

# Learning Actuator-Aware Spectral Submanifolds for Precise Control of Continuum Robots

Paul Leonard Wolff<sup>1,3</sup>, Hugo Buurmeijer<sup>1</sup>, Luis Pabon<sup>1</sup>, John Irvin Alora<sup>1</sup>, Mark Leone<sup>1</sup>,  
Roshan S. Kaundinya<sup>2</sup>, Amirhossein Kazempour<sup>3</sup>, Robert K. Katzschmann<sup>3</sup> and Marco Pavone<sup>1,4</sup>

**Abstract**—Continuum robots exhibit high-dimensional, nonlinear dynamics which are often coupled with their actuation mechanism. Spectral submanifold (SSM) reduction has emerged as a leading method for reducing high-dimensional nonlinear dynamical systems to low-dimensional invariant manifolds. Our proposed control-augmented SSMs (caSSMs) extend this methodology by explicitly incorporating control inputs into the state representation, enabling these models to capture nonlinear state-input couplings. Training these models relies solely on controlled decay trajectories of the actuator-augmented state, thereby removing the additional actuation-calibration step commonly needed by prior SSM-for-control methods. We learn a compact caSSM model for a tendon-driven trunk robot, enabling real-time control and reducing open-loop prediction error by 40% compared to existing methods. In closed-loop experiments with model predictive control (MPC), caSSM reduces tracking error by 52%, demonstrating improved performance against Koopman and SSM based MPC and practical deployability on hardware continuum robots.

## I. INTRODUCTION

Continuum robots are constructed from compliant materials and actuated through distributed deformation rather than rigid joints. These designs enable safe interaction in human-centered environments while expanding the design space for applications including manipulators, soft grippers, and morphing structures [1], [2]. However, the same compliance that yields these benefits complicates modeling and control: large deformations, geometric nonlinearities, and state-dependent couplings are common. Furthermore, continuum-mechanics ODEs discretize into high-dimensional systems, making real-time implementation of optimization-based methods like MPC computationally prohibitive [3], [4].

Model order reduction (MOR) seeks low-dimensional coordinates that capture a system’s essential behavior while enabling fast inference. Classical linear-subspace methods, such as proper orthogonal decomposition (POD) and principal component analysis (PCA), perform well when system dynamics align with a fixed global basis, but struggle with strong nonlinearities. Recent advances in nonlinear, data-driven reduction on low-dimensional spectral submanifolds (SSMs) have established them as a leading MOR technique

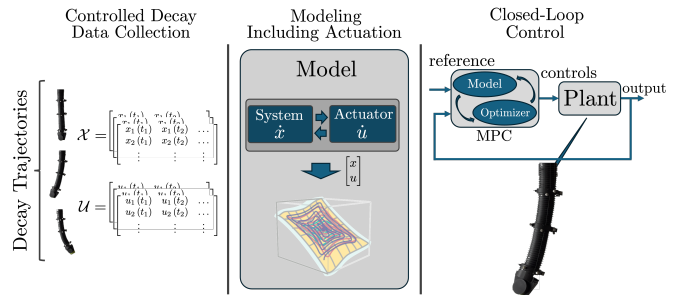


Fig. 1. Workflow for deploying SSMs on hardware for closed-loop control: (a) fully automated decay-based data collection, (b) actuator-aware manifold modeling from data, (c) real-time MPC rollout on hardware.

that preserves essential nonlinear dynamics while remaining computationally tractable for real-time control [5], [6], [7].

Equally important alongside MOR is recognizing that, in compliant robots, the actuator channel itself is dynamic. Tendons, pneumatic lines, valves, and transmissions introduce delays and bandwidth limits, while friction, stiction, saturation, and hysteresis cause inputs to enter the system dynamics in nonlinear, state-dependent ways [8]. Ignoring these effects leads to underfitting of transients and degraded predictive performance, whereas jointly modeling actuator and system dynamics yields a model that better captures the dynamics most relevant for control.

Based on these observations, our goal is to learn reduced-order models for high-dimensional systems and their actuators with dynamics of the kind:

$$\begin{aligned}\dot{x}(t) &= f(x(t), u(t)), \\ \dot{u}(t) &= \Lambda(u(t) - u_{\text{ref}}(t)),\end{aligned}\tag{1}$$

where  $x(t) \in \mathbb{R}^{n_f}$  denotes the full-order state,  $u(t) \in \mathbb{R}^m$  the control input applied through the actuators, and  $f: \mathbb{R}^{n_f} \times \mathbb{R}^m \rightarrow \mathbb{R}^{n_f}$  the nonlinear dynamics. We model actuator dynamics as a first-order ODE with the matrix  $\Lambda \in \mathbb{R}^{m \times m}$  capturing the dominant behavior of servo-driven tendon actuation. For pressure-driven fluidic actuators with higher-order or non-linear couplings, the state can be augmented with additional actuator-side states (e.g., pressure).  $u_{\text{ref}}(t)$  denotes the reference input commanded to the actuators.

To tackle this problem, we introduce *control-augmented Spectral Submanifolds* (caSSMs). Our contributions are:

- 1) **Framework:** caSSM embeds actuator states within the full state space to yield reduced-order models that capture coupled system–actuator dynamics.
- 2) **Theory:** derivation of external control contribution and linear actuator dynamics within caSSM.

<sup>1</sup> Autonomous Systems Lab, Stanford University, USA. Emails: {wolffp, hbuurmei, lpabon, jjalora, mleone, pavone}@stanford.edu.

<sup>2</sup> Institute of Mechanical Systems, ETH Zurich, Switzerland. Email: roshan.kaundinya@mavt.ethz.ch.

<sup>3</sup> Soft Robotics Lab, ETH Zurich, Switzerland. Emails: {wolffp, akazemi, rkk}@ethz.ch.

<sup>4</sup> NVIDIA Research, Santa Clara, USA.

- 3) **Pipeline:** an automated procedure that produces controller-ready models from decay trajectories (Fig. 1).
- 4) **Validation:** experiments on a tendon-driven trunk robot showing 40% lower prediction error and 52% higher tracking accuracy than state-of-the-art Koopman- and SSM-based MPC baselines.

