

Lie Group Formulation and Sensitivity Analysis for Shape Sensing of Variable Curvature Continuum Robots with General String Encoder Routing

Andrew L. Orekhov^{1,2}, Elan Z. Ahronovich¹, and Nabil Simaan¹

Abstract—This paper considers a combination of actuation tendons and measurement strings to achieve accurate shape sensing and direct kinematics of continuum robots. Assuming general string routing, a methodical Lie group formulation for the shape sensing of these robots is presented. The shape kinematics is expressed using arc-length-dependent curvature distributions parameterized by modal functions, and the Magnus expansion for Lie group integration is used to express the shape as a product of exponentials. The tendon and string length kinematic constraints are solved for the modal coefficients and the configuration space and body Jacobian are derived. The noise amplification index for the shape reconstruction problem is defined and used for optimizing the string/tendon routing paths, and a planar simulation study shows the minimal number of strings/tendons needed for accurate shape reconstruction. A torsionally stiff continuum segment is used for experimental evaluation, demonstrating mean (maximal) end-effector absolute position error of less than 2% (5%) of total length. Finally, a simulation study of a torsionally compliant segment demonstrates the approach for general deflections and string routings. We believe that the methods of this paper can benefit the design process, sensing and control of continuum and soft robots.

Index Terms—Continuum robots, soft robots, shape sensing, Lie group methods, human-robot collaboration

I. INTRODUCTION

CONTINUUM robots achieve infinitely varied shapes governed by their elastic equilibrium conformations rather than being solely determined by the values of their active joints, as shown in Fig. 1. As a result, the direct kinematics of these robots is corrupted with a large level of uncertainty due to structural deflections. Since active joint values alone are not sufficient for achieving accurate direct kinematics, an additional sensing source is needed. This need motivated efforts on shape sensing for continuum and soft robots, which focused on using extrinsic and/or intrinsic sensing.

Extrinsic sensing relies on use of external metrology (e.g. magnetic trackers [1] and computer vision [2–5]). Intrinsic sensing relies on sensors that augment joint-level data to produce shape estimates. The use of PVDF bimorph sensors for shape sensing of soft actuators under planar bending was explored in [6] along with considerations for the number of

sensors and their spacing to avoid Runge’s phenomenon when fitting a modal function series to estimate the shape. Others focused heavily on the use of fiber Bragg grating (FBG) sensors for shape sensing [7–9]. A mechanics model combined with moment sensing was presented in [10] to estimate the shape of whiskers and in [11] for Mckibben muscle actuated continuum robots, and a mechanics model was combined with passive cable displacement in [12] for shape sensing. Finally, string potentiometers were explored in [11], [13], [14] for shape sensing of continuum robots under planar or constant curvature shapes.

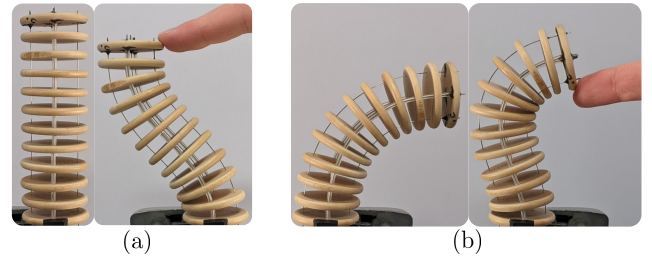


Fig. 1. A continuum segment subject to passive deflections (a) starting in a straight configuration (b) starting with a bent configuration. This paper addresses the problem of sensing the deflected shape using string encoders.

Despite previous works, there are two key technological and theoretical gaps that motivate this work. First, there is a need for a low cost solution to the problem of shape sensing. While use of FBG sensor arrays is possible, it is not an economical solution and seems to be at odds with the low cost of soft and continuum robots. Therefore, this paper is focused on presenting a modeling and design optimization framework for low-cost solutions to shape sensing of continuum robots using intrinsic sensing. This solution assumes that the continuum/soft robot is equipped with actively actuated tendons (henceforth referred to simply as *tendons*) and a set of passive measurement strings (henceforth referred to simply as *strings*). These strings can be in the form of string potentiometers (akin to spring-loaded lanyards) or string encoders. The combined set of length measurements of these tendons and strings is used to estimate the shape of these robots. The second gap stems from the lack of a theoretical study focused on the effect of string routing on the shape sensing problem and design guidelines for string routing optimization to minimize the sensitivity of the shape sensing problem to measurement noise. To address this gap, we present a Lie group mathematical formulation that enables us to answer these core questions:

¹Department of Mechanical Engineering, Vanderbilt University, Nashville, TN, USA. e-mail: aorekhov@andrew.cmu.edu, (elan.z.ahronovich, nabil.simaan}@vanderbilt.edu

²Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA

This work was supported by NSF award #CMMI-1734461 and by Vanderbilt University internal funds. A. Orekhov was partially supported by an NSF Graduate Research Fellowship under #DGE-1445197.

- What is the impact of string routing on the kinematic sensitivity of the shape sensing to sensory noise?
- How many sensors are needed per a continuum segment? and, if one has a fixed number of sensors, where should they be placed to minimize the shape error or to minimize the end effector error?

While addressing these questions, we present a Lie group formulation for the string routing. We use a modal representation for the curvature distribution along the robot length and derive the general string routing kinematics relating variations in bending shape to string length. While one may obtain the string routing kinematics as in [15] for calibration of continuum robots, we believe that the use of the Lie group formulation presented here leads to an elegant formulation of the body Jacobian. Finally, we adapt the noise amplification index presented by [16] within the context of robot calibration to guide the design of string routing.

Although accurate mechanics models have been presented in prior work on continuum robots, in this work we are interested in a formulation that could be used for online control, so we intentionally pursue a kinematics-based method, using lower-order polynomials that lead to either closed-form expressions or relatively cheap to compute optimization problems. Our kinematic deflection sensing approach does not require knowledge of applied loads from actuation, string encoders, or the environment, and it provides insight at the initial design stage for choosing string routings.

While our sensing approach does not rely on a mechanics model, our use of modal representation of curvature is related to continuum mechanics models for flexible [17] and continuum robots [18], [19]. Lie-group based kinematic integration formulations using the Magnus expansion have been used in other works for solving more general Cosserat rod mechanics models [18], [20] and a Lie-group finite-element formulation of beam dynamics was presented in [21]. Modal representations have also been used in prior works to parameterize the shape of hyper-redundant robots [22] and continuum/soft robots [23], with some works using modal functions to parameterize curvature/strain [19], [24] as we do here.

This paper is focused on shape sensing of larger scale hyper-redundant and continuum/soft robots, e.g. [25–27], but we believe the approach presented herein could also be applied to smaller continuum robots used in minimally-invasive surgery, e.g. [28], [29]. This could potentially be achieved by mounting the string encoder housings remotely away from the robot body, as is already done with actuation units in surgical continuum robots.

The contributions of this paper lies in addressing the two key gaps identified above for single-segment continuum robots. In doing so, we present: 1) a Lie group kinematics formulation for shape sensing with general string routing and modal shape deflections, 2) an approach for optimizing string routing to improve the shape sensing kinematic conditioning with recommendations for designing string routings, and 3) an experimental validation and a simulation study of the proposed approaches on a torsionally stiff continuum robot and a torsionally compliant soft robot, respectively.

II. LIE GROUP KINEMATIC FORMULATION

This section presents the kinematic formulation and modal representation for parameterizing the variable curvature backbone shape and deriving the resulting pose of local frames along the continuum segment. This parametrization will be used for describing general string routing and the associated kinematic Jacobians for shape sensing.

A. Central Backbone Kinematics

Referring to Fig. 2, we assume a robot with a total arc length L and define the arc length coordinate s as shown. We also assume that the robot's central backbone has a high slenderness ratio consistent with the assumption of negligible shear strains. With this assumption, the shape of the backbone can be described by its curvature distribution along its arc length s in three directions, $\mathbf{u}(s) = [u_x, u_y, u_z]^T \in \mathbb{R}^3$. For a given location, s , a local frame $\mathbf{T}(s)$ is assigned with its z -axis tangent and pointing in the direction of arc length growth and its two other axes in the backbone's local cross section:

$$\mathbf{T}(s) = \begin{bmatrix} {}^0\mathbf{R}_t(s) & {}^0\mathbf{p}(s) \\ 0 & 1 \end{bmatrix} \in SE(3), \quad s \in [0, L] \quad (1)$$

where ${}^0\mathbf{R}_t(s)$ and ${}^0\mathbf{p}(s)$ are expressed in the world frame $\{0\}$ ¹. As the backbone changes its local curvature, this local frame undergoes a twist $\boldsymbol{\eta}(s)$ defined with the angular velocity preceding the linear velocity. The set of local frames associated with local curvatures $\mathbf{u}(s)$ has a corresponding twist distribution $\boldsymbol{\eta}(s)$. Describing the twist $\boldsymbol{\eta}(s)$ in its local frame $\mathbf{T}(s)$, we can write $\boldsymbol{\eta}(s) = [\mathbf{u}(s)^T, \mathbf{e}_3^T]^T \in \mathbb{R}^6$ where $\mathbf{e}_3 = [0, 0, 1]^T$ denotes the local tangent unit vector.

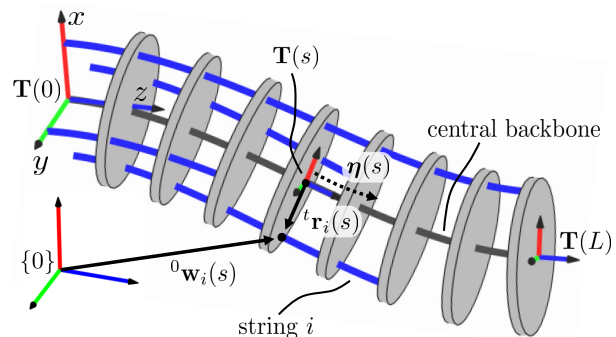


Fig. 2. Variables used in our kinematic model to describe variable curvature deflections and general string routing.

As the frame $\mathbf{T}(s)$ undergoes the body twist $\boldsymbol{\eta}$, it satisfies the following differential equation [30]:

$$\begin{aligned} \mathbf{T}'(s) &= \mathbf{T}(s)\hat{\boldsymbol{\eta}}(s), \quad \hat{\boldsymbol{\eta}}(s) = \begin{bmatrix} \hat{\mathbf{u}}(s) & \mathbf{e}_3 \\ 0 & 0 \end{bmatrix} \in se(3) \\ \hat{\mathbf{u}}(s) &= \begin{bmatrix} 0 & -u_z(s) & u_y(s) \\ u_z(s) & 0 & -u_x(s) \\ -u_y(s) & u_x(s) & 0 \end{bmatrix} \in so(3) \end{aligned} \quad (2)$$

where $(\cdot)'$ denotes the derivative with respect to s and the hat operator $(\hat{\cdot})$ forms the standard matrix representations of

¹The notation ${}^a\mathbf{y}$ designates vector \mathbf{y} described in frame $\{A\}$ and ${}^a\mathbf{R}_b$ is the orientation of frame $\{B\}$ with respect to frame $\{A\}$

$so(3)$ and $se(3)$ from their vector forms \mathbf{u} and $\boldsymbol{\eta}$, respectively. We also define the adjoint representation of $se(3)$, which will be used below for computing Jacobians:

$$\text{ad}(\boldsymbol{\eta}(s)) = \begin{bmatrix} \hat{\mathbf{u}}(s) & \mathbf{0} \\ \hat{\mathbf{e}}_3 & \hat{\mathbf{u}}(s) \end{bmatrix} \quad (3)$$

In the following analysis, we choose to represent the curvature distribution $\mathbf{u}(s)$ as a weighted sum of polynomial functions (similar to [18]). We denote the polynomial functions as $\phi_x(s)$, $\phi_y(s)$, and $\phi_z(s)$ and the weights as \mathbf{c}_x , \mathbf{c}_y , and \mathbf{c}_z , for the x , y , and z directions, respectively. The curvature distribution then takes the following form:

$$\begin{aligned} \mathbf{u}(s) &= \begin{bmatrix} \phi_x^T \mathbf{c}_x \\ \phi_y^T \mathbf{c}_y \\ \phi_z^T \mathbf{c}_z \end{bmatrix} = \begin{bmatrix} \phi_x^T & 0 & 0 \\ 0 & \phi_y^T & 0 \\ 0 & 0 & \phi_z^T \end{bmatrix} \begin{bmatrix} \mathbf{c}_x \\ \mathbf{c}_y \\ \mathbf{c}_z \end{bmatrix} \\ &= \boldsymbol{\Phi}(s)\mathbf{c}, \quad \boldsymbol{\Phi}(s) \in \mathbb{R}^{3 \times m}, \quad \mathbf{c} \in \mathbb{R}^m \end{aligned} \quad (4)$$

where the columns of $\boldsymbol{\Phi}(s)$ form a *modal shape basis*, and \mathbf{c} is a vector of constant *modal coefficients*.

The modal description of the curvature distribution offers a description of a variety of variable curvature deflections using a finite set of modal coefficients. This also provides simplified kinematic expressions for computing the workspace, solving for the shape using the string encoder measurements, and designing the string encoder routing to improve the numerical conditioning of the shape sensing problem.

For a given configuration \mathbf{c} , the frames $\mathbf{T}(s)$ are found by integrating (2). A number of Lie group integration methods could be used for this, as reviewed in [31], but here we use an approach based on the Magnus expansion because both fourth and sixth order expansions can be computed efficiently and because of the Magnus expansion's large convergence bound [31]. After integration with a Magnus expansion method (or another Lie group integration method), the spatial curve is given as a product of matrix exponentials [20]:

$$\mathbf{T}(s) = \mathbf{T}(0) \prod_{i=0}^k e^{\boldsymbol{\Psi}_i}, \quad \boldsymbol{\Psi}_i \in se(3) \quad (5)$$

Details on computing $\boldsymbol{\Psi}_i$ can be found in [20], [31]. We will show below that because we use a modal shape basis, Lie group integration is not needed to solve the shape sensing problem (i.e. determining the modal coefficients), however, once the configuration \mathbf{c} is determined, (2) must be integrated to compute the robot's forward kinematics and Jacobian, as shown in Section II-F. In this paper, we compute (5) with small step sizes (100 points along the backbone) at a rate of ~ 45 Hz in an unoptimized MATLAB implementation. Larger steps can be used in practice for faster computation times [20], but we use a fine discretization here to avoid introducing additional integration error into this study.

