9

# Discovering dominant dynamics for nonlinear continuum robot control

Check for updates

John Irvin Alora<sup>1</sup> 🖂, Mattia Cenedese<sup>2</sup>, George Haller<sup>2</sup> & Marco Pavone<sup>1</sup>

Continuum robots, which emulate biological organisms' dexterity and flexibility, hold transformative potential for terrestrial and extraterrestrial applications. While such capabilities present significant modeling and control challenges, these robots' highly dissipative nature allows their behavior to be explained by low-dimensional, dominant dynamics. Despite extensive research to uncover these dynamics, existing methodologies often fail to produce models that accurately capture them, hindering precise control in diverse and safety-critical tasks. This work addresses this gap by discovering these dynamics and leveraging them in a control methodology that substantially outperforms existing methods. Our approach, grounded in Spectral Submanifold theory, enhances accuracy up to *sixfold* and improves tracking performance by up to 150 times across a diverse set of control tasks, achieving Pareto dominance in precision and computational efficiency. These advances enable the development of simple yet robust models suitable for real-time control, moving us closer to deploying highly adaptive, efficient, and safe continuum robots.

In an era where the convergence of robotics and the natural world is set to redefine the boundaries of exploration, interaction, and intervention, our endeavor to engineer robots capable of navigating and manipulating their environment with the grace of living organisms has never been more pertinent. At the forefront of this challenge is the field of continuum robotics, which, by emulating the fluid movements of biological systems, presents unprecedented opportunities for innovation across a broad range of applications-from the unexplored depths of our oceans to the boundless expanses of space.

Unlike their rigid counterparts, continuum robots are characterized by their ability to assume an infinite range of shapes and movements within the bounds of their design. This unique characteristic affords them numerous advantages over their rigid counterparts, namely dexterous manipulation and locomotion, flexibility, and physical compliance. These advantages are ideal in various applications spanning multiple disciplines, as shown in Fig. 1. For example, soft surgical robots provide unparalleled precision and safety in navigating the complex and constrained environments of the human body<sup>1</sup>. In deep-sea exploration, soft robotic fish exhibit pressure resilience allowing scientists to push the boundaries of scientific discovery and explore the deepest parts of our oceans<sup>2</sup>. Space satellites equipped with deformable manipulators will allow for more precise space debris removal and satellite servicing operations<sup>3,4</sup>, while continuum robots will enable us to explore internal and enclosed dynamic terrain structures of scientific interest on Mars and Enceladus<sup>5</sup>. In all these applications, the robots must operate safely with precision under computational constraints.

The distinctive characteristics that grant continuum robots their flexibility and dexterity also present significant challenges in their modeling and control. Their compliance and intricate motion capabilities lead to complex dynamics, driven by both geometric and material nonlinearities<sup>6,7</sup> as well as nonlinear resonances. Accurately capturing these dynamics typically requires high-dimensional, nonlinear models which in turn poses computational challenges for realtime control. Moreover, the inherent deformability of these robots complicates the design of effective sensing systems to detect and measure their dynamic behavior<sup>8</sup>. Despite these challenges, there is hope to tractably model these systems for control. Continuum robots are highly dissipative, thus despite their high dimensionality, their high-frequency degrees of freedom rapidly synchronize with the robots' low-dimensional, persisting dynamics, *i.edominant dynamics*. Additionally, by carefully choosing the robot's sensor modality and arrangement, one can uncover these dominant dynamics from observed measurement transitions. The emergence of machine learning (ML) carries the promise of unveiling these dynamics from data at great fidelity.

This work addresses the task of achieving precise and safe control of continuum robots under computational constraints. To this end, we seek the following desiderata to learning dynamics for model-based control:

1. Faithful models: The algorithm should learn highly predictive models that generalize across different operating regimes, enabling precise control across a variety of tasks.

<sup>&</sup>lt;sup>1</sup>Department of Aeronautics and Astronautics, Stanford University, Stanford, USA. <sup>2</sup>Institute for Mechanical Systems, ETH Zürich, Zürich, Switzerland.



**Fig. 1** | **Examples of continuum systems and invariant manifolds. a** Phase portrait and 1-dimensional invariant manifold of a damped pendulum. **b** Origami-inspired artificial muscles as a deployable structure for on-orbit rendezvous operations. (PNAS<sup>60</sup>). **c** Electrically-driven bionic trunk made out of shape memory alloy with potential application to minimally invasive surgery (Wiley<sup>61</sup>). **d** A pressure-resilient,

soft robot fish that can operate in deep sea trenches. **e** Illustration of the eigenvalues of the linear part of the high-dimensional system, with the dominant modes being the eigenvalues closest to the imaginary axis. The SSM is parameterized by these slow modes and carries the dominant dynamics of the system.

- 2. Low model complexity: The model should be sufficiently lowdimensional to enable real-time control in resource and computeconstrained hardware.
- 3. Data efficiency: The learning algorithm should require a small, yet sufficient sample size to extract a performant model, making it feasible to train models with limited real-world data.
- 4. Sensor efficiency: To facilitate observability, the learning methodology should be able to incorporate data from as many sensors as available on the robot without increasing the model's complexity.
- Safety constraints: The control scheme should be able to handle state and actuation constraints, allowing the system to operate in safetycritical settings.

Existing methodologies have yet to satisfy these desiderata, largely due to their inability to discern the dominant dynamics governing continuum robot behavior in a mathematically precise fashion. These methodologies typically divide into two camps: those that learn a direct control policy and those that create a surrogate model for control.

Model-free reinforcement learning (RL) is an approach where the robot learns to make decisions by taking actions in its environment to maximize some notion of cumulative reward. Since continuum robots have intricate dynamics and are high-dimensional in nature, these approaches have gained traction in the literature<sup>9</sup> as they bypass the necessity to explicitly model these complexities. Model-free RL has proven effective across various robot designs and control tasks, ranging from real-time control of dynamic maneuvers<sup>10</sup> to locomotion in soft robotic fish<sup>11</sup>. Despite these successes, model-free RL lacks inherent safety guarantees during the exploration phase in which agents may take unsafe actions while learning. While some methods have been developed to address these safety concerns<sup>12</sup>, they often increase the computational complexity and require more conservative exploration strategies, which may limit generalizability and data-efficiency. This limits the practicality of applying RL to continuum robots operating in safety-critical environments.

As a result, much of the continuum robot literature leverages datadriven methods to infer the dynamic models of continuum robots for control. One strategy involves using data to find an optimal compression of the robot's finite-element model to a lower-dimensional representation<sup>13–15</sup>. However, this often results in reduced models that remain too large for realtime MPC on embedded systems with longer prediction horizons and higher control frequencies. This can make controllers less robust to model mismatch and overly focused on short-term actions, leading the system into states where it cannot effectively handle constraints. Furthermore, the development of high-fidelity analytical governing equations for compression is a cumbersome and ad-hoc process. As an alternative to this hybrid approach, learning the dynamics from time-series data of observed measurements using neural networks (NN) has become popular<sup>16-19</sup>. Despite their success in closed-loop control, these methods share the data inefficiency and generalization challenges seen in RL<sup>20</sup>. Another promising avenue in data-driven control involves systems-theoretic approaches. These leverage the Fundamental Lemma from behavioral systems theory to predict input-output trajectories directly from data while offering rigorous guarantees on stability, robustness, and constraint satisfaction<sup>21,22</sup>. Such approaches, however, typically require sufficiently rich data encoded in large data matrices to ensure accurate predictions, which can render them computationally intractable for real-time control<sup>23</sup>. Moreover, their effectiveness has yet to be demonstrated on the complex, high-dimensional dynamics of continuum robots.

Alternatively, data-driven modeling approaches grounded in Koopman operator theory<sup>24</sup> have been demonstrated to achieve notable performance on various soft robot applications<sup>25-27</sup>. Approaches like Dynamic Mode Decomposition (DMD)<sup>28</sup> and Extended DMD (EDMD)<sup>29</sup> can be interpreted as approximations to an infinite-dimensional operator that linearly represents the system dynamics through observable functions in a function space. These methods yield linear models conducive to fast and efficient control design and boast greater data efficiency than NNs<sup>25</sup>. However, a major pitfall with these linear procedures is their reliance on finite-dimensional approximations to an inherently infinite-dimensional operator. For these methods to generalize, the chosen observables must span a finite-dimensional Koopman-invariant subspace, which occurs with probability zero, even for generic observables<sup>30</sup>. While one might hope to overcome this limitation by enlarging the dimension of observables, such enlargements often require hundreds or thousands of dimensions. This can lead to overfitting, where the model fits training data well but struggles to generalize to unseen trajectories<sup>31-34</sup>. Furthermore, the dimensionality of these linear models scales exponentially with the number of observables and basis functions. These limitations are fundamentally at odds: increasing the expressivity of the model to better capture the nonlinear dynamics leads to an exponential increase in model dimension. Not only does this restrict

robot design and sensing capabilities, it also fundamentally limits the model's accuracy because onboard compute constraints prevent the deployment of such large models.

To overcome these challenges, we leverage Spectral Submanifold Reduction (SSMR)<sup>35</sup>, an emerging modeling paradigm that was developed less than a decade ago and only recently introduced to the field of ML<sup>3637</sup>. Originally conceived to model non-linearizable phenomena in engineering applications<sup>38–40</sup>, SSMR's application to control was recently established<sup>41–44</sup> due to two key model features: high predictive capability and low dimensionality. Spectral Submanifolds (SSMs) are attracting invariant manifolds residing in the robots' phase space and learning the internal dynamics of these structures leads to faithful, low-dimensional models because all nearby trajectories converge to these dynamics exponentially fast. Consequently, the low-dimensional dynamics on SSMs are the de facto *dominant dynamics* that can efficiently be leveraged for real-time control of the robot. SSMs have also been shown to vary smoothly with respect to general control forces<sup>45</sup>, which makes the inclusion of control in the SSM-reduced dynamics possible.

