Sampling-based multi-agent choreography planning: a metric space approach

Anton Lukyanenko and Damoon Soubakhsh

Abstract—We present a metric space approach for high-dimensional sample-based trajectory planning. Sample-based methods such as RRT and its variants have been widely used in robotic applications and beyond, but the convergence of such methods is known only for the specific cases of holonomic systems and sub-Riemannian non-holonomic systems. Here, we present a more general theory using a metric-based approach and prove the algorithm’s convergence for Euclidean and non-Euclidean spaces. The extended convergence theory is valid for joint planning of multiple heterogeneous holonomic or non-holonomic agents in a crowded environment in the presence of obstacles. We demonstrate the method both using abstract metric spaces ($\ell^p$ geometries and fractal Sierpinski gasket) and using a multi-vehicle Reeds-Shepp vehicle system. For multi-vehicle systems, the degree of simultaneous motion can be adjusted by varying the metric on the joint state space, and we demonstrate the effects of this choice on the resulting choreographies.

1 INTRODUCTION

In this paper, we address the problem of joint motion planning for a fleet of multiple heterogeneous holonomic or non-holonomic robots. Taking a metric space perspective, we expand the framework for sample-based algorithms commonly used for trajectory planning in such applications and demonstrate the algorithm in both abstract metric spaces and in the case of a fleet of collaborative vehicles.

Even in the absence of obstacles, robotic path planning can be a challenging task outside of a handful of simple cases (e.g. Euclidean straight lines and spherical great circles). In the Riemannian setting governing holonomic motion, the task is aided by Pontryagin’s Maximum Principle and the Riemannian geodesic equations. The non-holonomic case governed by sub-Riemannian and sub-Finsler geometry is well-explored, but obstacle-free path planning remains an active area of research, see e.g. [29], [15], [2], [3].

In the context of spatial restrictions and multi-vehicle collision avoidance, exact path planning becomes intractable, even with full knowledge of the underlying geometry. We therefore focus on sampling-based motion planners, and providing an in-depth analysis of the underlying geometry and auxiliary functions that are needed to find feasible and asymptotically optimal solutions.

In the holonomic case, our approach recovers the original convergence guarantees of [17], which had to be amended in [36] with a larger connection radius $c\left(\frac{\log n}{n}\right)^{\frac{1}{2m}}$, while largely maintaining the spirit of the original proof. As noted in [36], the original connection radius of $c'\left(\frac{\log n}{n}\right)^{\frac{1}{2}}$ has been successfully used in practical applications, and our proof confirms convergence with this tighter connection radius.

We address both types of common motion planning methods: the Probabilistic Roadmaps (PRMs) introduced by Kavraki et al. [21] and the Rapidly-exploring Random Trees (RRTs) of Lavalle [23]. Both algorithms sample a configuration space and build a graph-theoretic approximation of reachable destinations. PRM is optimized for multi-query applications, in which the starting and ending configurations may change between calls, and is performed in two stages: the roadmap phase that builds an approximating graph and a path creation phase that identifies trajectories through the graph. RRT, on the other hand, is optimized for single-query applications and builds a tree of possible trajectories.

The core PRM and RRT algorithms have been shown to be effective at finding feasible trajectories in a wide variety of single-robot holonomic and non-holonomic planning tasks [21], [23], [25], [33], [24], [1], [9], [12]. In [17] two influential variants of the algorithms called PRM* and RRT* were introduced, providing provable asymptotic optimality in the Euclidean setting without sacrificing performance speed. The results were extended to the sub-Riemannian setting in [16]. Recently, an error was discovered in the analysis of RRT*, and an alternate proof was provided for the holonomic case in [36].

Optimizing sample-based motion planning for a single-vehicle has been the subject of many new studies, focusing on developing hybrid discrete-continuous algorithms [32], addressing moving obstacles [5], [31], improving nearest-neighbor searches for non-holonomic robotics [42], [41], improving search regions [8], [43], [48], [47], and incorporating machine learning methods [4], [11], [49]. The computational time of RRT methods can be improved using biased or deterministic sampling [35], [14], [7], [13], [37].

Sample-based motion planning of multiple robots suffers from an exponential growth in the number of necessary samples [24]. The subdimensional expansion approach [45], [44] avoids this by planning each robot’s movements independently, and then resolving any potential collision through local planning in a “bubble space”. In previous studies, local planning was based on robot task reassignment [46], or priority-based motion with holding patterns [40], [39], [27], and localized RRT planning [19].

Here, we address the problem of coupled motion planning of a small number of robots. Such local planning increases the efficiency of multi-robot motion planning. The local trajectory planning is conducted after reducing the system’s dimension at a local level to resolve conflicts using subdimensional expansion or similar approaches. In such scenarios, the robots enter a box defined by the higher-order planner and leave it according to the poses defined by the planner. Our solution is for a general problem without requiring interchangeable robots, and all such simplifications using task allocation or similar methods have been taken into con-
sideration, and further simplifications are not possible. We use sample-based motion planning of higher-order systems (multiple vehicles) and provide a new proof of guaranteed convergence and optimality.

A preliminary version of this paper appeared as [26], which summarized the necessary assumptions for convergence and demonstrated motion planning for a small number of simulated and physical robots. Our key contributions here are as follows:

- Presenting a more general theory for sample-based motion planning of higher-order systems in a metric space.
- Proving the asymptotic convergence of the presented method under generalized the assumptions assumptions.
- Investigating the effect of the metric coupling parameter for the fleet on the resulting choreographies.
- Showing the applicability of the method in more general settings such as the fractal Sierpinski gasket.
- Verifying the results with simulations and planning of multiple nonholonomic car-like robots.

1.1 Statement of the Problem

Our goal in this paper is to provide a solution to a generalization of the Euclidean path-finding problem, which is often resolved using RRT*.

The path-finding problem is commonly studied under the following assumptions (e.g. in [17], [16]):

**Problem 1.** (Classic path-finding problem) Start with an obstacle set \( \text{Obst} \) inside a sampled search region \( X_{\text{rand}} \) inside Euclidean space \( X = \mathbb{R}^n \). Assume that:

1. A function \( \text{Extend} \) provides optimal paths between points \( x, y \) in \( X \) for a Euclidean, Riemannian, or sub-Riemannian metric \( d \) with Hausdorff dimension \( Q \).
2. A sampling method \( \text{Random} \) provides points that are uniformly distributed on \( X_{\text{rand}} \) with respect to Lebesgue measure (Euclidean volume).
3. The obstacle set \( \text{Obst} \) is open.
4. A cost functional \( \text{Cost} \) is defined for all finite-length paths by integrating a pointwise-defined cost function.

One seeks to find the minimizer of \( \text{Cost} \) among all paths between two given points \( x, y \in X_{\text{free}} = X_{\text{rand}} \setminus \text{Obst} \), if one exists. In relaxed versions of the problem, one is interested in finding approximate minimizers, or simply a finite-cost path joining \( x \) and \( y \) in \( X_{\text{free}} \). The starting and final points may be known at the start.

