



A topological approach to using cables to separate and manipulate sets of objects

Subhrajit Bhattacharya¹, Soonkyum Kim², Hordur Heidarsson³,
Gaurav S. Sukhatme³ and Vijay Kumar¹

Abstract

In this paper we study the problem of manipulating and transporting multiple objects on the plane using a cable attached at each end to a mobile robot. This problem is motivated by the use of boats with booms in skimming operations for cleaning oil spills or removing debris on the surface of the water. The goal in this paper is to automate the task of separating the objects of interest from a collection of objects by manipulating them with cables that are actuated only at the ends, and then transporting them to specified destinations. Because the cable is flexible, the shape of the cable must be explicitly modeled in the problem. Further, the robots must cooperatively plan motions to achieve the required cable shape and gross position/orientation to separate the objects of interest and then transport them as specified. The theoretical foundation for the problem is derived from topological invariants, homology and homotopy. We first derive the necessary topological conditions for achieving the desired separation of objects. We then propose a distributed search-based planning technique for finding optimal robot trajectories for separation and transportation. We demonstrate the applicability of this method using a dynamic simulation platform with explicit models of the cable dynamics, the contact between the cable and one or more objects, and the surface drag on the cable and on the objects. We also demonstrate the working of the proposed algorithm on an experimental platform consisting of a system of two cooperating autonomous surface vessels and stationary/anchored objects.

Keywords

Motion planning, topology, cable, object separation

1. Introduction

This paper addresses the motion planning for and control of the shape of a flexible cable to separate a specified set of objects from other objects and to transport the specified objects to a destination. Object manipulation is of course an important problem in robotics. Certainly conventional approaches to manipulation using robot arms with grippers has received considerable attention and is well understood (Dogar et al., 2012; Ivan et al., 2013). In contrast, we are interested in the use of mobile robots to contact and manipulate objects without special purpose effectors. This allows more versatility but leads to many challenges. One approach relies on caging an object using multiple mobile robots. This problem has been studied for planar objects (Fink et al., 2008). However, the ratio between the number of objects manipulated at a time, and the number of robots required for doing that is small, thus making such an approach highly inefficient for manipulating a large number of objects and for separating objects in a field with obstacles. In contrast, we propose a framework for manipulating a large number of objects with only a pair of robots.

The advantages of using ropes with robots for manipulation were demonstrated by Donald et al. (2000). An interesting problem that arises in these settings is the modeling of the shape of the cable and the motion planning for the robots to control the position and shape of the cable. Motion planning for manipulation of rope-like flexible objects is discussed by Saha et al. (2006). The problem of entangling and disentangling knots and the motion planning for this problem has been addressed by Lamiroux and Kavraki (2001). Our goal, however, is the motion planning that is required to manipulate objects on the plane and we are less interested in the specific configuration of the cable. The use of robots to tow objects using cables is discussed

¹University of Pennsylvania, Philadelphia, PA, USA

²Carnegie Mellon University, Pittsburgh, PA, USA

³Department of Electrical Engineering, University of Southern California, Los Angeles, CA, USA

Corresponding author:

Subhrajit Bhattacharya, University of Pennsylvania, David Rittenhouse Laboratory, 209 South 33rd Street, Philadelphia, PA 19104, USA.
Email: subhrahb@math.upenn.edu

by Jiang and Kumar (2010) and Cheng et al. (2009). An extension of these ideas leads to using a cable with its ends tied to robots to cage and tow objects. Indeed this method is widely used in skimming operations on water surfaces (Kerr, 2010; Robertson and Krauss, 2010). A description of the dynamics of such systems and an analysis of the problem of cooperative skimming are provided by Bhattacharya et al. (2011) and Aranda et al. (2006). However, this work does not explicitly address the manipulation of objects.

In this paper, we discuss the planning and control of the motions of two robots, each of which is tied to one end of a flexible cable, with the goals of: (a) separating a specified set of objects from other objects; and (b) to transport the specified objects to a destination. The first step, as one might expect, is to navigate the robots around the objects so that the cable separates the objects of interest from those that are not of interest. The problem of finding a hypersurface separating two types of objects is studied as part of statistical classification problems (Binder, 1981; Suykens and Vandewalle, 1999). However, such methods are susceptible to finding curves that can have disjoint components, do not have guarantees on optimality, and are statistical in nature. Moreover, the problem of finding a *separating* cable configuration (the curve) that separates the objects does not give us a necessary means of finding the trajectories of the robots that achieve that configuration. A deeper understanding of the topology of the configuration space of a system, and the use of topological techniques in problems of robot coverage and motion planning, have recently received significant attention within the robotics community (Tovar et al., 2008; Bhattacharya et al., 2012; Kuderer et al., 2013; Derenick et al., 2013; Narayanan et al., 2013; Ivan et al., 2013) and have proven to be very effective. In this paper we discover that a topological approach to the problem under consideration is the most natural and complete. The first key contribution of this paper is a topological description of the problem of separating two sets of objects and the algebraic formulation of the separation problem. The second contribution is a complete motion planning algorithm that relies on graph search (Cormen et al., 2001) to drive the robots in order to achieve separation and then transport the objects to specified destinations. We also derive a decoupled algorithm that has the advantage of only requiring to plan in the individual robot's configuration space instead of the joint state-space.

2. Problem description

We consider the scenario where there are two classes of objects present in a flat enclosed region, \mathcal{W} . For convenience we will refer to the two classes as “blue” and “red”. Without loss of generality, one of these classes of objects will be considered to be of interest (i.e. those need to be manipulated and transported), while the other consists of obstacles or objects that are not of interest. A flexible cable is attached, at its two ends, to two robots that are capable

of navigating on the flat surface. Given an initial configuration of the cable and the robots (Figure 1(a)), we need to first make the robots follow trajectories to the boundary of the enclosed region, $\partial\mathcal{W}$, such that the final cable configuration “separates” the objects from the ones, which we call a *separating configuration* (Figure 1(c)). Once that is achieved, the robots can move along $\partial\mathcal{W}$ to enclose one type of objects and “pull” them out, thus separating and transporting those objects.

Suppose \mathbf{e} and \mathbf{s} are the points on the boundary reached by the robots so that they split $\partial\mathcal{W}$ into $\partial\mathcal{W}_1$ and $\partial\mathcal{W}_2$ as in Figure 1(b). It is clear that the robot trajectories and cable configurations that describe the problem and achieve the desired objective are sufficiently described up to homotopy. That is, if C_1 and C_2 are two cable configurations that are in the same homotopy class (Bhattacharya et al., 2012), then, “ C_1 separates the two types of objects” “ C_2 separates the two types of objects” (Figure 2(a)). Likewise, if a particular set of robot trajectories, $\{\tau_1, \tau_2\}$, carry the cable from the initial configuration to the desired separating configuration (up to homotopy), another set of trajectories, $\{\tau'_1, \tau'_2\}$, that are homotopic to the first set (i.e. $\tau'_1 \sim \tau_1$ and $\tau'_2 \sim \tau_2$) will achieve the same objective.

In addition to this, it should also be noted that the homotopy class of the cable configuration that achieves the separation of the two types of objects is not unique either. For example, in Figure 2(a), the configuration C_3 is in a different homotopy class from C_1 or C_2 , but still separates the two types of objects; C' in Figure 2(c) is another example. Furthermore, for a given desired separating configuration of the cable (up to homotopy), the homotopy classes of the robot trajectories that can carry the cable from its initial configuration to the separating configuration, are not unique either (Figure 2(b)). Thus, it is useful to develop a notion of optimality to more precisely define the problem objectives. It is natural to use the length of the robot trajectories to the optimization criteria.

For the theoretical foundation and for setting up the optimization problem, we will make the following assumptions.

- (i) The objects are assumed to be stationary rigid bodies; that is, the cable cannot “pass through” any of the objects, and that on contact of the cable with the objects the objects do not move. In the implementation (Section 5.3) we will however relax the conditions that the objects need to be stationary.
- (ii) The cable is flexible, and there is no restriction on the length of the cable (i.e. the cable will not fall short and tug on the robots). We assume that the cable can either be spooled out as required from a cable reel residing on the robots, or may stretch as in an elastic band.

Note that by restricting the robots' motion to a plane and assuming that they can drive over the cable, we implicitly eliminate the possibility of creating knots in the cable. In

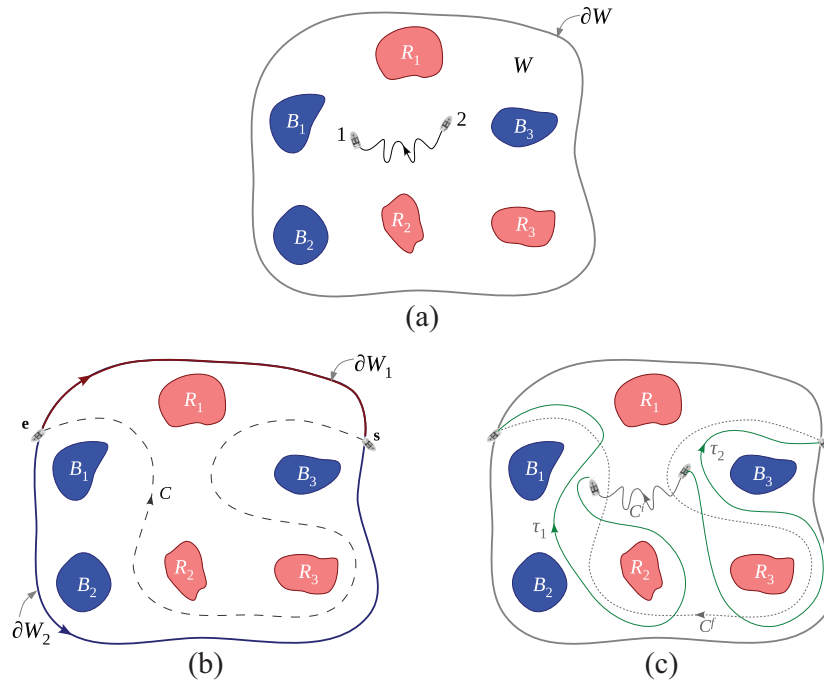


Fig. 1. The problem of separating the two types of objects (objects labeled as R_i and objects labeled as B_i). (a) The initial configuration of the cable and the two robots in the workspace W with boundary ∂W . (b) A separating configuration of the cable, C , that separates the two types of objects. (c) A possible set of trajectories that take the cable from the initial configuration, C^i , to a configuration homotopic to the separating configuration, C^f .

