A Cross-Entropy Motion Planning Framework for Hybrid Continuum Robots

Jibiao Chen, Junyan Yan, *Student Member, IEEE*, Yufu Qiu, Haiyang Fang, Jianghua Chen, and Shing Shin Cheng, *Member, IEEE*

***Abstract*—The sampling-based motion planners, including the Rapidly-exploring Random Trees (RRT) algorithms, are widely utilized in continuum robots, enabling efficient search for feasible motion plans in constrained environments. In surgical robotics, complex mapping among the high-dimensional kinematics of continuum robots, trajectory parameterization, and path redundancy may lead to non-optimal motion path, which in turn affects their efficiency and surgical task performance (e.g. path following), and ultimately the patient outcome. In this letter, a cross-entropy (CE) motion planning framework is proposed for continuum robots, wherein the RRT* planner is equipped with a CE estimation method serving as a probabilistic model to sample elite trajectories with optimal computation costs. It can asymptotically optimize the sampling distributions among individuals in terms of either robot states or parameterized trajectories. The presented CE motion planners were implemented on a hybrid continuum robot to enable obstacle avoidance, approximate follow-the-leader (FTL) motion, and navigation in a clinical scenario. They are shown to offer lower sampling cost and higher computational efficiency compared to existing approaches.**

***Index Terms*—Continuum robots, sampling-based motion planning, cross-entropy (CE) motion planning, Rapidly-exploring Random Trees (RRT*) approach.**

I. Introduction

MOTION planning for continuum robots is important to ensure a safe motion trajectory to reach the desired spatial target in constrained environments. This is mainly due to their complex kinematic and shape constraints when interacting with the environmental obstacles. In the case of surgical robotics, in particular, continuum robots are usually adopted to navigate in an uncertain and narrow environment with multiple anatomical constraints that must be taken into account to achieve real-time, precise task execution [1]. Thus, motion planning problems for continuum surgical robots should be solved with high computational efficiency and low sampling costs.

Among the many motion planners, sampling-based methods are known to offer relatively low computational complexityincluding Probabilistic RoadMaps (PRM) [2] and Rapidly-exploring Random Trees (RRT) [3]. The traditional RRT-like algorithms can be evolved to other variants, such as RRT-connect [4], RRT* [5], bidirectional-RRT [6], and informed RRT* [7]. Compared to PRM and RRT approaches, RRT* is uniquely capable of achieving asymptotic optimality and anytime solution (i.e. capable to be continuously optimized [5]) with minimum computational requirements in a single-query problem, leading to more adoption for RRT* in robotics.

Sampling-based motion planning approaches have also been proposed for continuum surgical robots comprising mostly three design architectures in the form of steerable needles, concentric tube robots, and multi-segment continuum robots with active segment bending. Patil et al. [8] presented a fast motion planner for flexible needle steering to avoid obstacles and handle uncertainties using the RRT approach, and extended studies were proposed for steerable needles respectively using backward planners [9], cost maps [10], and parallel sampling [11]. Additionally, derived from RRT*, path planners with optimized geometric Hermite curves [12], heuristic search [13], and an evolutionary sampling-based algorithm [14] were proposed for needle steering, leading to smooth path approximation and high computational efficiency. Compared to steerable needles, of which the needle shape passively follows the tip path in a soft tissue environment, the shapes of concentric tube robots and multi-segment continuum robots can actively and significantly change via internal forces from their respective actuation inputs. Therefore, the nonlinear robot shape constraint becomes another important consideration for motion planning problems for these continuum robots. A RRT-Shape approach was presented for motion planning of concentric tube robots [15], to perform an approximate follow-the-leader (FTL) motion. However, it can result in the solution getting trapped in local optima during path sampling without considering the nonlinear behavior of kinematics. The Rapidly-exploring random graph algorithm was presented [16] to perform a fast motion planner for concentric tube robots, and its node graph is a superset of the RRT graph [3]. In multi-segment continuum robots with active segment bending,

Manuscript received 8 June 2023; accepted 9 October 2023. Date of publication 18 October 2023; date of current version 1 November 2023. This letter was recommended for publication by Associate Editor L. Fichera and Editor P. Valdastri upon evaluation of the reviewers' comments. This work was supported in part by the Research Grants Council (RGC) of Hong Kong under Grant CUHK 24201219 and CUHK 14217822, in part by the Innovation and Technology Commission of Hong Kong under Grants ITS/136/20, ITS/234/21, and ITS/233/21, in part by Multi-scale Medical Robotics Center, InnoHK, Hong Kong, in part by The Chinese University of Hong Kong (CUHK) Direct Grant, and in part by the Shun Hing Institute of Advanced Engineering, CUHK. The content is solely the responsibility of the authors and does not necessarily represent the official views of the sponsors. *(Corresponding author: Shing Shin Cheng.)*

Jibiao Chen, Junyan Yan, Yufu Qiu, Haiyang Fang, and Jianghua Chen are with the Department of Mechanical and Automation Engineering and T Stone Robotics Institute, The Chinese University of Hong Kong, Hong Kong (e-mail: jibiaochen@foxmail.com; junyanyan@mae.cuhk.edu.hk; yufuqiu@outlook.com; haiyang.fang@link.cuhk.edu.hk; jhchen@mae.cuhk.edu.hk).

Shing Shin Cheng is with the Department of Mechanical and Automation Engineering and T Stone Robotics Institute, The Chinese University of Hong Kong, Hong Kong, and also with the Shun Hing Institute of Advanced Engineering and Multi-Scale Medical Robotics Center, Hong Kong (e-mail: sscheng@cuhk.edu.hk).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2023.3325777, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3325777

path redundancy is an important challenge to solve during its motion planning. Meng et al. proposed a workspace-based RRT* algorithm with enhancements in both trajectory costs and computation time [17], [18]. However, it may lead to inaccurate robot shape estimation due to singularities in pseudo-inverse kinematics. Model-based motion planners have also been presented for path following [19], tip tracking [20], and supervisory control [21] of multi-segment continuum robots. While significant progress has been made in motion planning for these different continuum robot types, the optimality of most existing motion planners is difficult to be satisfied without expensive computational effort. Such optimality is critical especially for continuum surgical robots to produce the desired motion paths to achieve optimal surgical outcomes.

Cross-entropy (CE) motion planning approach has been proposed in recent years to execute adaptive stochastic sampling, which can optimize the basic performance of sampling-based motion planning framework (such as RRT*) through elite selection of motion trajectories, leading to lower costs and less iterations [22]. Kobilarov et al. [22] firstly proposed the CE motion planning method for mobile robots and air vehicles, which was capable of producing lower robot trajectory costs compared to the RRT* algorithm. Suh et al. [23] proposed a cost-aware path planner combined with RRT* algorithm and CE method to find lower path costs for robot manipulation.

In this work, a CE motion planning framework is introduced for the first time, to our knowledge, for continuum robots. Compared with conventional CE motion planners [22] that focus only on implementation in the approximate point mass systems, our motion planning framework provides efficient and optimal sampling capability while considering the complex kinematic and shape constraints of continuum robots with high redundancy. It is designed for continuum robots to perform both non-FTL and FTL motion. Specifically, our contributions include the following: **1)** A CE-integrated RRT* algorithm is designed to optimally perform adaptive stochastic sampling in a newly converted robot arc coordinate domain and search for feasible trajectory solutions under path redundancy, kinematic constraint, and shape constraint of continuum surgical robots (i.e. multi-segment continuum robots and concentric precurved tube robots). **2)** Optimal motion plans are solved specifically for a recently proposed hybrid continuum robot with coupled geometric relationship between its precurved body and cable-driven flexible tip segment [1] to perform both collision-free and approximate follow-the-leader (FTL) motions in constrained environments.

The rest of this letter is organized as follows. Section II describes the kinematics for continuum robots compatible with the proposed CE motion planner. The procedure of CE estimation during sampling-based motion planning is introduced in Section III. Section IV proposes the CE motion planner in details. Section V validates the performance of the proposed CE motion planner by simulation tests. Concluding remarks and future works are provided in Section VI.

