Decentralized Control of Distributed Actuation in a Segmented Soft Robot Arm

Azadeh Doroudchi, Sachin Shivakumar, Rebecca E. Fisher, Hamid Marvi, Daniel Aukes, Ximin He, Spring Berman, and Matthew M. Peet

Abstract—Continuum robot manipulators present challenges for controller design due to the complexity of their infinite-dimensional dynamics. This paper develops a practical dynamics-based approach to synthesizing state feedback controllers for a soft continuum robot arm composed of segments with local sensing, actuation, and control capabilities. Each segment communicates to its two adjacent neighboring segments, requiring a tridiagonal feedback matrix for decentralized controller implementation. A semi-discrete numerical approximation of the Euler-Bernoulli beam equation is used to represent the robot arm dynamics. Formulated in state space representation, this numerical approximation is used to define an \( H_\infty \) optimal control problem in terms of a Bilinear Matrix Inequality. We develop three iterative algorithms that solve this problem by computing the tridiagonal feedback matrix which minimizes the \( H_\infty \) norm of the map from disturbances to regulated outputs. We confirm through simulations that all three controllers successfully dampen the free vibrations of a cantilever beam that are induced by an initial sinusoidal displacement, and we compare the controllers’ performance.

I. INTRODUCTION

Continuum robots \([31, 32]\) have high-dimensional configuration spaces, which can be leveraged to achieve versatile functionality over a wide variety of configurations. The implementation of decentralized control architectures \([2, 25]\) in continuum robots would enable scalability of the robot design, minimize expensive communication and power overhead, and increased robustness to partial failure. Furthermore, continuum robots composed of soft materials would exhibit high structural compliance in response to environmental inputs that can enhance the robot’s functionality.

Soft continuum robots with decentralized controllers can be used in manufacturing, surgery, and other applications requiring flexible manipulators that can operate safely in close proximity to humans. They can also be used to perform unstructured manipulation and locomotion tasks in uncertain, dynamic environments. Furthermore, novel soft materials such as smart hydrogels \([11]\), which can dramatically change volume and other properties in response to stimuli such as temperature, pH, and chemicals, present the possibility of constructing soft continuum robots with on-demand dynamic control of local properties through continuous sensing and actuation that is distributed throughout the robot. Such robots could offer new capabilities through self-regulated adaptive reconfiguration.

Challenges remain in the design of decentralized controllers for soft continuum robots. While there are many scalable and compliant soft robot designs, these designs are typically model-independent or use simplified models which do not accurately reflect either the nonlinear dynamics of highly deformable robots or the practical issues of sensor and actuator design and placement \([15, 17, 23]\). In addition, most soft robot designs still require complex sensing, control, and actuation to achieve even low-dimensional configuration spaces. Dynamic models of continuum robots would facilitate a variety of control techniques. However, many of the control-oriented models developed for these types of robots have thus far been governed by kinematic equations describing rigid links \([8, 21, 33]\), and hence are not useful for designing feedback controllers when both the forces produced by the actuators and the motion of the robot are distributed throughout the structure. While dynamic models have been formulated, e.g. a Partial Differential Equation (PDE) model of bending in a hyper-redundant continuum robotic arm \([13]\), their complexity often prevents their practical implementation in controller design and motion planning \([31, 32]\).

Work on the control of vibrations in beams \([6, 7]\) is closely related to the decentralized control strategies that we design in this paper. The optimal sensor/actuator placement problem has been well-studied in vibration control; see, e.g. \([5, 16]\). In addition, there has been significant research on the question of how to construct stabilizing decentralized feedback laws for a given network and, furthermore, whether there are necessary and sufficient conditions for the existence of such local feedback laws. The largest class of systems for which we know the answer to this question are those systems which are quadratically invariant \([18, 22]\). While testing quadratic invariance is known to be NP-hard, in practice, testing quadratic invariance under sparsity constraints for reasonably-sized systems is not difficult and furthermore, certain well-studied sparsity patterns are known to be quadratically invariant, with the most well-known case being when the controller is diagonal or both the controller...
and plant are upper- or lower-tridiagonal. Unfortunately, however, the tridiagonal sparsity constraint generated by discretization of beam-type equations (with zeros everywhere except the diagonal and first off-diagonal elements) is not quadratically invariant. Because the decentralized control problem with tridiagonal structure is difficult, the literature on vibration control of beams focuses on the case of diagonal decentralization, in which neighboring controllers do not communicate with each other.

