Submitted, 2011 International Conference on Intelligent Robots and Systems (IROS)
http://www.cds.caltech.edu/~murray/papers/hm11-iros.html

Geometric trajectory filtering via numerical conformal mapping

Shuo Han and Richard M. Murray

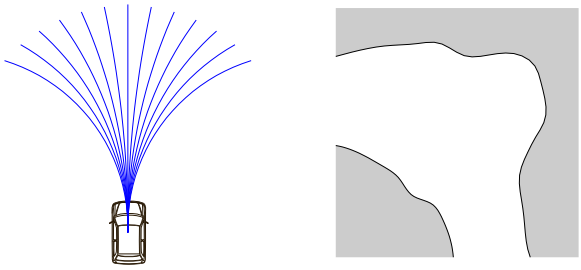
***Abstract*— The paper studies the problem which we refer to as geometric trajectory filtering, where only trajectories that satisfy the local safety constraints are selected from a library of trajectories. The goal is to speed up primitive-based motion planning while still maintaining a relatively a large collection of motion primitives. One way to solve this problem is to obtain a proper (preferably smooth) function, referred to as the containment indicator function, that describes the shape of the free space. To construct the containment indicator function for an arbitrary shape, the paper uses conformal mapping to transform the original shape of interest to a simpler target shape (e.g. disk, rectangle), which can then be characterized by elementary functions. Computational methods for finding the desired conformal maps are studied. It is shown that they can be formulated as convex optimization problems, whose solution can be obtained efficiently.**

I. Introduction

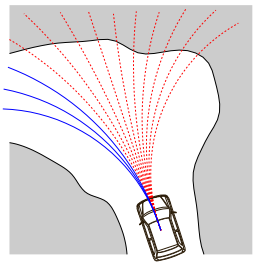
Motion planning is a field that has been studied extensively in robotics for over a couple decades. In its simplest form, the objective is to generate a trajectory (path) from a given initial state to a goal state (or a set of goal states), while satisfying (1) control input constraints and (2) safety constraints (e.g. avoiding obstacles). In general, there could be more than one feasible trajectory that satisfies these requirements. Therefore, in practice, one may be interested in finding the best trajectory under some cost metric, which gives rise to the problem of *optimal motion planning*. There are many methods for solving this problem, and can be classified mainly into the following categories.

The first class of methods is to formulate the problem as a constrained optimal control problem and solve for the optimal trajectory. Usually, some parametrization is adopted for the trajectory being optimized. In this way, the original optimal control problem can be converted into a nonlinear optimization problem over the parameters. This is known as the direct method in optimal control literature. There are also indirect methods that solve for both the trajectory and the associated Lagrange multipliers (also referred to as the "costates"or "adjoints"), but we do not review them here due to space constraints. The resulting nonlinear optimization problem can sometimes be computationally expensive to solve, unless special structure of the dynamics (e.g. differential flatness) is exploited [1], [2]. The other drawback is that the safety constraints need to be expressed in the form of a set of nonlinear inequalities, which can be difficult to handle. One way around this, for instance, is to use splines (e.g. non-uniform rational B-splines, or NURBS) and the convex hull property therein to remove the safety constraint, although the control points for splines still need to be chosen carefully [3].

The authors are with the Division of Engineering and Applied Science, California Institute of Technology, Pasadena, CA 91125, USA. hanshuo@caltech.edu, murray@cds.caltech.edu



(a) Possible motion primitives of a kine- (b) Local shape of the free space.
matic car.



(c) Filtered trajectories according to local environment. Blue lines correspond to admissible trajectories (the ones that lie in the free space).

Fig. 1. Illustration of geometric trajectory filtering.

The second class of methods is mainly based on subdividing the free space and solving the problem by constructing some discrete data structures such as trees or graphs over the subdivided space to represent the set of reachable states from the initial state. Each edge in the tree/graph normally corresponds to a certain "atomic motion" that happens within a short time step. By expanding the reachable set incrementally, the trajectory may be eventually obtained as soon as the goal state can be included in the set. Examples include the probabilistic roadmap (PRM) [4] and the rapidly-exploring random tree (RRT) [5], [6] or graph (RRG) [7]. The advantage of this is that it naturally handles the shape of the task space by using a discrete representation. However, most methods fail to handle optimality, since the data structure is primarily concerned with connectedness and/or reachability, although there are a few variants that cope with this issue (e.g. RRT*) [8].

