Planning Motions for a Planar Robot Attached to a Stiff Tether

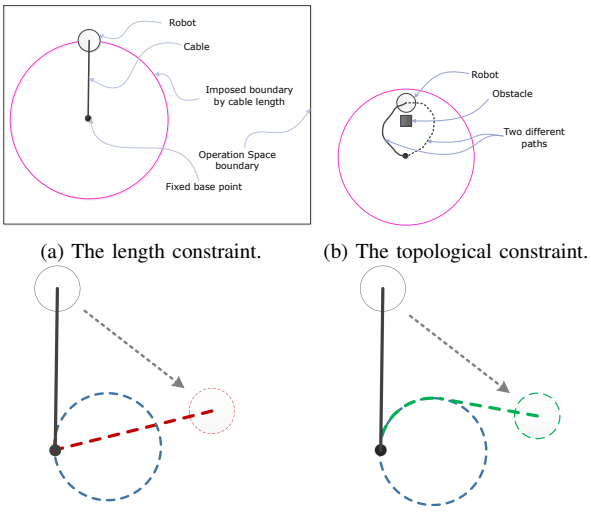
Reza H. Teshnizi and Dylan A. Shell

***Abstract*— There are several practical reasons to endow a mobile robot with a tether, but doing so adds considerable complexity to the problem of moving the robot. The feasibility of a particular motion in such systems depends on topological constraints imposed by the interplay of the robot's tether and its environment. The physical properties of the tether may also rule out configurations that would be possible otherwise. Little work has addressed these latter constraints, despite the considerable interest in motion planning for tethered robots recently. We examine the problem of planning motions of a planar robot connected via a cable of limited length to a fixed point in $\mathbb{R}^2$ when the tether has a constraint on its curvature, which adds appreciably to the realism of the cable model over existing work. We incorporate Dubins's theory of curves with work on planning with topological constraints to concisely represent the configuration space manifold, leading to an atlas of the manifold consisting of locally continuous charts that represent the cable's curvature limits conveniently. Any configuration of the tether and the robot is described in our representation with two elements: (1) a discrete structure that characterizes the cable's position and (2) an element within a single continuous chart. We provide an algorithm that explores the necessary parts of this atlas on-the-fly to locate paths efficiently.**

I. Introduction

The many practical reasons to tether a mobile robot include: the ability to provide power (electrical, pneumatic, hydraulic) from off-board sources, reliable high-speed communication, and security should the robot need to be physically retrieved after a failure. Cables have also been used for manipulation by helping robots *collect* [1], [2] or *separate* objects [3]. Motivated partly by its practical importance, recent research has begun exploring motion planning for tethered robots [4]–[8]. This body of work considers a simplified model of the tether (*e.g.,* assuming an infinitely thin, frictionless, or taut cable) to build and search over a representation of the configuration space (c-space). Within this vein, we also make several modeling simplifications, but this paper represents a step toward a more realistic model: we take into consideration the potential for limited flexibility, or *stiffness*, in the cable with a parameterized curvature constraint on the cable.

Even the simplest case involving a frictionless, taut and unconstrained flexible cable imposes two constraints on the motion of the robot: (1) the radius of the robot's movement is limited by the cable's length (Fig. 1a), and (2) topological constraints are imposed by the cable and the obstacles in the environment (Fig. 1b). For a stiff cable, an additional third constraint is imposed, not to the robot's motions, but to the feasible cable configurations: (3) the cable cannot be bent around sharp angles (Fig. 1c). One naïve approach is to consider a discretization of the cable which is then forced to abide by the curvature constraint. This technique, however, leads to a high dimensional space and requires computation that will be costly if it is not to be excessively imprecise.



(a) The length constraint. (b) The topological constraint.

(c) The blue dotted circle represents the cable's curvature constraint. Owing to the stiffness of the cable, it does not bend around sharp angles. The gray arrow shows a planned motion for the robot. The figure on the left depicts an invalid configuration for the cable after execution of the motion. The valid configuration is shown on the right.

Fig. 1: The three constraints imposed on the motion of a robot that are caused by a stiff cable.

We take advantage of two complementary ideas to tackle this problem. We generalize the decomposition method proposed in our earlier work [8], which separates topological regularities from the continuous aspects of the c-space. Then we employ Dubins's theory of optimal trajectories under curvature constraints [9] to exclude infeasible cable configurations from the robot-cable c-space. We build a concise representation of this manifold, enabling efficient and complete search for paths. In Section IV, we present an algorithm that explores the c-space while searching it, computing parts of the manifold as and when they are needed. This form of on-line search and exploration minimizes the memory requirements of the method. We also demonstrate that use of parameterized curvature makes the method a general enough technique to solve several related problems.

II. Related Work

Two distinct areas of motion planning research are closely related: planning for tethered mobile robots, and nonholo-

R. H. Teshnizi and D. A. Shell are with the Distributed AI and Robotics Lab, Dept. of Computer Science and Engineering , Texas A&M University, College Station, TX 77843, USA. Emails: {reza.teshnizi@gmail.com, dshell@cse.tamu.edu}

nomic motion planning. It is important to note that the nonholonomic constraint in this problem is not imposed by the robot's mechanics but reflects, instead, the feasible poses of the cable. The robot is treated as controllable in its degrees of freedom (x, y, and θ), but the cable must not be bent beyond a certain limit once anchored.

