Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

Sampling-Based Follow-the-Leader Motion Planning for Manipulator-Mounted Continuum Robots

Chengnan Shentu*, Nicholas Baldassini*, Oluwagbotemi D. Iseoluwa, Radian Gondokaryono, Jessica Burgner-Kahrs
University of Toronto
https://continuumroboticslab.github.io/sb-ftl-cr-planner/

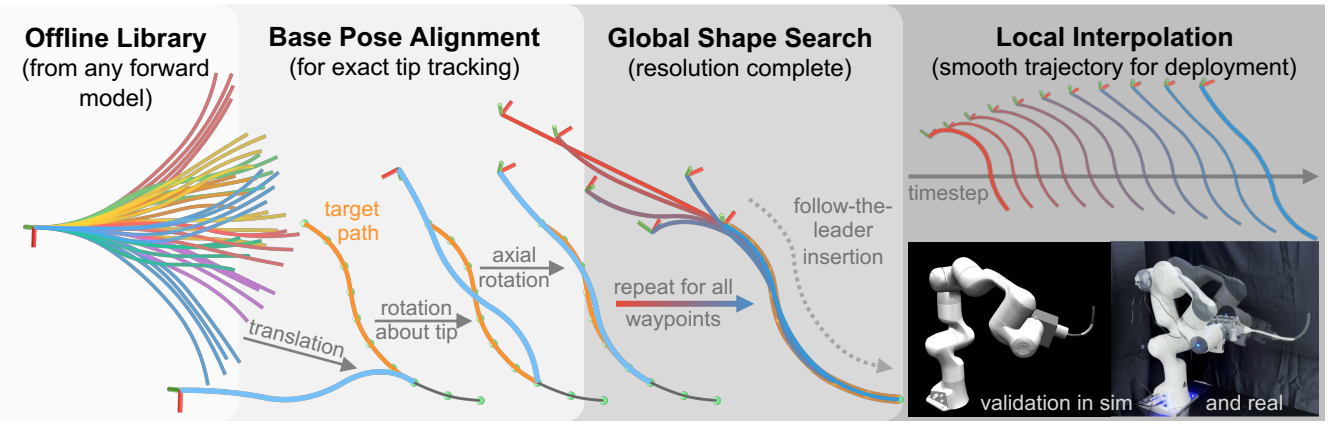

Fig. 1: **Overview** of our sampling-based follow-the-leader motion planning framework. **Offline Library:** shapes are precomputed from any forward model. **Base Pose Alignment:** for each candidate shape (blue), geometric transformations align the robot tip to the current waypoint (green dots) and shape along the target path (orange curve). Fully-actuated $SE(3)$ base poses are shown as coordinate axes. **Global Shape Search:** the best-matching shape is selected for each waypoint, producing a sparse plan (red to blue gradient indicates progression). **Local Interpolation:** smooth interpolation between waypoints yields a dense trajectory suitable for deployment, validated in simulation and on hardware.

***Abstract*—Follow-the-leader (FTL) motion exploits the unique morphology of continuum robots (CRs) to navigate confined spaces by having the body retrace the path of the tip. While extensively studied, existing FTL methods typically assume a fixed base or a single degree-of-freedom insertion mechanism, limiting their applicability to practical systems in which CRs are mounted on robotic manipulators with fully actuated $SE(3)$ base pose. This paper presents a sampling-based motion planner for FTL motion of manipulator-mounted CRs that jointly considers robot configuration and base pose. The key idea is to decouple global shape search from base pose determination by computing the base pose through a closed-form geometric construction, thereby avoiding iterative optimization during online planning. The approach supports general forward models and enables efficient planning by shifting the majority of computation offline. We establish theoretical guarantees including resolution complete shape search and converging tip tracking throughout waypoint traversal and interpolation. Experiments on 120 simulated paths over 3 test classes demonstrate 0% tip error and 1.9% mean shape deviation (w.r.t. robot length) at 100% success rate. We validate the practicality of our approach on a 6-DOF tendon-driven CR mounted on a serial manipulator. Code and visualization available at the project website.**

* Indicates equal contribution. Correspondence to {c.shentu, nicholas.baldassini}@mail.utoronto.ca.

I. Introduction

Continuum robots (CRs) are flexible, high-DOF robots inspired by biological structures such as elephant trunks and octopus arms, with promising potential for navigating confined, tortuous spaces inaccessible to rigid-link robots [29]. Their unique capabilities have led to increasing deployment in minimally invasive surgery [5, 11], industrial inspection [33] and search-and-rescue operations [38]. A key motion primitive enabling these applications is follow-the-leader (FTL), where the robot body follows the path traced by its tip during insertion [23]. FTL motion minimizes lateral forces on surrounding tissue or structures, reducing trauma in surgical procedures and avoiding collisions in cluttered environments.

FTL motion planning has been studied extensively for various continuum and soft robot architectures. Early work established the foundational concepts for snake-like robots [7]. Steerable needles [10] and vine robots [8] are designed to operate with FTL motion, achieving excellent results because they naturally extend during deployment. Similar principles have been applied to concentric tube robots (CTRs), where FTL is achieved through selection of suitable tube parameters and coordinated tube extension [15, 14, 16, 37].

For tendon-driven continuum robots (TDCRs), two main approaches have emerged. The first adds degrees of freedom

to the TDCR design itself through extensible segments [1], active stiffness modulation [27], magnetic actuation [21], or a combination such as the interlaced design with extensible segments and internal locking mechanism [19, 34]. While effective, these hardware-based solutions are difficult to generalize across applications as the additional mechanisms occupy valuable space that could otherwise accommodate payloads such as sensors, tools, or delivery channels.

The second approach addresses the more general case by introducing degrees of freedom at the TDCR base and coordinating them with the CR's configuration through differential [13], heuristic-based [22, 20], or optimization-based inverse kinematics [24, 12, 36]. However, these methods share common limitations. All rely on simplified kinematics such as the piece-wise constant curvature (PCC) model [18], which does not accurately capture the behavior of physical TDCRs. Additionally, most employ local solvers or iterative optimization, which can converge to suboptimal solutions in the high-dimensional, nonlinear configuration space of CRs and are sensitive to initialization. Such unpredictability can be a limitation for safety-critical applications where predictable planner behavior is important.

These limitations are compounded by a confining assumption shared by nearly all existing methods: the robot base is either fixed in space or constrained to 1-DOF linear insertion. This does not reflect the growing class of practical deployments where continuum robots are mounted on robotic manipulators with full 6-DOF base pose control. In surgical systems, flexible endoscopes have been integrated with robot arms for precise positioning [35, 3, 6]. In industrial inspection, snake-like robots have been mounted on serial manipulators [4], mobile platforms [36], and even aerial vehicles [25] to extend reach and dexterity.

In this paper, we consider the FTL motion planning problem for CRs whose base pose is fully actuated in SE(3), as arises when a CR is mounted on a serial manipulator. Given a spatial path specified by ordered waypoints, the objective is to generate a continuous motion such that (i) the robot tip tracks the path during insertion, and (ii) the inserted portion of the robot body follows the path with minimal deviation. While a fully actuated base offers greater flexibility and expanded motion capability, it also introduces computational challenges by expanding the planning space by six additional dimensions.

We address this problem with a sampling-based framework that decouples global shape search from local interpolation. The key enabler is a closed-form geometric construction that uniquely determines the 6-DOF base pose for any selected shape, guaranteeing exact tip placement without iterative optimization. This decoupling allows shape search to be performed via global search over a precomputed library—compatible with any forward model—while interpolation between waypoints exploits radial symmetry to ensure smooth transitions with a small number of forward model evaluations.