We are interested in designing decentralized controllers with tridiagonal structure for soft robot arms. Since the tridiagonal structure is not quadratically invariant, we instead consider the non-convex Bilinear Matrix Inequality (BMI) formulation of the problem and design algorithms to find solve this BMI directly using iteration and gradient descent.

Many algorithms have been developed for finding local solutions to BMI problems, several focusing on Branch and Bound [9], [27], [28], [29], [30]. However, many such global optimization algorithms have high computational complexity, making them impractical for the large state spaces induced by spatial discretization of a PDE [19]. For high dimensional problems, Yamada et al. suggest a modified triangle-covering based algorithm which reduces the computational cost [34]. Unfortunately, however, this approach is restricted to a class of Bilinear Matrix Inequalities (BMIs) that does not include the decentralized controller synthesis problem. The method proposed in [14] and the rank minimization approach in [12] will both typically converge to a local optimum given an initial feasible controller. Other approaches involve linearization of the BMI [10]. However many of these methods, as shown in [24], can fail to converge to even locally optimal solutions.

In this work, we develop a novel practical approach to designing decentralized state feedback controllers for soft continuum robot arms composed of segments with local sensing, actuation, and control capabilities. The control objective is to regulate the robot arm’s displacement in the presence of disturbance inputs; i.e., to dampen its disturbance-induced vibrations. Our approach does not require the use of a complex nonlinear model that describes the infinite-dimensional dynamics of the robot. Instead, we represent the robot arm’s spatiotemporal dynamics using a semi-discrete numerical approximation of the Euler-Bernoulli beam PDE (Section II). This numerical approximation is formulated as an ordinary differential equation (ODE) state space model for implementation in linear matrix inequality (LMI) methods. The state space model is used to define an $H_{\infty}$ optimal control problem in terms of a BMI (Section III). We present three algorithms of increasing stability and performance that solve this problem by computing the tridiagonal feedback matrix which minimizes the $H_{\infty}$ norm of the map from disturbances to regulated outputs (Section IV). Finally, we simulate the controllers computed by each algorithm for the case of a cantilever beam composed of hydrogel material and compare their performance (Section V). We conclude with a discussion of the simulation results and directions for future work (Section VI).

II. DYNAMIC MODEL

The robot arm is constructed from $N$ identical cylindrical segments that are arranged in a series configuration, as illustrated in Fig. 1. We assume that each segment is equipped with sensing, actuation, control, and communication elements. We also assume that each segment can apply local torques and can measure local deformations. For example, when a segment is composed of force-sensitive conductive hydrogel [11], local deformations can be sensed from a resulting change in resistivity across the segment, and this change in resistivity provides an electrical signal which can be used as an output to a local feedback controller. The local controller can then induce a current, which causes local temperature changes in the segment that produce prescribed deformations and resulting torques.

In this decentralized sensing and actuation model of a soft robot arm, we likewise impose a decentralized communication architecture with a similar chain topology, meaning that each segment can exchange state measurements only with adjacent segments.

A. Model definition

To model the segmented robot arm, we will use a discretized version of the cantilever beam, wherein the beam is composed of material that is elastic, homogeneous, and isotropic.

The material has Young’s modulus $E$ and mass density $\rho$. The beam has length $L$ and a constant circular cross-section with diameter $D$, area $A_c$, and area moment of inertia $I$ about the neutral axis. We specify that $L \geq 20D$, in which case the Euler-Bernoulli beam model yields an accurate approximation of the robot arm dynamics when the material properties satisfy the given assumptions [3], see my comment please.

Let $w(x, t)$ be the transverse displacement (see Fig. 1) of point $x \in [0, L]$ on the beam at time $t \in [0, T]$, where $T$ is a specified final time. The PDE describing a one-dimensional unforced Euler-Bernoulli beam is given by

$$b_c^2 \frac{\partial^4 w}{\partial x^4} + \frac{\partial^2 w}{\partial t^2} = 0, \quad b_c^2 = \frac{EI}{\rho A_c}. \quad (1)$$

We define boundary conditions for this model that describe a cantilever beam, in which the deflection and slope of the
fixed end and the bending moment and shear force at the free end are all set to zero:

\[
\begin{align*}
    w(0, t) &= 0, \quad \frac{\partial^2 w}{\partial x^2}(L, t) = 0, \\
    \frac{\partial w}{\partial x}(0, t) &= 0, \quad \frac{\partial^3 w}{\partial x^3}(L, t) = 0,
\end{align*}
\]  
where \(t \in [0, T]\).