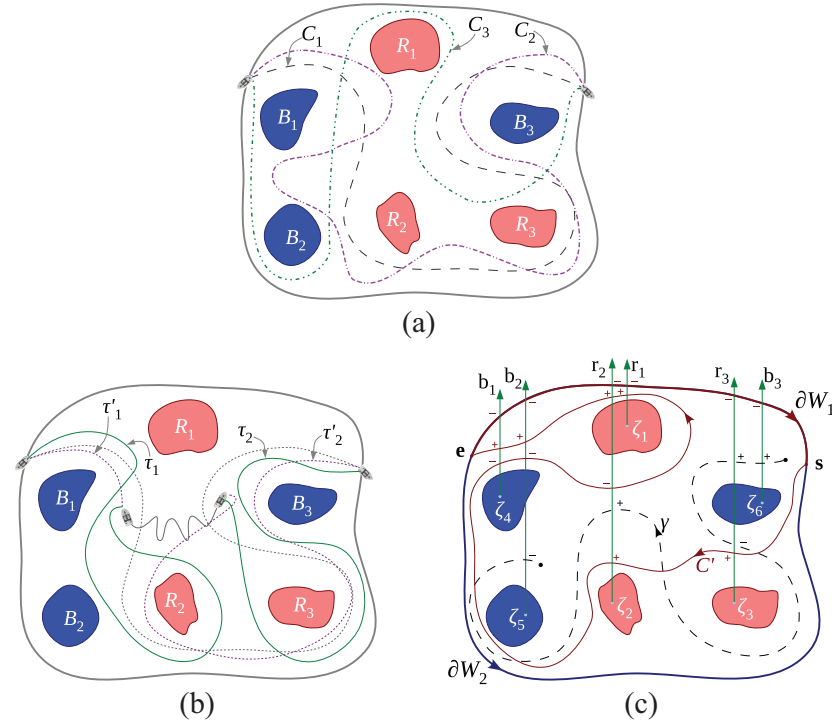


Fig. 2. Homotopy and homology classes of cable configurations and trajectories. (a) Three possible cable configurations separating the two types of objects. Here C_1 and C_2 are homotopic, but C_3 belongs to a different homotopy class. See the curve C' in Figure 2(c) for yet another cable configuration that separates the two types of objects. (b) The robot trajectories (up to homotopy) that can take the cable to a desired separating configuration (up to homotopy) are not unique. In this figure, τ_1 and τ'_1 are not homotopic, neither are τ_2 and τ'_2 . But either of the sets of trajectories, $\{\tau_1, \tau_2\}$ or $\{\tau'_1, \tau'_2\}$, take the cable to the homotopy class shown in Figure 1b. (c) Representative points ζ_i inside the objects, $R_1, R_2, \dots, R_r, B_1, B_2, \dots, B_b$ (in that order), and $r_i, i = 1, \dots, r$ and $b_j, j = 1, \dots, b$ are rays emanating from the respective points. Using the bump forms corresponding to the rays in defining the H -signature, $H(\gamma) = [0, 1, 0, 0, -1, 1]$. And, $h(\gamma) = "b_3 r_2 b_2^{-1}"$.

order to create a knot or a tangle a robot will need to move out of the plane, which we do not allow.

2.1. Main contributions and organization of the paper

The main contribution of the paper is the use of topology-based algorithmic tools to solve the problem of separation and manipulation of sets of objects using a flexible cable carried by two robots. As described above and illustrated in Figure 2, this problem is fundamentally topological. In Section 3 we thus start with some basic definitions related to topological descriptions of curves, and introduce some topological invariants that are fundamental to our algorithm. The main algorithmic contribution appears in Section 4, where we describe how we construct graphs that naturally incorporate the topological information such that finding a solution to the problem essentially boils down to performing an optimal search (such as A* or Dijkstra's) in the constructed graphs.

A notion of optimality is necessary in order to resolve the multiplicity of the possible solutions as illustrated above. In particular, in the joint state-space version of the algorithm described in Section 4.1 we will restrict ourselves to the optimization of sum of the lengths of the robot trajectories. This is an appropriate choice since trajectory lengths are directly related to the energy required to complete the task (e.g. amount of fuel that the robots need to burn to cover the trajectories) as well as the time required to execute the trajectories. However, more complex objective functions can be used. In the decoupled and distributed planning discussed in Section 4.2, we also consider the maximum of the two trajectory lengths of the two robots as the objective that is to be minimized.

We demonstrate the practicality of the proposed algorithm in solving the real physical problem through dynamic simulations and field experiments as discussed in Section 5. For all of the trajectory planning we inflate the obstacle/object map and plan the trajectories in the *inflated map* so that the planned trajectories maintain a finite separation from objects in order to avoid collision between the robots and the objects/obstacles.

3. Theoretical foundation

Let W be a two-dimensional simply connected and bounded region. Suppose it contains a set of objects, $\mathcal{O} = R_1 \cup R_2 \cup \dots \cup R_r \cup B_1 \cup B_2 \cup \dots \cup B_b \subseteq W$, where R_1, R_2, \dots, R_r are r counts of objects, and B_1, B_2, \dots, B_b are b counts of objects. Each object, R_i or B_j , is assumed to be connected. Thus, throughout this paper we will denote the "free" space (the configuration space) of the individual point robots as $X := (W - \mathcal{O})$ (we will use both notations interchangeably).

3.1. Curves in a topological space

Both cable configurations and robot trajectories are one-dimensional curves in $(W - \mathcal{O})$. They can thus be defined as continuous maps from the interval $[0, 1]$ to $(W - \mathcal{O})$.

In a topological space X , a curve, $\gamma: [0, 1] \rightarrow X$, is called *closed* if $\gamma(0) = \gamma(1)$. A curve, γ , is called *embedded* (Munkres, 1999) if $\gamma(t) \neq \gamma(t')$, for all $t \neq t'$ (i.e. the curve does not intersect itself). In our problem we will only require that the separating cable configuration be embedded (Proposition 1), but other curves need *not* be embedded.

Given a curve, $f: [0, 1] \rightarrow X$, we define the following curves:

- (i) For $0 \leq \alpha < \beta \leq 1$, $f|_{[\alpha, \beta]}: [0, 1] \rightarrow X$
 $t \mapsto f((\beta - \alpha)t + \alpha)$

That is, $f|_{[\alpha, \beta]}$ is the part of the curve lying between the parameter values of α and β .

- (ii) $-f: [0, 1] \rightarrow X$. That is, $-f$ is the same curve as f ,
 $t \mapsto f(1 - t)$

but with opposite orientation.

- (iii) for another curve $g: [0, 1] \rightarrow X$ with $f(1) = g(0)$,
 $f \sqcup g: [0, 1] \rightarrow X$, . That is, $f \sqcup g$ is
 $t \mapsto \begin{cases} f(2t), & \text{if } t < 1/2 \\ g(2t - 1), & \text{if } t \geq 1/2 \end{cases}$

the curve obtained by "joining" the end of f with the start of g .

The line integral of a differential 1-form, $\omega = udx + vdy$, over a curve γ is defined as $\int_{\gamma} \omega := \int_0^1 (u\dot{\gamma}_x + v\dot{\gamma}_y) dt$.

3.2. Homology and homotopy invariants

Definition 1 (Homology classes of curves). In a topological space X , two curves $\gamma_1, \gamma_2: [0, 1] \rightarrow X$ connecting the same start and end points, are called *homologous* (or belong to the same homology class) iff γ_1 together with γ_2 (the latter with opposite orientation) forms the complete boundary of a (oriented) two-dimensional subset embedded in X (see Figure 3).

Formally, in the notation of Hatcher (2001), γ_1 and γ_2 are homologous iff $\gamma_1 \sqcup -\gamma_2$ belongs to the trivial class of the first homology group of X , denoted by $H_1(X)$. That is, $[\gamma_1 \sqcup -\gamma_2] = \mathbf{0} \in H_1(X)$ (Bhattacharya et al., 2012; Hatcher, 2001).

A *homology invariant* is a function, H , from the space of all curves in $(W - \mathcal{O})$ (with fixed end points) to another much smaller space (in this case, a vector space), such that $H(\gamma_1) = H(\gamma_2)$ iff γ_1 is homologous to γ_2 .

In our previous study (Bhattacharya et al., 2012) we proposed a homology class invariant (called the H -signature) that is based on simple results from complex analysis. However, the possible choice of such invariants has been broadened in Bhattacharya et al. (2013), where the choice of the vector of differential 1-forms, which needs to be

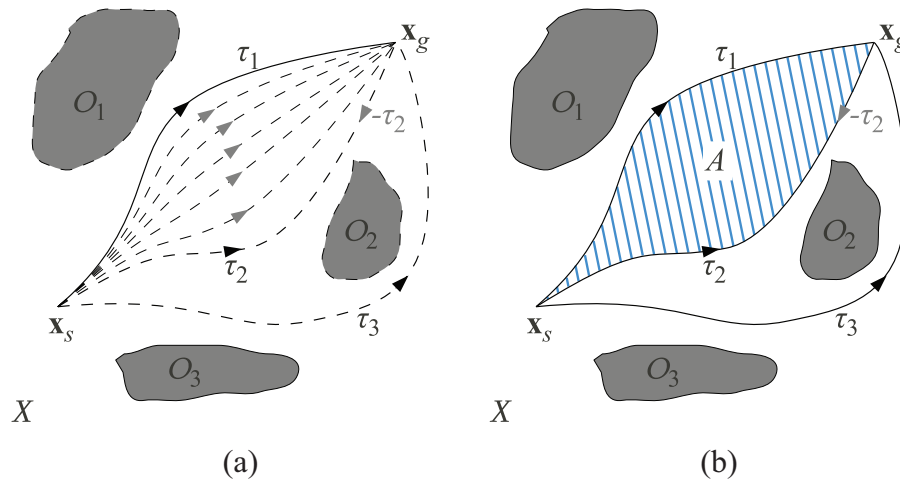


Fig. 3. Illustration of homotopy and homology equivalences. In this example τ_1 and τ_2 are both homotopic as well as homologous. (a) τ_1 is homotopic to τ_2 since there is a continuous sequence of trajectories representing deformation of one into the other. τ_3 belongs to a different homotopy class since it cannot be continuously deformed into any of the other two. (b) τ_1 is homologous to τ_2 since there exists an area A (shaded region) such that $\tau_1 \sqcup -\tau_2$ is the boundary of A . τ_3 belongs to a different homology class since such an area does not exist between τ_3 and any of the other two trajectories.

integrated over γ to obtain the invariant, has been proven to be any complete set of generators of the *de Rham cohomology group*, $H_{dR}^1(W - \mathcal{O})$. In particular, the *bump 1-forms* (Bott and Tu, 1982), $\omega_j = -v(y - \zeta_{j,y})\delta(x - \zeta_{j,x}) dx$, (where δ is the *Dirac delta function*, and its integral, v , is the *heaviside step function*; that is, informally speaking, ω_j are analogous to a *Dirac delta distribution* over rays emanating from *representative points* ζ_j along positive Y -axis) is a choice that has the simple interpretation of counting the number of times the curve, γ , intersects the ray emanating from ζ_j (see Figure 2(c)). In particular, define $\#_j \gamma := \int_\gamma \omega_j = (\text{Number of times } \gamma \text{ crosses the ray emanating from } \zeta_j \text{ from left to right}) - (\text{Number of times } \gamma \text{ crosses the ray emanating from } \zeta_j \text{ from right to left})$. Then, $H(\gamma) = [\#_1 \gamma, \#_2 \gamma, \dots, \#_n \gamma]^T$. For closed loops the value of H -signature will not depend (up to a scalar multiple) on the choice of the differential 1-forms, as long as they form a generating set of the first *de Rham cohomology group*, $H_{dR}^1(W - \mathcal{O})$ (Bott and Tu, 1982), and the elements in the vector will compute the *winding numbers* about ζ_j . Let $H(\gamma)$ be the H -signature of any curve, $\gamma : [0, 1] \rightarrow (W - \mathcal{O})$, with respect to the objects $R_1, R_2, \dots, R_r, B_1, B_2, \dots, B_b$ (in that order).

In this paper, however, we will require the concept of *homotopy* more often, which is described next. We use a lowercase ‘ h ’ for homotopy (as in h -signature or h -augmented graph as described later), as opposed to an uppercase ‘ H ’ for homology.

Definition 2 (Homotopy classes of curves). *In a topological space, X , two curves $\gamma_1, \gamma_2: [0, 1] \rightarrow X$ 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; refer to Figure 1(a) of Bhattacharya et al. (2012).*

Formally, if $\gamma_1: [0, 1] \rightarrow X$ and $\gamma_2: [0, 1] \rightarrow X$ represent the two trajectories (with $\gamma_1(0) = \gamma_2(0) = x_s$ and $\gamma_1(1) = \gamma_2(1) = x_g$), then γ_1 is homotopic to γ_2 iff there exists a continuous map $\eta: [0, 1] \times [0, 1] \rightarrow X$ 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 $\eta(0, \gamma) = x_s$, $\eta(1, \mu) = x_g \forall \mu \in [0, 1]$ (Bhattacharya et al., 2012; Hatcher, 2001). Alternatively, in the notation of Hatcher (2001), γ_1 and γ_2 are homotopic iff the closed curve $\gamma_1 \sqcup -\gamma_2$ belongs to the trivial class of the first homotopy group of X , denoted by $\pi_1(X)$. That is, $[\gamma_1 \sqcup -\gamma_2] = \mathbf{0} \in \pi_1(X)$.

As described by Bhattacharya et al. (2012), two curves being homotopic implies that they be homologous as well. But the converse is not necessarily true.

