Shape Control of Concentric Tube Robots via Approximate Follow-the-Leader Motion

Yunti Xu, Connor Watson, Jui-Te Lin, John T. Hwang, and Tania K. Morimoto, *Senior Member, IEEE*

***Abstract*—Concentric tube robots (CTRs) are miniaturized continuum robots that are promising for robotic minimally invasive surgeries. Control methods to date have primarily focused on controlling the robot tip. However, small changes in the tip position can result in large deviations in the shape of the robot body, motivating the need for shape control to ensure safe navigation in constrained environments. One proposed method for shape control, known as follow-the-leader (FTL) motion, allows the robot to deploy while occupying minimal volume but is limited to specific CTR designs and deployment sequences. In this letter, we propose a shape control method that approximates FTL motion and is applicable to arbitrary tip navigation tasks without requiring a predefined trajectory or specific tube design. This shape control method is framed as a nonlinear optimization problem, and through linearization of the CTR's kinematics, we turn it into a quadratic program solved by a shape controller that requires minimal knowledge of the robot's shape. Simulations show that our method enables better approximate FTL motion compared to a state-of-the-art Jacobian-based tip controller across different tube sets and tip paths while remaining computationally comparable. Furthermore, a hardware demonstration validates the effectiveness of the shape controller on a physical system during teleoperation.**

***Index Terms*—Surgical robotics: steerable catheters/needles, modeling, control, and learning for soft robots, medical robots and systems.**

I. Introduction

Robotic minimally invasive surgery (RMIS) has the potential to improve patient outcomes by minimizing negative surgical impacts [1]. However, these procedures often require navigation through winding, tortuous anatomy. Continuum robots, characterized by their continuously bending structure, offer high dexterity and the ability to maneuver around obstacles, making them promising for use in RMIS [2].

The concentric tube robot (CTR) represents a subset of continuum robots, consisting of pre-curved, elastic, telescoping tubes typically with diameters around 1 mm [3], [4]. The tubes are externally actuated at their base in translation and rotation, enabling the overall shape of the robot to change. Due to their compact size, CTRs have been considered for various minimally invasive surgeries, such as ENT [5], [6], transnasal [7], and intracerebral hemorrhage evacuation surgeries [2]. However, small changes in the tip position can result in large changes in the robot's backbone shape, making deployment through sensitive environments challenging [8], particularly during teleoperation where the deployment sequence is not known *a priori*. This challenge is especially relevant for needle steering applications through hollow lumens, where avoiding large motions of the backbone is crucial. In response, we propose a shape control method that minimizes shape changes to the robot's backbone, allowing the user to focus solely on teleoperating the tip (Fig. 1).

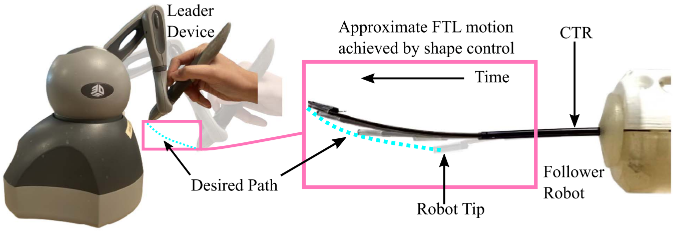

Fig. 1. Illustration of the approximate FTL motion achieved via the proposed shape control method in a teleoperation setup.

A. Control of CTRs

State-of-the-art control methods for CTRs primarily focus on controlling the CTR tip [9]. Typically these tip navigation tasks can be categorized as either autonomous tracking of a predefined tip path [10], [11], or as teleoperation, where the tip path is not defined *a priori* [7], [12]. A common approach is to use Jacobian-based methods, such as the damped-least-squares (DLS) method, originally developed for serial manipulators [13]. These methods have been applied for real-time CTR tip control by using efficient Jacobian computation methods based either on solving an initial value problem [14] or through approximation via Fourier series [15].

Beyond tip control, alternative Jacobian-based control strategies for CTRs use redundancy resolution to integrate additional kinematic tasks alongside tip navigation. One such strategy is the generalized damped-least-squares method, which employs weighting matrices to create a control scheme balancing multiple tasks [16]. This approach has been employed in CTRs to account for joint constraints and prevent instabilities [17]. Another strategy is to use nullspace projections to create a hierarchical control structure that integrates both higher and lower-priority tasks. Iterative nullspace projections of the Jacobians related to higher-priority tasks are used to also achieve secondary tasks, such as

Manuscript received 5 February 2024; accepted 6 June 2024. Date of publication 26 June 2024; date of current version 4 July 2024. This letter was recommended for publication by Associate Editor M. T. Chikhaoui and Editor J. Burgner-Kahrs upon evaluation of the reviewers' comments. This work was supported by National Science Foundation under Grant 2146095. *(Corresponding author: Yunti Xu.)*

The authors are with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, CA 92093 USA (e-mail: ytxu@ucsd.edu; cmwatson@eng.ucsd.edu; jul024@ucsd.edu; jhwang@ucsd.edu; tamorimoto@ucsd.edu).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3419649, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3419649

avoiding instabilities and enhancing manipulability [18], as well as preventing collisions in dual arm CTR configurations [19].

Despite the wide adoption of Jacobian-based methods for CTR control, a limitation of this approach is that constraints may still be violated since they are accommodated implicitly using suitable penalty functions [17], rather than enforced as explicit mathematical constraints within an optimization program. To this end, an alternative approach is to formulate the control problem as a constrained nonlinear optimization problem, where constraints, such as joint limits, are enforced explicitly [10], [12]. This approach balances tip control with additional objectives, including avoiding collision with the anatomy [12] or avoiding undesirable robot configurations [10]. Nevertheless, this approach requires a highly complex parallel system architecture to achieve computation times fast enough for teleoperation scenarios [12].

B. Follow-the-Leader (FTL) Motion

One approach to controlling the shape of a CTR is to use follow-the-leader (FTL) motion, where the robot's body precisely traces the path traversed by its tip. Originally introduced by Choset et al. [20] for hyper-redundant robots, FTL motion allows the robot to occupy a minimal volume during deployment, which is beneficial for navigating confined spaces. For CTRs, achieving perfect FTL motion is feasible only for specific combinations of tube geometries and sequences of joint values, as highlighted by Gilbert et al. [21]. Near-perfect FTL motion, alternatively, can be achieved by CTRs that have minimal torsional interaction—as demonstrated in Casanovas et al. [22]—but these results do not hold for more general CTR designs. To address a wider range of robot designs and joint motions, we consider the problem of approximate FTL motion, which attempts to minimize the occupied volume of a general CTR design during its deployment. Approximate FTL motion can be useful in applications such as navigation through a hollow lumen, where a tolerance exists between the robot and surrounding walls [21]. To the best of our knowledge, this problem has previously only been analyzed offline. Our work is the first to propose a method for shape control via approximating FTL motion while following any tip path for an arbitrary tube set.

