Topological Constraints in Search-based Robot Path Planning

S. Bhattacharya

Department of Mechanical Engineering
and Applied Mechanics,
University of Pennsylvania.
email: subhrabh@seas.upenn.edu

M. Likhachev

Robotics Institute,
Carnegie Mellon University.
email: maxim@cs.cmu.edu

V. Kumar

Department of Mechanical Engineering
and Applied Mechanics,
University of Pennsylvania.
email: kumar@seas.upenn.edu

Abstract

There are many applications in motion planning where it is important to consider and distinguish between different topological classes of trajectories. The two important, but related, topological concepts for classifying manifolds that are of importance to us are those of *homotopy* and *homology*. In this paper we consider the problem of robot exploration and planning in Euclidean configuration spaces with obstacles to (a) identify and represent different homology classes of trajectories; (b) plan trajectories constrained to certain homology classes or avoiding specified homology classes; and (c) explore different homotopy classes of trajectories in an environment and determine the least cost trajectories in each class. We exploit theorems from *complex analysis* and the *theory of electromagnetism* to solve the problem 2-dimensional and 3-dimensional configuration spaces respectively. Finally, we describe the extension of these ideas to arbitrary D-dimensional configuration spaces. We incorporate these basic concepts to develop a practical graph-search based planning tool with theoretical guarantees by combining integration theory with search techniques, and illustrate it with several examples.

1 Introduction

1.1 Motivation: Homotopy classes of trajectories

Homotopy classes of trajectories arise due to presence of obstacles in an environment. Two trajectories connecting the same start and goal coordinates are in the same homotopy class if they can be smoothly deformed into one another without intersecting any obstacle in the environment, otherwise they are in different homotopy classes. In many applications, it is important to distinguish between trajectories in different homotopy classes, as well as identify the different homotopy classes in an environment (*e.g.*, trajectories that go left around a circle in two dimensions versus right). For example, in order to deploy a group of agents to explore an environment (*e.g.,* for eliminating potential threats, searching for rewards/targets [20], as well as for updating obstacle map of a partially known environment [8]), an efficient strategy ought to be able to identify the different homotopy classes and deploy one robot in each homotopy class. One may also wish to determine the least cost path for each robot constrained to or avoiding specified homotopy classes. In many problems the notion of *visibility* is linked intrinsically with homotopy classes. In tracking of uncertain agents in an environment with dynamic obstacles, the ability to deal with occlusions during a certain time frame is important [23]. A knowledge of the possible homotopy classes of trajectories that a target can take in the environment when it is occluded can help more efficient belief propagation.

Classification of homotopy classes in two-dimensional workspaces has been studied in the robotics literature using geometric methods [12, 15], probabilistic road-map construction [19] techniques and triangulation-based path planning [9]. In two dimensions, topological information can be extracted using geometric methods by counting counting the number of times a trajectory intersects boundaries of the obstacles or rays emanating from the obstacles, or informed ways of dividing the free space into cells to keep track of the sequence in which a trajectory visits those cells. To the

best of our understanding, none of these methods in literature satisfactorily extend to configuration spaces of dimension higher than two. While in a 2-dimensional configuration space such methods can be used for telling whether or not two trajectories belong to the same homotopy class, efficient planning for least cost trajectories with homotopy class constraints is difficult using such representations even in 2-dimensions. Neither is it possible to efficiently explore/find optimal trajectories in different homotopy classes in an environment. To our knowledge, there has been no prior research on planning trajectories with topological constraints using search-based methods.

In this paper we propose a novel way of classifying and representing *homology* classes, a close analog of homotopy classes, in two and higher dimensional Euclidean configuration spaces, which are the types of configuration spaces we encounter most often in robot planning problems. For the 2-dimensional case we use theorems from complex analysis for developing a compact way of representing homology classes of trajectories, while for 3-dimensional configuration spaces we exploit theorems from electromagnetism.[1] Finally, we show that the formulae for 2 and 3 dimensional cases can in fact be extended to higher D-dimensional Euclidean configuration spaces with obstacles [3]. This is illustrated with examples in a 4-dimensional configuration space.

The novelty of our work lies in the fact that our proposed representation allows us to identify/distinguish trajectories in different classes and compute least-cost paths in non trivial configuration spaces with topological constraints using graph search-based planning algorithms. The representation we propose is designed to be independent of the type of the environment, the discretization scheme or cost function. Our proposed representation can also be used in configuration spaces with additional degrees of freedom that do not effect homotopy classes of the trajectories (*e.g.* for unicycle modes of mobile robots, the configuration space consists of variables X, Y and θ. But the last variable, θ, does not effect the homotopy classes of trajectories. Only the projection of the $x - y$ plane is enough to capture the topological information).

Using such a representation we show that topological constraints can be seamlessly integrated with graph search techniques for determining optimal paths subject to constraints. We also discuss how this method can be used to explore multiple *homotopy* classes in an environment using a single graph search.

1.2   Capturing Topological Information in Search-based Planning

In search-based planning algorithms one typically starts by discretizing a given environment to create a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Starting form an initial vertex, $v_s \in \mathcal{V}$, a typical graph-search algorithm *expands* the nodes of the graph by traversing the edges. Values are maintained and associated with each expanded node that capture the metric information (distance/cost) of shortest path leading to the expanded node from v_s. For example, A* search algorithm maintains two functions $g, f : \mathcal{V} \to \mathbb{R}$. $g(v)$ is the cost of the current path from the start node to node v, and $f(v) = g(v) + h(v)$ is an estimate of the total distance from start to goal going through v. The algorithm maintains an *open set*, the set of nodes to be *expanded*. Each time it expands a node v, it updates the values of $g(v')$ for each neighbor v' of v, by adding to $g(v)$ the cost of the edge $c(\overline{vv'})$ (the update happens only if the newly computed value is lower than the previous value). This process continues until a desired vertex $v_g \in \mathcal{V}$ is reached [13].

The fact that the value of $g(v')$ can be computed from $g(v) + c(\overline{vv'})$ is due to the fact that the cost function is additive (*i.e.* if α and β are two curves that share a common end point, then $c(\alpha \sqcup \beta) = c(\alpha) + c(\beta)$, where "$\sqcup$" indicates the *disjoint union*, and represent the total curve formed by the two curves together). This is because the metric information about the underlying space is captured using a differential 1-form ('quantities' that can be integrated over 1-dimensional manifolds, the trajectories [7, 22]), namely the infinitesimal length/cost, $\mathrm{d}l$. The cost of an edge, e, of the graph is then computed as an integral of the form $c(e) = \int_e \mathcal{J}(l)\,\mathrm{d}l$ (with some scaling function $\mathcal{J}$). This implies, in an arbitrary graph search algorithm, during the expansion of the vertices of the graph, the cost of the shortest path up to a vertex that is being expanded can simply be computed by adding to that of its parent (in terms of sequence of expansions) the cost of the edge connecting to it. This additive property of length/cost is key in developing such graph search algorithms.

While the differential 1-form, $\mathcal{J}(l)\,\mathrm{d}l$, yields metric information, there are other differential 1-forms that can incorporate other information about the underlying space and can be used for guiding the search algorithm. The main idea in this paper is to determine a differential 1-forms that encodes topological information about the space and let us guide the search accordingly.

---

[1]Parts of this paper have appeared elsewhere — the 2-dimensional case was introduced in [2] and the 3-dimensional case was analyzed in [5].

1.3 H-signature as class invariants for trajectories

We consider a very general differential 1-form in a given D-dimensional configuration space $\mathcal{C}$. If $x_1, x_2, \cdots, x_D$ are the coordinate variables describing the configuration space, a general differential 1-form can be written as $\mathrm{d}h := f_1(\mathbf{x})\mathrm{d}x_1 + f_2(\mathbf{x})\mathrm{d}x_2 + \cdots + f_D(\mathbf{x})\mathrm{d}x_D$. Thus, for any given trajectory/curve, τ, in this configuration space, one can compute $\mathcal{H}(\tau) = \int_\tau \mathrm{d}h$. We call this the H-signature of τ. In Sections 3.2 and 4.2.2 we will design the differential 1-forms, and hence the H-signature of a trajectory, for the 2 and 3 dimensional configuration spaces respectively, such that they are invariants for homology classes of trajectories.

   We want to design the 1-form $\mathrm{d}h$ and the H-signature of a trajectory such that it is an invariant across trajectories in the same homotopy class. However, because we use 1-forms and their integrals along closed curves to classify trajectories, we naturally obtain invariants for homology classes of trajectories [14, 18, 7]. But in most practical robotics problems the notion of homology and homotopy of trajectories can be used interchangeably, especially when finding the least cost path. This is discussed in greater detail with examples in Sections 5.1 and 6.

2   Homotopy and Homology Classes of Trajectories

Definition 1 (Homotopic trajectories). *Two trajectories τ_1 and τ_2 connecting the same start and end coordinates, $\mathbf{x}_s$ and $\mathbf{x}_g$ respectively, are homotopic iff one can be continuously deformed into the other without intersecting any obstacle.*

   Formally, if $\tau_1 : [0,1] \to \mathcal{C}$ and $\tau_2 : [0,1] \to \mathcal{C}$ represent the two trajectories (with $\tau_1(0) = \tau_2(0) = \mathbf{x}_s$ and $\tau_1(1) = \tau_2(1) = \mathbf{x}_g$), then τ_1 is homotopic to τ_2 iff there exists a continuous map $\eta : [0,1] \times [0,1] \to \mathcal{C}$ such that $\eta(\alpha, 0) = \tau_1(\alpha) \, \forall \alpha \in [0,1], \;\; \eta(\beta, 1) = \tau_2(\beta) \, \forall \beta \in [0,1]$, and $\eta(0, \gamma) = \mathbf{x}_s, \eta(1, \gamma) = \mathbf{x}_s \, \forall \gamma \in [0,1]$. Alternatively, in the notation of [14], τ_1 and τ_2 are homotopic iff $\tau_1 \sqcup -\tau_2$ belongs to the trivial class of the first homotopy group of $\mathcal{C}$, denoted by $\pi_1(\mathcal{C})$. That is, $[\tau_1 \sqcup -\tau_2] = \mathbf{0} \in \pi_1(\mathcal{C})$.

Definition 2 (Homologous trajectories). *Two trajectories τ_1 and τ_2 connecting the same start and end coordinates, $\mathbf{x}_s$ and $\mathbf{x}_g$ respectively, are homologous iff τ_1 together with τ_2 (the later with opposite orientation) forms the complete boundary of a 2-dimensional manifold embedded in $\mathcal{C}$ not containing/intersecting any of the obstacles.*

   Formally, in the notation of [14], τ_1 and τ_2 are homologous iff $\tau_1 \sqcup -\tau_2$ belongs to the trivial class of the first homology group of $\mathcal{C}$, denoted by $H_1(\mathcal{C})$. That is, $[\tau_1 \sqcup -\tau_2] = \mathbf{0} \in H_1(\mathcal{C})$.

   A set of homotopic trajectories form a homotopy class, while a set of homologous trajectories form a homology class.

   At an intuitive level the above two definitions may appear equivalent. For example, in Figure 1(a), τ_1 is homotopic to τ_2 since one can be continuously deformed into the other via a sequence of trajectories marked by the dashed curves. As a consequence, the area swept by this continuous deformation, A, forms a 2-dimensional region in the free configuration space whose boundary is the closed loop $\tau_1 \sqcup -\tau_2$. Indeed, the one-way implication is true as shown below.

Lemma 1. *If two trajectories are homotopic, they are homologous.*

Proof. This follows directly from the Hurewicz theorem [14] that guarantees the existence of an homomorphism from the homotopy groups to the homology groups of an arbitrary space. □   □

   The converse of Lemma 1 does not always hold true. There are subtle difference between homology and homotopy in spite of their similar notions, and one can create examples where two trajectories are not homotopic in spite of being homologous.