Homotopy invariants, in general, are much more difficult to design and compute. *Homotopy groups*, unlike *homology groups*, do not have the natural structure of a vector space (Hatcher, 2001). However, for curves in a two-dimensional plane with punctures (i.e. obstacles/objects) as in $(W - \mathcal{O})$, there are some relatively simple representations of the homotopy group and a way of computing the homotopy class of a given curve. In the literature (Hershberger and Snoeyink, 1991; Grigoriev and Slissenko, 1998; Tovar et al., 2008; Hatcher, 2001; Bhattacharya, 2012; Narayanan et al., 2013) several similar, but different representations have been proposed, all concerning construction of “words” as homotopy class invariants. As discussed in Appendix B, the equivalence of these representations can be shown. We choose a representation that traditionally appear in topology literature (Hatcher, 2001) and is one of the simplest as far as the required construction is concerned. This homotopy invariant is described as follows:

We consider *representative points*, ζ_i as before, and *non-intersecting rays* (we choose them to be parallel), r_1 ,

r_2, \dots, r_r and b_1, b_2, \dots, b_b , emanating from the and objects, respectively (Figure 2(c)). We form 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 right to left, and ‘-1’ if the crossing is from left to right. Thus, for example, the word for γ in Figure 2(c) will be “ $b_3 r_3 r_3^{-1} r_2 b_2^{-1}$ ”. We can *reduce* this word by canceling the same letters that appear consecutively but with opposite superscript signs. Thus, the word for γ in Figure 2(c) can be reduced to “ $b_3 r_2 b_2^{-1}$ ”. 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*). Unlike the *homology invariant*, this is not a vector, but an element of the *non-abelian* group *freely generated* (Scott and Scott, 1964; Hatcher, 2001) by $\{r_1, r_2, \dots, r_r, b_1, b_2, \dots, b_b\}$. Thus, although *words* cannot be added in the sense of vectors, they can be concatenated under the non-commutative *group operation*, ‘ \star ’. Also, the inverse of a word, w , written as 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 $w \star w^{-1} = “$, the identity element). Thus, $(w_1 \star w_2)^{-1} = w_2^{-1} \star w_1^{-1}$. As an example, $(“b_3 r_2 b_2^{-1}”)^{-1} = “b_2 r_2^{-1} b_3^{-1}”$.

However, if the curve is a closed loop (e.g. $(C' \sqcup \partial W_1)$ in Figure 2(c)), there is no preferred starting point from where we should start tracing the curve and write the word. Thus, for such curves we need to consider the *cyclic permutations* of the letters in the reduced words to be equivalent. That is, a word, “*abcde*” will be considered to be the same as “*cdeab*”. Thus, when reducing a word, we need to consider the cyclic permutations, and thus cancel a letter at the beginning of the word that appears at the end as well, but with opposite superscript signs. For example, in Figure 2(c), if we trace the curve, $C' \sqcup \partial W_1$, starting at the point e , we obtain

$$\begin{aligned} h(\partial W_1 \sqcup C') &= h(\partial W_1) \star h(C') \\ &= “b_1^{-1} b_2^{-1} r_2^{-1} r_1^{-1} r_3^{-1} b_3^{-1}” \star “r_3 r_2 b_1^{-1} b_2^{-1} r_2^{-1} r_1 r_2 b_2 b_1” \\ &= “b_1^{-1} b_2^{-1} r_2^{-1} r_1^{-1} r_3^{-1} r_3 r_2 b_1^{-1} b_2^{-1} r_2^{-1} r_1 r_2 b_2 b_1” \\ &= “r_3^{-1} b_3^{-1} r_3 r_2 b_1^{-1} b_2^{-1} r_2^{-1}” \\ &\quad (\text{after canceling the letters at the start and the end}) \end{aligned}$$

which is the completely reduced word.

The *homotopy invariant* of a curve, γ , is the *reduced word* constructed in the described way, with cyclic permutations of a word being considered equivalent when γ is closed. In Appendix B we present a more technical discussion, only short of a formal proof, on the justification behind such words being homotopy invariants of curves.

In Section 4 we will describe how we incorporate this homotopy information inside an *h-augmented* graph and

use an optimal graph search algorithm (such as A*) in it to find optimal trajectories in different homotopy classes in order of their path costs, without knowing the *words* for the classes to start with.

It is easy to note that for closed curves, the value of the *homology invariant* described earlier as integral over the bump 1-forms, does not depend on the choice of the direction of the rays emanating from ζ_i . But the *homotopy invariant* word is highly dependent on the choice of the direction of the rays.

3.2.1. The Hurewicz map. The characteristic distinction between *homotopy* and *homology* is that the first homotopy group, generally being non-abelian (non-commutative), yields a classification with a finer resolution, while the homology group, being always an abelian (commutative) group, yields a coarser classification of trajectories. Thus, while trajectories that are homotopic are also homologous, the converse is not necessarily true (see Figure 1 of Bhattacharya et al. (2012)).

The Hurewicz map (Hatcher, 2001) can be used to compute the *homology invariant* from a given *homotopy invariant* of a closed curve. We write h_* to denote this map from the space of *h-signatures* (words) to the space of *H-signatures* (vectors), and it is essentially the *abelianization* map.

Thus, to compute the *H-signature* from a given *h-signature*, we simply let the letters in the word commute. Thus, from the earlier example of Figure 2(c), we had $h(\partial W_1 \sqcup C') = “r_3^{-1} b_3^{-1} r_3 r_2 b_1^{-1} b_2^{-1} r_2^{-1}”$. Letting the letters commute we have the word “ $r_1^0 r_2^0 r_3^0 b_1^{-1} b_2^{-1} b_3^{-1}$ ” (with 0 superscript indicating that the letter is absent). Since the 6 components of the *H-signature* vector correspond to the objects R_1, R_2, R_3, B_1, B_2 and B_3 , respectively, we thus have $H(\partial W_1 \sqcup C') = h_*(“r_3^{-1} b_3^{-1} r_3 r_2 b_1^{-1} b_2^{-1} r_2^{-1}”) = [0, 0, 0, -1, -1, -1]^T$.

3.3. Propositions on object separation and cable manipulation

Proposition 1. Suppose that C is an embedded cable configuration such that $C(0), C(1) \in \partial W$ (i.e. the cable ends lie on the boundary of the environment). Say the end points of C splits ∂W into two parts: ∂W_1 and ∂W_2 (which themselves are curves in $(W - \mathcal{O})$). We assign orientation to ∂W_1 and ∂W_2 such that $C \sqcup \partial W_1$ and $C \sqcup \partial W_2$ are closed loops (Figure 1(b)). Then, C separates the two types of objects (i.e. it is a separating configuration) iff one of the following holds for the vector $H(C \sqcup \partial W_1)$:

- (i) the first r components are all 1 or all -1, and the last b components are all 0;
- (ii) the last b components are all 1 or all -1, and the first r components are all 0.

Note that from the definition of *H-signature*, $H(C \sqcup \partial W_1) = H(C) + H(\partial W_1)$. Also, in these conditions the choice of ∂W_1 over ∂W_2 is made without loss of

generality. The conditions could have been stated in terms of ∂W_2 as well.

Sketch of Proof. 1. The proof follows from the very definition of homology (see Figure 2(a)). First we note that $C \sqcup \partial W_1$ is a Jordan curve (Gamelin, 2001) inside W (since C is embedded). Hence, there is a simply connected region in W (not considering the objects) enclosed by $C \sqcup \partial W_1$. The objects (and their representative points) that this region will contain will manifest as a ± 1 in the corresponding components of the vector $H(C \sqcup \partial W_1)$. Since $C \sqcup \partial W_1$ is Jordan, it will wind around each of the enclosed points in the same direction (all clockwise or all anti-clockwise), thus making the corresponding components of the vectors either all $+1$ or all -1 . All the other components will be 0. The statement of the proposition simply states that the enclosed representative points will be ones corresponding to the objects or the objects, while the ones not enclosed will be ones corresponding to objects of the other color. ■

At this point it is instructive to illustrate why, in the above proposition, we used the homology invariant instead of homotopy invariant. Consider the curve C' in Figure 2(c), which clearly separates the objects from the ones. However, we previously saw that the *reduced word* for $(C' \sqcup \partial W_1)$ is $h(C') \star h(\partial W_1) = "r_3^{-1} b_3^{-1} r_3 r_2 b_1^{-1} b_2^{-1} r_2^{-1}"$. Likewise the *reduced word* $h(C') \star h(\partial W_2) = "r_3 r_2 b_1^{-1} b_2^{-1} r_2^{-1} r_1 r_2 b_2 b_1"$. Neither of these words are helpful in identifying the fact that C' separates the objects from the ones. However, $H(C') + H(\partial W_1) = [0, 0, 0, -1, -1, -1]^T$, and $H(C') + H(\partial W_2) = [1, 1, 1, 0, 0, 0]^T$ both satisfying the condition of Proposition 1 (note that the first 3 components of the vector correspond to R_1, R_2 & R_3 , while the last 3 correspond to B_1, B_2 & B_3), thus indicating that C' indeed separates the objects from the objects.

Proposition 2 (Refer to Figure 1(c)). *Let C be a starting cable configuration (which has an orientation from robot "2" to robot "1", as shown in Figure 1(a)) and C' be a final cable configuration (which may or may not be a separating configuration). Then the trajectories τ_1 and τ_2 for the two robots carry the cable from initial configuration to the separating configuration (up to homotopy) if and only if the closed loop $(C \sqcup \tau_1 \sqcup -C' \sqcup -\tau_2)$ is null homotopic (Hatcher, 2001), i.e. $h(C \sqcup \tau_1 \sqcup -C' \sqcup -\tau_2) = h(C) \star h(\tau_1) \star h(C')^{-1} \star h(\tau_2)^{-1} = "$, is the empty word (identity element).*

Sketch of Proof. We note that unlike in Proposition 1 we do not have the luxury of assuming that $(C \sqcup \tau_1 \sqcup -C' \sqcup -\tau_2)$ will be Jordan (see, for example, Figure 1(c)). First, suppose that trajectories τ_1 and τ_2 carry the cable from configuration C to final configuration C' . We choose two arbitrary points, p_1 and p_2 , on the trajectories τ_1 and τ_2 , respectively, as shown in Figure 4. Next consider the sequence of cable configurations from C to C' as the robots carry it. We can then construct a continuous function (a homotopy), $\bar{C} : [0, 1] \times [0, 1] \rightarrow (W - \mathcal{O})$, such that $\bar{C}(0, \cdot) \equiv C(\cdot)$ and $\bar{C}(1, \cdot) \equiv C'(\cdot)$, and $\bar{C}(t)$ is a general intermediate cable configuration. Such a curve, $\bar{C}(t)$, has

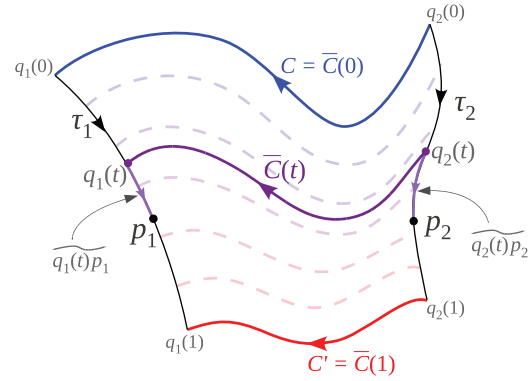


Fig. 4. Illustration for the Proof of Proposition 2.

its end points $q_1(t) \in \tau_1$ and $q_2(t) \in \tau_2$ (Figure 4). We consider the curve connecting $q_1(t)$ to p_1 and lying on τ_1 (call it $q_1(t)p_1$), and the one connecting $q_2(t)$ to p_2 and lying on τ_2 (call it $q_2(t)p_2$). Thus, the sequence of curves, $D(t) := \left(-(q_2(t)p_2) \sqcup \bar{C}(t) \sqcup (q_1(t)p_1) \right)$, defines a homotopy between curves connecting p_1 and p_2 . Thus, $D(0) \sqcup -D(1)$ is null-homotopic. That is, $\left(-(q_2(0)p_2) \sqcup \bar{C}(0) \sqcup (q_1(0)p_1) \right) \sqcup - \left(-(q_2(1)p_2) \sqcup \bar{C}(1) \sqcup (q_1(1)p_1) \right) \equiv (C \sqcup \tau_1 \sqcup -C' \sqcup -\tau_2)$, is null-homotopic.

