Minimizing the Fréchet Error of Task-Space Paths for Manipulators and Surgical Robots

Oren Salzman*, Rachel Holladay†, Sherdil Niyaz‡, Alan Kuntz§, Ron Alterovitz§ and Siddhartha Srinivasa‡
*Carnegie Mellon University, †Massachusetts Institute of Technology,
‡University of Washington, §The University of North Carolina at Chapel Hill

I. Introduction

We consider the problem of following a given path in *task space*—the space of all position of a robot's end effector. Here, the objective is to compute a path in the *configuration space* ($\mathcal{C}$-space), the space defined by the position and orientation of each of the robot's joints, "as best as possible" (we will formally define this shortly). We are motivated by settings where the task-space path is provided, but it is not clear if there exists a collision-free $\mathcal{C}$-space path that exactly traces it. This can occur due to obstacles in the workspace or due to kinematic constraints of the robot. One example where such a setting occurs is when robot manipulators operate in household environments performing tasks such as serving a cup of coffee. In order not to spill the coffee, the robot's end-effector has to stay roughly upright. A second example that motivates our work comes from medical robots such as steerable needles and concentric tube robots [3]. In this application, we envision the surgeon providing the reference path in task space and our algorithm producing the $\mathcal{C}$-space path for either the tip of the tube robot or the bevel edge of the steerable needle that follows the reference path.

There exists a multitude of planners that allow to follow paths in task space: Berenson et al. [1] present a $\mathcal{C}$-space planner that handles multiple constraints, including end-effector constraints. Yao and Gupta [9] use randomized gradient descent and a search that alternates between task space and $\mathcal{C}$-space using local-trajectory tracking to enforce the constraint. These local trajectory-tracking methods are similar to methods that leverage the null space of the Jacobian matrix [8]. Oriolo et al. [7] detail the limitations with these kinematic control methods.

Unfortunately, these methods do not optimize according to a metric that enforces the goal of the task: to follow a path. In this paper we employ the Fréchet distance, widely used in computational geometry, as a natural way to measure the distance between paths in task space [2]. Our key insight is that using Fréchet distance to measure error allows us to efficiently organize an anytime search for a path that closely follows our desired task-space path.

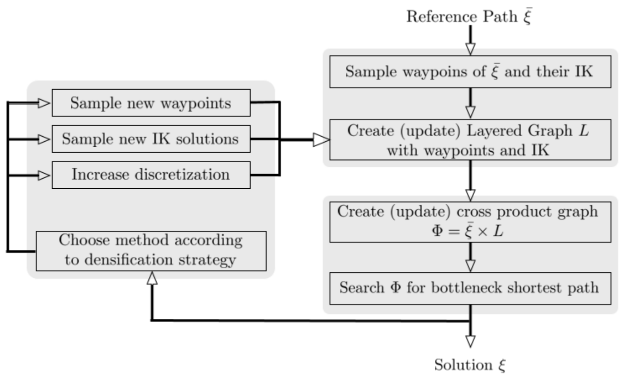

Reference Path $\bar{\xi}$
Sample waypoins of $\bar{\xi}$ and their IK
Create (update) Layered Graph L with waypoints and IK
Create (update) cross product graph $\Phi = \bar{\xi} \times L$
Search Φ for bottleneck shortest path
Solution ξ
Sample new waypoints
Sample new IK solutions
Increase discretization
Choose method according to densification strategy

Fig. 1: On top is a flowchart of our algorithm. We create data structures that allow us to efficiently compute a path that minimizes the Fréchet distance to the reference path and then to incrementally reduce this distance. Each grey box outlines a major step: (1) generating candidate paths, (2) searching over paths and (3) densifying.

