

Task-oriented Design of Concentric Tube Robots using Mechanics-based Models

Luis G. Torres, Robert J. Webster III, and Ron Alterovitz

Abstract—We introduce a method for task-oriented design of concentric tube robots, which are highly modular, tentacle-like robots with the potential to enable new minimally invasive surgical procedures. Our objective is to create a robot design on a patient-specific and surgery-specific basis to enable the robot to reach multiple clinically relevant sites while avoiding anatomical obstacles. Our method uses a mechanically accurate model of concentric tube robot kinematics that considers a robot’s time-varying shape throughout the performance of a task. Our method combines a search over a robot’s design space with sampling-based motion planning over its configuration space to compute a design under which the robot can feasibly perform a specified task without damaging surrounding tissues. To accelerate the algorithm, we leverage design coherence, the observation that collision-free configuration spaces of robots of similar designs are similar. If a solution exists, our method is guaranteed, as time is allowed to increase, to find a design and corresponding feasible motion plan. We provide examples illustrating the importance of using mechanically accurate models during design and motion planning and demonstrating our method’s effectiveness in a medically motivated simulated scenario involving navigation through the lung.

I. INTRODUCTION

A robot’s effectiveness at performing a task depends largely on its design. A robot’s design can be seen as a set of parameters that are fixed throughout the robot’s use. Given the tasks we wish the robot to perform, we can design the robot to most ably perform the tasks while ensuring that the robot does not damage itself or its environment.

In this paper we investigate *task-oriented design* of concentric tube robots, which are minimally invasive medical devices capable of following curved paths through body cavities. These robots may enable physicians to perform new surgical tasks requiring greater dexterity than possible with current instruments, including skull-base surgery [3], neurosurgery [1], operation on a fetus in the womb [8], and lung procedures [18]. In Fig. 1 we show by simulation how, if designed correctly, a single concentric tube robot can access multiple surgical sites in the bronchial tubes while avoiding damage to sensitive structures in the lung.

Concentric tube robots are composed of nested, pre-curved tubes, usually shaped with a straight section followed by a constant-curvature section. As each of the robot’s component tubes is independently rotated or extended, the entire device can change shape and trace intricate paths through open air

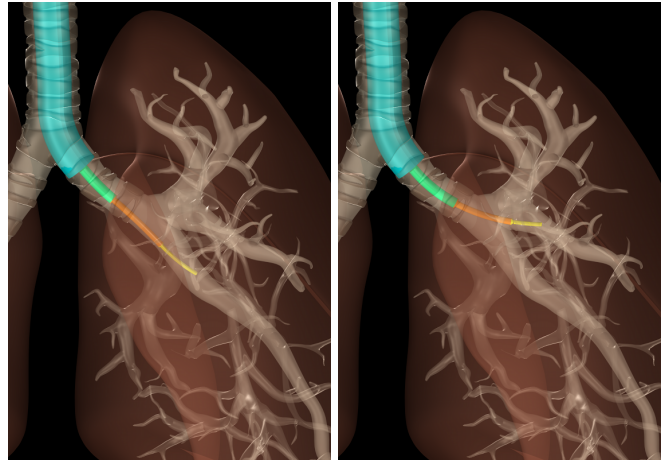


Fig. 1. Simulation of a single concentric tube robot reaching two pre-specified clinical targets in the bronchial tubes of a human lung. The robot is inserted through a bronchoscope (in cyan) and guided toward the specified targets while avoiding contact with the bronchial tube walls. The three component tubes of the robot are colored green, orange, and yellow.

or tissues. The pre-curvatures and lengths of the tubes have a significant impact on the set of clinical targets reachable by the device, so proper selection of design parameters for a patient’s anatomy is critical to the success of a surgical procedure.

Our goal is to computationally optimize the design of a concentric tube robot on a patient- and surgery-specific basis to enable a single robot inserted into the patient to reach multiple clinically relevant sites. Prior to the procedure, we assume that a volumetric image (e.g. CT scan or MRI) is available, from which we can extract the geometry of the anatomical environment through which the robot will navigate, including free space and anatomical obstacles such as nerves, vessels, sensitive organs, and bones.

Designing these robots to perform specific tasks is made challenging by their complex kinematics; unlike most manipulators, concentric tube robots cannot be separated into kinematically independent links because rotating or extending any tube affects the shape of the entire robot. Simplified kinematic models which assume *telescoping dominant stiffness* (meaning that each tube is assumed to be infinitely stiff compared to all within it, and all tubes are assumed to be infinitely torsionally stiff) have been used to mitigate modeling complexity for motion planning and design [17], [2], [1]. While these models’ computational simplicity allow for a good starting point, more advanced models are required to accurately describe these robots’ kinematics [6], [22], as shown in Fig. 2. Model accuracy has a significant impact

L. G. Torres and R. Alterovitz are with the Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA {luis, ron}@cs.unc.edu

R. J. Webster III is with the Department of Mechanical Engineering, Vanderbilt University, Nashville, TN, USA robert.webster@vanderbilt.edu



Fig. 2. Two overlaid time frames of a concentric tube robot. As the inner tube is extended past the outer tube, the inner tube interacts with the outer tube’s curvature, causing the entire robot’s shape to change. This illustrates that the tubes cannot necessarily be treated as kinematically independent during the design and control of these robots.

on the quality of motion plans and designs, as discussed in Sec. VI. Due to the importance of high quality motion plans and designs in minimally invasive surgery, the design process introduced in this paper uses one of the most accurate kinematic models available.