II. Robot Kinematics

A. Kinematics for Two Typical Types of Continuum Robots

The proposed CE motion planner is targeted in this work for two typical types of continuum robots, namely multi-segment continuum robots with active bending of each flexible segment and concentric precurved tube robots. The continuum robot is assumed to be capable of navigating (i.e. producing a robot trajectory) from an entry point to a target point by extending and bending its robot segments. Without loss of generality, the kinematic parameters for the robot trajectory of a multi-segment continuum robot with m segments or concentric precurved tube robot with m tubes can be defined as $\mathbf{y}_{tra} = [\theta_e\ \theta_{r_1}\ \theta_{r_2}\ \rho_1\ \Theta_1^{\mathrm{T}} \ldots \rho_i\ \Theta_i^{\mathrm{T}} \ldots \rho_m\ \Theta_m^{\mathrm{T}}]^{\mathrm{T}}$, where $i = 1, \ldots, m$, as shown in Fig. 1. θ_e, θ_{r_1}, and θ_{r_2} are angles of insertion orientation, rotation around the robot itself, and rotation around z_e-axis in the global frame $\{e\}$. The generalized coordinates ρ_i and Θ_i are the arc length and bending angle of segment i (or extension length and rotation angle of tube i) in the local frame $\{i\}$. l is the total robot (extension) length along the robot arc coordinate s in the global frame $\{e\}$. Additionally, considering two actuation structures for the multi-segment continuum robots, the generalized coordinate Θ_i of segment i can be further expressed as

$$\Theta_i = \begin{cases} \begin{bmatrix} \theta_i & \varphi_i \end{bmatrix}^{\mathrm{T}}, & \text{normal segment i} \\ \begin{bmatrix} \theta_{xi} & \theta_{yi} \end{bmatrix}^{\mathrm{T}}, & \text{orthogonal segment i} \end{cases} \tag{1}$$

where θ_i and φ_i are the bending angle and the bending plane angle for a normal segment, and θ_{xi} and θ_{yi} are the bending angles in the $y_i z_i-$ and $x_i z_i-$ planes for an orthogonal segment with N joint pairs. It should be noted that these two typical kinematic models are under the piecewise constant-curvature assumption. Other mathematical models, such as the Cosserat rod model [24], can be adopted for continuum robots with non-constant curvature behavior.

The homogeneous transformation matrix $\mathbf{T}$ describing the robot forward kinematics f_{FK} can be expressed as

$$\mathbf{T} = \mathbf{Rot}_3 \cdot \mathbf{Rot}_2 \cdot \mathbf{Rot}_1 \cdot \prod_{i=1}^{m} \mathbf{T}_i(\Theta_i) \tag{2}$$

where $\mathbf{Rot}_1$, $\mathbf{Rot}_2$, and $\mathbf{Rot}_3$ denote the rotation transformations by angles of θ_e, θ_{r_1}, and θ_{r_2}, while $\mathbf{T}_i(\Theta_i)$ denotes the transformation describing the geometric representation of single segment (tube) i. The discrete positions along the robot arc coordinate s can be determined by concatenating $\mathbf{T}_i(\Theta_i)$ between the local frames $\{i\}$ fixed in the adjacent robot segments (tubes). Therefore, the actual robot trajectory $\mathbf{p}_r(\mathbf{X}, s)$ can be determined by solving f_{FK} or inverse kinematics f_{IK} with the robot shape vector $\mathbf{X}$ expressed as either $\mathbf{p}_t$ (state-based) or $\mathbf{y}_{tra}$ (parameter-based), such that $f_{IK}(\mathbf{p}_t) \rightarrow \mathbf{p}_r$ or $f_{FK}(\mathbf{y}_{tra}) \rightarrow \mathbf{p}_r$, where $\mathbf{p}_t = [x_t\ y_t\ z_t]^{\mathrm{T}}$ denotes the robot tip position. The Jacobian matrix $\mathbf{J}_{jac}$ is then be expressed as

$$\mathbf{J}_{jac} = \partial \mathbf{p}_t / \partial \mathbf{y}_{tra} \tag{3}$$

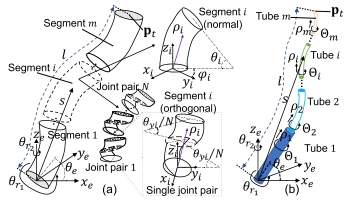

Fig. 1. Generalized coordinates for a (a) multi-segment continuum robot and (b) a concentric precurved tube robot.

B. Specific Kinematics for a Hybrid Continuum Robot

In the authors' previous work, a hybrid continuum robot consisting of a straight outer tube and a precurved inner tube with

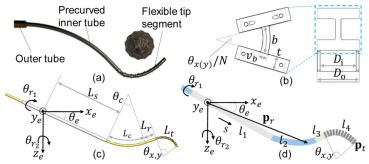

Fig. 2. (a) Hybrid continuum robot prototype. (b) Frame transformation of the 2-DOF notch pair in the tip segment. (c) Structural design parameters and (d) kinematic parameters of the hybrid continuum robot during its tube extension.

a monolithically integrated cable-driven flexible tip segment, as shown in Fig. 2(a) and (b), has been introduced [1]. Since the hybrid continuum robot combines the structural concepts of both concentric precurved tubes and active flexible segment, it is used as a target candidate for the implementation of the CE motion planner in this work. In this robot, $\mathbf{y}_{tra}$ can be expressed by $[\theta_e\ \theta_{r_1}\ \theta_{r_2}\ \rho_1\ \Theta_1^{\mathrm{T}}\ \rho_2\ \Theta_2^{\mathrm{T}}]^{\mathrm{T}}$, where ρ_1 and ρ_2 are extension lengths of outer and inner tubes, respectively. Θ_1 is empty because the outer tube cannot be rotated or bent. $\Theta_2 = [\theta_x\ \theta_y]^{\mathrm{T}}$, since the flexible tip is a notched tube backbone with two bending degrees of freedom (DOFs), as shown in Fig. 2(b).

The homogeneous transformation of each notch pair in the tip segment can be solved by taking the product of exponential mapping of the twists for the individual joints involved in the notch pair, which consists of key parameters such as notch beam width v_b, spacing height t, notch height b, and notch pair number N. Then, combining the notch pair transformation and the kinematic mapping among structural design parameters $\{L_s, L_c, L_r, \theta_c\}$, transformed length parameters $\{l_1, l_2, l_3, l_4\}$, and kinematic parameters $\{\theta_e, \theta_{r_1}, \theta_{r_2}, \rho_1, \theta_x, \rho_2, \theta_y\}$ (see detailed derivations in [1]), we can then determine the actual robot trajectory (shape) $\mathbf{p}_r(\mathbf{X}, s)$ as well as all the discrete points along the robot arc coordinate s in the global frame $\{e\}$, as shown in Fig. 2(c) and 2(d). It is also noted that the accuracy of the robot kinematics has been validated in [1], confirming the feasibility of using kinematics to calculate the actual robot trajectory $\mathbf{p}_r(\mathbf{X}, s)$.

III. Cross-Entropy Estimation

The cross-entropy (CE) method [25] employs stochastic estimation to select the elite trajectories (robot shapes) among random samples for every robot extension step. In the process, it adaptively generates a probabilistic model to sample parameters (i.e. $\mathbf{p}_t$ and $\mathbf{y}_{tra}$) using the optimized Gaussian distributions. Adopting the CE method in estimating $\mathbf{p}_t$ and $\mathbf{y}_{tra}$, individual trajectory samples are optimized and elite selection is conducted to ensure in particular collision-free or/and FTL motions for continuum robots. A unique and optimal path of sampled nodes (calculated by $\mathbf{p}_t$ or $\mathbf{y}_{tra}$) is eventually determined from the initial node to the target node.

