The Version of Record is available online at: http://dx.doi.org/10.1007/s41315-024-00330-5

A State-of-the-Art Review on Topology and Differential Geometry-Based Robotic Path Planning—Part I: Planning Under Static Constraints

Sindhu Radhakrishnan[1] and Wail Gueaieb[1*]

[1*]School of Electrical Engineering and Computer Science, University of Ottawa, 800 King Edward Avenue, Ottawa, K1N 6N5, ON, Canada.

*Corresponding author(s). E-mail(s): wgueaieb@uottawa.ca; Contributing authors: sradh006@uottawa.ca;

Abstract

Autonomous robotics has permeated several industrial, research and consumer robotic applications, of which path planning is an important component. The path planning algorithm of choice is influenced by the application at hand and the history of algorithms used for such applications. The latter is dependent on an extensive conglomeration and classification of path planning literature, which is what this work focuses on. Specifically, we accomplish the following: typical classifications of path planning algorithms are provided. Such classifications rely on differences in knowledge of the environment (known/unknown), robot (model-specific/generic), and constraints (static/dynamic). This classification however, is not comprehensive. Thus, as a resolution, we propose a detailed taxonomy based on a fundamental parameter of the space, i.e. its ability to be characterized as a set of disjoint or connected points. We show that this taxonomy encompasses important attributes of path planning problems, such as connectivity and partitioning of spaces. Consequently, path planning spaces in robotics may be viewed as simply a set of points, or as manifolds. The former can further be divided into unpartitioned and partitioned spaces, of which the former uses variants of sampling algorithms, optimization algorithms, model predictive controls, and evolutionary algorithms, while the latter uses cell decomposition and graph traversal, and sampling-based optimization techniques.This article achieves the following two goals: The first is the introduction of an all-encompassing taxonomy of robotic path planning. The second is to streamline the migration of path planning work from disciplines such as mathematics and computer vision to robotics, into one comprehensive survey. Thus, the main contribution of this work is the review of works for static constraints that fall under the proposed taxonomy, i.e., specifically under topology and manifold-based methods. Additionally, further taxonomy is introduced for manifold-based path planning, based on incremental construction or one-step explicit parametrization of the space.

Keywords: Path planning, Robotics, Manifolds, Topology

Abbreviations

APF Artificial Potential Field
CCM Closed Chain Manipulator
CDGT Cell Decomposition and Graph Traversal
DOF Degrees Of Freedom
DTD Dynamic Topology Detector
EA Evolutionary Algorithm
EE End-Effector
GVD Generalized Voronoi Diagram
MBM Model Based Methods
MPC Model Predictive Control
NF Navigation Functions
OA Optimization Algorithm
OCM Open Chain Manipulator
PDR Path Deformation Roadmap
PPP Path Planning Problem
PPS Path Planning Space
PRM Probabilistic Road Map
RRT Rapidly exploring Random Tree
SA Sampling Algorithm
SO Special Orthogonal Group
VG Visbility Graph

List of Latin Symbols

$\mathcal{C}$ Configuration space
$\mathcal{C}^{free}$ Configuration space free of constraints/obstacles
$\boldsymbol{f}^{fwd}$ Mapping defining forward kinematics
$\boldsymbol{f}^{inv}$ Mapping defining inverse kinematics
$\mathcal{P}$ Path planning space

$\mathcal{P}^{constraint} \equiv \mathcal{P}^{\text{mechanical limits}} \cup \mathcal{P}^{\text{no-go-zones}} \cup \mathcal{P}^{\text{obstacles}} \cup \mathcal{P}^{\text{singularities}}$ Space corresponding to all constraints

$\mathcal{P}^{free} \equiv \mathcal{P} \setminus \mathcal{P}^{\text{constraint}}$ Constraint-free Space

$\boldsymbol{p} \in \mathcal{P}$ A point in $\mathcal{P}$

$\mathcal{W}$ Workspace

1 Introduction

Planning a path for a robot to follow is an integral component of a paramount importance in any robotic application. The choice of a path planning algorithm is made based on some general requirements and others that are application-specific. An example is that of an automation chain in a factory or a semi/fully-autonomous warehouse. Here, robots have to be aware of their surroundings and must meet real-time constraints to interact with humans within their working environment while avoiding collisions with them and any obstacles that may be on the way such as other autonomous robots, for instance [1]. Such requirements require sifting through the vast path planning literature, and result in a subset of algorithms best suited for the application. Choosing an algorithm based on such requirements involves studying the classification of path planning approaches.

This work is the first in a series of two articles that comprehensively explore path planning algorithms that are centered around topology and differential geometry-based methods. Tackling the problem from this perspective distinguishes our survey from previous works and makes it unique. This paper focuses on path planning techniques under static constraints, while the second article in the series [2] covers techniques that are more suitable for dynamic constraints. In this work, we propose that path planning algorithms be classified based on the nature of the Path Planning Space (PPS). We do so, by first briefly discussing the taxonomies most commonly seen in literature. Since the demerits of existing taxonomies preclude the analysis of the nature of the PPS, we propose a classification that remedies the demerits. The proposed classification is general and encompasses all other traditional classifications, as it is based on the nature of the path planning space. Subsequently, it segments the literature into two main categories: set of points and topology/manifolds-based approaches. Then, we provide a comprehensive literature review of those path planning algorithms building on concepts from topology/manifolds. The aim of this article is to provide a comprehensive literature review of only those path planning algorithms that utilize notions of topology and manifold theory. This effort aims to minimize the gap in robotics literature pertaining to such a specific review of works in the area.

The three common taxonomies that divide the path planning literature are as follows. The first category classifies approaches based on whether or not the path planning environment is known apriori. Known environments can exploit approaches such as graphs, occupancy grids and any Cell Decomposition and Graph Traversal (CDGT) techniques [3–12]. Unknown environments can rely on Sampling Algorithms (SAs), such as Probabilistic Road Maps (PRMs) [13–21], Rapidly exploring Random Trees (RRTs) [6–9, 13, 14, 16, 22–28], and other variants of both. Path planning algorithms generally consist of two tasks. The first is generating a path segment towards the goal and the second is evaluation of the path segment for collision avoidance. The second category distinguishes algorithms based on which of the two tasks is optimized for path metrics. For example, Model Predictive Control (MPC) methods produce paths that are compatible with specific robot models and then evaluate the produced paths for collision [29–37]. Contrarily, force field methods, such as Artificial Potential Fields (APFs) [37–44] and Navigation Functionss (NFs) [45–50] focus on producing collision-free paths, which can later be processed for the robot models' configuration and workspaces. The third classification separates algorithms based on their ability to avoid static or dynamic constraints. Avoiding static constraints can be viewed as planning with knowledge of constraints, so it is deliberate collision avoidance. On the other hand, impromptu encounters of unknown static or dynamic constraints prompts the algorithm to reactively avoid them. Usually, this classification consists of variants of algorithms belonging to the first two categories. The classifications may be viewed pictorially in Fig. 1. Clearly, all classifications may share some algorithms, while also possessing some variants that are unique to the classification. The aforementioned distinctions in categories enable the choice of a suitable path planning algorithm for a specific application. However, viewing the Path Planning Problem (PPP) via the three categories has its advantages and disadvantages, which are now discussed.

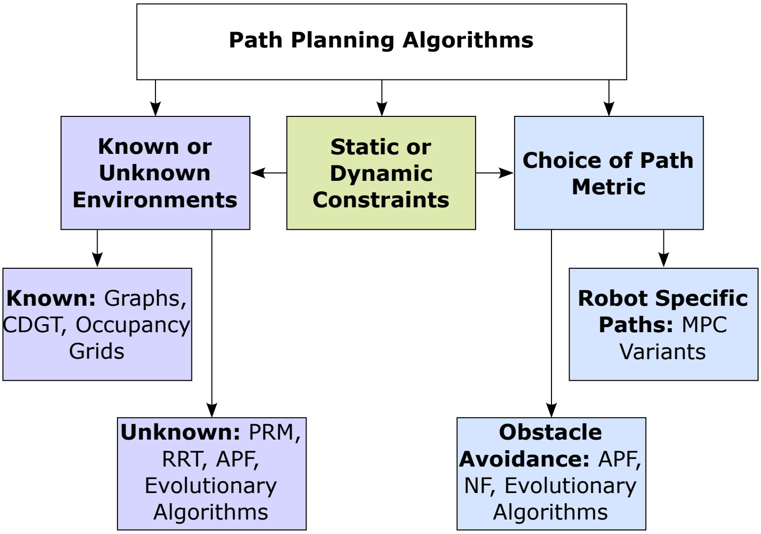

Fig. 1: Three common taxonomies in path panning literature

The three categories enable one to look at the PPP via different resource lenses. One may choose a suitable path planner based on the abundance or dearth of map information, or the precedence of robot specific path segments over the fastest collision avoidance path. However, the classifications still share a significant overlap of approaches, while missing an important branch of path planning literature. To minimize the overlap and make the classification more intuitive from the grassroots level, we propose a classification based on the nature of the PPS. That is, the PPS may be categorized as: (i) not having an explicit mathematical structure, such as a set of points, or (ii) possessing a mathematical structure of smoothness, such as manifolds. Now, using this taxonomy, the path planning literature can be seen to consist of approaches, such as SAs and Optimization Algorithms (OAs) for the former, and manifold-based methods for the latter. This taxonomy is seen in Fig. 2. Following such taxonomy then, one can easily insert the original three classifications under the respective division of point-set or manifold-based methods, if such a detailed branch structure is desired.

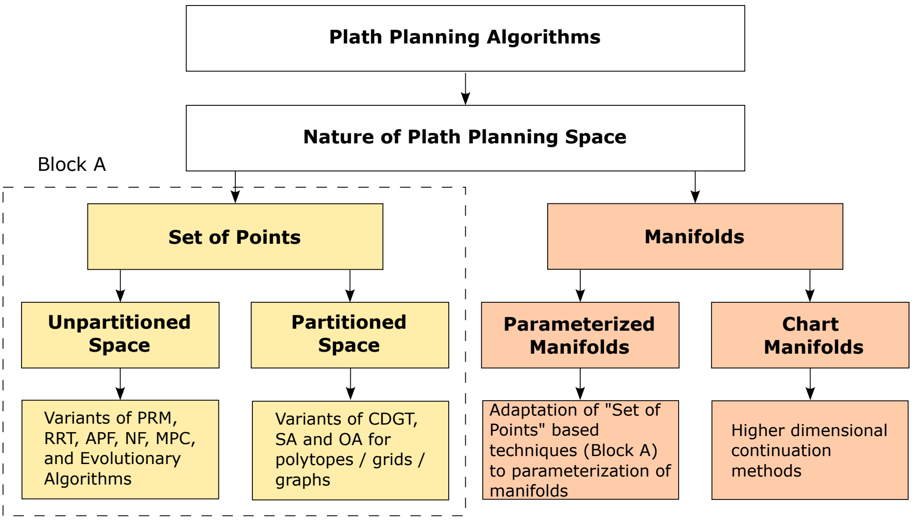

Fig. 2: Taxonomy based on the nature of the PPS

In this article, a comprehensive literature review of path planning algorithms in the manifold-based methods is provided. To establish context for work related to manifold based methods, a brief introduction to the algorithms pertaining to the point-set based PPS is provided in Section 2. Following this, Section 3 provides a precinct for manifold-based methods. These methods can further be categorized into two: path planning methods on parametrizations of manifolds as seen in Section 4 and path planning on manifolds that can be charted, as seen in Section 5. Finally, Section 6 concludes the literature review.