In this work, we present a model-based control methodology based on SSMR which satisfy the aforementioned desiderata and advance earlier results<sup>42,43</sup> towards,

- (a) making the procedure for learning dominant dynamics with arbitrary observables more rigorous and practical,
- (b) evaluating our proposed models on different robot morphologies in both high-fidelity simulation and hardware platforms,
- (c) aggregating these results to provide a novel methodology to address control of continuum robots.

Our method consistently improves the state-of-the-art along multiple dimensions including model accuracy by up to a factor of 6 and tracking performance by up to a factor of 150. Furthermore, our methods exhibit Pareto dominance with respect to baseline methods along both accuracy and computational efficiency. These previously unattainable results highlight the importance of inferring dominant dynamics for effective modelbased control. Critically, in contrast to other data-driven methods which fit to individual trajectories, SSMR discovers robust (*i.e* structurally stable) phase space structures, and uncovers their internal dynamics with high accuracy. This advancement brings us closer to deploying highly adaptive, efficient, and safe continuum robots across various application domains.

#### Results

In this section, we present our approach to modeling and controlling nonlinear, continuum robots with limited training data and computational resources. We first address the practical challenges of limited sensor measurements and propose a solution that allows us to leverage the dominant dynamics for control. We then conduct a thorough comparison with current state-of-the-art methods, including a robustness study to assess the generalization capability of each model to trajectories well outside the training distribution. Our evaluation focuses on the predictive accuracy over a finite horizon as well as closed-loop performance, both with and without constraints. We conduct a Pareto analysis to assess the tradeoff between tracking error and solve time, evaluating the real-time feasiblity of each model.

Our baselines include the trajectory piecewise linearization (TPWL) model reduction approach<sup>13</sup>, DMD<sup>46</sup>, Koopman (often referring to EDMD)<sup>26</sup>, and the recent Koopman static pregain method<sup>25</sup>. We learned DMD and Koopman models using a modified version of<sup>26</sup> and TPWL models using<sup>13</sup>. Since the performance of DMD and Koopman is sensitive to the user-specified regularization and model dimension, we learned multiple models corresponding to different parameters. We then chose models that achieved the best tracking performance on slow, *i.e* quasi-static control tasks, while still being computationally tractable. In TPWL, the model dimension is determined by the subspace to which we project the high-dimensional finite-element model. We find the optimal dimension by conducting singular value decomposition (SVD) on training trajectories and retaining the singular values which explain over 99.99% of variance in the dataset. The models deployed in simulation and hardware are outlined in Table 1.

Diamond Simulation (9768 dimensions)	Dimension	# Training Data	Type of Observables	# Delays	Degree Polynomial
SSMR	6	8692 (+40 IC)	tip position	1	R(z): 2; b <sub>r</sub> (z): 1
SSSR	6	8692 (+40 IC)	tip position	1	R(z): 1; b <sub>r</sub> (z): 1
DMD	11	8692	tip position & 4 inputs (not delayed)	1	1
Koopman/EDMD	66	8692	tip position & 4 inputs (delayed)	1	2
TPWL	42	4352	tip & four elbow positions	N/A	-
Trunk Simulation (4254 dimensions)					
SSMR	6	4201 (+40 IC)	tip position	1	R(z): 3; b <sub>r</sub> (z): 1
SSSR	6	4201 (+40 IC)	tip position	1	R(z): 1; b <sub>r</sub> (z): 1
DMD	15	7111	tip position & 8 inputs (not delayed)	1	1
Koopman/EDMD	120	7111	tip position & 8 inputs (delayed)	1	2
TPWL	28	9711	tip position	N/A	-
Diamond Experiment ( $\infty$ dimensions)	_		_		
SSMR	6	16,560 (+34 IC)	tip position	1	R(z): 3; b <sub>r</sub> (z): 1
SSSR	6	16,560 (+34 IC)	tip position	1	R(z): 1; b <sub>r</sub> (z): 1
Koopman/EDMD	66	34,098	tip position & 4 inputs (delayed)	1	2
Koopman LQR + Static Pregain	6	34,098 (+291 SP)	tip position	1	1
TPWL	42	43,522	tip and elbow positions	N/A	-

We list the model parameters of our approach and baselines for each continuum robot. In this work we consider a monomial basis for SSMR and Koopman/EDMD, while the TPWL model is extracted from the finite-element simulation. For SSMR, R(z) is the autonomous part of the dominant dynamics while b<sub>r</sub>(z) is the control-affine part. The amount of training data counts the number of controlled state transitions, while those in parentheses either count the number of initial conditions (IC) for the decay trajectories (SSMR and SSSR) or the number of setpoint (SP) states (DMD LQR + Pregain). The decaying trajectories are sampled at 100 Hz. Our approach gives the lowest dimensional model across all robot morphologies. *SSMR* spectral subspace reduction, *SSSR* spectral subspace reduction, *DMD* dynamic mode decomposition, *EDMD* extended dynamic mode decomposition, *TPWL* trajectory piecewise linearization, *LQR* linear quadratic regulator.

#### Table 1 | Model Parameters



**Fig. 2** | **Spectral Submanifold Reduction (SSMR) for Model Learning and Control. a** Collect decaying and controlled trajectories of the continuum robot. The decaying trajectories are truncated to remove initial transients, ensuring the data is close to the slow, attracting SSM of preselected dimension. Critically, the decaying trajectories will characterize the structure of the autonomous SSM while controlled trajectories will calibrate the internal dynamics of this invariant manifold as it smoothly deforms under control. **b** Use SSMLearn<sup>36</sup> to extract the dominant dynamics of the continuum robot on the autonomous SSM. Optionally, we can reparameterize the SSM to obtain a model with observables more favorable for control. Lastly, the calibration step learns the effect of control parallel to the autonomous manifold. **c** The SSMR-based, control-oriented model is used in a model predictive control scheme where planning and control is conducted in the reduced coordinates that parameterize the SSM.

We validate our findings through high-fidelity finite-element simulations on a diamond-shaped soft robot with 9768 dimensions and an elephant-inspired trunk robot with 4254 dimensions<sup>47</sup>. These differing robot morphologies are shown in Fig. 2a. Additionally, we conduct realworld experiments on a physical realization of the diamond robot for both unconstrained and constrained control tasks. The simulated and experimental diamond robots are made of soft silicone and actuated by motors which pull on cables attached to the robot's four elbows. The addition of cable tension controls the robot's deformation, allowing it to move continuously within its workspace. The trunk robot is also made of soft silicone but is controlled by eight cables, four of which terminate at the midpoint of its structure with the other four terminating at the endpoint. The robot is actuated by motors which pull on the cables, allowing it to assume a variety of geometries to navigate its workspace.

#### Spectral Submanifold reduction for optimal control

To achieve precise, real-time control of nonlinear continuum robots, we uncover their inherently low-dimensional dynamics on SSMs. These SSMs are invariant manifolds that carry the robots' dominant dynamics. Intuitively, there are two types of SSMs: *autonomous* SSMs, fixed at an equilibrium of the uncontrolled system, and *non-autonomous* SSMs, which evolve over time with the controlled system. Based on rigorous mathematical results from<sup>45</sup>, we approximate the dominant dynamics associated with the true, non-autonomous SSM by training on *generic* measurement data taken near an *autonomous* SSM. While in practice our data will almost never lie exactly on the primary SSM, it will lie with probability one on a nearby invariant manifold (*i.e* secondary SSM), of which infinitely many exist and whose Taylor expansion has the same leading order terms<sup>48</sup>. Subsequently, we identify the impact of control on the SSM-reduced dynamics via calibration. This procedure is summarized in Fig. 2.

First, we collect data tailored to approximate the autonomous SSM, as shown in Fig. 2a. This approach requires two datasets: a collection of decaying trajectories from different initial conditions and a collection of controlled trajectories. The decaying trajectory data are collected through a series of training experiments, performed by randomly applying constant control inputs to the robot and then removing the control input to allow the robot to decay to its rest state. The controlled trajectory datasets are collected by applying a random sequence of smooth inputs to the robot in open-loop. The training process is sample efficient requiring similar or less amount of data to infer a performant model compared to data-efficient baselines, as shown in Table 1.

The observables we use to infer the dominant dynamics of the robot are time-delayed measurements of the *x-y-z* position of the tracking markers placed on the end-effectors of each robot. In this way, we target the slowest 6D SSM of the system to capture the slowest oscillatory dynamics in the 3D space. In general, the desired dimension of the SSM-based model equals the dimension of the slowest spectral subspace corresponding to the dominant modes of the linearized system. We infer this dimension via an SVD of the decaying trajectories dataset (*i.e* principal component analysis)<sup>36,49</sup> by retaining the top six modes which capture more than 95% of the variance in the data. We stress that although this procedure is presented for a specific set of robots with measurements taken at their end-effectors, our approach applies to generic control systems<sup>42,45</sup>.

Second, to extract dominant dynamics on the SSM, we must uncover the SSM's chart and parameterization map, along with its calibrated internal dynamics, as shown in Fig. 2b. The chart map (*i.e* encoder) is inferred via singular value decomposition on the training set of decaying trajectories. The parameterization map (*i.e* decoder) and internal dynamics are fit using a monomial basis and by minimizing reconstruction error and prediction, respectively. Models with nonlinear (*i.e* 2nd order terms or higher) maps and dynamics are referred to as SSMR models, while those with only linear terms are referred to as Spectral Subspace Reduction (SSSR) models. In the latter, rather than fit a manifold, we fit a linear subspace with linear internal dynamics. Overall, we obtain a low-dimensional dynamics model and an output map which relates the reduced coordinates to our desired observables.