The robot trajectory at every robot extension step is determined by $\mathbf{p}_r(\mathbf{X}, s)$. The sampling cost function $J(\mathbf{X})$ for every robot trajectory can therefore be expressed as

$$J(\mathbf{X}) = a_1 \cdot cost_\Gamma + a_2 \cdot \frac{l}{L} + a_3 \cdot \left\|\mathbf{y}_{tra\_cur} - \mathbf{y}_{tra\_pre}\right\| + a_4 \cdot \left\|\mathbf{p}_r(\mathbf{X}, l) - \mathbf{p}_t^f\right\|^2, \text{ s.t. } \mathbf{p}_r(\mathbf{X}, s) \in \Lambda,\ \mathbf{p}_r(\mathbf{X}, l) = \mathbf{p}_t \tag{4}$$

$$cost_\Gamma = \begin{cases} \int_0^l \frac{1}{\Delta(\mathbf{p}_r(\mathbf{X}, s), \Gamma) + 0.001}, & \text{collision-free} \\ e_{FTL}, & \text{approximate FTL} \end{cases} \tag{5}$$

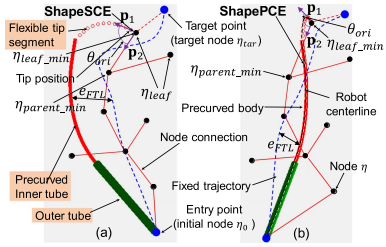

Fig. 3. Schematics showing the presented CE planner based on (a) ShapeSCE approach and (b) ShapePCE approach using the the hybrid continuum robot.

where $cost_\Gamma$ denotes the cost of the distance Δ of the environmental constraints Γ along the robot arc coordinate s. $\mathbf{y}_{tra\_cur}$ and $\mathbf{y}_{tra\_pre}$ are the kinematic parameter vectors of previous and current robot extension steps. $\mathbf{p}_t^f$ is the target point position. a_1, a_2, a_3, and a_4 are penalty factors for the distance of the robot from the environmental obstacles, robot extension length with a maximum length limit L, robot joint variables change, and distance between the current robot tip and the desired target, respectively. e_{FTL} is the FTL error expressed in (6), denoting the maximum shape deviation error between a fixed trajectory and the actual robot shape, as shown in Fig. 3.

$$e_{FTL} = \max\left(\|\mathbf{p}_r(\mathbf{X}, s) - \mathbf{p}_f(s)\|\right), \forall s \in [0, l] \tag{6}$$

where $\mathbf{p}_f(s)$ is a fixed trajectory, and it is user-predefined or predefined by surgeons.

The probability density function (pdf) of $\mathbf{X}$ used in the CE method can be expressed as

$$f(\mathbf{X}; \mathbf{v}) = \frac{\exp(-\frac{1}{2}(\mathbf{X} - \mu)^{\mathrm{T}}\Sigma^{-1}(\mathbf{X} - \mu))}{(2\pi)^{\frac{n_d}{2}}(\det\Sigma)^{\frac{1}{2}}} \tag{7}$$

where μ, Σ, and n_d denote the mean vector, the covariance matrix, and the mixture component number of the robot shape vector $\mathbf{X}$, respectively. $\mathbf{v} = \{\mu, \Sigma\}$ are the reference parameters in the pdf, which can be expressed as

$$\mu = \frac{1}{n_{elite}} \sum_{i=1}^{n_{elite}} \mathbf{X}_{elite,i} \quad \Sigma = \begin{bmatrix} \sigma_1^2 & & \Sigma_\rho \\ & \ddots & \\ \Sigma_\rho & & \sigma_{n_d}^2 \end{bmatrix}_{n_d \times n_d} \tag{8}$$

where the elite robot shape vector $\mathbf{X}_{elite}$ represents the elite robot trajectory (i.e. $\mathbf{X}$ with the smallest cost $J(\mathbf{X})$ among the n samples), $n_{elite} = \lceil \rho n \rceil$ is the number of $\mathbf{X}_{elite}$, and ρ is a user-specified parameter for the elite fraction. $\sigma_1, \ldots, \sigma_{n_d}$ are variances of all the component variables in $\mathbf{X}_{elite}$. The correlation coefficient matrix Σ_ρ is assumed to be $\mathbf{0}$, since all the component variables in $\mathbf{X}_{elite}$ are independent. if the coupling relationship exists in $\mathbf{X}_{elite}$. The reference parameters $\mathbf{v} = \{\mu, \Sigma\}$ of the pdf are optimized by CE minimization [25]:

$$\mathbf{v}* = \arg\max_{\mathbf{v}} \frac{1}{n} \Sigma_{k=1}^{n} I_{\{J(\mathbf{X}_k) \le \hat{\gamma}_{t_{CE}}\}} \ln f(\mathbf{X}_k; \mathbf{v}) \tag{9}$$

where $\hat{\gamma}_{t_{CE}} = J_{(\lceil \rho n \rceil)}$ is the sample quantile of the cost performance for the elite trajectories. $I(\cdot)$ is an indicator function, which determines the elite individuals $\mathbf{X}_{elite}$. As shown in Algorithm 1, the CE estimation can be summarized as follows: *Step 1:* Set iteration counter $t_{CE} = 1$. Calculate sampling cost J of each sampled robot shape vector $\mathbf{X}$ from its set $\mathcal{S}$, and then order them from smallest to largest, $J_1 \le J_2 \le \cdots \le J_n$. *Step 2:* Calculate $\hat{\gamma}_{t_{CE}}$ and use the current $\mathbf{X}_{elite}$ to calculate the pdf.

Algorithm 1: $(\mathbf{v}*, \mathbf{X}_{new}) \leftarrow$ `CE_Estimation`$(\mathbf{X}, \mathcal{S})$.

Set $\alpha_0 \leftarrow 0.2$ and $t_{CE} \leftarrow 1$; Find all $\mathbf{X} \in \mathcal{S}$
while `CollisionFree`$(\mathbf{X}_{new})$ & $t_{CE} \leq t_{CE,\max}$ **do**
  Calculate trajectory cost $J(\mathbf{X})$;
  $\hat{\gamma}_{t_{CE}} \leftarrow J_{(\lceil \rho n \rceil)}$;
  $\mathbf{v}* \leftarrow \arg\max_{\mathbf{v}} \frac{1}{n}\Sigma_{k=1}^{n} I_{\{J(\mathbf{X}_k) \leq \hat{\gamma}_{t_{CE}}\}} \ln f(\mathbf{X}_k; \mathbf{v})$;
  Calculate the pdf and then sample $\mathbf{X}_{new}$;
  $\mathbf{X}_{new} \sim \ln f(\cdot; \mathbf{v}*)$ and $t_{CE} \leftarrow t_{CE} + 1$;
end while
return $\mathbf{v}*$ and $\mathbf{X}_{elite,best}$;

Step 3: Set $t_{CE} = t_{CE} + 1$. Sample $\mathbf{X}_{new} \sim \ln f(\cdot; \mathbf{v}*)$ based on pdf using the updated optimal parameters $\mathbf{v}*$ according to (9) until reaching the maximum iteration number $t_{CE,\max}$.

During each CE estimation iteration t_{CE}, the newly sampled individuals $\mathbf{X}_{new}$ can be expressed as

$$\begin{bmatrix} \mathbf{X}_{new,1}^{\mathrm{T}} \\ \vdots \\ \mathbf{X}_{new,m}{}^{\mathrm{T}} \end{bmatrix} = randn(m, n_d) * \sqrt{\hat{\Sigma}_{t_{CE}}} + \begin{bmatrix} \hat{\mu}_{t_{CE}}^{\mathrm{T}} \\ \vdots \\ \hat{\mu}_{t_{CE}}^{\mathrm{T}} \end{bmatrix}_{m \times n_d} \tag{10}$$

$$\begin{aligned} \hat{\mu}_{t_{CE}} &= (1 - \alpha_0)\mu_{t_{CE}} + \alpha_0 \mu_{t_{CE}-1} \\ \hat{\Sigma}_{t_{CE}} &= (1 - \alpha_0)\Sigma_{t_{CE}} + \alpha_0 \Sigma_{t_{CE}-1} \end{aligned} \tag{11}$$

where m is the number of newly updated individuals $\mathbf{X}_{new}$ of the elite robot shape vector. $randn()$ returns m $1 \times n_d$ vectors with all the elements sampled from the Gaussian distribution $\mathcal{N}(0, 1)$. $\sqrt{\Sigma}$ can be solved by Cholesky decomposition. Equation (11) smoothens the elite seletion of $\mathbf{X}_{elite}$ with a smoothing factor α_0. The estimation is terminated after $t_{CE,\max}$ iterations, giving the most elite robot shape vector $\mathbf{X}_{elite,best}$ with the lowest cost among m individuals of $\mathbf{X}_{new}$.

