

# A Topological Approach to Using Cables to Separate and Manipulate Sets of Objects

Soonkyum Kim  
Department of Mechanical Engineering  
and Applied Mechanics  
University of Pennsylvania  
Philadelphia, PA 19104  
Email: soonkyum@seas.upenn.edu

Subhrajit Bhattacharya  
Department of Mathematics  
University of Pennsylvania  
Philadelphia,  
PA 19104  
Email: subhrabh@math.upenn.edu

Hordur Heidarsson  
Department of Electrical Engineering  
University of Southern California  
Los Angeles,  
CA 90089  
Email: heidarss@usc.edu

Gaurav S. Sukhatme  
Department of Computer Science  
University of Southern California  
Los Angeles,  
CA 90089  
Email: gaurav@usc.edu

Vijay Kumar  
Department of Mechanical Engineering  
and Applied Mechanics  
University of Pennsylvania  
Philadelphia, PA 19104  
Email: kumar@seas.upenn.edu

**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 describe our preliminary efforts to develop an experimental platform consisting of a system of two cooperating autonomous boats.

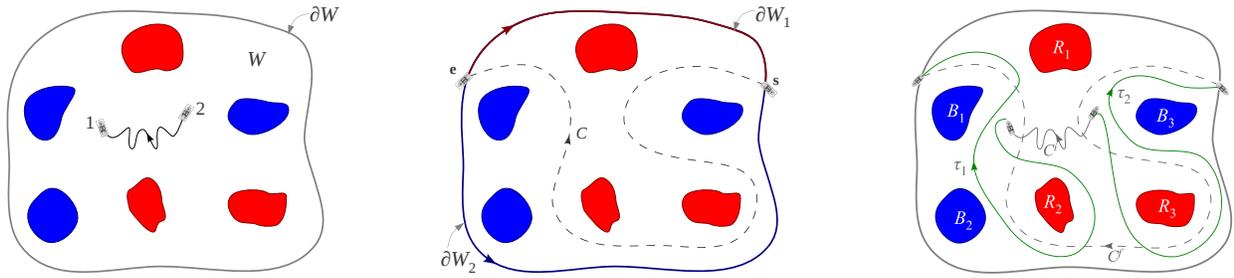
## I. INTRODUCTION

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 [30, 12]. 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 [14].

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* [13]. 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 in [25]. The problem of entangling and disentangling knots and the motion planning for this problem has been addressed in [22]. 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 in [20, 9]. 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 [24, 21]. A description of the dynamics of such systems and an analysis of the problem of cooperative skimming are provided in [4, 2]. 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 the ones that are not of interest. The problem of finding a hypersurface separating two



(a) The initial configuration of the cable and the two robots in the workspace  $W$  with boundary  $\partial W$ . (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. 1. The problem of separating the two types of objects.

types of objects is studied as part of statistical classification problems [7, 28]. 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. 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 [10] 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.

## II. PROBLEM DESCRIPTION

We consider the scenario where there are two classes of objects present in a flat enclosed region,  $W$ . 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,  $\partial W$ , such that the final cable configuration ‘separates’ the blue objects from the red, which we call the *separating configuration* (Figure 1(c)). Once that is achieved, the robots can move along  $\partial W$  to enclose one type of objects and “pull” them out, thus separating and transporting those objects.

Suppose  $e$  and  $s$  are the points on the boundary reached by the robots so that they split  $\partial W$  into  $\partial W_1$  and  $\partial W_2$  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 [5], then, “ $C_1$  separates the two types of objects”  $\iff$  “ $C_2$  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 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 V-C) 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.

## III. THEORETICAL FOUNDATION

Let  $W$  be a 2-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 red objects, and  $B_1, B_2, \dots, B_b$  are  $b$  counts of blue objects. Each object,  $R_i$  or  $B_j$ , is assumed to be connected.

### A. Curves in $(W - \mathcal{O})$

Both cable configurations and robot trajectories 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 say a curve,  $\gamma : [0, 1] \rightarrow (W - \mathcal{O})$ , is embedded [23] if  $\gamma(t) \neq \gamma(t'), \forall t \neq t'$  (*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. For a given curve,  $\gamma$ , we define  $-\gamma : t \mapsto \gamma(1 - t)$ . That is,  $-\gamma$  is the same curve

as  $\gamma$ , but with opposite orientation. The line integral of a differential 1-form,  $\omega = f dx + g dy$ , over  $\gamma$  is defined as  $\int_{\gamma} \omega := \int_0^1 (f \dot{\gamma}_x + g \dot{\gamma}_y) dt$ .