Path planning for advanced applications, such as choreography planning for a fleet of non-holonomic robots with asymmetric directional preferences, is not described by the classic path-finding problem. We broaden the framework to accommodate such examples, and take the opportunity to adjust \( \text{Obst} \) to avoid undetectable zero-width tunnels and non-robust optimizers (cf. [17]).

**Problem 2.** (Generalized path-finding problem, see §3 for the precise formulation) Start with a free space \( X_{\text{free}} \) contained in a sampled space \( X_{\text{rand}} \) inside an ambient space \( X \). Assume that:

1. A function \( \text{Extend} \) provides approximately-optimal paths between points \( x, y \in X \) and induces a proper pseudo-metric \( d \) on \( X \).
2. A sampling method \( \text{Random} \) provides points that are uniformly distributed on \( X_{\text{rand}} \) with respect to a Borel measure \( \mu \) such that the probability of sampling a ball of radius \( r \) is approximately \( r^Q \).
3. \( X_{\text{free}} \) is open and connected.
4. A cost functional \( \text{Cost} \) is defined for all finite-length paths, and is continuous with respect to the length \( \ell \).

With the above data, and given points \( x, y \in X_{\text{free}} \), one seeks to find a path from \( x \) to \( y \) within \( X_{\text{free}} \) that minimizes \( \text{Cost} \), approximately minimizes cost, or simply has finite cost.

A (sampling-based) algorithm solving the path-finding problem is complete if it (almost surely) finds a finite-cost solution within finite time, and is asymptotically optimal if it (almost-surely) approximately-minimizes cost to arbitrarily small error.

1.2 Contributions

In this paper, we show that sampling-based algorithms including RRT* and PRM* (see §3.2) are able to solve the generalized path-finding Problem 2. We then demonstrate path-finding in non-Euclidean and fractal spaces, focusing on coupled robotic systems.

Under our more general assumptions, we recover the following theorem. In the Euclidean setting, it recovers the original convergence guarantees of [17], and improves on the revised guarantees provided in [36].

**Theorem 3.** All RRT-type algorithms (see Definition 9), including RRT and RRT* are complete. Furthermore, if the proximity function \( r_n \) used by the RRT* and PRM* satisfies, for the constant \( C_r \) given by (5),

\[
r_n \geq C_r \left( \frac{\log n}{n} \right)^{1/Q},
\]

then both RRT* and PRM* are complete and asymptotically optimal.

We prove Theorem 3 in three steps (see §2 for a simplified example). First, we use a topological argument to show that RRT-type algorithms are complete, and in fact for every \( \epsilon > 0 \) almost surely provide an \( \epsilon \)-dense net in \( X \) within finite time. Second, we use this to reduce the analysis of RRT* to that of PRM* by showing that away from obstacles RRT* effectively turns into PRM* after finitely many iterations. This is done by defining a sequence of algorithms that interpolate between RRT* and PRM* in a small neighborhood of a given curve. Third, we update the proof of PRM* completeness and optimality to our broader assumptions. This third step largely follows the work of Karaman-Frazzoli, combining a Borel-Cantelli argument with the theory of random graphs and the idea of Poissonization (which decouples repeated samples of two or more correlated random variables), so we present a simplified proof of this step, under the assumption that \( r_n / \left( \log n \right)^{1/Q} \to \infty \), see Theorem 15.

The last step of the proof of Theorem 3 follows the study of PRM* by Karaman-Frazzoli, it fills in some necessary
details that are left unclear in their work, while also extending the result to a broader set of assumptions. In the case of RRT*, our interpolational approach substantially simplifies the proof of optimality, while also elucidating the relationship between RRT* and PRM*.

Following the proof of Theorem 3, we study applications to path planning in non-Euclidean spaces. We first discuss the simple example of coupled one-dimensional robots, showing how the choice of an ℓp metric on the joint space can affect the resulting paths. We then look at a more abstract example of path-finding in the Sierpinski gasket. Finally, we use RRT* to provide coordinated paths (“choreographies”) for a small fleet of Reeds-Shepp vehicles [34], [38]. We show that the algorithm is effective for localized planning, and demonstrate the influence of the choice of metric on the joint planning space on the resulting choreographies.

1.3 Organization

The paper is organized as follows. We set the stage by proving the convergence and optimality of the simplified probabilistic roadmap (sPRM) in §2. We lay out the assumptions and conventions for the Generalized Path Finding Problem in §3. We prove Theorem 3 in §4. We demonstrate path planning in Finsler and fractal non-Euclidean spaces in §5. We then present the application of the theorem to trajectory planning of multiple vehicles in §6. Concluding remarks are given in §7.

2 ILLUSTRATIVE EXAMPLE: sPRM

We first quickly recall the standard approach to analyzing path-finding algorithms using the simplest complete and asymptotically optimal algorithm: the Euclidean simplified probabilistic roadmap (sPRM) [21].

Algorithm 1. (sPRM) The sPRM algorithm returns a graph $G$ representing $X_{\text{free}}$ as follows. The vertices of $G$ are comprised of $n$ independent samples $x_1, \ldots, x_n \in X_{\text{free}}$. Two vertices $(x_i, x_j)$ are connected by an undirected edge whenever $|x_i - x_j| < r$ and the line segment between $x_i$ and $x_j$ remains in $X_{\text{free}}$.

The graph $G$ returned by sPRM is used to plan a path between points $p, q \in X_{\text{free}}$ by adding two more samples equal to $p$ and $q$ and then using a graph-based path finding algorithm. More precisely, let $x_p$ and $x_q$ be vertices of $G$ minimizing the distance to $p$ and $q$, respectively. The points $p, q$ are added to the vertex set of $G$, and connected to $x_p$ and $x_q$, respectively, if the corresponding line segments are in $X_{\text{free}}$ and satisfy $|x_i - x_j| < r$. Next, each edge of the graph is labeled with a weight corresponding to its length. A graph-based path-planning algorithm (e.g. Dijkstra or A*) can then be used to compute the optimal path from $p$ to $q$ within $G$, if it exists.

Theorem 4. The sPRM algorithm is complete and asymptotically optimal.

Proof. Let $p, q \in X_{\text{free}}$ be two points that can be connected by a finite-length curve in $X_{\text{free}}$. Let $L$ denote the infimum of the lengths of such curves, i.e. $L := \inf_\gamma \ell(\gamma)$ where the infimum is taken over all finite-length curves $\gamma$ connecting $p$ to $q$ in $X_{\text{free}}$.

We claim that we can find a path whose length is arbitrarily close to $L$, say within $2\epsilon$. Indeed, let $\gamma$ be a free-space path from $p$ to $q$ satisfying $\ell(\gamma) < 2\epsilon$. We claim that sPRM eventually finds a path $\gamma'$ that approximates $\gamma$ and satisfies $|\ell(\gamma') - \ell(\gamma)| < \epsilon$, so that $\ell(\gamma') - L < 2\epsilon$, as desired.

If necessary, reparametrize $\gamma$ by unit length. Let $t_1 = 0 < t_1 < \ldots < t_{m-1} < t_m = 4\ell(\gamma)$ be a sequence of time instants such that the discretized length agrees with the true length up to $\epsilon/2$:

$$\sum_{i=0}^{m-1} |\gamma(t_i) - \gamma(t_{i+1})| - \ell(\gamma) < \epsilon/2.$$  

If necessary, select a further refinement such that $|t_i - t_{i+1}| < r/2$, where $r$ is the connection radius in sPRM. Thicken up each point to create a “target” ball $B_i = B(\gamma(t_i), r')$, where $r' = \min(\ell(4m), r/4)$.

