Soft Robot Motion Simulation in 2D Vascular-mimicking Network

Mingzhang Zhu, Zhengqi Zhong, YaoHsing Tseng

***Abstract*— Soft continuum robots with active steerable probes capable of navigating in a complex vascular network hold great promise for medical applications. This paper reports a simulation of a soft robot controlled by an external magnetic field to navigate inside a 2D vascular network. We established a DER model for the soft robot and successfully controlled the equivalent force exerted by the magnetic field to follow the desired trajectory. A new algorithm is also developed in this paper to describe the robot kinematics with the effect of contact force. Future work includes iteration of vascular mimicking environment, animation rendering, adding friction between walls & robot, and refinement of robot model.**

I. INTRODUCTION

Robot devices constructed with soft and flexible materials have desirable traits for medical applications, including limb rehabilitation, soft robot-assisted ultrasound imaging, medical procedures, etc. [1–3]. Minimally Invasive Surgery (MIS) utilizes small-scale surgical instruments inserted into the body through small incisions to access the target anatomies. Soft robotics devices hold great promise, particularly for MIS, such as eliminating thrombosis or lesions located at hard-to-reach anatomies due to their flexibility and low force execution [4]. Therefore, exponentially efforts have been made to implement soft robots in intracorporal procedures to reduce the incision area, recovery time and enhance patient safety. However, traditional soft robotics devices such as flexible endoscopes with rigid distal tip, which is commonly used, can only treat lesions located on the straight path and aimed against the endoscope [5]. The limited access area of the device hinders the application of this approach.

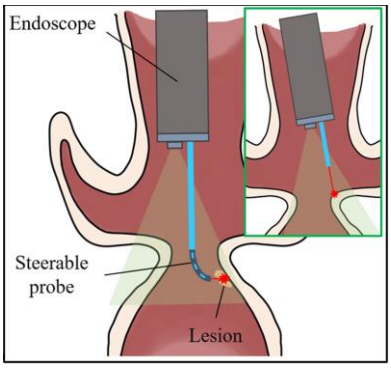

Figure 1. Schematic of a steerable probe that is able to treat lesions facing downward and inside laryngeal ventricles. (Inset) Schematic of a non-steerable endoscope with the forward-looking probe that can only reach tissue areas facing up.

Constructing an actuated distal steerable mechanism is a potential solution to tackle the above challenge. Recent work on the simulation of a steerable probe demonstrated the capability to access more than 70% larynx area than along-the-axis laser in a non-steerable fiber probe [6], which indicates that the steerable probe can treat some the patients that cannot be treated otherwise, as shown in Fig. 1. Despite the advantages of such a mechanism, several technical challenges blocked the clinical application. One major challenge is the large bending radius of the steerable probe due to the actuation mechanism, which results in a limited steering angle of the distal probe. Current research focuses on two types of actuation methods of the distal steering devices: 1) Tendon-driven continuum robot, where its steerable probe is driven by plastic wires embedded within the tube-like structures [7]. 2) Magnetically steerable continuum robot, where the steerable tip embedded with a tiny magnet deformed under external magnetic fields [8]. The cross-section radius of the latter mechanism can be much miniaturized due to its more straightforward structure, which results in greater flexibility compared with the tendon-drive one.

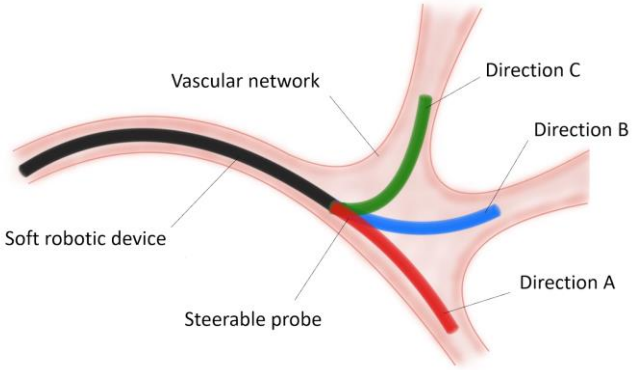

Figure 2. Illustration of an active steerable probe of a soft continuum robot navigating within a complex vascular network. Three configurations are listed to demonstrate different bending angles of the steerable distal tip.

Recent research on a soft robot utilizing a magnetically driven steerable probe has demonstrated its potential in the application of cardiac intervention, where its steerable probe was capable of navigating within a mimicked coronary artery [9]. However, due to the complexity of the real vascular structures with different radii of vessels, the real clinical implementation of this robot is still being investigated. Therefore, simulation of a soft robot navigating to the desired position within a constrained and complex environment is critical for the surgeons to determine the most suitable treatment for patients, as shown in Fig. 2. Additionally, the soft robot motion simulation paves the way for clinical application, especially in MIS.