### B. Homology and Homotopy Invariants

**Definition 1 (Homology classes of curves):** Two curves  $\gamma_1, \gamma_2 : [0, 1] \rightarrow (W - \mathcal{O})$  connecting the same start and end points, are homologous (or belong to the same *homology class*) iff  $\gamma_1$  together with  $\gamma_2$  (the latter with opposite orientation) forms the complete boundary of a 2-dimensional manifold embedded in  $(W - \mathcal{O})$  – refer to Figure 1(b) of [5] (not containing/intersecting any of the objects/obstacles) [5, 18].

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  $\gamma_1$  is homologous to  $\gamma_2$ . In [5] the authors proposed a homology class invariant (called the *H-signature*) that is based on simple results from complex analysis.

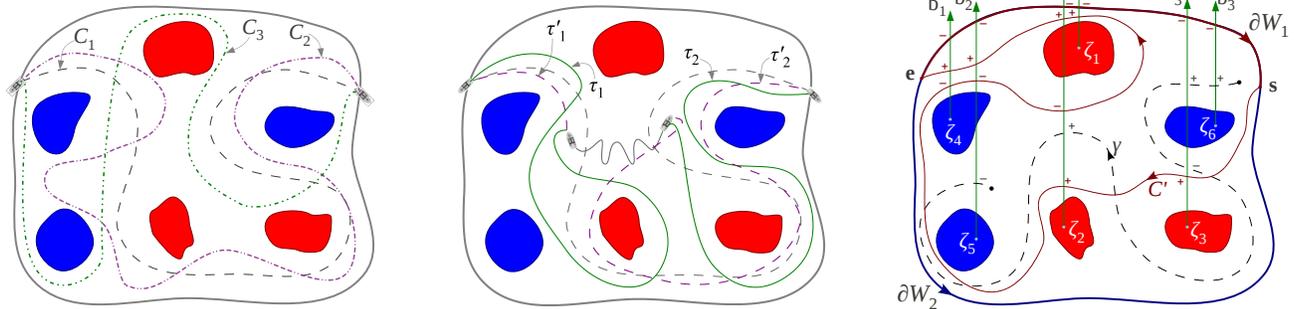
However, the possible choice of such invariants has been broadened in [6], where the choice of the vector of differential 1-forms, which needs to be integrated over  $\gamma$  to obtain the invariant, has been proven to be any complete set of generators of the *de Rham cohomology group*,  $H_{dR}^1(W - \mathcal{O})$ . In particular, the *bump 1-forms* [8],  $\omega_j = -v(y - \zeta_{j,y})\delta(x - \zeta_{j,x})dx$ , (where  $\delta$  is the *Dirac delta function*, and its integral,  $v$ , is the *heaviside step function* – that is, informally speaking,  $\omega_j$  are analogous to a *Dirac delta* distribution over rays emanating from  $\zeta_j$  along positive  $Y$  axis) is a choice that has the simple interpretation of counting the number of times the curve,  $\gamma$ , crosses rays emanating from  $\zeta_j$  (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* won't depend on the choice of the differential 1-forms, as long as they form a generating set of the first *de Rham cohomology group*,  $H_{dR}^1(W - \mathcal{O})$  [8], and will compute the *winding numbers* about  $\zeta_j$ .

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).

**Definition 2 (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 – refer to Figure 1(a) of [5].

Formally, if  $\gamma_1 : [0, 1] \rightarrow (W - \mathcal{O})$  and  $\gamma_2 : [0, 1] \rightarrow (W - \mathcal{O})$  represent the two trajectories (with  $\gamma_1(0) = \gamma_2(0) = \mathbf{x}_s$  and  $\gamma_1(1) = \gamma_2(1) = \mathbf{x}_g$ ), then  $\gamma_1$  is homotopic to  $\gamma_2$  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  $\eta(0, \gamma) = \mathbf{x}_s, \eta(1, \mu) = \mathbf{x}_g \forall \mu \in [0, 1]$  [5, 18].

*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 [18]. However, for curves in 2-dimensional plane with punctures (*i.e.* obstacles/objects), there is a relatively simple representation of the homotopy group and a way of computing the homotopy class of a given curve [16, 19, 29, 18, 3]: We consider *representative points*,  $\zeta_i$  as before, and *parallel non-intersecting rays*,  $r_1, r_2, \dots, r_r$  and  $b_1, b_2, \dots, b_b$ , emanating from the red and blue objects respectively (Figure 2(c)). We form a *word* by tracing  $\gamma$ , 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  $\gamma$  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  $\gamma$  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),  $\gamma$ , and we will write this as  $h(\gamma)$  and call it the “*h-signature of } \gamma”. 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*



(a) Three possible cable configurations separating the two types of objects.  $C_1$  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,  $\tau_1$  and  $\tau_1'$  are not homotopic, neither are  $\tau_2$  and  $\tau_2'$ . 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 1(b).

(c)  $\zeta_i$  are representative points 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,  $h(\gamma) = “b_3 r_2 b_2^{-1}”$ .

