

# Data-Driven Spectral Submanifold Reduction for Nonlinear Optimal Control of High-Dimensional Robots

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

**Abstract**—Modeling and control of high-dimensional, nonlinear robotic systems remains a challenging task. While various model- and learning-based approaches have been proposed to address these challenges, they broadly lack generalizability to different control tasks and rarely preserve the structure of the dynamics. In this work, we propose a new, data-driven approach for extracting *control-oriented*, low-dimensional models from data using Spectral Submanifold Reduction (SSMR). In contrast to other data-driven methods which fit dynamical models to training trajectories, we identify the dynamics on generic, low-dimensional attractors embedded in the full phase space of the robotic system. This allows us to obtain computationally-tractable models for control which preserve the system's dominant dynamics and better track trajectories radically different from the training data. We demonstrate the superior performance and generalizability of SSMR in dynamic trajectory tracking tasks *vis-à-vis* the state of the art, including Koopman operator-based approaches.

## I. INTRODUCTION

High-dimensional robotic systems promise to revolutionize the field of robotics due to the versatility brought forth by their large number of degrees of freedom (DOF). For example, continuum soft robots can exhibit *embodied intelligence* [1] in which they conform to surfaces and objects while maintaining a level of physical robustness unavailable to their more rigid counterparts. This level of compliance and elasticity make them well-suited to operate in delicate, geometrically constrained environments, which enable them to play crucial roles in settings where safe human-robot interaction is paramount.

Unfortunately, these advantages pose significant practical challenges for the modeling and control of these robots. This is due to the inherent nonlinearities and high DOF required to accurately capture the structural deformations that realize these compliant behaviors. While several model- and learning-based approaches have been proposed in literature to address some of these challenges, these methods suffer from their inability to tractably bridge the gap between having accurate, but low-dimensional models. This accuracy-dimensionality tradeoff results in methods that sacrifice predictive accuracy and structure preservation for a drastic decrease in dimensionality or vice versa.

J.A. is supported by the Secretary of the Air Force STEM Ph.D. Fellowship. This work was supported by the NASA University Leadership Initiative (grant #80NSSC20M0163) and KACST; this article solely reflects the opinions and conclusions of its authors and not any Air Force, NASA, nor KACST entity.

<sup>1</sup>Department of Aeronautics and Astronautics, Stanford University, Stanford, CA, 94305, USA {jjalora, schmerling, pavone}@stanford.edu

<sup>2</sup>Institute for Mechanical Systems, ETH Zurich, 8092 Zürich, Switzerland {mattiacc, georgehaller}@ethz.ch

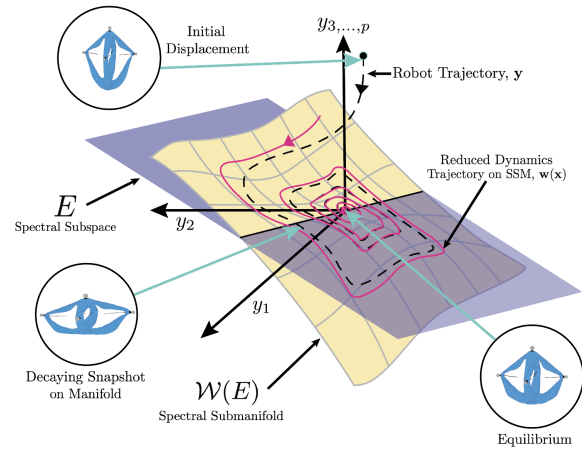


Fig. 1. The SSM is a low-dimensional invariant manifold in the robot's phase space which exponentially attracts full state trajectories, causing them to synchronize with the persistent dynamics on the SSM. These structures can capture highly nonlinear behaviors far away from the fixed point and can be approximated arbitrarily-well without increasing the dimension of the SSM.

Motivated by recent developments in *Spectral Submanifold* (SSM) theory [2] and its successful application to data-driven predictions of nonlinearizable phenomena [3], we propose a new data-driven *Spectral Submanifold Reduction* (SSMR) framework for learning low-dimensional, faithful dynamics of high-dimensional robots on SSMs. SSMs, as summarized in Figure 1, are low-dimensional, attracting invariant manifolds which capture highly nonlinear phenomena of high-dimensional systems. By learning the dynamics on these generic structures, we extract low-dimensional, control-oriented models that preserve the dominant physics of the system. This allows SSMR to overcome common drawbacks associated with data-driven approaches such as lack of generalizability, high-data requirement, and sensitivity to noise.

**Statement of Contributions:** (i) We present SSMR, the first data-driven approach for learning the dynamics of high-dimensional robots on SSMs for control. We extend recent work on *SSMLearn* [4] by providing the additional innovation of disambiguating the effect of control from the underlying dynamics on the SSM. This allows us to extract highly-accurate models for control in an *equation-free* manner.

(ii) We extend previous work on SSM-based control [5] to general control tasks by implementing a SSMR model predictive control (MPC) scheme and validating it on simulations of a high-dimensional soft robot. We show that SSMR outperforms the state-of-the-art methods in both trajectory tracking performance and computational efficiency, highlighting a key feature of SSMR: it neither compromises accuracy nor computational tractability.

**Related Work:** To address this issue, a body of work which involve linear [6]–[8] and nonlinear [9]–[11] techniques, seek to compress the governing equation of high-fidelity computational models using projection-based model reduction. This results in low-dimensional control surrogates suitable for online computation. A common limitation of these projection-based techniques is that the accuracy of the low-dimensional surrogate depends on the choice and size of the subspace, which can grow rapidly for only incremental improvement in accuracy. Additionally, since these approaches require knowledge of the governing equations, their application to real-world robots remain challenging. Indeed, the process of extracting accurate models from finite element code is an encumbering and code-intrusive process.

To overcome these challenges, there has been increasing interest in using machine learning techniques to construct data-driven models of high-dimensional robots. Much of the literature in this direction is focused on using different neural network (NN) architectures with varying levels of inductive bias, for learning approximations of these high-dimensional dynamics from observed transitions [12]–[14].