This paper presents the simulation of a soft robot device with a magnetically driven steerable tip capable of navigating through a complex 2D vascular-mimicking environment to the desired position. An elastic rod made of PDMS is adapted to

simulate the soft robots, while a two-dimensional maze with a size of 1m*1m is adapted to simulate the vascular network. The magnetically driven steerable distal probe is simplified as an external force implemented on a specific node of the elastic rod. A PD controller adjusts the rod motion to follow desired trajectories. We demonstrate a simulation of controlling the first node with an exerted force to run through the maze. With the integration of the DER model, the whole robot would move through the maze with the guidance of the magnetic force. A new algorithm is also developed in this paper to describe the robot kinematics with the effect of contact force. Future work includes iteration of vascular mimicking environment, animation rendering, adding friction between walls & robot, and refinement of robot model.

II. Design and Method

A. Vascular-mimicking Environment Model

To simulate the motion of a soft robot with a steerable probe navigating within a vascular network, we characterized both the soft robot and the vascular network with several conditions and assumptions. We only considered the robot's movement in a two-dimensional plane to simplify the construction of the virtual environment and reduce computational complexity. We created the 2D vascular mimicking environment in MATLAB with a size of 1m * 1m. To simulate the navigation capability of the steerable probe at an aneurysm, the intersection of several blood vessels, and the bending radius of the distal end, the 2D virtual network is designed to look like a maze with a random number of turns. The width of the channel is set to be 0.1m.

In the first stage of this virtual environment, all corners are designed to have a bending angle of 90 degrees. The starting point and ending point are located diagonally at the top-right and bottom-left corners of the maze. We also wrote a wall-follower algorithm to find the most direct path from the starting point to the endpoint, which is located at the centerline of every channel. This direct path is the ideal trajectory of the soft robot. The stiffness of the wall, as well as the friction and physical collision between the wall and the robot, are neglected for now.

B. Soft Continuum Robot Model

The soft continuum robot with a magnetically driven distal steerable probe is modeled as DER, a computation algorithm for rod-like object physically based simulation. With this approach, we can simulate a robot's bending, stretching, and twisting in a torturous environment such as a blood vessel. The modeled rod is assumed to be made of Polydimethylsiloxane (PDMS), a commonly used material in the soft robotics field. The material properties of PDMS and the robot dimensions that would be used in the simulation are shown in Table 1 below [10,11]. The stiffness and friction coefficient of the PDMS would be induced in the next step to model the physical intersection with the virtual blood vessels. Fig. 3 shows the magnetically responsive distal tip of the soft robot resulting from the embedded tiny magnet under a vertical magnetic field. We characterized the magnetic effect as a force F exerted on a node at the distal tip while the rest of the node on the rod can be moved in free space.

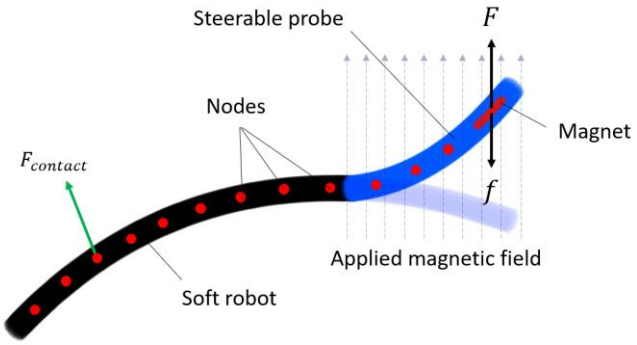

Figure 3. Schematic of the soft robot with the steerable probe under face-up vertical magnetic field. A magnet embedded in the frontal section of the distal tip produces force F that deflect the shape of the soft robot. Fluid resistance is represented by f, which is aligned to force F with the opposite direction.

TABLE I. Robot Model Parameters

Parameter	Value
Young's Modulus	1.47 MPa
Poisson's Ratio	0.48
Shear Modulus	0.497 MPa
Density	980kg/m^3
Length	0.5m
Outer Radius	2mm
Inner Radius	1mm

We induced fluid resistance to describe the friction when the soft robot passes through blood. The fluid resistance is assumed to be aligned with F in the opposite direction, which is defined as the equation shown below:

$$f = -\frac{1}{2}\rho v^2 A C_d$$

- ρ refers to the density of blood, which is approximately 1025 kg/m^3[12].
- v refers to the speed of a moving soft robot.
- A refers to the frontal area of the robot pushing through the blood. For our basic simulation, we consider this area to be equal to the cross-sectional area of the rod.
- C_d refers to the coefficient of drag, which is approximately 1.0536 [13].