The third class of methods can be seen as a mixture of the previous two, and often falls under the name *primitive-based motion planning* [9], [10]. The basic idea is to first adopt

optimal control methods to generate trajectory segments between some pairs of initial and goal states, while ignoring all the safety constraints. This is usually an offline process, and all the generated trajectory segments are stored as a library. After this, for a given specific environment and a goal state, the trajectory segments are retrieved from the library and pieced together, often by some graph search techniques, to form a feasible trajectory that satisfy the remaining constraints. Although there is no guarantee that the final formed trajectory will still be optimal, the trajectory is often acceptable, as long as each segment spans a considerable portion of the whole trajectory, since all the segments are optimal by themselves. In one extreme, where the entire trajectory consists of only one segment, the trajectory becomes optimal, recovering the optimal control method, but we would need to store a myriad of optimal segments in order to achieve this. In the other extreme, we can choose to store only the simplest motions that happen within short time durations and build a concise library; this will result in a trajectory consisting of a large number of small segments, thus recovering the discrete graph-search method.

During planning, one must only select that motion primitives that satisfy the local safety constraints, i.e., they are contained in the local free space. One prerequisite for doing this is to obtain a mathematical characterization of the local free space, usually by an indicator function that distinguishes the interior of the free space from the exterior. Since the shape of the free space can be arbitrary, the paper aims to find a universal method for obtaining the indicator function in an efficient way. Currently, our work is focused mainly on 2-dimensional free space, and uses tools in complex analysis for computing the indicator function. The specific method studied is numerical conformal mapping, which can be cast as convex optimization problems and solved efficiently.

II. Preliminaries and problem formulation

A. Notation and definitions

Since the paper relies heavily on results from complex analysis, we now introduce the notation used in this paper and familiarize the reader with some basic definitions. We use $\mathbb{C}$ to denote the entire complex plane. For any $z \in \mathbb{C}$, let $\mathrm{Re}(z)$ and $\mathrm{Im}(z)$ denote the real and imaginary part of z, respectively. The modulus of z is denoted as $|z| = [\mathrm{Re}^2(z) + \mathrm{Im}^2(z)]^{1/2}$; the argument of z is denoted as $\arg(z) = \{\mathrm{Arg}(z) + 2k\pi : k \in \mathbb{Z}\}$, where $\mathrm{Arg}(z) \in [0, 2\pi)$ is the principal value of argument[1]. Unless stated otherwise, every function (map) is a complex-valued function, defined from $\mathbb{C} \to \mathbb{C}$. A function f is called analytic (also referred to as holomorphic or regular in some references) in an open domain Ω if its complex derivative f' exists everywhere in Ω. Sometimes, the informal statement that "f is analytic at a point $z_0 \in \mathbb{C}$" is used to denote that f is analytic in an open neighborhood of z_0.

[1]Some references use $\mathrm{Arg}(z)$ to denote the argument and $\arg(z)$ the principal value.

An important class of analytical functions is conformal maps. In its original definition, an analytical map is called *conformal* if it preserves angles at every point z in its domain Ω. A straightforward application of the chain rule can show that an analytical map f is conformal if and only if $f'(z) \neq 0$ at every $z \in \Omega$. Also, all bijective maps are conformal.

B. Geometric trajectory filtering

Suppose we are given a discrete, time-indexed motion primitive $s = \{s_i\}_{i=0}^N$ in the configuration space $\mathcal{C}$, where $\mathcal{C}$ is a subset of $\mathbb{R}^n$, and each $s_i \in \mathcal{C}$. For motion planning, it is useful to select only the primitives that do not intersect with the obstacles present in the environment, namely the *admissible primitives*. Mathematically, we need to find an oracle that indicates whether a given point lies in the free space. One way of doing this is to construct a containment indicator function $g : \mathcal{C} \to \mathbb{R}$ that remains negative inside the free space and positive otherwise. If there is no smoothness requirement for the containment indicator function, one possible choice is to subdivide the free space into a disjoint union of convex polygons, similar to the occupancy grid, and construct g from the set of linear inequalities that characterize the interior of the polygons. This method will give an indicator function g that is piecewise linear.