B. Modal Shape Basis with Chebyshev Polynomials

Although any set of modal shape functions could be used to form $\boldsymbol{\Phi}$, (e.g. Euler curves [32], [33], monomials [15], Legendre polynomials [17], or trigonometric functions), in this paper we use Chebyshev polynomials of the first kind since each polynomial is bounded by ± 1 , which provides improved

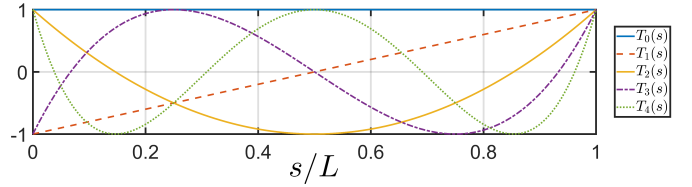


Fig. 3. The first five shifted Chebyshev polynomials.

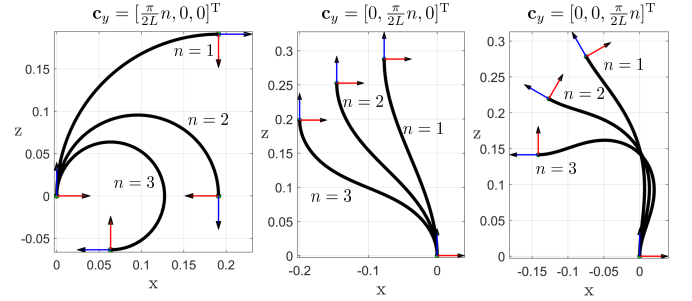


Fig. 4. Deflections generated on a rod with $L = 300$ mm and second order Chebyshev series modal shape basis on the y direction curvature, as given by (8). Deflections were generated by taking $\frac{\pi}{2L}$ steps in each of the three modal coefficient directions.

scaling of the modal coefficients and simplifies computation of the admissible workspace, as described in Section III. The Chebyshev polynomials can be expressed recursively as [34]:

$$\begin{aligned} T_0 &= 1, \quad T_1(x) = x \\ T_n(x) &= 2xT_{n-1}(x) - T_{n-2}(x), \quad n = 2, 3, \dots \end{aligned} \quad (6)$$

where $T_n(x)$, $x \in [-1, 1]$ is the n^{th} degree Chebyshev polynomial. Since the arc length coordinate is given by $s \in [0, L]$ and $T_n(x)$ is defined for $x \in [-1, 1]$, the following coordinate transformation is used:

$$x(s) = \frac{2s - L}{L} \quad (7)$$

We then evaluate $T_n(x(s))$ via (6). In the remainder of this paper, we refer to the polynomials as simply $T_n(s)$ with the transformation (7) implied. The first five Chebyshev polynomials, shifted to $s \in [0, L]$, are shown in Fig. 3.

As an example, to represent a y direction curvature with a second-order Chebyshev series, the modal shape basis and modal coefficients would be given by:

$$\begin{aligned} u_y(s) &= \phi_y(s)\mathbf{c}_y \\ \phi_y^T(s) &= [T_0 \quad T_1(s) \quad T_2(s)]^T, \quad \mathbf{c}_y \in \mathbb{R}^3 \end{aligned} \quad (8)$$

where the first three Chebyshev polynomials, shifted to $s \in [0, L]$ are given by:

$$T_0 = 1, \quad T_1(s) = \frac{2s - L}{L}, \quad T_2(s) = \frac{8s^2}{L^2} - \frac{8s}{L} + 1 \quad (9)$$

Figure 4 illustrates shapes generated by these first three Chebyshev polynomials for a planar continuum segment. Of note is that shapes in the T_0 and T_1 directions of the modal shape basis correspond to experimentally observed deflection shapes in tendon-actuated and multi-backbone robots. Shapes

along the T_0 direction correspond to constant curvature deflections exhibited in continuum segments with actuation wires equidistantly distributed about the central backbone. Shapes along the T_1 direction correspond to constant-orientation deflections due to external forces applied to the end disk of a tendon-actuated continuum segment, as shown in Fig. 1.

The shape sensing methods presented below would apply to other choices for Φ as well. For example, a modal shape basis with coupling between the modal coefficients could be used, e.g. [18]. If we choose Φ as the identity matrix, \mathbf{c} corresponds directly to constant curvatures in the x , y , and z directions (see Fig. 4), so the results here also apply to robots modeled with the commonly used constant-curvature assumption [35].

C. General String Routing Kinematics

In [36], a Cosserat-rod based mechanics model was presented for tendon-actuated robots with general tendon routing. In the kinematics model presented here, we use a similar method for describing the string routing. Assuming p strings, the string path, as shown in Fig. 2, is expressed in the moving frame $\mathbf{T}(s)$ and given by:

$${}^t\mathbf{r}_i(s) = [r_{x_i}(s), r_{y_i}(s), 0]^T, \quad i = 1, 2, \dots, p \quad (10)$$

Our kinematic formulation permits any differentiable function for $\mathbf{r}(s)$, but in Sections V and VI we consider constant pitch-radius paths and helical paths. The position of a point along the string/wire rope path in the world frame is given as the vectorial sum of the point $\mathbf{p}(s)$ along the central backbone and the radial vector $\mathbf{r}(s)$, defined in the moving frame:

$${}^0\mathbf{w}_i(s) = {}^0\mathbf{p}(s) + {}^0\mathbf{R}_t(s) {}^t\mathbf{r}_i(s) \quad (11)$$

Noting that vector norms are invariant under rotations, the length of the i^{th} string is given by:

$$\ell_i = \int_0^{s_{a_i}} \|{}^t\mathbf{w}'_i(s)\| ds, \quad i = 1, 2, \dots, p \quad (12)$$

where s_{a_i} designates arc length along the central backbone at which the string is anchored to the spacer disk/end disk. Taking the derivative of (11) with respect to s , substituting (4), and then using ${}^t\mathbf{w}'_i(s) = {}^0\mathbf{R}_t^T {}^0\mathbf{w}'_i(s)$ results in:

$${}^t\mathbf{w}'_i(s) = \mathbf{e}_3 - {}^t\hat{\mathbf{r}}_i(s)\Phi(s)\mathbf{c} + {}^t\mathbf{r}'_i(s), \quad i = 1, 2, \dots, p \quad (13)$$

Deriving the above result also requires (2), which states that ${}^0\mathbf{p}'(s) = {}^0\mathbf{R}_t(s)\mathbf{e}_3$ and ${}^0\mathbf{R}'_t(s) = {}^0\mathbf{R}_t(s)\hat{\mathbf{u}}(s)$.

Recalling that ${}^t\mathbf{r}_i(s)$ is in a moving frame having a body angular velocity $\hat{\mathbf{u}}(s)$, we can visualize (13) as the velocity of a point traversing the string path as the arc length s is increased at a unit speed. This point speed is given as the sum of the induced velocity due to the rotation of the moving frame and the velocity of that point relative to the moving frame due to traversal along the central backbone and the rate of change of the string's radial placement ${}^t\mathbf{r}'_i(s)$. Since (13) is a function only of the modal basis, the string routing function, and the modal coefficients, we can numerically integrate (12) with any quadrature rule (e.g. the trapezoid rule) and avoid the cost of integrating the Lie group differential equation in (2).

The string routings ${}^t\mathbf{r}_i$, $i = 1 \dots p$ can be chosen by the designer, but must each satisfy a geometric constraint that the string path in world frame must not have a cusp at any configuration of the continuum segment. To ensure physically realizable string paths, the local tangent to the actuation string must point in the same direction as the local tangent of the central backbone. It is possible for the angular rate of changes u_x and u_y to be large enough to cause ${}^t\mathbf{w}'_i$ to point in the opposite direction of \mathbf{e}_3 , which causes the string path to change directions. To avoid this scenario, we require that ${}^t\mathbf{w}'_i$ always point in the same direction as \mathbf{e}_3 . Using (13), we obtain the following set of p constraint equations for each string:

$$({}^t\mathbf{w}'_i)^T \mathbf{e}_3 = r_{y_i}(s)u_x(s) - r_{x_i}(s)u_y(s) + 1 > 0 \quad (14)$$

This can be rewritten as constraints on u_x and u_y :

$$u_x(s) \leq \frac{r_{x_i}(s)u_y(s) + 1}{r_{y_i}(s)}, \quad u_y(s) \leq \frac{r_{y_i}(s)u_x(s) + 1}{r_{x_i}(s)} \quad (15)$$

In the coming sections, the string routings will be incorporated into a model that captures their effect on increasing the robustness to noise in string length measurements to changes in the estimated shape of the continuum segment.

D. Solving for the Modal Coefficients

To solve for the shape of the robot, we concatenate (12) for each string together with the string length measurements ℓ^* :

$$\ell(\mathbf{c}) - \ell^* = 0, \quad \ell \in \mathbb{R}^p, \quad \ell^* \in \mathbb{R}^p \quad (16)$$

and solve this system of equations for the modal coefficients \mathbf{c} . We then integrate $\mathbf{T}'(s)$ once to find the backbone pose at any desired arc length. In some cases, for particular choices of ${}^t\mathbf{r}_i$ and Φ , unique closed-form solutions to (16) can be found by explicit integration of (12). This occurs below for the planar case and for the case of robots with high torsional stiffness.

In general however, the system of equations in (16) can be nonlinear in \mathbf{c} and may have multiple solutions, so it must be solved by an iterative numerical method, e.g. Gauss-Newton. A necessary condition for using the Gauss-Newton algorithm is $p \geq m$ where p is the number of strings and m is the number of columns in Φ . The Jacobian needed for each iteration of the Gauss-Newton method is provided in the next section.

E. Configuration Space Jacobian

Since the modal vector \mathbf{c} uniquely defines the shape of the continuum segment for a given modal basis Φ , we use \mathbf{c} as the configuration space variable. We also define the *configuration-space Jacobian* as the Jacobian relating small changes in the string lengths to small changes in the modal coefficients:

$$d\ell = \mathbf{J}_{\ell\mathbf{c}} d\mathbf{c}, \quad \mathbf{J}_{\ell\mathbf{c}} \in \mathbb{R}^{p \times m} \quad (17)$$

Following [19], the i^{th} row of $\mathbf{J}_{\ell\mathbf{c}}$ can be derived as:

$$\frac{d\ell_i}{d\mathbf{c}} = \int_0^{s_{a_i}} \left({}^t\mathbf{r}_i \times \frac{({}^t\mathbf{w}'_i)}{\|{}^t\mathbf{w}'_i\|} \right)^T \Phi ds \quad (18)$$

where we have dropped the dependence on s in the integrand terms for brevity and ${}^t\mathbf{w}'_i$ was given in (13). As with the integral in (12), (18) can be computed with any quadrature rule.

F. Body Jacobian

Next, we define the *body Jacobian* as the Jacobian relating the twists of the moving frame $\mathbf{T}(s)$ expressed in $\mathbf{T}(s)$ (also called body twists) to small changes in the modal coefficients:

$$\boldsymbol{\xi}(s) = \mathbf{J}_{\xi c}(s)dc, \quad \boldsymbol{\xi} \in se(3) \quad (19)$$

where $\boldsymbol{\xi}(s) = [\boldsymbol{\omega}(s)^T, \mathbf{v}(s)^T]^T$ is the instantaneous twist of $\mathbf{T}(s)$ produced by dc , expressed in the body frame $\mathbf{T}(s)$ and with angular velocity followed by linear velocity. The i^{th} column of $\mathbf{J}_{\xi c}(s)$, denoted by $\mathbf{J}_{\xi c}^{[i]}(s)$ is the twist produced by a small change in the i^{th} element of \mathbf{c} , denoted by c_i :

$$\mathbf{J}_{\xi c}^{[i]}(s) = \left(\mathbf{T}^{-1}(s) \frac{\partial}{\partial c_i} (\mathbf{T}(s)) \right)^\vee \in se(3) \quad (20)$$

where $(\cdot)^\vee$ denotes the inverse operation of $(\cdot)^\wedge$.

Referring back to (5), we require the terms $\frac{\partial}{\partial c_i} (e^{\Psi_j})$ to be able to compute $\frac{\partial}{\partial c_i} (\mathbf{T}(s))$. These derivatives are given by:

$$\frac{\partial}{\partial c_i} e^{\Psi_j} = e^{\Psi_j} \text{dexp} \left(\frac{\partial \Psi_j}{\partial c_i} \right) \quad (21)$$

where the *dexp* operator is defined in [37] as:

$$\text{dexp} \left(\frac{\partial \Psi_j}{\partial c_i} \right) = \left(\sum_{k=0}^{\infty} \frac{(-1)^k}{(k+1)!} \text{ad}^k (\Psi_j) \right) \left(\frac{\partial \Psi_j}{\partial c_i} \right) \quad (22)$$

Closed-form expressions for *dexp* for the case of $se(3)$ are also available in [38]. The value of $\frac{\partial \Psi_j}{\partial c_i}$ can be derived symbolically and will vary depending on the order of the Magnus expansion used. Alternatively, it can be estimated via finite difference approximation.

The kinematic expressions defined above provide the equations needed for solving the shape sensing problem and computing the forward/inverse kinematics for a particular continuum robot design and string routing function definition. In Section V we provide experimental validations of this modeling and shape sensing approach.

III. STRING ROUTING OPTIMIZATION

A benefit of our proposed kinematic formulation is that in addition to capturing variable curvature deflections, it provides a designer flexibility in the design of the string routing by allowing non-straight string routings. In some cases, there may be practical mechanical integration considerations that would benefit from non-straight routing, e.g. routing strings around proprioceptive sensing electronics embedded in the continuum structure [39] or routing strings in a tapered path for a segment with a decreasing diameter from base to tip [40]. In this section, we provide considerations for choosing the string routing paths to reduce the propagation of string measurement error to pose error while avoiding ill-conditioned Jacobians.