C. Contributions

The contributions of this letter are as follows. First, we introduce a formulation of CTR shape control to approximate FTL motion as a constrained nonlinear optimization problem. This approach applies to arbitrary tip navigation tasks, focusing on cases where the tip path is not predefined, and is not limited to specific tube designs. Second, we propose a shape controller that solves an approximated version of the nonlinear optimization problem via quadratic programming (QP), assuming minimal knowledge of the robot's shape. We note that the proposed algorithm is agnostic to how the desired robot shape is constructed, but here we focus specifically on the desired robot shape being the history of past tip position commands (i.e. approximate FTL motion). Then, we present simulation results that show that indeed our QP-based shape controller produces better approximate FTL motion compared to a state-of-the-art Jacobian-based tip controller across different tube sets and tip paths, while remaining computationally comparable. Finally, we demonstrate the effectiveness of our shape controller on a hardware system during teleoperation. Overall, by enabling approximate FTL motion, our proposed approach can help enhance the capabilities and safety of CTRs within narrow environments.

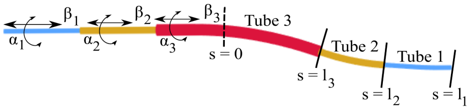

Fig. 2. Actuation parameters for a 3-tube CTR. The insertion point, $s = 0$, separates the actuation section from the active section of the CTR.

II. CTR Kinematics

CTRs consist of n concentrically nested tubes that can be rotated and translated at the base (see Fig. 2). Their actuation is represented by the joint values $\mathbf{q} = [l_1, \ldots, l_n, \alpha_1, \ldots, \alpha_n]^T \in \mathbb{R}^{2n}$, where l_i and α_i denote the deployment length and tube rotation at the base of tube i, respectively. The transmission length of tube i is given by, $\beta_i = l_i - L_i$, where L_i is the length of tube i. Each tube can be characterized by its geometric and material properties. Specifically, $L_{s,i}$ and $L_{c,i}$ are the lengths of the straight and curved sections of each tube, respectively, and $L_i = L_{s,i} + L_{c,i}$. The fixed pre-curvature of the curved section is denoted by κ_i. The inner and outer diameters of the tube are given as ID_i and OD_i, respectively, while E_i and G_i denote the elastic modulus and shear modulus.

The mapping from joint values to the CTR shape in 3D can be described by the forward kinematics model. In the absence of friction, external loads, and pre-torsion of the tubes, this mapping is governed by a set of differential equations with respect to the arclength variable s, and is subject to boundary conditions as follows [23]:

$$\ddot{\psi}_i = \frac{k_{ib}}{k_{it}k_b}\sum_{j=1}^{n} k_{jb}\kappa_i\kappa_j \sin(\psi_i - \psi_j)$$

$$\begin{cases} \psi_i(0) = \alpha_i - \beta_i\dot{\psi}_i(0) \\ \dot{\psi}_i(l_i) = 0 \end{cases} \tag{1}$$

where $k_{ib} = \frac{E_i(OD_i^4 - ID_i^4)\pi}{64}$ and $k_{it} = \frac{G_i(OD_i^4 - ID_i^4)\pi}{32}$ are the bending and torsional stiffnesses, respectively. The total bending stiffness is $k_b = \sum_{i=1}^{n} k_{ib}$, where n is the number of tubes in the considered CTR link. The term ψ_i denotes the axial rotation of the ith tube. The first boundary condition considers known tube angles grasped by the actuator at the proximal end, while the second boundary condition accounts for no torsion at the free end of each tube. The centerline position (i.e. shape) of the CTR, $\boldsymbol{\xi}(s) \in \mathbb{R}^3$, is defined by a set of ODEs with respect to s [23],

$$\begin{cases} \dot{\mathbf{R}} = \mathbf{R}\hat{\mathbf{u}} \\ \dot{\boldsymbol{\xi}} = \mathbf{R}\mathbf{e_3} \end{cases} \tag{2}$$

where $\mathbf{e_3}$ is the unit vector in the $z-$direction tangent to the robot backbone, $\hat{\mathbf{u}}$ is the skew-symmetric matrix of $\mathbf{u}$, which is the robot curvature vector computed using the solution of (1), and $\mathbf{R} \in \mathcal{SO}(3)$ is the backbone orientation. This integration is solved with the initial conditions $\mathbf{R}(0) = \mathbf{R_z}(\psi_1(0))$ and $\boldsymbol{\xi}(0) = \mathbf{0}_{3\times1}$. The shape of the CTR, $\boldsymbol{\xi}(s)$, along with the tip position, $\boldsymbol{p} = \boldsymbol{\xi}(l_1)$, are used to formulate the cost functions in the proposed shape control method.

III. Shape Control Method

In this section, we detail how to approximate FTL motion by formulating a nonlinear optimization problem and propose an algorithm for its solution. We explain the choice of objective function and propose a shape controller that solves the optimization via quadratic programming.

A. Problem Formulation

We propose framing the approximate FTL problem as an optimization problem that involves a robot design and a sequence of desired waypoints for the robot tip to reach. At each step k, a new waypoint, $\mathbf{p}_k^* \in \mathbb{R}^3$, is given, and the objective is to determine a local optimal change in joint values, $\Delta\mathbf{q}_k^* \in \mathbb{R}^{2n}$, from the current joint values, $\mathbf{q}_k \in \mathbb{R}^{2n}$, that minimizes both the tip error with respect to the new waypoint and the shape error between the backbone centerline and the desired tip path commanded so far, $\boldsymbol{\xi}_k^* = [\mathbf{p}_1^*, \ldots, \mathbf{p}_k^*] \in \mathbb{R}^{3\times k}$. To generalize this formulation, we consider only change in joint values, $\Delta\mathbf{q}_k$, as a design variable, and treat each new waypoint ($\mathbf{p}_k^*$), the tube set, and robot base pose as user-defined values for the optimization.

The proposed algorithm (Algorithm 1) starts with fixed initial joint values ($\mathbf{q}_{init}$) and for each waypoint, it calls a shape control scheme (see Section III-C) that solves the optimization problem. The tip error, $e_{tip,k}$—defined as the 2-norm between the current tip position and current waypoint—is computed. If this tip error exceeds a predefined threshold, $\tilde{e}_{tip}$, the algorithm reverts to a Jacobian-based tip controller to lower the error. Finally, the joint values are updated.