**Outline:** We survey related work in Section II and provide relevant SSM preliminaries in Section III. Next, Section IV introduces the caSSM formulation and derives the inclusion of actuator dynamics. We detail the hardware data pipeline and learning procedure in Section V. Finally, we evaluate open-loop prediction and closed-loop MPC on a tendon-driven trunk robot in Section VI and conclude in Section VII.

## II. RELATED WORK

Modeling soft robots spans physics-based discretizations and data-driven surrogates. Finite element analysis (FEA) couples nonlinear hyperelasticity with time integration to handle complex geometries, contacts, and boundary conditions accurately. However, its state dimension and per-step solve costs are prohibitive for real-time, optimization-based control [9]. MPC with horizon  $T$ ,  $n$  states, and  $m$  inputs yields dense quadratic programs scaling as  $\mathcal{O}(T^3(n+m)^3)$ , reducible via block-tridiagonal structure to  $\mathcal{O}(T(n+m)^3)$  [3]. This cubic dependence on  $n$  motivates MOR.

Projection-based MOR methods, such as proper orthogonal decomposition (POD), reduce dynamics onto linear subspaces and are optimal for linear systems, yet their accuracy quickly deteriorates under strong nonlinearities. Trajectory piecewise linearization (TPWL) extends applicability through multiple local linearizations but introduces residual modeling errors [10], [11], [12]. To overcome such limitations, snapshot-based, data-driven MOR instead extracts low-order structure directly from trajectories: dynamic mode decomposition (DMD) fits linear operators to observed dynamics [13], while Koopman/Extended DMD (EDMD) lifts states into feature spaces to represent nonlinear dynamics linearly [14], [15]. In both cases, performance hinges on the choice of observables or basis functions, where richer feature sets capture more dynamics but inflate dimensionality and hinder real-time MPC integration.

To capture nonlinearities beyond linear-operator approximations, machine learning approaches like Gaussian processes or deep networks learn input–deformation maps and embed ODE residuals for consistency. But they may require large curated datasets and performance can degrade under distribution shift, e.g., when extrapolating to unseen trajectories, amplitudes, or operating conditions [16], [17]. Related hybrid models combine analytical models with learned corrections to improve data efficiency and extrapolation, but inherit modeling bias from the chosen physics prior [18].

As an alternative to purely data-driven regression, SSMs provide a principled, dynamics-based reduction: they are smooth, low-dimensional invariant manifolds that capture the dominant attracting dynamics with few coordinates, generalizing slow eigenspaces to nonlinear settings. They can be identified directly from trajectory data using libraries

like `SSMLearn` and `fastSSM` [5], [6]. However, while SSM-based approaches have also been extended to non-autonomous systems, including recent work demonstrating effective control after additional calibration, actuator effects and input dynamics are typically not modeled explicitly [7], [19], [20]. Instead, control is often introduced through a separate calibration step. In contrast, our formulation incorporates actuator dynamics directly into the model from the outset. It thereby captures coupled actuation–system interactions while avoiding the additional experiments that post hoc calibration would require.

## III. BACKGROUND ON SPECTRAL SUBMANIFOLDS

We first consider Eq. (1) without external control. Taking the Jacobian of the dynamics  $f$  with respect to the full state at the origin gives  $A := D_x f(0, 0) \in \mathbb{R}^{n_f \times n_f}$ , such that we can write the autonomous dynamics as

$$\dot{x}(t) = f_{\text{aut}}(x(t)) := Ax(t) + f_{\text{nl}}(x(t)), \quad (2)$$

where the nonlinear terms are grouped into  $f_{\text{nl}}$  and  $A$  is assumed to be Hurwitz. We then define the associated slow linear subspace as

$$E := E_{j_1} \oplus \cdots \oplus E_{j_n},$$

where each  $E_{j_k}$  is the real eigenspace associated with the eigenvalue  $\lambda_{j_k}$  of  $A$ . The direct sum of eigenspaces is chosen so that

$$\min_{\lambda \in \Sigma_E} \text{Re}(\lambda) > \max_{\lambda \in \Sigma_{\text{out}}} \text{Re}(\lambda),$$

with  $\Sigma_E$  being the set of eigenvalues corresponding to  $E$ , and  $\Sigma_{\text{out}}$  being the complementary set of faster eigenvalues. From here, we define (adopted from [5]):

*Definition 1 (Autonomous SSM):* The *autonomous SSM*,  $W(E)$ , is the smoothest nonlinear continuation of  $E$  that remains invariant under the autonomous dynamics, with  $\dim W(E) = \dim(E) =: n$ .

For the dynamics in (2), this invariance is formally expressed as

$$x(0) \in W(E) \implies x(t) \in W(E) \quad \forall t \in \mathbb{R}^+. \quad (3)$$

Because trajectories remain confined to  $W(E)$ , the manifold can be parametrized by reduced coordinates with corresponding mappings to and from the ambient state space:

$$z = v(x), \quad (4)$$

$$x = w(z), \quad (5)$$

where  $z \in \mathbb{R}^n$  are reduced coordinates and  $x \in \mathbb{R}^{n_f}$  are the full coordinates, with invertibility on the manifold:

$$x = (w \circ v)(x), \quad z = (v \circ w)(z).$$

SSMs thus provide nonlinear, low-dimensional coordinates in which the dominant system behavior can be modeled efficiently.

To extend this approach to controlled systems, most SSM-based models assume control-affine dynamics [7], [19], [20]

$$\dot{x}(t) = f_{\text{aut}}(x(t)) + B(x(t))u(t), \quad (6)$$

which we will relax by focusing on the generic form of (1).