IV. Cross-Entropy Motion Planning

The RRT* approach is widely utilized for single-query problems, sampling random nodes (e.g. black dots in Fig. 3) starting from initial configuration at the root node (initial node) and then adding collision-free nodes to the node tree as leaf nodes. It finally generates feasible paths from the entry point to the target point by progressively searching eligible nodes until the target node is added to the node tree. In contrast to the RRT* performing basic sampling, the proposed CE motion planning methodology utilizes the CE state-based or parameter-based estimation to perform adaptive sampling, which can optimize trajectories with low costs and iterations. From the perspective of state-based estimation, this is a way to adaptively optimize the tip position to determine elite trajectories of the continuum robot. From the perspective of parameter-based estimation, the robot trajectory is firstly parameterized by a certain number of variables, which are then optimized by adaptive CE sampling to search for lower trajectory costs. Therefore, considering the kinematic and shape constraints, shape state cross-entropy (ShapeSCE) motion planner and shape parameterized cross-entropy (ShapePCE) motion planner are developed for continuum robots, corresponding to state-based and parameter-based CE estimations, as shown in Fig. 3. Fig. 3 depicts the difference

Algorithm 2: CE Motion Planning.

Set tip position error tolerance $e_1 \leftarrow 3$ mm;
$\mathbf{X} \leftarrow \varnothing$ and $\mathcal{N}_{leaf} \leftarrow \varnothing$;
$\mathcal{T} \leftarrow$ `InsertNode`$(\varnothing, \eta_0, \mathcal{T})$;
for $i \leftarrow 1 : N_\eta$ **do**
  $(\eta_{rand}, \mathbf{X}) \leftarrow$ `ShapeSCE(or ShapePCE)_Sample`$(i, \mathcal{N}_{leaf})$;
  $\eta_{nearest} \leftarrow$ `Nearest`$(\mathcal{T}, \mathcal{N}_{rand})$;
  $(\eta_{new}, \chi_{new}, \Delta d_2) \leftarrow$ `Steer`$(\eta_{nearest}, \eta_{rand})$
  if `CollisionFree`(χ_{new}) **then**
    $\mathcal{N}_{near} \leftarrow$ `Near_`$Neighbor(\mathcal{T}, \eta_{new})$;
    $\eta_{\min} \leftarrow$ `Parent`$(\mathcal{N}_{near}, \eta_{nearest}, \eta_{new})$;
    $\mathcal{T} \leftarrow$ `InsertNode`$(\eta_{\min}, \eta_{new}, \mathcal{T})$;
    $\mathcal{T} \leftarrow$ `Rewire`$(\eta_{\min}, \eta_{new}, \mathcal{N}_{near}, \mathcal{T})$;
    $\mathcal{N}_{leaf} \leftarrow \eta_{leaf} \bigcup \mathcal{N}_{leaf}$;
    $J(\mathbf{X}_{leaf}) \leftarrow$ `TrajectoryCost`$(\eta_{leaf}, \mathbf{X}_{leaf})$;
    if `CollisionFree`$(\chi_{new})\&\|\eta_{tar} - \eta_{new}\| \leq e_1$ **then**
      $J(\mathbf{X}_{tar}) \leftarrow$ `TrajectoryCost`$(\eta_{tar}, \mathbf{X}_{tar})$;
      `Break`;
    end if
  end if
end for
return $\mathcal{T}$ and $J(\mathbf{X})$;

and connection between ShapeSCE and ShapePCE. The black dots denote the sampled nodes, which are calculated by the tip position $\mathbf{p}_t$ or kinematic parameters $\mathbf{y}_{tra}$. The red solid and dotted lines respectively denote the edges among sampled nodes accepted by $\mathcal{T}$ and the potential connections between the accepted nodes and target node. ShapeSCE samples $\mathbf{p}_t$ that can make the robot under-defined since it requires less sampling information, while ShapeSCE samples $\mathbf{y}_{tra}$ that can fully define the robot shape denoted by the black dotted line.

In Algorithm 2, we extend the RRT* algorithm to a CE-based motion planning algorithm with several improvements: 1) Different from state-cross-entropy (SCE) and trajectory-cross-entropy (TCE) RRT* algorithms [22], our method considers the high-dimensional continuum robot configuration space. The time domain is therefore converted to the robot arc coordinate domain to perform both obstacle-free and approximate FTL motions, and the path redundancy is minimized by real-time iterative calculation of forward and inverse kinematics. 2) The entire generated robot trajectories are always iteratively checked to satisfy kinematic and shape constraints to ensure obstacle avoidance during the robot navigation. It is also noted that the robot tip orientation is constrained to limit the maximum extension orientation angle of the robot tip (see function `Orientation`() in Section IV(A)). 3) The rewiring of sampled nodes is governed by the damped least squares (DLS) method [26] to minimize the robot configuration change and the trajectory cost; this can also address the singularity issue in pseudo inverse kinematics solving and ensure a stable numerical solution to the robot configurations.

As shown in Algorithm 2, η_0, η_{rand}, η_{near}, η_{leaf}, and η_{tar} denote respectively the initial, sampled, neighbor, leaf, and target nodes. $\mathcal{N}_{rand}$, $\mathcal{N}_{near}$, and $\mathcal{N}_{leaf}$ are node sets of nodes η_{rand}, η_{near}, and η_{leaf}, respectively. $\mathcal{T}$ is a tree of sampled nodes η with the maximum iteration number N_η.

A. ShapeSCE Approach

The ShapeSCE approach is described in Algorithm 3, which implements `ShapeSCE_Sample()` function to estimate the robot trajectories by solving the inverse kinematics during adaptive sampling of new robot tip position, as shown in Fig. 3(a). In Algorithm 3, CE estimation ratio f_{ce} dictates the switch between the standard uniform sampling $\texttt{Uniform}(\mathcal{N}_{leaf}, \mathcal{V})$ and the CE estimation for the leaf nodes η_{leaf}. $\mathcal{V}$ is a spherical neighborhood with the center point at the leaf node η_{leaf} and the radius of each extension step length, or a randomly switched node set that is close to the target node η_{tar}, which can ensure the robot is extending toward η_{tar}. $\mathbf{p}_{t\_rand\_new}$, l_{rand_new}, and $\mathbf{p}_{r\_rand\_new}$ are the node position, the robot extension length, and the robot trajectory, corresponding to the sampled random node η_{rand_new}. Function `Orientation()` constrains the orientation range of the generated trajectories, which ensures the intersection angle θ_{ori} is less than the maximum angle $\theta_{ori,\max} = 120^\circ$ between the distal tip tangent unit vector $\mathbf{p}_1$ and the unit vector $\mathbf{p}_2$ in the direction of the connection between $\eta_{leaf\_\min}$ and $\eta_{parent\_\min}$, as shown in Fig. 3. $\eta_{leaf\_\min}$ denotes $N_{\min}$ leaf nodes with the lowest sampling cost $J(\mathbf{X})$ among all $\eta_{leaf} \in \mathcal{T}$, and $\eta_{parent\_\min}$ is a parent node of $\eta_{leaf\_\min}$. Therefore, the function `Orientation()` ensures an inequality of the tip orientation deviation, which can be expressed as

$$\arccos(\mathbf{p}_1 \cdot \mathbf{p}_2) \le \theta_{ori,\max} = 120^\circ \tag{12}$$

$$\mathbf{p}_1 = \frac{\mathbf{p}_r(\mathbf{X}, l - \Delta d_1) - \mathbf{p}_r(\mathbf{X}, l)}{\|\mathbf{p}_r(\mathbf{X}, l - \Delta d_1) - \mathbf{p}_r(\mathbf{X}, l)\|}$$

$$\mathbf{p}_2 = \frac{\mathbf{p}_{t\_parent\_\min} - \mathbf{p}_{t\_leaf\_\min}}{\left\|\mathbf{p}_{t\_parent\_\min} - \mathbf{p}_{t\_leaf\_\min}\right\|} \tag{13}$$

where Δd_1 is the extension step length. $\mathbf{p}_{t\_parent\_\min}$ and $\mathbf{p}_{t\_leaf\_\min}$ are node positions of $\eta_{parent\_\min}$ and $\eta_{leaf\_\min}$.

