A Generalized Framework for Concentric Tube Robot Design Using Gradient-Based Optimization

Jui-Te Lin, Cédric Girerd, Jiayao Yan, John Hwang, Tania K. Morimoto, *Member, IEEE*

***Abstract*—Concentric tube robots (CTRs) show particular promise for minimally invasive surgery due to their inherent compliance and ability to navigate in constrained environments. Due to variations in anatomy among patients and variations in task requirements among procedures, it is necessary to customize the design of these robots on a patient- or population-specific basis. However, the complex kinematics and large design space make the design problem challenging. Here we propose a computational framework that can efficiently optimize a robot design and a motion plan to enable safe navigation through the patient's anatomy. The current framework is the first fully gradient-based method for CTR design optimization and motion planning, enabling an efficient and scalable solution for simultaneously optimizing continuous variables, even across multiple anatomies. The framework is demonstrated using two clinical examples, laryngoscopy and heart biopsy, where the optimization problems are solved for a single patient and across multiple patients, respectively.**

***Index Terms*—Surgical robotics: steerable catheters and needles, Optimization and optimal control, Medical robots and systems, Concentric tube robots**

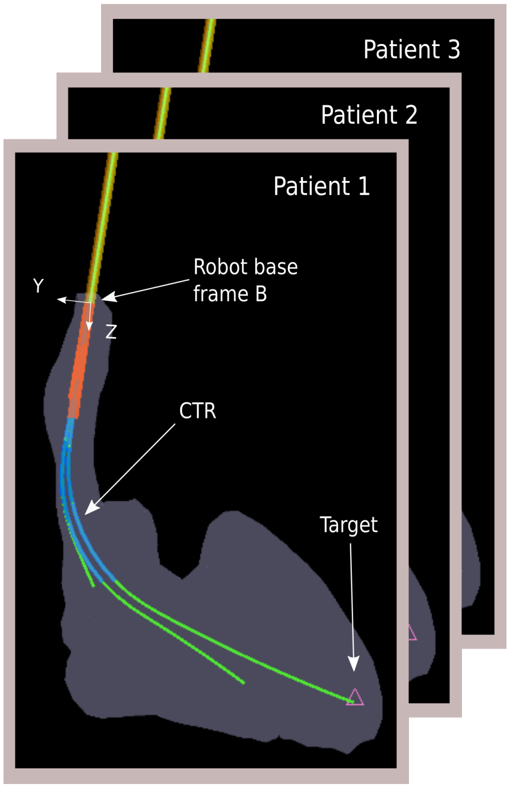


Fig. 1. The proposed gradient-based, computational framework is capable of both patient- and population-specific concentric tube robot (CTR) design optimization. The framework simultaneously optimizes a motion sequence that safely guides the CTR along a collision-free path to reach the surgical target. An example application, myocardial biopsy, is illustrated here.

I. Introduction

Continuum robots offer a number of potential advantages for minimally invasive surgery (MIS).In contrast to conventional rigid-linked robots, continuum robots have an infinite number of degrees of freedom, enabling navigation along highly curved paths [1], [2]. This ability, combined with their small size, can lead to increased accessibility to hard-to-reach places deep inside the body [3]. In addition, their natural compliance can result in increased patient safety and a decreased risk of significant tissue damage during interaction with the anatomy [4].

One type of continuum robot, known as a concentric tube robot (CTR) or active cannula, consists of a set of super-elastic, pre-curved tubes that are assembled concentrically [5], [6]. The compliance of the robot depends on the materials and diameters of the tubes. The relative translation and rotation of the tubes enable them to bend and twist, and the robot's overall shape can be determined based on the bending and torsional interactions between pre-curved tubes. In general, CTRs tend to fall into two categories: steerable needles and teleoperated robotic manipulators [7]. The complex snake-like motion, along with their small size, has enabled their use in many medical applications, including neurosurgery [8], lung biopsy [9], brain tumor surgery [10], and endonasal skull base surgery [11], [12].

It is not guaranteed that a given CTR design can adopt all the desired shapes in 3-D space, due to limitations of its tube geometries and material properties [10]. A single, generic robot will likely not be sufficient for all procedures or patients, and a common approach is to instead design a patient-specific [13] or population-specific [14] robot , as illustrated in Fig. 1. The design problem for CTRs is especially challenging due to the complex kinematics that requires solving 3-D beam bending and twisting problems with boundary conditions [13]. It is therefore not straightforward to design a robot based on intuition, and the large design space makes it nearly impossible to manually select a robot design for a particular scenario.

This work was supported in part by National Science Foundation grant 1850400.

Jui-Te Lin, Cédric Girerd, Jiayao Yan and John Hwang are with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, La Jolla, CA 92093 USA. Corresponding author: jul024@eng.ucsd.edu.

Tania K. Morimoto is with the Department of Mechanical and Aerospace Engineering and the Department of Surgery, University of California, San Diego, La Jolla, CA 92093 USA.

A. CTR Design Optimization Methods

Several approaches have been developed to try to solve this design optimization problem [15]. Many optimization algorithms use a torsionally-rigid model to optimize the tube length and curvature, while avoiding obstacles in the environment [16], [17]. Other approaches have taken the torsional interactions of the tubes into account and optimized the tube designs in order to reach a set of waypoints [13]. In order to reduce the complexity and improve the computational efficiency, the design space is often reduced and only the tube lengths and curvatures are considered as design parameters [13], [18], [19], [20]. The inner and outer tube diameters, which affect the robot's size and bending stiffness, are often not included as design parameters, despite their importance in determining the equilibrium shape of the robot. To date, the only work that considers the tube diameters is a gradient-free optimization framework, which can require long computation times [21].

Planning the motion of the robot is also important in order to safely navigate through the environment. However, CTR motion planning involves solving the inverse kinematics problem multiple times for each deployment step, which is computationally expensive. Several algorithms have been proposed for efficiently obtaining a collision-free motion plan [22], [23], [24]. In addition, approaches that combine both CTR design and motion planning have also been investigated. One such method has been to use an optimization-based approach to plan a collision-free path using a torsionally-rigid model [25], [26]. A more widely used approach is the application of RRT sampling-based motion planning methods to search for a safe motion plan using either a pre-defined robot design [27], or in combination with a design optimization algorithm [28], [29]. This method uses a stochastic optimization algorithm to search in the robot design space, while using a sampling-based motion planner to explore the configuration space. A recent approach has extended this work, to guarantee the optimality of the solutions and to avoid getting stuck in local minima [30]. These previous works represent an important step towards combining CTR design and motion planning into a single framework.

B. Gradient-Based Versus Gradient-Free Optimization

Simultaneously optimizing the tube design and joint variables of a CTR for multiple patient anatomies yields an optimization problem with at least one-hundred optimization variables (Table III). This large set of variables includes the inner diameter, outer diameter, length, curvature, rotation, and translation for each tube, with the latter two repeated for each configuration from the starting point to the target for each anatomy. Each objective and constraint evaluation has a non-trivial computation time (on the order of seconds), as this involves running a full CTR kinematic model for multiple configurations. Thus, it is critical to be able to solve the optimization problem with a minimal number of model evaluations.

Fig. 2 illustrates how the computation times for gradient-based and gradient-free optimizations scale with the number of optimization variables. In particular, we show results of a benchmarking study, computed using the multi-dimensional Rosenbrock problem (shown as dashed lines). The gradient-free optimizers scale quadratically or worse, and with one-hundred optimization variables, the gap in the computation time is several orders of magnitude [31]. Similar studies have been performed by other authors with comparable conclusions [32], [33]. We also compare a gradient-based and gradient-free approach to solving our proposed optimization problem in this work (shown as solid lines). Specifically, we illustrate results from solving the path optimization portion of the framework (described in Section IV-A), as well as from solving the simultaneous optimization (described in Section IV-C) for the laryngoscopy clinical example in Section V-A. The computation time for the path and CTR design optimization problems show the same trend as the benchmarking studies. The gradient-free optimizer, ALPSO (a popular particle swarm optimizer used in previous CTR design optimizations), has a slope of 2.6, which is quadratic, and the gradient-based optimizer, SNOPT, has a slope of 1.3, which is close to linear.

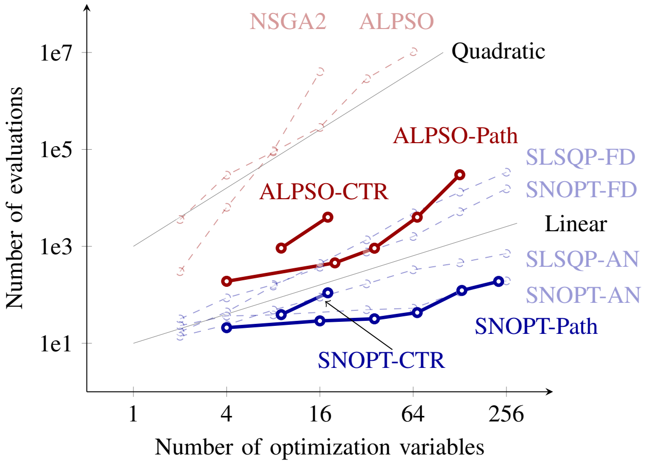


Fig. 2. Optimization algorithm scaling with the number of optimization variables, computed using the multi-dimensional Rosenbrock problem (shown in dashed lines), as well as using the proposed path and CTR optimization problems (shown in solid lines). For all cases, the gradient-free optimizers (ALPSO, NSGA2) scale quadratically or worse, while the gradient-based optimizers (SNOPT, SLSQP) scale linearly with finite-difference derivatives (FD), and better than linearly with analytic derivatives (AN) [31].

Gradient-free optimizers have many advantages, e.g., insensitivity to initial guess, simplicity of application, and higher likelihood of finding global minima. However, gradient-based optimization is more appropriate for the current approach, which is meant to handle problems with over one-hundred optimization variables. It should be noted that in order to enable gradient-based methods, the continuity of the optimization problem must be ensured. Although the majority of objective functions currently considered for CTR design optimization are discontinuous and non-differentiable, the proposed framework may help to solve similar problems if the objective functions and constraints can be reformulated into differentiable functions.

C. Contributions

The contributions of this paper are as follows. (1) We present the first fully gradient-based approach for CTR design optimization and motion planning. One main advantage of this approach is the scalability, which yields high efficiency in solving large-scale optimization problems. It enables the optimization of any set of continuous optimization variables, as well as the optimization across multiple anatomies to create population-specific robots for safe medical interventions. (2) The proposed tool is modular and open-source, and the code, along with associated documentation, is available here: https://ucsdmorimotolab.github.io/CTRoptimization/. The availability of such a tool can enable other researchers to design CTRs for their particular application, as well as to compare designs and optimization methods directly for benchmarking purposes. (3) We present and integrate two new methods for solving the challenges of the CTR design optimization problem. First, the tube plastic deformation is considered for each robot configuration, which ensures that the tubes remain within the material strain limit. Second, compared to current approaches in the literature, we present a more efficient approach to computing the reachability of a CTR, by simultaneously solving a number of inverse kinematics problems. It should be noted that these methods are general and can be added on top of other optimization frameworks, with the added cost of increased computation time for gradient-free approaches.

This paper is arranged as follows. The formulation and definition of the design problem are presented in Section II. In Section III, we describe an overview of the CTR design optimization framework, which is based on OpenMDAO (a software framework for optimization developed by NASA), along with the modules for optimization of a tube set for a single patient, or across multiple patients, simultaneously. Section IV demonstrates the three major steps of our design method. Section V showcases two clinical examples – myocardial biopsy and larynx surgery – with results obtained using the proposed approach. Finally, a conclusion and discussion is presented in Section VI. Table I introduces all variables used in the paper.

II. Design Problem Formulation

CTRs consist of n pre-curved, superelastic tubes that are nested inside one another. Each tube can be rotated and translated via the actuators located at their bases, which change the bending and torsional equilibrium between them. The optimization variables can be categorized into four subsets, as shown in Table I: tube design variables, configuration variables, kinematics variables, and path variables. The robot's joint vector is defined as $\mathbf{q} = [\alpha_1, \cdots, \alpha_n, \beta_1, \cdots, \beta_n]$, where α_i is the base angle of tube i, β_i is the position of tube i with respect to $s = 0$, and s represents the arc length of the robot. The deployed length of tube i is given by $L_i + \beta_i$, where $L_i = L_{s_i} + L_{c_i}$ is the total length of tube i. The CTR design space can be characterized by the geometric parameters and material properties of each tube, as shown in Fig. 3. The geometric parameters include the number of tubes (n), tube curvature (κ_i), length of the straight section (L_{s_i}), length of the curved section (L_{c_i}), inner diameter (ID_i), and outer diameter (OD_i). The material properties include the elastic modulus (E_i) and shear modulus (G_i). The discrete variables, including tube number and material related parameters, are not taken into consideration as optimization variables in the current gradient-based optimization framework. The tube design vector is given by $\mathbf{d} = [\kappa_1, L_{s_1}, L_{c_1}, ID_1, OD_1, \ldots, \kappa_n, L_{s_n}, L_{c_n}, ID_n, OD_n]$, where $\mathbf{d} \in \mathbb{R}^{5n}$. The robot shape can be described as a function, $c(\mathbf{d}, \mathbf{q}, s)$. The robot base coordinate frame, $\mathbf{B} \in \mathbb{R}^6$, includes the base location and orientation and is also considered an optimization variable.