Algorithm 1: Pseudocode for shape control algorithm.

```
Input:
$\mathbf{p}_k^*$: desired tip waypoint
$\mathbf{q}_{init}$: initial joint values
$\Delta\mathbf{q}_{init}$: initial change in joint values
Output:
$\mathbf{q}_k$: joint values for each waypoint
1: $\mathbf{q}_1 \leftarrow \mathbf{q}_{init}$
2: for each waypoint, $\mathbf{p}_k^*$ do
3:     $\boldsymbol{\xi}_k^* \leftarrow [\boldsymbol{\xi}_k^*, \mathbf{p}_k^*]$
4:     $\Delta\mathbf{q}_k \leftarrow \Delta\mathbf{q}_{init}$
5:     $\Delta\mathbf{q}_k^* \leftarrow ShapeControl(\mathbf{q}_k, \Delta\mathbf{q}_k, \mathbf{p}_k^*, \boldsymbol{\xi}_k^*)$
6:     $e_{tip,k} \leftarrow \|\mathbf{p}_k(\mathbf{q}_k + \Delta\mathbf{q}_k^*) - \mathbf{p}_k^*\|$
7:     if $e_{tip,k} > \tilde{e}_{tip}$ then
8:         $\Delta\mathbf{q}_k^* \leftarrow TipPositionControl(\mathbf{q}_k, \mathbf{p}_k^*)$
9:     end if
10:    $\mathbf{q}_{k+1} \leftarrow \mathbf{q}_k + \Delta\mathbf{q}_k^*$
11: end for
```

B. Objective Function

The objective function consists of four subfunctions: f_{tip}, f_{shape}, f_{deploy}, and f_{rot}. Note that the following derivations consider the kth step without loss of generality, thus for notational brevity, the subscript k is dropped. The tip objective function, f_{tip}, aims to navigate the robot's tip to the desired waypoint by minimizing the error between the current tip position, $\mathbf{p}(\mathbf{q} + \Delta\mathbf{q})$, and the current waypoint, $\mathbf{p}^*$:

$$f_{tip}(x) = \frac{1}{2}\left(\frac{\|\mathbf{p}(\mathbf{q} + \Delta\mathbf{q}) - \mathbf{p}^*\|}{\|\mathbf{p}_0\|}\right)^2, \tag{3}$$

where it is scaled by $\|\mathbf{p}_0\|$ for length-scale correction, which improves the generalizability by reducing manual tuning effort and normalizing the units. We selected $\|\mathbf{p}_0\|$ to be the initial tip error at the kth step prior to any optimization.

Next, f_{shape}, aims to align the robot's backbone shape, $\boldsymbol{\xi}(\mathbf{q} + \Delta\mathbf{q})$, with the desired tip path, $\boldsymbol{\xi}^*$, by minimizing the following shape error cost:

$$f_{\text{shape}}(x) = \frac{1}{2\,m}\sum_{i}^{m}\left(\frac{\|\boldsymbol{\xi}_i(\mathbf{q} + \Delta\mathbf{q}) - \boldsymbol{\xi}_i^*\|}{\|\boldsymbol{\xi}_0\|}\right)^2, \tag{4}$$

where the discretized shape error is defined as the mean Euclidean distance between the backbone and the tip path at m corresponding points, which exist at equal arclength s in the interval $[0, l_1]$ [24]. We note that the maximum m value, denoted by M, is the number of points discretizing l_1, which depends on the step size used in solving (1) and (2). This term is scaled by $\|\boldsymbol{\xi}_0\|$.

Since a large number of solutions may exist for the optimization problem, we introduce a regularizing heuristic term that selects a solution where the tubes are deployed evenly. Assuming a 3-tube CTR, this term penalizes significant differences between the tube deployment lengths since extreme tube configurations may affect the effectiveness of the shape control method:

$$f_{deploy}(x) = \frac{1}{2} \times \frac{((l_1+\Delta l_1)-(l_2+\Delta l_2))^2+((l_1+\Delta l_1)-(l_3+\Delta l_3))^2}{l_1^2}. \tag{5}$$

We normalize by the length of tube 1 to ensure the term remains bounded. Lastly, we add another regularizing heuristic term that penalizes large changes in tube rotations, $\Delta\alpha_i$, which could cause undesired oscillations during deployment,

$$f_{rot}(x) = \frac{1}{2}\left(\frac{\Delta\alpha_1^2 + \Delta\alpha_2^2 + \Delta\alpha_3^2}{\Delta\alpha_0^2}\right). \tag{6}$$

Overall, the objective function to minimize at the kth step is expressed as,

$$f(x) = c_1 f_{\text{tip}}(x) + c_2 f_{\text{shape}}(x) + c_3 f_{deploy}(x) + c_4 f_{rot}(x), \tag{7}$$

where the relative importance of each sub-objective is determined by the coefficients $c_i, i = 1, 2, 3, 4$. To ensure the outputs of the optimization, $\Delta\mathbf{q}$, are physically feasible, we enforce limit bounds on the deployment lengths,

$$-l_i \le \Delta l_i \le L_i - l_i, i = 1, 2, 3 \tag{8}$$

and linear inequality constraints to ensure that the tubes are at least $\epsilon > 0$ mm apart,

$$l_1 \le l_2 + \epsilon \le l_3 + \epsilon \tag{9}$$

C. Shape Controller

The computation time required to solve this nonlinear optimization problem would be too long for online control applications. We therefore transform the problem into a convex quadratic program (QP), which belongs to a special class of optimization problems that allows for online solvability and avoids local minima [25]. The standard convex quadratic program has the form: $h(x) = \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \boldsymbol{g}^T\mathbf{x}$, where $\mathbf{x}$ and $\boldsymbol{g}$ belong to $\mathbb{R}^n$, and $\mathbf{H}$ is an $n \times n$ symmetric, positive semidefinite matrix [25].

To reformulate, we first take a linear approximation of the CTR forward kinematics by deriving the first-order Taylor expansion of the kinematic model around the tip,

$$\mathbf{p}(\mathbf{q}+\Delta\mathbf{q}) \approx \mathbf{p}(\mathbf{q}) + \frac{\partial \mathbf{p}}{\partial \mathbf{q}}\Delta\mathbf{q} = \mathbf{p}(\mathbf{q}) + \mathbf{J}\Delta\mathbf{q}, \tag{10}$$

and around m points along the robot backbone,

$$\boldsymbol{\xi}_i(\mathbf{q}+\Delta\mathbf{q}) \approx \boldsymbol{\xi}_i(\mathbf{q}) + \frac{\partial \boldsymbol{\xi}_i}{\partial \mathbf{q}}\Delta\mathbf{q} = \boldsymbol{\xi}_i(\mathbf{q}) + \mathbf{J}_i\Delta\mathbf{q}. \tag{11}$$

Here, $\mathbf{J}$ and $\mathbf{J}_i$ are 3×6 matrices representing the Jacobians associated with the robot's tip and each point $i = 1$ through m, respectively. These Jacobians can be computed using various methods [12], [14]. For simplicity, we estimate them through a forward finite difference scheme using a step size of 0.1 mm for translation and 0.1 rd for rotation. By substituting (10) and (11) into (3) and (4), respectively, f_{tip} and f_{shape} can be approximated in a (constrained) least-squares sense and expanded as:

$$\begin{aligned} f_{tip}(x) &\approx \frac{1}{\|\mathbf{p}_0\|^2}\frac{1}{2}\|\mathbf{J}\Delta\mathbf{q} - \Delta\mathbf{p}\|^2 \\ &= \frac{1}{2}\Delta\mathbf{q}^T \underbrace{\frac{\mathbf{J}^T\mathbf{J}}{\|\mathbf{p}_0\|^2}}_{\mathbf{H}_{tip}} \Delta\mathbf{q} - \underbrace{\frac{\Delta\mathbf{p}^T\mathbf{J}}{\|\mathbf{p}_0\|^2}}_{\boldsymbol{g}_{tip}^T} \Delta\mathbf{q} + \underbrace{\frac{1}{2}\frac{\Delta\mathbf{p}^T\Delta\mathbf{p}}{\|\mathbf{p}_0\|^2}}_{\mathbf{c}_{tip}}, \end{aligned} \tag{12}$$