2 Planning on Point Set-Based Path Planning Space

The most fundamental representation of a path planning environment is by representing it as a union of all possible points that constitute the PPS. A PPS can be any space used in the path search, such as the configuration space $\mathcal{C}$, the workspace $\mathcal{W}$, joint velocity space, End-Effector (EE) velocity space, joint torque space, and EE force space, etc.. The point-set representation of the PPS is defined as follows. Denote a point in an n-dimensional Euclidean space $\mathcal{R}^n$ as $\mathbf{p}$, where $\mathbf{p} = \begin{bmatrix} p_1 & p_2 & \cdots & p_n \end{bmatrix}^T$. Define the span for each component of $\mathbf{p}$ as follows: $p_i \in [p_i^{\text{min}}, p_i^{\text{max}}] = p_i^{\text{span}}$, where $i = 1 : n$. Now, define the PPS $\mathcal{P} \subset \mathcal{R}^n$ as follows: $\mathcal{P} = \mathbf{p}^{\text{all}} = \prod_{i=1}^{n} p_i^{\text{span}}$. Thus, $\mathcal{P}$ as defined here, constitutes the point-set representation of the PPS. For the purposes of generality, the notation of $\mathcal{P}$ is used to denote any PPS.

A PPP needs to represent the region occupied by the robot, the constraint region consisting of obstacles, no-go zones and singularities, and the constraint-free region. The point-set representation allows two definitions of space: (i) un-partitioned and (ii) partitioned space. The former is a collection of disjoint discrete points with no explicit boundary association. The latter however, is a collection of discrete points satisfying explicit span or bounds. Examples of the former include the robot's pose at a given time, constraints such as robot singularities, obstacles and no-go zones. Examples of the latter include characterizing constraints using boundaries to enclose the point-set representing the constraints.

Unpartitioned spaces facilitate sampling algorithms, such as PRM and RRT-based variants, and OAs, such as APF, MPC, and Evolutionary Algorithm (EA) based approaches. Partitioned spaces on the other hand, enable region characterization as polytopes. That is, they are still a union of discrete points, however, connectivity relationships are satisfied at least for points on the boundaries of polytopes. A partitioned space facilitates a variety of approaches, including CDGT and sampling based OAs. It is in fact, common to notice a combination of such approaches and spaces to specifically suit the application. Since the focus of this article is manifold-based methods, the literature review pertaining to point-set based methods will not be elaborated much here. Instead, one may refer to Fig. 2 for a quick introduction to common methods associated with point-set approaches and to the following for an in-depth understanding:

- SAs [3–10, 13–28, 50–67], such as PRM- and RRT-based algorithms.
- OAs, such as APF in [33, 38–44, 68–70], MPC in [33] (OA-based) and [29–32, 34–37], EAs in [71–74], and NF in [50] (SA-based) and [45–49, 75].

3 Planning on Manifold-Based Path Planning Space

This section focuses on approaches that plan paths in spaces with an underlying structure. Such spaces contain points whose relationship to each other is defined by imposed characteristics. They include satisfying a system of equations, obeying laws of smoothness and adhering to a certain metric. One of the most obvious, intuitive characteristics of such spaces is that they are locally Euclidean, but not necessarily globally so (although they can be both). Such spaces are called manifolds. The advantage of formulating paths on manifolds is in utilizing the natural structure of the robotic workspace. This is especially of use when the resulting manifold of the configuration or workspace is free of constraints. Then, the path planning can proceed on the natural structure of manifold without having to repeatedly evaluate points in the space for constraint violation. Path planning on globally Euclidean spaces, such as planes, and globally non-Euclidean surfaces, such as manifolds, still aim to find an optimal path in their respective spaces. If the optimization is solely based on the path length, for example, then the shortest path in the former is a straight line as seen in Fig. 3. Its equivalent in the latter however, is called a geodesic as seen in Fig. 4. Informally, geodesics are the "straight lines" of manifolds, but appear non-straight since they adhere to the underlying curvature of the surface. Path planning between a source and a target configuration on curved surfaces therefore involves determining the geodesic, parts of it, or discrete approximations of either.

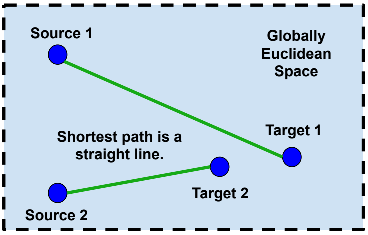

Fig. 3: A globally Euclidean space, such as a plane, where the shortest path between any two points is always a straight line.

Constructing geodesics requires knowledge of the manifold's representation. The difficulty lies in representing or characterizing the manifold. While some variants or shapes of manifolds exist that are represented using global parametrizations, others are seldom easily characterized by a global parametrization. The approaches to path planning therefore bifurcate into two main groups: the first group adapts methods used in the point-set space (such as SAs, APFs and CDGTs) to familiar manifolds. The second group incrementally builds charts or local representations of the unknown manifold to eventually build a global representation. Some planners simultaneously

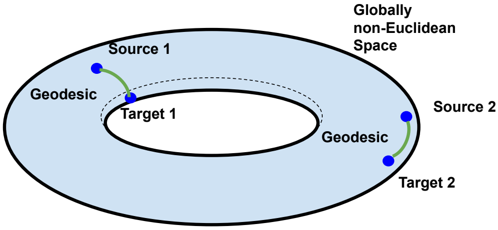

Fig. 4: A globally Non-Euclidean space, such as a Genus-1 Torus, where the shortest path between any two points is a geodesic.

attempt to search the built surface for a path. Others find paths after the manifold construction process. Sections 4 and 5 discuss the first and second groups, respectively.

4 Parametrization of manifolds

Point-sets in the workspace are often a union of points representing the constraints and those that represent the constraint free space, as seen in Fig. 5. It is from these point-sets that the surface denoting the constraint space $\mathcal{P}^{\text{constraint}}$ or the constraint-free space $\mathcal{P}^{\text{free}}$ is extracted mathematically. That is, the point-set is partitioned into $\mathcal{P}^{\text{constraint}}$ and $\mathcal{P}^{\text{free}}$ by employing systems of equations to construct surfaces from those regions. This can be especially challenging when both sets may not be compatible mathematically. The challenging nature of the problem led early researchers to approximate the PPS analytically in Section 4.1, geometrically in Section 4.2, numerically in Section 4.3, and graphically in Section 4.4. One may refer to Table 1 for a tabulated version of the works in this section.

Table 1: Overview of manifold based methods—Parametrization of manifolds

Parametrization method	OA	CDGT	APF	SA	EA	Topological tools	MBM	Numerical tools
Analytical approximation	[76, 77]	-	[45, 78]	-	-	-	[79–82]	-
Geometric approximation	[83]	[84]	[67, 85]	-	-	-	-	-
Numerical approximation	[86, 87]	-	[11]	-	-	-	-	[77, 88, 89]
Graphical approximation	[12, 89, 90]	[91]	[92]	[93]	[94]	[95]	-	-

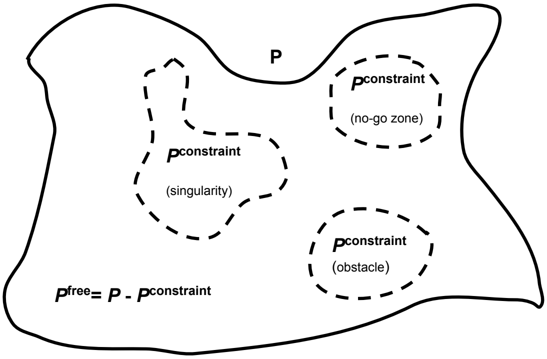

Fig. 5: Breakdown of PPS $\mathcal{P}$ into $\mathcal{P}^{\text{constraint}}$ and $\mathcal{P}^{\text{free}}$

4.1 Analytical Approximation of Path Planning Space

Analytical approximations of the PPS or its subsets rely on the characterization of the constraint and constraint-free spaces. However, it is challenging to do so because of two main reasons: the first deals with the availability of constraint information. The second involves a choice in characterizing the PPS. Both factors are now explained. First, information about the constraints must be available, since they help define the constraint-free space. Constraints can be any combination of several types: mechanical, kinematic, singularities, spacial obstacles, and no-go zones, as seen in Fig. 5. Since the constraints may be scattered across the path planning input and output spaces, the native space (input/output) of the constraints must first be accounted for. Then, one can define a constraint-free input/output space. However, the definition of a constraint space is still incomplete. All constraints must be accounted for in at least one of the path planning spaces for an unambiguous definition of the constraint space. This is done via a deliberate mapping between the input and output space. Now, one has a clear definition and choice of a constraint-free space, because the constraint-free property is now equivalent/transformable in the input and output spaces.

This section discusses path planning approaches taken to plan paths on analytical approximations of the PPS or its subsets as follows: Model Based Methods (MBM) in Section 4.1.1, OAs in Section 4.1.2, and force fields in Section 4.1.3.

4.1.1 Model-Based Methods

MBM rely on the knowledge of the PPS and/or that of the robot used. Attributes defining the former or latter are then used for path planning. The most fundamental approach that attempts this can be seen as early as 1983, where authors of [79] explored mapping the constraint space into the admissible configuration space. Although not explicitly defined as manifolds or geodesics, this paper is probably the earliest work on the projection of the constraint space into the admissible configuration space. The authors of [79] then find upper and lower bounds for the projected constraint space, and find paths in so called freeways, which are constraint-free regions represented by primary motions, such as a translation along a fixed axis.

Control theorists were familiar with the concepts of systems of equations defining surfaces, i.e., vector fields representing curvature and thus directions of motion leading to integral curves. Integral curves could then be viewed as feasible paths that the system could take. In the field of robotic path planning however, that was not the case. Authors of [80] were some of the early researchers who introduced the same concepts for path planning. They use kinematic constraints to come up with the structure of the manifold and the subsequent variety of paths resulting from such a structure. The work focuses on formulating the structure of the surface and move away from point-set based methods. This step is the precursor to constructing geodesics. By showing that vector fields already account for kinematic constraints, they show that the surface being operated on is free of kinematic constraints.

