

Vision-based Autonomous Blood Suction with a Concentric Tube Continuum Robot

Jinjie Sun^{1,2,3}, Courtney Amm^{1,3}, Jessica Burgner-Kahrs^{2,3,4}, Lueder A. Kahrs^{1,3,4,5,6}

¹ Medical Computer Vision and Robotics (MEDCVR) Lab, University of Toronto, Toronto, Canada

² Continuum Robotics Laboratory (CRL), University of Toronto, Toronto, Canada

³ Department of Computer Science, University of Toronto, Toronto, Canada

⁴ Department of Mathematical and Computational Sciences, University of Toronto Mississauga, Mississauga, Canada

⁵ Institute of Biomedical Engineering, University of Toronto, Toronto, Canada

⁶ Department of Otolaryngology – Head and Neck Surgery, University of Toronto, Toronto, Canada

{jinjie.sun, courtney.amm}@mail.utoronto.ca {lueder:kahrs, jessica.burgnerkahrs}@utoronto.ca

Abstract—Blood-water mixture removal is essential in many surgeries. Existing works adding a robotic assistant primarily focus on conventional robots. Limited attention has been given to using continuum robots for this task. This paper introduces a vision-based control framework for autonomous liquid suction using a concentric tube continuum robot (CTCR). The proposed method employs a controller using camera input combined with a hybrid control strategy that integrates differential inverse kinematics and a pre-computed lookup table to ensure stable and precise motion during suction. A CTCR simulator, implemented in the Unity Game Engine with photo-realistic rendering and robot-liquid interaction capabilities, as well as a benchtop robot system were developed as the experiment platform. The proposed method was evaluated through simulation and real-world experiments across four scenarios, demonstrating its generalizability and stability. In 32 real-world trials, less than 0.1 g of liquid remained after the suction, while over 99% of liquid was removed across 32 simulated trials. The results highlight the potential of CTCR for autonomous surgical liquid suction, showcasing the system’s adaptability and performance in dynamic environments.

Keywords-Autonomous Robotic Surgery, Robot Simulator, Continuum Robot

I. INTRODUCTION

In many surgical procedures, the removal of fluids, such as blood, sterile water for cleaning, or cooling purposes, is essential to maintain a clear operative field and ensure optimal visualization of anatomical structure. Traditionally, in microscope-based procedure, this task is handled by an assistant, requiring constant attention to the removal of fluids to prevent visual obstruction. However, the repetitive nature of the fluid suctioning task makes it an ideal candidate for automation. Autonomous surgical robots could efficiently handle this task, relieving operating room staff of routine duties and allowing them to focus on critical aspects of the operation [1]. For existing works addressing autonomous blood suction tasks, the majority of research has concentrated on traditional robotic systems, such as the da Vinci Research Kit [2]–[4] and custom-designed robots [5]. However, the application of continuum robots, particularly in the context of autonomous liquid removal tasks, remains underexplored.

Concentric Tube Continuum Robots (CTCR), composed of telescoping pre-curved tubes, have demonstrated promise in minimally invasive surgeries due to their dexterity and ability to navigate complex and confined anatomical spaces [6]–[8]. The telescopic structure of CTCR enables independent control of both tube rotation and translation. Additionally, their high compliance and hollow features make them particularly suitable for in-vivo operations, especially in narrow surgical corridors.

In this paper, we introduce a vision-based controller for autonomous liquid suction tasks, utilizing visual and depth inputs to guide the CTCR. A fine-tuned ResNet18 model [9] is employed for segmentation of the CTCR body and tip from real-time input images, enabling precise tracking of the CTCR tip relative to the liquid surface. Additionally, we developed a photo-realistic CTCR simulator using the Unity Game Engine, which incorporates realistic physical properties of liquid interaction using Zebra Liquid from Zebra AI Inc. This simulator is instrumental in generating training datasets for the segmentation model and validating the proposed autonomous CTCR control system in the simulated environments.

Our primary contributions are as follows:

- 1) A CTCR simulator developed in Unity that allows for liquid interaction and collision detection, facilitating data collection, training, and evaluation of robotic performance.
- 2) A closed-loop control strategy for blood suction tasks, including a decision-making algorithm to guide the robot to remove blood autonomously and a CTCR controller that translates the desired end-effector movements into joint-level commands using differential inverse kinematics and a pre-computed look-up table for rapid search.
- 3) Experimental validation of the proposed framework through real-world robotic setup and task execution, as demonstrated in Figure 1.