From Newton's second law, the equation of motion for this system is described in the equation shown below:

$$f_i = m_i \frac{q_i(t_{k+1}) - q_i(t_k)}{dt^2} - m_i \frac{\dot{q}_i(t_k)}{dt} + \frac{\partial}{\partial q_i}\left(E_k^s + E_k^b\right) + F + f + F_{contact}$$

- E_k^s refers to the stretching energy (Notes, equation 7.8).
- E_k^b refers to the bending energy (Notes, equation 7.10).
- f refers to the fluid resistance, which is derived in the above sections.

- F refers to the magnetic force, which is assumed to be exerted on a section of nodes near the distal end of the steerable probe.
- $F_{contact}$ refers to the contact force matrix of each node between the wall and the robot

As discussed in the class, Newton-Raphson iteration is implemented to solve the equation of motion at each time step. We also implemented the time marching method to compute the time evolution of the position of each node. The motion of the rod is controlled by changing the magnitude and direction of the force F in each time interval. In order to simplify the rod's movement, the force F is set to be equal to the fluid resistance so that the node with exerted force F would perform the uniform rectilinear motion.

C. Trajectory Following with Decentralized Feedforward PD Position Control

The trajectory of this soft continuum robot is indirectly controlled by the external magnetic field acting on the steerable probe part. By changing the direction and magnitude of the magnetic field, the movement of the robot's tip can be precisely controlled and follows a predefined trajectory. With the predefined 2-D maze and maze solution, the team generated a desired trajectory for the robot's tip to follow. In order to generalize this simulation and mimic the real-world situation, a sine wave disturbance with 2-unit Amplitude and 10 rad/sec frequency is added to the robot model, which would cause some offset between the robot and desired trajectory.

$$Disturbance = 2 * \sin(5 * t)$$

The team decided to implement the Decentralized Feedforward Position and Velocity Control on a single node of the robot's tip, track the desired trajectory, and compensate for disturbance. The general principle of this method is to compute the error values between the desired position, velocity, and real position and velocity. Then The team would apply a Proportional and Derivative control (PD control) to compensate for these errors and add them together with the desired trajectory. As a result, the real trajectory would slightly oscillate around the desired trajectory with proper proportional and derivative gain values.

D. Contact Force Characterization

Our contact force calculation implemented an iteration method inside the DER simulation process. When the DER first guess the positions for all the nodes, each node would be checked to see whether they collide with the boundary. If yes, the Contact Force Function would compute the distances between each node and the edges in X and Y directions. Then these distances would multiply a Step Force Gain Value(K_f). This Gain Value can be considered the step force increment related to the distance in each iteration. Therefore, a smaller K_f value can increase accuracy but require higher computational power. On the other hand, larger K_f computes faster but may be less accurate.

The contact force equation is shown below:

$$Force\ x\ =\ Force\ x\ +\ K\_f * dy$$

$$Force\ y\ =\ Force\ y\ +\ K\_f * dx$$

The smaller distance in the X or Y direction means a larger force is required in this direction. Therefore, the Force in the X direction should time dy, and Force in the Y direction should times dx. As a result, the nodes inside the boundary would be pushed a little bit outward during each iteration until all the nodes leave the edge, and the total contact forced also been computed during this process.

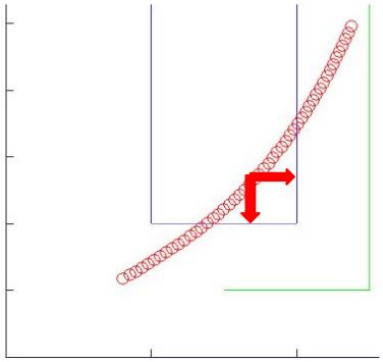

Figure 4. Decentralized Feedforward Position and Velocity Control Block Diagram with Sine Wave Disturbance.

E. Simulation Workflow

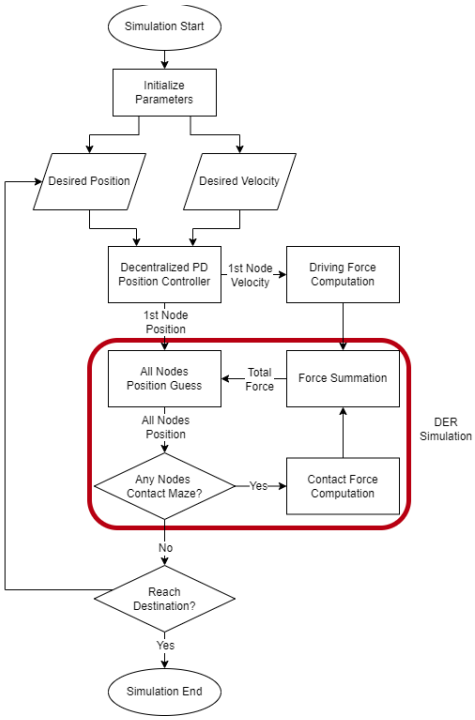

Figure 5. Flow chart of the complete simulation process.

