Soft Robot Optimal Control Via Reduced Order Finite Element Models

Sander Tonkens[1], Joseph Lorenzetti[2], and Marco Pavone[2]

Abstract—Finite element methods have been successfully used to develop physics-based models of soft robots that capture the nonlinear dynamic behavior induced by continuous deformation. These high-fidelity models are therefore ideal for designing controllers for complex dynamic tasks such as trajectory optimization and trajectory tracking. However, finite element models are also typically very high-dimensional, which makes real-time control challenging. In this work we propose an approach for finite element model-based control of soft robots that leverages model order reduction techniques to significantly increase computational efficiency. In particular, a constrained optimal control problem is formulated based on a nonlinear reduced order finite element model and is solved via sequential convex programming. This approach is demonstrated through simulation of a cable-driven soft robot for a constrained trajectory tracking task, where a 9768-dimensional finite element model is used for controller design.

I. INTRODUCTION

Soft robots are an emerging class of robots that leverage natural compliance through continuous deformation to simplify tasks such as object manipulation, moving in complex environments, safely interacting with humans, and even assisting in surgical procedures [1], [2]. However, significant challenges in modeling, simulation, and control have limited their practical use. One fundamental challenge is that continuously deforming soft robots are infinite-dimensional systems that exhibit significant nonlinear behavior during structural deformation. Another is that diversity among soft robot designs makes it challenging to develop modeling techniques that are generalizable.

One approach to soft robot modeling and control is to hand-engineer simplified models of the robot's motion by making approximations. For example piecewise constant curvature models [3], [4], beam models [5], and Cosserat models [6] have been used. However, these low-fidelity approximations are typically tailored to specific robot geometries and often model only the robot's kinematics, making it challenging to design controllers for dynamic tasks.

Data-driven methods have also been developed to generate models directly from input-output data. This approach has been used to develop both kinematic and dynamic controllers. For example, [7] learns a differentiable kinematics model for solving inverse kinematics problems, and [8], [9] learn neural network dynamics models to develop closed-loop controllers. Another data-driven approach is proposed in [10], which uses Koopman operator theory to build a dynamics model that is used for model predictive control. While data-driven methods are generalizable to different
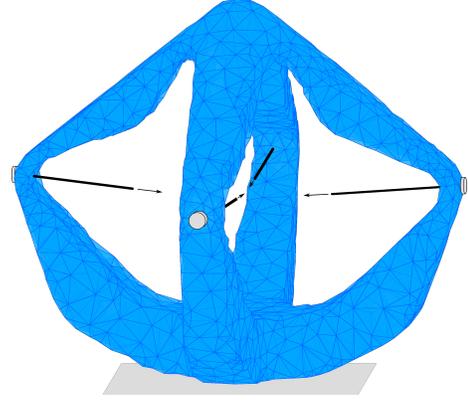
[1]Sander Tonkens is with the Department of Mechanical Engineering, Stanford University, Stanford, CA 94305, USA `tonkens@stanford.edu`

[2]Joseph Lorenzetti and Marco Pavone are with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, USA {`jlorenze, pavone`}`@stanford.edu`

Fig. 1: Finite element mesh for a "Diamond" soft silicone robot with $\mathcal{N} = 1628$ mesh nodes. This robot is fixed at the base and is actuated by controlling the tension in four cables (black) attached to the robot's "elbows". Finite element methods offer a systematic approach to generating high-fidelity dynamics models of soft robots.

types of soft robots, they fail to leverage physics-based principles and there is no systematic procedure for developing them. Additionally, reliance on experimental data for model identification and controller validation precludes the use of data-driven modeling approaches in the design process.

Alternatively, finite element methods (FEMs) provide a systematic, physics-based approach to soft robot modeling. These approaches can be used to develop high-fidelity models for a wide variety of soft robots and can be directly incorporated into the design process. While some techniques directly use FEM models for inverse kinematics based control [11] or trajectory optimization [12], the high-dimensionality of FEM models (e.g. thousands to tens of thousands of degrees of freedom) makes the design of real-time dynamic controllers challenging.

In this work, we propose an approach that leverages high-fidelity finite element models to control cable-actuated soft robots. The challenge of computational efficiency is addressed by using model order reduction techniques to compress the high-dimensional FEM models without significant loss in modeling accuracy. The resulting reduced order models (ROMs) are then used to efficiently solve optimal control problems, such as trajectory tracking tasks.

Related Work: A variety of principled model order reduction techniques have been developed for compressing high-dimensional dynamics models [13]. In the context of soft robotics, these methods can be useful for reducing the dimensionality of FEM models by orders of magnitude without significant loss in accuracy.

Specifically, linearized reduced order FEM models have been used to design regulating output-feedback controllers for cable-actuated [14] and pneumatically-actuated [15] soft robots, as well as trajectory tracking controllers [16]. Linear ROMs have also been used in contexts beyond soft robotics for solving constrained optimal control problems [17], [18]. However, using a *linearized* FEM model is not sufficient for many dynamic control problems due to the significant nonlinearities arising from the soft robot's deformation.

While *nonlinear* FEM model reduction techniques exist, they have generally been developed for simulation applications and are difficult to use for controller synthesis. In fact, in the context of soft robotics, *nonlinear* FEM model reduction has only been used to design inverse kinematics-based controllers [19], which are not sufficient for dynamic control tasks such as trajectory tracking. In summary, due to the use of either *linearized* models or kinematics-based controllers, existing approaches that leverage reduced order FEM models cannot adequately address dynamic soft robot control tasks.