Paths on this kinematic constraint-free surface, then simply need to be checked for intersections with the obstacle region's boundary. Thus, feasible paths incorporate both kinematic and obstacle constraints. Since there may exist multiple feasible paths, a cost or metric is chosen for optimality. Authors of [80] choose the Riemannian metric, where-from one can compute the geodesics of that metric. Geodesics are paths corresponding to "straight lines" in the metric space. However, unlike straight lines, geodesics are not always feasible paths. In order to be so, they need to be in the total flow of the vector fields of control. In general, and especially for non-holonomic problems, no such situation exists. In that case, the geodesics are approximated by pieces of flow lines of the field. The discretization procedure of geodesics computation was accomplished by using the cost wave propagation algorithm [80].

Properties of homeomorphism and diffeomorphism (both rely on continuity) are intertwined with analytical approximations of the PPS. So clearly, continuity is an important aspect of the complexity of path planning, and more so on manifolds. In [81], Farber illustrates precisely this, i.e., the notion of topological complexity of the motion planning problem. The topological complexity is a number that measures discontinuity of the motion planning process in the configuration space. Farber establishes an upper and lower bound for the topological complexity as a function of the dimension of the configuration space. Farber uses examples of two-dimensional surfaces such as spheres, products of spheres, etc., which arguably form the most commonly seen manifolds in a systematic $\mathcal{W}$ or $\mathcal{C}$ derivation.

A specific extension to robotics is seen in the example where the topological complexity of the motion planning problem for a robot arm in a configuration space without obstacles is computed. The idea presented in [81] is of importance, because it is able to present a mathematical measure of what surfaces are viable for motion planning. Such a measure is useful if and when there exists a choice of path planning surfaces.

Alternatively, one could approach the problem of designing paths around obstacles by both extracting and imposing topological information on the path planning algorithm. Authors of [82] show that the topological complexity, as previously seen in [81], can be extended to smooth compact Riemannian manifolds (manifolds equipped with a smooth metric). This accounts for motion planners that aim to construct paths with the lowest possible lengths. It also shows that the topological complexity of motion planners on supposedly more restrictive Riemannian manifolds only differs by a factor of 1. This is important to note since Riemannian manifolds admit smooth paths and their metric allows definitions of angles at intersections, curvature, area and volume, etc., which are useful in path planning applications. The PPS, as considered in [82], can then be considered as a space of continuous paths or motions that the robotic system can take.

4.1.2 Optimization Based Methods

In this section, metric/s are optimized on the analytical approximation of the PPS to generate a path. Trinkle and Milgram of [76] tackle path planning on the singularity-free workspace for a planar closed chain spherical manipulator. They determine $\mathcal{W}$, $\mathcal{C}$, and the singular sets in either space analytically, by approximating the Closed Chain Manipulator (CCM) as an equivalent Open Chain Manipulator (OCM). The OCM's fixed joint EE is equivalent to the CCM's first and last fixed joints.

The following steps are taken to determine $\mathcal{W}$, $\mathcal{C}$, and the singular sets. First, it can be seen that the $\mathcal{C}$ of the OCM is a product of the Special Orthogonal Group (SO) and the EE of this open chain is free to map anywhere inside the annular workspace $\mathcal{W}$. Second, they determine the complete $\mathcal{C}$ corresponding to the forward kinematics' image space $\mathrm{Im}\{\mathbf{f}^{\mathrm{fwd}}\} = V$. This image V takes the form of either a closed ball or an annulus, and lies in $\mathcal{W}$. Subsequently, the inverse location of the removed joint e is obtained using $\mathbf{f}^{\mathrm{inv}}(e)$, as a special case. In this case, the singular points of $\mathbf{f}^{\mathrm{fwd}}$ are disjoint spheres in $\mathcal{C}$. This completes defining the singular sets of the CCM and the singularity-free space of the CCM in $\mathcal{C}$, all of which are some variants of SO, thus resulting in manifolds.

Next, the image of the singular set decomposes $V \in \mathcal{W}$ into connected regions, which are open annuli centered at the base of the open chain. Since the singular set in $\mathcal{C}$ consists of disjoint spheres (i.e., compact), the map $\mathbf{f}^{\mathrm{fwd}}$ restricted to the inverse image of each open region in the annuli in $\mathcal{W}$, is a closed compact manifold (sphere) in $\mathcal{C}$. Consequently, the inverse images for any two points in one of these regions are at least locally diffeomorphic. Now, the problem reduces to two basic steps: first, find the inverse image in one of these regions; and, second, understand how the image changes when passed through the image of a singular point.

Standard methods of differential topology and algebraic geometry aid in determining how the global structure of the space $\mathbf{f}^{\mathrm{inv}}$ changes as one passes through the image of the singular points. The structure and the number of connected components in the inverse image provides the solution to the path existence problem. That is, a path from $\mathbf{q}^{\text{start}}$ to $\mathbf{q}^{\text{goal}}$ exists only if $\mathbf{q}^{\text{start}}$ and $\mathbf{q}^{\text{goal}}$ are in the same path component in $\mathbf{f}^{\text{inv}}$. Further information about the global structure of $\mathbf{f}^{\text{inv}}$ gives insight on how to construct efficient paths consisting of long geodesic segments. They use accordion moves to generate such paths. Accordion moves are so termed, since they optimize the number and selection of joints that remain fixed and actuated, based on the analytical approximation of the PPS. It is this optimization that achieves path planning. The authors of [76] extend their work in [77] for star shaped planar manipulators. They establish global connectivity by using cell decomposition of the analytically derived workspace and configuration space. They then device a complete polynomial algorithm for motion planning.

4.1.3 Force Fields

APF are powerful concepts as attractors toward the goal configurations, however cannot be directly adapted to curved surfaces, because of the fundamental difference in the nature of curved and planar surfaces. As an adaptation of the APF, Koditschek and Rimon of [45] pioneered path planning by formulating an equivalent global attractor function on manifolds. Their work was based on analytic manifolds with a boundary and the attractor functions were referred to as NFs. NFs are extensions of APF in point sets to manifolds, specifically in a sphere world. A sphere world is a compact connected subset whose boundary is formed by the disjoint union of a finite number of spheres. It follows that there is one large sphere which bounds the workspace, and smaller spheres which bound the obstacles. The free space $\mathcal{P}^{\text{free}}$ is obtained after removing all the smaller sphere-bounded obstacles from the bigger encompassing sphere-bounded space. Koditschek and Rimon show that using non-degenerate vector fields (isomorphic or structure preserving vector fields) that are transverse on the boundary of the free workspace, does not provide a globally attracting equilibrium state. This is because a smooth vector field on any sphere world which has a unique attractor, must have at least as many saddles as there are obstacles.

Thus, a globally attracting equilibrium state is topologically impossible [45]. The presence of such saddles or local minima is typical of what is experienced in APF in point sets. As a solution instead, Koditschek and Rimon of [45] impose further restrictions on the obstacle free workspace. They do so by defining it as a compact connected analytic manifold $\mathcal{P}^{\text{free}}$, that admits a NF. A proper navigation function, as defined by Koditschek and Rimon in [45], is one that satisfies being a composition of the three following functions:

- The first function is polar, almost everywhere Morse and analytic. That is, it has a unique and global minimum at the goal, must have directions where the function increases in value and can be locally approximated by a convergent power series, respectively.

- The second function is a diffeomorphism. That is, it is continuous and differentiable (smooth) on the path to the goal in the connected set, and uniformly maximal on the boundaries.
- The third function addresses the target point.

The NF therefore is posed to look ideal in that it is of maximum value at the boundary of the obstacle, directed away from the boundary and towards the workspace with a minimum at the goal. The first function enables a natural sinkhole towards the unique minimum, which is towards the goal. The second function is a diffeomorphism to limit the output range, resulting in a polar, admissible, and analytic function which is non-degenerate everywhere on $\mathcal{P}^{\text{free}}$ except the target point in $\mathcal{P}^{\text{free}}$. The third function caters towards the desired target point. The authors point out that analytic NF are harder to construct, but once defined, yield a provably correct control algorithm. Unquestionably, real world scenes do not often admit even a smooth, much less an analytic representation, and extending the approach beyond the class of a ball of obstacles may require relaxation of some constraints [45]. Tanner and Kumar of [78] also use NF in the context of multi-agent robotics to avoid local minima.

4.2 Geometric approximation of Path Planning Space

As opposed to analytical approximations of the spaces associated with the PPS, some algorithms use geometric methods to perform the task. Such techniques usually characterize a subset of the constraint-free space. Naturally, it does not provide a complete characterization, but it attempts to simplify the process. Path planning methods include force fields in Section 4.2.1, heuristic methods in Section 4.2.2 and optimization methods in Section 4.2.3.

4.2.1 Force Fields

Similar to potential or energy based OA, [85] performs path planning based on topology and thermal conduction. It identifies the source and target configurations of the robot as the heat source and sink of a conducting plate, respectively. Then, the PPP is formulated as a topology optimization problem that minimizes thermal compliance. Obstacles are modelled as regions of zero-conductivity or varying non-uniform levels depending on the application. Analogous to an APF, this technique uses artificial mass. The formulation of the problem makes obstacle regions less conducive to move towards and may be viewed as an indirect way of extracting the manifold (without any formal characterization) of the obstacle-free space. Yet another attempt to capture the constraint-free space is seen in [67]. Here, the concept of reachable volumes is derived by computing the union of points that result from the reachable confines of the joints' extremities.

4.2.2 Heuristics

Capturing global connectedness is a complex task. In an attempt to simplify the solution, Roy of [83], formulates a subset of the configuration manifold by breaking the C of a high Degrees Of Freedom (DOF) robot into smaller parts. Every joint's C is treated as a plane, and so are obstacles. The intersection of the joint space planes and obstacles result in sets (points, lines, circles, etc.), for which a C map is computed. The final set of the collision areas are the union of all those sets of computed C maps. A visibility map-based heuristic algorithm is used to generate near-optimal safe path in the formulated subset of the manifold.

4.2.3 Graphical Optimization

Authors of [84] explore the possibility of capturing topological information in an environment where the robot can sense obstacles as point clouds. The authors propose a method to define convex corridors that specify free space and provide linear constraints required to find the optimal trajectory for polygonal robots in cluttered environments. Obstacles sensed are represented by point clouds, and by using a point-wise Minkowski sum, the occupied regions are defined for every orientation of the robot. An initial seed path is generated using a graph/sampling based planner. Subsequently, the algorithm uses obstacle points to define hyperplanes, which in turn define polytopes that enclose the chords of the seed path. Since the polytopes that enclose the chords of the path can overlap, the inherent overlap produces a chain of overlapping polytopes, and hence convex regions.

These convex regions now form an approximated representation of the obstacle free space. The initial seed path can then be optimized for velocity using quadratic programming, resulting in smooth, short trajectories, or any higher desired derivative the selection of which is driven by robot dynamics, kinematics, or power constraints. This approach lends itself to the advantage of constructing adaptable polytopes to maximize the free workspace representation. Additionally, it can be scaled to several obstacles; however it is not without its disadvantages. Without prior knowledge of the arrangement of obstacles and the seed path, choosing obstacle points for polytope creation could be challenging. A naive approach is to rank the obstacle points for each chord. This can become computationally intensive depending on the number of obstacles points [84].

