

Homotopy aware kinodynamic planning using RRT-based planners

Basak Sakcak, Luca Bascetta and Gianni Ferretti

Abstract—This paper introduces a method for kinodynamic planning with homotopy class constraints, and proposes a homotopy class identifier that establishes a geometric relation between a trajectory and a union of convex partitions of the 2D robot workspace. The proposed identifier is shown to be invariant with respect to the trajectories that belong to the same homotopy class, in such a way that each class has its own unique signature. Furthermore, we show that the proposed homotopy class identifier can be easily incorporated in a RRT-based planner, without changing the planning algorithm, while restricting the solution trajectories to a designated homotopy class.

I. INTRODUCTION

Planning with topological constraints has gained an increasing attention in recent years. In fact, besides obtaining a collision free trajectory that minimizes a selected objective, one might be interested in how the resulting trajectory avoids obstacles. This problem is relevant in a number of different robotics and autonomous vehicle applications, ranging from avoiding dangerous zones and coordinating multiple agents for better exploration, to imposing rules on how to avoid obstacles. Furthermore, it is also a useful tool to decompose an otherwise complex problem into simpler sub-problems.

A homotopy class in the context of trajectory planning is the set of all trajectories that can be morphed to each other without any discontinuity. In other words, trajectories that belong to the same class avoid obstacles in the same way. The introduction of this concept into the planning scenario has gained an increased interest in the last decade. Most of the works that fall into this class are focused on returning paths that are different in how they avoid obstacles. The main idea is to represent the environment in an abstract form, determine a graph representation, and traverse the graph to obtain a number of different paths using a homotopy invariant. Examples of homotopy invariants include the winding angle of the path [1], or a topology equivalence function based on Complex Analysis ([2], [3]). These approaches couple the identification of the homotopy class with motion planning; in other words, the algorithm has to explore all possible paths before returning k -best solutions.

In most of the robotic applications, finding a collision free path in the robot workspace does not guarantee its feasibility, this is indeed the case when the robot at hand is differentially constrained. In order to overcome this issue some motion planners focus on kinodynamic planning [4], which combines the search for a collision-free path with the underlying dynamics of the system. Including also

the homotopy-awareness within the kinodynamic planning problem, in [5] the authors decompose the space using triangulation, and obtain a series of linear constraints that can be included in a Model Predictive Control (MPC) framework to ensure feasibility while satisfying the homotopy class constraints. Similarly, Park et al. [6] enforce the trajectory to a homotopy class by means of cell sequences, and use Mixed Integer Quadratic Programming (MIQP) to compute trajectories for a corresponding dynamical system that follow the cell sequence and avoid obstacles.

Sampling-based algorithms, such as Rapidly-exploring Random Tree (RRT) [7], are one of the most popular approaches in solving the kinodynamic planning problem owing to their practicality and scalability for higher dimensions. With the increased concern on the quality of the resulting trajectory RRT* [8] was proposed, which achieves asymptotic optimality.

Sampling based planners that take into account homotopy class constraints exist as well, like for example H-RRT [9] that partitions the search space similarly to the reference frame method [10], preliminarily obtaining the homotopy class invariant in terms of a sequence of boundary crossings. This information is then embedded in the RRT framework to enforce that the planned trajectory belongs to the homotopy class. Note that, RRT builds a tree whose edges do not change once they are added to the tree. On the other hand, the asymptotic optimality property of RRT* requires some edges to be removed and new ones to be added, i.e., rewiring the tree, at each iteration. Keeping track of a homotopy identifier in terms of boundary crossings is then problematic due to the dynamic nature of the built tree.

The main contribution of this paper is a homotopy class invariant which is based on the union of convex cells used to partition the 2D robot workspace. It can then be easily embedded in a RRT-based algorithm, without any additional change. We explicitly describe this case using the example of RRT*, for which keeping track of a homotopy identifier can be challenging. We also show that using a preliminary information on the topology of the robot workspace some informed decisions can be made that improve the convergence of the algorithm to a lower cost solution.

II. PROBLEM DEFINITION

In this work, we consider dynamic systems with state vector $s \in S \subset \mathbb{R}^d$ and control input $u \in U \subset \mathbb{R}^m$, where $d \geq 2$ and $m \geq 1$, governed by the following state transition equation

$$\dot{s}(t) = f(s(t), u(t)) \quad (1)$$

The authors are with Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Piazza Leonardo Da Vinci 32, 20133, Milano, Italy {name.surname}@polimi.it

where f is continuously differentiable as a function of both arguments, and S and U are compact sets. The state $s(t)$ is assumed to include the vehicle position $\pi(t)$, defined with respect to an absolute reference frame.