The standard approach for planning motions of tethered robots is to create a graph approximation of the configuration space that is then searched to find some solution among paths in the same homotopy class. Xavier [10] first utilized the visibility graph to trace back the cable and store changes in the visibility of the vertices in the environment. He constructs a path from the collected vertices and modifies it to its shortest equivalent in the same homotopy class. Grigoriev and Slissenko [11] and Narayanan *et al.* [12] addressed the problem by representing paths in a given homotopy class by defining an alphabet used to describe ray crossing events. Determining whether two paths are homotopy equivalent then requires comparison of their associated strings (the latter paper also addresses homological equivalence). Bhattaharya *et al.* [4] employ the *Cauchy Integral Theorem* in 2D space to define homotopy classes of trajectories. Influenced by the work of Igarashi and Stilman [5], we introduced a technique to decompose the c-space into cells [8]. The cells are then used to create an atlas whose structure is stored in an accompanying graph, which factors out the discrete properties of the c-space from those that are continuous.

There has been considerable work on the general topic of *nonholonomic motion planning*. Dubins [9] proved that any optimal path in an obstacle-free unbounded environment consists of 3 segments which are either straight line segments or circle sectors. Jacobs and Canny [13] provided an algorithm that finds a *δ-robust* approximate shortest path by using a plane sweep technique along with quadtrees. Their method describes a graph that divides the c-space into simple trajectories made up of subpaths that pass between the points on the boundaries of the obstacles. Agarwal *et al.* [14] extended Dubins's theory to deal with environments with so-called moderate obstacles. By improving on Jacobs and Canny's work, the paper provided an algorithm that builds a graph of straight lines and circles. This graph was then searched to find an optimal path. The total time of this method is $O(n^2 \log(n))$, where n is the total number of edges. Their later work [15] took a similar approach to find the shortest path in an obstacle free environment but with polygonal boundaries with n edges in time $O(n^2 \log(n))$. There are also several works addressing the problem from a control perspective (see [16], [17] and the references therein.)

Unlike prior work, we consider both curvature and topological constraints while modeling the robot-cable configurations. We believe the method introduced herein describes the robot-cable configuration space precisely and elegantly, capturing the necessary constraints before the search phase.

III. The Preliminaries

A. Curvature Constraint and Stiff Cables

Definition 1. ***Nonholonomic Constraint:*** *If r is the radius vector of an object in $\mathbb{R}^2$, a constraint f of the form $f(r, \dot{r}, t) = 0$ is nonholonomic if it cannot be expressed as $f(r, t) = 0$ [18].*

Definition 2. ***Average Curvature:*** *For a path $P : I \to \mathbb{R}^2$, the Average Curvature is*

$$\kappa(s) = \frac{d^2 P(s)}{ds^2}, \forall s \in I, \tag{1}$$

and therefore is a nonholonomic condition.

Definition 3. ***Curvature Constraint:*** *If there exists an upper-bound $\kappa_0 \in [0, \infty)$ on the average curvature of a path, we say the path has a curvature constraint of κ_0. That is*

$$\kappa(s) \leq \kappa_0 = \frac{1}{r_0}, \forall s \in I, \tag{2}$$

where r_0 is the radius of the smallest circle that the path can turn around.

Since a stiff cable cannot be bent beyond a certain limit defined by some curvature constraint, κ_0, any robot-cable configuration in which the cable is bent with a radius less than r_0 is infeasible and is excluded from c-space (see Fig. 1c). Notice that equation 2 provides a general model since a cable without curvature constraint can be parametrized as $\lim_{r_0 \to 0} \kappa_0 = \infty$.

B. Problem Statement

Definition 4. ***Pose:*** *A pose, P_k, is a pair $P_k = (p_k, \theta_k)$ where $p_k \in \mathbb{R}^2$ is a point and $\theta_k \in S^1$ representing the location and orientation, respectively. Alternatively, the orientation can be represented by a velocity vector $v_k = (v_{kx}, v_{ky}) \in \mathbb{R}^2$ of unit length where $\theta_k = \arctan \frac{v_{ky}}{v_{kx}}$.*

This work considers the problem of planning from an initial pose, P_i, to a goal pose, P_g, for an oriented point robot[1] situated in $\mathbb{R}^2$ among polygonal obstacles whose vertices are known to the robot. The robot is tethered to a fixed base with pose P_{base}, via a cable with curvature constraint $\kappa_0 = 1/r_0$. It is assumed that the cable has a maximum length l and is always retracted to its tightest form. By *tight* we mean there is no perturbation that can be applied to the path such that it would make the length of the given path shorter without violating either the length or curvature constraints.

C. Shortest Curvature Constrained Paths

Let M be a 1-connected unbounded manifold in $\mathbb{R}^2$ and τ be a path contained in M. Following the terminology in [14], a *C-segment* of τ is a non-empty maximal subpath of τ that has the form of a circular arc with radius r_0; an *L-segment* is a non-empty maximal subpath of τ that has the form of a straight line segment. We say, for example, a path is of type CLC if it is made up of only three segments of types C, L, and C in this order.

Definition 5. ***Shortest Curvature Constrained Path:*** *Denoted by $P_a \rightsquigarrow P_b$, a shortest curvature constrained path*

[1]We discuss how to apply the technique developed in this work to a tethered robot with extent in Section IV-A.

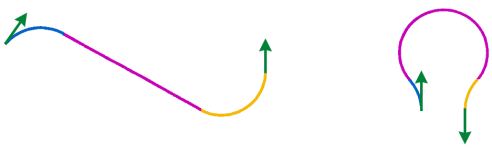
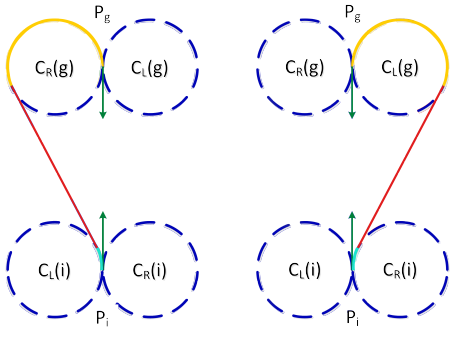


Fig. 2: The two types of Dubins Path. The path on the left is of the type CLC, and the path on the right is of the type CCC. The three segments are colored blue, magenta, and yellow, respectively. The arrows at the beginning and at the end of the paths depict the initial and goal poses, P_i and P_g, respectively.



