



HAL
open science

On High Performance Control of Concentric Tube Continuum Robots through Parsimonious Calibration

Quentin Boyer, Sandrine Voros, Pierre Roux, François Marionnet, Kanty Rabenorosoa, M. Taha Chikhaoui

► **To cite this version:**

Quentin Boyer, Sandrine Voros, Pierre Roux, François Marionnet, Kanty Rabenorosoa, et al.. On High Performance Control of Concentric Tube Continuum Robots through Parsimonious Calibration. IEEE Robotics and Automation Letters, inPress. hal-04685717

HAL Id: hal-04685717

<https://hal.science/hal-04685717v1>

Submitted on 3 Sep 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Open licence - etalab

On High Performance Control of Concentric Tube Continuum Robots through Parsimonious Calibration

Quentin Boyer, Sandrine Voros, Pierre Roux, François Marionnet,
Kanty Rabenorosoa, *Member, IEEE*, and M. Taha Chikhaoui, *Member, IEEE*

Abstract—Continuum robots deform continuously, compared to conventional robots composed of rigid links and joints, and require dedicated calibration methods. Indeed, calibration is an essential step to obtain high performance control, as it directly influences robot accuracy. In this paper, we investigate how model parameters influence both model accuracy and model-based closed-loop control accuracy of Concentric Tube Continuum Robots (CTCR). A fast, robust, and real-time implementation of the Cosserat rod model is first introduced. Then, a model-based Jacobian control scheme is presented. A parsimonious calibration procedure focused on control accuracy is finally proposed to achieve submillimetric tracking errors along a 3D trajectory at velocity reaching 5 mm/s in complex scenarios including actuation constraints, obstacle avoidance, and external forces. Results are demonstrated both in simulation and on an experimental setup of a 3-tube CTCR.

Index Terms—Medical Robots and Systems, Surgical Robotics: Steerable Catheters/Needles, Calibration and Identification, Performance Evaluation and Benchmarking, Continuum Robots

I. INTRODUCTION

CALIBRATION is an essential part of robotic applications. It involves identifying an accurate relationship between actuator displacement sensor readings and tip pose in the workspace [1]. It has been widely studied for rigid-link robots but far less for continuum robots (CRs). Compared to conventional robots, CRs have an elastic structure that deforms continuously. They can be miniaturized due to the absence of joints and their ability to follow curved trajectory makes them promising for applications in tortuous environments, such as medical applications [2].

Concentric Tube Continuum Robots (CTCRs) are one of the smallest types of CRs and address a large number of minimally invasive surgical applications [3]. They are composed of a set of precurved elastic tubes assembled in a telescopic manner.

Manuscript received: April, 30, 2024; Revised July, 31, 2024; Accepted August, 27, 2024. This paper was recommended for publication by Editor Pietro Valdastri upon evaluation of the Associate Editor and Reviewers' comments. This work is the results of the research projects funded by grants ANR-11-LABX-0004-01, ANR-21-ESRE-0015, ANR-17-CE19-0005, ANR-17-EURE-0002, and ANR-20-CE33-0001.

Q. Boyer, S. Voros, and M. T. Chikhaoui are with Univ. Grenoble Alpes, CNRS, UMR5525, VetAgro Sup, Grenoble INP, INSERM, TIMC, F38000 Grenoble, France firstname.lastname@universite-grenoble-alpes.fr

Q. Boyer, P. Roux, F. Marionnet, and K. Rabenorosoa are with FEMTO-ST Institute, Univ. Franche-Comté, SUPMICROTECH, CNRS, F-25000 Besançon, France firstname.lastname@femto-st.fr

Digital Object Identifier (DOI): see top of this page.

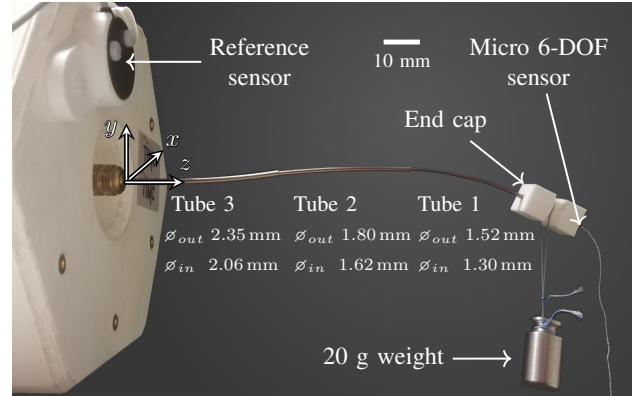


Fig. 1: Experimental setup of our 3-tube CTCR with electromagnetic sensors, and an example of weight used to generate external forces.

Each tube is actuated at its base in translation and rotation and the elastic interaction between the tubes deforms the whole shape of the robot. Because of their simple structure, they can be designed with smaller diameters than other types of CRs and have an interesting ratio of lumen to external diameter, allowing tool integration inside the structure and navigation in restricted environments. Numerous prototypes have been developed, and the CR community is showing increasing interest in control to guarantee accurate positioning and trajectory tracking while considering interaction with the environment [4]. Nevertheless, the continuum backbone and complex phenomena involved in CTCR motions make the calibration challenging. Improper calibration impacts all aspects that rely on the model including control performances, handling of actuation limits, and running into singular configurations. In this paper, we are interested in CTCR calibration methods, and their influence on model accuracy and on control accuracy. Therefore, we review the current state of the art concerning modeling, control, and calibration of CTCR.

A. Related Works

1) *CTCR standard model*: The nominal working regime for most CTCR applications is at the kinetostatic level, which can be accurately captured by a Cosserat rod model [5]. This model accounts for torsion in the tubes, which often have non-negligible effects and can even lead to instabilities known as

snapping [6]. This model can also consider external loads and is now the standard model for CTCRs.

The model computation time is a crucial consideration for online applications. Several approximations of the canonical model [5] are proposed for faster computation. Using truncated Fourier series, the model computes at 1 kHz but does not consider external loads [7]. Linearization with first-order Taylor series expansion is used in [8] to achieve 1 kHz computation rate but it requires additional force and torque sensors. A nonlinear observer is also proposed in [9] to rapidly estimate the solution of the Cosserat rod equations for a single rod. A fast implementation of the canonical model is proposed in [10], achieving up to tens of kilohertz, but it relies on an optimized C++ implementation that is not straightforward to reproduce.