   Homotopy equivalence arises naturally in many robotics problems. On the other hand, homology is less natural. However, it is much simpler to compute homologies. One can establish direct correspondence between homology groups of trajectories and differential 1-forms whose integrals yield homology class invariants for trajectories via the De Rham theorem [7]. Since, according to the discussion of Section 1.2, we desire such differential forms, the rest of the paper will be developed with homology classes of trajectories under consideration rather than their homotopy classes. The assumption will be that in many of the practical robotics problems where homotopy classes of trajectories are of greater significance, homology classes of trajectories will serve as a fair analog. We will justify this claim in Section 5.1 and through experimental results (Section 6).

(a) τ_1 is homotopic to τ_2 since there is a continious 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.

Figure 1: Illustration of homotopy and homology equivalences. In this example τ_1 and τ_2 are both homotopic as well as homologous.

(a) In 2-dimensions.

(b) In 3-dimensions.
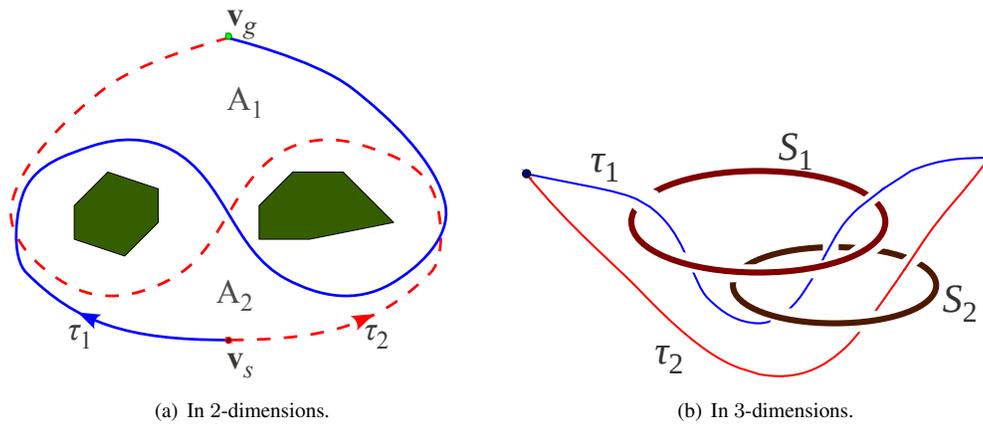
Figure 2: Examples where the trajectories are homologous, but not homotopic.

To clarify the distinction between homotopy equivalence and homology equivalence of trajectories, we present two examples where homology is not same as homotopy. The first example is in 2-dimensions. In Figure 2(a) we observe that the trajectories τ_1 and τ_2 are not homotopic, but they are homologous (since their H-signatures, as defined in Section 3.2, are equal). This is seen perhaps more easily by considering the interior defined by the union of the areas marked by A_1 and A_2 which indeed forms the boundary for $\tau_1 \sqcup -\tau_2$. In Figure 2(b), one can observe that the two trajectories are not homotopic. However, they are homotopic if we only consider S_1 or S_2 but not both. Hence their H-signatures are the same (*i.e.* they are homologous). Thus, if we were exploring different homotopy classes in this environment using the described method, we would be finding one trajectory for these two homotopy classes.

(a) The integrals over contours γ_1 and γ_2 are equal.

(b) Only the poles enclosed by γ influence the value of the integral of F.
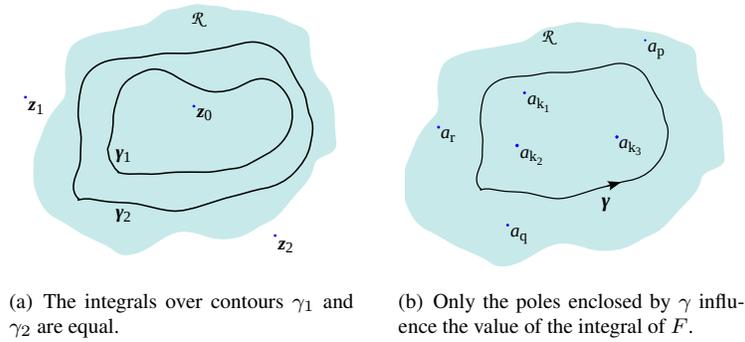
Figure 3: Cauchi Integral Theorem and Residue Theorem.

3 H-signature in 2-dimensional Euclidean Configuration Space

We consider a 2-dimensional subset of $\mathbb{R}^2$ as the configuration space. The obstacles are thus *punctures* or discontinuities in that subset. The approach for designing a H-signature for such a 2-dimensional configuration space is based on theorems from Complex Analysis, specifically the Cauchy Integral theorem and Residue theorem.

3.1   Background: Complex Analysis

Cauchy Integral Theorem.   The Cauchy Integral Theorem states that if $f : \mathbb{C} \to \mathbb{C}$ is an holomorphic (analytic) function in some simply connected region $\mathcal{R} \subset \mathbb{C}$, and γ is a closed oriented (i.e. directed) contour completely contained in $\mathcal{R}$, then the following holds,

$$\oint_\gamma f(z)\mathrm{d}z = 0 \tag{1}$$

Moreover, if z_0 is a point inside the region enclosed by γ, which has an anti-clockwise (or positive) orientation, then for the function $F(z) = f(z)/(z - z_0)$ with a simple pole at z_0, the following holds

$$\oint_\gamma \frac{f(z)\mathrm{d}z}{z - z_0} = 2\pi i f(z_0) \tag{2}$$

The Residue Theorem.   A direct consequence of the Cauchy Integral Theorem, the Residue Theorem, states that, if $F : \mathcal{R} \to \mathbb{C}$ is a function defined in some simply connected region $\mathcal{R} \subset \mathbb{C}$ that has simple poles at the distinct points $a_1, a_2, \cdots, a_M \in \mathcal{R}$, and holomorphic (analytic) everywhere else in $\mathcal{R}$, and say γ is a closed positively oriented Jordan curve completely contained in $\mathcal{R}$ and enclosing only the points $a_{k_1}, a_{k_1}, \cdots, a_{k_m}$ out of the poles of F, then the following holds,

$$\oint_\gamma F(z)\mathrm{d}z = 2\pi i \sum_{l=1}^{m} \lim_{\xi \to a_{k_l}} (\xi - a_{k_l})F(\xi) \tag{3}$$

The scenario is illustrated in Figure 3(b).

It is important to note that in both the Cauchy Integral Theorem and the Residue Theorem the value of the integrals are independent of the exact choice of the contour γ as long as the mentioned conditions are satisfied (see Figure 3(a)).

3.2   Designing a H-signature

We exploit the above theorems for designing a differential 1-form that can be used to construct a homology class invariant for 2-dimensional configuration space.

We start by representing the 2-dimensional configuration space as a subset of the complex plane $\mathbb{C}$. Thus a point in the configuration space, $(x, y) \in \mathcal{C}$, is represented as $x + iy \in \mathbb{C}$. The obstacles are assumed to be simply-connected regions in $\mathbb{C}$ and are represented by $\mathcal{O}_1, \mathcal{O}_2, \cdots, \mathcal{O}_N$.

Construction 1 [Representative points]   We define one "representative point" in each connected obstacle such that it lies in the interior of the obstacle. The exact location of the representative points is not of particular significance

as long as they each lie inside the respective obstacles. Thus we define the points $\zeta_l \in \mathcal{O}_l$, $\forall l = 1, \cdots, N$. Figure 4(a) shows such representative points inside three obstacles.

Definition 3 (Obstacle Marker Function). *For a given set of "representative points", we define the "Obstacle Marker Function" function $\mathcal{F} : \mathbb{C} \to \mathbb{C}^N$ as follows,*

$$\mathcal{F}(z) = \begin{bmatrix} \frac{f_1(z)}{z-\zeta_1} \\ \frac{f_2(z)}{z-\zeta_2} \\ \vdots \\ \frac{f_N(z)}{z-\zeta_N} \end{bmatrix} \tag{4}$$

where f_l, $l = 1, 2, \cdots, N$ are analytic functions over entire $\mathbb{C}$ such that $f_l(\zeta_l) \neq 0$, $\forall l$. Typical examples of such f_l are polynomials in z.

Thus, $\mathcal{F}$ is a complex vector function, the lth component of which has a single simple pole/singularity at ζ_l.

Definition 4 (H-signature in 2-dimensional configuration space). *For the given configuration space and set of obstacles, we define the* obstacle marker function *as described above, and hence define the H-signature of a trajectory τ the vector function $\mathcal{H}_2 : C_1(\mathbb{C}) \to \mathbb{C}^N$*

$$\mathcal{H}_2(\tau) = \int_\tau \mathcal{F}(z) dz$$

where $C_1(\mathbb{C})$ is the set of all curves/trajectories in $\mathbb{C}$.

It is to be noted that the value of the H-signature of a trajectory in the 2-dimensional configuration space is simply a vector of N complex numbers.

Lemma 2. *Two trajectories τ_1 and τ_2 connecting the same points in the described 2-dimensional configuration space are homologous if and only if $\mathcal{H}_2(\tau_1) = \mathcal{H}_2(\tau_2)$*

Proof. We note that by changing the orientation of a path over which an integration is being performed, we change the sign of the integral. If τ is a path, its oppositely oriented path is represented as $-\tau$. Thus, as we see from Figure 4(a), τ_1 along with $-\tau_2$ forms a positively oriented closed loop.

If τ_1 and τ_2 are in the same homology class, the area enclosed by τ_1 and τ_2 does not contain any of the "representative points", ζ_i, hence rendering the function $\mathcal{F}$ analytic in that region. Hence from the Cauchy Integral Theorem we obtain,

$$\int_{\tau_1 \sqcup -\tau_2} \mathcal{F}(z) \mathrm{d}z = \mathbf{0}$$
$$\Rightarrow \int_{\tau_1} \mathcal{F}(z) \mathrm{d}z + \int_{-\tau_2} \mathcal{F}(z) \mathrm{d}z = \mathbf{0}$$
$$\Rightarrow \int_{\tau_1} \mathcal{F}(z) \mathrm{d}z = \int_{\tau_2} \mathcal{F}(z) \mathrm{d}z$$

where the $\mathbf{0}$ in bold implies that it is a N-vetor of zeros.

If τ_1 and τ_2 are in different homology classes, we can easily note that the closed positive contour formed by τ_1 and $-\tau_2$ will enclose one or more of the obstacles, and hence their corresponding "representative points". This is illustrated in Figure 4(b). Let us assume that enclosed "representative points" are $\zeta_{\kappa_1}, \zeta_{\kappa_2}, \cdots, \zeta_{\kappa_n}$. Moreover we note that at least one component of the vector function $\mathcal{F}$ has a simple pole at ζ_l for each $l = 1, 2, \cdots, N$. Thus, by
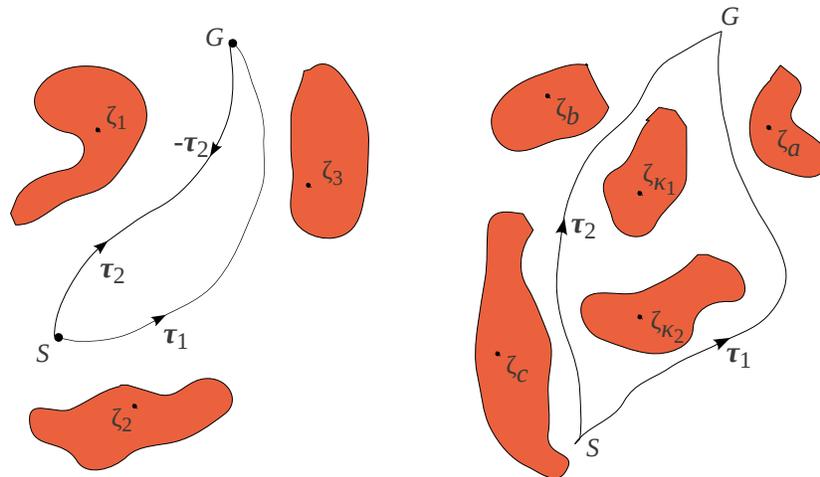
(a) In same Homotopy class, forming a closed contour.   (b) In different Homotopy classes, enclosing obstacles.