(a) (b)

Fig. 3: In the case of a symmetry between $C_L(i)$ and $C_L(g)$ with $C_R(i)$ and $C_R(g)$ finding a shortest curvature constraint path makes it possible to find the second shortest curvature constraint path by doing a reflection. Figures 3a and 3b show the two optimal ways of reaching the goal pose when the path begins with turning left first and right first, respectively. Different colors in the paths illustrate their different segments.

is a minimal length path in M from initial pose P_a to goal pose P_b, with curvature constraint κ_0.

Theorem 1. ***Dubins's Theorem [9]:*** *Every shortest curvature constrained path in M is necessarily a path of type CLC or CCC or a substring of either of these (see Fig. 2).*

Let $P_x = (p_x, v_x)$ be a pose in the environment. Let L_x be the line parallel to v_x which passes through p_x. We use the notation $C_L(x)$ and $C_R(x)$ for the circles to the left and right (according to v_x) of L_x which are tangent to L_x at p_x. We will call $C_L(x)$ and $C_R(x)$ the tangent circles to P_x.

Let P_i and P_g be the initial pose and the goal pose respectively. It is important to notice that the shortest curvature constrained path in M, as defined in Definition 5, is not unique in a set of special configurations of P_i and P_g. When p_g is on L_i, there is a symmetry between $C_L(i)$ and $C_L(g)$ with $C_R(i)$ and $C_R(g)$. Therefore, if there is one shortest curvature constrained path $\tau = P_i \rightsquigarrow P_g$, a second path can be created by reflecting τ about L_i (see Fig. 3).

Now let M be an unbounded[2] subspace of $\mathbb{R}^2$ with polygonal obstacles and τ be a curvature constrained path in M. We have C- and L-segments defined as before, but now introduce an O-segment of τ as a maximal segment of τ that lies on the boundary of an obstacle. A C-segment is called *terminal* if it is the first or last segment of a path.

Lemma 1. ***Non-terminal C-segment [14]:*** *A non-terminal C-segment of a locally shortest path[3] is either tangent to at least one obstacle or it is adjacent (on the path) to a terminal C-segment.*

Corollary 1. *If any subpath of a locally shortest path contains O-segments, they are either at the beginning or at the end of the subpath.*

Proof. Follows from Lemma 1. □

Lemma 2. ***Dubins Subpaths [13], [14], [19]:*** *Any subpath of a locally shortest curvature constrained path is itself a shortest curvature constrained path if the subpath does not touch any obstacle.*

D. C-Space Skeleton

Throughout this work, we use the term "*decomposition*" to emphasize the difference between our approach and that of "*discretization*." Our decomposition is a representation of c-space as a graph whose vertices represent the largest subspaces of c-space in which the solution to the motion planning problem is calculated in $O(1)$. Whereas, a discretization method provides an approximation of c-space. A cable induces structure reflected in the c-space manifold which can be exploited in an elegant way: an appropriate decomposition of c-space yields a concise discrete topological skeleton and a set of continuous subspaces. This observation was made in our earlier work [8] where the cable in each subspace is always in the form of a straight line. Here, we construct solutions from simpler subpaths found in subspaces but must now demand that the cable be in the form a shortest path in each subspace as defined by Definition 5. This means the same data structures and path finding algorithms function as before, but with modifications to adapt the method for finding shortest curvature constrained paths. Once this method has been determined, we define the appropriate cell decomposition, as is discussed next.

E. Decomposition Method and Dubins Cells

Since the cable is always taut, it lies on a path from P_{base} to P_g, where each of its segment is a locally shortest path.

Corollary 2. ***Taut Cable Decomposition:*** *Every taut configuration of a curvature-bounded cable that lies on the path τ_c can be decomposed into subpaths, each of which is a Dubins Path.*

Proof. If the cable does not touch any obstacle along its path, τ_c from P_{base} to P_g, then the locally shortest path is also the globally shortest path, and hence following Theorem 1 it is a Dubins path. In this proof, we employ regular expression notation to describe the string representing the C-, L-, and O-segments in a path. Let $[C, L]^*$ be a path composed of zero or more C- or L-segments. Moreover, let $O[C, L]^*$ denote a path beginning with an O-segment and followed by a path of type $[C, L]^*$. $[O[C, L]^*]^+$ denotes one or more consecutive paths of type $O[C, L]^*$. Finally, $C[O[C, L]^*]^+C$ is a path

[2]If M is bounded then we require that the boundary of the environment be define via boundary obstacles.

[3]A path is locally shortest if any perturbation in the path is physically impossible or lengthens the path.

beginning with a C-segment, followed by a path of type $[O[C,L]^*]^+$, and ending in a C-segment. By Corollary 1, if the cable does touch obstacles, then τ_c is always of type $C[O[C,L]^*]^+C$ since the O-segments only occur at the beginning or at the end of each subpath. Then each such subpath is in the form of $O[C,L]^*$. If $[C,L]^*$ is an empty string, then it represents a path of zero length which is trivially a shortest curvature constrained path. Otherwise, since it is a subpath of a locally shortest path and it does not touch any obstacle, by Lemma 2, it is a shortest curvature constrained path. Thus, breaking a locally shortest path into subpaths at the points where obstacles are touched will yield a set of subpaths such that each is a shortest curvature constrained path. □

This decomposition corollary is the basis for constructing a skeleton of the c-space. Since the environment contains semi-algebraic obstacles, it can be partitioned into a semi-algebraic set M_{obs} consisting of all the obstacles and the free space M_{free} that is the complement of M_{obs} [11].