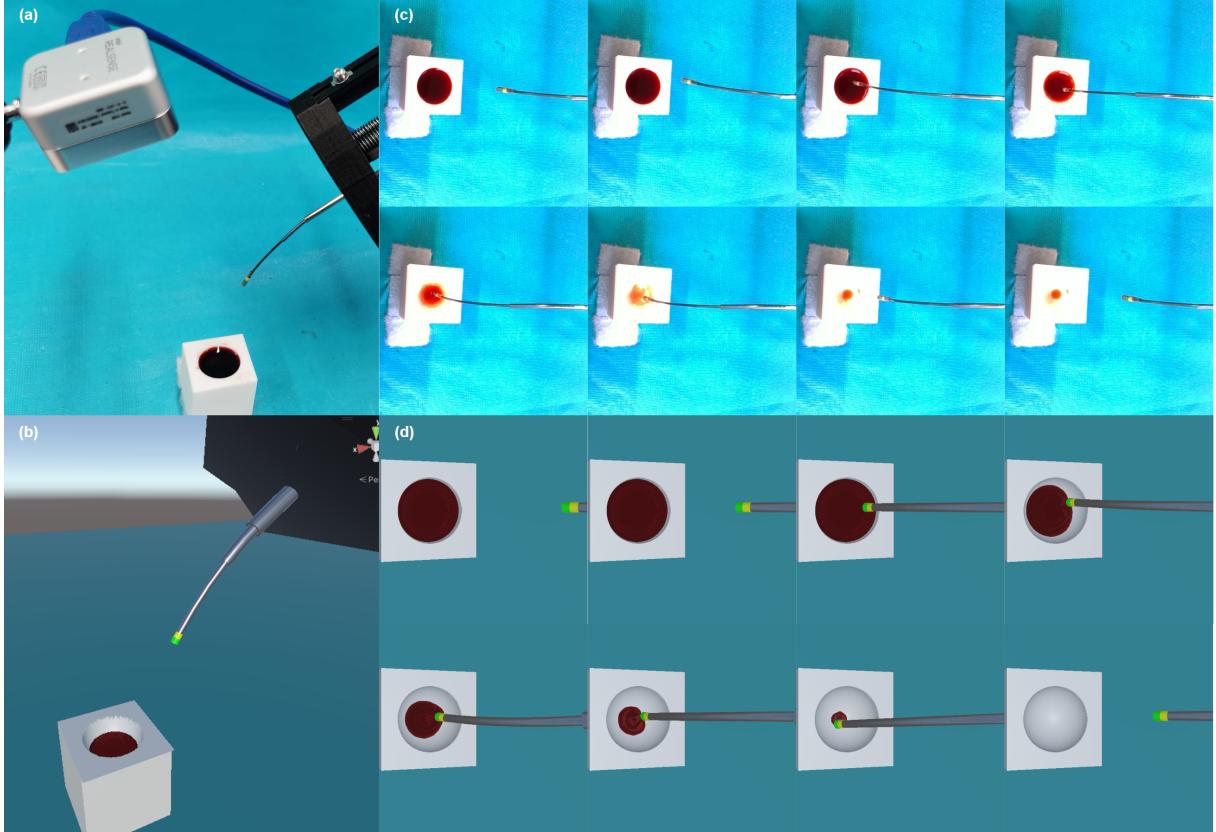


Figure 1: Auto-Suction Experiment Setup and Results. (a) Real-world experiment environment (b) Snapshot from simulation environment. (c) Temporal sequence images from the real-world bowl experiment. (d) Temporal sequence images from the simulation bowl experiment.

II. RELATED WORK

Autonomous blood suction using surgical robots remains an emerging research direction, with few studies to date addressing this challenge. Su et al. [5] introduced a robotic system for autonomous blood removal by employing a pre-trained Mask R-CNN model and fine-tuned using real-world images to detect blood contours from dual cameras. Although their method demonstrated effectiveness, it primarily focused on superficial blood regions and approached the problem as a planar path-planning task, assuming the blood contour could be represented as a planar closed curve, which limits its broader applicability. Richter et al. [2] proposed an approach based on blood flow detection, using optical flow estimates from a pre-trained CNN model. They employed the da Vinci Research Kit (dVRK) [10] to carry out the task. However, their method relies on detectable liquid flow in the input, and the open-loop controller prevents path adjustment during task execution, making it difficult to handle dynamic changes in the environment. Huang et al. [3] used model predictive control (MPC) in conjunction with differentiable position-based fluids (PBF) model to optimize the path for maximum liquid removal. While effective in simulation,

their method relies on an accurate PBF model, which poses challenges when applied to real-world scenarios.

Ou et al. [4] developed a particle-based fluid (PBF) simulator using NVIDIA PhysX 5 and the Unity Game Engine to train a reinforcement learning (RL) agent for autonomous blood suction tasks. Their approach effectively leverages RL techniques; however, it necessitates detailed information about the environment, including container characteristics like height maps, segmented liquid masks, and the robot's end-effector position.

Unlike previous studies that primarily utilized traditional robots, Lai et al. [11] introduced a two-segment cable-driven soft manipulator for liquid suction tasks focusing on pose control rather than autonomous suction, requiring manual positioning of the robot beneath the liquid surface to initiate the task, with no ability for trajectory re-planning. Godage et al. [12] demonstrated a two-segment CTCR for intracerebral hemorrhage evacuation, utilizing intraoperative CT scans to iteratively re-plan trajectories during the evacuation process. While their approach incorporates intraoperative imaging, it relies on intermittent CT updates and open-loop execution between scans, which may not capture rapid changes of the

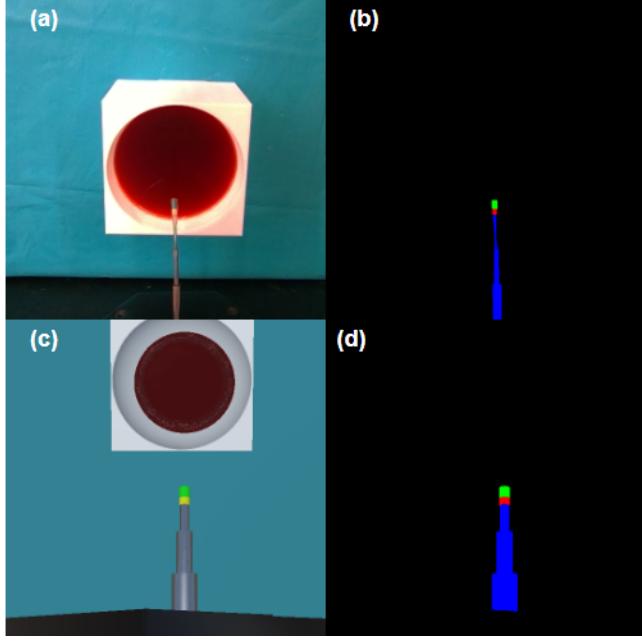


Figure 2: RGB image and mask pair of real-world and simulation experiments. (a) Image of real-world experiment, (b) Mask of real-world experiment, (c) Image of simulation experiment, (d) Mask of simulation experiment

pathology blood flow.