Figure 4: Two trajectories in same and different homotopy classes.

the Residue Theorem and Definition 3,

$$\int_{\tau_1} \mathcal{F}(z)\mathrm{d}z \; + \; \int_{-\tau_2} \mathcal{F}(z)\mathrm{d}z \;\; =$$

$$2\pi i \sum_{u=1}^{n} \lim_{\xi \to \zeta_{\kappa_u}} \left(\xi - \zeta_{\kappa_u}\right) \begin{bmatrix} \frac{f_1(\xi)}{\xi-\zeta_1} \\ \frac{f_2(\xi)}{\xi-\zeta_2} \\ \vdots \\ \frac{f_N(\xi)}{\xi-\zeta_N} \end{bmatrix}$$

$$\Rightarrow \quad \int_{\tau_1} \mathcal{F}(z)dz - \int_{\tau_2} \mathcal{F}(z)dz \;\; = \;\; \begin{bmatrix} \cdots \\ f_{\kappa_1}(\zeta_{\kappa_1}) \\ \vdots \\ f_{\kappa_2}(\zeta_{\kappa_2}) \\ \vdots \\ f_{\kappa_n}(\zeta_{\kappa_n}) \\ \cdots \end{bmatrix} \neq \mathbf{0}$$

Hence proved.   □

We have hence shown that $\mathcal{H}_2$ gives a *homology invariant* for trajectories in 2-dimensional Euclidean configuration space with obstacles.

3.3   Computation for a Line Segment

As discussed earlier in Section 1.2, and will be discussed later in Section 5, we discretized the given configuration space and create a graph out of it. In many practical implementations we assume that every edge in the graph is a line segment. Thus it is for those line segments that we really need to compute the H-signatures. Thus it is important that we are able to do so efficiently. In this section we will show how to compute the H-signature for a *small* line segment in a 2-dimensional configuration space using a closed-form formula.

Given a line segment e connecting points z_1 and z_2, we can parametrize the segment using the variable $z = (1 - \lambda)z_1 + \lambda z_2$, where $\lambda \in [0, 1]$ is the parameter. Thus we have,

$$\begin{aligned} \mathcal{H}_2(e) \;\; &= \;\; \int_e \mathcal{F}(z)\mathrm{d}z \\ &= \;\; \int_0^1 \mathcal{F}\Big((1 - \lambda)z_1 + \lambda z_2\Big)(z_2 - z_1)\,\mathrm{d}\lambda \end{aligned} \tag{5}$$

For computing the H-signature of $e = \{z_1 \rightarrow z_2\}$ analytically, we assume that f_l are chosen to be constants. Let $f_l = A_l$ *(const.)* for all $l = 1, 2, \cdots, N$.

Now, a standard integration result gives for the lth component of $\mathcal{H}_2(e)$

$$\int_0^1 \frac{A_l}{(1 - \lambda)z_1 + \lambda z_2 - \zeta_l} (z_2 - z_1) \, \mathrm{d}\lambda \quad = \quad A_l \left(\ln(z_2 - \zeta_i) - \ln(z_1 - \zeta_l) \right)$$

However we note that the *logarithm of a complex number* does not have an unique value. For any $z' \in \mathbb{C}$, $\ln(z') = \ln(|z'| e^{i(\arg(z')+2k\pi)}) = \ln(|z'|) + i(\arg(z') + 2k\pi)$, $\forall k = 0, \pm 1, \pm 2, \ldots$ (where $\arg(x + iy) = \mathrm{atan2}(y, x)$). Hence, following the assumption that e is a small line segment, we choose the *smallest* of all the possible values over different k's. Thus, the lth component of $\mathcal{H}_2(e)$ is computed as,

$$A_l \left[\ln(|z_2 - \zeta_l|) - \ln(|z_1 - \zeta_l|) \quad + \quad i \ \mathrm{absmin}_{k \in \mathbb{Z}} \left(\arg(z_2 - \zeta_l) - \arg(z_1 - \zeta_l) + 2k\pi \right) \right]$$

where $\mathrm{absmin}_{k \in \mathbb{Z}} G(k)$ returns the value of $G(k)$ that has the minimum absolute value (*i.e.* closest to 0) over all $k \in \mathbb{Z}$. Typically, we can get away with checking a few values of k around 0 and picking the local minimum, since the value of $\arg(z_2 - \zeta_l) - \arg(z_1 - \zeta_l) + 2k\pi$ is monotonic in k.

4 H-signature in 3-dimensional Euclidean Configuration Space

While in the two-dimensional case, theoretically any finite obstacle on the plane can induce multiple homotopy and homology classes for trajectories joining two points, the notion of homotopy/homology classes in three dimensions can only be induced by obstacles with *genus* [2] one or more, or with obstacles stretching to infinity. Figure 6 shows some examples of obstacles that can or cannot induce such classes for trajectories. A sphere or a solid cube, for example, cannot induce multiple homotopy classes in an environment.

4.1 Background: Electromagnetism

Biot-Savart law. Consider a single hypothetical current-carrying *curve* (a current conducting wire) embedded in a 3-dimensional space carrying a steady current of unit magnitude (Figure 5(a)). There is no source for the current nor any sink - only a steady flow persisting inside the conductor due to absence of any dissipation. It is to be noted that such a steady current is possible iff the *curve* is closed (or open, but extending to infinity, where we close the curve using a loop at infinity. See Figure 7(a) and Construction 2). We denote the curve by $\mathcal{S}$. Then, according to the Biot-Savart Law [11], the magnetic field $\mathbf{B}$ at any arbitrary point $\mathbf{r}$ in the space, due to the current flow in $\mathcal{S}$, is given by,

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi} \int_{\mathcal{S}} \frac{(\mathbf{x} - \mathbf{r}) \times \mathrm{d}\mathbf{x}}{\|\mathbf{x} - \mathbf{r}\|^3} \tag{6}$$

where, $\mathbf{x}$, the integration variable, represents the coordinate of a point on $\mathcal{S}$, and $\mathrm{d}\mathbf{x}$ is an infinitesimal element on $\mathcal{S}$ along the direction of the current flow.

Ampere's Law. While Biot-Savart law gives a recipe for computing the magnetic field from a given current configuration, Ampere's Law [11], in a sense, gives the inverse of it. Given the magnetic field $\mathbf{B}$ at every point in the space, and a closed loop γ (Figure 5(a)), the line integral of $\mathbf{B}$ along γ gives the current *enclosed* by the loop γ. That is,

$$\Xi(\mathcal{C}) := \int_{\gamma} \mathbf{B}(\mathbf{l}) \cdot \mathrm{d}\mathbf{l} = \mu_0 I_{encl} \tag{7}$$

where, $\mathbf{l}$, the integration variable, represents the coordinate of a point on γ, and $\mathrm{d}\mathbf{l}$ is an infinitesimal element on $\mathcal{C}$.

In Biot-Savart Law and Ampere's Law one can conveniently choose the constant μ_0 to be equal to 1 by proper choice of units. Moreover, by choice, the value of the current flowing in the conductor is unity. Thus, for any closed loop γ, the value of $\Xi(\gamma)$ is zero iff γ does not enclose the conductor, otherwise it is ± 1 (the sign depends on the direction of integration performed on γ). Thus in Figure 5(a), $\Xi(\gamma_1) = 1$ and $\Xi(\gamma_2) = 0$.

---

[2] The *genus* of an obstacle refers to the number of *holes* or *handles* [17].

(a) Magnetic field due to current in $\mathcal{S}$, & its integration along closed loop γ_i.

(b) 2 trajectories, τ_1 & τ_2, connecting the same points form a closed loop.
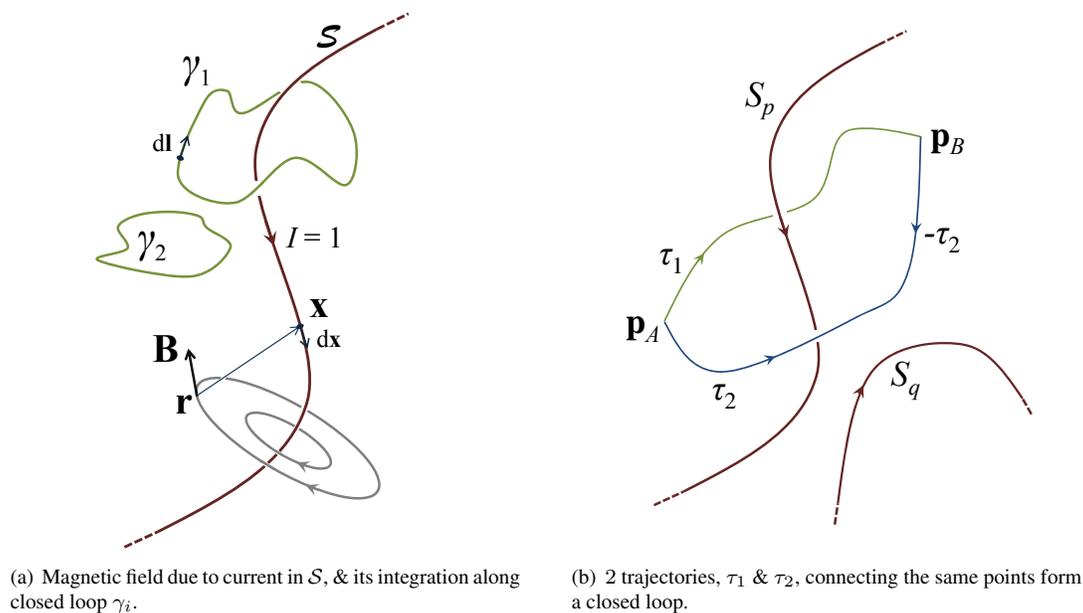
Figure 5: Theorems from electromagnetism, and their application in defining H-signature in 3-dimensions.

(a) Skeleton of a generic genus 1 obstacle is modeled as a current-carrying conductor.

(b) A torus-shaped genus 1 obstacle.

(c) A genus 2 obstacle.

(d) An infinite tube is a genus 1 obstacle.

(e) A knot-shaped obstacle with genus 1.

(f) A sphere **does not** induce homotopy classes and has genus 0.
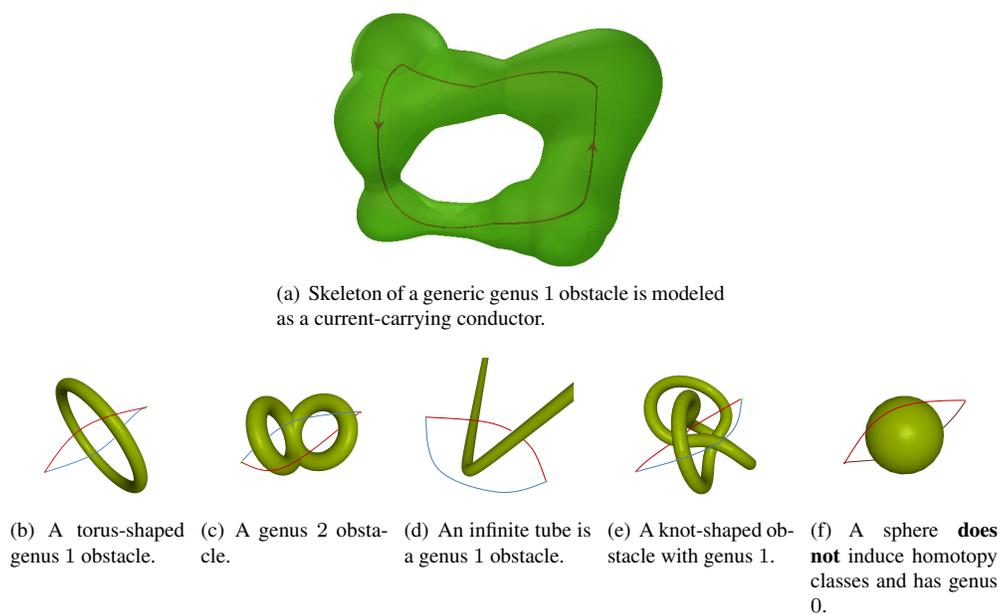
Figure 6: Examples of obstacles in 3-D. (a-e) induce homotopy classes, (f) does not.