With the models and methods described in the previous section, the complete flowchart of our simulation is shown in Fig. x below. The desired velocity and position of the first node can be determined with a given maze and trajectory. The PD controller would calculate the updated position and velocity, which is required to find the external force. These two values would enter the DER simulation module, including contact force determination. The DER simulation would compute the actual position of the whole robot and feed the first node position back to the PD controller to calculate the node's

position in the next timestamp. The detailed composition of each module is discussed in the algorithm section.

III. Algorithm

Three main modules are included in this simulation: Decentralized PD controller for controlling the first node, Actuation force computation, and DER simulation module. We will discuss the principle of each module and their connections in the following paragraphs.

A. Decentralized PID Controller

Algorithm 1 shows the steps of computing the velocity, velocity error, and updated position of the first node. With the desired position and velocity, the actual position, and velocity from the last timestamp, the error values of position and velocity can be computed. We would apply PD control with appropriate K_p and K_d values to compensate for the error and add to the desired trajectory. The velocity and velocity error would be used to compute the actual actuation force, while the updated position of the first node would be used to guide the rest nodes within the DER simulation. The Simulink subsystem of this PID controller is shown in Fig. 6.

Algorithm 1 First Node Decentralized PD Controller

Require: K_p, K_d ▷ Gain values
Require: $v1(t_{i-1})$
Require: $q_{1\_update}(t_{i-1})$
Require: $ref\_q1(t_i), ref\_v1(t_i)$ ▷ Desired position and velocity
Ensure: $F_{1st\_Node}, q_{1\_new}(t_i), v1(t_i)$
function FIRST_NODE_CONTROL($F_{1st\_Node}, q_{1\_new}(t_i), v1(t_i)$)
  for $k \leftarrow 1 : t_{end}$ **do**
    Compute $ref\_F1(t_i)$
    $pos\_error \leftarrow ref\_q1(t_i) - q_{1\_update}(t_{i-1})$
    $vel\_error \leftarrow ref\_v1(t_i) - v1(t_{i-1})$
    Compute $extra\_F1(t_i)$
    $F_{1st\_Node} = ref\_F1(t_i) + extra\_F1(t_i)$
    $u \leftarrow pos\_error \times K_p + vel\_error \times K_d$
    $q_{1\_new}(t_i) \leftarrow ref\_q1(t_i) + u$
    $v1(t_i) \leftarrow (q_{1\_new}(t_i) - q_{1\_update}(t_{i-1}))/dt$
  end for
  return $F_{1st\_Node}, q_{1\_new}(t_i), v1(t_i)$
end function

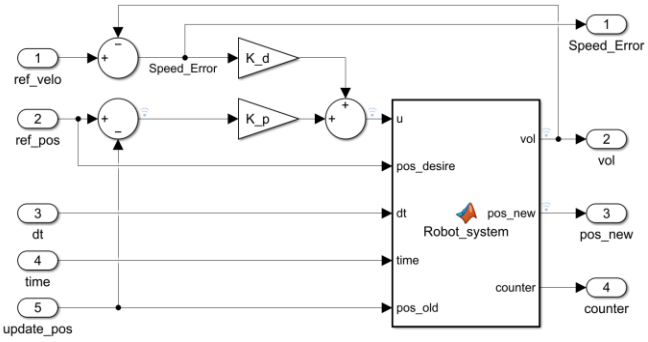

Figure 6. PD controller subsystem of controlling the first node position built in Simulink.

B. Actuation Force Computation

To calculate the required force vector to keep the first node of the robot moving on the desired trajectory, the reference velocity and velocity error computed in the PD controller is needed. The relationship between the velocity and force is described in the soft continuum robot section. The forces computed by two velocities would be added to be the real actuation force which would be used to be a critical component of the external force for DER simulation. Fig. 7 below shows the block diagram of the actuation force computation module in Simulink.

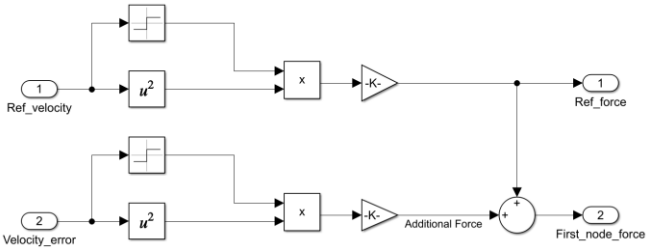

Figure 7. Actuation forces a computation module to be built in Simulink.