Statement of Contributions: This work proposes an approach for constrained optimal control of soft robots that leverages high-fidelity *nonlinear* finite element models. We handle the computational challenges induced by the high-dimensionality of the FEM models by exploiting existing model order reduction techniques in the design of our control scheme. In particular, we approximate the FEM model with a low-dimensional piecewise affine model that is amenable to optimal control. We design an output feedback receding horizon control scheme based on sequential convex programming, which can be used for a variety of dynamic control tasks including setpoint and trajectory tracking. We demonstrate the performance of the proposed controller in simulation, using the soft robot shown in Figure 1 with a finite element model of dimension 9768. An implementation of the approach[1] that leverages SOFA [20], an open-source FEM software toolkit, is also provided.

Organization: We begin by presenting a general form of the nonlinear soft robot FEM model and the associated optimal control problem in Section II. Next, in Section III we use reduced order modeling techniques to construct a ROM, which is then used for computationally efficient optimal control in Section IV. We present simulation results in Section V and conclude with the merits of our approach and avenues of future work in Section VI.

II. PROBLEM FORMULATION

We begin by defining the soft robot finite element model and formulating the control problem.

A. Soft Robot Finite Element Model

Finite element methods are a numerical approach for solving partial differential equations (PDEs) and have been used for physics-based modeling and simulation in many domains, including fluid and structural mechanics as well as soft robotics [20]. These methods solve PDEs by performing a spatial discretization at a finite number of nodes defined by a mesh, which results in a finite set of ordinary differential equations corresponding to the state of each mesh node.

In the context of soft robotics, the state of each node consists of the node's position and velocity, resulting in six degrees of freedom. Therefore, for a spatial discretization with $\mathcal{N}$ nodes the resulting FEM model will have a total dimension of $n^f = 6\mathcal{N}$ (FEM model variables are denoted using $(\cdot)^f$). In particular, the soft robot FEM model is defined as in [19] using Newton's second law:

$$M^f \dot{v}^f = P^f - F^f(q^f, v^f) + H^f(q^f)u, \quad \dot{q}^f = v^f, \tag{1}$$

where $q^f(t) \in \mathbb{R}^{3\mathcal{N}}$ is the vector of node positions, $v^f(t) \in \mathbb{R}^{3\mathcal{N}}$ is the vector of node velocities, and $u(t) \in \mathbb{R}^m$ is the

[1]github.com/StanfordASL/soft-robot-control

control input. Additionally, $M^f \in \mathbb{R}^{3\mathcal{N} \times 3\mathcal{N}}$ is a constant mass matrix, $P^f \in \mathbb{R}^{3\mathcal{N}}$ represents constant external forces (e.g. gravity), $F^f(q^f, v^f) : \mathbb{R}^{3\mathcal{N}} \times \mathbb{R}^{3\mathcal{N}} \to \mathbb{R}^{3\mathcal{N}}$ is a nonlinear function that defines the internal forces (e.g. stresses from deformation), and $H^f(q^f) : \mathbb{R}^{3\mathcal{N}} \to \mathbb{R}^{3\mathcal{N} \times m}$ is a nonlinear function that specifies the input matrix. The internal forces $F^f(q^f, v^f)$ can be modeled in several ways depending on the material properties of the robot. A classic example is to use a linear-elastic model, which assumes a linear stress-strain relationship through Hooke's law.

To enable output feedback control we assume a linear measurement model given by:

$$y = C_y^f x^f, \tag{2}$$

where $x^f = [v^{f^T}, q^{f^T}]^T \in \mathbb{R}^{n^f}$ is the combined state vector and $y(t) \in \mathbb{R}^p$ is the measurement. Additionally, in FEM-based control problems only a small set of variables may be of particular interest (e.g. end-effector position and velocity). We denote these performance variables as $z(t) \in \mathbb{R}^o$ and assume a linear performance output model:

$$z = C_z^f x^f. \tag{3}$$

This output model can be used to simplify the formulation of the optimal control problem by expressing the cost function and constraints in terms of the relevant performance variables rather than the full state x^f of the FEM model.

B. Constrained Optimal Control Problem

In this section we introduce the soft robot optimal control problem (OCP). First, a set of constraints on the performance variables z and control u are assumed to be defined by:

$$u \in \mathcal{U}, \qquad z \in \mathcal{Z}, \tag{4}$$

where $\mathcal{U} := \{u \mid H_u u \leq b_u\}$ and $\mathcal{Z} := \{z \mid H_z z \leq b_z\}$ are convex polytopes with $H_u \in \mathbb{R}^{n_u \times m}$ and $H_z \in \mathbb{R}^{n_z \times o}$. Second, a cost function is defined over a finite horizon as:

$$J^f = \|\delta z(t_f)\|_{Q_f}^2 + \int_{t_0}^{t_f} \|\delta u(t)\|_R^2 + \|\delta z(t)\|_Q^2 dt, \tag{5}$$

where $[t_0, t_f]$ defines the time horizon, $Q, Q_f \in \mathbb{R}^{o \times o}$ are positive semi-definite performance variable cost matrices and $R \in \mathbb{R}^{m \times m}$ is a positive definite control cost matrix. The terms $\delta z(t) = z(t) - z_d(t)$ represent deviations of the performance variables with respect to a desired target trajectory $z_d(t)$, and $\delta u(t) = u(t) - u_d(t)$ represents deviations from a desired target input $u_d(t)$. For example, in trajectory tracking tasks $z_d(t)$ defines the reference trajectory (and typically $u_d(t) = 0$), and in regulation tasks $z_d(t) = z_0$, $u_d(t) = u_0$ where z_0 and u_0 are the desired equilibrium values.