Fig. 2. Homotopy and homology classes of cable configurations and trajectories.

element of the *non-abelian* group *freely generated* [26, 18] by  $\{r_1, r_2, \dots, r_r, b_1, b_2, \dots, b_b\}$ . 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  $\mathfrak{w} \diamond \mathfrak{w}^{-1} = \text{“ ”}$ , the identity element). Thus,  $(\mathfrak{w}_1 \diamond \mathfrak{w}_2)^{-1} = \mathfrak{w}_2^{-1} \diamond \mathfrak{w}_1^{-1}$ . As an example,  $(\text{“}b_3 r_2 b_2^{-1}\text{”})^{-1} = \text{“}b_2 r_2^{-1} b_3^{-1}\text{”}$ .

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,  $\text{“}abcde\text{”}$  will be considered to be the same as  $\text{“}cdeab\text{”}$ . 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 get,

$$\begin{aligned} h(\partial W_1 \sqcup C') &= h(\partial W_1) \diamond h(C') \\ &= \text{“}b_1^{-1} b_2^{-1} r_2^{-1} r_1^{-1} r_3^{-1} b_3^{-1}\text{”} \diamond \text{“}r_3 r_2 b_1^{-1} b_2^{-1} r_2^{-1} r_1 r_2 b_2 b_1\text{”} \\ &= \text{“}b_1^{-1} b_2^{-1} r_2^{-1} r_1^{-1} r_3^{-1} b_3^{-1} r_3 r_2 b_1^{-1} b_2^{-1} r_2^{-1} r_1 r_2 b_2 b_1\text{”} \\ &= \text{“}r_3^{-1} b_3^{-1} r_3 r_2 b_1^{-1} b_2^{-1} r_2^{-1}\text{”} \end{aligned}$$

(*after canceling the letters at the start & the end*), which is the completely reduced word.

The *homotopy invariant* of a curve,  $\gamma$ , is the *reduced word* constructed in the described way, with cyclic permutations of a word being considered equivalent when  $\gamma$  is closed. 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  $\zeta_i$ . But the *homotopy invariant* word is highly dependent on the choice of the direction of the rays.

**The Hurewicz map:** While trajectories that are homotopic are also homologous, the converse is not necessarily true (see Fig. 1 of [5]). The Hurewicz map [18] 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  $h(\partial W_1 \sqcup C') = \text{“}r_3^{-1} b_3^{-1} r_3 r_2 b_1^{-1} b_2^{-1} r_2^{-1}\text{”}$ . Letting the letters commute we have the word  $\text{“}r_1^0 r_2^0 r_3^0 b_1^{-1} b_2^{-1} b_3^{-1}\text{”}$  (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  $H(\partial W_1 \sqcup C') = h_*(\text{“}r_3^{-1} b_3^{-1} r_3 r_2 b_1^{-1} b_2^{-1} r_2^{-1}\text{”}) = [0, 0, 0, -1, -1, -1]^T$ .

**C. Propositions on Object Separation & Cable Manipulation**

**Proposition 1:** Suppose  $C$  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  $\partial W$  into two parts:  $\partial W_1$  and  $\partial W_2$  (which themselves

are curves in  $(W - \mathcal{O})$ ). We assign orientation to  $\partial W_1$  and  $\partial W_2$  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  $\partial W_1$  over  $\partial W_2$  is made without loss of generality. The conditions could have been stated in terms of  $\partial W_2$  as well.

*Sketch of Proof:* 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 [15] 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  $\pm 1$  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 red objects or the blue 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 red objects from blue. However we previously saw that the *reduced word* for  $(C' \sqcup \partial W_1)$  is,  $h(C') \diamond h(\partial W_1) = \text{“}r_3^{-1} b_3^{-1} r_3 r_2 b_1^{-1} b_2^{-1} r_2^{-1}\text{”}$ . Likewise the *reduced word*  $h(C') \diamond h(\partial W_2) = \text{“}r_3 r_2 b_1^{-1} b_2^{-1} r_2^{-1} r_1 r_2 b_2 b_1\text{”}$ . Neither of these words are helpful in identifying the fact that  $C'$  separates the blue objects from the red. 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 blue from the red 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  $\tau_1$  and  $\tau_2$  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* [18], *i.e.*  $h(C \sqcup \tau_1 \sqcup -C' \sqcup -\tau_2) = h(C) \diamond h(\tau_1) \diamond h(C')^{-1} \diamond h(\tau_2)^{-1} = \text{“ ”}$ , is the empty word (*identity element*).

