Identification and Representation of Homotopy Classes of Trajectories for Search-based Path Planning in 3D

— Paper-ID 107

Abstract— **There are many applications in motion planning where it is important to distinguish between and consider the different *homotopy classes* of trajectories. Two trajectories are homotopic if one trajectory can be continuously deformed into another without passing through an obstacle, and a homotopy class is a collection of homotopic trajectories. In this paper we consider the problem of robot exploration and planning in three-dimensional configuration spaces to (a) identify and classify different homotopy classes and (b) plan trajectories constrained to certain homotopy classes or avoiding some others. In [1], the authors solve this problem for two-dimensional, static environments using the Cauchy Integral Theorem in concert with incremental graph search techniques. The robot workspace is mapped to the complex plane and obstacles are poles in this plane. The Residue Theorem allows the use of integration along the path to distinguish between trajectories in different homotopy classes. However, this idea is fundamentally limited to two dimensions. In this work we develop new techniques to save the same problem but in three dimensions. The integral form of Ampere's Law allows us to identify homotopy classes in three dimensions. Skeletons of obstacles in the robot world are extracted and are modeled by current-carrying conductors. It can be shown using the Kelvin-Stokes theorem that the line integral for closed loops are non-zero if and only if the loop encloses one of the current carrying conductors, thus letting us achieve both objectives (a) and (b). We describe the development of a practical graph-search based planning tool with theoretical guarantees by combining integration theory with incremental search techniques, and illustrate it with examples in three-dimensional spaces such as two-dimensional, dynamic environments and three-dimensional static environments.**

I. Introduction

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 application, it is important to distinguish between trajectories of 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 [3], an efficient strategy ought to be able to identify the multiple homotopy classes and deploy at least one agent in each homotopy class. One may also wish to determine the least cost path for each agent 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 [13]. A knowledge of the possible homotopy classes of trajectories that the agent can take in the environment during the period of occlusion can help more efficient belief propagation.

Classification of homotopy classes and motion planning with homotopy class constraints in two-dimensional workspaces have been studied in the past using geometric methods [7, 9], probabilistic road-map construction [12] techniques, triangulation-based path planning [4] and more recently, using complex analysis and incremental path planning techniques [1]. The last method is of importance because it gives us a compact way of representing homotopy classes of trajectories which is independent of the geometry, discretization of the environment, cost function or search algorithm used to find trajectories in the environment. The method is also robust to noise in the environment created by sensor data.

In this paper we propose a novel way of classifying and representing homotopy classes in a 3-dimensional configuration space using theorems from electromagnetism. The representation is designed to be independent of the type of the environment, the discretization scheme or cost function. Using such a representation we show that homotopy class constraints can be seamlessly integrated with graph search techniques for determining optimal paths constrained to specified homotopy classes or forbidden from others. We also discuss how one can explore multiple homotopy classes in an environment using a single graph search.

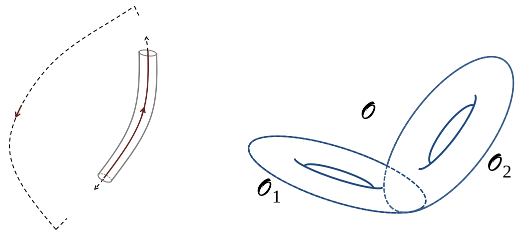
II. Background

A. Homotopy Classes in Three Dimensional Spaces

We re-state the definition of *Homotopy Classes* [1].

Definition 1 (Homotopy Class of Trajectories): Two trajectories (or paths), τ_1 and τ_2 connecting a pair of fixed points are said to be in the same homotopy class (or simply *homotopic*) iff one can be smoothly deformed into the other without intersecting any obstacle. Otherwise they belong to different homotopy classes.

While in the two-dimensional case, theoretically any finite obstacle on the plane can induce multiple homotopy classes for trajectories joining two points, the notion of homotopy classes in three dimensions can only be induced by obstacles with *genus* one or more, or with obstacles stretching to infinity in two directions (The *genus* of an obstacle refers to the number



(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 $\mathcal{O}_2$.

Fig. 1. Illustration of Constructions 1 and 2.

of *holes* or *handles* [11]. See Figure 2). For example, a torus-shaped obstacle in a three-dimensional environment creates two primary homotopy classes: *i.* The trajectories passing through the "hole" of the torus, and *ii.* the trajectories passing outside the "hole" of the torus. Figure 2 shows some examples of obstacles that can or cannot induce homotopy classes for trajectories. A sphere or a solid cube, for example, cannot induce multiple homotopy classes in an environment.

Lemma 1: Two trajectories τ_1 and τ_2 connecting the same points are homotopic if and only if they are homotopic with respect to each and every obstacle in the environment.

Proof: Follows from definition of homotopy class. ■

Definition 2 (Simple Homotopy-Inducing Obstacle): A *Simple Homotopy-inducing Obstacle* (SHIO) is a bounded obstacle of *genus* 1, for example a torus (Figure 2(a), 2(b)) or a knot (Figure 2(e)).