For a general purpose manipulator, the external loading magnitude and location may not be known *a priori*. For this reason, here we propose a Jacobian-based method to optimize the string routing path ${}^t\mathbf{r}_i(s)$ without assuming any particular loading conditions. We do this by designing ${}^t\mathbf{r}_i(s)$ to improve the numerical conditioning of both the task space and configuration space Jacobians to reduce the upper bound

on error propagation from the string length measurements to errors in the spatial curve $\mathbf{T}(s)$. We validate the approach in simulations and experiments in Sections IV, V, and VI.

We first define the noise amplification index, a design measure used in robot calibration [16]. For an expression $\mathbf{Ax} = \mathbf{b}$ the noise amplification is given by:

$$\aleph(\mathbf{A}) = \frac{\sigma_{min}^2(\mathbf{A})}{\sigma_{max}(\mathbf{A})} \quad (23)$$

where $\sigma_{min}(\mathbf{A})$ and $\sigma_{max}(\mathbf{A})$ are the minimum and maximum singular values of \mathbf{A} , respectively. As shown in [16], the noise amplification index provides a bound on how errors in \mathbf{b} propagate to errors in \mathbf{x} :

$$\|\delta\mathbf{x}\| \leq \frac{1}{\aleph(\mathbf{A})} \|\delta\mathbf{b}\| \quad (24)$$

Within the context of shape sensing, (17) is the mapping relating noise in string length measurement $d\ell$ to a change in modal coefficients dc . Also, (19) provides the mapping relating twist and changes in dc . We solve (17) for dc and substitute into (19), then solve the equation for $d\ell$ to produce:

$$d\ell = \mathbf{J}_{\ell\xi}\boldsymbol{\xi}(s), \quad \mathbf{J}_{\ell\xi} = (\mathbf{J}_{\xi c}\mathbf{J}_{\ell c}^+)^+, \quad (25)$$

where $(+)$ denotes the pseudoinverse. Using (24), (25) has the following noise amplification bound:

$$\|\delta\boldsymbol{\xi}(s)\| \leq \frac{1}{\aleph(\mathbf{J}_{\ell\xi})} \|\delta d\ell\| \quad (26)$$

Since the pose $\mathbf{T}(s)$ is an integral of the twist, minimizing the effect of string encoder measurement noise on the estimates of the segment shape requires minimizing $\|\delta\boldsymbol{\xi}(s)\|$, which in turn requires maximizing $\aleph(\mathbf{J}_{\ell\xi})$. Since $\aleph(\mathbf{J}_{\ell\xi})$ contains linear/angular velocity units, we multiply the first three rows corresponding to angular velocity by a characteristic length c_ℓ [41]. For our experimental and simulation results, we chose c_ℓ to be the segment's kinematic radius so that angular velocities are scaled to represent linear velocities at the edge of the disk.

We must also consider the numerical conditioning of the configuration space Jacobian $\mathbf{J}_{\ell c}$, since it is used when iteratively solving (16). Since maximizing $\aleph(\mathbf{J}_{\ell\xi})$ does not guarantee a well-conditioned $\mathbf{J}_{\ell c}$, we seek to also prevent $\aleph(\mathbf{J}_{\ell c}) = 0$, which would indicate a singular $\mathbf{J}_{\ell c}$. We define this design problem as a constrained optimization problem:

$$\max_{\mathbf{k}} \aleph_g(\mathbf{J}_{\ell\xi}) \quad \text{s.t.} \quad \aleph(\mathbf{J}_{\ell c}) \geq \epsilon \quad (27)$$

where \aleph_g is the global noise amplification index defined below, \mathbf{k} are a set of string path design parameters, and ϵ is a lower-bound on the noise amplification index. We will provide two examples of defining \mathbf{k} to ensure the string paths are physically realizable in the simulation and experimental studies below. For these two examples, \mathbf{k} are restricted to a set of integers such that (27) can be solved by a brute-force search.

We define $\aleph_g(\mathbf{A})$ as the global noise amplification index, which is the average of $\aleph(\mathbf{A})$ over an admissible configuration workspace \mathcal{C}_a denoting all admissible configurations \mathbf{c} . This global performance measure can be numerically approximated

by sampling \mathcal{C}_a along j sample points $\mathbf{c}_1 \dots \mathbf{c}_j$ and computing $\aleph(\mathbf{A})$ at each sampled configuration:

$$\aleph_g(\mathbf{A}) \triangleq \frac{\int_{\mathcal{C}_a} \aleph(\mathbf{A}) d\mathcal{C}_a}{\int_{\mathcal{C}_a} d\mathcal{C}_a} \approx \frac{1}{j} \sum_{i=1}^j \aleph(\mathbf{A}_i) \quad (28)$$

We define the admissible workspace \mathcal{C}_a as the set of all shapes that a segment can achieve:

$$\mathcal{C}_a = \{\mathbf{c} \mid \mathbf{f}(\mathbf{c}) \leq \mathbf{0}\} \quad (29)$$

where $\mathbf{f}(\mathbf{c})$ is a vector of constraints on the robot's configuration. To compute the admissible workspace, we take samples in the configuration space \mathbf{c} and discard samples that violate the constraints $\mathbf{f}(\mathbf{c})$. The constraints $\mathbf{f}(\mathbf{c})$ need to be defined on a case-by-case basis, but in this paper we define three constraints that are relevant to the robots considered in our simulation studies and experimental results.

The first set of constraints we consider are maximal strain limits on the continuum structure $\epsilon_{max} = [\epsilon_{max,x}, \epsilon_{max,y}, \epsilon_{max,z}]^T$. Assuming a backbone diameter d_b and a beam model following linear elasticity, the local curvature limits are given by:

$$\max_s (u_j(\mathbf{c}, s)) - \frac{\epsilon_{max,j}}{(d_b/2)} \leq 0, \quad j \in \{x, y, z\} \quad (30)$$

where \max_s denotes the maximum over $s \in [0, L]$ for a configuration \mathbf{c} .

The second set of constraints we consider are the maximum curvatures to prevent the intermediate disks from colliding. Considering a subsegment of the robot between two interme-

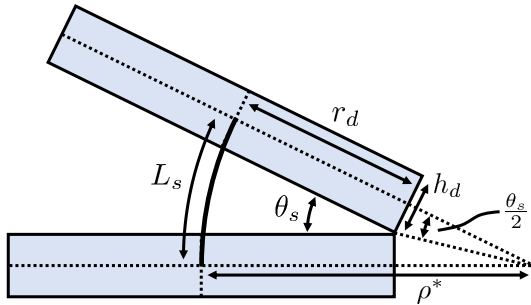


Fig. 5. Kinematic variables for a subsegment in a constant curvature configuration with disk collision.

mediate disks with length L_s , and assuming the subsegment is under constant curvature as shown in Fig. 5, we have:

$$\tan\left(\frac{\theta_s}{2}\right) = \frac{h_d}{2(\rho^* - r_d)} \quad (31)$$

where θ_s is the angle between the intermediate disks, h_d is the height of each intermediate disk, r_d is the radius of each disk, and ρ^* is the radius of curvature when the disks collide. Substituting $\theta_s = \frac{L_s}{\rho^*}$ into (31), we have the following

$$2(\rho^* - r_d) \tan\left(\frac{L_s}{2\rho^*}\right) = h_d \quad (32)$$

which we solve numerically for ρ^* . To prevent disk collision, we then require that the curvatures in the x and y direction are low enough to avoid this collision condition:

$$\max_s (\|\mathbf{u}_{x/y}(s)\|) \leq \frac{1}{\rho^*} \quad (33)$$

where $\mathbf{u}_{x/y}(s) = [u_x(s), u_y(s)]^T$ and \max_s again denotes the maximum over $s \in [0, L]$.

The third set of constraints we include are given by (15), which ensure that the string paths are physically realizable. Using these three sets of constraints (as specified by (30), (15), and (33)), we determine the admissible workspace by searching for configurations that do not violate the constraints, and compute \aleph_g for these admissible configurations. This allows us to explore how to design the string routing paths to maximize \aleph_g , which we do below for several simulation and experimental examples.

IV. PLANAR CASE STUDY

In this section, we present a simulation case study for a segment subject only to planar deflections. We show that for planar deflections and string routings with constant pitch radius, the configuration space Jacobian $\mathbf{J}_{\ell c}$ is constant for any choice of the modal basis $\Phi(s)$. We then explore choices of string anchor points and pitch radii that improve the noise amplification indices $\aleph_g(\mathbf{J}_{\ell c})$ and $\aleph(\mathbf{J}_{\ell c})$.

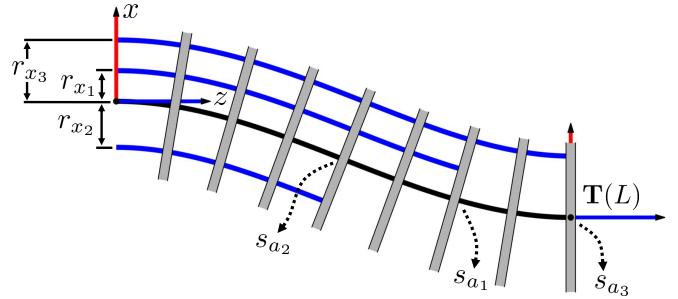


Fig. 6. Variables of the kinematic model used in the planar case study.

Consider a planar continuum segment that is restricted to deflect in the $x - z$ plane, i.e. $u_x = u_z = 0$, as shown in Fig. 6. We choose to represent the curvature distribution using a second-order Chebyshev series as given in (8), and we assume three strings are routed within the segment and anchored at arc lengths s_{a1} , s_{a2} , and s_{a3} . We also assume the strings are routed in a path with a constant pitch radius, i.e. ${}^t\mathbf{r}_i(s) = [r_{x_i}, 0, 0]^T$, where $r_{x_i} \in \mathbb{R}$ is a constant scalar. Noting that ${}^t\mathbf{r}'_i(s) = [0, 0, 0]^T$, (13) simplifies to:

$${}^t\mathbf{w}'_i(s) = \left[0, 0, \left(1 - r_{x_i} \phi_y^T(s) \mathbf{c}_y\right)\right]^T \quad (34)$$

Applying the requirement from (14) that $({}^t\mathbf{w}'_i)^T \mathbf{e}_3 > 0$, for this planar case, (12) simplifies to:

$$\ell_i = s_{a_i} - r_{x_i} \int_0^{s_{a_i}} \phi_y^T(s) \mathbf{c}_y ds \quad (35)$$

We will now consider how many strings are needed to accurately predict the tip pose $\mathbf{T}(L)$ of this planar segment.

We assume here that the number of columns in the modal shape basis is equal to the number of strings, i.e. $p = m$ and $\mathbf{J}_{\ell c}$ is square. This means each additional sensing string enables an additional higher-order term to be added to the shape basis to further reduce the tip pose error. Prior works that used shape functions for modeling continuum robots have shown that low-order shape functions can be sufficient for capturing variable curvature deflections [18], [32], [42]. We will also demonstrate this here for this simulation study and experimentally in Sections V and VI.

We used a Cosserat rod mechanics model from [20], which neglected shear strains and extension, to simulate a Nitinol rod with a length $L = 300$ mm and a diameter of 4 mm (similar to the central backbone Nitinol rod used in the robot in Section V). To generate a variety of variable curvature rod shapes, we subjected the rod to planar forces in the world frame's x direction and moments in the world frame's y direction, with the world frame assigned as shown in Fig. 6. A subset of these variable curvature shapes is shown in Fig. 7. The shape of the rod was obtained as a solution to a boundary value problem using the shooting method for each applied wrench. The maximum applied force and moment were $\mathbf{f}_e = [\pm 60, 0, 0]$ N and $\mathbf{m}_e = [0, \pm 6, 0]^T$ Nm, with 10 wrenches selected between these maximum values, for a total of 100 applied wrenches and rod shapes that were solved for.

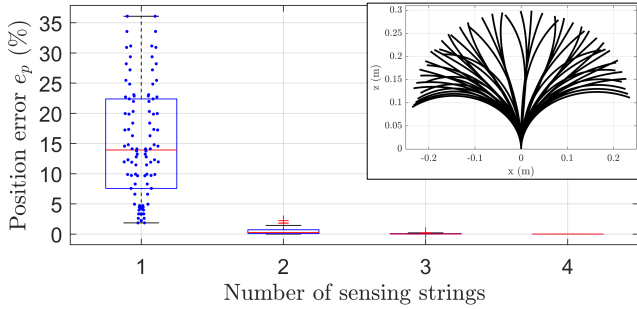


Fig. 7. The tip position error between our kinematic model and a simulated Nitinol rod rapidly converges as the number of strings p is increased. Individual data points are shown for $p = 1$, and a subset of the 100 simulated variable curvature rod shapes is shown in the inset.

For each set of 100 shapes and number of strings considered, we determined the string routing radii and anchor points by solving the following constrained optimization problem:

$$\max_{r_{x_i}, s_{a_i}} \aleph(\mathbf{J}_{\ell c}) \quad \text{s.t.} \quad 0 \leq s_{a_i} \leq L \quad i = 2, 3, \dots, p \quad (36)$$

where we have assumed that $r_{x_1} = 0.25$ and $s_{a_1} = L$ to represent an actuation tendon anchored at the end disk. We solved (36) using the interior point solver provided by MATLAB's `fmincon()` with an initial guess of evenly spaced radii and anchor points. For each additional string, we used the Cosserat rod model to determine the string lengths, then used these simulated string lengths to predict the shape and tip pose $\mathbf{T}(L)$ by solving (16).