To summarize our contribution, we present a FTL motion planning framework for manipulator-mounted CRs with exact tip tracking by construction. The framework supports general forward models beyond PCC while enabling efficient online planning through user-adjustable clustering. We establish formal guarantees including resolution completeness for shape search and asymptotic convergence for tip tracking, and validate the approach in simulation and on hardware. The implementation is open-sourced to support reproducibility. This is the first FTL method for continuum robots with a fully-actuated $SE(3)$ base and formal planning guarantees.

II. Problem Formulation

Following the standard FTL problem formulation, the desired path is given as an ordered set of 3D waypoints $\mathcal{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n\}$, $\mathbf{w}_i \in \mathbb{R}^3$. Unlike methods that assume specific path representations (e.g., constant-curvature arcs), our formulation accepts arbitrary waypoint sequences, whether user-defined, discretized from continuous curves, or generated by higher-level planners. The shape of a CR in configuration $\mathbf{q}$ is described by backbone points $\mathbf{p}(s)$ for arc length $s \in [0, S]$, computed via a forward model (FM), where $\mathbf{p}(0)$ is the base and $\mathbf{p}(S)$ is the tip. In practice, we discretize this shape as $\mathbf{P} = [\mathbf{p}_0, \ldots, \mathbf{p}_D]^\top \in \mathbb{R}^{(D+1)\times 3}$, with $\mathbf{p}_0 = \mathbf{p}(0)$ and $\mathbf{p}_D = \mathbf{p}(S)$.

We consider a CR mounted to a 6-DOF base, representing practical systems where CRs are attached to serial manipulators. The CR configuration space is denoted $\mathcal{Q} \subset \mathbb{R}^d$, and the combined system configuration space is $\mathcal{K} = SE(3) \times \mathcal{Q}$. Let $f : \mathcal{Q} \to \mathbb{R}^{(D+1)\times 3}$ denote the FM function mapping a CR configuration to its discretized backbone points in the CR base frame. Given a base pose $\mathbf{T}_b \in SE(3)$, the corresponding world-frame backbone points are given by $\mathbf{T}_b \cdot f(\mathbf{q})$.

An FTL motion plan is defined as a sequence of N configurations $\{(\mathbf{T}_{b,1}, \mathbf{q}_1), \ldots, (\mathbf{T}_{b,N}, \mathbf{q}_N)\} \subset \mathcal{K}$ that guides the robot through all waypoints in $\mathcal{W}$. While there are n waypoints, the full motion plan consists of $N \gg n$ configurations to ensure continuous interpolation between waypoints. At each configuration index k, the planner must jointly choose the CR shape parameters $\mathbf{q}_k$ and the base pose $\mathbf{T}_{b,k}$.

As the robot progresses along the path, increasing portions of the CR backbone are considered "inserted". We associate each waypoint $\mathbf{w}_i$ with an insertion depth $s \in [0, S]$, representing the arc length along the backbone that should coincide with the path when the tip reaches $\mathbf{w}_i$.

We define two properties that should hold throughout the trajectory:

- **Tip Tracking:** The tip should be placed at waypoint $\mathbf{w}_i$ when passing through it, and follow a smooth interpolation (e.g., linear) between consecutive waypoints.
- **Shape Following:** The inserted portion of the robot should minimize deviation from the corresponding segment of the path.

A key distinction from classical FTL formulations is that the base pose $\mathbf{T}_b$ is not fixed. Instead, the base pose is free to vary throughout the motion and is derived jointly with the CR configuration. This additional freedom is essential for modeling CRs mounted on serial manipulators, but it

fundamentally alters the FTL problem structure by removing the fixed reference frame typically assumed in prior work.

As a concrete instantiation, this work describes the method for a 3-segment TDCR with configuration space parameterized by Clarke coordinates [17], giving $\mathbf{q} \in \mathbb{R}^6$; the method is generalizable to any CR with any forward model.

III. Method

As discussed in Section I, the high-dimensional configuration space $\mathcal{K} = SE(3) \times \mathcal{Q}$ and nonlinear CR FM make iterative optimization approaches prone to local minima and computationally expensive. Our method addresses this by decoupling the problem into global shape search and local interpolation, yielding three key benefits:

- **Model-agnostic:** FM evaluations occur primarily offline, enabling use of complex kinetostatic models that would be impractical for online optimization.
- **Efficient:** Online planning involves shape evaluation and closed-form geometric construction, with only small number of FM evaluations for interpolation.
- **Theoretically Guaranteed:** The sampling-based formulation provides formal guarantees including resolution completeness (Section IV).

The method proceeds as follows: offline library generation (Section III-A), online base pose alignment and shape search at each waypoint (Sections III-B–III-C), and interpolation between waypoints (Section III-D).

A. Shape Library

1) Library Generation: The shape library is constructed offline by sampling N_{lib} configurations uniformly from the CR configuration space $\mathcal{Q}$. For each sampled configuration $\mathbf{q}_i$, we compute FM to obtain a discretized shape representation. Each shape is stored as a tuple $S^{(i)} = (\mathbf{q}^{(i)}, \mathbf{P}^{(i)})$, where $\mathbf{P}^{(i)} = [\mathbf{p}_0^{(i)}, \ldots, \mathbf{p}_D^{(i)}]^\top \in \mathbb{R}^{(D+1)\times 3}$ contains $D+1$ points along the backbone. D should be larger than n to meaningfully capture the CR shape. D remains as a fixed constant for all shapes. The complete library is $\mathcal{L} = \{S^{(i)}\}_{i=1}^{N_{lib}}$. The shape library approximates the CR's reachable shape space. Since library lookup is much faster than forward model evaluation, this shifts the computational burden offline and enables efficient online planning.

2) Threshold Clustering: At each waypoint, we query $\mathcal{L}$ to find the shape minimizing deviation from the active path. The base case is linear search: evaluating all N_{lib} shapes guarantees finding the optimal shape within the library, but incurs $\mathcal{O}(N_{lib})$ evaluations per waypoint.

To reduce computation, we introduce clustering as an optional mechanism that trades accuracy for speed. Shapes are grouped based on pairwise similarity:

$$M(S^{(a)}, S^{(b)}) = \sum_{j=0}^{D} \|\mathbf{p}_j^{(a)} - \mathbf{p}_j^{(b)}\| \tag{1}$$

Clusters are formed using a similarity threshold γ: iterating through ungrouped shapes, each becomes a cluster center and all shapes within distance γ are added to its clusters, forming local balls of bounded radius in shape space.

The optimal γ depends on the specific CR and its shape distribution. In practice, we empirically choose γ by analyzing k-nearest neighbor distances in the library, targeting approximately $\lfloor \sqrt{N_{lib}} \cdot 1.5 \rfloor$ clusters as a suggested default for the speed-accuracy tradeoff; see Appendix A1 for an ablation study on the effect of cluster size. Setting $\gamma = 0$ recovers the linear search behavior.

During online execution with clustering enabled, we first evaluate only cluster centers, then search exhaustively within the best-scoring cluster. This two-pass strategy reduces complexity from $\mathcal{O}(N_{lib})$ to $\mathcal{O}(\sqrt{N_{lib}})$, at the cost of potentially missing the globally optimal shape if it resides in a cluster whose center scores poorly.