Finally, we design a Model Predictive Control (MPC) scheme (as shown in Fig. 2c), which predicts the future states of the robot using our SSMR-based model to optimize control actions in real-time, while considering safety constraints. In summary, our MPC scheme plans in a drastically reduced space, then maps these predictions into our desired measurements on which we evaluate our cost function and state constraints. The receding horizon nature of MPC provides closed-loop feedback that robustly handles the uncertainties in our model. As the optimal control problem is non-convex, we approximate it as a series of convex problems<sup>50</sup> making it tractable by using reliable convex optimization solvers. Given that the control scheme exploits the low-dimensionality of our model, we benefit from low computational overhead, which in turn enables real-time, safe control.

## Practical challenges with time delays

Numerous factors, including cost, power, and geometric constraints, may limit the number of sensors we can place on our system. Despite these limitations, time-delayed measurements allow us to uncover the dominant dynamics from observed data. This is because with enough delays, embedding theorems<sup>51,52</sup> guarantee that we can detect the dominant dynamics from measured transitions.

As the dimension of the SSM increases, the increased time delay requirements imposed by these theorems result in filtering of non-autonomous, controlled behaviors. We refer to this filtering phenomenon due to increased time delays as the "curse of delays."

Recall that SSMR projects the delayed observables onto the leading singular vectors resulting from the SVD of the decaying trajectory dataset. These leading modes can be interpreted as principal component trajectories (PCT)<sup>53</sup> which represent the dominant patterns of decay and capture the most significant modes of change in the system's dynamics over time. The challenge is that for an increased number of delays, the decaying PCTs are less capable of representing transient trajectories. This is because projection to the PCTs effectively acts like a bandpass filter with respect to the frequencies of the PCTs, with decreasing bandwidth as the delays increase.

To overcome these practical challenges, we can reparameterize the time-delayed model to other observables (*e.g.* using fewer time delays) as long as the dimension of the observables is at least the same as the SSM. Section 4.3 gives more details on how to conduct such a procedure. To evaluate this claim, we learn two SSM models: one purely parameterized by  $n_d \in \{4, 10, 15, 20\}$  delays and the other which reparameterizes the original  $n_d$  model to a single delay observable. We then pass the figure-eight trajectory of the simulated diamond robot through the chart and parameterization maps of the two models at  $n_d$  to evaluate its reconstruction accuracy. Since this trajectory is sampled every 10 ms, the delays correspond to the past  $T_d \in \{40, 100, 150, 200\}$  ms (simulation time) of end-effector measurements.

Figure 3 shows that models featuring many time delays have low reconstruction accuracy. As suggested, reparameterizing the model to a

**Fig. 3** | **Effect of time delay on non-autonomous trajectories. a** Percent reconstruction of controlled trajectories of the diamond robot's end-effector as a function of the number of delays in the model. The 0.744762,0.317516,0.028023orange plot shows the figure-eight's reconstruction from a model with increasing time delays without reparameterization. The 0.054870,0.440281,0.010655green plot represents reconstruction for a model that is reparameterized to single time delay. **b** Qualitative depiction of the reconstruction of the figure-eight trajectory along the *x-y* plane for the original and reparameterized models.





**Fig. 4 | Model accuracy heat map of simulated diamond robot.** Open-loop performance of diamond robot models over a 50 ms horizon and evaluated across 500 different points along a given trajectory. The box spans the middle 50% of the data, with whiskers extending to the smallest and largest values within 1.5 times the interquartile range, and points outside the whiskers indicating outliers. The best

performing model is TPWL with an RMSE of 0.317 mm. SSMR and SSSR give comparable performance at 0.492 mm and 0.457 mm, respectively. DMD and Koopman/EDMD give the worst average RMSE at 3.104 mm and 2.962 mm, respectively.



**Fig. 5** | **Model accuracy heat map of simulated trunk robot.** Open-loop performance of trunk robot models over a 50 ms horizon, evaluated across 500 different points along a given trajectory. The box spans the middle 50% of the data, with whiskers extending to the smallest and largest values within 1.5 times the interquartile range, and points outside the whiskers indicating outliers. The best

performing model is SSMR with an average RMSE of 0.54 mm. TPWL comes next at 1.90 mm, while SSSR gives an average RMSE of 2.60 mm. DMD and Koopman/ EDMD give the worst performance at 4.24 mm and 3.68 mm average RMSE, respectively.

single time delay results in almost constant reconstruction accuracy, despite the base model being of higher  $n_d$  delays. Interestingly, we observed that both models exhibit good reconstruction accuracy for uncontrolled, decaying trajectories. This suggests that increased delays act as a bandpass filter with decreased passband about the frequencies of the principal modes detected in the decaying trajectories. Thus, if the desired control trajectory has a frequency different from the principal modes, increasing the number of delays will filter out important transients that characterize this trajectory. Thus, we adopt this reparameterization trick to construct SSMR and SSSR models with a single time delay, thereby avoiding the curse of delays.

#### Simulation open-loop prediction accuracy

In this section, we compare the predictive accuracy of our SSMR model against SSSR and established baselines in the literature. To conduct this evaluation, we explore the actuation space of the robot using a randomly sampled input sequence different from the ones used to learn both the SSMR and baseline models. We record the resulting open-loop trajectory and corresponding inputs, and at random, chose n = 500 points { $p_1, ..., p_n$ } along this trajectory. Starting from each point  $p_i$  as an initial state, we roll out

the system's trajectory over the next five time steps using the respective segment of the original input sequence. We quantify predictive accuracy by calculating the Root Mean Square Error (RMSE) between the predicted and actual trajectories for each  $p_i$  over a 50 ms horizon. By replicating this finite-horizon prediction process across all selected points, we achieve a comprehensive evaluation of our model's accuracy throughout the robot's configuration space. To provide further context on the dynamics of the trajectories, the average speed of the reference trajectory for the diamond robot was 74.7 mm/s, while that of the trunk robot was 30.3 mm/s.

Figures 4, 5 show the simulation open-loop evaluations for the diamond and trunk robot, respectively. SSMR models achieve excellent predictive performance for both robots. Intuitively, this means that we approximate the 6D autonomous SSM well and that trajectories of the full system indeed synchronize fast with our model, which approximates the internal dynamics of the SSM. This results if the control minimally excites the outer modes, or if the effect of outer modes are insignificant due to large timescale difference between the dominant and off-manifold dynamics. In both cases, the off-manifold transients only occur momentarily and the dominant dynamics capture most of the robot's behavior. In general, the **Fig. 6 | Simulation closed-loop results for diamond robot. a** Closed-loop trajectory for the slow figure-eight trajectory in the *x-y* plane with **b** Relative tracking ISE of SSMR for the slow figureeight compared to the other models. SSMR and SSSR perform similarly while Koopman/EDMD performs noticeably worse. DMD and TPWL perform significantly worse than SSMR. **c** Closed-loop trajectory for the fast figure-eight trajectory in the *x-y* plane with **d** Relative tracking ISE of SSMR for the fast figure-eight. This desired trajectory is far from the training distribution and SSMR and SSSR are the only models that remain robust to this out-ofdistribution task.



Fig. 7 | Simulation Closed-loop Results for Trunk Robot. a Closed-loop trajectory for the slow ASL trajectory in the *x-y* plane. b Relative tracking ISE of SSMR compared to the other models for the ASL trajectory. c Closed-loop trajectory for the slow pacman trajectory in the *x-y* plane. d Relative tracking ISE of SSMR for the pacman trajectory. Overall, SSMR outperforms all other models with respect to tracking performance. While SSSR still outperforms all other baselines, it is performs significantly worse than SSMR due to its inability to capture the inherent geometric nonlinearities of the robot.



SSMR models' prediction accuracy degrades as the end-effector moves further from the robot's uncontrolled, resting point. This is because our model includes only a first-order correction to the uncontrolled dynamics. To improve accuracy at positions further from the origin, we would need to include higher-order corrections.

For the diamond robot, TPWL outperforms all other models in this evaluation but degrades in accuracy for the trunk robot. This indicates that the diamond robot's behavior near this trajectory is close to linear. On the other hand, Koopman/EDMD and DMD exhibit the worst performance among all models. As expected, the Koopman/EDMD and DMD models in Figs. 4 and 5 exhibit no clear pattern in model accuracy with respect to the configuration of the robot. Since Koopman/EDMD and DMD will learn a Koopman-invariant subspace with probability zero, the Koopman/EDMD and DMD models overfit to their training data and do not generalize well to the evaluation dataset.

Interestingly, SSMR only slightly outperforms the SSSR models on the diamond robot evaluations, indicating that the diamond robot's behavior in this open-loop trajectory is nearly linear. This can be attributed to the robot's truss-like structure, where most of the motion occurs at virtual hinges, leading to mostly linear behaviors. Furthermore, SSSR outperforms Koopman/EDMD and DMD on both robots, highlighting that our model discovery approach infers the underlying linear dynamics from data more accurately than fundamentally linear methods which attempt to approximate the Koopman operator. However, the SSSR model's predictive accuracy deteriorates quickly further away from the robot's resting position and, hence, performs worse overall than the SSMR models. This is clearly visible for the simulated trunk robot whose end-effector was made to follow trajectories further from its equilibrium position.

## Simulated closed-loop trajectory-tracking performance

We now evaluate the closed-loop performance of the models on a variety of trajectory-tracking tasks. For these evaluations, the end-effectors of the robots were commanded to trace out various dynamic trajectories with a control frequency of 50 Hz and an MPC horizon of 100 ms. We evaluate trajectory tracking accuracy using the Integrated Squared Error (ISE) which sums the squares of the error over time. Like RMSE, it heavily penalizes larger errors, but it does so over the integral of the control process. For the diamond robot, we consider two 10 second long trajectories: a slow figure-eight trajectory at an average speed of 4.5 mm/s and a fast figure-eight at an average speed of 89.4 mm/s, both in the *x*-*y* plane. The results are shown in Fig. 6. For the trunk robot, we commanded its tip to follow two 10 second 2D trajectories in the *x*-*y* plane: one tracing the "ASL" symbol at an average speed of 11.9 mm/s, and the other tracing a Pacman symbol at 4.6 mm/s. The results for the trunk are shown in Fig. 7.