B. Skeleton *of a SHIO*

In [1], each obstacle in a 2-dimensional plane that induces the notion of multiple homotopy classes is 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 3 (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* (*i.e.*, if the obstacle $\mathcal{O}$ is replaced by an equivalent obstacle S, then the *homotopy equivalence* between two arbitrary trajectories, τ_1 and τ_2, connecting every pair of fixed points in the environment, will remain unchanged.)

In the literature, algorithms for constructing skeletons of solid objects is well-studied [2, 10]. 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 2(a) demonstrates how a skeleton can be constructed for a generic genus 1 obstacle. There is definitely no unique way of constructing such a skeleton. For the results in this paper with the $X-Y-Z$ domain, we either hand-picked key-points inside obstacles to construct skeletons, or created obstacles around a *skeleton* to begin with. For the $X-Y-Time$ domain we used similar notion as *representative points* [1] inside moving obstacles, that automatically creates a skeleton for that obstacles in $X-Y-Time$ domain because of extrusion along the time axis.

C. Conversion of generic obstacles into SHIOs

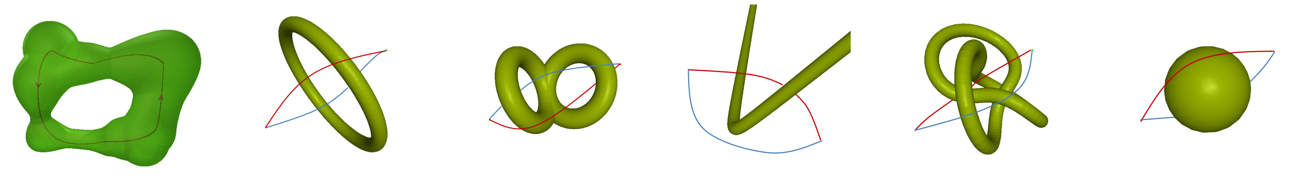
Given a set of obstacles in a three-dimensional environment, we perform the following two constructions/reduction on the obstacles so that the only kind of obstacle we have in the environment are *Simple Homotopy-Inducing Obstacles*. The Construction 1 is mostly trivial in the sense that it can be easily automated for arbitrary obstacles. Construction 2 on the other hand is linked with the construction of *skeleton* of the obstacles (Definition 3) and is discussed later.

***Construction* 1 (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 2(d)) at a large distance from the domain of interest (Figure 1(a)). While this construction does not effect the problem itself, as we will discuss later, it will help us with numerical integration and computation of *homotopy signature* of trajectories.

***Construction* 2 (Decomposing obstacles with genus >1):** After closing all infinite, unbounded obstacles in an environment according to Construction 1, if there is an obstacle with genus k (*e.g.* Figure 2(c)), we can decomposed/split it into k obstacles, possibly overlapping and touching each other, but each with genus 1 (Figure 1(b)). This does not change the obstacles or the problem in any way. This construction just changes the way we identify obstacles. For example in Figure 1(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 homotopy signatures of trajectories.

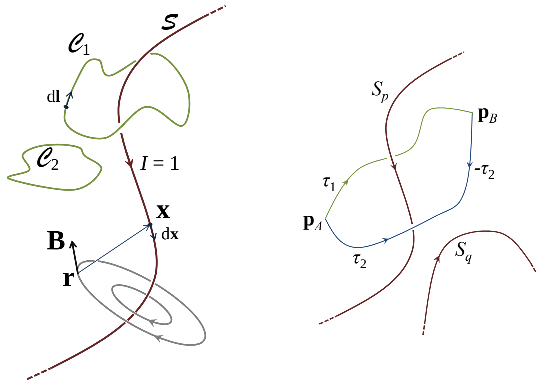
D. Biot-Savart law

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 3(a)). 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 1(a) and Construction 1). We denote the curve by $\mathcal{S}$. Then, according to the Biot-Savart Law [6], 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,



(a) A skeleton of a generic obstacle of genus 1 can be modeled as a current-carrying conductor. (b) A torus-shaped obstacle of genus 1 obstacle. (c) A genus 2 obstacle. (d) An infinite tube is a genus 1 obstacle. (e) A knot-shaped obstacle which has genus 1. (f) A sphere **does not** induce homotopy classes and has genus 0.

Fig. 2. Obstacles that do and do not induce homotopy classes in a 3-dimensional space.



(a) Magnetic field due to current in a conductor $\mathcal{S}$, and its integration along a closed loop $\mathcal{C}_i$. (b) Two trajectories, τ_1 and τ_2, connecting the same points form a closed loop.

Fig. 3.

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi} \int_{\mathcal{S}} \frac{(\mathbf{x} - \mathbf{r}) \times \mathrm{d}\mathbf{x}}{\|\mathbf{x} - \mathbf{r}\|^3} \tag{1}$$

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

E. Ampere's Law

While Biot-Savart law gives a recipe for computing the magnetic field from a given current configuration, Ampere's Law [6], in a sense, gives the inverse of it. Given the magnetic field $\mathbf{B}$ at every point in the space, and a closed loop $\mathcal{C}$ (Figure 3(a)), the line integral of $\mathbf{B}$ along $\mathcal{C}$ gives the current *enclosed* by the loop $\mathcal{C}$. That is,

$$\Xi(\mathcal{C}) := \int_{\mathcal{C}} \mathbf{B}(\mathbf{l}) \cdot \mathrm{d}\mathbf{l} = \mu_0 I_{encl} \tag{2}$$

where, $\mathbf{l}$, the integration variable, represents the coordinate of a point on $\mathcal{C}$, and $d\mathbf{l}$ 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 $\mathcal{C}$, the value of $\Xi(\mathcal{C})$ is zero iff $\mathcal{C}$ does not enclose the conductor, otherwise it is ± 1 (the sign depends on the direction of integration performed on $\mathcal{C}$). Thus in Figure 3(a), $\Xi(\mathcal{C}_1) = 1$ and $\Xi(\mathcal{C}_2) = 0$.