Definition 6. ***Dubins Cell:*** *Given a base pose P_b, a Dubins Cell is a chart, (U,φ), where $U \subseteq M_{free}$; the homeomorphism φ is $\varphi(x,y) = (x,y)$; and every point inside U can be reached from P_b via a shortest curvature constrained path.*

A Dubins Cell can be represented with the following four attributes (see Fig. 4):

- ***Base Pose:*** $P_b = (p_b, v_b)$, where p_b is the location of the base and v_b is the orientation of the cable at p_b.
- ***Parent Cell:*** a reference or pointer to the cell describing the robot-cable configuration directly before occurrence of the O-segment.
- ***Cable Length:*** L, determines the maximum allowed distance between the robot and the base pose. The exact value is $L = Parent.L - Length(Parent.P_b \rightsquigarrow P_b)$
- ***Stitch Line:*** this is the line where one chart is connected to another is the interface between them. Formally, it is the domain for the transition map, φ, between the two charts. Once the robot crosses this line, a contact is made or released and the robot will be transfered from a chart (cell) to the other.

Fig. 5 shows a single Dubins Cell in an environment without any obstacles. Notice how the boundary of farthest reachable points take the shape of an Archimedean spiral, which arises from the curvature constraint. To understand why the boundary is this shape, imagine the cable is anchored with pose $P_0 = (p_0, v_0)$ where $p_0 = [r_0, 0]$ and $v_0 = [0, 1]$. If the robot moves along the boundary of maximal curvature for θ radians, then $(r_0\theta)$ of the cable's length will be consumed. Therefore the polar equation $r = l - r_0\theta$ will describe the farthest boundary of the cell. The general equation can easily be obtained.

F. The Boundaries of a Cell

The curvature constraint means that the boundaries of a cell are dependent on the base pose and goal pose. This subsection gives a better understanding of the boundaries.

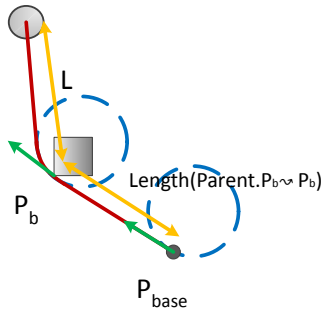


Fig. 4: The defining attributes of a Dubins Cell. The blue dotted circles are the boundaries of maximum curvature. The green arrows are the orientations at the base poses of each cell.

Let $c_l(x)$ and $c_r(x)$ be the centers of $C_L(x)$ and $C_R(x)$ respectively, and let $d(p_x, p_y)$ denote the Euclidean distance in $\mathbb{R}^2$ between the points p_x and p_y. There are four possible conditions regarding the Euclidean distance between the centers of the tangent circles of P_i and P_g.

1) $d(c_l(i), c_l(g)) \geq 4$ and $d(c_r(i), c_r(g)) \geq 4$.
2) $d(c_l(i), c_l(g)) < 4$ and $d(c_r(i), c_r(g)) \geq 4$.
3) $d(c_l(i), c_l(g)) \geq 4$ and $d(c_r(i), c_r(g)) < 4$.
4) $d(c_l(i), c_l(g)) < 4$ and $d(c_r(i), c_r(g)) < 4$.

Notice that condition 2 and condition 3 give the same planar configuration via reflection. Also, condition 4 raises three possible scenarios [20]:

4.1) Shortest curvature constrained path from P_i to P_g is a single C-segment with length less than πr_0, or the concatenation of two C-segments with a total length less than πr_0.
4.2) The four tangent circles enclose a region Ω. The boundary of Ω is concatenation of six circles with radius r_0 (see Fig. 6).
4.3) Shortest curvature constrained path contains either a C-segment with length at least πr_0, or an L-segment with length at least $4r_0$.

We adopt the following four proximity conditions from Ayala and Hyam in [20].

Proximity Condition A. If P_i and P_g satisfy condition 1.
Proximity Condition B. If P_i and P_g satisfy condition 2 or 3.
Proximity Condition C. If P_i and P_g satisfy condition 4.1 or 4.2.
Proximity Condition D. If P_i and P_g satisfy condition 4.3.

A path $\tau : [0,1] \rightarrow \mathbb{R}^2$ is not in Ω if $\exists s \in [0,1]$ such that $\tau(s) \notin \Omega$. Therefore, Ω divides all paths from P_i to P_g into two disjoint sets of paths: $\Delta(\Omega)$ the set of all paths in Ω, and $\Delta'(\Omega)$ the set of all paths not in Ω. The importance of

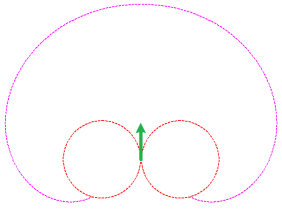
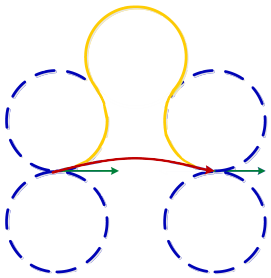
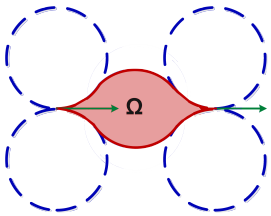


Fig. 5: A single Dubins Cell in an environment without obstacles. The red dotted circles show boundary of maximal curvature. The magenta spirals show the boundary of farthest reachable points (without considering a goal pose). The green arrow shows the orientation of the robot at the base pose. Cable's length is 60 units with curvature constraint $\kappa_0 = 1/20$.



(a) There is no continuous transformation that can convert the yellow path to the red path that can maintain the curvature constraint throughout the transformation. The boundaries of maximum curvature (dotted blue) and the yellow path form a bight.



(b) Any path that leaves the Ω region is not homotopic to the paths that are entirely inside the Ω region.