The FEM-based soft robot constrained optimal control problem is then formulated as:

$$\begin{aligned} \underset{q^f, v^f, u}{\text{minimize}} \quad & J^f, \\ \text{subject to} \quad & M^f \dot{v}^f = P^f - F^f(q^f, v^f) + H^f(q^f)u, \\ & \dot{q}^f = v^f, \\ & u \in \mathcal{U}, \quad z \in \mathcal{Z}, \quad z = C_z^f \begin{bmatrix} v^f \\ q^f \end{bmatrix}. \end{aligned} \tag{6}$$

A common approach for solving these types of optimal control problems is to transcribe the finite-horizon OCP into

a nonlinear programming problem that can be solved with existing numerical algorithms. This problem can then be used for closed-loop feedback control by repeatedly solving the optimization problem in a receding horizon fashion.

However, practical implementations of receding horizon control schemes require the OCP to be solved in real-time, which is challenging in this context for several reasons. First, the FEM model (1) is nonlinear and therefore the OCP contains non-convex constraints. Second, even the simple case where the optimization problem is a convex quadratic program has a computational complexity that is $\mathcal{O}(T(\mathcal{N} + m)^3)$, where T is the number of time steps in the OCP horizon [21]. This generally precludes the direct use of FEM models for optimization-based control of soft robots since fine spatial discretizations (a large number of nodes $\mathcal{N}$) are required for high-fidelity modeling. In fact, it is not uncommon for the full state dimension n^f to be on the order of thousands to tens of thousands, making it impossible to solve the OCP in real-time with existing state-of-the-art optimization algorithms.

Our proposed approach to address this computational challenge is to leverage model order reduction techniques to derive a high-fidelity (but low-dimensional) approximation to the original soft robot FEM model. This reduced order model can then be directly used for real-time soft robot control.

III. REDUCED ORDER MODELING FOR SOFT ROBOTS

Principled model order reduction techniques have been developed to derive high-fidelity reduced order models (ROMs) from high-dimensional models. For nonlinear systems, the model order reduction process generally consists of two successive steps. The first is to project the high-dimensional model onto a reduced-order subspace, and the second is to define an efficient approach for approximately evaluating the model's high-dimensional nonlinear terms.

A. Projection-based Model Order Reduction

In this work we consider projection-based methods, which are a general class of model order reduction methods that project the high-dimensional model onto a reduced order subspace and include widely-used approaches such as balanced truncation and proper orthogonal decomposition (POD) [13]. In particular, we use POD to define a Galerkin projection specified by a projection matrix UU^T, where $U \in \mathbb{R}^{3\mathcal{N} \times r}$ is an orthogonal basis matrix and r is the dimension of the reduced-order subspace (typically $r \ll 3\mathcal{N}$). This projection is applied to the high-dimensional FEM position and velocity vectors to define reduced order position and velocity vectors:

$$q = U^T(q^f - q_{\text{ref}}^f), \quad v = U^T(v^f - v_{\text{ref}}^f), \quad (7)$$

and to approximately reconstruct the original states by:

$$q^f \approx Uq + q_{\text{ref}}^f, \quad v^f \approx Uv + v_{\text{ref}}^f, \quad (8)$$

where $q_{\text{ref}}^f, v_{\text{ref}}^f \in \mathbb{R}^{3\mathcal{N}}$ are constant reference states that are used to better condition the basis matrix U. In particular, we select q_{ref}^f and v_{ref}^f to correspond to a static equilibrium (i.e. $v_{\text{ref}}^f = 0$) of the soft robot.

The reduced order model is then defined by projecting the FEM model (1) onto the reduced order subspace:

$$M\dot{v} = P - U^T F^f(Uq + q_{\text{ref}}^f, Uv + v_{\text{ref}}^f)$$
$$+ U^T H^f(Uq + q_{\text{ref}}^f)u, \quad (9)$$
$$\dot{q} = v + v_{\text{ref}},$$

with $M = U^T M^f U$, $P = U^T P^f$, and $v_{\text{ref}} = U^T v_{\text{ref}}^f$. Additionally, with the combined reduced order state $x = [v^T, q^T]^T \in \mathbb{R}^n$ (where $n = 2r$) and reference state $x_{\text{ref}}^f = [v_{\text{ref}}^{f\,T}, q_{\text{ref}}^{f\,T}]^T \in \mathbb{R}^n$, the reduced order measurement and performance models are given by:

$$y = C_y x + y_{\text{ref}}, \quad z = C_z x + z_{\text{ref}}, \quad (10)$$

where $C_y = C_y^f V$ and $C_z = C_z^f V$ with V defined as $V = \text{blkdiag}(U, U)$, and $y_{\text{ref}} = C_y^f x_{\text{ref}}^f$ and $z_{\text{ref}} = C_z^f x_{\text{ref}}^f$ are constants. Crucially, this projection results in a ROM (9) with combined dimension n, which can be orders of magnitude smaller than the original FEM model (1) of dimension n^f.