Our work addresses two key limitations of existing systems. First, current approaches require prior knowledge of the surface. In contrast, we propose a framework that leverages RGB-D camera outputs and registration techniques, reducing reliance on detailed pre-calibration. Second, while existing controllers depend on intermittent imaging, our approach achieves closed-loop autonomy by processing continuous intraoperative RGB-D data (30 Hz input, 4 – 7 Hz control updates). This allows dynamic adjustment of suction actions in response to real-time fluid flow—a critical capability for handling the unpredictable movement of bleeding sites during surgery.

III. METHODS

We propose a vision-based closed-loop control system that utilizes color and depth images as references to drive the CTCR in completing the suction task step-by-step. For the CTCR control, we extend an existing approach using differential inverse kinematics with a pre-computed loop-up table.

A. Blood Detection and CTCR Segmentation

For blood region detection, we utilize color-based segmentation using a selected color feature as suggested by [13]: the red ratio of a pixel. The red ratio, calculated as the ratio of the red channel value to the sum of the RGB values of the pixel, $\frac{r}{r+g+b}$ is used to classify pixels as blood pixels

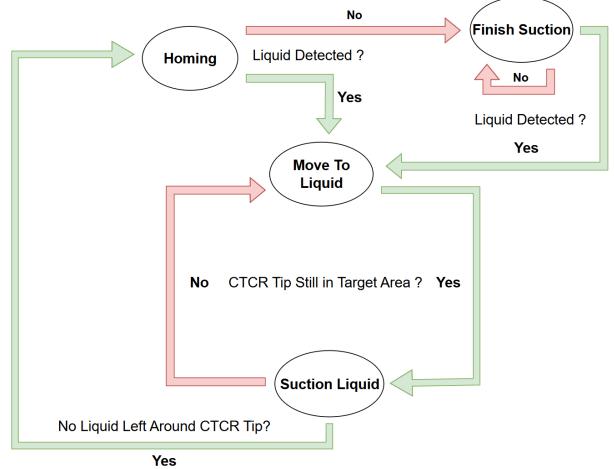


Figure 3: Blood suction decision-making unit structure.

if their ratio exceeds a threshold. This method generates a binary mask indicating the blood region.

The control algorithm relies on information about the relative position between the CTCR tip and the target blood region to compute movement commands. For CTCR tip tracking, we adopted a pre-trained CNN model based on the U-Net architecture with a ResNet18 encoder [14] as the backbone model, fine-tuned with our dataset and achieved an average Intersection-over-Union of 0.8. Our dataset comprises 762 images of a CTCR and a white bone-like phantom from both real-world (450) and simulated (312) scenarios. The ground truth of the tube segmentation mask is obtained by manually labeling the real-world images; for simulation, a customized rendering pipeline has been used to generate the ground truth masks. Figure 2 shows the image and mask pairs for both scenarios. The CNN model outputs the CTCR segmentation mask and additional segmentation masks for two color tapes placed at the distal ends of the CTCR, which serve as markers to indicate the relevant depth between the distal ends and the liquid surface. In real-world experiments, an additional color segmentation process is applied to the detected tip region using a small window size to enhance the detection results of the two color tapes.

B. Autonomous Blood Suction Controller

The controller comprises two components: an autonomous blood suction decision-making unit and a robot controller. The decision-making unit employs a state-based logic that determines high-level movement commands based on the current operational state and the segmentation results from color-depth image pairs. This logic operates through four distinct states: **Homing**, **Move to Liquid**, **Suction Liquid**, and **Finish Suction**. Transitions between states are conditionally triggered by the system's evaluation of the segmented image data and task progress. In each state, the system decides on a movement type and step size, which is

equivalent to the desired end-effector velocity with discrete implementations. The algorithm will process and generate new motion commands once the new images are received and the previous motion has been completed.

1) *Decision-making Unit*: The decision-making unit begins in the **Homing** state, directing the robot to a predefined home position. Upon arrival, a non-empty blood segmentation mask triggers the **Move to Liquid** state, generating 3D directional commands (CTCR's tip position to the estimated liquid center position) derived from color/depth images and segmentation masks. These commands guide the CTCR tip toward a target subregion near the blood center. If multiple blood regions are detected, the algorithm selects the largest region as the primary target and computes its center from the segmentation mask.

Upon entering the target area, the **Move to Liquid** state transitions to the **Suction Liquid** state, moving the CTCR tip below the liquid surface to perform the suction task. Two color markers are placed at the distal tip to indicate the CTCR's depth below the liquid surface: green and yellow ring marker, where the green ring is closer to the distal tip of the CTCR. Downward movement commands are sent when the green marker is visible and there are pixels from the fluid around the CTCR distal ends, indicating the tip of the CTCR is still above the liquid surface. If only the yellow marker is detected, it signifies the tip is at the desired depth below the surface, and the CTCR remains in position for suctioning. When the yellow marker cannot be detected, it indicates the tip is too deep into the liquid; as a result, upward movement commands are generated until the yellow marker is detected. All movement commands are in the base frame and are converted to the robot frame before being sent to the robot controller to handle various robot positions and orientations.