C. DER Module

Algorithm 2 DER Calculation

Require: F_{1st_Node} ▷ 1st Node Force Calculated from PD Controller
Require: EA, EI ▷ Elastic stiffness
Require: M ▷ Mass Matrix
Require: $q_{1\_new}(t_i)$ ▷ 1st Node Position Calculated from PD Controller
Ensure: $q(t_i), q_{1\_update}(t_i)$
Ensure: $F_{contact}(t_i)$
function DER_SIMULATION($q(t_i), q_{1\_update}(t_i), F_{contact}(t_i)$)
  for $k \leftarrow 1 : t_{end}$ **do**
    Guess: $q(t_i) \leftarrow q(t_{i-1})$
    $q(t_i)(1:2) \leftarrow q_{1\_new}(t_i)$ ▷ Update 1st node position
    $u \leftarrow (q(t_i) - q(t_{i-1})/dt$
    while $error > tol$ **do**
      Compute F and J
      for $k \leftarrow 1 : N-1$ **do** ▷ Stretching Energy
        $ind \leftarrow$ locations of x_k, x_{k+1}
        [dF, dJ] $\leftarrow$ gradEs_hessEs(x_k, x_{k+1})
        F(ind) $\leftarrow$ F(ind) + dF
        J(ind) $\leftarrow$ J(ind, ind) + dJ
      end for
      for $k \leftarrow 2 : N-1$ **do** ▷ Bending Energy
        $ind \leftarrow$ locations of x_{k-1}, x_k, x_{k+1}
        [dF, dJ] $\leftarrow$ gradEb_hessEb(x_{k-1}, x_k, x_{k+1})
        F(ind) $\leftarrow$ F(ind) + dF
        J(ind) $\leftarrow$ J(ind, ind) + dJ
      end for
      Compute $F_{contact}(t_i)$ ▷ See Contact_Force Calculation
      $Ex\_Force(1:2) \leftarrow F_{1st\_Node}$ ▷ Actuation force
      $Ex\_Force(t_i) \leftarrow Ex\_Force(t_i) + F_{contact}(t_i)$
      $F(t_i) = F(t_i) + Ex\_Force(t_i)$
      $\delta q(t_i) \leftarrow J_{free} \backslash F_{free}$
      error $\leftarrow sum(abs(F_{free}))$
    end while
    $q_{1\_update}(t_i) \leftarrow q(t_i)(1:2)$
  end for
  return $q(t_i), q_{1\_update}(t_i), F_{contact}(t_i)$
end function

Algorithm 2 shows the steps to compute the actual position of each node and the external force matrix of the DER simulation. This Newton-Raphson iteration of DER was modified from the one professor discussed in class. Instead of using the position from the previous timestamp, the position of the first node would be updated by the q1_new(ti). Besides, a switch and a memory module were used in Simulink to store the position of the previous timestamp except the initial configuration. The contact force computation is embedded in the iteration so that the contact force required to push the node out of the wall can be computed for every node in each timestamp. Then both of contact force and actuation force would be added to the external force matrix in the equation of motion to calculate nodes' position. The actual position of the first node would be sent back to the PD controller to execute

simulation for the next timestamp. Fig. 8 below shows the block diagram of the DER computation module in Simulink.

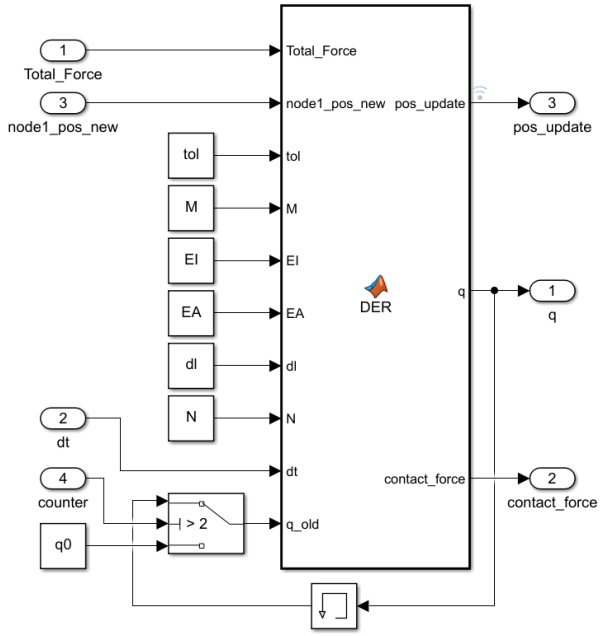

Figure 8. Soft Continuum Robot tip node trajectory following, with total magnetic force direction and magnitude applied to the node.