III. Application of Theory of Electromagnetism in Identifying Homotopy Classes

A. Skeleton of SHIOs as Current Carrying Manifolds

***Construction* 3: (Modeling *skeleton* of a SHIO as a current carrying manifold)** This is the key construction: 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 1 and 2), 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 2(a), 1(a)). The direction of the currents is not of importance, but by convention, each is of unit magnitude.

Definition 4 (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 Ampere's Law as follows,

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

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.

B. Homotopy Signature

Definition 5 (Homotopy Signature): Given an arbitrary trajectory, τ, in the 3-dimensional environment with M skeletons, we define the *homotopy signature* of τ to be the following M-vector,

$$\mathcal{H}(\tau) = [h_1(\tau),\ h_2(\tau),\ \ldots,\ h_M(\tau)]^T \tag{4}$$

where,

$$h_i(\tau) = \int_{\tau} \mathbf{B}_i(\mathbf{l}) \cdot \mathrm{d}\mathbf{l} \tag{5}$$

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.

Lemma 2: Two trajectories τ_1 and τ_2 connecting the same pair of fixed end points belong to the same homotopy class if and only if their homotopy signatures are the same.

Proof: Since τ_1 and τ_2 connect the same points, $\tau_1 \cup -\tau_2$, *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 3(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 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

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

is non-zero if and only if the closed loop formed by $\tau_1 \cup -\tau_2$ encloses the current carrying conductor S_i. Otherwise it is zero. For example, in Figure 3(b), $h_p(\tau_1 \cup -\tau_2) = 1$ and $h_q(\tau_1 \cup -\tau_2) = 0$. A direct consequence of this fact is that $h_i(\tau_1 \cup -\tau_2) = 0$ if and only if τ_1 can be smoothly deformed into τ_2 without intersecting S_i. Now, by the definition of line integration we have the following identity,

$$\begin{aligned} h_i(\tau_1 \cup -\tau_2) &= \int_{\tau_1 \cup -\tau_2} \mathbf{B}_i(\mathbf{l}) \cdot \mathrm{d}\mathbf{l} \\ = \int_{\tau_1} \mathbf{B}_i(\mathbf{l}) \cdot d\mathbf{l} - \int_{\tau_2} \mathbf{B}_i(\mathbf{l}) \cdot \mathrm{d}\mathbf{l} &= h_i(\tau_1) - h_i(\tau_2) \end{aligned} \tag{6}$$

Thus, $h_i(\tau_1) = h_i(\tau_2)$ if and only if τ_1 can be smoothly deformed into τ_2 without intersecting S_i (*i.e* homotopic).

Now by Lemma 1, τ_1 and τ_2 are homotopic if and only if they are homotopic with respect to each and every obstacle in the environment. That is, τ_1 and τ_2 are homotopic if and only if $h_i(\tau_1) = h_i(\tau_2)$, $\forall i = 1, 2, \ldots, M$, *i.e.* $\mathcal{H}(\tau_1) = \mathcal{H}(\tau_2)$, *i.e.* the *homotopy signatures* of τ_1 and τ_2 are identical. From Definition 3 it is clear we can replace the obstacles with the skeletons $S_1, S_2, \ldots, S_M$ for the purposes of computing homotopy signatures.

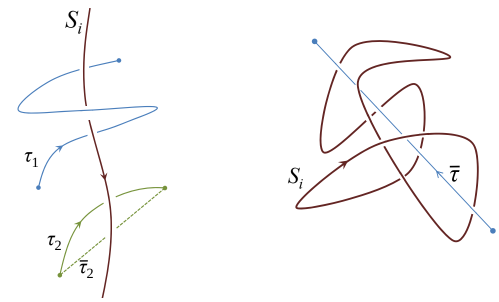
■

C. Some notes on the value of Homotopy Signature

"Looping" of a trajectory around an obstacle (Figure 4(a)) is similar in essence to *non-Jordan* curves on two-dimensional planes. However in three dimensions their precise and universal definition is more difficult. One way of identifying one of the homotopy classes of trajectories (joining a given start and an end coordinate) that do not loop around a skeleton S_i is by joining the start and the end coordinates using a straight line segment (call it $\overline{\tau}$). Then the trajectories that are homotopic to $\overline{\tau}$ form a particular homotopy class of *non-looping trajectories* w.r.t. S_i (for example, in Figure 4(a), the homotopy class to which $\overline{\tau}_2$, and hence τ_2, belong are non-looping). However, for more complex obstacles (like knots), the notion of a *non-looping* trajectory being a straight line segment breaks down (See Figure 4(b)). In fact the notions of *looping* and *non-looping* is imprecise in such cases. In Appendix VII-A we show that for the special simple case when S_i is an infinitely long line, the component of the homotopy signature $h_i(\overline{\tau})$ for a line segment $\overline{\tau}$ lies between -1 and 1. We hence propose the following mathematical definition of a *non-looping* trajectory,