The state exits the **Suction Liquid** phase if there are insufficient blood pixels around the tip area, suggesting the completion of the suction task for the local region. The CTCR will return to the home position, and the state will change to the initial state, ready to move once a new blood region has been detected. Figure 3 illustrates the proposed decision-making unit.

2) *Robot Controller*: The robot controller is based on a weighted damped least-squares method [15] which maps desired task space velocities to corresponding joint space velocities (tube rotations and translations) using the pseudoinverse of the CTCR's Jacobian. The Jacobian is computed from the forward kinetostatic model of the CTCR using finite differences [15]. As a local controller, the WDLS differential inverse kinematics smoothes solutions and mitigates singularity issues. However, CTCR exhibit highly nonlinear kinematics, making it challenging to find a global solution using only differential methods. As shown by Dupont et al. [16], a lookup table can provide a coarse, globally informed estimate of the inverse kinematics. Our

Algorithm 1 CTCR dIK Controller

Require: Robot Jacobian Matrix J_θ , Desired Motion δp , Current EE Position p_{EE} , Current Joint Value q , Sampled Matrix M , Window Size w

Ensure: Updated Joint Values δq

- 1: $\delta q \leftarrow \text{DLSdIK}(J(q), \delta p)$
- 2: $q_{\text{new}} \leftarrow q + \delta q$, $p_{\text{est}} \leftarrow J(q_{\text{new}})q_{\text{new}}$
- 3: $v_{\text{dIK}} \leftarrow \|p_{\text{est}} - p\|$
- 4: **if** $\cos \text{sim}(v_{\text{dIK}}, \delta p) < \text{Threshold}$ **then**
- 5: $\delta q \leftarrow \text{SearchFromLookUpTable}(p_{EE}, \delta p, M, w)$
- 6: **end if**
- 7: **return** δq

Algorithm 2 Look-up Table Search

Require: Current EE Position P_{EE} , Desired Movement δp , Matrix M , Window Size w , Current Joint Value q

Ensure: Updated Joint Values δq

- 1: $P_t \leftarrow P_{EE} + \delta p$, $L1 \leftarrow \text{GetLookUpTable}(M)$, $M_2 \leftarrow \text{Get3DPositionToTableKey}(M)$
- 2: $M_{\text{sub}} \leftarrow M_2[P_t.x \pm w, P_t.y \pm w, P_t.z \pm w]$
- 3: $d_{\min} \leftarrow \infty$, $p_{\min} \leftarrow \text{NA}$
- 4: **for** each row i in M_{sub} **do**
- 5: $(p_{\text{temp}}, d_{\text{temp}}) \leftarrow \text{ClosestPoint}(M_{\text{sub}}[i], P_t)$
- 6: **if** $d_{\text{temp}} < d_{\min}$ **then**
- 7: $d_{\min} \leftarrow d_{\text{temp}}$, $p_{\min} \leftarrow p_{\text{temp}}$
- 8: **end if**
- 9: **end for**
- 10: $\delta q \leftarrow L1[p_{\min}] - q$
- 11: **return** δq

approach therefore includes a pre-computed lookup table, to identify joint values which are then refined by the damped least squares procedure. This hybrid strategy enhances robustness and convergence in real time by exploiting both a precomputed global model and a locally optimal correction.

Algorithm 1 outlines the CTCR controller's pseudo-code. It first computes joint velocities via differential inverse kinematics. The controller evaluates the cosine similarity between desired and actual end-effector motions; if below a threshold, indicating significant deviation, it switches to a search-based method. This method identifies alternative joint velocities to minimize positional error while penalizing excessive joint motion.

In [15], the differential inverse kinematics problem is framed into an optimization problem that aims to minimize the following loss function:

$$F = \frac{1}{2} \left((\mathbf{J}\dot{\mathbf{q}} - \mathbf{v}_0)^T \mathbf{W}_0 (\mathbf{J}\dot{\mathbf{q}} - \mathbf{v}_0) + (\dot{\mathbf{q}} - \mathbf{v}_1)^T \mathbf{W}_1 (\dot{\mathbf{q}} - \mathbf{v}_1) \right) \quad (1)$$

where \mathbf{W}_0 and \mathbf{W}_1 are positive definite diagonal weight-

ing matrices, \mathbf{J} is the Jacobian matrix, \mathbf{v}_0 is the desired end-effector velocity, \mathbf{v}_1 is the gradient of the undesired configuration penalty w.r.t. each joint, and $\dot{\mathbf{q}}$ is the joint velocity. For CTCR, the first three entries of $\dot{\mathbf{q}}$ represent the speed of rotation and the last three represent the speed of translation. By setting $\frac{\partial F}{\partial \dot{\mathbf{q}}} = 0$, the $\dot{\mathbf{q}}$ to minimize the loss function can be computed as follows:

$$\dot{\mathbf{q}} = (\mathbf{J}^T \mathbf{W}_0 \mathbf{J} + \mathbf{W}_1)^\dagger (\mathbf{J}^T \mathbf{W}_0 \mathbf{v}_0 + \mathbf{W}_1 \mathbf{v}_1) \quad (2)$$

