Path Planning for a Tethered Mobile Robot

Soonkyum Kim[1] Subhrajit Bhattacharya[2] Vijay Kumar[3]

***Abstract*— In this paper we investigate the problem of navigation for a planar mobile robot tethered to a base by a flexible cable of length L. Obstacles present in the environment, coupled with the cable length constraint, makes the problem highly non-trivial. We adopt a topological approach along with graph search-based techniques to solve this problem, wherein we use the notion of a *homotopy augmented graph* to capture the information about the homotopy class of the cable. This lets us plan traversable optimal trajectories from an initial robot position and cable configuration to a final position of the robot. We demonstrate the algorithm by planning trajectories in several cluttered environments and with different cable lengths. As a demonstration of practical applicability, using a dynamic simulation testbed we simulate a robot-cable system following a planned trajectory.**

I. Introduction

There are important applications in which mobile robots must be tethered to a base station. A tether can provide a communication link enabling teleoperation in environments where wireless signals either may not work or be sufficient [22], [21]. This was the case when robots were deployed in the reactor building after the tragic accident in the Fukushima because the radioactive environment disrupted communication links [13]. In disaster recovery operations a tether may also facilitate high bandwidth communications between the robot and a remotely located human user [6]. As in many mobile robot applications, energy can be a scarce resource and a tether allows a robot to be connected to a power source allowing for longer missions.

In such applications, the presence of a tether introduces geometric constraints that are both metric and topological in flavor. First, because the tether generally has a finite length, the reachable workspace of the robot is limited. Second, in the presence of obstacles, the configuration space of a tethered robot is naturally partitioned into equivalence classes, each consisting of cable configurations such that one can be reached from the other without passing through an obstacle.

In this work, we consider the path planning of robot whose workspace is limited by a cable. The robot moves in a known, bounded workspace, W (a subset of $\mathbb{R}^2$, which is of interest). One end of a cable, whose length is L, is attached on the robot while the other end is anchored on a fixed point, *base*, at $\mathbf{q}_b$. In the absence of any obstacles, the reachable space of robot will be the intersection of W and a disk of radius L centered at the base, $\mathbf{q}_b$. However, the presence of obstacles introduces geometric constraints as shown in Figure 1. In addition, obstacles introduce topological constraints. As shown in Figure 2(a), the point $\mathbf{q}_g$ can only be reached if the cable configuration lies in the appropriate homotopy class

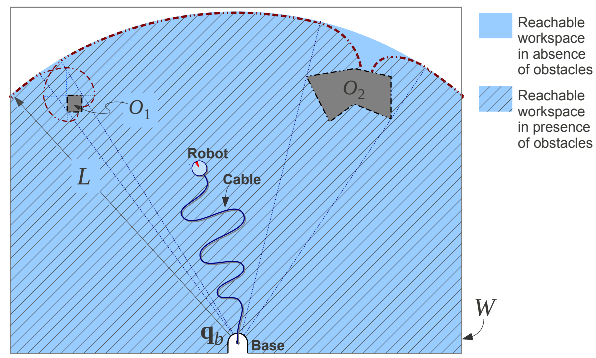


Fig. 1. The reachable workspace for a robot tethered by a cable of length L in a workspace W. The blue filled region is the reachable workspace in absence of any obstacle. However, in presence of obstacles, O_1 and O_2, the reachable workspace shrinks to the hatched region (with workspace boundary traced by the robot with *taut* cable shown in dash-dots).

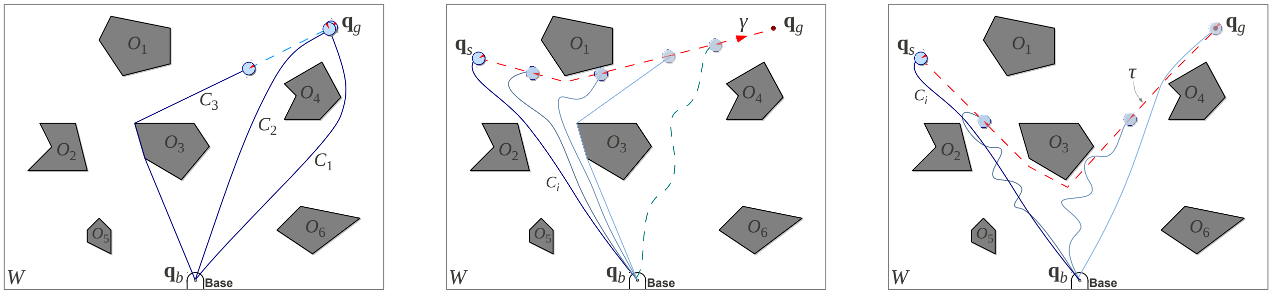
(the homotopy classes of C_1 or C_2, for example, but not the homotopy class of C_3).

Our objective is to be able to find the shortest feasible trajectory to any specified goal position, $\mathbf{q}_g \in (W - \mathcal{O})$ (where $\mathcal{O} = \cup_{i=1}^{m} O_i$, the set of obstacles), if such a trajectory exists (that is, if the goal lies in the reachable workspace, via any homotopy class), given an initial configuration of the robot and the cable. Clearly the shortest trajectory in $(W - \mathcal{O})$ connecting $\mathbf{q}_s$ to $\mathbf{q}_g$ may not be sufficient (as illustrated in Figure 2(b)). This is because the cable length constraint may prevent the traversal of the trajectory. Even if every point on the trajectory is individually reachable (using some homotopy class), the trajectory as a whole may not be traversable. However, the shortest trajectory in some other homotopy class may be traversable (as illustrated in Figure 2(c)).

The cable can be considered to be of fixed length, L, that is flexible and can be allowed to slack. We will assume that the robot can drive over the cable, if required, and hence we will not be concerned by the issues like damage to or self-entanglement of the cable. Alternatively, we can assume that the cable is a flexible and stretchable elastic band, with maximum length of L, which remains taut all the time. Either of these models are compatible with the proposed solution.

There are several relevant papers that report similar investigations. Among the earliest works, [17] have considered tangle-free planning for multiple tethered robots in an environment without obstacles.

In [27] the author approaches the problem by triangulation of an environment with polygonal obstacles, followed by a visibility graph-like construction, selection of a particular homotopy class, and performing an almost-exhaustive enumeration of graph paths in the class. However this approach



(a) The point $\mathbf{q}_g$ can be reached via some homotopy classes, but not others, because of the cable length constraint.

(b) The globally shortest trajectory, γ, from $\mathbf{q}_s$ to $\mathbf{q}_g$, with initial cable configuration C_i is not traversable because of cable length constraint, even though every point on γ is reachable via some homotopy class.

(c) The shortest trajectory in a different homotopy class (the trajectory τ), with initial cable configuration C_i, is however traversable.

Fig. 2. A tethered robot in a cluttered environment, $(W - \mathcal{O})$, where $\mathcal{O} = O_1 \cup O_2 \cup \cdots$.

sufferes from the computational complexities involved and the limitations surrounding construction of a visibility graph. This technique has also appeared in [28].

In [18] the authors have solved the problem using a notion of homotopy classes of the cables, not very unlike our present approach. However, in the graph construction presented there, instead of formally identifying homotopy classes using a *homotopy invariant*, the distance from the initial vertex is used. Such a representation of homotopy, solely based on a metric, under many situation may fail to identify that two vertices in the graph represent two different homotopy classes. For example, when a point and the start point are on an axis of symmetry of a symmetric obstacle, the shortest paths in the two homotopy classes connecting the points will be of equal length.