Definition 6 (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. If another trajectory, τ', connecting the same start and end points as a non-looping trajectory τ, goes on the "other side of the obstacle" without looping around it, then $\tau \cup -\tau'$ forms a closed loop that encloses S_i. Then, $h_i(\tau \cup -\tau') = \pm 1 = \text{sign}(h_i(\tau \cup -\tau'))$. But since, τ and $\tau \cup -\tau'$ goes around S_i in the same orientation, we have $\text{sign}(h_i(\tau \cup -\tau')) = \text{sign}(h_i(\tau))$. Again by property of line integration, $h_i(\tau \cup -\tau') = h_i(\tau) - h_i(\tau')$. Thus, $h_i(\tau') = h_i(\tau) - \text{sign}(h_i(\tau))$. Thus we have the following definition.



(a) Trajectory that loops around a skeleton and trajectory that doesn't. In this figure $h_i(\tau_1) > 1$ and $0 < h_i(\tau_2) = h_i(\overline{\tau}_2) < 1$.

(b) In the most general case, it is difficult to precisely identify a *non-looping* homotopy class.

Fig. 4.

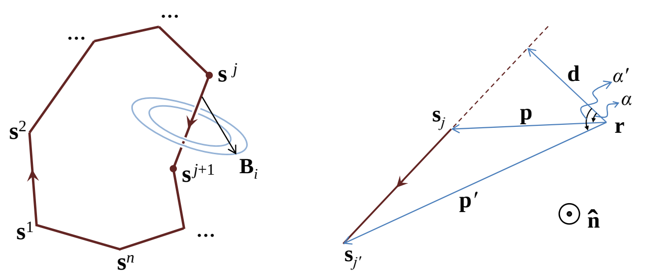
Definition 7 (Complementary Homotopy Class): Given a trajectory τ that is *non-looping* w.r.t. all the skeletons in the environment (*i.e.* $h_i(\tau) \in (-1, 1)\ \forall\ i = 1, 2, \ldots, M$), we define the *Complementary Homotopy Class* of the homotopy class of τ to be the one for which the homotopy signature is $\mathcal{H}(\tau) - \text{sign}(\mathcal{H}(\tau))$, where $\text{sign}(\cdot)$ gives the vector of signs of the elements of its input vector.

IV. Search-based Planning in Three Dimensions with Homotopy Class Constraints

We now investigate the problem of search-based path planning for trajectories in 3-dimensional configuration spaces. Primarily we investigate two types of problems: (i.) Exploration of the different homotopy classes of trajectories connecting a given start and goal coordinates in the environment, and (ii.) Planning for trajectories with specified homotopy class constraints (where we are required to find trajectories restricted to specified homotopy classes, and/or avoiding other specified homotopy classes). We perform these tasks in two kinds of environments: *a)* two-dimensional dynamic environment (*i.e. X-Y-Time* domain), and *b)* three-dimensional static environment (*i.e. X-Y-Z* domain).

In the discussion that follows, we represent a point in the 3-dimensional configuration space using the coordinates $\mathbf{v} = (x, y, z)$, with the understanding that z can represent $time$ in the time-varying 2-dimensional environment.

The approach, as in [1], is to discretize the configuration space, construct a directed graph out of it, and perform a graph search in it. The discretization can be quite general. Approximate or exact cell decompositions can be used to generate a roadmap. The roadmap can be probabilistic or deterministic. Or a uniform grid representation can be used to generate a graph, which is the representation used here.



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

(b) Magnetic field at $\mathbf{r}$ due to the current in a line segment $\overrightarrow{\mathbf{s}_i^j\mathbf{s}_i^{j'}}$ can be computed analytically.

Fig. 5.

The discretized space is represented by the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, in which each node $\mathbf{v} = (x, y, z) \in \mathcal{V}$ represents the coordinate of a discretized cell. Depending on the type of configuration space, the nodes are connected to their relevant neighboring nodes by weighted edges, where the weights are equal to the cost of traversing the edge. A directed edge connecting node $\mathbf{v}_1$ to $\mathbf{v}_2$ is represented by $\{\mathbf{v}_1 \to \mathbf{v}_2\}$. Inaccessible coordinates (lying inside obstacles or outside a specified workspace) do not constitute nodes of the graph. A path in this graph represents a trajectory of the robot in the 3-dimensional configuration space. Moreover, small obstacles (*e.g.* created by sensor noise), or obstacles that we don't desire to contribute towards the homotopy class of trajectories, can be chosen not to have a skeleton, thus preventing them from claiming a component in the homotopy signature vector.

We will discuss the connectivity of the graph, $\mathcal{G}$, and the cost function in greater details for each of the two types of configuration space we present in Section V.