Conversely, if $(C \sqcup \tau_1 \sqcup -C' \sqcup -\tau_2)$ is null-homotopic, one can construct a homotopy, D , as before, and hence construct a sequence of curves \bar{C} , that takes the cable from C to C' .

4. Implementation

For simplicity, we assume that the environment, W , is a rectangular region, and all of the rays, $r_j, j = 1, 2, \dots, r$ and $b_j, j = 1, 2, \dots, b$, are parallel, pointing along the positive Y axis. Furthermore, we restrict the final goals of the robots to the left and right boundaries of the environment (∂W_l at $x = x_l$ and ∂W_r at $x = x_r$, respectively), but they need to reach the opposite edges. Thus, a part of the boundary, ∂W_2 , will never intersect any of the rays (Figure 5(a)), and hence $H(\partial W_2) = [0, 0, \dots, 0]^T$ and $h(\partial W_2) = "$. This simplifies the computation of $H(C' \sqcup \partial W_2)$ for Proposition 1 to the computation of $H(C')$.

We use a discrete representation of the environment, and construct a graph, \mathcal{G} , by placing a vertex in every discrete cell and by establishing an edge between the vertices of adjacent cells. From such a graph we can construct an H -augmented graph, \mathcal{G}_H (for keeping track of the homology invariants), or an h -augmented graph, \mathcal{G}_h (for keeping track of the homotopy invariants), as described by Bhattacharya et al. (2012).

While the graph, \mathcal{G} , itself can be quite arbitrary, for simplicity we used a uniform 8-connected discrete representation (see Figure 5(b)) of the environment for all our simulations and experiments.

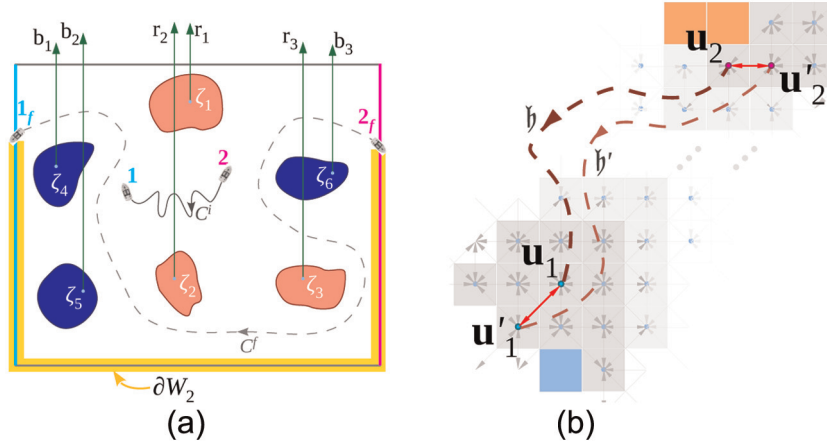


Fig. 5. The environment and its discretization. (a) The rectangular environment with the goal of the robots being the left (cyan line) and right (magenta line) boundaries. ∂W_2 does not intersect any of the rays (all of which point in the positive Y direction). (b) Robots 1 and 2 navigating on copies of graph, \mathcal{G} , formed by uniform discretization of configuration space. Change in the h -signature of the cable due to transitions within the graph is also illustrated.

4.1. Planning in joint state-space

The problem under consideration is to plan optimal trajectories that would take a given initial cable configuration, C^i , to a separating cable configuration, and the robot 1 reaches the left (or right) edge of W , while robot 2 reaches the right (or left) edge. In this section we describe an algorithm that searches for the optimal solutions in the joint state space of the two robots and the homotopy class of the cable. A graph, $\mathcal{J} = \mathcal{G} \times \mathcal{G}$, is defined as the *graph Cartesian product* of two copies of \mathcal{G} . Thus, for every pair of vertices, $\mathbf{u}_1, \mathbf{u}_2 \in V(\mathcal{G})$, a vertex in $V(\mathcal{J})$ is of the form $(\mathbf{u}_1, \mathbf{u}_2)$, and for every pair of edges of the form $[\mathbf{u}_1 \rightsquigarrow \mathbf{u}'_1], [\mathbf{u}_2 \rightsquigarrow \mathbf{u}'_2] \in E(\mathcal{G})$ (“ \rightsquigarrow ” indicating the direction of an edge), there exists an edge $[(\mathbf{u}_1, \mathbf{u}_2) \rightsquigarrow (\mathbf{u}'_1, \mathbf{u}'_2)] \in E(\mathcal{J})$. We are given an initial vertex in the joint state space, $(\mathbf{u}'_1, \mathbf{u}'_2)$, the initial positions of the robots, and an initial configuration of the cable (up to homotopy) in form of the h -signature of the cable, h_i (which, as defined earlier, is a *reduced word*).

In order to incorporate the information about the homotopy class of the cable in the search problem, we define an augmented graph, \mathcal{J}_h , such that a vertex in this graph contains the additional information of the h -signature of the cable that is being carried by the robots. This, in essence, is similar to the H -augmented graph construction detailed in Bhattacharya et al. (2012). The explicit construction is as follows: the initial vertex in \mathcal{J}_h is $\mathbf{v}_i = (\mathbf{u}'_1, \mathbf{u}'_2, h^i)$, which contain the information about the initial positions of the robots and the h -signature of the initial cable configuration, $h^i = h(C^i)$. A transition of the robots from $(\mathbf{u}_1, \mathbf{u}_2, h)$ to $(\mathbf{u}'_1, \mathbf{u}'_2, h')$ will mean (due to Proposition 2) that the h -signature of the resultant cable configuration is equal to $h' = h(-\tau_2) \star h \star h(\tau_1)$ (recall, “ \star ” is concatenation, followed by reduction), where τ_1 and τ_2 are trajectories taken

by the robots for the transition (see Figure 5(b)). Thus, for each edge of the form $[(\mathbf{u}_1, \mathbf{u}_2) \rightsquigarrow (\mathbf{u}'_1, \mathbf{u}'_2)] \in E(\mathcal{J})$, emanating from $[(\mathbf{u}_1, \mathbf{u}_2)] \in V(\mathcal{J})$, the vertex $(\mathbf{u}_1, \mathbf{u}_2, h) \in V(\mathcal{J}_h)$ is connected to neighbors $(\mathbf{u}'_1, \mathbf{u}'_2, h(\mathbf{u}'_2 \mathbf{u}_2) \star h \star h(\mathbf{u}_1 \mathbf{u}'_1)) \in V(\mathcal{J}_h)$ (where $[\mathbf{a} \rightsquigarrow \mathbf{b}]$ is used to indicate an edge in edge set, $E(\mathcal{G})$, from vertex \mathbf{a} to \mathbf{b} , and $\overleftarrow{\mathbf{ab}}$ is the curve/line segment that constitutes the edge. $\overrightarrow{\mathbf{ba}}$ is the same curve but with opposite orientation). This gives us the recipe to construct \mathcal{J}_h incrementally starting at $(\mathbf{u}'_1, \mathbf{u}'_2, h^i)$.

We choose the optimization objective to be the sum of the lengths of the robot trajectories since trajectory lengths are directly related to the energy required to complete the task as well as the time required to execute the trajectories. Thus, the cost of the edge $[(\mathbf{u}_1, \mathbf{u}_2, h) \rightsquigarrow (\mathbf{u}'_1, \mathbf{u}'_2, h(\mathbf{u}'_2 \mathbf{u}_2) \star h \star h(\mathbf{u}_1 \mathbf{u}'_1))] \in E(\mathcal{J}_h)$ is chosen to be the sum of the lengths of the edges $[\mathbf{u}_1 \rightsquigarrow \mathbf{u}'_1]$ and $[\mathbf{u}_2 \rightsquigarrow \mathbf{u}'_2]$ in $E(\mathcal{G})$. For this cost and with the left and right boundaries as goal, an *admissible heuristic function* is $f(\mathbf{u}_1, \mathbf{u}_2, h) = \min((u_{1,x} - x_l) + (x_r - u_{2,x}), (u_{2,x} - x_l) + (x_r - u_{1,x}))$, which is a lower bound on the cost to reach a goal from $(\mathbf{u}_1, \mathbf{u}_2, h)$ (where, $u_{j,x}$ is the X coordinate of a vertex \mathbf{u}_j and $u_{j,y}$ is its Y coordinate).

Starting at the initial vertex, $(\mathbf{u}'_1, \mathbf{u}'_2, h^i)$, we thus keep *expanding* the vertices in the graph, \mathcal{J}_h , using a search algorithm (we use Dijkstra’s (Dijkstra, 1959) or A* (Hart et al., 1968) since they are complete, optimal and deterministic) until we reach a goal vertex. A vertex $(\mathbf{u}_1, \mathbf{u}_2, h)$ is deemed as a goal if $\mathbf{u}_1 \in \partial W_l$ and $\mathbf{u}_2 \in \partial W_r$ (or vice versa), and if $h_*(h) + H(\partial W_2) (= h_*(h))$ satisfies the condition of Proposition 1 (i.e. it is a separating cable configuration).

Planning in the joint state space gives the flexibility of easily incorporating additional constraints (such as inter-robot collision avoidance, communication constraints, etc.) as well as using more complex cost functions.

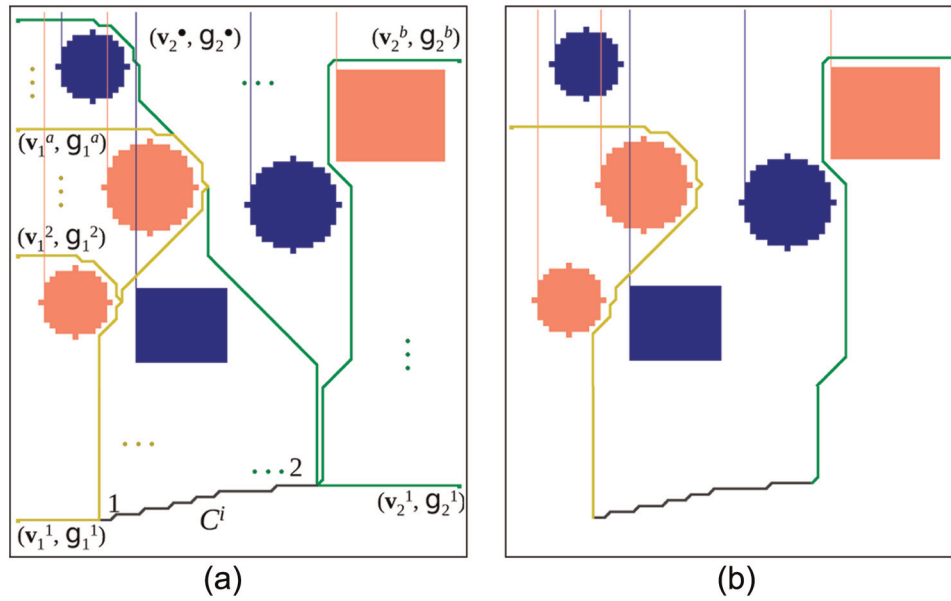


Fig. 6. Decoupled and distributed planning: Optimal paths with different h -signatures found for the two robots in parallel threads, and costs of compatible pairs are compared to find the optimal compatible pair. (a) Optimal paths in different homotopy classes corresponding to goal vertices $(v_j^1, g_j^1), (v_j^2, g_j^2), \dots \in \mathcal{G}_{h,j}$ are found in parallel threads for robot $j = 1$ (yellow) and robot $j = 2$ (green). (b) A compatible combination of paths (with end vertices (v_1^a, g_1^a) and (v_2^b, g_2^b)) is such that the end points, v_1^a & v_2^a , lie on the opposite edges of W , and $h_*(g_2^b)^{-1} \star h(C^i) \star g_1^a$ satisfies the condition of Proposition 1.