### A. Fitting SSMs from Observational Data

For high-dimensional systems, such as soft robots, one generally cannot observe their full state. Therefore, we instead construct the SSM in the observable space of measurements, which can be made sufficiently large by adding time-delay embeddings. The fitting of SSMs from observational data generally involves three steps: 1) fitting the chart (4) and parameterization maps (5), 2) finding the on-manifold dynamics, and 3) calibrating the dynamics to include the effect of control inputs. The first two steps use data obtained by launching the robot from various initial conditions and recording its autonomous decay trajectories. The slow spectral subspace,  $E$ , can directly be found from these observations by performing a PCA and selecting the  $n$  most dominant components. This defines the chart map as

$$z = v(x_o) := V^\top x_o, \quad (7)$$

where  $x_o \in \mathbb{R}^o$  is the observed state vector and the columns of  $V \in \mathbb{R}^{o \times n}$  span  $E$ . Assuming analyticity of  $f_{\text{nl}}$ ,  $W(E)$  admits a convergent Taylor expansion over  $E$ , motivating a finite-order polynomial feature map for the parameterization [5], i.e.,

$$x_o = w(z) := Vz + W_{\text{nl}}z^{2:n_w}, \quad (8)$$

where  $z^{2:n_w}$  collects all monomials of  $z$  from order 2 up to  $n_w$ , and the coefficient matrix  $W_{\text{nl}}$  is obtained via polynomial regression. The reduced dynamics similarly is

$$\dot{z} = R_0z + R_{\text{nl}}z^{2:n_r} + B_r u, \quad (9)$$

where  $z^{2:n_r}$  denotes monomials of order 2 through  $n_r$ ,  $R_0$  and  $R_{\text{nl}}$  are fitted on autonomous data via polynomial regression, and  $B_r$  on controlled data via least-squares (see [7]). We will refer to these resulting models as the *origin SSMs* (oS<sub>SMs</sub>) to differentiate from the proposed approach.

### B. Assumptions and Practical Limitations

SSM-based reductions adopt the following modeling assumptions: (i) manifold-shaping dynamics arise solely from the system's intrinsic behavior, neglecting actuator interactions [5]; (ii) actuator dynamics are much faster than the system, enabling near-instant setpoint tracking [7]; (iii) the control influence on dynamics is affine (linear input contribution) as in (6).

In robotic systems, these assumptions are often violated [8], [21]. In particular, slow actuator dynamics can be non-negligible and therefore affect the choice of modal basis used for the reduced-coordinate representation. To assess when this matters, Fig. 2 gives a practical bandwidth diagnostic through the relative location of actuator modes (blue) and robot modes (green): when the actuator modes are well separated on the fast side, the actuators can be approximated as ideal. If the spectra overlap, actuator dynamics should be modeled explicitly. When actuator modes are slower than the robot modes of interest, control authority becomes insufficient for reliable tracking. Consequently, omitting actuator modes can yield reduced coordinates that miss relevant physical behavior and lead to significant model mismatch.

These limitations also affect data collection. Slow actuators may be incapable of providing step inputs to generate decay trajectories that trace out the manifold, necessitating manual displacements to create nonzero initial conditions followed by instant release, thereby limiting practicality. Finally, restricting control effects to affine terms impedes capturing nonlinear actuation characteristics (e.g., friction) common in tendon-driven soft robots.

### IV. CONTROL-AUGMENTED SSMs

We introduce *control-augmented SSMs* (caSSMs), a reduced modeling framework that explicitly embeds actuator dynamics into the SSM formulation.

To capture actuator behavior, we augment the state with the control input:

$$\tilde{x} = \begin{bmatrix} x \\ u \end{bmatrix} \in \mathbb{R}^{n_f+m}.$$

By linearizing the state dynamics from (1) around the origin, we obtain the (open-loop) system–actuator model

$$\dot{\tilde{x}} = \begin{bmatrix} \dot{x} \\ \dot{u} \end{bmatrix} = \underbrace{\begin{bmatrix} A & A_u \\ 0 & \Lambda \end{bmatrix}}_{\text{linear}} \begin{bmatrix} x \\ u \end{bmatrix} + \underbrace{\begin{bmatrix} g(x, u) \\ 0 \end{bmatrix}}_{\text{nonlinear}} - \underbrace{\begin{bmatrix} 0 \\ \Lambda u_{\text{ref}}(t) \end{bmatrix}}_{\text{setpoint drive}}, \quad (10)$$

with  $\Lambda$  being Hurwitz. We thereby replace the control-affine assumption by a general coupling via a linear cross term  $A_u \in \mathbb{R}^{n_f \times m}$  and a nonlinear term  $g : \mathbb{R}^{n_f} \times \mathbb{R}^m \rightarrow \mathbb{R}^{n_f}$ , thus allowing nonlinear input dependence. When the coupling mechanism (e.g. tendons) introduces identifiable slow modes on time scales relevant to the robot dynamics, the corresponding internal states should be explicitly included in  $x$ .  $u_{\text{ref}}(t)$  encodes the desired control input setpoints, which we prescribe and optimize over, while  $u$  is the actual input applied to the system. The Hurwitz condition on  $\Lambda$  enforces dissipative actuator dynamics whose eigenvalues set an actuation rate limit. The spectrum of the linearized augmented system is the union of those of  $A$  and  $\Lambda$ .

To describe the caSSM, we use a graph-style parameterization over reduced coordinates  $z \in \mathbb{R}^n$ :

$$z = V^\top \tilde{x}, \quad V^\top V = I, \quad (11)$$

$$\tilde{x} = w(z) = Vz + w_{\text{nl}}(z), \quad (12)$$

$$\dot{z} = r(z, t) = R_0z + r_{\text{nl}}(z) + r_{\text{ref}}(t) \quad (13)$$

where  $r_{\text{ref}}(t)$  captures the effect of external control inputs to the reduced system. Similar to [22], we choose the SSM parameterization specified in (12) to be independent of time. Moreover, an orthogonal projection is chosen in accordance with [7], while recent work has explored generalizing to oblique projections [20], [23].



Fig. 2. Eigenvalue layout: slow system modes are shown in green and actuator modes in blue. The region left of  $\gamma \ll 0$  indicates rapidly decaying modes that can safely be neglected. Spectral overlap of modes signals that neglecting actuator dynamics can induce significant modeling errors.

### A. Derivation of External Control Contribution

We derive how the reference inputs appear on the SSM by identifying the reduced input term  $r_{\text{ref}}(t)$  that preserves manifold invariance under the augmented dynamics (10). Throughout, we adopt the SSM assumption that the reference input remains small and bounded,  $|u_{\text{ref}}(t)| < \delta$ , such that the manifold is approximately preserved under external forcing.