To create designs in a manner that considers the time-varying shape of the robot during deployment, we simultaneously compute both the design of the robot and time-sequences of control inputs that will feasibly guide the robot to the user-specified goal sites. The latter requires solving a motion planning problem from the robot’s start location to the goal sites while avoiding anatomical obstacles. Motion planning for complex robots such as concentric tube robots is PSPACE-hard and computationally intractable to completely solve [21], so like most practical motion planners, we use a sampling-based approach. We build on a previously developed motion planner for concentric tube robots [26], which searches for feasible trajectories in the robot’s configuration space, and augment it to also search for designs in the robot’s design space. By properly blending motion planning and design, we retain one of the key features of sampling based motion planners: probabilistic completeness [13]. Extending probabilistic completeness to design of concentric tube robots, we guarantee that if a task-feasible design exists, as computation time is allowed to increase, the probability of our algorithm finding a task-feasible design for performing a given task approaches 100%.

To accelerate the algorithm, we leverage *design coherence*, the observation that collision-free configuration spaces of robots of similar designs are similar. This coherence implies that information gained about the free configuration space of a robot under one design can be partially re-used for a similar robot design.

We demonstrate our method’s effectiveness for concentric tube robot design by (1) showing that we can successfully design a robot for scenarios where prior concentric tube robot design techniques would fail, and (2) applying our new method to a medically motivated simulated scenario involving navigation through bronchial anatomy to reach several points of clinical interest in the lung.

II. RELATED WORK

The task-oriented design of concentric tube robots was first investigated in the work of Bedell et al. [2] and Anor et al. [1]. They both used an optimization-based approach: they formulate an objective function that codifies a design’s

fitness for performing a particular task, and then optimize this function using a global pattern search. These approaches make the significant assumption of telescoping dominant stiffness, which limits their applicability to designs that satisfy this assumption. With this assumption, a design can be quickly evaluated by considering only its final configuration. The assumption ensures the shape of the device up to the tip does not change when the tip is advanced using a sequential tube insertion strategy: the tubes are first rotated to their final orientations and then deployed such that inner tubes do not protrude from outer tubes until the outer tubes have reached their final insertion lengths [17]. To consider a broader class of concentric tube robots for which this assumption does not hold, motion planning will be required to consider the device’s time-varying shape changes during task performance.

Our approach to blending task-oriented design with motion planning requires that we have an accurate model of the time-varying curve of a concentric tube robot as a task is performed. Early geometric models assumed torsional rigidity [7], [25]; later models considered torsion in straight transmission segments [28], [5], full torsional compliance [6], [23], external loading [22], [16], and frictional effects [15]. The model used in this paper considers torsional effects in all sections of the tube [23]; we do not consider external loading since our designs are intended to avoid collisions with obstacles.

Our design approach also requires a motion planner that can plan sequences of tube rotations and insertions that will maneuver the robot from a start location to a target while avoiding obstacles. Prior motion planning methods have used improved kinematic models as they have been developed: from optimization-based motion planning using a simplified kinematic model [17], [18] to a sampling-based method using the fully torsionally compliant kinematic model [26]. The planner from the latter paper is incorporated into our design method in order to provide probabilistic completeness guarantees and solution accuracy.

Optimization approaches have been applied to the design of other types of robots. A sampling-based approach was applied to optimizing the base location of a robot manipulator [9], but this approach requires an initial collision-free plan. Merlet used interval analysis to find robot designs which fulfill a set of requirements [19]. Methods such as Synergy [11] and Darwin2k [14] rely on genetic algorithms to perform multiple simulations to optimize the structure of a robot for various metrics. This methodology has been applied to the design of manipulators [4] and surgical robots [24]. However, these approaches provide no guarantees on performance. Other methods aim to optimize structural properties of robots for specific problem domains, including geometric methods for planar manipulators [27], and grid-based methods for manipulators [20]. In contrast to these methods, our sampling-based approach is applicable to the complex kinematics of concentric tube robots, accounts for obstacles in the workspace, and provides probabilistic completeness guarantees.

III. PROBLEM DEFINITION

We define the design of a concentric tube robot as a set of parameters that are selected before its use and cannot be changed once it begins to perform a task. Specifically, we define a design d of an n -tube concentric tube robot as a vector of $3n$ real-valued parameters where we associate with each tube the following three parameters:

- L_i^s : length of tube i 's straight transmission section
- L_i^c : length of tube i 's pre-curved section
- κ_i : curvature of tube i 's pre-curved section

We assume that the *design space* $D \subset \mathbb{R}^{3n}$ is a bounded set. We only consider design parameters which do not modify the size or topology of the robot's configuration space, e.g. we do not consider the number of tubes as a design parameter (although this could be addressed by running our method for various numbers of tubes and selecting the best option). We also do not consider pre-curved sections with non-constant curvatures, which will be considered in future work.