Another popular approach is the Koopman operator, which seeks to find a finite-dimensional approximation of an infinite-dimensional linear operator. Since observed dynamics under the Koopman operator are linear, the approach lends itself to established linear control techniques and has found success in the real-time control of soft robots [15], [16]. While conceptually appealing, most physical systems do not admit exact finite-dimensional, linear representations [17]. The approach also suffers from the accuracy–dimensionality tradeoff since, in theory, more accurate Koopman models require increased number of *a-priori* chosen observable functions.

A common drawback with current data-driven approaches is that they typically result in models that rarely preserve all, if any, of the inherent structure of the dynamics (*e.g.*, structural modes, passivity, etc). This results in models that do not generalize well to control tasks which involve trajectories outside the training set. By explicitly targeting rigorous and generic structures in the high-dimensional robot’s phase space, we are able to extract reduced-order models which overcome these issues.

**Organization:** We begin in Section II by detailing the class of high-dimensional systems we consider and posing the associated nonlinear optimal control problem. In Section III, we summarize relevant results from SSM theory and outline the data-driven procedure for learning control dynamics on SSMs. We then discuss our proposed control procedure in Section IV and present simulation results in Section V.

## II. PROBLEM FORMULATION

### A. High-Dimensional Optimal Control Problem

We consider control-linear mechanical systems with  $\mathcal{N} \in \mathbb{N}$  DOF. These systems encompass a wide range of robots such as manipulators, drones, and highly-articulated robots. In the continuum limit (*i.e.*,  $\mathcal{N} \rightarrow \infty$ ), these systems can also converge to the exact model of control-linear soft robots [18]. Such systems can be written in first-order form with state

vector  $\mathbf{x}^f(t) \in \mathbb{R}^{n_f}$  (where  $f$  denotes *full* state, as opposed to the *reduced* state  $\mathbf{x}$  introduced in Section III-B) as

$$\dot{\mathbf{x}}^f(t) = \mathbf{A}\mathbf{x}^f(t) + \mathbf{f}_{nl}(\mathbf{x}^f(t)) + \epsilon\mathbf{B}\mathbf{u}(t), \quad (1)$$

where  $n_f = 2\mathcal{N}$  since the states represent the position and velocity of each node,  $\mathbf{A} \in \mathbb{R}^{n_f \times n_f}$  is assumed to be Hurwitz (*i.e.*, the origin is an asymptotically stable fixed point for  $\epsilon = 0$ ) and  $\mathbf{B} \in \mathbb{R}^{n_f \times m}$  represents the linear control matrix. The nonlinear term  $\mathbf{f}_{nl} : \mathbb{R}^{n_f} \rightarrow \mathbb{R}^{n_f}$  belongs to the class of analytic functions and satisfies  $\mathbf{f}_{nl}(\mathbf{0}) = \mathbf{0}$ ,  $\partial\mathbf{f}_{nl}(\mathbf{0})/\partial\mathbf{x}^f = \mathbf{0}$ , while the parameter  $0 < \epsilon \ll 1$  introduces our assumption that the magnitude of the control inputs should be moderate compared to the autonomous dynamics. In this work, we consider control tasks near the vicinity of the robot’s equilibrium point (*e.g.*, a highly-articulated manipulator arm conducting pick and place tasks in a constrained workspace). Thus, if the desired trajectories are reasonable, this assumption is typically satisfied. Derivation of Equation (1) from a second-order mechanical system can be found in Appendix A [19].

We now pose the problem of controlling Equation (1) to follow arbitrary and dynamic trajectories in the vicinity of the origin. Consider the following continuous-time, optimal control problem (OCP) with quadratic cost and polytopic constraints in states and control:

$$\begin{aligned} & \underset{\mathbf{u}(\cdot)}{\text{minimize}} \quad \|\delta\mathbf{z}(t_f)\|_{\mathbf{Q}_f}^2 + \int_{t_0}^{t_f} \left( \|\delta\mathbf{z}(t)\|_{\mathbf{Q}}^2 + \|\mathbf{u}(t)\|_{\mathbf{R}}^2 \right) dt, \\ & \text{subject to} \quad \mathbf{x}^f(0) = \mathbf{g}(\mathbf{z}(0)), \\ & \quad \dot{\mathbf{x}}^f(t) = \mathbf{A}\mathbf{x}^f(t) + \mathbf{f}_{nl}(\mathbf{x}^f(t)) + \epsilon\mathbf{B}\mathbf{u}(t), \\ & \quad \mathbf{y}(t) = \mathbf{h}(\mathbf{x}^f(t)), \quad \mathbf{z}(t) = \mathbf{C}\mathbf{y}(t), \\ & \quad \mathbf{u} \in \mathcal{U}, \quad \mathbf{z} \in \mathcal{Z}. \end{aligned} \quad (2)$$

Here,  $\delta\mathbf{z}(t) = \mathbf{z}(t) - \bar{\mathbf{z}}(t)$  is the tracking difference between the performance variable,  $\mathbf{z}(t) \in \mathbb{R}^o$  and the desired trajectory  $\bar{\mathbf{z}}(t) \in \mathbb{R}^o$ . The observed state is denoted as  $\mathbf{y}(t) \in \mathbb{R}^p$  and  $[t_0, t_f]$  represents the time horizon.  $\mathbf{Q}, \mathbf{Q}_f \in \mathbb{R}^{o \times o}$  are positive semi-definite matrices which represent the stage and terminal costs, respectively, over the performance variables, while  $\mathbf{R}$  is a positive-definite matrix representing the cost on controls. The functions  $\mathbf{g} : \mathbb{R}^o \rightarrow \mathbb{R}^{n_f}$  and  $\mathbf{h} : \mathbb{R}^{n_f} \rightarrow \mathbb{R}^p$  map the performance variable to the full state and the full state to the observed state, respectively, while  $\mathbf{C} \in \mathbb{R}^{o \times p}$  is a selection matrix of states that we observe. Lastly, the constraint sets are defined as  $\mathcal{U} := \{\mathbf{u}(t) \in \mathbb{R}^m : \mathbf{M}_u\mathbf{u}(t) \leq \mathbf{b}_u\}$  and  $\mathcal{Z} := \{\mathbf{z} \in \mathbb{R}^o : \mathbf{M}_z\mathbf{z}(t) \leq \mathbf{b}_z\}$  with  $\mathbf{M}_u \in \mathbb{R}^{n_u \times m}$  and  $\mathbf{M}_z \in \mathbb{R}^{n_z \times o}$ , where  $n_u$  and  $n_z$  represent the number of constraints in the inputs and the observed states, respectively.