The novelty of our approach, on the other hand, is that we actually use a true *homotopy invariant* (h-signature) to construct what we call a h-augmented graph that explicitly bears the topological information ([5], [19] and Section III-A). This not only makes our approach more robust, it also makes the graph construction a lot simpler and intuitive.

In context of coverage using a tethered robot, a Morse theory-like approach has been used in [24], where the authors have identified the critical points ("split cells" and "saddle lines") of the Euclidean distance from the base, and hence guaranteed coverage of the workspace.

In our own previous work we have investigated cooperative control of two autonomous boats towing a boom or a cable with the motivation to address oil skimming and environmental clean up [4]. Most recently, we have also addressed cooperative manipulation of objects in which tools from algebraic topology were used to find robot trajectories for separating and manipulating objects [19].

II. THEORETICAL FOUNDATION

A. Homotopy Class Of Paths And Cables

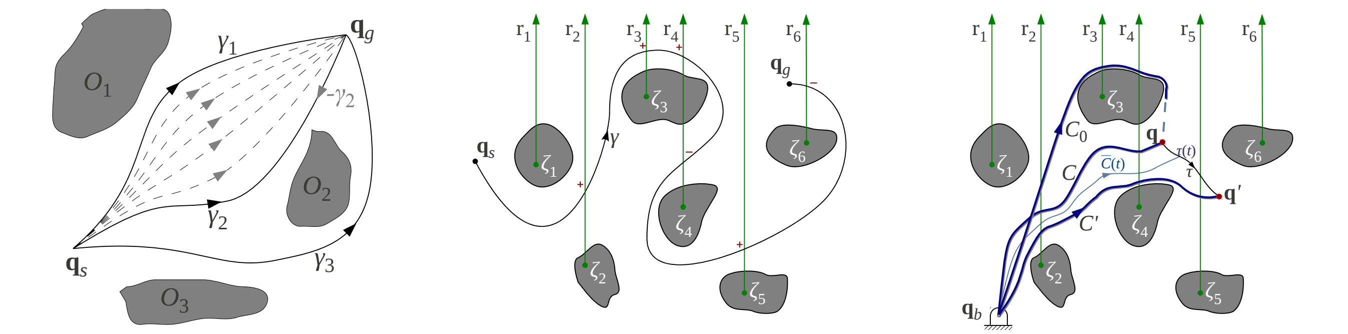
Let W be a 2-dimensional simply connected and bounded region. Suppose it contains a set of obstacles, $\mathcal{O} = O_1 \cup Q_2 \cup \cdots \cup O_m \subseteq W$, where $O_1, O_2, \cdots, O_m$ are m counts of obstacles.

Both cable configuration and robot paths are 1-dimensional curves in $(W - \mathcal{O})$. They can thus be defined as continuous maps from the interval $[0,1]$ to $(W - \mathcal{O})$. We will need to consider the homotopy class of both cables and robot trajectories. So we start by reviewing some of the standard definitions related to homotopy.

Definition 1 (Homotopy classes of curves): Two curves $\gamma_1, \gamma_2 : [0,1] \rightarrow (W - \mathcal{O})$ connecting the same start and end points, are homotopic (or belong to the same *homotopy class*) iff one can be continuously deformed into the other without intersecting any obstacle (see Figure 3(a)).

Formally, if $\gamma_1 : [0,1] \rightarrow (W - \mathcal{O})$ and $\gamma_2 : [0,1] \rightarrow (W - \mathcal{O})$ represent the two trajectories (with $\gamma_1(0) = \gamma_2(0) = \mathbf{q}_s$ and $\gamma_1(1) = \gamma_2(1) = \mathbf{q}_g$), then γ_1 is homotopic to γ_2 iff there exists a continuous map $\eta : [0,1] \times [0,1] \rightarrow (W - \mathcal{O})$ such that $\eta(\alpha, 0) = \gamma_1(\alpha)\ \forall \alpha \in [0,1]$, $\eta(\beta, 1) = \gamma_2(\beta)\ \forall \beta \in [0,1]$, and $\eta(0, \mu) = \mathbf{q}_s, \eta(1, \mu) = \mathbf{q}_g\ \forall \mu \in [0,1]$ [5], [15].

Homotopy invariants of curves (a function of curves that uniquely identifies its input's homotopy class), in general, are difficult to design and compute. *Homotopy groups* do not have the natural structure of a vector space, unlike its more computationally favorable cousin, *homology groups* [15]. However, for curves in 2-dimensional plane with punctures (*i.e.* obstacles), there is a relatively simple representation of the homotopy group and a way of computing the homotopy class of a given curve [12], [16], [26], [15], [3]: We consider *representative points*, ζ_i, inside the i^{th} obstacle O_i [5], and *parallel non-intersecting* rays, $r_1, r_2, \cdots, r_m$, emanating from the obstalces (Figure 3(b)). If γ is a given curve whose homotopy class we are trying to identify, we construct a *word* by tracing γ, and consecutively placing the letters of the rays that it crosses, with a superscript of '$+1$' (assumed implicitly) if the crossing is from left to right, and '-1' if the crossing is from right to left. Thus, for example, the word for γ in Figure 3(b) will be "$r_2 r_3 r_4 r_4^{-1} r_5 r_6^{-1}$". We then *reduce* this word by canceling the same letters that appear consecutively but with opposite superscript signs. Thus, the word for γ in Figure 3(b) can be reduced to "$r_2 r_3 r_5 r_6^{-1}$". This reduced word representation is a *homotopy invariant* for open curves (with fixed end points), γ, and we will write this as $h(\gamma)$ and call it the "h-signature of γ". However, it is important to note that we cannot exchange position for arbitrary pairs of letters in the word (*i.e.* the juxtaposition of letters is *non-commutative*). h-signature is not a vector, but an



(a) γ_1 is homotopic to γ_2 since there is a continuous sequence of trajectories representing deformation of one into the other, but not to γ_3 since it cannot be continuously deformed into any of the other two.

(b) ζ_i are representative points inside the obstacles, $O_1, O_2, \cdots, O_m$ (in that order), and r_i, $i = 1, \cdots, m$ are rays emanating from the respective points. The homotopy invariant of this curve γ is $h(\gamma) =$ "$r_2 r_3 r_5 r_6^{-1}$".

(c) Point $\mathbf{q}$ is *reachable* in homotopy class "$r_2 r_4$", but not in the class "$r_2 r_3 r_4$".

Fig. 3. Homotopy classes and their word representation.

element of the *non-abelian* group *freely generated* [23], [15] by $\{r_1, r_2, \cdots, r_m\}$. Thus, although *words* can't be added in the sense of vectors, they can be concatenated under the non-commutative *group operation*, '$\diamond$'. Also, the inverse of a word, $\mathfrak{w}$, written as $\mathfrak{w}^{-1}$, is the h-signature of the same curve but with opposite orientation (*i.e.* $h(-\gamma) = (h(\gamma))^{-1}$), and is a word where the order of the letters are reversed, and the exponent of each letter is flipped (so that $\mathfrak{w} \diamond \mathfrak{w}^{-1} =$" ", the identity element). Thus, $(\mathfrak{w}_1 \diamond \mathfrak{w}_2)^{-1} = {\mathfrak{w}_2}^{-1} \diamond {\mathfrak{w}_1}^{-1}$. As an example, ("$r_2 r_3 r_5 r_6^{-1}$")$^{-1}$ ="$r_6 r_5^{-1} r_3^{-1} r_2^{-1}$". Furthermore, if the end point of γ coincides with the start point of γ', then $h(\gamma \cup \gamma') = h(\gamma) \diamond h(\gamma')$.