B. Nonlinear Model Reduction

While the ROM (9) has a reduced dimension, evaluation of the nonlinear terms is still computationally expensive because of their dependence on the high-dimensional state (e.g. the evaluation of the internal force $U^T F^f(Uq+q_{\text{ref}}^f, Uv+v_{\text{ref}}^f)$ scales as $\mathcal{O}(n^f)$). In this work we address this by approximating the nonlinear terms by reduced order piecewise affine functions, resulting in a piecewise affine ROM.

Specifically, the piecewise affine approximations are generated by considering N linearization points (q_i, v_i), which are represented in the high-dimensional space by $q_i^f = Uq_i + q_{\text{ref}}^f$ and $v_i^f = Uv_i + v_{\text{ref}}^f$. Around each of these linearization points, the internal forces F^f are approximated by a first-order Taylor expansion:

$$F^f(q^f, v^f) \approx F^f|_{q_i^f, v_i^f} + \frac{\partial F^f}{\partial q^f}|_{q_i^f, v_i^f}(q^f - q_i^f)$$
$$+ \frac{\partial F^f}{\partial v^f}|_{q_i^f, v_i^f}(v^f - v_i^f).$$

The internal force terms from (9) can then be written as:

$$U^T F^f(Uq + q_{\text{ref}}^f, Uv + v_{\text{ref}}^f) \approx$$
$$F_i + K_i(q - q_i) + D_i(v - v_i),$$

with reduced order terms:

$$F_i = U^T F^f|_{q_i^f, v_i^f}, \quad K_i = U^T \frac{\partial F^f}{\partial q^f}|_{q_i^f, v_i^f} U,$$

$$D_i = U^T \frac{\partial F^f}{\partial v^f}|_{q_i^f, v_i^f} U.$$

The input matrix H^f in (9) is then simply approximated by the zeroth-order Taylor series expansion:

$$H^f(Uq + q_{\text{ref}}^f) \approx H^f(q_i^f).$$

With these simplifications the ROM (9) can be approximated in the vicinity of $x_i = [v_i^T, q_i^T]^T$ by:

$$\dot{x} = A_i x + B_i u + d_i, \quad (11)$$

where:

$$A_i = \begin{bmatrix} -M^{-1}D_i & -M^{-1}K_i \\ I & 0 \end{bmatrix}, \quad B_i = \begin{bmatrix} M^{-1}H_i \\ 0 \end{bmatrix},$$
$$d_i = \begin{bmatrix} M^{-1}(P - F_i + K_i q_i + D_i v_i) \\ v_{\text{ref}} \end{bmatrix}.$$

Combined, these N linearization points provide a global approximation of (9) via the piecewise affine ROM:

$$\dot{x} = \{A_i x + B_i u + d_i, \quad i = \arg\min_j \|x - x_j\|_W, \quad (12)$$

where $W \in \mathbb{R}^{n \times n}$ is a positive semi-definite weighting matrix that defines a distance metric for determining the nearest linearization point. For example, we choose $W =$ blkdiag$(0, I)$ for the soft robot in Figure 1 since the internal forces are heavily dependent on the robot's configuration q^f.

Since the matrices A_i, B_i, and d_i are of reduced order, the ROM (12) can easily be stored in memory, efficiently used for simulation, and its Jacobians can be trivially extracted for use in optimization-based control. A discrete-time version of this model can be defined by:

$$x_{k+1} = g(x_k, u_k) := \begin{cases} A_{i,d}x_k + B_{i,d}u_k + d_{i,d}, \\ i = \arg\min_{j} \|x_k - x_j\|_W, \end{cases} \quad (13)$$

where $A_{i,d}, B_{i,d}$ and $d_{i,d}$ are the discretized reduced order matrices, which can also be computed *a priori*.

Not only is the use of piecewise affine models common for nonlinear control applications, it is also a popular approach in the model order reduction community and is known as the trajectory piecewise linear (TPWL) method [22].

C. Building Piecewise Affine ROM

This section proposes an automated approach for the development of the piecewise affine ROM (12), specifically the computation of the reduced order basis matrix U and the selection of the linearization points (q_i, v_i). This process relies on simulations of the high-dimensional FEM model (1), but can be done entirely *offline*.

As mentioned in Section III-A, the basis matrix U that defines the reduced order subspace is computed via POD, a data-driven method for compressing high-dimensional physics models. In this work, POD is implemented by simulating the FEM model (1) to collect a set of "snapshots", which can include the soft robot's configuration q^f, velocity v^f, and acceleration $\dot{v}^f$, and implicitly defines a basis that characterizes the robot's behavior. A principal component analysis of the snapshots is then used to identify the reduced order subspace. In particular, the subspace is selected by defining a "snapshot matrix" S with columns corresponding to snapshots, and then computing a singular value decomposition $S = \bar{U}\Sigma\bar{V}$. The basis matrix U is then defined by taking the r columns of $\bar{U}$ associated with the r largest singular values. The dimension of the subspace, r, is typically chosen to be as small as possible while still providing good approximation accuracy. In practice, a commonly used heuristic for quantifying the approximation accuracy is the "energy" of the truncated singular values [13].

The linearization points used to define the piecewise affine ROM (12) are also determined via an *offline* data-driven procedure. In particular, at each time step of a FEM simulation a linearization point is added to the ROM if the reduced order state predicted by the ROM diverges too significantly from the FEM result. In other words, the ROM is built incrementally over the course of the simulation.

