Computationally Efficient Lookahead Search for Contact-aided Navigation for Tendon-driven Continuum Robots

Priyanka Rao[1], Itai Spigelman[2], Oren Salzman[2] and Jessica Burgner-Kahrs[1]

I. Introduction

Tendon-Driven Continuum robots (TDCRs), are adept at navigating cluttered environments due to their flexible and compliant backbones. While multi-segment designs allow variations in backbone curvature, they also increase the complexity of actuation unit design and tendon control. We propose the use of a single-segment long TDCR, leveraging obstacle contacts to adjust curvature, along with a motion planner [1] to determine the sequence of actuation inputs to reach a target orientation. A major bottleneck for the computational efficiency for these searches is the forward model of these TDCRs. We propose a lookahead motion planner adapted from Lazy Receding Horizon A* (LRA*) [2] that employs a novel crude model for *lazy* estimation of the robot shape, and consequently the cost-to-go. The proposed method aims to direct the search effectively towards promising regions while minimizing the need for costly full model computations.

Modeling the behavior of TDCRs under contact forces is complex due to the need to consider both tendon and contact interactions. Its continuous backbone requires an infinite number of parameters for exact representation which can be reduced by applying geometrical assumptions, allowing for simpler state-space representations. A backbone experiencing multiple contacts can be divided into a series of portions, each experiencing a force at the tip. Linear curvature profiles approximated by Euler arc splines (EAS) have been demonstrated [3] to approximate the backbone shape under the influence of a tip force. Using piecewise first-order polynomials [4] for the curvature profile has been shown to provide a good approximation of the robot shape, this representation is yet to be utilized to provide computationally efficient approximation of the TDCRs contact mechanics. By utilizing a series of EAS for the backbone representation, the proposed crude model employs a simplified parameterization to faster computations.

II. Lookahead Search

We consider the same motion planning problem as defined in [1] of a planar single-segment TDCR in a 2D workspace, $\mathcal{W} \in \mathbb{R}^2$, populated with o known obstacles. A *joint-space value*, $\mathbf{q} = (\ell_{\text{seg}}, \ell_{\text{ten}})$ denotes the total inserted length of the robot in the workspace and the tendon length, respectively. A *joint-space action* $\delta = (\delta_{\text{seg}}, \delta_{\text{ten}})$ adjusts the robot's base and tendon lengths respectively by insertion/retraction and pulling/releasing actions. We will use σ to denote a *sequence* of such joint-space actions. Actions leading to outcomes that fail to meet the problem's constraints by the model are termed invalid actions, whereas those that produce feasible solutions are classified as valid actions. The goal is to find a sequence of valid joint space actions σ that when applied to an initial joint-space value $\mathbf{q}_{\text{init}}$, align the robots end-effector position $\mathbf{p}(\mathbf{q}_{\text{init}} + \sigma)$ and orientation $\psi(\mathbf{q}_{\text{init}} + \sigma)$ with target position and orientations, within specified tolerances for position ε and orientation ω.

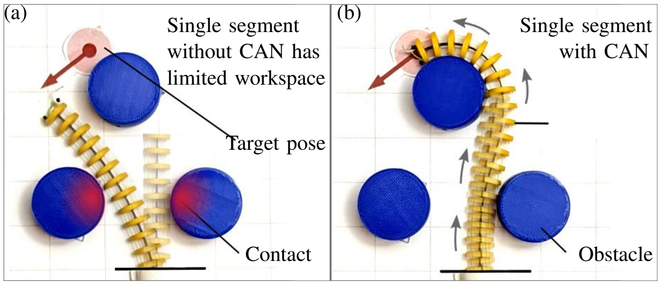
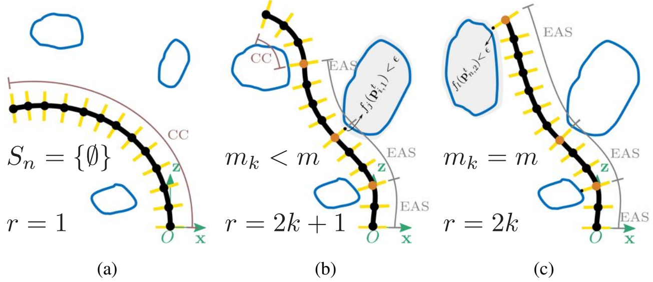


Fig. 1: Navigating a cluttered environment to reach a target by a single-segment (a) with obstacle avoidance : cannot reach the target pose, (b) with CAN : reaches target pose successfully.