4.2. Decoupled planning: a distributed approach

While the approach of planning in joint state-space is complete and optimal, it suffers from the obvious drawback of being slow and inefficient since the graph, \mathcal{J} , is very large and is of high degree, being a discrete representation of a four-dimensional space. However, it is possible to decouple the searches for the two robots in two copies of \mathcal{G}_h (the h -augmented graph of \mathcal{G} , described next), each of which are discrete representation of two-dimensional spaces, and run those searches in parallel (in parallel threads in our C++ implementation). Consequently, collecting the solutions obtained from each parallel process as they progress, and checking whether they together give a valid solution for attaining a separating configuration, we are able to conclude when the optimal solution is found, and thus halt the threads to conclude the search process.

The h -augmented graph, \mathcal{G}_h , is very similar to the concept of the H -signature augmented graph, \mathcal{G}_H described by Bhattacharya et al. (2012), only with the homology invariants being replaced by the homotopy invariants. Corresponding to a given $\mathbf{u} \in V(\mathcal{G})$, there exists a discrete set of the augmented states, $(\mathbf{u}, \mathbf{h}) \in V(\mathcal{G}_h)$, for each homotopy class of trajectories (with h -signature \mathbf{h}) from an initial vertex, \mathbf{u}^i , to the vertex \mathbf{u} . Clearly $(\mathbf{u}_j^i, "") \in V(\mathcal{G}_h)$ is the start vertex corresponding to the trajectory of zero length. Edges emanating from (\mathbf{u}, \mathbf{h}) are thus of the form $[(\mathbf{u}, \mathbf{h}) \rightsquigarrow (\mathbf{u}', \mathbf{h} + h(\mathbf{u}\mathbf{u}'))] \in E(\mathcal{G}_h)$, corresponding to every $[\mathbf{u} \rightsquigarrow \mathbf{u}'] \in E(\mathcal{G})$. The cost of such an edge is chosen to be the Euclidean length of $\mathbf{u}\mathbf{u}'$. An *admissible heuristic function* for this choice of cost, and with goal as $\partial W_l \cup \partial W_r$, is $f(\mathbf{u}, \mathbf{h}) = \min(u_x - x_l, x_r - u_x)$.

Thus, we start with two copies of the augmented graph, $\mathcal{G}_{h,1}$ and $\mathcal{G}_{h,2}$, in two parallel threads (that branch off from a main thread), for robots 1 and 2. In robot j 's copy of the graph, we start expanding the vertices from (i.e. initiate the *open list* with) the vertex $(\mathbf{u}_j^i, "") \in \mathcal{G}_{h,j}$, $j = 1, 2$. We keep expanding the vertices in the respective graphs, and keep storing a path every time ∂W_l or ∂W_r is reached via a new homotopy class for the robot (i.e. if (\mathbf{v}, \mathbf{g}) is expanded, with $\mathbf{v} \in \partial W_l \cup \partial W_r$, then the vertex is bookmarked as a potential desired goal for robot j if the homotopy class \mathbf{g} is not same for any of the previously bookmarked vertices for the robot). It is important to note that for each of the robots such optimal paths with different h -signatures are found in the order of their costs since we use an optimal search algorithm (Dijkstra's/A* (Hart et al., 1968)).

Suppose for robot " j " such potential goal vertices are $\{(v_j^1, g_j^1), (v_j^2, g_j^2), (v_j^3, g_j^3), \dots\}$ with costs of the respective optimal paths $c_j^1 \leq c_j^2 \leq c_j^3 \leq \dots$, for $j = 1, 2$ (Figure 6(a)). In order to find the optimal combination of paths, we define a *partial order* (Stanley and Rota, 2000), \preceq , on \mathbb{R}^2 , to compare the cost of pairs of paths, (c_1^m, c_2^n) . One obvious choice is to compare the sum of the path costs: $(c_1^m, c_2^n) \preceq (c_1^\mu, c_2^\nu) \Leftrightarrow c_1^m + c_2^n \leq c_1^\mu + c_2^\nu$. However, the issue with this choice is that a lower joint cost may be highly asymmetric, while one would desire that the task of carrying the cable is evenly distributed among the two robots, and not one of the robots end up traveling most of the distance while the other travels very little. For this, we choose to minimize the maximum of the costs of the two trajectories instead of their sum. Thus, we define the partial order to be

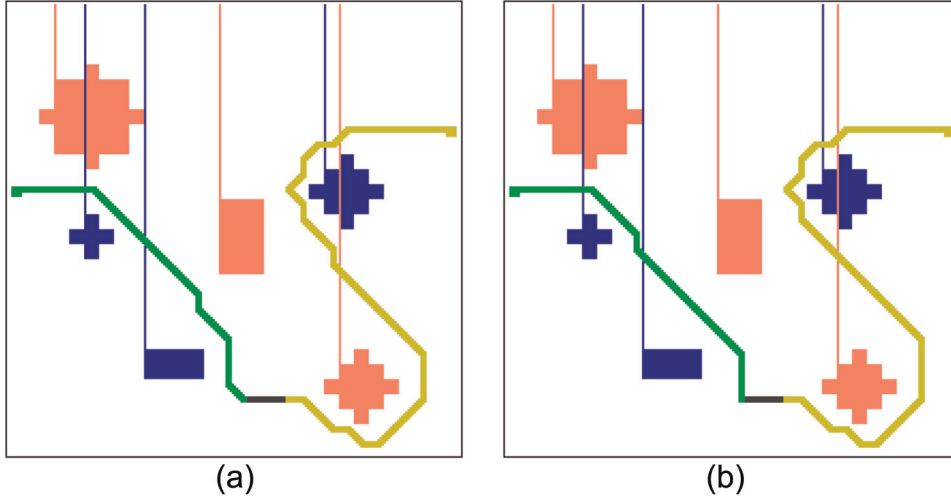


Fig. 7. A simple 30×30 environment with $r = b = 3$. The green and yellow are the trajectories of the robots. The rays emanating from ζ_j are also shown. The dark grey segment indicates the initial cable configuration. (a) Planning in the joint state space took 4250 s. The sum of the costs of two trajectories is 65.598 discretization units. (b) The distributed decoupled planning gives result with the same optimal cost, but takes about 2 s to run.

$$(c_1^m, c_2^n) \preccurlyeq (c_1^\mu, c_2^\nu) \Leftrightarrow \text{Either } \max(c_1^m, c_2^n) < \max(c_1^\mu, c_2^\nu), \text{ or,} \\ (\max(c_1^m, c_2^n) = \max(c_1^\mu, c_2^\nu) \\ \text{and } \min(c_1^m, c_2^n) \leq \min(c_1^\mu, c_2^\nu))$$

which we call the *sorted lexicographic order*.

Thus, as the main thread of the program receives the two sequences of optimal paths to the left/right boundaries with different h -signatures from the two different threads, it keeps checking them in pairs. A pair of potential goal vertices, (v_1^m, g_1^m) and (v_2^n, g_2^n) , is deemed “compatible” (Figure 6(b)) if the corresponding final cable configuration is a separating configuration. Now, recall that due to Proposition 2, the h -signature of the final cable configuration is equal to $(g_2^n)^{-1} \star h(C^i) \star g_1^m$. Thus, for this to be a separating configuration, due to Proposition 1, the required condition is that $h_*((g_2^n)^{-1} \star h(C^i) \star g_1^m) + H(\partial W_2)$ be a vector with first r components ± 1 and rest zeros, or last b components ± 1 and rest zeros. We keep record of the most optimal of such compatible pairs (i.e. the one with lowest (c_1^a, c_2^b) , where comparisons are made using “ \preccurlyeq ”).

Say at an instant the most optimal pair has cost (c_1^*, c_2^*) . Since the optimal paths with different h -signatures are found in order of their costs, if robot j finds a path such that its cost is greater than current value of $\max(c_1^*, c_2^*)$ (or, if we were using the sum of the pairs in defining the partial order, then $c_1^* + c_2^*$), we can say for sure that none of the paths to be discovered for robot j after that point can be part of a more optimal pair. Hence, we stop the search for robot j . When the searches for both the robots end, it can be concluded that the current optimal pair is the global optimal one. We also do check for and eliminate robot trajectories that loop around objects either by checking explicit self-intersection or checking whether the h -signature contains the same letter consecutively more than once. A

similar check is also done on the h -signature of the final cable configuration to prevent wrapping around objects.

As a side-note, we would like to point out that a distributed implementation over multiple threads, as proposed in this section, can be used in more generic setups. Multiple problems coupled by constraints can be decoupled and solved in parallel threads, while performing cross-validation for constraint satisfaction in a central main thread, until a solution satisfying the coupling constraints is found. An interested reader may refer to Samar et al. (2007) and Bhattacharya et al. (2010) for similar approaches.

4.3. Some notes on complexity

The complexity of the graph searches involved is the usual for an uniform degree graph: $O(V_e \log(V_e))$, where V_e is the number of vertices expanded during the search. It is to be noted that we do not construct the graph \mathcal{G}_h from before, and that graph itself is infinite because of the infinitely many possible homotopy classes. However, in our search algorithm we generate the vertices and edges of the graphs on the fly, starting at the start vertices. The number of vertices that need to be generated or expanded before a separating configuration is found will depend largely on the configuration of the and the objects, and is difficult to estimate for the most general setup. However, in particular, there may be simple configurations of the objects such that the very first pair of robot trajectories (obtained from the parallel threads in the decoupled planning approach) may be a separating configuration. However, the worst-case complexity will in fact be exponential in the number of objects present in the environment because of the combinatorial nature of the homotopy classes in the environment.

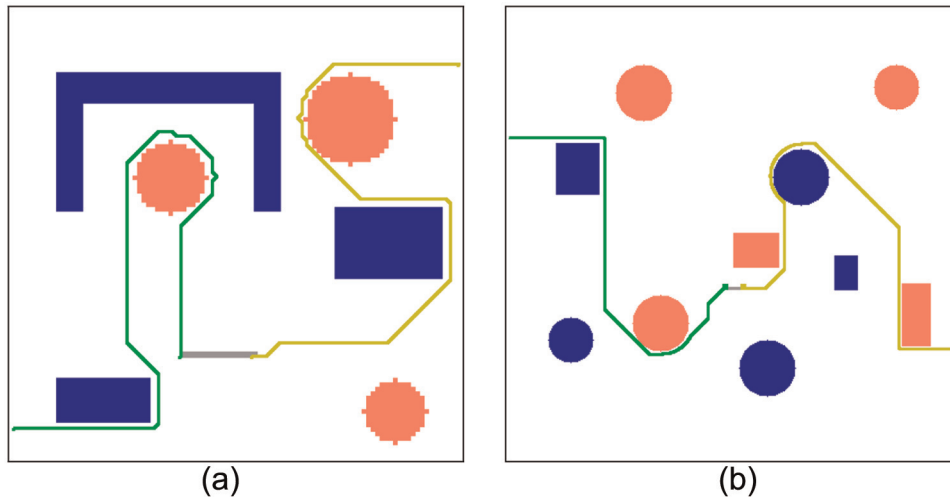


Fig. 8. Decoupled, distributed plans. Initial cable is shown in grey/black. Optimal planned trajectories are in green and yellow. (a) The planned trajectories in a 100×100 discretized environment. (b) The planned trajectories in a 400×400 discretized environment.

5. Results

We implemented the planning in the joint state-space as well as the decoupled planning in C++ programming language with ROS integration, and used the A* algorithm to search in the respective h -augmented graphs. All computations were performed on a system with dual-core processor with clock speed 2.6 MHz and 4 GB memory. Throughout this paper we consider an uniform discretization of the environment for simplicity. However, the techniques developed in this paper is not restricted to any specific discretization scheme or even a specific search algorithm. A more detailed discussion on the generality of the technique can be found in Bhattacharya et al. (2012).