However, under certain circumstances, a smooth indicator function is preferred. In primitive-based motion planning, for example, we do not solely want to select the motion primitives s that satisfy $g(s_i) \leq 0$ for all $s_i \in s$, which is too restrictive. For most mechanical systems in practice, even if a motion primitive does not satisfy the local safety constraint, we may still be able to apply rotation and translation to the trajectory and "fit" it inside the free space. More formally, this means there exists some group $\mathcal{Q} = \mathrm{SE}(n)$ whose action relates one feasible trajectory to another: if s is a feasible trajectory, then $\Phi_q(s)$ is also a feasible trajectory for any $q \in \mathcal{Q}$, where $\Phi : \mathcal{Q} \times \mathcal{C} \to \mathcal{C}$ denotes the group action of a group element q on s. We name this procedure of selecting the admissible motion primitives while allowing group actions as the *geometric trajectory filtering* problem, since the filtering (selection) depends on the local geometric configuration of the obstacles:

Problem 1 (Geometric trajectory filtering): Given a trajectory $s = \{s_i\}_{i=0}^N$ and a containment indicator function g, find a rigid-body transformation $q \in \mathrm{SE}(n)$ such that $g(\Phi_q(s_i)) \leq 0$ for all $s_i \in s$, if such q exists.

When solving the geometric trajectory filtering problem, it is preferred that the indicator function g is smooth, since otherwise it can prohibit the use of gradient methods in nonlinear optimization and/or root finding. Therefore, we are mainly interested in finding a smooth containment indicator function g to characterize any arbitrary free space. If we restrict ourselves to 2-dimensional free space, i.e. $\mathcal{C} \subset \mathbb{R}^2$, then it turns out one powerful method for obtaining g is to view $\mathbb{R}^2$ as the complex plane $\mathbb{C}$, and use numerical conformal mapping described in the following section.

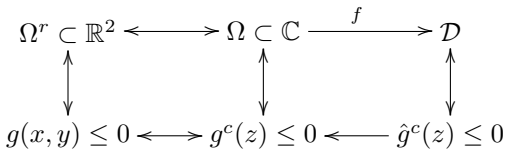


Fig. 2. The procedure of obtaining the indicator function g via conformal mapping f in the complex plane $\mathbb{C}$. The target domain $\mathcal{D}$normally has a simple shape that can be characterized by an elementary function $\hat{g}^c(z)$, which is then used to obtain the (complex) indicator function $g^c = \hat{g}^c \circ f$. The real containment indicator function g is related to g^c by the natural isomorphism from $\mathbb{R}^2$ to $\mathbb{C}$.

C. Computing the containment indicator function by numerical conformal mapping

In the 2D case, a free space of arbitrary shape, $\Omega^r \subset \mathbb{R}^2$, can be viewed alternatively as a domain $\Omega \subset \mathbb{C}$ by applying the natural isomorphism: $(x, y) \mapsto x + iy$. Therefore, finding an indicator function g that characterizes Ω^r is equivalent to finding g^c that characterizes Ω: $g^c(z) \le 0$ if and only if $z \in \Omega$. The basis idea in this paper for obtaining g^c is as follows: since g^c is generally difficult to obtain directly for an arbitrarily shaped domain Ω, we choose to first use a map f that transforms the original domain $\Omega = \{z : g^c(z) \le 0\}$ to a simpler target domain (e.g. unit disk, rectangle) $\mathcal{D} = \{z : \hat{g}^c(z) \le 0\}$, where the containment indicator function $\hat{g}^c(\cdot)$ can be expressed in a simple form. From there, the original containment indicator function can then be expressed as $g^c = \hat{g}^c \circ f$, if f is also bijective (one-to-one and onto).

This approach may seem unproductive at first glance, since it is still unclear how to find the bijective map f that maps the original domain Ω to a simpler domain $\mathcal{D}$. However, if we allow ourselves to extend the class of bijective maps to *conformal* ones (recall that bijectivity implies conformality), we can make use of some powerful computational tools to numerically obtain the map f (and hope that it is also bijective). These tools have traditionally been used for solving partial differential equations (e.g. Laplace equations) in two dimensions. Unfortunately, many existing results in computational conformal mapping do not directly apply in this case. For example, the Schwarz–Christoffel mapping can only transform a simple domain (the upper half-plane) to a more complex one (a simple polygon), but not vice versa. In the following, we will explore other numerical approaches that use existing theorems in complex analysis to obtain the desired conformal map f by exploiting the computational aspect of the theorems. It turns out that the required computation is solving a convex optimization problem, which can be carried out efficiently.