A. Computation of Homotopy Signature for an Edge of $\mathcal{G}$

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 5(a)). Thus, the integration of equation (3) can be split into summation of n_i integrations,

$$\mathbf{B}_i(\mathbf{r}) = \frac{1}{4\pi}\sum_{j=1}^{n_i}\int_{\overrightarrow{\mathbf{s}_i^j\mathbf{s}_i^{j'}}} \frac{(\mathbf{x}-\mathbf{r})\times \mathrm{d}\mathbf{x}}{\|\mathbf{x}-\mathbf{r}\|^3} \tag{7}$$

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

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 [6] (also, see Figure 5(b)),

$$\begin{aligned}\int_{\overrightarrow{\mathbf{s}_i^j\mathbf{s}_i^{j'}}} \frac{(\mathbf{x}-\mathbf{r})\times \mathrm{d}\mathbf{x}}{\|\mathbf{x}-\mathbf{r}\|^3} &= \frac{1}{\|\mathbf{d}\|}\left(\sin(\alpha')-\sin(\alpha)\right)\hat{\mathbf{n}} \\ &= \frac{1}{\|\mathbf{d}\|^2}\left(\frac{\mathbf{d}\times\mathbf{p}'}{\|\mathbf{p}'\|} - \frac{\mathbf{d}\times\mathbf{p}}{\|\mathbf{p}\|}\right)\end{aligned} \tag{8}$$

where, $\mathbf{d}, \mathbf{p}$ and $\mathbf{p}'$ are functions of $\mathbf{s}_i^j, \mathbf{s}_i^{j'}$ and $\mathbf{r}$ (Figure 5(b)), and can be expressed as,

$$\begin{gathered}\mathbf{p} = \mathbf{s}_i^j - \mathbf{r}\ , \quad \mathbf{p}' = \mathbf{s}_i^{j'} - \mathbf{r}\ , \\ \mathbf{d} = \frac{(\mathbf{s}_i^{j'} - \mathbf{s}_i^j)\times(\mathbf{p}\times\mathbf{p}')}{\|\mathbf{s}_i^{j'} - \mathbf{s}_i^j\|^2}\end{gathered} \tag{9}$$

We define and write $\mathbf{\Phi}(\mathbf{s}_i^j, \mathbf{s}_i^{j'}, \mathbf{r})$ for the RHS of Equation (8) for notational convenience. Thus we have,

$$\mathbf{B}_i(\mathbf{r}) = \frac{1}{4\pi}\sum_{j=1}^{n_i} \mathbf{\Phi}(\mathbf{s}_i^j, \mathbf{s}_i^{j'}, \mathbf{r}) \tag{10}$$

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

Given an edge $e \in \mathcal{E}$, we can now compute the *homotopy signature*, $\mathcal{H}(e) = [h_1(e),\ h_2(e),\ \ldots,\ h_M(e)]^T$, where,

$$h_i(e) = \frac{1}{4\pi}\int_e \sum_{j=1}^{n_i} \mathbf{\Phi}(\mathbf{s}_i^j, \mathbf{s}_i^{j'}, \mathbf{l})\cdot \mathrm{d}\mathbf{l} \tag{11}$$

can be computed numerically.

B. Homotopy Signature Augmented Graph

Let $\mathbf{v}_s = (x_s, y_s, z_s)$ be the start coordinate in the configuration space, and $\mathbf{v}_g = (x_g, y_g, z_g)$ be the goal coordinate. By Lemma 2, any two trajectories from $\mathbf{v}_s$ to $\mathbf{v}$ that belong to the same homotopy class will have the same homotopy signature. The homotopy signature can assume different, but discrete values corresponding on the homotopy 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.

1) Allowed and Blocked Homotopy Classes: Suppose it is required that we restrict all our search for trajectories connecting $\mathbf{v}_s$ and $\mathbf{v}_g$ to certain homotopy classes, or not allow certain homotopy classes. We denote the set of allowed homotopy signatures of trajectories leading up to $\mathbf{v}_g$ by the set $\mathcal{A}$, and the set of blocked homotopy 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 homotopy signatures of all the homotopy classes of trajectories joining $\mathbf{v}_s$ and $\mathbf{v}_g$), and $\mathcal{B}$ can be an empty set when all homotopy classes are allowed. Following the discussion in Section III-C, it is also possible to restrict search to *non-looping* trajectories by putting all homotopy signatures that have at least one element outside $(-1, 1)$ into the set $\mathcal{B}$.

2) Homotopy Signature Augmented Graph: Once we have the means of computing homotopy signature for each edge, we introduce the concept of *homotopy signature augmented graph*. We define the *homotopy 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 homotopy 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 homotopy 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 and assumes the values of the homotopy signatures corresponding to the discrete homotopy classes. Thus, we define the *homotopy signature augmented graph* of $\mathcal{G}$ as follows,