A trajectory $\sigma : [0, \tau] \rightarrow S$ is defined as a state evolution along the time interval $[0, \tau]$, where τ is the duration of the trajectory, that satisfies the differential equation (1) for $t \in [0, \tau]$.

We assume that obstacles can be represented in a 2D space and any information about collision and obstacle avoidance can be retrieved using the projection of the robot state onto this 2D plane, i.e. the robot footprint. Let $\mathcal{P} \subset \mathbb{R}^2$ define the portion of the 2D plane where the robot operates, and $\mathcal{P}_{obs} \subset \mathbb{R}^2$ is the obstacle region. We can thus introduce the obstacle-free region, defined on this 2D plane as $\mathcal{P}_{free} := \mathcal{P} \setminus \mathcal{P}_{obs}$. Let $\mu : S \rightarrow \mathbb{R}^2$ be a transformation that maps a state to its position in this 2D plane. Consequently, the free state space, S_{free} , is defined as the set of states that can be mapped to \mathcal{P}_{free} , i.e., $S_{free} := \{s \in S \mid \mu(s) \in \mathcal{P}_{free}\}$.

Let $\phi(\lambda) : [0, 1] \rightarrow \mathcal{P}$, where $\lambda \in [0, 1]$, be the geometric path that corresponds to the trajectory $\sigma(t)$ with final time τ , i.e., $t \in [0, \tau]$. Two trajectories σ and σ' having the same initial and final positions, π_0 and π_f , i.e., $\phi(0) = \phi'(0) = \pi_0$ and $\phi(1) = \phi'(1) = \pi_f$, are homotopic if there exists a continuous mapping $\psi : [0, 1] \times [0, 1] \rightarrow \mathcal{P}_{free}$ that can deform ϕ into ϕ' . Furthermore, we can define a homotopy class, $\Sigma_{\mathcal{H}}$, as the set of all the trajectories which have the same initial and final positions, and their respective geometric paths can be deformed to each other using a continuous mapping function.

Based on the previous definitions, for a given homotopy class, $\Sigma_{\mathcal{H}}$, the solution of an *optimal homotopic kinodynamic planning problem* is the control input $u(t) \in U$ for $t \in [0, \tau]$ and the trajectory duration, τ , which result in the trajectory $\sigma(t) \in \Sigma_{\mathcal{H}}$ that is optimal according to a cost criterion $J(s, u, \tau) : \Sigma_{\mathcal{H}} \rightarrow \mathbb{R}_{\geq 0}$ that is expressed as

$$J(s, u, \tau) = \int_0^{\tau} g(s(t), u(t)) dt \quad (2)$$

where $g : S \times U \rightarrow \mathbb{R}_{\geq 0}$ is an instantaneous cost function.

Finally, the overall problem addressed here implies two different steps: first, given the representation of the obstacles in the 2D plane, a homotopy class invariant has to be identified; second, the homotopic kinodynamic planning problem has to be solved imposing the required topological characteristics on the solution trajectories.

III. HOMOTOPY CLASS IDENTIFICATION

Computing an identifier to determine if a trajectory belongs to an homotopy class is the first step of a planning algorithm with homotopy class constraints. This identifier should capture the topology of the environment, and has to be different for each homotopy class.

The topology of an environment can be easily defined as a union of partitions of the workspace, together with their connectivity rules. Thus, the first part of homotopy

class identification refers to partitioning the free space, and obtaining the connectivity information. The second part is focused on determining a correspondence between a trajectory and the topology of the robot workspace. Furthermore, this correspondence should be invariant with respect to homotopic trajectories.

In this paper we use a convex cell decomposition to describe the topology of the free space, and we show that a geometric relation can be established between the trajectories that belong to the same homotopy class and the union of a subset of the convex partitions of the workspace. This will allow us to decouple the solution of the planning problem from imposing homotopy class constraints.

A. Free Space Partitioning

A convex polygonal decomposition [11] is a finite collection of convex polygons $c_i \in \mathcal{C}$, where each convex polygon c_i is called a *cell*, such that the union of the cells is equal to the closed interior of the obstacle free region, i.e., $\bigcup\{c_i \in \mathcal{C}\} = cl(\mathcal{P}_{free})$.

Once the free space is decomposed into convex cells, the topology information can be captured in the form of an *adjacency graph*. Two cells c and c' are adjacent if $c \cap c'$ is a line segment of non-zero length. The associated non-directed graph, \mathcal{G} , is then formed such that each cell corresponds to a node in \mathcal{G} , and two nodes are connected by an edge if the corresponding cells are adjacent.