D. System Structure

Fig. 9 below shows the high-level system structure and how the three subsystems described above communicated to perform simulation. For DER, the force required to navigate through the anatomies is computed in the actuation force module, while the position of the first node is computed in the PD controller module. The contact force is self-recursive within DER to calculate the actual external force matrix. The output of the whole simulation includes the real position of the robot, actuation force, and contact force for further analysis.

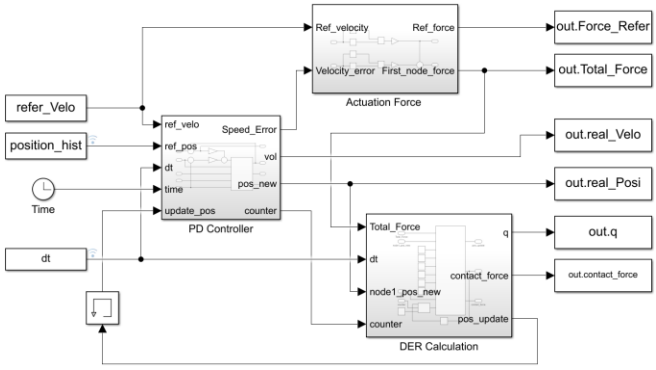

Figure 9. High-level block diagram of the whole simulation with inputs and outputs.

IV. Result

A. 2D Vascular-mimicking Maze

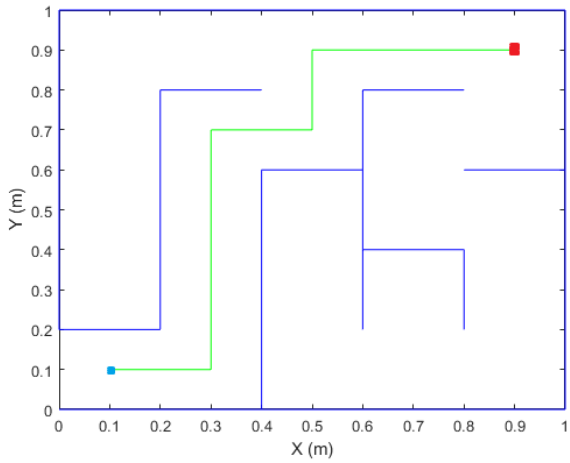

Figure 10. Illustration of the random-generated 2D vascular-mimicking maze. Red dot represents the starting point, and the blue dot represents the end point. Blue lines represent the boundary wall; Green line represents the most direct path from the starting point to the end point.

The first step of our simulation is creating a 2D maze with a size of 5*5 blocks to mimic the blood vessels' environment, then finding the path to solve this maze. As shown in the figure above, the blue dot represents the starting point, red is the ending point, and the green lines are the solution for this maze.

B. Trajectory Following Simulation of a Single Node

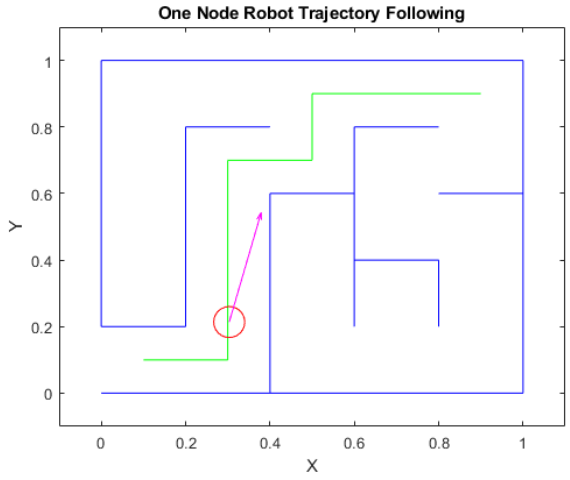

Figure 11. Soft Continuum Robot tip node trajectory following, with total magnetic force direction and magnitude applied to the node.

By implementing a Feedforward PD position controller on the first node of our robot, the node can follow the desired trajectory very well, even with a certain random disturbance that we assigned to the robot model. During this process, we assume the robot is moving at constant speed in each line segment with constant friction force by the environmental blood viscosity. The arrow line represents the external driven force applied by the magnetic field that is needed to control the robot's movement.

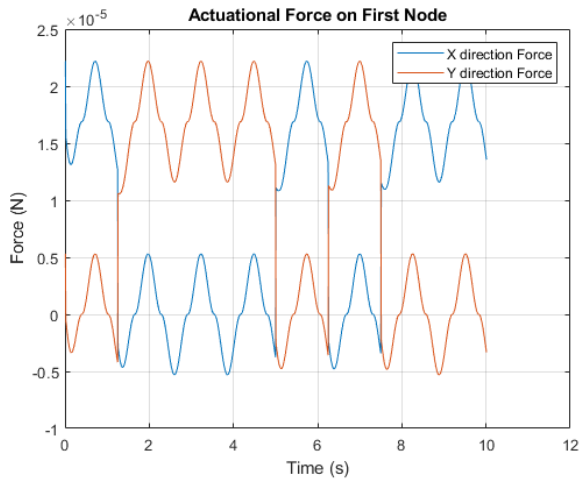

Figure 12. Magnetic Actuation Force acted on Soft Continuum Robot tip node Actuation Force.