*Theorem 1:* Under the augmented dynamics (10) and the graph parameterization (11)–(13), the reduced input term that preserves manifold invariance is

$$r_{\text{ref}}(t) = V^\top \begin{bmatrix} 0 \\ -\Lambda u_{\text{ref}}(t) \end{bmatrix}. \quad (14)$$

*Proof:* On the manifold,  $\tilde{x} = w(z)$  by (12). Differentiating and substituting (13) gives

$$\begin{aligned} \dot{\tilde{x}} &= V\dot{z} + D_z w_{\text{nl}}(z)\dot{z} \\ &= [V + D_z w_{\text{nl}}(z)](R_0 z + r_{\text{nl}}(z) + r_{\text{ref}}(t)). \end{aligned} \quad (15)$$

The same derivative follows from the full augmented dynamics (10) while plugging in (12), giving

$$\begin{aligned} \dot{\tilde{x}} &= \begin{bmatrix} A & A_u \\ 0 & \Lambda \end{bmatrix} (Vz + w_{\text{nl}}(z)) \\ &\quad + \begin{bmatrix} g(x, u) \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ \Lambda u_{\text{ref}}(t) \end{bmatrix}. \end{aligned} \quad (16)$$

Equating (15) and (16) and isolating the time-dependent terms, we obtain

$$D_z w(z) r_{\text{ref}}(t) = \begin{bmatrix} 0 \\ -\Lambda u_{\text{ref}}(t) \end{bmatrix}. \quad (17)$$

At this point, recall that the manifold parameterization is smooth and can be written as  $w(z) = Vz + w_{\text{nl}}(z)$  (12). Its Jacobian therefore has the form

$$D_z w(z) = V + D_z w_{\text{nl}}(z),$$

where  $D_z w_{\text{nl}}(0) = 0$  and  $\|D_z w_{\text{nl}}(z)\| = \mathcal{O}(\|z\|)$ .<sup>1</sup>

Close to the equilibrium, the nonlinear contribution is higher order in  $z$  such that we approximate

$$V r_{\text{ref}}(t) = \begin{bmatrix} 0 \\ -\Lambda u_{\text{ref}}(t) \end{bmatrix}. \quad (18)$$

Left-multiplying by  $V^\top$  and using (11) yields (14). ■

This result is consistent with a more general statement found in [22], Section 4.2.

### B. Obtaining Linear Actuator Dynamics

Inspecting the derived term (14) shows its dependency on the  $\Lambda$ -matrix, the linear actuator dynamics. However, this is an unknown quantity and shall be identified. To do so, we examine the linear terms in  $z$  when equating (15) with (16), yielding the linear invariance condition

$$V R_0 = \begin{bmatrix} A & A_u \\ 0 & \Lambda \end{bmatrix} V. \quad (19)$$

<sup>1</sup>Tangency gives  $D_z w(0) = V$  hence  $D_z w_{\text{nl}}(0) = 0$ ; analyticity yields  $w_{\text{nl}}(z) = \mathcal{O}(\|z\|^2)$  and thus  $D_z w_{\text{nl}}(z) = \mathcal{O}(\|z\|)$  [5].

Practically, the matrix  $R_0 \in \mathbb{R}^{n \times n}$  is approximated by taking the Jacobian of the identified reduced dynamics, with the caSSM dimension  $n$  chosen to include the modes associated with actuation, as illustrated in Fig. 2.

Let  $E_u \in \mathbb{R}^{m \times (n_f + m)}$  be defined as  $E_u = [0_{m \times n_f} \ I_m]$ , which extracts the actuator components. We can decompose  $V = [V_x^\top, V_u^\top]^\top \in \mathbb{R}^{(n_f + m) \times n}$ , such that left-multiplying (19) with  $E_u$  gives

$$\begin{aligned} E_u V R_0 &= E_u \begin{bmatrix} A & A_u \\ 0 & \Lambda \end{bmatrix} V, \\ V_u R_0 &= \Lambda V_u. \end{aligned}$$

Assuming  $\text{rank}(V_u) = m$ , due to appropriate tuning, the linear actuator dynamics follows from least squares as

$$\Lambda = (V_u R_0) V_u^\dagger = (E_u V R_0) (E_u V)^\dagger, \quad (20)$$

where  $(\cdot)^\dagger$  denotes the Moore–Penrose pseudoinverse. In essence,  $V$  captures dominant robot-actuator motion patterns. Projecting them onto the actuator coordinates and fitting the best linear map yields the actuator dynamics matrix  $\Lambda$  consistent with manifold invariance.

Alternatively, future work could treat  $\Lambda$  as a design parameter with additional degrees of freedom: (i) placing real eigenvalues near measured actuator rates for realism; (ii) choosing  $\text{Re}(\lambda(\Lambda))$  slightly more negative than dominant system modes to separate time scales and aid identification; (iii) introducing mild oscillations via complex-conjugate pairs to excite weakly damped modes; (iv) cautiously aligning actuator eigenfrequencies with systems resonances to amplify informative responses. This design choice trades realism, numerical conditioning, and excitation richness for identification and control.

### C. Augmenting with Actuator-Side Feedback

Besides data-driven identification, we can incorporate state feedback into the actuator dynamics via the linear gain  $H$  or the nonlinear term  $h(x, u)$ . For SSM existence in Section IV, we considered the stable case ( $A$  Hurwitz). Now, we analyze actuator-assisted stabilization when the system's linearization is not Hurwitz, denoted by  $A^u$ , leading to the augmented state dynamics

$$\dot{\tilde{x}} = \begin{bmatrix} A^u & A_u \\ H & \Lambda \end{bmatrix} \tilde{x} + \begin{bmatrix} g(x, u) \\ h(x, u) \end{bmatrix} - \begin{bmatrix} 0 \\ \Lambda u_{\text{ref}}(t) \end{bmatrix}. \quad (21)$$

Linearizing the actuator channel by setting  $h(x, u) \equiv 0$  and  $u_{\text{ref}} \equiv 0$  shows how the feedback acts:

$$\dot{u} = Hx + \Lambda u \Rightarrow$$

$$U(s) = (sI - \Lambda)^{-1} H X(s) = K(s) X(s).$$

This yields a dynamic feedback filter with low-pass behavior, where  $K(s) \approx -\Lambda^{-1} H$  for  $s \approx 0$  and  $K(s) \rightarrow 0$  as  $|s| \rightarrow \infty$ . Hence, the feedback strongly influences slow modes while attenuating high-frequency content, in line with the actuator bandwidth.