We report the error between the tip pose predicted by the mechanics model and the solution given by solving (16) as a percent of the segment length:

$$e_p = \frac{\|\mathbf{p}_s - \mathbf{p}_p\|}{L} \times 100 \quad (37)$$

where \mathbf{p}_s and \mathbf{p}_p are the tip positions given by the mechanics model and predicted via (16), respectively. The position errors across the 100 shapes and for different numbers of strings are shown in Fig. 7. We do not report rotation errors because, as shown in [43], one string anchored to the end disk is sufficient to provide the tip angle of a planar segment and the rotation errors were therefore within numerical precision across all of the simulations.

Fig. 7 shows rapid convergence of the tip position error as the number of strings is increased. The average tip position errors across the 100 simulated shapes was 15%, 0.47%, 0.059% and 0.0052% of the segment length for one, two, three and four strings, respectively. We proceed with this case study by choosing $p = 3$.

Concatenating the string lengths for $i = [1, 2, 3]$ results in

$$\ell = \underbrace{\begin{bmatrix} s_{a_1} \\ s_{a_2} \\ s_{a_3} \end{bmatrix}}_{\mathbf{J}_{\ell c}} - \underbrace{\begin{bmatrix} r_{x_1} & 0 & 0 \\ 0 & r_{x_2} & 0 \\ 0 & 0 & r_{x_3} \end{bmatrix}}_{\mathbf{J}_{\ell c}} \underbrace{\begin{bmatrix} \int_0^{s_{a_1}} \phi_y^T(s) ds \\ \int_0^{s_{a_2}} \phi_y^T(s) ds \\ \int_0^{s_{a_3}} \phi_y^T(s) ds \end{bmatrix}}_{\mathbf{c}_y} \quad (38)$$

where we have denoted the Jacobian $\mathbf{J}_{\ell c}$ with an underbrace. For the planar case, $\mathbf{J}_{\ell c}$ is independent of the configuration \mathbf{c} (i.e. it is constant throughout the workspace).

Solving for the modal coefficients \mathbf{c}_y requires inverting $\mathbf{J}_{\ell c}$. This Jacobian is the product of a diagonal matrix whose elements are the pitch radii of each string and a Vandermonde-like matrix of polynomial functions determined by the choice of modal basis. Here we seek to design the string routing paths to improve the numerical conditioning of $\mathbf{J}_{\ell c}$. For a given choice of the modal basis ϕ_y , the design parameters for each string are 1) the pitch radii r_{x_i} , and 2) the anchor points s_{a_i} .

There are several string routing design insights that can be observed directly from (38). First, r_{x_i} must not be zero to avoid rank deficiency. Second, increasing the pitch radius r_{x_i} (while keeping the ratios of the pitch radii close to one) will increase $\aleph(\mathbf{J}_{\ell c})$. Third, if any two string anchor points s_{a_i} are equal, $\mathbf{J}_{\ell c}$ will lose rank, so the strings should be anchored at unique s_{a_i} along the segment. Noting that actuation tendons endowed with motor encoder sensing provide the same shape information as a passive string encoder, this means that although actuation tendons are commonly anchored at the end disk to expand the segment's workspace [35], any additional actuation tendon anchored to the end disk provides no additional shape information (in the planar case). In Section V, we provide conditions under which strings or tendons anchored at the end disk do provide additional shape information for out-of-plane deflections.

We will now consider how to choose s_{a_i} to maximize $\aleph(\mathbf{J}_{\ell c})$ (the noise amplification index of the configuration space Jacobian) for our planar example. To represent an actuation tendon that is anchored at the end disk, we choose $s_{a_3} = L$. We choose $r_{x_3} = 0.25L$ to correspond to the typical length-diameter ratio of most continuum robots. We now investigate how to optimally choose the radii and anchor points of the two remaining strings.

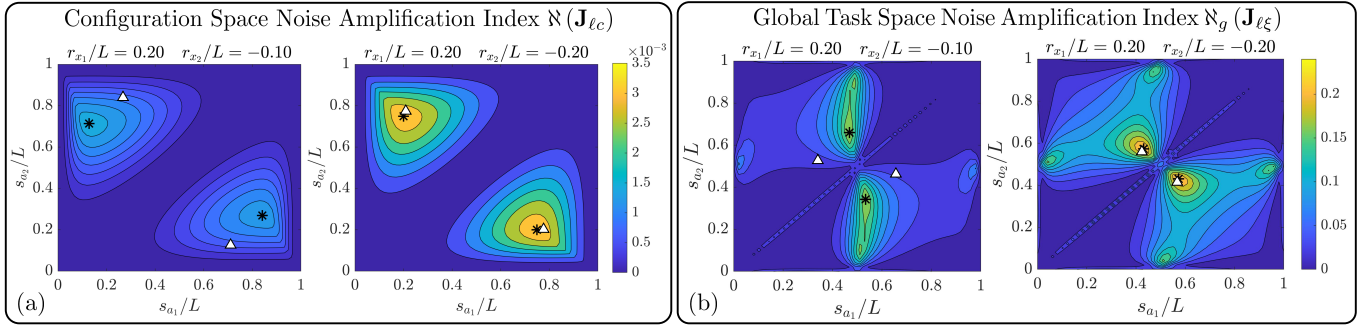


Fig. 8. The values of a) $\aleph(\mathbf{J}_{\ell c})$ and b) $\aleph_g(\mathbf{J}_{\ell \xi})$ for different choices of string anchor points ($\frac{s_{a_1}}{L}$ and $\frac{s_{a_2}}{L}$) shown for two sample string radii ($\frac{r_{x_1}}{L}$ and $\frac{r_{x_2}}{L}$) in the planar robot shown in Fig. 5. The location of the two peaks on each plot are marked with an asterisk (*), and a triangle (Δ) denotes the peak values for the other designs included in Table I and Table II. The contours for the peaks denoted with triangles are not shown for clarity, but they follow a similar pattern to the ones shown.

TABLE I

PLANAR CASE: LOCAL MAXIMA OF $\aleph(\mathbf{J}_{\ell c})$ COMPARED TO EQUIDISTANT SPACING OF STRING PLACEMENT. β SHOWS THE PERCENT IMPROVEMENT IN $\aleph(\mathbf{J}_{\ell c})$

r_{x_1}/L	r_{x_2}/L	s_{a_1}/L	s_{a_2}/L	$\aleph(\mathbf{J}_{\ell c}) (\times 10^{-3})$	β
0.10	-0.10	0.204	0.772	1.03	64%
		0.772	0.204	1.03	64%
0.10	-0.20	0.269	0.841	1.32	54%
		0.714	0.128	1.50	76%
0.20	-0.10	0.128	0.714	1.50	65%
		0.841	0.269	1.32	46%
0.20	-0.20	0.200	0.749	3.29	53%
		0.749	0.200	3.29	53%

TABLE II

PLANAR CASE: LOCAL MAXIMA OF $\aleph_g(\mathbf{J}_{\ell \xi})$ COMPARED TO EQUIDISTANT SPACING OF STRING PLACEMENT. β SHOWS THE PERCENT IMPROVEMENT IN $\aleph_g(\mathbf{J}_{\ell \xi})$

r_{x_1}/L	r_{x_2}/L	s_{a_1}/L	s_{a_2}/L	$\aleph_g(\mathbf{J}_{\ell \xi}(L))$	β
0.10	-0.10	0.573	0.428	0.159	146%
		0.428	0.573	0.159	146%
0.10	-0.20	0.343	0.534	0.181	204%
		0.660	0.468	0.181	204%
0.20	-0.10	0.468	0.660	0.181	213%
		0.534	0.343	0.181	213%
0.20	-0.20	0.573	0.431	0.228	76%
		0.431	0.573	0.228	76%

Figure 8a and Table I show $\aleph(\mathbf{J}_{\ell c})$ for different choices of string radius and anchor point, from which we make several observations. First, we note that each plot in Fig. 8a contains two peaks, and that $\aleph(\mathbf{J}_{\ell c}) = 0$ when $s_{a_1} = s_{a_2}$ and when $s_{a_i} = 0$ or $s_{a_i} = L$. Second, we observe that increasing the pitch radius of both string increases $\aleph(\mathbf{J}_{\ell c})$ while not substantially changing the peak location. Third, we note that increasing a single string's pitch radius increases $\aleph(\mathbf{J}_{\ell c})$ while allowing the location of the peak to be adjusted.

Of note is the fact that the intuitive design choice of evenly spacing out the anchor points along $s = [0, L]$ does not result in the best kinematic conditioning for $\mathbf{J}_{\ell c}$. Table I shows the values of the noise amplification index for each of the peaks in Fig. 8 as well as for designs where the string anchor points are evenly spaced, i.e. $s_{a_1} = s_{a_2}$. Table I also shows the improvement in the noise amplification index over the evenly spaced designs, calculated as:

$$\beta = \left[\left(\aleph(\mathbf{J}_{\ell c}) - \aleph(\tilde{\mathbf{J}}_{\ell c}) \right) / \aleph(\tilde{\mathbf{J}}_{\ell c}) \right] \times 100 \quad (39)$$

where $\tilde{\mathbf{J}}_{\ell c}$ denotes the Jacobians for designs having evenly spaced wire anchor points at $s_{a_1} = L/3, s_{a_2} = 2L/3, s_{a_3} = L$. As shown in Table I, placing the anchor points at one of the peaks instead of using evenly spaced anchor points resulted in 46% or greater improvement in $\aleph(\mathbf{J}_{\ell c})$ for all the cases considered.

We now consider improving the numerical conditioning of the full kinematic mapping (joint to task space) $\aleph_g(\mathbf{J}_{\ell \xi}(L))$. We computed $\aleph_g(\mathbf{J}_{\ell \xi}(L))$ for this robot using 67 configura-

tions sampled in the admissible workspace, which we defined as any configuration that did not exceed 5% strain for a 4 mm central backbone. Figure 8b shows the noise amplification for different string radii and anchor points. We observe that the design parameters that optimize $\aleph_g(\mathbf{J}_{\ell \xi}(L))$ are not the same parameters that optimize $\aleph(\mathbf{J}_{\ell c})$ in Fig. 8a, and in fact there is a conflict between these two design objectives, since the peaks in Fig. 8b correspond to valleys in Fig. 8a. Since the design objective is usually to minimize the pose error at a particular location, designing for maximizing $\aleph_g(\mathbf{J}_{\ell \xi}(L))$ is sufficient as long as the string routing solution avoids singularity of $\mathbf{J}_{\ell c}$.

We also observe again that the intuitive design of using evenly spaced anchor points does not result in an optimally conditioned kinematic mapping. Table II shows the values of $\aleph_g(\mathbf{J}_{\ell \xi}(L))$ for the peaks shown in Fig. 8b as well as the percent improvement in $\aleph_g(\mathbf{J}_{\ell \xi}(L))$ as compared to a design using evenly spaced anchor points, i.e. $s_{a_2} = \frac{L}{3}$ and $s_{a_3} = \frac{2L}{3}$. The peak values of $\aleph_g(\mathbf{J}_{\ell \xi}(L))$ were increased by 76% or greater for all cases considered.

In this planar example, we have shown that increasing the number strings reduces the pose estimation error, provided string path design considerations, and showed that the intuitive choice of evenly spaced anchor points does not lead to optimal values for the noise amplification. The optimal placement of string anchor points depends to some degree on the family of deflected shapes that a given robot experiences under target design operating conditions (loading and reach). We have simulated the same case using a monomial basis as opposed to a Chebyshev polynomial basis and noted that the peaks

in Fig. 8 experience negligible changes. Nevertheless, the designer should carry out a simulation as in Fig. 8 to achieve a qualitative understanding of the optimal location of wire anchor points and then determine these points while respecting practical design considerations. Building on these simulation results, we will now demonstrate our modeling and shape sensing approach on two other continuum robot embodiments that are subject to more realistic spatial deflections.

V. ROBOTS WITH HIGH TORSIONAL STIFFNESS AND CONSTANT PITCH STRING PATHS

We now consider robots that are subject to spatial deflections but have sufficiently high torsional stiffness that renders the torsional deflections negligible. We consider this category of robots because a number of continuum robots with high torsional stiffness have been presented in prior work [44–47], and we will present here a new modular collaborative continuum robot in this category and use it to experimentally validate the model we present. This model is also directly applicable to hyper-redundant robots with torsionally stiff universal joint backbones, e.g. [25], [48]. This category of robots is also of interest because, as we will show, if the string paths are restricted to constant pitch radius paths, the configuration space Jacobian is constant and the modal coefficients are linear with respect to the string lengths, which simplifies the solution of the shape sensing problem.

First, we will present the kinematic formulation for this category of robots and provide considerations for designing the string paths. In particular, we show that *two strings anchored to the same disk add distinct information only if their polar coordinates are distinct by angle difference other than 0° and 180° (i.e. they are not collinear in the radial direction of the disk)*. We then present the results of the string routing optimization for our collaborative continuum segment and experimentally validate the sensing approach, showing that the end disk position can be sensed with position errors below 5% of arc length using information from four passive string encoders and two actuators.

A. Kinematic model for torsionally stiff continuum robots

Under the assumption of negligible torsional deflections, i.e. $u_z(s) = 0$ the shape basis (4) takes the form

$$\mathbf{u}(s) = \begin{bmatrix} \phi_x^T & 0 \\ 0 & \phi_y^T \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{c}_x \\ \mathbf{c}_y \end{bmatrix} = \Phi(s)\mathbf{c} \quad (40)$$

We restrict our consideration here to constant pitch-radius string routings give by:

$${}^t\mathbf{r}_i(s) = [r_{x_i} \quad r_{y_i} \quad 0]^T, \quad r_{x_i} \in \mathbb{R}, \quad r_{y_i} \in \mathbb{R} \quad (41)$$

Noting that ${}^t\mathbf{r}'_i(s) = [0, 0, 0]^T$, the string path derivative (13) simplifies to:

$${}^t\mathbf{w}'_i(s) = \left[0, 0, \left(r_{y_i} \phi_x^T \mathbf{c}_x - r_{x_i} \phi_y^T \mathbf{c}_y + 1 \right) \right]^T \quad (42)$$

Applying the requirement from (14) that $({}^t\mathbf{w}'_i)^T \mathbf{e}_3 > 0$ and using (12) results in the string length as:

$$\ell_i = s_{a_i} + \left(\int_0^{s_{a_i}} [r_{y_i} \phi_x^T, -r_{x_i} \phi_y^T] ds \right) \mathbf{c} \quad (43)$$

where $\mathbf{c} = [\mathbf{c}_x^T, \mathbf{c}_y^T]^T$. Concatenating for each string $i \in [1, \dots, p]$ gives:

$$\boldsymbol{\ell} = \begin{bmatrix} s_{a_1} \\ \vdots \\ s_{a_p} \end{bmatrix} + \underbrace{\begin{bmatrix} \int_0^{s_{a_1}} [r_{y_1} \phi_x^T, -r_{x_1} \phi_y^T] ds \\ \vdots \\ \int_0^{s_{a_p}} [r_{y_p} \phi_x^T, -r_{x_p} \phi_y^T] ds \end{bmatrix}}_{\mathbf{J}_{\ell c}} \mathbf{c} \quad (44)$$

As in the planar case, we observe that $\mathbf{J}_{\ell c}$ is independent of the configuration \mathbf{c} .