When executing an approximate FTL motion, the maximum lateral deviation error $\bar{e}_{FTL}$ is the minimum value calculated iteratively in increments of 0.2 mm starting with an initial lateral FTL error of 2 mm. It is terminated when the following three conditions are satisfied: 1) there is no collision, 2) the target node η_{tar} can be reached, and 3) $e_{FTL} \le \bar{e}_{FTL}$. e_{FTL} is calculated by Eq. (6). The mixture component dimension n_d is equal to 3 in the ShapeSCE motion planner. Finally, the new node position $\mathbf{p}_{t\_new}$ is assumed to be the tip position, therefore, the actual robot trajectory $\mathbf{p}_r(\mathbf{X}, s)$ can be solved by the mapping: $f_{IK}(\mathbf{p}_t) \to \mathbf{p}_r(\mathbf{X}, s)$, which is added to the collision-free robot trajectory set Λ after collision detection.

B. ShapePCE Approach

The main goal of the ShapePCE approach is to sample a parameterized trajectory (robot shape denoted by black dotted line as shown in Fig. 3(b)) for collision detection, which utilizes the sampled variable vector $\mathbf{y}_{tra}$ to calculate the actual robot trajectory, as shown in Algorithm 4. Different from the ShapeSCE approach, the actual robot trajectory $\mathbf{p}_r(\mathbf{X}, s)$ can be easily determined during the adaptive sampling of $\mathbf{y}_{tra}$ with the mapping $f_{FK}(\mathbf{y}_{tra}) \to \mathbf{p}_r(\mathbf{X}, s)$, instead of inverse kinematics calculation $f_{IK}(\mathbf{p}_t) \to \mathbf{p}_r(\mathbf{X}, s)$. Function `Orientation()` also sets a constrained orientation range for the robot tip under the control of the CE estimation ratio f_{ce}. Additionally, the discrete points along the robot centerline can be extracted from the individual trajectory $\mathbf{p}_r(\mathbf{X}, s)$, therefore, it can take the overall individual trajectory $\mathbf{p}_r(\mathbf{X}, s)$ into account for cost calculation. Generally, ShapeSCE and ShapeSCE motion planning approaches can be converted to each other by the mappings $f_{FK}(\mathbf{y}_{tra}) \to \mathbf{p}_r(\mathbf{X}, s)$ and $f_{IK}(\mathbf{p}_t) \to \mathbf{p}_r(\mathbf{X}, s)$. Thus, similar to the adaptive robot state (tip position) estimation of the ShapeSCE approach, the robot state is also added to $\mathcal{T}$ after the obstacle-free trajectory parameterization, leading to feasible trajectories $\mathbf{p}_r(\mathbf{X}, s) \in \Lambda$.

Algorithm 3: $(\eta_{rand}, \mathbf{p}_{t\_new}) \leftarrow \texttt{ShapeSCE\_Sample}(i, \mathcal{N}_{leaf})$.

Set CE estimation ratio $f_{ce} \leftarrow 0.5$, minimum required number $N_{\min}$ of leaf nodes, and extension step length $\Delta d_1 \leftarrow 3$ mm; Set $n_d \leftarrow 3$;
Find parent node $\eta_{parent\_\min}$ of $\eta_{leaf\_\min}$;
while $\texttt{CollisionFree}(\mathbf{p}_{r\_rand\_new})$ & $\mathbf{p}_r(\mathbf{p}_{t\_rand\_new}, l_{rand\_new}) \in \mathcal{V}$ & $\texttt{Orientation}(\eta_{parent\_\min}, \eta_{leaf\_\min}, \mathbf{p}_{r\_rand\_new})$ & $e_{FTL} \le \bar{e}_{FTL}$ **do**
  if $\texttt{rand}(0,1) < f_{ce}$ & $|\mathcal{N}_{leaf}| > N_{\min}$ **then**
    $(\mathbf{v}*, \mathbf{p}_{t\_rand\_new}) \leftarrow \texttt{CE\_Estimation}(\mathbf{p}_t, \mathcal{N}_{leaf})$;
    $\mathbf{p}_{t\_rand\_new} \leftarrow \mathbf{p}_{t\_rand\_new} + \Delta d_1 \cdot (\mathbf{p}_t^f - \mathbf{p}_{t\_rand\_new})/\|\mathbf{p}_t^f - \mathbf{p}_{t\_new}\|$;
    $\eta_{rand\_new} \leftarrow \eta(\mathbf{p}_{t\_rand\_new})$;
  else
    Sample $\eta_{rand\_new} \sim \texttt{Uniform}(\mathcal{N}_{leaf}, \mathcal{V})$;
  end if
end while
return $\eta_{rand} \leftarrow \eta_{rand\_new}$ and $\mathbf{p}_{t\_new}$;

Algorithm 4: $(\eta_{rand}, \mathbf{y}_{tra\_new}) \leftarrow \texttt{ShapePCE\_Sample}(i, \mathcal{N}_{leaf})$.

Set CE estimation ratio $f_{ce} \leftarrow 0.5$ and minimum required number $N_{\min}$ of leaf nodes. Set $n_d \leftarrow 7$;
Find parent node $\eta_{parent\_\min}$ of $\eta_{leaf\_\min}$; $l_{pre} \leftarrow$ robot length of the previous extension;
while $\texttt{CollisionFree}(\mathbf{p}_{r\_rand\_new})$ & $\mathbf{p}_r(\mathbf{p}_{t\_rand\_new}, l_{rand\_new}) \in \mathcal{V}$ & $\texttt{Orientation}(\eta_{parent\_\min}, \eta_{leaf\_\min}, \mathbf{p}_{r\_rand\_new})$ & $e_{FTL} \le \bar{e}_{FTL}$ **do**
  if $\texttt{rand}(0,1) < f_{ce}$ & $|\mathcal{N}_{leaf}| > N_{\min}$ & $l_{rand\_new} > l_{pre}$ **then**
    $(\mathbf{v}*, \mathbf{y}_{tra\_new}) \leftarrow \texttt{CE\_Estimation}(\mathbf{y}_{tra}, \mathcal{N}_{leaf})$;
    $\eta_{rand\_new} \leftarrow \eta(\mathbf{y}_{tra\_new})$;
  else
    Sample $\eta_{rand\_new} \sim \texttt{Uniform}(\mathcal{N}_{leaf}, \mathcal{V})$;
  end if
end while
return $\eta_{rand} \leftarrow \eta_{rand\_new}$ and $\mathbf{y}_{tra\_new}$;

C. Remaining Procedures

Each node $\eta \in \mathcal{T}$ in the rest of the CE motion planner in Algorithm 2 is determined by `InsertNode()`, `Steer()`, `Parent()`, `Rewire()`, and `TrajectoryCost()`, which are modified and improved from those in the conventional algorithm [5], [22] to cater to the new domain of continuum robot arc length while

satisfying the kinematics and shape constraints. These functions can be described as follows:

(1) $\texttt{InsertNode}(\eta_1, \eta_2, \mathcal{T})$ inserts a node η_2 to become a child of an existing node η_1, and creates an edge between the two nodes inside $\mathcal{T}$ using $\texttt{AddNodeEdge}(\eta_2, \eta_1, \mathcal{T})$, where η_1 and η_2 are arbitrary nodes. **(2)** Once the function $\texttt{Nearest}()$ determines a node $\eta_{nearest}$ closest to $\eta_{rand} \in \mathcal{N}_{rand}$, which is the output of either the ShapeSCE or the ShapePCE approach, $\texttt{Steer}(\eta_{nearest}, \eta_{rand})$ returns a robot motion transition χ_{new} from $\eta_{nearest}$ to η_{rand}. It also creates a new node η_{new} between $\eta_{nearest}$ and η_{rand} with a tip position movement that is within the predefined permissible movement distance Δd_2 (e.g. 3 mm) for $\texttt{Steer}()$. The linear interpolation between two configurations of $\eta_{nearest}$ and η_{new} are conducted to obtain the interpolated configurations that can ensure the interval distance among the corresponding tip position trajectory is less than a small distance (e.g. 2 mm). The tip trajectory is unconditionally accepted if collision detection for χ_{new} (including both tip trajectory and robot shape change) is successful by $\texttt{CollisionFree}()$. Otherwise, η_{new} is unacceptable to $\mathcal{T}$. Additionally, safety margin along the robot is adopted during collision detection [13], [20]. The function $\texttt{CollisionFree}()$ is called to ensure there is no collision with the surrounding obstacles before the trajectory estimation proceeds further. **(3)** After $\texttt{Near\_}Neighbor()$ collects all the neighbor nodes η_{near} close to η_{new} in the node set $\mathcal{N}_{near}$, $\texttt{Parent}(\mathcal{N}_{near}, \eta_{nearest}, \eta_{new})$ returns a selection of the optimal parent nodes $\eta_{\min}$ for η_{new}, as seen in Algorithm 5. Here, the total cost of the potential parent nodes ($\eta_{near} \in \mathcal{N}_{near}$) takes into account the current trajectory cost $J(\mathbf{X}_{near})$ and the change of the parameter-based variable vector $\mathbf{y}_{tra}$ (robot configuration $\mathbf{q}$) between nodes η_{near} and η_{new} with α_1 and α_2 being the weighting factors. **(4)** As seen in Algorithm 6, $\texttt{Rewire}(\eta_{\min}, \eta_{new}, \mathcal{N}_{near}, \mathcal{T})$ sets node η_{new} to be a parent node of $\eta_{near} \in \mathcal{N}_{near} \backslash \{\eta_{\min}\}$ if the trajectory cost $J(\mathbf{X})$ decreases when the robot is extending (i.e. $l_{new} < l_{near}$, where l_{new} and l_{near} are the robot extension lengths relative to η_{new} and η_{near}). Otherwise, the rewiring procedure is omitted so that the edge connected to the new node η_{new} remains unchanged. Combined with the method of Lagrange multipliers, the robot configuration $\mathbf{q}_{near}$ of η_{near} is always redetermined by the DLS method to minimize the robot configuration change $\Delta\mathbf{q}$ between η_{new} and η_{near}, where the robot configuration $\mathbf{q}$ is equal to $\mathbf{y}_{tra}$. Hence, it is desired to find the minimum $\Delta\mathbf{q}$ that satisfies the kinematic constraint, which can be solved by [26]

$$\min_{\Delta\mathbf{q}} \left(\left\| \mathbf{J}_{jac}^{\mathrm{T}} \mathbf{J}_{jac} - (\mathbf{p}_{t\_near} - \mathbf{p}_{t\_new}) \right\|^2 + \lambda^2 \left\| \Delta\mathbf{q} \right\|^2 \right)$$

$$\Rightarrow \Delta\mathbf{q} = \mathbf{J}_{jac}^{\mathrm{T}} (\mathbf{J}_{jac} \mathbf{J}_{jac}^{\mathrm{T}} + \lambda^2 \mathbf{I})^{-1} (\mathbf{p}_{t\_near} - \mathbf{p}_{t\_new}) \quad (14)$$

$$\hat{\mathbf{p}}_{t\_near} = \mathbf{p}_r(\Delta\mathbf{q} + \mathbf{q}_{new}, l) \quad (15)$$

where $\mathbf{I}$ and λ are an identity matrix and a damping constant, respectively. $\mathbf{p}_{t\_near}$ and $\mathbf{p}_{t\_new}$ are the node positions of η_{near} and η_{new}. $\mathbf{q}_{near}$ and $\mathbf{q}_{new}$ are the robot configurations of η_{near} and η_{new}. The estimated configuration $\hat{\mathbf{q}}_{near} = \mathbf{q}_{new} + \Delta\mathbf{q}$ is updated for the node η_{near} if the cost $J(\mathbf{X})$ decreases with a small tip position movement $3e_1$ (from $\mathbf{p}_{t\_near}$ to the estimated node position $\hat{\mathbf{p}}_{t\_near}$) of η_{near}, where e_1 is the tip position error tolerance. It is also noted that the DLS method can avoid the null space variance of the Jacobian configuration-generation (JCG) method during pseudo inverse kinematics solving for motion plans of continuum robots [17]. **(5)** The function $\texttt{TrajectoryCost}(\eta_1, \mathbf{X})$ returns the trajectory cost $J(\mathbf{X})$ (according to (4)) at each extension step, where node η_1 corresponds to the robot tip.

Finally, the target node η_{tar} is assumed to be reached when the Euclidean distance between η_{new} and η_{tar} is less than e_1. Besides, the FTL error e_{FTL} between the fixed trajectory and actual robot trajectory (shape) is minimized at every extension step, when the approximate FTL motion is performed in both ShapeSCE and ShapePCE, as shown in Fig. 3.

Algorithm 5: $\eta_{\min} \leftarrow \texttt{Parent}(\mathcal{N}_{near}, \eta_{nearest}, \eta_{new})$.

Set $\alpha_1 \leftarrow 0.8$ and $\alpha_2 \leftarrow 0.2$; Find $\mathbf{y}_{tra\_\min}$ and $\mathbf{X}_{\min}$ corresponding to $\eta_{\min}$, and find $\mathbf{y}_{tra\_new}$ corresponding to η_{new};
$\eta_{\min} \leftarrow \eta_{nearest}$;
$J(\mathbf{X}_{\min}) \leftarrow \texttt{TrajectoryCost}(\eta_{\min}, \mathbf{X}_{\min})$;
$Cost\_\min \leftarrow \alpha_1 \cdot J(\mathbf{X}_{\min}) + \alpha_2 \cdot \|\mathbf{y}_{tra\_\min} - \mathbf{y}_{tra\_new}\|^2$;
for $\eta_{near} \in \mathcal{N}_{near}$ **do**
  Find $\mathbf{y}_{tra\_near}$ corresponding to node η_{near};
  $\mathbf{X}_{near} \leftarrow \mathbf{y}_{tra\_near}$;
  $J(\mathbf{X}_{near}) \leftarrow \texttt{TrajectoryCost}(\eta_{near}, \mathbf{X}_{near})$;
  if $\alpha_1 \cdot J(\mathbf{X}_{near}) + \alpha_2 \cdot \|\mathbf{y}_{tra\_near} - \mathbf{y}_{tra\_new}\|^2 < Cost\_\min$ & $\texttt{CollisionFree}(\eta_{near}, \eta_{new})$ **then**
    $Cost\_\min \leftarrow \alpha_1 \cdot J(\mathbf{X}_{near}) + \alpha_2 \cdot \|\mathbf{y}_{tra\_near} - \mathbf{y}_{tra\_new}\|^2$;
    $\eta_{\min} \leftarrow \eta_{near}$;
  end if
end for
return $\eta_{\min}$;

Algorithm 6: $\mathcal{T} \leftarrow \texttt{Rewire}(\eta_{\min}, \eta_{new}, \mathcal{N}_{near}, \mathcal{T})$.