The search-based method utilizes the kinetostatic model of the CTCR to compute the resulting end-effector position in the spatial frame using sampled joint values. Before running the search, a lookup table and a 4D matrix are generated. The lookup table matches each sampled point to a unique key value, while the 4D matrix represents a spatial grid that divides the sampled workspace into grids, each containing sampled points. During the search process, a predefined search size is used to select all the grids around the target point. Every point in the selected grid is then checked to find the point with the minimum distance to the target point in both spatial and joint space, based on the preceding equations. Algorithm 2 provides the pseudo-code of this approach, where M is an $N \times 9$ matrix with N being the number of sampled points, and the columns correspond to the six joint values along with the corresponding end-effector spatial position in Cartesian space. The lookup table L_1 is structured as a dictionary with a $N \times 9$ (key, value) pair, where each sampled point corresponds to a unique key. The second matrix, M_2 , is the 4D matrix sized $X \times Y \times Z \times D$, where X , Y , and Z are the maximum differences in the sampled workspace in mm, representing the spatial grid. Each entry $M_2[x, y, z]$ contains a list of key values corresponding to rounded end-effector positions.

For the CTCR used in this paper, we choose $\mathbf{W}_0 = \text{diag}(1e4, 1e4, 1e4, 82, 82, 820)$ and $\mathbf{W}_1 = \text{diag}(82, 82, 82, 1e3, 1e3, 1e3)$ for stable performance, and set the penalty function L of undesired configuration as follows:

$$L = \sum_{i=1}^3 \frac{1}{d_{i,i+1}} + \frac{1}{p_{i,i+1}} \quad (3)$$

where $d_{i,i+1}, p_{i,i+1}$ are the distance of the proximal and distal ends between tube i and tube $i + 1$. With discrete implementations, the joint velocities represent the updated joint values in the next step, which will move the CTCR tip in the desired direction, following the movement command sent by the decision-making unit.

C. CTCR Simulator

We developed the CTCR liquid interaction simulator using the Unity3D game engine as the platform. Unity provides photo-realistic, physical, and real-time rendering results, along with an active developer community that offers many assets readily available for integration into the simulator. In

our simulator, we utilized the Zibra Liquid Asset (Version 2.1.5, Zibra AI Inc., Ukraine) to achieve realistic liquid interaction effects and integrated a kinematics solver [15] to obtain a physically-based model of the CTCR. Additionally, to generate tube-like meshes following the physical model, we build upon an open source Unity tube generation implementation [17] and incorporated material asset [18] to achieve metallic material rendering results.

IV. EXPERIMENTS

In this section, we present the details of the experimental setup for testing the performance of our blood suction controller. We evaluate its capabilities both in simulation and real-world environments under various scenarios, which are designed to assess the generalizability of the controller across different liquid states and surface conditions.

A. Experiment Scenarios

We designed four distinct containers, each with increasing complexity:

Bowl: This scenario evaluates the controller's ability to perform a complete blood suction task, including accurate classification and tracking of the CTCR tip, generating high-level commands, and converting them into appropriate joint velocities for removing the liquid. This is a simple test case with liquid in a single region. The semi-sphere located in the center of the

Two Spheres: This setup tests the controller's performance in handling a multi-region liquid scenario, where two separate regions of liquid must be removed. The controller needs to dynamically detect and handle each of the liquid areas during the task.

Terrain Flat: Initially, the liquid appears as one continuous pool. As the suction progresses, the liquid will flow into a semi-sphere with a larger radius. This tests the controller's ability to replan and adjust its strategy during the suction process as the liquid state changes.

Terrain Curved: The most complex scenario, in which the container is initially filled with liquid. After some of the liquid is removed, the remaining liquid pools in two semi-spheres with different depths. The controller must detect these changes and generate the appropriate commands to remove the remaining liquid, while ensuring stable and efficient movement.

Each container is a cube with a white, bone-like color and an edge length of 3 cm, with the diameter of each semi-sphere shown in Figure 4. One surface of each cube contains embedded semi-spherical holes varying in number, size, and position. The shape of the removed portion of the containers is inspired by bone drilling scenarios similar to a mastoidectomy during temporal bone surgery, e.g., for cochlear implantation [19].

Both simulation and real-world experiments were conducted using these four containers, with eight trials to

evaluate the generalizability of the controller. For the Bowl scenario, the container was placed at four distinct locations relative to the CTCR and camera. For the Two Sphere scenario, the container was rotated at four different angles (0° , 45° , 90° , and 180°) around its vertical axis. Similarly, for the Terrain Flat and Terrain Curved scenarios, both containers were rotated at four angles, increasing by 90° each time, to test the controller's generalizability.

B. Hardware and Robot Setup

Table I summarizes the configuration of the concentric tubes used in both experiments. In this table, L_i^s denotes the straight-section length of the tube, L_i^c the curved-section length, and κ the curvature. OD and ID refer to the tube's outer and inner diameters, respectively. The inner and middle tubes are fabricated from nitinol, whereas the outer tube is made of 304 stainless steel. All experiments were conducted on a workstation with an NVIDIA GeForce GTX 3070 GPU and an Intel Core i7-10700F CPU. Figure 5 illustrates the robot configuration for real-world experiments, and Section IV-D provides further details on the experimental setup.