In this work, we use a common cell-decomposition approach, named *trapezoidal decomposition* [12], which decomposes \mathcal{P}_{free} into a set of trapezoidal or triangular cells by extending rays normal to a decomposition direction from each of the vertices of the polygonal obstacles. Furthermore, we have applied the same strategy as in [13], which decomposes the space along the direction of motion, i.e., the straight line connecting the initial and goal positions, by extending rays normal to this direction.

B. Homotopy Class Invariant

Recall that $\phi(\lambda)$ is the geometric path which corresponds to the trajectory $\sigma(t)$, and denote the set of independent variables λ as $\Lambda = \{\lambda_i\}$ where $0 \leq \lambda_i \leq 1$. Then, there exists a convex cell, c_i , such that for $\lambda \in (\lambda_{i-1}, \lambda_i)$ the path $\phi(\lambda)$ is confined in c_i , and for $\lambda = \lambda_{i-1}$ and $\lambda = \lambda_i$ the path intersects the boundaries shared with c_{i-1} and c_{i+1} , i.e., $\phi(\lambda_{i-1}) \in c_{i-1} \cap c_i$ and $\phi(\lambda_i) \in c_i \cap c_{i+1}$. As a consequence, one can define a sequence of cells $\mathcal{CS} = \{c_0 \rightarrow c_1 \rightarrow \dots \rightarrow c_f\}$ that fully contains $\phi(\lambda)$.

The canonical representation of a cell sequence can be defined as its standard representation after eliminating the repetitive sub-sequences. For example, for the two trajectories shown in Figure 1(a) with their corresponding cell sequences, $\mathcal{CS}_1 = \{c_0 \rightarrow c_2 \rightarrow c_3 \rightarrow c_5 \rightarrow c_4\}$ (blue) and $\mathcal{CS}_2 = \{c_0 \rightarrow c_2 \rightarrow c_3 \rightarrow c_5 \rightarrow c_3 \rightarrow c_5 \rightarrow c_4\}$ (red), the canonical representations can be computed by eliminating the repetitions. Since \mathcal{CS}_1 does not have any repetitive sub-sequences, it is already a canonical representation, while \mathcal{CS}_2 can be reduced to its canonical representation $\mathcal{CS}_2^{can} :=$

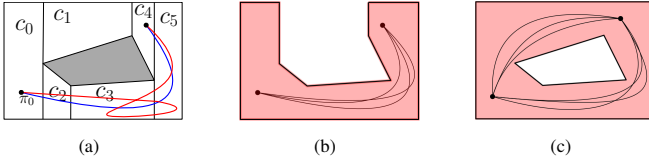


Fig. 1. (a) Two trajectories having different cell sequences that can be reduced into the same canonical representation. (b) The polygon that corresponds to the cell sequence $\{c_0 \rightarrow c_2 \rightarrow c_3 \rightarrow c_5 \rightarrow c_4\}$. (c) The polygon that corresponds to the union of two distinct loopless cell sequences $\{c_0 \rightarrow c_2 \rightarrow c_3 \rightarrow c_5 \rightarrow c_4\}$ and $\{c_0 \rightarrow c_1 \rightarrow c_4\}$ has a hole.

$\{c_0 \rightarrow c_2 \rightarrow c_3 \rightarrow c_5 \rightarrow \cancel{c_3} \rightarrow \cancel{c_5} \rightarrow c_4\}$. Finally, both trajectories share the same canonical representation.

Let \mathcal{P}' denote the polygon induced by the union of the cells in a loop-less cell sequence \mathcal{CS} , i.e. $\mathcal{P}' = \bigcup\{c_i \in \mathcal{CS}\}$. The geometric paths of any two trajectories sharing the same canonical representation, \mathcal{CS} , are entirely confined in \mathcal{P}' .

In the following, we show that the polygon \mathcal{P}' is a homotopy class invariant.

Proposition 1. *The set of all trajectories with the same initial and final positions, that share the same canonical representation corresponding to a loop-less cell sequence, are homotopic.*

Proof. Let \mathcal{P}' be the polygonal area corresponding to the cell sequence \mathcal{CS}' , any path with the same canonical cell sequence representation would be confined in \mathcal{P}' . Since the cell sequence does not have any loop, it does not encircle any obstacle, then the interior of \mathcal{P}' does not have any hole (Figure 1(b)). Without loss of generality, we assume that the initial and goal cells are not adjacent otherwise the interior of \mathcal{P}' might have a hole¹.