Fig. 6: The Ω region divides the set of all path going from P_i to P_g into two disjoint sets. One contains all the paths that are completely contained within this region and the other set contains the rest of the paths.

the existence of the Ω region is that the paths in $\Delta(\Omega)$ and $\Delta'(\Omega)$ belong to different homotopy classes [20] and hence they should be uniquely identified. To better understand this see Fig. 6. The yellow path in 6a is of type CCC. This path cannot be transformed into the red path as we have $d(c_l(i), c_l(g)) < 4$ and $d(c_r(i), c_r(g)) < 4$. Therefore it is not possible to push the middle C-segment of the blue path through $C_L(i)$ and $C_L(g)$; in other words, the blue path and $C_L(i)$ and $C_L(g)$ form a bight.

The next subsection explains how one may uniquely identify all the possible conflagrations of a robot and its taut curvature-bounded cable.

IV. The Planning Algorithm

Let L_i be the line tangent to pose $P_i = (p_i, v_i)$, *i.e.,* it passes through p_i and is parallel to v_i. Also, let $C_L(i)$ and $C_R(i)$ be the circles tangent to L_i at point p_i on the left and on the right side of L_i, respectively, when looking in the direction v_i from p_i.

Algorithm 1 details how to find a Dubins path from P_i to P_g. Line 4 produces four tangents for each pair of circles, of which only one is compatible with direction of both v_i and v_g (see Fig 7). Line 5 will remove the tangents that are incompatible, yielding in only one Dubins path. In line 9, the length of each path is calculated and is then compared to the current minimum in line 10. Finding the length of a Dubins path is done by accumulating the length of the L-segment and $r_0 \times \theta$ for C-segment(s) (arcs), where θ is the angle traversed on C-segment in radians.

Recursive Algorithm 2 solves the problem of planning from an initial pose, P_i, to a goal pose, P_g, for a planar robot,

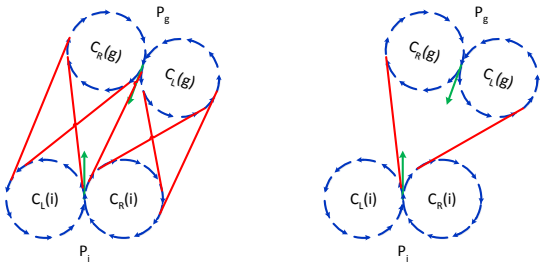
Algorithm 1: Find Shortest Curvature Constrained Path

1: **FindDubinsPath**(P_a, P_b)
2: Find $C_L(a)$, $C_R(a)$, $C_L(b)$, and $C_R(b)$
3: $minPath = \varnothing$
4: $t = \{$ all common tangents between $\big(C_i(a), C_j(b)\big); i, j \in \{L, R\}\}$
5: Remove *incompatible* tangents from t
6: **for all** $t_k \in t$ **do**
7: $\quad p_{a,k}$ = point at which t_k touches $C_i(a)$ for $i = R$ or L
8: $\quad p_{k,b}$ = point at which t_k touches $C_i(b)$ for $i = R$ or L
9: $\quad d = Length\big(arc(p_a, p_{a,k}) + (p_{a,k} - p_{k,b}) + arc(p_{k,b}, p_b)\big)$
10: $\quad$ **if** $d < Length(minPath)$ **then**
11: $\quad\quad minPath = arc(p_a, p_{a,k}) + (p_{a,k} - p_{k,b}) + arc(p_{k,b}, p_b)$
12: $\quad$ **end if**
13: **end for**
14: **return** $minPath$

whose pose is denoted P_r, situated in $\mathbb{R}^2$ in the presence of polygonal obstacles. Information regarding the cable's maximum length and its base pose is encoded within $cell$ passed as an input argument. The algorithm is initiated by passing the following arguments: current cell in which the robot is situated, P_i, P_g, and an empty path. The condition in Line 4, checks whether there exists a shortest curvature constrained path which is completely contained in $cell$. Finding the shortest curvature constrained path is achieved by Algorithm 1. The condition on line 4 ensures that the robot will not leave $cell$. If so, a *robot path* is created and is appended to the end of the path taken up to this point (see Section V-B). Otherwise the algorithm proceeds by searching down the tree in a depth-first fashion and storing the minimum length path that is found (or $\varnothing$ if no such path exists). Any graph search method can be used in place of the Depth-First Search. The next step is to look up the tree (ancestors) for a solution and compare the result of this search to the minimum length path found in child nodes. The root cell of the atlas does not have a parent (the value being set to $\varnothing$). Line 7 is a critical part of Algorithm 2, allowing the method to search for child cells and expand the tree only as needed. Algorithm 3 provides details.

A. A Note on a Tethered Robot with Extent

Throughout this work, we have considered the motion planning problem for a point robot. This eases understanding of the representation introduced in this work, but it is possible to plan motions for robots with extent using the



(a) All the possible tangents from initial circles to goal circles.

(b) Compatible tangents from initial circles to goal circles.

Fig. 7: The blue dotted circles show the boundaries of maximum curvature. The red lines show the tangents from initial circles to goal circles. The green arrows show the orientation of the initial and goal poses.

