



Published in final edited form as:

IEEE Int Conf Robot Autom. 2015 May ; 2015: 2361–2367. doi:10.1109/ICRA.2015.7139513.

## A Motion Planning Approach to Automatic Obstacle Avoidance during Concentric Tube Robot Teleoperation

**Luis G. Torres,**

Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, USA

**Alan Kuntz,**

Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, USA

**Hunter B. Gilbert,**

Department of Mechanical Engineering, Vanderbilt University, Nashville, TN 37235, USA

**Philip J. Swaney,**

Department of Mechanical Engineering, Vanderbilt University, Nashville, TN 37235, USA

**Richard J. Hendrick,**

Department of Mechanical Engineering, Vanderbilt University, Nashville, TN 37235, USA

**Robert J. Webster III, and**

Department of Mechanical Engineering, Vanderbilt University, Nashville, TN 37235, USA

**Ron Alterovitz**

Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, USA

Luis G. Torres: luis@cs.unc.edu; Alan Kuntz: adkuntz@cs.unc.edu; Ron Alterovitz: ron@cs.unc.edu

### Abstract

Concentric tube robots are thin, tentacle-like devices that can move along curved paths and can potentially enable new, less invasive surgical procedures. Safe and effective operation of this type of robot requires that the robot's shaft avoid sensitive anatomical structures (e.g., critical vessels and organs) while the surgeon teleoperates the robot's tip. However, the robot's unintuitive kinematics makes it difficult for a human user to manually ensure obstacle avoidance along the entire tentacle-like shape of the robot's shaft. We present a motion planning approach for concentric tube robot teleoperation that enables the robot to interactively maneuver its tip to points selected by a user while automatically avoiding obstacles along its shaft. We achieve automatic collision avoidance by precomputing a roadmap of collision-free robot configurations based on a description of the anatomical obstacles, which are attainable via volumetric medical imaging. We also mitigate the effects of kinematic modeling error in reaching the goal positions by adjusting motions based on robot tip position sensing. We evaluate our motion planner on a teleoperated concentric tube robot and demonstrate its obstacle avoidance and accuracy in environments with tubular obstacles.

## I. Introduction

Tentacle-like surgical devices called concentric tube robots have the potential to enable new minimally invasive medical procedures in the human body, including the heart, the lungs, and the skull base, among other applications [1]. These robots are composed of thin, pre-curved, elastic, concentric tubes. These tubes can be telescopically inserted and rotated relative to one another, causing the entire robot's shape to change, enabling the device to curve around anatomical obstacles to reach anatomical sites inaccessible to traditional, straight instruments (see Fig. 1).

During operation, care must be taken to ensure that the robot does not collide with sensitive anatomical obstacles, e.g., blood vessels, critical nerves, and sensitive organs. Physicians can attempt to manually avoid obstacles while steering the robot's tip using position control [2], [3], [4], but some tip motions can require dramatic changes in the robot's shape and therefore cause hard-to-predict collisions. Furthermore, requiring a physician to guarantee collision avoidance along the entire tentacle-like shape of the device would require considerable mental load on the physician.

In order to tackle this problem, we developed a motion planning approach for teleoperation of concentric tube robots that frees the physician to focus on controlling the robot's tip, while our integrated motion planner automatically handles obstacle avoidance along the robot's entire shaft. Motion planning can harness these robots' redundant degrees of freedom in order to curve around obstacles and still reach desired tip positions.

The user specifies goal positions by clicking with a 3D mouse on an augmented reality interface, and the concentric tube robot automatically moves its tip to the specified goal position while avoiding contact with surrounding obstacles. Our approach also reduces the error between the commanded goal position and the robot's tip position by using tip position sensing to account for kinematic modeling error.

We demonstrate our motion planner's ability to accurately steer concentric tube robots while avoiding obstacles in physical experiments. The environments in the experiments include tubular obstacles and are inspired by the need to avoid critical vessels during surgical procedures. To the best of our knowledge, this is the first demonstration of a teleoperated concentric tube robot with automatic obstacle avoidance.

## II. Related Work

Automatic avoidance of obstacles requires accurate estimation of the concentric tube robot's shape before executing a motion. Kinematic models for predicting the shape of concentric tube robots have rapidly improved in accuracy and sophistication in recent years, with models considering bending [5], torsional compliance [6], [7], external loading [8], [9], and friction [10]. Our motion planner uses a mechanically accurate model developed by Rucker et al. [3].

Much of the prior research in concentric tube robot control focuses on quickly computing instantaneous velocities for the robot's actuators to achieve a specified tip motion. Xu et al.

and Rucker et al. quickly compute the manipulator Jacobian to achieve fast tip position control [4], [3]. Dupont et al. achieve fast tip position control by evaluating the inverse kinematics of a Fourier series approximation of the robot's kinematics [2]. We build on this prior work by adding a motion planner that enables automatic obstacle avoidance while moving the tip toward a specified position.

Prior work on automatic obstacle avoidance for concentric tube robots has introduced algorithms with varying speed and kinematic accuracy tradeoffs. Algorithms include fast motion planning by assuming simplified kinematics [11], [12], noninteractive motion planning using accurate kinematic models [13], and accurate, interactive-rate motion planning via precomputation [14]. Motion planning algorithms have helped to identify stable configurations [15]. This paper's motion planning approach builds on work by Torres et al. [14]. Whereas prior motion planners for obstacle avoidance have only been evaluated in simulation, here we integrated motion planning into concentric tube robot teleoperation, incorporated position sensing to improve the accuracy of the tip motion, and demonstrated motion planner performance by conducting physical experiments.