For good ROM accuracy, the FEM simulation that is used for POD snapshot collection and linearization point selection should sufficiently cover the full range of possible robot motions. To ensure sufficient data is collected, a simple yet effective approach is to apply an open-loop control sequence in the FEM simulation that approximately spans the range of possible actuations for the robot. Specifically, we choose this sequence through Latin hypercube sampling of the soft robot's admissible actuations. A summary of the automated

procedure for developing the piecewise affine ROM is given in Algorithm 1, where **FEM** simulates the FEM model (1) over one time step, **POD** computes the reduced order basis U, **project** corresponds to the projection (7), and **ROM** corresponds to simulating the piecewise affine ROM (12).

Algorithm 1 Defining Piecewise Affine ROM (Offline)

1: **procedure** DEFINEROM(T_{sim}, r, W_q, W_v, η)
2:     $\{u_k\}_{k=0}^{T_{\text{sim}}} \leftarrow$ **LatinHypercubeSample**(T_{sim})
3:     $\mathcal{X}^f = \{(q_0^f, v_0^f, 0)\}$
4:     **for** $k = \{0, \ldots, T_{\text{sim}}\}$ **do**
5:         $(q_{k+1}^f, v_{k+1}^f) \leftarrow$ **FEM**(u_k, q_k^f, v_k^f)
6:         $\mathcal{X}^f \leftarrow \mathcal{X}^f \cup (q_{k+1}^f, v_{k+1}^f, v_{k+1}^f - v_k^f)$
7:     $U \leftarrow$ **POD**($\mathcal{X}^f$, r)
8:     $\mathcal{P} = \{(q_0^f, v_0^f)\}$
9:     **for** $k = \{0, \ldots, T_{\text{sim}}\}$ **do**
10:         $(q_{k+1}^f, v_{k+1}^f) \leftarrow$ **FEM**(u_k, q_k^f, v_k^f)
11:         $(q_{k+1}, v_{k+1}) \leftarrow$ **project**(q_{k+1}^f, v_{k+1}^f)
12:         $(\tilde{q}_{k+1}, \tilde{v}_{k+1}) \leftarrow$ **ROM**($u_k, q_k, v_k, \mathcal{P}$)
13:         **if** $\|q_{k+1} - \tilde{q}_{k+1}\|_{W_q} + \|v_{k+1} - \tilde{v}_{k+1}\|_{W_v} > \eta$ **then**
14:             $\mathcal{P} \leftarrow \mathcal{P} \cup (q_k^f, v_k^f)$
    return U, $\mathcal{P}$

IV. REDUCED ORDER OPTIMAL CONTROL

We now leverage the ROM (12) to formulate a reduced order OCP to approximately solve the original OCP (6). An output feedback control scheme is then defined which consists of three components: (1) the reduced order OCP, (2) a reduced order state estimator, and (3) a reduced order feedback control law. In this scheme, the reduced order OCP optimizes a reduced order trajectory that the soft robot should follow and the state estimator incorporates measurements to provide an estimate of the robot's current (reduced order) state. The feedback control law then uses the state estimate to drive the robot to track the optimized trajectory.

A. Reduced Order Optimal Control Problem

A discretized optimal reduced order trajectory $(\mathbf{x}^*, \mathbf{u}^*)_k = (\{x_i^*\}_{i=k}^{k+T}, \{u_i^*\}_{i=k}^{k+T-1})$ for the robot is defined over a finite horizon T by solving a reduced order approximation of the high-dimensional OCP (6):

$$
\begin{aligned}
\underset{x,u}{\text{minimize}} \quad & \|\delta z_{k+T}\|_{Q_f}^2 + \sum_{j=k}^{k+T-1} \|\delta u_j\|_R^2 + \|\delta z_j\|_Q^2, \\
\text{subject to} \quad & x_{i+1} = g(x_i, u_i), \\
& x_k = x_{0,k}, \\
& u_i \in \mathcal{U}, \quad z_i \in \mathcal{Z}, \quad z_i = C_z x_i + z_{\text{ref}},
\end{aligned}
\tag{14}
$$

where $i = k, \ldots, k + T - 1$, and $x_{0,k}$ is the initial state at time step k. This finite horizon problem is then solved in a receding horizon fashion to define the optimal trajectory over an arbitrarily long horizon. Specifically, the OCP is initialized at time $k = 0$ by setting $x_{0,0} = \hat{x}_0$, where $\hat{x}_0$ is the reduced order state estimate. The OCP is then recursively solved every $T_r < T$ time steps (i.e. at time steps $k = T_r, 2T_r, \ldots$ over the receding horizon $[k, k + T]$) by setting $x_{0,k} = x_k^*$, where x_k^* is the optimal state from the previous solution (computed at time step $k - T_r$).

We solve the nonconvex OCP (14) using sequential convex programming (SCP), by solving a sequence of quadratic program (QP) approximations until convergence. Specifically, we use a slightly modified version of [23]. Crucially, since the ROM is constructed such that $n \ll n^f$ this approach can enable real-time control.

B. Reduced Order Controller and State Estimator

The reduced order OCP (14) defines an optimized *open-loop* reduced order trajectory that the robot should follow. A simple output feedback control scheme is now defined to drive the robot to track this trajectory.

To estimate the robot's current reduced order state from measurements a reduced order state estimator is defined as:

$$\hat{x}_k = g(\hat{x}_{k-1}, u_{k-1}) + L_k(y_k - C_y g(\hat{x}_{k-1}, u_{k-1})), \quad (15)$$

where $L_k \in \mathbb{R}^{n \times p}$ is the estimator gain, y_k is the robot measurement, and u_{k-1} is the previous control input. We choose the gain L_k to be the extended Kalman filter gain based on the discrete-time ROM dynamics (13).