Definition 5 (Simple Homotopy-Inducing Obstacle in 3-dimensional Configuration Space). *A Simple Homotopy-inducing Obstacle (SHIO) is a bounded obstacle of* genus 1*, for example a torus (Figure 6(a), 6(b)) or a knot (Figure 6(e)).*

(a) An unbounded obstacle and its skeleton can be closed at a large distance to create a *closed loop*.

(b) An obstacle with genus 2, $\mathcal{O}$, can be decomposed into 2 obstacles, each with genus one, $\mathcal{O}_1$ and $\mathcal{O}_2$.
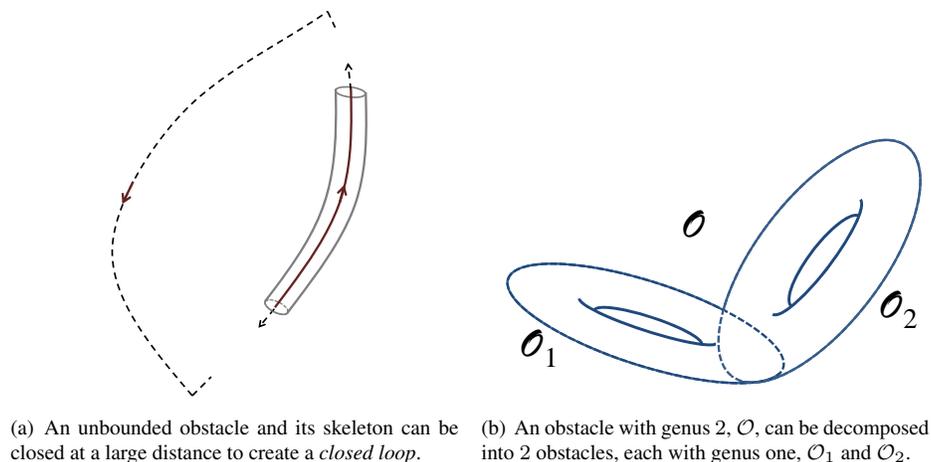
Figure 7: Illustration of Constructions 2 and 3.

4.2   Designing a H-signature

For the 2-dimensional case, each obstacle on the plane that induces the notion of multiple homotopy classes was assigned a *representative point*. Analogously, for the 3-dimensional case, we need to define a *skeleton* for every SHIO. Intuitively, a skeleton of a 3-dimensional obstacle is a 1-dimensional curve that is completely contained inside the obstacle such that the surface of the obstacle can be "shrunk" onto the skeleton in a continuous fashion without altering the topology of the surface of the obstacle. Formally, we define the skeleton of an obstacle in terms of *homotopy equivalence*.

Definition 6 (Skeleton). *A* 1-*dimensional manifold, S, is called a* skeleton *of a SHIO, $\mathcal{O}$, iff S is homeomorphic to $\mathbb{S}^1$ (a circle), S is completely contained inside $\mathcal{O}$, and if S and $\mathcal{O}$ are* homotopy equivalent.

Thus, the fact that τ_1 and τ_2 are of the same or of different homotopy/homology classes is not altered by replacing $\mathcal{O}$ by S.

In the literature, algorithms for constructing skeletons of solid objects is a well-studied [6, 16]. However in the present context we have a much relaxed notion of skeleton. While we can adopt any of the different existing algorithms for automated construction of skeleton from a 3-dimensional obstacles, this discussion is out of the scope of the present work. Figure 6(a) demonstrates skeletons for several genus 1 obstacles.

4.2.1   Conversion of generic obstacles into SHIOs

Given a set of obstacles in a three-dimensional environment, we perform the following two constructions/reduction on the obstacles so that the only kind of obstacle we have in the environment are *Simple Homotopy-Inducing Obstacles*.

Construction 2 [Closing infinite, unbounded obstacles] In most of the problems that we are concerned with, the domain in which the trajectories of the robots lie is finite and bounded. This gives us the freedom of altering/modifying the obstacles or parts of obstacles lying outside that domain without altering the problem. One consequence of this freedom is that we can *close* infinite and unbounded obstacles (*e.g.* Figure 6(d)) at a large distance from the domain of interest (Figure 7(a)).

Construction 3 [Decomposing obstacles with genus > 1] After closing all infinite, unbounded obstacles in an environment according to Construction 2, if there is an obstacle with genus k (*e.g.* Figure 6(c)), we can decompose/split it into k obstacles, possibly overlapping and touching each other, but each with genus 1 (Figure 7(b)). This does not change the obstacles or the problem in any way. This construction just changes the way we identify obstacles and construct their skeletons. For example in Figure 7(b) the original obstacle $\mathcal{O}$ with genus 2 is realized as two obstacles $\mathcal{O}_1$ and $\mathcal{O}_2$, each with genus 1 and overlapping each other. The decomposition of obstacles into SHIOs allows us define k skeletons for each obstacle of genus k and simplify computations of h-signatures of trajectories.

Note that in this paper we do both constructions manually — the automation of these steps is beyond the scope of this paper.

4.2.2 Skeleton of SHIOs as Current Carrying Curves for H-signature Construction

Construction 4 [Modeling *skeleton* of a SHIO as a current carrying manifold] Given m obstacles in an environment, $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_m$, with genus $k_1, k_2, \ldots, k_m$ respectively, we can construct $M = k_1 + k_2 + \cdots + k_m$ skeletons from M SHIOs (obtained using Constructions 2 and 3), namely $S_1, S_2, \ldots, S_M$. Each S_i is a closed, connected, boundary-less 1-dimensional manifold. We model each of them as a current-carrying conductor carrying current of unit magnitude (Figures 6(a), 7(a)). The direction of the currents is not of importance, but by convention, each is of unit magnitude.

Definition 7 (Virtual Magnetic Field due to a Skeleton). *Given S_i, the skeletons of a Simple Homotopy-Inducing Obstacle, we define a Virtual Magnetic Field vector at a point $\mathbf{r}$ in the space due to the current in S_i using Biot-Savart Law as follows,*

$$\mathbf{B}_i(\mathbf{r}) = \frac{1}{4\pi} \int_{S_i} \frac{(\mathbf{x} - \mathbf{r}) \times d\mathbf{x}}{\|\mathbf{x} - \mathbf{r}\|^3} \tag{8}$$

where, $\mathbf{x}$, the integration variable, represents the coordinates of a point on S_i, and $d\mathbf{x}$ is an infinitesimal element on S_i along the chosen direction of the current flow in S_i.

Definition 8 (H-signature in 3-dimensional Configuration Space). *Given an arbitrary trajectory, τ, in the 3-dimensional environment with M skeletons, we define the H-signature of τ to be the function $\mathcal{H}_3 : C_1(\mathbb{R}^3) \to \mathbb{R}^M$,*

$$\mathcal{H}_3(\tau) = [h_1(\tau), \; h_2(\tau), \; \ldots, \; h_M(\tau)]^T \tag{9}$$

where, $C_1(\mathbb{R}^3)$ is the space of all curves/trajectories in $\mathbb{R}^3$, and

$$h_i(\tau) = \int_\tau \mathbf{B}_i(\mathbf{l}) \cdot d\mathbf{l} \tag{10}$$

is defined in an analogous manner as the integral in Ampere's Law. In defining h_i, $\mathbf{B}_i$ is the Virtual Magnetic Field vector due to the unit current through skeleton S_i, $\mathbf{l}$ is the integration variable that represents the coordinate of a point on τ, and $d\mathbf{l}$ is an infinitesimal element on τ.

It is to be noted that the value of the H-signature of a trajectory in the 3-dimensional configuration space is simply a vector of M real numbers.

Lemma 3. *Two trajectories τ_1 and τ_2 connecting the same points in the described 3-dimensional configuration space are homologous if and only if $\mathcal{H}_3(\tau_1) = \mathcal{H}_3(\tau_2)$.*

Proof. Since τ_1 and τ_2 connect the same points, $\tau_1 \sqcup -\tau_2$, *i.e.* τ_1 and $-\tau_2$ together (where $-\tau$ indicates the same curve as τ, but with the opposite orientation) form a closed loop in the 3-dimensional environment (Figure 5(b)). We replace the obstacles $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_m$ in the environments with the skeletons $S_1, S_2, \ldots, S_M$.

Consider the presence of just the skeleton S_i. By the direct consequence of Ampere's Law and our construction in which a unit current flows through S_i, the value of

$$h_i(\tau_1 \sqcup -\tau_2) = \int_{\tau_1 \sqcup -\tau_2} \mathbf{B}_i(\mathbf{l}) \cdot d\mathbf{l}$$

is non-zero if and only if the closed loop formed by $\tau_1 \sqcup -\tau_2$ encloses the current carrying conductor S_i (*i.e.* there does not exist a surface not intersecting S_i, the boundary of which is $\tau_1 \sqcup -\tau_2$). For example, in Figure 5(b), $h_p(\tau_1 \sqcup -\tau_2) = 1$ and $h_q(\tau_1 \sqcup -\tau_2) = 0$. Now, by the definition of line integration we have the following identity,

$$\begin{aligned} h_i(\tau_1 \sqcup -\tau_2) &= \int_{\tau_1 \sqcup -\tau_2} \mathbf{B}_i(\mathbf{l}) \cdot d\mathbf{l} \\ &= \int_{\tau_1} \mathbf{B}_i(\mathbf{l}) \cdot d\mathbf{l} - \int_{\tau_2} \mathbf{B}_i(\mathbf{l}) \cdot d\mathbf{l} = h_i(\tau_1) - h_i(\tau_2) \end{aligned} \tag{11}$$

Thus, $h_i(\tau_1) = h_i(\tau_2)$ if and only if the closed loop formed by τ_1 and τ_2 does not enclose S_i (*i.e.* homologous in presence of S_i).

Now in presence of skeletons $S_1, S_2, \ldots, S_M$ the same argument extends for each skeleton individually. Thus τ_1 and τ_2 are homologous if an only if $\mathcal{H}_3(\tau_1) = \mathcal{H}_3(\tau_2)$. $\qquad\square$

Hence we have shown that the proposed formula for H-signature is a homology class invariant for trajectories in 3-D.

(a) A skeleton of an obstacle can be constructed/approximated so that it is made up of n line segments.

(b) Magnetic field at $\mathbf{r}$ due to the current in a line segment $\overrightarrow{\mathbf{s}_i^j \mathbf{s}_i^{j'}}$.
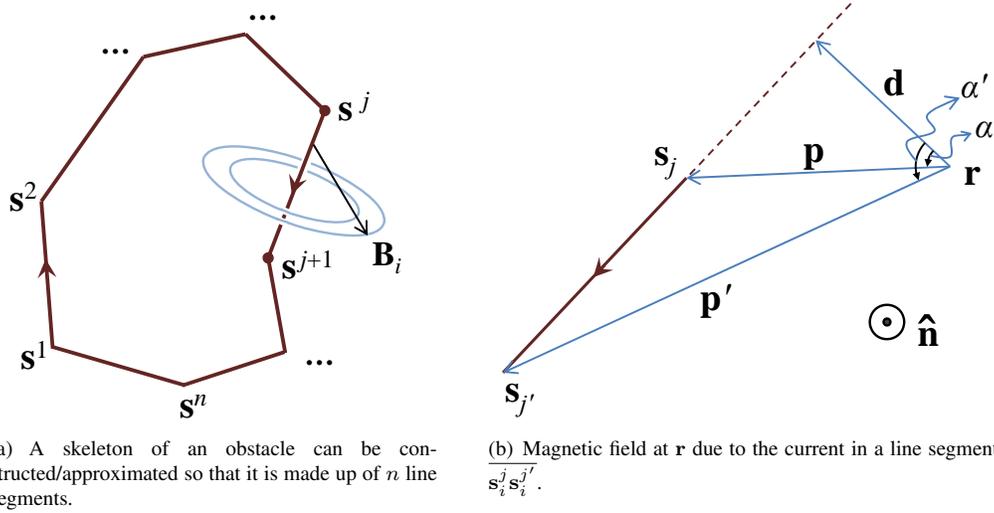
Figure 8: Closed-form, analytic computation of virtual magnetic field vector.