By definition, $\mathcal{P}' \subset \mathcal{P}_{free}$. Therefore, for the set of all the trajectories Σ' that can be projected to a set of geometric paths Φ' with the same initial and final positions, and such that if $\phi' \in \Phi'$ then $\phi'(\lambda) \in \mathcal{P}'$ for $\lambda \in [0, 1]$, then there exists a continuous mapping $\psi : [0, 1] \times [0, 1] \rightarrow \mathcal{P}'$ that deforms any $\phi'_i \in \Phi'$ to any other $\phi'_j \in \Phi'$. \square

Note that, the above proposition refers to a region that can be used as a reference for a set of trajectories belonging to the same homotopy class, but it is not restricted to the one-to-one correspondence between a trajectory and its corresponding cell sequence. This means that the set of all trajectories that can be mapped to the same polygonal area defined as the union of cells in a loop-less cell sequence – in other words, the set of trajectories whose geometric path corresponds to a loop-less cell sequence in its canonical form – are homotopic if they share the same initial and final position.

Proposition 2. *Two different sets of trajectories, having two different loop-less cell sequences as their canonical representations, belong to two distinct homotopy classes.*

Proof. Let \mathcal{P}_1 and \mathcal{P}_2 denote the free space obtained as the

¹Note that, this situation can be easily avoided by adding a pseudo-obstacle that ensures the initial and the goal cells do not share a boundary, this excludes the cell sequence composed of these two only.

union of convex cells in loop-less cell sequences \mathcal{CS}_1 and \mathcal{CS}_2 , i.e., $\mathcal{P}_1 = \bigcup\{c_i \in \mathcal{CS}_1\}$ and $\mathcal{P}_2 = \bigcup\{c_i \in \mathcal{CS}_2\}$. Furthermore, let \mathcal{P}' be the union of these two polygons, i.e., $\mathcal{P}' = \mathcal{P}_1 \cup \mathcal{P}_2$. Since the two cell sequences \mathcal{P}_1 and \mathcal{P}_2 are loop-less and different, there exists at least one cell in \mathcal{CS}_1 that is not shared in \mathcal{CS}_2 .

Assuming that the workspace is bounded, and workspace boundaries can be considered as boundaries of an obstacle, each convex cell has two boundaries with obstacles. For this reason, the polygon \mathcal{P}' obtained as the union of convex cells in \mathcal{CS}_1 and \mathcal{CS}_2 has holes, and the space contained inside the outer boundary is not continuous (Figure 1(c)). Therefore, a continuous mapping that deforms a trajectory with a geometric path $\phi_1 \in \mathcal{P}_1$ into another whose geometric path is $\phi_2 \in \mathcal{P}_2$ does not exist. This means that two trajectories having different loop-less cell sequences as their canonical representation belong to two distinct homotopy classes. \square

Proposition 2 states that the union of the convex cells in a loop-less cell sequence is a unique classifier for a homotopy class, therefore it can be used to distinguish different classes.

IV. HOMOTOPY AWARE KINODYNAMIC RRT*

The homotopy class invariant introduced in the previous section can be used within a RRT-based planning framework to impose the desired obstacle avoidance characteristic, while ensuring that the solution is feasible given a set of differential constraints (1). In particular, an asymptotically optimal RRT-based planner, RRT* [8], is used to obtain homotopically distinct (asymptotically) optimal trajectories.

Before introducing the proposed homotopy aware kinodynamic RRT*, we briefly recall the main steps of kinodynamic RRT*. The algorithm builds a tree $T = (V, E)$, whose nodes, V , are states of the dynamic system (1), and edges, E , are optimal trajectories that connect two nodes solving a two point boundary value problem. The tree is initialized with $V = \{s_0\}$, where $s_0 \in S_{free}$ is the initial state, and $E = \emptyset$. Every node $s \in V$ is connected to s_0 via a single sequence of intermediate nodes $s_i \in V$, $i = 1, \dots, n-1$, $n \leq N$, and associated edges $e_i = e_{s_i, s_{i+1}} \in E$, $i = 0, 1, \dots, n-1$, with $s_n = s$. One can then associate to this sequence a cost $C(\rightarrow s_n)$ given by

$$C(\rightarrow s_n) = \sum_{i=0}^{n-1} C(e_i)$$

where $C(e_i)$ denotes the cost associated with edge $e_i \in E$, and computed as in (2).