Set $\mathbf{q}_{new} \leftarrow \mathbf{y}_{tra\_new}$ corresponding to node η_{new}; Set $\lambda \leftarrow 0.1$;
for $\eta_{near} \in \mathcal{N}_{near} \backslash \{\eta_{\min}\}$ **do**
  Find $\mathbf{y}_{tra\_near}$ corresponding to node η_{near};
  $\Delta\mathbf{q} \leftarrow \mathbf{J}_{jac}^{\mathrm{T}} (\mathbf{J}_{jac} \mathbf{J}_{jac}^{\mathrm{T}} + \lambda^2 \mathbf{I})^{-1} (\mathbf{p}_{t\_near} - \mathbf{p}_{t\_new})$;
  $\hat{\mathbf{q}}_{near} \leftarrow \mathbf{q}_{new} + \Delta\mathbf{q}$; $\hat{\mathbf{p}}_{t\_near} \leftarrow \mathbf{p}_r(\hat{\mathbf{q}}_{near}, l)$;
  if $\texttt{CollisionFree}(\eta_{new}, \eta_{near})$ & $\|\hat{\mathbf{p}}_{t\_near} - \mathbf{p}_{t\_near}\| < 3e_1$ **then**
    $\mathbf{X}_{near} \leftarrow \hat{\mathbf{q}}_{near}$; $Cost\_current \leftarrow J(\mathbf{X}_{near})$;
    if $Cost\_current < J(\mathbf{X}_{new})$ & $l_{new} < l_{near}$ **then**
      $\mathcal{T} \leftarrow \texttt{AddNodeEdge}(\eta_{new}, \eta_{near}, \mathcal{T})$;
    end if
  end if
end for
return $\mathcal{T}$;

V. Results and Evaluations

The proposed motion planners based on both the ShapeSCE and ShapePCE approaches were implemented on the hybrid continuum robot to evaluate their ability in enabling 1) obstacle avoidance, 2) approximate FTL motion, and 3) navigation in a simulated clinical scenario. The proposed CE planners and other algorithms used for comparison were individually evaluated in the three aforementioned randomized single-query tasks with each evaluation consisting of 30 test

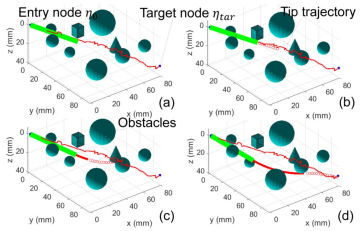

Fig. 4. Obstacle avoidance by ShapeSCE planner, demonstrated by four selected extension steps ((a)–(d)). The red solid line denotes the tip trajectory during the robot motion.

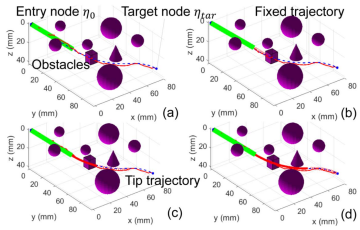

Fig. 5. Approximate FTL motion by ShapePCE planner, demonstrated by four selected extension steps ((a)–(d)). The blue dotted line denotes the fixed trajectory $\mathbf{p}_f(s)$.

TABLE I
PERFORMANCE COMPARISON FOR OBSTACLE AVOIDANCE

	Cost $J(\mathbf{X})$	Time t_p	SR
ShapeSCE	$(1.01 \pm 0.06) \times 10^4$	†266 ± 55 s	87%
ShapePCE	†$(9.05 \pm 0.6) \times 10^3$	†259 ± 43 s	83%
RRT*	$(1.11 \pm 0.03) \times 10^4$	1031 ± 561 s	43%
WRRT*	$(9.65 \pm 0.8) \times 10^3$	8760 ± 960 s	27%

TABLE II
COMPARISON FOR APPROXIMATE FTL MOTION

	$\bar{e}_{FTL}$	Cost $J(\mathbf{X})$	Time t_p	SR
ShapeSCE	5.2 mm	†$(1.61 \pm 0.1) \times 10^4$	†467 ± 139 s	97%
ShapePCE	2.8 mm	†$(1.45 \pm 0.1) \times 10^4$	†286 ± 32 s	93%
RRT-Shape	8.4 mm	$(2.27 \pm 0.2) \times 10^4$	726 ± 399 s	90%

trials. All the simulations and tests were executed in MATLAB R2020b installed on a DELL desktop computer with Intel(R) Core(TM) i5-9500 CPU @3.00 GHz 8.00 GB RAM. The robot structural parameters are as follows: $[L_s\ L_c\ L_r\ L_t\ b\ t\ v_b]^{\mathrm{T}} = [46\ 57\ 3\ 30\ 0.77\ 1.37\ 0.31]^{\mathrm{T}}$ mm and $N = 7$. The comparison algorithms adopt the same parameter setting as the presented CE motion planning approaches (e.g. cost calculation, rewiring process, and selection of the optimal parent node). The results were compared using a nonparametric Friedman test to calculate statistical significance ($p < 0.05$), followed by a post-hoc Wilcoxon matched-pairs signed-rank test with Bonferroni correction ($p < 0.0125$), † is labeled next to the result from either ShapeSCE or ShapePCE if it is statistically significant relative to all the methods used in comparison.

A. Experiment 1: Obstacle Avoidance

In the obstacle avoidance experiment shown in Fig. 4, multiple shapes of obstacles are adopted such as spheres, cones, and cubes, and the entry and target nodes were (0, 0, 0) and (80, 90, 30) mm, respectively. The ShapeSCE and ShapePCE motion planners were both executed and compared with other motion planners previously introduced for continuum robots, including RRT* algorithm [5] and workspace-based RRT* (WRRT*) algorithm [17]. As shown in Fig. 4, four robot extension steps were selected to showcase the ShapeSCE motion plans. Similar results were obtained using the ShapePCE planner and are therefore not shown in the same figure. As shown in Table I, ShapeSCE and ShapePCE greatly reduce both the average cost $J(\mathbf{X})$ of all sampled trajectories and the average computation time t_p, and enhance the success rate (SR) during obstacle avoidance. The cost $J(\mathbf{X})$ depends mainly on the distance from the robot tip to the target node η_{tar} at its early stage of adaptive sampling, and it starts to be dominated by the robot environmental, kinematic, and shape constraints at the later stage. The lower cost of ShapeSCE and ShapePCE indicates that the distance from the environmental obstacle is larger and the actuation input is less during robot motion. SR is defined by the ratio of the number of successful collision-free trials to the total number of test trials. A trial in the FTL cases is considered successful if the robot can satisfy two conditions, namely 1) achieve e_{FTL} less than the permissible maximum lateral deviation error $\bar{e}_{FTL}$ and 2) reach the target node η_{tar} within the maximum iteration number N_η. In the obstacle avoidance only (non-FTL) cases, only condition 2 is required. Compared with ShapeSCE, ShapePCE samples the trajectories with similar t_p but less $J(\mathbf{X})$. This is mainly because the paramerized robot trajectory can be fully defined and then solved by forward kinematics f_{FK}, avoiding the multi-solution problem of inverse kinematics f_{IK} present in ShapeSCE. As shown in Table I, $J(\mathbf{X})$ of ShapeSCE and ShapePCE are respectively reduced by by 9% and 18%, compared to RRT*. Besides, although WRRT* can also reduce the inverse kinematics computation time using the JCG method, there exists null space variance and singularities in pseudo inverse kinematics, which leads to a large tip position error, especially when the robot Jacobian matrices are complicated.

B. Experiment 2: Approximate FTL Motion

During the approximate FTL motion experiment, the fixed trajectory $\mathbf{p}_f(s)$ can be predetermined by an optimization-based motion planner or directly by the surgeon [1]. The motion planners evaluated here need to generate sequential motion plans to navigate between the entry point and the target point while minimizing e_{FTL}. As shown in Fig. 5, the entry and target nodes were respectively defined as $(0, 0, 0)$ and $(71.3, 97.0, 27.8)$ mm in a new map consisting of purple obstacles. In addition, four robot extension steps were also selected to be demonstrated during a random approximate FTL motion executed by the ShapePCE planner.