Let

$$\tilde{A} := \begin{bmatrix} A^u & A_u \\ H & \Lambda \end{bmatrix}.$$

be the linearized augmented system around an unstable equilibrium of  $A^u$ . If  $(A^u, A_u)$  is stabilizable,  $\Lambda$  is Hurwitz, and the actuator can influence the unstable modes of  $A^u$ , then there exists  $H$  such that  $\tilde{A}$  is Hurwitz. In practice, the actuator bandwidth must cover these unstable or slow modes: the eigenvalues of  $\Lambda$  should lie sufficiently left of those modes so that  $K(s) \approx -\Lambda^{-1}H$  holds over the relevant frequency range, otherwise feedback authority is insufficient.

## V. DEPLOYING CASSMS ON HARDWARE

This section details the practical pipeline to identify and employ a control-augmented SSM (caSSM) on hardware: (i) the data collection procedure; (ii) feature-map selection; (iii) control reference injection; and (iv) the reduced-order MPC formulation.

### A. Data Collection

To capture the dominant nonlinear behavior and system-actuator coupling, we collect trajectories that (a) cover a broad portion of the operating range and (b) excite transverse modes through orthogonal perturbations that generate oscillatory transients. Such experiments increase signal variance, reveal mode interactions, and make actuator-driven effects more identifiable. Raw measurements are denoised offline using a Kalman filter with Rauch–Tung–Striebel (RTS) smoothing, which stabilizes learning without inducing phase shifts. During online control, we use raw measurements, as non-causal filtering would reintroduce phase shifts and the controllers proved robust enough to handle the measurement noise.

Since the full state of a continuum robot cannot be measured, we restrict the model to measurable observables. We denote the observable augmented state by  $\tilde{x}_o = [x_o^\top, u^\top]^\top \in \mathbb{R}^{L(o+m)}$ , which may include up to  $L-1$  delay embeddings of the observed control-augmented state. For an appropriate choice of  $L$  or  $o$ , we refer to [6], [7].

### B. Feature Maps for $w_{nl}$ and $r_{nl}$

The oSSM models use polynomial feature maps as seen in (8) and (9). However, with finite data and potential overfitting, this can cause divergence in certain data regimes. Similar to [24], who used radial basis functions to train nonlinear SSM feature maps for capturing chaotic attractors, we revisited this design choice and evaluated three families of feature maps: (1) low-degree polynomial monomials, (2) random Fourier features (RFFs) approximating an RBF kernel as described in [25], and (3) small neural networks. In an ablation study, RFF-inspired cosine features achieved the highest in-domain accuracy and the least out-of-domain divergence, resulting in the best overall performance. Therefore, we use the cosine feature map motivated by the random Fourier features (RBF approximation) for all fitted nonlinear mappings.

### C. Control Reference Vector

As for the observed augmented state, the control reference vector as defined in (14) shall be embedded. For embedding

depth  $L$ , the controller maintains a buffer of the most recent inputs. The contribution per embedding is

$$\tilde{J} = \begin{bmatrix} 0_{o \times o} & 0_{o \times m} \\ 0_{m \times o} & -\Lambda \end{bmatrix}, \quad \tilde{J}_L = I_L \otimes \tilde{J} \in \mathbb{R}^{(L(o+m))^2},$$

which places zeros on the system rows and  $-\Lambda u_{\text{ref}}(\cdot)$  on the actuator rows.

With this definition, the control reference vector in reduced coordinates is

$$r_{\text{ref}}(t) = V^\top \tilde{J}_L \tilde{x}_o(t). \quad (22)$$

At runtime, the controller optimizes the embedded  $u_{\text{ref}}$ , causing the MPC complexity to grow linearly with the number of delay embeddings.

As an illustration, for embedding depth  $L = 2$  the expression becomes

$$r_{\text{ref}} = V^\top \begin{bmatrix} 0_o \\ -\Lambda u_{\text{ref}}(t) \\ 0_o \\ -\Lambda u_{\text{ref}}(t-dt) \end{bmatrix}, \quad (23)$$

showing how the current input and its one-step delay appear in the embedded formulation.

For faster solves, we approximate the input history by repeating the current input across all lags. With the same block structure  $\tilde{J}_L$ , the control reference vector becomes

$$r_{\text{ref}}(t) \approx V^\top \tilde{J}_L (1_L \otimes u_{\text{ref}}(t)). \quad (24)$$

This keeps the number of decision variables independent of  $L$  (only the current input is optimized), yielding much faster MPC solves with only minor accuracy loss. For example, with  $L = 2$  the approximation expands to

$$r_{\text{ref, approx}} = V^\top \begin{bmatrix} 0_o \\ -\Lambda u_{\text{ref}}(t) \\ 0_o \\ -\Lambda u_{\text{ref}}(t) \end{bmatrix}. \quad (25)$$

### D. Low-Dimensional MPC Formulation

For closed-loop control on hardware, GuSTO-based MPC [26] is used and implemented in JAX to allow for just-in-time compilation. The exact low-dimensional MPC formulation is given by

$$\begin{aligned} \min_{u_{\text{ref}}(\cdot)} \quad & \|\delta y_{\text{perf}}(t_f)\|_{Q_f}^2 + \int_{t_0}^{t_f} (\|\delta y_{\text{perf}}(t)\|_Q^2 + \|\Delta u(t)\|_{R_\Delta}^2) dt, \\ \text{s.t.} \quad & z(t_0) = V^\top (\tilde{x}_o(t_0)), \\ & \dot{z}(t) = r(z(t)) + r_{\text{ref}}(t), \\ & y_{\text{perf}}(t) = C w(z(t)), \\ & y_{\text{perf}}(t) \in \mathcal{Y}, \quad u(t) \in \mathcal{U}, \end{aligned} \quad (26)$$

where  $y_{\text{perf}} \in \mathbb{R}^p$  is the performance state, which is a subset of the observed states, i.e.  $p < o$ . The cost penalizes deviations of the performance trajectory  $\delta y_{\text{perf}}(t) := y_{\text{perf}}(t) - y_{\text{ref}}(t)$  as well as variations in the control input  $\Delta u(t) := u(t) - u(t-dt)$ . This formulation emphasizes computational tractability by restricting the optimization to the reduced coordinates while still enforcing feasibility in the full space.