III. Containment indicator function from numerical conformal mapping

A. A motivating example

As a motivating example, consider the bent rectangular region as shown in Fig. 3, which is defined as

$$\Omega^r = \{p = (x, y) : \exists t,\ \|p - c(t)\| \le w/2\},$$

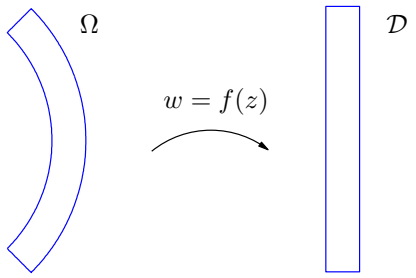


Fig. 3. An example of conformal mapping from a bent rectangle to a (regular) rectangle using logarithm.

where w is the width of the bent rectangle and

$$c(t) = (x_0 + r\cos t\theta_{\max}, y_0 + r\sin t\theta_{\max}),\ t \in [-1, 1],$$

for some $x_0, y_0, r, \theta_{\max} \in \mathbb{R}$, forms its center line.

This kind of region may appear locally, for example, for car race tracks. Although this region may seem complicated to characterize using some simple function g, we can try to first transform the region to some simpler region that is easier to describe. By viewing Ω^r as the equivalent subset in $\mathbb{C}$, namely Ω, we can use the following (complex) logarithmic map[2]

$$f(z) = \log(z - z_0), \quad z_0 = x_0 + iy_0$$

to convert Ω into a rectangle $\mathcal{D}$ shown in Fig. 3:

$$\mathcal{D} = \{z : \mathrm{Re}(z) \in [\log(r - w/2), \log(r + w/2)], \\ \mathrm{Im}(z) \in [-\theta_{\max}, \theta_{\max}]\}.$$

In this case, it is not difficult to write out the indicator function $\hat{g}^c$ for $\mathcal{D}$:

$$\hat{g}^c(z) \le 0 \quad \text{iff} \\ \mathrm{Re}(z) \in [\log(r - w/2), \log(r + w/2)], \\ \mathrm{Im}(z) \in [-\theta_{\max}, \theta_{\max}]$$

or more concisely,

$$\hat{g}^c(z) = \max \begin{bmatrix} \mathrm{Re}(z) - \log(r + w/2) \\ -\mathrm{Re}(z) + \log(r - w/2) \\ \mathrm{Im}(z) - \theta_{\max} \\ -\mathrm{Im}(z) + \theta_{\max} \end{bmatrix},$$

where the maximum is taken over all the entries in the vector. After this, we can combine the above to get the containment indicator function g^c for Ω:

$$g^c(z) = \max \begin{bmatrix} \mathrm{Re}(\log(z - z_0)) - \log(r + w/2) \\ -\mathrm{Re}(\log(z - z_0)) + \log(r - w/2) \\ \mathrm{Im}(\log(z - z_0)) - \theta_{\max} \\ -\mathrm{Im}(\log(z - z_0)) + \theta_{\max} \end{bmatrix}.$$

In addition, it is not difficult to verify that $f(z)$ is a bijective conformal map. Note that the key ingredient of the above procedure is to find the proper conformal map f that transforms Ω to a simple domain $\mathcal{D}$. Of course, the region

[2]The complex logarithm is defined as: $\log z \triangleq \log|z| + i\arg(z)$.

of interest in this particular example is a special shape that renders the use of logarithmic map possible. In the following, more general approaches will be introduced for transforming arbitrary (but simply connected) regions to simpler shapes.

B. Numerical conformal mapping by derivative maximization

We start by considering the unit disk as the target domain: $\mathcal{D}_1 = \{z \in \mathbb{C} : |z| \leq 1\}$. The Riemann mapping theorem in complex analysis ensures the existence of a bijective conformal map from any simply connected region (other than $\mathbb{C}$ itself) to a unit disk:

Theorem 1 (Riemann mapping theorem): For any simply connected domain $\Omega \subset \mathbb{C}$, there exists a bijective analytic (therefore conformal) function f that maps Ω to the unit disk.

Proof: The proof of this theorem involves quite some technical details (cf. [11]). However, some excerpts of the proof form the computational foundation of the algorithm that will be described in this section, so we will briefly sketch the proof below:

Consider the set of all the one-to-one (injective) functions that map Ω to $\mathcal{D}_1$. It can be shown that this set is non-empty and uniformly bounded. The proof then shows that there exists f in this set that maximizes $|f'(z_0)|$, where $z_0 \in \Omega \backslash \partial\Omega$ is an arbitrary (but fixed) point inside the domain of interest. It turns out that this particular f actually maps Ω onto $\mathcal{D}_1$ (i.e. f is surjective), and therefore is bijective. ■

The above proof gives a way to compute this desired f by casting an optimization problem to maximize the derivative at a given point strictly inside Ω, while restricting $|f(z)| \leq 1$ (inside the unit disk) for every $z \in \Omega$:

$$\begin{aligned} \underset{f}{\text{maximize}} \quad & |f'(z_0)|, \quad z_0 \in \Omega \backslash \partial\Omega \\ \text{subject to} \quad & \max_{z \in \Omega} |f(z)| \leq 1. \end{aligned}$$

Unfortunately, there are a couple of technical difficulties in solving the above problem in practice:

- First, the maximum in the constraint is taken over the entire region Ω; naively, this implies that we need to enumerate every points in Ω, which is impossible.
- Second, the optimization is performed over a function space that is infinite-dimensional.