We now consider how to choose the anchor points and string paths to avoid singularities in $\mathbf{J}_{\ell c}$. Consider the scenario where two anchor points are equal, $s_{a_i} = s_{a_j}, i \neq j$. To prevent singularities, we must avoid a scenario where two rows of $\mathbf{J}_{\ell c}$ become dependent, e.g. one row is a scalar multiple of another. Considering a cross section of the segment at $s = s_{a_i} = s_{a_j}$, this will occur if the string radii $[r_{x_i}, r_{y_i}]$ and $[r_{x_j}, r_{y_j}]$ lie on a line passing through the origin of the body frame $\mathbf{T}(s_{a_i}) = \mathbf{T}(s_{a_j})$.

Furthermore, anchoring more than two strings will also cause $\mathbf{J}_{\ell c}$ to become rank-deficient. This is apparent since two radially non-collinear strings define the plane of the disk. Therefore, for an inextensible, torsionally stiff continuum segment, no more than two parallel-routed strings should be anchored to the same intermediate disk (to prevent rank-deficiency in $\mathbf{J}_{\ell c}$).

B. Experimental validation on a collaborative continuum robot module

We now validate our kinematic model on a continuum segment, shown in Fig. 9 and in the multimedia extension, that is a portion of a robotic arm currently under development for collaborative manufacturing in confined spaces. Additional discussion on the motivations for development of this device can be found in [39], [49].

The flexible structure of the segment consists of aluminum intermediate disks with torsionally stiff metal bellows attached to the disks with an acrylic adhesive. Each bellow has a torsional stiffness of approximately 63 Nm/°, making the bellows approximately 1950 times stiffer in torsion than in bending. For the target application, the segments are expected to experience static torsional loads of up to 64 Nm, resulting in approximately 1° of torsional deflection per bellow. Since these maximum deflections are small relative to the bending deflections, we neglect the torsional deflections in our kinematic model below. These bellow subassemblies are bolted together, and a $\varnothing 4$ mm solid Nitinol rod is passed through the center of the structure to prevent contraction of the segment. The length of the segment, measured from the top of the actuation unit to the bottom of the end disk, is 300.65 mm.

The segment is actuated by a pair of actuation tendons that pass through bronze bushings in the intermediate disks and are

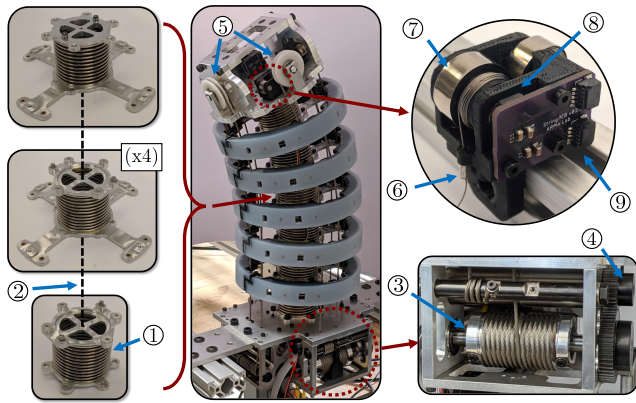


Fig. 9. The modular continuum segment consists of torsionally stiff metal bellows ① with a $\varnothing 4$ mm Nitinol rod ② passing through their centers. The segment is actuated with a capstan mounted on a linear ball spline ③ that is actuated by a brushed DC motor/geartrain ④. Idler pulleys ⑤ provide a 2:1 reduction in the actuation tendon force. Each continuum segment contains four string encoders, which consist of a wire rope ⑥, a constant-torque return spring ⑦, and a magnetic encoder ⑧ with an I²C interface ⑨.

made from $\varnothing 2.38$ mm steel wire rope. The actuation tendons are actuated in a differential manner (as shown in Fig. 10) by a capstan mounted on a linear ball spline, and this ball spline is actuated by a gearmotor with a 111:1 gear ratio (maxon DCX22L/GPX22HP) through a single spur gear stage with a gear ratio of 1.851. In the distal endplate assembly of the segment, idler pulleys route the tendons back towards the base of the segment, and the tendons are anchored to a manual pretensioning mechanism within the actuation unit. These idler pulleys provide a 2:1 reduction in the tendon force, which reduces the required sizing of the mechanical components in the actuation unit.

Each intermediate disk is also equipped with 8 time-of-flight proximity sensors and 8 contact sensors which enable mapping of the environment, human-robot physical interaction, and bracing against the environment along the entire body of the segment. The potential for whole-body contact with this segment is one motivation for the study of sensitivity analysis along different arc-lengths that we carry out below. Additional details on these sensing disks can be found in [39], [50].

Four custom-built string encoders are mounted in the distal portion of the segment, as shown in Fig. 9. The encoders have a $\varnothing 0.33$ mm diameter wire rope wrapped around a $\varnothing 12$ mm capstan and a constant-torque return spring (Vulcan Spring SV3D48) that results in a constant preload of 3.3 N on each string. This preload was sufficient to overcome friction and keep the string taut. Although the preload had a negligible effect on the robot shape, we note that our kinematic approach captures any deflections that occur due to the string preload. A printed circuit board (PCB) with a magnetic encoder (Renishaw AM4096 12 bit incremental) measures the angle of the capstan. The PCB also has connectors for the encoder's I²C interface. All of these components are housed in a 3D-printed PLA housing. The magnetic encoder angle is read via I²C by a Teensy 4.1 microcontroller at a rate of ~ 150 Hz. The Teensy then provides the angle via UDP using a Wiznet WIZ850IO ethernet board to a Robot Operating System (ROS) node that

publishes the string extensions on a ROS topic.

The segment motors are controlled with a control box containing six motor drivers (maxon ESCON 50/5), encoder reading cards (Sensoray 526) and a PC/104 CPU (Diamond Systems Aries) running Ubuntu/Linux with the PREEMPT-RT real-time patch. A higher-level ROS controller computes a fifth-order quintic polynomial trajectory and sends desired velocities to the motor control box via UDP. A real-time PI controller then commands desired motor currents based on the desired position/velocity profile.

The radius of the capstan on each string encoder was calibrated by extending the string in 0.2 mm increments using a 3-DoF Cartesian robot. The linear stages of the Cartesian robot are Parker 404XR ballscrew linear stages actuated with brushed DC motors (Maxon RE35) and equipped with 1000 counts-per-turn encoders. The motion control accuracy of the linear stages was evaluated at ± 15 μ m. The Cartesian robot was used to extend/release the string over $n_c = 1201$ positions given by $\mathbf{d}_r = [0, 0.2, 0.4, \dots, 120, 119.8, 119.6, \dots, 0]^T$ mm. Using the wrapping capstan model while neglecting the helix angle, we obtain the measurement model $\mathbf{d}_r = \boldsymbol{\theta}_e r_e$, where $\boldsymbol{\theta}_e \in \mathbb{R}^{n_c}$ is the vector of magnetic encoder angles and r_e is radius of the encoder capstan. We then solve for the value of r_e that minimizes the least-squares error between the predicted string extension based on string encoder angle and the distance traveled by the Cartesian robot by evaluating $r_e = \boldsymbol{\theta}_e^+ \mathbf{d}_r$. We calibrated all four of the string encoders and found that the average extension measurement error was below 0.1 mm (0.08% of the total stroke) and the maximum extension measurement error across all four string encoders was 0.27 mm (0.23% of total stroke).

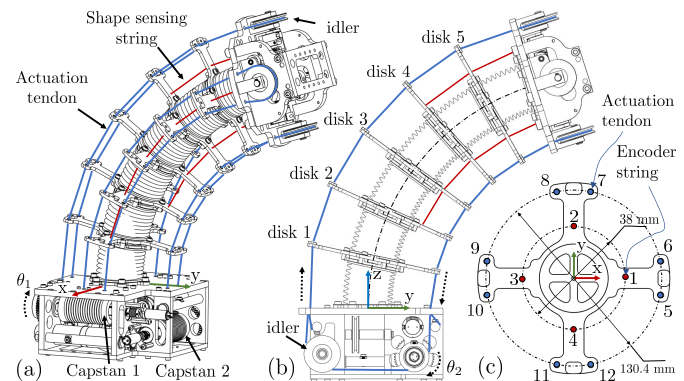


Fig. 10. The path of the actuation tendons (numbered 5-12) and string encoders (numbered 1-4) are shown in (a) a 3D view, and (b) a simplified side view. Shown in (c) is top view of an intermediate disk with the locations where each string/tendon passes.

Due to space considerations for integrating sensing electronics in the intermediate disks [39], the strings are restricted to be routed in constant pitch radius paths. The strings paths are therefore given by (41), where the values for r_{x_i} and r_{y_i} are determined from the geometry in Fig. 10c. Since the string encoders are mounted in the distal endplate (instead of in the robot's base), the string lengths ℓ_i , $i \in [1, 2, 3, 4]$ are found via (12), except integration is performed from s_{a_i} to L rather than from 0 to s_{a_i} , resulting in an expression similar to (43).

We also use the actuation tendon lengths ℓ_i , $i \in [5, \dots, 12]$ as additional input to the problem of shape estimation. The routing path definitions for the actuation tendons require special consideration because of the idler pulleys in the end plate that reduce the tendon force by rerouting the tendons to the base of the robot. We separate the actuation tendon paths into eight different curves between the base plate and the end plate, as partially shown in Fig. 10a as blue curves. We then require the difference in the tendon path lengths on each side of the actuation capstan be equal to the change in length due to the actuation capstan rotation:

$$\begin{aligned} \Delta \ell_{\theta_1} &= \Delta(\ell_5 + \ell_6) = -\Delta(\ell_9 + \ell_{10}) \\ \Delta \ell_{\theta_2} &= -\Delta(\ell_7 + \ell_8) = \Delta(\ell_{11} + \ell_{12}) \end{aligned} \quad (45)$$

where $\ell_5 \dots \ell_{12}$ refer to the actuated tendon lengths corresponding with the numbered bushings shown in Fig. 10c and ℓ_{θ_1} and ℓ_{θ_2} are the change in tendon length due to rotation of the capstan for joints 1 and 2, respectively. The values of ℓ_{θ_1} and ℓ_{θ_2} are determined from the motor angles while accounting for the helical wrapping pattern on the capstan:

$$\ell_{\theta_i} = \frac{\theta_i}{2\pi} \sqrt{(2\pi r_c)^2 + \gamma^2}, \quad i \in [1, 2] \quad (46)$$

where θ_1 and θ_2 are the angles of the first and second actuation capstans, respectively, r_c is the radius of the capstan, and γ is the lead of the helical groove on the capstan. The actuation tendon lengths ℓ_i , $i \in [5 \dots 12]$ are found via (43).

Information from the two motor encoders and the four string encoders allows for a modal basis with six columns ($p = 6$). Neglecting torsional deflections, the curvature distribution is given by (40) with the following shape functions:

$$\phi_x(s) = \phi_y(s) = [T_0, T_1(s), T_2(s)]^T \quad (47)$$

The string encoder length equations for $i \in [1, 2, 3, 4]$ are concatenated together with (45) to give the string lengths in a similar form as in (44), but accounting for the string encoder routing as described above while adding the tendon lengths:

$$\begin{bmatrix} \ell_1 \\ \ell_2 \\ \ell_3 \\ \ell_4 \\ \ell_{\theta_1} \\ \ell_{\theta_2} \end{bmatrix} = \begin{bmatrix} L - s_{a1} \\ L - s_{a2} \\ L - s_{a3} \\ L - s_{a4} \\ 0 \\ 0 \end{bmatrix} + \mathbf{J}_{\ell c} \mathbf{c}, \quad \mathbf{J}_{\ell c} \in \mathbb{R}^{6 \times 6}, \quad \mathbf{c} \in \mathbb{R}^6 \quad (48)$$

Given a set of string measurements, we then solve (48) for the modal coefficients \mathbf{c} with a single matrix inversion. Our MATLAB 2019b implementation computes \mathbf{c} at a rate of ~ 125 kHz. As with the planar example above, the configuration space Jacobian $\mathbf{J}_{\ell c}$ is constant when assuming zero torsional deflections, so $\mathbf{J}_{\ell c}$ and $\mathbf{J}_{\ell c}^{-1}$ can be computed once and stored.

Based on the analysis in Section V-A of singularities in $\mathbf{J}_{\ell c}$, the two actuation tendon equations in (45) do not introduce singularities into $\mathbf{J}_{\ell c}$, because the tendon pairs that are collinear with the central backbone point are in the same equations in (45). However, any additional string anchored at the end disk (or equivalently at the base disk for the strings mounted in the endplate) would not provide additional shape

information. The radius of the passive sensing strings is fixed, so in designing the string paths for this segment, we can only change the string anchor points. The anchor points are also restricted to the discrete points along the central backbone where the five intermediate disks lie. We therefore have five possible choices for s_{a_i} for each of the four strings.