The robot Jacobian J and compliance C matrices, describing changes of the robot shape with respect to actuation variations and external loads variations, respectively, have many applications such as control and force sensing. They can be estimated using finite differences of the model but it is more efficient to compute them along with the model [11]. Computation of the canonical model, Jacobian, and compliance matrices at high rate is still an ongoing challenge.

2) *Control*: Most of the work regarding CTCR closed-loop control is addressed from a quasi-static inverse kinematics perspective using the robot Jacobian matrix for Resolved-Rate Motion Control (RRMC). The Jacobian can be model-based [12], or data-based with online updates of the Jacobian [13]. RRMC is studied in simple scenarios, without constraints, external loads or obstacles in the workspace. Recently, model-based Nonlinear Model Predictive Control (NMPC) has been proposed to handle explicit actuation constraints and to avoid unstable configurations [14]. It is compared to RRMC which fails to achieve those tasks. Using NMPC, linear trajectories are performed without external loads, with 1.2 mm root-mean-squared error (RMSE) and 2.9 mm maximum error for 3-tube CTCR. Data-based linear Model Predictive Control (MPC) is also proposed, claiming that model-based approaches have difficulties due to the inability to predict unknown external forces and disturbances [15]. Using MPC for a 2-tube CTR, various trajectories are performed, some with external loads, with mean errors from 0.748 mm to 2.4 mm and standard deviation from 0.45 mm to 1.7 mm depending on the scenarios. However, in [16], a model-based Jacobian method considering a manipulability criterion is used to steer an unstable 3-tube CTCR, in presence of external loads, while respecting actuation constraints and avoiding snapping. Tracking errors under external loads are not provided, but 0.85 mm RMSE and 5.6 mm maximum error are achieved in unloaded cases. In open-loop control, stability and stiffness criteria are used in a model-based Jacobian scheme to perform instability avoidance and stiffness tuning of an unstable 3-tube CTCR and achieving 4.18 ± 1.98 mm mean error in [17]. This shows that model-based Jacobian methods are promising for CTCR control in complex scenarios but accurate control in complex scenarios is still challenging.

3) *Calibration*: The model of the complete system studied here can be divided in three parts, each one requiring a proper calibration. First, the actuation model describes the

relationship between sensor readings at actuator level and actual tube displacements. Modes et al. [18] focuses on calibration at the actuation level and proposes an automatic and repeatable method to determine rotational offsets. Then, the model of the effective part of the CTCR relates the motion of the backbone with reference to the tube displacements. In this paper, we consider the standard CTCR model including external loads [5]. The considered parameters of this model include geometric and material parameters. Finally, the sensor model relates the sensor readings to the actual signal of interest, often the tip pose or the backbone position.

On the one hand, some parameters, such as reduction ratio, are known with sufficient accuracy, and nominal values are often used. On the other hand, some parameters can have a large range of possible values (e.g. the tubes' Young modulus provided by manufacturers) and calibration procedures are often used to identify them [16], [19], [15]. Furthermore, parameters such as the relative pose between the robot base frame and a sensor reference frame are considered known accurately enough or calibrated [6].

Two approaches are mainly used to identify CR parameters. The first one consists of identifying multiple parameters at once by minimizing the error between the tip position [12] or several positions along the robot backbone [5], predicted by the model and experimental measurements on the complete assembled system. This approach is used to identify parameters before implementing a model-based Jacobian control in [12], [16]. We initially used this approach and obtained maximum model errors of around 3% of robot length. However, this approach involves a number of difficulties. Choosing which parameters to identify and the proper robot configurations to identify them is generally not trivial [20] and it can be particularly challenging for CTCRs because of nonlinearities and coupling effects. It can lead to fitted values that are not physically realistic, as pointed out in [21] and [22]. Moreover, the shape of the robot does not only depend on the current actuator positions but also on the past trajectory [23]. The second approach consists of measuring each parameter independently. It avoids the aforementioned difficulties but can be time consuming as it requires several parameter-specific measurements and may require disassembling parts of the robot. For example, Young's modulus can be identified from static cantilever deflection measurements and the tube precurvature from image processing [19].

B. Contribution

In the literature, fast implementation of the standard mechanics-based model is described as a challenge for online applications such as control. Model-based Jacobian control is described as unsuited to handle actuation constraints or unknown external loads [14], [15] and calibration is sparsely investigated. Our first contribution is to propose a fast and robust implementation of the standard CTCR model, along with the robot Jacobian and compliance matrices. Our implementation is at least 5 times faster than the fastest implementation in the literature [10]. Moreover, our implementation being real-time, it can be used in online applications. We provide our

code¹ for reproducibility and to facilitate future work. We then use our implementation in a closed-loop control scheme to show that model-based Jacobian control can efficiently handle complex scenarios with actuation constraints, obstacles in the workspace, and in presence of external loads. We finally provide extensive studies on the influence of model parameters both on the model predictions and on control performances. We identify the most influential parameters in order to determine on which parameters to put the calibration efforts, aiming to help standardizing and simplifying calibration routines of CTCRs, which can be especially challenging in real-life usage. A dedicated and parsimonious calibration procedure for CTCR is proposed to increase control accuracy and results are presented for a 3-tube CTCR both in simulation and on an experimental setup.

II. EXPERIMENTAL AND SIMULATION SETUPS

The 3-tube CTCR used in the experiments and the reference frame are shown in Figure 1. The actuation unit uses 6 stepper motors (Beckhoff AS1020, 0.5Nm holding torque and 1.8° step angle) with their incremental encoders (1024 steps/revolution). The control scheme is implemented on a computer with Real-Time Operating System (Xenomai), handling the actuators and sensors using Beckhoff EtherCAT terminals. We use Nitinol tubes with a straight and a curved part. Tube parameters are summarized in Table II. The tubes are designed to ensure that the robot is globally stable, meaning that there is no snapping in the entire workspace. An electromagnetic (EM) tracking system (Aurora, NDI) is used to measure the tip position with 0.88mm accuracy and orientation with 0.48° accuracy (95% CI) at 40 Hz.