Each tube of a concentric tube robot provides two degrees of freedom to the robot, as each tube can be independently rotated or inserted. For an n -tube concentric tube robot with design d , let $Q \subset (\mathbb{S}^1)^n \times \mathbb{R}^n$ be its bounded, $2n$ -dimensional configuration space and let $q \in Q$ denote a configuration. While the configuration space of any robot in the design space is the same, the collision-free subset $Q_{\text{free}}^d \subseteq Q$ of the configuration space will vary based on the robot design d . Given a workspace goal region R we can define for a design d the subspace of configurations Q_{goal}^d which correspond to the robot's end-effector lying in R . Similar to Q_{free}^d , the goal region Q_{goal}^d can vary depending on d (a given design d may not even be able to reach the goal region, meaning $Q_{\text{goal}}^d = \emptyset$).

We will consider the task of reaching a goal region with a concentric tube robot's end-effector while avoiding collisions between the robot and its environment. For this task, we must consider paths in configuration space for particular robot designs. A path in Q is a continuous function $\Pi : [0, s] \rightarrow Q$ where s is the length of the path. A *collision-free path* under a design d is a path which lies entirely in Q_{free}^d . Given a start configuration q_0 and a workspace goal region R , a *feasible path* under a design d is a collision-free path Π which begins at q_0 and ends at a configuration corresponding to the robot reaching R . To simplify method exposition, we assume that q_0 is collision-free for all $d \in D$.

Formally, given R as a workspace goal region, D as the design space of a concentric tube robot, and q_0 as the robot's start configuration, our objective is to compute a *task-feasible design*: a design $d \in D$ for which there exists a feasible path Π from configuration q_0 to a configuration that reaches R .

IV. METHOD FOR TASK-ORIENTED DESIGN

Our task-oriented design method searches for a robot design in design space and simultaneously searches for motion plans in configuration space. The method explores a set $\hat{D} \subseteq D$ of designs. We compute a motion plan for design d in Q_{free}^d using a sampling-based approach based on a rapidly-exploring random tree (RRT) [13], which produces a motion

planning graph that incrementally explores the configuration space. We achieve probabilistic completeness guarantees on finding a task-feasible design by properly interleaving the searches of the design and configuration spaces.

To accelerate the algorithm, we leverage *design coherence*, the observation that collision-free configuration spaces of robots of similar designs are similar. Our design method can be seen as an "RRT of RRTs," where the former RRT is in design space and the latter are in configuration space. We explore design space using an RRT that utilizes configuration space information from nearby design samples to accelerate configuration space exploration at new design samples.

A. Method Inputs

Our method requires as input a specification of the environment geometry (i.e. anatomical obstacles) and the location of the goal region. The environment geometry is implicitly defined by a user-defined function `collision_free(q_1, q_2, d)`, which returns `false` if under design d the path linearly interpolated from configuration q_1 to configuration q_2 collides with an obstacle. We define the goal regions with a function `is_in_goal(q, d)` which returns `true` if the end effector of a robot of design d at configuration q lies within the goal region. The user must also specify the number of tubes and the allowable ranges for each of the design parameters defined in Sec. III.

B. Concentric Tube Robot Design Algorithm

As described in Alg. 1, our method begins by adding a random initial design d_0 to its set of design samples \hat{D} . At each method iteration, we perform one of the following two functions, where the probability of invoking the former is user-specified weight p_{design} :

- `generate_new_design`: Generate a new design space sample d' and use a nearby design's motion planning graph to populate the graph of d' (Alg. 2).
- `explore_prior_design`: Expand the motion planning graph of a previously considered design sample (Alg. 3).

C. Generating New Designs

We associate a distinct motion planning graph with each design sample because the collision-free configuration space of a robot is dependent on its design. For a design d , the sampling-based motion planning graph $G^{d_i} = (V^{d_i}, E^{d_i})$ contains vertices V^{d_i} corresponding to sampled configurations $q_i \in Q$ which were found to be collision-free under design d_i , and edges E representing a collision-free connection between pairs of configurations.

In `generate_new_design`, we add new design space samples as in an RRT: using a Voronoi bias. We generate a random sample $d_{\text{sample}} \in D$ and use `nearest $_D(\hat{D}, d_{\text{sample}})$` to select the configuration $d_{\text{near}} \in \hat{D}$ that is nearest to d_{sample} in design space. We then use `extend $_D(d_{\text{near}}, d_{\text{sample}}, \alpha)$` to generate a new design d_{new} such that d_{new} has a more similar design to d_{sample} than d_{near} . This α value is analogous to RRT's maximum stepsize parameter. We choose α to be

Alg. 1 CTRDesign_main: Compute the free configuration spaces of designs in D .