4.3 Numerical Approximation of Path Planning Space

The PPS or a subset of it may be represented using numerical approximations. Such representations may be the result of geometric, analytical and point-set approximations, which are not sufficient enough to represent the environment. The numerical combination may help in imposing structure. The path planning methods that operate on such representations are discussed in this section. Section 4.3.1, Section 4.3.2 and Section 4.3.3 discuss methods using force fields, optimization and numerical algorithms for path planning.

4.3.1 Force Fields

As the topological nature of PPS became more apparent, efforts to combine the benefits of topology while not being mathematically intensive began to appear. Take for example [11], where the authors recognized the need to account for environmental topology such as terrain curvature. However, they leverage tools such as Generalized Voronoi Diagram (GVD), Visbility Graph (VG) and APF that are typical of discrete data sets representing the environment. The study in [11] is an important example that shows the gradual transition from planning paths in point set representative environments, to requiring the imposition of structure to define the environment, albeit using small steps. Masehian and Amin-Naseri of [11] demonstrate the effectiveness of combining GVD, VG, APF and maximal inscribed discs to produce an obstacle-free configuration space. The assumptions include a point robot in a two-dimensional space, where the environment with static obstacles is known apriori.

4.3.2 Optimization

Olmstead and Yang of [86] take an alternative view to using geodesics for robotic path planning. Unlike the approach that uses manifolds to model the environment as seen in [11, 45, 76, 80, 81], the authors of [86] model the obstacle as a manifold instead. Typically, the shortest path from the source to the target configurations in a planar environment is a straight line connecting the two configurations. However, with the presence of an obstacle, the straight line path may not be feasible anymore. In an effort to minimize the deviation of the original path, the segment of the path around the obstacle must be such that it is, at least locally, the shortest path. In other words, it must be a geodesic.

To compute a geodesic around the obstacle, the obstacle must be a surface that is complete, convex and edge-free. Since obstacles need not comply with such requirements on their own, authors of [86] model an ellipsoid around the obstacle (similar to elliptical potential functions used to model repulsive APF around obstacles by [39]). An illustration of such a path around a spherical manifold enveloping an obstacle is seen in Fig. 6. Subsequently, geodesics on the surface of the ellipsoid model of the obstacle can be computed mathematically. The authors compute the discrete geodesic around the obstacle and append it to the remaining points that make up the path. Mathematical computation is involved since geodesics require some constants (Christoffel Symbols) to be computed.

In [87], the authors investigate path planning algorithms that are based on level set methods, which reflect the underlying mathematical structure of the space. These are for applications in which the environment is static, but where an apriori map is inaccurate and the environment is sensed in real-time. As the authors mention, the principal contribution is not a new path planning algorithm, but rather a formal analysis of path planning algorithms based on level set methods [87].

Approaches of this kind, i.e., formal and mathematically complete analysis as seen in [87, 96], even in seemingly simple environments is what will aim to provide a representation as close to the actual environment as possible. Computational costs when planning paths with level set methods are associated with the creation of the level set function. Once the level set function is computed, the optimal path is obtained by performing gradient descent down the level set function. The approach in [87] rests on the formal analysis of how the value of the level set function varies, when the changes in the environment are detected. The authors show that in many practical cases, only a small domain of the level set function needs to be re-computed when the environment changes [87].

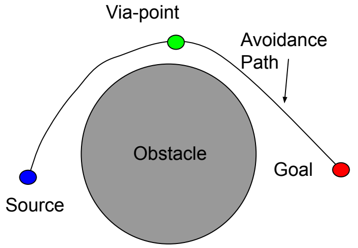

Fig. 6: Path around spherical manifold based obstacle

4.3.3 Numerical Path planning

As an extension to the work done in [76], Shvalb et al. [77] examine the global connectivity of the planar star-shaped manipulators, for efficient path planning. The authors determine the topological complexity of $\mathcal{C}$ similar to [76]. Through the analysis of the critical set of $\mathcal{C}$, the authors derive necessary and sufficient conditions for the global connectivity of $\mathcal{C}$, and those for path existence. Based on these results, a complete polynomial algorithm is devised for motion planning [77].

Although the focus of [88] is optimizing multi-robot coverage of a region, the approach that the authors develop is for a non-Euclidean space. Specifically, they adapt Lloyd's algorithm (discrete-time cost-functional minimizing) algorithm originally meant for Euclidean surfaces, to a typical PPS. That is, the PPS considered is a manifold punctured (absence of region) where obstacles would be located. They are able to demonstrate uniform coverage of the path planning manifold with a hole by multiple robots by reducing it to a discrete graph. The algorithm was shown to be effective in the case where multiple robots achieved uniform coverage on a two-sphere and in an indoor environment with walls and obstacles. Such work is essential in showing that mathematically intensive formulations can be approximated for computation purposes, while not compromising on the topological representation of the environment.

Complete reliance on geodesics for 3D robotic path planning is seen by Wu et al. [89]. They compute paths on a non-flat surface, such as a parameterized regular surface, using geodesics. The authors assume a simple parameterized surface for simulation purposes and compute discretized geodesics to show its efficiency for path planning purposes. Since the computed geodesic from the source to the target configuration has to be discretized for computational purposes, there exists a number of discretization intervals to choose from. To evaluate the choice of discrete geodesics with appropriate intervals, the criteria of minimized cost of geodesics is used. Geodesics locally minimize the arc length [89] between points that are sufficiently close. Therefore, a geodesic connecting source and target configurations that are not sufficiently close are comprised of geodesic segments between intermediate configurations. As the number of intermediate configurations tends to infinity, the discretized geodesic will approach the true geodesic between the start and target points. For a given source configuration therefore, the authors use potential fields to determine the next suitable robot configuration such that it complies with the geodesic equation and possesses the minimum cost, i.e least deviation from the true geodesic.

4.4 Cell/Graph Based Approximation of the Path Planning Space

Some algorithms portray a subset or all of their PPS (overall a manifold) as a cell/graph based representation. This section discusses the different path planning approaches in such environments. Note that some algorithms do not explicitly discuss manifolds, but utilize tools that can be used on manifolds. They are also discussed here. Section 4.4.1, Section 4.4.2, Section 4.4.3, Section 4.4.4 and Section 4.4.5 discuss path planning algorithms that utilize optimization, graph traversal, force fields, sampling and evolutionary based principles for path planning.

4.4.1 Heuristic/Optimization Methods

Similar to [11], the work in [12] also relies on grid-based GVD to find paths. Incrementally constructed GVD in the scope of robotics are used to extract the underlying metric skeletons of grids. The result is typically a graph structure with vertices and edges representing the connectivity of the space. However, the resulting structure contains no topological information about the environment, thus making it difficult to combine with path planners that need more heuristic information, such as terrain change and elevation about the PPS. Authors of [12] introduce a new algorithm called Dynamic Topology Detector (DTD) that combines GVD with discrete grid-based occupancy maps. The DTD then modifies the GVD stages to account for topological information. For instance, when a cell is marked as occupied instead of free, the authors introduce operations that allow for elevated or depressed scanning. That is, the occupancy grid along with the voronoi cells now store information that is specific to the elevation.

Along with retaining connectivity amongst nodes, DTD's topological information enable obstacle and terrain structures to be incrementally updated. Since DTD works incrementally, it is possible to use it in dynamic environments. One must note that works such as [11, 12], attempt to represent a subset of the PPS. Since it is not a comprehensive solution, the risks of incompletely representing the environment and possibly misinterpreting the true topological nature based on simply what was sensed are high. Therefore, the use of mathematical (topology and differential geometry) tools may provide a better comprehensive foundation towards a topologically representative solution of a path in its environment, as seen in much of the work in this section. This approach may be restrictive across parameters, such as the kinds of surfaces used, assumptions about apriori knowledge of the map, and computational restrictions. However, such restrictions are what allow a foundational analysis of the problem. Of course, there are several tools that can provide valid approximation procedures that transcend such restrictions such as charting of manifolds seen in Section 5.

Authors of [90] also model their path planning terrain as a manifold and then compute geodesics as paths. Since geodesics imbibe the curvature of the underlying space, they represent the shortest paths on the surface. The work in [90] differs from that in [89], with respect to the presence of obstacles. While the terrain assumes no obstacles in [89], it is not the case in [90]. Authors of [90] assume an initial calculated geodesic between the desired source and target configurations, for example, see Fig. 7. An encountered obstacle is characterized by a circle with a certain radius. This circle is the projection of the bounding volume (sphere) around the obstacle onto the manifold. The original geodesic is now evaluated for any intersection with the characterized obstacle. Intersections are handled as follows.

At the point of intersection, the algorithm computes the tangent plane (restricted to a certain local range to preserve the structure of the manifold) and picks a direction with magnitude on the tangent plane as the next movement. This chosen line, along with its end points are projected on the manifold to be connected as an intermediate segment to the original geodesic. However, should this new segment still intersect with the obstacle, the algorithm can pick yet another direction of motion from the tangent plane to proceed with. One may be concerned about the computational resource/time used to correctly find the direction of motion away from the obstacle, since there exists an infinite number of directions on the tangent plane to choose from. To assuage that, the authors of [90] use the Gauss-Bonnet theorem and the geodesic triangle to show that choosing either of the basis vectors of the tangent plane at the point of intersection is sufficient. This is because, one of the two tangent vectors is guaranteed to be collision-free; additionally, it is guaranteed to conform to the curvature of the manifold. The authors show such an example on an ellipsoid representation of their PPS.

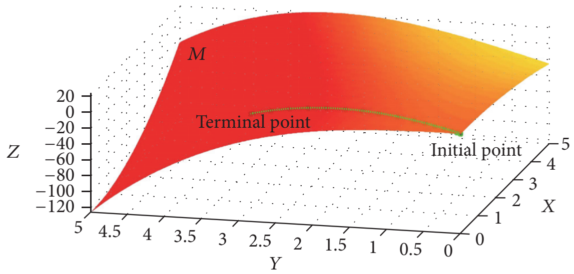

Fig. 7: An initial calculated geodesic between the source and target locations on a hypersurface [90].

4.4.2 Graph Search

Bhattacharya approaches the problem of path planning in non-Euclidean spaces differently in [91] and proves to be an interesting transition from manifolds as seen so far, towards gradually discretizing them. The work proposes the tasks of effectively building a non-Euclidean space and planning a path on it using the Rips complex of graph based representations. Using the Rips complex allows tremendous flexibility in defining connections between vertices and also across faces, i.e., connections are no longer simply between neighbouring vertices. The removal of such a restrictive connection and the imposition the Rips complex, allows maximal simplices to traverse the graph of the manifold. To build and estimate cost at the same time, the study in [91] proposes an algorithm called S*, which is a descendant of the Dijkstra algorithm. S* works like A*, however, the difference is that it operates on maximal simplices and constructs local embeddings of the identified maximal simplices to create a more geometric representation of the manifold. The work in [91] is a crucial marker of how discrete mathematical tools are essential to find a suitable approximation to the complex problem of representing the PPS accurately while still adhering to their fundamentally mathematical nature.