For high-dimensional dynamical systems, *i.e.*,  $n_f \gg 1$ , Equation (1) becomes a bottleneck and it is intractable to solve the OCP (2) in an online fashion. Recent advancements in nonlinear dynamics [2] show that the autonomous part of (1) lie on uniquely-smooth manifolds embedded in the phase space of the robot. These manifolds contain low-dimensional dynamics which we leverage to approximately solve OCP (2).

## III. DATA-DRIVEN MODELING OF LOW-DIMENSIONAL DYNAMICS

In this section, we describe our data-driven SSMR procedure to construct controlled, predictive models of soft robots

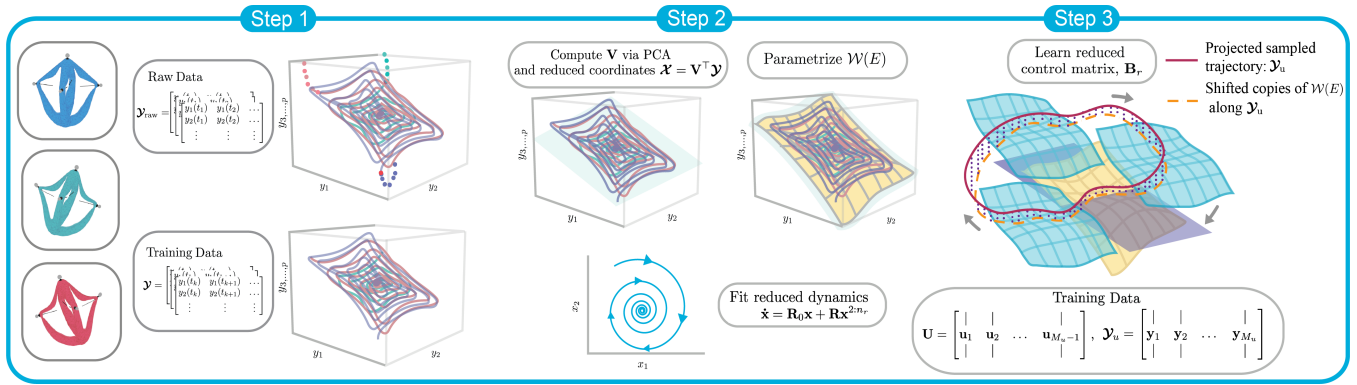


Fig. 2. Three-step procedure to learn control-oriented dynamics on the SSM. Step 1 depicts the data collection procedure whereby we displace the robot across various parts of its workspace and collect decaying trajectories. We then form our training data by truncating the dataset to approximate trajectories that near the manifold. Step 2 computes the SSM parameterization and autonomous dynamics while Step 3 regresses the control matrix which best explains how the autonomous SSM is translated under the influence of control. The “Diamond” soft robot is shown in its various displaced configurations on the far left.

from data. Our approach entails learning low-dimensional models directly as the reduced dynamics on attracting, low-dimensional invariant manifolds that generically exist in dissipative physical systems characterized by (1).

#### A. SSMs in a Nutshell

We define an  $n$ -dimensional spectral subspace  $E$  as the direct sum of an arbitrary collection of  $n$  eigenspaces of  $\mathbf{A}$ ,

$$E := E_{j_1} \oplus E_{j_2} \oplus \dots \oplus E_{j_n},$$

where  $E_{j_k}$  denotes the real eigenspace corresponding to an eigenvalue  $\lambda_{j_k}$  of  $\mathbf{A}$ . Let  $\Lambda_E$  be the set of eigenvalues related to  $E$  and  $\Lambda_{\text{out}}$  be that of eigenvalues not related to  $E$ . If  $\min_{\lambda \in \Lambda_E} \text{Re}(\lambda) > \max_{\lambda \in \Lambda_{\text{out}}} \text{Re}(\lambda)$ , then  $E$  represents the slowest spectral subspace of order  $n$ . Intuitively, the slowest spectral subspace corresponds to the dominant modes representing the persisting dynamics of the robot and can be extracted via modal analysis or principal component analysis (PCA).

Let us first assume that  $\epsilon = 0$ . For purely linear systems without external forcing (*i.e.*, the linearization of Equation (1)), any trajectories that start in  $E$  will remain in  $E$ , by the Spectral Mapping Theorem. When nonlinearities are introduced, superposition is lost and the autonomous part of Equation (1) is no longer invariant on  $E$ . The autonomous SSM corresponding to  $E$ ,  $\mathcal{W}(E)$ , is the smoothest  $n$ -dimensional manifold in the robot’s phase space which nonlinearly extends the invariance of  $E$ , *i.e.*, for the autonomous part of Equation (1)

$$\mathbf{x}_{\text{aut}}^f(0) \in \mathcal{W}(E) \implies \mathbf{x}_{\text{aut}}^f(t) \in \mathcal{W}(E), \quad \forall t \in \mathbb{R}, \quad (3)$$

where  $\dot{\mathbf{x}}_{\text{aut}}^f(t) = \mathbf{A}\mathbf{x}_{\text{aut}}^f(t) + \mathbf{f}_{\text{nl}}(\mathbf{x}_{\text{aut}}^f(t))$  and  $\mathbf{x}_{\text{aut}}^f \in \mathbb{R}^{n_f}$ . Given the smoothness of the SSM, the parameterization of  $\mathcal{W}(E)$  and the corresponding reduced dynamics can be represented by polynomial maps [20], as detailed in Section III-B.

Low-dimensional slow SSMs corresponding to slow spectral subspaces are ideal candidates for model reduction as nearby full system trajectories become exponentially attracted towards these manifolds and synchronize with the slow dynamics. Figure 1 gives a visual depiction of this property as well as the relationship between  $E$  and  $\mathcal{W}(E)$ . For a detailed definition of SSMs, see Appendix B [19].