Therefore, we need to make a few approximations to the original problem. For the first one, we can choose to sample a number of points $z_i \in \Omega$, and hope to obtain a good approximation by choosing the number to be large enough. In fact, there exists a more efficient solution than this. Recall that the maximum modulus principle (cf. [12]) guarantees that the modulus $|f(z)|$ can only attain its maximum on $\partial\Omega$, the boundary of Ω, unless f is constant in Ω:

Theorem 2 (Maximum modulus): If f is analytic in a domain Ω and $|f(z_0)| = \max_{z \in \Omega} |f(z)|$ for some $z_0 \in \Omega \backslash \partial\Omega$, then $f(z) \equiv c$ $(\forall z \in \Omega)$ for some constant c.

Therefore, we can ignore all the points that lie in the interior of Ω and only sample on $\partial\Omega$. For the second one, we choose to restrict the form of f to a linear combination of basis functions:

$$f(z) = \sum_{i=1}^{n} w_i \phi_i(z) = \Phi^T(z) w,$$

where each $\phi_i(z)$ is analytical. A possible choice for the basis function is monomials in z, which makes $f(z)$ a polynomial. This not only allows us to cast the original infinite-dimensional optimization problem into a finite-dimensional one, but also makes the optimization problem convex, as will be seen later. We now rewrite the approximate optimization problem below:

$$\begin{aligned} \underset{w}{\text{maximize}} \quad & \left|\Phi^{'}(z_0)^T w\right|, \quad z_0 \in \Omega \backslash \partial\Omega \\ \text{subject to} \quad & \left|\Phi^T(z_i) w\right| \leq 1 \quad (i = 1, 2, \cdots, N). \end{aligned}$$

In its natural form, the above optimization problem is non-convex, because the objective function being maximized is not concave (in fact, it is convex). However, note that for any w^* that attains maximum, γw^* ($|\gamma| = 1$, $\gamma \in \mathbb{C}$) also gives the same optimal value, since:

$$\left|\Phi^{'}(z_0)^T (\gamma w^*)\right| = \left|\Phi^{'}(z_0)^T w^*\right| |\gamma| = \left|\Phi^{'}(z_0)^T w^*\right|.$$

Therefore, it is without loss of generality to require that $\Phi^{'}(z_0)^T w$ be real and recast the optimization problem as

$$\begin{aligned} \underset{w}{\text{maximize}} \quad & \mathrm{Re}\{\Phi^{'}(z_0)^T w\}, \quad z_0 \in \Omega \backslash \partial\Omega \\ \text{subject to} \quad & \left|\Phi^T(z_i) w\right| \leq 1 \quad (i = 1, 2, \cdots, N), \end{aligned}$$

where now the problem becomes convex[3], since $\mathrm{Re}(\cdot)$ is linear and thus the objective function is linear.

C. Numerical conformal mapping by mimimax modulus

Another seemingly different way of computing the conformal map f is to minimize the radius (instead of unity) of the target disk region that f maps to [13]. Without loss of generality, the approach assumes that the region of interest Ω contains $z = 0$, which can always be achieved by translation. The existence and uniqueness of the conformal map to be computed in given by the following theorem:

Theorem 3: (cf. [14]) Let $\Omega \subset \mathbb{C}$ be a simply connected region that includes $z = 0$. Denote $\mathcal{H}_1$ as the following set:

$$\mathcal{H}_1 = \{f : f(0) = 0,\, f'(0) = 1,\, f \text{ is analytic in } \Omega\}.$$

Then there exists a unique bijective $f_{\min}$ that minimizes the quantity $M(f) = \max_{z \in \Omega} |f(z)|$:

$$f_{\min} = \arg \min_{f \in \mathcal{H}_1} \max_{z \in \Omega} |f(z)|.$$

Furthermore, the function $f_{\min}$ maps Ω to a disk $\mathcal{D} = \{z : |z| \leq r^*\}$, where $r^* = M(f_{\min})$.