The *homotopy invariant* of a curve, γ, is the *reduced word* constructed in the described way. We call this the h-signature of γ, and write it as $h(\gamma)$. It uniquely identifies the homotopy class of a curve. That is, the reduced word for two curves connecting the same points are same if and only if the curves are homotopic.

B. Feasible Path to Goal

Definition 2: (Reachability of point via a particular homotopy class): Given a point in the workspace, $\mathbf{q} \in (W - \mathcal{O})$, and a homotopy class of curves connecting $\mathbf{q}_b$ to $\mathbf{q}$ (represented by the corresponding *word*, $\mathfrak{w}$), we say $\mathbf{q}$ is *reachable via homotopy class* $\mathfrak{w}$ if the Euclidean length of the shortest path connecting $\mathbf{q}_b$ and $\mathbf{q}$, but restricted to the homotopy class corresponding to word $\mathfrak{w}$, is less than or equal to L (see Figure 3(c)).

Proposition 1: Let $\mathbf{q}$ be a robot position and C be a corresponding cable configuration connecting $\mathbf{q}_b$ to $\mathbf{q}$. C is given an orientation from base to robot, as shown in Figure 3(c). Let τ be a trajectory connecting $\mathbf{q}$ to another point $\mathbf{q}' \in (W - \mathcal{O})$ (with that orientation). If $\tau(t)$ is reachable via homotopy class $h(C) \diamond h(\tau([0,t]))$, for every $t \in [0,1]$ (*i.e.*, for all points on the trajectory), then the robot can traverse τ in its entirety to reach $\mathbf{q}'$. Furthermore, the word corresponding to the homotopy class of the final cable configuration (*i.e.*, its h-signature) will be $h(C') = h(C) \diamond h(\tau)$.
[Note: By $\tau([0,t])$ we mean the part of the trajectory, τ, up to the point $\tau(t)$. That is, $\tau([0,t]) = \cup_{u \in [0,t]} \tau(u)$.]

Proof: We need to prove that each point, $\tau(t)$, that is reached as the robot traverses the trajectory, is reachable via the homotopy class corresponding to the cable configuration at that t.

Say the cable configuration at the instant t (*i.e.*, when the robot is at $\tau(t)$) be $\overline{C}(t)$. It is easy to notice that then C, $\tau([0,t])$ and $-\overline{C}(t)$ will form a closed region (Figure 3(c)) that has been swept by the cable itself. Thus $C \cup \tau([0,t]) \cup (-\overline{C}(t))$ is null-homotopic [15] – that is, it is a closed loop that can be contracted to a point. Thus, $h(C \cup \tau([0,t]) \cup (-\overline{C}(t))) =$ " " (the empty word). This implies $h(\overline{C}(t)) = h(C \cup \tau([0,t])) = h(C) \diamond h(\tau([0,t]))$.

Thus, due to the hypothesis in the statement, every point on the trajectory, $\tau(t)$, is reachable via the homotopy class that the cable will be in as the robot follows the trajectory. Thus the trajectory will be traversable.

The last statement is obvious since $C' \equiv \overline{C}(1)$. ■

III. OPTIMAL PATH GENERATION

A. Homotopy Augmented Graph

We use a discrete representation of the workspace, W, and construct a graph, G (with vertex set $\mathcal{V}(G)$ and edge set $\mathcal{E}(G)$), by placing a vertex in every accessible discrete cell (cells not intersecting with an obstacle) and by establishing an edge between the vertices of adjacent cells. While the graph, G, itself can be quite arbitrary, for simplicity we used a uniform square discretization and an 8-connected graph representation of the environment for all our simulations (Figure 4). From such a graph we construct an h-augmented graph, G_h, for keeping track of the homotopy class of the cable. This construction, in essence, is very similar to that of the *homology augmented graph* (or H-augmented graph) as was described in [5]. However here, instead of the *homology invariant*, we use the described *homotopy invariant*. Similar construction also appears in our recent work [19].

A vertex in this h-augmented graph is of the form $(\mathrm{q}, \mathfrak{w})$ where $\mathrm{q} \in \mathcal{V}(G)$ is the position of the robot in the workspace (as a vertex in the discrete representation graph, $\mathcal{G}$) and $\mathfrak{w}$ is the *word* (*i.e.* the homotopy invariant) corresponding to the homotopy class of the cable. We write the tuple as $\mathrm{v} = (\mathrm{q}, \mathfrak{w}) \in \mathcal{V}(G_h)$. We assume that there is a vertex

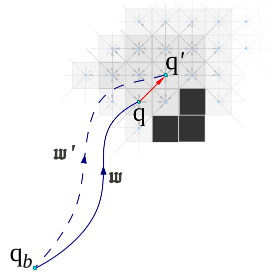


Fig. 4. The h-augmented graph, G_h, is created from the discrete graph representation of the environment, G, so as to incorporate information about the homotopy class of the cable.

in G corresponding to the base, $\mathrm{q}_b \in \mathcal{V}(G)$. This point is reachable by the robot with the cable not wrapping around any obstacle (the cable curled up or spooled completely, for example). In this configuration, the h-signature of the cable is the empty word, " ". Thus in the h-augmented, the corresponding vertex is $\mathrm{v}_b := (\mathrm{q}_b, \text{`` ''})$.

Suppose $\mathrm{q} \in \mathcal{V}(G)$ and $\mathrm{q}' \in \mathcal{N}_G(\mathrm{q}) \subset \mathcal{V}(G)$ (where by $\mathcal{N}_G(\mathrm{q})$ we mean the neighbors of q in G). Thus, $[\mathrm{q} \rightsquigarrow \mathrm{q}'] \in \mathcal{E}(G)$ (where by $[\mathrm{q} \rightsquigarrow \mathrm{q}']$ we mean a directed edge from q to q′). Then, if $\mathrm{v} = (\mathrm{q}, \mathfrak{w}) \in \mathcal{V}(G_h)$ is a vertex in the h-augmented graph, and if the robot travels from q to q′ via the line segment $\overrightarrow{\mathrm{qq}'}$ (line segment corresponding to the edge), the h-signature of the cable will be $\mathfrak{w} \diamond h(\overrightarrow{\mathrm{qq}'})$ when the robot reaches q′ (due to the last statement of Proposition 1). This means that the neighbors of $(\mathrm{q}, \mathfrak{w})$ in G_h will be of the form $(\mathrm{q}', \ \mathfrak{w} \diamond h(\overrightarrow{\mathrm{qq}'})), \ \ \forall \ \mathrm{q}' \in \mathcal{N}_G(\mathrm{q})$ (refer to Figure 4).