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

where,

1. $$\mathcal{V}_H = \left\{ \{\mathbf{v}, \mathbf{h}\} \;\middle|\; \begin{array}{l} \mathbf{v} \in \mathcal{V}, \text{ and,} \\ \mathbf{h} = \mathcal{H}(\widetilde{\mathbf{v}_s\mathbf{v}}) \text{ for some trajectory} \\ \qquad\qquad \widetilde{\mathbf{v}_s\mathbf{v}} \in \mathcal{P}(\mathbf{v}_s, \mathbf{v}), \text{ and,} \\ \mathbf{h} \in \mathcal{A} \text{ (equivalently, } \mathbf{h} \notin \mathcal{B}) \\ \qquad\qquad\qquad \text{when } \mathbf{v} = \mathbf{v}_g \end{array} \right\}$$
2. An edge $\{\{\mathbf{v}, \mathbf{h}\} \rightarrow \{\mathbf{v}', \mathbf{h}'\}\}$ 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 $\{\mathbf{v} \rightarrow \mathbf{v}'\} \in \mathcal{E}$, and,
   ii. $\mathbf{h}' = \mathbf{h} + \mathcal{H}(\mathbf{v} \rightarrow \mathbf{v}')$, where, $\mathcal{H}(\mathbf{v} \rightarrow \mathbf{v}')$ is the *homotopy 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 the cost/weight associated with edge $\{\mathbf{v} \rightarrow \mathbf{v}'\} \in \mathcal{E}$.

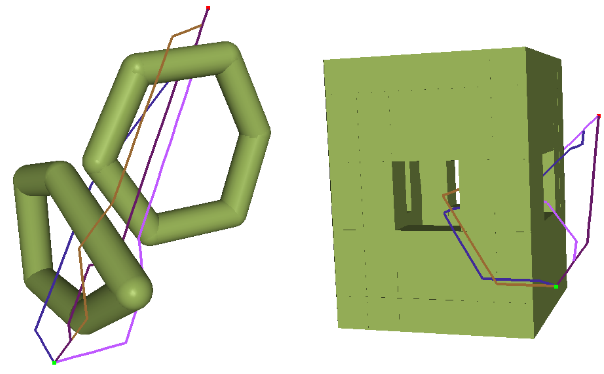
The consequence of point 3 in the above definition is that an *admissible heuristics* for search in $\mathcal{G}$ will remain admissible in $\mathcal{G}_H$. That is, if $f(\mathbf{v}, \mathbf{v}_g)$ was the heuristic function in $\mathcal{G}$, we define $f_H(\{\mathbf{v}, \mathbf{h}\}, \{\mathbf{v}_g, \mathbf{h}'\}) = f(\mathbf{v}, \mathbf{v}_g)$ 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 homotopy signature is that now nodes are distinguished not only by their coordinates, but also the homotopy class 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 $\{\mathbf{z}_s, \mathbf{0}\}$ (where by $\mathbf{0}$ we mean a M-dimensional vector of zeros). For exploration of homotopy classes, whenever we expand a state $\{\mathbf{z}_g, \tilde{\mathbf{h}}\} \in \mathcal{V}_H$, 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 homotopy classes are explored. That way we explore homotopy classes in order of their path costs. For searches with homotopy class constraint, we stop upon expansion of a goal coordinate $\{\mathbf{z}_g, \tilde{\mathbf{h}}\}$ for some $\tilde{\mathbf{h}} \notin \mathcal{B}$ (or equivalently, $\tilde{\mathbf{h}} \in \mathcal{A}$).

C. 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 Homotopy class 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 homotopy class 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 homotopy class constraints defined in $\mathcal{G}_H$. ■



(a) Two hoops. (b) A room with windows.

Fig. 6. Exploring homotopy classes in $X - Y - Z$ space.

V. Results

We implemented the graph structure, $\mathcal{G}_H$, and A* search algorithm [8] to search in the graph using C++ programing language. For the numerical integration in Equation (11) we used the GNU Scientific Library [5]. For the graphic visualization we used *OpenCV* and *OpenGL* libraries.

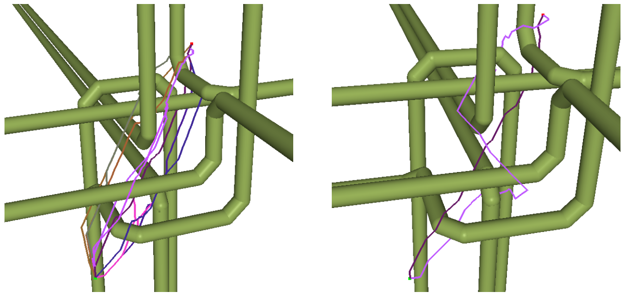
A. Planning in 3-dimensional space with static obstacles

The first domain in which we implement the planning algorithm is the space of 3 spatial dimensions, X, Y and Z. For a particular problem, 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.

1) Simple environments with bounded obstacles: Figure 6(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. That is, we set $\mathcal{B} = \left\{\mathbf{h} = [h_1, h_2]^T \;\middle|\; |h_1| > 1 \text{ or } |h_2| > 1\right\}$. We search for 4 homotopy classes of trajectories connecting a given start and goal coordinate. As shown in Figure 6(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 6(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 2. 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.

2) Environment with unbounded Pipes: Figure 7(a) shows a more complex environment consisting of 7 pipes stretching to infinity. The workspace of choice is $44 \times 44 \times 44$ discretized,



(a) Exploring 10 distinct homotopy classes. (b) Plan in the *complementary homotopy class* of the least cost path.

Fig. 7. An environment with 7 unbounded pipes.