Tree expansion continues until a maximum number of iterations, N , defined by the user, is reached. At each iteration a random node, s_{rand} , is sampled from the free space, S_{free} to be connected to the tree. The parent node, s_{parent} , for s_{rand} is selected among a set of nodes within the local neighbourhood of s_{rand} , i.e., S_{near} . For each node $s_{near} \in S_{near}$, the optimal trajectory connecting s_{near} to s_{rand} is computed which minimizes the cost function (2) and satisfies the differential constraints (1). Consecutively,

the parent node, $s_{parent} \in S_{near}$, is the node in S_{near} that minimizes the cost $\text{Cost}(\rightarrow s_{rand})$ and for which the trajectory connecting s_{parent} with s_{rand} is collision-free. Every time a new node, s_{rand} , is added to the tree it is checked if existing nodes in the neighbourhood of s_{rand} can be connected to the tree through s_{rand} , and the nodes that satisfy this criteria are *rewired*, i.e., the previous edges that connect them are removed from the tree and new edges rooting from s_{rand} are added. Finally, the trajectory that reaches the goal with minimum cumulative cost is returned.

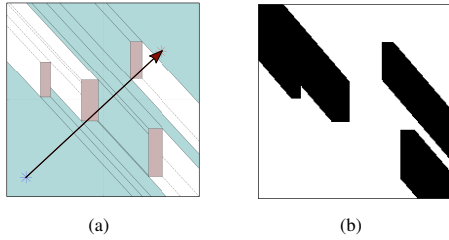


Fig. 2. Feasible region for homotopy aware RRT*. (a) Homotopy class (blue region) defined as a union of a sequence of convex cells from the initial cell (containing blue star) to a final one (containing red star). The workspace is decomposed along the direction of motion (black solid line with arrowhead). (b) Corresponding feasible region (in the form of an occupancy grid) that correspond to the cell sequence, free space is shown in white.

From Proposition 1, it can be concluded that any two trajectories having the same initial and final positions, whose projections onto the 2D plane are confined in a polygonal region defined as a union of loop-less sequence of convex cells, are homotopic. This allows us to define a homotopy class constraint as a feasible region (Figure 2), without explicitly tracking the corresponding sequence of cells passed for each node that is added to the tree. Any trajectory that can be projected onto this free space will be homotopic to the others, as long as they share the same initial and final positions, since they share the same canonical cell sequence. Once the feasible region that corresponds to the homotopy class is determined, RRT* can be modified to ensure that the tree built by the algorithm lies within it.

Given a loop-less cell sequence $\mathcal{CS} = \{c_0 \rightarrow c_1 \rightarrow \dots \rightarrow c_f\}$, and the corresponding polygonal region, i.e. $\mathcal{P}' = \bigcup \{c_i \in \mathcal{CS}\}$, the free space is defined as

$$S'_{free} = \{s \in \mathcal{S}: \mu(s) \in \mathcal{P}'\}$$

which corresponds to the set that RRT* samples at each iteration to obtain a random state, s_{rand} , for expanding the tree. Furthermore, introducing this definition of the free space, the collision check ensures that the tree of trajectories lies entirely in the selected polygon, hence, the trajectory reaching the goal would lie in the selected homotopy class.

A. Topology based informed sampling

The exact decomposition of the free space is also useful for making informed decisions during the sampling phase. In particular, the shortest geometric path can be used to sample in a restricted promising region.

In order to avoid confusion, in this section a shortest geometric path represents the shortest path connecting the initial position to the final one, neglecting system dynamics, in the polygon which corresponds to the homotopy class. Therefore, an approach similar to Theta*-RRT [14] – which first computes the shortest path in an environment using the Theta* algorithm [15], and then draws samples in the vicinity of this path while solving for the kinodynamic planning problem using RRT – can be adopted in order to impose the full dynamics using RRT* framework while biasing the sampling using the shortest path.

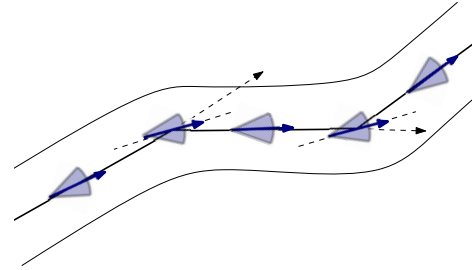


Fig. 3. Flow of the shortest geometric path. Solid black line at the center represents the shortest path found on the visibility graph (the sampleable zone around the shortest path is also shown). The sampling range allowed for orientations is shown with blue cones.

In order to use an informed sampling approach based on the shortest path in a polygon, the shortest path that connects the initial position to a final one inside the polygonal region \mathcal{P}' , which corresponds to the desired cell sequence \mathcal{CS} , is first computed. To this end, the visibility graph containing the line segments that are confined inside the designated polygonal zone is traversed using the A* algorithm [16].