*Sketch of Proof:* We note that unlike in Proposition 1 we don't 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

trajectories  $\tau_1$  and  $\tau_2$  carries the cable from configuration  $C$  to final configuration  $C'$ . We choose two arbitrary points,  $p_1$  and  $p_2$ , on the trajectories  $\tau_1$  and  $\tau_2$  respectively, as shown in Figure 3. Next consider the sequence of cable configurations from  $C$  to  $C'$  as the robots carry it. We can thus construct a continuous function (a homotopy),  $\overline{C} : [0, 1] \times [0, 1] \rightarrow (W - \mathcal{O})$ , such that  $\overline{C}(0, \cdot) \equiv C(\cdot)$  and  $\overline{C}(1, \cdot) \equiv C'(\cdot)$ , and  $\overline{C}(t)$  is a general intermediate cable configuration. Such a curve,  $\overline{C}(t)$ , has its end points  $q_1(t) \in \tau_1$  and  $q_2(t) \in \tau_2$  (Figure 3). We consider the curve connecting  $q_1(t)$  to  $p_1$  and lying on  $\tau_1$  (call it  $\widetilde{q_1(t)p_1}$ ), and the one connecting  $q_2(t)$  to  $p_2$  and lying on  $\tau_2$  (call it  $\widetilde{q_2(t)p_2}$ ). Thus, the sequence of curves,  $D(t) := \left( -\widetilde{q_2(t)p_2} \sqcup \overline{C}(t) \sqcup \widetilde{q_1(t)p_1} \right)$ , defines a homotopy between curves connecting  $p_1$  and  $p_2$ . Thus,  $D(0) \sqcup -D(1)$  is null-homotopic. That is,  $\left( -\widetilde{q_2(0)p_2} \sqcup \overline{C}(0) \sqcup \widetilde{q_1(0)p_1} \right) \sqcup -\left( -\widetilde{q_2(1)p_2} \sqcup \overline{C}(1) \sqcup \widetilde{q_1(1)p_1} \right) \equiv (C \sqcup \tau_1 \sqcup -C' \sqcup -\tau_2)$ , is null-homotopic.

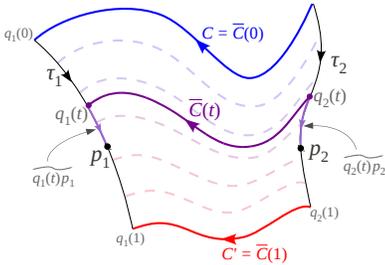


Fig. 3. Illustration for Proof of Prop. 2.

#### IV. IMPLEMENTATION

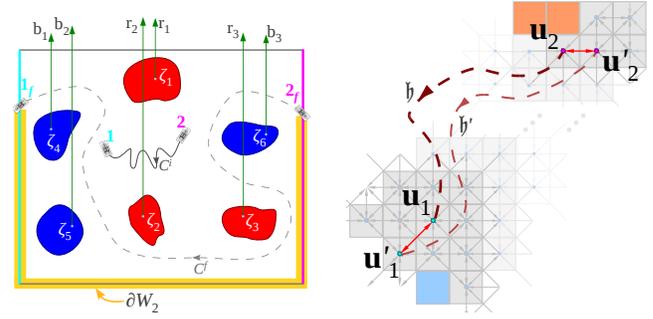
For simplicity, we assume that the environment,  $W$ , is a rectangular region, and all the rays,  $r_j$ ,  $j = 1, 2, \dots, r$  and  $b_j$ ,  $j = 1, 2, \dots, b$ , are parallel, pointing along the positive  $Y$  axis. Furthermore, we restrict the final goals of the robots to the left and right boundaries of the environment ( $\partial W_1$  at  $x = x_l$  and  $\partial W_r$  at  $x = x_r$  respectively), but they need to reach the opposite edges. Thus a part of the boundary,  $\partial W_2$ , will never intersect any of the rays (Figure 4(a)), and hence  $H(\partial W_2) = [0, 0, \dots, 0]^T$  and  $h(\partial W_2) = ""$ . This simplifies the computation of  $H(C^f \sqcup \partial W_2)$  for Proposition 1 to the computation of  $H(C^f)$ .

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 in [5].

While the graph,  $\mathcal{G}$ , itself can be quite arbitrary, for simplicity we used a uniform 8-connected discrete representation (see Figure 4(b)) of the environment for all our simulations and experiments.

##### A. 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



(a) The rectangular environment with the goal of the robots being the left (cyan line) and right (magenta line) boundaries.  $\partial W_2$  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.

Fig. 4. The environment and its discretization.

(or left) edge. In the first approach we plan trajectories in the joint state-space of the two robots. 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  $(\mathbf{u}_1, \mathbf{u}_2)$ . We are given an initial vertex in the joint state-space,  $(\mathbf{u}_1^i, \mathbf{u}_2^i)$ , 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*).