SSMR and SSSR consistently outperform the baseline models across all control tasks for both robots. In the case of the diamond robot, both methods effectively capture the dynamics of slow and fast trajectories, with SSMR achieving the best closed-loop performance. The performance gap





between SSMR and SSSR substantially widens for the fast figure-eight task, likely due to near-resonant behavior increasing nonlinearity. For the trunk robot, SSMR models surpass all other models, including SSSR, indicating that the trunk robot's significant geometric nonlinearities are better handled by nonlinear models. The enhanced relative performance of SSMR over SSSR across different robot morphologies and control tasks underscores the necessity for nonlinear models to improve performance in tasks that induce nonlinear behaviors.

Our results underscore two additional observations, highlighting the efficacy of our approach compared to the state-of-the-art. First, both Koopman/EDMD and DMD perform reasonably for the slow figure-eight but poorly on the fast counterpart, with Koopman/EDMD faring worse than DMD by over a factor of 20. Second, SSSR significantly outperforms both despite it extracting a simpler linear model. The former further suggests that these methods overfit to the training data, which consists of trajectories slower than the desired fast figure-eight trajectory. It is then unsurprising that Koopman/EDMD performs the worst because it has nonlinear features which pick up quasi-random variations in the data rather than learn the underlying dynamics. The latter, on the other hand, results from the fact that we are explicitly targeting resilient structures in the robots' phase space. This allows us to uncover even linear models which generalize to control tasks well outside of its training distribution. This resilience stems from the fact that SSMs and their associated spectral subspaces persist under perturbation, e.g. due to actuation and noise35.

## Trade-off between accuracy and computational complexity

To rigorously compare the models, we evaluate their performance along two dimensions: ISE and time to solve the optimal control problem. Figure 8 shows the Pareto log-plot comparing SSMR and SSSR against the baseline methods for the simulated diamond and trunk robot. Each point represents a time-discretized model in  $\Delta t = 0.02$  s, which is commensurate with its control period. Since each model's dimension is fixed, the solve time to control period ratio is the time to solve a single optimization problem divided by the time-discretization,  $\Delta t$ . Any model to the right of unit 1 on the solve time to control ratio axis is not real-time capable. In addition to the control tasks considered in Figs. 6, 7, we evaluate the tradeoff on an additional control task for each robot. For the diamond robot, we consider a fast circle trajectory where the diamond's end-effector is required to track a circle in the *y*–*z* plane (e.g., see Fig. 10b). For the trunk robot, we consider the task of tracking the Stanford symbol in the *x*–*y* plane (*e.g.* see Fig. 10c).

Overall, SSSR and SSMR demonstrate Pareto dominance across various robot morphologies. Both robots lie on the Pareto front, with SSMR exhibiting the lowest tracking error. While DMD also lies on the Pareto front, it has a higher ISE than both SSSR and SSMR. Koopman/EDMD and TPWL are Pareto-dominated by DMD, SSSR, and SSMR, and none of the TPWL models are real-time compatible. Although SSMR, SSSR, and DMD lie on the Pareto curve, they do not perform equally well. Generally, we aim



exhibit Pareto dominance, DMD has consistently higher ISE. **b** Pareto Log-plot for models of the trunk robot for various control tasks. SSMR, SSSR, and DMD are Pareto-dominant, with SSMR exhibiting the lowest ISE.

for minimal error while maintaining real-time computation. Critically, Fig. 8 shows that with a slight increase in computation time (still real-time), SSMR and SSSR achieve superior tracking performance with reasonable qualitative behavior, as shown in Figs. 6, 7. We observe a similar tradeoff between SSMR and SSSR for the trunk robot in Fig. 8b.

While SSMR and SSSR are both 6D, the jump in computationally complexity for SSMR stems from the fact that the majority of the MPC computation for the SSMR models results from computing the Jacobians at each timestep in the MPC horizon. Fortunately, the computational complexity of SSMR-based MPC does not scale with the observables, whereas for Koopman-based MPC this complexity scales exponentially. This is seen in the trunk robot, where the number of states in the Koopman/EDMD model is more than 100 and the MPC solve time grows rapidly. The TPWL models are relatively high-dimensional for both robots, also leading to large solve times. Thus, the high computational complexity of these models result in the loss of real-time control capability, as shown in Fig. 8.

#### Constrained control for simulated trunk

We now compare the closed-loop performance of each model on the trunk robot in an obstacle avoidance setting to evaluate their ability to achieve precise control while remain safe. Specifically, the trunk's tip is made to follow a 2D projection of the "ETH" symbol on the x-y plane with 20 circular obstacles at varying radii. The period of the trajectory is 10 seconds and we evaluate each model's closed-loop tracking performance over 100 randomly generated obstacles. We consider only the single time-delayed tip positions for the SSMR model and seek to evaluate each model's performance based on their ability to track the trajectory (*i.e* low RMSE) while minimizing the amount of constraint violations (*i.e* low violation ratio). Should it violate the constraints, the method should minimize the constraint violation (*i.e* low max violation).

The results of this evaluation are shown in Fig. 9. The planning horizon for all models is 60 ms, which is myopic and leads to abrupt changes in the control strategy near the boundary of a constraint. Furthermore, the density of obstacles results in narrow free space, which causes these abrupt changes to happen frequently throughout the control task. These abrupt control changes result in discontinuous behavior which poses a challenge for all methods. For example, the Koopman/EDMD model fails to generalize to these behaviors, leading to erratic and unsafe behavior. While SSSR outperforms Koopman/EDMD, geometric nonlinearities induced by this trajectory coupled with the discontinuous behavior near constraint boundaries drive the linear model out of its region of validity, leading to worse performance than SSMR. Lastly, TPWL does slightly better than SSSR, likely due to its piecewise linear models providing a degree of robustness at the discontinuities.

Overall, the SSMR model effectively handles these abrupt behaviors near the constraint allowing it to outperform all other models in terms of tracking accuracy and safety. Recent results<sup>54</sup> show that data-driven



**Fig. 9** | **Closed-loop Obstacle Avoidance for Simulated Trunk Robot. a** Trajectory of most accurate tracking of "ETH" symbol among the 100 simulations for each model. **b** Statistics for tracking performance (ISE). **c** Statistics for how often the robot

violated constraints (violation ratio). **d** Statistics for how much the robot violated the constraints (max violation). SSMR outperforms all models in terms of precision and safety.

Fig. 10 | Actuation, sensing, and control tasks of (a) റ റ േ hardware diamond robot. a Actuation and sensing setup for control of the real-world diamond robot.  $\square$ A **b** Diamond robot and its three control tasks: (from left to right) quasi-static figure eight in x-y, "Stan-A A ford" symbol with obstacles, and quasi-static circle in *y*–*z*. (b) 220 mm

approximation of a non-smooth SSM carrying the discontinuous trajectory is justified as long as the discontinuity is moderate (*i.e* the two dynamical systems on both sides of the discontinuity do not differ significantly). Analogously, the SSMR procedure in Fig. 2 approximates the flow of the system on a non-smooth SSM induced by the abrupt control inputs. These results highlight that this smooth approximation is effective and useful when dealing with constraints, but compute constraints limit the planning horizon.

#### Hardware Experimental Validation

We now evaluate each model's closed-loop performance on a hardware platform of the diamond robot. We expect that the hardware platform will exhibit more nonlinearity than the simulation due to unmodeled boundary conditions (*e.g.* imperfect fixation at the base, unmodeled friction at the interface, etc), viscoelasticity, manufacturing imperfections, and actuator hysteresis. Our experimental setup and the control tasks are shown in Fig. 10. In this evaluation, we run 10 closed-loop experiments across various tasks including a 2D figure eight in the *x-y* plane, a 2D Stanford symbol with obstacles in the *x-y* plane, and a slow circle trajectory in the *y-z* plane. The control frequencies for each model were chosen for real-time control: 20 Hz

for the SSMR and SSSR models, 10 Hz for the Koopman/EDMD model, 50 Hz for the DMD-based LQR model, and 2.5 Hz for the TPWL model.

For the unconstrained figure eight and circle control tasks, we baseline against the Koopman static pregain method<sup>25</sup>, which is essentially DMD-based LQR, augmented with a feedforward control term. This method amounts to computing a linear inverse kinematic map which is used in a feedforward fashion to track a desired trajectory. Intuitively, this feedforward term controls the robot close to the reference trajectory, while the feedback term is used to correct small deviations from the trajectory. Following<sup>25</sup>, we collect training data by commanding random step inputs and partitioning them into dynamic and static components to learn the DMD model and linear feedforward term, respectively. For TPWL, we calibrate the dynamic response of the model from the finite-element simulator to align it with our hardware platform.

The results for the unconstrained control tasks are reported in Fig. 11 while those of the constrained tasks are shown in Fig. 12. Overall, SSMR outperforms all baseline models in terms of closed-loop tracking performance and safety across all the control tasks. Furthermore, although SSSR is a 6D linear model, it outperforms the more complex Koopman/EDMD and TPWL baselines in terms of tracking performance across all tasks. Despite

 $\frac{135}{\mathrm{mm}}$ 



**Fig. 11 | Closed-loop Results for Diamond Robot Hardware Platform.** Ten closedloop experiments were conducted for each model. The bar plots and displayed values represent the average ISE for each model while the error bars represent 95% confidence intervals. The distribution of data points is shown to the left of each bar plot. **a** Qualitative performance of each model on quasi-static figure eight task in *x*–*y*.