In order to assess our method performances in simulation in a wide range of conditions, random configurations were simulated with random external point force up to 0.3 N applied at the tip. Actuation variables are randomly sampled similarly to [24]. Simulations are computed using an Intel Xeon W-2275 CPU at 3.30 GHz.

III. EFFICIENT MODEL IMPLEMENTATION

In this section, we propose an efficient implementation to compute the CTCR shape, Jacobian, and compliance matrices.

A. Modeling Background

We use the standard forward kinetostatic model considering external loads for a n -tube CTCR [5]. This model consists of a multipoint Boundary Value Problem (BVP) on a set of nonlinear differential equations:

$$\dot{\mathbf{y}} = \mathbf{f}(s, \mathbf{y}, \mathbf{q}, \mathbf{w}) \quad (1)$$

where \mathbf{y} is the set of state variables, the dot operator ($\dot{\cdot}$) represents derivative with respect to the arc length s , $\mathbf{q} = [\beta, \alpha]$ contains the n translation actuation variables β , and n rotation actuation variables α . \mathbf{w} is an external point wrench applied

at the robot tip. The boundary conditions (BC) can be written in the following form:

$$\mathbf{b}(\mathbf{y}, \mathbf{q}, \mathbf{w}) = \mathbf{0} \quad (2)$$

To solve the BVP, a shooting method is commonly used [12]. First, the related Initial Value Problem (IVP) is solved at iteration k for a certain guess of the unknown initial conditions $\mathbf{y}_u^k(0)$ using numerical integration. Discontinuities where the tubes end or have a step change in precurvature and where external point loads are applied, prevent the use of a standard IVP solver over the entire robot. Instead, the robot is divided in segments delimited by the discontinuity locations and piecewise integration is used. Then, BC residuals $\mathbf{b}(\mathbf{y}^k, \mathbf{q}, \mathbf{w})$ are computed from the IVP solution and a new guess $\mathbf{y}_u^{k+1}(0)$ is estimated using a nonlinear root-finding algorithm to iteratively converge to a guess satisfying the BC. Not only do we consider computation of the robot shape but also of the robot Jacobian \mathbf{J} and compliance \mathbf{C} matrices along the model using IVP finite differences method [11]. \mathbf{J} will be used for model-based control in Section IV and \mathbf{C} is included in our framework to facilitate future work.

B. Numerical Implementation

Many elements influence the model convergence and computation rate. Here we present the choice of state variables, the resolution methods for the IVP and BVP, the computation of the Jacobian matrices, and implementation details.

1) *State variables*: Using curvatures along x and y axis as state variables, as in [11], can cause step change in the initial conditions that the BVP solver has to estimate for certain configurations. Instead, we use the sums of internal moments along x and y axis, as in [12], leading to better convergence. When there are no external loads, the number of state variables and unknown initial conditions is reduced for faster computation, as in [10].

2) *Solving the IVP*: It has been shown in [10] that a single 8th order Runge-Kutta (RK8) step for each segment can accurately solve the IVP. We simulated 10^3 random designs and for each, 10^3 random configurations, as described in Section II. We investigate the use of lower order schemes to achieve faster computation. Using RK3 (SSPRK3 [25]) or classical RK4 is approximately 3.5 and 2.9 times faster than using RK8 (RK8(7)13M [26]), respectively. The errors between the tip positions computed using a single RK3, RK4, and RK8 step per segment and the reference tip positions computed using 100 RK8 steps per segment are shown in Figure 2 with errors relative to robot length. Maximal errors using RK3 are 1% of robot length, which is the same order of magnitude as the modeling error. RK4 is slightly slower but more accurate with 0.1% maximal error and 0.003% mean error. RK8 is much more accurate but slower. We choose to use RK4 as it produces negligible integration error compared to modeling error while being faster than RK8.

3) *Solving the BVP*: We then use the Gauss-Newton algorithm to solve the BVP:

$$\mathbf{y}_u^{k+1}(0) = \mathbf{y}_u^k(0) - \mathbf{B}_u^\dagger \mathbf{b}(\mathbf{y}^k, \mathbf{q}, \mathbf{w}) \quad (3)$$

¹<https://github.com/TIMClab-CAMI/Modeling-and-Control-of-Concentric-Tube-Continuum-Robots>

TABLE I: Model computation rate

Computed elements	Required number of IVP computations	Number of threads							
		1	4	6	8	10	12	14	18*
Unloaded	4	154 kHz	277 kHz	-	-	-	-	-	-
Unloaded + J	10	63 kHz	128 kHz	150 kHz	150 kHz	181 kHz	-	-	-
Loaded	6	41 kHz	91 kHz	134 kHz	-	-	-	-	-
Loaded + J	12	20 kHz	54 kHz	67 kHz	67 kHz	67 kHz	87 kHz	-	-
Loaded + J + C	18	14 kHz	36 kHz	50 kHz	50 kHz	63 kHz	63 kHz	59 kHz	68 kHz

* 14 physical cores, 18 threads

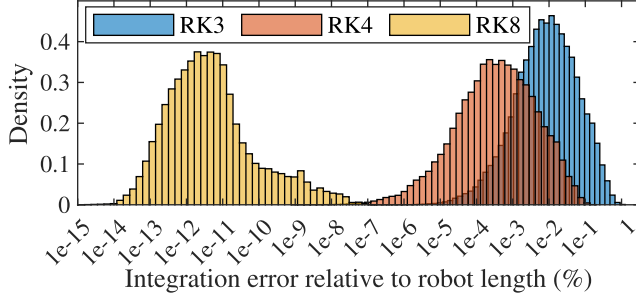


Fig. 2: Comparison of the IVP integration error depending on the integration scheme (log scale).

where the matrix $B_u = \frac{\partial b}{\partial \mathbf{y}_u(0)}$ describing changes of \mathbf{b} with respect to changes in $\mathbf{y}_u(0)$ is computed using IVP finite differences method. The algorithm stops when $\|\mathbf{b}\| < 10^{-10}$. This threshold is chosen according to the desired accuracy.

4) *Parallel computation of the Jacobian matrices:* The IVP finite differences method lends itself well to parallel computation. We investigate the computation rates for different cases, depending on the number of available CPU cores, the presence of external loads, and whether the Jacobian and compliance matrices are computed. The required number of IVP computations and computation rates are in Table I.