The goal of the proposed optimization framework is to search for an optimal design, $\mathbf{d*}$, such that the robot can navigate through the environment and reach the target without any collisions. Therefore, there is a need to solve a motion

TABLE I
Nomenclature.

Optimization variables	
Tube design variables	
ID_i	Inner diameter of tube i
OD_i	Outer diameter of tube i
L_{s_i}	Length of straight section of tube i
L_{c_i}	Length of curved section of tube i
κ_i	Curvature of curved section of tube
Configuration variables	
ϕ_i	Tip angle of tube i
β_i	Transmission length of tube i
$\mathbf{B}$	Robot base frame
Path variables	
$\mathbf{c_P}$	B-spline control points
Model parameters	
Dependent parameters	
α_i	Base angle of tube i
$\mathbf{K}_i$	Stiffness tensor of tube i
k_{it}	Torsional stiffness of tube i
k_{ib}	Bending stiffness of tube i
ψ_i	Angle between tube i material frame and $\mathbf{R_B}$
$\mathbf{u}_i^*$	Pre-curvature vector of tube i
ϵ_{i_x}	Bending strain of tube i
$\gamma_{i_{xy}}$	Shear strain of tube i
$\mathbf{p_P}$	B-spline path points
$\mathbf{p}_{tip}$	Tip position of the CTR in 3-D space
t_r	Tangent vector of the CTR tip
Independent parameters	
n	Number of tubes in the CTR
s	Linear abscissa along the CTR backbone
$\mathbf{R}_B$	Rotation matrix of the bishop frame
ϵ_{imax}	Material strain limit of tube i
E_i	Young's modulus of tube i
G_i	Shear modulus of tube i
m	Number of links in the discretized CTR
h	Number of patients
b	Number of waypoints
v	Number of 3-D points in the anatomy
c	Number of B-spline control points
a	Number of B-spline path points
S	B-spline curve
ρ	Penalty term
λ	Lagrange multiplier
R_{reach}	Reachable percentage
t_w	Tangent vector of septum wall
$\mathbf{p}_{des}$	Desired position of the CTR tip in 3-D space
$\mathbf{e}_3$	Unit vector in z-direction
N_r	Reachable 3-D points in the workspace

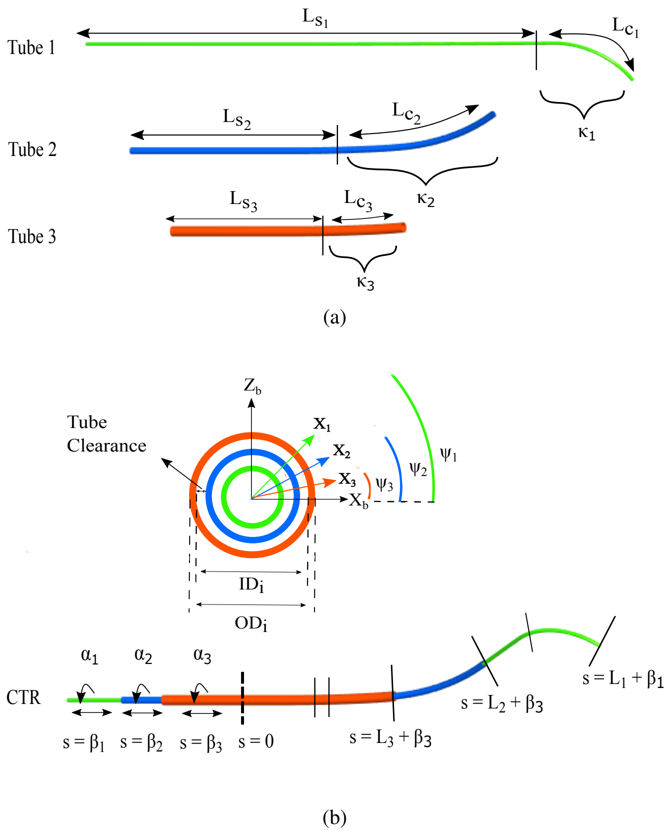


Fig. 3. (a) CTR composed of 3 tubes, each of them consisting of a straight section followed by a pre-curved section. (b) Illustration of the resultant shape of the CTR, along with a cross-sectional view of a segment where all 3 tubes overlap.

planning problem. We chose to use an optimization-based approach that uses a set of waypoints, $\Omega \in \mathbb{R}^3$, to guide the robot from the entry point to the target location, while avoiding collisions. Given a task with b waypoints, we must then solve b inverse kinematic problems in order to derive a motion sequence, $\mathbf{Q} = [\mathbf{q}_1 \ldots \mathbf{q}_b]$. As a result, a collision-free motion plan can be found if the robot, $c(\mathbf{d}, \mathbf{Q}, s)$, does not collide with the anatomy, $\mathbf{\Gamma}$, where $\mathbf{\Gamma} \in \mathbb{R}^{3v}$ is a set of 3-D points representing the anatomy.

In conclusion, the optimization problem is to find a robot design $\mathbf{d}^*$ and a motion plan $\mathbf{Q}^*$ such that the robot, $c(\mathbf{d}^*, \mathbf{Q}^*, s)$, has collision-free deployment. The design space can finally be modified and written as:

$$D = \{\mathbf{d} \in \mathbb{R}^{5n}, \mathbf{Q} \in \mathbb{R}^{2n \times b}, \mathbf{B} \in \mathbb{R}^{6}\}. \tag{1}$$

III. CTR Design Optimization Framework

This section describes the mathematical and software framework that provides the building blocks for the sequence of optimization problems solved in our new CTR design method (Section IV). We first describe OpenMDAO, the general-purpose optimization library with which our framework is implemented. We then discuss how the CTR kinematic model is implemented within OpenMDAO, and, lastly, we describe the formulation of the constraints.

A. OpenMDAO: General-Purpose Optimization Library

OpenMDAO is a NASA-developed open-source Python library for large-scale optimization with complex models [34]. It provides an interface with which complex models (e.g., CTR kinematic model) can be broken down and implemented as a set of modular components that represent smaller units of computation, therefore simplifying efficient derivative computation. With gradient-based optimization, derivatives must be computed accurately since any round-off or truncation can lead to robustness issues in the optimization algorithm, an increase in the number of optimization iterations (and thus computation time), or an inability to converge to the specified termination criterion. Efficiency in the derivative computation method is also critical because the difference between a simple, but slow method (e.g., a finite-difference approximation) and an advanced, but efficient method (e.g., the adjoint method) is a factor of n decrease in the computation time, where n is the number of design variables [35].

However, accurate, efficient derivative computation is especially challenging since the right choice of method depends on the model structure. In the CTR kinematics model, for example, the presence of matrix inversions and explicit time-marching in the model means that the most efficient method is a complex combination of the chain rule and the adjoint method. The challenge of identifying and implementing the right method for computing derivatives is resolved by the unified derivatives equation (UDE), a matrix equation that generalizes the methods for computing derivatives of models [35]. The chain rule, adjoint method, and all other methods and hybrids can be derived from this one equation, simply by making a different choice of which variables to expose and which to consider as hidden within black-box components. The practical benefit is that an optimization library can solve the UDE to effectively automate the computation of model-level total derivatives from the partial derivatives of each component of the model [36]. OpenMDAO is an example of such an optimization library (see Appendix A for more details), and we use it for our CTR design optimization framework.

The implementation of the CTR optimization in OpenM-

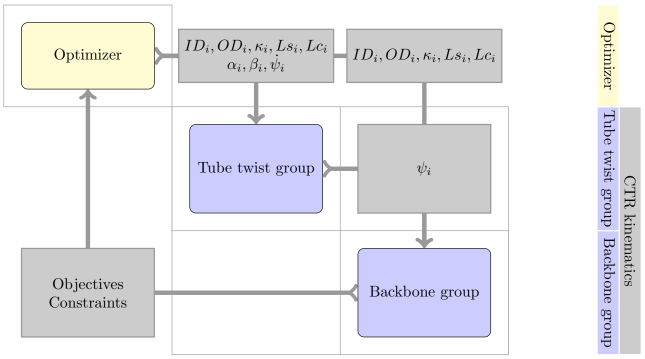


Fig. 4. Design structure matrix, which represents the structure of the CTR kinematics model, used in both the sequential and simultaneous optimization problems, described in Section IV. It consists of the optimization variables (grey), groups (purple), and the optimizer (yellow).

DAO is shown in a design structure matrix in Fig. 4. The grey boxes contain the variables that are passed from one component to another. The CTR optimization consists of an optimizer, optimization variables, the tube twist group, and the backbone group. We connect the CTR model with a commercial sequential quadratic programming (SQP) algorithm called SNOPT [37]. The tube twist group involves the solution to the ODEs for the tube angles, while the backbone group then calculates the backbone position using the tube angles computed from the tube twist group. These groups are described in the sections that follow.

B. Tube Twist Group

CTR tubes generally have a proximal straight section and a distal constant-curvature section, as shown in Fig. 3a, which is the geometry adopted in our work. Once the tubes are assembled, as shown in Fig. 5, the CTR can be divided into a number of links. The overall curvature of each link is determined based on the geometric and mechanical properties of the tubes contained in the link [38]. The forward kinematics problem solves the resultant shape of the robot for a given set of tubes and joint variables. In order to address the torsion and bending interaction between the tubes in each link, an energy minimization approach is used to derive the differential equations for obtaining the tube angle, $\psi_i(s)$, along their length. Assuming no friction, no external loading, and tubes with transverse isotropy, the differential equations with respect to the arc length, s, are given by [39]

$$\ddot{\psi}_i = \frac{k_{ib}}{k_{it}k_b}\sum_{j=1}^{n} k_{jb}\kappa_i\kappa_j \sin(\psi_i - \psi_j), \qquad (2)$$

where i is the tube index, $k_{ib} = \frac{E_i(OD_i^4 - ID_i^4)\pi}{64}$ and $k_{it} = \frac{G_i(OD_i^4 - ID_i^4)\pi}{32}$ are the bending and torsional stiffnesses for tube i (with E_i and G_i being the Youngs and shear modulus of the material of tube i, respectively), k_b is $\sum_{i=1}^{n} k_{ib}$, κ_i is the precurvature of tube i, and ψ_i is the angle for tube i along its main axis. We note that derivatives are taken with respect to the curvilinear abscissa, s, of the robot, if not otherwise specified.

The presence of gauge freedom, i.e. obtaining the same robot shape in 3-D space for a different set of design parameters, should be noted. This phenomenon is visible in the set of Eq. (2), where scaling the tube stiffnesses leaves the system of equations unchanged. Since the bending and torsional stiffnesses are linearly coupled ($k_{it}(1+\nu_i) = k_{ib}$), we perform our analysis of gauge freedom on the bending stiffnesses, k_{ib}. The vector of bending stiffnesses, $\begin{bmatrix} k_{1b} & \cdots & k_{nb} \end{bmatrix}$, is of dimension n, while it is defined by the set of design variables, which are the tube diameters, $\begin{bmatrix} OD_1 & ID_1 & \cdots & OD_n & ID_n \end{bmatrix}$, of dimension $2n$. Thus, in order to provide a set of diameters such that the bending stiffnesses are not linearly scaled, a set of $n+1$ constraints must be added to the set of tube diameters. This can be done by fixing some diameter values, or by constraining the tube clearances between outer and inner diameters of neighboring tubes to a specific value (instead of a range). While it can be important in optimization problems to remove the gauge freedom and ensure a unique solution exists for the robot shape to design parameters, the SNOPT optimizer used in our work can handle such optimization problems that lead to non-unique design parameterizations. Thus, we chose to constrain a minimal set of tube diameters, based on application requirements and allow the gauge freedom to remain.

Based on the number of links, a set of differential equations that considers the boundary conditions, the continuity of the robot, and the bending and moment equilibrium must be solved [38], [39]. The boundary conditions at the proximal ends of the tubes are based on the assumption that the tubes inside the actuation unit are forced to be straight, where $\psi_i(0) = \alpha_i - \beta_i\dot{\psi}_i(0)$. Due to the assumption of no friction between the tubes, there is no axial moment applied from one tube to another. The distal, free ends of the tubes are not subjected to any moments, which leads to the set of boundary conditions $\dot{\psi}_i(L_i + \beta_i) = 0$ at these locations [39]. However, with this set of boundary conditions, the solution to the kinematics may not be unique, due to the fact that multiple robot shapes, with different distal angles, are observed in the case of unstable robots, for the same proximal tube angles, α_i. The alternative approach in order to ensure the uniqueness of the solution for a given set of boundary conditions is to use the distal tube angles, instead of the proximal tube angles, as boundary conditions. The BVP then becomes an initial value problem (IVP), where the two boundary conditions are set at the distal ends of the tubes [40]. In summary, the boundary conditions are now described as in Eq. (3)

$$\begin{aligned} \psi_i(L_i + \beta_i) &= \phi_i \\ \dot{\psi}_i(L_i + \beta_i) &= 0, \end{aligned} \qquad (3)$$

where ϕ_i is the distal angle of tube i. Then, the joint values α_i can be easily derived using the relationship $\psi_i(0) = \alpha_i - \beta_i\dot{\psi}_i(0)$. Unlike the common approach of solving the BVP, which is typically done using a shooting method [39], [41], solving the IVP appears to also be more efficient, as it does not require guessing and iterating the initial condition for the torsion state in order to satisfy the boundary conditions. In order to solve the kinematics, an open-source tool, Ozone, is used to solve the ordinary differential equations (ODEs). Ozone is built on top of the OpenMDAO environment, and therefore the derivatives of the ODE integration are also required. Ozone enables computation of the derivatives across the time-marching scheme for any multi-stage integrator by differentiating the general linear methods equations [42].