The control applied to the robot is then computed using a reduced order linear feedback control law:

$$u_k = u_k^* + G_k(\hat{x}_k - x_k^*), \quad (16)$$

where $G_k \in \mathbb{R}^{m \times n}$ is the controller gain matrix, and (x_k^*, u_k^*) is the optimized state-input pair computed by the OCP (14). To define the controller gains G_k we propose a simple and computationally efficient method based on gain scheduling:

$$G_k = \big\{ G_i, \quad i = \arg\min_j \|x_k^* - x_j\|_W, \quad (17)$$

where the gains G_i are the discrete-time LQR gains at each linearization point (q_i, v_i), which can be computed *a priori*. In particular, the gains G_i are computed using the discrete-time reduced order dynamics matrices $A_{i,d}$ and $B_{i,d}$, and positive semi-definite cost matrix $Q_G \in \mathbb{R}^{n \times n}$ and positive definite control cost matrix $R_G \in \mathbb{R}^{m \times m}$ (e.g. $Q_G = C_z^T Q C_z$ and $R_G = R$).

The formulation of the proposed control scheme has a couple of practical advantages. First, since the OCP (14) defines the initial condition based on the previous solution the next solve can be started as soon as the previous solve is completed. For example, suppose the optimized trajectory $(\mathbf{x}^*, \mathbf{u}^*)_k$ has already been computed at time step k, then the OCP problem associated with time step $k + T_r$ can be solved starting at time step k since $x_{k+T_r}^*$ is already known. If the model is discretized with a sampling time of h this gives hT_r seconds to solve the next OCP. Additionally, this choice of initialization can help with warm-starting of the SCP algorithm. Second, even though the optimal trajectory computed by the OCP is discretized with a sampling time h, the state estimator and control law can be operated at a higher frequency by simply interpolating the optimal trajectory $(\mathbf{x}^*, \mathbf{u}^*)_k$.

V. SIMULATION RESULTS

We now compare our method against two alternative approaches. First, we use a reduced order model predictive control (ROMPC) scheme based on a *linearized* FEM model [17] to demonstrate the significant benefits of using a nonlinear model. Second, we demonstrate that our method performs comparably to a data-driven Koopman operator-based control

scheme [10] without suffering common drawbacks of data-driven methods, such as a loss in generality from building problem or task-specific models. These comparisons are discussed in more detail in Section V-B.

We compare these approaches in simulation using the elastomer "Diamond" soft robot shown in Figure 1. This robot is fixed at the base and is actuated by controlling the tension in four cables that are attached at the robot's "elbows". The "top" of the robot will be referred to as the "end effector". We use the open-source SOFA framework [20], [24] for finite element model-based simulations, and the mesh used in our experiments can be found in the SOFA Soft Robots plugin[2]. The FEM model used to simulate the Diamond robot consists of $\mathcal{N} = 1628$ nodes (i.e. $n^f = 9768$), the nonlinear internal forces $F^f(q^f, v^f)$ are defined by a linear stress-strain law, and gravity is assumed to be the only external force P^f. The measurement model includes the position and velocity of the end effector and the four "elbows" of the robot (i.e. $y \in \mathbb{R}^{30}$), and additive Gaussian measurement noise is included in the FEM simulation. We consider a control application where the performance variable is the position of the robot's end effector (i.e. $z = [x_{ee}, y_{ee}, z_{ee}]^T$). In particular, a constrained optimal control problem is formulated to drive the end effector position (x_{ee}, y_{ee}) to track a desired trajectory $(x_{d,ee}, y_{d,ee})$ subject to position constraints, as shown in Figures 2 and 3. Note that this type of dynamic control problem cannot be addressed with kinematics-based controllers, and the addition of constraints precludes the use of many data-driven control methods.

A. Controllers

For the proposed approach we construct a piecewise affine ROM for the Diamond robot using the procedure in Algorithm 1. In particular, we use acceleration snapshots in the POD step to define a reduced order subspace with dimension $r = 21$ (i.e. $n = 42$), we choose $v_{\text{ref}}^f = 0$, and select q_{ref}^f to be the equilibrium configuration of the robot with no actuation. For selecting the linearization points we choose the threshold parameter η (with $W_q = 0, W_v = I$) such that a total of $N = 642$ points (q_i, v_i) are selected. Additionally, the piecewise affine ROM is defined with $W = \text{blkdiag}(0, I)$, such that only the robot's configuration q is used to select the nearest linearization point. The reduced order OCP (14) is then defined with a horizon of $T = 5$, a rollout horizon of $T_r = 2$, and the ROM is discretized with a sampling time of $h = 0.05$ seconds. By interpolating the optimal trajectory computed by the OCP, the feedback controller (16) and state estimator (15) operate with a sampling time of 0.01 seconds.