Our motion planner for concentric tube robot teleoperation achieves its interactive speed by executing motion planning queries on a precomputed roadmap of collision-free robot configurations. We use a sampling-based motion planning approach, which is well suited for robots with many degrees of freedom and complex kinematics [16]. Our motion planner uses the rapidly-exploring random graph (RRG) algorithm [17]. We use RRG instead of a canonical multi-query planner like the probabilistic roadmap (PRM) [16] because RRG only considers configurations reachable from the robot's insertion point. As an alternative to sampling-based approaches, recently developed optimization-based motion planners [18], [19], [20] have been shown to achieve high execution speeds for many types of robotic manipulators. These methods require many online evaluations of the robot's kinematic model, which is infeasible for concentric tube robots due to these robots' computationally expensive kinematic model.

### III. Overview of Approach

We provide in Fig. 2 an overview of our motion planning approach for concentric tube robot teleoperation. Before operation, we require a geometrical specification of the anatomical obstacles that must be avoided by the concentric tube robot during the procedure. The obstacles  $O$  are specified using polygonal meshes, which can be obtained by segmenting volumetric medical images (e.g., CT scans) either manually or using automatic segmentation software [21].

During operation, the user repeatedly specifies a new goal position  $\mathbf{x}_{\text{goal}}$  for the concentric tube robot to reach. The robot immediately responds by moving its tip to reach the latest given goal position while automatically avoiding contact with obstacles. The user specifies the goal positions with a 3D mouse in an augmented reality display that shows a cursor overlaid on a video feed of the surgical site (e.g., from an endoscopic camera).

## A. Concentric Tube Robot Kinematics

A concentric tube robot includes  $N$  telescoping tubes. Each tube  $i$  consists of a straight segment of length  $L_i^s$  followed by a pre-curved portion of length  $L_i^c$  and radius of curvature  $r_i$ . The device is inserted at a point  $\mathbf{x}_{\text{start}}$  along a vector  $\mathbf{v}_{\text{start}}$ .

Each tube may be (1) inserted or retracted from the previous tube, and (2) axially rotated, yielding 2 degrees of freedom per tube. Hence, the robot's configuration is a  $2N$  dimensional vector  $\mathbf{q} = (\theta_i, \beta_i : i = 1, \dots, N)$  where  $\theta_i$  is the axial angle at the base of the  $i$ 'th tube and  $\beta_i$  is the arc-length position of the base of tube  $i$  behind the robot insertion point (i.e.,  $\beta = 0$  corresponds to  $\mathbf{x}_{\text{start}}$ ). The configuration space is  $Q = (S^1)^N \times \mathbb{R}^N$ .

For a given configuration  $\mathbf{q} \in Q$ , we represent the device's shape as a 3D space curve  $\mathbf{x}(\mathbf{q}, s) : Q \times [0, 1] \mapsto \mathbb{R}^3$  where  $\mathbf{x}(\mathbf{q}, 0) = \mathbf{x}_{\text{start}}$  and  $\mathbf{x}(\mathbf{q}, 1)$  is the robot's tip position.

## B. Intraoperative Execution

Each time the user provides a new  $\mathbf{x}_{\text{goal}}$ , the robot's motion planner computes a collision-free sequence of configurations to guide the robot to the vicinity of  $\mathbf{x}_{\text{goal}}$ , and then performs a tip error correction step guided by sensed measurements of the tip position. These steps make up a motion plan, denoted by a sequence of configurations  $\Pi = (\mathbf{q}_1, \dots, \mathbf{q}_n)$ , to move the robot from its current configuration  $\mathbf{q}_1$  to a new configuration  $\mathbf{q}_n$  such that

1. the distance between the robot's tip  $\mathbf{x}(\mathbf{q}_n, 1)$  and the goal  $\mathbf{x}_{\text{goal}}$  is minimized,
2. each configuration  $\mathbf{q}_i$  of the motion plan  $\Pi$  is free of contact with obstacles, and
3. each configuration  $\mathbf{q}_i$  satisfies the robot's kinematic constraints, which includes limits on tube translations.

To enable interactivity, each plan  $\Pi$  should be computed sufficiently fast.

Due to the unmodeled effects such as friction and tube tolerances, there will be some difference between mechanics-based model predictions and the robot's actual shape. Thus, we consider the model's predictions to be an estimate  $\hat{\mathbf{x}}$  of the robot's true shape  $\mathbf{x}$ . Accurately reaching desired goal positions while ensuring collision avoidance becomes especially challenging when we only have an approximation  $\hat{\mathbf{x}}$  of the robot's true shape. We describe in Sec. IV how we select motion plans that encourage collision avoidance and use sensing to correct tip error.

## IV. Methods

Our motion planning approach for concentric tube robot teleoperation consists of the following components.

- a user interface for specifying desired goal points for the robot to reach.
- an integrated motion planner that computes a path that moves the robot as close as possible to the goal while avoiding collisions. The planner has two parts:
  - a roadmap-based planner that generates collision-free paths to the goal's vicinity.

- a tip error correction (TEC) that senses the robot's tip position and moves the robot's tip closer to the goal.
- a physical concentric tube robot actuation unit that executes the commands computed by the motion planner.

### A. Providing User Input

The user specifies goal positions for the robot by pointing and clicking with a Geomagic® Touch™ X [22]. We currently do not use the Touch™ X's haptic rendering features. To facilitate selection of goal positions, we created an augmented reality display that shows a video feed of the concentric tube robot with an overlaid 3D cursor as the intended goal. We registered the video camera view with the robot's frame of reference using the OpenCV computer vision library [23]. In clinical settings, we expect to use an endoscopic video camera attached to the robot in a manner such that this registration can be pre-calibrated. The user moves the 3D cursor with the handheld stylus of the Touch™ X, and clicking the stylus triggers the motion planner to move the robot's tip to the specified goal position. The augmented reality interface is shown in Fig. 3.