5) *Implementation details:* The model is implemented in C++ and compiled with g++ with `-O3` optimization flag. Equations are implemented using Eigen linear algebra library to take advantage of vectorization optimization. Vectors and matrices are declared with fixed size, known at compile time, allowing more efficient optimization and implementation in real-time context, where dynamic allocation is prohibited.

C. Performances

In order to assess our implementation performances, 10^3 random designs, and for each 10^3 configurations were simulated (cf. Section II). The proposed implementation can compute one step of the Gauss-Newton algorithm, i.e. computation of \mathbf{y}^k , \mathbf{b}^k , and B_u^k , along with the Jacobian and compliance matrices, at rates from 14 kHz to 277 kHz depending on the presence of external loads and which elements need to be computed (cf. Table I). Using parallel computation significantly increases the computation rates by about 3 to 5 times. To the best of our knowledge, the fastest implementation in the literature is [10], where computation rates of 36.9 kHz, 9.7 kHz, 27 kHz, and 2.7 kHz are achieved for the following computed elements: unloaded, unloaded + J, loaded, and loaded + J + C, respectively. These results are obtained using

single-thread computation on an Intel i7-4790 at 3.6 GHz. Depending on the computed elements, our implementation is 5 to 25 times faster.

Convergence of the BVP solver typically takes 1 iteration for small actuation variations ($\sim 1^\circ$, 1 mm) with reference to the previous computation, as in a control scenario. For globally stable CTCRs, the Gauss-Newton algorithm converges even for actuation large variations ($\sim 360^\circ$, 10 mm) but may need up to 2 or 3 iterations. However, in the case of CTCR prone to snapping, one could use our method with small actuation variations to ensure convergence.

The accuracy of our implementation is similar to that of a naive one, with very fine discretization and strict tolerances on BVP resolution. Numerical errors are negligible compared with modeling errors. This efficient implementation allows to use the model intensively for a wide range of applications, both offline and online, such as design, trajectory planning, calibration, model-based control, and force sensing. Moreover, we provide a real-time implementation to ensure a desired sampling frequency in online applications.

IV. CONTROL

Using our efficient implementation, we introduce a model-based Jacobian control scheme. We then present the tuning of the gains, the complex scenarios we consider, and we finally assess the performances in simulation.

A. Control Scheme

We use the generalized-damped least squares (GDLS) method for its ability to easily integrate several competing tasks [12]. We consider the following objective function:

$$F = \frac{1}{2} \sum_{i=1}^p (\mathbf{J}\mathbf{q}' - \mathbf{v}_{\mathbf{X},i})^T \mathbf{W}_i (\mathbf{J}\mathbf{q}' - \mathbf{v}_{\mathbf{X},i}) + \frac{1}{2} \sum_{j=p+1}^m (\mathbf{q}' - \mathbf{v}_{\mathbf{q},j})^T \mathbf{W}_j (\mathbf{q}' - \mathbf{v}_{\mathbf{q},j}) \quad (4)$$

accounting for m competing goals, where the p first tasks are described using the desired tip velocities $\mathbf{v}_{\mathbf{X},i}$ and the next $m - p$ tasks are described using the desired actuator velocities $\mathbf{v}_{\mathbf{q},j}$. \mathbf{W}_i is a nonnegative symmetric weighting matrix associated with the i^{th} task. The prime operator ($'$) represents derivative with respect to time. Setting $\frac{\partial F}{\partial \mathbf{q}'} = 0$ gives the necessary condition for \mathbf{q}' to minimize F .

We consider four tasks: trajectory tracking, $\mathbf{v}_{\mathbf{X},tra} = \mathbf{X}'_{des}$ with \mathbf{X}'_{des} the desired tip velocity computed using a standard feedback/feedforward control scheme and trapezoidal velocity profile generation; damping (actuator velocity limitation),

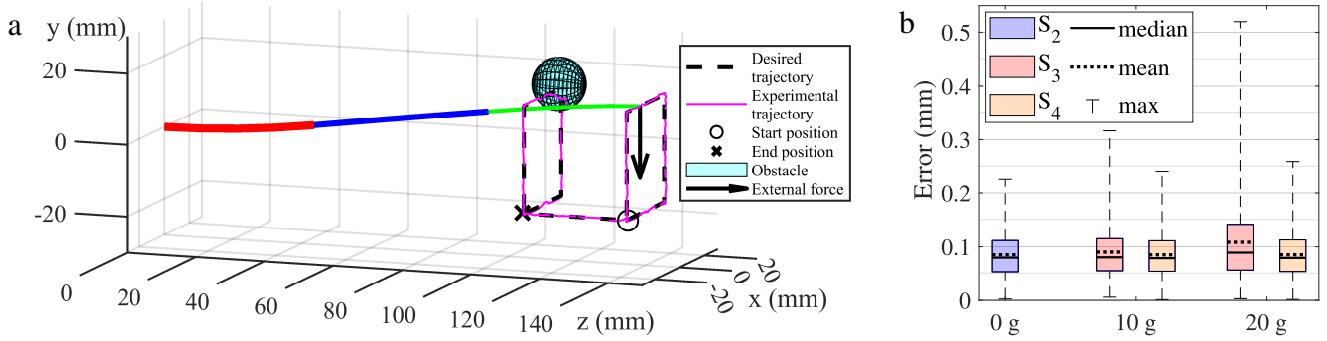


Fig. 3: a) Trajectory tracking of a 30 mm-sided cube in presence of a spherical obstacle and under external point force applied at the tip. b) Tracking error for different forces in S_2 (no force), S_3 , and S_4 scenarios in simulation.

$\mathbf{v}_{q,dam} = \mathbf{0}_{2n \times 1}$; actuator limit avoidance, $\mathbf{v}_{q,act} = -\frac{\partial L}{\partial \mathbf{q}}$ with L the sum of the squared distance between the tube bases, the actuation unit base plate, and the tube tips; obstacle avoidance with $\mathbf{v}_{X,obs}$ pointing away from the obstacle and whose norm is inversely proportional to the squared distance between the tip and the obstacle.

The EM tracking system has the lowest update rate in the setup at 40 Hz, so the entire system is operated at this rate.