5.1. Joint state space plan

The search in the discrete representation of the four-dimensional joint state space, \mathcal{J}_h , is prohibitively expensive for large environments. Figure 7(a) shows the result in a simple environment, 30×30 discretized, and with 3 objects of each type. The search took about 4250 s and expanded 1,484,999 vertices in \mathcal{J}_h . Figure 7(b) shows the result obtained for same problem, but using the decoupled planning (and using sum of the cost of the trajectories for defining the partial order, \preceq , for being consistent). The result has the same optimal cost as the joint state space planning, but took less than 1 s with 19,144 and 19,593 vertices being expanded in $\mathcal{G}_{h,1}$ and $\mathcal{G}_{h,2}$. All of the objects were inflated to avoid collision.

5.2. Decoupled planning

In this section we present results obtained using the decoupled, distributed implementation. The sorted lexicographic order was used for the partial order, “ \preceq ”.

Figure 8(a) show the plans obtained for two robots in a 100×100 discretized environment. The planning took about 1.3 s, and expanded 39,764 and 40,066 vertices in the graphs of the two robots. Figure 8(b) shows the result in a much larger (400×400 discretized) environment. The planning time for this case was 490 s, with 1,086,182 and 1,079,670 vertices being expanded.

Finally, we present a result along with a simple dynamic simulation created using the open-source two-dimensional physics engine, *Box2D* (Catto, 2011), illustrating the process of separating the two types of objects. The environment in Figure 9(a) is 150×200 discretized. The objects are static obstacles, and the ones are those that need to be separated and transported. The ends of the cables were modeled as point robots and controlled using a simple PD controller to follow the planned trajectories. The objects were given very high inertia so that their drifts are minimal when the robots are navigating to attain the separating configuration and the cable touches the objects. Figures 9(b)–(f) shows snapshots from the dynamic simulation (also see Extension 1). It is only near the end of attaining the separating configuration that enough *pull* is applied by the roots to transport the objects.

5.3. Fast re-planning

So far we have planned the trajectories with the assumptions that the object remain stationary as the robots follow the planned trajectories. This is a fair practice even when the objects move or drift moderately from their original positions: the homotopy classes remain indifferent to small perturbations in the positions of the objects. This, in the first place, was one of the advantages behind using a topological approach: topology is robust and invariant to small perturbations in the metric information.

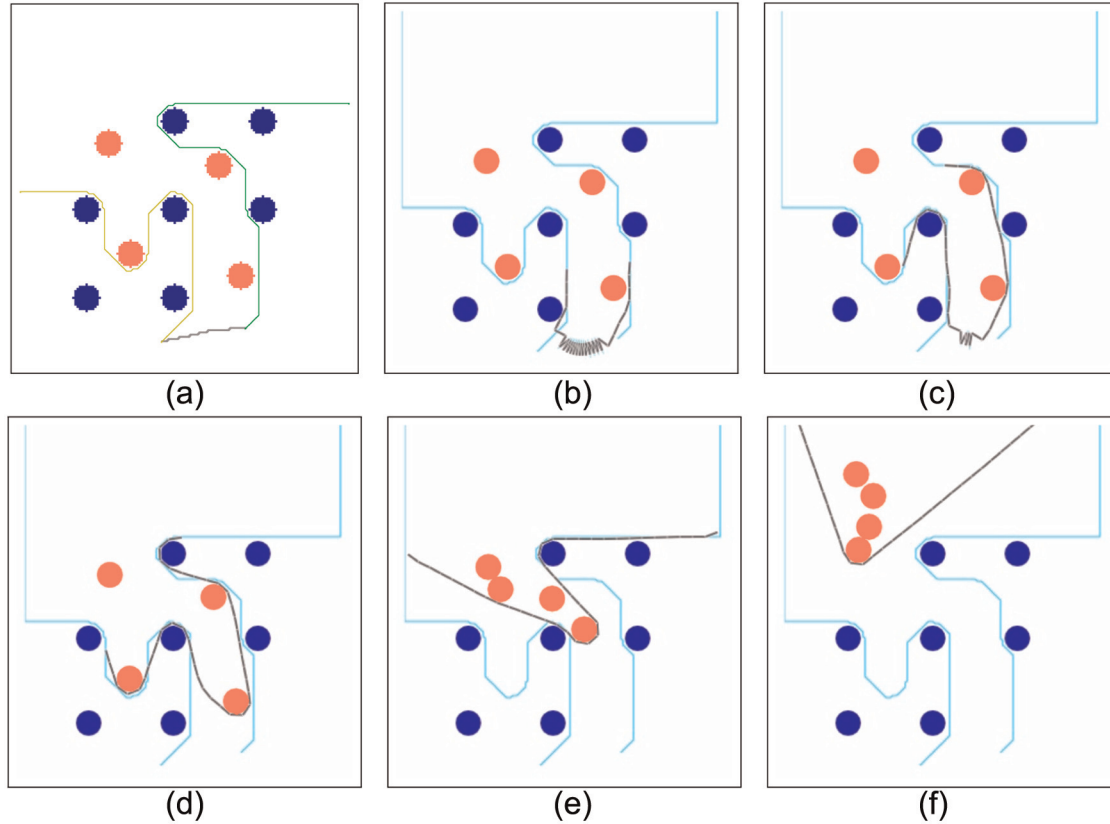


Fig. 9. Separation and transportation of objects in presence of obstacles (also see Extension 1). (a) Planned optimal trajectories. The environment with obstacles (blue) and objects (red) to be transported. (b)–(f) Dynamic simulation illustrating the process of separation and transportation: (b) $t = 5$ s; (c) $t = 11$ s; (d) $t = 19$ s; (e) $t = 55$ s; (f) $t = 97$ s. Grey curve is the cable, while the planned trajectories are shown in cyan.

However, a re-planning is necessary when the objects move significantly such that the topological classes of the planned trajectories change (e.g. an object drifts and crosses the remaining planned robot trajectory or the rays emanating from the representative points of two objects cross each other). Such motions are possible due to the interactions between the cable and the objects when the objects are free to move. Instead of solving the entire problem every time the environment changes by small amounts, we invoke a re-planning algorithm only when two objects exchange the order of the X coordinates of their representative points (i.e. the rays emanating from ζ_j cross each other) or one of the planned trajectories become invalid (due to an object moving on top of it).

Suppose g_1 and g_2 are the h -signatures for the trajectories lying ahead of the robots (i.e. the part yet to be traversed) just before one of the triggers for re-planning happens. We set $g'_j = g_j$, $j = 1, 2$. If the trigger was caused due to switching of the X coordinates of two representative points, we interchange the positions of the corresponding letters in the words g'_1 and g'_2 wherever they appear side-by-side. We thus re-plan trajectories for the robots in $\mathcal{G}_{h,j}$, $j = 1, 2$ (starting at vertex $(p_j, \text{" "})$, where $p_j \in \mathcal{G}$ is the current vertex position of robot j) and with the constraints that the new trajectories need to have h -signature of g'_j .

Since for these re-plannings we know the desired h -signatures, it is possible to construct a more efficient, yet admissible, heuristic function than before for performing the new searches using A* algorithm: Consider the vertex (u, h) in the h -augmented graph used for the re-planning for robot ' j '. The h -signature of the path to a goal vertex from that vertex is thus $h^{-1} * g'_j$. Suppose the representative points corresponding to the letters appearing in $h^{-1} * g'_j$ are $\zeta_{\sigma_1}, \zeta_{\sigma_2}, \dots, \zeta_{\sigma_m}$. It is obvious that the remaining part of a valid trajectory after (u, h) will have to cross each of the rays emanating from these ζ_{σ_k} in this order (besides possibly crossing others that will cancel out). This allows us to design the following heuristic function as a lower bound of the cost (i.e. admissible) for reaching a goal state:

$$f((u, h)) = \sum_{i=1}^{m-1} |\zeta_{\sigma_{i+1}, x} - \zeta_{\sigma_i, x}| + |\zeta_{\sigma_m, x} - x_g| + \begin{cases} f_8(u, \zeta_{\sigma_1}), & \text{if } u_y < \zeta_{\sigma_1, y}, \\ |u_x - \zeta_{\sigma_1, x}|, & \text{otherwise} \end{cases}$$

where $f_8(u, v) = \sqrt{2} \min(|u_x - v_x|, |u_y - v_y|) + ||u_x - v_x| - |u_y - v_y||$ is an admissible heuristic for a 8-connected graph.

Such re-planning of trajectories will be able to take into account arbitrary motions of the objects, including collision

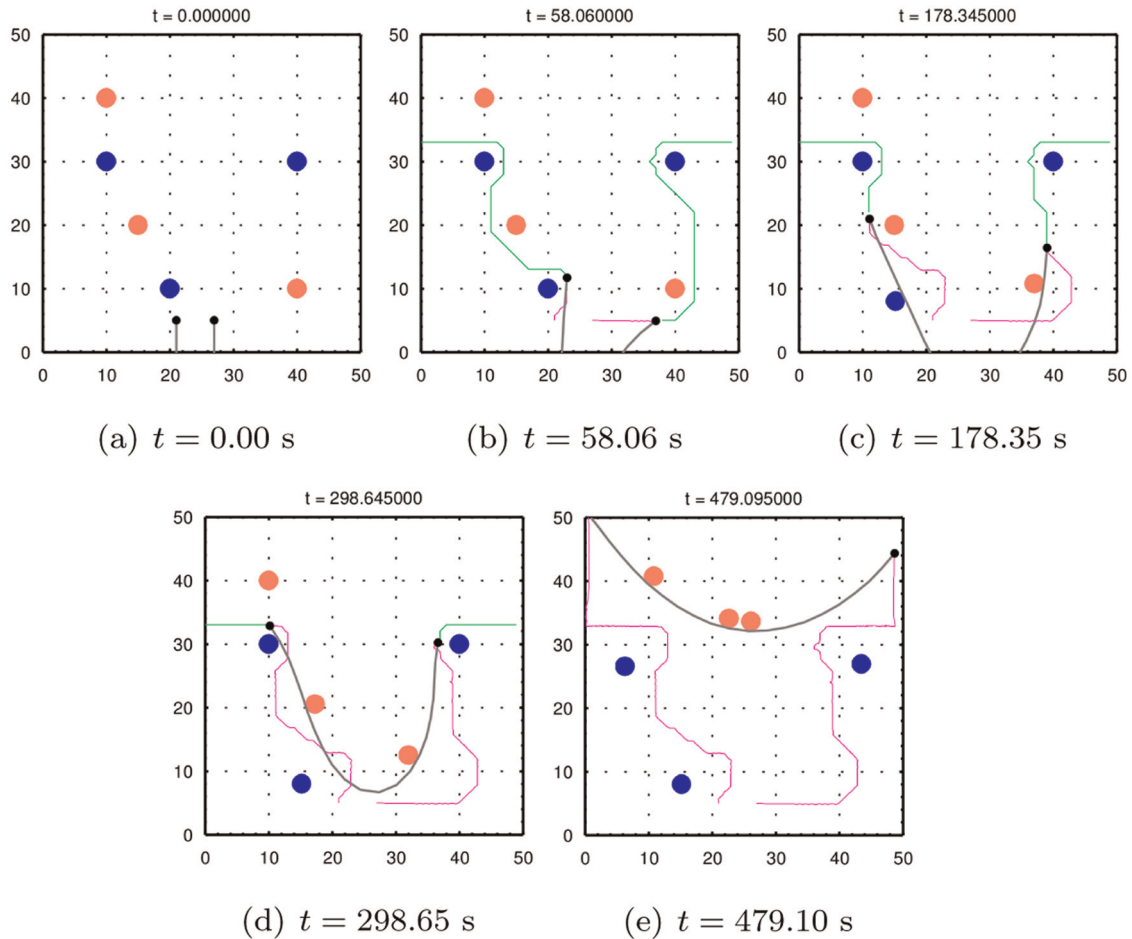


Fig. 10. Dynamic simulation with fast re-planning for separation of objects. The grey curve is the cable, with black dots marking robots at its ends. Green curves are the planned trajectories. Magenta curves are the robot footprints. The and disks are the rigid freely floating objects. Also see Extension 1.

between the objects, as long as a valid separating configuration exists. It is possible that a separating configuration ceases to exist due to objects of different colors drifting and ending up touching each other, and hence re-planning would not find a solution. This situation is however unlikely and would be prevented in the first place due to the re-planning being triggered when one of the objects cross a robot trajectory. However, if such a case indeed arises, we reduce the size of the different-colored touching objects (or the safety padding around them) in performing the re-plan, so that the subsequent interactions between the freely floating objects and the cable/robots *opens up* a solution. We demonstrate the practicality of this approach in solving the physical problem of separating the objects through a dynamic simulation as described next.