4.4.3 Force Fields

Methods such as those used in [92] revert to attempt creating an approximate, discrete topological map of the environment. They combine probabilistic path planning and APF. The authors of [92] compute probabilities of reliability to visible points and use the Floyd-Warshall algorithm to compute paths. The algorithm uses APF to model obstacles and correspondingly update the probabilities of the region and edges that constitute the path. Regions with obstacles have their vertices updated with a low reliability score. Simulations and real-world experiments show the generation of a reliable path. However, the environment is planar and capturing the topology of the environment is not clearly elaborated. The study in [11] shows another example where APF is used in conjunction with GVD. The terrain of the path planning space contains changes in elevation and curvature. Using obstacle features in a distributed grid form, a simple Voronoi diagram is constructed. This is eventually converted to a GVD, which is pruned for edges extending into dead-end regions. The pruned GVD provides an abstraction of the environment. Then, an obstacle-free space is formed in the pruned GVD by centering scaled, maximal inscribed discs that are centred on points of the pruned GVD. The resulting abstraction now reflects the topology of the environment. The path planning phase uses the visibility feature to track the target location and/or better paths towards the target. The use of APFs in this case is to mark already traversed edges or paths with higher potentials and to mark desirable visible paths to the goal with lower potential. So, this work accomplishes capturing the topology of the space with the combination of graphical methods and path planning with force field based methods.

4.4.4 Sampling Algorithms

Connectivity is a property of manifolds that is inherent to its definition. It is so, since if globally parameterized, every point on the manifold is related to each other by virtue of the global parametrization. Alternatively, if locally parameterized, every neighbourhood on the manifold must be able to transition to its neighbour using a local map. This connectivity allows the manifold to be well defined. Thus, for the purposes of path planning, it can be queried for a path multiple times with different source and target configurations. Multiple motion planning queries in a static environment have enjoyed the benefits of SAs. Since PRM has been successful in solving problems in configuration spaces of dimensions approaching 100, many researchers have worked to make the method more efficient.

For example, modifications have been made to solve more challenging types of problems, such as those with closed kinematic loops, non-holonomic constraints, dynamics, and intermittent contact motion plans, which are difficult to solve using traditional PRM [77]. Take for example, Fig. 8, which shows paths that can be deformed into one another (left hand side), and paths that cannot be deformed, due to the presence of an obstacle in between them (right hand side). These path characteristics are intrinsic to the nature of space where they lie. The nature of the space can be characterized by constraints on dynamics, kinematics and contact points. Traditional PRMs, do not account for such underlying attributes of the sampled points. In systems that model real-life applications however, $\mathcal{C}$ is often most naturally viewed as a lower-dimensional space embedded in an ambient space (typically the joint space), where embeddings are special maps between manifolds. The embedding results from equality constraints corresponding to kinematic loop closure or contact constraints. Thus, it is challenging to obtain an explicit description of $\mathcal{C}$ with minimal number of parameters and a suitable metric to guide sample generation. These problems make it difficult to construct a roadmap with the requisite properties, and hence difficult to solve motion planning problems for systems with kinematic loops using PRM (or any SA). Unless SAs account for the global structure of $\mathcal{C}$, they may fail to sample globally relevant regions [77].

Jaillet and Simeon capitalize exactly on that issue in [93] by developing a new variant of PRM that inherently encodes the connectedness of the configuration space. While not explicitly aimed at manifolds, this can be extended to manifolds as well. The encoding is inside small yet representative graphs which capture the different varieties of free paths well. The graphs are produced using the proposed Path Deformation Roadmap (PDR). PDRs depend on whether paths can be continuously deformed into another existing path, and use a deformation criteria that is simpler than permitted for homotopic paths. Informally defined, functions are homotopic if they can be continuously deformed into one another. It implies that the functions are equivalent topologically under the homotopy class. This is of use in constructing paths for robotic applications, since, should one path not be viable, a deformation may be applied to result in an alternative, equivalent path. A simple illustration is seen in Fig. 8.

The relaxed deformation in [93] allows the use of the Visibilty-PRM technique to construct road-maps that encode better suited information than representative paths of homotopy classes. As an added advantage, PDRs contain additional useful cycles between paths in the same homotopy class that otherwise would be hardly deformed into each other [93]. This approach is important in recognizing multiple aspects of the PPP such as global and connected nature of the space, characterizing paths into classes and planning for alternative paths within every class. Alternatives in the form of deformable paths are of utmost importance for contingent environment changes, such as the appearance of new static and dynamic obstacles. It is also important to know that while the PDR is still a variant of the PRM, it is capable of using the topological basis of the space, rather than assuming the relationship, as often seen in pure sampling based methods.

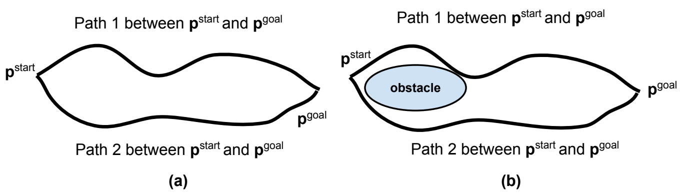

Fig. 8: (a) Homotopic or deformable paths (b) Paths are not homotopic due to the visibility of an obstacle.

4.4.5 Evolutionary Algorithms

Using an entirely different perspective, authors of [94] also attempt to create a representation of the path planning manifold by learning the underlying structure of the manifold from existing trajectories. The authors argue that their approach is model-free. That is, an explicit and complete analytical specification of the task which is difficult to obtain for some realistic robots, need not be assumed. Instead, they focus on maneuver sets, which are sets containing the solutions/trajectories to optimal PPPs by accounting for constraints and variations in trajectories/skills. As noted before in this section, these sets typically admit a description in the form of manifolds embedded in the configuration space of the system being controlled. The focus of [94] is on learning such solution sets from data in the form of a manifold, generalizing from a sample set of trajectories that share the underlying qualitative structure, and utilizing this manifold as the basis for subsequent motion synthesis, by a process of constrained geodesic trajectory-generation [94].

The skill manifold is learned in an offline phase directly from demonstrated data. Such data can be either the result of a computationally expensive optimization procedure (which is feasible offline) or examples of trajectories as demonstrated by a human being. The learnt skill manifold can then be used by the robot online and autonomously, accommodating novel constraints and goals. The disadvantages of combining learning techniques with complex, higher-dimensional problems are as follows. First, providing an appropriate number of and appropriately admissible solution sets/trajectories that are amply representative of the true manifold is seldom easy. This may result in the degradation of the algorithm's ability to correctly learn the characteristics that define the manifold, and additionally risk never being able to converge to a valid solution. Second, the learning algorithm runs into risks of learning the underlying structure incompletely, as learning is proportional to the training set provided and the amount of training permitted. Finally, such learning based approaches are not robust to parameter changes; for example, changes in robot dimension, placement of the obstacle, definition of the environment, and even the definition of the task. With even one such change, training must commence from the start.

4.5 Topological Tools on the Path Planning Space

Topological tools are useful in determining if paths can be collectively defined by virtue of sharing some equivalence relationship. So, when one path is deemed unfit, another equivalent path may be chosen. While not explicitly aimed at manifolds, Bhattacharya et al. [95] answer exactly this question and take it further by designing classes of paths around obstacles using concepts of homotopy and homology. Note that such an effort to produce homotopic paths was undertaken earlier in [93] as illustrated in Fig. 8.

Bhattacharya et al. [95] show that their proposed representation allows them 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 two classes of paths used are homotopy and homology, respectively. Informally defined as homotopy earlier, homotopic trajectories can be described in the context of path planning as follows. Two trajectories τ_1 and τ_2 connecting the same start and end configurations, are homotopic only if one can be continuously deformed into the other without intersecting any obstacle. As an alternative equivalence class in homology, the same pair of trajectories above (but this time τ_2 possessing an opposite orientation) forms the complete boundary of a 2-dimensional manifold embedded in $\mathcal{C}$ not containing/intersecting any of the obstacles. This definition stems from the fact that homology informally defines the empty space or holes (genus) that characterizes the manifold. Paths in the homology and homotopy classes are seen in Fig. 9.

The authors in [95] use this strong mathematical feature to stay clear of obstacles. They note that in a typical path planning environment that has been discretized into a graph, the common cost function is that of the distance assigned to a graph node/vertex. Such a function is a differential one-form (mapping from several dimensions to a single dimension/real value) but only accounts for distance and does not capture topological information. The authors therefore use properties of differential one-forms such as its integral, to design a homology class that is invariant for 2-dimensional $\mathcal{C}$. They use representative points to mark the interior of obstacles and thus regions around obstacles form distinct homotopy classes (boundaries of obstacles would form homology classes). Subsequently, an obstacle marker function is used, that allows trajectories to stop deforming once they reach the obstacle's boundaries. Using this imposed structure on an obstacle filled environment, they successfully generate obstacle avoiding trajectories, and are able to compute alternative paths within the classes. Successful examples include up to 20 different homotopy classes are seen in the 2D case.

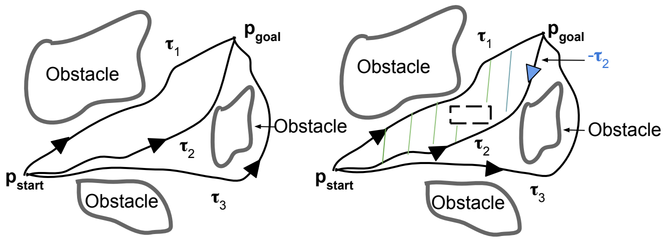

Fig. 9: Left: Homotopic paths τ_1 and τ_2, but not τ_3, with either of the other paths. Right: Homologous paths τ_1 and $-\tau_2$ enclose shaded area; τ_3 is not homologous with τ_1 or τ_2.

5 Charting of Manifolds

This section provides a brief overview of the alternative approach to formally parametrizing the manifold. This approach may involve any or a combination of the following. One may define the surface using a system of implicitly defined equations, and create parametrizations approximating the surface locally, called charts. Some authors construct all charts that cover the surface (called the atlas of a manifold), following which suitable charts (points located there) are picked by the path planner. Alternatively, there may exist situations where the bounds of the parameters defining the surface may not be defined, i.e., resulting surface is without a boundary. In such a situation, one may want to construct charts selectively in the direction of the target point to save time and computational resources. Several variants of chart construction methods for manifolds or their discretization exist, although mostly in the form of mathematical tools that can later be applied in the context of robotics. Some others visualize manifolds as triangulated surfaces either from point clouds representing the surface or from closed-form parametrizations. The following therefore focuses on how mathematical tools and path planning/graph traversal algorithms can be used to represent and compute paths on parts of/the entire manifold describing the PPS. Section 5.1 and Section 5.2 focus on complete charting and goal biased charting of manifolds respectively. One may refer to Table 2 for a tabulated version of the works in this section.