Once the shortest path is found, it is possible to define a sampling set that restricts the states to be sampled to the ones that are in the vicinity of this path. Note that, no other restrictions are considered for the remaining states, e.g., on the velocity components or on the higher order derivatives. This results in a more focused set, that contains promising samples which can contribute to finding lower-cost trajectories. In particular, a sampleable zone is designed such that the distance between the position component of any drawn sample and the shortest geometric path is not larger than a predefined threshold², and the orientation of a sample is bounded by the limits determined by the *flow* of this shortest path (Figure 3). For the samples that are in the vicinity of a line segment the orientation bounds are determined such that, $\theta \in (\theta_i^i - \theta_{th}, \theta_i^i + \theta_{th})$, where θ is the orientation component of the state to be sampled, θ_i^i is the orientation of the i^{th} line segment which is a part of the shortest geometric path, and θ_{th} is a threshold that determines a cone of desired orientation. Furthermore, for the circular zone around the points that correspond to the intersection of two consecutive line segments, characterized by the radius equal to the distance threshold, the bounds are computed using the minimum and the maximum among

²The distance is computed using Euclidean norm.

the two limits determined by the two consecutive lines. The same is also performed for the initial and the final positions, using the initial and final states and the corresponding line segments.

The completeness of the algorithm is not compromised with the inclusion of this biased sampling assuming *small-time locally controllable* systems, such that a Euclidean ball centered at any state, s , contains a neighborhood of s , i.e., a d -dimensional set of states including s , that are reachable from s in time $0 < t \leq T$, $\forall T > 0$ [17]. The completeness then follows from [8], such that for any homotopy class which admits a trajectory with ϵ -clearance from the obstacles, where $\epsilon \geq 0$, the probability of the algorithm not finding a solution is 0 as the number of iterations tends to infinity. Note that, if the system at hand is not small-time locally controllable, for example a car-like robot with path curvature constraint, this decoupled approach may fail.

B. Multiple distinct trajectories

Beside constraining the solution to the selected homotopy class, homotopy aware RRT* can also be used to return homotopically distinct trajectories. Given a cell-sequence selection strategy a set of promising classes can be preliminarily identified, for example as in [13] that exploits a graph traversal algorithm adopting a *Depth First Search* (DFS) strategy to extract the branches (cell sequences) of the adjacency graph, \mathcal{G} , from an initial node to the final one, while imposing some constraints on their characteristics. In particular, the algorithm stops exploring a branch if it contains a duplicate of a node, or if the total number of nodes exceeds a threshold value.

Once a set of promising homotopy classes has been identified, each problem can be solved independently from the others. This also allows the overall problem of determining distinct trajectories to be solved in a parallel computing scheme.

V. NUMERICAL EXAMPLE

In order to show the effectiveness of the algorithm, we provide a numerical example. A 4D state space is considered, representing a unicycle-like robot moving on a planar surface. The equations of motion are given by

$$\begin{aligned} \dot{x} &= v \cos \theta & \dot{\theta} &= \omega \\ \dot{y} &= v \sin \theta & \dot{v} &= a \end{aligned} \quad (3)$$

where (x, y) and θ are the robot position and orientation with respect to a global reference frame, v and ω are the linear and angular velocities, and a is the linear acceleration. The model given in (3) is feedback linearized to obtain the following double integrator model

$$\begin{aligned} \dot{x} &= v_x & \dot{v}_x &= u_1 \\ \dot{y} &= v_y & \dot{v}_y &= u_2 \end{aligned} \quad (4)$$

where $v_x = v \cos(\theta)$ and $v_y = v \sin(\theta)$. We can then apply the approach in [18] to obtain the optimal trajectory