Since each $B_i$ has positive volume, for large values of $n$ the sampling process will provide a point $x^i$ in each $B_i$ (more formally, the probability of a sampling failure at this step goes to 0 as $n \to \infty$). If we are able to join consecutive points by collision-free edges, we obtain a path $\gamma'$ whose length is bounded above, by repeated applications of the triangle inequality at each edge, by

$$\sum_{i=0}^{m-1} |x^i - x^{i+1}| \leq \sum_{i=0}^{m-1} 2\epsilon + |t_i - t_{i+1}| \leq \epsilon + \ell(\gamma),$$

as desired.

Fig. 1: The path $\gamma$ (black) is approximated by a path $\gamma'$ (blue) using samples inside targets (red). Limiting the radius of the targets controls oscillation in $\gamma'$, ensuring that $\text{Cost}(\gamma')$ is close to $\text{Cost}(\gamma)$.

The only remaining question is whether the connections are, in fact, made. This can fail in two cases: if $|x^i - x^{i+1}| > r$, or if the segment between $x^i$ and $x^{i+1}$ intersects an obstacle. The first situation is avoided by our choice of $r'$. The second situation can be avoided by computing the distance from $\gamma$ to the closest obstacle and restricting all distances accordingly (see Lemma 8 for details).

We have thus obtained an approximation $\gamma'$ of $\gamma$ to error $\epsilon$, and therefore have an approximate optimizer for the search problem, with error at most $2\epsilon$, as desired.

\section{Assumptions and Algorithms}

The sPRM algorithm in §2 searches for the shortest-length path within a region $X_{\text{free}}$ in Euclidean space. This is the most-studied set of assumptions for holonomic motion planning, and provides a piecewise-linear solution to the problem. The sPRM algorithm suffers from two drawbacks: restrictive assumptions and sub-optimal memory utilization.

\footnote{This is possible by the definition of length via bounded variation.}
We now broaden the assumptions on the underlying geometry, elaborating on Problem 2, to allow path planning for all holonomic and non-holonomic motion, as well as in certain more general metric spaces. We then recall the definitions of the more memory-efficient algorithms PRM*, RRT, and RRT*.

### 3.1 Assumptions and conventions

We now carefully discuss our assumptions and conventions for Problem 2, which can be informally summarized as follows:

1. The function \( d(x, x') \) is a quasi-distance on a space \( X \).
2. \( \mu \) is a \( Q \)-dimensional measure on \( X \), and \( \text{Random} \) is uniformly distributed on a search space \( X_{\text{rand}} \subset X \) with respect to \( \mu \).
3. The function \( \text{Extend}(p, q) \) provides a path from \( p \) to \( q \) for sufficiently nearby points in \( X \), and is asymptotically geodesic.
4. The \( \text{Cost} \) functional on paths is continuous with respect to the length functional and spatial convergence.
5. The free space \( X_{\text{free}} \subset X_{\text{rand}} \) is path-connected and open.
6. The path-finding algorithm makes use of a proximity threshold \( r_n \), with \( r'_n \geq C_r \left( \frac{\log n}{n} \right)^Q \) where the constant \( C_r \) is given by (5).
7. It is sufficient for the path-finding algorithm to return a certain marked graph.

#### 3.1.1 Distances: We first adjust the ambient space and distance metric.

In practice, \( X \) will be identified with \( \mathbb{R}^n \), and \( d \) will specify a non-Euclidean metric on \( X \).

**Assumption 1.** (Distances) We assume that the function \( d \) on \( X \) satisfies the usual axioms for a metric function, with the weakened symmetry assumption: \( d(x, x') \leq C_d d(x', x) \) for some \( C_d > 0 \) and all \( x, x' \in X \); and that the induced topology is proper: that is, closed balls \( B(x, r) \) are compact.

The weakened symmetry assumption allows encoding directional systems, such as vehicles with a preference for forward rather than backward motion. In discussing metric balls, we will work with left-sided balls unless otherwise stated: \( B(x, r) = \{ x' : d(x, x') < r \} \).

The metric \( d \) provides a notion of convergence and continuity (that is, a topology\(^2\)), and the properness assumption ensures that these concepts behave as expected at small scales.

**Question 5.** Is it possible to fully remove the symmetry assumption? This would be desirable for, e.g., prohibiting backward motion as in the Dubins model of vehicle motion. However, this introduces multiple pathologies, such as \( \epsilon \)-ball having arbitrarily large diameter, see e.g. [28].

#### 3.1.2 Sampling and dimension: We next generalize the notion of volume \( \mu \), with respect to which \( \text{Random} \) is uniformly distributed on \( X_{\text{rand}} \).

---

\(^2\)See [30] for the topological terminology. Note also that the topologies induced by left-sided and right-sided balls coincide under the weakened symmetry assumption.

---

**Assumption 2.** (Sampling and dimension) We assume that \( \mu \) is a Borel measure on \( X \) satisfying

\[
\mu(B(x, r)) \geq C_\mu r^Q \tag{2}
\]

for any ball \( B(x, r) \), that \( X_{\text{rand}} \) has finite diameter and finite measure \( \mu(X_{\text{rand}}) \), that \( \text{Random} \) is a random variable uniformly distributed on \( X_{\text{rand}} \) with respect to \( \mu \).

Equation (2) provides a lower bound for the size of metric balls in \( X \) and implies that, for any ball \( B \) fully contained in \( X_{\text{rand}} \), the probability of sampling a point in \( B \) is at least \( C_\mu r^Q \). In practice, \( Q \) will correspond to the Hausdorff dimension of the metric space \( (X, d) \).

For the sake of analysis, we will use \( \text{Random} \) to generate an infinite sequence of identically-distributed samples in \( X \) (even if we only use the first few samples). This infinite sequence is itself a random variable, and is uniformly distributed in the infinite Cartesian product \( X_{\text{rand}}^\infty \) with respect to the probability measure \( \left( \frac{\mu}{\mu(X_{\text{rand}})} \right)^\infty \). We will say that an event happens almost surely (a.s.) if the measure of all sample sequences for which the event does not happen is 0.

#### 3.1.3 Curves and local connections: A curve is a continuous function \( \gamma : [0, T] \to X \). We will often conflate the function \( \gamma \) with its image in \( X \).

The length (or bounded variation) \( \ell(\gamma) \) of \( \gamma \) can be approximated by discretizing it as a sequence of finitely many points, and is defined as the supremum of such discretizations. That is,

\[
\ell(\gamma) := \sup \sum_i d(\gamma(t_i), \gamma(t_{i+1})) \tag{3}
\]

where the supremum is taken over all partitions \( 0 = t_0 < \ldots < t_n = T \). It suffices to consider any subclass of partitions with gap going to zero.

**Remark 1.** Under Assumption 1 the length of a curve \( \gamma : [0, T] \to X \) and its reversal \( \eta(t) = \gamma(T - t) \) satisfy \( \ell(\eta) \leq C_d(\gamma) \).