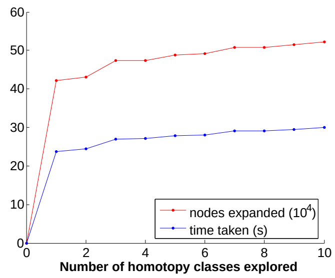


Fig. 8. Cumulative time taken and number of states expanded while searching $\mathcal{G}_H$ for 10 homotopy classes in the problem of Figure 7(a).

with the start and goal coordinates at two opposite corners of the discretized space. In Figure 7(a) we find the least cost paths in 10 different homotopy classes.

3) Planning with Homotopy Class Constraint: Figure 7(b) demonstrates a planning problem with homotopy class 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 homotopy signature for that trajectory was computed, and hence we computed the signature of the *complementary homotopy class* (Definition 7), and put it in $\mathcal{A}$. The lighter trajectory is the one planned with that $\mathcal{A}$ as the set of allowed homotopy signature. This trajectory goes on the *opposite side* of each and every pipe in the environment as compared to the darker trajectory.

4) Search Speed and Efficiency: We now present the running time for the case in Figure 7(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 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 homotopy 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 7(a) took about $30s$ and expansion of 521692 nodes in $\mathcal{G}_H$. Figure 8 shows the cumulative time required and the number of nodes in $\mathcal{G}_H$ expanded during search for the 10 homotopy classes.

B. 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 in Figure 9) in the $X-Y-Time$ space. The skeletons are closed outside and far from the discretized domain (Construction 1). 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 $\mathbb{R}^3$.

Figure 9 shows the screen-shots from 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, $\mathcal{O}_1$ and $\mathcal{O}_2$, 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 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 homotopy class consideration.

VI. Conclusion

In this paper we have proposed a novel and efficient way of representing homotopy classes in 3-dimensional configuration spaces by exploiting laws from Theory of Electromagnetism. We have shown that this representation is well suited for use with incremental graph search techniques for finding least cost paths respecting given homotopy class constraints as well as for exploring different homotopy classes in an environment. The method is independent of the discretization scheme, the cost function or geometry of the environment. We have demonstrated the efficiency, applicability and versatility of the method in our results.

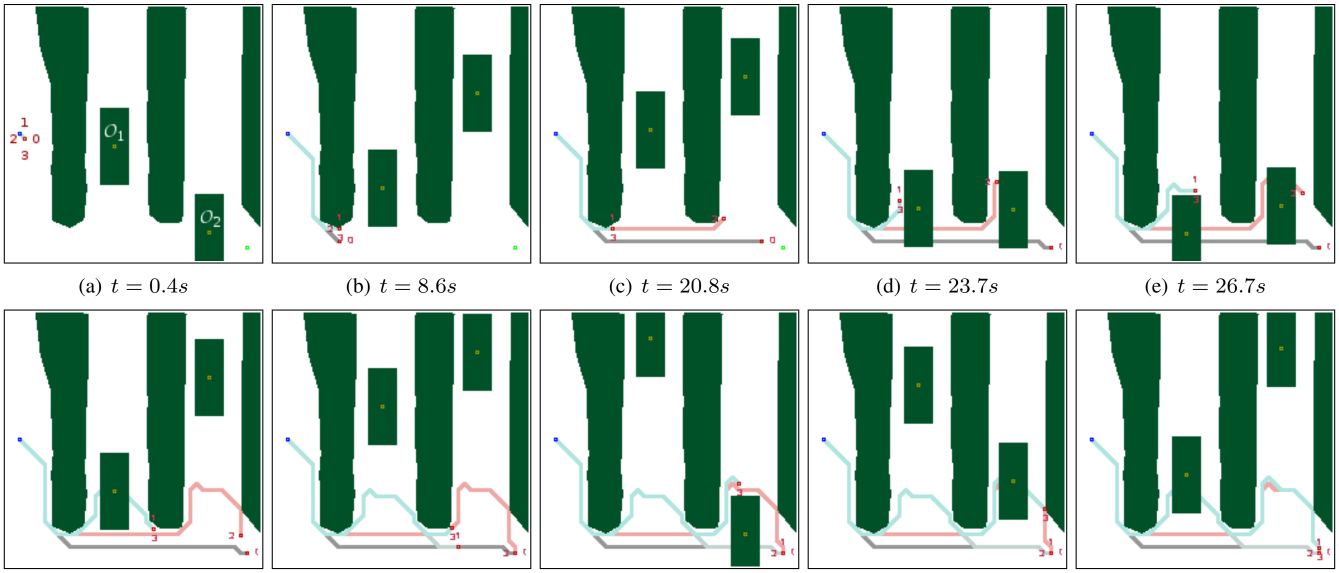
VII. Appendix

A. *Homotopy signature of a* non-looping *trajectory w.r.t. an infinite straight line skeleton*

Making use of the result from Equation (8), if the current carrying line segment stretches to infinity in both direction (*i.e.* it becomes a line), we have $\alpha' = \frac{\pi}{2}$ and $\alpha = -\frac{\pi}{2}$. The virtual magnetic field due to S_i at a point $\mathbf{r}$ becomes

$$\mathbf{B}_i = \frac{1}{4\pi} \frac{2\,\hat{\mathbf{n}}}{\|\mathbf{d}\|} = \frac{1}{2\pi} \frac{\hat{\mathbf{n}}}{\|\mathbf{d}\|} \tag{12}$$

Note that the contribution of the closing curve at infinity (Construction 1) becomes zero in the above quantity.