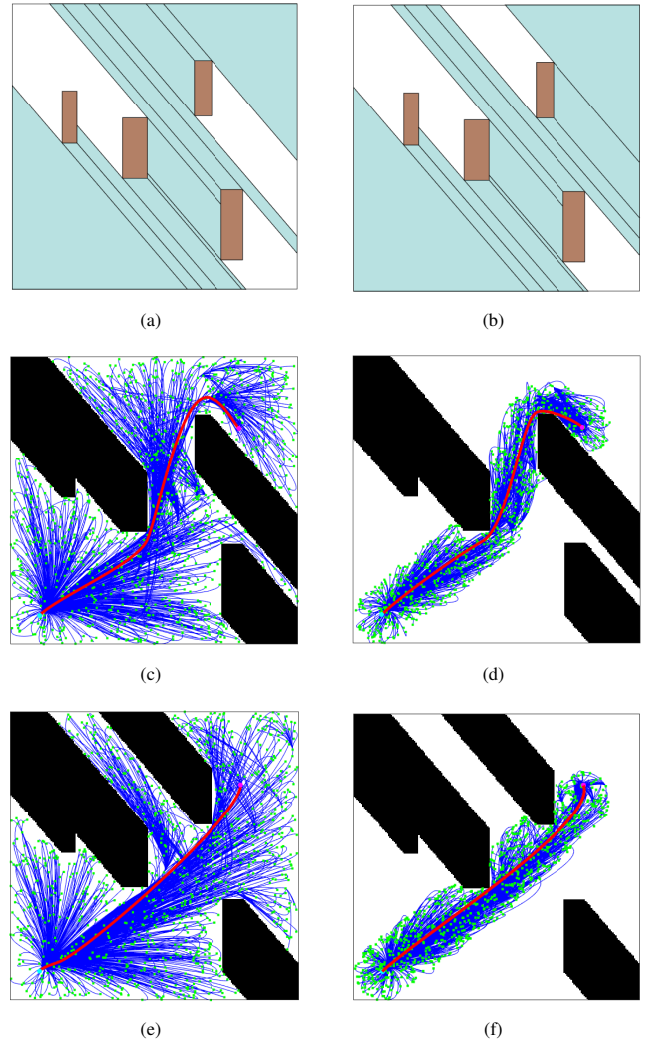


Fig. 4. For two different homotopy classes, $\Sigma_{H_1}, \Sigma_{H_2}$: (a) cell sequence corresponding to Σ_{H_1} (b) cell sequence corresponding to Σ_{H_2} (c-d) the resulting tree and the best cost trajectory at 10^3 iterations for Σ_{H_1} using uniform and informed sampling. (e-f) the resulting tree and the best cost trajectory at 10^3 iterations for Σ_{H_2} using uniform and informed sampling.

connecting two states $s_i = [x_i, y_i, \theta_i, v_i]^T$ and $s_{i+1} = [x_{i+1}, y_{i+1}, \theta_{i+1}, v_{i+1}]^T$ considering the following cost function

$$J(u) = \int_0^\tau (1 + u^T R u) dt \quad (5)$$

which minimizes the total time, τ , of the trajectory while penalizing the control input for the double integrator model, $u = [u_1, u_2]^T$, using the weight $R = 0.25I_2$.

The planning environment has the size $10 \times 10 m$ and it is populated with 4 obstacles. The algorithm is constrained to return a solution that follows a selected homotopy class given as a sequence of convex cells that partition the robot workspace. With a slight abuse of the previously introduced definition of homotopic trajectories, we consider any trajectory having its final position inside a $0.1 \times 0.1 m$ sized region centered at the goal position as a solution.

Once the desired homotopy class is given in terms of a

cell sequence from an initial cell to a final one, the algorithm starts computing the free region (Figure 2). Figure 4 shows the trajectories computed after 10^3 iterations for the homotopy classes given in Figure 4(a,b) using different sampling techniques. Clearly, the homotopy class restrictions are satisfied since the corresponding occupancy grid for collision-detection assigns every cell other than those corresponding to the desired cell sequence as an obstacle.

Using the shortest path in the polygon representing the corresponding homotopy class allows us to explore a more restricted area focusing on the samples that can contribute to find a trajectory that solves the homotopic trajectory planning problem. Figure 5 shows the average solution cost for 10 independent simulations for the homotopy class given in Figure 4(a) as a function of the number of iterations for each of the sampling strategies (uniform and informed). It is clear that, using an informed sampling that accounts for the *flow* of the shortest path provides better solutions faster compared to sampling the complete free region. Furthermore, it also allows to find the first solution at a lower number of iterations. Though arguing for the optimality when such a sampling strategy is used is problem dependent and out of the scope of this work, we can conclude, based on the empirical results, that the proposed sampling strategy increases the convergence rate to lower cost solutions.

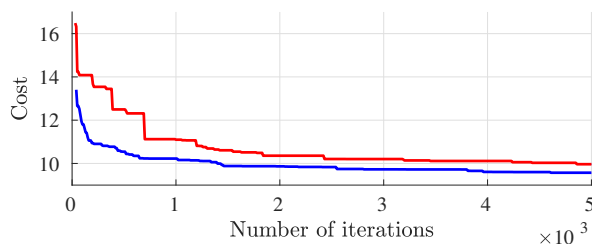


Fig. 5. Solution cost as a function of the number of iterations for uniform sampling (red solid line) and for informed sampling (blue) strategies.

Note that, other than imposing certain rules on how a trajectory should avoid obstacles, identifying a number of homotopy classes is also beneficial for biasing the sampling along the shortest path for each of the classes. Especially for systems with dynamic constraints or with bounds on actuation as in these cases obtaining a shortest path in a homotopy class does not necessarily guarantee the existence of a feasible trajectory.