**Assumption 3.** (Local connections) We assume \( X \) is a length metric: that for any pair of points \( p, q \in X \) the distance \( d(p, q) \) can be recovered as \( d(p, q) = \inf \ell(\gamma) \), where the infimum is taken over all curves \( \gamma \) joining \( p \) to \( q \). Indeed, we assume that there is a constant \( C_E > 0 \) such that, given points \( p, q \in X \) satisfying \( d(p, q) < C_E \), the function \( \text{Extend}(p, q) \) provides a finite-length curve starting at \( p \) and terminating at \( q \). While we do not assume that \( \text{Extend}(p, q) \) provides a geodesic joining \( p \) and \( q \), we do assume that it is asymptotically geodesic:

\[
\lim_{d(p,q) \to 0} \frac{\ell(\text{Extend}(p, q))}{d(p, q)} = 1. \tag{4}
\]

In particular, there is a constant \( C_E' > 0 \) such that points \( p, q \) satisfying \( d(p, q) < C_E' \) are connected by curves satisfying \( \ell(\text{Extend}(p, q)) < 2d(x, y) \).

#### 3.1.4 Convergence and cost: We are ultimately interested in approximating cost-optimal curves using cost-near-optimal curves, without any regard for spatial convergence. However,
we will need two additional notions of proximity for continuous curves \( \gamma : [0, T_\gamma] \to X, \eta : [0, T_\eta] \to X \).

**Definition 6.** (Spatial proximity) Reparametrize \( \eta \) and \( \eta \), if necessary, so that both are parametrized at constant speed with time interval \([0, 1]\). The spatial proximity between \( \gamma \) and \( \eta \) is then given by the pointwise comparison

\[
d_s(\gamma, \eta) := \max_{t \in [0, 1]} d(\gamma(t), \eta(t)).
\]

**Definition 7.** (Length proximity) The length proximity between \( \gamma \) and \( \eta \) is given by \( d_\ell(\gamma, \eta) := |\ell(\gamma) - \ell(\eta)| \).

Clearly, length proximity does not imply spatial proximity since \( \gamma \) and \( \eta \) can be in separate regions of \( X \). Likewise, spatial proximity does not imply length proximity. For example, one can start with any finite-length curve and introduce high-frequency oscillations to increase its length, while remaining arbitrarily close to the original curve in terms of spatial proximity.

**Remark 2.** In the Euclidean context, when working with paths defined on the same interval one can consider the path \( \gamma - \eta \) to obtain \( d_s(\gamma, \eta) = |\gamma - \eta|_{\infty} \) and \( d_\ell(\gamma, \eta) \leq |\gamma - \eta|_{\infty} + |\ell(\gamma) - \ell(\eta)| \). Simultaneous proximity in both senses is then captured by the bounded variation (BV) norm \( |\gamma - \eta|_{BV} = |\gamma - \eta|_{\infty} + |\ell(\gamma - \eta)| \). Outside of Euclidean space, this interpretation is not natural since subtraction is not available. It can be recovered by using the Kuratowski embedding of \( X \) into the space of continuous functions on \( X \), but for us it is sufficient to analyze geometric proximity and length proximity separately.

**Assumption 4.** (Continuity of Cost) We assume that \( \text{Cost} \) is a functional assigning a number in \([0, \infty]\) to every finite-length curve in \( X \), in a way that is additive under concatenation. Furthermore, we assume that \( \text{Cost} \) is continuous under combined spatial and length convergence: if \( \gamma_i \to \gamma \) in the sense that \( d_s(\gamma_i, \gamma) + d_\ell(\gamma_i, \gamma) \to 0 \), then \( \text{Cost}(\gamma_i) \to \text{Cost}(\gamma) \).

The most natural choice for \( \text{Cost} \) is the length functional \( \ell \), but others can also be chosen. In \$5\$, we explore other options for \( \text{Cost} \) and explain why **Assumption 4 is usually hard to satisfy**.

It may also be possible to weaken Assumption 4 somewhat, since the assumption doesn’t fully capture the approximation processes described in \$2\$ and \$4.4\$.

### 3.1.5 Obstacles and Free Space:

**Assumption 5.** We assume that the unobstructed space \( X_{\text{free}} \subset X_{\text{rand}} \) is open and connected\(^3\). That is, at every point \( x \in X_{\text{free}} \) we can find a radius \( r > 0 \) such that \( x \notin B(x, r) \subset X_{\text{free}} \), and any pair of points \( x, x' \in X_{\text{free}} \) is connected by a finite-length curve \( \gamma \). Conversely, the obstacle set \( \text{Obst} := X_{\text{rand}} \setminus X_{\text{free}} \) is closed, i.e. contains its limit points.

Furthermore, we assume that we can effectively determine whether a point \( x \) given by \text{Random}, or a curve given by \text{Extend} is fully contained in \( X_{\text{free}} \).

\(^3\)Under our assumptions, it is straightforward to show that the relevant notions of connectivity (connected, path connected, and rectifiably connected) are equivalent.

Note that under Assumption 5, we cannot generally speak about optimal paths, since in some scenarios (e.g. when going around a corner) only nearly-optimal paths will exist. Since we are only interested in asymptotic optimality, this will not affect us. On the other hand, it provides us with wiggle room for convergence arguments and avoids some pathologies that come with assuming that \( X_{\text{free}} \) is closed, including undetectable tunnels with width 0 (cf. the robust optimality assumption in Karaman-Frazzoli [18]). We formalize this using the curve neighborhood lemma as follows. Here, the \( r \)-neighborhood of a set \( A \subset X \) is denoted by

\[ N_r(A) = \{ x \in X : d(x, x') < r \text{ for some } x' \in A \}. \]

**Lemma 8.** (Curve neighborhoods) Let \( \gamma \in X_{\text{free}} \). Then there is an \( \epsilon \) such that the \( \epsilon \)-neighborhood of \( \gamma \) remains in free space, i.e. \( N_\epsilon(\gamma) \subset X_{\text{free}} \).

**Proof.** Since \( \gamma \) is a continuous image of a closed interval, which is a compact set, it is itself compact. Consider the cover of \( \gamma \) by sets \( \gamma \times N_{1/n}(\text{Obst}) \), indexed by \( n \in \mathbb{N} \). The sets cover \( \gamma \) such that \( X_{\text{free}} \) is open, for each \( x \in \gamma \subset X_{\text{free}} \) there is an \( r > 0 \) such that \( B(x, r) \subset X_{\text{free}} \) and therefore \( B(x, r/2) \cap N_{1/n}(\text{Obst}) = \emptyset \) for sufficiently large \( n \). Since \( \gamma \) is compact and the sets are nested, for sufficiently large \( n \) we have \( \gamma \cap N_{1/n}(\text{Obst}) = \emptyset \), so that each point of \( \gamma \) is at least \( 1/n \)-far from \( \text{Obst} \), as desired.

#### 3.1.6 Proximity radius

While sPRM makes all connections in a fixed radius, the proximity radius for PRM* and RRT* is variable. For PRM* it is computed as \( r_n \) based on the number of samples used, while for the incremental RRT* algorithm it is computed as \( r_{\text{free}} \) based on the number of vertices discovered so far, cf. Theorem 14.