Input:

q_0 : initial configuration
 p_{design} : weight for exploring design space relative to configuration space at each algorithm iteration
 α : maximum size of extend step in D
 β : maximum size of extend step in Q
 n : number of iterations to perform

Output:

Set of graphs $\{G^{d_i} = (V^{d_i}, E^{d_i})\}$, one graph for each considered design $d_i \in \hat{D}$

```

1   $d_0 \leftarrow$  random design in  $D$ 
2   $\hat{D} \leftarrow \{d_0\}$ ,  $V^{d_0} \leftarrow \{q_0\}$ ,  $E^{d_0} \leftarrow \emptyset$ 
3  for  $i = 1 \dots n$ 
4      if (random number in  $[0, 1] < p_{\text{design}}$ )
5          generate_new_design()
6      else
7          explore_prior_design()

```

Alg. 2 generate_new_design: Generate a new design sample and populate its motion planning graph.

Input:

Global variables $\{V^{d_i}\}, \{E^{d_i}\}, \hat{D}, \alpha, \text{is_checked}$ from CTRDesign_main

Output:

Global variables $\{V^{d_i}\}, \{E^{d_i}\}, \text{is_checked}$, and \hat{D} are updated

```

1   $d_{\text{sample}} \leftarrow$  random design in  $D$ 
2   $d_{\text{near}} \leftarrow$  nearest $_D(\hat{D}, d_{\text{sample}})$ 
3   $d_{\text{new}} \leftarrow$  extend $_D(d_{\text{near}}, d_{\text{sample}}, \alpha)$ 
4   $\hat{D} \leftarrow \hat{D} \cup \{d_{\text{new}}\}$ 
5   $V^{d_{\text{new}}} \leftarrow V^{d_{\text{near}}}$ ,  $E^{d_{\text{new}}} \leftarrow E^{d_{\text{near}}}$ 
6  for each  $(q_1, q_2) \in E^{d_{\text{new}}}$ 
7      is_checked $[d_{\text{new}}, q_1, q_2] \leftarrow$  false

```

large enough to gain new and useful information about D , but small enough so that we can exploit design coherence between d_{near} and d_{new} . We add d_{new} to \hat{D} .

We then utilize design coherence to populate the empty motion planning graph of d_{new} with information from d_{near} . Because d_{new} , by construction, is a similar design to d_{near} , we can expect many of the vertices and edges in $G^{d_{\text{near}}}$ to be valid in $G^{d_{\text{new}}}$. So, we optimistically copy *all* of $G^{d_{\text{near}}}$ into $G^{d_{\text{new}}}$. Instead of collision-checking all of the edges under the new design, we take a lazy approach: we tag them all as “unchecked” edges in d_{new} using the hash structure `is_checked`. We only collision check an unchecked edge if it is expanded upon in `explore_prior_design`, thus speeding exploration while reducing unnecessary checks.

Alg. 3 explore_prior_design: Explore motion planning graphs of a previously considered design.

Input:

Global variables $\{V^{d_i}\}, \{E^{d_i}\}, \hat{D}, \beta$, `is_checked` from CTRDesign_main

Output:

Global variables $\{V^{d_i}\}, \{E^{d_i}\}, \hat{D}, \text{is_checked}$ are updated

```

1   $d \leftarrow$  random design in  $\hat{D}$ 
2   $q_{\text{sample}} \leftarrow$  random sample from  $Q^d$ 
3   $q_{\text{near}} \leftarrow$  nearest $_Q(V^d, q_{\text{sample}})$ 
4   $q_{\text{new}} \leftarrow$  extend $_Q(q_{\text{near}}, q_{\text{sample}}, \beta)$ 
5   $V^d \leftarrow V^d \cup \{q_{\text{new}}\}$ ,  $E^d \leftarrow E^d \cup \{(q_{\text{near}}, q_{\text{new}})\}$ 
6  for each edge  $(q_1, q_2)$  on path from  $q_0$  to  $q_{\text{new}}$ 
    where is_checked $[d, q_1, q_2] =$  false
7      if collision_free $(q_1, q_2, d)$ 
8          is_checked $[d, q_1, q_2] \leftarrow$  true
9      else
10         Delete subtree of  $(V^d, E^d)$  rooted at  $q_2$ 

```

D. Exploring Previously Considered Designs

In `explore_prior_design`, we gain more information about the free configuration spaces of the set of designs \hat{D} we have already considered. First, we uniformly sample a design d from the set of previously considered designs \hat{D} . Next, we expand the motion planning graph of d . This would be identical to RRT expansion except that there may be edges in G^d which are “unchecked” due to the lazy heuristic. So, we begin by adding a new configuration q_{new} (generated by `extend $_Q$`) to G^d without collision-checking the new edge. Then, we ensure that there exists a collision-free path from q_0 to q_{new} by collision-checking any unchecked edges between q_0 and q_{new} in G^d . If the path is found to be collision-free, q_{new} remains in the tree. If not, we delete the entire subtree of configurations rooted at the first in-collision configuration found when checking the path from q_0 . In this way we combine configuration space exploration with validation of the information we imported from neighboring designs’ motion planning graphs.