VI. CONCLUSIONS

In this paper a homotopy aware variant of RRT* is presented, which restricts the tree growth to obtain solution trajectories that belong to a selected homotopy class. In particular, a homotopy class constraint based on the convex cell decomposition which can easily be included within the RRT-based planner scheme is proposed. Integration of the proposed constraint does not call for any modification in the original algorithm.

A theoretical analysis is also presented, to show that as long as loop-less cell sequences are considered, the homotopy class identifier is invariant with respect to trajectories that belong to same class, and unique to the designated class.

ACKNOWLEDGMENT

Published in the context of the project TEINVEIN: Tecnologie INnovative per i VEicoli Intelligenti, CUP (Codice Unico Progetto - Unique Project Code): E96D17000110009 – Call “Accordi per la Ricerca e l’Innovazione”, cofunded by POR FESR 2014-2020 (Programma Operativo Regionale, Fondo Europeo di Sviluppo Regionale – Regional Operational Programme, European Regional Development Fund).

REFERENCES

- [1] M. Kuderer, C. Sprunk, H. Kretzschmar, and W. Burgard, “Online generation of homotopically distinct navigation paths,” in *IEEE International Conference on Robotics and Automation*, 2014, pp. 6462–6467.
- [2] S. Bhattacharya, M. Likhachev, and V. Kumar, “Topological constraints in search-based robot path planning,” *Autonomous Robots*, vol. 33, no. 3, pp. 273–290, 2012.
- [3] S. Bhattacharya, V. Kumar, and M. Likhachev, “Search-based path planning with homotopy class constraints,” in *Third Annual Symposium on Combinatorial Search*, 2010.
- [4] B. Donald, P. Xavier, J. Canny, and J. Reif, “Kinodynamic motion planning,” *Journal of the ACM*, vol. 40, no. 5, pp. 1048–1066, 1993.
- [5] S. J. Anderson, S. B. Karumanchi, K. Iagnemma, and J. M. Walker, “The intelligent copilot: A constraint-based approach to shared-adaptive control of ground vehicles,” *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 2, pp. 45–54, 2013.
- [6] J. Park, S. Karumanchi, and K. Iagnemma, “Homotopy-based divide-and-conquer strategy for optimal trajectory planning via mixed-integer programming,” *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1101–1115, 2015.
- [7] S. M. LaValle and J. J. Kuffner Jr, “Randomized kinodynamic planning,” *The international journal of robotics research*, vol. 20, no. 5, pp. 378–400, 2001.
- [8] S. Karaman and E. Frazzoli, “Optimal kinodynamic motion planning using incremental sampling-based methods,” in *IEEE Conference on Decision and Control*, 2010, pp. 7681–7687.
- [9] E. Hernandez, M. Carreras, and P. Ridao, “A comparison of homotopic path planning algorithms for robotic applications,” *Robotics and Autonomous Systems*, vol. 64, pp. 44–58, 2015.
- [10] K. D. Jenkins, “The shortest path problem in the plane with obstacles: A graph modeling approach to producing finite search lists of homotopy classes,” Ph.D. dissertation, Monterey, California. Naval Postgraduate School, 1991.
- [11] J.-C. Latombe, *Robot motion planning*. Springer Science & Business Media, 2012.
- [12] B. Chazelle, “Approximation and decomposition of shapes,” *Algorithmic and Geometric Aspects of Robotics*, vol. 1, pp. 145–185, 1985.
- [13] B. Sakcak, L. Bascetta, and G. Ferretti, “An exact optimal kinodynamic planner based on homotopy class constraints,” in *Modelling and Simulation for Autonomous Systems. MESAS 2017. Lecture Notes in Computer Science*, vol. 10756. Springer, Cham, 2018, pp. 140–152.
- [14] L. Palmieri, S. Koenig, and K. O. Arras, “RRT-based nonholonomic motion planning using any-angle path biasing,” in *IEEE International Conference on Robotics and Automation*, 2016, pp. 2775–2781.
- [15] A. Nash, K. Daniel, S. Koenig, and A. Felner, “Theta*: Any-angle path planning on grids,” in *AAAI Conference on Artificial Intelligence*, 2007, pp. 1177–1183.
- [16] J. Pearl, “Heuristics: intelligent search strategies for computer problem solving,” 1984.
- [17] H. M. Choset, *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.
- [18] D. J. Webb and J. Van Den Berg, “Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics,” in *IEEE International Conference on Robotics and Automation*, 2013, pp. 5054–5061.