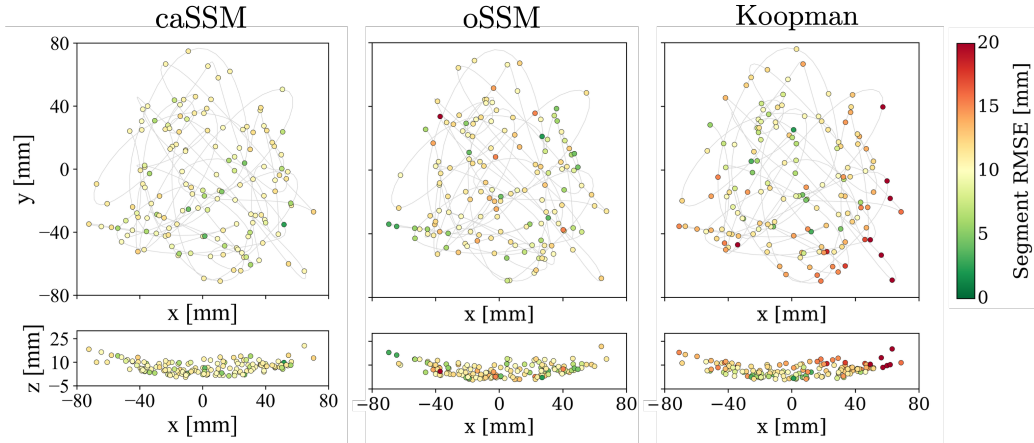


Fig. 3. RMSE of predictive results across three models for the same ground truth trajectory. caSSM shows lowest RMSE across all segments, while Koopman diverges near training data boundaries.

## VI. EXPERIMENTS

This section presents our experimental evaluation. We first detail the experimental setup and then demonstrate caSSM effectiveness in two experiments: open-loop predictive performance and closed-loop reference trajectory tracking. In both settings we benchmark against two baselines.

### A. Experimental Setup

Experiments are performed on a tendon-driven continuum “trunk” robot actuated by six servos in antagonistic tendon pairs, grouped into two independent inputs ( $u \in \mathbb{R}^2$ ), with 3D position measurements coming from motion capture tracking of reflective markers placed on three segments along the soft body ( $x_o \in \mathbb{R}^9$ ). We do not measure additional tendon-internal states explicitly, as no slow tendon-related modes are observed in the measured response. Residual tendon effects are therefore captured implicitly through the nonlinear coupling term  $g(x, u)$ . Relevant nonidealities include actuation dead bands near the origin position, trial-to-trial variability

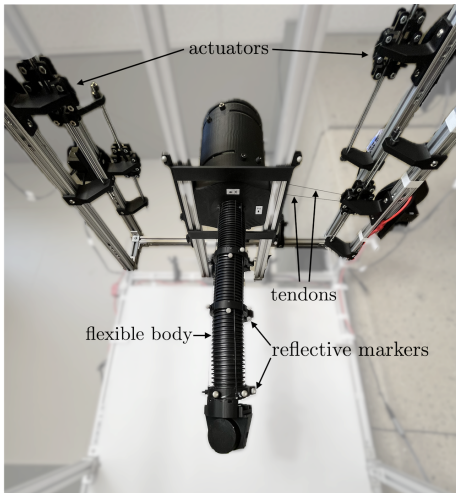


Fig. 4. The tendon-driven trunk hardware system, shown in its rest position. The flexible body has a length of 31 cm. Tendons extend from the actuators to rigid clamps attached to the flexible body. Observations were captured using an OptiTrack motion capture system.

(uncertainty bands), and hysteresis (path dependency). The system is shown in Fig. 4.

To evaluate the proposed caSSM method, we compare three reduced-order models: (i) **caSSM**, a 7D manifold including actuator modes with  $\Lambda$  recovered via (20) and two delay embeddings; (ii) **oSSM**, a 5D polynomial manifold (degree 2) with a separately calibrated affine control map following [20]; and (iii) **Koopman**, using polynomial lifting (degree 2) and one delay, yielding a 120-dimensional lifted linear model following [14]. All models are trained on data sampled at 50 Hz, and their hyperparameters are tuned for best performance on a shared validation set.<sup>2</sup>

### B. Open-Loop Predictive Performance

We assess the models’ predictive capabilities independent of controller feedback using a finite-horizon *random-actuation* protocol. A 15 s random-input sequence is partitioned into 150 segments of 5 timesteps each (0.1 s). For each segment, models are initialized at the measured start state and simulated in open loop with RK4 at  $\Delta t = 0.02$  s. We compute the RMSE between predicted and measured states, report the distribution across segments in Fig. 3, and summarize the segment-averaged RMSE values in Table I.

The caSSM model achieved the lowest average RMSE of 2.8 mm across random segments, maintaining accuracy under fast transients and varied initializations. For large-amplitude circular trajectories, accuracy decreases—likely due to hysteresis not captured by decay-based training—yet the model remains more numerically stable than the baselines, aided by the chosen feature map.

TABLE I  
AVERAGE RMSE [MM] FOR OPEN-LOOP PREDICTIONS.

Model	caSSM	oSSM	Koopman
RMSE	<b>2.8</b>	4.6	7.3

<sup>2</sup>Baseline results for oSSM models with RFF features are omitted, as their performance within the training domain was comparable to that of polynomial features.

The oSSM model showed moderate performance (average RMSE of 4.6 mm), with accuracy deteriorating primarily during rapid state transitions. For trajectories with significant hysteresis effects (particularly large-amplitude circular motions), oSSM occasionally outperformed caSSM, likely because its calibration step partially compensates for hysteresis phenomena not explicitly modeled.

The Koopman baseline exhibited the highest average RMSE of 7.3 mm and the least consistent performance. Its predictions degraded near the boundaries of the training data, with numerous segments showing numerical instability. This suggests that the Koopman model overfits to measurement noise rather than capturing the underlying dynamics.

### Key Takeaways

- (i) Embedding actuator dynamics (caSSM) significantly improves both prediction quality and numerical stability on hardware.
- (ii) The decay-only training introduces a modeling bias for hysteresis effects; caSSM is most robust, while oSSM trades modeling bias for reduced robustness.
- (iii) Koopman lifting exhibits reliable performance in-distribution and for small-amplitude motions, but fails to generalize robustly to larger amplitudes.