**Assumption 6.** (Proximity Radius) We assume that the proximity radius \( r_n > 0 \) satisfies \( \lim_{n \to \infty} r_n = 0 \) and \( \log n \geq C_r \left( \frac{\log n}{n} \right)^{1/Q} \), where \( Q \) is the dimension parameter in Assumption 2 and \( C_r \) satisfies

\[
C_r > (1 + C_d) \left( 1 + 1/Q \right) \frac{\mu(X_{\text{rand}})}{C_\mu} \left( \frac{\log n}{n} \right)^{1/d}.
\]

In the Euclidean case of dimension \( d \), Assumption 6 reduces to the requirement that \( r_n \to 0 \) while

\[
r_n \geq 2 \left( 1 + 1/d \right) \frac{\mu(X_{\text{rand}})}{C_\mu} \left( \frac{\log n}{n} \right)^{1/d}.
\]

This returns to the connection radius of [17], compared to the revised analysis of [36] where an exponent of \( \frac{1}{d+1} \) is introduced.

#### 3.1.7 Algorithm response:

**Assumption 7.** We will assume that, in addition to the above data, the algorithm is provided with a starting point \( p \in X_{\text{free}} \) and destination \( x \in X_{\text{free}} \). As in the case of sPRM (§2), we will allow an algorithm to simply return a graph \( G = (V, E) \) such that:

1) \( p \in V \subset X_{\text{free}} \),
2) if \((v_1, v_2) \in E\), then \( d(v_1, v_2) < C_E \) so that the curve \( \text{Extend}(v_1, v_2) \) is defined
3) if \((v_1, v_2) \in E\), then the curve \(\text{Extend}(v_1, v_2)\) lies in \(X_{\text{free}}\).

A path \(\gamma \subset X_{\text{free}}\) will be extracted from such a graph, if possible, by joining \(q\) to the nearest vertex \(v\) using \(\text{Extend}(v, q)\) and then using a graph-theoretic algorithm such as Dijkstra’s or \(\Lambda^*\) to connect \(p\) to \(v\) along the weighted graph \((V, E)\) with weights \(\text{Cost}(\text{Extend}(v_1, v_2))\) on the edges.

3.2 Algorithms

While sPRM provides an asymptotically optimal approach to the path-finding problem, its run time and space complexity of \(O(n^2)\) is prohibitive\(^4\). The PRM* algorithm reduces complexity by making \(r\) a decreasing function of \(n\). For single-query applications, RRT* furthermore takes an incremental approach by joining each new sample to its nearest neighbor, if possible, and maintaining a tree of optimal paths to a fixed point, including a rewiring of edges if possible, by joining.

Algorithm 2. (PRM*) Sample \(n\) points in \(X_{\text{free}}\) using Random, setting \(V\) to be this collection of points. For each pair of vertices \(v_1, v_2 \in V\), create an edge \((v_1, v_2)\) if \(f(\text{Extend}(v_1, v_2)) \leq r_n\) and \(\text{Extend}(v_1, v_2) \subset X_{\text{free}}\). Return the resulting graph.

Unlike PRM*, RRT* iteratively builds a weighted tree of optimal known paths to each vertex. Each edge \((v_1, v_2)\) of the tree is labeled with its cost \(\text{Cost}(\text{Extend}(v_1, v_2))\), and an auxiliary function \(\text{Cost}_*(v)\) provides the cumulative cost of reaching the vertex \(v\) from the tree’s root. Instead of using the rewiring radius \(r_n\), we use \(r_1 = r_{\#V}\), where we set \(i\) to equal the number of vertices in \(V\) at a given iteration of the algorithm.

Algorithm 3. (RRT*) Let \(V = \{p\}\), \(E = \emptyset\), and iterate until \(V\) contains \(n\) points, returning the resulting tree at the end.

At each iteration:
1) Sample a point \(x_{\text{new}} \in X_{\text{free}}\) using Random.
2) Make an initial connection:
   a) Let \(v_{\text{nearest}} = \arg\min_{v \in V} d(v, x_{\text{new}})\).
   b) If \(\text{Extend}(v_{\text{nearest}}, x_{\text{new}}) \subset X_{\text{free}}\), add the point \(x_{\text{new}}\) to \(V\) and add the edge \((v_{\text{nearest}}, x_{\text{new}})\) to \(E\). Otherwise, proceed to the next iteration.
3) Identify the locally-optimal parent:
   a) Let \(A\) be the set of vertices \(v \in V\) such that \(d(v, x_{\text{new}}) < r_{\#V}\) and \(\text{Extend}(v, x_{\text{new}}) \subset X_{\text{free}}\), excluding \(x_{\text{new}}\).
   b) If \(A\) is empty, proceed to Step 4.
   c) Letting \(x_{\text{best}} = \arg\min_{a \in A} (\text{Cost}_*(a) + \text{Cost}(\text{Extend}(a, x_{\text{new}})))\), remove the edge \((x_{\text{nearest}}, x_{\text{new}})\) from \(E\) and add the edge \((x_{\text{best}}, x_{\text{new}})\).
4) Identify the locally-optimal children:

Note that at sufficiently small scales, all pairwise connections are computed and stored.

a) Let \(A\) be the set of vertices \(v \in V\) such that \(d(x_{\text{new}}, v) < r_{\#V}\) and \(\text{Extend}(x_{\text{new}}, v) \subset X_{\text{free}}\), excluding \(x_{\text{new}}\).

b) For each \(a \in A\), if \(\text{Cost}_*(x_{\text{new}}) + \text{Cost}(\text{Extend}(x_{\text{new}}, a)) < \text{Cost}_*(a)\), remove the edge from \(a\) to its parent and add the edge \((x_{\text{new}}, a)\).

Note that the initial connection created in Step 2 may be at greater distance than \(r_{\#V}\). We will see that this step is critical in the first phase of the algorithm, but becomes unnecessary once \(X_{\text{free}}\) is sufficiently well-explored, see Theorem 14.

Algorithm 4. (RRT) The RRT algorithm is identical to RRT*, but excludes the optimization steps 3 and 4.

We will prove the convergence of a broader class of RRT-type algorithms, which include RRT, RRT*, and RRG (defined in the proof of Theorem 14).

Definition 9. (RRT-type algorithm) A path-finding algorithm \(\text{ALG}\) is an RRT-type algorithm if it:
1) iteratively builds a connected graph, initializing \(V = \{p\}\) and \(E = \emptyset\).
2) at each iteration, samples a point \(x_{\text{new}} \in X_{\text{free}}\) using Random,
3) adds \(x_{\text{new}}\) to the vertex set if the nearest existing vertex \(x_{\text{nearest}}\) satisfies \(\text{Extend}(x_{\text{nearest}}, x_{\text{new}}) \subset X_{\text{free}}\).

4 CONVERGENCE AND OPTIMALITY OF RRT* AND PRM*

4.1 Target Sequences

As in Figure 1, we will be interested in approximating a curve by reaching target regions along it.