(a) $t = 0.4s$ (b) $t = 8.6s$ (c) $t = 20.8s$ (d) $t = 23.7s$ (e) $t = 26.7s$

(f) $t = 29.6s$ (g) $t = 32.5s$ (h) $t = 37.0s$ (i) $t = 40.0s$ (j) $t = 43.1s$

Fig. 9. Screen-shots from an example with two moving obstacles ($\mathcal{O}_1$ and $\mathcal{O}_2$) showing the exploration of 4 homotopy classes in a dynamic environment. The blue trajectory (3) passes above both $\mathcal{O}_1$ and $\mathcal{O}_2$. The red trajectory (2) passes above $\mathcal{O}_2$, but not $\mathcal{O}_1$. The light blue-gray trajectory (1) passes above $\mathcal{O}_1$, but not $\mathcal{O}_2$. The dark gray trajectory (0) is the trivial shortest path.

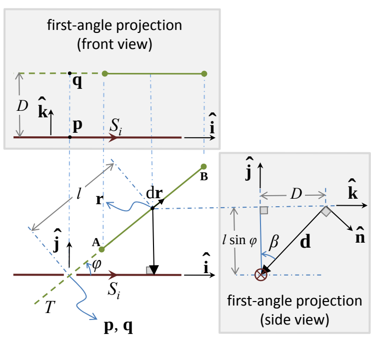


Fig. 10. An infinitely long skeleton and homotopy signature of a straight line segment.

Now consider the straight line segment trajectory $\overline{\tau} = \overline{\mathbf{r}_A\mathbf{r}_B}$. Let the line containing $\overline{\tau}$ (*i.e.* formed by extending $\overline{\tau}$ to infinity in both directions) be T (Figure 10). Consider the shortest distance between S_i and T and let it be D. Assuming S_i and T are not parallel, there is a unique point on each of these line ($\mathbf{p}$ and $\mathbf{q}$ respectively) that are closest and are separated by the distance D. The line segment joining the closest points, $\overline{\mathbf{pq}}$, is perpendicular to both S_i and T. The main diagram of Figure 10 shows the projection of S_i and T on a plane perpendicular to $\overline{\mathbf{pq}}$. Note that this plane (the plane of the paper) is parallel to both S_i and T, since it is perpendicular to $\overline{\mathbf{pq}}$.

We define an orthonormal coordinate system with unit vectors $\hat{\mathbf{i}}$ pointing along S_i in the direction of the current, and unit vector $\hat{\mathbf{k}}$ pointing along $\overline{\mathbf{pq}}$. Using these, and referring to Figure 10, we now can write the following equations,

$$\begin{gathered} \|\mathbf{d}\|^2 = D^2 + l^2 \sin^2\phi \\ \hat{\mathbf{n}} = \cos\beta\ \hat{\mathbf{k}} - \sin\beta\ \hat{\mathbf{j}}\ , \quad \mathrm{d}\mathbf{r} = (\cos\phi\ \hat{\mathbf{i}} + \sin\phi\ \hat{\mathbf{j}})\ \mathrm{d}l \end{gathered} \tag{13}$$

where, ϕ is a constant angle between S_i and T on the plane of the paper, $\cos\beta = \frac{l\sin\phi}{\|\mathbf{d}\|}$, $\sin\beta = \frac{D}{\|\mathbf{d}\|}$, and l is the length parameter along T starting at $\mathbf{q}$. Thus from (12),

$$\mathbf{B}_i \cdot \mathrm{d}\mathbf{r} = -\frac{1}{2\pi}\frac{\sin\beta\ \sin\phi}{\|\mathbf{d}\|}\mathrm{d}l = \ -\frac{D\sin\phi}{2\pi}\frac{\mathrm{d}l}{D^2 + l^2\sin^2\phi} \tag{14}$$

Thus, $$\int_{\overline{\tau}} \mathbf{B}_i \cdot \mathrm{d}\mathbf{r} = -\frac{D\sin\phi}{2\pi}\int_{l_A}^{l_B}\frac{\mathrm{d}l}{D^2 + l^2\sin^2\phi}$$
$$= -\frac{1}{2\pi}\left(\arctan\left(\frac{l_B}{D/\sin\phi}\right) - \arctan\left(\frac{l_A}{D/\sin\phi}\right)\right) \tag{15}$$

An arctangent of a quantity, with consideration for proper quadrants, can assume values between $-\pi$ and π. Thus the quantity within the outer brackets of Equation (15), that is the difference of two arctangents, can assume values between -2π and 2π. Thus the integral $\int_{\overline{\tau}} \mathbf{B}_i \cdot \mathrm{d}\mathbf{r}$, can assume values between -1 and 1. Thus, as claimed in Section III-C, a straight line segment trajectory indeed has the value of $h_i(\tau)$ in $(-1, 1)$ for this simple case of infinitely long line S_i.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] Mark Galassi, Jim Davies, James Theiler, Brian Gough, Gerard Jungman, Michael Booth, and Fabrice Rossi. *Gnu Scientific Library: Reference Manual*. Network Theory Ltd., February 2003.

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

[7] 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.

[8] 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.

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

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

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

[12] 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.

[13] 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.