The ODE solver, a second order implicit integrator Lobatto2, integrates along the length of the robot, from $s = L_i + \beta_i$ to $s = 0$, using m time steps with the time marching method. The time steps in the integration process are analogous to links in our kinematic model. However, based on the deployed length of each tube, the curvilinear abscissa, s, of the robot is different among the tubes, leading to a discontinuity in the implementation when solving for the tube angle, ψ_i, from the differential equations. We must address this problem in order to vectorize the kinematic model in OpenMDAO.

The solution illustrated in Fig. 5 is to create a $m \times n$ matrix, where m represents the number of links and n is the number of tubes. Since the inner-most tube has the longest deployed

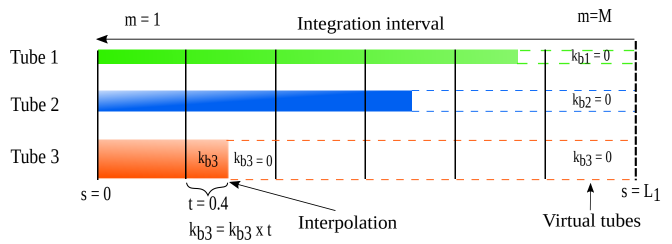


Fig. 5. Illustration of a discretized CTR with m links. The ODE solver integrates from the last link, m, to the first link, for each tube. The virtual tubes are visualized as dotted lines, and an example of the interpolation of bending stiffness, k_{ib}, of each tube is shown here. Since tube 3 ends in between two links (at link 1.4), interpolation of the bending stiffness must be performed. The equation is shown, where t is the percentage of the tube that exists in the link.

length in practice, we can consider the tube length of the inner tube for discretization. The number of links is thus determined by the total length of the inner tube, L_1. In other words, the entire CTR will be discretized into m links, which begins at $s = 0$ and ends at $s = L_1$. However, when the inner tube is not reaching its maximum deployed length, it means that there will be links where no tube exists. To guarantee continuity for solving the differential equations, we use the virtual tube concept in our CTR model [43]. The idea is to extend all distal ends of the tubes so that they are located at $s = L_1$. Since the physical end of tube i is located at $s = L_i$, a virtual part is thus added in the location $[L_i + \beta_i, L_1], i \in [1, n]$. Zero bending stiffness and infinite torsional stiffness are applied to the links where the tubes do not physically exist, in order to ensure that the virtual part does not affect the overall shape of the robot.

To implement the virtual tubes in our framework, there are two challenges to address. First, for our gradient-based approach, we must ensure continuity in the bending stiffness and the curvature of the tubes so that they are differentiable. The hyperbolic tangent function, $1 - \tanh x$, is chosen to generate a smooth transition in the stiffness, k_{ib}, from the physical to virtual section, represented as 1 and 0, respectively. For the curvature matrix, $\tanh x$ is used since the proximal section of each tube is straight, and the distal section is pre-curved with curvature κ_i. The second challenge to address is the fact that the ends of the tubes are, in general, not aligned with the ends of the links, as illustrated in Fig. 5. We propose to use a linear interpolation method for the approximation for both the stiffness and curvature matrices when solving the differential equations. In addition, we vectorize the tube twist group into a 3-D matrix, which includes the number of links (m), number of tubes (n), and the number of waypoints (b). The tube twist group is capable of solving b initial value problems at the same time. It should be noted that the use of interpolation to remove the discontinuities in the tube curvature and stiffness (i.e. where the physical tubes end and the virtual tubes start) introduces some errors in the kinematics models. However, the interpolation error is small compared to, for instance, the impact of friction and tube clearance [44], and can be further reduced by increasing the number of links.

C. Backbone Group

After solving for the tube twist angles, $\psi_i(s)$, the 3-D robot shape is reconstructed in Cartesian space. The deformed curvature vector of the robot from the base to the tip can be obtained using [39],

$$\mathbf{u} = \mathbf{K}^{-1}\sum_{i=1}^{n}\mathbf{K}_i(\mathbf{R}_{\psi_i}\mathbf{u}_i^* - \dot{\psi}_i\mathbf{e}_3), \tag{4}$$

where $\mathbf{K}_i$ is a 3×3 stiffness tensor of tube i, $\mathbf{K} = \sum_{i=1}^{n}\mathbf{K}_i$, $\mathbf{u}_i^*$ is the pre-curvature vector of tube i, and $\mathbf{e}_3$ is the unit vector in z-direction, tangent to the robot backbone. Since the inner tube is the only tube that extends from the proximal end to the distal end, reconstructing the shape of this tube alone is sufficient to obtain the robot shape. Two differential equations for determining the backbone position are as follows:

$$\begin{aligned}\dot{\mathbf{R}} &= \mathbf{R}\hat{\mathbf{u}},\\ \dot{\mathbf{p}} &= \mathbf{R}\mathbf{e}_3,\end{aligned} \tag{5}$$

with the initial conditions $\mathbf{R}(0) = \mathbf{R}_z(\psi_1(0))$ and $\mathbf{p}(0) = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$. The same ODE integrator Lobatto2 and time-marching approach are used to solve the differential equations. The first differential equation allows us to obtain the backbone orientation, $\mathbf{R} \in SO(3)$. The second differential equation gives us the robot backbone position, $\mathbf{p} \in \mathbb{R}^3$. Since the deployed length may not align with the link, interpolation of the robot tip position is also performed.

D. Constraints

This section presents the primary constraints that are explicitly enforced in the optimization problem.

a) Kinematic and geometric constraints: The boundary conditions at the distal ends of the tubes are treated as constraints in the optimization problem. The torsion at the tip of each tube, $\dot{\psi}(L_i + \beta_i)$, is constrained to be zero because the tubes have free distal ends. In addition, we constrain the minimum wall thickness of the tubes based on the particular material and fabrication method. The clearance between each tube must also be constrained with a lower bound to ensure that they can be actuated without too much friction, and an upper bound to limit the impact of tube clearance on the physical robot shape. In addition, the proximal end of each tube must be exposed in order to be grabbed by their respective actuators in physical prototypes, leading to the inequality $\beta_i - \beta_{i+1} \leq 0$. Finally, we enforce constraints on the exposed length of the distal end of each tube: ($L_i + \beta_i$)-($L_{i-1} + \beta_{i-1}$) ≥ 0.

b) Tube plastic deformation constraints: For all optimized robot configurations, it is important for the material strain to remain in the elastic region to avoid plastic deformation [38]. When the tubes deform to reach an equilibrium configuration, the material strain changes. Most of the existing frameworks only consider the pre-curvature limit and enforce a constraint on the maximum curvature or diameter [45]. However, ensuring no plastic deformation during the entire surgical operation is critical.

For planar tubes arranged in a plane, the maximum allowable curvature can be computed using $\delta\kappa_{max,i} = k_i - \kappa_{eq}$ [38].

However, tubes aligned in the same plane with opposite curvatures represent a worst-case scenario for the strain that may not happen during operation, and therefore such a formulation may unnecessarily constrain the robot design. We instead examine the tube strain for the specific joint values obtained using our optimization algorithm. To do so, the tube strains are computed for each link, along the entire tube lengths. In addition to examining the tube strain related to bending, like in previous approaches, we also include the strain related to torsion of the tubes along their lengths. This method assumes that the heat-treated tube materials, such as Nitinol, have zero strain in the pre-curved state, before assembly [46], [47].

To ensure that the strain limit for each robot configuration is under the material strain limit, ϵ_{imax}, the principal strains are computed using Eq. (6) [48]

$$\epsilon_{i_I}, \epsilon_{i_{II}} = \frac{\epsilon_{i_x} + \epsilon_{i_y}}{2} \pm \sqrt{\left(\frac{\epsilon_{i_x} - \epsilon_{i_y}}{2}\right)^2 + \frac{\gamma^2_{i_{xy}}}{2}}, \tag{6}$$

where ϵ_{ix} and ϵ_{iy} are local bending strains of tube i in the x and y directions and γ_{ixy} is the local shear strain of tube i. To compute the bending strain in each link of the tubes, we assume that each link is planar and has a constant curvature, and compute the change in arc-length between the tube's natural plane and equilibrium plane. To compute the arc length along the tubes around their entire circumference, before and after reaching the equilibrium we use the set of Eq. (7) [38]

$$\begin{aligned} \chi_i &= l\kappa_i \left(\tfrac{OD_i}{2}\sin\left(\theta - \tfrac{\pi}{2}\right) + \tfrac{1}{\kappa_i}\right) \\ \chi_{eq} &= l\kappa_{eq}\left(\tfrac{OD_i}{2}\sin\left(\theta - \tfrac{\pi}{2} + \xi\right) + \tfrac{1}{\kappa_{eq}}\right), \end{aligned} \tag{7}$$

where l is the link length, κ_i is the curvature of tube i, κ_{eq} is the equilibrium curvature of the robot, OD_i is the tube outer diameter, ξ is the difference between the tube angle and the robot equilibrium angle, and $\theta \in [0, 2\pi]$. The bending strain ϵ_{i_x} is then given by

$$\epsilon_{i_x} = \frac{|\chi_i - \chi_{eq}|}{\chi_i}, \tag{8}$$

and using the assumption of planar, constant-curvature links, we have $\epsilon_{i_y} = 0$. The shear strain $\gamma_{i_{xy}}$ can be computed using the torsion of the tubes along their lengths. For a circular hollow tube, the maximum shear strain occurs on the outer surface. Therefore, the outer diameter OD_i of the tube is used, and the torsional strain is thus given by Eq. (9).

$$\gamma_{i_{xy}} = \dot{\psi}_i \frac{OD_i}{2} \tag{9}$$

Fig. 6 illustrates both the bending and torsional strain as computed by Eq. (8) and Eq. (9) for an example tube. Finally, after computing the principal strains ϵ_{i_I} and $\epsilon_{i_{II}}$, we apply the failure criterion $\epsilon_{i_I}, \epsilon_{i_{II}} \leq \epsilon_{imax}$ [49] to verify that the strain levels are below that of the material's limits.

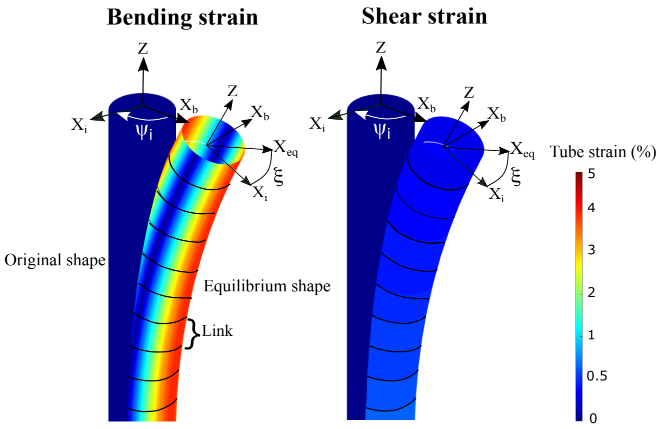


Fig. 6. Illustration of the change in tube strain of a section of a single tube of a CTR from its original state to the equilibrium state due to the changes in arc-length and presence of torsion. χ_i and χ_{eq}, described in Eq. (7) are used to compute the arc-length around the tube circumference for natural plane (X_i) and equilibrium plane (X_{eq}), respectively.

IV. Optimization-Based CTR Design Method

This section presents the details of the optimization framework, which can be decomposed into three steps, as shown in Fig. 7. The first step is to solve the path-finding problem independent of the CTR model. The second step solves a series of sequential optimization problems, each of which involves finding a configuration that reaches a waypoint along the previously found path. The third step is the simultaneous optimization of the CTR tube design and joint variables, that considers all waypoints simultaneously and uses the results from the previous step to provide initial guesses.