B. State space representation

To represent the segmented arm, we construct a discretized approximation of the continuum PDE beam model (1), (2), which results in a set of linear ODEs. As in [4], we apply the central finite difference method with second-order accuracy to obtain a semi-discrete space approximation of model (1), (2) (discrete in the spatial coordinate \(x\) and continuous in time \(t\)). We define \(h = L/N\) as the length of each segment and \(x_j\) as the \(x\) position of the right boundary of segment \(j \in \{1, ..., N\}\). Then we have that \(x_j = jh\) for each segment \(j\), and we define \(x_0 = 0\). For the boundary conditions, we also introduce two external points \(x_{-1} = -h\) and \(x_{N+1} = L + h\). The semi-discretization version of model (1) is then given by the following system of \(N\) linear equations, each describing the dynamics of the transverse displacement of point \(x_j\) on the beam at time \(t \in [0, T]\):

\[
\begin{align*}
    \ddot{w}(x_j, t) &= -\frac{b^2}{h^2}[w(x_{j+2}, t) - 4w(x_{j+1}, t) + 6w(x_j, t) - 4w(x_{j-1}, t) + w(x_{j-2}, t)), \\
    w(x_{N+1}, t) &= w(x_N, t), \quad w(x_{N+2}, t) = w(x_{N-1}, t).
\end{align*}
\]  

Note that the dynamics of each segment’s displacement is approximated as a function of its own displacement and that of the two closest segments on either side. The boundary conditions (2) are expressed as

\[
\begin{align*}
    w(x_0, t) &= 0, \quad w(x_{-1}, t) = -w(x_1, t), \\
    w(x_{N+1}, t) &= w(x_N, t), \quad w(x_{N+2}, t) = w(x_{N-1}, t).
\end{align*}
\]  

We define the system state variables as \(w(x_j, t)\), \(\dot{w}(x_j, t)\), \(j = 1, ..., N\) and arrange them in the vectors \(w = [w(x_1, t) \ w(x_2, t) ... w(x_N, t)]^T\), \(\dot{w} = [\dot{w}(x_1, t) \ \dot{w}(x_2, t) ... \ \dot{w}(x_N, t)]^T\). The system of equations (3) and the boundary conditions (4) can then be represented in state space form as follows:

\[
\begin{align*}
    \begin{bmatrix}
        \ddot{w} \\
        \dot{w}
    \end{bmatrix} &=
    \begin{bmatrix}
        A_{11} & A_{12} \\
        A_{21} & A_{22}
    \end{bmatrix}
    \begin{bmatrix}
        w \\
        \dot{w}
    \end{bmatrix},
\end{align*}
\]  

where

\[
\begin{align*}
    A_{11} &= [0]_{N \times N}, \quad A_{12} = I_{N \times N}, \\
    A_{21} &= -\frac{b^2}{h^2}A_h, \quad A_{22} = [0]_{N \times N},
\end{align*}
\]  
and the matrix \(A_h \in \mathbb{R}^{N \times N}\) is defined as

\[
A_h =
\begin{bmatrix}
    5 & -4 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\
    -4 & 6 & -4 & 1 & 0 & 0 & 0 & \cdots & 0 \\
    1 & -4 & 6 & -4 & 1 & 0 & 0 & \cdots & 0 \\
    0 & 1 & -4 & 6 & -4 & 1 & 0 & \cdots & 0 \\
    0 & 0 & 1 & -4 & 6 & -4 & 1 & \cdots & 0 \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
    0 & \cdots & 0 & 0 & 1 & -4 & 6 & -4 & 1 \\
    0 & \cdots & 0 & 0 & 0 & 0 & 1 & -4 & 6 & -3 \\
    0 & \cdots & 0 & 0 & 0 & 0 & 0 & 1 & -3 & 2
\end{bmatrix}.
\]  