**b** ISE statistics of all models for figure eight trajectory. **c** Qualitative performance of each model on quasi-static circle task in y–z. **d** ISE statistics of all models for circle trajectory. Bolded trajectory represents the trajectory with best tracking performance. SSMR and SSSR outperform the baselines on both control tasks.



**Fig. 12 | Closed-loop Obstacle Avoidance for Hardware Diamond Robot. a** Trajectory of most accurate tracking of "Stanford" symbol among the ten experiments for each model. **b** Statistics for tracking performance (ISE). **c** Statistics for how often the robot violated constraints (violation ratio). **d** Statistics of how



much the robot violated the constraints (max violation). The bar plots and displayed values represent the average value of each metric while the error bars represent the 95% confidence interval. SSMR outperforms all models in terms of precision and safety.

the fact that we derive the TPWL model from a high-fidelity simulator calibrated for the real robot, its performance still suffers due to the sim-toreal gap. This highlights the appeal of data-driven methods due to their flexibility in learning the dynamics of the actual system directly from realworld data. Interestingly, while Koopman/EDMD is trained on data from the real robot, its MPC variant performs worse than TPWL in tracking performance for unconstrained tasks and comparably to TPWL for constrained tasks. This suggests that the Koopman/EDMD model overfits to the training data and does not learn the underlying dynamics of the robot. When we opt for an LQR controller and add the pregain term, DMD outperforms TPWL but falls short of our SSMR model's performance. These results reaffirm the efficacy of our approach to learning dominant dynamics useful for real-time, predictive control.

#### Discussion

We have introduced SSMR, a data-driven methodology to model and control continuum robots. We leverage the recently developed SSM theory from dynamical systems to infer low-dimensional, yet faithful models of the robots' dominant dynamics from numerical and experimental data. This low-dimensionality affords computational efficiency, allowing us to design an MPC scheme that provides closed-loop control in real-time while enforcing safety constraints. Using a model learning algorithm that is theoretically justified, we obtain models that are simple yet generalizable across robot morphologies and control tasks. Furthermore, our models are simple (requiring only 6-dimensions and no more than third-order polynomials) and do not require significant amounts of training data. Despite this simplicity, we demonstrate that SSMR exceeds the performance of other available data-driven modeling methods in prediction accuracy, trajectory tracking, and safety.

Our zeroth-order approximation of the time-varying SSM amounts to assuming that the desired trajectory of the robot remains close to the origin and hence, can be modeled by the SSM computed at the origin. Although the trajectories considered in the previous sections are not particularly close to the origin, this approximation still provides an SSM-reduced model that outperforms other approaches when used in MPC. The robustness of SSMR confirms that we should infer the dominant dynamics of System 1 by learning persistent attracting structures in the phase space rather than model inherently non-robust individual trajectories. While here we take a coarse approximation to minimize computational complexity, an important next step will be to efficiently compute the full SSM structure along the desired orbit up to higher orders. This will be crucial to maintaining precision during more dynamic tasks that involve larger control inputs.

Additionally, while SSMR scales independently of the number of measurements and hence allows for more complex sensor arrangements, we have simplified our study here by using only a single sensor to measure the robot's end-effector position. Since different sets of observables might further improve the parameterization for the SSM, our future work will explore more complex sensor configurations. Another important direction for future work is addressing complex real-world dynamics, such as mechanical instabilities (e.g. buckling) and significant discontinuities like contact or fluid-structure interactions. Recent advances in SSM theory<sup>45</sup> also apply to systems with fixed points of varying stability, offering a promising approach to controlling structural buckling. To handle discontinuities, future work can extend non-smooth SSM modeling techniques from<sup>54</sup> and<sup>55</sup> to enhance SSMR-based models in contact-rich environments. These controllers will be particularly useful for applications such as soft robotic fish operating in flow-disturbed environments, such as deep sea trenches (as depicted in Fig. 1d) where strong currents interact with complex geological structures.

Finally, we plan to compare our approach with methods based on first principles, such as the Cosserat rod dynamics presented in<sup>56</sup>, to further motivate the practical advantages of data-driven models.

SSMR's flexibility coupled with its low computational cost and data efficiency makes it a promising choice for safely deploying continuum robots in high-stakes and resource-constrained environments. While the present work focuses on applications to continuum robots, our approach is independent of the specifics of the robots considered here and hence is generally applicable across robotics. Thus, we believe that our approach has the potential to enable safe and efficient control for emerging classes of autonomous systems across a spectrum of operating environments.

## Methods

## Spectral Submanifold theory

**SSMR:** Spectral Submanifold Reduction. Inspired by recent advances in nonlinear dynamics<sup>35,36,45,48</sup>, our work aims to identify the dominant dynamics of continuum robots from data for efficient model-based control. Throughout this work, we consider robots as control-affine, dynamical systems of the form

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{g}(\mathbf{x}(t))\mathbf{u}(t),$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t)),$$
(1)

where  $\mathbf{x} \in \mathbb{R}^{n_f}$  represents a state in the high-dimensional state space of the robot,  $\mathbf{u} \in \mathbb{R}^m$  is the control input,  $\mathbf{y} \in \mathbb{R}^o$  are the observed states, and

Eq. (1) is locally asymptotically stable at x = 0. Note that the finite-dimensional system in Eq. (1), can be seen as the result of a spatial discretization of the underlying PDE which govern the robot's continuum dynamics.

The success of data-driven modeling is typically attributed to the manifold hypothesis: that data sets lie along a low-dimensional submanifold, allowing for their parsimonious representation. In the context of time series, this submanifold is an attractor representing the set of states (or points in the phase space) towards which a system tends to evolve, regardless of its initial conditions. In other words, if the manifold hypothesis holds, it suggests that the dynamics of the data-generating process are confined to a low-dimensional attractor. This means that even though the time series data exist in a high-dimensional space, the process' behavior and evolution are governed by a simpler, lower-dimensional structure. Recent results in dynamical systems confirm the existence and uniqueness of these attracting structures for dissipative systems, called Spectral Submanifolds (SSMs)<sup>35</sup>, under a set of well-defined mathematical conditions that we now review.

Consider the linearized dynamics  $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$  with A: = Df(0) being the Jacobian of f at x = 0 and one of its spectral subspaces,

$$E := E_{j_1} \oplus E_{j_2} \oplus \dots E_{j_n}. \tag{2}$$

where  $E_{i_{\mu}}$  denotes the real eigenspace corresponding to an eigenvalue  $\lambda_{i_{\mu}}$  of A. Let  $\Lambda_E^{\kappa}$  be the set of eigenvalues related to E and  $\Lambda_{out}$  be the eigenvalues not related to *E*. If  $\max_{\lambda \in \Lambda_{out}} \operatorname{Re}(\lambda) < \min_{\lambda \in \Lambda_E} \operatorname{Re}(\lambda) < 0$ , then *E* is a *slow spectral subspace* corresponding to the *n* slowest decay modes. Based on SSM theory, for  $\mathbf{u}(t) = \mathbf{0}$ , the uncontrolled System 1 admits infinitely many invariant manifolds of the same dimension as  $E^{35}$ . These manifolds are tangent to E at the origin and attract all trajectories in the domain of attraction of x = 0 at exponential rates faster than their internal decay rates. If there are no resonances between eigenvalues in  $\Lambda_E$  and  $\Lambda_{out}$ , then precisely one of those invariant manifolds is as smooth as the original dynamical System 1. The internal dynamics of this special invariant manifold serve as the de facto reduced-order model with which nearby trajectories converge to exponentially fast. This smoothest invariant manifold, denoted  $\mathcal{W}_0(E)$ , is called the (primary) SSM fixed to the origin of the uncontrolled system. The internal dynamics is the observed dominant dynamics of System 1 for u(t) = 0 within the domain of attraction of the origin. For typical continuum robots, the entire range of movements and configurations they are designed to perform falls within domain of attraction of  $W_0(E)$ . The autonomous SSM, which represents the primary behavior of the uncontrolled system, effectively describes the main dynamics observed across all the robot's intended operations.

When  $\mathbf{u}(t) \neq 0$ , then recent results by<sup>45</sup> guarantee the survival of a timedependent SSM,  $\mathcal{W}_{\mathbf{u}(t)}(E, t)$ , in the controlled system, as long as either  $\|\mathbf{g}(\mathbf{x})\mathbf{u}(t)\|$  is a small perturbation to  $\mathbf{f}(\mathbf{x})$  or  $\|\mathbf{D}_t\mathbf{g}(\mathbf{x})\mathbf{u}(t)\|$  is small relative to  $\|\frac{d}{dt}\mathbf{f}(\mathbf{x})\|$ . The latter condition is arguably satisfied in our setting because the intended trajectories result in control actions that do not vary faster than the time scales of the robot's internal structural vibrations. In this setting<sup>45</sup>, calls  $\mathcal{W}_{\mathbf{u}(t)}(E, t)$  an adiabatic SSM, showing that it varies slowly along a unique anchor trajectory that has the same stability type as the uncontrolled  $\mathbf{x} = 0$ fixed point of System 1. In our setting, this anchor trajectory is the desired trajectory of the robot.

In the present study, we use the zeroth-order approximation  $(\mathcal{W}_{\mathbf{u}(t)}(E, t) \approx \mathcal{W}_0(E))$ , *i.e* we globally approximate the time-dependent SSM with the autonomous SSM anchored at the origin. To approximate the actual dominant dynamics within  $\mathcal{W}_{\mathbf{u}(t)}(E, t)$ , we include the first-order correction to the uncontrolled dynamics, which turns out to be simply the projection of  $B(\mathbf{x})\mathbf{u}(t)$  onto the spectral subspace *E* along the direction of fast eigenspaces outside  $E^{15}$ . To avoid the costly identification of those fast modes, here we simply take the projection onto *E* to be orthogonal, as is generally assumed by the simplest variant of the SSMLearn algorithm we use here from<sup>36</sup>.