### C. Closed-Loop Tracking on Hardware

We compare caSSM against oSSM and Koopman in a closed loop on the trunk robot. All controllers use the MPC formulation in (26) and performance is reported as RMSE of end-effector tracking.

Both SSM models achieved the best results using the same MPC parameters: horizon  $N=10$  at 50 Hz model rate (actuation 25 Hz), state costs  $Q = \text{diag}(7, 7, 0)$ , terminal costs  $Q_N = \text{diag}(20, 20, 0)$ , input-change penalty  $R_\Delta = \text{diag}(0.16, 0.16)$ , and no absolute input penalty. State and input constraints are imposed uniformly. The Koopman model with lifted dimension  $n=120$  achieved its best results in closed loop by using a shorter horizon ( $N=2$ ), which maintained predictive stability while meeting real-time solve-time constraints. Its tuned costs are  $Q = \text{diag}(1, 1, 0)$ ,  $Q_N = \text{diag}(20, 20, 0)$ , and  $R_{\Delta u} = \text{diag}(0.05, 0.05)$ .

We compare tracking performance on circles and figure-eights of different radii at nominal (0.5 rad/s) and doubled (1 rad/s, denoted "fast") angular velocity. Qualitative results for selected trajectories are shown in Fig. 5, and RMSE values are summarized in Table II.

Among the compared models, caSSM achieves the highest accuracy and robustness, reaching 4.6 mm RMSE on a 5 cm

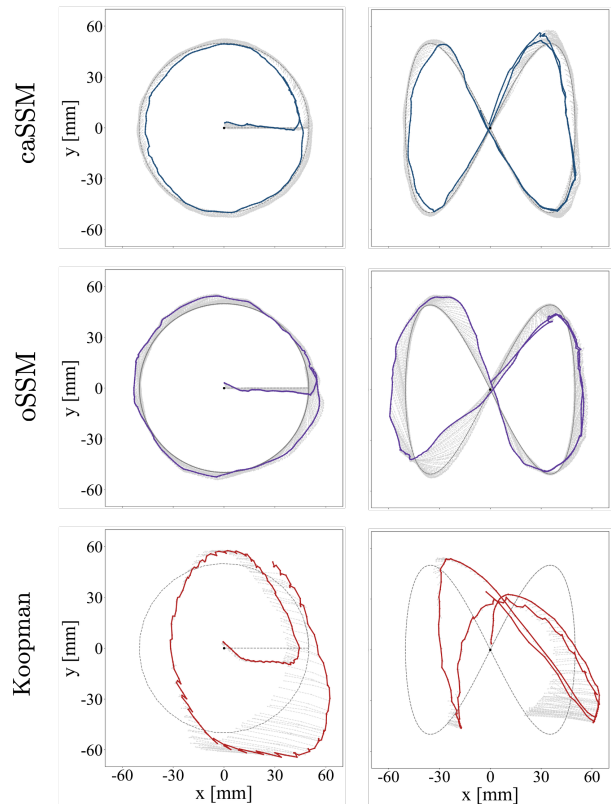


Fig. 5. Closed-loop MPC performance on circular and figure-eight trajectories. Reference paths are shown in gray, with MPC rollouts in lighter gray. The Koopman model fails to track the reference, oSSM struggles on sharp curves while only caSSM tracks consistently across both shapes.

circle (vs. 9.89 mm for oSSM and 20.40 mm for Koopman) and remaining stable at larger radii and higher speeds. At higher amplitudes, a small steady offset appears, consistent with hysteresis effects discussed in Section VI-A, yet tracking remains symmetric and reliable. In contrast, oSSM tracks smaller and slower trajectories but degrades as amplitude and speed increase. While the separate calibration step in oSSM can partly mitigate hysteresis effects, its overall robustness remains below that of caSSM.

Despite solving a *linear* optimal control problem, the Koopman baseline has the longest average MPC solve time of 64 ms and the fastest prediction divergence, which together require shorter horizons for robustness and real-time feasibility. As a result, it can follow only small motions and yields the highest RMSE among successful runs. In contrast, caSSM's added modeling capacity increases solve time only moderately, from 14 ms for oSSM to 18 ms, while significantly improving tracking performance.

TABLE II

AVERAGE RMSE [MM] FOR CLOSED-LOOP TRAJECTORIES.

Model	caSSM	oSSM	Koopman
Circle 5 cm	<b>4.60</b>	9.89	20.40
Circle 5 cm – "fast"	<b>5.30</b>	13.44	<i>Diverged</i>
Circle 8 cm	<b>9.13</b>	19.80	<i>Diverged</i>
Circle 10 cm	<b>13.36</b>	<i>Diverged</i>	<i>Diverged</i>
Figure-eight	<b>4.40</b>	8.64	26.72
Figure-eight – "fast"	<b>8.12</b>	15.10	<i>Diverged</i>

### Key Takeaways

- (i) Explicit actuator dynamics in caSSM expand the controllable workspace, improve tracking accuracy.
- (ii) caSSM enables more stable execution across diverse trajectories by providing more robust predictions.
- (iii) Compared to Koopman, SSM-based models sustain real-time MPC with longer horizons, yielding improved closed-loop stability.

## VII. CONCLUSION AND OUTLOOK

By explicitly embedding actuator dynamics, our caSSMs close key modeling gaps in continuum robots and enable real-time control with minimal manual calibration.

Compared to regular SSM and Koopman baselines, caSSM delivers higher-fidelity open-loop predictions and superior closed-loop tracking with real-time MPC on a continuum robot hardware platform. Practically, the pipeline streamlines data collection to a single automated step (decay trajectories), removes separate control calibration, and maintains tractable solve times via a simplified reference-injection scheme. Together, these elements make caSSMs a practical and scalable tool for modeling and control of continuum robots.

Our approach exhibits two notable limitations. First, training exclusively on controlled decay trajectories can introduce residual hysteresis bias, which manifests in our system as small steady-state offsets in predictions of higher-amplitude trajectories. Despite these offsets, the closed-loop system under MPC maintains both stability and symmetry properties. Second, the method's effectiveness depends on accurate actuator mode identification; improperly identified actuator dynamics can degrade closed-loop performance beyond what would be observed without explicit actuator modeling.