### B. Collision-free Roadmap Planner

We use a motion planner that can generate collision-free paths (as approximated by the kinematic model) for concentric tube robots at interactive rates [14]. The motion planner achieves its speed by separating planning into two phases: a preoperative phase and an intraoperative phase.

**1) Preoperative Phase**—The inputs to the preoperative phase of motion planning are:

- anatomical obstacles represented as 3D polygonal surface meshes.
- physical specifications of the concentric tube robot.
- insertion pose of the concentric tube robot.

The output of the preoperative phase is a discretization of the collision-free subset of the robot's configuration space  $Q$  in the form of a *roadmap*. The roadmap stores a set of configurations and motions between these configurations that have all been verified to be collision-free, according to the kinematic model  $\mathbf{x}^{\wedge}$ . The motion connecting two adjacent configurations in the roadmap is assumed to be linear interpolation. Paths in the roadmap are represented as sequences of adjacent configurations, which means the roadmap generates piecewise linear motions in configuration space.

We compute the collision-free roadmap using a *rapidly-exploring random graph* [17], or RRG. RRG focuses roadmap construction only on regions of the configuration space that are reachable from the robot's insertion pose. RRG also stores multiple alternative paths between configurations, thus allowing us to select paths that meet application-specific criteria.

The RRG algorithm requires a predicate  $\text{is\_collision\_free}(\mathbf{q})$  that evaluates whether the robot satisfies kinematic constraints and is collision-free at configuration  $\mathbf{q}$ . We compute

this predicate by generating a 3D polygonal mesh on-the-fly of our kinematic model  $\mathbf{x}(\mathbf{q}, s)$  and using the Flexible Collision Library [24] to check for collisions between the robot's mesh and the anatomical obstacle meshes. For this work we assume the obstacles are described in the robot's coordinate frame, but in future work and for clinical applications, we will need to register to the preoperative volumetric image coordinate system.

We note that the function  $\text{is\_collision\_free}(\mathbf{q})$  is approximate because it uses the kinematic model  $\mathbf{x}$ . In order to reduce false negatives in collision detection due to modeling error, we add 1 mm of padding to the anatomical obstacle meshes used in computing  $\text{is\_collision\_free}(\mathbf{q})$ .

**2) Intraoperative Phase**—In the intraoperative phase, the motion planner performs the following steps each time the user specifies a new desired goal position  $\mathbf{x}_{\text{goal}}$ .

1. Find the configuration  $\mathbf{q}_{\text{near\_goal}}$  in the roadmap with the minimum predicted tip distance to the goal.
2. Given that the concentric tube robot is currently at configuration  $\mathbf{q}_{\text{start}}$ , find the configuration  $\mathbf{q}_{\text{near\_start}}$  in the roadmap that is nearest to  $\mathbf{q}_{\text{start}}$ .
3. Use Dijkstra's graph search algorithm to find the optimal sequence of configurations  $\Pi^*$  on the roadmap from  $\mathbf{q}_{\text{near\_start}}$  to  $\mathbf{q}_{\text{near\_goal}}$ . We will describe below how we define edge costs.
4. Send  $\Pi^*$  to the robot for execution.

As previously noted, the roadmap stores multiple alternative collision-free paths between configurations. In prior work [14], we selected paths that resulted in smooth robot motion (in simulation) by choosing paths that minimized the total distance traveled by the robot's tip. However, this metric does not consider anatomical obstacles. Paths that optimize this metric can lead the robot to closely trace the boundaries of anatomical obstacles. Although in simulation these paths are considered collision-free, kinematic modeling error can cause unpredicted collisions when the real robot executes these paths.

In this work we prioritize collision avoidance over smooth tip motion. Therefore, we select paths based on a metric that considers the robot's *clearance* from anatomical obstacles. Given a function  $\text{clear}(\mathbf{q})$  that computes the minimum distance between the robot at configuration  $\mathbf{q}$  and the anatomical obstacles, we define the *cost* of the motion connecting configurations  $\mathbf{q}_1$  and  $\mathbf{q}_2$  as

$$c(\mathbf{q}_1, \mathbf{q}_2) = \|\mathbf{q}_1 - \mathbf{q}_2\| \int_0^1 \frac{1}{\text{clear}(\mathbf{q}_1 + t(\mathbf{q}_2 - \mathbf{q}_1))} dt. \quad (1)$$

This function integrates  $(1/\text{clear}(\mathbf{q}))$  along the linear interpolation between  $\mathbf{q}_1$  and  $\mathbf{q}_2$ . We use the reciprocal of the clear function because we want to incur *higher* cost on a path with *smaller* robot-obstacle clearance. We use the Flexible Collision Library [24] to compute clear. In our implementation we compute the integral in Eq. 1 using the trapezoid rule.

Dijkstra's shortest path algorithm uses the cost function in Eq. 1 in order to return paths on the roadmap that tend to steer farther away from anatomical obstacles. We cache these motion costs during precomputation of the collision-free roadmap in order to avoid online evaluation of the expensive clear function.

### C. Tip Error Correction using Tip Position Measurements

In the prior section we explained how the planner uses a precomputed roadmap to take a (predicted-to-be) collision-free path to a configuration in the roadmap  $\mathbf{q}_{\text{near\_goal}}$ . However,  $\mathbf{x}(\mathbf{q}_{\text{near\_goal}}, 1) \neq \mathbf{x}_{\text{goal}}$  in general for two reasons:

1. the roadmap is a finite discretization of the robot's configuration space  $\mathcal{Q}$ , so a configuration  $\mathbf{q}_{\text{goal}}$  where  $\mathbf{x}(\mathbf{q}_{\text{goal}}, 1) = \mathbf{x}_{\text{goal}}$  generally is not exactly represented in the roadmap, and
2. the roadmap planner uses an approximate kinematic model  $\hat{\mathbf{x}}$  to select  $\mathbf{q}_{\text{near\_goal}}$ .