Remark 1: The function $f_{\min}$ is conformal, since $f'_{\min}(0) = 1$ by the definition of $\mathcal{H}_1$.

[3]In fact, a quadratic program.

The above theorem also gives us a numerical way to solve for the conformal map by formulating the following infinite-dimensional optimization problem:

$$\begin{aligned} \underset{f\in\mathcal{H}_1}{\text{minimize}} \quad & \max_{z\in\Omega} |f(z)| \\ \text{subject to} \quad & f(0)=0,\, f'(0)=1. \end{aligned}$$

To make the optimization computationally tractable, we follow the approach similar to the previous one and approximate f as a linear combination of basis functions: $f(z)=\sum_i w_i\phi_i(z)=\Phi^T(z)w$. By invoking the maximum modulus principle again, we only sample $\partial\Omega$ at a finite number of locations $\{z_i\}_{i=1}^N$. The point $z_0\in\Omega\backslash\partial\Omega$ can be chosen arbitrarily:

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & \max_{i} \left|\Phi^T(z_i)w\right| \quad (i=1,2,\cdots,N) \\ \text{subject to} \quad & \Phi^T(0)w=0,\, \Phi'(0)w=1. \end{aligned}$$

The objective function is the maximum of norms, and is therefore convex; the constraints are linear equalities in w. Therefore, the above optimization problem is convex. It can be shown that this computed conformal map f only differs possibly by a (complex) scaling factor and a displacement from the one computed in the previous section, by the uniqueness of the conformal map as stated in Theorem 3, since both algorithms will produce a conformal map that transform the same domain (up to a displacement) to a disk. Moreover, it has been shown that by replacing the modulus $|\cdot|$ in the objective function with other admissible functions $v(\cdot)$, the above method can also map a given region to other simply connected domains (e.g., infinite stripes). The requirement on v is that its needs to be a subnorm (see [13] for definitions). For any subnorm v and radius $r>0$, we can define its associated disk as:

$$\mathcal{D}_v(r)=\{z:\, v(z)\le r\}.$$

For example, $v(z)=|\mathrm{Re}(z)|$ is a subnorm, and the corresponding $\mathcal{D}_v(r)$ is an infinite stripe. The maximum principle still holds for subnorms, which avoids the need of sampling inside the region of interests and allows us to form a more generalized optimization problem similar to the previous one:

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & \max_{i} v(\Phi^T(z_i)w) \\ \text{subject to} \quad & \Phi^T(0)w=0,\, \Phi'(0)w=1, \end{aligned}$$

where z_i are points sampled on $\partial\Omega$. The only caveat is that not all subnorms v are convex, so the generalized problem may not remain convex. However, there are some convex subnorms, such as $v(z)=|\mathrm{Re}(z)|$, under which we can still solve the optimization problem efficiently.

IV. Numerical examples and discussions

We now give a few numerical examples with different simply connected shapes for Ω (both convex and non-convex). Although the various theorems in the previous section dictates the existence of such a bijective map f from Ω to a certain target domain (disk, or some other simply connected domain defined by the subnorm v), both computational approaches by themselves do not guarantee that the result will give a bijective map. The reason is that they do not optimize over the set of all admissible functions on $\mathbb{C}$ and have instead made the approximation to restrict f within a special class of functions. Therefore, we should check if the generated map f is bijective. Unfortunately, there is no known efficient method to check whether a given map f is bijective other than to naively check against every point $z\in\Omega$ and their image $f(z)$. However, it can be proved that it suffices to check the points on $\partial\Omega$ to test whether a map f is injective, as stated in the following theorem.

Theorem 4: (cf. [15]) If the map $f\colon \Omega\to\mathcal{D}$ is analytic in $\Omega\backslash\partial\Omega$ and on $\partial\Omega$. Suppose $f(z)$ is one-to-one on $\partial\Omega$, and Ω is simply connected, then (1) $f(z)$ maps $\partial\Omega$ to $\partial\mathcal{D}$, and (2) $f(z)$ is also one-to-one from Ω to $\mathcal{D}$.

Informally, this means we can check if f is injective by checking if the image of $\partial\Omega$ self-intersects.