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 [5]. Thus the initial vertex in the graph is  $\mathbf{v}_i = (\mathbf{u}_1^i, \mathbf{u}_2^i, h^i)$ , 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  $h' = h(-\tau_2) \diamond h \diamond h(\tau_1)$  (recall,  $\diamond$  is concatenation, followed by reduction), where  $\tau_1$  and  $\tau_2$  are trajectories taken by the robots for the transition (see Figure 4(b)). Thus, for each edge  $[(\mathbf{u}_1, \mathbf{u}_2) \rightsquigarrow (\mathbf{u}'_1, \mathbf{u}'_2)] \in E(\mathcal{J})$ , the vertex  $(\mathbf{u}_1, \mathbf{u}_2, h)$  is connected to neighbors  $(\mathbf{u}'_1, \mathbf{u}'_2, h(\mathbf{u}'_2 \mathbf{u}_2) \diamond h \diamond h(\mathbf{u}_1 \mathbf{u}'_1))$  (where,  $[a \rightsquigarrow b]$  is used to indicate an edge in edge set,  $E(\mathcal{G})$ , from vertex  $a$  to  $b$ , and  $\overrightarrow{ab}$  is the curve/line segment that constitutes the edge.  $\overleftarrow{ba}$  is the same curve but with opposite orientation).

We choose the optimization objective to be the sum of the length of the robot trajectories. Thus, the cost of the edge  $[(\mathbf{u}_1, \mathbf{u}_2, h) \rightsquigarrow (\mathbf{u}'_1, \mathbf{u}'_2, h \diamond h(\mathbf{u}_1 \mathbf{u}'_1) \diamond h(\mathbf{u}'_2 \mathbf{u}_2))]$   $\in E(\mathcal{J}_h)$  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 at a vertex  $\mathbf{u}_j$ ).

Starting at  $(\mathbf{u}_1^i, \mathbf{u}_2^i, h^i)$  we thus keep expanding the vertices in the graph,  $\mathcal{J}_h$ , using a search algorithm (we use Dijkstra's [11] or A\* [17] since they are complete, optimal and deterministic). A vertex  $(\mathbf{u}_1, \mathbf{u}_2, h)$  is deemed as goal if

$\mathbf{u}_1 \in \partial W_l$  and  $\mathbf{u}_2 \in \partial W_r$  (or vice-versa), and if  $h_*(\mathbf{h}) + H(\partial W_2)$  ( $= h_*(\mathbf{h})$ ) 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 like inter-robot collision avoidance, communication constraints, etc.

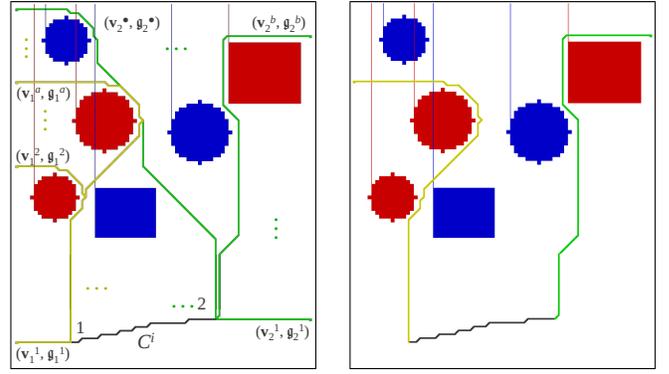
### B. Decoupled Planning: A Distributed Approach

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 4-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), and run those searches parallelly (parallel threads in our C++ implementation), comparing the solutions obtained from each parallel process as they progress, and being able to conclude when the optimal solution is found, and thus halting the threads.

The  $h$ -augmented graph,  $\mathcal{G}_h$ , is very similar to the concept of the  $H$ -signature augmented graph,  $\mathcal{G}_H$  described in [5], only with the homology invariants being replaced by the homotopy invariants. Corresponding to a given  $\mathbf{u} \in V(\mathcal{G})$ , there exists discrete number 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  $\mathbf{u}$ . 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 set* with) the vertex  $(\mathbf{u}_j^i, \dots) \in \mathcal{G}_{h,j}$ ,  $j = 1, 2$ . We keep expanding the vertices in the respective graphs, and keep storing a path every time  $\partial W_l$  or  $\partial W_r$  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 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\* [17]). Suppose for robot ' $j$ ' such goal vertices are  $\{(\mathbf{v}_j^1, \mathbf{g}_j^1), (\mathbf{v}_j^2, \mathbf{g}_j^2), (\mathbf{v}_j^3, \mathbf{g}_j^3), \dots\}$  with costs of the respective optimal paths  $c_j^1 \leq c_j^2 \leq c_j^3 \leq \dots$ , for  $j = 1, 2$ .