To design the string routings, we run a brute-force search across all possible combinations of s_{a_i} (625 possible designs) to find the combination with the largest $\aleph_g(\mathbf{J}_{\ell \xi}(s))$. We chose the characteristic length to be 0.0652, the kinematic radius of the actuation tendons. The global noise amplification index \aleph_g was computed for a set of 320 configurations in the segment's admissible workspace, and the noise amplification index for the end disk, denoted as $\aleph_g(\mathbf{J}_{\ell \xi}(L))$, and the noise amplification index for the third disk, denoted as $\aleph_g(\mathbf{J}_{\ell \xi}(s_3))$, where s_3 is the arc length at which the third disk is located, were computed at each configuration.

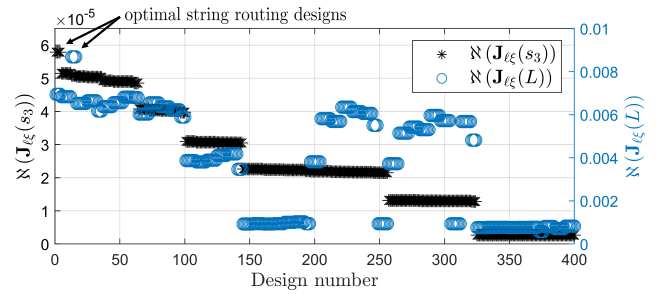


Fig. 11. Noise amplification indices at the end disk and the third disk for all physically realizable string routing designs of the collaborative continuum robot. A large range of noise amplification indices are possible, but the noise amplification indices for the design optimal for end disk pose estimation is not significantly different than for the design optimal for Disk 3 pose estimation.

Out of the 625 string routing designs considered, 225 resulted in a singular $\mathbf{J}_{\ell \xi}(L)$ and $(\mathbf{J}_{\ell \xi}(s_3))$. Figure 11 shows the noise amplification indices for 400 of the string routing designs in our brute-force search that did not result in singularities. Some designs resulted in the noise amplification being particularly close to zero. For example, one poorly conditioned routing design was a design with anchor points at disk 2, disk 4, disk 3, and disk 5, for strings 1-4, respectively, with the disk numbers given in Fig. 10. This design places one string on each disk except disk 1. Although this choice might seem reasonable to a designer at first glance, its noise amplification index is $\aleph_g(\mathbf{J}_{\ell \xi}(L)) = 8.3e-3$, which is two orders of magnitude smaller than for the optimal design. This highlights the importance of carrying out the sensitivity analysis we present herein when choosing a string routing design to avoid these ill-conditioned string routings.

From the designs in Fig. 11, we found that the anchor points that maximize $\aleph_g(\mathbf{J}_{\ell \xi}(L))$ were disk 4, disk 4, disk 2, and disk 2, for strings 1-4 respectively. Below, we will refer to this string routing design as the *end disk routing*. For maximizing $\aleph_g(\mathbf{J}_{\ell \xi}(s_3))$, the optimal anchor points were disk 1, disk 1, disk 3, and disk 3, for strings 1-4, respectively. Below, we will refer to this string routing design as the *third disk routing*.

The noise amplification indices for these two optimized designs are given in Table III. Although the brute-force search

TABLE III
NOISE AMPLIFICATION INDICES FOR OPTIMIZED STRING ROUTING
DESIGNS ON COLLABORATIVE CONTINUUM MODULE

	Disk Used For Routing Optimization	
	End	3rd
$\aleph_g(\mathbf{J}_{\ell\xi}(L))$	8.69e-3	6.94e-3
$\aleph_g(\mathbf{J}_{\ell\xi}(s_3))$	5.08e-5	5.79e-5

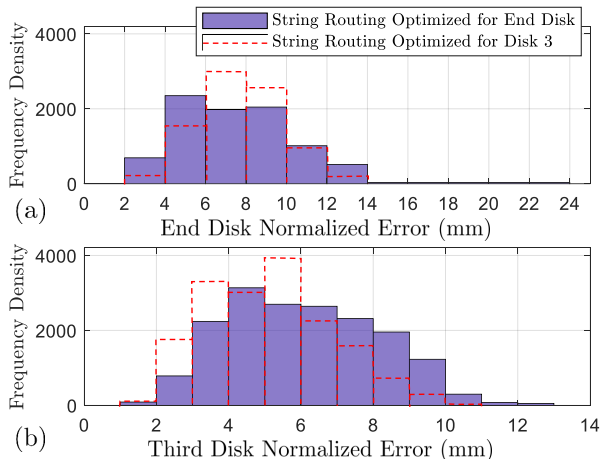


Fig. 12. The frequency histogram of pose error when the string routing is optimized to minimize the effect of measurement noise for either the end disk pose (end disk routing) or the pose of the third disk (third disk routing). (a) The end-disk position error histograms for both routings showing that the error distribution for $\mathbf{T}(L)$ is shifted leftward compared to the third disk routing (b) The third-disk position error histograms for both routings showing that the error distribution for $\mathbf{T}(s_3)$ is shifted leftward compared to the end disk routing.

optimization increased $\aleph_g(\mathbf{J}_{\ell\xi}(L))$ for the end disk routing by 25% compared to the middle disk routing, and the value of $\aleph_g(\mathbf{J}_{\ell\xi}(L))$ for the middle disk routing was increased by 14% compared to the end disk routing, we will show in our experimental results below that these changes in $\aleph_g(\mathbf{J}_{\ell\xi}(L))$ are not large enough to have a significant effect on the pose error either at the end disk or at the third disk. Either of these string routing designs could be chosen for this robot without having a significant effect on the pose error.

We experimentally validated the shape sensing approach on the physical continuum robot. We routed the string encoders according to the two routing designs given by our optimization procedure, and for each string routing design, we measured the shape of the segment across a large variety of variable curvature shapes, a subset of which are shown in Fig. 13. The segment was mounted on a revolute joint driven by an off-the-shelf actuator (Dynamixel PH54-200-S500-R), and the angle of the revolute joint was commanded to 0, 45°, and 90°. For each of these three angles, we commanded the segment to move from the initial home configuration to four different configurations: $\boldsymbol{\theta} = [0^\circ, 500^\circ]$, $\boldsymbol{\theta} = [0^\circ, -500^\circ]$, $\boldsymbol{\theta} = [500^\circ, 0^\circ]$, and $\boldsymbol{\theta} = [-500^\circ, 0^\circ]$, where $\boldsymbol{\theta} = [\theta_1, \theta_2]^T$ contains the angles of the actuation capstans. The motion profile to reach these four poses was generated using a fifth-order polynomial trajectory planner, with a time of 45 seconds to move to the desired configuration from the initial home

configuration. During these motions, we continuously captured the string lengths using the *roscop* ROS package. We captured ground-truth pose of the third intermediate disk and the end disk using a stereo vision optical tracker (ClaroNav H3-60) and optical markers mounted on the robot. The data collection rate is limited by the optical tracker's ~ 8.5 Hz sample rate. This experiment was carried out with the end disk routing, and then repeated for the third disk routing.

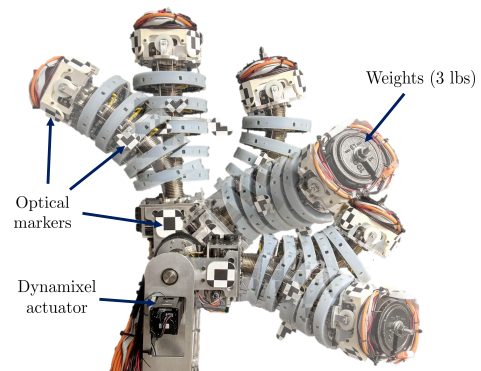


Fig. 13. A subset of the variable curvature spatial configurations used to validate our shape sensing approach for a torsionally stiff continuum segment with straight string routing. The third and end disk poses were captured using optical trackers with and without weights attached to the end disk.

Using the string length measurements acquired with the string encoders, we solved for the modal coefficients using the full shape sensing model given by (48). To compare against a scenario where the robot did not have shape sensing encoders, we also reconstructed the shape using only the actuation variables $\boldsymbol{\theta}$. For this case, we used a modal basis with two columns where $\phi_x = \phi_y = 1$. This is identical to the commonly used constant-curvature model [35]. We report the constant-curvature model results using the data set collected with the third disk routing, but note that the errors were similar for this constant-curvature model using the end disk routing.

TABLE IV
AVERAGE (MAXIMAL) ABSOLUTE POSITION AND ORIENTATION ERRORS
IN MM AND ° FOR THE SEGMENT IN FIG. 9. THE ERRORS ARE SPECIFIED
FOR THE END DISK ($s = L$) AND THE 3rd DISK ($s = s_3$)

	End Disk Routing	Third Disk Routing	w/o Passive Strings
$\bar{p}_e(L), \max(p_e(L))$	5.9 (14.4)	6.0 (13.8)	56.2 (104.9)
$\bar{p}_e(s_3), \max(p_e(s_3))$	3.8 (10.2)	3.3 (9.2)	31.8 (58.6)
$\bar{\theta}_e(L), \max(\theta_e(L))$	1.5 (8.6)	1.5 (3.9)	3.6 (14.4)
$\bar{\theta}_e(s_3), \max(\theta_e(s_3))$	2.0 (6.0)	1.6 (4.2)	15.4 (28.1)

The mean and maximum errors of our shape sensing model (with the two different routing designs) as well as the constant curvature model are given in Table IV. The position error is given by $p_e(s) = \|\mathbf{p}_{model}(s) - \mathbf{p}_{meas}(s)\|$, where $\mathbf{p}_{model}(s) \in \mathbb{R}^3$ is the model-predicted position, and $\mathbf{p}_{meas}(s) \in \mathbb{R}^3$ is the measured position, and we denote the average of $p_e(s)$ across all configurations as $\bar{p}_e(s)$. The angular error is given by:

$$\theta_e(s) = \cos^{-1} \left(\frac{\text{trace}(\mathbf{R}_{meas}(s)\mathbf{R}_{model}(s)^T) - 1}{2} \right) \quad (49)$$

where $\mathbf{R}_{meas}(s) \in SO(3)$ is the measured rotation matrix and $\mathbf{R}_{model}(s) \in SO(3)$ is the model-predicted rotation matrix. Both the end disk routing and the middle disk routing reduced the mean end disk errors compared to the constant curvature model by more than 89% in position and 58% in angle. The maximum end disk error was reduced compared to the constant curvature model by more than 85% in position and 40% in angle for both string routing designs. Both routing designs have average tip position errors below 2.0% of the total arc length, a significant improvement compared to the constant curvature model error of 18.7% of total arc length. The maximum tip position error of our approach was 4.8% of the total arc length, compared to 34.9% for the constant-curvature model.

The third disk routing reduced the mean position error at s_3 by 0.5 mm and the maximum by 1 mm. The third disk routing design also reduced the mean angular error at s_3 by 0.4° and the maximum by 1.8° . Figure 12 shows the histograms of the normalized error $e_n \in \mathbb{R}$ at the end disk and the third disk:

$$e_n = \sqrt{\|\mathbf{p}_{model} - \mathbf{p}_{meas}\| + c_\ell \theta_e} \quad (50)$$

where c_ℓ is the characteristic length used in the string routing design optimization procedure (in this case, we used the kinematic radius of the disk $0.1304/2$ m as shown in Fig. 10). We observe that, as expected, the error distribution of $\mathbf{T}(s_3)$ is shifted leftward towards reduced error due to the increase in $N_g(\mathbf{J}_{\ell\xi}(s_3))$. However, we also observe that the maximum errors for $\mathbf{T}(L)$ were higher using the routing optimized for the end disk. This indicates that the change in the noise amplification index at $\mathbf{T}(L)$ when using the routing optimized for the third disk was not significant enough to affect the tip pose error in a way that would overcome other sources of error, i.e. friction, mechanical clearances, and the continuously parallel routing assumption. Overall, while the general trends in the errors match the expected behavior due to the noise amplification index, we observe that the pose error of the middle and end disk was not substantially affected, meaning that either of the two optimized string routing designs could be used. This provides flexibility in the string routing design.

In this section, we have presented the kinematic formulation for sensing deflections of robots with negligible torsional stiffness. The formulation results in a constant configuration space Jacobian and linear shape sensing equations. We validated the model and approach for a collaborative continuum robot with high torsional stiffness and constant pitch radius string paths, showing that our approach sensed the tip position with errors below 4.8% of total arc length. We now demonstrate the approach for the more general case of robots with non-negligible torsional deflections and helical string routing.

VI. SENSING TORSIONAL DEFLECTIONS WITH HELICAL STRING PATHS

In the section above, we validated our shape sensing approach on a segment with high torsional stiffness. Neglecting torsional stiffness significantly simplifies the model equations and reduces the computation cost of solving for the shape. However, many continuum robots have relatively low torsional

stiffness, so torsional deflections cannot always be neglected. In this section, we consider helical string routing as a way to sense torsional deflections. We then validate the approach in a simulation study using a Cosserat rod mechanics model.

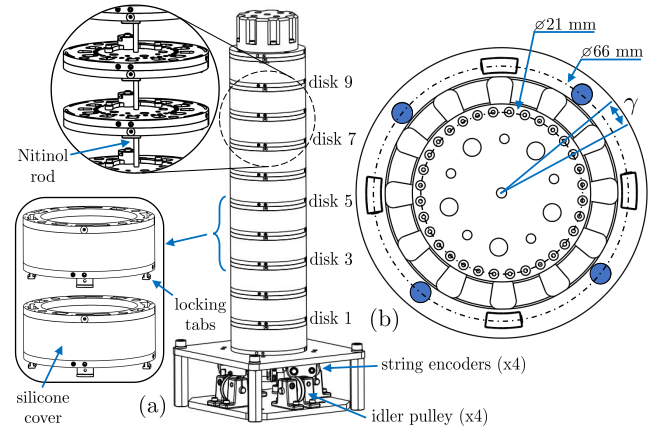


Fig. 14. To validate our approach for torsional deflections and helical string routing, we simulated a modular soft continuum segment. The segment's interlocking subsegments are over-molded with silicone, and a Nitinol rod passes through their centers. Four string encoders are mounted at the segment's base, and each intermediate disk (b) has 32 holes to enable helical string paths.