For small  $\epsilon$ , the SSM  $\mathcal{W}(E)$  is still relevant for control. From a theoretical point of view, results on the existence of

non-autonomous SSMs subject to quasi-periodic forcing were established in [2]. Since quasi-periodic signals over a finite time interval are dense in the space of continuous signals over the same interval, we can interpret the trajectory of the system under control input as lying approximately on a time-varying, invariant manifold that is  $\epsilon$ -perturbed from  $\mathcal{W}(E)$ .

#### B. Reduced-order Models on SSMs

In general, since we seldom have access to the full state  $\mathbf{x}^f$ , we must construct the SSM and the reduced dynamics of our system in the space of observed states such that  $p \geq 2n + 1$  either by Whitney or Takens embedding theorems [3]. In case  $\mathbf{y}$  does not satisfy this condition, we use time-delay embeddings of  $\mathbf{y}$ , whereby our new observed measurements include current and past measurements of  $\mathbf{y}$ , in order to embed  $\mathcal{W}(E)$  in a space with sufficient dimension.

To describe the geometry of  $\mathcal{W}(E)$ , we seek a pair  $\mathbf{w}(\mathbf{x})$ ,  $\mathbf{v}(\mathbf{y})$  of smooth, invertible functions where  $\mathbf{y} = \mathbf{w}(\mathbf{x})$  uniquely maps the reduced state on the SSM to the observed state and  $\mathbf{x} = \mathbf{v}(\mathbf{y})$  maps the observed state to the reduced coordinates, where  $\mathbf{x} \in \mathbb{R}^n$  is the reduced state. By definition of the invariance and tangency properties of the SSM, the two maps that parameterize  $\mathcal{W}(E)$  must satisfy the invertibility relations [2],  $\mathbf{y} = (\mathbf{w} \circ \mathbf{v})(\mathbf{y})$  and  $\mathbf{x} = (\mathbf{v} \circ \mathbf{w})(\mathbf{x})$  such that

$$\begin{aligned} \mathbf{x} &= \mathbf{v}(\mathbf{y}) := \mathbf{V}^T \mathbf{y}, \\ \mathbf{y} &= \mathbf{w}(\mathbf{x}) := \mathbf{W}_0 \mathbf{x} + \mathbf{W} \mathbf{x}^{2:n_w}, \end{aligned} \quad (4)$$

where  $\mathbf{x}^{2:n_w}$  is the family of all monomials from order 2 to  $n_w$ , and  $n_w$  is the desired order of the Taylor series expansion for approximating the SSM. Also, the columns of  $\mathbf{V} \in \mathbb{R}^{n_f \times n}$  span the spectral subspace of  $E$  and  $\mathbf{W}_0, \mathbf{W}$  represent coefficient matrices of the SSM parameterization. In addition, the reduced dynamics on  $\mathcal{W}(E)$  are represented by

$$\dot{\mathbf{x}}_{\text{aut}} = \mathbf{r}_{\text{aut}}(\mathbf{x}) := \mathbf{R}_0 \mathbf{x} + \mathbf{R} \mathbf{x}^{2:n_r}, \quad (5)$$

where  $\mathbf{r}_{\text{aut}} : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is the autonomous reduced dynamics on the SSM;  $\mathbf{R}_0$  and  $\mathbf{R}$  represent the corresponding coefficient matrices. Since  $\mathcal{W}(E)$  is locally a graph over the spectral subspace  $E$  [3], we can identify the full state trajectory of System (1) on  $\mathcal{W}(E)$  described by Equation (5). Figure 1 gives an intuitive depiction of this concept.

We seek to learn the SSM-reduced dynamics for control and construct mappings that describe the trajectory of our observed

states on SSMs. Our three-step SSMR procedure involves: (1) collecting trajectories at or near the SSM. (2) learning the SSM geometry and the reduced dynamics in Equation (5), followed by (3) learning a linear control matrix that describes the effect of the controls in the reduced coordinates. Figure 2 summarizes the complete data-driven SSMR approach.

### C. Learning Autonomous Dynamics on SSMs

To learn the geometry and reduced dynamics on the SSM, the training data should involve only trajectories that are near the SSM. Decaying trajectories converge exponentially-fast onto the SSM even when far away from the equilibrium point. Thus, we obtain training data snapshots by displacing the robot along various directions in its workspace, then collect the observed state trajectory as it decays to its equilibrium position. In other words, we form an augmented matrix of  $M_y$  (possibly time-delayed) decay datasets  $\mathcal{Y}_{\text{raw}} = [\mathcal{Y}_1, \dots, \mathcal{Y}_{M_y}]$ , as shown in Figure 2 (left). We remove initial transients converging to the SSM in our datasets  $\mathcal{Y}_{\text{raw}}$  by truncating the first few states in the decay trajectories [4] and forming the dataset,  $\mathcal{Y}$ .

To start, we first compute  $\mathbf{V}$  by finding the  $n$  dominant modes of Equation (1). To do this we carry out principle component analysis on the trajectory dataset  $\mathcal{Y}$  and pick the  $n$  leading directions that capture a majority of the variance in the data. Indeed, for systems that do not feature strong nonlinearities, PCA is able to obtain a close estimate for the spectral subspace  $E$  to which the SSM is tangent [21].