In the following, we will test a few combinations of original domains Ω and target domains $\mathcal{D}$ listed in Table I. For the last two examples, we also compare the effect of using different target domains. All the examples search over polynomials up to degree n:

$$f(z)=\sum_{k=0}^{n} a_k z^k,$$

except for the last one, where additional basis functions consisting of logarithms are also considered:

$$f(z)=\sum_{k=0}^{n} a_k z^k+\sum_{k=1}^{m} b_m \log^m z.$$

#	Original domain Ω	Target domain $\mathcal{D}$
1	Deformed disk	Disk
2	Square	Disk
3	Nonconvex polygon	Disk
4	Elongated rectangle	Disk and ∞-normed disk
5	Bent rectangle	Disk and ∞-normed disk

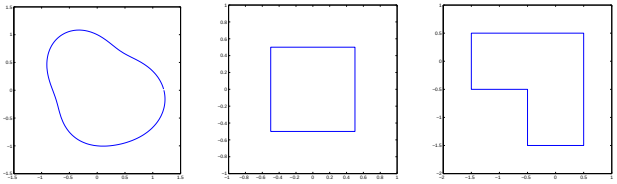
TABLE I

EXAMPLES OF ORIGINAL AND TARGET DOMAINS UNDER TEST.

It can be seen, from Fig 4, that the computed conformal maps transform the original domain to the target domain perfectly for Example 1 and 2, and almost perfectly for Example 3 (the boundary loops itself over a bit) in Table I, possibly due to the non-convexity of Ω and the polynomial basis functions being used (since $|z|^k$ is strictly increasing). Two issues have been found critical for this method of numerical conformal mapping. First, the target domain (i.e. the subnorm $v(z)$) needs to be chosen properly based on the shape of the original domain. This can be seen in Example 4, where the computed conformal map fails to transform it to a disk properly (Fig. 5b), because the original region has a high aspect ratio. Instead, a different subnorm $v(z)$ is used, which is a weighted ∞-norm:

$$v(z)=\max(|\mathrm{Re}(z)|,\mu\,|\mathrm{Im}(z)|) \quad (\mu>0),$$

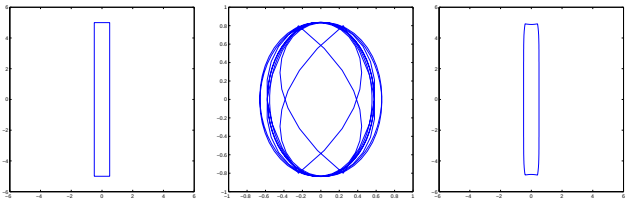
under which the "normed disk" becomes a rectangle again. Not surprisingly, it transforms the original region properly (Fig. 5c), due to the fact that the normed disk resembles the original domain itself. Second, choice of the basis functions plays a key role in transforming more irregular shapes. In Example 5, the polynomial basis performs poorly and is not able to generate a bijective conformal map (Fig. 6b), whereas adding logarithmic basis functions helps greatly (Fig. 6c).



(a) #1 (original) (b) #2 (original) (c) #3 (original)

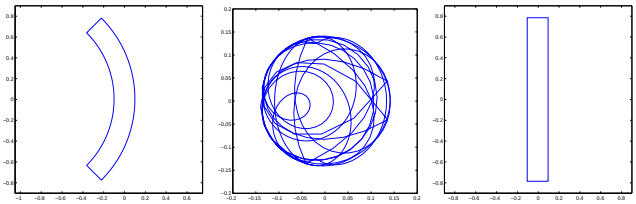
(d) #1 (target) (e) #2 (target) (f) #3 (target)

Fig. 4. Original and target domains for Example 1–3. Only the boundaries of the domains are shown for clarity. The computed conformal maps have successfully transformed the original domains of interest to the target domains (disks for all the examples).



(a) #4 (original) (b) #4 (target = disk) (c) #4 (target = rect, i.e., ∞-normed disk)

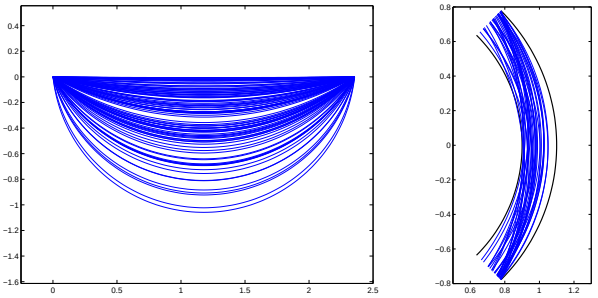
Fig. 5. Original and target domains for Example 4. Only the boundaries of the domains are shown for clarity.



(a) #5 (original) (b) #5 (target = disk, basis functions = polynomials) (c) #5 (target = rect, basis functions = polynomials + logarithms)

Fig. 6. Original and target domains for Example 5. Only the boundaries of the domains are shown for clarity.