We chose the geometry of the simulated continuum robot based on a concept for a modular soft continuum robot, shown in Fig. 14. The subsegments of the robot are built with cylindrical silicone over-molded on the outer circumference of the intermediate disks to act as a soft outer cover, and the bottom plate of each subsegment has locking tabs that mate with a slot in the top plate of the previous subsegment. A 293 mm long solid Nitinol rod passes through the center of the continuum robot to prevent compression of the structure.

Four string encoders (as described in Section V) are mounted below the segment, and four tendons are routed to actuators via idler pulleys below the segment's base. In this embodiment, the actuation tendons are routed in straight paths with the tendon path given by (41). We assume that two of the tendons are anchored at the end disk, and that two of the tendons are anchored at disk 7. This choice in anchor points for the actuation tendons allows all four actuation tendons to provide shape information (since more than two strings anchored to a disk will result in a singular $\mathbf{J}_{\ell c}$ when the robot is straight) while still allowing the segment to bend in all four directions and have a large reachable workspace. The string encoders are routed in helical paths given by (51). As shown in Fig. 14c, the bottom plate of each subsegment has 32 holes through which the strings pass, allowing the strings to be routed in the desired helical shape. We now optimize the anchor points of the four string encoders and the twist rate of their helical paths.

The helical routing string path function is given by [36]:

$${}^t\mathbf{r}_i(s) = r_s[\cos(\omega s + \alpha_i), \sin(\omega s + \alpha_i), 0]^T \quad (51)$$

where r_s is the radius of the helical path, ω is the twist rate of the helical path (which we assumed was constant and equal for all four strings to prevent the string paths from intersecting), and α_i is an angular offset for each string. Since we have eight

inputs to our shape sensing model (four passive string encoders and four active actuation tendons), we choose a modal basis with eight columns. The modal shape basis is given by (4) with the following shape functions:

$$\begin{aligned}\phi_x(s) &= \phi_y(s) = [T_0, T_1(s), T_2(s)]^T \\ \phi_z(s) &= [T_0, T_1(s)]^T\end{aligned}\quad (52)$$

where the Chebyshev functions $T_i(s)$ are given by (9). We compute the string lengths and the configuration space Jacobian $\mathbf{J}_{\ell c}$ by numerically integrating (12) and (18), respectively.

We solved the optimization problem (27) with $\epsilon = 1e-7$ through a brute-force search of all possible string anchor points (disks 1-10) and helical path design parameters. Given the 32 holes on the intermediate disks, as shown in Fig. 14, the helical path twist rate can be approximated as $\omega = n_\omega \gamma$, where $n_\omega \in \mathbb{R}$ determines the number of routing holes to skip in between the intermediate disks when routing the string, and $\gamma = \frac{2\pi}{32}$ is the angle between the routing holes, as shown in Fig. 14. We constrained the twist rate to $n_\omega < 2$ to prevent large twist rates, since excessive twist rates increase the possibility of binding between the string and the routing holes. With the ten possible string anchor points and two possible twist rates for four strings, there were 20,000 possible string routing designs that we evaluated. Using a prototype subsegment, we experimentally determined that each subsegment can bend up to 10° and twist up to 7.5° without mechanical failure, so the admissible workspace was defined as a set of 32 sampled configurations that did not exceed 10° of bending and 7.5° of twist in each subsegment. We chose the characteristic length to be radius of the segment, 37.5 mm. We discarded all designs that violated $\aleph_g(\mathbf{J}_{\ell c}) \geq 1e-7$. We will consider below the fourth, sixth, and end disk as representative examples to optimize $\mathbf{J}_{\ell \xi}$ for, so we stored the noise amplification indices for $s = L$, $s = s_4$ (the location of the fourth disk), and $s = s_6$ (the location of the sixth disk).

TABLE V
NOISE AMPLIFICATION INDICES FOR OPTIMIZED STRING ROUTING DESIGNS ON SIMULATED SEGMENT WITH HELICAL ROUTING

	Disk Used For Routing Optimization		
	End	6th	4th
$\aleph_g(\mathbf{J}_{\ell \xi}(L))$	3.47e-3	3.50e-3	3.35e-4
$\aleph_g(\mathbf{J}_{\ell \xi}(s_6))$	5.24e-4	5.40e-4	2.27e-4
$\aleph_g(\mathbf{J}_{\ell \xi}(s_4))$	7.91e-5	8.27e-5	8.85e-5

Figure 15 shows the noise amplification indices across all string routing designs that did not violate $\aleph_g(\mathbf{J}_{\ell c}) \geq 1e-7$, sorted in descending order of $\aleph_g(\mathbf{J}_{\ell \xi}(L))$. Table V shows the values $\aleph_g(\mathbf{J}_{\ell \xi}(L))$, $\aleph_g(\mathbf{J}_{\ell \xi}(s_6))$, and $\aleph_g(\mathbf{J}_{\ell \xi}(s_4))$ for the designs that maximize each of these three values (which are also indicated with arrows in Fig. 15). We observe that the design that maximizes $\aleph_g(\mathbf{J}_{\ell \xi}(s_6))$ results in only a 0.8% decrease in $\aleph_g(\mathbf{J}_{\ell \xi}(L))$, however, the design that maximizes $\aleph_g(\mathbf{J}_{\ell \xi}(s_4))$ results in a 90% decrease in $\aleph_g(\mathbf{J}_{\ell \xi}(L))$. Choosing the string routing design that maximizes $\aleph_g(\mathbf{J}_{\ell \xi}(s_4))$ would therefore tend to increase the pose error at $s = L$. We also note that $\aleph_g(\mathbf{J}_{\ell \xi}(s_4))$ only increases by 4.6% between the end disk routing and the fourth disk routing, indicating that we would

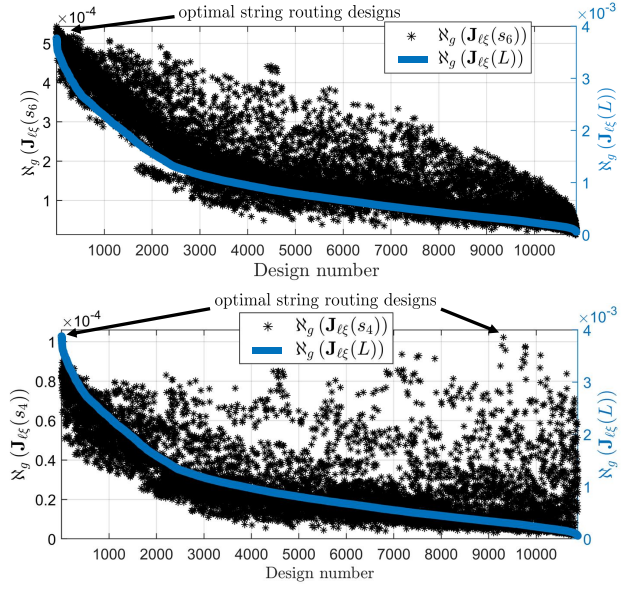


Fig. 15. The noise amplification indices across all physically realizable string routing designs for the simulated soft robot with torsional deflections. The routing that maximizes $\aleph_g(\mathbf{J}_{\ell \xi}(s_4))$ results in a significantly reduced $\aleph_g(\mathbf{J}_{\ell \xi}(L))$, but for disk 6, $\aleph_g(\mathbf{J}_{\ell \xi}(L))$ does not significantly change.

not expect to see a significant change in the pose error at s_4 between these two designs. We will now validate these predicted behaviors in a simulation study.

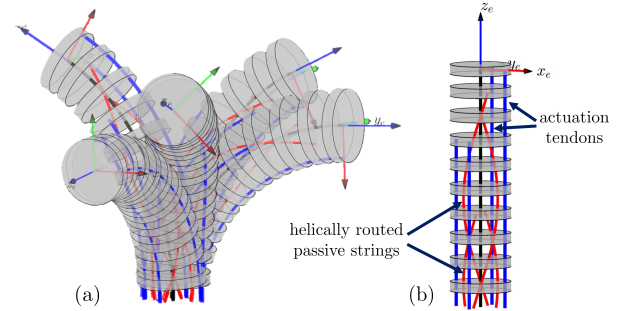


Fig. 16. (a) A subset of the spatial configurations used to validate our shape sensing approach on a segment subject to torsional loads utilizing helical routing, and (b) the segment in its zero-curvature configuration.

We simulated the robot in Fig. 14 using the Cosserat rod model from [36]. This model takes as inputs the applied tensions on the actuation tendons as well as the external forces/moments applied to the tip of the segment and returns the curvature, shear, and extension of the segment. We simulated the robot with a preload force of 25 N applied on all four tendons, and applied additional forces of up to 300 N on the tendons to bend the segment in 8 different directions. For each direction, we applied 12 different external wrenches with forces of ± 20 N and moments of ± 4 Nm, expressed in the world frame. We selected these loads to generate a large variety of shapes without exceeding the maximum curvature to keep the segment within its admissible workspace. We also included constant forces of 3.3 N on the helical strings due to the constant-torque spring in the string encoder housing.

For each tendon tension/external wrench combination, we started with the segment initially unloaded and incremented the external wrench with 5 steps to incrementally apply the load and solve for the shape at each external wrench increment. We did this to ensure that a good initial guess was provided to the solver. This simulation resulted in 480 different configurations of the continuum segment. For each configuration, we stored the string/tendon lengths, and then used these lengths as inputs to our kinematic shape sensing model to compare the accuracy of our shape sensing approach. Our unoptimized MATLAB 2019b implementation used MATLAB's *lsqnonlin()* to solve (16) for \mathbf{c} at a rate of ~ 4 Hz on average, using an 80 point trapezoid rule to compute $\mathbf{J}_{\ell c}$. Future code can be orders of magnitude faster with direct implementation in C++.

We compared the mechanics model to two different shape sensing models. The first used all 8 length measurements (4 string encoders and 4 actuation tendons) with the modal basis given by (52). The second used only the 4 actuation tendons to compare our shape sensing model to a scenario without any helical shape sensing strings. For this second model, the modal basis has 4 columns and is given by (40), with the modal functions given by $\phi_x(s) = \phi_y(s) = [T_0, T_1(s)]^T$.

TABLE VI

AVERAGE (MAXIMAL) ABSOLUTE POSITION AND ORIENTATION ERRORS IN MM AND $^\circ$ FOR THE SEGMENT IN FIG. 14. ERRORS ARE SPECIFIED FOR THE END DISK ($s = L$) AND THE 4th DISK ($s = s_4$)

	End Disk Routing	Fourth Disk Routing	w/o Passive Strings
$\bar{p}_e(L), \max(p_e(L))$	1.29 (3.39)	1.94 (5.64)	5.61 (32.87)
$\bar{p}_e(s_4), \max(p_e(s_4))$	0.45 (0.90)	0.52 (1.51)	0.63 (3.07)
$\bar{\theta}_e(L), \max(\theta_e(L))$	1.17 (3.04)	6.72 (21.78)	4.55 (23.44)
$\bar{\theta}_e(s_4), \max(\theta_e(s_4))$	0.61 (1.55)	0.57 (1.20)	2.15 (12.17)

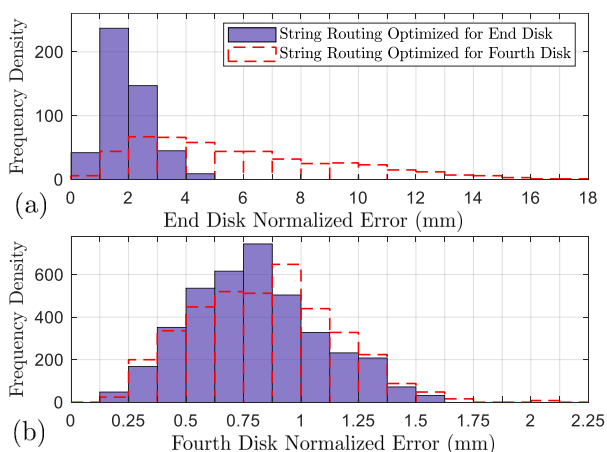


Fig. 17. Histograms of normalized pose error at the end disk and at the fourth intermediate disk for the routing that maximizes $\aleph_g(\mathbf{J}_{\ell\xi}(L))$ and for the routing that maximizes $\aleph_g(\mathbf{J}_{\ell\xi}(s_4))$. The pose error at the fourth intermediate disk is not significantly effected by the change in routing design, but the tip pose error is significantly effected due to the larger change in $\aleph_g(\mathbf{J}_{\ell\xi}(L))$ between the two designs.

The statistical results are given in Table VI, where we denote the string routing design that maximizes $\aleph_g(\mathbf{J}_{\ell\xi}(L))$

as the *end disk routing* and the string routing design that maximizes $\aleph_g(\mathbf{J}_{\ell\xi}(s_4))$ as the *fourth disk routing*. Both optimized string routing designs had maximum absolute end disk position errors below 2% of arc length. However, the fourth disk routing resulted in a 616% increase in $\max(\theta_e(L))$ compared to the end disk routing. There was also a 474% increase in $\bar{\theta}_e(L)$, and small increases in both $\bar{p}_e(L)$ and $\max(p_e(L))$ when compared to the fourth disk routing. This confirms the expected behavior due to the large decrease in $\aleph_g(\mathbf{J}_{\ell\xi}(L))$ in the fourth disk routing given in Table V.

Figure 17 shows the histograms of the error at the end disk and the fourth disk, normalized using (50). We observe that the end disk error with the end disk routing is shifted leftward compared to the fourth disk routing errors, as expected due to the large change in $\aleph_g(\mathbf{J}_{\ell\xi}(L))$. Furthermore, since $\aleph_g(\mathbf{J}_{\ell\xi}(s_4))$ did not significantly change, we do not see a significant difference in the error distribution for the fourth disk pose, and in fact see a small shift rightward with the fourth disk routing. This increase in error using the fourth disk routing is explained by the known fact that kinematic conditioning indices are not guaranteed to directly correlate with the true errors (see [51]). Large changes in the conditioning index should be sought to increase the possibility of reducing the true errors.