Once we project  $\mathcal{Y}$  onto the reduced coordinate such that  $\mathcal{X} = \mathbf{V}^\top \mathcal{Y}$ , we can then learn the parameterization of  $\mathcal{W}(E)$  (i.e., learn the map  $\mathbf{w}$ ) by finding  $\mathbf{W}$  and  $\mathbf{W}_0$  via polynomial regression

$$(\mathbf{W}_0^*, \mathbf{W}^*) = \arg \min_{\mathbf{W}_0, \mathbf{W}} \|\mathcal{Y} - \mathbf{W}_0 \mathcal{X} - \mathbf{W} \mathcal{X}^{2:n_r}\|_F^2. \quad (6)$$

In a similar fashion, we can compute the polynomial form of the reduced dynamics in Equation (9) by finding the coefficients  $\mathbf{R}_0$  and  $\mathbf{R}$  via the regression

$$(\mathbf{R}_0^*, \mathbf{R}^*) = \arg \min_{\mathbf{R}_0, \mathbf{R}} \|\dot{\mathcal{X}} - \mathbf{R}_0 \mathcal{X} - \mathbf{R} \mathcal{X}^{2:n_r}\|_F^2, \quad (7)$$

The time derivative in Equation (7) can be computed using standard finite difference schemes if the sampling time of  $\mathcal{X}$  is much smaller than the Nyquist sampling time of the fastest mode in the SSM dynamics. Otherwise, we can also compute a discrete-time alternative to Equation (9) using a similar procedure through simple shifting operations on the dataset as in [22].

The procedure outlined above is suitable for both numerical and experimental data. For the former, we can choose  $\mathbf{y}(t) = \mathbf{x}^f(t)$  and let the dataset  $\mathcal{Y}$  consist of full state information during decay. We implement this procedure using a modified version of *SSMLearn*<sup>1</sup> [3].

### D. Learning the Control Matrix

Keeping in mind our moderate control assumption, we assume  $\epsilon = 1$ , without loss of generality, for the rest of the exposition. Once the reduced autonomous dynamics on  $\mathcal{W}(E)$  is known, we seek to learn the contribution of control

in the reduced coordinates. Our goal is to find the best linear control matrix  $\mathbf{B}_r \in \mathbb{R}^{n \times m}$  which best explains the difference between the controlled dynamics and our model of the autonomous dynamics. We explore the actuation space of the robot by randomly sampling a sequence of inputs,  $\mathbf{U}$ , and recording the corresponding (possibly time-delayed) observed state trajectory  $\mathcal{Y}_u$ , as depicted in Figure 2 (right).

We then project the observed states down to the reduced coordinates and form the reduced state matrix  $\mathcal{X}_u = \mathbf{V}^\top \mathcal{Y}_u$ . Additionally, we evaluate our model of the autonomous dynamics and form the matrix  $\dot{\mathcal{X}}_{\text{aut}} = \mathbf{r}_{\text{aut}}(\mathcal{X}_u)$ . Learning the (continuous-time) control matrix from data amounts to solving the minimization problem

$$\mathbf{B}_r^* = \arg \min_{\mathbf{B}_r} \|\dot{\mathcal{X}}_u - \dot{\mathcal{X}}_{\text{aut}} - \mathbf{B}_r \mathbf{U}\|_F^2, \quad (8)$$

where  $\dot{\mathcal{X}}_u$  is computed by finite differencing  $\mathcal{X}_u$ . Our learned, low-dimensional control dynamics is thus,

$$\dot{\mathbf{x}} = \mathbf{r}(\mathbf{x}, \mathbf{u}) := \mathbf{R}_0 \mathbf{x} + \mathbf{R} \mathbf{x}^{2:n_r} + \mathbf{B}_r \mathbf{u}. \quad (9)$$

In general, the introduction of control causes  $\mathcal{W}(E)$  to lose its invariance. Intuitively, though, we expect that the trajectories will remain within a small neighborhood of  $\mathcal{W}(E)$  since the effect of our control input is moderate compared to the system dynamics. Thus, we interpret this step as regressing a linear matrix that optimally translates the autonomous SSM under control inputs to be as close as possible to off-SSM trajectories.

## IV. SSM-BASED NONLINEAR MPC

### A. Reduced Order Optimal Control Problem

Learning the parameterization of  $\mathcal{W}(E)$  enables us to learn the intrinsic physics of our system, leading to low-dimensional and accurate reduced models with  $n \ll n_f$ . This allows us to approximate the OCP in (2) by posing an optimization problem with respect to the dynamics on the SSM as follows

$$\begin{aligned} \underset{\mathbf{u}(\cdot)}{\text{minimize}} \quad & \|\delta \mathbf{z}(t_f)\|_{\mathbf{Q}_f}^2 + \sum_{k=1}^{N-1} \left( \|\delta \mathbf{z}(t)\|_{\mathbf{Q}}^2 + \|\mathbf{u}(t)\|_{\mathbf{R}}^2 \right) \\ \text{subject to} \quad & \mathbf{x}(0) = \mathbf{V}^\top (\mathbf{y}(0) - \mathbf{y}_{\text{eq}}), \\ & \dot{\mathbf{x}}(t) = \mathbf{r}(\mathbf{x}(t)) + \mathbf{B}_r \mathbf{u}(t), \\ & \mathbf{z}(t) = \mathbf{C} \mathbf{w}(\mathbf{x}(t)) + \mathbf{z}_{\text{eq}}, \\ & \mathbf{z}(t) \in \mathcal{Z}, \quad \mathbf{u}(t) \in \mathcal{U}, \end{aligned} \quad (10)$$

where  $\mathbf{z}_{\text{eq}} \in \mathbb{R}^o$  and  $\mathbf{y}_{\text{eq}} \in \mathbb{R}^p$  are the performance and observed states at equilibrium. To solve the approximate OCP (10) numerically, we discretize the continuous-time system and use Sequential Convex Programming (SCP) to transform (10) into a sequence of quadratic programs. If  $n$  is small enough, we can compute the solution to the resulting approximate OCP in real-time. We implement our SSMR-based controller on top of the open-source soft robot control library<sup>2</sup> presented in [9]. See Appendix C [19] for more details on the SCP setup.