A. Path Optimization

Path-finding problems are relevant to many robotics applications. Some of the approaches, including artificial potential fields, road-map approaches, cell decomposition, and polynomial interpolation, are used to search for a smooth, continuous and collision-free path for robot navigation [50], [51]. In this work, we propose a new formulation that relies on a 3-D B-spline curve parametrization to optimize the desired path in a constrained environment. B-splines are often used for curve-fitting and shape optimization. Here, we optimize the control points that define the B-spline curve to generate a 3-D path inside the anatomy, while passing through the starting point and the target point. A B-spline curve, $\mathbf{S}(x)$, is given by a linear combination of the basis functions, $B_{i,k}(x)$, and the 3-D control points, $\mathbf{c}_{pi}$, as given by,

$$\mathbf{S}(x) = \sum_{i=0}^{n-1} \mathbf{c}_{pi} B_{i,k;t}(x), \tag{10}$$

where $B_{i,k;t}$ are B-spline basis functions of degree k and knots t. Basis functions are given by,

$$\begin{aligned} B_{i,0}(x) &= \begin{cases} 1 & \text{if} \quad t_i \leq x \\ 0 & \text{otherwise} \end{cases} \\ B_{i,k}(x) &= \tfrac{x - t_i}{t_{i+k} - t_i} B_{i,k-1}(x) + \tfrac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(x) \end{aligned} \tag{11}$$

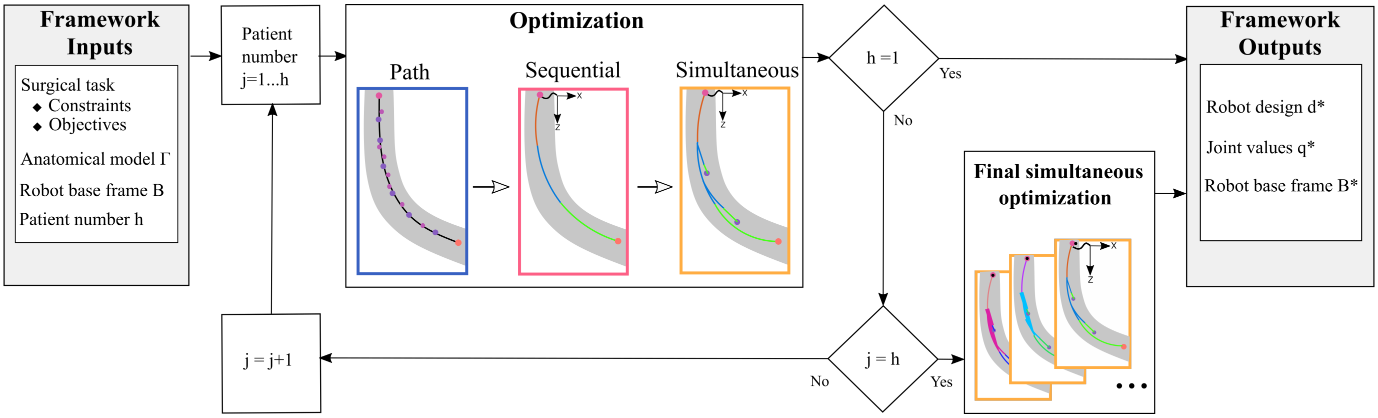


Fig. 7. Diagram illustrating the workflow of the proposed framework, which enables patient- and population-specific design optimization. For a specific medical application, four inputs are required from the user in order to optimize a robot design, a series of joint values, and the robot base frame.

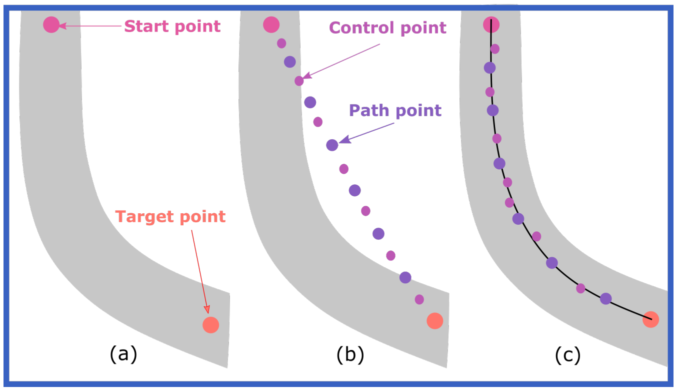


Fig. 8. Illustration of the path optimization including (a) selection of two end-points, (b) formation of a straight line that consists of a number of B-spline path points and control points, and (c) finding a collision-free path.

TABLE II
BEHAVIOR OF THE FUNCTION, $g(x)$,

Location	Close to wall	Far from wall
Inside the anatomy	Negative small	Negative large
Outside the anatomy	Positive small	Positive large

The starting point and final target point must first be selected by the user, as shown in Fig. 8a. The other user-defined parameters are the number of path points (a) and control points (c). All the points on the B-spline curve will be referred to as the path points. The algorithm initializes all the control points and path points on a straight line between the starting point and target point, as shown in Fig. 8b.

Then to create a collision-free motion plan, a method to ensure that all the B-spline's path points remain inside the particular patient anatomy must be implemented. The current framework requires the user to import a 3-D model represented by a triangular mesh, which can be obtained by segmentation and 3-D reconstruction of MRI or CT scans. Unlike the discontinuous KD-tree approach that has been used in some gradient-free frameworks [40], [13], our implementation must be formulated as a continuous and differentiable function, in order to be integrated into our gradient-based method. Our proposed algorithm extracts the point cloud from the imported mesh and computes the normal of each vertex in order to implement collision avoidance. We propose a new formulation, $g(x)$, that can be introduced into this path-finding problem in Eq. (12),

$$f_1(x) = \zeta \underbrace{\sum_{i=1}^{a}\left(\sum_{j=1}^{v} dc_{i,j}(x)^{0.125} P_{i,j}(x)\right)}_{g(x)} + \epsilon_e \frac{\sum_{i=1}^{a-1} dp_i^2(x)}{\left(\sum_{i=1}^{a-1} dp_i(x)\right)^2} \tag{12}$$

where a is the number of path points on the B-spline, v is the number of points in the point cloud representing the anatomy, and x represents the 3-D position for the path points. The $dc_{i,j}(x)$ function calculates the Euclidean distance between an anatomical point and a given point. The sign function, $P(x)$, is used to compute whether this given point is inside the anatomy or outside. In addition, the 0.125 norm allows distant anatomical points to be weighted less, therefore focusing on anatomical points close to the given point. The behavior of the function $g(x)$ is shown in Table II. When a 3-D point is inside the anatomy and remains far away from the wall, the smallest value of the function can be found. In contrast, the function's largest value appears if a 3-D point is located outside and far away from the anatomy. $dp_i(x)$ is the Euclidean distance between path point i and $i + 1$, and ζ and ϵ_e are the tuning parameters.

The penalty method is used to solve our path optimization problem. Once the spline curve is inside the anatomy, the distances between adjacent path points are forced to be as equal as possible to ensure a smooth path, as illustrated in Fig. 8(c).

B. Sequential Optimization

The path points from our path optimization step serve as an input for the sequential optimization. The user can define

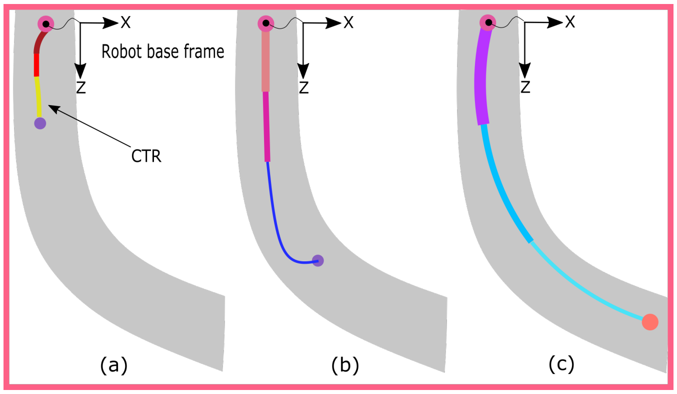


Fig. 9. Illustration of the sequential optimization step that aims to guide the robot to the surgical site safely. The tube design variables are different among the configurations since each waypoint is solved individually. Both (a) and (b) show the CTR hitting the waypoints in order to reach the final target, as visible in (c).

the number of waypoints, b, which will be selected from the B-spline path points in order to guide the robot to the target site, represented by the final path point. The robot must avoid collisions throughout the entire sequence, for each waypoint. Due to the complexity of the optimization problem, the large design space considered, and the use of a gradient-based approach, a good initial guess is needed to ensure convergence. As a result, the algorithm solves a sequence of optimization problems for each waypoint independently. Each optimized robot design and associated joint variables are used as an initial guess to solve the subsequent optimization problem, as shown in Fig. 9.

The objective function in this sequential optimization problem is divided into three sub-functions: f_1 , f_2, and f_3. To compute f_1, Eq. (12) will be used, where the function $dc_{i,j}(x)$ will now be computed as the Euclidean distance between the point cloud representing the anatomy and the backbone points, such that increasing ζ will force the robot to remain inside the anatomy. In addition, the function $dp_i(x)$ is instead used to equalize the deployed length of each tube in order to prevent the robot from favoring the deployment of particular tubes over others. For instance, the optimization algorithm might find a CTR that only deploys the inner tube to reach certain target points, resulting in a very long inner tube, which is not an ideal configuration for practical use.

In addition, in order to reach the desired position in 3-D space, an inverse kinematics problem must be solved. The approach is to optimize the robot design and joint variables so that the Euclidean distance between the robot tip position and the target position is minimized. The second sub-function, f_2, handles the tip position error, and is given by

$$f_2(x) = \frac{\rho}{2}\left(\frac{||\mathbf{p}_{tip}-\mathbf{p}_{des}||_2}{||\mathbf{p}_0||_2}\right)^2 + \lambda\frac{||\mathbf{p}_{tip}-\mathbf{p}_{des}||_2}{||\mathbf{p}_0||_2}, \tag{13}$$

where p_{tip} is the robot tip position, p_{des} is the desired position, and $||\mathbf{p}_0||$ is a normalization term that allows it to be more generally applicable to other optimization problems by reducing the manual effort on parameter tuning. We implemented the Augmented Lagrangian method to handle the tip position error. By definition, ρ is the penalty term and λ is the Lagrange multiplier. This approach will obtain the exact solution if we have a perfect Lagrange multiplier, λ^*. Another benefit is that we can apply a lower value of ρ, which can give us the same desired accuracy on the tip position and offer a better-conditioned Hessian matrix. The function f_3 accounts for deviations in position and orientation with respect to the entry point and is given by

$$f_3(x) = \epsilon_p \frac{||\delta\mathbf{pos}||_2}{||\delta\mathbf{p}||_2} + \epsilon_r \frac{||\delta\mathbf{rot}||_2}{||\delta\mathbf{r}||_2}, \tag{14}$$

where $\delta\mathbf{pos}$ and $\delta\mathbf{rot}$ are the deviations in the base location and orientation with respect to an initial user-defined frame, respectively. ϵ_p and ϵ_r are the tuning parameters. $\delta\mathbf{p}$ and $\delta\mathbf{r}$ are the normalization terms. This function ensures that the robot will remain close to the insertion point defined by the user while allowing some freedom to adjust the base frame. As a result, the final objective function can be written as

$$f(x) = \sum_{i=1}^{b}\left(f_1(x)_i + f_2(x)_i + f_3(x)_i\right), \tag{15}$$

where b is the number of the waypoints. The Augmented Lagrangian method requires solving a sequence of optimization problems by gradually increasing the tuning parameter until the solution is found. Thus, in order to reduce the effort of manually tuning the parameters across different anatomies, we apply a continuation approach to automatically find the two best tuning parameters, ζ and ρ, that allow us to reach the target while avoiding collisions with the anatomy. The tuning parameters are tuned outside the optimization, instead of during each iteration. Therefore, the property of continuity of the objective function is still preserved. The value of ζ will be increased if there is a collision between the robot and the anatomy, and ρ will be increased if the tip position error is larger than the tolerance.

C. *Simultaneous Optimization*

This section presents a new approach that enables optimization of the robot design for single or multiple patient anatomies. The patient-specific simultaneous optimization needs to be performed first for each anatomy in order to solve the population-specific simultaneous optimization.

1) Patient-specific: The simultaneous optimization problem shares similarities with the sequential optimization problem. The penalty method and the Augmented Lagrangian method are used, and all the constraints remain the same. The tube design variables of the final robot configuration in the sequential optimization step are used as the initial guess for this simultaneous optimization, since we already know that it can reach the target location. However, it is not guaranteed that the other configurations during robot deployment will satisfy the constraints, as shown in Fig. 10a. As a result, we propose a new approach that allows us to optimize a robot design and a sequence of joint values simultaneously in the same optimization problem. Note that only one tube set and base frame should be optimized since the same robot design will be used throughout the entire surgical task. The difference between the robot configurations will therefore be the joint values. Our framework thus includes another dimension—the number of robot configurations or waypoints, b. This allows us to simultaneously optimize the robot design ($\mathbf{d}$), base frame ($\mathbf{B}$), and the motion plan ($\mathbf{Q}$). Fig. 10 shows an example before and after the simultaneous optimization process, and the patient specific optimization algorithm is shown in Algorithm 1.

2) Population-specific: After the patient-specific simultaneous optimization step is performed for each patient, an extra step can be introduced in order to design a population-specific robot design that works across multiple patients. To do so, we can take advantage of the modular approach in OpenMDAO and consider each patient anatomy as a *group* that consists of the sub-*groups* for the CTR kinematics, including the tube twist and the backbone *groups*. Once all the *groups* are constructed, we can combine them into a *model* to form the CTR design optimization problem for multiple patient anatomies. The optimization procedure is shown in Fig. 7 and is as follows: 1) We solve the patient-specific design and motion planning problem (using the path, sequential, and simultaneous optimization steps) from patient 1 to patient h. 2) We average the tube design variables among the patients and use these as an initial tube design for the population-specific optimization. 3) We perform a population-specific simultaneous optimization across multiple patient anatomies using the following modified objective function,

$$f(x) = \sum_{i=1}^{h} \sum_{j=1}^{b} \left(f_1(x)_{i,j} + f_2(x)_{i,j} + f_3(x)_{i,j}\right), \qquad (16)$$

where h is the number of patients and b is the number of waypoints.