Including inputs and outputs, we obtain a state space representation given by

\[
\begin{bmatrix}
    \dot{w} \\
    \ddot{w}
\end{bmatrix} = A \begin{bmatrix}
    w \\
    \dot{w}
\end{bmatrix} + B_u, \quad y = C \begin{bmatrix}
    w \\
    \dot{w}
\end{bmatrix} + Du,
\]  
in which the system control input is denoted by \(u \in \mathbb{R}^N\) and the output by \(y \in \mathbb{R}^2\). The \(A, B, C,\) and \(D\) matrices are defined as

\[
\begin{align*}
    A &= \begin{bmatrix}
        A_{11} & A_{12} \\
        A_{21} & A_{22}
    \end{bmatrix}_{2N \times 2N}, \quad B = \begin{bmatrix}
        [0]_{N \times N} \\
        I_{N \times N}
    \end{bmatrix}_{2N \times N}, \\
    C &= I_{2N \times 2N}, \quad D = \begin{bmatrix}
        [0]_{N \times N} \\
        I_{N \times N}
    \end{bmatrix}_{2N \times N}.
\end{align*}
\]  

In Section III, we discuss how the decentralized communication constraint leads to structural constraints on the gain from input to output.

III. CONTROLLER SYNTHESIS

In this section, we use the linear ODE model developed in Section II-B to define a decentralized control problem assuming local full-state feedback. We first impose the mild assumption that the uncontrolled system is neutrally stable and controllable. To define the \(H_{\infty}\)-optimal control problem, we use the standard regulator framework, yielding the 2-input, 2-output system representation \(R \in \mathbb{R}^{7N \times 7N}\) as:

\[
R = \begin{bmatrix}
    A & B_1 & B_2 \\
    C_1 & D_{11} & D_{12} \\
    C_2 & D_{21} & D_{22}
\end{bmatrix},
\]  
where

\[
\begin{align*}
    B_1 &= \begin{bmatrix}
        B & 0
    \end{bmatrix}_{2N \times 4N}, \quad B_2 = B, \\
    C_1 &= \begin{bmatrix}
        C & 0
    \end{bmatrix}_{3N \times 2N}, \quad C_2 = C, \\
    D_{11} &= \begin{bmatrix}
        D & 0
    \end{bmatrix}_{3N \times 4N}, \quad D_{12} = \begin{bmatrix}
        D & I
    \end{bmatrix}_{3N \times N}, \\
    D_{21} &= \begin{bmatrix}
        D & I
    \end{bmatrix}_{2N \times 4N}, \quad D_{22} = D.
\end{align*}
\]  

Because \(C_2 = I\), the control problem is one of full-state feedback. The control problem, then, is to find the feedback controller \(u = Ky, K \in \mathbb{R}^{N \times 2N}\), that minimizes the \(H_{\infty}\) norm of the map from disturbing inputs \(u\) to regulated outputs \(y\). However, we now add a communication constraint in which we specify the structure of \(K\) to be tridiagonal. This structure implies that the moment generated by each
segment is based only on measurements of its own state and the states of its two neighboring segments. The desired tridiagonal matrix structure is shown below:

\[
\begin{bmatrix}
    k_{1,1} & k_{1,2} & 0 & \cdots & 0 & 0 & 0 \\
    k_{2,1} & k_{2,2} & k_{2,3} & \cdots & 0 & 0 & 0 \\
    0 & k_{3,2} & k_{3,3} & \cdots & 0 & 0 & 0 \\
    \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
    0 & 0 & 0 & \cdots & k_{n-2,n-2} & k_{n-2,n-1} & 0 \\
    0 & 0 & 0 & \cdots & k_{n-1,n-2} & k_{n-1,n-1} & k_{n-1,n} \\
    0 & 0 & 0 & \cdots & 0 & 0 & k_{n,n}
\end{bmatrix}
\]

Defining \( T \) as the set of matrices with structure (9), we denote the set of admissible controller gains by \( S \), where

\[ S = \{ [K_1 \ K_2] \mid K_1, K_2 \in T \}. \]

This allows us to represent the controller information constraint as \( K \in S \). By applying Kalman–Yakubovich–Popov (KYP) lemma, the optimization problem is written in (10) where \( '*' \) is used to represent symmetric elements in the matrix inequality. Specifically, the \( H_\infty \)-optimal decentralized controller \( K \in S \) is the solution to the optimization problem (10).

\[
\text{minimize } \gamma > 0 \text{ such that } \\
\begin{bmatrix}
    (A + B_2 K)^T P + P (A + B_2 K) & *^T & *^T \\
    B_1^T P & -\gamma I & *^T \\
    (C_1 + D_{12} K) & D_{11} & -\gamma I
\end{bmatrix} < 0
\]

for some \( K \in S \) and \( P > 0 \).