The ROMPC controller [17] uses a ROM generated by linearizing the FEM model (1) around the point x_{ref}^f and then reusing the POD projection from our proposed controller. We consider the same horizons T, T_r and sampling time h as in our controller, and the ROMPC feedback controller and state estimator also operate with a sampling time of 0.01 seconds by interpolating the optimized trajectory. Finally, for the Koopman operator-based controller [10] we build a model for the performance variables z with a delay $d = 1$ and all monomials of maximum degree 2 to define a lifted state of dimension $N_K = 66$ and a sampling time of $h = 0.05$ seconds. The data used to build this model came from the same FEM model simulation used to define our proposed controller.
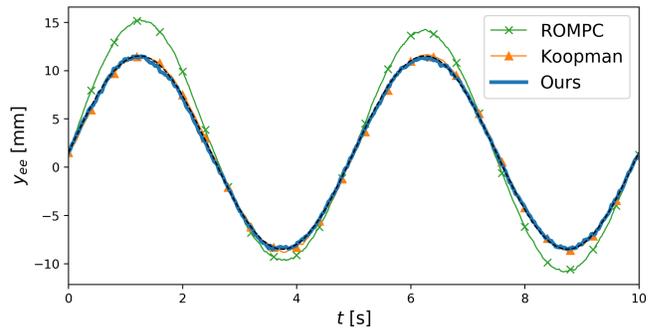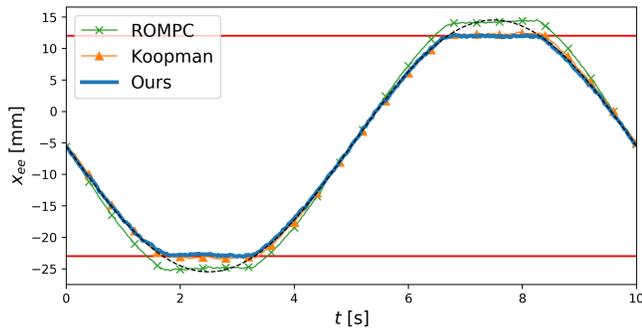
Fig. 2: FEM model simulation results for trajectory tracking with the Diamond soft robot. The green line represents the ROMPC controller [17], orange is the Koopman controller [10], blue is our proposed control scheme. The red lines represent constraints, and the black dashed line indicates the desired trajectory (which is partially infeasible due to the constraints). These results demonstrate the ability of our proposed controller to allow the soft robot to perform dynamic tasks such as constrained trajectory tracking.
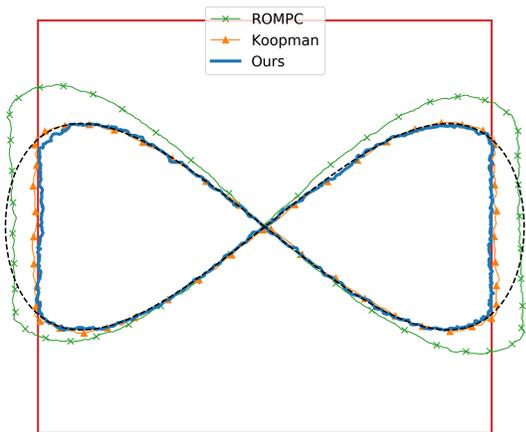

Fig. 3: FEM model simulation results for the Diamond soft robot with the ROMPC [17], Koopman [10], and our proposed controller. The black dashed line indicates the desired trajectory, and the interior of the red box is the admissible region defined by the constraints. Note that in addition to the advantages discussed in Section V-B, our proposed approach performs comparably to the Koopman approach with respect to tracking and constraint satisfaction. The poor performance of ROMPC motivates the need for nonlinear model-based controllers.

TABLE I: A comparison of the mean square error (MSE) and the time spent solving QPs in each controller using Gurobi [25], where the reported value for our method considers the *cumulative* sum of all QP solve times in the SCP algorithm (on a 2.5 GHz Intel Core i5 processor with 8GB of RAM). These results show our FEM-based optimal control scheme achieves state-of-the-art performance and is real-time capable.

Method	Tracking Error	Computation		
	MSE (mm^2)	Mean (ms)	Min. (ms)	Max. (ms)
ROMPC	5.80	10	9	18
Koopman	0.17	21	17	38
Ours	0.07	17	10	34

B. Results

Simulation results of the Diamond FEM model are presented for each controller in Figures 2 and 3. As can be seen, the ROMPC scheme, which uses a *linearized* ROM, offers poor tracking and severely violates the constraints. In contrast, our approach and the Koopman controller offer good tracking performance and generally satisfy the constraints, highlighting the importance of capturing the soft robot's nonlinear behavior.

To demonstrate the computational requirements of each controller we report the amount of time spent solving QPs (which is the most significant computational component) in Table I. Note that the ROMPC and Koopman controllers only solve one QP at a time while the proposed SCP approach may require multiple QPs to be solved. These results show that the proposed FEM model-based control scheme is real-time capable.

Our proposed approach offers similar performance to the Koopman operator-based controller in this simulation (see Table I), while offering several advantages. First, the Koopman approach only models the behavior of a *prespecified* choice of the performance variables z, while the FEM model in our approach captures the robot's entire state (independent of the choice of z) and can therefore be used for any number of different control problems. In contrast to our approach, the Koopman controller is also restricted to only consider outputs z that are a subset of the measured variables y. Second, the dimension of the Koopman model does not scale well with the number of measured variables. For example, including all the measured variables y (i.e. setting $z = y \in \mathbb{R}^{30}$) would result in a model of dimension $N_K = 2145$ (using the same delay $d = 1$ and all monomials with maximum degree 2). Third, the Koopman controller must operate at whatever frequency the model is discretized at (and the frequency the QP is solved at), while our controller can be operated at much higher frequencies by subsampling the optimized trajectory.