B. Base Pose Alignment

At each waypoint $\mathbf{w}_i$, we query the shape library to find the best-matching shape for the active path $\mathcal{W}_i = \{\mathbf{w}_1, \ldots, \mathbf{w}_i\}$. For each candidate shape $S = (\mathbf{q}, \mathbf{P})$, evaluation involves two steps: (1) align the shape to the path via a closed-form geometric construction that determines the base pose $\mathbf{T}_b$, and (2) compute the shape deviation after alignment. The shape with the minimum deviation is selected. Since alignment and evaluation are independent across shapes, the process is easily parallelizable. To align the shape to the path, the following steps are performed:

1) Active Shape Subset: Since the robot is only partially inserted at waypoint $\mathbf{w}_i$, we first identify the active portion of the shape. Let $|\mathcal{W}_i| = \sum_{j=1}^{i-1} \|\mathbf{w}_j - \mathbf{w}_{j+1}\|$ denote the arc length of the active path. We select the subset of shape points from shape tip $\mathbf{p}_D$ whose arc length best matches $|\mathcal{W}_i|$:

$$m^* = \arg\min_m \left| |\mathcal{W}_i| - \sum_{j=m}^{D-1} \|\mathbf{p}_j - \mathbf{p}_{j+1}\| \right| \tag{2}$$

The active shape subset is $\mathbf{P}_{act} = \{\mathbf{p}_{m^*}, \ldots, \mathbf{p}_D\}$. Intuitively, we select the portion of the shape whose arc length best matches the inserted path length, ensuring meaningful shape-to-path comparison.

2) Base Pose Alignment: We compute the base pose $\mathbf{T}_b \in SE(3)$ through three sequential transformations that align the shape to the path with exact tip placement (Fig. 2).

Transformation 1 (Translation): Translate the shape by $(\mathbf{w}_i - \mathbf{p}_D)$ so the tip coincides exactly with waypoint $\mathbf{w}_i$. This ensures the tip-tracking property by construction.

Transformation 2 (Rotation about tip): Rotate the shape about $\mathbf{w}_i$ to align the base-to-tip directions. Define unit vectors $\mathbf{v}_1 = (\mathbf{w}_i - \mathbf{w}_1)/\|\mathbf{w}_i - \mathbf{w}_1\|$ along the path and $\mathbf{v}_2 = (\mathbf{w}_i - \mathbf{p}_{m^*})/\|\mathbf{w}_i - \mathbf{p}_{m^*}\|$ along the translated shape. The rotation axis is $\mathbf{v} = \mathbf{v}_2 \times \mathbf{v}_1$ and angle is $\theta = \cos^{-1}(\mathbf{v}_1 \cdot \mathbf{v}_2)$. Since this rotation is about $\mathbf{w}_i$, tip placement is preserved.

Transformation 3 (Axial rotation): A rotational degree of freedom remains about the path axis $\mathbf{v}_1$. To resolve this, we select a third correspondence point $\mathbf{w}_k \in \mathcal{W}_i$ that maximizes distance to the line through $\mathbf{w}_1$ and $\mathbf{w}_i$, with corresponding

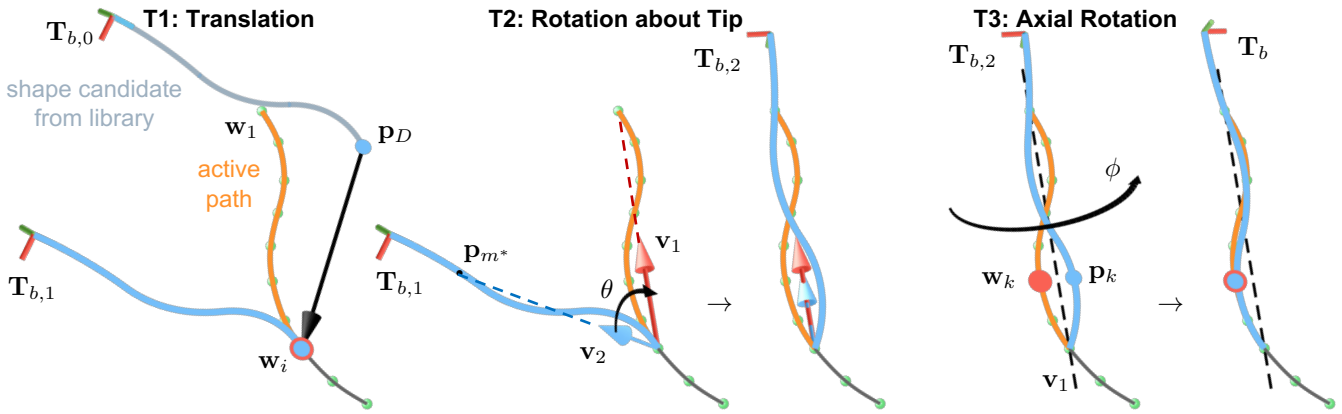

Fig. 2: **Geometric base pose alignment** via three sequential transformations. The robot shape (blue) is aligned to the active path (orange) while guaranteeing exact tip placement at waypoint $\mathbf{w}_i$. **T1:** Translation by $(\mathbf{w}_i - \mathbf{p}_D)$ places the tip exactly at the waypoint. **T2:** Rotation about $\mathbf{w}_i$ by angle θ aligns the tip-to-robot direction $\mathbf{v}_2$ with the path direction $\mathbf{v}_1$. **T3:** Axial rotation about the $\mathbf{v}_1$ axis by angle ϕ resolves the remaining degree of freedom using a third correspondence pair $(\mathbf{w}_k, \mathbf{p}_k)$. Together, these transformations uniquely determine the 6-DOF base pose $\mathbf{T}_b$ with exact tip tracking by construction.

shape point $\mathbf{p}_k$ matched by arc length ratio. The rotation angle ϕ is computed from the projections of $(\mathbf{w}_k - \mathbf{w}_1)$ and $(\mathbf{p}_k - \mathbf{w}_1)$ onto the plane orthogonal to $\mathbf{v}_1$. Since, this rotation is also about $\mathbf{v}_1$ passing through $\mathbf{w}_i$, tip placement is preserved.

Together, Transformations 1–3 uniquely determine all six degrees of freedom of $\mathbf{T}_b$ while guaranteeing $\mathbf{p}_D = \mathbf{w}_i$ by construction. Note that this construction requires at least three non-collinear active waypoints. For the first two waypoints ($\mathbf{w}_1$ and $\mathbf{w}_2$), the base pose can be trivially determined using a home configuration (e.g., straight) or by reusing the shape found at $\mathbf{w}_3$, depending on application requirements.

C. Shape Evaluation

After alignment, we compute the deviation between the transformed shape and the path using the symmetric Chamfer distance:

$$\frac{1}{|\mathcal{W}_i|}\sum_{\mathbf{x}\in\mathcal{W}_i}\min_{\mathbf{y}\in\mathbf{P}_{act}}\|\mathbf{x}-\mathbf{y}\| + \frac{1}{|\mathbf{P}_{act}|}\sum_{\mathbf{y}\in\mathbf{P}_{act}}\min_{\mathbf{x}\in\mathcal{W}_i}\|\mathbf{y}-\mathbf{x}\| \quad (3)$$

where $\mathbf{P}_{act}$ denotes active shape after alignment. We choose symmetric Chamfer distance as it is robust to differences in point density between the path and shape discretizations, and penalizes deviations bidirectionally. The shape minimizing the deviation cost is selected, and its associated $\mathbf{T}_b$ forms the complete configuration $(\mathbf{q}, \mathbf{T}_b) \in \mathcal{K}$.