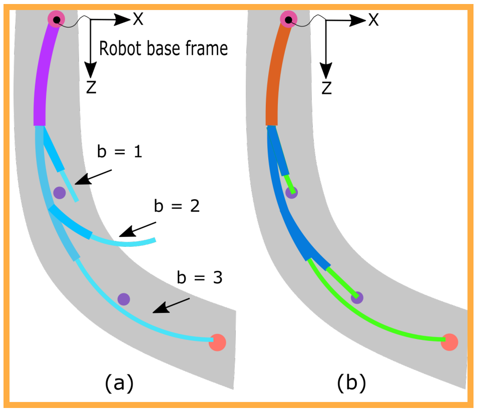


Fig. 10. (a) Initialization of the simultaneous optimization step is done based on the final configuration resulting from the sequential optimization step. Therefore, the first few configurations may not have good initial guesses and may be unable to initially reach the waypoints with that particular tube design. (b) After the simultaneous optimization simulation step, a single robot design that can reach all waypoints and the target, while avoiding collisions with the anatomy, is found.

Algorithm 1 CTR patient-specific optimization.

```
Inputs:
$\mathbf{\Gamma}$: Anatomical model
$\mathbf{S}_t$: Surgical target
$\mathbf{d}_{init}$: Initial robot design
$\mathbf{B}_{init}$: Initial robot base frame
Outputs:
$\mathbf{d}^*$: Robot design
$\mathbf{Q}^*$: Motion plan
$\mathbf{p}_p \leftarrow$ PathOpt($\mathbf{\Gamma}, \mathbf{S}_t$)
for each $\mathbf{p}_{p_i} \in \mathbf{p}_p, j = 1, 2 \ldots k$ do
    while collision = True or tip error $\geq$ tolerance do
        $\mathbf{d}_i$, tip error $\leftarrow$ SeqOpt($\mathbf{d}_{i-1}, \mathbf{\Gamma}, \mathbf{S}_t, \mathbf{B}_{init}$)
        collision $\leftarrow$ CollisionDetection($\mathbf{d}_i, \mathbf{\Gamma}$)
        if collision = True then
            $\zeta \leftarrow$ increase $\zeta$
        if tip error > tolerance then
            $\rho \leftarrow$ increase $\rho$
$\mathbf{d}^*, \mathbf{Q} \leftarrow$ SimOpt($\mathbf{d}_{1 \ldots j}, \mathbf{\Gamma}, \mathbf{S}_t$)
return $\mathbf{d}^*, \mathbf{Q}$
```

V. Clinical Examples

This section presents three use cases for the proposed optimization framework that illustrate its ability to optimize a CTR design and provide a path plan for different patients and procedures. The general optimization problem statement is presented in Table III. All optimizations were run on a standard PC with Intel Core i7-8700, 3.20GHz x 12 and 16GB RAM.

A. Clinical Example 1: Laryngoscopy

Laryngoscopy is a minimally invasive surgery that enables a close-up view of the larynx, as shown in Fig. 11. Some conditions, including bloody cough, bad breath, and difficulty swallowing, may require this procedure to inspect the abnormal area [52]. In addition, it can be used for vocal cord biopsy and for removing foreign objects [53]. In general, there are two visualization approaches — indirect and direct laryngoscopy. Indirect laryngoscopy is considered to be a less invasive approach, since the surgeon only needs to insert a mirror in order to get an image of the larynx. However, this method

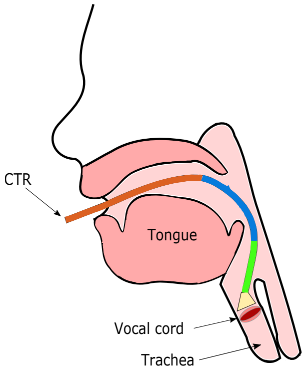
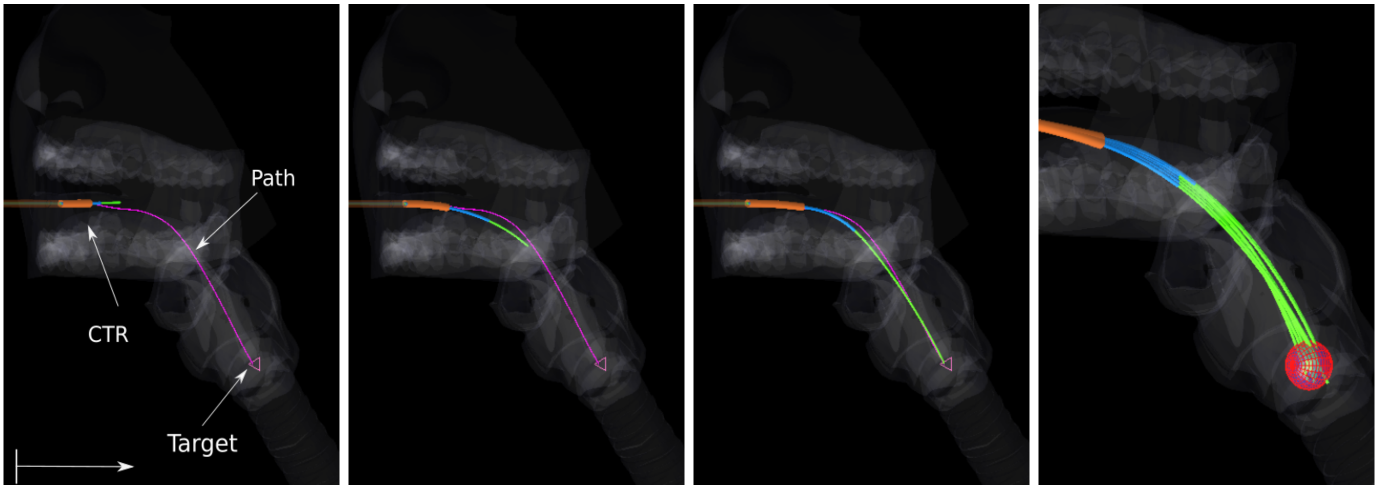


Fig. 11. CTRs have the potential to be used as flexible larynscopes to navigate through the mouth and into the larynx, where the surgeon can then obtain a clear view from a camera installed at the tip.



(a) (b) (c) (d)

Fig. 12. Timelapse of the optimized CTR deployed in the larynx. The CTR tubes are shown in green (tube 1), blue (tube 2), and orange (tube 3), and the optimized B-spline path is shown in pink. The pre-defined volume to be swept is represented by red spheres in (d).

has a high learning curve, is more expensive, and obtaining a clear view may be more difficult [54]. Direct laryngoscopy, on the other hand, requires a flexible tool that can go inside the throat to visualize the larynx area for further analysis. However, the narrow and constrained passage increases the contact between the surgical tool and the anatomy, often leading to complications, including infection, bleeding, and hoarseness [55].

A new flexible robotic scope, Flex Robotic System (Medrobotics, Raynham, USA), was introduced in 2015. It is a steerable scope, ranging from 18 mm to 28 mm in diameter, that can be used by the surgeon to navigate through the anatomy [56]. CTRs have the potential to be used as a steerable laryngoscope and would provide the surgeon with a more miniaturized system. The view angle and the shape of the robot can be controlled in order to obtain a desired set of images, while ensuring the absence of collisions with the anatomy. During operation, it is important to have a complete and clean view of the vocal cord area, making the reachability of the CTR an important goal.

TABLE III
THE OPTIMIZATION PROBLEM STATEMENT.

	Description	Lower*	Variable	Upper*	Optimization quantity		
					Path	Sequential	Simultaneous
Objective	$f_1(x)$ (Eq. (12))				✓		
	$f(x)$ (Eq. (15))					✓	✓
Optimization variables	Path control points		$\mathbf{c_p}$		$c \times 3$		
	Inner Diameter (mm)	0	ID_i	3.5		$n-1$	$n-1$
	Outer Diameter (mm)	0	OD_i	3.5		n	n
	Curved section length (mm)	0	L_{ci}			n	n
	Straight section length (mm)	0	L_{si}			n	n
	Tube curvature (mm^{-1})	0	κ_i			n	n
	Tube rotation at tip (rad)		ϕ_i			n	$b \times n \times h$
	Tube translation (mm)		β_i	0		n	$b \times n \times h$
	Robot base frame (3-D position and orientation)		$\mathbf{B}$			6	$6 \times h$
	Total (e.g. $c = 25, n = 3, b = 10, h = 1$)				75	26	80
Constraints	Start point		$\mathbf{s}_p$		3		
	Final point		$\mathbf{f}_p$		3		
	Tube clearance (mm)	0.1	$ID_i - OD_{i-1}$	0.16		$n-1$	$n-1$
	Tube wall thickness (mm)	0.05	$(OD_i - ID_i)/2$			n	n
	Distal exposed length (mm)	0	$(L_i + \beta_i)$-$(L_{i-1} + \beta_{i-1})$			$n-1$	$b \times (n-1) \times h$
	Proximal exposed length (mm)		$\beta_i - \beta_{i+1}$	0		$n-1$	$b \times (n-1) \times h$
	Material strain (Nitinol)	0	ϵ_i	0.08		$n \times 2$	$n \times h \times 2$
	Tip orientation (example 2)	0	$\mathbf{t_w} \cdot \mathbf{t_r}$	0		1	h
	Total (e.g. $n = 3, b = 10, h = 1$)				6	16	52

*: the absence of a value means that the corresponding optimization variable or constraint has no lower or upper bound.

Previous work aiming to maximize the percent coverage in a particular volume or workspace, W, mainly used a forward kinematics model with sampled joint values to generate robot configurations [11], [19], [57]. The percentage of the volume swept was then computed as the ratio of voxels containing tip positions, over the total number of voxels. Such approaches were inefficient as many robot configurations with tip positions far away from the desired volume were computed. A more recent approach used an inverse kinematics model, enabling the computation of only the robot configurations reaching desired positions in 3-D space [58]. However, a torsionally-rigid kinematic model was used, which lacks generality for robot designs where tubes experience torsion. Finally, there have been a few methods proposed for incorporating a motion plan needed to reach the desired 3-D volume to be swept [29], [30].

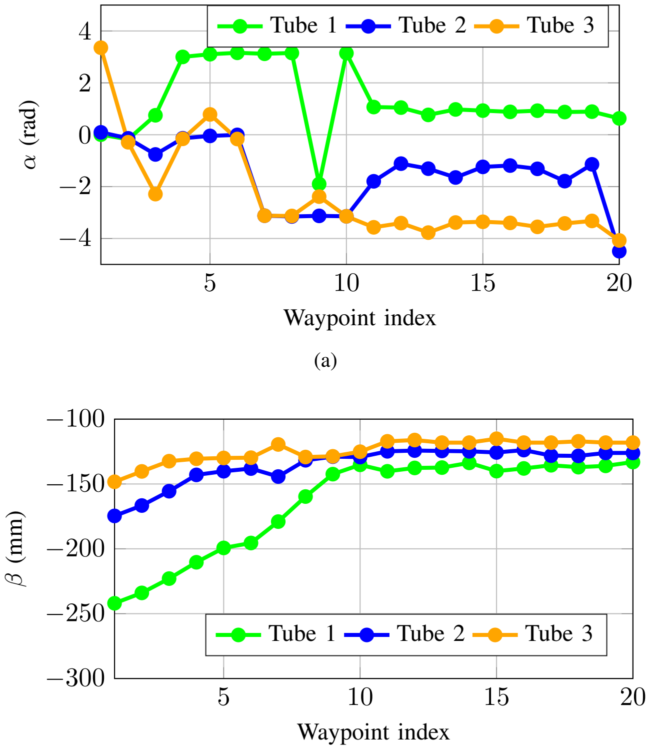
We present a new approach that takes advantage of the proposed framework and can efficiently solve the k CTR inverse kinematics problems simultaneously. We reformulate the reachability problem into an inverse kinematics problem with distance minimization between the robot tip position and the target positions inside the pre-defined volume. Thus, we generate a number of points, N, that are inside the workspace $W \in \mathbb{R}^3$, which are uniformly distributed inside the pre-defined volume. The reachable percentage of the optimized tube design, $\mathbf{d}^*$, is given by,

$$R_{reach}(\mathbf{d}^*) = \frac{N_r}{N}, \tag{17}$$

where N_r is the number of 3-D points that are considered reachable.