4.3   Computation for a Line Segment

Once again, we are interested in efficient computation of the H-signature for small line segments since those are the ones that will make up edges of the graph formed by discretization of the environment. For all practical applications we assume that a skeleton of an obstacle, S_i, is made up of finite number (n_i) of line segments: $S_i = \{\overrightarrow{\mathbf{s}_i^1 \mathbf{s}_i^2}, \; \overrightarrow{\mathbf{s}_i^2 \mathbf{s}_i^3}, \; \ldots, \; \overrightarrow{\mathbf{s}_i^{n_i-1} \mathbf{s}_i^{n_i}}, \; \overrightarrow{\mathbf{s}_i^{n_i} \mathbf{s}_i^1}\}$ (Figure 8(a)). Thus, the integration of equation (8) can be split into summation of n_i integrations,

$$\mathbf{B}_i(\mathbf{r}) = \frac{1}{4\pi} \sum_{j=1}^{n_i} \int_{\overrightarrow{\mathbf{s}_i^j \mathbf{s}_i^{j'}}} \frac{(\mathbf{x} - \mathbf{r}) \times d\mathbf{x}}{\|\mathbf{x} - \mathbf{r}\|^3} \tag{12}$$

where $j' \equiv 1 + (j \mod n_i)$. It is to be noted that a skeleton of an unbounded obstacle created from Construction 2 can be made up of finite and few line segments. The only feature of such a skeleton might be that some of the points that make up the line segments ($\mathbf{s}_i^j$) might be located at a large distance from the domain of interest, which is used to close the skeleton.

One advantage of this representation of skeletons is that for the straight line segments, $\overrightarrow{\mathbf{s}_i^j \mathbf{s}_i^{j'}}$, the integration can be computed analytically. Specifically, using a result from [11] (also, see Figure 8(b)),

$$\int_{\overrightarrow{\mathbf{s}_i^j \mathbf{s}_i^{j'}}} \frac{(\mathbf{x} - \mathbf{r}) \times d\mathbf{x}}{\|\mathbf{x} - \mathbf{r}\|^3} = \frac{1}{\|\mathbf{d}\|} \left(\sin(\alpha') - \sin(\alpha) \right) \hat{\mathbf{n}}$$

$$= \frac{1}{\|\mathbf{d}\|^2} \left(\frac{\mathbf{d} \times \mathbf{p}'}{\|\mathbf{p}'\|} - \frac{\mathbf{d} \times \mathbf{p}}{\|\mathbf{p}\|} \right) \tag{13}$$

where, $\mathbf{d}$, $\mathbf{p}$ and $\mathbf{p}'$ are functions of $\mathbf{s}_i^j, \mathbf{s}_i^{j'}$ and $\mathbf{r}$ (Figure 8(b)), and can be expressed as,

$$\mathbf{p} = \mathbf{s}_i^j - \mathbf{r}, \;\; \mathbf{p}' = \mathbf{s}_i^{j'} - \mathbf{r}, \;\; \mathbf{d} = \frac{(\mathbf{s}_i^{j'} - \mathbf{s}_i^j) \times (\mathbf{p} \times \mathbf{p}')}{\|\mathbf{s}_i^{j'} - \mathbf{s}_i^j\|^2} \tag{14}$$

We define and write $\mathbf{\Phi}(\mathbf{s}_i^j, \mathbf{s}_i^{j'}, \mathbf{r})$ for the RHS of Equation (13) for notational convenience. Thus we have,

$$\mathbf{B}_i(\mathbf{r}) = \frac{1}{4\pi} \sum_{j=1}^{n_i} \mathbf{\Phi}(\mathbf{s}_i^j, \mathbf{s}_i^{j'}, \mathbf{r}) \tag{15}$$
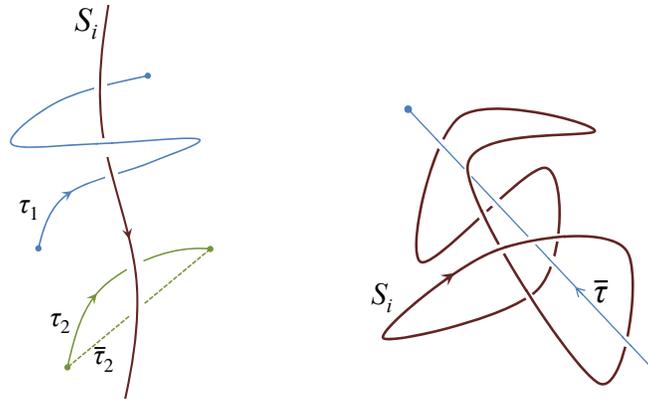
(a) Trajectory that loops around a skeleton & one that   (b) In the most general case, it is difficult to precisely
doesn't. In this figure $h_i(\tau_1) > 1$ and $0 < h_i(\tau_2) =$ identify a *non-looping* homotopy class.
$h_i(\overline{\tau}_2) < 1$.

Figure 9: Looping vs. non-looping trajectories.

where, $j' \equiv 1 + (j \mod n_i)$.

Given a small line segment, e, we can now compute the *H*-signature, $\mathcal{H}(e) = [h_1(e), \ h_2(e), \ \ldots, \ h_M(e)]^T$, where,

$$h_i(e) = \frac{1}{4\pi} \int_e \sum_{j=1}^{n_i} \boldsymbol{\Phi}(\mathbf{s}_i^j, \mathbf{s}_i^{j'}, \mathbf{l}) \cdot \mathbf{dl} \tag{16}$$

can be computed numerically. For the numerical integration, in all our experimental results we used the GSL (GNU Scientific Library), which has a highly efficient implementation of adaptive integration algorithms with desired precision. We used a cache for storing the H-signature of edges that has been computed in order to avoid re-computation.

4.4 'Looping' and 'Non-looping' Trajectories

"Looping" of a trajectory around an obstacle (Figure 9(a)) is similar in essence to *non-Jordan* curves on two-dimensional planes. However in three dimensions their precise definition is difficult. In fact, the notions of *looping* and *non-looping* is imprecise when the skeleton of the obstacle is complex as a knot (Figure 9(b)). However, equipped with the definition of H-signature, we propose the following definition. A more elaborate discussion on this can be found in [4].

Definition 9 (Non-looping trajectory w.r.t. S_i)**.** *A trajectory τ is said to be* non-looping *w.r.t. S_i if $h_i(\tau) \in (-1, 1)$. The value is in $[0, 1)$ if the trajectory goes around S_i in accordance with the* right-hand rule *with thumb pointing along the direction of the current in S_i. If the direction is opposite, the value lies in $(-1, 0]$.*

This definition, in many cases, conform to our general intuition of *non-looping* trajectories. A natural consequence of this definition is the notion of a trajectory that is in a "Complementary Class" of a non-looping trajectory, *i.e.* one that goes on the opposite side of every obstacle.

Definition 10 (Complementary Homotopy Class)**.** *Given a trajectory τ that is* non-looping *w.r.t. all the skeletons in the environment (i.e. $h_i(\tau) \in (-1, 1) \ \forall \ i = 1, 2, \ldots, M$), we define the* Complementary Homotopy Class *of the homotopy class of τ to be the one for which the* h*-signature is $\mathcal{H}(\tau) - sign(\mathcal{H}(\tau))$, where $sign(\cdot)$ gives the vector of signs of the elements of its input vector.*

5 H-signature Augmented Graph

Once we have the means of computing H-signature for each edge (small line segments), we introduce the concept of *H-signature augmented graph*. Typically, a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is created for the purpose of graph-search based planning
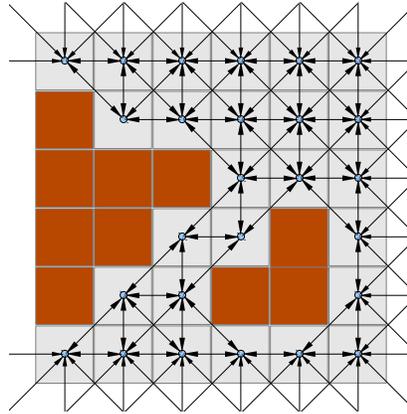
Figure 10: Graph formed by uniform discretization of configuration space and connecting each node with its neighbors. Dark red indicate location of obstacles.

by discretization of an environment, placing a vertex at each discretized cell, and by connecting the neighboring cells with edges (See Figure 10 for an example in 2-dimensional configuration space). In the following discussion we perform a construction without distinguishing between 2 and 3-dimensional configuration spaces explicitly, once we have discretized the environment, and perform a general treatment with the graph $\mathcal{G}$. For H-signature trajectories or line segments we use the generic function $\mathcal{H}$, which we understand to be $\mathcal{H}_2$ or $\mathcal{H}_3$ depending on the dimensionality of the configuration space.

Let $\mathbf{v}_s$ be the start coordinate in the configuration space, and $\mathbf{v}_g$ be the goal coordinate. By Lemma 2 or 3, any two trajectories from $\mathbf{v}_s$ to $\mathbf{v}$ that belong to the same homology class will have the same H-signature. The H-signature can assume different, but discrete values corresponding on the class of the trajectory. We also write $\mathcal{P}(\mathbf{v}_s, \mathbf{v})$ to denote the set of all trajectories from $\mathbf{v}_s$ to $\mathbf{v}$, and $\widetilde{\mathbf{v}_s\mathbf{v}} \in \mathcal{P}(\mathbf{v}_s, \mathbf{v})$ to denote a particular trajectory in that set.

Definition 11 (Allowed and Blocked Homology Classes). *Suppose it is required that we restrict all our search for trajectories connecting $\mathbf{v}_s$ and $\mathbf{v}_g$ to certain homology classes, or not allow some other. We denote the set of allowed* H-*signatures of trajectories leading up to $\mathbf{v}_g$ by the set $\mathcal{A}$, and the set of blocked* H-*signatures as $\mathcal{B}$. $\mathcal{A}$ and $\mathcal{B}$ are essentially complement of each other ($\mathcal{A} \cup \mathcal{B} = \mathcal{U}$, where the universal set, $\mathcal{U}$, is the set of the* H-*signatures of all the classes of trajectories joining $\mathbf{v}_s$ and $\mathbf{v}_g$), and $\mathcal{B}$ can be an empty set when all classes are allowed.*

We define the *H-signature augmented graph* of $\mathcal{G}$ as the graph $\mathcal{G}_H(\mathcal{G}) = \{\mathcal{V}_H, \mathcal{E}_H\}$, such that each node in this new graph has the *H*-signature of a trajectory leading up to the coordinate of the node from $\mathbf{v}_s$ appended to it. That is, each node in this augmented graph is given by $\{\mathbf{v}, \mathcal{H}(\widetilde{\mathbf{v}_s\mathbf{v}})\}$, for some $\widetilde{\mathbf{v}_s\mathbf{v}} \in \mathcal{P}(\mathbf{v}_s, \mathbf{v})$. Thus, corresponding to a given $\mathbf{v} \in \mathcal{V}$, since there are discrete homology classes of trajectories from $\mathbf{v}_s$ to $\mathbf{v}$, there are a discrete number of the augmented states, $\{\mathbf{v}, \mathbf{h}\} \in \mathcal{V}_H$, where $\mathbf{h}$ is a M-vector of reals (M being the number of *representative points* or the number of *SHIOs* depending on whether it's a 2 or 3-dimensional configuration space) and assumes the values of the H-signatures corresponding to the discrete homology classes. Thus, we define the *H-signature augmented graph* of $\mathcal{G}$ as follows,

$$\mathcal{G}_H = \{\mathcal{V}_H, \mathcal{E}_H\}$$

where,

1.

$$\mathcal{V}_H = \left\{ \{\mathbf{v}, \mathbf{h}\} \;\middle|\; \begin{array}{l} \mathbf{v} \in \mathcal{V}, \text{ and,} \\ \mathbf{h} = \mathcal{H}(\widetilde{\mathbf{v}_s\mathbf{v}}) \text{ for some trajectory} \\ \qquad \widetilde{\mathbf{v}_s\mathbf{v}} \in \mathcal{P}(\mathbf{v}_s, \mathbf{v}), \text{ and,} \\ \mathbf{h} \in \mathcal{A} \text{ (equivalently, } \mathbf{h} \notin \mathcal{B}) \\ \qquad \text{when } \mathbf{v} = \mathbf{v}_g \end{array} \right\}$$