The figure above shows the force required from the external magnetic field to act on the magnet, which is embedded in the first node of the Soft Continuum Robot. The force's shape makes sense since we added a sine wave disturbance into the robot model.

C. *Discrete Elastic Rod Trajectory Following*

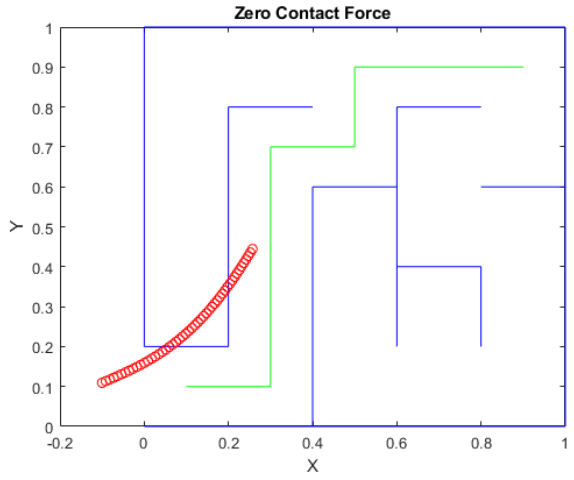

Figure 13. Soft Continuum Robot whole body simulation by DER algorithm (without Contact Force) in blood vessels environment.

After getting the first node position and the magnetic field's required driving force, we can implement the DER algorithm method to compute the elastic and bending forces and obtain the position for the rest of the robot body. However, we can notice that the DER method does not consider the contact between the robot body and the boundary (blood vessel walls). Therefore, we need to include the contact force computation inside our DER computation process.

D. *Discrete Elastic Rod Simulation (With Contact force)*

After adding the Contact Force computation into our DER algorithm, the robot body can "feel" the wall and curve at the place where it touches the wall. At this stage, this continuum robot simulation can mimic the performance in the ideal environment.

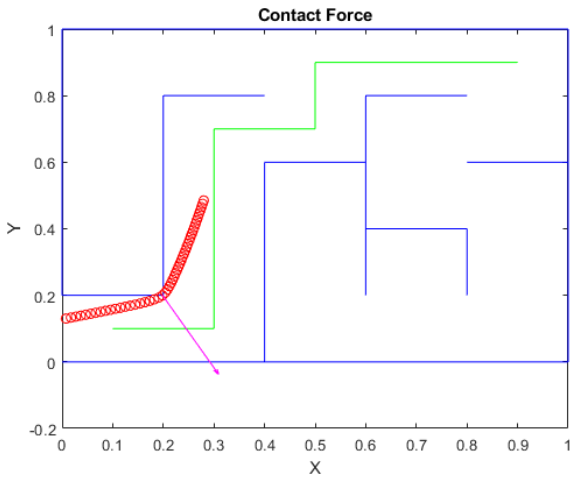

Figure 14. Soft Continuum Robot whole body simulation by DER algorithm (with Contact Force) in blood vessels environment. Arrow line represents the Contact Force direction and magnitude.

V. Discussion & Ongoing Work

A. *Soft Robot Model Refinement*

The current soft robot model was simplified so that we could quickly verify the feasibility of our simulation method, which can be further refined mainly in two ways. We assumed that in each force direction, the contact area of the fluid resistance remains the same and equal to the cross-sectional area of the rod. In the actual case, the fluid resistance would become larger while the angle between the looking-forward direction and the force direction becomes larger. Additionally, displacement restrictions between each node can be added to improve the accuracy of the simulation.

We would also recalculate the magnetic force that was exerted on the robot head. We assumed that the magnetic force is a single vector that acts on the robot in the currently existing system. Though it works perfectly for the navigation algorithm, the result will still be different if we apply a magnetic field, which would be the real-life scenario, on the robot. By implementing the field concept into our simulation, the result would be more intact and much closer to reality.

B. *Contact Force*

It is critical to consider the collision and friction between the soft robot and the inner wall of the blood vessels in clinical application. The position of each node at every timestamp would be stored and checked whether the robot has physical interaction with the wall. Node positions are also used to calculate the friction and normal force indirectly. The additional force produced during the rod movement would be added to the equation of motion in the next timestamp to determine the actual pose of the rod. Other parameters such as friction coefficient and the stiffness of both rod and walls would be included to calculate the resultant force.