In this laryngoscopy example, the user pre-defined parameters are the tube number ($n = 3$), the number of links ($m = 50$), the tube material (Nitinol, where $\epsilon_{max} \approx 8\%$, and $E = 80$ GPa [59]), the number of waypoints ($b = 10$), the number of B-spline control points ($c = 25$), and the number of B-spline path points ($a = 100$). In addition, we select tube clearances between 0.1 mm to 0.16 mm for Nitinol tubes, which is standard for most CTR prototypes. The wall thickness is set to be 0.05 mm. ID_1 is selected to be 0.66 mm, based on a camera (minnieCam-XS, Enable, Inc.) that could be used for this application. Finally, a set of points ($N = 10$) are selected and distributed inside a pre-defined volume, represented by a sphere with a diameter of 12 mm. These extra points are treated as additional waypoints that the tip needs to reach in the simultaneous optimization step. Therefore the total number of configurations to solve becomes $b = 20$. The optimization quantity can be computed from Table III. This overall problem includes a total of 140 optimization variables and 91 constraints.

The deployment sequence in Fig. 12 shows that the optimized CTR can safely navigate through the anatomy by following the B-spline curve path (represented in pink) from the mouth to the vocal cords. Fig. 12d also shows the 10 robot configurations that can reach the pre-defined points inside the volume. The reachable percentage, R_{reach}, is calculated using Eq. (17) to be 60 %, based on a tip position tolerance of 2 mm.



(a)

(b)

Fig. 13. The optimized joint values in the Laryngoscopy example, starting from the first waypoint and ending at the target point. (a) Shows the tube base angles of each tube during deployment, and (b) shows their translations.

The optimized tube design parameters are reported in Table IV. Fig. 13 shows the optimized joint values that form a safe motion plan during deployment. Since the joint values of the robot configurations from waypoints 11 to 20 are optimized in order to reach target points in a pre-defined volume that is relatively small, the translation β_1, has only small variations between configurations.

In addition, we confirmed the accuracy of our kinematics model implementation that requires an interpolation, as previously explained. For the optimized tube set with 50 links, the average tip position error compared to the exact kinematics model is less than 3 mm, corresponding to $0.2-1\%$ of the arc length. The computation time for the laryngoscopy example is 6 hours and 35 minutes.

TABLE IV
OPTIMIZED TUBE DESIGN VARIABLES

Variable	Optimal values			Computation time
	Tube 1	Tube 2	Tube 3	
ID (mm)	0.66	0.88	1.29	6 hr, 35 mins
OD (mm)	0.76	1.16	3.31	
L_s (mm)	162.98	130	84.79	
L_c (mm)	114.77	70.33	79.28	
κ (mm^{-1})	0.0038	0.0174	0.0028	

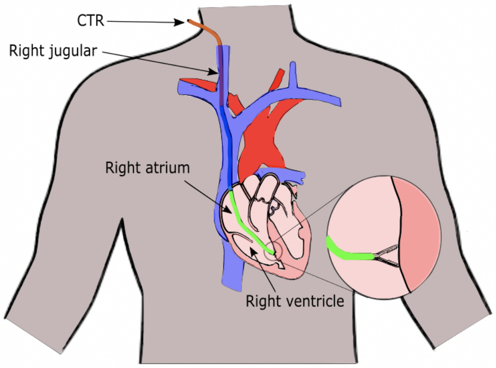


Fig. 14. Illustration of the relevant anatomy for myocardial biopsy procedures. The CTR would be inserted from the right jugular close to the neck, and it would navigate through the right atrium in order to take the tissue sample from the septum inside the right ventricle.

B. Clinical Example 2: Myocardial Biopsy

Myocardial biopsy, also known as heart biopsy and cardiac biopsy, is a minimally invasive procedure that removes a small sample of heart tissue located in the right ventricular septum. The use of this diagnostic technique is primarily for the detection of heart disease and the monitoring of cardiac transplantation rejections [60]. Typically, a miniature catheter, ranging from 1.5 mm to 4 mm in diameter, is manually deployed into the heart via the right jugular vein [61]. To visualize the catheter, most procedures are performed using fluoroscopy, which can result in substantial radiation and large sampling errors. Alternatives include the use of cardiovascular magnetic resonance (CMR) and echocardiography. However, the challenges in visualization of the soft tissue and the catheter tip directionality [62] result in uncertainties that can lead to an increased risk for the patient's safety. The complications in the biopsy procedure includes carotid puncture, bleeding, and arterial damage.

The use of advanced surgical tools can help to reduce patient discomfort, tissue damage, and risks of complications [63]. CTRs, in particular, have the potential to be used for heart biopsies due to their flexibility and steerability. The complex and constrained environment requires a specific robot design that can navigate along a collision-free path from the jugular, through the atrium and ventricle, in order to reach the septum, as shown in Fig. 14. In addition, one task-specific constraint is the orientation of the robot tip, since it can reduce the user's effort and patient's risk when taking the sample tissue [64]. Thus, based on the suggestion from clinicians at the University of California, San Diego, the desired angle between the tip and the septum is set to 90°. The tip orientation constraints can be written as

$$\mathbf{t_w} \cdot \mathbf{t_r} = 0, \tag{18}$$

where t_w is the tangent vector of the septum wall where the target is located and t_r is the tangent vector of the robot tip.

In this myocardial biopsy example, the user pre-defined parameters are the tube number ($n = 3$), the number of links ($m = 50$), the tube material (Nitinol, which has a strain

TABLE V
OPTIMIZED TUBE DESIGN VARIABLES.

Variable	Optimal values			Computation time
	Tube 1	Tube 2	Tube 3	
ID (mm)	0.60	0.87	1.14	
OD (mm)	0.76	0.99	2.31	
L_s (mm)	200.79	127.31	85.00	7 hr, 33 mins
L_c (mm)	127.31	120.42	93.98	
κ (mm^{-1})	0.0115	0.0104	0.0042	

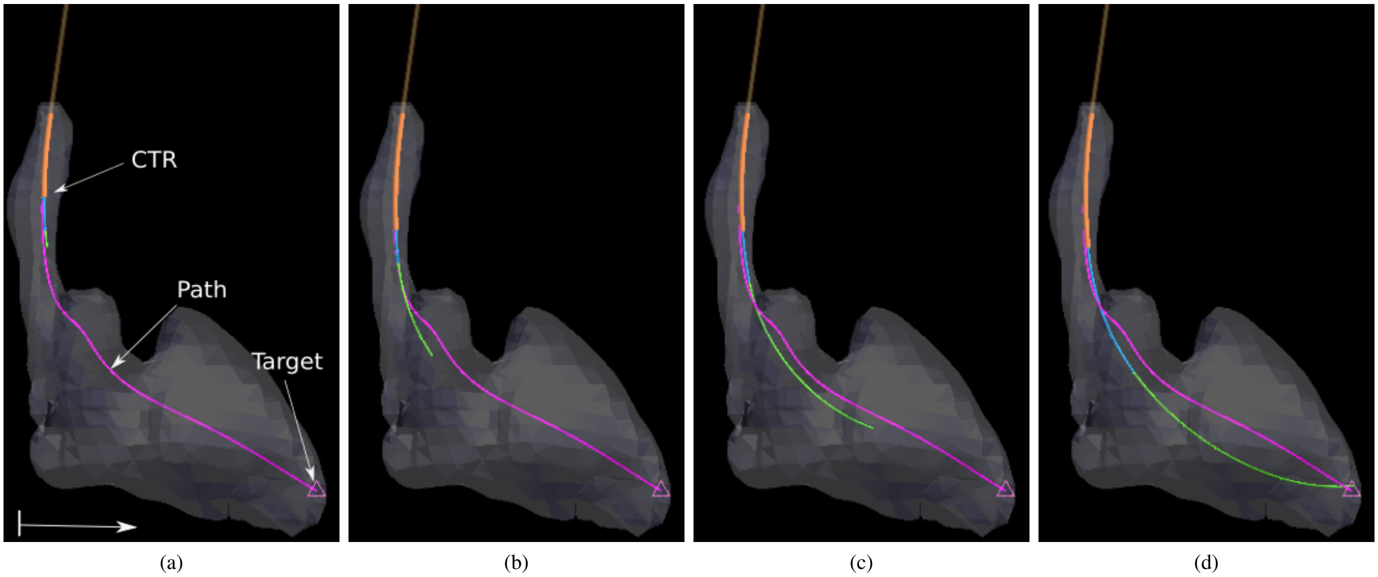


Fig. 15. Illustration of the obtained deployment sequence for the biopsy procedure. The design parameters and joint values of the three-tube CTR are optimized in order to safely navigate through the jugular vein and right atrium to reach the right ventricle. (d) shows the final configuration that reaches the surgical target.

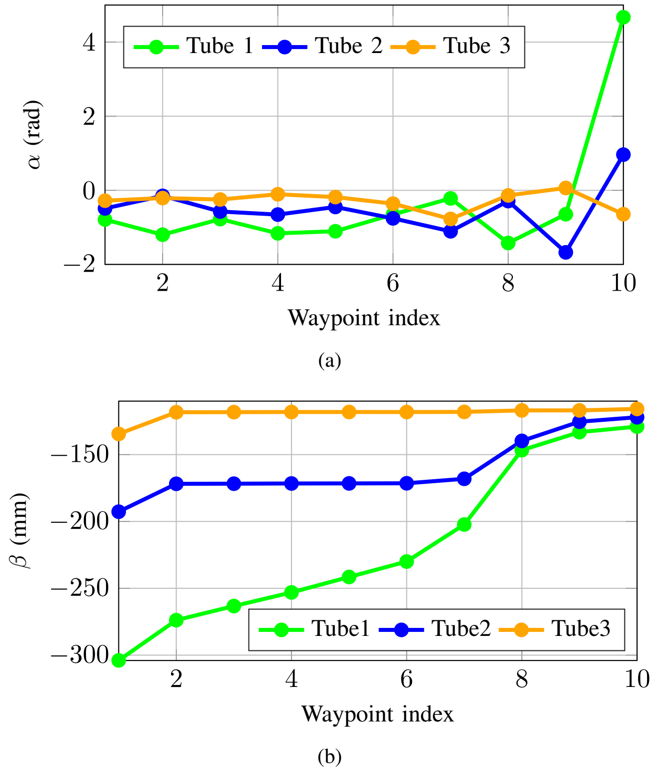


Fig. 16. The optimized joint values from the first waypoint to the target point of the Myocardial biopsy example are shown here. (a) shows the rotation of the each tube (b) shows the translation of each tube

limit, $\epsilon_{max} \approx 8\%$ and $E = 80$ GPa [59]), the number of waypoints ($b = 10$), the number of B-spline control points ($c = 25$), and the number of B-spline path points ($a = 100$). ID_1 is chosen to be 0.6 mm due to the size of the biopsy forceps. The wall thickness and the tube clearance are remain the same as previous example. The optimization quantity can be computed from Table III. This overall problem includes a total of 80 optimization variables and 52 constraints. The final result is presented in Fig. 15. The deployment sequence of the optimized CTR from the initial configuration to the final configuration is shown in Fig. 15a - 15d. As seen in Fig. 15d, the CTR can reach the final surgical target, with a tip position error of 2 mm and at an angle of 84° between the robot tip and the septum wall.

The optimized robot design, $\mathbf{d}^*$, is presented in Table V. Tube 1 and tube 2 have higher optimized curvatures (κ_1 and κ_2) compared to tube 3, whose curvature (κ_3) is close to the lower bound ($\kappa = 0$). The joint values of each tube are illustrated in Fig. 16. It can be seen that the tubes have significant changes in rotation towards the end of the deployment. In addition, tube 1 is the only tube with significant translation during the deployment. The average interpolation error with the optimized tube sets is 0.2-1.2% of the arc length with 50 links. The computation time of the entire optimization, including the path, sequential and patient-specific simultaneous optimization, is 7 hours and 33 minutes.

C. Clinical Example 3: Design Across Multiple Anatomies

The previous two examples demonstrated that the framework can successfully optimize a robot design for a specific patient anatomy. In some scenarios, it could be beneficial to have a population-specific robot, or one that would work across multiple patient anatomies. Having a single robot design could be feasible if the surgical task is the same and the population is sufficiently similar, at least with regards to the anatomy in the area of interest.