This formulation can be viewed as nearest-neighbor search in a task-induced metric space: rather than measuring distance in configuration space as in classical sampling-based planners, we evaluate candidates by task performance (shape deviation) after analytically aligning the base pose. Our formulation readily accommodates flexible cost design. For instance, different weights can be assigned to waypoints along the path, e.g., prioritizing recent waypoints as the robot progresses to emphasize accuracy near the tip. Additional terms such as base movement penalties, orientation change from the previous step, or collision constraints may also be incorporated to encourage smooth motion or address application-specific requirements.

D. Interpolation

At this stage, we have a sparse motion plan with one configuration $(\mathbf{q}_i, \mathbf{T}_b^i)$ at each waypoint $\mathbf{w}_i$. To execute on real hardware, this sparse plan must be converted into a continuous trajectory in both CR joint space and base pose space. However, naive interpolation between consecutive configurations can result in large deviations from the FTL path, as the intermediate shapes may not align well with the path.

The main challenge is that consecutive configurations may differ significantly in both joint state and base pose (Fig. 3), requiring large coordinated motions to maintain FTL path tracking. We address this in two steps: first, we exploit the radial symmetry of CRs to pre-align consecutive configurations, reducing orientation discontinuities; then, we interpolate smoothly in joint space while constructing base poses that guarantee exact tip placement throughout.

1) Exploiting Radial Symmetry: CRs exhibit radial symmetry about their backbone axis: rotating the robot about this axis produces a shape in a different bending plane without changing the shape itself. We exploit this to pre-align consecutive configurations before interpolation.

Given the sparse plan $\{(\mathbf{q}_j, \mathbf{T}_b^j)\}_{j=1}^n$, we rotate each configuration to maximize alignment with a common reference direction (e.g., the base x-axis of the first configuration, x_1). The optimal rotation angle θ_j is computed in closed form by maximizing the dot product between the rotated x-axis and the reference direction:

$$\theta_j = \operatorname{atan2}\left(-\mathbf{x}_1^\top \mathbf{y}_j,\ \mathbf{x}_1^\top \mathbf{x}_j\right) \quad (4)$$

where $\mathbf{x}_j$ and $\mathbf{y}_j$ are the x and y axes of the base orientation $\mathbf{T}_b^j$. Depending on the CR design, three types of radial symmetry determine how θ_j is applied:

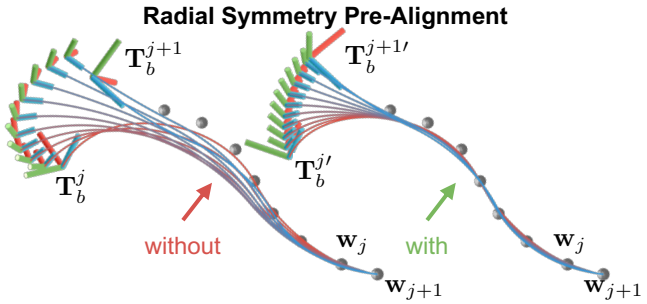

Fig. 3: **Pre-alignment via radial symmetry.** Fanned frames show interpolated configurations between waypoints $\mathbf{w}_j$ (red shape) and $\mathbf{w}_{j+1}$ (blue shape). Left: without alignment, differing bending planes cause large base rotations during interpolation. Right: pre-aligning reduces orientation discontinuity while preserving shape, resulting in better shape following.

1. Continuous: For CRs with fully symmetric actuation (e.g., PCC), θ_j is applied directly.
2. Discrete: For TDCRs with k tendons per segment, θ_j is snapped to the nearest $360°/k$ increment.
3. Data-driven: When analytical symmetry properties are unknown, we identify radially symmetric shapes empirically. Using the same threshold-based clustering method from Section III-A, we cluster shapes that have been rotated to align their tips to a common plane. Shapes within the same cluster are considered radially symmetric variants, allowing substitution with a similar shape with a more favorable base orientation, denoted $(\mathbf{q}'_j, \mathbf{T}_b^{j\prime})$.

2) Local Interpolation Between Waypoints: Between waypoints $\mathbf{w}_j$ and $\mathbf{w}_{j+1}$, we generate h intermediate configurations to ensure smooth motion for practical deployment. The key idea is to define a smooth tip trajectory first, then compute the base pose that places the tip exactly on this trajectory.

We linearly interpolate the tip position and use SLERP [30] for tip orientation, both obtained from the sparse plan, yielding a smooth desired tip pose ${}^k\mathbf{T}_{tip}$ at each interpolation step $k \in \{0, \dots, h\}$ with $\alpha = k/h$:

$$ {}^k\mathbf{w} = (1-\alpha)\mathbf{w}_j + \alpha\mathbf{w}_{j+1} \tag{5} $$

$$ {}^k\mathbf{R}_{tip} = \text{SLERP}(\mathbf{R}_{tip}^{j\prime}, \mathbf{R}_{tip}^{(j+1)\prime}, \alpha) \tag{6} $$

Simultaneously, we linearly interpolate the CR joint state (after pre-alignment via radial symmetry):

$$ {}^k\mathbf{q}' = (1-\alpha)\mathbf{q}'_j + \alpha\mathbf{q}'_{j+1} \tag{7} $$

Evaluating $f({}^k\mathbf{q}')$ gives the tip pose $\mathbf{T}_{tip}({}^k\mathbf{q}')$ in the CR's local frame. To place the tip exactly at the desired pose ${}^k\mathbf{T}_{tip}$, we construct the base pose as:

$$ {}^k\mathbf{T}_b = {}^k\mathbf{T}_{tip} \cdot \mathbf{T}_{tip}({}^k\mathbf{q}')^{-1} \tag{8} $$

This construction guarantees exact tip tracking at every interpolation step. Each step requires one FM evaluation, and with h steps between each of $n-1$ waypoint pairs, the complete motion plan consists of $N = (n-1)\cdot h + 1$ configurations with $(n-1)\cdot h$ total FM evaluations—a fixed, predictable cost independent of library size.

IV. Theoretical Guarantees

We analyze the theoretical properties of our motion planner, establishing resolution completeness and characterizing tip-tracking optimality. We present the analysis for the base case of linear search over the shape library. Section IV-D discusses the speed-accuracy tradeoff when using clustered search.

A. Assumptions

We require the following mild conditions:

- **A1** The CR configuration space $\mathcal{Q}$ is compact.
- **A2** C^1 Forward Model: $f : \mathcal{Q} \to \mathcal{S}$ is continuously differentiable on $\mathcal{Q}$, where $\mathcal{S}$ is the space of robot shapes.
- **A3** Smooth Feasibility: For a feasible path $\mathcal{W}$, there exists a continuous function $\mathbf{q}^* : [0,1] \to \mathcal{Q}$ such that the corresponding shapes achieve bounded shape deviation at each insertion depth.

A1 holds for all practical robots with bounded joint limits. A2 is satisfied by standard CR models: PCC, Cosserat rod, FEM, and neural networks with smooth activations [28]. A2 further implies a deterministic forward model. A3 formalizes the existence of a smooth FTL motion and is standard in sampling-based planning.