Data-driven computations of SSMs from experimental data were recently developed and successfully applied to model complex, real-world systems<sup>36,37</sup>. The practicality of these methods stem from Whitney's

embedding theorem<sup>51</sup> which guarantees that almost all smooth maps, h, will embed an *n*-dimensional SSM in an *o*-dimensional observable space if  $o \ge 2n + 1$ . In case we do not have enough observables, Takens' embedding theorem<sup>52</sup> asserts that delay-embeddings of smaller sets of measurements allow one to reconstruct the dynamics on an SSM. As long as the observables are generic<sup>49</sup>, this theorem guarantees that an n-dimensional attractor of Equation (1) can be smoothly embedded in an observable space of dimension  $o \ge 2n + 1$ . Intuitively, this means that as long as we choose our sensor arrangement to sufficiently characterize the robots' n mode shapes, we detect its dominant dynamics from transitions in the observables. To do this one can increase the number of sensors along the robot's structure to obtain finer measurements of its geometric configuration. As a simplification, we only measure the x, y, and z positions of the end effectors in both the diamond and trunk robots, as shown in Fig. 2. This approach is sufficient for the considered control tasks, hence, we relegate more complex sensor arrangements to future work, as discussed in Section 3. We then take appropriate time-delayed measurements of the end-effector to achieve the dimensionality required by the Takens embedding theorem. For more details on observables and their genericity, we refer the reader to<sup>49</sup>.

#### SSMR for control

Combining our discussion in Section 4.1 with the SSMLearn algorithm of  $^{36}$ , dynamics in Eq. (1) of continuum robots by learning the dynamics on the autonomous SSM anchored at the robot's resting position. We then calibrate a control vector field which aligns these reduced dynamics with those on the time-varying SSM. Specifically, we aim to learn the parameterization map, w, and reduced dynamics, r, of the following form

$$\mathbf{y} = \mathbf{w}(\mathbf{z}) = \mathbf{U}_n \mathbf{z} + \mathbf{W} \mathbf{z}^{2:n_w},$$
  
$$\dot{\mathbf{z}} = \mathbf{r}(\mathbf{z}, \mathbf{u}) = \mathbf{R} \mathbf{z}^{1:n_r} + \mathbf{b}_r(\mathbf{z})\mathbf{u},$$
(3)

where **y** denotes the observable vector in Eq. (1),  $\mathbf{z} \in \mathbb{R}^n$  is the reduced coordinate vector which parameterizes the SSM,  $\mathcal{W}_0(E)$ , and  $\mathbf{z}^{cn_w}$  represent monomials of z from order *c* to order  $n_w$ . The coefficient matrices  $\mathbf{U}_n \in \mathbb{R}^{o \times n}$  and  $\mathbf{W} \in \mathbb{R}^{o \times q}$  are the linear and nonlinear parts of the parameterization map, where  $q = \binom{n+n_w}{n} - n$ ,  $\mathbf{U}_n^{\top}\mathbf{U}_n = \mathbf{I}_{n \times n}$ , and  $\mathbf{U}_n^{\top}\mathbf{W} = \mathbf{0}$ . For the reduced dynamics,  $\mathbf{R} \in \mathbb{R}^{n \times r}$  is the coefficient of the drift (*i.e* autonomous) term, while  $\mathbf{b}_r : \mathbb{R}^n \to \mathbb{R}^{n \times m}$  is the reduced state control input matrix. This procedure is fully data-driven and conducted with limited sensor measurements. Since the resulting model is low-dimensional, we can formulate a receding-horizon optimal control problem within  $\mathcal{W}_0(E)$  (as shown in Section 4.4) which can be solved in real-time.

Specifically, we tailor our sensor arrangement, data collection, and learning algorithm to infer the unknown coefficients in Eq. (3). First, we arrange sensors on the robot so that our observable measurements are generic<sup>49</sup> and produce feature-rich training data. Intuitively, genericity suggests that choosing a comprehensive sensor arrangement capable of detecting the robots' primary modes which span its expected operating configurations enables us to accurately model the SSM and its dynamics.

We then collect two training datasets: one consists of decaying trajectories from a collection of initial conditions, denoted Y, and the other consists of controlled trajectories, denoted  $Y_{u}$ . To learn an approximation of the dominant dynamics, we follow a three-step process.

1. We conduct SVD on the decaying trajectory dataset,  $\mathbf{Y} \in \mathbb{R}^{o \times N_{\text{traj}}}$ , to approximate the tangent space of the SSM *i.e* 

$$\mathbf{Y} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^{\top},\tag{4}$$

where  $N_{\text{traj}}$  represents the number of training trajectories and  $U_n$  represents the leading *n* columns of U, which are a good estimate of the tangent space of the *n*-dimensional SSM<sup>49</sup>.

2. To infer the unknown coefficients in the parameterization and autonomous dynamics of the SSM in Eq. (3), we solve the optimization

problems

$$\mathbf{W} = \arg\min_{\mathbf{W}} \|\mathbf{Y} - \mathbf{U}_n \mathbf{Z} - \mathbf{W} \mathbf{Z}^{2:n_w}\|,$$
  

$$\mathbf{R} = \arg\min_{\mathbf{n}} \|\dot{\mathbf{Z}} - \mathbf{R} \mathbf{Z}^{1:n_r}\|,$$
(5)

where  $\mathbf{Z} = \mathbf{U}_n^{\top} \mathbf{Y}$  encodes the training decay data matrix into reduced coordinates and the derivatives are comuted via numerical differentiation. 3. We approximate the dominant dynamics on the approximating SSM,

 $W_0(E)$ , via calibration that amends the autonomous SSM dynamics learned from decaying trajectories with the time-varying SSM dynamics of the controlled system. If  $b_r(z)$  is a smooth function of z, then this calibration amounts to solving the optimization problem

$$(\mathbf{B}, \mathbf{B}_r) = \arg\min_{\mathbf{B}, \mathbf{B}_r} \left\| \dot{\mathbf{Z}}_{u} - \mathbf{R} \mathbf{Z}_{u}^{1:n_r} - \mathbf{B} \mathbf{U} - \mathbf{B}_r \mathbf{Z}_{u}^{1:n_u} \otimes_c \mathbf{U} \right\|_F^2.$$
(6)

Here,  $\mathbf{X} \otimes_c \mathbf{U} := [\mathbf{X}_{\cdot,1} \otimes \mathbf{U}_{\cdot,1}, \dots, \mathbf{X}_{\cdot,N} \otimes \mathbf{U}_{\cdot,N}]$  is the Kronecker product applied columnwise,  $\mathbf{Z}_u = \mathbf{U}_n^\top \mathbf{Y}_u$  encodes the controlled training data matrix into reduced coordinates, and the control vector field is given by

$$\mathbf{b}_{\mathbf{r}}(\mathbf{z})(\mathbf{u}) \coloneqq \mathbf{B}\mathbf{u} + \mathbf{B}_{\mathbf{r}}(\mathbf{z}^{1:n_u} \otimes \mathbf{u}). \tag{7}$$

Our SSMR algorithm operates under the assumption that the control effort is sufficiently small, allowing the leading-order component tangential to the SSM to serve as an adequate approximation. In practice, we observe that the excitation of outer modes due to orthogonal control components lead to fast transients back to the manifold. Thus, the effect of excited external modes are short lived and our approximation is justified.

#### Reparameterizing the SSM for control

We avoid excessive time delays in the control loop by reparameterizing the SSM with a different set of *control observables*, thereby mitigating the curse of delays. At a high level, we detect the dominant dynamics from sufficiently time-delayed decay data, then embed the SSM in our chosen control observable space. In this section, we simplify notation by reusing the variable y to refer to our new control observable.

To detect the dominant dynamics, we first need to embed the SSM in an appropriate observable space. By following Step 1 in Section 4.2, we obtain a projection onto the reduced coordinates, z, which parameterizes the SSM within the observable space. This allows us to infer reduced coordinates whose dynamics approximate the dominant dynamics. These reduced coordinates are valid because the SSM is guaranteed to be embedded in  $\mathbb{R}^o$ and SVD of the observables' decaying trajectories in this space is a good approximation of the SSM's spectral subspace. To avoid the curse of delays, we then reparameterize the SSM in the new observable space, by finding a parameterization map,  $\mathbf{w} : \mathbb{R}^n \to \mathbb{R}^p$ , and chart map,  $\mathbf{v} : \mathbb{R}^p \to \mathbb{R}^n$ , commensurate with our control observable,  $\mathbf{y} \in \mathbb{R}^p$ ,

$$\mathbf{z} = \mathbf{v}(\mathbf{y}) = \mathbf{V}_n^{\top} \mathbf{y}$$
  
$$\mathbf{y} = \mathbf{w}(\mathbf{z}) = \mathbf{V}_n \mathbf{z} + \mathbf{W} \mathbf{z}^{2:n_w},$$
 (8)

such that  $v \circ w = I_{n \times n}$  and  $p \ge n$ . Since we have inferred valid reduced coordinates, fitting a parameterization map of the SSM on an observable space,  $\mathbb{R}^p$ , of at least the same dimension as the SSM is theoretically justified.