To address both of these challenges, we use measurements from a tip position sensing system combined with iterative inverse kinematics (IK) to "step out" of the precomputed roadmap and toward the specified goal point. See Sec. V-A.2 for details on the tip position measurement system used in our experiments.

To step off the roadmap and move the robot's tip from its sensed position to the goal position, we used the damped least squares (DLS) IK algorithm [25], [3]. If we do not consider kinematic uncertainty (as in our prior work [14] which we only evaluated in simulation), then we could compute the off-roadmap steps using tip error computed from the kinematic model:

$$\Delta \mathbf{q} = \text{DLS\_IK}(\mathbf{x}_{\text{goal}} - \hat{\mathbf{x}}(\mathbf{q}, 1)). \quad (2)$$

In this work, we incorporate feedback from tip position measurements in order to mitigate the effects of kinematic modeling error. We take the off-roadmap steps using the tip error measured by our position sensing system:

$$\Delta \mathbf{q} = \text{DLS\_IK}(\mathbf{x}_{\text{goal}} - \mathbf{x}_{\text{measured}}). \quad (3)$$

We iteratively sense the position of the robot's tip and adjust the robot's configuration using Eq. 3 until convergence or until we fail to make positive progress toward the goal position. We only perform off-roadmap steps if they are predicted to be collision-free with  $\text{is\_collision\_free}(\mathbf{q})$ .

## V. Evaluation

We evaluated our motion planner's ability to accurately maneuver a concentric tube robot's tip to goal positions while avoiding collisions with obstacles. We assess *obstacle avoidance* by whether any point on the concentric tube robot touches an obstacle at any time during a procedure, and we assess accuracy by *tip error*, the Euclidean distance between the specified



goal position and the actual robot tip position (as measured by an external vision-based sensor) after the robot's motion has completed.

### A. Experimental Setup

Our experimental setup consisted of a physical concentric tube robot platform, a stereo vision system, and a pair of test environments.

**1) Physical Concentric Tube Robot Platform**—Our robot's mechanical design is similar to prior work in a quadramanual concentric tube robot system [26]. Our robot is composed of 3 tubes. Each tube  $i$  has outer and inner diameters  $OD_i$  and  $ID_i$ , respectively; a straight segment of length  $L_i^s$ ; and a curved segment of length  $L_i^c$  with radius of curvature  $r_i$ . The specifications of the robot's component tubes are in the table below (all units in millimeters).

Tube	$OD_i$	$ID_i$	$L_i^s$	$L_i^c$	$r_i$
1	0.889	0	219	59	169
2	1.1176	0.9652	114	50	160
3	2.18	2.02	56	21	200

We note that the innermost “tube” in our experiments is actually a wire. We designed the tubes to be slight in curvature in order to maximize the robot's stability and preclude snapping behavior [27], [28], [2].

**2) Tip Position Measurement using Stereo Vision**—We note the concentric tube robot's tip position using a calibrated stereo camera system consisting of two Point Grey Research Flea2 cameras with resolution  $1600 \times 1200$ . We mounted the cameras on a tripod and calibrated the camera system using the Camera Calibration Toolbox for MATLAB [29]. We implemented a basic tracker that automatically locates a small, brightly-colored marker at the tip of the concentric tube robot in each camera image and then triangulates the tip's 3D position in space. This stereo vision setup provided accurate and automated measurements of the concentric tube robot's tip. In a clinical setting, we anticipate using electromagnetic trackers or intraoperative medical imaging to estimate tip position.

**3) Test Environments**—For evaluating our approach, we considered two environments with cylindrical obstacles, environment A and environment B, shown in Fig. 4. Environment A contains 2 thin cylindrical obstacles at 1.4 cm spacing and environment B contains 3 thicker cylindrical obstacles at 1.35 cm spacing. The environments are inspired by the need to avoid thin tubes (e.g., critical blood vessels) during surgical procedures.

In our approach's preoperative phase, we generated 3D polygonal surface meshes of environments A and B. We used these surface meshes to generate the collision-free roadmaps used by the motion planner, as described in Sec. IV-B.1. We generated these roadmaps from 20,000 RRG iterations, which resulted in roadmaps with 19,331



configurations and 17,797 configurations for environments A and B, respectively. Roadmap computation took approximately 3 hours each.

## B. Experimental Trials

We evaluated the effectiveness of the robot with a series of experimental trials. Each experimental trial involved randomly sampling a start and a goal position, executing the motion planner, and evaluating the path taken by the robot for both obstacle avoidance and tip error at the end of the path. Specifically, each experimental trial consisted of the following steps.

1. Sample a start position and a goal position from a given subset of the robot's feasible workspace.
2. Move the robot to the start position (for convenience we use the collision-free roadmap for this).
3. Use the motion planner to compute a motion plan to the goal, and execute the motion plan and tip error correction.
4. Record (a) the tip error, and (b) whether a collision occurred during execution of the motion. If a collision occurred, we measure the tip error at the last collision-free configuration.

We labeled an experimental trial a success if the robot's tip reached within 1 mm of the goal position without colliding with environmental obstacles, and a failure otherwise.

The feasible workspace subsets used for sampling start and goal positions in environments A and B are illustrated in Fig. 4. We estimated these regions using a rapidly-exploring random tree (RRT) algorithm [30] to densely sample points reachable by collision-free paths, and sampled from these points for our planning queries. We note that these configurations were distinct from those sampled when generating our collision-free roadmaps.

We compare our motion planning approach with several related methods. Specifically, we evaluated the robot's performance using the following methods for computing and executing motions in step 3 above.