## V. SIMULATION RESULTS

### A. Simulation

We now compare our proposed SSMR method against the Trajectory Piecewise-Linear (TPWL) approach [9] and

<sup>1</sup><https://github.com/StanfordASL/SSMR-for-control>

<sup>2</sup><https://github.com/StanfordASL/soft-robot-control>



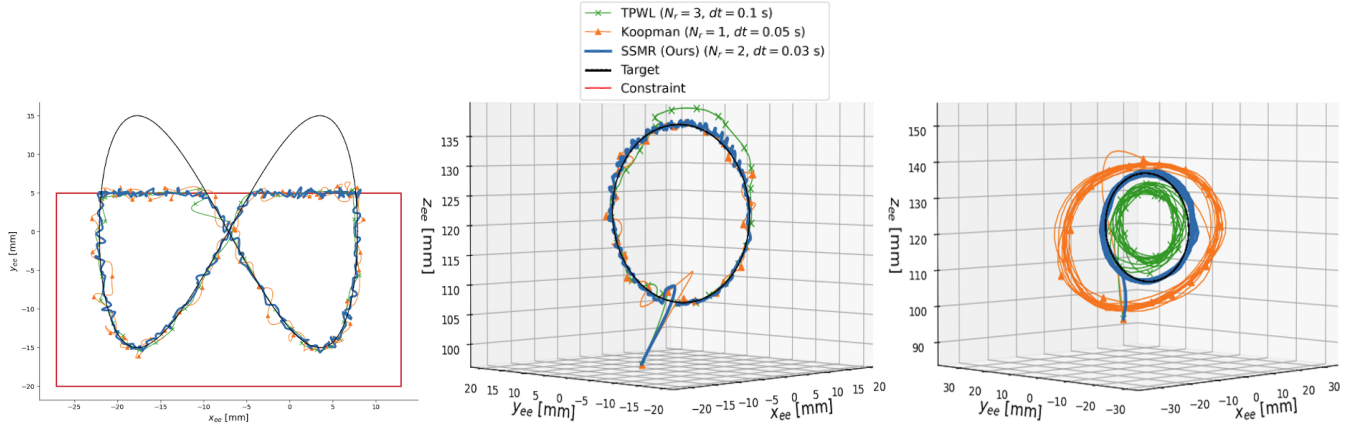


Fig. 3. Simulation results of tracking performance for tasks (1), (2), and (3) from left to right, respectively, with horizon length of  $N = 3$ . The parameters used were tuned for each method to yield the best, real-time performance across all tasks. The TPWL trajectory is shown in green, the Koopman trajectory in orange, and the SSMR in blue. The dotted black line represents the reference trajectory while the red line represent constraints. The quasi-static circle (task 2) MSEs (in  $\text{mm}^2$ ) are  $e_{\text{TPWL}} = 3.35$ ,  $e_{\text{koop}} = 0.91$ , and  $e_{\text{SSM}} = 0.53$ . The near-resonance circle (task 3) MSEs are  $e_{\text{TPWL}} = 21.75$ ,  $e_{\text{koop}} = 133.6$ , and  $e_{\text{SSM}} = 1.87$ .

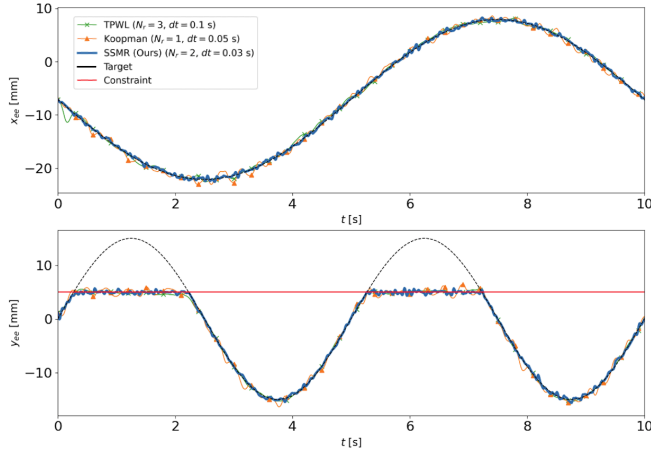


Fig. 4. Time-series simulation results of tracking performance for the quasi-static Figure Eight (task 1). The controller parameters for each approach are set similarly to those reported in Figure 3. The MSE (in  $\text{mm}^2$ ) for the TPWL, Koopman, and SSMR approaches are  $e_{\text{TPWL}} = 0.22$ ,  $e_{\text{koop}} = 0.38$ , and  $e_{\text{SSM}} = 0.13$ .

Koopman operator-based control approach [15] in simulations of the elastomer “Diamond” robot (shown in Figure 2), as detailed in [9] and [23]. We show that our approach outperforms these baselines while mitigating their drawbacks, namely lack of generalization, lack of robustness to noise, and computational intractability.

We carry out simulations using the finite-element based SOFA framework [25]; the Diamond robot mesh we used for simulation can be found in the *SoftRobots* plugin [26]. The parameters of our Diamond robot mirror those reported for a hardware replica in [23], where  $E = 175$  MPa is the Young’s modulus,  $\nu = 0.45$  is the Poisson ratio, and  $\alpha = 2.5, \beta = 0.01$  represent the usual parameters for Rayleigh damping *i.e.*,  $C = \alpha \mathbf{M} + \beta \mathbf{K}$ . Additionally,  $N_r \leq N$  represents the rollout horizon of the optimal solution,  $\mathbf{u}^*$ , to OCP (10) while the controller sampling time is  $T_c = N_r dt$ , where  $dt$  is the time-discretization of the dynamics.

In this work, we consider control tasks in which the end effector of the robot is made to follow various trajectories.

Thus, the performance variable  $\mathbf{z} = [x_{ee}, y_{ee}, z_{ee}]^T$  denotes the position of the top of the robot in its workspace. We also introduce additive Gaussian measurement noise to simulate real-world conditions. We consider three control tasks which include following (1) a figure eight in the  $x$ - $y$  plane subject to constraints, (2) a circle in the  $y$ - $z$  plane, and (3) the same circle but near resonance with the dominant mode of the system.

Since we are in a simulation environment, we collect full state information as training data *i.e.*, the  $i$ -th dataset is  $\mathcal{Y}_i = [\mathbf{x}_1^f, \mathbf{x}_2^f, \dots, \mathbf{x}_M^f]^T$ . We obtain this data by displacing the robot along 44 different points in its workspace and observe the decaying trajectory state transitions sampled at  $T_s = 1$  ms. This is consistent with the highest frequency mode in the SSM which has a period of roughly 330 ms. After conducting PCA on our training data, we found that the 3 leading configuration modes (6 modes in phase space) captured more than 95% of the variance in our dataset. Hence, we learn a cubic order, 6-D autonomous SSM parametrization described in (6) and its continuous-time, reduced dynamics (9) using the procedure in Section III. Lastly, we learn the control matrix by randomly sampling controls and then collect the resulting state transitions sampled at 10 ms.