For this example, we selected three sets of CT images, which we reconstructed into 3-D meshes. The user-defined parameters remained the same, with $m = 50$, $n = 3$, $b = 10$. The only new parameter is the number of patients, $h = 3$. Therefore, this optimization problem has a total number of 212 optimization variables and 146 constraints for the population-specific simultaneous optimization. The results are shown in Fig. 17, where the left column shows the three robot designs, including three tube sets and cross-sectional view of the robots, resulting from the patient-specific simultaneous optimization step. The right column in Fig. 17 shows the final robot

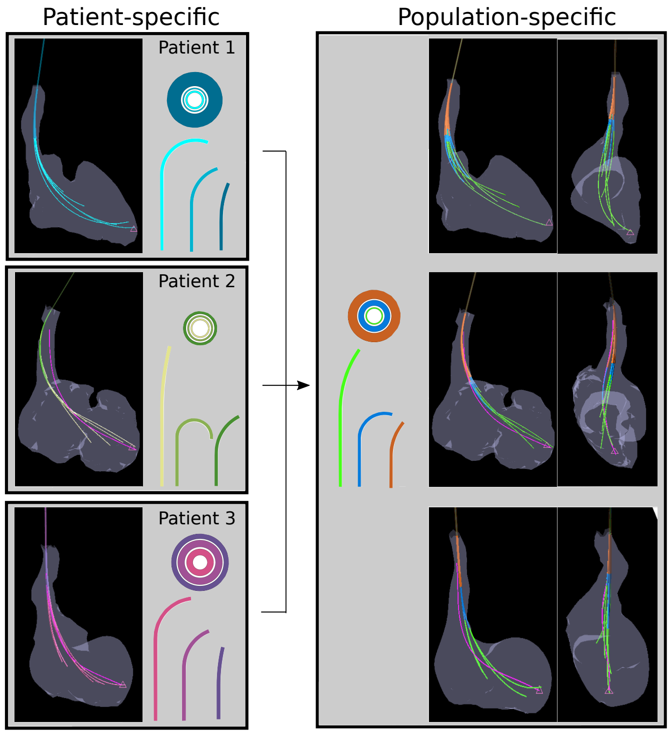
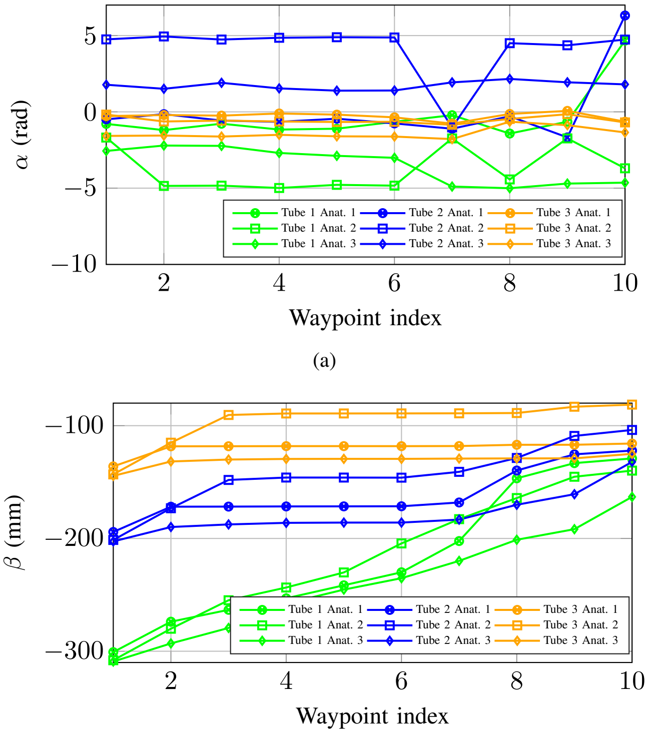


Fig. 17. The result of the population-specific ($h = 3$) design optimization is illustrated. Three patient's heart model was selected in this example. The patient-specific simultaneous optimization was performed individually, as shown on the left. The final population-specific robot design showing the robot can safely deploy through three different anatomies.

TABLE VI
OPTIMIZED TUBE DESIGN VARIABLES.

Variable	Optimal values			Computation time
	Tube 1	Tube 2	Tube 3	
ID (mm)	0.60	0.84	1.4	Patient 1: 7 hr, 33 min
OD (mm)	0.70	1.29	2.15	Patient 2: 8 hr, 15 min
L_s (mm)	199.61	127.41	85.35	Patient 3: 6 hr, 24 min
L_c (mm)	164.37	118.66	92.56	Final opt.: 4 hr, 27 min
κ (mm^{-1})	0.0037	0.0156	0.0073	



(a)

(b)

Fig. 18. The optimized joint values from the first waypoint to the target point of the three patient anatomies are shown here. (a) shows the rotation of each tube for each anatomy (b) shows the translation of each tube for each anatomy.

design visualized in three patient anatomies with two viewing angles. This example demonstrates the ability of our proposed approach to handle the variations between three anatomical models and find a robot design that can reach the target with an average of 3.2 mm of tip position error. Also, the angle between the robot tip and the septum wall was 93° on average, which satisfies the requirements. The optimized tube design variables for the three anatomies are shown in Table VI.

It should be noted that although customized Nitinol tubes might be expensive, the cost is generally a secondary concern to functionality and safety in medical procedures. Moreover, 3-D printed tubes and other low-cost materials could instead be used as an alternative solution [65]. As shown in Fig. 18, the optimized joint variable, β, for each patient is almost identical throughout deployment. The only noticeable difference among the three patients is the value of α towards the end of the deployment sequence. In particular, differences in the rotation of tube 1 can be seen in the side view of the three anatomies in Fig. 17. Qualitatively, the geometry of patient 3 is noticeably different from the other two, including the more tortuous nature of the jugular vein. Although the target can still be reached for all patients, future work should include methods for grouping patients into populations based on anatomical similarities.

VI. Conclusion

We presented a gradient-based optimization approach that is based on the modular and unified derivative framework, OpenMDAO. The gradient-based approach allows us to efficiently solve large scale optimization problems without adding assumptions or reducing the design space. We introduced a new method consisting of three major steps, which can solve the CTR patient-specific and population-specific design optimization and motion planning, simultaneously. The new formulation of the anatomical and plastic deformation constraints are integrated into the optimization problem. In addition, due to the modular approach, the framework is highly general, and offers an efficient way to formulate new optimization problems by customizing task-specific constraints, objectives, and patient numbers. Furthermore, the proposed framework can be applied to different patients and surgical tasks and requires a very low number of inputs from the user.

As part of our future work, we plan to integrate CTR stability, as well models that consider the external loading, tube clearance, and friction. Taking these effects into account can help guarantee safety during deployment in a real-world setting. There is also a need to investigate a new formulation for the collision avoidance with the anatomy that does not rely on computing the entire point cloud, which is computationally expensive. In addition, the computation time can be reduced by parallelizing the algorithm across anatomies and configurations.

Acknowledgment

The authors would like to thank Dr. Philip Weissbrod and Dr. Albert Hsiao for their assistance with providing clinical context, motivation, and data for this work.

Appendix: Derivative Computation in OpenMDAO

The OpenMDAO optimization library uses an object-oriented programming paradigm to facilitate the modular implementation of complex models, in a manner that facilitates derivative computation.

The hierarchy of classes in OpenMDAO consists of *components*, *groups*, the *model*, and the *problem*. In OpenMDAO, the user performs all the computations using *components*. The *components* are organized into a hierarchy using *groups*. The user specifies connections between outputs and inputs of *components* to form the complete analysis, which is called the *model*. With this architecture, the user can easily change the objective and constraints, as well as intermediate computations by switching to different *components*.

OpenMDAO automates the computation of total derivatives using the unified derivatives equation (UDE). The UDE unifies the adjoint method, chain rule, and all other known methods for computing discrete derivatives of models using a single matrix equation, given by [36]

$$\frac{\partial \boldsymbol{\mathcal{R}}}{\partial \boldsymbol{\nu}} \frac{d\boldsymbol{\nu}}{d\boldsymbol{r}} = \mathcal{I} = \frac{\partial \boldsymbol{\mathcal{R}}}{\partial \boldsymbol{\nu}}^T \frac{d\boldsymbol{\nu}}{d\boldsymbol{r}}^T , \tag{19}$$

where $\boldsymbol{\nu}$ is the concatenated vector of the model's inputs, outputs, and the states; $\boldsymbol{\mathcal{R}}$ is a vector-valued function that acts as a residual function for $\boldsymbol{\nu}$; $\boldsymbol{r}$ is simply the output of the $\boldsymbol{\mathcal{R}}$ function; and $\mathcal{I}$ is the identity matrix.

The significance of the UDE is that different choices of the $\boldsymbol{\nu}$ vector (at different levels of model decomposition) leads to different methods for computing derivatives. For instance, excluding all states and treating the model as a black-box function mapping inputs to outputs yields the trivial result that the total derivatives are equal to the partial derivatives. Alternatively, considering a model in which the states are explicitly defined and have no feedback loops yields a linear equation that, when applying back-substitution, yields the chain rule. Yet another example is defining $\boldsymbol{\nu}$ as the inputs, outputs, and a single vector of state variables—block solution of the UDE with this definition of $\boldsymbol{\nu}$ yields the adjoint method.

Remark: By "solution of the UDE", we are referring to the solution of the multiple-right-hand-side linear system that is defined by Eq. (19). Choosing the left or right equality in Eq. (19) amounts to a choice of the commonly known forward or reverse modes of differentiation. In the forward mode, solving the left equality for the ith column yields the derivatives of all variables in ν with respect to the ith variable, while in the reverse mode, solving the right equality for the ith column yields the derivatives of the ith variable with respect to all variables in ν. In this work, we use the reverse mode, because it yields the full gradient vector for one output (objective or constraint) at the cost of a single linear solution, and this cost does not increase with the number of optimization variables.

The inputs, outputs, and states mentioned previously are defined here from the perspective of the optimization model, rather than that of the overall framework (shown in Fig. 7). The model's inputs consist of optimization variables, as well as parameters that are fixed during optimization. The model's outputs consist of the objective and constraint variables of the optimization problem, and the model's states consist of all other intermediate variables computed within the model in the process of mapping the inputs to the outputs. In the sequential and simultaneous optimization problems shown in Table III, the inputs would include all the optimization variables listed in the table for the respective problem, the outputs would include the objective ($f(x)$) and all the constraints listed in the table for the respective problem, and the states would include all intermediate variables in the model, e.g., tube angle, ψ_i, rotation matrix, $\boldsymbol{R}$, backbone position, $\boldsymbol{p}$, and tube stiffnesses, k_{ib} and k_{it}.

The residual function, $\mathcal{R}$, is defined differently for inputs, outputs, and states. For an input, such as κ_i, that takes on a value of 1 in a particular optimization iteration (for instance), the residual would be defined as $\kappa_i - 1$, such that when the function yields zero, $\kappa_i = 1$, as desired. For an output or an explicitly defined state, such as bending strain ($\epsilon_{i_x} = |\chi_i - \chi_{eq}|/\chi_i$), the residual would be defined as $\epsilon_{i_x} - |\chi_i - \chi_{eq}|/\chi_i$. For a state that is implicitly defined, such as the tube pre-curvature ($\mathbf{u} = \mathbf{K}^{-1}\sum_{i=1}^{n}\mathbf{K}_i(\mathbf{R}_{\psi_i}\mathbf{u}_i^* - \dot{\psi}_i\mathbf{e}_3)$), the residual function would be $\mathbf{K}\mathbf{u} - \sum_{i=1}^{n}\mathbf{K}_i(\mathbf{R}_{\psi_i}\mathbf{u}_i^* - \dot{\psi}_i\mathbf{e}_3)$. Again, the residual is defined in this way so that when the residual evaluates to zero, the desired solution is obtained for state $\mathbf{u}$. In the case of the residual definition, $\mathbf{K}\mathbf{u} - \sum_{i=1}^{n}\mathbf{K}_i(\mathbf{R}_{\psi_i}\mathbf{u}_i^* - \dot{\psi}_i\mathbf{e}_3)$, the significance of rearranging to remove the matrix inverse is that due to this choice, solving the UDE is equivalent to the adjoint method. Otherwise, if we kept the matrix inverse, applying the finite-difference method to compute the partial derivatives of $\mathbf{u}$ would require n applications of the matrix inverse where n is the size of the matrix.

In our model, we have a mix of explicitly and implicitly defined states. Therefore, when OpenMDAO solves the UDE to compute derivatives, this process is equivalent to applying a mixture of the chain rule and the adjoint method, which is the desired (most efficient) approach for our model. However, the complexity of our model would have made manual formulation and implementation of the chain rule and adjoint equations a very laborious and error-prone process, which our use of OpenMDAO has automated.

Because of OpenMDAO's use of the UDE, the model is implemented as a set of small units of code—the *components* described previously. The analytical partial derivative of each output with respect to each input needs to be provided to OpenMDAO. Since each small unit performs relatively simple computations, the partial derivatives are easy to compute. OpenMDAO then assembles the $\partial\mathcal{R}/\partial\boldsymbol{\nu}$ Jacobian matrix from these partial derivatives of each component and solves the UDE.

REFERENCES

[1] G. S. Chirikjian and J. W. Burdick, "A hyper-redundant manipulator," *IEEE Robotics Automation Magazine*, vol. 1, no. 4, pp. 22–29, 1994.

[2] G. Robinson and J. B. C. Davies, "Continuum robots - a state of the art," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 4, 1999, pp. 2849–2854 vol.4.

[3] I. D. Walker, "Continuous backbone "continuum" robot manipulators," *ISRN Robot*, pp. 1–19, 2013.

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

[5] P. Sears and P. Dupont, "A steerable needle technology using curved concentric tubes," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 2850–2856.

[6] R. J. Webster, A. M. Okamura, and N. J. Cowan, "Toward active cannulas: Miniature snake-like surgical robots," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 2857–2863.

[7] H. B. Gilbert, D. C. Rucker, and R. Webster III, "Concentric tube robots: The state of the art and future directions," *Robotics Research*, vol. 114, no. 6, pp. 253–269, 2016.

[8] T. Anor, J. R. Madsen, and P. Dupont, "Algorithms for design of continuum robots using the concentric tubes approach: A neurosurgical example," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 667–673.

[9] L. G. Torres, R. J. Webster, and R. Alterovitz, "Task-oriented design of concentric tube robots using mechanics-based models," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 4449–4455.