1. *Iterative IK*: The robot is controlled to move from the start to the goal using the DLS IK algorithm from Eq. 2. This method does not use the collision-free roadmap or tip position measurements.
2. *TEC*: The robot is controlled to move from the start to the goal using only the tip error correction (TEC) algorithm based on Eq. 3. This method does not use the collision-free roadmap.
3. *Roadmap*: The teleoperation system uses the collision-free roadmap to compute a motion plan to the configuration in the roadmap that is closest to the goal, and then executes this motion plan in an open-loop manner. This approach does not consider tip position error due to inaccuracies in the kinematic model.

4. *Roadmap + IK*: The above “Roadmap” method is executed. It then uses DLS IK to move the robot’s tip closer to the goal. This approach does not consider tip position error due to inaccuracies in kinematic model.
5. *Roadmap + TEC (our complete method)*: The above “Roadmap” method is executed. It then uses TEC to attempt to move the robot’s tip closer to the goal.

We performed 10 trials for environment A and for environment B for each of the above methods. We used the same set of start/goal position pairs to evaluate each method.

### C. Experimental Results

We show failure rates for each method in Fig. 5. In both environments, the roadmap-based variants caused fewer collisions than the variants without roadmap planning; this shows that explicit obstacle avoidance using a collision-free roadmap approach is beneficial. Our complete method (Roadmap + TEC) had the lowest overall failure rate due to its consideration of both obstacle avoidance (using the roadmap) and accuracy (using the sensing-based tip error correction).

We show the average tip position errors over all trials for each planner variant in Fig. 6. We note that, if a trial resulted in collision, we recorded the tip position error for that trial at the last collision-free configuration. The roadmap-based variants exhibited significantly lower tip position errors because roadmap planning enables globally collision-free routing to points near the target. The benefit of tip error correction is shown in the low errors for the Roadmap + TEC variant: average errors of 0.21 mm and 0.18 mm. The other roadmap-based variants exhibit average errors around 3 mm, which push the boundary of required precision in minimally invasive surgery.

New goal position requests resulted in motion response from the concentric tube robot within an average of 0.118 seconds. This slight latency (caused by intraoperative phase motion planning) is hidden by the time taken for the robot to reach each new goal position.

## VI. Conclusion

To the best of our knowledge, this is the first teleoperated concentric tube robot with integrated motion planning for automatic obstacle avoidance. Our experiments show that our motion planning approach enabled powerful obstacle avoidance capabilities, but there is always room for improvement. The motion plans generated by the roadmap-based planner are executed in an open-loop fashion until the tip error correction step. Our current approach uses a 1 mm buffer around obstacles to account for uncertainty in the robot’s kinematic model, but a more principled approach that considers models of kinematic uncertainty could provide better guarantees for collision avoidance during the roadmap execution stage. This problem enables exciting potential research in closed-loop sensing, planning, and execution for continuum robots with kinematic modeling error. For instance, recent work in continuum robot shape sensing [31], [32], [33] can potentially be combined with uncertainty-aware manipulation planning [34], [13] in order to maximize the effectiveness of automatic obstacle avoidance.

Our motion planning approach for concentric tube robot teleoperation achieves interactive execution speeds with high accuracy via precomputation. The precomputation requires knowledge of obstacle location, e.g., from preoperative medical imaging. While this may be appropriate for relatively static surgical sites (e.g., in the skull base), the clinical need to operate where the anatomy is in flux (e.g., beating hearts and breathing lungs) motivates investigation of motion planning methods that can handle dynamic environments while maintaining interactive planning rates.

Our use of a 3D mouse combined with an augmented reality interface makes the robot's responses to the user's inputs more intuitive, and this approach raises other interesting questions about robot-physician interfaces. For instance, the user's spatial reasoning of the surgical site could be improved with a stereoscopic display or visual overlay of surgically relevant anatomical structures. Additionally, force feedback could provide the physician with tactile information about the surroundings of the concentric tube robot. This and future work in robot-physician interfaces for concentric tube robot teleoperation should be evaluated via user studies.

In order to encourage stability, we used concentric tubes with slight curvatures, but higher curvatures allow concentric tube robots to attain more diverse shapes and can augment their ability to curve around obstacles. Augmenting our motion planner with recent work in avoiding or mitigating the unstable snapping effects [27], [15], [35], [36] caused by tight tube curvatures may enable stable yet highly dexterous obstacle avoidance for concentric tube robots.

## Acknowledgments

This research is supported in part by the National Science Foundation (NSF) Graduate Research Fellowship Program under Grant No. DGE-1144081 as well as NSF award IIS-1149965 and by the National Institutes of Health (NIH) under awards R01EB017467 and R21EB017952.

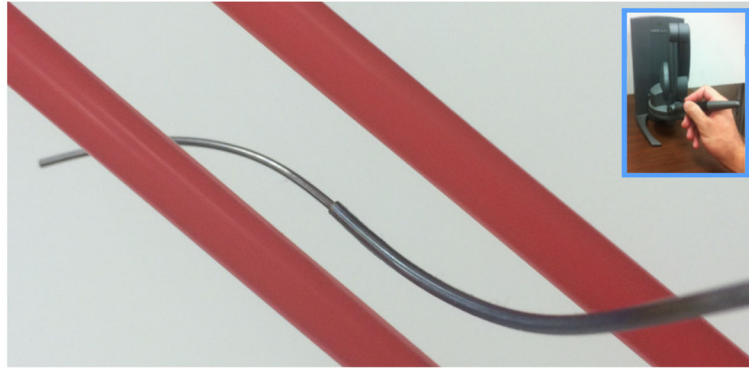
We thank Jim Mahaney for his assistance with the experimental setup, Andrew Maimone for his insights on camera calibration, and Dr. Brent Senior for insights on the clinical application of these robots.