B. Weight Tuning

We use diagonal weighting matrices with equal first three diagonal elements $w_{i,t}$ relating to translation and equal last three diagonal elements $w_{i,r}$ relating to rotation, of either the tip or the actuators depending on the considered task. The tuning of the weights is done in two steps. The first step is to roughly estimate the weights, allowing to run the control scenario in simulation. Estimating the relative importance of each task gives a first rough estimation of the weights. Using SI units for all variables and choosing that a 1 mm (10^{-3} m) tracking error is penalized as much as an actuator velocity of 1 mm/s (10^{-3} m/s) and 10° /s ($\frac{10\pi}{180}$ rad/s), we set $w_{tra,t} = \frac{1}{(10^{-3})^2} = 10^6$, $w_{tra,r} = 0$, $w_{dam,t} = 10^6$, $w_{dam,r} = (\frac{180}{10\pi})^2 \approx 33$. The second step is to leverage the simulation to finely tune the weights, starting with the tracking and damping tasks, then the actuation limit task, and finally the obstacle avoidance task. After manual tuning, we obtain $w_{tra,t} = 10^7$, $w_{tra,r} = 0$, $w_{dam,t} = 10^6$, $w_{dam,r} = 10$, $w_{act,t} = 100$, $w_{act,r} = 0$, $w_{obs,t} = 10$, $w_{obs,r} = 0$.

C. Scenarios

We designed several control scenarios to investigate how the considered tasks, the amplitude of the external forces, and their knowledge by the controller influence the performances. In all the scenarios, the reference trajectory is cube-shaped as shown in Figure 3a, with 30 mm sides, representing a significant portion of the robot workspace, and it is performed at a velocity of 5 mm/s and an acceleration of 2.5 mm/s^2 . Without loss of generality, trajectory tracking is performed considering the tip position only. An obstacle modeled as a 10 mm diameter sphere is included onto the reference trajectory. We consider the following scenarios:

- S_1 : Unloaded, and the objective function does not consider the actuator limit avoidance task.
- S_2 : Unloaded, and all tasks are considered.
- S_3 : All tasks are considered and an external, constant point force is applied on the tip, with the force considered unknown for the controller.
- S_4 : S_3 but the force is considered known for the controller.

Results are presented for loads applied along y-axis, as this loading direction produces larger deflections than along z-axis. Forces along x-axis produce similar results. In simulation, Gaussian noise X_n is added to the tip simulated position, where $X_n \sim \mathcal{N}([0, 0, 0], \text{diag}([\sigma^2, \sigma^2, \sigma^2]))$ with $\sigma = 35 \mu\text{m}$ such as X_n magnitude is similar to the magnitude of the measurement noise observed on the experimental setup. Each simulation is repeated 10 times.

D. Simulation Results

In scenario S_1 , the task fails because the actuator limits are violated. In S_2 , the addition of the actuator limit avoidance allows the controller to steer the tip along the desired trajectory while respecting the actuation constraints. For S_3 and S_4 , trajectory tracking results while applying a force along the y-axis, with a magnitude corresponding to weights ranging from 0 g (no force) to 20 g, are shown in Figure 3b. For weights higher than 20 g, the controller fails in both S_3 and S_4 because of actuation constraint violation. When the force is known by the controller, it does not influence the performances, and when it is unknown, the tracking error slightly increases with the magnitude of the force. The controller is quite robust to the external loads, as a 20 g mass increases the maximum and mean error by only 0.3 mm and 0.01 mm, respectively, in closed-loop, whereas it produces a 12 mm deflection in open-loop. Figure 3a shows that the robot accurately follows the trajectory when far from the obstacle, and deviates near the obstacle to avoid collision.

The model-based Jacobian control scheme implemented in this section is relatively straightforward and has successfully performed several simultaneous tasks, such as accurate and fast trajectory tracking, while limiting actuator velocities, respecting actuation limits and avoiding obstacles in the task space. For experimental validation, we now focus on the development of a dedicated calibration procedure.

V. CALIBRATION

In this section, we investigate the influence of calibration on model-based Jacobian control accuracy. We first provide an exhaustive sensitivity analysis of the influence of model parameters on both the model accuracy, considering the tip position, and the impact on control accuracy. We then propose a calibration procedure to estimate the model parameters of our experimental setup and assess the obtained control performances. We consider all the parameters involved in the model of the complete system (actuation, CTCR effective part, and sensor). The actuation is modeled by $q_i = \rho_i \theta_i + \Delta_i$, with ρ_i the transmission reduction ratio, θ_i the motor angular position, and Δ_i the offset. Parameters of the CTCR standard model are for each tube i : Young modulus E_i , Poisson ratio ν_i , precurvatures κ_i , effective length (between the point where the tube is grasped in the actuation unit and the tube tip) l_i , precurved length $l_{\kappa,i}$, inner and outer diameters φ_{in} and φ_{out} , respectively. The sensor is finally modeled by $X_m^{R_{ref}} = T_{ref} X_m^{R_0}$, with $X_m^{R_{ref}}$ and $X_m^{R_0}$ the measured pose in the reference sensor frame and in the robot base frame, respectively, and T_{ref} the homogeneous transformation between the two frames.

A. Proposed method

Considering the challenges induced by the optimization procedure described in Section I, we propose to individually identify the model parameters and to focus the calibration efforts on the most influential parameters. In order to determine which parameters require accurate identification, we first investigate the influence of model parameters by performing a sensitivity analysis in simulation. We consider a nominal model, using the parameters in Table II, and an uncertain model for which one parameter at a time varies in a large range: ± 5 mm for $\Delta_{\beta,i}$, l_i , and $l_{\kappa,i}$, $\pm 5^\circ$ for $\Delta_{\alpha,i}$, ± 15 GPa for E_i , $[0; 0.5]$ for ν_i , $\pm 10\%$ for κ_i , $\pm 15\%$ for tube wall thickness, ± 3 mm for the translational part of T_{ref} , and $\pm 3^\circ$ for the rotational part of T_{ref} . Uncertainties of transmission reduction ratios are considered negligible as we use standard

mechanical components manufactured with tight tolerances. Influence on model accuracy is assessed using 10^6 random configurations sampled as described in Section II and influence on control performance is assessed using the same trajectory tracking scenarios as in Section IV-C. For control, the nominal model is used to simulate the robot, and the uncertain model is used in the controller to estimate the robot Jacobian. A first interesting result is that the maximum variation of the model prediction is a few centimeters but around a hundred times less with only a few hundred of microns in closed-loop, so the closed-loop control is quite robust to parameters uncertainties. We then examine the sensitivity analysis in closed-loop to determine the admissible uncertainties for each parameter, regarding the desired control accuracy.