Algorithm 2: The Shortest Path Algorithm

1: **FindPath**$(cell, P_i, P_g, path)$
2: mark $cell$ as visited
3: $minPath = \varnothing$
4: **if** $cell.P_b \rightsquigarrow P_g$ is completely within $cell$ **then**
5: $\quad minPath = path.Append(cell.P_b \rightsquigarrow P_g)$
6: **else**
7: $\quad children =$ **GetChildCells**$(cell)$
8: $\quad$ **for all** $c \in children$ **do**
9: $\quad\quad \tau =$
$\quad\quad path.Append$(Find robot's path inside $cell$ to $c.StitchLine$)
10: $\quad\quad \tau =$ **FindPath**$(c, robot.P_r, P_g, temp)$
11: $\quad\quad$ **if** τ is shorter than $minPath$ **then**
12: $\quad\quad\quad minPath = \tau$
13: $\quad\quad$ **end if**
14: $\quad$ **end for**
15: $\quad$ **if** $Parent \neq \varnothing$ and is not yet visited **then**
16: $\quad\quad \tau = path.Append$(Find robot's path inside $cell$ to $cell.P_b$)
17: $\quad\quad \tau =$ **FindPath**$(Parent, cell.P_b, P_g, temp)$
18: $\quad\quad$ **if** τ is shorter than $minPath$ **then**
19: $\quad\quad\quad minPath = \tau$
20: $\quad\quad$ **end if**
21: $\quad$ **end if**
22: **end if**
23: **return** $minPath$

Algorithm 3: Algorithm for Getting Child Cells

1: **GetChildCells**$(cell)$
2: $children = \varnothing$
3: $C = \{C_L(cell.P_b), C_R(cell.P_b)\}$
4: keep *valid* circles of C
5: **for all** $c \in C$ **do**
6: $\quad$ **for all** $p_o \in$ Vertices such that p_o is inside $cell$ **do**
7: $\quad\quad L = \{$ tangents to c passing from $p_o\}$
8: $\quad\quad$ remove *incompatible* tangents from L
9: $\quad\quad$ **for all** $l \in L$ **do**
10: $\quad\quad\quad v_o =$ direction of l
11: $\quad\quad\quad P_o =$ Pose(p_o, v_o)
12: $\quad\quad\quad child =$ cell with $Parent = cell$ and $P_b = P_o$
13: $\quad\quad\quad children = children \cup \{child\}$
14: $\quad\quad$ **end for**
15: $\quad$ **end for**
16: **end for**
17: **return** $children$

representation too. The standard technique is to reduce the problem of a robot with extent to a point robot using Minkowski sums [21]. In the context of planning for a tethered robot, the map produced by such procedures cannot be applied to the cable as the cable always makes contact with the *actual* vertices of the obstacles (as opposed to the vertices in the reduced problem). To overcome this, one plans the path for the robot using the map of reduced problem, but the length of the cable and the boundaries of the cells should be calculated using the actual location of the vertices of the obstacles.

V. Discussion of the Method

To demonstrate the method and evaluate its performance, we developed a simulation environment and implemented the algorithm in MATLAB (see Figs. 8–17). In the following sections provide some observations.

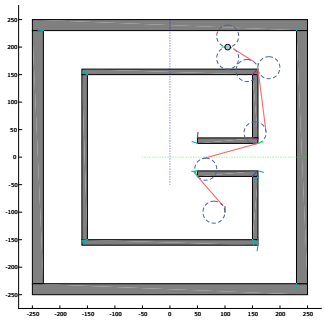


Fig. 8: An example a final configuration of the robot and its cable after execution of the shortest path plan. ($r_0 = 20\ cm, \kappa_0 = 5\ m^{-1}$, cable = 5 m).

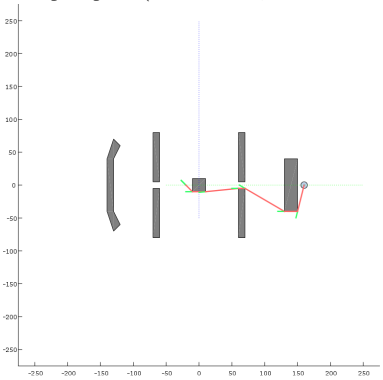


Fig. 9: The performance of the curvature parameter. As demonstrated in this figure and the four that follow, increasing the radius of curvature requires the robot to take longer paths to reach its destination. Note the comparatively sharp angle in the cable's configuration in this figure. ($r_0 \approxeq 0, \kappa_0 = \infty$, cable $= 5\ m$).

A. Memory Consumption

The method uses few graph vertices in memory during the search because it benefits from on-line exploration of the c-space. The atlas uses a data structure for all of the Dubins Cells stored in the graph and doing so requires only a constant amount of memory regardless of the cell's volume. These memory saving aspects of the approach is discussed in greater detail in our previous paper [8].

B. Discussion of the Algorithm

The main algorithm presented in this work is a graph search problem which expands the unexplored parts of the c-space only as needed. In this problem, the combinatorial nature of the homotopy classes induced by cable configurations can result in a space which is exponential in the number of vertices.

However, the basic function calls in Algorithm 2 are all polynomial in the number of vertices in the environment, n. Algorithm 1, for finding a Dubins Path is a constant time operation, resulting in $O(n)$ time in line 4. Finding a path for the robot in line 5 is also computed in $O(n)$ since all the boundaries of a Dubins Cell are known. The robot is assumed to be controllable in all three degrees of freedom, so its path inside a cell is a straight line when not obstructed by the boundary of maximal curvature or any obstacle. Otherwise, the path is a line segment tangent to the boundary followed by a segment supported by the boundary, followed

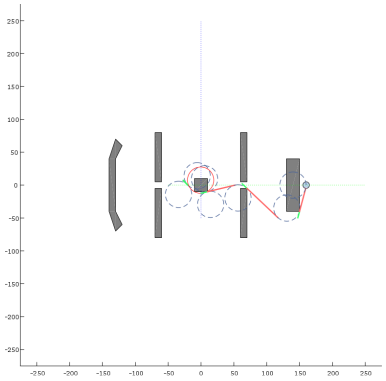


Fig. 10: Parameterized curvature continued. Note the last portion of the path is a CCC path. ($r_0 = 20\ cm, \kappa_0 = 5\ m^{-1}$, cable $= 5\ m$).