We define a *partial order* [27],  $\preceq$ , on  $\mathbb{R}^2$ , to compare the cost of pairs of paths of robots 1 and 2. One obvious choice is to compare the sum of the path costs:  $(\alpha_1, \alpha_2) \preceq (\beta_1, \beta_2) \Leftrightarrow \alpha_1 + \alpha_2 \leq \beta_1 + \beta_2$ . However, 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 the 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 (rather than their sum). Thus, we define the partial order to be



(a) Optimal paths in different homotopy classes corresponding to goal vertices  $(\mathbf{v}_1^a, \mathbf{g}_1^a), (\mathbf{v}_1^b, \mathbf{g}_1^b), (\mathbf{v}_2^a, \mathbf{g}_2^a), (\mathbf{v}_2^b, \mathbf{g}_2^b) \in \mathcal{G}_{h,j}$  are found in parallel threads for robot  $j = 1$  (yellow) and robot  $j = 2$  (green). (b) A compatible combination of paths (with end vertices  $(\mathbf{v}_1^a, \mathbf{g}_1^a)$  and  $(\mathbf{v}_2^b, \mathbf{g}_2^b)$ ) is such that the end points,  $\mathbf{v}_1^a$  &  $\mathbf{v}_2^a$ , lie on the opposite edges of robot  $j = 1$  ( $W$ , and  $h_*(\mathbf{g}_2^b)^{-1} \diamond h(C^i) \diamond \mathbf{g}_1^a$  satisfies the condition of Proposition 1.

Fig. 5. 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.

$$(\alpha_1, \alpha_2) \preceq (\beta_1, \beta_2) \Leftrightarrow \max(\alpha_1, \alpha_2) < \max(\beta_1, \beta_2), \text{ or,} \\ (\max(\alpha_1, \alpha_2) = \max(\beta_1, \beta_2) \text{ and} \\ \min(\alpha_1, \alpha_2) \leq \min(\beta_1, \beta_2))$$

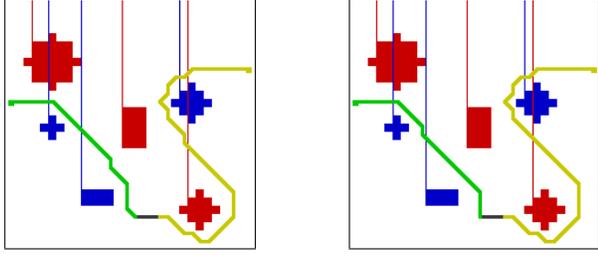
which we call the *sorted lexicographic order*.

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,  $(\mathbf{v}_1^a, \mathbf{g}_1^a)$  and  $(\mathbf{v}_2^b, \mathbf{g}_2^b)$ , is deemed '*compatible*' (Figure 5(b)) if the corresponding final cable configuration (whose  $h$ -signature, by Proposition 2, is equal to  $(\mathbf{g}_2^b)^{-1} \diamond h(C^i) \diamond \mathbf{g}_1^a$ ) is a separating configuration. That is, due to Proposition 1, a pair is compatible if  $h_*(\mathbf{g}_2^b)^{-1} \diamond h(C^i) \diamond \mathbf{g}_1^a + H(\partial W_2)$  is a vector with first  $r$  components  $\pm 1$  and rest zeros, or last  $b$  components  $\pm 1$  and rest zeros. We keep record of the most optimal compatible pair (*i.e.*, one with lowest  $(c_1^*, c_2^*)$ , where comparisons are made using ' $\preceq$ ').

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, the current optimal pair is the global optimal one.

## V. SIMULATION RESULTS

We implemented the search in the joint state-space as well as the decoupled search in C++ programming language with ROS integration, and used A\* search algorithm. 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



(a) Planning in the joint state-space took 4250s. The sum of the costs of two training gives result with the same optimal cost, but takes about 2s to run.

(b) The distributed decoupled planning gives result with the same optimal cost, but takes about 2s to run.

Fig. 6. A simple  $30 \times 30$  environment with  $r = b = 3$ . The green & yellow are the trajectories of the robots. The rays emanating from  $\zeta_j$  are also shown. The dark gray segment indicates the initial cable configuration.

more detailed discussion on the generality of the technique can be found in [5].

### A. Joint State-space Plan

The search in this 4-dimensional environment is prohibitively expensive for large environments. Figure 6(a) shows the result in a simple environment,  $30 \times 30$  discretized, and with 3 objects of each type. The search took about 4250 s and expanded 1484999 vertices in  $\mathcal{J}_h$ . Figure 6(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 19144 and 19593 vertices being expanded in  $\mathcal{G}_{h,1}$  and  $\mathcal{G}_{h,2}$ . All objects were *inflated* to avoid collision.

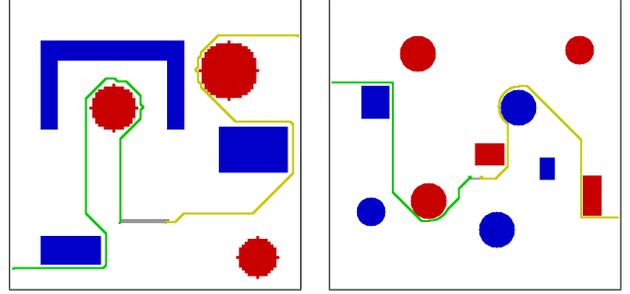
### B. Decoupled Planning

In this section we present results obtained using the decoupled, distributed implementation. The sorted lexicographic order was used for ' $\preceq$ '. Figure 7(a) show the plans obtained for two robots in a  $100 \times 100$  discretized environment. The planning took about 1.3 s, and expanded 39764 and 40066 vertices in the graphs of the two robots. Figure 7(b) shows the result in a much larger ( $400 \times 400$  discretized) environment. The planning time for this case was 490 s, with 1086182 and 1079670 vertices being expanded.