2. An edge $\{\{\mathbf{v}, \mathbf{h}\} \to \{\mathbf{v}', \mathbf{h}'\}\}$ is in $\mathcal{E}_H$ for $\{\mathbf{v}, \mathbf{h}\} \in \mathcal{V}_H$ and $\{\mathbf{v}', \mathbf{h}'\} \in \mathcal{V}_H$, iff

    i. The edge $\{\mathbf{v} \to \mathbf{v}'\} \in \mathcal{E}$, and,
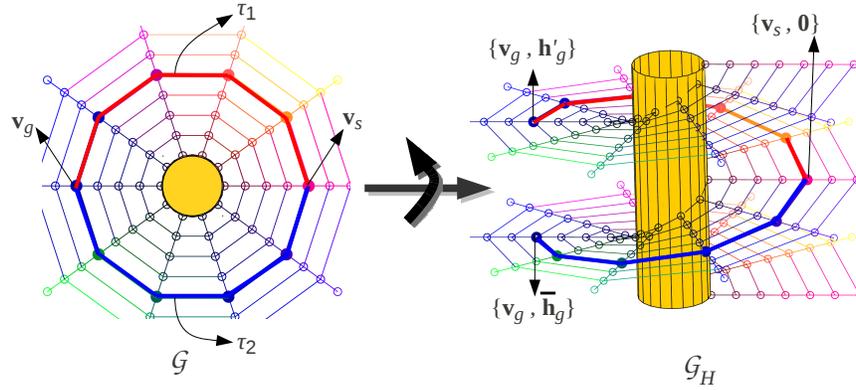
Figure 11: The topology of the augmented graph, $\mathcal{G}_H$ (right), compared against $\mathcal{G}$ (left), for a cylindrically discretized 2-dimensional configuration space around a circular obstacle

    ii. $\mathbf{h}' = \mathbf{h} + \mathcal{H}(\mathbf{v} \rightarrow \mathbf{v}')$, where, $\mathcal{H}(\mathbf{v} \rightarrow \mathbf{v}')$ is the *H-signature* of the edge $\{\mathbf{v} \rightarrow \mathbf{v}'\} \in \mathcal{E}$.

3. The cost/weight associated with an edge $\{\{\mathbf{v}, \mathbf{h}\} \rightarrow \{\mathbf{v}', \mathbf{h}'\}\}$ is same as that associated with edge $\{\mathbf{v} \rightarrow \mathbf{v}'\} \in \mathcal{E}$.

The consequence of point 3 in the above definition is that an *admissible heuristics* for search in $\mathcal{G}$ will remain admissible in $\mathcal{G}_H$. That is, if $f(\mathbf{v}, \mathbf{v}_g)$ was the heuristic function in $\mathcal{G}$, we define $f_H(\{\mathbf{v}, \mathbf{h}\}, \{\mathbf{v}_g, \mathbf{h}'\}) = f(\mathbf{v}, \mathbf{v}_g)$ as the heuristic function in $\mathcal{G}_H$ for any $\mathbf{h}' \in \mathcal{A}$.

    The consequence of augmenting each node of $\mathcal{G}$ with a H-signature is that now nodes are distinguished not only by their coordinates, but also the H-signature of the trajectory followed to reach it. Typically we use graph search algorithms like A* (or variants like D* or D*-lite) where nodes in the graph $\mathcal{G}_H$ are expanded starting from the node $\{\mathbf{v}_s, \mathbf{0}\}$ (where by $\mathbf{0}$ we mean a M-dimensional vector of zeros).

    The topology of this augmented graph for a 2-dimensional case is illustrated in Figure 11. A goal state $\mathbf{v}_g$ is the same in $\mathcal{G}$ irrespective for the path (τ_1 or τ_2) taken to reach it. Whereas in the H-signature augmented graph, the states are differentiated by the additional value of $\mathbf{h}_g$. We can perform a graph search in the augmented graph, $\mathcal{G}_H$, using any standard graph search algorithm starting from the state $\{\mathbf{v}_s, \mathbf{0}\}$. The goal state (i.e. the state, upon expansion of which we stop the graph search) is potentially any of the states $\{\mathbf{v}_g, \mathbf{h}_g\}$ for any $\mathbf{h}_g \in \mathcal{A}$ (or $\mathbf{h}_g \notin \mathcal{B}$ if $\mathcal{B}$ is provided instead of $\mathcal{A}$). We can use the same heuristic that we would have used for searching in $\mathcal{G}$, i.e. $f_H(\mathbf{v}, \mathbf{h}) = f(\mathbf{v})$. It is to be noted that $\mathcal{G}_H$ is essentially an infinite graph, even if $\mathcal{G}$ is finite. However the search algorithm needs to expand only a finite number of states. Since for a given $\mathbf{v}$, the states $\{\mathbf{v}, \mathbf{h}\}$ can assume some discrete values of $\mathbf{h}$ (corresponding to the different homology classes). To determine if $\{\mathbf{v}, \mathbf{h}'_g\}$ and $\{\mathbf{v}, \overline{\mathbf{h}}_g\}$ are the same states, we can simply compare the values of $\mathbf{h}'_g$ and $\overline{\mathbf{h}}_g$.

5.1   Uses of the H-signature Augmented Graph

There are primarily two distinct but related ways we would like to use the H-signature augmented graph with search algorithms:

    i. *Exploration of environment for different homotopy classes of trajectories connecting $\mathbf{v}_s$ and $\mathbf{v}_g$:* For this problem, whenever we expand a state $\{\mathbf{v}_g, \tilde{\mathbf{h}}\} \in \mathcal{V}_H$, for some $\tilde{\mathbf{h}} \notin \mathcal{B}$, we store the path up to that node, and continue expanding more states until the desired number of classes are explored. Although H-signature is a homology class invariant, and not a homotopy class invariant, by Lemma 1, two trajectories are homotopic implies that they are homologous. Thus, two trajectories that are homotopic will be in the same homology class, and hence their H-signatures will be the same. Thus, in such problems where we find least cost trajectories with different H-signatures in a configuration space using the said method, we are always guaranteed to obtain trajectories in distinct homotopy classes as well.

    ii. *Planning with H-signature constraint:* For searches with H-signature constraint, we stop upon expansion of a goal coordinate $\{\mathbf{v}_g, \tilde{\mathbf{h}}\}$ for some $\tilde{\mathbf{h}} \notin \mathcal{B}$ (or equivalently, $\tilde{\mathbf{h}} \in \mathcal{A}$).

5.2 Theoretical Analysis

Theorem 1. *If* $\mathbf{P}_H^* = \{ \{\mathbf{v}_1, \mathbf{h}_1\}, \{\mathbf{v}_2, \mathbf{h}_2\}, \cdots, \{\mathbf{v}_p, \mathbf{h}_p\} \}$ *is an optimal path in* $\mathcal{G}_H$, *then the path* $\mathbf{P}^* = \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_p\}$ *is an optimal path in the graph* $\mathcal{G}$ *satisfying the* H-*signature constraints specified by* $\mathcal{A}$ *and* $\mathcal{B}$

Proof. By construction of $\mathcal{G}_H$, the path $\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_p\}$ satisfies the given H-signature constraints. Moreover by definition, $\mathbf{P}_H^*$ is a minimum cost path in $\mathcal{G}_H$. Since the cost function in $\mathcal{G}_H$ is the same as the one in $\mathcal{G}$ and does not involve $\mathbf{h}_j$, it follows that the projection of $\mathbf{P}_H^*$ on $\mathcal{G}$ given by $\mathbf{P}^* = \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_p\}$ is an optimal path in the graph $\mathcal{G}$ satisfying the constraints defined in $\mathcal{G}_H$. $\qquad\square$

6 Results

The method described in this paper was implemented in C++ and MATLAB. In the sections below we present results in 2, 3 and 4-dimensional configuration spaces.

6.1 Two-dimensional Configuration Space

6.1.1 Path prediction by homotopy class exploration

Figure 12(a) shows a large 1000×1000 discretized environment with circular and rectangular obstacles. We explore trajectories in different classes in order of their path costs using the method 'i.' described in Section 5.1. The implementation was done in C++ running on an Intel Core 2 Duo processor with 2.1 GHz clock-speed and 4GB RAM. All the different trajectories in different homotopy classes were determined in a single run of graph search on $\mathcal{G}_H$ as described earlier.

As discussed earlier, in such exploration problems, although we use the H-signature as the class invariants in the search algorithms, since non-homologous trajectories are guaranteed to be non-homotopic, we are guaranteed to obtain trajectories in different homotopy classes.

We also constructed 10 such environments using random circular and rectangular obstacles. Table 12(c) demonstrate the efficiency of the searches. The time indicates the cumulative time during the search until a shortest-path trajectory in a particular homotopy class is found. This is relevant to problems of tracking dynamic entities, such as people, where one often needs to predict possible paths in order to bias the tracker or to deal with occlusion by anticipating where the dynamic entity will appear. Since people can choose different paths to their destinations, we need to be able to predict least cost paths that lie in different homotopy classes.

6.1.2 H-signature constraint: H-signature defined by Key-points

Figure 13(a) demonstrates an example where we define homology classes using a sample (suboptimal) trajectory specified by key-points. One can then compute the H-signature for such a trajectory. It can then be used to search $\mathcal{G}_H$ for an optimal path in the same class (or different) as the sample trajectory (Figure 13(b)).

Although technically we have imposed homology class constraint by imposing the H-signature constraint, we observe that the optimal trajectory that we obtain is in fact in the same homotopy class as the key-point generated trajectory. In fact we observe that in most robotics planning problems imposing H-signature constraints indeed impose the corresponding homotopy class constraint as well.

6.1.3 Multiple robot visibility problem

The problem of path planning for multiple robots with visibility constraints can also make use of our approach. If one robot needs to plan its path such that it is never obstructed from the view of another robot by some obstacle, we can apply the technique of planning with H-signature constraint to obtain the desired trajectories. In Figure 14(a)-(c) two robots plan trajectories to their respective goals. The robot on the right needs to plan a trajectory such that it is in the "visibility" of the robot on the left, whose trajectory is given. Thus, in order to determine the H-signature of the desired homotopy class it first constructs a suboptimal path by connecting its own start and goal points to the start and goal of the left robot, such that the trajectory of the left robot is completely contained in it (Figure 14(b)) as key points. The H-signature of this path gives the desired homology class, thus re-planning with that class as the only allowed class gives the desired optimal plan (Figure 14(c)).

(a) Paths in 20 different homotopy classes. (b) Run-time & states expanded for finding the least-cost paths in a particular run.
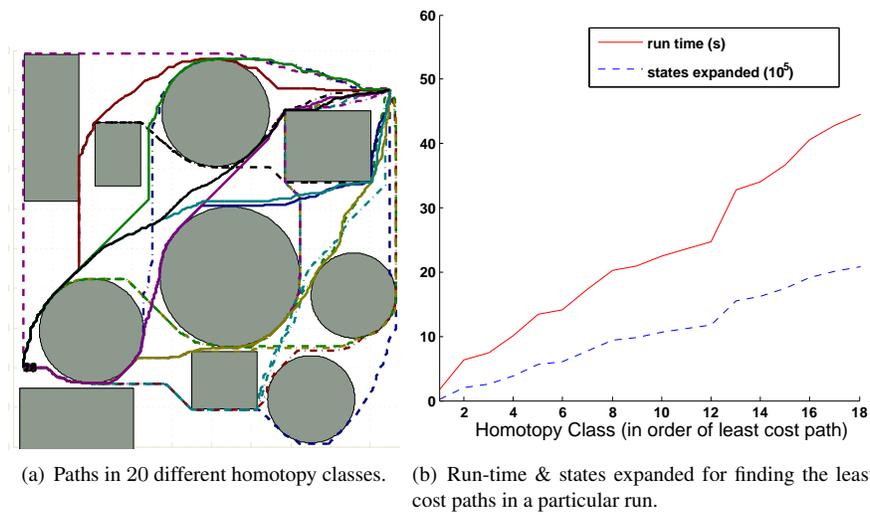
Hmtp. class (i)	time ellapsed until i^{th} hmtp. class explored (s)			states expanded cumulative (10^6)		
	min	max	mean	min	max	mean
1	1.41	2.01	1.71	0.021	0.039	0.032
2	3.58	8.58	5.15	0.099	0.313	0.170
3	5.09	9.69	6.77	0.180	0.375	0.244
4	6.13	12.46	8.92	0.237	0.494	0.345
5	7.80	17.74	11.50	0.285	0.776	0.472
6	10.53	18.56	13.05	0.422	0.825	0.555
7	10.92	19.74	15.38	0.473	0.888	0.681
8	13.35	20.32	17.01	0.604	0.935	0.773
9	15.08	21.76	18.60	0.693	1.027	0.858
10	15.53	26.28	20.87	0.720	1.252	0.978

(c) Statistics of searching least-cost paths in first 10 homotopy classes in 10 randomly generated environments. The numbers represent the cumulative values till the ith homotopy class is explored.