IV. PROPOSED ALGORITHMS FOR SOLVING THE BMI

The optimization problem (10) is a Bilinear Matrix Inequality (BMI) in the matrix variables \( K \) and \( P \). Solving BMIs is known to be an NP-hard problem [26]. In this section, we evaluate three possible algorithms for obtaining locally optimal solutions to this BMI, two based on iteration and one based on gradient descent.

A. Initialization

In all three algorithms, we require an initial feasible solution to the BMI. Furthermore, the selection of initial values can significantly influence convergence to an optimal solution. Unfortunately, however, there are no canonical rules for finding an initial feasible solution. In our algorithm, we address this problem as follows. Under the assumption that the nominal system is controllable, the following LMI has solution \( P > 0 \):

\[
\text{controllability: } A^T P + P A - B B^T < 0 \tag{11}
\]

We use this solution as an estimate of the initial value of \( P (P_0) \). Using this \( P_0 \) to find the initial value of \( K (K_0) \) is problematic, however, because of the additional constraint \( K \in S \). To resolve this, we initialize \( K \) without the sparsity constraint and solve the resulting LMI version of (10) for \( P \), and then use this as our new estimate of \( P_0 \). Given this new value of \( P_0 \), we solve the resulting LMI version of (10) for \( K \) with the relaxed constraint that only the last row of \( K \) is required to have the sparsity structure \( K \in S \) of the tridiagonal matrix (9). Using this \( K \), we solve the LMI version of (10) for \( P \). This procedure is repeated by progressively constraining more rows of \( K \) to have the structure of the corresponding row of matrix (9) until the entire matrix \( K \) has the desired tridiagonal structure.

We have developed the following three algorithms to obtain an \( H_\infty \) optimal solution for \( K \). The algorithms all use the initialization procedure described above.

B. Iterative optimization algorithm

Algorithm 1 is a standard iteration-based method used to solve a bilinear system of equations. It is similar to our initialization procedure for the variables \( P \) and \( K \). Initializing a value for \( P (P_0) \) yields the LMI from (10), which is solved by optimizing over \( K \). Afterward, we fix \( K \) in (10) and optimize over \( P \). These steps are repeated until the values of \( K \) and \( P \) converge to optimal values, at which point the change in \( \gamma \) is minimal.

This algorithm has two drawbacks: it does not converge for certain initial values of \( P (P_0) \), especially if the \( A \) matrix is numerically ill-conditioned, and the solution for \( K \) could have a large magnitude that makes the feedback controller physically impractical to implement. However, imposing additional constraints on the magnitude of \( K \) could potentially cause the \( H_\infty \) norm to diverge. We next propose two modified versions of this algorithm that address these problems.

C. Modified iterative optimization algorithm

Algorithm 1, depending on the choice of \( P_0 \), can end up oscillating between suboptimal solutions for \( K \). This was observed to happen for poor choices of \( P_0 \). To reduce these oscillations, we define \( P \) and \( K \) at each iteration as weighted averages of their current values and their optimized values, obtained by solving the optimization problem (10) during the current iteration. The weight factor \( \alpha \) is chosen to be a value between 0 and 1. The \( \alpha \) value can be selected to produce small changes in the solution between iterations, thus preventing the solution from making large jumps in the non-convex subspace. An \( \alpha \) close to 0 would result in very small changes in \( P \) and \( K \) over successive iterations.

D. Gradient descent algorithm

Although both Algorithms 1 and 2 are quick to converge, they do not converge at all when the matrices \( A \) and \( B \) are numerically ill-conditioned. In addition, the solution for \( K \) computed by these procedures often has a magnitude that is too large for implementation in practice. We address this problem in Algorithm 3 by splitting optimization problem (10) into two optimization problems with LMI constraints, shown below. The difference here is that we can directly restrict changes in the solution over successive iterations and also limit the values taken by the variables. We redefine the optimization variables as \( \Delta K \in \mathbb{R}^{N \times 2N} \) and \( \Delta P \in \mathbb{R}^{2N \times 2N} \), whose \( L_\infty \)-norms are constrained to be small.
Algorithm 1 Standard iterative algorithm