The ShapeSCE and ShapePCE motion planners are compared with the RRT-shape approach to validate their shape following capability, as shown in Table II. The RRT-shape is an effective methodology to perform an approximate FTL motion for concentric tube robots [15]. As shown in Table II, $J(\mathbf{X})$ of ShapeSCE and ShapePCE are respectively reduced by 29% and 36%, compared to RRT-Shape. It can be clearly seen that ShapePCE results in sequential motion plans within the smallest $\bar{e}_{FTL}$ (less than 3 mm). Due to the elite trajectory selection of the CE estimation, ShapePCE can avoid unstable trajectory sampling generated by a wide range of random configurations by RRT-Shape. In addition, ShapeSCE is more time-consuming and inaccurate compared to ShapePCE, due to its higher dimensional Gaussian mixture model.

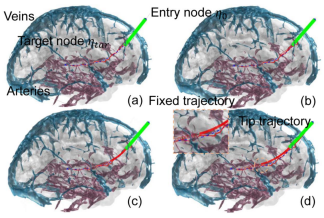

Fig. 6. Safe navigation in the brain under approximate FTL motion by ShapePCE motion planner, demonstrated by four selected insertion steps (a)–(d).

TABLE III
COMPARISON IN CLINICAL SCENARIO NAVIGATION

	$\bar{e}_{FTL}$	Cost $J(\mathbf{X})$	Time t_p	SR
ShapeSCE	6.0 mm	†$(1.56 \pm 0.1) \times 10^4$	†519 ± 116 s	87%
ShapePCE	4.4 mm	†$(1.50 \pm 0.2) \times 10^4$	†321 ± 72 s	93%
RRT-Shape	9.2 mm	$(2.42 \pm 0.2) \times 10^4$	806 ± 236 s	73%

C. Experiment 3: Clinical Scenario

The anatomical environment was segmented from medical images of real patients using intensity-based approaches involving manual labeling, thresholding, and region growing techniques implemented in 3D Slicer. The resulting segmentation was then utilized to generate a point cloud map in MATLAB. The CE motion planners were then implemented to perform an approximate FTL motion in a clinical scenario, which ensures a safe navigation in the brain by avoiding collision with intracranial blood vessels (arteries and veins) during neurosurgical tasks. As shown in Fig. 6, motion plans generated by the ShapePCE approach were demonstrated in a FTL manner. Table III also indicates the ShapePCE motion planner has better shape following performance with shortest t_p, smallest FTL error (<4.4 mm), and highest SR of 95%.

VI. CONCLUSION

In this work, we have proposed a new motion planning framework presented in state-based (ShapeSCE) and parameter-based (ShapePCE) approaches for continuum robots. It is equipped with a cross-entropy (CE) estimation method offering a stochastic model to adaptively realize elite trajectory sampling in constrained environments. The superiority of the presented CE motion planner is demonstrated in both collision-free motion and FTL motion compared to existing state-of-the-art motion planners for continuum robots, leading to less computation time, lower sample cost, and better shape following capability. Although ShapePCE provides more precise motion plans, ShapeSCE requires less input information which makes it suitable for continuum robots with less sensing capabilities (e.g. only tip positions need to be known). In our future work, the implementations of the CE motion planners for other general continuum surgical robots, such as steerable needles, will also be investigated.

REFERENCES

[1] J. Yan et al., "A continuum robotic cannula with tip following capability and distal dexterity for intracerebral hemorrhage evacuation," *IEEE Trans. Biomed. Eng.*, vol. 69, no. 9, pp. 2958–2969, Sep. 2022.
[2] M. Baumann, S. Leonard, E. A. Croft, and J. J. Little, "Path planning for improved visibility using a probabilistic road map," *IEEE Trans. Robot.*, vol. 26, no. 1, pp. 195–200, Feb. 2010.
[3] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.
[4] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. ICRA. Millennium Conf. IEEE Int. Conf. Robot. Automat. Symposia Proc.*, 2000, pp. 995–1001.
[5] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the RRT," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 1478–1483.
[6] X. Wang, J. Wei, X. Zhou, Z. Xia, and X. Gu, "AEB-RRT*: An adaptive extension bidirectional RRT* algorithm," *Auton. Robots*, vol. 46, no. 6, pp. 685–704, 2022.
[7] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed RRT: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 2997–3004.
[8] S. Patil, J. Burgner, R. J. Webster, and R. Alterovitz, "Needle steering in 3-D via rapid replanning," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 853–864, Aug. 2014.
[9] J. Hoelscher et al., "Backward planning for a multi-stage steerable needle lung robot," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 3987–3994, Apr. 2021.
[10] M. Fu, A. Kuntz, R. J. Webster, and R. Alterovitz, "Safe motion planning for steerable needles using cost maps automatically extracted from pulmonary images," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4942–4949.
[11] W. Sun, S. Patil, and R. Alterovitz, "High-frequency replanning under uncertainty using parallel sampling-based motion planning," *IEEE Trans. Robot.*, vol. 31, no. 1, pp. 104–116, Feb. 2015.
[12] M. Pinzi, S. Galvan, and F. R. y Baena, "The adaptive hermite fractal tree (AHFT): A novel surgical 3D path planning approach with curvature and heading constraints," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 14, pp. 659–670, 2019.
[13] A. Favaro, L. Cerri, S. Galvan, F. R. Y. Baena, and E. De Momi, "Automatic optimized 3D path planner for steerable catheters with heuristic search and uncertainty tolerance," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 9–16.
[14] A. Favaro, A. Segato, F. Muretti, and E. De Momi, "An evolutionary-optimized surgical path planner for a programmable bevel-tip needle," *IEEE Trans. Robot.*, vol. 37, no. 4, pp. 1039–1050, Aug. 2021.
[15] K. Wu, L. Wu, and H. Ren, "Motion planning of continuum tubular robots based on centerlines extracted from statistical atlas," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 5512–5517.
[16] L. G. Torres, C. Baykal, and R. Alterovitz, "Interactive-rate motion planning for concentric tube robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014, pp. 1915–1921.
[17] B. H. Meng, I. S. Godage, and I. Kanj, "RRT*-based path planning for continuum arms," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 6830–6837, Jul. 2022.
[18] B. H. Meng, I. S. Godage, and I. Kanj, "Smooth path planning for continuum arms," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 7809–7814.
[19] B. Ouyang, Y. Liu, H.-Y. Tam, and D. Sun, "Design of an interactive control system for a multisection continuum robot," *IEEE/ASME Trans. Mechatron.*, vol. 23, no. 5, pp. 2379–2389, Oct. 2018.
[20] J. Lai, B. Lu, Q. Zhao, and H. K. Chu, "Constrained motion planning of a cable-driven soft robot with compressible curvature modeling," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 4813–4820, Apr. 2022.
[21] M. Bentley, C. Rucker, and A. Kuntz, "Interactive-rate supervisory control for arbitrarily-routed multitendon robots via motion planning," *IEEE Access*, vol. 10, pp. 80999–81019, 2022.
[22] M. Kobilarov, "Cross-entropy motion planning," *Int. J. Robot. Res.*, vol. 31, no. 7, pp. 855–871, 2012.
[23] J. Suh, J. Gong, and S. Oh, "Fast sampling-based cost-aware path planning with nonmyopic extensions using cross entropy," *IEEE Trans. Robot.*, vol. 33, no. 6, pp. 1313–1326, Dec. 2017.
[24] Y. Liu, J. Chen, J. Liu, and X. Jing, "Nonlinear mechanics of flexible cables in space robotic arms subject to complex physical environment," *Nonlinear Dyn.*, vol. 94, pp. 649–667, 2018.
[25] R. Rubinstein, "The cross-entropy method for combinatorial and continuous optimization," *Methodol. Comput. Appl. Probability*, vol. 1, pp. 127–190, 1999.
[26] S. R. Buss, "Introduction to inverse kinematics with Jacobian transpose, pseudoinverse and damped least squares methods," *IEEE J. Robot. Automat.*, vol. 17, no. 16, pp. 1–19, Apr. 2004.