The Fréchet distance was proposed as a method for measuring distance between task-space paths by Holladay and Srinivasa [5], who used trajectory optimization to produce a path that closely follows the reference path. However, their approach suffers from the fact that it maps each task-space point to one arbitrary $\mathcal{C}$-space point in the set of inverse kinematic (IK) solutions. Instead, we search over the space of IK solutions, approximating the search space by a layered graph that organizes IK solutions by their task-space location along the path. Further borrowing from computational geometry, we can efficiently compute the Fréchet distance between candidate paths in the layered graph and our reference path by treating them as a set of simplicial complexes [4]. We can efficiently search the cross product of our two complexes with a simple variant of Dijkstra's graph-search algorithm.

II. Algorithmic Approach

Our algorithmic approach, depicted in Fig. 1 can be summarized as follows: Given a reference path $\bar{\xi}$ in task space, we generate a set of candidate paths by sampling waypoints along $\bar{\xi}$. For each waypoint, we compute a set of different IK solutions to create a layered graph L. Vertices of this graph correspond to configurations and we define a layer as all configurations whose forward kinematics map to the same waypoint. The edges of

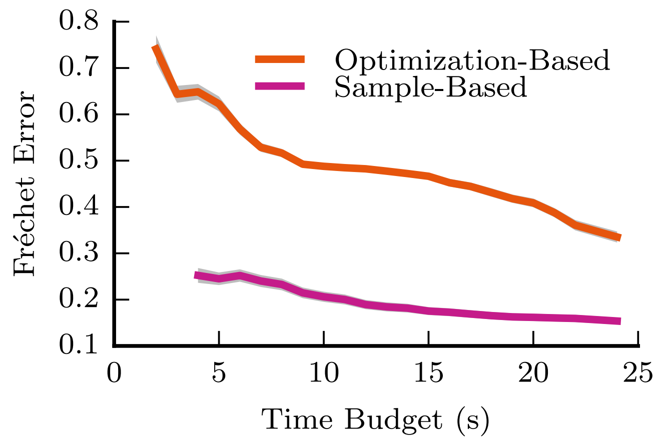
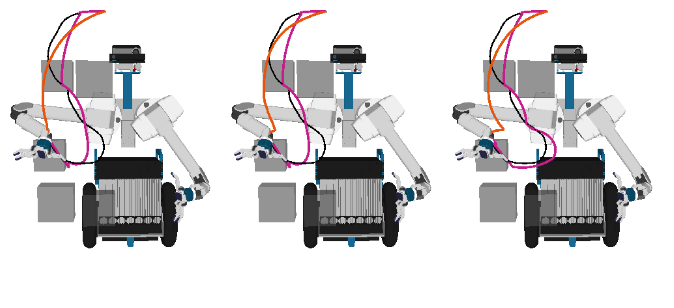


Fig. 2: Quality of the solution produced by our sample-based approach and the optimation-based method used by Holladay and Srinivasa [5].



(a) Initial Paths (b) Midway Progress (c) Final Paths

Fig. 3: We show the progression of the optimization-based approach (orange) and the sample-based approach (pink) as they try to follow the reference path (black). Randomly generated obstacles in the environment are shown in grey. These figures only capture the differences in position, not orientation.

this graph connect any two configurations that belong to the same or to adjacent layers. In order to obtain the path $\xi \in L$ that minimizes the Fréchet distance with the reference path $\bar{\xi}$, we compute a cross-product graph $\Phi = \bar{\xi} \times L$. Following Har-Peled and Raichel [4], we compute the minimal-bottleneck path in Φ. This, in turn, induces the path $\xi \in L$ that indeed minimizes the Fréchet distance with $\bar{\xi}$. To improve the quality of the solution in an online fashion, we employ various densification strategies to produce new candidate paths. We can efficiently incorporate these new paths by updating our graphs and searching again the cross-product graph.

III. Evaluation

A. Manipulation tasks

To evaluate our approach, we start by comparing our sampling-based algorithm with the optimization-based approach presented by Holladay and Srinivasa [5]. We compare the two methods by examining the anytime performance in Fig. 2 and their progression in Fig. 3. To compare anytime performance, we query each planner after t seconds for their best solution so far.