## References

1. Gilbert HB, Rucker DC, Webster RJ III. Concentric tube robots: The state of the art and future directions. *Int Symp Robotics Research (ISRR)*. Dec.2013
2. Dupont PE, Lock J, Itkowitz B, Butler E. Design and control of concentric-tube robots. *IEEE Trans Robotics*. Apr; 2010 26(2):209–225. [PubMed: 21258648]
3. Rucker, DC. PhD dissertation. Vanderbilt University; 2011. The mechanics of continuum robots: model-based sensing and control.
4. Xu, R.; Asadian, A.; Naidu, AS.; Patel, RV. Position control of concentric-tube continuum robots using a modified Jacobian-based approach. *IEEE Int. Conf. Robotics and Automation (ICRA)*; May 2013; p. 5793-5798.
5. Sears, P.; Dupont, PE. A steerable needle technology using curved concentric tubes. *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*; Oct. 2006; p. 2850-2856.
6. Rucker DC, Webster RJ III. Parsimonious evaluation of concentric-tube continuum robot equilibrium conformation. *IEEE Trans Biomedical Engineering*. Sep; 2009 56(9):2308–2311.
7. Dupont, PE.; Lock, J.; Butler, E. Torsional kinematic model for concentric tube robots. *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*; May 2009; p. 3851-3858.

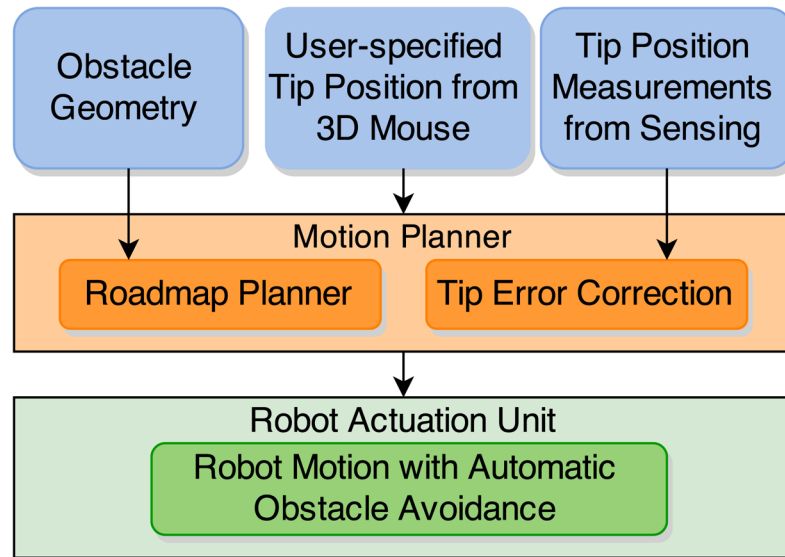
8. Rucker DC, Jones BA, Webster RJ III. A geometrically exact model for externally loaded concentric-tube continuum robots. *IEEE Trans Robotics*. Jan; 2010 26(5):769–780. [PubMed: 21566688]
9. Lock, J.; Laing, G.; Mahvash, M.; Dupont, PE. Quasistatic modeling of concentric tube robots with external loads. *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*; Oct. 2010; p. 2325-2332.
10. Lock, J.; Dupont, PE. Friction modeling in concentric tube robots. *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*; May 2011; p. 1139-1146.
11. Lyons, LA.; Webster, RJ., III; Alterovitz, R. Motion planning for active cannulas. *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*; Oct. 2009; p. 801-806.
12. Trovato K, Popovic A. Collision-free 6D non-holonomic planning for nested cannulas. *Proc SPIE Medical Imaging*. Mar.2009 7261
13. Torres, LG.; Alterovitz, R. Motion planning for concentric tube robots using mechanics-based models. *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*; Sep. 2011; p. 5153-5159.
14. Torres, LG.; Baykal, C.; Alterovitz, R. Interactive-rate motion planning for concentric tube robots. *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*; May 2014; p. 1915-1921.
15. Bergeles, C.; Dupont, PE. Planning stable paths for concentric tube robots. *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*; Nov. 2013; p. 3077-3082.
16. Choset, H.; Lynch, KM.; Hutchinson, SA.; Kantor, GA.; Burgard, W.; Kavraki, LE.; Thrun, S. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press; 2005.
17. Karaman S, Frazzoli E. Sampling-based algorithms for optimal motion planning. *Int J Robotics Research*. Jun; 2011 30(7):846–894.
18. Zucker M, Ratliff N, Dragan AD, Pivtoraiko M, Matthew K, Dellin CM, Bagnell JA, Srinivasa SS. CHOMP: Covariant Hamil-tonian optimization for motion planning. *Int J Robotics Research*. Aug; 2012 32(9):1164–1193.
19. Kalakrishnan, M.; Chitta, S.; Theodorou, E.; Pastor, P.; Schaal, S. STOMP: Stochastic trajectory optimization for motion planning. *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*; May 2011; p. 4569-4574.
20. Schulman J, Duan Y, Ho J, Lee A, Awwal I, Bradlow H, Pan J, Patil S, Goldberg K, Abbeel P. Motion planning with sequential convex optimization and convex collision checking. *International Journal of Robotics Research*. Aug; 2014 33(9):1251–1270.
21. Johnson, HJ.; McCormick, M.; Ibáñez, L. Insight Software Consortium. The ITK Software Guide. Dec. 2013 Available: <http://www.itk.org/ItkSoftwareGuide.pdf>
22. Geomagic® Touch™ X Haptic Device. [Online]. Available: <http://geomagic.com/en/products/phantom-desktop/overview>
23. Bradski, G. Dr Dobb's Journal of Software Tools. 2000. The OpenCV Library.
24. Pan, J.; Chitta, S.; Manocha, D. FCL: A general purpose library for collision and proximity queries. *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*; May 2012; p. 3859-3866.
25. Wampler CW. Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods. *IEEE Trans Systems, Man and Cybernetics*. 1986; 16(1):93–101.
26. Swaney PJ, Croom JM, Burgner J, Gilbert HB, Rucker DC, Webster RJ III, Weaver KD, Russell PT III. Design of a quadramanual robot for single-nostril skull base surgery. *ASME Dynamic Systems and Control*. Oct.2012 :387–393.
27. Xu, R.; Atashzar, SF.; Patel, RV. Kinematic instability in concentric-tube robots: modeling and analysis. *Proc. IEEE RAS/EMBS Int. Conf. Biomedical Robotics and Biomechatronics (BioRob)*; Aug. 2014; p. 163-168.
28. Webster RJ III, Romano JM, Cowan NJ. Mechanics of precurved-tube continuum robots. *IEEE Trans Robotics*. Feb; 2009 25(1):67–78.
29. Bouguet, J-Y. Camera Calibration Toolbox for Matlab. [Online]. Available: <http://www.vision.caltech.edu/bouguetj/calibdoc/>
30. LaValle, SM. *Planning Algorithms*. Cambridge, U.K: Cambridge University Press; 2006.