Table I: CTCR Configurations for real-world and simulation experiment

Tube	outer	middle	inner
L_i^s	42	33.5	80.1
L_i^c	0	48.5	58.9
κ	0	0.0087	0.0066
OD	3.04	2.00	1.2
ID	2.30	1.80	0.8

All values in mm, κ in mm^{-1}

C. Simulation Experiment

The simulation experiments were conducted using a custom-built CTCR simulator developed in Unity. The controller receives RGB input from a Unity camera utilizing the Universal Rendering Pipeline (URP) and depth data from a secondary camera through the URP camera's depth buffer. Both cameras are fixed in position and orientation to ensure precise alignment of the RGB and depth data. The yellow ring marker has a width of 3 Unity units, and the green marker's width is 2 Unity units.

The camera's orientation is assumed to be known and remains constant throughout the suction process—a standard assumption in both simulated and real-world scenarios. Images are captured at a resolution of 720×1280 and are then cropped to match the input requirements of the segmentation model, resulting in 540×540 images. These cropped images are subsequently rotated to ensure the CTCR enters from the right side of the image and moves left toward the target. This orientation assumption is essential for the controller to function correctly in the blood suction controller.

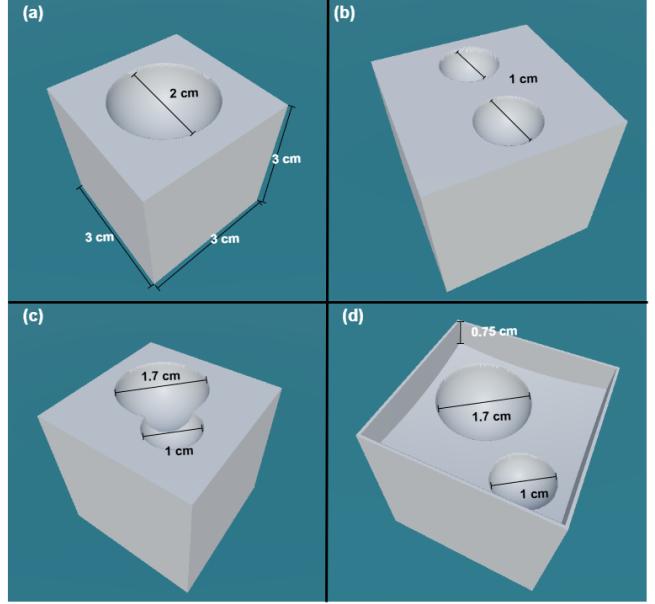


Figure 4: Four Phantom Shapes in Simulation. (a) Bowl, (b) Two Spheres, (c) Terrain Flat, (d) Terrain Curved



Figure 5: Real-world Experiment Setup

The containers are inside the CTCR's workspace, including the semi-spherical regions. Each container scenario undergoes eight trials, with two trials conducted for each unique position and orientation, as described in the previous section, to validate the controller's generalizability.

The performance of the controller is evaluated based on the following metrics:

- 1) Total Liquid Particles: The amount of liquid particles generated in the scene.
- 2) Removed Particles %: The amount of liquid left after the suction task.
- 3) Execution Time: The time taken to complete the task

Table II presents the results of the simulation experiments.

Table II: Simulation Experiment Results

Experiment	Total Liquid Particles	Removed Particles %	Execution Time (s)
Bowl	46166 ± 4698	99.84	27.7 ± 6.8
Two Sphere	12808 ± 21	98.62	40.5 ± 10.3
TerrainCurved	240501 ± 1503	99.63	72.0 ± 12.9
TerrainFlat	29853 ± 379	99.54	52.1 ± 18.0

Table III: Real-world Experiment Results

Experiment	Total Liquid Amount (g)	Removed Liquid %	Execution Time (s)
Bowl	1.5 ± 0.2	100	28.1 ± 4.0
Two Spheres	0.5 ± 0.1	100	48.2 ± 10.7
Terrain Curved	4.6 ± 0.4	97.8	77.5 ± 23.6
Terrain Flat	1.3 ± 0.1	99.1	54.0 ± 15.5

D. Real-world Experiment

For real-world testing, the same four container scenarios were recreated using 3D printing. The containers were fabricated using a Bambu lab A1 mini 3D printer and printed with white PLA material, and an Intel RealSense D405 depth camera was used to provide RGBD images for input to the model. The camera's exposure and white balance parameters were set to 40000 and 4080. The output RGBD images have 1280x720 resolution, then cropped at the center and reshaped into 540x540 to meet the CNN model input image size. The camera was mounted at a fixed position above the container, angled upward at -20 degrees about the x-axis in the camera frame, as shown in Figure 5 on the left side. This configuration ensured optimal visibility of the container surface and liquid regions. The CTCR was positioned along the positive y-axis of the camera's frame, with its tube directed towards the negative y-axis. This setup ensured consistent alignment with the controller's assumptions.

The captured images are preprocessed following a similar procedure described in Section IV-C, ensuring they meet the requirements for both the segmentation model and the controller. Liquid removal was performed using the Drive Medical Heavy Duty Suction Machine, with the inner tube of the CTCR inserted into the suction nozzle to maintain reliable suction performance. The CTCR actuation unit is described in previous publications [20]–[22]. Figure 5 illustrates the complete setup used for the real-world experiments.

In real-world experiments, the completeness of the suction is evaluated by measuring the weight of the liquid before and after suction using an electronic scale. The blood-like

liquid is simulated using water mixed with red food coloring. Similar to the simulation, the execution time and controller's output frequency are recorded during each trial. Table III provides a summary of the results.

V. RESULTS

In this section, we summarize the outcomes of both real-world and simulation experiments.