Table 2: Overview of manifold based methods—Charting of manifolds

Charting method	General mathematical tools	Robotics path planning oriented tools
Unbiased charting	[97–99]	[100–105]
Goal-biased charting	-	[66, 102–105]

5.1 Unbiased Chart Construction

Sethian first pioneered the Fast Marching Method in 1995, which was later published in [99] as a tool to compute the position of propagating fronts. That is, topological changes, corner and cusp development, and accurate determination of geometric properties, such as curvature and normal direction are naturally obtained in this setting. Sethian and Kimmel [98] formally introduce the Fast marching method to propagate level set functions and build a surface. As an application, they provide an optimal time algorithm for computing the geodesic distances and thereby extracting the shortest paths on triangulated manifolds. Mathematical tools, such as Fast Marching methods [97–99], enable the robotics community to choose from a wide variety of methods as solutions to visualize complex surfaces and subsequently compute a geodesic distance. The aforementioned work predominantly solved mathematical problems, which made their way into computer vision and with time, slowly forayed into more robotics-related applications.

It is interesting to note that such techniques were precursors and alternatives to the formal notions of defining metrics (like the Rips Complex) on a Riemannian manifold, as seen in [91]. The onus of constructing a minimal number of charts to represent the manifold is always one of importance [100]. Equivalently in the discrete world, several authors explore constructing charts in a discrete fashion, that enable to pave the manifold. For example, Henderson [101] presents a new continuation method for computing implicitly-defined manifolds. The manifold is represented as a set of overlapping neighborhoods, and extended by an added neighborhood of a boundary point. The boundary point is found using a boundary expression in terms of the vertices of a set of finite convex polyhedra. The resulting algorithm allows adaptive spacing of the computed points, and deals with the problems of local and global overlap in a natural way; see Fig. 10. The algorithm pioneered the technique to chart a manifold incrementally, based on implicit equations defining a surface, and has spawned several extensions [102–105]. Note that one could adapt PDRs

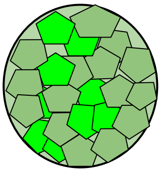

Fig. 10: Several charts (green polygons) covering/charting the spherical manifold; the union of charts forms an atlas

in [93] to suit higher dimensional problems. This was originally described in Section 4.4.4. It trades off resolution completeness with efficiency and probabilistic completeness. Of course, its resolution impacts how well it captures the connectivity of the space.

The work in [100], which is similar to that of [76], still adheres to restrictions of manifolds without immediately discretizing them for computational ease. Instead, it uses some fundamental properties specific to planar serial manipulators to minimize chart construction, where the union of charts represents the manifold. Usually, several charts are required to accurately map the manifold completely, however the authors of [100] prove that only two coordinate charts to parameterize the configuration space are required. Assuming that the planar serial manipulator has m joints and $(m-1)$ links, each such chart is embedded in an $(m-3)$ dimensional torus (since every joint is an SO group, as seen in [76]).

The resulting formulated $\mathcal{C}$ allows obstacles to be defined by their convex hulls and their boundaries be defined as varieties (solutions to set of polynomials). This forms two structural sets: one using the boundary varieties to define the obstacles, and the other defining charts to represent the manifold. They then sample across the boundary varieties and charts to obtain a path planning solution. Compared to traditional algorithms, this technique not only avoids massive time-consuming projections from the ambient space to the path planning space, but also is able to solve problems for which the PPS contains bifurcations and narrow passages [100]. This work is an amalgamation of recognizing the PPS to be a surface of some known parameters (embedding into a specific kind of torus) and then, utilizing charts to numerically define the manifold. So, it capitalizes on both Section 4 and Section 5.

5.2 Goal-Biased Chart Construction

Bohigas et al. [102–104] use the higher dimensional continuation method developed by Henderson in [101] to build a parametrization of $\mathcal{C}$. The approach in the aforementioned works extract from the constraint-filled $\mathcal{C}$, a system of equations that formulate $\mathcal{C}^{\text{free}}$ as an implicitly defined manifold. Subsequently, higher-dimensional continuation techniques are used to progressively construct an atlas of the manifold that contains the start configuration. The construction stops either when the goal configuration is reached, or when path non-existence is proven at the chosen resolution of the atlas.

An example of a charted, implicitly defined manifold of $\mathcal{C}^{\text{free}}$ as constructed in [102] is seen in Fig. 11. In this case, the path planner used on the atlas is A*, which uses the centres of charts as nodes for path planning. Chart creation, as adopted from [101], is customized so that only those charts that comply with both A* and the manifold curvature are admitted. The approaches described thus far and throughout this section, use incremental constructions of manifolds. They are vastly different from the methods seen in Section 4, since no explicit and global parametrization of the path planning manifold is required. Complex configuration spaces are seldom easily explicitly characterized, and more so in the presence of constraints, such as singularities and obstacles.

Charting via implicitly defined equations avoids the need of representing the singularity locus explicitly as an obstacle, i.e., no explicit representation of the constraint region is required. Constraints are also represented implicitly as the region that does not comply with the system of equations defining the original desired manifold. The solution manifold of this system can then be freely navigated without fear of encountering any constraints, which in [102, 103] for instance, were manipulator singularities. With the ability to both progressively and exhaustively generate the atlas from a reachable configuration, the method lends itself to multi-query planning and visualization. The resolution completeness of the approach is at the expense of a computational cost that scales exponentially with the dimension of $\mathcal{C}$. The work in [103] also uses goal-

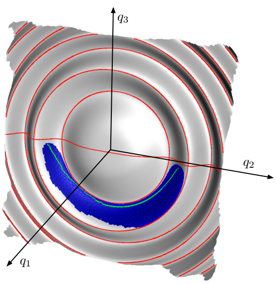

Fig. 11: Charting an implicitly defined manifold of $\mathcal{C}^{\text{free}}$ [102]

biased continuation methods by progressively constructing an atlas of the PPS.

Such continuation methods do not require explicit parametrizations of the constraints. Another advantage is the almost discrete nature of the intermediate results, i.e., that of the charts. The discrete nature, which is the individualized representation for each chart or patch, allows them to be viewed as nodes in a graph. Subsequently, any graph traversal algorithm can be used to pick directions to expand the chart generation or to plan paths. As seen in [102–104], A* was the planner of choice. But with such flexibility, improvements in the form of other graph traversal/tree growing methods can be seen in conjunction with higher dimensional continuation. For example, Jaillet and Porta of [105] combine RRTs with the higher dimensional continuation method to propose a new method called AtlasRRT. The AtlasRRT uses an atlas to efficiently explore a configuration space manifold implicitly defined by kinematic constraints. Similarly, the approach in[66] also uses an optimized goal-biased sampling strategy which, additionally, guarantees asymptotic optimality [103].

To combat the expensive process of defining the full atlas for a given manifold, the AtlasRRT algorithm intertwines the construction of the atlas and the RRT. The partially constructed atlas is used to sample new configurations and to generate new branches for the RRT, with the latter used to determine directions of expansion for the atlas. This approach retains the exploration bias typical of RRT approaches, i.e., the tree is strongly pushed towards yet unexplored regions of the configuration space manifold. Jaillet and Porta of [105] suggest that experimentally, the AtlasRRT is generally more efficient for highly constrained spaces. As opposed to highly constrained regions, mildly constrained spaces imply that the embedded PPS is very similar to the ambient space.

6 Conclusion

Path planning is an integral component of autonomous robotics. While the decision on an appropriate path planner is application dependent, classifications of existing literature can aid the choice. In this work, we accomplish the following: First, the common classifications of path planning approaches are provided. The common categories of algorithms are based on operation in known/unknown environments, prioritizing between generating robot-specific paths and generic paths better suited for collision avoidance, and finally, handling static and dynamic constraints. This paper specifically analyzes the categorization of path planning algorithms under static constraints. It was seen that such a classification is not comprehensive from the grass roots level. Thus, to be more comprehensive, we propose a taxonomy of path planning algorithms based on the mathematical structure of the PPS. We show that a PPS can be viewed as a set of points or as manifolds. The former can further be divided into un-partitioned and partitioned spaces, of which the former uses variants of SAs, OAs, MPCs, and EAs, while the latter uses CDGT and sampling based optimization techniques. The main contribution of this work is the review of works for static constraints that fall under the proposed taxonomy, i.e., specifically under topology and manifold based methods. This comprehensive literature review pertaining to manifold based robotic path planning algorithms sees a bifurcation when the PPSs can either be explicitly parameterized or incrementally constructed for path planning purposes. This article achieves the following two goals: First, it introduces a systematic taxonomy that is now inclusive of all schools of approaches in path planning. Second, it streamlines the migration of path planning work from disciplines such as mathematics and computer vision to robotics, into one comprehensive reference document.

Statements and Declarations

- Funding: This work was partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC). Grant RGPIN-2014-06512.
- Competing interests: The authors have no competing interests to declare that are relevant to the content of this article.
- Ethics approval: Not applicable
- Consent to participate: Not applicable
- Consent for publication: Not applicable
- Availability of data and materials: There is no data or materials to share.
- Code availability: Not applicable
- Authors' contributions: All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Sindhu Radhakrishnan. The first draft of the manuscript was written by Sindhu Radhakrishnan and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

References

[1] Molina, C.P., Ortego, R.G., Pérez, F.M.: Perspectives on Technological Developments Applied to Robotics, pp. 59–86. Springer, London (2014). https://doi.org/10.1007/978-1-4471-5358-0_5

[2] Radhakrishnan, S., Gueaieb, W.: A state-of-the-art review on topology and differential geometry-based robotic path planning—Part II: planning under dynamic constraints. International Journal of Intelligent Robotics and Applications (Springer) (2023). (Submitted: Nov. 2023)

[3] Lengyel, J., Reichert, M., Donald, B.R., Greenberg, D.P.: Real-time robot motion planning using rasterizing computer graphics hardware. Computer graphics (New York, N.Y.) **24**(4), 327–335 (1990)

[4] Stentz, A.: Optimal and efficient path planning for partially-known environments. In: Proceedings of the 1994 IEEE International Conference

on Robotics and Automation, pp. 3310–33174 (1994). https://doi.org/10.1109/ROBOT.1994.351061

[5] Radhakrishnan, S.: Observable 2D SLAM and Evidential Occupancy Grids. Master's thesis, Carleton University (2014)

[6] Kuffner, J.J., LaValle, S.M.: RRT-connect: An efficient approach to single-query path planning. In: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), vol. 2, pp. 995–10012 (2000). https://doi.org/10.1109/ROBOT.2000.844730

[7] Ferguson, D., Stentz, A.: Anytime RRTs. In: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5369–5375 (2006). https://doi.org/10.1109/IROS.2006.282100