5.3.1. Dynamic simulation with re-planning. For the purpose of testing the re-planning algorithm, we built an accurate real-time dynamic simulation platform for the cable (modeled as a serial chain) and freely floating disk-shaped objects on a fluid. Using Lagrangian mechanics we

developed the equations of motion with realistic modeling of drag forces (Bhattacharya et al., 2011), and modeled the contacts using linear complementarity conditions (Kim et al., 2013). We used a simple feedback (PD) controller to make each robot follow the paths generated by the planner as described by Kim et al. (2013).

Figure 10 (and Extension 1) shows the dynamic simulation result on a simple environment with 3 and 3 objects. All of the objects are identical in size and inertia, and can move freely. Figure 10(a) shows the initial configuration of the system. As the objects move and the environment changes significantly, the planned paths of the robot are re-computed (shown by green curves in Figure 10(b)–(e)). The final figure illustrates that we were able to successfully separate the objects from the ones.

5.4. Experimental results

The field experiments were conducted in Puddingstone Lake, San Dimas, CA. Two identical autonomous surface vessels (ASVs) were used, each around 2 m long and 0.8 m wide, capable of speeds up to 1.6 m/s, using two electric

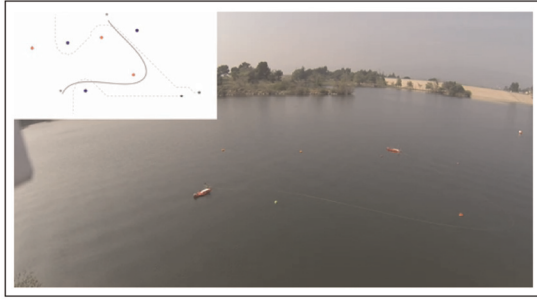


Fig. 11. Experimental setup showing the ASVs.

thrusters and a rudder for control. Both were equipped with a GPS, an IMU with integrated compass and an onboard computer for control. Six buoys (the objects to be separated) were placed in the water, anchored fairly tightly with

weights to the bottom to prevent movement, in an area $25 \text{ m} \times 35 \text{ m}$, and their locations recorded manually using GPS. The displacements of the buoys from their original positions were minimal during the experiment, moving moderately only when they are tugged upon by the rope. The anchoring was necessary because without it the buoys would have drifted very fast due to current and wind and would be stranded on shore before the experiment could be started. For the experiments we assume that the objects remain stationary, and hence the objective of the experiment is to plan and attain a separating configuration for the cable without transporting the objects. An un-actuated cable spool was mounted on each of the ASVs and spooled on them was a floating rope, connected between the ASVs. This was done to reduce the drag that the water exerts on the rope during movement and also to be able to provide sufficient length of rope during the experiment. When the

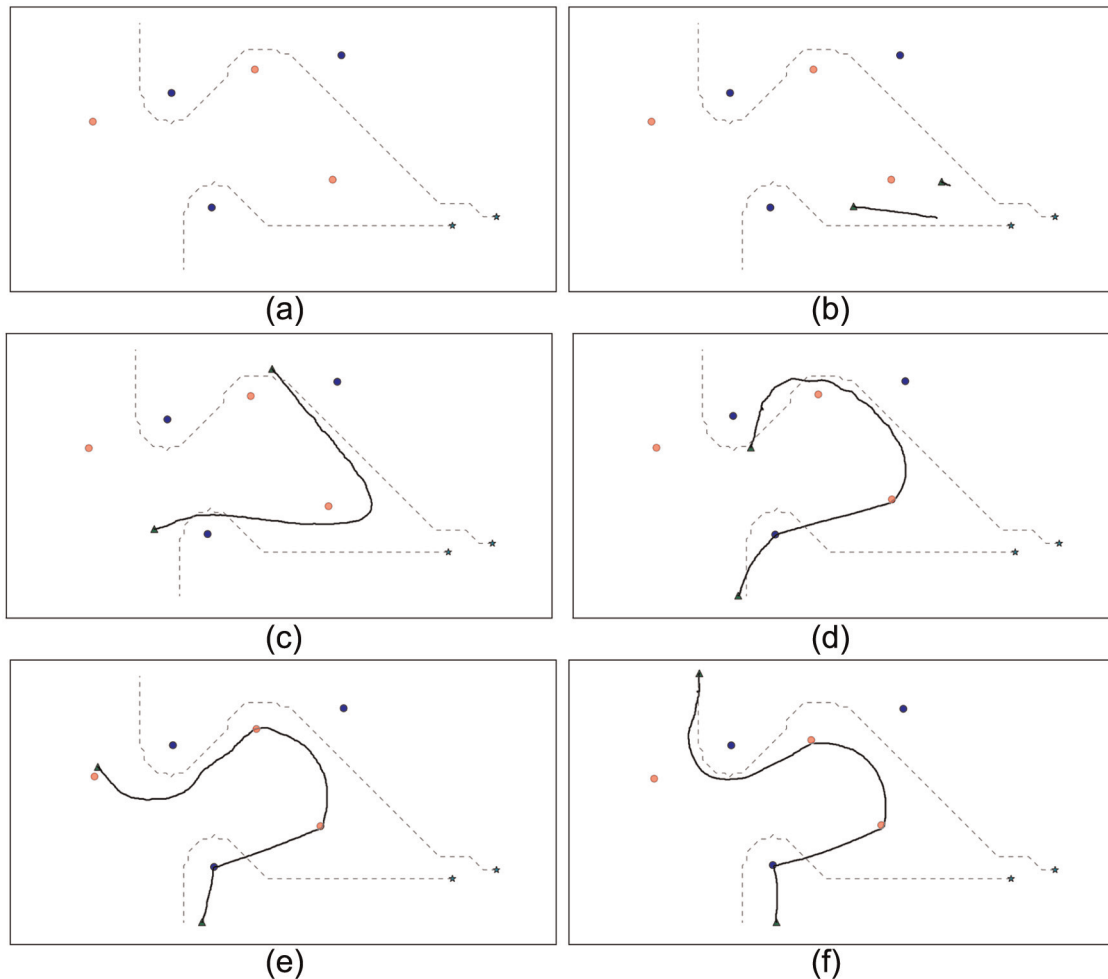


Fig. 12. Snapshots of experimental data at different instants of time during execution: (a) $t = 0 \text{ s}$; (b) $t = 25 \text{ s}$; (c) $t = 50 \text{ s}$; (d) $t = 75 \text{ s}$; (e) $t = 100 \text{ s}$; (f) $t = 118 \text{ s}$. The disks represent the buoys to be separated, the triangles are the ASVs and the stars are their initial positions. Dashed curves are the planned trajectories. Solid curves are the (post-processed) shape of the cable as obtained from the camera data. Mild drifting of the objects (due to tugging by the cable) during experiment can be observed. Deviation of the actual paths followed by the ASVs from the planned paths (dotted curves) was due to the inability of the underactuated ASVs to compensate for the large drag on the floating cable. Also see Extension 1.

ASVs moved, the drag exerted on the rope caused more rope to pull of the spools causing the ASVs to effectively lay down the rope in their paths. In order to show the change in position of the objects as the cable tugged on them, post-processed live positions of the buoys over the course of the experiment were extracted from the overhead camera data. Figure 11 shows the experimental setup.

To record the experiment, particularly the position and shape of the rope, a camera, in an adjustable tilt mount, was mounted on top of a 9 m mast which stood on shore close to the experiment area, overlooking it. The ASVs were moved to a starting position and the planner was run based on the configuration of the targets and ASVs. The resulting plan was then executed on the ASVs.

The camera data processing was done as follows. First, the rope shape and buoy positions were manually extracted from one frame per second. Since the positions were extracted from uncalibrated images, they were then undistorted using parameters from a previous camera calibration that had been performed using a checkerboard. A perspective projection was computed from the initial positions of the buoys and applied to the images to get a top-down view of the rope line with a known scale.

During the experiment, the ASV on the right-hand side of the formation was commanded a path of 62 m while the one on the left was commanded a path of 41 m. The length of the actual traveled paths were 86 m for the ASV on the right and 46 m for the ASV on the left. The underactuated ASVs (with limited propeller motor power) were unable to compensate for the drag on the floating rope and thus were not able to follow the commanded paths exactly. However, the executed paths were homotopic to the commanded ones and the correct separation was achieved. The experiment was halted (the separated set of buoys not dragged away) once the ASVs reached their end positions due the buoys being anchored.

The accompanying video (Extension 1) shows one run of the experiment from the view of the mast-mounted camera. The animation in the top left corner (as well as the screenshots in Figure 12) shows the processed top-down view of the experiment with the assigned colors of the targets, planned path and ASV, buoy and rope movements during the experiment.

6. Conclusions

In this paper we have presented a formal mathematical description of the problem of planning and control for a flexible cable towed by two robots so as to separate two types of objects in a planar environment. We have developed a graph search-based implementation, and distribute the computation for efficiency. We demonstrate the working of the algorithms through simulations, and the practical applicability of the method using a dynamic simulation and experiments with ASVs. More elaborate experimentation with ASVs is within the scope of future research.

Acknowledgements

We would like to thank the reviewers, whose valuable comments and suggestions helped in vastly improving the quality of the paper. Parts of this work have appeared in the proceedings of the ninth Robotics: Science and Systems (RSS), 2013.

Funding

This work was supported by 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

- Aranda J, González de Santos P and Cruz JMdl (2006) *Robotics and Automation in the Maritime Industries*. Producción Gráfica Multimedia (PGM).
- Bhattacharya S (2012) *Topological and Geometric Techniques in Graph-Search Based Robot Planning*. PhD Thesis, University of Pennsylvania. Available at: <http://subhrajit.net/wiki/index.php?n=Projects.Publications&action=bibentry&bibfile=PublicationsBibfile&bibref=PhDthesis:12>.
- Bhattacharya S, Heidarsson H, Sukhatme GS and Kumar V (2011) Cooperative control of autonomous surface vehicles for oil skimming and cleanup. In: *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. Available at: http://www.subhrajit.net/files/Projects-Work/Oil Boom_Catenary_2010/ICRA2011.pdf.
- Bhattacharya S, Kumar V and Likhachev M (2010) Distributed optimization with pairwise constraints and its application to multi-robot path planning. In: *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain.
- Bhattacharya S, Likhachev M and Kumar V (2012) Topological constraints in search-based robot path planning. *Autonomous Robots*. DOI: 10.1007/s10514-012-9304-1.
- Bhattacharya S, Lipsky D, Ghrist R and Kumar V (2013) Invariants for homology classes with application to optimal search and planning problem in robotics. *Annals of Mathematics and Artificial Intelligence (AMAI)* 67(3–4): 251–281.
- Binder D (1981) Approximations to Bayesian clustering rules. *Biometrika* 68: 275–285.
- Bott R and Tu L (1982) *Differential Forms in Algebraic Topology* (Graduate Texts in Mathematics). New York: Springer-Verlag.
- Catto E (2011) Box2D – a 2D physics engine for games. Available at: <http://box2d.org/>.
- Cheng P, Fink J and Kumar V (2009) Cooperative towing with multiple robots. *ASME Transactions: Journal of Mechanisms and Robotics* 1(1): 1–8.
- Cormen TH, Leiserson CE, Rivest RL and Stein C (2001) *Introduction to Algorithms*, 2nd edn. Cambridge, MA: MIT Press.
- Crowell RH (1959) On the van Kampen theorem. *Pacific Journal of Mathematics* 9(1): 43–50.
- Derenick J, Speranzon A and Ghrist R (2013) Homological sensing for mobile robot localization. In: *2013 IEEE International Conference on Robotics and Automation (ICRA)*. New York: IEEE, pp. 572–579.
- Dijkstra EW (1959) A note on two problems in connexion with graphs. *Numerische Mathematik* 1: 269–271.