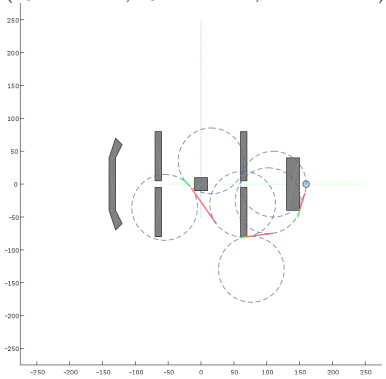


Fig. 11: Parameterized curvature cont. ($r_0 = 50\ cm, \kappa_0 = 2\ m^{-1}$, cable $= 5\ m$).

by another tangent from the boundary to the goal point. Finally, Algorithm 3 returns a set with $O(n)$ elements.

C. Parameterized Curvature Constraint

An advantage of this method is its versatility in modeling different cable specifications. To demonstrate this capability, we tested the algorithm with different curvature upper-bounds. Fig. 9 illustrates a cable without curvature constraint (*i.e.,* $r_0 = \lim x \to 0^+$). As a result, the cable is able to have a sharp angle bend around the corner of the obstacle on the path $P_b \rightsquigarrow P_g$. In contrast, Figs. 10–12 r_0 show a nonzero positive values for r_0 limiting the freedom with which the cable may bend. For example, the algorithm is unable to find any feasible solution the problem in Fig. 12. Note that increasing the cable length in Fig. 13 leads to a solution from the initial to goal pose.

VI. Conclusion

This paper addresses the problem of planning motions for tethered robots when the tether has a constraint on its curvature—an investigation motivated by the fact that in practice cables have limited flexibility. Since existing methods ignore this constraint, they may plan motions that pass through unreachable configurations. Even in environments with few obstacles, the topology of the configuration space for a robot with a curvature constrained tether is complex; though we know of no existing implementations, naïve approaches to this problem that are sound and complete require extremely costly computations.

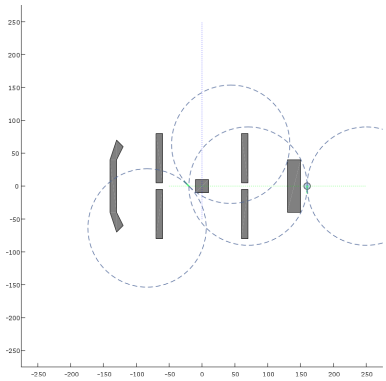


Fig. 12: Parameterized curvature cont. In this figure there is no solution to the problem because the cable has insufficient length. ($r_0 = 1\ m$, $\kappa_0 = 1\ m^{-1}$, cable $= 5\ m$).

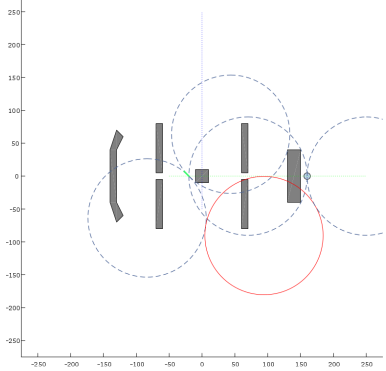


Fig. 13: Parameterized curvature cont. In this instance we have tested the problem in Fig. 12 with a longer cable and the solution is a CCC path.($r_0 = 1\ m, \kappa_0 = 1$, cable $= 10\ m$).

In tackling this problem, we found it useful to combine ideas from two distinct areas of motion planning research, *planning for tethered mobile robots* and *nonholonomic motion planning*. We generalized the decomposition method proposed in our earlier work [8], to separate topological regularities from the continuous aspects of the c-space by representing it with an atlas and its accompanying graph. We also employ Dubins's theory [9] to exclude infeasible cable configurations from the robot-cable c-space. This paper presents an algorithm to explore the c-space lazily while searching it. Our method benefits from the on-line exploration to minimize the memory requirements. Finally, we note that the use of the parameterized curvature constraint, makes this method suitable for solution of several other related problems.

Acknowledgment

The authors are grateful for the support provided by the National Science Foundation grants IIS-1302393 and IIS-1453652 for this work.

References

[1] B. Donald, L. Gariepy, and D. Rus, "Distributed manipulation of multiple objects using ropes," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1. IEEE, 2000, pp. 450–457.

[2] S. Bhattacharya, H. Heidarsson, G. Sukhatme, and V. Kumar, "Cooperative control of autonomous surface vehicles for oil skimming and cleanup," in *Robotics and automation (ICRA), 2011 IEEE international conference on*. IEEE, 2011, pp. 2374–2379.

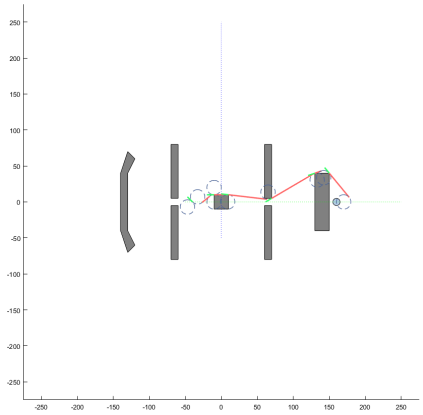


Fig. 14: Untangling the cable. Figures 14–17 show how the robot untangles its cable to execute the shortest path. ($r_0 = 10\ cm, \kappa_0 = 10\ m^{-1}$, cable $= 5\ m$).

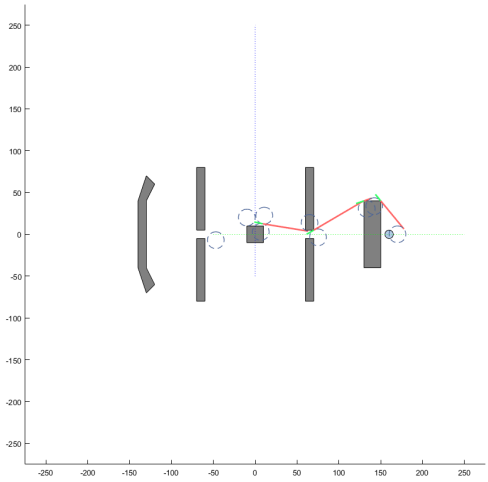


Fig. 15: Untangling the cable cont. ($r_0 = 10\ cm, \kappa_0 = 10\ m^{-1}$, cable $= 5\ m$).