We start with the parameters that are simple to measure accurately: $\Delta_{\beta,i}$, l_i , and T_{ref} . For $\Delta_{\beta,i}$, the robot is set to its home configuration, and distances between the insertion point and each tube prehension point (where the actuation unit grasps the tube) is manually measured, providing millimetric accuracy. The deployed length of each tube when the robot is in home position is then measured with the incremental encoders, by retracting each tube so the tips are flush with the insertion point. Adding the previously measured translational offsets gives l_i with similar millimetric accuracy. T_{ref} is identified by registering several reference points on the actuation unit baseplate. Positions in the robot base frame are known using the CAD and positions in the reference sensor frame are manually measured using an EM probe. The identified transformation is consistent with the CAD, with difference around 0.5 mm and 0.3° . The cumulative influence of uncertainties of $\Delta_{\beta,i}$, l_i , and T_{ref} increases the maximum tracking error by at most $15 \mu\text{m}$, which we consider negligible.

We then consider $\Delta_{\alpha,i}$. A simple identification method consists of visually aligning the tube with reference to the robot base frame and the associated uncertainties are reported to be around $\pm 5^\circ$ in [18]. Based on our sensitivity analysis (cf. Figure 4a), 5° uncertainties for each tube would increase the maximum tracking error by $210 \mu\text{m}$. We therefore propose an automatic procedure, similar to [18], to identify rotational

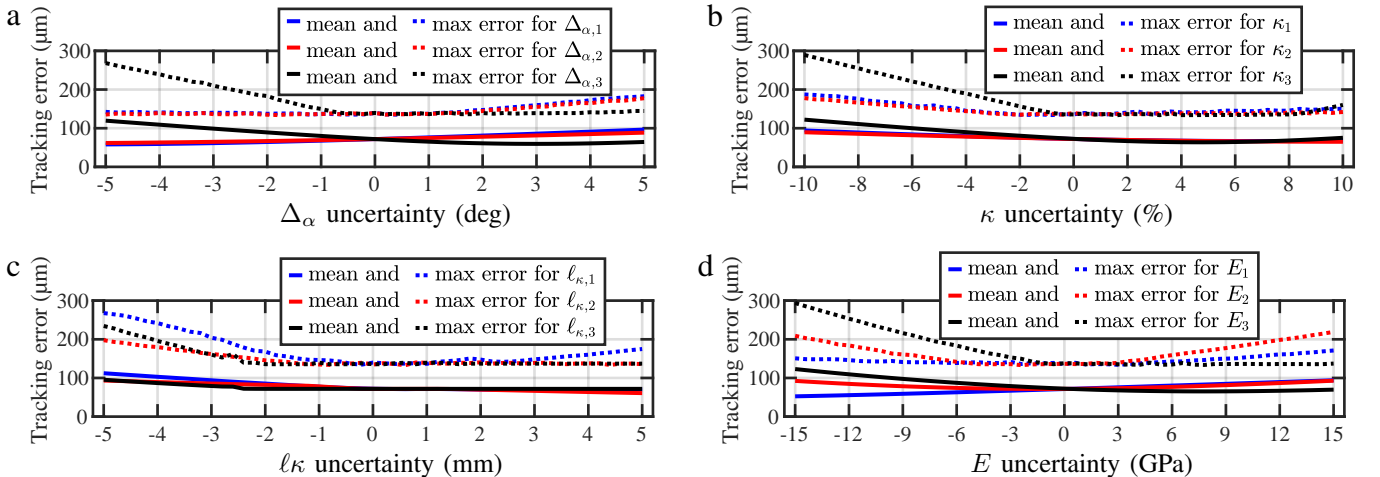


Fig. 4: Influence of uncertainties of a) rotational offsets Δ_{α} , b) precurvatures κ , c) precurved lengths l_{κ} , and d) Young's moduli E on tracking error in closed-loop control.

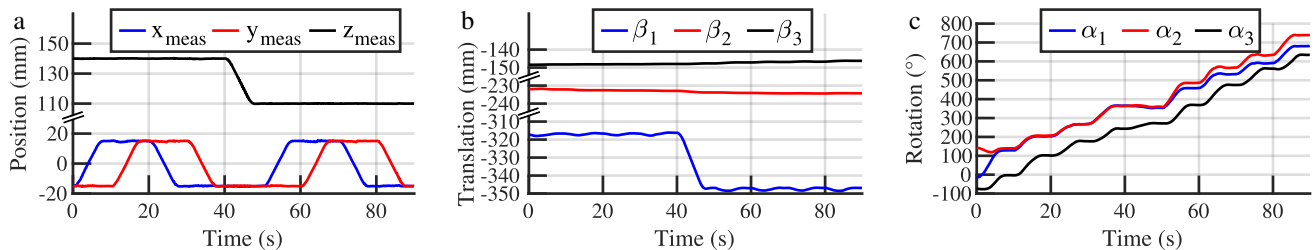


Fig. 5: a) Measured tip trajectory, b) translational actuation, and c) rotational actuation, for experimental scenario S_2 .

offset more accurately and limit the error increase. To identify the offset associated with one tube, only this tube is deployed out of the actuation unit, with an EM sensor attached to its tip. The tube is then rotated so its tip is assumed to follow a circular trajectory in a plane parallel to the $x - y$ plane. We then search for the offset minimizing the error between the measurements and the circular trajectory:

$$\min_{\Delta_{\alpha,i}} \left((x_{center} + radius \cdot \cos(\rho_i \theta_i + \Delta_{\alpha,i}) - x_{meas})^2 + (y_{center} + radius \cdot \sin(\rho_i \theta_i + \Delta_{\alpha,i}) - y_{meas})^2 \right) \quad (5)$$

As we assume the torsional effects to be symmetrical with reference to the rotational direction, we perform the procedure in both directions and keep the mean value as the identified offset. This procedure is performed for each tube independently and allows to reduce the uncertainties to a few tenths of degrees, resulting in negligible influence on maximum control tracking error of a few microns.