Table I reports the mean-squared error tracking performance and average *cumulative* time to solve the QP for all trajectories at various controller parameters and time discretization of Equation (9). To enable real-time control, we seek control parameters such that the controller sampling time is at least an order of magnitude less than the solve time. Figures 3 and 4 depict simulation results for trajectories (1), (2), and (3) for controller parameters chosen to maximize performance while enabling real-time control.

These results show that our SSMR-based MPC scheme outperforms the TPWL and the Koopman approach in tracking performance across all trajectories considered, for small enough time discretization. Thus, our approach exhibits superior generalizability to control tasks as shown in the above figures and tables. Due to the low dimensionality of our learned model, we can solve the SCP iterations quickly and the computational burden grows modestly as the MPC

TABLE I. The table on the left shows mean squared error ( $mm^2$ ) for all considered trajectories while the right shows average cumulative QP solve times (in milliseconds) for the SCP algorithm. The Koopman model consists of polynomials up to order 2 over  $\mathbf{z} = [x_{ee}, y_{ee}, z_{ee}]^T$  with a single time delay ( $d_{\text{koop}} = 66$ ). We train separate, discrete-time Koopman models corresponding to the various time-discretizations,  $dt$ . The TPWL model parameters are set similarly to the ones reported in [23] ( $d_{\text{TPWL}} = 42$ ). We learn a single, continuous-time SSM model of cubic order and although it is low-dimensional ( $d_{\text{SSM}} = 6$ ), it outperforms the other approaches in both tracking performance (at low enough time-discretization) and solve time. The QP is solved using Gurobi [24] on a 1.6 GHz Intel Core i5 processor with 8 GB of RAM.

		Figure Eight					Circle				Near-Resonance Circle			Average Solve Times (ms)			
		$dt$ (ms)	10	20	50	100	10	20	50	100	10	20	50	10	20	50	100
$N=3$	SSMR (Ours)	<b>0.131</b>	<b>0.123</b>	0.227	0.718	<b>0.481</b>	<b>0.342</b>	1.353	4.480	<b>0.893</b>	<b>1.861</b>	32.87	<b>0.85</b>	<b>0.97</b>	<b>0.97</b>	<b>0.92</b>	
	TPWL	0.166	0.149	<b>0.164</b>	<b>0.191</b>	3.996	3.033	3.197	3.216	4.077	3.789	<b>4.472</b>	25.31	26.19	27.75	31.32	
	Koopman EDMD	1.053	1.230	0.500	0.725	2.985	1.806	<b>0.914</b>	<b>2.060</b>	1.864	6.707	135.3	6.08	6.10	5.95	5.92	
$N=5$	SSMR (Ours)	<b>0.153</b>	<b>0.136</b>	0.205	0.757	<b>0.466</b>	<b>0.348</b>	1.287	4.561	<b>0.810</b>	<b>1.816</b>	33.43	<b>1.55</b>	<b>1.49</b>	<b>1.62</b>	<b>1.26</b>	
	TPWL	0.160	0.155	<b>0.160</b>	<b>0.196</b>	3.350	3.278	3.265	3.254	3.698	3.900	<b>4.585</b>	52.20	51.23	55.51	58.52	
	Koopman EDMD	1.540	1.286	0.515	0.679	2.919	1.632	<b>0.895</b>	<b>2.119</b>	1.818	7.149	133.6	15.81	16.18	18.01	19.65	

horizon increases. As shown in Table I, the solve times for our approach are magnitudes lower than for the TPWL and Koopman-based methods, giving us more freedom to choose the controller parameters to enable real-time control. Observed deterioration of performance as  $dt$  increases in Table I is likely due to numerical errors introduced by coarser time-discretization of the dynamics since we learn a continuous-time model of Equation (9).

### B. Discussion

Additionally, the SSMR approach offers several practical advantages over the alternatives. First, our SSM-based model exhibits good closed-loop performance at longer horizons and does not suffer from numerical conditioning issues that plague the Koopman approach. We found that at horizons  $N \geq 10$ , the Koopman QPs were no longer solvable, which is likely due to ill-condition of the Koopman matrices. It is well-known that approximation of the Koopman operator is numerically challenging when many observables are considered [27]. Since we explicitly reason about the dynamics of the system in the learning process, we find that SSMR yields radically low-dimensional and thus, numerically well-behaved, models.

Second, our approach involves only two parameters: the order approximation and dimension of the manifold. Of these two, the dimension of the SSM is a property of the system dynamics, which can be inferred via a frequency analysis of the available data. The polynomial order of the SSM approximation controls the accuracy and the trade-off between generalization and overfitting. Given the limited amount of training data, we found that exceeding a third-order polynomial approximation led to overfitting. With more data, one can systematically increase the polynomial order to achieve higher accuracy. The primary drawback is computation of the jacobian in OCP (10), whose time-complexity grows exponentially with respect to  $n$  as a function of the polynomial order.

The size of the Koopman model grows rapidly with the number of observed states while the dimension of the projection basis for TPWL needs to be fairly large for acceptable closed-loop performance. In contrast, since off-manifold dynamics are sufficiently approximated by those on the SSM for closed-loop control, we can learn models of minimal size and tune the SSM order iteratively to increase model fidelity, as needed. This has considerable practical advantage over

learning-based approaches where it is well-known that closed-loop performance is highly sensitive to choice of dictionary features, size, and regularization.

## VI. CONCLUSION AND FUTURE WORK

In this work, we proposed a new data-driven approach for constructing control-oriented, reduced models of soft robots on spectral submanifolds. Using our approach, we can construct faithful, predictive, low-dimensional models which can be effectively used for real-time optimal control. We demonstrated that our SSM-based MPC scheme outperforms the state of the art significantly in both tracking error and computation time. The success of *SSMLearn* [4] in the experimental domain hints at the prospects of our SSMR approach for application to real world robots. Bolstered by promising results in a high-fidelity, finite-element simulation environment, we plan to validate the data-driven SSMR approach on our hardware platform detailed in [23].