1: Choose a small $\epsilon > 0$. Initialize $P$ to $P_0$.
2: while $|\gamma_k - \gamma_{k-1}| > \epsilon$ do
3: Use the last known value for $P$.
4: Solve for $K$ in problem (10), minimizing $\gamma$.
5: Use the solution for $K$ in the next step.
6: Solve for $P$ in problem (10), minimizing $\gamma$.
7: $\gamma_k$ is the minimized value of $\gamma$ in step 6.
8: $k = k + 1$
9: end while

Algorithm 2 Modified iterative algorithm

1: Choose a small $\epsilon > 0$ and $\alpha \in (0, 1)$. Initialize $P$ to $P_0$.
2: while $|\gamma_k - \gamma_{k-1}| > \epsilon$ do
3: Use the last known value for $P$.
4: Solve for $K$ in problem (10), minimizing $\gamma$.
5: $K_{k+1} = K_k + \alpha (K - K_k)$
6: Use $K_{k+1}$ as the current value of $K$.
7: Solve for $P$ in problem (10), minimizing $\gamma$.
8: $P_{k+1} = P_k + \alpha (P - P_k)$
9: $\gamma_k$ is the optimal value of $\gamma$ in step 7.
10: $k = k + 1$
11: end while

in order to prevent large changes in $K$ and $P$ between iterations.

minimize $\gamma_0 > 0$ such that $\|\Delta K\| < \epsilon_1$ and

$$
\begin{bmatrix}
(A + B_2K_0)^T P + P(A + B_2K_0) & *^T \\
B_1^T P & -\gamma_0 I \\
(C_1 + D_{12}K_0) & D_{11} - \gamma_0 I
\end{bmatrix} < 0
$$

(12)

for some $K_0 \in S$, where $K_0 \equiv K + \Delta K$.

minimize $\gamma_0 > 0$ such that $\|\Delta P\| < \epsilon_2$ and

$$
\begin{bmatrix}
(A + B_2K)^T P_a + P_a(A + B_2K) & *^T \\
B_1^T P_a & -\gamma_b I \\
(C_1 + D_{12}K) & D_{11} - \gamma_b I
\end{bmatrix} < 0
$$

(13)

for some $P_a > 0$, where $P_a \equiv P + \Delta P$.

In these two problems, $\epsilon_1$ and $\epsilon_2$ are small positive numbers.

The optimization procedure is performed alternately over $\Delta K$ and $\Delta P$ as follows such that $\gamma$ converges to a local minimum. At the beginning of each iteration, problem (12) is solved for $\Delta K$ using the current values of $K$ and $P$, and then $K$ is increased by $\Delta K$. Next, problem (13) is solved for $\Delta P$, and $P$ is increased by $\Delta P$.

V. SIMULATION RESULTS

We validated our numerical approximation of the beam model and investigated the performance of our decentralized state feedback controllers in simulation. YALMIP [20], an optimization toolbox for MATLAB with the MOSEK solver [1], was used to solve the optimization problems in Algorithms 1, 2, and 3. The beam model was simulated using the MATLAB lsim command with $N = 40$ segments and the parameters listed in Table I, where $E$ and $\rho$ are defined for hydrogel material\(^1\).

A. Validation of semi-discrete approximation of beam model

In order to evaluate the accuracy of the numerical approximation (3), (4), we compare it to the analytical solution of the cantilever Euler-Bernoulli beam model (1), (2). The initial conditions of the beam model were set to

$$w(x,0) = \sin \left( \frac{3\pi}{2L} x \right), \quad \frac{\partial w}{\partial t}(x,0) = 0, \quad x \in [0,L].$$

(14)

For these initial conditions, the solution to the beam model (1), (2) can be obtained using the separation of variables method:

$$w(x,t) = \sin \left( \frac{3\pi}{2L} x \right) \cos \left( \frac{9\pi^2 b t}{4L^2} \right), \quad t \in [0,T], \quad x \in [0,L].$$

(15)

This solution describes the first mode shape of the beam. In the simulations, we set $T = 20$.