Algorithm: The motion planner utilizes a best-first search on a discretized representation of the continuous joint space. Each node n is uniquely identified by the pair $(\mathbf{q}^n, \kappa^n)$, where $\mathbf{q}^n = \mathbf{q}_{\text{init}} + \delta_1 + \ldots + \delta_k$ denotes a joint-space vector and κ^n represents the individual curvatures of the robot's sub-segments. While the primitive operations, data structures, and heuristic remain consistent with those proposed in [1], the main algorithm is adapted to employ a hybrid model evaluation strategy for each node, alternating between the computationally expensive but accurate full model [5] and the faster, less accurate crude model.

Initialization involves populating an open list with the node corresponding to $\mathbf{q}_{\text{init}}$, followed by expansion in a best-first manner according to the heuristic estimate of each node. Nodes within a certain depth α from the last full model evaluation undergo shape approximation using the crude model . The estimated end-effector location from this approximation provides a heuristic estimate $\bar{h}(n)$, which is then inserted into the open list. Upon a node's depth exceeding α (indicated by its budget $b[n]$), $\mathcal{K}_f(n)$ is used to evaluate its true heuristic value $h(n)$, which is then reinserted into the open list for re-prioritization. Node expansion occurs only upon reselection, and if the crude model fails, the full model is utilized to re-evaluate the node, resetting its budget value to zero. This iterative process continues until either a

[1] Continuum Robotics Laboratory, Department of Mathematical & Computational Sciences, University of Toronto, ON, Canada
[2] Department of Computer Science, Technion, Haifa 3200003, Israel
priyankaprakash.rao@mail.utoronto.ca.



(a) (b) (c)

Fig. 2: Diagrammatic representation of a TDCR interacting with obstacles in different scenarios (i) as a *contact-free* configuration (ii) *in-contact* with a freely moving end (iii) *in-contact* configuration with its tip in contact with an obstacle.

node evaluated by the full model reaches the goal, or the maximum number of search iterations is attained, signifying an unsuccessful search.

Proposed Crude Model: We propose a crude model for node expansion, assuming the contact profile of neighboring node c mirrors that of node n. This method represents the backbone with EAS curves, creating a solvable model. Contact points divide the segment, each with a tip force. If the last disk has no contact, a *contact-free* portion exists. We use CC and EAS to describe the segment's curvature, requiring one and two curvature parameters for *in-contact* and *contact-free* portions, respectively. The segment between the base and a subsequent contact point, and between every pair of contact points, is represented as an EAS [3]. Between a contact point and the last disk, the segment is modeled as a CC arc. For a *contact-free* configuration, the entire segment is depicted as a CC arc.

For a given node n, we denote $S_n = \{m_1, m_2, ..m_k\}$ as the subsegments in contact with obstacles, where $m_i = 0, 1, 2, ..m$ indicates the index of the disk in contact with an obstacle. This set is determined by assessing whether the positions of the disks fall within a specified tolerance, denoted as $\epsilon_{\text{contact}}$, relative to the proximity of all obstacles. In the event of a configuration being *contact-free* , S_n is empty. Consequently, three possible scenarios arise, as delineated in Fig. 2. The optimization structure follows that of the full model [1], with r curvature parameters as unknowns instead of m. Initialization involves using node n's curvature parameters as an initial guess, approximated as piecewise linear and constant curvature profiles using linear regression. If the contact profile shifts between nodes n and c, indicating model failure, the full model is computed to accurately determine curvature and contact profile.