$$\begin{aligned} f_{\text{shape}}(x) &\approx \frac{1}{\|\boldsymbol{\xi}_0\|^2}\frac{1}{2\,m}\sum_i^m \|\mathbf{J}_i\Delta\mathbf{q} - \Delta\boldsymbol{\xi}_i\|^2 \\ &= \frac{1}{m}\sum_i^m \left(\frac{1}{2}\Delta\mathbf{q}^T \underbrace{\frac{\mathbf{J}_i^T\mathbf{J}_i}{\|\boldsymbol{\xi}_0\|^2}}_{\mathbf{H}_{\text{shape}}} \Delta\mathbf{q} - \underbrace{\frac{\Delta\boldsymbol{\xi}_i^T\mathbf{J}_i}{\|\boldsymbol{\xi}_0\|^2}}_{\boldsymbol{g}_{\text{shape}}^T} \Delta\mathbf{q} \right. \\ &\qquad \left. + \underbrace{\frac{1}{2}\frac{\Delta\boldsymbol{\xi}_i^T\Delta\boldsymbol{\xi}_i}{\|\boldsymbol{\xi}_0\|^2}}_{\mathbf{c}_{\text{shape}}}\right), \end{aligned} \tag{13}$$

where $\Delta\mathbf{p} := \mathbf{p}^* - \mathbf{p}(\mathbf{q})$ and $\Delta\boldsymbol{\xi}_i := \boldsymbol{\xi}_i^* - \boldsymbol{\xi}_i(\mathbf{q})$. Both quadratic weight matrices, $\mathbf{H}_{tip}$ and $\mathbf{H}_{\text{shape}}$, are symmetric and positive semidefinite. When we compare these quadratic objective terms to the standard form for convex QP, we observe that they are the same up to a constant, which can be disregarded as it does not affect the optimization process.

We rewrite f_{deploy} in quadratic form,

$$\mathbf{f}_{deploy}(x) = \frac{1}{2}\Delta\mathbf{q}^T\mathbf{H}_{deploy}\Delta\mathbf{q} + \boldsymbol{g}_{deploy}^T\Delta\mathbf{q} + \mathbf{c}_{deploy}$$

$$\mathbf{H}_{deploy} = \frac{1}{l_1^2}\left(\begin{array}{ccc|c} 2 & -1 & -1 & \\ -1 & 1 & 0 & \mathbf{0}_{3\times 3} \\ -1 & 0 & 1 & \\ \hline & \mathbf{0}_{3\times 3} & & \mathbf{0}_{3\times 3} \end{array}\right),$$

$$\boldsymbol{g}_{deploy}^T = \frac{1}{l_1^2}\left[2l_1 - l_2 - l_3 \quad l_2 - l_1 \quad l_3 - l_1 \quad \mathbf{0}_{1\times 3}\right]$$

$$\mathbf{c}_{deploy} = \frac{1}{2}\frac{(l_1 - l_2)^2 + (l_1 - l_3)^2}{l_1^2} \tag{14}$$

Again, we note that $\mathbf{H}_{deploy}$ is symmetric and positive semidefinite and the constant term can be neglected. The objective f_{rot} can be directly written in quadratic form,

$$f_{rot}(x) = \frac{1}{2}\Delta\mathbf{q}^T\mathbf{H}_{rot}\Delta\mathbf{q}, \mathbf{H}_{rot} = \frac{1}{\alpha_0^2}\left(\begin{array}{c|c} \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \mathbf{I}_{3\times 3}. \end{array}\right) \tag{15}$$

TABLE I
TUBE SET PARAMETERS

Tube no.	κ_i [1/mm]	$[L_{s,i}, L_{c,i}]$ [mm]	ID_i [mm]	OD_i [mm]	E [GPa]
		Tube set 1			
1	[0 0.0061]	[162 15]	0.650	0.880	80
2	[0 0.0131]	[71 50]	1.076	1.296	80
3	[0 0.0021]	[15 50]	1.470	2.180	80
		Tube set 2			
1	[0 0.0030]	[120 25]	0.50	1	80
2	[0 0.0090]	[65 50]	1.20	1.60	80
3	[0 0.0160]	[15 50]	1.80	2.50	80

Finally, we combine (12), (13), (14), and (15) to form an overall quadratic objective function,

$$\begin{aligned} &h(\Delta\mathbf{q}) \\ &= \frac{1}{2}\Delta\mathbf{q}^T \underbrace{\left(c_1\mathbf{H}_{tip} + c_2\mathbf{H}_{\text{shape}} + c_3\mathbf{H}_{deploy} + c_4\mathbf{H}_{rot}\right)}_{\mathbf{H}} \Delta\mathbf{q} \\ &\quad + \underbrace{\left(c_1\boldsymbol{g}_{tip}^T + c_2\boldsymbol{g}_{\text{shape}}^T + c_3\boldsymbol{g}_{deploy}^T\right)}_{\boldsymbol{g}^T} \Delta\mathbf{q}, \end{aligned} \tag{16}$$

subjected to (8) and (9). Since we consider a linearized, local kinematics relationship, the bounds on both the deployment lengths and tube rotations are adjusted as follows: -2 mm $\leq \Delta l_i \leq 2$ mm and $-\pi/4 \leq \alpha_i \leq \pi/4$. The selection of these numerical values is based on the proximity of sequential waypoints. The goal is to ensure that each waypoint can be reached through a feasible change in joint values, denoted as $\Delta\mathbf{q}$. The solution to the convex quadratic program is then used in the shape control scheme on line 5 in Algorithm 1. In our QP formulation, the robot kinematics are linearized, resulting in significant computational savings compared to solving the nonlinear kinematics at each iteration of the optimization process. We further note that a single evaluation of the kinematic model is sufficient to determine the tip and shape position errors, while an additional 6 evaluations are adequate to compute Jacobians (via finite difference).

IV. Simulation Experiments and Results

In this section, we conduct numerical experiments to assess how the tip and shape errors vary across different CTR designs and tip paths. We compare the performance of our shape controller with a standard tip controller.