Definition 10. (Target sequence) Given an arc-length parametrized curve \(\gamma\) and parameters \(\sigma, \rho\), let \(0 = t_1 < t_2 < \ldots < t_M = \ell(\gamma)\) be a sequence of times such that \(t_i - t_{i-1} = \sigma\) for \(i < M\) and \(t_M - t_{M-1} < \sigma\). The target sequence \(B(\sigma, \rho)\) is then the set of balls \((B(\gamma(t_i), \rho))_{i=1}^M\).

We say that a target sequence is connectable if any pair of points \(a, b\) in consecutive balls satisfies \(\text{Extend}(a, b) \subset X_{\text{free}}\).

Lemma 11. Let \(B(\rho, \sigma)\) be a target sequence along a finite-length curve \(\gamma\). Then:
(a) The ball centers are at most \(\sigma\) apart: \(d(\gamma(t_i), \gamma(t_{i+1})) \leq \sigma\).
(b) The number of balls satisfies \(M \leq \frac{\ell(\gamma)}{\sigma} + 1\).
(c) Given points \(a, b\) in consecutive balls, we have \(d(a, b) \leq C_\rho + \sigma + \rho\).
(d) For sufficiently small values of \(\rho\) and \(\sigma\), depending on the clearance of the curve \(\gamma\), the target sequence \(B(\sigma, \rho)\) is connectable.

Proof. The first and second claim are immediate from the definition of length, see Equation (3). The third follows from the triangle inequality and symmetry constant \(C_\rho\): if \(a \in B(\gamma(t_i), \rho)\) and \(b \in B(\gamma(t_{i+1}), \rho)\), then
\[
d(a, b) \leq d(a, \gamma(t_i)) + d(\gamma(t_i), \gamma(t_{i+1})) + d(\gamma(t_{i+1}), b) \\
\leq C_\rho + \sigma + \rho.
\]
We prove the fourth claim via two estimates on $l(\text{Extend}(a,b))$. For sufficiently small $\rho$ and $\sigma$, we have that $d(a,b) < C_d \rho + \sigma + \rho < C' E$, where $C' E$ is the length-doubling constant in Assumption 3, so that

$$l(\text{Extend}(a,b)) \leq 2d(a,b) \leq 2(C_d \rho + \sigma + \rho).$$

On the other hand, let $x$ be a point along $\text{Extend}(a,b)$ and assume by way of contradiction that $x \notin X_{\text{free}}$. Let $\epsilon > 0$ be given by Lemma 8, so that $N_{x}(\gamma) \subseteq X_{\text{free}}$. Then $d(a,x) \geq \epsilon$ and $d(b,x) \geq \epsilon$, and therefore $l(\text{Extend}(a,b)) \geq (1 + C_d)\epsilon$. For sufficiently small $\rho$ and $\sigma$, the two length estimates contradict, so $x$ cannot be outside of $X_{\text{free}}$. \hfill $\square$

### 4.2 Convergence of RRT-type algorithms

The convergence of RRT to feasible solutions for systems with finite controls\(^5\) was proven by Lavalle [23] and for Euclidean systems with continuous controls by Kuffner-Lavalle [22]; it extends immediately to RRT* [17]. We provide the following result for RRT-type systems under our broader assumptions.

**Theorem 12.** Every RRT-type algorithm $\text{ALG}$ is probabilistically complete.

**Proof.** We need to show that the graph $(V,E)$ returned by $\text{ALG}$ eventually contains a vertex $v$ such that $\text{Extend}(v,q) \subseteq X_{\text{free}}$.

By Assumption 5, there is a continuous curve $\gamma : [0,1] \rightarrow X_{\text{free}}$ joining $p$ to $q$. Let $B = (B(\gamma(t_i),\rho))_{i=1}^{M}$ be a connectable target sequence given by Lemma 11.

The first ball in the target sequence already contains the point $x_1 = p$, which is included in the vertex set maintained by $\text{ALG}$. The second ball has radius $\rho$ and therefore has a probability $\frac{C_d}{\mu(X_{\text{free}})} \epsilon^\rho$ of being sampled by $\text{Random}$. Almost surely, such a sample $x_2$ will be found within a finite number of iterations. Since $B$ is connectable, the algorithm $\text{ALG}$ will add $x_2$ to its graph and connect it by an edge either to $p_2$ or another appropriate point. Proceeding inductively, a point $x_i$ is eventually discovered in each ball of the target sequence, until a point $x_{M-1}$ is found that can be connected to $q$. \hfill $\square$

In particular, we conclude that an RRT-type algorithm explores every region of the free space arbitrarily-well within finite time:

**Corollary 13.** Let $\text{ALG}$ and $\delta > 0$. Let $V_n$ be the set of vertices produced by $\text{ALG}$ after $n$ iterations. For sufficiently large $n$, $N_{\delta}(V_n) \subseteq X_{\text{free}}$, almost surely.

**Proof.** Note first that $X_{\text{free}}$ has compact closure; indeed, by assumption 1, $X$ is proper, and by assumption 2, $X_{\text{rand}}$ has bounded diameter, so that $X_{\text{free}} \subseteq X_{\text{rand}}$ is also bounded and therefore has compact closure. Cover $X_{\text{free}}$ by balls $\{B(x,\delta/C_d) : x \in X_{\text{free}}\}$. By compactness, finitely many of these balls cover $X_{\text{free}}$. As in the proof of Theorem 12, we almost surely reach each of these balls and add them to the vertex set within finite time, as desired. \hfill $\square$

\(^5\)Critically, in this case, the function $\text{Extend}$ is unavailable. Instead, one finds the nearest vertex and then chooses one of finitely many controls that minimizes distance to $x_{\text{new}}$. Under these conditions, the set of feasible solutions is countable.

### 4.3 Reducing RRT* to PRM*

Unlike PRM*, RRT* adds each sample to the vertex set incrementally, and the sample is discarded if the connection to the nearest existing vertex is obstructed by the obstacle set. This, a priori, makes RRT* less effective than PRM*. One does not expect this to be a problem if the sample is drawn sufficiently far from the obstacle set and if there is an existing vertex close to the sample. Indeed, we showed in Corollary 13 that RRT-type algorithms obtain an arbitrarily dense vertex set within finite time. We now show that, once this phase is complete along a given curve, RRT* performs no worse than PRM*. This clarifies the long-term behavior of RRT* and simplifies the proof of its optimality by reducing it to the simpler analysis of PRM*.

**Theorem 14.** (Reduction to PRM*) Fix a proximity radius sequence $r_n$ such that PRM* is complete and asymptotically optimal with this proximity radius sequence. Then the same is true for RRT*.

**Proof.** Completeness of RRT* has already been proven, irrespective of the proximity radius, in Theorem 12. To prove asymptotic optimality, we fix a finite-length curve $\gamma$ connecting the starting point $p$ to the endpoint $q$ within $X_{\text{free}}$, and show that the optimal solutions produced by RRT* will, in the limit, be no worse than $\gamma$.

We consider a series of modifications of RRT*, all of which will be either RRT-type algorithms and therefore complete by Theorem 12, or equivalent to PRM* and therefore assumed to be complete. We will label objects created by an algorithm $\text{ALG}$ with a superscript, e.g. the vertex set $V_{\text{ALG}}^n$ produced by $\text{ALG}$ with parameter $n$. In particular, we will denote the best finite-length curve produced (almost surely for sufficiently large $n$) by $\text{ALG}^\gamma$.