V. METHOD ANALYSIS

A. Probabilistic Completeness

We now prove *probabilistic completeness for design*: as the number of iterations $n \rightarrow \infty$, the probability of our design method finding a design which feasibly reaches a goal region R , given one exists, approaches one. We define $p[\text{returns } d \in D_{\text{goal}}]$ as the probability that the method returns a robot design $d \in D_{\text{goal}}$, where $D_{\text{goal}} \subseteq D$ is the set of designs capable of reaching a point in the workspace goal region R . We assume the set D_{goal} is a measurable set.

To simplify analysis, we consider a weaker form of the method which does not use design coherence and performs one `generate_new_design` step and one `explore_prior_design` step at each iteration. These simplifications do not modify the algorithm’s asymptotic behavior.

To show probabilistic completeness, we must show that as $n \rightarrow \infty$: (1) a design will be sampled in D_{goal} , and (2) the motion planning graph for a design $d \in D_{\text{goal}}$ will explore the configuration space sufficiently to determine that d reaches the goal region R . To prove the former point, we use the fact that, in the limit, our RRT-like sampling approach in design space will approximate a uniform distribution. Since D_{goal} is a measurable set, we guarantee that some $d \in D_{\text{goal}}$ will be sampled. To prove the latter point, we use the fact that we are building an RRT for design d in d 's configuration space. To guarantee that the RRT will find a path to a point in R , we require the number of samples in this RRT to approach infinity, which is non-trivial since our method samples in both design space and in the configuration space of each design. For a design d_i created in iteration i , the number of configuration samples $|V^{d_i}|$ created as $n \rightarrow \infty$ is:

$$\lim_{n \rightarrow \infty} E[|V^{d_i}|] = \sum_{j=i}^{\infty} E \left[\begin{array}{l} \# \text{ of samples added} \\ \text{to } d_i \text{ at iteration } j \end{array} \right] = \sum_{j=i}^{\infty} \frac{1}{j}.$$

This is a harmonic series whose limit approaches ∞ . Hence, the configuration space of a design created at iteration i of our method is fully explored, proving probabilistic completeness of our design method for concentric tube robots.

B. Computational Complexity

For the simplified version of the design method described in Sec. V-A, the dominating asymptotic factors are the nearest neighbor searches. When implemented efficiently, these calls are logarithmic in complexity to the number of elements searched over [10]. At iteration i , our method will have created at most i samples, so the complexity of iteration i of the simplified method is $O(\log i)$. We note that our integrated design and motion planning method is no worse asymptotically than RRT for motion planning alone. The heuristics in our full method add computational complexity: copying configuration samples from one motion planning graph to another of a similar design requires, in the worst case, $O(i)$ time at iteration i . Therefore, the complexity of iteration i of our full method is $O(i)$.

Although our full method is asymptotically slower per iteration than the simplified method, it adds far more nodes to the designs' motion planning graphs than the simplified method and therefore comes to solutions in fewer iterations. This results in a significant speed-up: averaged over 40 executions in the twisted half torus environment described in Sec. VI-A, the full method computes task-feasible designs approximately 94% faster than the simplified method, and 20% faster than a method with the same p_{design} -based exploration but without design coherence.

VI. RESULTS

We experimentally demonstrated the effectiveness of our concentric tube robot design method by considering two tasks: (1) navigating through a twisted half-torus, and (2) reaching two specified points of interest in the bronchial anatomy, where in both cases we wish to avoid collisions between the robot and the surrounding environment.

A. Comparison with Prior Methods

In our first experiment, we consider a 2-tube concentric tube robot maneuvering through a simple tubular environment as shown in Fig. 3. The environment consists of a half-torus for which one half is rotated by 45 degrees such that the environment is non-planar. The robot starts at the center of the proximal end of the environment and the goal is to reach a point at the distal end of the environment.

We applied four different design methodologies to the problem of designing a robot to navigate this environment:

- 1) *Dominant stiffness with sequential insertion*: We analytically derive a design d_{dom} and configuration q_{seq} for which q_{seq} is collision free and reaches the goal under the assumption of telescoping dominant stiffness using the insertion strategy described in Sec. II.
- 2) *Dominant stiffness with motion planning*: Given d_{dom} and q_{seq} from above, we execute a motion planner that considers an accurate mechanics-based model of the robot kinematics in order to compute a plan to reach the goal.
- 3) *Our method with sequential insertion*: We use our proposed method which computes a design d^* that can navigate the environment without the assumption of telescoping dominant stiffness, but perform the task using the sequential insertion strategy.
- 4) *Our method with motion planning*: We use our proposed method to compute a design d^* and a motion plan that can navigate the environment.