This work motivates several extensions that we plan to investigate in future studies: (i) **Adiabatic extensions:** interpolating local models around forced steady states (as done in [19]) may improve setpoint projection in regions where global decays underrepresent effects like hysteresis. (ii) **Design-space tuning:** principled selection of actuator spectra  $\Lambda$  (and its excitation profile) to balance identifiability, numerical conditioning, and control authority. (iii) **Controller-side coupling:** structured design of  $H$  and  $h(x, u)$  for bandwidth-aware stabilization of unstable modes, including actuation-awareness.

## ACKNOWLEDGMENTS

This work was partially supported by Toyota Motor Engineering & Manufacturing North America (TEMA). The views expressed in this article are solely those of the authors and do not necessarily reflect those of the supporting entity. We thank Patrick Benito Eberhard for valuable discussions on hardware evaluations.

## REFERENCES

- [1] D. Trivedi, C. D. Rahn, W. M. Kier, and I. D. Walker, "Soft robotics: Biological inspiration, state of the art, and future research," *Applied Bionics and Biomechanics*, vol. 5, no. 3, pp. 99–117, 2008.
- [2] R. K. Katzschmann, J. DelPreto, R. MacCurdy, and D. Rus, "Exploration of underwater life with an acoustically controlled soft robotic fish," *Science Robotics*, vol. 3, no. 16, p. eaar3449, 2018.
- [3] G. Rao, M. J. Wright, Stephen and B. Rawlings, J. "Application of interior-point methods to model predictive control," *Journal of Optimization Theory and Applications*, vol. 99, no. 3, pp. 731–757, 1998.
- [4] A. Malyshev, R. Quirynen, and A. Knyazev, "Preconditioned krylov iterations and condensing in interior point mpc method," Mitsubishi Electric Research Laboratories, Cambridge, MA, USA, Technical Report TR2018-121, Aug. 25 2018.
- [5] G. Haller and S. Ponsioen, "Nonlinear normal modes and spectral submanifolds: existence, uniqueness and use in model reduction," *Nonlinear Dynamics*, vol. 86, pp. 1493–1534, Nov 2016.
- [6] M. Cenedese, J. Ax as, H. Yang, M. Eriten, and G. Haller, "Data-driven nonlinear model reduction to spectral submanifolds in mechanical systems," *Philosophical Transactions of the Royal Society A*, vol. 380, no. 2229, p. 20210194, 2022.
- [7] J. I. Alora, M. Cenedese, G. Haller, and M. Pavone, "Discovering dominant dynamics for nonlinear continuum robot control," *npj Robotics*, vol. 3, no. 1, p. 5, 2025.
- [8] T. N. Do, T. Tjahjowidodo, M. W. S. Lau, and S. J. Phee, "Performance control of tendon-driven endoscopic surgical robots with friction and hysteresis," *arXiv preprint*, 2017, arXiv:1702.02063.
- [9] C. Duriez, "Control of elastic soft robots based on real-time finite element method," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 3982–3987.
- [10] K. Kunisch and S. Volkwein, "Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics," *SIAM Journal on Numerical Analysis*, vol. 40, no. 2, pp. 492–515, 2002.
- [11] M. Rewienski and J. White, "A trajectory piecewiselinear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 2, pp. 155–170, Feb. 2003.
- [12] S. Tonkens, J. Lorenzetti, and M. Pavone, "Soft robot optimal control via reduced order finite element models," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 12 010–12 016.
- [13] P. J. Schmid, "Dynamic mode decomposition of numerical and experimental data," *Journal of Fluid Mechanics*, vol. 656, pp. 5–28, 2010.
- [14] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley, "A data-driven approximation of the koopman operator: Extending dynamic mode decomposition," *Journal of Nonlinear Science*, vol. 25, no. 6, pp. 1307–1346, 2015.
- [15] D. Bruder, C. D. Remy, and R. Vasudevan, "Nonlinear system identification of soft robot dynamics using koopman operator theory," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6244–6250.
- [16] M. T. Gillespie, C. M. Best, E. C. Townsend, D. Wingate, and M. D. Killpack, "Learning nonlinear dynamic models of soft robots for model predictive control with neural networks," in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, 2018, pp. 39–45.
- [17] Y. Yang and C. Duriez, "Real-time simulation of soft robots with physics-informed neural networks," *Soft Robotics*, vol. 8, no. 2, pp. 242–254, 2021.
- [18] R. F. Reinhart, Z. Shareef, and J. J. Steil, "Hybrid analytical and data-driven modeling for feed-forward robot control," *Sensors*, vol. 17, no. 2, p. 311, 2017.
- [19] R. S. Kaundinya, J. I. Alora, J. G. Matt, L. A. Pabon, M. Pavone, and G. Haller, "Data-driven soft robot control via adiabatic spectral submanifolds," *arXiv preprint*, 2025, arXiv:2503.10919.
- [20] H. Buurmeijer, L. A. Pabon, J. I. Alora, R. S. Kaundinya, G. Haller, and M. Pavone, "Taming high-dimensional dynamics: Learning optimal projections onto spectral submanifolds," in *Proc. IEEE Conf. on Decision and Control*, 2025.
- [21] Z. Ma and D. Sameoto, "A review of electrically driven soft actuators for soft robotics," *Micromachines*, vol. 13, no. 11, p. 1881, 2022.
- [22] G. Haller, *Modeling Nonlinear Dynamics from Equations and Data—with Applications to Solids, Fluids, and Controls*. Philadelphia, PA: SIAM, 2025.
- [23] L. Bettini, B. Kasz as, B. Zybach, J. Dual, and G. Haller, "Data-driven nonlinear model reduction to spectral submanifolds via oblique projection," *Chaos*, vol. 35, no. 4, 2025.
- [24] B. Kasz as and G. Haller, "Joint reduced model for the laminar and chaotic attractors in plane couette flow," *arXiv preprint*, 2025, arXiv:2505.05881.
- [25] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in Neural Information Processing Systems*, vol. 20, 2007.
- [26] R. Bonalli, A. Cauligi, A. Bylard, and M. Pavone, "Gusto: Guaranteed sequential trajectory optimization via sequential convex programming," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6741–6747.