While the optimization-based approach finds an initial solution faster, its Fréchet error solution is significantly higher than the one found by our sampling-based approach. As shown in Fig. 3a, the initial solution of the optimization-based approach (shown in orange) poorly captures the reference path (black) compared to the sample-based approach (pink). While both approaches continue to improve their solutions over time, the optimization-based approach does so at a faster rate. However, given the fixed time budget, the sample-based approach produces a better quality solution. Given more time, we would expect both solution to continue to improve.

Our sample-based approach is able to converge to a path that more closely follows the reference path because it searches over sets of IK solutions and it leverages the Fréchet distance to efficiently search. Indeed, we can show that under some assumptions, our algorithm's output will converge to a path that exactly traces the reference paths [6].

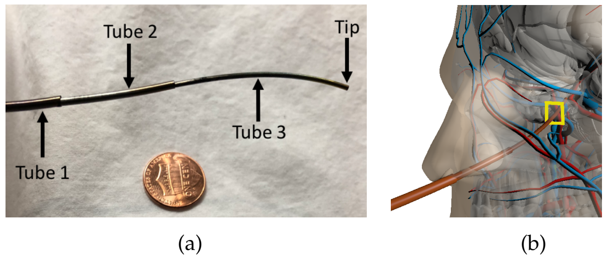


Fig. 4: To perform minimally-invasive surgery through anatomical regions that are highly constrained, such as accessing the pituitary gland in the brain via the sinuses, the CTR could be inserted through the nasal cavity and a small window cut through the sinuses. We envision the surgeon providing one reference path to reach the sinus, a second to cut an opening through the sinus and a third to reach the brain. In this figure, we see an example of such a window, which could be traced by the surgeon, in purple and the CTR accessing the brain, in orange.

B. Surgical tasks

We are currently exploring the applicability of our approach for the case of minimally invasive surgery by concentric tube robots (CTRs) [3]. These systems are composed of multiple telescoping, concentric, precurved, superelastic tubes that can be axially translated and rotated at their bases relative to one another. They derive bending not from tendon wires or other external mechanisms, but from elastic tube interaction in the backbone itself, permitting high dexterity and small diameter. Controlling such tubes is highly un-intuitive for humans, motivating the need for new user interfaces and efficient planning algorithms

We envision the surgeon providing a reference path and our algorithm producing the actual path for the tip of the tube that follows the reference path.

For a depiction of the robot and the reference path, see Fig. 4. For videos demonstrated initial results in simulation, see goo.gl/tzjvEd.

References

[1] Dmitry Berenson, Siddhartha Srinivasa, Dave Ferguson, and James Kuffner. Manipulation planning on constraint manifolds. In *ICRA*, pages 625–632. IEEE, 2009.

[2] M Maurice Fréchet. Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, 22(1):1–72, 1906.

[3] Hunter B. Gilbert, D. Caleb Rucker, and Robert J. Webster III. Concentric tube robots: The state of the art and future directions. In *ISRR*, pages 253–269, 2013.

[4] Sariel Har-Peled and Benjamin Raichel. The Fréchet distance revisited and extended. *TALG*, 10(1):3, 2014.

[5] Rachel Holladay and Siddhartha Srinivasa. Distance metrics and algorithms for task space path optimization. In *IROS*, pages 5533–5540. IEEE, 2016.

[6] Rachel M. Holladay, Oren Salzman, and Siddhartha Srinivasa. Minimizing task space frechet error via efficient incremental graph search. *CoRR*, abs/1710.06738, 2017.

[7] Giuseppe Oriolo, Massimo Cefalo, and Marilena Vendittelli. Repeatable motion planning for redundant robots over cyclic tasks. *Transactions on Robotics*, 2017.

[8] Rodney Roberts and Anthony Maciejewski. Repeatable generalized inverse control strategies for kinematically redundant manipulators. *Transactions on Automatic Control*, 38(5):689–699, 1993.

[9] Zhenwang Yao and Kamal Gupta. Path planning with general end-effector constraints. *RAS*, 55(4):316–327, 2007.