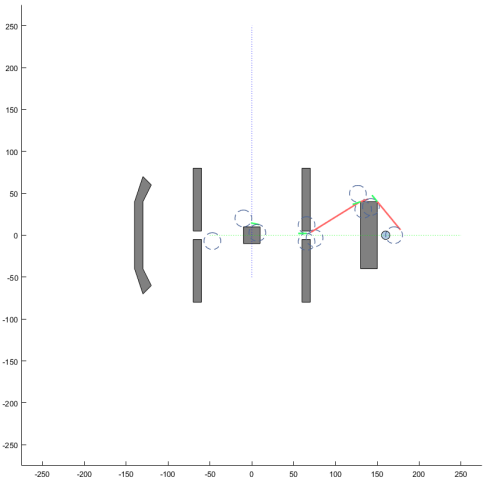


Fig. 16: Untangling the cable cont. ($r_0 = 10\ cm, \kappa_0 = 10\ m^{-1}$, cable $= 5\ m$).

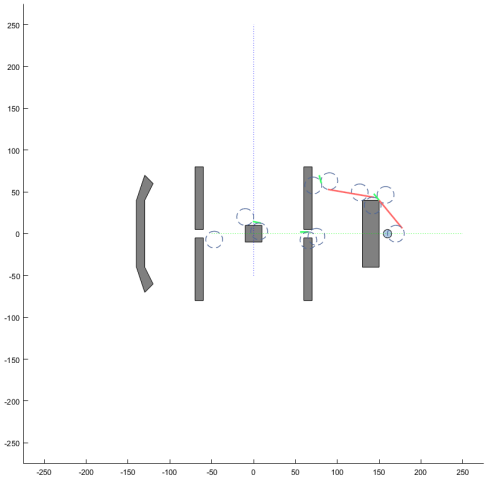


Fig. 17: Untangling the cable cont. ($r_0 = 10\ cm, \kappa_0 = 10\ m^{-1}$, cable $= 5\ m$).

[3] S. Kim, S. Bhattacharya, H. K. Heidarsson, G. Sukhatme, and V. Kumar, "A topological approach to using cables to separate and manipulate sets of objects." in *Robotics: Science and Systems*, 2013.

[4] S. Bhattacharya, V. Kumar, and M. Likhachev, "Search-based path planning with homotopy class constraints," in *Annual Symposium on Combinatorial Search*, 2010.

[5] T. Igarashi and M. Stilman, "Homotopic path planning on manifolds for cabled mobile robots," in *Algorithmic Foundations of Robotics IX*. Springer, 2011, pp. 1–18.

[6] D. S. Yershov, P. Vernaza, and S. M. LaValle, "Continuous planning with winding constraints using optimal heuristic-driven front propagation," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 5551–5556.

[7] I. Shnaps and E. Rimon, "Online coverage by a tethered autonomous mobile robot in planar unknown environments," in *Proceedings of Robotics: Science and Systems*, Berlin, Germany, June 2013.

[8] R. H. Teshnizi and D. A. Shell, "Computing cell-based decompositions dynamically for planning motions of tethered robots," in *Robotics and automation (ICRA), 2014 IEEE international conference on*, 2014.

[9] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of mathematics*, pp. 497–516, 1957.

[10] P. G. Xavier, "Shortest path planning for a tethered robot or an anchored cable," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 2. IEEE, 1999, pp. 1011–1017.

[11] D. Grigoriev and A. Slissenko, "Polytime algorithm for the shortest path in a homotopy class amidst semi-algebraic obstacles in the plane," in *Proceedings of the 1998 international symposium on Symbolic and algebraic computation*. ACM, 1998, pp. 17–24.

[12] V. Narayanan, P. Vernaza, M. Likhachev, and S. M. LaValle, "Planning under topological constraints using beam-graphs," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 431–437.

[13] P. Jacobs and J. Canny, "Planning smooth paths for mobile robots," in *Nonholonomic Motion Planning*. Springer, 1993, pp. 271–342.

[14] P. K. Agarwal, P. Raghavan, and H. Tamaki, "Motion planning for a steering-constrained robot through moderate obstacles," in *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*. ACM, 1995, pp. 343–352.

[15] P. K. Agarwal, T. Biedl, S. Lazard, S. Robbins, S. Suri, and S. Whitesides, "Curvature-constrained shortest paths in a convex polygon," *SIAM Journal on Computing*, vol. 31, no. 6, pp. 1814–1851, 2002.

[16] F. Lamiraux and J.-P. Lammond, "Smooth motion planning for car-like vehicles," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 4, pp. 498–501, 2001.

[17] R. Takei, R. Tsai, H. Shen, and Y. Landa, "A practical path-planning algorithm for a simple car: a hamilton-jacobi approach," in *American Control Conference (ACC), 2010*. IEEE, 2010, pp. 6175–6180.

[18] H. Goldstein, *Classical Mechanics*, 2nd ed. Addison-Weseley, 1980.

[19] S. Fortune and G. Wilfong, "Planning constrained motion," *Annals of Mathematics and Artificial Intelligence*, vol. 3, no. 1, pp. 21–82, 1991.

[20] J. Ayala and H. Rubinstein, "Non-uniqueness of the homotopy class of bounded curvature paths," *arXiv preprint arXiv:1403.4911*, 2014.

[21] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry, Algorithms and Applications*. Springer, 1991.