A. Selection of Tube Sets and Tip Paths

To evaluate our QP-based shape control algorithm, we select two CTR tube sets (Table I), where each tube has one straight and one curved section. Specifically, tube set 1 is adopted from [26]. We generate a set of B-splines for each tube set by first sampling B total path points denoted as $\mathbf{v}_b^* \in \mathbb{R}^3, b = 0, \ldots, B$, from the robot's workspace, with increasing z-values, starting from an initial point $\mathbf{v}_0^*$. We construct the set $\mathbf{V}^* = \{\mathbf{v}_0^*, \mathbf{v}_1^*, \ldots, \mathbf{v}_b^*, \ldots, \mathbf{v}_B^*\}$ such that the difference in z-values between successive path points $\mathbf{v}_b$ and $\mathbf{v}_{b+1}$ randomly varies within a specified lower and upper bound, which we set to be $[\lfloor\frac{z_{\max}-z_{\min}}{B}\rfloor, \lfloor\frac{z_{\max}-z_{\min}}{B-1}\rfloor]$, where $z_{\max}$ and $z_{\min}$ are the maximum and minimum z values in the robot's workspace, and $B = 3$. Next, we fit a smoothing cubic B-spline [27] through this set of path points $\mathbf{V}^*$. In the case of a smoothing B-spline,

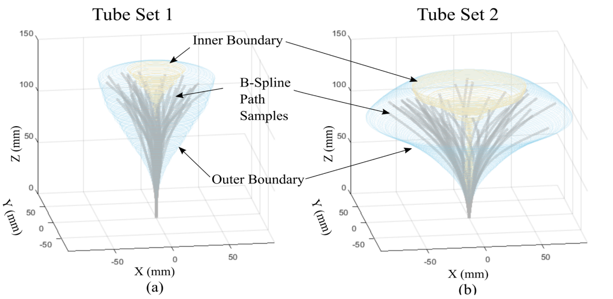

Fig. 3. (a) A subset of general B-spline paths generated for tube set 1 and (b) tube set 2. The inner and outer boundaries of the robot workspace are also shown. These paths are then discretized to form waypoints.

$\mathbf{f}_s$, its control points are computed by minimizing the functional L:

$$L := \mu \sum_{b=0}^{B} \omega_r \|\mathbf{f}_{s,b} - \mathbf{v}_b^*\|^2 + (1-\mu) \int \left| \frac{d\mathbf{f}_s^2}{du^2} \right|^2 du \qquad (17)$$

where the parameters $\mu \in [0, 1]$ trade-off between fitting the spline close to each path point and the overall smoothness of the spline, and ω_b determines the local importance of each path point [27]. For our simulations, we set $\mu = \frac{1}{6(10^{-4})+1}$ and set $\omega_b = \|\mathbf{v}_{b,xy}^* - \mathbf{c}_{xy}\|^{\rho_b}$, such that it is the Euclidean distance in the x-y plane of point $\mathbf{v}_b^*$ from the center of the robot's workspace raised to the power of $\rho_b = -1$, for all points except $\mathbf{v}_B^*$, where $\rho_B = -0.5$.

The B-spline path must then be checked to ensure it remains contained within the CTR workspace. We calculate the workspace boundaries of the CTR following the method in [26], and discard any B-spline paths that do not lie inside these boundaries. We note that this approach is independent of tube design parameters, such as tube curvatures. Furthermore, there is no guarantee that the shape of the CTR can feasibly follow the B-spline paths in a perfect FTL manner. Finally, to create waypoints from B-spline path samples, we discretize the path such that the Euclidean distance between two successive waypoints is 0.9 ± 0.1 mm. Examples of the generated sample paths are presented in Fig. 3.

B. Evaluation Parameters and Metrics

We compare our shape controller to the Newton-Raphson tip control algorithm using the damped-least-squares (DLS) method, represented in the general form as in [13]:

$$\Delta\mathbf{q} = -(\mathbf{J}^T\mathbf{W}_0\mathbf{J} + \mathbf{W}_1)^{-1}\mathbf{J}^T\mathbf{W}_0(\mathbf{p}(\mathbf{q}) - \mathbf{p}^*). \qquad (18)$$

Here, $\Delta\mathbf{q}$ is the change in joint values, $\mathbf{p}(\mathbf{q}) - \mathbf{p}^*$ is the tip position error vector, $\mathbf{J} = \frac{\partial \mathbf{p}}{\partial \mathbf{q}}$ is the CTR's tip Jacobian, and $\mathbf{W}_0$ and $\mathbf{W}_1$ are positive definite matrices whose values are tuned to balance error tracking and damping. We use $\mathbf{W}_0 = \text{diag}\{1, 1, 1\}$ for all tube sets, $\mathbf{W}_1 = \lambda I$, $\lambda = 0.1$ for tube set 1, and $\mathbf{W}_1 = \text{diag}\{0.25, 0.25, 0.25, 0.98, 0.98, 0.98\}$ for tube set 2. We define the termination condition as either reaching a maximum of 50 iterations or achieving a tip error less than or equal to 0.1 mm.

To evaluate these control approaches, we use the following error metrics defined per sampled path: maximum tip error, $\bar{e}_{tip}$, and maximum shape error, $\bar{e}_{\text{shape}}$, representing the largest tip error and shape error encountered during the entire tip path traveled [24], respectively. A low maximum shape error indicates good approximate FTL motion, since this measure would be zero for perfect FTL. To maintain consistency across paths, we initialize the joint values as $\mathbf{q}_{init} = [15\,\text{mm}; 10\,\text{mm}; 5\,\text{mm}; 0\,\text{rad}; 0\,\text{rad}; 0\,\text{rad}]$, and set $\Delta\mathbf{q}_{init}$ to zero. We implement all algorithms in MATLAB on an Intel Core i7–800 K Processor. For the QP implementation, we set the quadratic objective term coefficients $c_{1,2,3}$ to 1 and c_4 to 0.05 (unless otherwise specified) and use the quadprog solver with the active set algorithm.

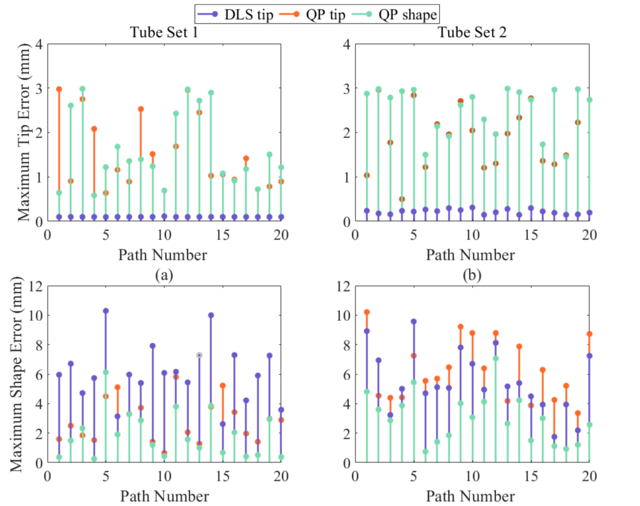

Fig. 4. The maximum tip error and shape error obtained from using the QP shape controller, as well as QP and DLS tip controllers for 20 sampled paths for two different CTRs. The top row ((a), (b)) shows maximum tip error and the bottom row ((c), (d)) shows maximum shape error.

C. Effects of Incorporating the Shape Objective

The goal of this experiment is to determine how the QP shape controller compares to a standard tip controller and a QP-based controller without the shape error objective. We randomly select 20 sample tip paths generated via the procedure from Section IV-A for each tube set. To simulate a teleoperation scenario, we provide one waypoint from the sample path at a time to the three control algorithms: shape control via QP (referred to as the QP shape controller), tip control with QP (referred to as the QP tip controller, where $\alpha_1 = 0$ and the shape error term is ignored), and tip control via DLS (referred to as the DLS tip controller).