[10] M. Boushaki, C. Liu, B. Herman, V. Trevillot, M. Akkari, and P. Poignet, "Optimization of concentric-tube robot design for deep anterior brain tumor surgery," in *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2016, pp. 1–6.

[11] J. Burgner, P. J. Swaney, D. C. Rucker, H. B. Gilbert, S. T. Nill, P. T. Russell, K. D. Weaver, and R. J. Webster, "A bimanual teleoperated system for endonasal skull base surgery," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 2517–2523.

[12] H. Alfalahi, F. Renda, and C. Stefanini, "Concentric tube robots for minimally invasive surgery: Current applications and future opportunities," *IEEE Transactions on Medical Robotics and Bionics*, vol. 2, no. 3, pp. 410–424, 2020.

[13] 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 Transactions on Robotics*, vol. 31, no. 1, pp. 67–84, 2015.

[14] C. Girerd, T. Lihoreau, K. Rabenorosoa, B. Tamadazte, M. Benassarou, L. Tavernier, L. Pazart, E. Haffen, N. Andreff, and P. Renaud, "In vivo inspection of the olfactory epithelium: Feasibility of robotized optical biopsy," *Annals of Biomedical Engineering (ABME)*, vol. 46, pp. 1951–1961, June 2018.

[15] Z. Mitros, S. H. Sadati, R. Henry, L. Da Cruz, and C. Bergeles, "From theoretical work to clinical translation: Progress in concentric tube robots," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, 2021.

[16] C. Bedell, J. Lock, A. Gosline, and P. E. Dupont, "Design optimization of concentric tube robots based on task and anatomical constraints," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 398–403.

[17] J. Burgner, P. J. Swaney, D. C. Rucker, H. B. Gilbert, S. T. Nill, P. T. Russell, K. D. Weaver, and R. J. Webster, "A bimanual teleoperated system for endonasal skull base surgery," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 2517–2523.

[18] D. P. Anor T, Madsen JR, "Algorithms for design of continuum robots using the concentric tubes approach: A neurosurgical example," in *IEEE Int Conf Robot Autom*, 2011, pp. 667–673.

[19] J. Burgner, H. B. Gilbert, and R. J. Webster, "On the computational design of concentric tube robots: Incorporating volume-based objectives," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 1193–1198.

[20] J. Ha, F. C. Park, and P. E. Dupont, "Achieving elastic stability of concentric tube robots through optimization of tube precurvature," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 864–870.

[21] J. Granna, "Multi-objective particle swarm optimization for the structural design of concentric tube continuum robots for medical applications," in *Dissertation, Leibniz Universität Hannover, Germany*, 2019.

[22] A. Kuntz, M. Fu, and R. Alterovitz, "Planning high-quality motions for concentric tube robots in point clouds via parallel sampling and optimization," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 2205–2212.

[23] K. Leibrandt, C. Bergeles, and G.-Z. Yang, "Concentric tube robots: Rapid, stable path-planning and guidance for surgical use," *IEEE Robotics Automation Magazine*, vol. 24, no. 2, pp. 42–53, 2017.

[24] S. Niyaz, A. Kuntz, O. Salzman, R. Alterovitz, and S. S. Srinivasa, "optimizing motion-planning problem setup via bounded evaluation with application to following surgical trajectories," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1355–1362.

[25] L. Lyons, R. Webster, and R. Alterovitz, "Motion planning for active cannulas," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 801–806.

[26] L. Lyons, R. Webster, and R. Alterovitz, "Planning active cannula configurations through tubular anatomy," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 2082–2087.

[27] L. G. Torres and R. Alterovitz, "Motion planning for concentric tube robots using mechanics-based models," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 5153–5159.

[28] L. G. Torres, R. J. Webster, and R. Alterovitz, "Task-oriented design of concentric tube robots using mechanics-based models," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 4449–4455.

[29] C. Baykal, L. G. Torres, and R. Alterovitz, "Optimizing design parameters for sets of concentric tube robots using sampling-based motion planning," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 4381–4387.

[30] C. Baykal, C. Bowen, and R. Alterovitz, "Asymptotically optimal kinematic design of robots using motion planning," in *Autonomous Robots*, vol. 43, no. 2, 2019, pp. 345–357.

[31] J. T. Hwang, "A modular approach to large-scale design optimization of aerospace systems," Ph.D. dissertation, University of Michigan, 2015.

[32] R. E. Perez, P. W. Jansen, and J. R. Martins, "pyopt: a python-based object-oriented framework for nonlinear constrained optimization," *Structural and Multidisciplinary Optimization*, vol. 45, no. 1, pp. 101–118, 2012.

[33] J. Li and M. Zhang, "Data-based approach for wing shape design optimization," *Aerospace Science and Technology*, vol. 112, p. 106639, 2021.

[34] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor, "OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization," *Structural and Multidisciplinary Optimization*, vol. 59, no. 4, pp. 1075–1104, April 2019.

[35] J. R. Martins and J. T. Hwang, "Review and unification of methods for computing derivatives of multidisciplinary computational models," *AIAA journal*, vol. 51, no. 11, pp. 2582–2599, 2013.

[36] J. T. Hwang and J. R. Martins, "A computational architecture for coupling heterogeneous numerical models and computing coupled derivatives," *ACM Transactions on Mathematical Software (TOMS)*, vol. 44, no. 4, pp. 1–39, 2018.

[37] P. E. Gill, W. Murray, and M. A. Saunders, "Snopt: An sqp algorithm for large-scale constrained optimization," *SIAM review*, vol. 47, no. 1, pp. 99–131, 2005.

[38] R. J. Webster, J. M. Romano, and N. J. Cowan, "Mechanics of precurved-tube continuum robots," *IEEE Transactions on Robotics*, vol. 25, no. 1, pp. 67–78, 2009.

[39] D. C. Rucker, R. J. Webster III, G. S. Chirikjian, and N. J. Cowan, "Equilibrium conformations of concentric-tube continuum robots," *The International journal of robotics research*, vol. 29, no. 10, pp. 1263–1280, 2010.

[40] C. Bergeles and P. E. Dupont, "Planning stable paths for concentric tube robots," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 3077–3082.

[41] R. J. Webster and D. C. Rucker, "Parsimonious evaluation of concentric-tube continuum robot equilibrium conformation," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 9, pp. 2308–2311, 2009.

[42] J. T. Hwang and D. Munster, "Solution of ordinary differential equations in gradient-based multidisciplinary design optimization," in *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2018, p. 1646.

[43] J. Ha, F. C. Park, and P. E. Dupont, "Elastic stability of concentric tube robots subject to external loads," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 6, pp. 1116–1128, 2016.

[44] J. Ha, G. Fagogenis, and P. E. Dupont, "Modeling tube clearance and bounding the effect of friction in concentric tube robot kinematics," *IEEE Transactions on Robotics*, vol. 35, no. 2, pp. 353–370, 2019.

[45] B.-K. J. Granna J, Nabavi A, "Computer-assisted planning for a concentric tube robotic system in neurosurgery," *Int J Comput Assist Radiol Surg.*, vol. 14, no. 2, pp. 335–344, 2019.

[46] S. W. Robertson, A. R. Pelton, and R. O. Ritchie, "Mechanical fatigue and fracture of nitinol," *International Materials Reviews*, vol. 57, no. 1, pp. 1–37, 2012.

[47] R. Adharapurapu and K. Vecchio, "Superelasticity in a new bioimplant material: Ni-rich 55niti alloy," *Experimental mechanics*, vol. 47, no. 3, pp. 365–371, 2007.

[48] B. Yang, *Stress, strain, and structural dynamics: an interactive handbook of formulas, solutions, and MATLAB toolboxes*. Academic Press, 2005.

[49] E. J. Barbero, *Finite element analysis of composite materials using AbaqusTM*. CRC press Boca Raton, 2013, vol. 2103.

[50] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, "Path planning and trajectory planning algorithms: A general overview," *Motion and operation planning of robotic systems*, pp. 3–27, 2015.

[51] A. Ravankar, A. A. Ravankar, Y. Kobayashi, Y. Hoshino, and C.-C. Peng, "Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges," *Sensors*, vol. 18, no. 9, p. 3170, 2018.

[52] American Cancer Society. Laryngoscopy. [Online]. Available: https://www.cancer.org/treatment/understanding-your-diagnosis/tests/endoscopy/laryngoscopy.html

[53] S. Harris. A Close-Up Look at Laryngoscopy. (2018, Sep 1). [Online]. Available: https://www.healthline.com/health/laryngoscopy#side-effects

[54] D. R. Cheyne and P. Doyle, "Advances in laryngoscopy: rigid indirect laryngoscopy," *F1000 medicine reports*, vol. 2, 2010.

[55] obpmedical. What is Laryngoscopy and Who Needs This Procedure? (2018, Sep 17). [Online]. Available: https://obpmedical.com/what-is-laryngoscopy/

[56] P. Gorphe, "A contemporary review of evidence for transoral robotic surgery in laryngeal cancer," *Frontiers in oncology*, vol. 8, no. 121, pp. 209–225, 2018.

[57] J. Burgner, P. J. Swaney, R. A. Lathrop, K. D. Weaver, and R. J. Webster, "Debulking from within: A robotic steerable cannula for intracerebral hemorrhage evacuation," *in IEEE Transactions on Biomedical Engineering*, vol. 60, no. 9, pp. 2567–2575, 2013.

[58] J. Granna, Y. Guo, K. D. Weaver, and J. Burgner-Kahrs, "Comparison of optimization algorithms for a tubular aspiration robot for maximum

coverage in intracerebral hemorrhage evacuation," *Journal of Medical Robotics Research*, vol. 02, no. 01, p. 1750004, 2017.

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

[60] L. C. Francis R, "Myocardial biopsy: techniques and indications," *Heart*, vol. 104, no. 11, pp. 950–958, 2018.

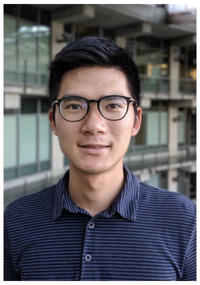
[61] O. Kiamanesh and M. Toma, "The state of the heart biopsy: A clinical review," *CJC Open*, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2589790X20302055

[62] A. M. From, J. J. Maleszewski, and C. S. Rihal, "Current status of endomyocardial biopsy," in *Mayo Clinic Proceedings*, vol. 86, no. 11. Elsevier, 2011, pp. 1095–1102.

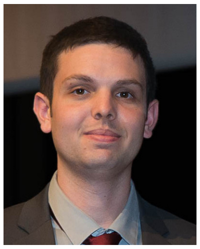
[63] R. Grankvist, A. Chireh, M. Sandell, A. K. Mukarram, N. Jaff, I. Berggren, H. Persson, C. Linde, F. Arnberg, J. Lundberg *et al.*, "Myocardial micro-biopsy procedure for molecular characterization with increased precision and reduced trauma," *Scientific reports*, vol. 10, no. 1, pp. 1–11, 2020.

[64] M. M. Sandra V. Chaparro. Endomyocardial Biopsy. [Online]. Available: https://thoracickey.com/endomyocardial-biopsy-5/

[65] T. K. Morimoto and A. M. Okamura, "Design of 3-d printed concentric tube robots," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1419–1430, 2016.



Jui-Te Lin received the B.S degree in mechanical and electro-mechanical engineering from Tamkang University, New Taipei City, Taiwan in 2017. He received the M.S. degree in 2020 and is currently working towards the Ph.D. degree in mechanical engineering at the University of California, San Diego. His research focuses on the design and control of continuum robots.



Cédric Girerd received an Engineering degree in Mechatronics from SIGMA Clermont and a Master of Science in Robotics from University Blaise Pascal in Clermont-Ferrand, France, in 2014. He also received a Ph.D. degree in Robotics from the University of Strasbourg, France, in 2018. He is currently working as a Postdoctoral Researcher at the University of California, San Diego. His research focuses on the design and control of continuum and soft robots.

Jiayao Yan received the B. Eng degree from Jilin University, China, in 2017, and the M.S. degree from the University of California San Diego (UCSD) in 2020. She is currently pursuing the Ph.D. degree with UCSD. Her research focuses on the structural design and control of soft robots using large-scale gradient-based optimization.

John T. Hwang received the B.A.Sc. degree in engineering science from the University of Toronto in 2010, and he received the M.S.E. degree in aerospace engineering (2012), M.S. degree in mathematics (2013), and Ph.D. degree in aerospace engineering (2015) from the University of Michigan. He is currently an Assistant Professor in the Department of Mechanical and Aerospace Engineering at the University of California, San Diego. His research focuses on multidisciplinary design optimization methods and their application to the design of aircraft, spacecraft, robots, and other engineering systems.

Tania K. Morimoto received the B.S. degree from Massachusetts Institute of Technology, Cambridge, MA, in 2012 and the M.S. and Ph.D. degrees from Stanford University, Stanford, CA, in 2015 and 2017, respectively, all in mechanical engineering. She is currently an Assistant Professor in the Mechanical and Aerospace Engineering Department and in the Surgery Department at the University of California, San Diego. Her research interests include robotics, haptics, and engineering education.