B. Resolution Completeness

A motion planning algorithm is *resolution complete* if, for a given discretization level, it is guaranteed to find a valid plan when one exists and terminate in finite time [2]. In our framework, the discretization level corresponds to the discrete resolution library size N_{lib}, not the continuous configuration space. The key insight is that if a feasible configuration $\mathbf{q}^*$ exists, a sufficiently large library will contain a nearby configuration; Lipschitz continuity of shape deviation then ensures this neighbor achieves comparable performance, yielding a valid plan. We formalize this below.

Lemma 4.1 (Coverage Probability): Let $\mathcal{L}_N = \{\mathbf{q}_1, \dots, \mathbf{q}_N\}$ be i.i.d. uniform samples from compact $\mathcal{Q}$. For any $\epsilon_C > 0$ and $\delta > 0$, there exists N_0 such that for $N \geq N_0$:

$$ \mathbb{P}[\mathcal{L}_N \text{ is an } \epsilon_C\text{-cover of } \mathcal{Q}] \geq 1 - \delta \tag{9} $$

Intuitively, uniform sampling eventually places points densely throughout $\mathcal{Q}$, ensuring every configuration has a nearby sample. See Appendix B for the full proof.

Lemma 4.2 (Lipschitz Continuity of Shape Deviation): Under A1–A2, the shape deviation $E(S, \mathcal{W}_i)$ after optimal base pose alignment is Lipschitz continuous in the configuration:

$$ |E(\mathbf{q}_1, \mathcal{W}_i) - E(\mathbf{q}_2, \mathcal{W}_i)| \leq L_E \|\mathbf{q}_1 - \mathbf{q}_2\| \tag{10} $$

for some constant $L_E < \infty$.

Proof. Under A1–A2, the forward model f is continuously differentiable on compact $\mathcal{Q}$, so ∇f is bounded and f is Lipschitz by the mean value theorem. The base pose construction (Transformations 1–3) depends continuously on the

shape points, and shape deviation (3) is continuous in the transformed shape. The composition of Lipschitz functions is Lipschitz. See Appendix C for the full proof. □

Theorem 4.3 (Resolution Completeness): Under A1–A3, for any $\delta > 0$, there exists library size N_0 such that for $N_{lib} \geq N_0$:

$$\mathbb{P}[\text{valid plan found} \mid \text{feasible plan exists}] \geq 1 - \delta \quad (11)$$

Proof. By A3, at each waypoint $\mathbf{w}_i$ there exists $\mathbf{q}^*(i) \in \mathcal{Q}$ achieving shape deviation below some threshold ϵ^*. By Lemma 4.1, for $N_{lib} \geq N_0$, with probability at least $1 - \delta$, there exists $\hat{\mathbf{q}}_i \in \mathcal{L}_{N_{lib}}$ satisfying $\|\mathbf{q}^*(i) - \hat{\mathbf{q}}_i\| < \epsilon_C$. By Lemma 4.2:

$$E(\hat{\mathbf{q}}_i, \mathcal{W}_i) \leq E(\mathbf{q}^*(i), \mathcal{W}_i) + L_E \epsilon_C < \epsilon^* + L_E \epsilon_C \quad (12)$$

The planner selects the configuration minimizing $E(\cdot, \mathcal{W}_i)$, achieving deviation at most $\epsilon^* + L_E \epsilon_C$. As $N_{lib} \to \infty$, $\epsilon_C \to 0$, so the achieved deviation approaches ϵ^*. □

C. Tip-Tracking Convergence

A central property of base pose alignment (Transformations 1–3) is that exact tip placement is guaranteed by geometric construction, independently of the selected shape and without iterative optimization:

Theorem 4.4 (Exact Tip Tracking By Construction): For any selected configuration $\mathbf{q} \in \mathcal{Q}$:

1) At each waypoint $\mathbf{w}_i$, the tip position satisfies $\|\mathbf{p}_D - \mathbf{w}_i\| = 0$.
2) At each interpolation step k between waypoints $\mathbf{w}_j$ and $\mathbf{w}_{j+1}$, the tip position satisfies $\mathbf{p}_D^{(k)} = (1-\alpha)\mathbf{w}_j + \alpha \mathbf{w}_{j+1}$ exactly, where $\alpha = k/h$.

Proof. Property (1): Transformation 1 translates the shape by $(\mathbf{w}_i - \mathbf{p}_D)$, placing the tip at $\mathbf{w}_i$. Transformations 2–3 rotate about $\mathbf{w}_i$, preserving this constraint. Property (2): The interpolated base pose is constructed as ${}^k\mathbf{T}_{tip} \cdot \mathbf{T}_{tip}({}^k\mathbf{q})^{-1}$ where ${}^k\mathbf{T}_{tip}$ has position component ${}^k\mathbf{w} = (1-\alpha)\mathbf{w}_j + \alpha \mathbf{w}_{j+1}$, ensuring the tip matches the interpolated waypoint. □

While tip error is exactly zero at interpolation steps, hardware execution requires continuous joint motion between them. During this inter-step motion, the tip may deviate from the linear FTL path. We show this deviation is bounded and vanishes with finer interpolation:

Theorem 4.5 (Asymptotic Tip-Tracking Convergence): Between consecutive interpolation steps, the tip deviation from the linearly-interpolated FTL path satisfies:

$$\max_{\beta \in [0,1]} \|\mathbf{p}_{tip}(\beta) - \mathbf{p}_{ftl}(\beta)\| \leq \frac{C}{h^2} \Delta_{max}^2 \quad (13)$$

where h is the number of interpolation steps, Δ_{max} bounds the configuration change between waypoints, and C depends on FM smoothness. Consequently, as $h \to \infty$, the supremum tip error over the entire trajectory vanishes.

Proof. The tip error is zero at both endpoints of each inter-step interval (Theorem 4.4). Define the error $\mathbf{e}(\beta) = \mathbf{p}_{tip}(\beta) - \mathbf{p}_{ftl}(\beta)$. Since $\mathbf{e}(0) = \mathbf{e}(1) = \mathbf{0}$ and $\mathbf{e}$ is smooth (by A1–A2′), the maximum deviation satisfies $\max_\beta \|\mathbf{e}(\beta)\| \leq \frac{1}{8} \max_\beta \|\mathbf{e}''(\beta)\|$ (up to dimensional constants). Since $\|\mathbf{e}''(\beta)\|$ scales as $(\Delta_{max}/h)^2$, the $\mathcal{O}(1/h^2)$ bound follows. See Appendix D for the full derivation. □

D. Computational Tradeoff with Clustering

The theoretical guarantees established above require linear search over the full shape library $\mathcal{L}$, incurring $\mathcal{O}(N_{lib})$ shape evaluations per waypoint. The clustered variant (Section III-A2) reduces this to approximately $2\sqrt{N_{lib}}$ evaluations, but sacrifices resolution completeness: if the optimal configuration resides in a cluster whose center scores poorly, that cluster will not be explored.

This presents a user-adjustable tradeoff: linear search retains full guarantees, while clustering enables faster planning at the cost of potential suboptimality. Crucially, tip-tracking guarantees (Theorems 4.4–4.5) remain valid regardless, as they depend only on base pose alignment. In practice, we observe minimal increase in shape deviation with significant reduction in computation time (Section V), suggesting clustering is a viable choice for most applications.

V. Evaluation

We evaluate our method in three settings of increasing fidelity: (1) comparison against an optimization-based baseline using a PCC kinematic model, (2) simulation with a system-identified pseudo-rigid-body model in MuJoCo, and (3) hardware demonstration on a physical tendon-driven CR mounted on a 7-DOF manipulator. Together, these experiments validate our method's accuracy, efficiency, and practical applicability.