### C. Dynamic Simulation and 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. However, in a practical implementation, where the objects will be free to move on the surface, the interaction between the cable and the objects will change the configuration of the environment. Consequently there comes the need for re-planning. For the purpose of testing this scenario we build 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 [4], and modeled the contacts using linear complementarity conditions [1]. We use a simple feedback (PD) controller to make each robot follow the paths generated by the planner. See [1].

Instead of solving the entire problem every time the environment changes, we invoke a re-planning algorithm whenever two objects exchange the order of the  $X$  coordinates of their



(a) The planned trajectories in a  $100 \times 100$  discretized environment. (b) The planned trajectories in a  $400 \times 400$  discretized environment. Fig. 7. Decoupled, distributed plans. Initial cable is shown in gray/black. Trajectories are in green and yellow.

representative points (*i.e.*, the *rays* emanating from  $\zeta_j$  cross each other) or one of the planned trajectories become invalid (due to an object moving on top of it). Suppose  $\mathbf{g}_1$  and  $\mathbf{g}_2$  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. Set  $\mathbf{g}'_j = \mathbf{g}_j$ ,  $j = 1, 2$ . 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 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  $(\mathbf{p}_j, \cdot)$ , where  $\mathbf{p}_j \in \mathcal{G}$  is the current vertex position of robot  $j$ ) with the constraints that the new trajectories need to have  $h$ -signature of  $\mathbf{g}'_j$ .

Since we know the target  $h$ -signature, it is possible to construct a more efficient, yet admissible, heuristic function than before for performing optimal search using A\* algorithm. We consider the vertex  $(\mathbf{u}, \mathbf{h})$  in the  $h$ -augmented graph. Suppose the representative points corresponding to the letters appearing in  $\mathbf{h}^{-1} \diamond \mathbf{g}'_j$  is  $\zeta_{\sigma_1}, \zeta_{\sigma_2}, \dots, \zeta_{\sigma_m}$ . It is obvious that the remaining part of a valid trajectory after  $(\mathbf{u}, \mathbf{h})$  will have to cross each of the rays emanating from these  $\zeta_{\sigma_k}$  in this order (besides possibly crossing others that will cancel out). Thus the following heuristic function is a lower bound of the cost (*i.e.*, admissible) for reaching a goal state,

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

where  $f_8(\mathbf{u}, \mathbf{v}) = \sqrt{2} \min(|u_x - v_x|, |u_y - v_y|) + ||u_x - v_x| - |u_y - v_y||$  is an admissible heuristic for a 8-connected graph. Figure 8 shows the simulation result. Figure 8(a) shows the initial configuration of the system. As the objects move and the map change, the planned paths of robot are re-computed (shown by green curves in Figures 8(b)-(e)). We are able to successfully separate the red objects from the blue ones.

### D. Experimental Results

We are in the processing of conducting a field experiment for validating the planner using two Autonomous Surface Vessels (ASV) at Puddingstone Reservoir in San Dimas, CA. The two ASVs are identical, each around 2 m long and 0.8 m wide, capable of speeds up to 1.6 m/s, using two electric thrusters and a rudder for control. Both are equipped with a GPS, an IMU with integrated compass and an onboard computer for control. The two ASVs have a 40 m long floating rope attached between them with periodically spaced markers on it for increased visibility and to use as fixed sampling points. Buoys (the objects to be separated) are placed

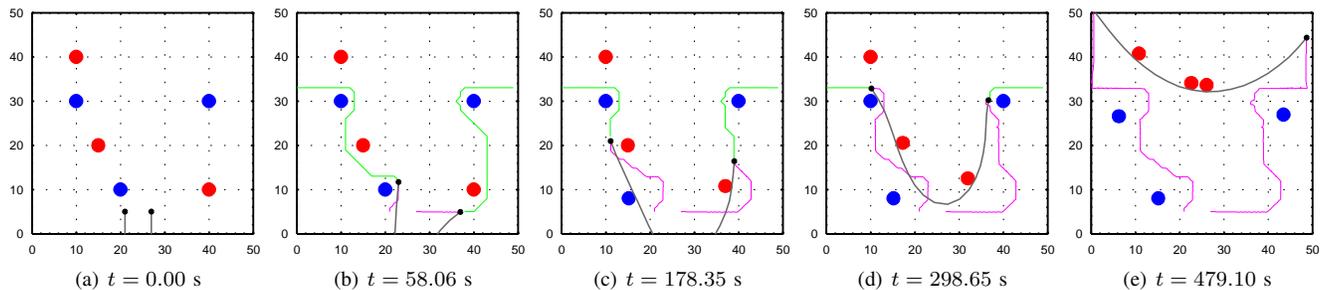


Fig. 8. Dynamic simulation for separation of objects. The gray curve is the cable, with black dots marking robots at its ends. Green curves are the planned trajectories. Magenta curves are the robot footprints. Red & blue disks are the rigid freely-floating objects. See <http://youtu.be/GyCn-8yDzO0> for video.