A. Performance in Simulation

For the simulation experiments, a total of 32 trials were conducted across four different container scenarios. On average, 99% of the liquid particles were successfully removed. Total running time varied by container complexity. In the simple Bowl scenario, where the CTCR removed liquid from one location, the task completed in under 30 seconds. In contrast, the more complex Terrain Curved scenario took about 72 seconds due to repeated homing and re-entry movements for re-planning and error correction.

During the suction process, the controller outputs commands at a frequency range of 4 to 7 Hz, including tasks such as image processing, model inference, decision-making, and CTCR controller computations. The re-planning features and their effects will be further discussed in Section VI.

B. Performance in Real-world

Across the total 32 trials, the remaining liquid was generally less than or equal to 0.1 g. Among the four container types, the Terrain Curved scenario has the highest amount of liquid remaining, typically around 0.1 g. This result is attributed to the larger initial liquid volume added to this container and the complex geometry, which caused the liquid to adhere to the curved surface.

Running times in real-world experiments were consistent with simulations. The Bowl scenario achieved the shortest completion time, while the Terrain Curved scenario averaged approximately 78 seconds, with delays arising from redundant homing commands when liquid was incompletely removed. Additionally, collisions between the CTCR and containers occurred during suction and homing, as our methods do not include collision avoidance strategies. However, the CTCR's low stiffness mitigated the impact of these minor collisions.

VI. DISCUSSION

The experimental results demonstrate the robustness of the decision-making unit in adapting to dynamic changes in liquid distribution during the suction process. As shown in Figure 6, the system effectively detects shifts in liquid position and replans the CTCR's trajectory to target new regions. This behavior highlights the capability of the decision-making unit to handle liquid dynamics and adaptively replan trajectories during the suction process.

However, the replanning feature can occasionally result in redundant movements. Factors such as camera occlusion

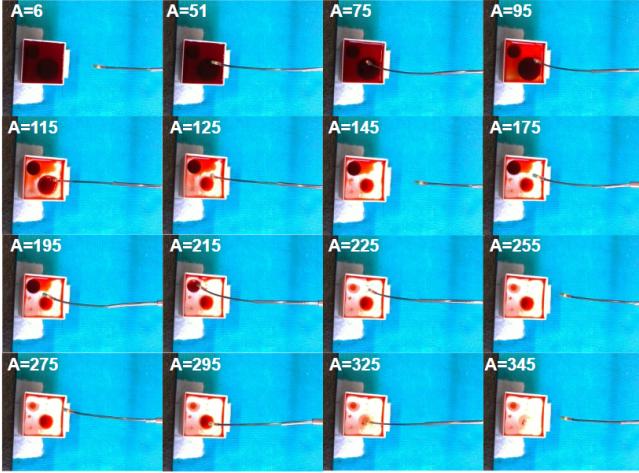


Figure 6: Sequential images from an experimental trial in the Terrain Curved container captured by a D405 camera. The number of A on the upper corner indicates the number of actions executed. Images have been cropped to enhance visibility

or incorrect segmentation may cause the system to enter undesired states, leading to inaccurate commands being issued to the robot. Despite these challenges, the incremental nature of the commands allows errors to be corrected in subsequent iterations. As a result, the suction task is completed successfully, although the running times will increase in some scenarios. For example, in the most complex container type, Terrain Curved, the average completion time was 77.5 ± 23.6 seconds, reflecting the impact of repeated homing and re-entry movements for error correction and re-planning. Our current approach favours task completion over speed, a trade-off which requires consideration in the future.

While the proposed method performs effectively in both simulation and real-world scenarios, several limitations arise from its underlying assumptions. First, the current liquid segmentation approach is color-based, assuming that no red-like pixels exist in the background. This assumption restricts the method's applicability in surgical contexts, where surrounding tissues often share a similar color to blood. Recent advancements in blood segmentation using learning-based methods demonstrate promising solutions, leveraging surgical videos to distinguish blood from irrelevant red regions. Integrating such models into the current framework could significantly enhance its applicability in diverse surgical environments. Secondly, camera occlusion can lead to incorrect or redundant movements, which substantially increase task completion times. Addressing this issue could involve incorporating additional cameras to provide multiple views of the suction scene, thereby reducing blind spots. Alternatively, enhancing the state space of the decision-making unit might mitigate the effects of occlusion, although

this would require more prior knowledge about the environment to ensure effective implementation. Lastly, challenges related to CTCR motion, including collisions and deviations near its workspace boundary, contribute to extended running times and limit broader usage. These issues could be mitigated through the integration of advanced motion planners, which would enable smoother trajectories while avoiding singular configurations and reducing collision risks. Such enhancements would be particularly valuable in scenarios with complex container geometries or edge-case end-effector positions, such as blood suction for in-vivo scenarios. Despite these limitations, the experimental results validate the method's robustness and generalizability, highlighting its potential for real-world blood suction tasks.

VII. CONCLUSION AND FUTURE WORK

This paper presented a novel approach for autonomous liquid suction using concentric tube continuum robot (CTCR), demonstrating its effectiveness in both simulation and real-world environments. Our controller leverages a vision-based framework that incorporates continuous segmentation of the CTCR body and tip through a fine-tuned ResNet18 model. Additionally, a hybrid strategy combining differential inverse kinematics with a pre-computed lookup table was employed to ensure stable and efficient motion.

In both simulated and real-world settings, the controller successfully adapted to dynamic changes in liquid states without requiring prior information about the container. This capability underscores the robustness and adaptability of our approach, allowing the controller to autonomously adjust its suction strategies based solely on real-time feedback, independent of predefined container geometries or surface characteristics.