While these results are promising, there are many open problems. For example, although the setting we consider involves tasks around an equilibrium point (e.g., manipulation tasks in a constrained workspace), many high-dimensional systems are not fixed to a point and can freely navigate their environment. Extending our SSMR framework to handle these settings would generalize our approach to a broader class of systems. Also, since most robotic systems have configuration-dependent actuation constraints, we plan to extend our approach to learning dynamics with state-affine control. Lastly, we plan to estimate errors arising from SSM approximation *a-priori* and derive error bounds for constraint-tightening control schemes in an MPC framework.

*Acknowledgements.* The authors thank Elisabeth Alora and Matteo Zallio for generating the instructive figures, Florian Mahlkecht for thoughtful discussions and help with initial implementations, and Spencer M. Richards for his careful review of the manuscript.

## REFERENCES

- [1] G. Mengaldo, F. Renda, S. L. Brunton, *et al.*, “A concise guide to modelling the physics of embodied intelligence in soft robotics,” *Nature Reviews Physics*, pp. 1–16, 2022.

- [2] G. Haller and S. Ponsioen, "Nonlinear normal modes and spectral submanifolds: Existence, uniqueness and use in model reduction," *Nonlinear Dynamics*, vol. 86, no. 3, pp. 1493–1534, Nov. 2016, ISSN: 1573-269X.
- [3] M. Cenedese, J. Axås, B. Bäuerlein, K. Avila, and G. Haller, "Data-driven modeling and prediction of non-linearizable dynamics via spectral submanifolds," *Nature Communications*, vol. 13, no. 1, p. 872, Feb. 2022, ISSN: 2041-1723.
- [4] M. Cenedese, J. Axås, 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.
- [5] F. Mahlknecht, J. I. Alora, S. Jain, *et al.*, "Using spectral submanifolds for nonlinear periodic control," *arXiv preprint arXiv:2209.06573*, 2022.
- [6] M. Thieffry, A. Kruszewski, C. Duriez, and T.-M. Guerra, "Control design for soft robots based on reduced-order model," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 25–32, 2018.
- [7] R. K. Katzschmann, M. Thieffry, O. Goury, *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)*, IEEE, 2019, pp. 717–724.
- [8] C. Della Santina, R. K. Katzschmann, A. Bicchi, and D. Rus, "Model-based dynamic feedback control of a planar soft robot: Trajectory tracking and interaction with the environment," *The International Journal of Robotics Research*, vol. 39, no. 4, pp. 490–513, 2020.
- [9] S. Tonkens, J. Lorenzetti, and M. Pavone, "Soft robot optimal control via reduced order finite element models," in *Proc. IEEE Conf. on Robotics and Automation*, 2021.
- [10] O. Goury and C. Duriez, "Fast, generic, and reliable control and simulation of soft robots using model order reduction," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1565–1576, 2018.
- [11] C. Farhat, T. Chapman, and P. Avery, "Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models," *International journal for numerical methods in engineering*, vol. 102, no. 5, pp. 1077–1110, 2015.
- [12] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi, "Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 124–134, 2018.
- [13] S. Greydanus, M. Dzamba, and J. Yosinski, "Hamiltonian neural networks," *Advances in neural information processing systems*, vol. 32, 2019.
- [14] M. Cranmer, S. Greydanus, S. Hoyer, P. Battaglia, D. Spergel, and S. Ho, "Lagrangian neural networks," *arXiv preprint arXiv:2003.04630*, 2020.
- [15] D. Bruder, B. Gillespie, C. D. Remy, and R. Vasudevan, "Modeling and control of soft robots using the koopman operator and model predictive control," *arXiv preprint arXiv:1902.02827*, 2019.
- [16] D. Bruder, C. D. Remy, and R. Vasudevan, "Non-linear system identification of soft robot dynamics using koopman operator theory," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 6244–6250.
- [17] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz, "Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control," *PloS one*, vol. 11, no. 2, e0150171, 2016.
- [18] C. Della Santina, C. Duriez, and D. Rus, "Model based control of soft robots: A survey of the state of the art and open challenges," *arXiv preprint arXiv:2110.01358*, 2021.
- [19] J. I. Alora, M. Cenedese, E. Schmerling, G. Haller, and M. Pavone, "Data-driven spectral submanifold reduction for nonlinear optimal control of high-dimensional robots (extended version)," *arXiv preprint arXiv:2209.05712*, 2022.
- [20] S. Jain and G. Haller, "How to compute invariant manifolds and their reduced dynamics in high-dimensional finite element models," *Nonlinear Dynamics*, Oct. 2021, ISSN: 1573-269X.
- [21] J. Axås, M. Cenedese, and G. Haller, "Fast data-driven model reduction for nonlinear dynamical systems," *arXiv preprint arXiv:2204.14169*, 2022.
- [22] J. L. Proctor, S. L. Brunton, and J. N. Kutz, "Dynamic mode decomposition with control," *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 1, pp. 142–161, 2016.
- [23] J. Lorenzetti, "Reduced order model predictive control of high-dimensional systems," Ph.D. dissertation, Stanford University, Dept. of Aeronautics and Astronautics, Stanford, California, Aug. 2021.
- [24] Gurobi Optimization, LLC, *Gurobi Optimizer Reference Manual*, 2022.
- [25] J. Allard, S. Cotin, F. Faure, *et al.*, "Sofa-an open source framework for medical simulation," in *MMVR 15-Medicine Meets Virtual Reality*, IOP Press, vol. 125, 2007, pp. 13–18.
- [26] E. Coevoet, T. Morales-Bieze, F. Largilliere, *et al.*, "Software toolkit for modeling, simulation, and control of soft robots," *Advanced Robotics*, vol. 31, no. 22, pp. 1208–1224, 2017.
- [27] S. Dahdah and J. R. Forbes, "System norm regularization methods for koopman operator approximation," *Proceedings of the Royal Society A*, vol. 478, no. 2265, p. 20220162, 2022.