Thus, starting from $\mathrm{v}_b = (\mathrm{q}_b, \text{`` ''})$, one can generate the vertices in G_h. Corresponding to every edge $[\mathrm{q} \rightsquigarrow \mathrm{q}'] \in \mathcal{E}(G)$, there are edges of the form $[(\mathrm{q}, \mathfrak{w}) \rightsquigarrow (\mathrm{q}', \mathfrak{w} \diamond h(\overrightarrow{\mathrm{qq}'})] \in \mathcal{E}(G_h)$. We take the cost of an edge in G_h to be the same as the cost of the projected edge in G. That is, $c_{G_h}([(\mathrm{q}, \mathfrak{w}) \rightsquigarrow (\mathrm{q}', \mathfrak{w} \diamond h(\overrightarrow{\mathrm{qq}'})]) = c_G([\mathrm{q} \rightsquigarrow \mathrm{q}'])$ (where c_G and c_{G_h} represent the cost functions in the respective graphs). In our implementation we choose $c_G([\mathrm{q} \rightsquigarrow \mathrm{q}'])$ to be the Euclidean length of the line segment that constitutes the edge, $\overline{\mathrm{qq}'}$

In the above construction, for a $(\mathrm{q}, \mathfrak{w}) \in \mathcal{V}(G_h)$, we do not however take into account the reachability of q via homotopy class $\mathfrak{w}$. Thus in the next section we describe the first step in implementation, that is to determine the reachable workspace via the different homotopy classes.

B. Algorithm: Step 1 — Determining the Reachable Workspace (the Pre-processing Step)

This step is required for determining the reachable vertices in G_h. This is a one-time pre-processing step — that means this step needs to be executed only once for a given environment. Following that, every time we need to plan an optimal path from any arbitrary robot-cable configuration to a goal position of the robot, we need not execute this step, and will only need to run *Step* 2 of the algorithm.

Starting from the vertex $\mathrm{v}_b = (\mathrm{q}_b, \text{`` ''}) \in \mathcal{V}(G_h)$, we use Dijkstra's [10], [8] algorithm to *expand* the vertices in the graph. For a $\mathrm{v} = (\mathrm{q}, \mathfrak{w}) \in \mathcal{V}(G_h)$, this simultaneously computes the cost of the shortest path connecting the vertex and v_b. Let us represent the cost of this shortest path in the graph by $g_{G_h}(\mathrm{v})$ (typically called the g-score of the vertex).

It is to be noted that $g_{G_h}(\mathrm{v})$ is the cost of a path in the discrete graph representation of the environment, and in general is greater than the actual Euclidean length of the shortest curve with h-signature $\mathfrak{w}$ connecting $\mathbf{q}_b$ and $\mathbf{q}$ (where $\mathbf{q}_*$ is the position of the vertex q_* in $W - \mathcal{O}$). While one can simply eliminate vertices with g-score greater than L, that approach will get rid of some vertices that are actually accessible when the maximum cable length is L (because Euclidean length of the shortest curve up to the vertex in the particular homotopy class is less than L, but the shortest path in the graph leading to the vertex is greater than L).

Thus, we try to do better by reducing the elimination of physically accessible vertices: While executing the Dijkstra's algorithm, if for an expanded vertex $\mathrm{v} = (\mathrm{q}, \mathfrak{w})$ we have $g_{G_h}(\mathrm{v}) > L$ (*i.e.* it is potentially inaccessible), instead of marking it as inaccessible right away, we take the shortest path in G_h leading up to v (call it P), and use a curve shortening algorithm (Algorithm '*CurveShorten*') to compute the Euclidean length of the shortest curve with h-signature $\mathfrak{w}$ leading up to $\mathbf{q}$. *CurveShorten(P)* gives a value that is closer (though not equal) to the Euclidean length of the shortest curve in $W - \mathcal{O}$ connecting $\mathbf{q}_b$ to $\mathbf{q}$, and in the homotopy class $\mathfrak{w}$.

However, since the *CurveShorten* algorithm is expensive to run, we do not run it for every vertex, v, with $g_{G_h}(\mathrm{v}) > L$. Instead, we run it for such a vertex only when the g-score of its parent (call it v_p) is less than or equal to L as well. We represent this *shortened distance* of v (from v_b via the correct homotopy class, $\mathfrak{w}$) as $d_{G_h}(\mathrm{v})$ and store it as a data associated with the vertex v. Otherwise, if the g-score of the parent, v_p, of a vertex, v, is greater than L (which implies, using an inductive argument, that $d_{G_h}(\mathrm{v}_p)$ has been computed) we designate its 'shortened distance' as $d_{G_h}(\mathrm{v}) = d_{G_h}(\mathrm{v}_p) + c_{G_h}([\mathrm{v}_p \rightsquigarrow \mathrm{v}'])$. This effectively performs the curve shortening only for points on a sphere of (approximate) radius L in G_h (*i.e.*, distance L in metric restricted to the graph), while for points outside the ball the curve shortening happens *partially*. Thus, only if for a vertex $g_{G_h}(\mathrm{v}) \geq d_{G_h}(\mathrm{v}) > L$, we mark v to be inaccessible (and hence do not generate its children).

We next describe the *CurveShorten* algorithm briefly.

Algorithm *CurveShorten*: l = *CurveShorten(P)*:

Input: A path (a sequence of $n+1$ vertices) in graph G_h,
$P = [\mathrm{v}_0{=}\mathrm{v}_b, \mathrm{v}_1, \mathrm{v}_2, \cdots, \mathrm{v}_{n-1}, \mathrm{v}_n{=}\mathrm{v}], \ \ \mathrm{v}_i = (\mathrm{q}_i, \mathfrak{w}_i) \in \mathcal{V}(G_h)$

Output: The length of the "shortened" curve, l.

```
l := 0
i := 0,  j := 0
while  j ≤ n
   if  j < n  AND  q_i q_{j+1} ∩ O = ∅
      j := j + 1
   else
      l := l + ||q_i q_j||
      i := j
   end if
end while
return  l
```

By $\overline{\mathrm{q}_i\mathrm{q}_{j+1}}$ we mean the line segment (in the workspace W) joining the i^{th} and $(j+1)^{th}$ vertices in the path. $\|\overline{\mathrm{q}_i\mathrm{q}_j}\|$

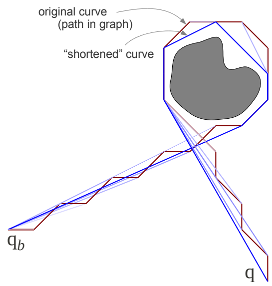


Fig. 5. Illustration of algorithm *CurveShorten*.

indicates the Euclidean length of the line segment $\overline{\mathrm{q}_i \mathrm{q}_j}$. Starting from the base, the *CurveShorten* algorithm tries to sequentially join the vertices (that constitute the path P) using straight line segments until the line segment intersects an obstacle (which is checked by sampling points on the line segment and checking if any lies inside obstacles). Upon intersection with an obstacle it continues to try joining the last point with the following points (Figure 5). The *shortened* curve that this algorithm thus uses is a valid curve joining $\mathbf{q}_b$ and $\mathbf{q}$ and is in the same homotopy class as that of the original path provided to the algorithm (since we prevent intersection, and hence 'crossing across' an obstacle). The algorithm however is not guaranteed to return the lowest Euclidean length path in the homotopy class of the given path. However, the resulting curve has a smaller length that is closer to the true Euclidean length of the shortest path. A more extensive curve shortening algorithm may be employed (see [7] for a comprehensive list of curve shortening algorithms), but at a higher computational expense.