31. Lobaton, E.; Fu, J.; Torres, LG.; Alterovitz, R. Continuous shape estimation of continuum robots using X-ray images. *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*; May 2013; p. 717-724.
32. Ryu, SC.; Dupont, PE. FBG-based shape sensing tubes for continuum robots. *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*; May 2014; p. 3531-3537.
33. Kim, B.; Ha, J.; Park, FC.; Dupont, PE. Optimizing curvature sensor placement for fast, accurate shape sensing of continuum robots. *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*; May 2014; p. 5374-5379.
34. Sun, W.; Torres, LG.; van den Berg, J.; Alterovitz, R. Safe motion planning for imprecise robotic manipulators by minimizing probability of collision. *Proc. Int. Symp. Robotics Research (ISRR)*; Dec. 2013;
35. Kim, J-S.; Lee, D-Y.; Kim, K.; Kang, S.; Cho, K-J. Toward a solution to the snapping problem in a concentric-tube continuum robot: grooved tubes with anisotropy. *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*; May 2014; p. 5871-5876.
36. Azimian, H.; Francis, P.; Looi, T.; Drake, J. Structurally-redesigned concentric-tube manipulators with improved stability. *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*; Sep. 2014; p. 2030-2035.



**Fig. 1.**

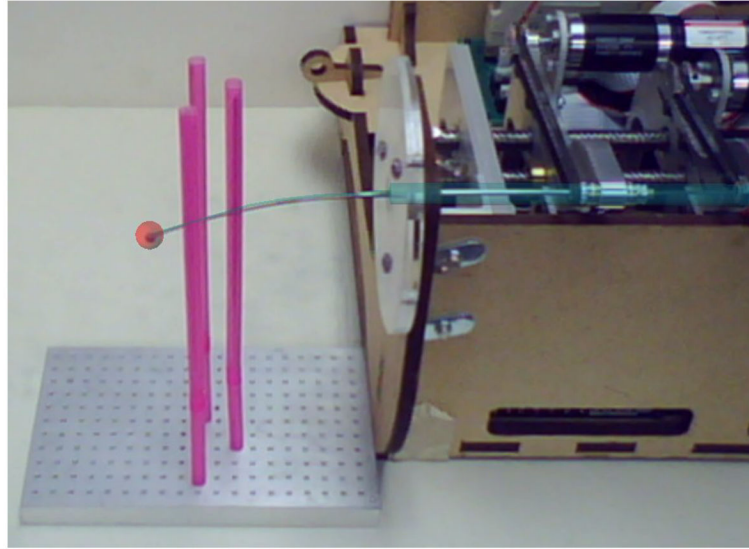
A concentric tube robot steering between and avoiding two tubular obstacles. We developed a fast motion planner to enable a teleoperated concentric tube robot to maneuver its tip to points specified by a 3D mouse (see inset) while automatically avoiding collisions with obstacles along its shaft.