Given the decay data matrix in reduced coordinates, Z obtained from Step 1 in Section 4.2, we parameterize to the new observables by solving the following optimization problem

$$\mathbf{V}_{n}, \mathbf{W} = \arg\min_{\mathbf{V}_{n}, \mathbf{W}} \|\mathbf{Y} - \mathbf{V}_{n}\mathbf{Z} - \mathbf{W}\mathbf{Z}^{2:n_{w}}\|,$$
  
subject to :  $\mathbf{V}_{n}^{\top}\mathbf{V}_{n} = \mathbf{I}_{n \times n},$   
 $\mathbf{V}_{n}^{\top}\mathbf{W} = \mathbf{0},$  (9)

where our chart map is given by  $\mathbf{z} = \mathbf{v}(\mathbf{y}) = \mathbf{V}_n^\top \mathbf{y}$ . We then carry out Steps 2 and 3 in Section 4.2 to obtain our control-calibrated reduced dynamics,

which we can then use for efficient control synthesis. To solve the constrained optimization problem in Eq. (9), we restrict  $V_n$  to lie on the Steifel manifold of orthonormal matrices<sup>57</sup>. We can then use existing software packages<sup>58,59</sup> to numerically solve for the unknown coefficients.

In this work, we adopt a simpler approach, bypassing the need to formulate a manifold optimization problem. Let us denote  $\bar{z} \in \mathbb{R}^n$  as the reduced coordinates inferred from decaying trajectories of the potentially many time delayed measurements in  $\mathbb{R}^o$ .

1. We fit a parameterization map between our new control observable y and  $\bar{z}$  by solving the following unconstrained problem

$$\bar{\mathbf{V}}_{n}, \bar{\mathbf{W}} = \arg\min_{\bar{\mathbf{V}}_{n}, \bar{\mathbf{W}}} \left\| \mathbf{Y} - \bar{\mathbf{V}}_{n} \bar{\mathbf{Z}} - \bar{\mathbf{W}} \bar{\mathbf{Z}}^{2:n_{w}} \right\|,$$
(10)

where Y is the decay data matrix of the new control observables and Z is the reduced coordinate data matrix computed in Step 1 of Section 4.2.

- Since Eq. (10) is unconstrained, V
  <sub>n</sub> need not be orthonormal. Thus, we scale and rotate Z
   *Z* to obtain the new reduced coordinates Z = V<sup>T</sup><sub>n</sub> V
  <sub>n</sub>Z
   , where the orthonormal matrix V<sub>n</sub> ∈ R<sup>n×p</sup> is obtained by orthonormalizing V
  <sub>n</sub> via Gram-Schmidt and defines the chart map, v, in Eq. (8).
- 3. This chart map allows us to define a graph-style parameterization between the new reduced coordinates, z and the control observables, y, satisfying the orthogonality constraints in Eq. (9). Specifically, we solve another unconstrained optimization problem

$$\mathbf{V}_{n}, \mathbf{W} = \arg\min_{\mathbf{V}_{n}, \mathbf{W}} \|\mathbf{Y} - \mathbf{V}_{n}\mathbf{Z} - \mathbf{W}\mathbf{Z}^{2:n_{w}}\|,$$
(11)

where  $\mathbf{V}_n^{\mathsf{T}} \mathbf{V}_n = \mathbf{I}_{n \times n}$ . The solved coefficients define the parameterization map, w, in Eq. (8) between the reduced coordinates and our control observable.

We complete the procedure by conducting Steps 2 and 3 in Section 4.2 to fit the calibrated reduced dynamics in the new reduced coordinates, *z*. By using this reparameterization trick, we can approximate the dominant dynamics while overcoming the curse of delays during control operation.

#### SSMR-based model predictive control

Model predictive control (MPC) is a modern control strategy based on optimization that has widespread applications in both industry and academic settings. Its core concept involves continuously solving finite-horizon, open-loop control problems in real-time. Feedback is implicitly generated through the application of the initial segment of the optimized control sequence, followed by the repetition of the real-time optimization process at subsequent time steps. The advantages of MPC are threefold: (1) optimization of performance criteria; (2) handles state and input constraints; (3) applicable to general nonlinear systems.

MPC is amenable for precise, real-time control when the prediction model is accurate and sufficiently low-dimensional. By approximating the dominant dynamics of Eq. (1), SSMR generates faithful, low-dimensional models that enable real-time MPC designs. To design the MPC, we discretize Eq. (3) in time and, with slight abuse of notation, formulate the optimal control problem,

$$\begin{split} \min_{\mathbf{u}_{0},\dots,\mathbf{u}_{N-1}} & \parallel \mathbf{y}_{N} - \hat{\mathbf{y}}_{N} \parallel_{\mathbf{Q}_{f}}^{2} \\ & + \sum_{k=0}^{N-1} \left( \parallel \mathbf{y}_{k} - \hat{\mathbf{y}}_{k} \parallel_{\mathbf{Q}}^{2} + \parallel \mathbf{u}_{k} \parallel_{\mathbf{R}}^{2} \right) \\ \text{s.t.} \quad \mathbf{z}_{0} = \mathbf{v}(\mathbf{y}_{0}) \\ \mathbf{z}_{k+1} = \mathbf{r}(\mathbf{z}_{k}, \mathbf{u}_{k}), \quad k \in \mathbb{Z}_{[0,N-1]}, \\ \mathbf{y}_{k} = \mathbf{w}(\mathbf{z}_{k}), \quad k \in \mathbb{Z}_{[0,N-1]}, \\ \mathbf{y}_{k} \in \mathcal{Y}, \quad k \in \mathbb{Z}_{[1,N]}, \\ \mathbf{u}_{k} \in \mathcal{U}, \quad k \in \mathbb{Z}_{[0,N-1]}, \end{split}$$
(12)

where  $\hat{\mathbf{y}}_k$  is the reference state at time  $k, N \in \mathbb{Z}$  is the prediction horizon, Q,  $Q_f$  are positive semi-definite matrices which represent the observed state and terminal costs, while R is a positive-definite matrix that penalizes control input. The sets  $\mathcal{Y}$  and  $\mathcal{U}$  represent state and input constraints, respectively, that are potentially non-convex.

Intuitively, our MPC makes predictions on the approximated, lowdimensional SSM. It first infers the reduced state,  $z_{ko}$  from the observed state,  $y_{k}$ , at each time-step k via a state estimation step. Then, it predicts the evolution of the system in the reduced space (*i.e* tangent space) over the next N time-steps, and maps back to the approximate manifold to obtain the observables' evolution. While (12) is non-convex, we use sequential convex programming (SCP) to transform the MPC into a sequence of quadratic programs, which is solved iteratively until convergence. While SCP has been shown to be useful in for nonlinear MPC, the computational burden of computing Jacobians at each solve step necessitates low-dimensional models. Fortunately, the reduced states in SSMR are low-dimensional, making (3) possible to solve in real-time. Our SSMR-based MPC scheme is outlined in Fig. 2c.

## **Data Availability**

The datasets generated and/or analyzed during the current study have been deposited in the Open Science Framework repository at https://doi.org/10.17605/OSF.IO/AMFQ7.

## Code availability

• All code for visualizing and reproducing the results presented here is publicly available in the DominantDynamicsViz repository at https://github.com/jjalora/DominantDynamicsViz. • All code for generating SSMR and SSSR models is publicly available in the SSMR-for-control repository at https://github.com/jjalora/SSMR-for-control. • All code for generating DMD, Koopman/EDMD, and Koopman static pregain models is publicly available in the soft-robot-koopman repository at https://github.com/jjalora/soft-robot-koopman. • All code for running the model predictive controller and interfacing with the Simulation Open Framework Architecture for soft robot simulation is publicly available in the soft-robot-control repository at https://github.com/StanfordASL/soft-robot-control.

Received: 10 July 2024; Accepted: 22 January 2025; Published online: 05 March 2025

## References

- Burgner-Kahrs, J., Rucker, D. C. & Choset, H. Continuum robots for medical applications: a survey. *IEEE Transactions on Robotics* **31**, 1261–1280 (2015).
- 2. Li, G. et al. Self-powered soft robot in the mariana trench. *Nature* **591**, 66–71 (2021).
- Nahar, D., Yanik, P. M. & Walker, I. D. Robot tendrils: Long, thin continuum robots for inspection in space operations. In 2017 IEEE aerospace conference, 1–8 (IEEE, 2017).
- Tonapi, M. M., Godage, I. S. & Walker, I. D. Next generation rope-like robot for in-space inspection. In 2014 IEEE Aerospace Conference, 1–13 (IEEE, 2014).
- Zhang, Y. et al. Progress, challenges, and prospects of soft robotics for space applications. *Advanced Intelligent Systems* 5, 2200071 (2023).
- Khaniki, H. B., Ghayesh, M. H., Chin, R. & Amabili, M. A review on the nonlinear dynamics of hyperelastic structures. *Nonlinear Dynamics* 110, 963–994 (2022).
- Trivedi, D., Lotfi, A. & Rahn, C. D. Geometrically exact models for soft robotic manipulators. *IEEE Transactions on Robotics* 24, 773–780 (2008).
- Shi, C. et al. Shape sensing techniques for continuum robots in minimally invasive surgery: A survey. *IEEE Transactions on Biomedical Engineering* 64, 1665–1678 (2016).