in the water and anchored in place in an area  $50 \text{ m} \times 50 \text{ m}$ , and their approximate locations recorded using GPS.



Fig. 9. Experimental setup showing the ASVs. See <http://youtu.be/GyCn-8yDzO0> for video.

overlooking it. Figure 9 shows the experimental setup.

## VI. CONCLUSIONS

In this paper we present 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 develop 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 complete dynamic simulation. Experiment with real robots is under progress and results will be presented in a succeeding journal paper.

## ACKNOWLEDGEMENTS

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] Dynamic simulation of autonomous boats for cooperative skimming and cleanup. Technical report. <http://db.tt/FXsAAw4r>.
- [2] Joaquin Aranda, Pablo Gonzalez de Santos, and Jess Manuel de la Cruz. *Robotics and Automation in the maritime industries*. Produccion Grafica Multimedia (PGM), 2006.
- [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] Subhrajit Bhattacharya, David Lipsky, Robert Ghrist, and Vijay Kumar. Invariants for homology classes with application to optimal search and planning problem in robotics. *Electronic pre-print*, Aug 2012. <http://arxiv.org/abs/1208.0573> [math.AT].
- [7] D.A. Binder. Approximations to bayesian clustering rules. *Biometrika*, 68:275–285, 1981.
- [8] R. Bott and L.W. Tu. *Differential Forms in Algebraic Topology*. Graduate texts in mathematics. Springer-Verlag, 1982.
- [9] P. Cheng, J. Fink, and V. Kumar. Cooperative towing with multiple robots. *ASME Transactions: Journal of Mechanisms and Robotics*, 1, February 2009.
- [10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT Press, 2nd edition, 2001.
- [11] Edsger W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
- [12] Mehmet Dogar, Kaijen Hsiao, Matei Ciocarlie, and Siddhartha Srinivasa. Physics-based grasp planning through clutter. In *Robotics: Science and Systems VIII*, July 2012.
- [13] Bruce Donald, Larry Garipey, and Daniela Rus. Distributed manipulation of multiple objects using ropes. In *IEEE International Conference on Robotics and Automation*, pages 445–457, 2000.
- [14] Jonathan Fink, M. Ani Hsieh, and Vijay Kumar. Multi-robot manipulation via caging in environments with obstacles. In *IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, May 2008.
- [15] Theodore W. Gamelin. *Complex analysis*. Springer Science, 2001.
- [16] 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.
- [17] 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.
- [18] Allen Hatcher. *Algebraic Topology*. Cambridge Univ. Press, 2001.
- [19] J. Hershberger and J. Snoeyink. Computing minimum length paths of a given homotopy class. *Comput. Geom. Theory Appl*, 4:331–342, 1991.
- [20] Qimi Jiang and Vijay Kumar. The inverse kinematics of 3-d towing. *Advances in Robot Kinematics: Motion in Man and Machine*, pages 321–328, 2010.
- [21] Richard A. Kerr. A lot of oil on the loose, not so much to be found. *Science*, 329(734), 2010.
- [22] F. Lamiroux and L. E. Kavraki. Planning paths for elastic objects under manipulation constraints. *International Journal of Robotics Research*, 20(3):188–208, 2001.
- [23] James Munkres. *Topology*. Prentice Hall, 1999.
- [24] Campbell Robertson and Clifford Krauss. Gulf spill is the largest of its kind, scientists say. *The New York Times*, August 2010.
- [25] Mitul Saha, Pekka Isto, and Jean claude Latombe. Motion planning for robotic manipulation of deformable linear objects. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 2478–2484, 2006.
- [26] W.R. Scott and W.R. Scott. *Group Theory*. Dover Books on Mathematics Series. Dover Publ., 1964.
- [27] R.P. Stanley and G.C. Rota. *Enumerative Combinatorics:.* Cambridge studies in advanced mathematics. Cambridge University Press, 2000.
- [28] J. A. K. Suykens and J. P. L. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, Jun 1999.
- [29] 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.
- [30] Dmitry Zarubin, Vladimir Ivan, Marc Toussaint, Taku Komura, and Sethu Vijayakumar. Hierarchical motion planning in topological representations. In *Proc. Robotics: Science and Systems (RSS)*, Sydney, Australia, 2012.