We also do not expand vertices that correspond to *looping configurations* — vertices, $\mathrm{v} = (\mathrm{q}, \mathfrak{w})$, such that the *homology signature* (computed using the *Hurewicz map*) corresponding to $\mathfrak{w}$ has elements with absolute value greater than 1, or such that the projection of the shortest curve leading to v on to G is self-interesting (see [19], [5] for a detailed discussion).

We continue expanding vertices in G_h using Dijkstras, and pruning vertices in the graph as described, until the *open set* is empty (this will be encountered since only finitely many vertices in G_h are reachable due to the length constraint). Thus, at the end of *Step 1* we have a subgraph of G_h that contains vertices $(\mathrm{q}, \mathfrak{w})$ such that $\mathbf{q}$ is reachable via the homotopy class $\mathfrak{w}$ (where, *reachability* is determined by the condition $d_{G_h}((\mathrm{q}, \mathfrak{w})) \leq L$). We write this subgraph as $\widehat{G}_h$.

C. Algorithm: Step 2 — Finding Optimal Path in $\widehat{G}_h$

Given an initial robot position, $\mathrm{q}_s \in \mathcal{V}(G)$ (as a vertex in G), and a cable configuration, C_s, the vertex in $\widehat{G}_h$ that this state corresponds to is $\mathrm{u}_s = (\mathrm{q}_s, h(C_s))$. This vertex will exist in $\mathcal{V}(\widehat{G}_h)$ since $\mathbf{q}_s$ is reachable via the homotopy class $h(C_s) =: \mathfrak{h}_s$.

Say $\mathrm{q}_g \in \mathcal{V}(G)$ is the goal vertex that the robot wants to reach ($\mathbf{q}_g$ being the location of the vertex in W). Thus, we perform an A* [14], [8] search in $\widehat{G}_h$ starting from vertex $(\mathrm{q}_s, \mathfrak{h}_s)$ (*i.e.*, seeding the *open set* with this vertex) until a vertex of the form $(\mathrm{q}_g, \mathfrak{h}_g) =: \mathrm{u}_g$ is expanded (for some word $\mathfrak{h}_g$). Following that one can reconstruct the path in $\widehat{G}_h$ connecting $(\mathrm{q}_s, \mathfrak{h}_s)$ with $(\mathrm{q}_g, \mathfrak{h}_g)$. Let's call this path $Q = [\mathrm{u}_s{=}\mathrm{u}_0, \mathrm{u}_1, \mathrm{u}_2, \cdots, \mathrm{u}_{m-1}, \mathrm{u}_m{=}\mathrm{u}_g]$, $\mathrm{u}_i = (\mathrm{q}_i, \mathfrak{h}_i) \in \mathcal{V}(\widehat{G}_h)$. Now we make couple of observations about this path:

i. Every vertex on this path is reachable via some homotopy class due to the construction of $\widehat{G}_h$ (which we constructed by eliminating non-reachable vertices of G_h).
ii. Due to the construction of G_h (and since $\widehat{G}_h$ is just a sub-graph of G_h), Q being a path in the graph will imply that for any two consecutive vertices in it, say u_j and u_{j+1}, we have $\mathfrak{h}_{j+1} = \mathfrak{h}_j \diamond h(\overrightarrow{\mathrm{q}_j \mathrm{q}_{j+1}})$. Using an induction over j this implies $\mathfrak{h}_{j+1} = \mathfrak{h}_s \diamond h(\overrightarrow{\mathrm{q}_s \mathrm{q}_1} \sqcup \overrightarrow{\mathrm{q}_1 \mathrm{q}_2} \sqcup \cdots \sqcup \overrightarrow{\mathrm{q}_{j-1} \mathrm{q}_j} \sqcup \overrightarrow{\mathrm{q}_j \mathrm{q}_{j+1}})$.

We will assume that the discretization is small enough so that if u_j and u_{j+1} are reachable (via their respective homotopy classes), the points on the straight edge joining them will also be reachable. Thus, from the above observations, and using Proposition 1, we can conclude that the robot will be able to follow the trajectory defined by the projection of the path Q on to G, *i.e.*, the path $Q_G = [\mathrm{q}_s, \mathrm{q}_1, \mathrm{q}_2, \cdots, \mathrm{q}_{m-1}, \mathrm{q}_g]$.

Optimality follows from the fact that we used an optimal search algorithm to find path to a vertex of the form $(\mathrm{q}_g, *) \in \mathcal{V}(\widehat{G}_h)$, and that cost of the edges in $\widehat{G}_h$ are the Euclidean lengths of the edges. Note however that this optimality is in the discrete graph (*i.e.*, we found the shortest path in the discrete graph leading to a valid goal vertex).

For the A* search we use an admissible Heuristic function for 8-connected graph representation with Euclidean length of edges as their costs: $\underline{c}_{G_h}(\alpha, \beta) = |\Delta_x - \Delta_y| + \sqrt{2}\min(\Delta_x, \Delta_y)$, where $\alpha = (\mathrm{a}, \mathfrak{a}), \beta = (\mathrm{b}, \mathfrak{b}) \in \mathcal{V}(G_h)$; $\mathbf{a}, \mathbf{b} \in W$ are respectively the positions of the vertices $\mathrm{a}, \mathrm{b} \in \mathcal{V}(G)$; and $\Delta_x = |\mathbf{a}_x - \mathbf{b}_x|, \Delta_y = |\mathbf{a}_y - \mathbf{b}_y|$ are the differences in their x and y coordinates respectively. It can be proved that $\underline{c}_{G_h}$ indeed gives a lower bound on the actual cost of the path in G_h connecting the two vertices, and thus being an admissible heuristics will speed up the A* search without sacrificing optimality.

IV. DYNAMIC SIMULATION

To demonstrate the working of the algorithm we developed a dynamic simulation platform where we used a point model for the robot, modeled the cable as a series of rigid cylindrical segments connected by revolute joints, as well as considered the interaction between the cable and the obstacles (disk-shaped) by formulating the problem as a Linear Complementarity Problem (LCP). This dynamic simulation platform is built on our previous work [20], [4]. Here we briefly describe the mathematical model.

The cable is modeled as a series of n rigid cylindrical segments connected by revolute joints. The i^{th} segment is modeled as an uniform cylinder of length L_i, and its diameter, d_i, is assumed to be much smaller than its length. We choose the generalized coordinates to be $\theta = [\theta_1, \ldots, \theta_n]^T$, where θ_i is the angles made by the i^{th} cylindrical segment with respect to positive X axis of the global inertial frame of reference (see Fig. 6).

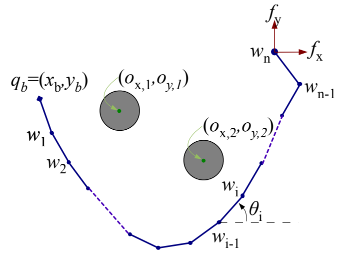


Fig. 6. The dynamic model.