We will prove that for all algorithms below (RRT* and $\text{ALG}1-5$), we a.s. have

$$\lim_{n \to \infty} \text{Cost}(\gamma_n^{\text{ALG}}) \leq \text{Cost}(\gamma).$$

First, let $\text{ALG}1$ be an algorithm that is identical to RRT*, except that it terminates after $n$ iterations rather than the RRT* stopping condition $\#V_n^{\text{RRT}*} = n$. Since less iterations are performed, the graphs produced by $\text{ALG}1$ and RRT* satisfy $G_n^{\text{ALG}1} \subseteq G_n^{\text{RRT}*}$, so that $\text{Cost}(\gamma_n^{\text{ALG}1}) \leq \text{Cost}(\gamma_n^{\text{RRT}*})$ for each $n$.

Second, let $\text{ALG}2$ be identical to $\text{ALG}1$, except that the proximity radius is set to $r_n$, rather than decreasing with the iteration number. This does not affect the vertex set but does restrict edge set from which the pieces of $\gamma$ are selected, so that $\text{Cost}(\gamma_n^{\text{ALG}2}) \leq \text{Cost}(\gamma_n^{\text{ALG}1})$ for each $n$.

Now, by Lemma 8, there is an $\epsilon > 0$ such that $N_{x}(\gamma) \subseteq X_{\text{free}}$. Let $\text{ALG}3$ be identical to $\text{ALG}2$, except that samples are discarded if they do not belong to the space $X_{\text{free}}$. This forces the tree to grow along $\gamma$, but allows edges to leave the restricted region $X_{\text{free}}$. Since less samples are used, we have $V_n^{\text{ALG}1} \subseteq V_n^{\text{ALG}2}$ and furthermore an inductive argument shows that at each iteration and for every vertex $v \in V_n^{\text{ALG}2}$, we have that the cost-to-root of $v$ satisfies $\text{Cost}_n^{\text{ALG}1}(v) \geq \text{Cost}_n^{\text{ALG}2}(v)$. In particular, $\text{Cost}_n^{\text{ALG}1}(\gamma_n) \geq \text{Cost}_n^{\text{ALG}2}(\gamma_n)$ for each $n$.
Fourth, let ALG4 be identical to ALG3, but with the optimization stage replaced by making all valid connections in the proximity radius. That is, an edge \((x_{\text{new}}, v)\) or \((v, x_{\text{new}})\) is created if and only if \(d(x_{\text{new}}, v) < r_n\) or \(d(v, x_{\text{new}}) < r_n\), respectively. The modification to the algorithm does not alter the vertex set, but does increase the edge set. However, the cost from \(p\) to each vertex \(v\) remains unchanged, since ALG3 at each stage selects the optimal connections out of those created by ALG4, maintaining a minimum-cost spanning tree. In particular, we have \(\text{Cost}_{\text{ALG4}}(\gamma_n) = \text{Cost}_{\text{ALG3}}(\gamma_n)\) for all \(n\).

There remain two ways in which ALG4 differs from PRM* running with reduced free space \(X_{\text{free}}^{\text{PRM*}} = X_{\beta/2}\). First, ALG4 creates an initial connection between \(x_{\text{new}}\) and the nearest vertex \(v\), disposing of any samples for which the curve \(\text{Extend}(v, x_{\text{new}})\) intersects the original obstacle set. Second, while vertices are restricted to lie in \(X_{\beta/2}\), the curve \(\text{Extend}(v, x_{\text{new}})\) may leave \(X_{\beta/2}\). We now show that the first difference disappears after finite time, and then make the further restriction to the algorithm that removes the second difference.

As in the proof of Lemma 11 let \(\epsilon' < \epsilon/2\) be sufficiently small so that for any \(x, y \in X_{\epsilon/2}\) with \(d(x, y) < \epsilon'\), we have \(\ell(\text{Extend}(x, y)) < \epsilon/2\), so that \(\text{Extend}(x, y) \in N_\epsilon(\gamma) \subset X_{\text{free}}\). By Corollary 13, the graph produced by ALG4 a.s. becomes \(\epsilon'\)-dense in \(X_{\epsilon/2}\) in some finite time \(m\), which can depend on the choice of \(\epsilon\) and the sequence of sampled points, but not on the proximity radius.

We now condition the analysis on how long the exploration phase takes, and create an algorithm that ignores the samples taken during the exploration, allowing us to focus on the optimization phase of ALG4.

Fix \(m > 0\) and consider only those sample sequences for which the conclusion of Corollary 13 becomes true for ALG4 on the \(m^{th}\) iteration. Observe that after \(m\) iterations, each newly sampled point \(x_{\text{new}}\) satisfying \(x_{\text{new}} \in X_{\epsilon/2}\) will be within distance \(\epsilon'\) of some vertex \(v\), and is added by ALG4 to \(V\), so we are very close to simply running PRM*. Let ALG5m be the PRM* algorithm with the following modifications: (1) the free space is reduced to \(X_{\text{free}}^{\text{ALG5m}} = X_{\epsilon/2}\), (2) the first \(m\) samples are discarded, and only the remaining \(n-m\) samples are used, (3) the connection radius remains \(r_n\). Under our restriction of the set of sample sequences, we have that the graph \(G_{\text{ALG5m}}\) is contained in \(G_{\text{ALG4}}\), so \(\text{Cost}(\gamma_n; G_{\text{ALG5m}}) \geq \text{Cost}(\gamma_n; G_{\text{ALG4}})\) for all \(n\). Observe that, under the sample sequence restriction, ALG5m is equivalent to simply running PRM* with free space \(X_{\text{free}}^{\text{ALG5m}} = X_{\epsilon/2}\), but (through bad luck) obtaining samples in the obstacle region for the first \(m\) iterations. This is an event with low but positive probability, so PRM* remains asymptotically optimal even under this restriction. Therefore, ALG5m is asymptotically optimal under the sampling assumption.

We thus have that (6) holds a.s. for each conditional algorithm ALG5m, and therefore for ALG4 under each of the sub-cases indexed by \(m\). Thus, (6) holds a.s. for ALG4, as well as ALG3, ALG2, ALG1, and RRT*.

### 4.4 Completeness and Optimality of PRM*

We now prove that PRM* is a.s. complete if and optimal, under the assumptions listed in §3.1. We prove here a slightly relaxed version of Theorem 3, see Theorem 14 and Remark 3:

**Theorem 15.** PRM* is complete, and is furthermore optimal if the proximity sequence \(r_n\) satisfies \(\lim n \rightarrow \infty r_n/\left(\log n\right)^{1/Q} = \infty\).

**Proof.** As in Theorem 15, it suffices to show that PRM* can approximate any finite-length curve \(\gamma\) in \(X_{\text{free}}\), given a random infinite sequence of samples, see Assumption 2.