Figure 12: Exploring homotopy classes in 1000×1000 discretized environments to find least cost paths in each.

(a) Suboptimal key-point generated trajectory. (b) Optimal trajectory in same class as key-point generated trajectory.
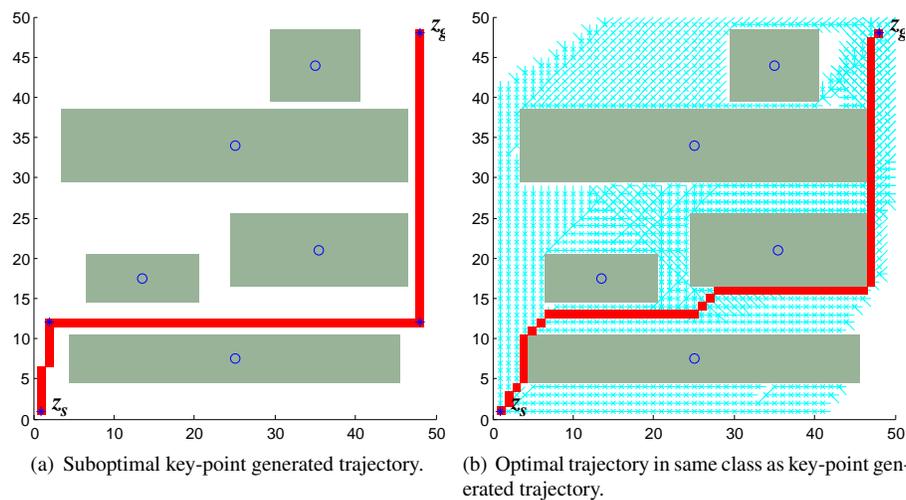
Figure 13: Homotopy class constraint determined using suboptimal key-point generated trajectory.

(a) Unconstrained plans of two robots from their respective start to goal coordinates.

(b) Robot 2 determines H-signature of desired homotopy class by constructing a suboptimal path.

(c) Optimal plan with *visibility constraint* satisfied – obtained using search-based planning with H-signature constraints.
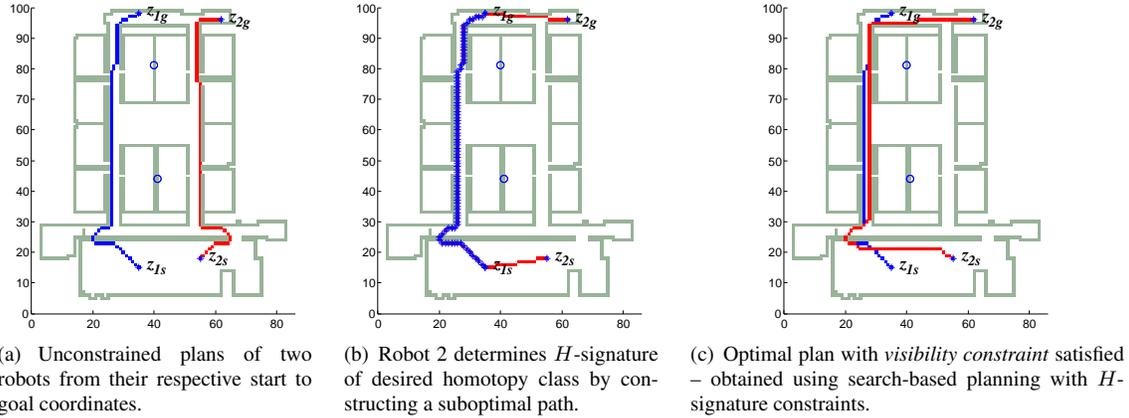
Figure 14: 100×100 discretized environment with 2 *representative points* on the central large connected walls.

(a) $w = 0.0, \mathcal{B} = \{\}$.   (b) $w = 0.01, \mathcal{B} = \{\}$.   (c) $w = 0.0, \mathcal{B} = \{\mathbf{h}_0\}$.   (d) $w = 0.01, \mathcal{B} = \{\mathbf{h}_0\}$.
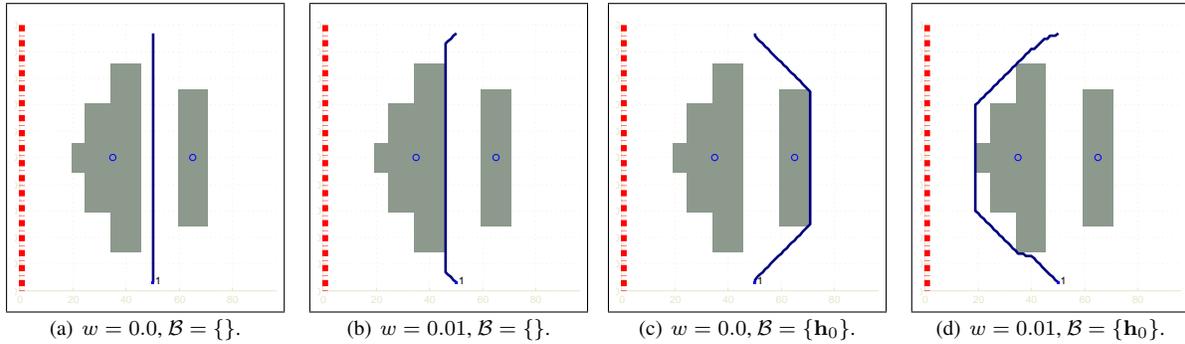
Figure 15: Planning with non-Euclidean length as cost as well as homotopy class constraint.

The natural constraint in this situation is that of homotopy. But we once again observe that even imposing the H-signature constraint we do obtain trajectory in the desired homotopy class.

6.1.4 Arbitrary cost functions

Our method is not limited to Euclidean length cost functions. It can deal with arbitrary cost functions. For example, in Figure 15 there are two large obstacles and a *communication base* to the left of the environment marked by the bold dotted line, $x = 0$. An agent is supposed to plan its path from the bottom to the top of the environment, while minimizing a weighted sum of the length of the trajectory and the distance of the trajectory from the *communication base*. Thus, in this case, besides the transition costs of the states in $\mathcal{G}$, each state, $z = x + iy \in \mathcal{G}$, is assigned a cost $w \cdot x$, the penalty on separation from the *communication base*. Thus the net penalized cost of the trajectory, τ, that is being minimized is of the form $c = \int_\tau ds + w \int_\tau x(s) ds$, where x is the x-coordinate of the points on the trajectory, parametrized by s, the length of the trajectory. The trajectories in figures 15(a) and (b) with penalty weights $w = 0$ and $w = 0.01$ respectively have H-signature of $\mathbf{h}_0$. Blocking this class, but having a small penalty over distance from *communication base* gives the trajectory in 15(d) that passes close to the *communication base*.

6.1.5 Implementation using Visibility Graph

To demonstrate the versatility of the proposed algorithm we implemented it using a Visibility Graph as the state graph, $\mathcal{G}$. Figure 16 shows the visibility graph generated in an environment with polygonal obstacles and the shortest paths in 9 homotopy classes that we explore. Obstacles were *inflated* in order to incorporate collision safety and circular obstacles were approximated by polygons. *Representative points* were placed only on the large obstacles (*i.e.,* relevant obstacles that contribute towards the practical notion of homotopy classes, and not, for example, ones created by
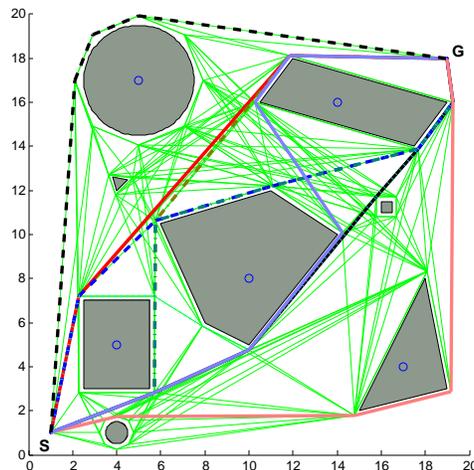
Figure 16: Exploring homotopy classes using a Visibility Graph

sensor noise – determined by threshold on diameter and marked by blue circles in the figure) and visibility graph was constructed. A* search was used for searching the H-signatue augmented graph. The implementation was made in MATLAB. The average run-time of the search until the 9th homotopy class was explored was 0.4 seconds and about 100 states were expanded.

6.2 Three-dimensional Configuration Space

The first 3-dimensional domain in which we implement the planning algorithm is the space of 3 spatial dimensions, X, Y and Z. We also demonstrate the algorithm in the 3-dimensional space of X, Y and $time$, *i.e.* an environment with planar dynamic obstacles (Section 6.2.5).

For a problem in 3 spatial dimensions, the domain of interest is bounded by upper and lower limits of the 3 coordinates. The domain is then uniformly discretized into cubic cells and a node of $\mathcal{G}$ is placed at the center of each cell. Connectivity is established between a node and its 26 neighbors (all cells that share at least one corner, edge or face with it). Each edge is bi-directional and its cost is the Euclidean length.

6.2.1 Simple environments with bounded obstacles

Figure 17(a) demonstrates a simple environment, $20 \times 20 \times 18$ discretized, with two *torus-shaped* obstacles. The skeleton of each obstacle is made up of line segments passing through the central axis of the cylindrical segments. Here we restrict search to *non-looping* trajectories (See [4] for a precise definition). That is, we set $\mathcal{B} = \left\{ \mathbf{h} = [h_1, h_2]^T \mid |h_1| > 1 \text{ or } |h_2| > 1 \right\}$. We search for 4 homotopy classes of trajectories connecting a given start and goal coordinate. As shown in Figure 17(a), the algorithm finds four such trajectories: (i) going through hoops 1 and 2; (ii) going through hoop 1 but not through hoop 2; (iii) going through hoop 2 but not through hoop 1; and (iv) not going through either hoops. According to Theorem 1 each path is the least cost one in the graph and in its respective homotopy class.

Figure 17(b) shows the exploration of 4 homotopy classes in and around a room with windows on each wall. The skeletons for this obstacle are defined as loops around each window according to Construction 3. The trivial shortest path from the given start to goal configuration goes outside the room (the dark violet trajectory). Trajectories in other homotopy classes pass through the room.

6.2.2 Environment with unbounded Pipes

Figure 18(a) shows a more complex environment consisting of 7 pipes stretching to infinity. The workspace of choice is $44 \times 44 \times 44$ discretized, with the start and goal coordinates at two opposite corners of the discretized space. In Figure 18(a) we find the least cost paths in 10 different homotopy classes.

(a) Two hoops.   (b) A room with windows.
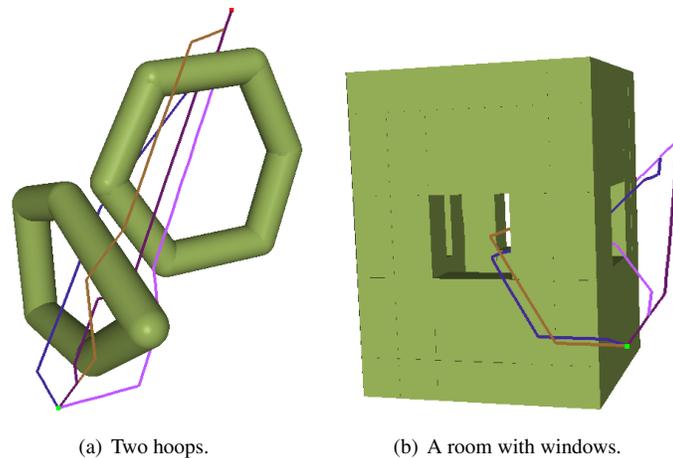
Figure 17: Exploring homotopy classes in $X - Y - Z$ space.