VI. CONCLUSION

In this work we propose a novel approach for model-based optimal control of soft robots based on high-fidelity nonlinear finite element models. Notably, computational efficiency is achieved by defining a reduced order piecewise affine model to approximate the high-dimensional FEM model. The proposed controller enables output feedback constrained optimal control problems to be addressed, including setpoint and trajectory tracking problems. Simulation results are used to demonstrate the performance of the proposed approach in comparison to a state-of-the-art data-driven and linear FEM model-based constrained control method.

Future Work: Future work includes validation of the proposed approach through hardware experiments as well as several extensions, including the use of more advanced nonlinear model reduction techniques (e.g. the energy-conserving sampling and weighting (ECSW) method [26]), application to different types of soft robots (e.g. pneumatically actuated robots, which have a distributed actuation force), the ability to handle scenarios where the robot makes and breaks contact with the environment, and whether data-driven techniques could be used to augment the current approach. Additional theoretical analysis is also needed to study whether performance, stability, and constraint satisfaction guarantees can be made within the proposed framework.

References

[1] D. Rus and M. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.

[2] T. G. Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control strategies for soft robotic manipulators: A survey," *Soft Robotics*, vol. 5, no. 2, pp. 149–163, 2018.

[3] R. J. Webster, III and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *Int. Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010.

[4] C. Della Santina, R. Katzschmann, A. Bicchi, and D. Rus, "Model-based dynamic feedback control of a planar soft robot: Trajectory tracking and interaction with the environment," *Int. Journal of Robotics Research*, vol. 39, no. 4, pp. 490–513, 2020.

[5] I. A. Gravagne, C. D. Rahn, and I. D. Walker, "Large deflection dynamics and control for planar continuum robots," *IEEE Transactions on Mechatronics*, vol. 8, no. 2, pp. 299–307, 2003.

[6] F. Renda, F. Boyer, J. Dias, and L. Seneviratne, "Discrete cosserat approach for multisection soft manipulator dynamics," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1518–1533, 2018.

[7] J. M. Bern, Y. Schnider, P. Banzet, N. Kumar, and S. Coros, "Soft robot control with a learned differentiable model," in *IEEE Int. Conf. on Soft Robotics*, 2020.

[8] M. T. Gillespie, C. M. Best, E. C. Townsend, D. Wingate, and M. D. Killpack, "Learning nonlinear dynamic models of soft robots for model predictive control with neural networks," in *IEEE Int. Conf. on Soft Robotics*, 2018.

[9] 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, 2019.

[10] D. Bruder, B. Gillespie, C. D. Remy, and R. Vasudevan, "Modeling and control of soft robots using the koopman operator and model predictive control," in *Robotics: Science and Systems*, 2019.

[11] C. Duriez, "Control of elastic soft robots based on real-time finite element method," in *Proc. IEEE Conf. on Robotics and Automation*, 2013.

[12] J. M. Bern, P. Banzet, R. Poranne, and S. Coros, "Trajectory optimization for cable-driven soft robot locomotion," in *Robotics: Science and Systems*, 2019.

[13] A. Antoulas, *Approximation of Large-Scale Dynamical Systems*. SIAM, 2005.

[14] M. Thieffry, A. Kruszewski, C. Duriez, and T. Guerra, "Control design for soft robots based on reduced-order model," *IEEE Transactions on Robotics and Automation*, vol. 4, no. 1, pp. 25–32, 2019.

[15] R. K. Katzschmann, M. Thieffry, O. Goury, A. Kruszewski, T. Guerra, C. Duriez, and D. Rus, "Dynamically closed-loop controlled soft robotic arm using a reduced order finite element model with state observer," in *IEEE Int. Conf. on Soft Robotics*, 2019.

[16] M. Thieffry, A. Kruszewski, T. Guerra, and C. Duriez, "Trajectory tracking control design for large-scale linear dynamical systems with applications to soft robotics," *IEEE Transactions on Control Systems Technology*, pp. 1–11, 2019.

[17] J. Lorenzetti and M. Pavone, "Error bounds for reduced order model predictive control," in *Proc. IEEE Conf. on Decision and Control*, 2020, in Press.

[18] J. Lorenzetti, A. McClellan, C. Farhat, and M. Pavone, "UAV aircraft carrier landing using CFD-based model predictive control," in *AIAA Scitech Forum*, 2020.

[19] O. Goury and C. Duriez, "Fast, generic and reliable control and simulation of soft robots using model order reduction," *IEEE Transactions on Robotics*, 2018.

[20] E. Coevoet, T. Morales-Bieze, F. Largilliere, Z. Zhang, M. Thieffry, M. Sanz-Lopez, B. Carrez, D. Marchal, O. Goury, J. Dequidt, and C. Duriez, "Software toolkit for modeling, simulation, and control of soft robots," *Advanced Robotics*, vol. 31, no. 22, pp. 1208–1224, 2017.

[21] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 267–278, 2010.

[22] M. Rewienski and J. White, "A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 2, pp. 155–170, 2003.

[23] R. Bonalli, A. Cauligi, A. Bylard, and M. Pavone, "GuSTO: guaranteed sequential trajectory optimization via sequential convex programming," in *Proc. IEEE Conf. on Robotics and Automation*, 2019.

[24] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik, and S. Cotin, "SOFA: A multi-model framework for interactive physical simulation," in *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*, 2012.

[25] Gurobi Optimization, LLC, *Gurobi Optimizer Reference Manual*, 2020, available at http://www.gurobi.com.

[26] 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," *Int. Journal for Numerical Methods in Engineering*, vol. 102, no. 5, pp. 1077–1110, 2015.