A. Metrics

We evaluate performance using three metrics:

- **Tip deviation**: distance between the CR tip and the target waypoint, normalized by robot length (%)
- **Shape deviation**: symmetric Chamfer distance (Eqn. 3) between the inserted robot shape and the active path, normalized by robot length (%)
- **Compute time**: wall-clock time to generate a complete motion plan (seconds)

B. PCC Model: Comparison with Optimization Baseline

1) Setup: We compare our method against an optimization-based FTL baseline using a piece-wise constant curvature (PCC) kinematic model. The baseline minimizes shape deviation using `SciPy`'s L-BFGS-B solver [32], initialized with the solution from the previous waypoint or the zero configuration at the first waypoint. This sequential optimization approach aligns with prior FTL methods [24, 13].

For our method, the shape library contains $N_{lib} = 20,000$ configurations uniformly sampled from $\mathcal{Q}$, chosen empirically to balance speed and accuracy. The CR has 3 segments of unit length each (total length 3).(Section V-C3 evaluates the effect of library size). We evaluate both the linear search variant and the clustered variant ($\gamma = 2.5$), with continuous radial symmetry for pre-alignment. All experiments were run on a 13th Gen Intel Core i7-13700F with 32GB memory.

TABLE I: Evaluation results across 120 paths with PCC. Our method outperforms the optimization-based baseline with exact tip tracking. The clustered variant maintains competitive shape deviation with significantly reduced computation time. Tip and shape deviation are reported as percentages of the total robot length (3 unit lengths).

	Tip Deviation (%) ↓			**Shape Deviation (%) ↓**			**Compute Time (s) ↓**		
Test Class	Optimization	Linear	Clustered	Optimization	Linear	Clustered	Optimization	Linear	Clustered
C Curves	1.24	**0.0**	**0.0**	1.74	**1.25**	1.61	41.3	36.4	**1.81**
S Curves	1.21	**0.0**	**0.0**	2.36	**1.94**	2.09	39.3	46.7	**2.35**
Robot Curves	0.62	**0.0**	**0.0**	1.77	**1.71**	2.03	43.4	46.9	**2.32**

2) Test Paths: We evaluate on 120 paths across three curve classes (40 per class):

- **C Curves**: half circles with start at origin and endpoint sampled from $[0.5, 1.5] \times [-0.75, 0.25] \times [1, 2]$, with bending plane uniformly sampled from $[0, 2\pi]$
- **S Curves**: planar cubic Bézier curves with start at origin and endpoint sampled from $[-2.25, -1.25] \times [-0.5, 0.5] \times \{1.5\}$, with control points forming an S shape
- **Robot Curves**: 3D paths generated by FM on configurations sampled from a bounded subset of $\mathcal{Q}$, representing scenarios where the goal shape is known-feasible (e.g., retraction from a current configuration); they test planning quality rather than shape generalization.

Each path has $n = 10$ waypoints with $h = 10$ interpolation steps between consecutive waypoints.

3) FTL Performance: Table I summarizes the results. Our method achieves 0% tip deviation across all test classes, confirming the exact tip tracking guarantee (Theorem 4.4). For shape deviation, the linear search variant outperforms the optimization baseline on all curve classes, confirming the hypothesis that sampling-based methods are better suited for FTL planning since they avoid local minima. Notably, the clustered variant reduces computation time by over an order of magnitude while sacrificing minimal shape accuracy.

C. MuJoCo Simulation: Pseudo-Rigid-Body Model

1) Setup: We evaluate our method using a custom implementation of a pseudo-rigid-body (PRB) model in MuJoCo [31], where the CR is modeled by a series of 30 rigid links connected by elastic universal joints. System parameters such as joint stiffness are identified from a physical 3-segment tendon-driven CR to capture realistic deformation behavior, including segment coupling effects not modeled by PCC. The simulated CR is mounted on a 7-DOF manipulator (Franka Robotics) with a constant offset transformation $\mathbf{T}_{\text{offset}}$ accounting for the actuation unit and hardware mount. For each planned base pose $\mathbf{T}_b$, the manipulator end-effector pose is $\mathbf{T}_{ee} = \mathbf{T}_b \cdot \mathbf{T}_{\text{offset}}^{-1}$, which is converted to joint-level commands for the Franka robot using an inverse kinematics solver [9].

We test the clustered variant with shape library size of $N_{lib} = 20,000$ and $\gamma = 2.5$. We use the same three curve classes (C, S, Robot) with 40 paths per class, randomly initialized with the same distributions as in V-B2.

2) Results: Table II summarizes the results. The method achieves 0% tip deviation across all test classes. Shape deviation is slightly higher than the PCC experiments, reflecting the increased complexity of the PRB model with segment coupling and only three-fold discrete radial symmetry for pre-alignment. These results demonstrate that our method scales effectively to more realistic CR models: given any forward model, the planner generates motion plans with guaranteed tip tracking and competitive shape following.

TABLE II: Evaluation results across 120 paths with MuJoCo simulation using a system-identified PRB model. Our method achieves exact tip tracking and reasonable shape deviation despite increased model complexity. Tip and shape deviation are reported as percentages of the total robot length.

	Tip Dev. (%) ↓	**Shape Dev. (%) ↓**	**Time (s) ↓**
Test Class	Clustered	Clustered	Clustered
C Curves	0.0	2.46	2.07
S Curves	0.0	3.17	2.69
Robot Curves	0.0	2.43	2.85

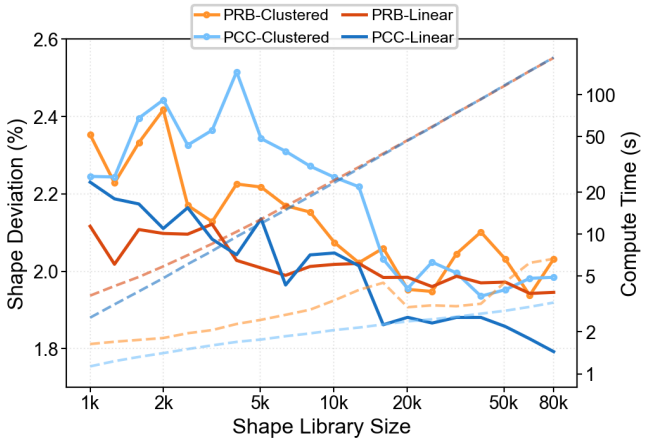

Fig. 4: **Tradeoff between library size, accuracy, and computation time.** As N_{lib} increases, shape deviation (solid lines) decreases with diminishing returns, while computation time (dashed lines) grows linearly for linear search. Clustering (light colors) maintains computation time with small accuracy loss. Results shown for both PCC (blue) and PRB (orange).

3) Effect of Shape Library Size: Fig. 4 illustrates the effect of the shape library size N_{lib} on shape-following accuracy and computation time for both PCC and PRB models. As N_{lib} increases, shape deviation decreases with diminishing

returns: rapid improvement is observed for small to moderate library sizes, followed by saturation as the library becomes sufficiently dense. This behavior is consistent with the resolution completeness analysis in Theorem 4.3, which predicts improved approximation of feasible shapes with increasing library resolution. The computation time grows linearly with N_{lib} for the linear variant. In contrast, the clustered variant maintains nearly constant computation time across library sizes while achieving comparable shape deviation, confirming the effectiveness of the two-pass search strategy (Section IV-D). The gap between PCC and PRB curves reflects the increased model complexity, but both exhibit the same qualitative trends. For time-critical applications, clustering provides an effective mechanism to maintain this accuracy with over an order of magnitude reduction in online computation.