- Thuruthel, T. G., Falotico, E., Renda, F. & Laschi, C. Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators. *IEEE Transactions on Robotics* 35, 124–134 (2018).
- Jitosho, R., Lum, T. G. W., Okamura, A. & Liu, K. Reinforcement learning enables real-time planning and control of agile maneuvers for soft robot arms. In *Conference on Robot Learning*, 1131–1153 (PMLR, 2023).
- 11. Yang, T. et al. A soft artificial muscle driven robot with reinforcement learning. *Scientific reports* **8**, 14518 (2018).
- 12. Gu, S. et al. A review of safe reinforcement learning: Methods, theories and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).
- Tonkens, S., Lorenzetti, J. & Pavone, M. Soft robot optimal control via reduced order finite element models. In 2021 IEEE International Conference on Robotics and Automation (ICRA), 12010–12016 (IEEE, 2021).
- Goury, O. & Duriez, C. Fast, generic, and reliable control and simulation of soft robots using model order reduction. *IEEE Transactions on Robotics* 34, 1565–1576 (2018).
- Katzschmann, R. K. et al. Dynamically closed-loop controlled soft robotic arm using a reduced order finite element model with state observer. In 2019 2nd IEEE international conference on soft robotics (RoboSoft), 717–724 (IEEE, 2019).
- George Thuruthel, T., Ansari, Y., Falotico, E. & Laschi, C. Control strategies for soft robotic manipulators: a survey. *Soft robotics* 5, 149–163 (2018).
- Thuruthel, T. G., Shih, B., Laschi, C. & Tolley, M. T. Soft robot perception using embedded soft sensors and recurrent neural networks. *Science Robotics* 4, eaav1488 (2019).
- Zheng, G., Zhou, Y. & Ju, M. Robust control of a silicone soft robot using neural networks. *ISA transactions* **100**, 38–45 (2020).
- Chen, G., Yang, X., Xu, Y., Lu, Y. & Hu, H. Neural network-based motion modeling and control of water-actuated soft robotic fish. *Smart Materials and Structures* **32**, 015004 (2022).
- 20. Kim, D. et al. Review of machine learning methods in soft robotics. *Plos one* **16**, e0246102 (2021).
- Berberich, J. & Allgöwer, F. An overview of systems-theoretic guarantees in data-driven model predictive control. *Annual Review of Control, Robotics, and Autonomous Systems* 8 (2024).
- Coulson, J., Lygeros, J. & Dörfler, F. Data-enabled predictive control: In the shallows of the deepc. In 2019 18th European Control Conference (ECC), 307–312 (IEEE, 2019).
- Zhang, K., Zheng, Y., Shang, C. & Li, Z. Dimension reduction for efficient data-enabled predictive control. *IEEE Control Systems Letters* (2023).
- Koopman, B. O. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences* 17, 315–318 (1931).
- 25. Haggerty, D. A. et al. Control of soft robots with inertial dynamics. *Science robotics* **8**, eadd6864 (2023).
- Bruder, D., Fu, X., Gillespie, R. B., Remy, C. D. & Vasudevan, R. Datadriven control of soft robots using koopman operator theory. *IEEE Transactions on Robotics* 37, 948–961 (2020).
- Castaño, M. L. et al. Control-oriented modeling of soft robotic swimmer with koopman operators. In 2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), 1679–1685 (IEEE, 2020).
- Tu, J. H.Dynamic mode decomposition: Theory and applications. Ph.D. thesis, Princeton University (2013).
- Williams, M. O., Kevrekidis, I. G. & Rowley, C. W. A data-driven approximation of the koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science* 25, 1307–1346 (2015).
- Haller, G. & Kaszás, B. Data-driven linearization of dynamical systems. *Nonlinear Dynamics* **112**, 18639–18663 (2024).
- Brunton, S. L., Budišić, M., Kaiser, E. & Kutz, J. N. Modern Koopman Theory for Dynamical Systems. *SIAM Review* 64, 229–340 (2022).

- Otto, S. E. & Rowley, C. W. Koopman operators for estimation and control of dynamical systems. *Annual Review of Control, Robotics, and Autonomous Systems* 4, 59–87 (2021).
- Shi, L., Liu, Z. & Karydis, K. Koopman operators for modeling and control of soft robotics. *Current Robotics Reports* 4, 23–31 (2023).
- 34. Shi, L. et al. Koopman operators in robot learning. *arXiv preprint arXiv:2408.04200* (2024).
- Haller, G. & Ponsioen, S. Nonlinear normal modes and spectral submanifolds: existence, uniqueness and use in model reduction. *Nonlinear dynamics* 86, 1493–1534 (2016).
- Cenedese, M., Axås, J., Bäuerlein, B., Avila, K. & Haller, G. Data-driven modeling and prediction of non-linearizable dynamics via spectral submanifolds. *Nature Communications* 13, 872 (2022).
- Axås, J., Cenedese, M. & Haller, G. Fast data-driven model reduction for nonlinear dynamical systems. *Nonlinear Dynamics* **111**, 7941–7957 (2023).
- Cenedese, M., Marconi, J., Haller, G. & Jain, S. Data-assisted nonintrusive model reduction for forced nonlinear finite elements models. *CoRRabs*/2311.17865 (2023).
- Xu, Z. et al. Data-driven modelling of the regular and chaotic dynamics of an inverted flag from experiments. *Journal of Fluid Mechanics* 987, R7 (2024).
- Kaszás, B., Cenedese, M. & Haller, G. Dynamics-based machine learning of transitions in couette flow. *Physical Review Fluids* 7, L082402 (2022).
- Mahlknecht, F. et al. Using spectral submanifolds for nonlinear periodic control. In 2022 IEEE 61st Conference on Decision and Control (CDC), 6548–6555 (IEEE, 2022).
- Alora, J. I., Cenedese, M., Schmerling, E., Haller, G. & Pavone, M. Data-driven spectral submanifold reduction for nonlinear optimal control of high-dimensional robots. In 2023 IEEE International Conference on Robotics and Automation (ICRA), 2627–2633 (IEEE, 2023).
- Alora, J. I., Cenedese, M., Schmerling, E., Haller, G. & Pavone, M. Practical deployment of spectral submanifold reduction for optimal control of high-dimensional systems. *IFAC-PapersOnLine* 56, 4074–4081 (2023).
- Alora, J. I. et al. Robust nonlinear reduced-order model predictive control. In 2023 62nd IEEE Conference on Decision and Control (CDC), 4798–4805 (IEEE, 2023).
- 45. Haller, G. & S Kaundinya, R. Nonlinear model reduction to temporally aperiodic spectral submanifolds. *Chaos* **34** (2024).
- Proctor, J. L., Brunton, S. L. & Kutz, J. N. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems* 15, 142–161 (2016).
- Coevoet, E. et al. Software toolkit for modeling, simulation, and control of soft robots. *Advanced Robotics* 31, 1208–1224 (2017).
- Haller, G., Kaszás, B., Liu, A. & Axås, J. Nonlinear model reduction to fractional and mixed-mode spectral submanifolds. *Chaos* 33 (2023).
- Axås, J. & Haller, G. Model reduction for nonlinearizable dynamics via delay-embedded spectral submanifolds. *Nonlinear Dynamics* 22079–22099.
- Bonalli, R., Cauligi, A., Bylard, A. & Pavone, M. Gusto: Guaranteed sequential trajectory optimization via sequential convex programming. In 2019 International conference on robotics and automation (ICRA), 6741–6747 (IEEE, 2019).
- 51. Whitney, H. The self-intersections of a smooth n-manifold in 2nspace. *Annals of Mathematics* 220–246 (1944).
- 52. Takens, F. Detecting strange attractors in turbulence. In *Dynamical* Systems and Turbulence, Warwick 1980: proceedings of a symposium held at the University of Warwick 1979/80, 366–381 (Springer, 2006).
- Dylewsky, D., Kaiser, E., Brunton, S. L. & Kutz, J. N. Principal component trajectories for modeling spectrally continuous dynamics as forced linear systems. *Physical Review E* 105, 015312 (2022).

- Bettini, L., Cenedese, M. & Haller, G. Model reduction to spectral submanifolds in piecewise smooth dynamical systems. *Int J. Nonlinear Mech.* 163, 104753 (2024).
- 55. Morsy, A. A., Xu, Z., Tiso, P. & Haller, G. Reducing finite element models of bolted joints using spectral submanifolds. *arXiv preprint arXiv:2409.05012* (2024).
- Till, J., Aloi, V. & Rucker, C. Real-time dynamics of soft and continuum robots based on cosserat rod models. *The International Journal of Robotics Research* 38, 723–746 (2019).
- 57. Boumal, N.An introduction to optimization on smooth manifolds (Cambridge University Press, 2023).
- Bergmann, R. Manopt.jl: Optimization on manifolds in Julia. Journal of Open Source Software 7, 3866 (2022).
- Townsend, J., Koep, N. & Weichwald, S. Pymanopt: a python toolbox for optimization on manifolds using automatic differentiation. *Journal* of Machine Learning Research **17**, 1–5 (2016).
- Li, S., Vogt, D. M., Rus, D. & Wood, R. J. Fluid-driven origami-inspired artificial muscles. *Proceedings of the National academy of Sciences* 114, 13132–13137 (2017).
- Goergen, Y., Rizzello, G. & Motzki, P. Systematic methodology for an optimized design of shape memory alloy-driven continuum robots. *Advanced Engineering Materials* 26, 2301502 (2024).

## Acknowledgements

We are grateful to Luis Pabon and Edward Schmerling for several helpful comments on an earlier version of this manuscript. We thank Matteo Zallio for designing and generating Figs. 2 and 10 as well as Aleksei Diuzhov for creating Fig. 1d. J.A. and M.P. disclose support for the research of this work from NASA University Leadership Initiative [grant number 80NSSC20M0163] and Office of Naval Research [grant number N00014-23-S-B001]. J.A. acknowledges funding from the Secretary of the Air Force Ph.D. Fellowship. The funders played no role in study design, data collection, analysis and interpretation of data, or the writing of this manuscript.

# **Author contributions**

J.A. and M.C. designed the research. J.A. carried out the research. J.A. and M.C. developed the software and analyzed the examples. J.A. wrote the paper. G.H. and M.P. led the research team and edited the paper.

# **Competing interests**

The authors declare no competing interests.

# Additional information

**Correspondence** and requests for materials should be addressed to John Irvin Alora.

Reprints and permissions information is available at http://www.nature.com/reprints

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

This is a U.S. Government work and not under copyright protection in the US; foreign copyright protection may apply 2025