Using Lagrangian mechanics, the equations of motion for the system can be written as

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{\theta}}\right) - \frac{\partial K}{\partial \theta} - Q_\theta = 0 \tag{1}$$

where, K, the kinetic energy of the system, is given by $K = \sum_{i=1}^{n}\left(\frac{1}{2}m_i\left(\dot{\mathbf{p}}_i \cdot \dot{\mathbf{p}}_i\right) + \frac{1}{2}\frac{m_i L_i^2}{12}\dot{\theta}_i^2\right)$ with m_i the mass, and $\mathbf{p}_i$ the center of mass of the i^{th} segment of the cable. The generalized forces are $Q_{\theta_i} = f_x L_i \sin\theta_i + f_y L_i \cos\theta_i,\ \forall i \in \{1, 2, \ldots, n\}$. For a simple demonstrative simulation we assume the absence of any friction or drag force.

Using a first order approximation for the acceleration in the equations of motion, (1), we can write at the k^{th} time instant,

$$M(\theta^k)\frac{\dot{\theta}^{k+1} - \dot{\theta}^k}{\Delta t^k} - \mathbf{V}(\dot{\theta}^k, \theta^k) - \mathbf{Q}^k = 0 \tag{2}$$

where $\mathbf{V}(\dot{\theta}^k, \theta^k) \in \mathbb{R}^n$ and $\mathbf{Q}^k = [Q_{\theta_1}, \ldots, Q_{\theta_n}]^T \in \mathbb{R}^n$ is the vector of generalized forces. Thus, using Euler integration, we can find the generalized states and velocities at the $(k+1)^{th}$ time step.

We assume a frictionless contact between the cable and the obstacles, and adopt a time-stepping algorithms to solve Linear Complementarity Problem at every step [1], [2], [25]. In order to account for *noninterpenetration* between the rigid bodies with smooth distance function, we assume that all the objects are disk-shaped, and that the thickness of cable segments are negligible, which is consistent with our earlier assumptions. We only consider the cable-obstacle contacts using distance functions as described in earlier work [20], and ignore the self-collision of the cable.

At a given generalized coordinate, θ^k, at time, t^k, we select and stack up all the distance functions whose values are less than some threshold, δ, (say m^k of them) into a vector, $\mathbf{f}(\theta^k) = \left[f_1(\theta^k), \ldots, f_{m^k}(\theta^k)\right]^T$, *i.e.* $f_p(\theta^k) < \delta$ for $p = 1, 2, \ldots, m^k$. Then our goal is to find the state in the next step which satisfies $f_p(\theta^{k+1}) \geq 0$ (*i.e.*, there is no penetration of the cable segments into the obstacles). This results in a nonlinear problem. We thus linearize the distance functions around θ^k using a first order approximation as follows [1], [2], [25],

$$f_p(\theta^{k+1}) \simeq f_p(\theta^k) + \Delta t^k \nabla_\theta f_p(\theta^k)\dot{\theta}^{k+1} \tag{3}$$

which is linear with respect to the generalized velocity of the next time step. We thus incorporate impulse, λ^k, due to the contacts into the generalized force term of Eqn. (2), and thus construct the equations of motion which satisfy the noninterpenetration constraints

$$\begin{aligned} M^k\dot{\theta}^{k+1} &= M^k\dot{\theta}^k + \Delta t^k\left(\mathbf{V}^k + \mathbf{Q}^k\right) + \nabla_\theta \mathbf{f}(\theta^k)^T\lambda^k \\ &= M^k\dot{\theta}_u^k + \nabla_\theta \mathbf{f}(\theta^k)^T\lambda^k \end{aligned} \tag{4}$$

where $M^k = M(\theta^k)$, $\mathbf{V}^k = \mathbf{V}(\dot{\theta}^k, \theta^k)$, $\mathbf{Q}^k = \mathbf{Q}(\theta^k)$ and $\lambda^k = \left[\lambda_1^k, \ldots, \lambda_{m^k}^k\right]^T$ is a vector of nonnegative components (since the impulse due contact of two rigid bodies should be in a direction such that the distance functions increase). To simplify the notation, we define $\dot{\theta}_u^k$ to be the generalized velocity of the next time step when there is no impulse due to noninterpenetration constraints, which satisfies the Eqn. (2). Then we substitute this equation into the noninterpenetration constraints to obtain a linear inequality,

$$\begin{aligned} &\mathbf{f}(\theta^k) + \Delta t^k \nabla_\theta \mathbf{f}(\theta^k)\left(M^k\right)^{-1}\left(M^k\dot{\theta}_u^k + \nabla_\theta \mathbf{f}(\theta^k)^T\lambda^k\right) \\ =&\mathbf{f}(\theta^k) + \Delta t^k \nabla_\theta \mathbf{f}(\theta^k)\dot{\theta}_u^k + \Delta t^k \nabla_\theta \mathbf{f}(\theta^k)\left(M^k\right)^{-1}\nabla_\theta \mathbf{f}(\theta^k)^T\lambda^k \\ =&\mathbf{b}^k + A^k\lambda^k \geq 0 \end{aligned} \tag{5}$$

where $\mathbf{b}^k = \mathbf{f}(\theta^k) + \Delta t^k \nabla_\theta \mathbf{f}(\theta^k)\dot{\theta}_u^k \in \mathbb{R}^{m^k}$ and $A^k = \Delta t^k \nabla_\theta \mathbf{f}(\theta^k)\left(M^k\right)^{-1}\nabla_\theta \mathbf{f}(\theta^k)^T \in \mathbb{R}^{m^k \times m^k}$ are known, given by the current generalized coordinate and velocity, θ^k and $\dot{\theta}^k$. Hence we need to solve a linear complementarity problem (LCP),

$$\begin{gathered} \mathbf{b}^k + A^k\lambda^k \geq 0, \quad \lambda^k \geq 0 \\ \lambda^k \cdot \left(\mathbf{b}^k + A^k\lambda^k\right) = 0 \end{gathered} \tag{6}$$

at each time step. We can solve this LCP efficiently with Lemke's method [9], [11]. We then substitute the solution of λ^k into the following equation (derived from Eqn. (4)) to find the generalized state and velocity for the next time step.

$$\dot{\theta}^{k+1} = \dot{\theta}_u^k + \left(M^k\right)^{-1}\nabla_\theta \mathbf{f}(\theta^k)^T\lambda^k \tag{7}$$

$$\theta^{k+1} = \theta^k + \Delta t^k\dot{\theta}^{k+1}. \tag{8}$$

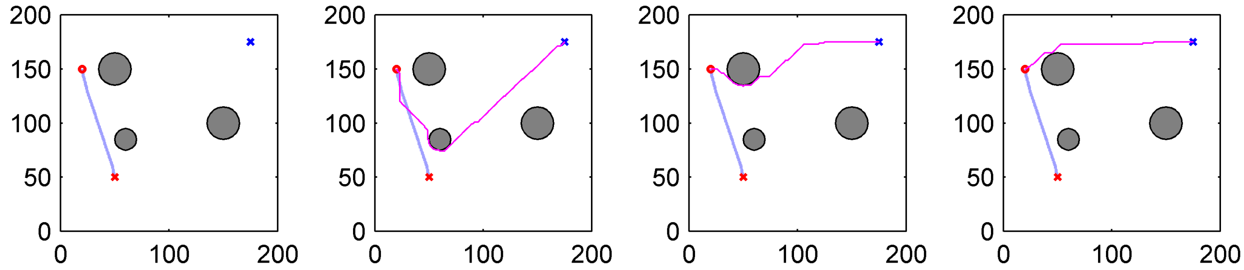
V. SIMULATION RESULTS

We implemented the search algorithms (*Step* 1 and *Step* 2) in C++ programming language with ROS integration. The lengths of the cable mentioned in this section are all in the discretization units of the respective examples. Obstacles shown in the figures are assumed to be *inflated* versions of the original given obstacles, so as to avoid collisions, and thus letting us consider a point model for the robot.