In general, we see from the top row of Fig. 4 that the DLS tip controller outperforms our QP-based controllers in terms of maximum tip error most of the time. This outcome aligns with our expectations, as our QP approach addresses a multiobjective optimization problem, which may not always converge to the lowest possible tip error due to the need to balance additional, potentially conflicting costs. Despite this tradeoff, even the resulting maximum tip errors, which represent the worst-case scenario, remain low. Specifically, they remain below the threshold of $\tilde{e}_{tip} = 3$ mm, indicating the effectiveness of falling back to the DLS tip controller.

Importantly, we see from the bottom row of Fig. 4 that the QP shape controller consistently reduces the maximum shape error for the large majority of paths across different tube sets compared to both the DLS and QP tip controllers. These results indicate that incorporating the proposed shape objective improves the approximation of FTL motion in scenarios where perfect FTL

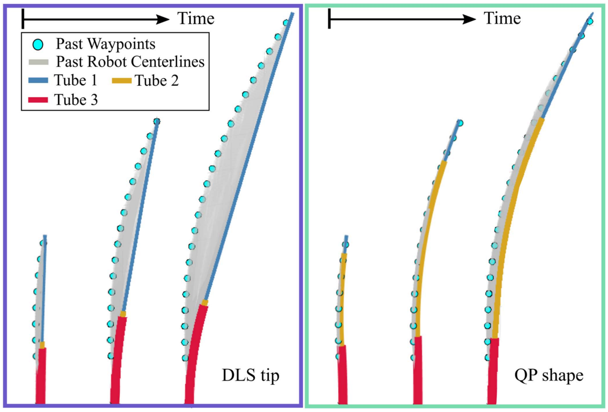

Fig. 5. Time evolution of the shape of tube set 2 for a B-spline path, resulting from using the DLS tip controller (left) and QP shape controller (right). Previous robot shapes are shown in gray and the shape at the current timestep is in color.

TABLE II
STATISTICS SUMMARY OF THE MAXIMUM TIP AND SHAPE ERRORS ACROSS 100 B-SPLINE PATHS

Controller	Tube set	Mean Tip Error [mm]	Mean Shape Error [mm]	Avg. Shape Reduction %
QP Shape	1	1.86 ± 0.89	1.93 ± 1.28	68.6
	2	2.24 ± 0.67	3.84 ± 2.76	44.0
DLS Tip	1	0.10 ± 0.002	6.25 ± 2.25	0
	2	0.12 ± 0.03	6.40 ± 2.84	0

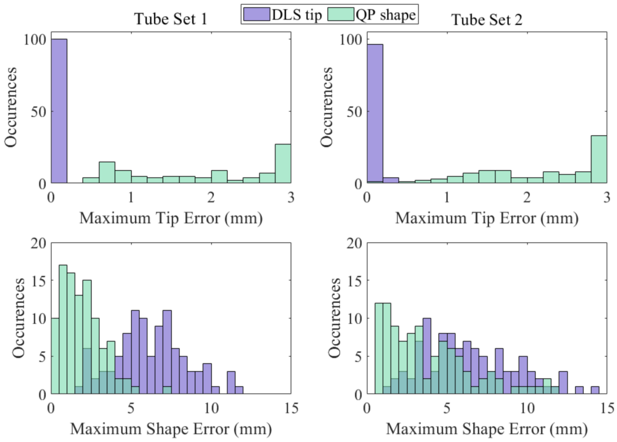

Fig. 6. Distributions of the maximum tip and shape error for tube sets 1 (left) and 2(right) across 100 B-spline paths.

motion may not be achievable, particularly for general tube sets following B-spline paths. Although the shape error is not zero, we see a visible reduction of shape error when the QP shape controller is used. Fig. 5 depicts the evolution of the centerline of the robot across multiple waypoints, along with the final configuration of the robot. It is evident that the robot's shape, under the influence of tip control, can occupy a substantial volume in task space when navigating between waypoints. In contrast, the robot's shape resulting from shape control is observed to occupy a significantly smaller volume.

D. Effect of Limited Shape Error Information

This experiment assesses how reducing m, the number of points on the robot backbone where the shape error is calculated, impacts the performance of the shape control method. Using the same 20 B-spline paths as the previous experiment, we run the QP shape controller on these paths for $m = 3, 6, 9, 12, 15, M$. The distribution of these points along the robot's backbone is as follows: at a minimum, 1 point is placed at the end of each tube, followed by locations at the midpoint of each tube, and then at regular intervals along the tube's length. We observed that neither the maximum tip nor shape error showed a discernible trend as m decreased, indicating that although a larger m may give more accurate shape error estimation, it does not necessarily enhance the shape control performance. We found that the maximum tip error for tube set 1 at $m = 3$ and $m = M$ is 1.79 ± 0.86 mm and 1.60 ± 0.84 mm, respectively, and for tube set 2 they are 2.27 ± 0.70 mm and 2.46 ± 0.48 mm. The maximum shape error for tube set 1 at $m = 3$ and $m = M$ is 1.75 ± 1.15 mm and 1.15 ± 1.55 mm, respectively, and for tube set 2 they are 3.01 ± 1.70 mm and 3.23 ± 1.48 mm. Thus, for the following experiment, we set $m = 3$.

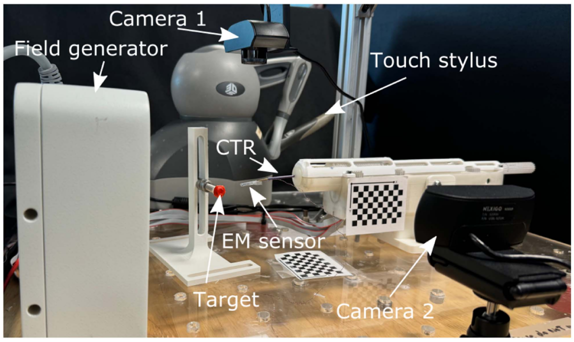

Fig. 7. The experimental setup for the hardware demonstration. Users teleoperate the CTR's tip with an input stylus toward a target.

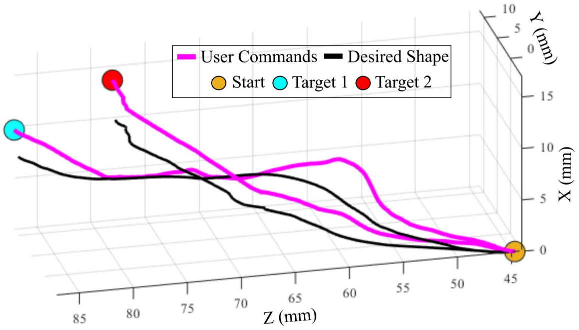

Fig. 8. 3D plot of the recorded user-commanded tip positions, the smoothed desired shape, the starting tip position, and the target locations.