**Fig. 2.**

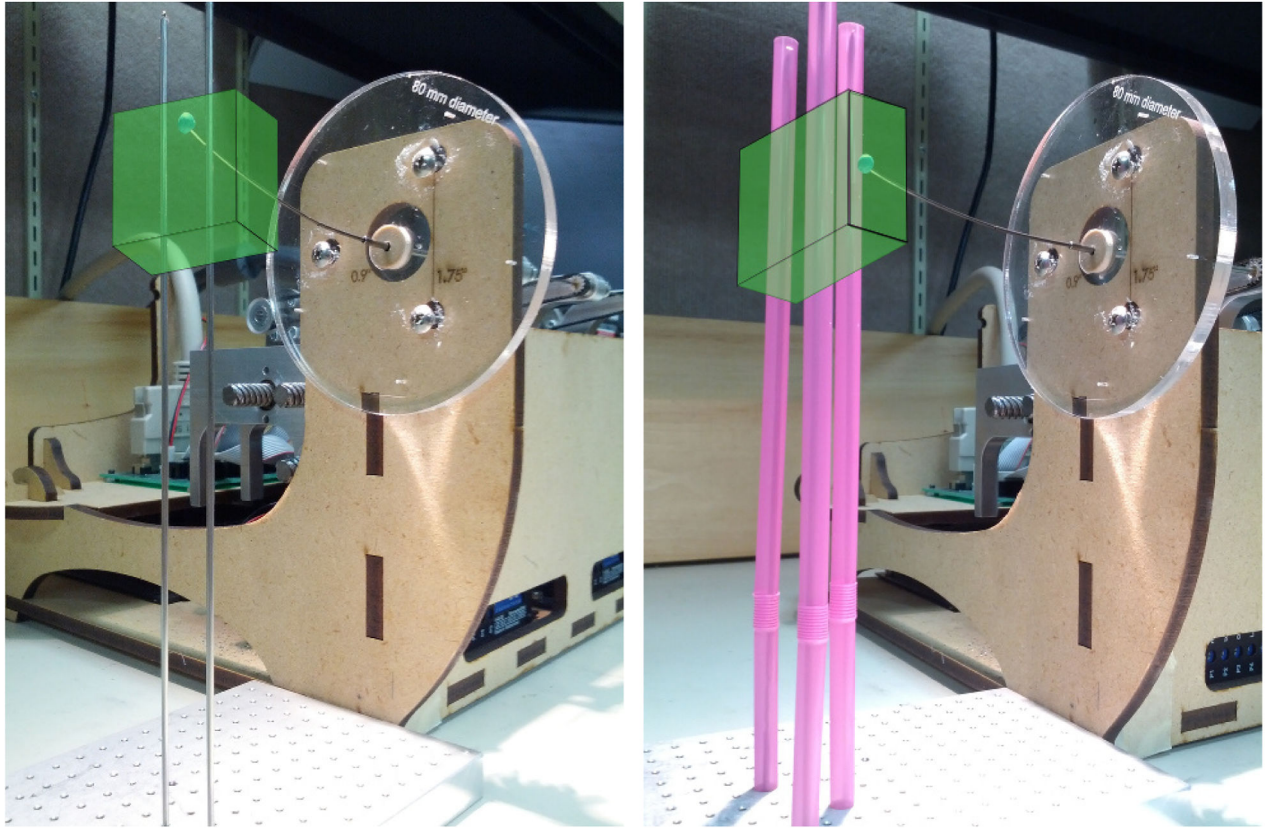
A schematic overview of our motion planning approach for concentric tube robot teleoperation with automatic obstacle avoidance.



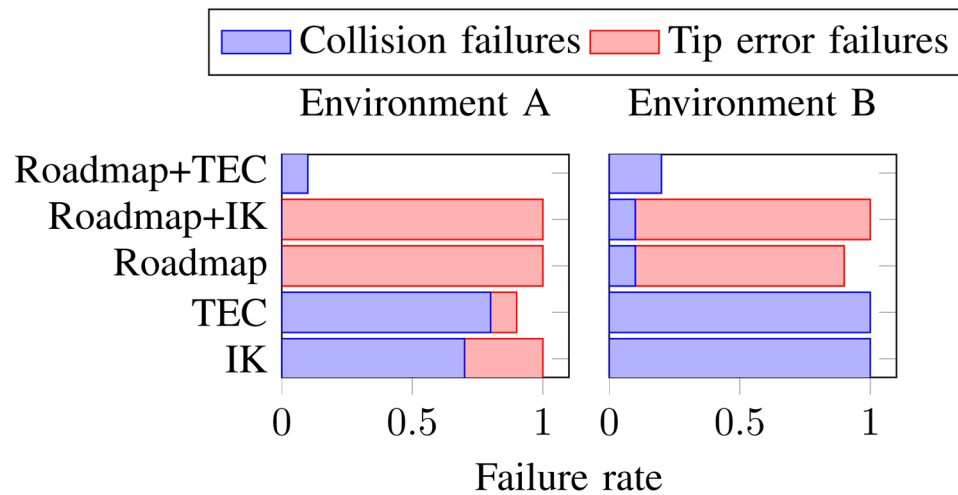


**Fig. 3.**

A screen capture of the augmented reality interface for teleoperating concentric tube robots. The translucent aqua curve is the robot's shape as estimated by the kinematic model, which almost exactly corresponds to the actual tubes in the video feed. The translucent red sphere corresponds to the 3D cursor, which the user moves with a 3D mouse to select goal positions for the robot tip.

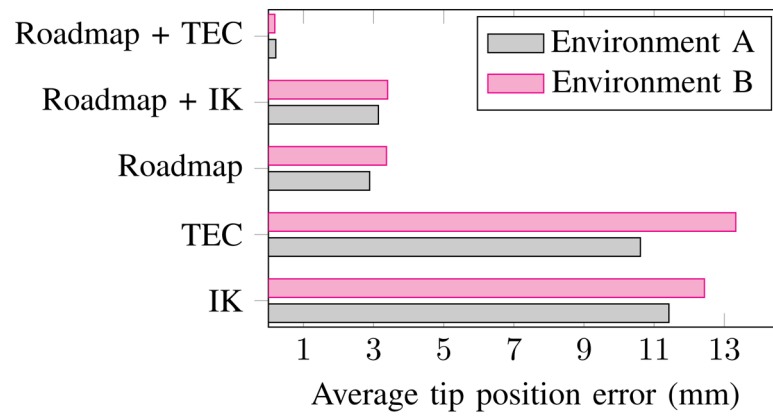


**Fig. 4.** Environment A (left) and Environment B (right). Start and goal positions sampled from a region approximately rendered in green.



**Fig. 5.**

Rates of procedure failure of several planners in both environments. Bars are color-coded to distinguish between (1) failures due to obstacle collision or (2) failure to place the robot's tip within 1 mm of the goal. Roadmap-based planners cause far fewer collisions than planning without a roadmap. Our complete method, which combines roadmap-based planning with sensing-based tip error correction to overcome kinematic modeling error, achieves the lowest failure rate.



**Fig. 6.**

Average tip position errors for each planner variant in both environments. If a trial resulted in collision, we recorded the error at the last collision-free configuration. The Roadmap + TEC variant combines roadmap-based collision-free routing with sensing-based tip error correction to achieve the lowest errors (0.21 mm and 0.18 mm for environments A and B, respectively).