In this simple environment we can analytically derive a design d_{dom} which would allow the robot to navigate the passage under the assumption of telescoping dominant stiffness: both tubes would have radii of curvature κ_i equal to that of the torus from which the environment was generated and the curved section lengths would each have to be at least as long as half the environment's center-line length. A collision-free configuration q_{seq} which would reach the goal can also be analytically defined: have the outer tube's plane of curvature coincide with that of the first portion of the torus, and the inner tube's plane of curvature coincide with that of the rotated portion. The insertion length of the outer tube would be the length of the first part of the environment and the insertion length of the inner tube would be the length of the entire passage.

Under the telescoping dominant stiffness assumption, a concentric tube robot of design d_{dom} should be able to reach the goal while avoiding collisions by following a sequential insertion strategy ending in configuration q_{seq} . To best approximate a real-world concentric tube robot which may exhibit the telescoping dominant stiffness property, we simulated a robot with both the thickest outer tube and the thinnest inner tube found in prior literature [6], [23], with the inner tube having an inner radius large enough to still allow for passing tools through. We chose the inner tube's inner and outer radii to be 0.824 mm and 0.924 mm respectively (for a thickness of 0.1 mm), and the outer tube's inner and outer radii to be 0.925 mm and 1.1175 mm respectively (for

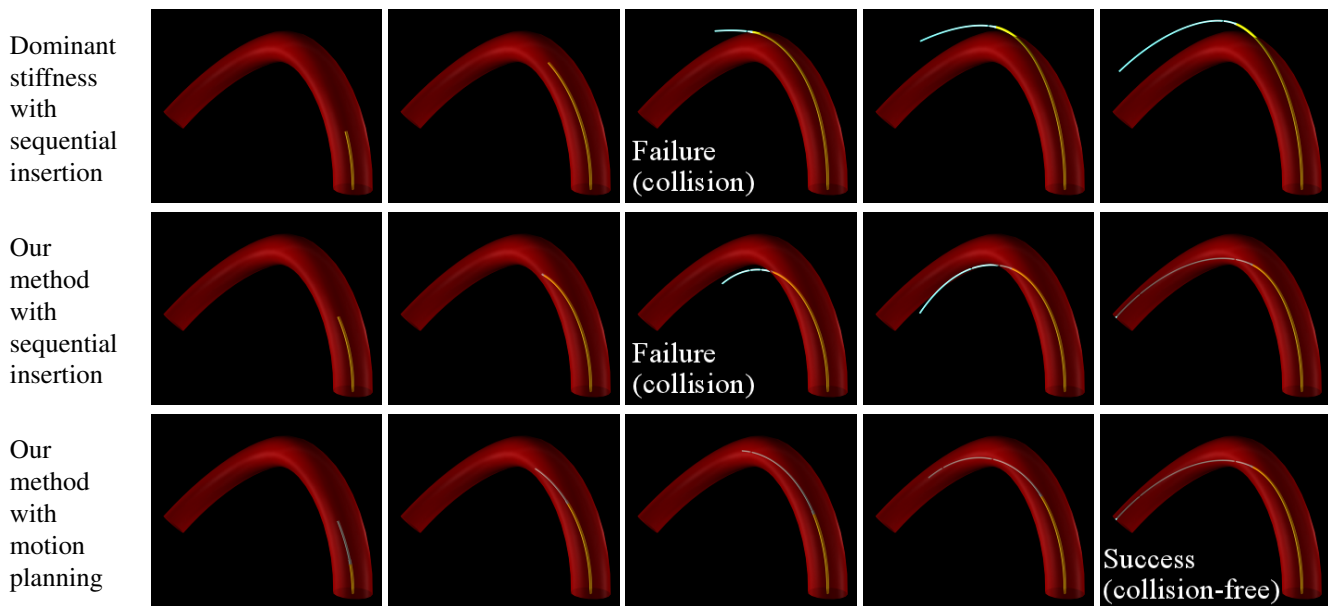


Fig. 3. Three mechanically accurate simulations of the insertion of a 2-tube concentric tube robot through a red twisted torus environment with varying designs and insertion strategies. The robot’s outer tube is pictured in yellow and the inner tube in light blue. The top row shows a robot under design d_{dom} which collides with the environment because it was designed and inserted under the assumption of telescoping dominant stiffness, which does not hold under a mechanically accurate kinematic model of concentric tube robots. When motion planning was applied to this design, a collision-free path through the tube under design d_{dom} could not be found (not pictured). The middle row shows a robot design d^* computed by our design method, but it is inserted with a sequential insertion strategy which does not require motion planning; when simulated under realistic robot kinematics, the insertion collides with the environment and fails. The bottom row shows a robot of our computed design d^* successfully navigating the environment without collisions because it combines an accurate kinematic model of concentric tube robots with motion planning to enable collision-free performance of the task.

a thickness of 0.25 mm). In order to realistically simulate the robot’s behavior, we use a highly accurate, mechanics-based kinematic model of the robot [23].