D. Hardware Demonstration

1) Setup: We demonstrate our method on a physical setup consisting of a 3-segment tendon-driven CR mounted on a 7-DOF manipulator (Franka Robotics). The CR has 3 tendons per segment with parallel tendon routing. Motion plans are generated using the MuJoCo PRB model from Section V-C.

2) Execution: The joint-level motion plan is post-processed using `toppra` [26] for time-optimal path parameterization, reducing jerk while ensuring synchronized execution between the CR and manipulator. The resulting trajectories are sent to servo motors with built-in PID controllers (Dynamixel XL430-W250-T) for tendon actuation, and a joint-impedance controller for the serial manipulator. We executed representative paths from each of the three curve classes.

3) Results: Fig. 5 shows the hardware execution for C, S, and robot curve paths. The robot successfully tracks all paths, with the manipulator and CR moving in coordination throughout the trajectory. We measured tip tracking error using an optical tracker (Lyra, NDI, Canada), with a single marker attached to the CR tip and a four-marker rigid body defining the base frame relative to the manipulator base. Across the three paths, normalized by the CR length (187.5 mm), the C-shape achieved a mean error of 5.5% (max 8.0%), the S-shape 9.1% mean (max 20.6%), and the robot-curve shape 8.3% mean (max 20.9%), with visible deviations between planned and executed shapes.

In simulation, the planner achieves zero tip error by construction, since waypoints are matched exactly against the same kinematic model used for execution. The hardware error therefore reflects the gap between this model and the physical system rather than a limitation of the planner itself. Unmodelled effects such as tendon stretch, friction, and hysteresis cause the physical robot to bend less than predicted, and because our pipeline executes plans open-loop from the system-identified model, these discrepancies accumulate without correction. The resulting hardware metrics are effectively a noisy version of the simulation results, with the noise attributable to system-specific factors (modeling fidelity, controller performance) that would not transfer across platforms. We therefore report these tip errors as a reference point for the specific hardware realization rather than as an evaluation of planner quality. The simulation experiments in Sections V-B–V-C isolate planner performance, while the hardware demonstration validates that our method produces smooth, executable trajectories that achieve qualitatively correct FTL motion on a physical manipulator-mounted CR system.

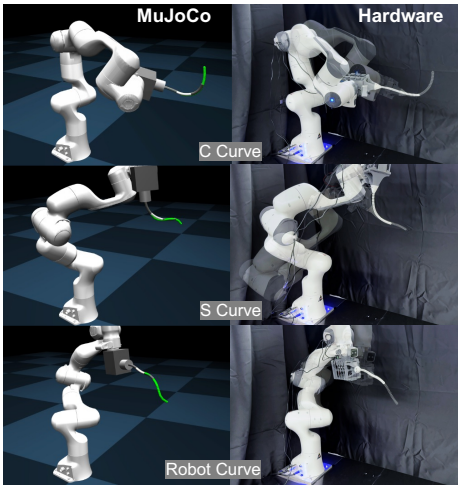

Fig. 5: **Simulated (left) and hardware (right) evaluation** on three path classes. The setup consists of a 3-segment tendon-driven CR mounted on a 7-DOF manipulator. Hardware images overlay two frames: solid (mid-trajectory) and semi-transparent (final configuration), illustrating the coordinated base motion during FTL execution. See supplementary video for full execution sequences.

Closing this gap is orthogonal to the planning framework and can be approached from several directions: improving model fidelity through richer parametrizations of tendon-driven dynamics, more thorough system identification, generating the shape library directly from hardware-recorded shapes rather than from a model (which would absorb unmodelled effects into the library itself), or adding closed-loop shape control during execution. See the supplementary video for full execution sequences.

VI. Conclusion

This work introduced a sampling-based motion planning framework for follow-the-leader motion of manipulator-mounted continuum robots with full 6-DOF base pose control. By decoupling global shape search from local interpolation, the approach achieves exact tip tracking by design while supporting general forward models that would be impractical for iterative optimization. This decoupled structure also enables formal guarantees that optimization-based approaches cannot provide: resolution completeness from sampling-based search, exact tip tracking from geometric base pose construction, and asymptotic tip-tracking convergence from interpolation. User-adjustable clustering further provides a principled tradeoff between computation time and shape accuracy.

A. Limitations and Future Directions

The current work assumes the forward model is independent of base pose and external loading, whereas gravity can induce pose-dependent shape changes; pose-dependent shape prediction or active gravity compensation could address this. The per-step nature of shape search precludes trajectory-wide guarantees on shape-following quality; jointly optimizing across waypoints would provide stronger guarantees, though at the cost of substantially expanding the planning space. The uniform sampling strategy for library generation does not guarantee uniform coverage in shape space; importance sampling or adaptive refinement targeting shape space coverage could improve library quality for complex CRs without increasing library size. Finally, closed-loop shape control during execution, combined with more accurate forward models, would improve hardware execution fidelity.

More broadly, this work shows that FTL motion for manipulator-mounted CRs can be addressed without iterative optimization by decoupling shape search from base pose construction. We believe this opens the door to scalable FTL planning for increasingly complex CR systems, including hybrid rigid–continuum platforms.

Acknowledgments

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), [funding reference number RGPIN-2025-04343]. This research was also supported by the Canada Foundation for Innovation (CFI) through the John R. Evans Leaders Fund (JELF) [project number 40110].

References

[1] Ernar Amanov, Thien-Dang Nguyen, and Jessica Burgner-Kahrs. Tendon-driven continuum robots with extensible sections—a model-based evaluation of path-following motions. *The International Journal of Robotics Research*, 40(1):7–23, 2021.

[2] Jerome Barraquand and Jean-Claude Latombe. Robot motion planning: A distributed representation approach. *The International Journal of Robotics Research*, 10(6): 628–649, 1991.

[3] Pierre Berthet-Rayne, Gauthier Gras, Konrad Leibrandt, Piyamate Wisanuvej, Andreas Schmitz, Carlo A Seneci, and Guang-Zhong Yang. The i2 snake robotic platform for endoscopic surgery. *Annals of Biomedical Engineering*, 46(10):1663–1675, 2018.

[4] Rob Buckingham and Andrew Graham. Nuclear snake-arm robots. *Industrial Robot*, 39(1):6–11, 2012.

[5] Jessica Burgner-Kahrs, D. Caleb Rucker, and Howie Choset. Continuum robots for medical applications: A survey. *IEEE Transactions on Robotics*, 31(6):1261–1280, 2015.

[6] Jian Chen, Mingcong Chen, Qingxiang Zhao, Shuai Wang, Yihe Wang, Ying Xiao, Jian Hu, Danny Tat Ming Chan, Kam Tong Leo Yeung, David Yuen Chung Chan, et al. Design and visual servoing control of a hybrid dual-segment flexible neurosurgical robot for intraventricular biopsy. In *IEEE International Conference on Robotics and Automation*, pages 5906–5912. IEEE, 2024.

[7] Howie Choset and Wade Henning. A follow-the-leader approach to serpentine robot motion planning. *Journal of Aerospace Engineering*, 12(2):65–73, 1999.