We finally examine the parameters that are challenging to measure accurately: κ_i , $\ell_{\kappa,i}$, E_i , and ν_i . Using image processing of tubes' photos before assembly, similarly to [27], we validated that κ_i and $\ell_{\kappa,i}$ are consistent with the nominal values from the CAD and we estimated the image processing accuracy of 10% for κ_i and 5 mm for $\ell_{\kappa,i}$. Such uncertainties increase the maximum tracking error by at most 236 μm for κ_i and 281 μm for $\ell_{\kappa,i}$ (cf. Figure 4b and c). The manufacturer provides a large range for E_i , between 41 and 75 GPa, and a wall thickness tolerance of 15%. We observe that such variations increase the maximum tracking error by at most 265 μm for E_i (cf. Figure 4d) and 85 μm for wall thickness. A variation of ν_i in a range of $[0; 0.5]$ has a negligible influence on the tracking error. Considering the worst-case scenario, where all parameters produce the maximum possible error, the cumulative influence of κ_i , $\ell_{\kappa,i}$, E_i , and wall thickness uncertainties are an 867 μm increase of the maximum error and a 304 μm increase of the mean tracking error, which are below the accuracy of our tracking system. We therefore propose to simply use the nominal values for these parameters. Calibrated parameters are presented in Table II.

B. Experimental Results

We finally implement the control scheme presented in Section IV on the experimental setup using the calibrated parameters. For repeatability assessment, each experiment is repeated 5 times. In order to apply external forces on the tip, 10 g, 20 g, and 30 g weights are attached to the end-cap.

TABLE II: Calibrated parameters

Parameter	Symbol	Calibration method	Value
Reduction ratio (translation)	ρ_β	CAD	2 mm/tr
Reduction ratio (rotation)	ρ_α	CAD	$-1 \cdot \text{tr}_{(\text{tube})}/\text{tr}_{(\text{motor})}$
Offset (translation)	Δ_β	Measurement	$[-298.0; -220.5; -145.0]$ mm
Offset (rotation)	Δ_α	Fit EM measurements and circular model	$[3.55; -0.22; 2.71]$ deg
Young modulus	E	Datasheet	$[58; 58; 58]$ GPa
Poisson ratio	ν	Datasheet	$[0.3; 0.3; 0.3]$
Precurvature	κ	CAD	$[10; 5; 7]$ m^{-1}
Precurved length	ℓ_κ	CAD	$[50; 50; 50]$ mm
Effective length	ℓ	Measurement	$[463.0; 330.5; 199.0]$ mm
Inner diameter	\varnothing_{in}	Datasheet	$[1.30; 1.62; 2.03]$ mm
Outer diameter	\varnothing_{out}	Datasheet	$[1.52; 1.80; 2.35]$ mm
Transformation between the reference sensor frame and the robot base frame	$T_{ref}(xyz)$	Fit EM measurements and CAD	$[-48.25; -7.25; 5.61]$ mm
	$T_{ref}(\phi\theta\psi)$		$[120.26; -0.17; 180.13]$ deg

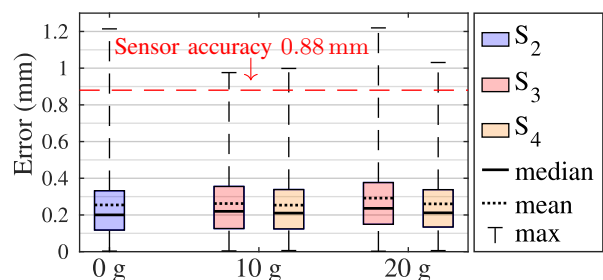


Fig. 6: Tracking error for different forces in S_2 (no force), S_3 , and S_4 scenarios in experiments.

When the weight exceeds 20 g, the controller fails in both S_3 and S_4 due to actuation constraint violation, similarly to the simulation. Figure 5 shows that the actuation variables and the tip position vary smoothly. Tracking errors are presented in Figure 6, where each box represents the distribution of 18000 datapoints (5 repetitions of a 90-second scenario sampled at 40Hz), and a video of the experimental trajectory tracking is also available as supplementary material. Among all scenarios, the maximal error is 1.22 mm, 99.8% of errors are submillimetric and 99.2% are below 0.88 mm (accuracy of the EM measurement). The errors are higher than in the simulation results, probably due in part to parameter estimation errors

but also to modeling errors. However, the vast majority of the measurements are below the sensor accuracy. Combining the GDLS method and our dedicated calibration procedure, the robot is able to accurately follow a 3D trajectory under external loads, in presence of an obstacle while respecting the actuator limit constraints.

VI. DISCUSSION AND CONCLUSION

We proposed a CTCR calibration method aiming to increase closed-loop control accuracy. Performing an exhaustive sensitivity analysis, we identified the influence of the model parameters on control accuracy with reference to the parameter uncertainties. We then focused the calibration efforts on the most influential parameters and introduced dedicated procedures to accurately identify them. Experimental validation of our calibration method on a 3-tube CTCR yielded very satisfying control performances. We compare our results to recent work, where sufficient details on trajectory data are provided [15]. In the context of trajectory tracking in presence of unknown external loads, we obtain mean error, standard deviation, and velocity of 0.29 mm, 0.20 mm and 5 mm/s, respectively vs. 1.7 mm, 2.0 mm and 1 mm/s in [15].

As the influence of the parameters highly depends on the design used and the desired performances depend on the targeted application, our recommendation for a different application, e.g. open-loop control where sensitivities are significantly higher, would be to follow the approach proposed in this paper, by performing a sensitivity analysis, followed by a decision on which parameters should deserve accurate identification based on the targeted accuracy.

Our system can consider unknown external point forces with significant magnitude up to 0.2 N. Handling of distributed forces is straightforward but we show results for point forces applied at the tip as it is the most unfavorable case, producing the largest deflection. For an application requiring larger forces, a mechanical approach, such as tube patterning [28], or a dedicated control scheme [16] could be considered.