III. Empirical Evaluation

In our study, we examine the computational efficiency of lookahead search techniques applied to the case studies detailed in [1]. The baseline method involves a greedy search without lookahead, where the forward model (full model) is executed at every node during node expansion ($\mathcal{M}_{\text{search, f}}$). To evaluate the effectiveness of the proposed lookahead search, particularly the proposed model (crude model), we conduct experiments using two different models for heuristic estimation: 1) the proposed crude model , denoted as $\mathcal{M}_{\text{lookahead, c}}$, and 2) the same full model as in [5], but with relaxed optimization constraints, referred to as $\mathcal{M}_{\text{lookahead, f}}$. The full model considers function and constraint tolerances of the order 1×10^{-10} to ensure the obtained robot shape is precise, while both models used for node approximation consider tolerances of the order 1×10^{-6}. Both approaches employ a lookahead depth of $\alpha = 5$ nodes.

We perform 75 searches in two workspaces from [1]: one with uniform circular obstacles ($\mathcal{W}_1$) and another resembling the inside of a jet engine ($\mathcal{W}_2$). Implemented in C++ using NLopt's sequential quadratic programming (SQP) algorithm, models run on an x86 16-core processor at 2.2 GHz. Results are tabulated in Table I. Inclusion of lookahead search reduces search times compared to the baseline greedy search, with average model computation decreasing due to lightweight models. Slightly lower success rates suggest errors in lightweight model approximations may divert the search from solution. Investigating the effect of the lookahead factor and incorporating duplicate detection to avoid prolonged exploration of nearby nodes may alleviate this issue.

The performance of the two lookahead searches is comparable. While it's plausible to suggest that relaxing tolerances could enhance the computation time of the full model, leveraging the proposed crude model provides an additional advantage of being applicable to static models as it respects the physics behind TDCR bending [3]. In static models, achieving force and moment balance equations necessitates careful consideration, making the relaxation of tolerances less straightforward. In future work, we intend to assess the performance of this search approach on static models. Furthermore, we aim to investigate the impact of hyperparameters such as α and tolerances to further refine and optimize the search process.

TABLE I: Search Times (in s) and success rate (in brackets in %) for workspaces $\mathcal{W}_1$ and $\mathcal{W}_2$.

		Avg.	Median	Max.
$\mathcal{W}_1$	$\mathcal{M}_{\text{search, f}}$ (82.7%)	89.9	19.1	575.7
	$\mathcal{M}_{\text{lookahead, f}}$ (74.7%)	88.2	19.6	431.8
	$\mathcal{M}_{\text{lookahead, c}}$ (74.7%)	85.7	16.3	379.5
$\mathcal{W}_2$	$\mathcal{M}_{\text{search, f}}$ (86.7%)	264.1	98.1	2264.9
	$\mathcal{M}_{\text{lookahead, f}}$ (85.3%)	186.9	66.7	1644.5
	$\mathcal{M}_{\text{lookahead, c}}$ (84.0%)	184.5	51.9	1747.3

References

[1] P. Rao, O. Salzman, and J. Burgner-Kahrs, "Contact-aided navigation for tendon-driven continuum robots," 2024. [Online]. Available: https://arxiv.org/abs/2402.14175

[2] A. Mandalika, O. Salzman, and S. Srinivasa, "Lazy receding horizon A* for efficient path planning in graphs with expensive-to-evaluate edges," *Proceedings International Conference on Automated Planning and Scheduling, ICAPS*, pp. 476–484, 2018.

[3] P. Rao, Q. Peyron, and J. Burgner-Kahrs, "Shape representation and modeling of tendon-driven continuum robots using euler arc splines," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8114–8121, 2022.

[4] O. Al-Ahmad, M. Ourak, J. Vlekken, and E. Vander Poorten, "FBG-Based Estimation of External Forces Along Flexible Instrument Bodies," *Frontiers in Robotics and AI*, vol. 8, pp. 1–12, 2021.

[5] K. P. Ashwin, S. Kanti, and A. Ghosal, "Profile and contact force estimation of cable-driven continuum robots in presence of obstacles," *Mechanism and Machine Theory*, vol. 164, p. 104404, 2021.