Under design d_{dom} and using a sequential insertion strategy ending at q_{seq} , the robot failed to navigate the passage as shown in Fig. 3 (top row). We also ran a motion planner [26] to see if there existed some other motion plan which, under design d_{dom} , could still navigate the passage. The planner failed to find such a plan, even when allowed 10 hours of computation time. A design deemed task-feasible under the telescoping dominant stiffness assumption is unlikely to fully avoid the obstacle wall for real-world concentric tube robots.

Next, we applied our new design method to this problem, which incorporates motion planning under the realistic, torsionally compliant kinematic model. Using $p_{\text{design}} = 0.0001$, we found a robotic design d^* which allows for collision-free navigation through the environment when simulated with the realistic kinematic model. The inner and outer tube curvatures of d^* were 0.00525mm^{-1} and 0.00885mm^{-1} , respectively; the inner and outer tube curved segment lengths were 351mm and 205mm; the inner and outer tube straight segment lengths were 659mm and 240mm. Averaged over 20 executions, the method required 24 minutes on a 2.40 GHz Intel®Xeon Quad-Core PC with 12 GB RAM. The design d^* and collision-free path are shown in Fig. 3 (bottom row).

We also simulated a sequential insertion of the robot under design d^* that ended in the final configuration of the collision-free path yielded by our method. This failed to navigate the passage as shown in Fig. 3 (middle row). This demonstrates that motion planning is indeed vital to the design process of concentric tube robots under realistic

kinematics.

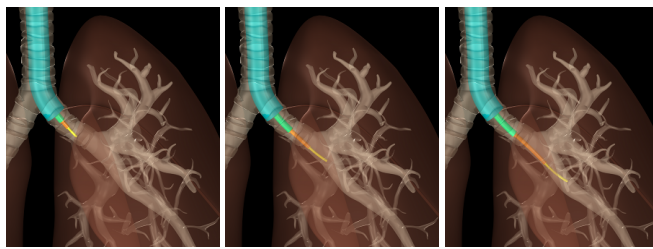
B. Bronchial Anatomy Scenario

Biopsy is required for a definitive diagnosis of lung cancer. However, many sites within the lung currently cannot be safely accessed for biopsy without highly invasive procedures [12], which makes early definitive diagnosis and treatment of lung cancer impossible for many patients who cannot tolerate highly invasive procedures. A properly designed concentric tube robot may have the dexterity to safely reach sites deep within the lung that currently available medical instruments cannot.

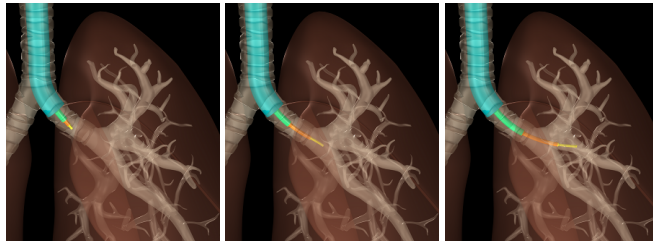
We apply our presented method to design a 3-tube concentric tube robot that can access two sites in distinct bronchi without damaging (e.g. colliding with or piercing) the walls of the bronchial tubes. Irregularity of the bronchial tubes’ shapes combined with very narrow passageways make this a difficult problem. A simulation of one of the resulting designs is pictured in Fig. 4 with two configurations reaching the specified points with the robot’s end-effector. Averaged over 50 runs, our method took 174 seconds to compute designs which can reach these two points without colliding with the surrounding environment. Since pre-operative CT scans are typically obtained at least a day before an actual biopsy, this is a clinically acceptable computation time.

VII. CONCLUSION

We developed a task-oriented design method for concentric tube robots which combines a search of the robot’s design space and sampling-based motion planning of the robot’s configuration space in order to find a task-feasible design for performing a given task without colliding with anatomical



(a) Target 1 in Lung



(b) Target 2 in Lung

Fig. 4. Sequential snapshots of virtual simulations of two concentric tube robot motion plans. Both simulations are of one robot design computed by our design method in order to navigate to two specified points in narrow bronchial anatomy without colliding with the bronchial walls.

obstacles. Our method relaxes assumptions made in prior work in order to consider a broader class of concentric tube robots and generalizes probabilistic completeness to design space.

In future work, we will extend the scope of our design method to other modular robots. Our method is easily generalizable because its formulation makes almost no assumptions about the robot's kinematics. Specifically concerning concentric tube robots, we plan to consider cases in which the robot is allowed to make contact with some tissues subject to constraints on forces. We also will evaluate our computed designs on concentric tube robots operating in tissue phantoms.

VIII. ACKNOWLEDGMENT

The authors thank Sachin Patil, Chris Bowen, and Philip J. Swaney for their insights regarding concentric tube robots and their design, and Nathan Dierk for creating visualizations of these devices. This research was supported in part by the National Science Foundation (NSF) through an NSF Graduate Research Fellowship and awards #IIS-0905344 and #IIS-1149965 and by the National Institutes of Health (NIH) under award #R21EB011628.