Figures 2(a),(b) plot the vibrations of the beam over time from the analytical solution (15) and the numerical approximation, respectively. Figure 2(c) plots the error between the analytical solution and the numerical approximation. Although this error grows over time due to numerical approximation error propagation, it remains relatively small (absolute maximum error of 0.072 after 200 iterations at second 2) compared to the maximum amplitude of the beam vibrations within the first few seconds of the simulation, when the controllers effectively damp the vibrations (see next section). Thus, the numerical approximation is sufficiently accurate for use in our optimization methods to synthesize

\(^1\)N-isopropylacrylamide, variously abbreviated PNIPA, PNIPAAm, NIPA, PNIPAA or PNIPAm

TABLE I: Beam material and geometric properties

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E$</td>
<td>Young’s modulus at 25$^\circ$C</td>
<td>5.0</td>
<td>kPa</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Mass density</td>
<td>1.1</td>
<td>g/cm$^3$</td>
</tr>
<tr>
<td>$D$</td>
<td>Diameter</td>
<td>5.0</td>
<td>cm</td>
</tr>
<tr>
<td>$L$</td>
<td>Length</td>
<td>1.0</td>
<td>m</td>
</tr>
<tr>
<td>$A_c$</td>
<td>Cross-section area</td>
<td>19.6</td>
<td>cm$^2$</td>
</tr>
<tr>
<td>$I$</td>
<td>Area moment of inertia</td>
<td>30.7</td>
<td>cm$^4$</td>
</tr>
</tbody>
</table>
the controllers. In addition, pay attention that these simulations are done for a low number \((N = 40)\) comparing to usual numerical approximations.

B. Comparison of optimal decentralized controllers for damping beam vibrations

Decentralized state feedback controllers were synthesized with Algorithms 1, 2, and 3, and the beam dynamics were simulated for each controller using the numerical approximation (6). All the variables were initialized using the procedure described in Section IV-A.

Figures 3a, 4a, and 5a plot the evolution of the closed-loop \(H_\infty\) norm bound, \(\gamma\), over the execution of each algorithm when the optimization is performed alternately over the variables \(K\) and \(P\) during each iteration. Figures 3b, 4b, and 5b display the resulting closed-loop beam response for each controller given the initial condition (14). These figures show that all controllers successfully dampen the beam vibrations that are induced by the initial beam displacement within the first 5 seconds. From the convergence rates of the plots in Fig. 3a, 4a, and 5a, it is evident that Algorithm 1 is the least computationally intensive procedure, followed by Algorithm 2 and then by Algorithm 3. This is because \(K\) is least constrained in Algorithm 1, which therefore permits large changes in \(K\) between iterations and hence has the fastest convergence, followed by the other two algorithms. Algorithm 2 shows superior performance to Algorithm 1, in that it converged to a controller with a smaller \(H_\infty\) norm bound \((\gamma = 1.05, \text{versus } \gamma = 1.42 \text{ for Algorithm 1})\) at the expense of a slight increase in computational demands. Algorithm 3 converged to the highest \(H_\infty\) norm bound \((\gamma = 2.31)\) of the three methods since the controller gain values were subject to additional constraints. However, the controller computed by this algorithm would be the most feasible one to implement in practice, since the constraints limit the magnitudes of the controller gains.

VI. CONCLUSIONS

In this paper, we developed three algorithms for synthesizing a decentralized controller for the discretized Euler-Bernoulli beam model by solving an \(H_\infty\) optimal control problem. We found that when the system matrix is numerically ill-conditioned, which is a common property of discretized beam models, convergence of the \(H_\infty\) norm is not always guaranteed. In addition, we found that iterative approaches are in general sensitive to the initial selections of \(P\) and \(K\). The modifications proposed in the algorithms solved these problems of convergence and sensitivity for the discretized beam model. The iterative and modified iterative methods quickly reach a converged \(H_\infty\) norm value, but they do not guarantee convergence for different selections of initial \(P\) and \(K\). The gradient descent approach, while slightly slower at reaching a converged \(H_\infty\) norm value, is less sensitive to different choices of initial \(P\) and \(K\). It provides a bounded solution for the controller gains, which is often a necessity in physical systems.

In future work, we plan to develop decentralized controllers for beam models that more accurately describe the dynamics of a soft continuum robot arm composed of hydrogel. We will design these controllers to produce diverse types of arm motions and deformations that are useful for manipulation and locomotion tasks.

VII. ACKNOWLEDGMENTS

The authors gratefully acknowledge Prof. Marc Mignolet for helpful discussions on structural dynamics of beam models and Roozbeh Khodambashi for information about hydrogel functional properties.
### References


Fig. 5: (a) $H_\infty$ norm bound converging in the two alternating steps of Algorithm 3. (b) Closed-loop response of the simulated beam with initial conditions (14) and the controller from Algorithm 3.