C. *Vascular-mimicking Environment Iteration*

For the first iteration, the vascular-mimicking maze only includes walls in the horizontal and vertical directions. For the soft robot model, this maze could only test a scenario of a 90-degree bending angle. In the next iteration, a vascular network

with sharp turns and multiple-channel-intersections will be developed to mimic the real vascular network, and We will also approach three-dimensional computation. The result will show more detail about the interaction between the robot and the environment, which cannot be foreseen in the current simulation.

Constructing a sectional model of a real human vascular network is another option to simulate the real environment further. Even with the specific design, there are still some differences between the maze and the real human vascular. A large gap between real-world scenarios and the simulation would make it impossible to optimize the robot system to finish the tasks that its users, such as doctors, assign. In that case, simulating the soft robot in the environment that formed on real human blood vessels becomes necessary. Therefore, the further improvements to the robot system would be based on the blood-vessels simulation results.

D. Animation Rendering

Animation rendering will be applied to visualize the result to help with improving the robot system. The environment in which the soft robot is deployed contains blood vessels and highly fragile soft tissues. Therefore, the development of the soft robot guiding system would need doctors' and experts' cooperation. However, the system would easily be mistranslated by its reader with only calculation results. This error would let the patients take the fatal risk of going under their operation. As a result, animation rendering becomes a crucial part of the simulation. With this technique, doctors can easily understand the whole story behind the design and outline the additional specific requirements for the robot. Hence, the engineer can refine the origin model to eliminate potential issues and develop higher-level control of the robot to make it easier to use, which helps pave the way for wider implementation of the soft robot in clinical applications.

REFERENCES

[1] Zion Tsz Ho Tse et al., Soft Robotics in Medical Applications, Journal of Medical Robotics Research, doi: 10.1142/S2424905X18410064

[2] B. Vucelic et al., "The Aer-O-Scope Proof of Concept of a Pneumatic Skill-Independent, Self-Propelling Self- Navigating Colonoscope," Gastroenterology, vol. 130, no. 3, pp. 672-677, 2006.

[3] H. G. Ren, X. & Tan, K. L., "Human-Compliant Body-Attached Soft Robots Towards Automatic Cooperative Ultrasound Imaging," 2016 20th IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD 2016), 2016.

[4] Mark Runciman, Ara Darzi, and George P. Mylonas.Soft Robotics.Aug 2019.423-443. http://doi.org/10.1089/soro.2018.0136

[5] M. Zhu, Y. Shen, A. J. Chiluisa, J. Song, L. Fichera and Y. Liu, "Optical Fiber Coupling System for Steerable Endoscopic Instruments," 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 2021, pp. 4871-4874, doi: 10.1109/EMBC46164.2021.9629658.

[6] I. A. Chan, J. F. d'Almeida, A. J. Chiluisa, T. L. Carroll, Y. Liu, and L. Fichera, "On the merits of using angled fiber tips in office-based laser surgery of the vocal folds," in Medical Imaging 2021: Image-Guided Procedures, Robotic Interventions, and Modeling, vol. 11598, p. 115981Z, 2021.

[7] D. B. Camarillo, C. R. Carlson, J. K. Salisbury, Configuration tracking for continuum manipulators with coupled tendon drive. IEEE Trans. Robot. 25, 798–808 (2009).

[8] M. P. Armacost, J. Adair, T. Munger, R. R. Viswanathan, F. M. Creighton, D. T. Crud, R. Sehra, Accurate and reproducible target navigation with the Stereotaxis Niobe® magnetic navigation system. J. Cardiovasc. Electrophysiol. 18, S26–S31 (2007).

[9] A. K. Hoshiar, S. Jeon, K. Kim, S. Lee, J.-Y. Kim, H. Choi, Steering algorithm for a flexible microrobot to enhance guidewire control in a coronary angioplasty application. Micromachines 9, 617 (2018).

[10] Johari, Shazlina & Shyan, L.. (2017). Stress-strain relationship of PDMS micropillar for force measurement application. EPJ Web of Conferences.

[11] Shim, Sang & Yashin, Victor & Isayev, Avraam. (2004). Environmentally-friendly physico-chemical rapid ultrasonic recycling of fumed silica-filled poly(dimethyl siloxane) vulcanizate. Green Chemistry - GREEN CHEM.

[12] Kenner, T. The measurement of blood density and its meaning. Basic Res Cardiol 84, 111–124 (1989). https://doi.org/10.1007/BF01907921

[13] DAVIS, A. M. J., and K. B. RANGER. "A STOKES FLOW MODEL FOR THE DRAG ON A BLOOD CELL." Quarterly of Applied Mathematics, vol. 45, no. 2, 1987, pp. 305–11