[8] Margaret M. Coad, Laura H. Blumenschein, Sadie Cutler, Javier A. Reyna Zepeda, Nicholas D. Naclerio, Haitham El-Hussieny, Usman Mehmood, Jee-Hwan Ryu, Elliot W. Hawkes, and Allison M. Okamura. Vine robots. *IEEE Robotics and Automation Magazine*, 27(3):120–132, 2020.

[9] Peter Corke and Jesse Haviland. Not your grandmother's toolbox–the robotics toolbox reinvented for python. In *IEEE International Conference on Robotics and Automation*, pages 11357–11363. IEEE, 2021.

[10] Noah J Cowan, Ken Goldberg, Gregory S Chirikjian, Gabor Fichtinger, Ron Alterovitz, Kyle B Reed, Vinutha Kallem, Wooram Park, Sarthak Misra, and Allison M Okamura. Robotic needle steering: Design, modeling, planning, and image guidance. In *Surgical Robotics: Systems Applications and Visions*, pages 557–582. Springer, 2010.

[11] Pierre E Dupont, Nabil Simaan, Howie Choset, and Caleb Rucker. Continuum robots for medical interventions. *Proceedings of the IEEE*, 110(7):847–870, 2022.

[12] GuoHua Gao, Pengyu Wang, and Hao Wang. Follow-the-leader motion strategy for multi-section continuum robots based on differential evolution algorithm. *Industrial Robot*, 48(4):589–601, 2021.

[13] Yuanqian Gao, Kiyoshi Takagi, Takahisa Kato, Naoyuki Shono, and Nobuhiko Hata. Continuum robot with follow-the-leader motion for endoscopic third ventriculostomy and tumor biopsy. *IEEE Transactions on Biomedical Engineering*, 67(2):379–390, 2020.

[14] Arnau Garriga-Casanovas and Ferdinando Rodriguez y Baena. Complete follow-the-leader kinematics using concentric tube robots. *The International Journal of Robotics Research*, 37(1):197–222, 2018.

[15] Hunter B. Gilbert, Joseph Neimat, and Robert J. Webster. Concentric tube robots as steerable needles: Achieving follow-the-leader deployment. *IEEE Transactions on Robotics*, 31(2):246–258, 2015.

[16] Cedric Girerd, Andrey V. Kudryavtsev, Patrick Rougeot, Pierre Renaud, Kanty Rabenorosoa, and Brahim Tamadazte. Slam-based follow-the-leader deployment of concentric tube robots. *IEEE Robotics and Automation Letters*, 5(2):548–555, 2020.

[17] Reinhard M. Grassmann and Jessica Burgner-Kahrs. Clarke coordinates are generalized improved state parametrization for continuum robots. *IEEE Robotics and Automation Letters*, 10(10):10362–10369, 2025.

[18] Bryan A Jones and Ian D Walker. Kinematics for multisection continuum robots. *IEEE Transactions on Robotics*, 22(1):43–55, 2006.

[19] Byungjeon Kang, Risto Kojcev, and Edoardo Sinibaldi. The first interlaced continuum robot, devised to intrinsically follow the leader. *PloS One*, 11(2):e0150278, 2016.

[20] Congjun Ma, Quan Xiao, Liangcheng Liu, Xingxing You, and Songyi Dian. Geometric iterative approach for efficient inverse kinematics and planning of continuum robots with a floating base under environment constraints. *arXiv preprint arXiv:2503.14848*, 2025.

[21] Liyang Mao, Yang Peng, Chenyao Tian, Xingjian Shen, Feihao Wang, Hao Zhang, Xianghe Meng, and Hui Xie. Magnetic steering continuum robot for transluminal procedures with programmable shape and functionalities. *Nature Communications*, 15, 2024.

[22] Abdelkhalick Mohammad, Matteo Russo, Yihua Fang, Xin Dong, Dragos Axinte, and James Kell. An efficient follow-the-leader strategy for continuum robot navigation and coiling. *IEEE Robotics and Automation Letters*, 6 (4):7493–7500, 2021.

[23] Maria Neumann and Jessica Burgner-Kahrs. Considerations for follow-the-leader motion of extensible tendon-driven continuum robots. In *IEEE International Conference on Robotics and Automation*, pages 917–923, 2016.

[24] David Palmer, Salvador Cobos-Guzman, and Dragos Axinte. Real-time method for tip following navigation of continuum snake arm robots. *Robotics and Autonomous Systems*, 62(10):1478–1485, 2014.

[25] Rui Peng, Yu Wang, Minghao Lu, and Peng Lu. A dexterous and compliant aerial continuum manipulator for cluttered and constrained environments. *Nature Communications*, 16(1):889, 2025.

[26] Hung Pham and Quang-Cuong Pham. A new approach to time-optimal path parameterization based on reachability analysis. *IEEE Transactions on Robotics*, 34(3):645–659, 2018.

[27] Chen Qian, Tangyou Liu, and Liao Wu. Jammingsnake: A follow-the-leader continuum robot with variable stiffness based on fiber jamming. *IEEE/ASME Transactions on Mechatronics*, 30(4):3100–3107, 2025.

[28] Priyanka Rao, Quentin Peyron, Sven Lilge, and Jessica Burgner-Kahrs. How to model tendon-driven continuum robots and benchmark modelling performance. *Frontiers in Robotics and AI*, 7 - 2020, 2021. ISSN 2296-9144.

[29] Graham Robinson and J. Bruce C. Davies. Continuum robots-a state of the art. In *IEEE iInternational Conference on Robotics and Automation*, volume 4, pages 2849–2854, 1999.

[30] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, 1985.

[31] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

[32] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[33] Mingfeng Wang, Xin Dong, Weiming Ba, Abdelkhalick Mohammad, Dragos A. Axinte, and Andy Norton. Design, modelling and validation of a novel extra slender continuum robot for in-situ inspection and repair in aeroengine. *Robotics and Computer-Integrated Manufacturing*, 67, 2019.

[34] Peiyi Wang, Sheng Guo, Fuqun Zhao, Xiangyang Wang, and Majun Song. Follow-the-leader deployment of the interlaced continuum robot based on the unpowered lock mechanism. In *Intelligent Robotics and Applications: 14th International Conference*, page 448–459. Springer-Verlag, 2021. ISBN 978-3-030-89133-6.

[35] Paul Wilkening, Farshid Alambeigi, Ryan J. Murphy, Russell H. Taylor, and Mehran Armand. Development and experimental evaluation of concurrent control of a robotic arm and continuum manipulator for osteolytic lesion treatment. *IEEE Robotics and Automation Letters*, 2(3):1625–1631, 2017.

[36] Quan Xiao, Congjun Ma, Xuke Zhong, Yuqi Zhu, Xingxing You, and Songyi Dian. Design, modeling and motion planning of a mobile continuum robot for in-situ inspection and maintenance in gas-insulated switchgear. *Robotics and Computer-Integrated Manufacturing*, 99: 103205, 2026.

[37] Yunti Xu, Connor Watson, Jui-Te Lin, John T Hwang, and Tania K Morimoto. Shape control of concentric tube robots via approximate follow-the-leader motion. *IEEE Robotics and Automation Letters*, 9(8):7198–7205, 2024.

[38] Yu Yamauchi, Yuichi Ambe, Hikaru Nagano, Masashi Konyo, Yoshiaki Bando, Eisuke Ito, Solvi Arnold, Kimitoshi Yamazaki, Katsutoshi Itoyama, Takayuki Okatani, et al. Development of a continuum robot enhanced with distributed sensors for search and rescue. *Robomech Journal*, 9(1):8, 2022.