Our approach is compatible with image-guided medical applications as it only requires measuring the tip position and not the complete shape or the external loads applied on the robot. Future works will include applying our method in scenarios closer to clinical interventions using medical imaging.

REFERENCES

- [1] Z. Roth, B. Mooring, and B. Ravani, "An overview of robot calibration," *IEEE Journal on Robotics and Automation*, vol. 3, no. 5, pp. 377–385, 1987.
- [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, 2015.
- [3] H. Alfalahi, F. Renda, and C. Stefanini, "Concentric Tube Robots for Minimally Invasive Surgery: Current Applications and Future Opportunities," *IEEE Trans. Med. Robot. Bionics*, vol. 2, no. 3, pp. 410–424, 2020.
- [4] C. J. Nwafor, C. Girerd, G. J. Laurent, T. K. Morimoto, and K. Rabenorosoa, "Design and Fabrication of Concentric Tube Robots: A Survey," *IEEE Trans. Robot.*, vol. 39, no. 4, pp. 2510–2528, 2023.
- [5] D. C. Rucker, B. A. Jones, and R. J. Webster III, "A Geometrically Exact Model for Externally Loaded Concentric-Tube Continuum Robots," *IEEE Trans. Robot.*, vol. 26, no. 5, pp. 769–780, 2010.
- [6] R. J. Webster III, J. M. Romano, and N. J. Cowan, "Mechanics of Precurved-Tube Continuum Robots," *IEEE Trans. Robot.*, vol. 25, no. 1, pp. 67–78, 2009.
- [7] 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, 2010.
- [8] R. Xu and R. V. Patel, "A fast torsionally compliant kinematic model of concentric-tube robots," in *Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2012.
- [9] B. Thamo, K. Dhaliwal, and M. Khadem, "Rapid Solution of Cosserat Rod Equations via a Nonlinear Partial Observer," in *IEEE Int. Conf. Robot. Autom.*, 2021, pp. 9433–9438.
- [10] H. B. Gilbert, "Concentric Tube Robots: Design, Deployment, and Stability," Ph.D. dissertation, 2016, pp. 66–72.
- [11] D. C. Rucker and R. J. Webster III, "Computing Jacobians and compliance matrices for externally loaded continuum robots," in *IEEE Int. Conf. Robot. Autom.*, 2011, pp. 945–950.
- [12] J. Burgner-Kahrs, D. C. Rucker, H. Gilbert, P. Swaney, P. Russell, K. Weaver, and R. J. Webster III, "A Telerobotic System for Transnasal Surgery," *IEEE/ASME Trans. Mechatron.*, vol. 19, no. 3, pp. 996–1006, 2014.
- [13] K. Wu, L. Wu, C. M. Lim, and H. Ren, "Model-free image guidance for intelligent tubular robots with pre-clinical feasibility study: Towards minimally invasive trans-orifice surgery," in *IEEE Int. Conf. Inf. Autom.*, 2015, pp. 749–754.
- [14] M. Khadem, J. O'Neill, Z. Mitros, L. d. 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, 2020.
- [15] B. Thamo, D. Hanley, K. Dhaliwal, and M. Khadem, "Data-Driven Steering of Concentric Tube Robots in Unknown Environments via Dynamic Mode Decomposition," *IEEE Robot. Autom. Lett.*, vol. 8, no. 2, pp. 856–863, 2023.
- [16] M. Khadem, J. O'Neill, Z. Mitros, L. Cruz, and C. Bergeles, "Autonomous Steering of Concentric Tube Robots for Enhanced Force/Velocity Manipulability," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 2197–2204.
- [17] P. L. Anderson, R. J. Hendrick, M. F. Rox, and R. J. Webster, "Exceeding traditional curvature limits of concentric tube robots through redundancy resolution," *Int. J. Robot. Res.*, vol. 43, no. 1, pp. 53–68, 2024.
- [18] V. Modes and J. Burgner-Kahrs, "Calibration of Concentric Tube Continuum Robots: Automatic Alignment of Precurved Elastic Tubes," *IEEE Robot. Autom. Lett.*, vol. 5, no. 1, 2020.
- [19] J. Till, V. Aloï, K. E. Riojas, P. L. Anderson, R. J. Webster III, and D. C. Rucker, "A Dynamic Model for Concentric Tube Robots," *IEEE Trans. Robot.*, vol. 36, no. 6, pp. 1704–1718, 2020.
- [20] J. Hollerbach, W. Khalil, and M. Gautier, "Model Identification," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, pp. 321–344.
- [21] J. Ha, G. Fagogenis, and P. E. Dupont, "Modeling Tube Clearance and Bounding the Effect of Friction in Concentric Tube Robot Kinematics," *IEEE Trans. Robot.*, vol. 35, no. 2, pp. 353–370, 2019.
- [22] J. Lock and P. E. Dupont, "Friction Modeling in Concentric Tube Robots," *IEEE Int. Conf. Robot. Autom.*, pp. 1139–1146, 2011.
- [23] J. Ha, G. Fagogenis, and P. Dupont, "Effect of Path History on Concentric Tube Robot Model Calibration," in *Int. Symp. Med. Robot.*, 2017, pp. 77–78.
- [24] J. Burgner-Kahrs, H. B. Gilbert, J. Granna, P. J. Swaney, and R. J. Webster, "Workspace characterization for concentric tube continuum robots," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 1269–1275.
- [25] S. Gottlieb, C.-W. Shu, and E. Tadmor, "Strong Stability-Preserving High-Order Time Discretization Methods," *SIAM Review*, vol. 43, no. 1, pp. 89–112, 2001.
- [26] P. J. Prince and J. R. Dormand, "High order embedded Runge-Kutta formulae," *J. Comput. Appl. Math.*, vol. 7, no. 1, pp. 67–75, 1981.
- [27] P. Sears and P. E. Dupont, "A Steerable Needle Technology Using Curved Concentric Tubes," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 2850–2856.
- [28] J. Kim, W.-Y. Choi, S. Kang, C. Kim, and K.-J. Cho, "Continuously Variable Stiffness Mechanism Using Nonuniform Patterns on Coaxial Tubes for Continuum Microsurgical Robot," *IEEE Trans. Robot.*, vol. 35, no. 6, pp. 1475–1487, 2019.