Figure 7 shows the simulation result in a simple 200×200 discretized environment. The base is located at $\mathbf{q}_b = [50, 50]$ (the red X mark), the initial robot position is $\mathbf{q}_s = [20, 150]$ (the red dot), and the initial cable configuration is schematically shown in light blue (Figure 7(a)). Figure 7(d) shows the unrestricted shortest path, which is traversable if the cable is sufficiently long (plan obtained with $L = 260$ and $L = 280$). However, as the cable gets shorter the robot needs to travel longer distance using different homotopy classes to reach the goal as shown in Figures 7(c) and 7(b).

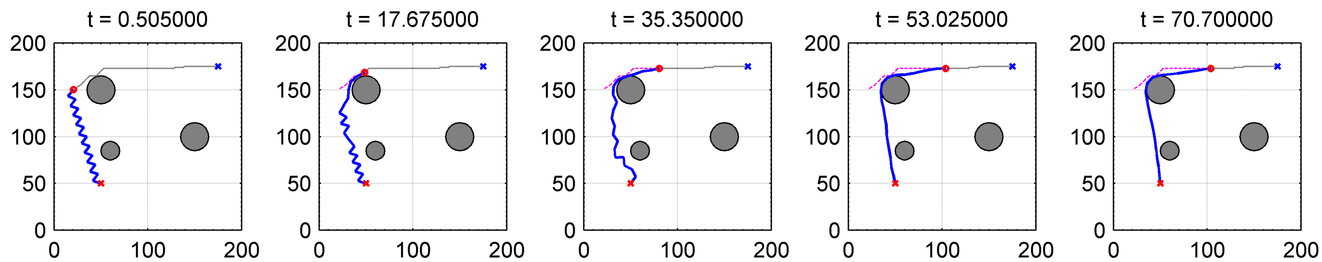
We demonstrate the applicability of the algorithm in a dynamic simulation using the same environment as in Figure 7. In the dynamic simulation (screen-shots shown in Figure 8) we set the cable length at 180 discretization units. Figure 8(a) shows the robot trying to travel the path obtained with $L = 260$ (significantly long so as to obtain the globally shortest path). As evident, the cable length of 180 units falls short in trying to traverse the planned path. However, the path obtained with $L = 180$ (Figure 8(b)) is traversable.

Figures 9 and 10 illustrates the result in larger and more realistic environments. In both the figures the base of the cable is marked by the red cross, the robot's initial position is the red circle, the initial cable configuration is shown in light blue, and the target position of the robot is shown in

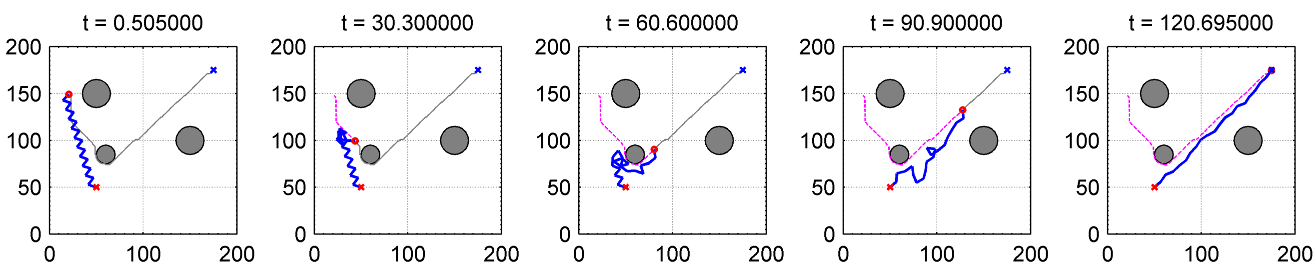


(a) The intial cable configuration and robot. (b) The optimal path when the cable length is $L = 180$. (c) The optimal path when the cable length is $L = 200, 220, 240$. (d) The optimal path when the cable length is $L = 260, 280$.

Fig. 7. Planned trajectories in simple 200×200 discretized environment for different values of maximum cable length, L. The gray cells are the obstacles. The light blue curve is the cable, with the red cross as its base and red dot the initial robot position. The magenta curves are the planned robot trajectories.



(a) The dynamic simulation of traversing the globally shortest path planned not considering the cable length constraint.

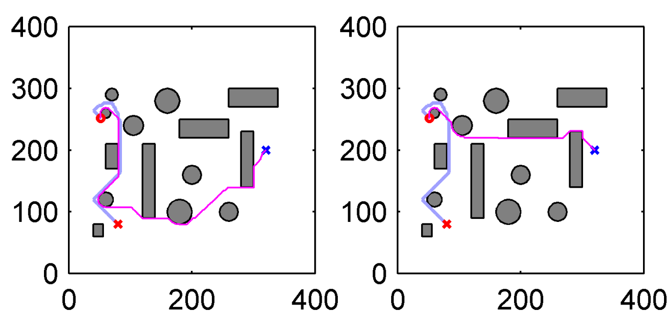


(b) The dynamic simulation of traversing path planned considering the cable length constraint of $L = 180$.

Fig. 8. The dynamic simulation results in the same 200×200 environment as in Figure 7. The cable length in the dynamic model is fixed at 180 discretization units.

blue cross. The planned trajectory for the different values of L are shown in the magenta curves.

All computations were performed on a Intel i7-3770 3.40 GHz processor with 16 GB memory. Table I shows the number of expanded vertices and computation time for both *step* 1 and *step* 2 of the algorithm for each simulation. A clear positive correlation can be observed between the length of the cable and the number of states expanded in *step* 1. However, longer the cable is, the robot is allowed to move in a larger feasible subgraph of G_h, and hence it reduces the number of expanded vertices and computation time for *step* 2. Note that the computation time mentioned is up to two significant places of decimal only.

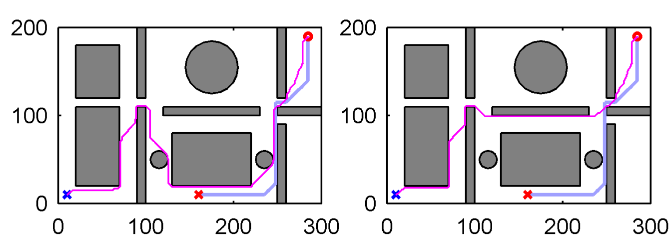


(a) The optimal path with $L = 300$. (b) The optimal path with $L = 400$.

Fig. 9. A 400×400 discretized environment. $\mathbf{q}_b = [80, 80]$ (red cross), $\mathbf{q}_s = [52, 252]$ (red circle), $\mathbf{q}_g = [320, 200]$ (blue cross).

VI. CONCLUSIONS

In this paper we describe an efficient algorithm for planning trajectories for robots tethered to a base by a flexible cable of fixed length (or an elastic cable of maximum length), L, in a cluttered environment. The objective is to find the shortest path from the initial robot-cable configuration to a final robot position. Obstacles in the environment and cable length constraints give rise to topological non-triviality in the problem, which we resolve in two steps: *i.* By constructing



(a) The optimal path with $L = 350$. (b) The optimal path with $L = 450$.