[8] Lee, J., Pippin, C., Balch, T.: Cost based planning with RRT in outdoor environments. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 684–689 (2008). https://doi.org/10.1109/IROS.2008.4651052

[9] Diankov, R., Kuffner, J.: Randomized statistical path planning. In: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1–6 (2007). https://doi.org/10.1109/IROS.2007.4399557

[10] Jaillet, L., Cortés, J., Siméon, T.: Sampling-based path planning on configuration-space costmaps. IEEE Transactions on Robotics **26**(4), 635–646 (2010)

[11] Masehian, E., Amin-Naseri, M.R.: A voronoi diagram-visibility graph-potential field compound algorithm for robot path planning. Journal of Robotic Systems **21**(6), 275–300 (2004)

[12] Qin, L., Yin, Q., Zha, Y., Peng, Y.: Dynamic detection of topological information from grid-based generalized voronoi diagrams. Mathematical Problems in Engineering **2013**, 1–11 (2013)

[13] M.LaValle, S.: Rapidly-exploring random trees: A new tool for path planning. Technical report, Iowa State University, Ames, IA 50011 USA (June 1998)

[14] Kavraki, L.E., Svestka, P., Latombe, J.-C., Overmars, M.H.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Transactions on Robotics and Automation **12**(4), 566–580 (1996). https://doi.org/10.1109/70.508439

[15] Elbanhawi, M., Simic, M.: Sampling-based robot motion planning: A

review. IEEE Access **2**, 56–77 (2014). https://doi.org/10.1109/ACCESS.2014.2302442

[16] Karaman, S., Frazzoli, E.: Sampling-based algorithms for optimal motion planning. The International Journal of Robotics Research **30**(7), 846–894 (2011)

[17] Hsu, D., Zheng Sun: Adaptively combining multiple sampling strategies for probabilistic roadmap planning. In: IEEE Conference on Robotics, Automation and Mechatronics, 2004., vol. 2, pp. 774–7792 (2004). https://doi.org/10.1109/RAMECH.2004.1438016

[18] Bohlin, R., Kavraki, L.E.: Path planning using lazy PRM. In: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), vol. 1, pp. 521–5281 (2000). https://doi.org/10.1109/ROBOT.2000.844107

[19] Akbaripour, H., Akbaripour, H., Masehian, E., Masehian, E.: Semi-lazy probabilistic roadmap: a parameter-tuned, resilient and robust path planning method for manipulator robots. International Journal of Advanced Manufacturing Technology **89**(5), 1401–1430 (2017)

[20] LaValle, S.M., Branicky, M.S., Lindemann, S.R.: On the relationship between classical grid search and probabilistic roadmaps. The International Journal of Robotics Research **23**(7-8), 673–692 (2004)

[21] Dale, L.K., Amato, N.M.: Probabilistic roadmaps-putting it all together. In: Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164), vol. 2, pp. 1940–19472 (2001). https://doi.org/10.1109/ROBOT.2001.932892

[22] LaValle, S.M., Kuffner, J.J.: Randomized kinodynamic planning. The International Journal of Robotics Research **20**(5), 378–400 (2001)

[23] Atramentov, A., LaValle, S.M.: Efficient nearest neighbor searching for motion planning. In: Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292), vol. 1, pp. 632–6371 (2002). https://doi.org/10.1109/ROBOT.2002.1013429

[24] Lin, Y., Saripalli, S.: Sampling-based path planning for uav collision avoidance. IEEE Transactions on Intelligent Transportation Systems **18**(11), 3179–3192 (2017)

[25] Kuwata, Y., Teo, J., Karaman, S., Fiore, G., Frazzoli, E., How, J.: Motion planning in complex environments using closed-loop prediction. In: AIAA Guidance, Navigation and Control Conference and Exhibit

(2008). https://doi.org/10.2514/6.2008-7166. AIAA

[26] Wang, W., Zuo, L., Xu, X.: A learning-based multi-rrt approach for robot path planning in narrow passages. Journal of Intelligent & Robotic Systems **90**(1-2), 81–100 (2017)

[27] Suh, J., Gong, J., Oh, S.: Fast sampling-based cost-aware path planning with nonmyopic extensions using cross entropy. IEEE Transactions on Robotics **33**(6), 1313–1326 (2017)

[28] Noreen, I., Khan, A., Ryu, H., Doh, N.L., Habib, Z.: Optimal path planning in cluttered environment using RRT-AB. Intelligent Service Robotics **11**(1), 41–52 (2017)

[29] Park, J.M., Kim, D.W., Yoon, Y.S., Kim, H.J., Yi, K.S.: Obstacle avoidance of autonomous vehicles based on model predictive control. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering **223**(12), 1499–1516 (2009)

[30] Gao, Y., Lin, T., Borrelli, F., Tseng, E., Hrovat, D.: Predictive Control of Autonomous Ground Vehicles With Obstacle Avoidance on Slippery Roads. Dynamic Systems and Control Conference, vol. ASME 2010 Dynamic Systems and Control Conference, Volume 1, pp. 265–272 (2010). https://doi.org/10.1115/DSCC2010-4263. ASME. https://doi.org/10.1115/DSCC2010-4263

[31] Yoon, Y., Shin, J., Kim, H.J., Park, Y., Sastry, S.: Model-predictive active steering and obstacle avoidance for autonomous ground vehicles. Control Engineering Practice **17**(7), 741–750 (2009)

[32] Rasekhipour, Y., Khajepour, A., Chen, S., Litkouhi, B.: A potential field-based model predictive path-planning controller for autonomous road vehicles. IEEE Transactions on Intelligent Transportation Systems **18**(5), 1255–1267 (2017). https://doi.org/10.1109/TITS.2016.2604240

[33] Rasekhipour, Y., Fadakar, I., Khajepour, A.: Autonomous driving motion planning with obstacles prioritization using lexicographic optimization. Control Engineering Practice **77**, 235–246 (2018)

[34] Sgorbissa, A.: Integrated robot planning, path following, and obstacle avoidance in two and three dimensions: Wheeled robots, underwater vehicles, and multicopters. The International journal of robotics research **38**(7), 853–876 (2019)

[35] Abbas, M.A., Milman, R., Eklund, J.M.: Obstacle avoidance in real time with nonlinear model predictive control of autonomous vehicles. In: 2014

IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 1–6 (2014). https://doi.org/10.1109/CCECE.2014.6901109

[36] Gao, Y., Gray, A., Tseng, H.E., Borrelli, F.: A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles. Vehicle System Dynamics **52**(6), 802–823 (2014) https://doi.org/10.1080/00423114.2014.902537. https://doi.org/10.1080/00423114.2014.902537

[37] Montiel, O., Orozco-Rosas, U., Sepúlveda, R.: Path planning for mobile robots using bacterial potential field for avoiding static and dynamic obstacles. Expert Systems with Applications **42**(12), 5177–5191 (2015)

[38] Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. The International Journal of Robotics Research **5**(1), 90–98 (1986)

[39] Volpe, R., Khosla, P.: Artificial potentials with elliptical isopotential contours for obstacle avoidance. In: 26th IEEE Conference on Decision and Control, vol. 26, pp. 180–185 (1987). https://doi.org/10.1109/CDC.1987.272738

[40] Faverjon, B., Tournassoud, P.: A local based approach for path planning of manipulators with a high number of degrees of freedom. In: Proceedings. 1987 IEEE International Conference on Robotics and Automation, vol. 4, pp. 1152–1159 (1987). https://doi.org/10.1109/ROBOT.1987.1087982

[41] Koditschek, D.: Exact robot navigation by means of potential functions: Some topological considerations. In: Proceedings. 1987 IEEE International Conference on Robotics and Automation, vol. 4, pp. 1–6 (1987). https://doi.org/10.1109/ROBOT.1987.1088038

[42] Khosla, P., Volpe, R.: Superquadric artificial potentials for obstacle avoidance and approach. In: Proceedings. 1988 IEEE International Conference on Robotics and Automation, pp. 1778–17843 (1988). https://doi.org/10.1109/ROBOT.1988.12323

[43] Volpe, R., Khosla, P.: Manipulator control with superquadric artificial potential functions: theory and experiments. IEEE Transactions on Systems, Man, and Cybernetics **20**(6), 1423–1436 (1990). https://doi.org/10.1109/21.61211

[44] Agirrebeitia, J., Avilés, R., de Bustos, I.F., Ajuria, G.: A new APF strategy for path planning in environments with obstacles. Mechanism and machine theory **40**(6), 645–658 (2005)

[45] Koditschek, D.E., Rimon, E.: Robot navigation functions on manifolds with boundary. Advances in Applied Mathematics **11**(4), 412–442 (1990)

[46] Quillen, P., Muñoz, J., Subbarao, K.: Path planning to a reachable state using minimum control effort based navigation functions. The Journal of the astronautical sciences **66**(4), 554–581 (2019)

[47] Kowalczyk, W., Kowalczyk, W., Przybyla, M., Przybyla, M., Kozlowski, K., Kozlowski, K.: Set-point control of mobile robot with obstacle detection and avoidance using navigation function - experimental verification. Journal of Intelligent & Robotic Systems **85**(3), 539–552 (2017)

[48] Kowalczyk, W.: Rapid navigation function control for two-wheeled mobile robots. Journal of Intelligent & Robotic Systems **93**(3-4), 687–697 (2018)

[49] Tao, S., Tan, J.: Path planning with obstacle avoidance based on normalized r -functions. Journal of Robotics **2018**, 1–10 (2018)

[50] Barraquand, J., Latombe, J.-C.: Robot motion planning: A distributed representation approach. The International Journal of Robotics Research **10**(6), 628–649 (2016)

[51] Dash, A.K., Chen, I.-M., Yeo, S.H., Yang, G.: Singularity-free path planning of parallel manipulators using clustering algorithm and line geometry. In: 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422), vol. 1, pp. 761–766 (2003)

[52] Carpin, S., Pillonetto, G.: Motion planning using adaptive random walks. IEEE Transactions on Robotics **21**(1), 129–136 (2005)

[53] Chu, K., Lee, M., Sunwoo, M.: Local path planning for off-road autonomous driving with avoidance of static obstacles. IEEE Transactions on Intelligent Transportation Systems **13**(4), 1599–1616 (2012)

[54] Kavraki, L.E., Kolountzakis, M.N., Latombe, J.-.: Analysis of probabilistic roadmaps for path planning. IEEE Transactions on Robotics and Automation **14**(1), 166–171 (1998). https://doi.org/10.1109/70.660866

[55] Donald, B., Lynch, K.K.M.., Rus, D. (eds.): Algorithmic and Computational Robotics : New Directions 2000 WAFR, 1st edn. A K Peters/CRC Press, an imprint of Taylor and Francis, Boca Raton, FL (2001)