REFERENCES

- [1] T. Anor, J. R. Madsen, and P. E. Dupont, "Algorithms for design of continuum robots using the concentric tubes approach: A neurosurgical example," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2011, pp. 667–673.
- [2] C. Bedell, J. Lock, A. Gosline, and P. E. Dupont, "Design optimization of concentric tube robots based on task and anatomical constraints," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2011, pp. 398–403.
- [3] J. Burgner, P. J. Swaney, D. C. Rucker, H. B. Gilbert, S. T. Nill, P. T. Russell III, K. D. Weaver, and R. J. Webster III, "A bimanual teleoperated system for endonasal skull base surgery," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Sept. 2011, pp. 2517–2523.
- [4] O. Chocron, "Evolutionary design of modular robotic arms," *Robotica*, vol. 26, no. 3, pp. 323–330, 2008.

- [5] P. E. Dupont, J. Lock, and E. Butler, "Torsional kinematic model for concentric tube robots," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2009, pp. 3851–3858.
- [6] P. E. Dupont, J. Lock, B. Itkowitz, and E. Butler, "Design and control of concentric-tube robots," *IEEE Trans. Robotics*, vol. 26, no. 2, pp. 209–225, Apr. 2010.
- [7] J. Furusho, T. Katsuragi, T. Kikuchi, T. Suzuki, H. Tanaka, Y. Chiba, and H. Horio, "Curved multi-tube systems for fetal blood sampling and treatments of organs like brain and breast," *J. Comput. Assist. Radiol. Surg.*, vol. 1, pp. 223–226, 2006.
- [8] M. Heverly, P. E. Dupont, and J. Friedman, "Trajectory optimization for dynamic needle insertion," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Apr. 2005, pp. 1646–1651.
- [9] D. Hsu, J.-C. Latombe, and S. Sorkin, "Placing a robot manipulator amid obstacles for optimized execution," in *Proc. IEEE Int. Symp. Assembly and Task Planning*, July 1999, pp. 280–285.
- [10] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robotics Research*, vol. 30, no. 7, pp. 846–894, June 2011.
- [11] L. K. Katragadda, "A language and framework for robotic design," Ph.D. dissertation, Carnegie Mellon University, 1997.
- [12] G. Krishna and M. K. Gould, "Minimally invasive techniques for the diagnosis of peripheral pulmonary nodules," *Current Opinion in Pulmonary Medicine*, vol. 14, pp. 282–286, 2008.
- [13] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006.
- [14] C. Leger, "Automated synthesis and optimization of robot configurations: An evolutionary approach," Ph.D. dissertation, Carnegie Mellon University, 1999.
- [15] J. Lock and P. E. Dupont, "Friction modeling in concentric tube robots," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2011, pp. 1139–1146.
- [16] J. Lock, G. Laing, M. Mahvash, and P. E. Dupont, "Quasistatic modeling of concentric tube robots with external loads," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Oct. 2010, pp. 2325–2332.
- [17] L. A. Lyons, R. J. Webster III, and R. Alterovitz, "Motion planning for active cannulas," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Oct. 2009, pp. 801–806.
- [18] —, "Planning active cannula configurations through tubular anatomy," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, May 2010, pp. 2082–2087.
- [19] J.-P. Merlet, "Optimal design of robots," in *Proc. Robotics: Science and Systems*, 2005.
- [20] J. Park, P. Chang, and J. Yang, "Task-oriented design of robot kinematics using the grid method," *J. Advanced Robotics*, vol. 17, no. 9, pp. 879–907, 2003.
- [21] J. H. Reif, "Complexity of the mover's problem and generalizations," in *20th Annual IEEE Symp. on Foundations of Computer Science*, Oct. 1979, pp. 421–427.
- [22] D. C. Rucker, B. A. Jones, and R. J. Webster III, "A geometrically exact model for externally loaded concentric-tube continuum robots," *IEEE Trans. Robotics*, vol. 26, no. 5, pp. 769–780, Jan. 2010.
- [23] D. C. Rucker and R. J. Webster III, "Parsimonious evaluation of concentric-tube continuum robot equilibrium conformation," *IEEE Trans. Biomedical Engineering*, vol. 56, no. 9, pp. 2308–2311, Sept. 2009.
- [24] D. Salle, P. Bidaud, and G. Morel, "Optimal design of high dexterity modular MIS instrument for coronary artery bypass grafting," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Apr. 2004, pp. 1276–1281.
- [25] P. Sears and P. E. Dupont, "A steerable needle technology using curved concentric tubes," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Oct. 2006, pp. 2850–2856.
- [26] L. G. Torres and R. Alterovitz, "Motion planning for concentric tube robots using mechanics-based models," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Sept. 2011, pp. 5153–5159.
- [27] R. Vijaykumar, K. J. Waldron, and M. J. Tsai, "Geometric optimization of serial chain manipulator structures for working volume and dexterity," *Int. J. Robotics Research*, vol. 5, no. 2, pp. 91–103, 1986.
- [28] R. J. Webster III, A. M. Okamura, and N. J. Cowan, "Toward active cannulas: Miniature snake-like surgical robots," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Oct. 2006, pp. 2857–2863.