Using the computed conformal map, we solve the geometric trajectory filtering problem for the original domain Ω in Example 5 listed in Table I. Fig. 7a shows the



(a) (b)

Fig. 7. An example of geometric trajectory filtering. (a) Motion primitives under consideration. (b) Filtered motion primitives according to the shape of the free space, using the containment indicator function as in Fig. 6c.

synthetic library of motion primitives of a kinematic car, $\mathcal{L} = \{s^{(k)}\}_{k=1}^{M}$, which consists of a total M arcs with different curvature. Formally, we need to obtain feasible solutions $\bar{s} = \{\bar{s}_i\}_{i=0}^{N}$ that satisfy:

1) $\bar{s}_i = \Phi_q(s_{i+j}^{(k)})$ $(0 \le i \le N)$ for some $j, N \in \mathbb{Z}$, $s^{(k)} \in \mathcal{L}$, and $q \in \mathrm{SE}(2)$. This implies that $\bar{s}$ is generated by applying rigid-body transformation on part of some trajectory in the library.
2) $g(\bar{s}_i) \le 0$ for all $s_i \in s$, where g is the conformal map computed previously.

The filtered trajectories are computed by solving a nonlinear optimization problem (with constant objective function, for feasibility only) using `fmincon` in MATLAB. A number of trajectories are plotted in Fig. 7b, which indicates that they all lie in the domain of interest. These trajectories can then be viewed as admissible motion primitives during the later motion planning stage.

V. Conclusions

The paper has described methods for numerically computing the containment indicator function, which is used to describe the local shape of the free space for *geometric trajectory filtering*. The key ingredient is to transform the original shape of interest to a simpler target shape via *numerical conformal mapping*, which can be cast as convex optimization problems, at least approximately. Several different shapes have been tested for being converted into simpler shapes, e.g., disks and rectangles. Two issues have been found critical for the algorithms described: (1) target region needs to be chosen properly according to the original shape of interest; (2) the basis functions can affect the quality of the conformal map greatly (e.g., sometimes polynomials are not good enough to provide a useful conformal map). The computed conformal map has also been used in trajectory filtering. Some preliminary results have been shown, using a synthetic trajectory library consisting of motion primitives of a kinematic car (i.e. arcs). Future issues to be addressed include investigating a richer class of basis functions, and testing the algorithm under more realistic environment configurations.

Acknowledgements. This work is supported by the National Science Foundation (NSF) grant CNS-0931746.

REFERENCES

[1] M. Milam, K. Mushambi, and R. Murray, "A new computational approach to real-time trajectory generation for constrained mechanical systems," in *IEEE Conf. on Decision and Control*, vol. 1, pp. 845–851, IEEE, 2000.

[2] M. Milam, *Real-time optimal trajectory generation for constrained dynamical systems*. PhD thesis, California Institute of Technology, 2003.

[3] M. Flores and M. Milam, "Trajectory generation for differentially flat systems via NURBS basis functions with obstacle avoidance," in *American Control Conf.*, p. 7, IEEE, 2006.

[4] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *Trans. on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[5] S. LaValle and J. Kuffner, "Randomized kinodynamic planning," *Int. J. of Robotics Research*, vol. 20, no. 5, p. 378, 2001.

[6] D. Hsu, R. Kindel, J. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *Int. J. of Robotics Research*, vol. 21, no. 3, p. 233, 2002.

[7] S. Karaman and E. Frazzoli, "Sampling-based motion planning with deterministic μ-calculus specifications," in *IEEE Conf. on Decision and Control*, 2009.

[8] S. Karaman and E. Frazzoli, "Incremental sampling-based optimal motion planning," in *Robotics: Science and Systems*, 2010.

[9] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 25, no. 1, pp. 116–129, 2002.

[10] K. Hauser, T. Bretl, J. Latombe, K. Harada, and B. Wilcox, "Motion planning for legged robots on varied terrain," *Int. J. of Robotics Research*, vol. 27, no. 11-12, p. 1325, 2008.

[11] S. Lang, *Complex Analysis*. Springer Verlag, 1999.

[12] E. Saff and A. Snider, *Fundamentals of complex analysis with applications to engineering, science, and mathematics*. Prentice Hall, 2003.

[13] G. Opfer, "Conformal mappings onto prescribed regions via optimization techniques," *Numerische Mathematik*, vol. 35, no. 2, pp. 189–200, 1980.

[14] G. Goluzin, *Geometric theory of functions of a complex variable*. AMS Bookstore, 1983.

[15] M. Ablowitz and A. Fokas, *Complex variables: introduction and applications*. Cambridge Univ Pr, 2003.