Compared to the model without string measurements, the end disk routing reduced the maximum end disk position error by 90% and the maximum angular end disk error by 87%. The fourth disk routing reduced the fourth disk maximum position error by 51% and the fourth disk maximum angular error by 90%. These simulation results demonstrate that 4 passive string (together with the four actuation tendons) can significantly improve the accuracy of continuum robot kinematics over actuation-based sensing alone.

VII. CONCLUSIONS

In this paper, we have presented a Lie group kinematic formulation for capturing variable curvature deflections of continuum robots using general string encoder routing. We used this formulation for a sensitivity analysis of the error propagation from error in string extension measurements to error in the modal coefficients and error in the central backbone shape. This analysis allows the designer to avoid string encoder routings that lead to ill-conditioned Jacobians. We then applied the approach on a planar example, a segment with high torsional stiffness, and a robot subject to torsional deflections using helical routing, showing that this shape sensing approach can result in mean and maximal absolute position error below 2% and 5% of arc length, respectively.

Our results provided several simple design guidelines for routing the strings to improve numerical conditioning, which we summarize here. For a planar segment, the strings should not be anchored at the same disk, and the pitch radius should be maximized. For a segment with high torsional stiffness, no more than two strings should be anchored to a disk, and if two strings are anchored to a disk, their anchor points should not be collinear in the radial direction of the disk. For segments with general deflections including torsion, simple design rules are

more difficult to define, but the numerical conditioning can be improved using our proposed design optimization procedure.

This sensing approach utilizes standard mechanical and electrical components, providing a relatively low-cost way of sensing the variable curvature deflections of large continuum robots, and the kinematic formulation and analysis presented herein enables practitioners to design string routings that improve the numerical conditioning of shape sensing. We have shown that four string encoders can provide accurate shape sensing, and have demonstrated a physical embodiment of a segment utilizing this approach, but a drawback of this approach compared to other sensing methods is the physical space required to mount the string encoders within the robot. To overcome this drawback, directions for future work include investigating more tightly integrated design of the string encoder mechanical components into the robot body or mounting of the string encoder housing remotely outside of the robot body. Future work also includes evaluation on multi-segment continuum robots and using this shape sensing approach and Lie group kinematic formulation to provide updates to continuum robot mechanics models.

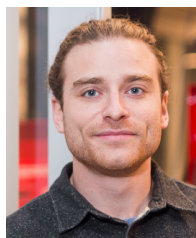
ACKNOWLEDGEMENTS

Thank you to Zachary Taylor for help with the code to read and calibrate the string encoders.

REFERENCES

- [1] S. Song, Z. Li, H. Yu, and H. Ren, "Electromagnetic positioning for tip tracking and shape sensing of flexible robots," *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4565–4575, 2015.
- [2] D. B. Camarillo, K. E. Loewke, C. R. Carlson, and J. K. Salisbury, "Vision based 3-D shape sensing of flexible manipulators," in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 2940–2947.
- [3] A. Reiter, R. E. Goldman, A. Bajo, K. Iliopoulos, N. Simaan, and P. K. Allen, "A learning algorithm for visual pose estimation of continuum robots," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 2390–2396.
- [4] A. Reiter, A. Bajo, K. Iliopoulos, N. Simaan, and P. K. Allen, "Learning-based configuration estimation of a multi-segment continuum robot," in *2012 4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechanics (BioRob)*, 2012, pp. 829–834.
- [5] R. Reilink, S. Stramigioli, and S. Misra, "3D position estimation of flexible instruments: marker-less and marker-based methods," *International Journal of Computer Assisted Radiology and Surgery*, vol. 8, no. 3, pp. 407–417, 2013.
- [6] Y. Shapiro, G. Kósa, and A. Wolf, "Shape tracking of planar hyperflexible beams via embedded PVDF deflection sensors," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 4, pp. 1260–1267, 2014.
- [7] Y.-L. Park, S. Elayaperumal, B. Daniel, S. C. Ryu, M. Shin, J. Savall, R. J. Black, B. Moslehi, and M. R. Cutkosky, "Real-time estimation of 3-D needle shape and deflection for MRI-guided interventions," *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 6, pp. 906–915, 2010.
- [8] R. Xu, A. Yurkewich, and R. V. Patel, "Curvature, torsion, and force sensing in continuum robots using helically wrapped FBG sensors," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 1052–1059, 2016.
- [9] R. J. Roesthuis, M. Kemp, J. J. van den Dobbelsteen, and S. Misra, "Three-dimensional needle shape reconstruction using an array of fiber Bragg grating sensors," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 4, pp. 1115–1126, 2014.
- [10] T. Clements and C. Rahn, "Three-dimensional contact imaging with an actuated whisker," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 844–848, 2006.
- [11] D. Trivedi and C. D. Rahn, "Model-based shape estimation for soft robotic manipulators: The planar case," *Journal of Mechanisms and Robotics*, vol. 6, no. 2, p. 021005, May 2014.
- [12] W. S. Rone and P. Ben-Tzvi, "Multi-segment continuum robot shape estimation using passive cable displacement," in *2013 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, 2013, pp. 37–42.
- [13] K. Xu, Y. Chen, Z. Zhang, S. Zhang, N. Xing, and X. Zhu, "An insertable low-cost continuum tool for shape sensing," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. Macau: IEEE, Dec. 2017, pp. 2044–2049.
- [14] C. G. Frazelle, A. Kapadia, and I. Walker, "Developing a kinematically similar master device for extensible continuum robot manipulators," *Journal of Mechanisms and Robotics*, vol. 10, no. 2, 02 2018, 025005.
- [15] L. Wang and N. Simaan, "Geometric calibration of continuum robots: Joint space and equilibrium shape deviations," *IEEE Transactions on Robotics*, vol. 35, no. 2, pp. 387–402, 2019.
- [16] A. Nahvi and J. M. Hollerbach, "The noise amplification index for optimal pose selection in robot calibration," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 1, 1996, pp. 647–654.
- [17] L. U. Odhner and A. M. Dollar, "The smooth curvature model: An efficient representation of Euler–Bernoulli flexures as robot joints," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 761–772, 2012.
- [18] F. Renda, C. Armanini, V. Lebastard, F. Candelier, and F. Boyer, "A geometric variable-strain approach for static modeling of soft manipulators with tendon and fluidic actuation," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4006–4013, 2020.
- [19] F. Boyer, V. Lebastard, F. Candelier, and F. Renda, "Dynamics of continuum and soft robots: A strain parameterization based approach," *IEEE Transactions on Robotics*, vol. 37, no. 3, pp. 847–863, 2020.
- [20] A. L. Orekhov and N. Simaan, "Solving Cosserat rod models via collocation and the Magnus expansion," in *IEEE/RSJ International Conference on Robots and Systems (IROS)*, Oct. 2020.
- [21] R. Jödicke, U. Jungnickel, and A. Müller, "Lie group modeling of nonlinear helical beam elements," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 6, Aug 2014, V006T10A030.
- [22] G. S. Chirikjian and J. W. Burdick, "A modal approach to hyper-redundant manipulator kinematics," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 3, pp. 343–354, 1994.
- [23] S. H. Sadati, S. E. Naghibi, I. D. Walker, K. Althoefer, and T. Nanayakkara, "Control space reduction and real-time accurate modeling of continuum manipulators using Ritz and Ritz–Galerkin methods," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 328–335, 2017.
- [24] L. Wang and N. Simaan, "Investigation of error propagation in multi-backbone continuum robots," in *Advances in Robot Kinematics*. Springer, 2014, pp. 385–394.
- [25] M. W. Hannan and I. D. Walker, "Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots," *Journal of Robotic Systems*, vol. 20, no. 2, pp. 45–63, 2003.
- [26] W. McMahan, V. Chitrakaran, M. Csencsits, D. Dawson, I. D. Walker, B. A. Jones, M. Pritts, D. Dienno, M. Grissom, and C. D. Rahn, "Field trials and testing of the OctArm continuum manipulator," in *2006 IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 2336–2341.
- [27] W. Felt, M. J. Telleria, T. F. Allen, G. Hein, J. B. Pompa, K. Albert, and C. D. Remy, "An inductance-based sensing system for bellows-driven continuum joints in soft robots," *Autonomous robots*, vol. 43, no. 2, pp. 435–448, 2019.
- [28] J. Ding, R. E. Goldman, K. Xu, P. K. Allen, D. L. Fowler, and N. Simaan, "Design and coordination kinematics of an insertable robotic effectors platform for single-port access surgery," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 5, pp. 1612–1624, 2012.
- [29] N. Sarli, G. Del Giudice, S. De, M. S. Dietrich, S. D. Herrell, and N. Simaan, "TURBot: A system for robot-assisted transurethral bladder tumor resection," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 4, pp. 1452–1463, 2019.
- [30] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017.
- [31] A. Iserles, H. Z. Munthe-Kaas, S. P. Nørsett, and A. Zanna, "Lie-group methods," *Acta Numerica*, vol. 9, pp. 215–365, Jan. 2000.
- [32] P. S. Gonthina, A. D. Kapadia, I. S. Godage, and I. D. Walker, "Modeling variable curvature parallel continuum robots using Euler curves," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 1679–1685.
- [33] P. Rao, Q. Peyron, and J. Burgner-Kahrs, "Using Euler curves to model continuum robots," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1402–1408.

- [34] J. C. Mason and D. C. Handscomb, *Chebyshev Polynomials*. CRC Press, 2002.
- [35] R. J. Webster III and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010.
- [36] D. C. Rucker and R. J. Webster, "Statics and dynamics of continuum robots with general tendon routing and external loading," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1033–1044, Dec. 2011.
- [37] W. Rossmann, *Lie Groups: An Introduction Through Linear Groups*. Oxford University Press on Demand, 2006, vol. 5.
- [38] J. M. Selig, *Geometric Fundamentals of Robotics*. Springer Science & Business Media, 2004.
- [39] C. Abah, A. L. Orekhov, G. L. H. Johnston, and N. Simaan, "A multi-modal sensor array for human–robot interaction and confined spaces exploration using continuum robots," *IEEE Sensors Journal*, vol. 22, no. 4, pp. 3585–3594, 2022.
- [40] F. Renda, M. Giorelli, M. Calisti, M. Cianchetti, and C. Laschi, "Dynamic model of a multibending soft robot arm driven by cables," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1109–1122, 2014.
- [41] P. Cardou, S. Bouchard, and C. Gosselin, "Kinematic-sensitivity indices for dimensionally nonhomogeneous Jacobian matrices," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 166–173, 2010.
- [42] J. Zhang, J. Roland, J. Thomas, S. Manolidis, and N. Simaan, "Optimal path planning for robotic insertion of steerable electrode arrays in cochlear implant surgery," *Journal of Medical Devices*, vol. 3, no. 1, 12 2008.
- [43] N. Simaan, R. Taylor, and P. Flint, "A dexterous system for laryngeal surgery," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 1, 2004, pp. 351–357.
- [44] X. Dong, M. Raffles, S. Cobos-Guzman, D. Axinte, and J. Kell, "A novel continuum robot using twin-pivot compliant joints: design, modeling, and validation," *Journal of Mechanisms and Robotics*, vol. 8, no. 2, p. 021010, 2016.
- [45] J. A. Childs and C. Rucker, "Concentric precurved bellows: new bending actuators for soft robots," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1215–1222, 2020.
- [46] J. Santoso and C. D. Onal, "An origami continuum robot capable of precise motion through torsionally stiff body and smooth inverse kinematics," *Soft Robotics*, Jul. 2020.
- [47] J. A. Childs and C. Rucker, "Leveraging geometry to enable high-strength continuum robots," *Frontiers in Robotics and AI*, vol. 8, 2021.
- [48] V. C. Anderson and R. C. Horn, "Tensor arm manipulator," Feb. 24 1970, US Patent 3,497,083.
- [49] G. L. Johnston, A. L. Orekhov, and N. Simaan, "Kinematic modeling and compliance modulation of redundant manipulators under bracing constraints," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4709–4716.
- [50] C. Abah, A. L. Orekhov, G. L. Johnston, P. Yin, H. Choset, and N. Simaan, "A multi-modal sensor array for safe human-robot interaction and mapping," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3768–3774.
- [51] J. P. Merlet, "Jacobian, manipulability, condition number, and accuracy of parallel robots," *Journal of Mechanical Design*, vol. 128, no. 1, pp. 199–206, Jun. 2005.



Elan Z. Ahronovich received his B.S. degree in biological sciences from the Ohio State University, Columbus, OH, USA, in 2014 and his B.S. degree in mechanical engineering from Virginia Polytechnic Institute and State University, Blacksburg, VA, USA, in 2019. His current research interests include collaborative surgical robotics and soft continuum manipulators.



Nabil Simaan (F'20) received his Ph.D. degree in mechanical engineering from the Technion—Israel Institute of Technology, Haifa, Israel, in 2002. During 2003, he was a Postdoctoral Research Scientist at Johns Hopkins University National Science Foundation (NSF) ERC-CISST. In 2005, he joined Columbia University, New York, NY. In 2009 he received the NSF Career award for young investigators to design new algorithms and robots for safe interaction with the anatomy. In Fall 2010 he joined Vanderbilt University. In 2020 he was named an IEEE Fellow for contributions to dexterous continuum robotics. In 2021 he was elected Fellow of the ASME for contributions to continuum and soft robotics for surgery. His research interests include medical robotics, kinematics, robot modeling and control and human-robot interaction.



Andrew L. Orekhov received the B.S. degree in mechanical engineering from the University of Tennessee, Knoxville, TN, in 2016, and the Ph.D. in mechanical engineering from Vanderbilt University, Nashville, TN, in 2022. He is currently a Postdoctoral Fellow in the Robotics Institute at Carnegie Mellon University and his research interests include design, modeling, and control of continuum manipulators and snake robots. He received the NSF Graduate Research Fellowship in 2016.