Fig. 10. A 300×200 discretized environment. $\mathbf{q}_b = [160, 10]$ (red cross), $\mathbf{q}_s = [285, 190]$ (red circle), $\mathbf{q}_g = [10, 10]$ (blue cross).

Environment		200×200 Figure 7(b)	200×200 Figure 7(c)	200×200 Figure 7(d)	400×400 Figure 9(a)	400×400 Figure 9(b)	300×200 Figure 10(a)	300×200 Figure 10(b)
L (discretization units)		180	200	260	300	400	350	450
Step 1	Number of vertices expanded	105505	131877	260584	1385422	9668053	396162	1244385
	Computation time (s)	0.75	1.14	3.13	33.12	459.71	6.04	29.07
Step 2	Number of vertices expanded	23515	4406	548	219811	47579	110313	33995
	Computation time (s)	0.01	0.00	0.00	0.11	0.03	0.05	0.02

TABLE I

COMPUTATION TIME AND NUMBER OF VERTICES EXPANDED.

a *homotopy augmented graph* we find the subgraph of it that is reachable via different homotopy classes of the cable, and, *ii.* once the reachable workspace in different homotopy classes is constructed, the shortest path in the subgraph of the homotopy augmented graph becomes the desired trajectory of the robot. We illustrated the algorithm using simulations in cluttered environments, and demonstrated its applicability using a dynamic simulation testbed.

ACKNOWLEDGMENT

We gratefully acknowledge the support of the Office of Naval Research grant numbers N00014-07-1-0829 and N00014-09-1-1031, the Army Research Laboratory grant number W911NF-10-2-0016 and the Air Force Office of Scientific Research grant number FA9550-10-1-0567.

REFERENCES

[1] Mihai Anitescu. A fixed time-step approach for multibody dynamics with contact and friction. In *International Conference on Intelligent RObots and Systems - IROS*, 2003.

[2] Mihai Anitescu and Gary D. Hart. A fixed-point iteration approach for multibody dynamics with contact and small friction. *Mathematical Programming*, 101(1):3–32, 2004.

[3] Subhrajit Bhattacharya. *Topological and Geometric Techniques in Graph-Search Based Robot Planning*. PhD thesis, University of Pennsylvania, January 2012.

[4] Subhrajit Bhattacharya, Hordur Heidarsson, Gaurav S. Sukhatme, and Vijay Kumar. Cooperative control of autonomous surface vehicles for oil skimming and cleanup. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 9-13 May 2011.

[5] Subhrajit Bhattacharya, Maxim Likhachev, and Vijay Kumar. Topological constraints in search-based robot path planning. *Autonomous Robots*, pages 1–18, June 2012. DOI: 10.1007/s10514-012-9304-1.

[6] Stephen Cass. DARPA Unveils Atlas DRC Robot, Jul 2013. http://spectrum.ieee.org/automaton/robotics/humanoids/darpa-unveils-atlas-drc-robot.

[7] K.S. Chou and X.P. Zhu. *The Curve Shortening Problem*. Taylor & Francis, 2010.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press, 2nd edition, 2001.

[9] R. Cottle, J.S. Pang, and R.E. Stone. *The linear complementarity problem*. Classics in applied mathematics. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 1992.

[10] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[11] Paul L. Fackler and Mario J. Miranda. Lemke: Solver for standard linear complementarity problems (lcps), 2002.

[12] D. Grigoriev and A. Slissenko. Polytime algorithm for the shortest path in a homotopy class amidst semi-algebraic obstacles in the plane. In *ISSAC '98: Proceedings of the 1998 international symposium on Symbolic and algebraic computation*, pages 17–24, New York, NY, USA, 1998. ACM.

[13] Erico Guizzo. Fukushima Robot Operator Writes Tell-All Blog, Aug 2011. http://spectrum.ieee.org/automaton/robotics/industrial-robots/fukushima-robot-operator-diaries.

[14] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems, Science, and Cybernetics*, SSC-4(2):100–107, 1968.

[15] Allen Hatcher. *Algebraic Topology*. Cambridge Univ. Press, 2001.

[16] J. Hershberger and J. Snoeyink. Computing minimum length paths of a given homotopy class. *Comput. Geom. Theory Appl*, 4:331–342, 1991.

[17] S. Hert and V. Lumelsky. The ties that bind: motion planning for multiple tethered robots. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pages 2734–2741 vol.4, 1994.

[18] Takeo Igarashi and Mike Stilman. Homotopic path planning on manifolds for cabled mobile robots. In David Hsu, Volkan Isler, Jean-Claude Latombe, and MingC. Lin, editors, *Algorithmic Foundations of Robotics IX*, volume 68 of *Springer Tracts in Advanced Robotics*, pages 1–18. Springer Berlin Heidelberg, 2011.

[19] Soonkyum Kim, Subhrajit Bhattacharya, Hordur Heidarsson, Gaurav Sukhatme, and Vijay Kumar. A topological approach to using cables to separate and manipulate sets of objects. In *Proceedings of the Robotics: Science and System (RSS)*, Syndey, Australia, June 24-28 2013.

[20] Soonkyum Kim, Subhrajit Bhattacharya, and Vijay Kumar. Dynamic simulation of autonomous boats for cooperative skimming and cleanup. In *Proceedings of the ASME International Design Technical Conferences and Computer and Information in Engineering Conference*, Portland, OREGON, Aug 4-7 2013.

[21] Nathan Michael, Shaojie Shen, Kartik Mohta, Yash Mulgaonkar, Vijay Kumar, Keiji Nagatani, Yoshito Okada, Seiga Kiribayashi, Kazuki Otake, Kazuya Yoshida, Kazunori Ohno, Eijiro Takeuchi, and Satoshi Tadokoro. Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *Journal of Field Robotics*, 29(5):832–841, 2012.

[22] K.S. Pratt, R.R. Murphy, J.L. Burke, J. Craighead, C. Griffin, and S. Stover. Use of tethered small unmanned aerial system at berkman plaza ii collapse. In *Safety, Security and Rescue Robotics, 2008. SSRR 2008. IEEE International Workshop on*, pages 134–139, 2008.

[23] W.R. Scott and W.R. Scott. *Group Theory*. Dover Books on Mathematics Series. Dover Publ., 1964.

[24] Iddo Shnaps and Elon Rimon. Online coverage by a tethered autonomous mobile robot in planar unknown environments. In *Proceedings of Robotics: Science and Systems*, Berlin, Germany, June 2013.

[25] David Stewart and J. C. Trinkle. An implicit time-stepping scheme for rigid body dynamics with coulomb friction. *International Journal Of Numerical Methods In Engineering*, 39:2673–2691, 1996.

[26] Benjamn Tovar, Fred Cohen, and Steven M. LaValle. Sensor beams, obstacles, and possible paths. In *Workshop on the Algorithmic Foundations of Robotics*, pages 317–332, 2008.

[27] 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*, volume 2, pages 1011–1017 vol.2, 1999.

[28] Ning Xu, Peter Brass, and Ivo Vigan. An improved algorithm in shortest path planning for a tethered robot. Technical report.