E. Validation of Shape Control Method Across Paths

Finally, we compare the performance of the QP shape controller with limited shape error information to the DLS tip controller for a larger set of general B-spline paths. We randomly select 100 paths generated using the procedure from Section IV-A and apply both controllers. In terms of maximum tip errors, Fig. 6(a) and (b) indicate that using the DLS tip controller results in a right-skewed error distribution for both tube sets that is shifted left compared to the left-skewed error distribution of using the QP shape controller. This trend reiterates the idea that the shape control method sacrifices tip error performance to accommodate conflicting objectives. For maximum shape errors, Fig. 6(c) and (d) show that using the QP shape controller

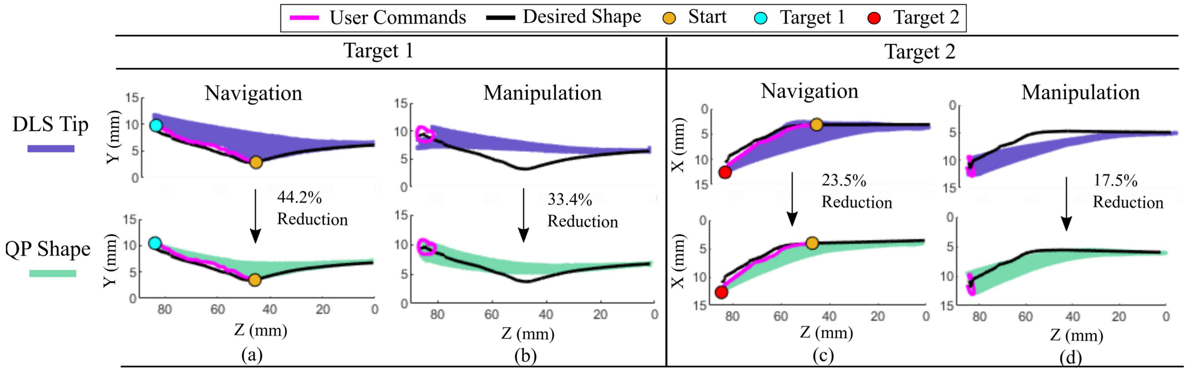

Fig. 9. Time evolution of the robot shape for target 1 using DLS tip (top, in purple) and QP shape (bottom, in mint), during navigation (a) and manipulation (b) captured in the yz plane. Time evolution of the robot shape for target 2 using DLS tip (top, in purple) and QP shape (bottom, in mint), during navigation (c) and manipulation (d) captured in the xz plane.

leads to a right-skewed error distribution that is shifted left compared to a more symmetric distribution achieved by the DLS tip controller. The form of the shape error distribution highlights the potential of the shape control method to reduce shape errors across different tube sets and tip paths, enabling an improvement of approximate FTL motion.

A summary of error statistics is provided in Table II. To quantify the shape error reduction between DLS tip control and shape control per sampled path, we define the shape error reduction ratio (in %): $r = \frac{\bar{e}_{\text{shape,DLS}} - \bar{e}_{\text{shape,QP}}}{\bar{e}_{\text{shape,DLS}}} \times 100$, and average it across 100 paths. The average reduction ratios are 68.6% and 44.0% for tube set 1 and 2, respectively. Furthermore, we hypothesize that a lower shape reduction ratio is associated with an increased frequency of fallbacks, or instances where the shape control algorithm reverts to using the DLS tip controller solution, since a 100% fallback rate would yield a shape reduction ratio of 0. Indeed, we find that on average, tube set 1 and 2 experience fallbacks of 1.67% versus 6.29% of the time per path, respectively.

Finally, we measured the average computation time per waypoint over the 100 paths. For the DLS tip controller, the computation times were 1.61 and 2.55 seconds for tube sets 1 and 2, respectively, and for the QP shape controller they were 1.64 and 1.62 seconds for tube sets 1 and 2, respectively. These results demonstrate that our QP shape controller can solve a multi-objective optimization problem with explicitly enforced constraints while its computational time remains comparable to that of a standard tip-only control strategy. We note that substantial improvements in computation time can be achieved by optimizing the kinematic model and Jacobian implementations for speed. For example, speeds up to $200 - 400$ Hz have been demonstrated for implementations in C++ [7]. Alternatively, computation speed-ups can also be obtained by employing analytical approximations of the torsionally compliant model to compute the kinematics and Jacobian, such as approaches where the torsional interactions between tubes are neglected [3], [28], or where the kinematics is approximated by multidimensional Fourier series [15].

V. Hardware Demonstration

In this section, we demonstrate the effectiveness of the QP shape controller on a physical system during a teleoperation scenario that involves both navigation and manipulation tasks. In the context of minimally invasive surgery, navigation refers to moving the CTR from an initial configuration to reach a surgical target, and manipulation involves subsequent tasks performed at the robot's tip (eg. ablation, suction) [29]. We represent these two scenarios by tasking a user with teleoperating a CTR to reach two different targets and to draw a circle once the target is reached.

A. Experimental Setup

The experimental setup is shown in Fig. 7 and uses tube set 1 from the simulation study and the CTR actuation unit developed in [26]. A Touch Device (3D Systems) is used for teleoperation, where the stylus position is sent to MATLAB through the Robot Operating System (ROS) and mapped to a desired CTR tip position, with a one-to-one scaling in the z direction and one-to-five scaling in the x-y direction. An electromagnetic (EM) tracking system (NDI) and associated 6 DOF EM sensing tool are used for calibration. Two external cameras (NexiGo N1080p) placed in the yz and xz planes are used to capture the shape of the CTR as it deploys. To obtain sufficiently high loop rates, we choose to compute the CTR kinematics and Jacobian via Fourier series approximation [15]. Simulation showed that kinematic information at the end of each tube is sufficient for the QP shape controller, thus, we sampled 75,000 positions for each tube set from the torsionally compliant model to construct the Fourier coefficients.

B. Controller Comparison During Teleoperation

The user is assigned navigation and manipulation tasks employing first the QP shape controller and then the standard DLS controller. For all trials, the CTR starts from initial joint values $[45; 35; 25; 0; 0; 0]$, dictated by hardware constraints. They are tasked with teleoperating the CTR to two different targets, and to then draw a circle at the end of their navigation trajectory. The actual tip positions and the user-commanded tip positions are recorded, and the motion is captured via cameras. To establish the desired shape from the commanded tip positions so far, the user input is smoothed using a kernel estimator with bandwidth $\lambda = 5$ [30]. For each step k, ξ_k is constructed by appending $p_{k,s}$ to $\xi_{k,s}$, where $p_{k,s}$ is a smoothed version of the desired waypoint p_k^*. The x, y coordinates of $p_{k,s}$ are predicted from $z_{k,s} = z_k^*$ by using a Gaussian Kernel. Fig. 8 shows the user-commanded tip positions, desired shape, and target locations. For fair comparsion, the recorded user commands from using the shape controller are then sent one by one to the DLS controller, therefore replicating the same teleoperation input.

To experimentally assess the controller performance, we compute the shape error reduction ratio as described in Section IV. Shape reduction is quantified in 2D by measuring the maximum shape error, which is determined as the pixel difference between the CTR's segmented centerline from images and the desired shape projected onto the most relevant plane for each trajectory. This approach is chosen to mitigate errors associated with 3D shape reconstruction. Visual inspection of Fig. 9 reveals that the QP shape controller aligns the CTR backbone more closely with the desired shape compared to the DLS tip controller, consistent with simulation results. The shape controller reduces shape error from the tip controller by 44.2% during navigation and 33.4% during manipulation for target 1. For target 2, the reduction is 23.5% during navigation and 17.5% during manipulation. The average computation time per waypoint is 0.0016 and 0.0070 seconds, for the DLS tip controller and QP shape controller, respectively.