- Dogar M, Hsiao K, Ciocarlie M and Srinivasa S (2012) Physics-based grasp planning through clutter. In: *Robotics: Science and Systems VIII*.
- Dold A (1995) *Lectures on Algebraic Topology. Classics in Mathematics*, 2nd edn. New York: Springer.
- Donald B, Garipey L and Rus D (2000) Distributed manipulation of multiple objects using ropes. In: *IEEE international conference on robotics and automation*, pp. 450–457.
- Fink J, Hsieh MA and Kumar V (2008) Multi-robot manipulation via caging in environments with obstacles. In: *IEEE international conference on robotics and automation (ICRA)*, Pasadena, CA.
- Gamelin TW (2001) *Complex Analysis*. New York: Springer Science.
- Grigoriev D and Slissenko A (1998) 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*. New York: ACM Press, pp. 17–24.
- Hart PE, Nilsson NJ and Raphael B (1968) A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems, Science, and Cybernetics* 4(2): 100–107.
- Hatcher A (2001) *Algebraic Topology*. Cambridge: Cambridge University Press.
- Hershberger J and Snoeyink J (1991) Computing minimum length paths of a given homotopy class. *Computational Geometry: Theory and Applications* 4: 331–342.
- Ivan V, Zarubin D, Toussaint M, Komura T and Vijayakumar S (2013) Topology-based representations for motion planning and generalisation in dynamic environments with interactions. *The International Journal of Robotics Research* 32(9–10): 1151–1163.
- Jiang Q and Kumar V (2010) The inverse kinematics of 3-D towing. *Advances in Robot Kinematics: Motion in Man and Machine*. New York: Springer, pp. 321–328.
- Kerr RA (2010) A lot of oil on the loose, not so much to be found. *Science* 329(734): 734–735.
- Kim S, Bhattacharya S and Kumar V (2013) Dynamic simulation of autonomous boats for cooperative skimming and cleanup. In: *ASME 2013 international design engineering technical conferences and computers and information in engineering conference volume 6b: 37th mechanisms and robotics conference*, Portland, OR.
- Kuderer M, Kretschmar H and Burgard W (2013) Teaching mobile robots to navigate in populated environments. In: *Proceedings of the international conference on intelligent robots and systems (IROS)*, Tokyo, Japan.
- Lamiroux F and Kavraki LE (2001) Planning paths for elastic objects under manipulation constraints. *The International Journal of Robotics Research* 20(3): 188–208.
- Munkres J (1999) *Topology*. Englewood Cliffs, NJ: Prentice Hall.
- Narayanan V, Vernaza P, Likhachev M and LaValle SM (2013) Planning under topological constraints using beam-graphs. In: *2013 IEEE international conference on robotics and automation (ICRA)*. New York: IEEE, pp. 431–437.
- Robertson C and Krauss C (2010) Gulf spill is the largest of its kind, scientists say. *The New York Times*, 2 August. Available at: http://www.nytimes.com/2010/08/03/us/03spill.html?_r=0.
- Saha M, Isto P and Claude Latombe J (2006) Motion planning for robotic manipulation of deformable linear objects. In: *Proceedings IEEE international conference on robotics and automation (ICRA)*, pp. 2478–2484.
- Samar S, Boyd S and Gorinevsky D (2007) Distributed estimation via dual decomposition. In: *Proceedings European control conference (ECC)*, Kos, Greece, July 2007, pp. 1511–1516.
- Scott W and Scott W (1964) *Group Theory (Dover Books on Mathematics Series)*. New York: Dover.
- Stanley R and Rota G (2000) *Enumerative Combinatorics (Cambridge Studies in Advanced Mathematics)*. Cambridge: Cambridge University Press.
- Suykens JAK and Vandewalle JPL (1999) Least squares support vector machine classifiers. *Neural Processing Letters* 9(3): 293–300.
- Tovar B, Cohen F and LaValle SM (2008) Sensor beams, obstacles, and possible paths. In: *Workshop on the algorithmic foundations of robotics*, pp. 317–332.

Appendix A: Index to Multimedia Extension

Archives of IJRR multimedia extensions published prior to 2014 can be found at <http://www.ijrr.org>, after 2014 all videos are available on the IJRR YouTube channel at <http://www.youtube.com/user/ijrrmultimedia>

Table of Multimedia Extension

Extension	Media type	Description
1	Video	Dynamic simulations and field experiment.

Appendix B: Words as homotopy invariants—generalization and justification

For a pointed space, X , with basepoint x_0 (i.e. a space with a preferred point through which all closed curves pass), the construction of a *homotopy invariant* for closed curves can be formally (and with greater generality) justified using the van Kampen's theorem (Hatcher, 2001; Crowell, 1959) as follows:

Given a N -dimensional connected topological space, X (the *free* workspace), suppose we can find α counts of $(N - 1)$ -dimensional subsets, $U_1, U_2, \dots, U_\alpha \subset X$ (the U_i constitute the “rays” in the earlier discussion; see Figure 13) such that

- (i) $X_0 := X - \bigcup_{i=1}^{\alpha} U_i$ is simply-connected;
- (ii) $X_j := X - \bigcup_{i=1, i \neq j}^{\alpha} U_i$ are path-connected and $\pi_1(X_j) \simeq \mathbb{Z}$ for all $j = 1, 2, \dots, \alpha$; and
- (iii) $U_i \cap U_j = \emptyset, \forall i \neq j$.

Note that all of these conditions are satisfied by choosing the *rays* emanating from ζ_i as the U_i , as in our previous discussion.

Using these properties it is easy to see that $X_i \cap X_j = X_0, \forall i \neq j$, and thus the set $C = \{X_0, X_1, X_2, \dots, X_\alpha\}$ constitutes an open cover of X that is closed under

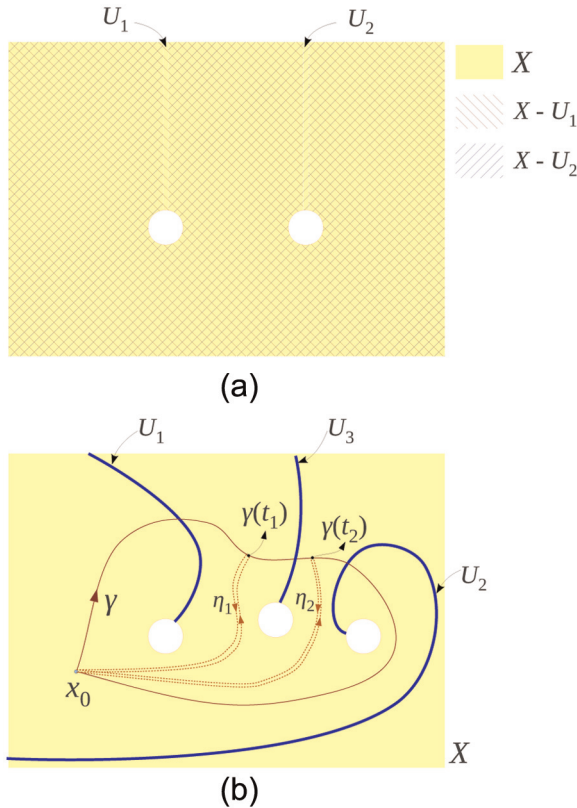


Fig. 13. Use of van Kampen's theorem for justification of proposed homotopy invariants and its generalization. (a) The topological space X and its cover constituting of $X_i = X - U_i$, $i = 1, 2, \dots$. (b) U_i different from "rays". The closed curve γ , and the division of the interval $[0, 1]$ is shown.

intersection. These are essential requirements for the van Kampen's theorem (see Section 1.2 of Hatcher (2001)) to be applied on an open cover of X .

Now it follows from the van Kampen's theorem that the first homotopy group of X is isomorphic to the (non-abelian) *free product* (Scott and Scott, 1964) of the first homotopy groups of $X_0, X_1, X_2, \dots, X_\alpha$. Furthermore, note that $\pi_1(X_0) \simeq 0$ and $\pi_1(X_i) \simeq \mathbb{Z}$, $i = 1, 2, \dots, \alpha$. Thus, it is not surprising that homotopy classes in $\pi_1(X)$ can be expressed as "words" generated by α letters: one for each generator

of a subgroup $\pi_1(X_i) \simeq \mathbb{Z}$. The more challenging part is to actually describe the map at the level of closed curves that induces the isomorphism of the homotopy groups. For that we borrow the construction appearing in proof of van Kampen's theorem (Theorem 1.20) of Hatcher (2001).

Given a closed curve $\gamma: [0, 1] \rightarrow X$, with $\gamma(0) = \gamma(1) = x_0$ (the base-point in X), we divide the interval $[0, 1]$ as $0 = t_0 < t_1 < \dots < t_m = 1$ such that $\gamma(t_k) \in X_0, \forall k$, and $\gamma([t_k, t_{k+1}])$ lies entirely in $X_{u(k)}$, $u(k) \in \{1, 2, \dots, \alpha\}$ (equivalently, $\gamma([t_k, t_{k+1}])$ intersects only with $U_{u(k)}$). This is always possible since we assume everything is Hausdorff. We can then construct curves, η_i such that $\eta_i(0) = \gamma(t_i)$ and $\eta_i(1) = x_0$, lying completely in X_0 , as illustrated in Figure 13(b) (this is always possible since X_0 is path connected). Then clearly γ is homotopic to $\xi := \xi_0 \sqcup \xi_1 \sqcup \dots \sqcup \xi_{m-1}$, where $\xi_0 := \gamma|_{[t_0, t_1]} \sqcup \eta_1$, $\xi_1 := -\eta_1 \sqcup \gamma|_{[t_1, t_2]} \sqcup \eta_2$, $\xi_2 := -\eta_2 \sqcup \gamma|_{[t_2, t_3]} \sqcup \eta_3$, \dots , $\xi_k := -\eta_k \sqcup \gamma|_{[t_k, t_{k+1}]} \sqcup \eta_{k+1}, \dots$, $\xi_{m-1} := -\eta_{m-1} \sqcup \gamma|_{[t_{m-1}, t_m]}$. It is easy to observe that each ξ_k is a closed curve that lies completely inside $X_{u(k)}$, and since $\pi(X_{u(k)}) \simeq H_1(X_{u(k)}; \mathbb{Z}) \simeq \mathbb{Z}$ (the first isomorphism holds due to the Hurewicz theorem (Hatcher, 2001)), the homotopy and homology invariants of closed curves in $X_{u(k)}$ are the same. Thus, $[\xi_k] \in \pi(X_{u(k)})$ is represented by the winding number of ξ_{k+1} , which in turn is computed by the *intersection number* (Dold, 1995; Bhattacharya et al., 2013) between ξ_k and $U_{u(k)}$ (which is equal to the "intersection number" between $\gamma([t_k, t_{k+1}])$ and $U_{u(k)}$, since the η_* do not intersect any of the U_*).

Thus, in conclusion, the homotopy class of γ is represented as $[\gamma] = [\xi] = [\xi_0] \star [\xi_1] \star \dots \star [\xi_{m-1}]$, where " \star " indicates the free product, and the elements $[\xi_k] \in \pi(X_{u(k)}) \simeq H_1(X_{u(k)}; \mathbb{Z}) \simeq \mathbb{Z}$ are represented as unique "letters" exponentiated by the intersection number between ξ_k and $U_{u(k)}$.

It is to be noted that the choice of the subsets U_i such that the mentioned conditions are satisfied is not unique. This indeed hints towards the equivalence of the different representations of homotopy invariants as *words* that are found in the literature Hershberger and Snoeyink (1991); Grigoriev and Slissenko (1998); Tovar et al. (2008); Hatcher (2001); Bhattacharya (2012); Narayanan et al. (2013).