Looking ahead, future work will focus on enhancing the simulation's physical accuracy to better replicate real-world conditions and refining the controller's ability to manage unpredictable liquid dynamics. We also aim to extend this framework to other surgical tasks involving fluids. Additionally, the simulation platform will be utilized to explore learning-based methods in more complex scenarios, such as blood suction in constrained workspaces with collision avoidance and other challenging tasks. Furthermore, human-robot interaction mechanisms could potentially be integrated into the current system to support expert supervision and intervention during the autonomous suction procedure.

REFERENCES

- [1] J. Yang, J. A. Barragan, J. M. Farrow, C. P. Sundaram, J. P. Wachs, and D. Yu, “An adaptive human-robotic interaction architecture for augmenting surgery performance using real-time workload sensing—demonstration of a semi-autonomous suction tool,” *Human Factors*, vol. 66, no. 4, pp. 1081–1102, 2024.
- [2] F. Richter, S. Shen, F. Liu, J. Huang, E. K. Funk, R. K. Orosco, and M. C. Yip, “Autonomous robotic suction to clear the surgical field for hemostasis using image-based blood flow detection,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1383–1390, 2021.
- [3] J. Huang, F. Liu, F. Richter, and M. C. Yip, “Model-predictive control of blood suction for surgical hemostasis using differentiable fluid simulations,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 12 380–12 386.
- [4] Y. Ou, A. Soleymani, X. Li, and M. Tavakoli, “Autonomous blood suction for robot-assisted surgery: A sim-to-real reinforcement learning approach,” *IEEE Robotics and Automation Letters*, vol. 9, no. 8, pp. 7246–7253, 2024.
- [5] B. Su, S. Yu, X. Li, Y. Gong, H. Li, Z. Ren, Y. Xia, H. Wang, Y. Zhang, W. Yao, J. Wang, and J. Tang, “Autonomous robot for removing superficial traumatic blood,” *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 9, pp. 1–9, 2021.
- [6] 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.
- [7] P. E. Dupont, N. Simaan, H. Choset, and C. Rucker, “Continuum robots for medical interventions,” *Proceedings of the IEEE*, vol. 110, no. 7, pp. 847–870, 2022.
- [8] C. J. Nwafor, C. Girerd, G. J. Laurent, T. K. Morimoto, and K. Rabenorosoa, “Design and fabrication of concentric tube robots: A survey,” *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 2510–2528, 2023.
- [9] I. Z. Yalniz, H. Jégou, K. Chen, M. Paluri, and D. Mahajan, “Billion-scale semi-supervised learning for image classification,” 2019.
- [10] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, “An open-source research kit for the da vinci® surgical system,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 6434–6439.
- [11] J. Lai, K. Huang, B. Lu, Q. Zhao, and H. K. Chu, “Verticalized-tip trajectory tracking of a 3d-printable soft continuum robot: Enabling surgical blood suction automation,” *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 3, pp. 1545–1556, 2022.
- [12] I. S. Godage, A. A. Remirez, R. Wirz, K. D. Weaver, J. Burgner-Kahrs, and R. J. Webster, “Robotic intracerebral hemorrhage evacuation: An in-scanner approach with concentric tube robots,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 1447–1452.
- [13] T. Okamoto, T. Ohnishi, H. Kawahira, O. Dergachyava, P. Jannin, and H. Haneishi, “Real-time identification of blood regions for hemostasis support in laparoscopic surgery,” *Signal, Image and Video Processing*, vol. 13, no. 2, pp. 405–412, 10 2018.
- [14] P. Iakubovskii, “Segmentation models pytorch,” https://github.com/qubvel/segmentation_models.pytorch, 2019.
- [15] J. Burgner, D. C. Rucker, H. B. Gilbert, P. J. Swaney, P. T. Russell, K. D. Weaver, and R. J. Webster, “A telerobotic system for transnasal surgery,” *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 3, pp. 996–1006, 2014.
- [16] P. E. Dupont, J. Lock, and B. Itkowitz, “Real-time position control of concentric tube robots,” in *IEEE International Conference on Robotics and Automation*, 2010, pp. 562–568.
- [17] S. Shumikhin, “Tubegenerator,” <https://github.com/sshumihin/TubeGenerator>, 2019.
- [18] N. Yughues, “Yughues free metal materials,” <https://assetstore.unity.com/packages/2d/textures-materials/metals/yughues-free-metal-materials-12949>, 2021.
- [19] C. Yuan, D. Shin, T. N. Le, V. Y. Lin, J. M. Chen, L. A. Kahrs, and J. T. Lui, “Instrument segmentation with ternausnet during temporal bone surgery,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 18, pp. S49–S50, 2023.
- [20] V. Modes, S. Ihler, A. Nabavi, T. Ortmaier, L. A. Kahrs, and J. Burgner-Kahrs, “Towards concentric tube robots for microsurgery: First results in eye-to-hand visual servoing,” *The Hamlyn Symposium on Medical Robotics*, p. 77–78, 2018.
- [21] R. M. Grassmann, R. Z. Chen, N. Liang, and J. Burgner-Kahrs, “A dataset and benchmark for learning the kinematics of concentric tube continuum robots,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 9550–9557.
- [22] R. Grassmann, V. Modes, and J. Burgner-Kahrs, “Learning the forward and inverse kinematics of a 6-dof concentric tube continuum robot in $se(3)$,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 5125–5132.