[56] Berenson, D., Srinivasa, S., Kuffner, J.: Task space regions: A framework for pose-constrained manipulation planning. The International Journal of Robotics Research **30**(12), 1435–1460 (2011)

[57] Jaillet, L., Porta, J.: Asymptotically-optimal path planning on manifolds. Robotics Science and Systems VIII (2012)

[58] Quinlan, S.: Real-time modification of collision-free paths. ProQuest Dissertations Publishing (1995)

[59] Quinlan, S., Khatib, O.: Elastic bands: connecting path planning and control. In: [1993] Proceedings IEEE International Conference on Robotics and Automation, pp. 802–8072 (1993). https://doi.org/10.1109/ROBOT.1993.291936

[60] Ademovic, A., Lacevic, B.: Path planning for robotic manipulators using expanded bubbles of free c-space. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), vol. 2016-, pp. 77–82. IEEE, Stockholm, Sweden (2016). https://doi.org/10.1109/ICRA.2016.7487118

[61] Qureshi, A.H., Iqbal, K.F., Qamar, S.M., Islam, F., Ayaz, Y., Muhammad, N.: Potential guided directional-RRT* for accelerated motion planning in cluttered environments. In: 2013 IEEE International Conference on Mechatronics and Automation, pp. 519–524 (2013). https://doi.org/10.1109/ICMA.2013.6617971

[62] Qureshi, A.H., Mumtaz, S., Iqbal, K.F., Ali, B., Ayaz, Y., Ahmed, F., Muhammad, M.S., Hasan, O., Kim, W.Y., Ra, M.: Adaptive potential guided directional-RRT. In: 2013 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1887–1892 (2013). https://doi.org/10.1109/ROBIO.2013.6739744

[63] Qureshi, A.H., Qureshi, A.H., Ayaz, Y., Ayaz, Y.: Potential functions based sampling heuristic for optimal path planning. Autonomous Robots **40**(6), 1079–1093 (2016)

[64] Qureshi, A.H., Ayaz, Y.: Intelligent bidirectional rapidly-exploring random trees for optimal motion planning in complex cluttered environments. Robotics and Autonomous Systems **68**, 1–11 (2015)

[65] Wei, K., Ren, B.: A method on dynamic path planning for robotic manipulator autonomous obstacle avoidance based on an improved RRT algorithm. Sensors (Basel, Switzerland) **18**(2), 571 (2018)

[66] Kang, G., Kim, Y.B., Lee, Y.H., Oh, H.S., You, W.S., Choi, H.R.: Sampling-based motion planning of manipulator with goal-oriented sampling. Intelligent Service Robotics **12**(3), 265–273 (2019)

[67] McMahon, T., Thomas, S., Amato, N.M.: Sampling-based motion planning with reachable volumes for high-degree-of-freedom manipulators.

The International Journal of Robotics Research **37**(7), 779–817 (2018)

[68] Siciliano, B.: Robotics Modelling, Planning and Control, 1st ed. 2009. edn. Advanced Textbooks in Control and Signal Processing. Springer, London (2009). https://doi.org/10.1007/978-1-84628-642-1

[69] Oriolo, G.: Motion Planning 3 Artifical Potential Fields. Professor Oriolo's notes for his class in Artificial Intelligence and Robotics. (2020). http://diag.uniroma1.it/oriolo/amr/slides/MotionPlanning3_Slides.pdf

[70] Leica, P., Chavez, D., Rosales, A., Roberti, F., Toibero, J.M., Carelli, R.: Strategy based on multiple objectives and null space for the formation of mobile robots and dynamic obstacle avoidance. Revista Politécnica (Quito) **33**(1) (2014)

[71] Aghababa, M.P.: 3D Path planning for underwater vehicles using five evolutionary optimization algorithms avoiding static and energetic obstacles. Applied Ocean Research **38**, 48–62 (2012)

[72] Wang, C., Mao, Y.S., Du, K.J.: Simulation on local obstacle avoidance algorithm for unmanned surface vehicle. International Journal of Simulation Modelling **15**(3), 460–472 (2016)

[73] Kim, H., Cheang, U.K., Rogowski, L.W., Kim, M.J.: Motion planning of particle based microrobots for static obstacle avoidance. Journal of Micro-Bio Robotics **14**(1-2), 41–49 (2018)

[74] Wang, D., Wang, P., Zhang, X., Guo, X., Shu, Y., Tian, X.: An obstacle avoidance strategy for the wave glider based on the improved artificial potential field and collision prediction model. Ocean Engineering **206**, 107356 (2020)

[75] Campana, M., Lamiraux, F., Laumond, J.-P.: A gradient-based path optimization method for motion planning. Advanced Robotics **30**(17-18), 1126–1144 (2016)

[76] Trinkle, J.C., Milgram, R.J.: Complete path planning for closed kinematic chains with spherical joints. The International Journal of Robotics Research **21**(9), 773–789 (2002)

[77] Shvalb, N., Shoham, M., Liu, G., Trinkle, J.C.: Motion planning for a class of planar closed-chain manipulators. The International Journal of Robotics Research **26**(5), 457–473 (2007)

[78] Tanner, H.G., Kumar, A.: Towards decentralization of multi-robot navigation functions. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, pp. 4132–4137. IEEE, Barcelona, Spain (2005)

[79] Brooks, R.A.: Planning collision- free motions for pick-and-place operations. The International Journal of Robotics Research **2**(4), 19–44 (1983)

[80] Dorst, L., Mandhyan, I., Trovato, K.: The geometrical representation of path planning problems. Robotics and Autonomous Systems **7**(2), 181–195 (1991)

[81] Farber, M.: Topological complexity of motion planning. Discrete & Computational Geometry **29**(2), 211–221 (2003)

[82] Błaszczyk, Z., Carrasquel-Vera, J.G.: Topological complexity and efficiency of motion planning algorithms. Revista Matemática Iberoamericana **34**(4), 1679–1684 (2018)

[83] Roy, D.: Algorithmic path planning of static robots in three dimensions using configuration space metrics. Robotica **29**(2), 295–315 (2011)

[84] Kennedy, M., Thakur, D., Ani Hsieh, M., Bhattacharya, S., Kumar, V.: Optimal paths for polygonal robots in SE(2). Journal of Mechanisms and Robotics **10**(2) (2018)

[85] Ryu, J.C., Ryu, J.C., Park, F.C., Park, F.C., Kim, Y.Y., Kim, Y.Y.: Mobile robot path planning algorithm by equivalent conduction heat flow topology optimization. Structural and multidisciplinary optimization **45**(5), 703–715 (2012)

[86] Olmstead Muhs, J.C., Yang, J.: A geodesics-based model for obstacle avoidance. In: 2005 Digital Human Modeling for Design and Engineering Symposium. SAE International, Iowa City, IA, USA (2005). https://doi.org/10.4271/2005-01-2692. https://doi.org/10.4271/2005-01-2692

[87] Xu, B., Xu, B., Stilwell, D.J., Stilwell, D.J., Kurdila, A.J., Kurdila, A.J.: Fast path re-planning based on fast marching and level sets. Journal of Intelligent & Robotic Systems **71**(3), 303–317 (2013)

[88] Bhattacharya, S., Ghrist, R., Kumar, V.: Multi-robot coverage and exploration on riemannian manifolds with boundaries. The International Journal of Robotics Research **33**(1), 113–137 (2014)

[89] Wu, K., Lo, C., Lin, Y., Liu, J.: 3D path planning based on nonlinear

geodesic equation. In: 11th IEEE International Conference on Control Automation (ICCA), pp. 342–347 (2014). https://doi.org/10.1109/ICCA.2014.6870943

[90] Wu, K.-L., Ho, T.-J., Huang, S.A., Lin, K.-H., Lin, Y.-C., Liu, J.-S.: Path planning and replanning for mobile robot navigation on 3D terrain: An approach based on geodesic. Mathematical Problems in Engineering **2016**, 1–12 (2016)

[91] Bhattacharya, S.: Towards optimal path computation in a simplicial complex. The International Journal of Robotics Research **38**(8), 981–1009 (2019)

[92] Zhang, B., Liu, Y., Lu, Q., Wang, J.: A path planning strategy for searching the most reliable path in uncertain environments. International Journal of Advanced Robotic Systems **13**(5), 172988141665775 (2016). https://doi.org/10.1177/1729881416657751

[93] Jaillet, L., Simeon, T.: Path deformation roadmaps: Compact graphs with useful cycles for motion planning. The International Journal of Robotics Research **27**(11-12), 1175–1188 (2008). https://doi.org/10.1177/0278364908098411

[94] Havoutis, I., Ramamoorthy, S.: Motion planning and reactive control on learnt skill manifolds. The International Journal of Robotics Research **32**(9-10), 1120–1150 (2013)

[95] Bhattacharya, S., Likhachev, M., Kumar, V.: Topological constraints in search-based robot path planning. Autonomous Robots **33**(3), 273–290 (2012)

[96] Bhattacharya, S., Pivtoraiko, M.: A classification of configuration spaces of planar robot arms for a continuous inverse kinematics problem. Acta applicandae mathematicae **139**(1), 133–166 (2015)

[97] Sethian, J.A.: Fast marching methods. SIAM review **41**(2), 199–235 (1999)

[98] Kimmel, R., Sethian, J.A.: Computing geodesic paths on manifolds. Proceedings of the National Academy of Sciences - PNAS **95**(15), 8431–8435 (1998)

[99] Sethian, J.A.: A fast marching level set method for monotonically advancing fronts. Proceedings of the National Academy of Sciences - PNAS **93**(4), 1591–1595 (1996)

[100] Liu, G., Trinkle, J., Yang, Y., Luo, S.: Motion planning of planar closed

chains based on structural sets. IEEE Access **8**, 117203–117217 (2020)

[101] Henderson, M.E.: Multiple parameter continuation: Computing implicitly defined k-manifolds. International Journal of Bifurcation and Chaos in Applied Sciences and Engineering **12**(3), 451–476 (2002)

[102] Bohigas, O., Henderson, M.E., Ros, L., Porta, J.M.: A singularity-free path planner for closed-chain manipulators. In: 2012 IEEE International Conference on Robotics and Automation, pp. 2128–2134. IEEE, Saint Paul, MN, USA (2012). https://doi.org/10.1109/ICRA.2012.6224899

[103] Bohigas, O., Henderson, M.E., Ros, L., Manubens, M., Porta, J.M.: Planning singularity-free paths on closed-chain manipulators. IEEE Transactions on Robotics **29**(4), 888–898 (2013). https://doi.org/10.1109/TRO.2013.2260679

[104] Porta, J.M., Jaillet, L., Bohigas, O.: Randomized path planning on manifolds based on higher-dimensional continuation. The International Journal of Robotics Research **31**(2), 201–215 (2012)

[105] Jaillet, L., Porta, J.M.: Path planning under kinematic constraints by rapidly exploring manifolds. IEEE Transactions on Robotics **29**(1), 105–117 (2013). https://doi.org/10.1109/TRO.2012.2222272