VI. Conclusion

This letter introduced a novel QP-based shape control method for CTRs that approximates FTL motion. Our controller is computationally comparable to standard tip control methods and requires minimal shape information, making it feasible with limited sensors. Simulation results demonstrate that our shape control method indeed reduces shape error, enabling better approximations of FTL motion across various CTR designs and tip paths. These results were also confirmed in a hardware demonstration with a physical CTR system. Future work includes considering additional kinematic tasks (e.g. instability and singularity avoidance) or incorporating different formulations of the desired shape. The method could also be extended to optimize both joint values and the CTR's base pose. Finally, effects on usability and cognitive load during teleoperation can be studied.

Acknowledgment

The authors would like to thank Dr. Cedric Girerd for his discussions during method development.

References

[1] M. J. Mack, "Minimally invasive and robotic surgery," *JAMA*, vol. 285, no. 5, pp. 568–572, 2001.

[2] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Trans. Robot.*, vol. 31, no. 6, pp. 1261–1280, Dec. 2015.

[3] P. Sears and P. Dupont, "A steerable needle technology using curved concentric tubes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 2850–2856.

[4] R. III, J. S. Kim, N. Cowan, G. Chirikjian, and A. Okamura, "Nonholonomic modeling of needle steering," *Int. J. Robotic Res.*, vol. 25, pp. 509–525, 2006.

[5] P. Swaney, H. Gilbert, R. Webster, P. Russell, and K. Weaver, "Endonasal skull base tumor removal using concentric tube continuum robots: A phantom study," *J. Neurol. Surg. Part B, Skull Base*, vol. 76, pp. 145–149, 2014.

[6] L. Wu, S. Song, K. Wu, and C. Lim, "Development of a compact continuum tubular robotic system for nasopharyngeal biopsy," *Med. Biol. Eng. Comput.*, vol. 55, pp. 403–417, 2016.

[7] J. Burgner et al., "A telerobotic system for transnasal surgery," *IEEE/ASME Trans. Mechantronics*, vol. 19, no. 3, pp. 996–1006, Jun. 2014.

[8] C. Girerd, K. Rabenorosoa, P. Rougeot, and P. Renaud, "Towards optical biopsy of olfactory cells using concentric tube robots with follow-the-leader deployment," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots. Syst.*, 2017, pp. 5661–5887.

[9] Z. Mitros, S. H. Sadati, R. Henry, L. Da Cruz, and C. Bergeles, "From theoretical work to clinical translation: Progress in concentric tube robots," *Annu. Rev. Control Robot. Auton. Syst.*, vol. 5, no. 1, pp. 335–359, 2022.

[10] M. Khadem, J. O'Neill, Z. Mitros, L. da Cruz, and C. Bergeles, "Autonomous steering of concentric tube robots via nonlinear model predictive control," *IEEE Trans. Robot.*, vol. 36, no. 5, pp. 1595–1602, Oct. 2020.

[11] B. Thamo, F. Alambeigi, K. Dhaliwal, and M. Khadem, "A hybrid dual jacobian approach for autonomous control of concentric tube robots in unknown constrained environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 2809–2815.

[12] K. Leibrandt, C. Bergeles, and G.-Z. Yang, "On-line collision-free inverse kinematics with frictional active constraints for effective control of unstable concentric tube robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 3797–3804.

[13] A. S. Deo and I. D. Walker, "Overview of damped least-squares methods for inverse kinematics of robot manipulators," *J. Intell. Robot. Syst.*, vol. 14, no. 1, pp. 43–68, Sep. 1995.

[14] D. C. Rucker and R. J. Webster, "Computing jacobians and compliance matrices for externally loaded continuum robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 945–950.

[15] P. E. Dupont, J. Lock, B. Itkowitz, and E. Butler, "Design and control of concentric-tube robots," *IEEE Trans. Robot.*, vol. 26, no. 2, pp. 209–225, Apr. 2010.

[16] C. W. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods," *IEEE Trans. Syst., Man, Cybern.*, vol. TSMC-16, no. 1, pp. 93–101, Jan. 1986.

[17] R. J. Hendrick, "System design and elastic stability modeling of transendoscopic continuum robots," Ph.D. dissertation, Mech., Eng. Dept., Vanderbilt Univ., 2017.

[18] M. Khadem, J. O'Neill, Z. Mitros, L. d. Cruz, and C. Bergeles, "Autonomous steering of concentric tube robots for enhanced force/velocity manipulability," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot Syst.*, 2019, pp. 2197–2204.

[19] M. T. Chikhaoui, J. Granna, J. Starke, and J. Burgner-Kahrs, "Toward motion coordination control and design optimization for dual-arm concentric tube continuum robots," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 1793–1800, Jul. 2018.

[20] H. Choset and W. Henning, "A follow-the-leader approach to serpentine robot motion planning," *J. Aerosp. Eng.*, vol. 12, pp. 65–73, 1999.

[21] H. B. Gilbert, J. Neimat, and R. J. Webster, "Concentric tube robots as steerable needles: Achieving follow-the-leader deployment," *IEEE Trans. Robot.*, vol. 31, no. 2, pp. 246–258, Apr. 2015.

[22] A. Garriga-Casanovas and F. R. y Baena, "Complete follow-the-leader kinematics using concentric tube robots," *Int. J. Robot. Res.*, vol. 37, no. 1, pp. 197–222, 2018.

[23] D. Rucker, R. III, G. Chirikjian, and N. Cowan, "Equilibrium conformations of concentric-tube continuum robots," *Int. J. Robotic Res.*, vol. 29, pp. 1263–1280, 2010.

[24] M. Neumann and J. Burgner-Kahrs, "Considerations for follow-the-leader motion of extensible tendon-driven continuum robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 917–923.

[25] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.

[26] C. Girerd and T. K. Morimoto, "Design and control of a hand-held concentric tube robot for minimally invasive surgery," *IEEE Trans. Robot.*, vol. 37, no. 4, pp. 1022–1038, Aug. 2021.

[27] L. Biagiotti and C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots*. Berlin, Germany: Springer, 2008.

[28] R. J. Webster, J. M. Romano, and N. J. Cowan, "Mechanics of precurved-tube continuum robots," *IEEE Trans. Robot.*, vol. 25, no. 1, pp. 67–78, Feb. 2009.

[29] C. Bergeles, A. H. Gosline, N. V. Vasilyev, P. J. Codd, P. J. del Nido, and P. E. Dupont, "Concentric tube robot design and optimization based on task and anatomical constraints," *IEEE Trans. Robot.*, vol. 31, no. 1, pp. 67–84, Feb. 2015.

[30] R. C. Smith, *Uncertainty Quantification: Theory, Implementation, and Applications*. Philadelphia, PA, USA: SIAM, 2014.