(a) Exploring 10 distinct homotopy classes.   (b) Plan in the *complementary homotopy class* of the least cost path.
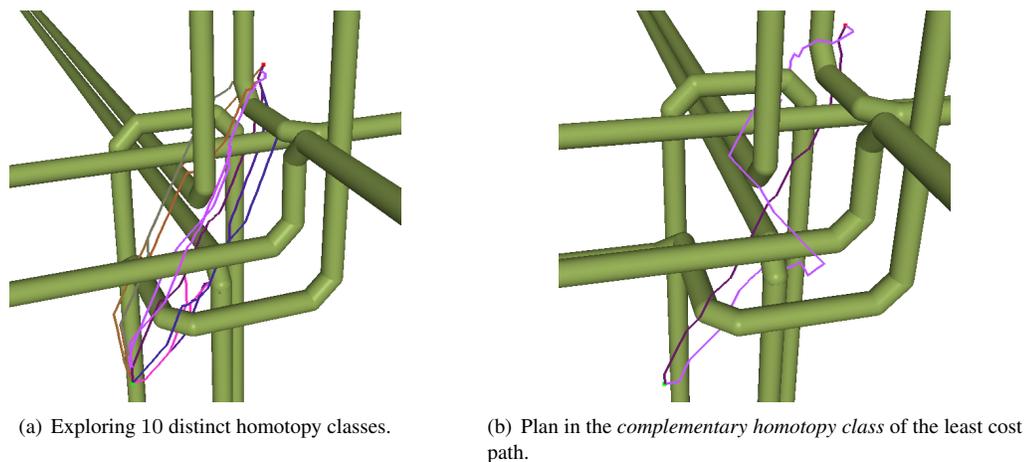
Figure 18: An environment with 7 unbounded pipes.

6.2.3 Planning with H-signature Constraint

Figure 18(b) demonstrates a planning problem with H-signature constraint. The darker trajectory is the global least cost path found from a search in $\mathcal{G}$ for the given start and goal coordinates. The H-signature for that trajectory was computed, and hence we computed the signature of the *complementary class* (*i.e* the class corresponding to the trajectory that passes on the *other side* of every SHIO - see [4] for a precise definition), and put only that in $\mathcal{A}$. The lighter trajectory is the one planned with that $\mathcal{A}$ as the set of allowed H-signature. This trajectory goes on the *opposite side* of each and every pipe in the environment as compared to the darker trajectory.

We note that in this example the notion of *complementary homology class* concurs with that of *complementary homotopy class*.

6.2.4 Search Speed and Efficiency

We now present the running time for the case in Figure 18(a). The environment, as described earlier, is $44 \times 44 \times 44$ discretized, and hence $\mathcal{G}$ contains 85184 nodes. Due to each node being connected to 26 of its neighbors, there are almost 13 times as many edges in $\mathcal{G}$. The program was run on a Intel Core 2 Duo processor with 2.1 GHz clock-speed and 3GB RAM. We first compute the values of $\mathcal{H}(e)$ for all edges $e \in \mathcal{E}$ and store them in a cache, which takes
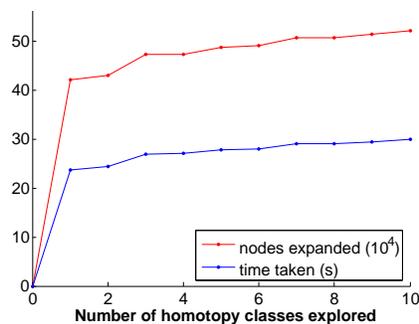
Figure 19: Cumulative time taken and number of states expanded while searching $\mathcal{G}_H$ for 10 homotopy classes in the problem of Figure 18(a).

about $2273s$. Then we perform the A* search in $\mathcal{G}_H$, using the values from the cache whenever required. By doing so we eliminate the requirement of re-computing the *h*-signatures of the edges every time we perform a search, even with changed start and goal coordinates. The search for the 10 homotopy classes in Figure 18(a) took about $30s$ and expansion of 521692 nodes in $\mathcal{G}_H$. Figure 19 shows the cumulative time required and the number of nodes in $\mathcal{G}_H$ expanded.

6.2.5 Planning in 2-dimensional plane with moving obstacles

The next 3-dimensional domain that we experiment with is that of the two-dimensional plane, but with dynamic entities. Thus the variables of interest are X, Y and $time$. The node set was formed by uniform discretization of the domain of interest. The connectivity of the graph is such that the *time* variable can increase only in the positive direction (each node connected to 9 neighboring nodes in next time step, including the same x & y). The cost of an edge, e, with differences in the coordinates of its end points $\Delta x, \Delta y$ and Δt is computed as $c(e) = \sqrt{\Delta x^2 + \Delta y^2 + \epsilon \Delta t^2}$, where ϵ is a small value for avoiding *zero cost edges* in $\mathcal{G}_H$. The skeleton of the moving obstacles are the curves traced by their centers (yellow dots on the oscillating rectangles in Online Resource 1) in the $X - Y - Time$ space. The skeletons are closed outside and far from the discretized domain (Construction 2). Note that in doing so, segments of the skeleton may point along negative time. However that does not effect the planning since the $X - Y - Time$ space itself can be treated no differently from $\mathbb{R}^3$.

The first supplementary video (Online Resource 1) shows the exploration of 4 homotopy classes in $X - Y - Time$ domain. The environment is 40×40 discretized in X and Y directions, and have 100 discretization cells in time. There are two dynamic rectangular obstacles, that undergo a known oscillatory motion inside a narrow passage between other static obstacles. The 4 different trajectories in the different homotopy classes are marked by different colors as well as different numbers at their current locations. The blue trajectory (3) passes above both the obstacles. The red trajectory (2) passes above the right obstacles, but not the left one. The light blue-gray trajectory (1) passes above the obstacle on the left, but not one on the right. The dark gray trajectory (0) is the trivial shortest path. The trajectories in the non-trivial homotopy classes *go behind* the obstacles, a region that would otherwise not be visited by the least cost path without any H-signature consideration.

6.3 Four-dimensional Configuration Space

A natural extension of the example provided in Section 6.2.5 would be to explore homotopy classes of trajectories in a 3-dimensional space with moving obstacles. However that makes the configuration space a 4-dimensional one consisting of the coordinates X, Y, Z and $Time$. So far we have developed representations of H-signature for 2 and 3 dimensional configuration spaces. While doing so we have noticed the similarity in the approach of both. In fact it is possible to unify the formulae and generalize it to arbitrary dimensional configuration space using certain concepts from algebraic and differential topology. We discuss the derivation of such a formula in [3], where we use exterior calculus and the Stokes theorem [22, 10, 21] to design differential 1-forms in arbitrary D-dimensional configuration spaces, the integration of which serve as the desired H-signature in such spaces. This analysis is reminiscent of a more

general treatment that we are currently investigating [1].

    Thus we present a result in a $X - Y - Z - Time$ configuration space. The second supplementary video (Online Resource 2) shows the exploration of 3 homotopy classes in a 4-dimensional configuration space consisting of a dynamic obstacle in 3-dimensions. The loop-shaped obstacle is rotating about an axis. The blue axes are the X, Y and Z axes. The apparent rotation of the axes themselves is due to movement of the *camera* for viewing the trajectories from different angles. As we observe in the video, trajectories numbered 0 and 1 take off from the start coordinate (green dot) and move towards the "center" of the loop. They wait there while 2 takes a different homotopy class to reach the center later. From there 0 and 2 head together towards the goal (red dot), while 1 wait to take a different trajectory to the goal. Thus the 3 trajectories are in different homotopy classes.

7   Conclusion

In this paper we have proposed a novel and efficient way of representing topological information of trajectories for robot motion planning. We have shown that this representation is well suited for use with graph search techniques for finding least cost paths respecting given *homology* class constraints as well as for exploring different *homotopy* classes in an environment. The method is independent of the discretization scheme, cost function or the search algorithm used. We have demonstrated the efficiency, applicability and versatility of the method in our results with examples in two, three and four dimensions.

Acknowledgements

References

[1] S. Bhattacharya, M. Likhachev, and V. Kumar. A homotopy-like class invariant for sub-manifolds of punctured euclidean spaces. Technical report, University of Pennsylvania. See https://fling.seas.upenn.edu/~subhrabh/cgi-bin/wiki/index.php?WPage=chihomotopy.

[2] Subhrajit Bhattacharya, Vijay Kumar, and Maxim Likhachev. Search-based path planning with homotopy class constraints. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, Atlanta, Georgia, July 11 2010.

[3] Subhrajit Bhattacharya, Maxim Likhachev, and Vijay Kumar. *H*-signature for general euclidean spaces. Technical report, The University of Pennsylvania, May 2011. See https://fling.seas.upenn.edu/~subhrabh/cgi-bin/wiki/index.php?SFile=HSignatureHigherDim.

[4] Subhrajit Bhattacharya, Maxim Likhachev, and Vijay Kumar. *H*-signature of a non-looping trajectory with respect to an infinite straight line skeleton. Technical report, The University of Pennsylvania, May 2011. See https://fling.seas.upenn.edu/~subhrabh/cgi-bin/wiki/index.php?SFile=HSignatureNonLooping.

[5] Subhrajit Bhattacharya, Maxim Likhachev, and Vijay Kumar. Identification and representation of homotopy classes of trajectories for search-based path planning in 3d. In *Proceedings of Robotics: Science and Systems*, 27-30 June 2011.

[6] Harry Blum. A Transformation for Extracting New Descriptors of Shape. In Weiant W. Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge, 1967.

[7] R. Bott and L.W. Tu. *Differential forms in algebraic topology*. Graduate texts in mathematics. Springer-Verlag, 1982.

[8] Frederic Bourgault, Alexei A. Makarenko, Stefan B. Williams, Ben Grocholsky, and Hugh F. Durrant-Whyte. Information based adaptive robotic exploration. In *in Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS*, pages 540–545, 2002.

[9] Douglas Demyen and Michael Buro. Efficient triangulation-based pathfinding. In *AAAI'06: Proceedings of the 21st national conference on Artificial intelligence*, pages 942–947. AAAI Press, 2006.

[10] Harley Flanders. *Differential Forms with Applications to the Physical Sciences*. Dover Publications, New York, 1989.

[11] David J. Griffiths. *Introduction to Electrodynamics (3rd Edition)*. Benjamin Cummings, 1998.

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

[13] 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.

[14] Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2001.

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

[16] Anil K. Jain. *Fundamentals of digital image processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[17] James Munkres. *Topology*. Prentice Hall, 1999.

[18] Joseph J. Rotman. *An Introduction to Algebraic Topology*. Springer, 1988.

[19] E. Schmitzberger, J.L. Bouchet, M. Dufaut, D. Wolf, and R. Husson. Capture of homotopy classes with probabilistic road map. In *International Conference on Intelligent Robots and Systems*, volume 3, pages 2317–2322, 2002.

[20] Mac Schwager, Philip Dames, Vijay Kumar, and Daniela Rus. Multi-robot mapping and exploration of environments with hazards. In *RSS 2011 Workshop on 3D Exploration, Mapping, and Surveillance with Aerial Robots*, 2011.

[21] A. Svec. *Global Differential Geometry*. Springer, 2001.

[22] Yves Talpaert. *Differential Geometry with Applications to Mechanics and Physics*. CRC Press, 2000.

[23] Yan Zhou, Bo Hu, and Jianqiu Zhang. Occlusion detection and tracking method based on bayesian decision theory. In Long-Wen Chang and Wen-Nung Lie, editors, *Advances in Image and Video Technology*, volume 4319 of *Lecture Notes in Computer Science*, pages 474–482. Springer Berlin / Heidelberg, 2006.