For each \(n\), consider a target sequence \(B(\sigma_n, \sigma_n a_n)\), consisting of balls \(B_{n,1}, \ldots, B_{n,m}\), spaced \(\sigma_n\) apart with radius \(\sigma_n a_n\), where the parameter sequences \(a_n\) and \(\sigma_n\) will be determined later. Our first goal is to show that, for sufficiently large values of \(n\), all target balls \(B_{n,i}\) are sampled and connected to their neighbors, providing a curve that travels along \(\gamma\). We will then control oscillations in the approximating curve by taking \(a_n \rightarrow 0\).

According to Lemma 11(c), given samples \(a, b\) in consecutive balls, an edge is created between \(a\) and \(b\) by PRM* if \(r_n \geq (C_d \sigma_n a_n + \sigma_n a_n + \sigma_n)\). We therefore set \(\sigma_n = r_n/(C_d a_n + a_n + 1)\). By Lemma 11(d), the path \(\text{Extend}(a, b)\) will avoid obstacles once \(r_n\) is sufficiently small. Thus, for our first goal it suffices to prove that for sufficiently large \(n\) all of the balls \(B_{n,i}\) are in fact sampled.

We now estimate the probability of failure at stage \(n\). Each ball \(B_{n,i}\) satisfies, by Assumption 2,

\[
\mathbb{P}(\text{Random} \in B_{n,i}) \geq \frac{C_{\mu}}{\mu(X_{\text{rand}})} \left(\frac{a_n r_n}{C_d a_n + a_n + 1}\right)^Q.
\]

The probability of the ball \(B_{n,i}\) not being sampled in \(n\) iterations is therefore less than or equal to

\[
\left(1 - \frac{C_{\mu}}{\mu(X_{\text{rand}})} \left(\frac{a_n r_n}{C_d a_n + a_n + 1}\right)^Q\right)^n,
\]

which is less than or equal to

\[
\exp\left(-n \cdot \frac{C_{\mu}}{\mu(X_{\text{rand}})} \left(\frac{a_n r_n}{C_d a_n + a_n + 1}\right)^Q\right)\]

by the inequality \((1 + a)^b \leq e^{ab}\). Bounding the number of balls using Lemma 11(b), the probability of some ball not being sampled during \(n\) iterations is then bounded above (ignoring overlaps) by

\[
p_n = \left(1 + \frac{C_{\mu} a_n + a_n + 1}{r_n} \ell(\gamma)\right) \exp\left(-n \cdot \frac{C_{\mu}}{\mu(X_{\text{rand}})} \left(\frac{a_n r_n}{C_d a_n + a_n + 1}\right)^Q\right).
\]

According to the Borel-Cantelli lemma, if \(\sum\mathbb{P}(\text{Random} \in B_{n,i}) < \infty\), then we (almost certainly) fail to sample all the balls \(B_{n,i}\) for only finitely many values of \(n\), i.e. we almost certainly successfully sample all of the balls in the target sequence for sufficiently large \(n\).

Removing terms that do not affect summability, denoting the corresponding equivalence relation by \(\approx\), and setting \(b_n = n r_n / \left(\log n\right)^{1/ Q}\) we then have

\[
p_n = \frac{1}{r_n} \exp\left(-n \cdot \frac{C_{\mu} r_n}{\mu(X_{\text{rand}})} \left(\frac{a_n r_n}{C_d a_n + a_n + 1}\right)^Q\right)
\]

\[
\approx \frac{n}{b_n^{Q}} \exp\left(-n \cdot \frac{C_{\mu} r_n}{\mu(X_{\text{rand}})} \left(\frac{a_n r_n}{C_d a_n + a_n + 1}\right)^Q\right)
\]

\[
\approx \frac{n}{b_n^{Q}} \exp\left(-n \cdot \frac{C_{\mu} r_n}{\mu(X_{\text{rand}})} \left(\frac{a_n r_n}{C_d a_n + a_n + 1}\right)^Q\right).
\]
Disregarding the denominator $5$ in $Q \log n$, we are reduced to
\[ p_n \approx n^{1/\ell} \left( \frac{c_n}{\mu(X_{\text{rand}})} \right)^Q \left( \frac{a_n}{c_d a_n + \sigma_n} \right)^Q, \]
which is summable, by the p-test, if $1/Q - \frac{c_n}{\mu(X_{\text{rand}})} \left( \frac{a_n}{c_d a_n + \sigma_n} \right)^Q < -1$ which can be rewritten as
\[ 1/a_n < b_n \left( \frac{C_\mu}{\mu(X_{\text{rand}})} \frac{1}{1 + 1/Q} \right)^{1/Q} - 1 - C_d, \tag{7} \]
where $a_n$ is a yet-undetermined parameter providing the spacing of the balls $B(\sigma_n, a_n \rho_n)$. A positive choice of $a_n$ is possible as long as the right side of the inequality is positive, which is guaranteed by Assumption 6.

Once we make the corresponding choice of $a_n$, the sequence $p_n$ becomes summable, and we conclude using Borel-Cantelli that for sufficiently large $n$ the sampling sequence samples all of the balls $B_{n,i}$ and PRM* connects these into a curve $\gamma_n$. By Assumption 3, these connections are asymptotically optimal, so $\gamma_n$ pointwise converges to $\gamma$. Combining this with the definition of $\ell(\gamma)$, we furthermore have $\lim_{n \to \infty} \ell(\gamma_n) \geq \ell(\gamma)$.

To obtain the converse bound, we note first that repeated applications of the triangle inequality give that the length of $\gamma_n$ is bounded above by
\[
\ell(\gamma_n) \leq (M_n - 1)(\sigma_n + \rho_n + C_d \rho_n)
\leq \frac{\ell(\gamma)}{\sigma_n}(\sigma_n + a_n \sigma_n + C_d \sigma_n a_n)
= \ell(\gamma)(1 + a_n + C_d a_n)
\]
which converges to $\ell(\gamma)$ if we take $a_n \to 0$, which is allowed by (7) under our assumption that $b_n \to \infty$.

Lastly, since $\gamma_n$ converges to $\gamma$ in terms of both spatial proximity and length proximity, we conclude that $\text{Cost}(\gamma_n) \to \text{Cost}(\gamma)$, so that PRM* is asymptotically optimal for both distances and costs.

\textbf{Remark 3.} The assumption $b_n \to \infty$ can be removed by using Poissonization to show that an arbitrary large proportion of the balls $B_n$ are sampled arbitrarily close to the center, see [17]. Note that, in [17], the estimate $(\sqrt{2} \alpha + \beta(1 - \alpha))\ell(\gamma)$ for the BD norm of $\gamma - \gamma_n$ appears to be incorrect, but can be replaced (for a symmetric metric) with $\ell(\gamma_n) \leq M_n \theta q_n + 2(\alpha M_n) q_n + 2(1 - \alpha) M_n \beta q_n \leq \ell(\gamma) + 2(\alpha + \beta)\ell(\gamma)$.

\section{Non-Euclidean Path Planning}

We now illustrate the flexibility of our framework by demonstrating path-finding in non-Euclidean spaces. We first consider coupled one-dimensional systems, which are directly relevant to the case of fleets of vehicles considered in §6 and lead to path planning with respect to $\ell^p$ norms. Second, we provide path-finding in the fractal Sierpinski Gasket geometry, which requires us to use highly non-Euclidean distances, measures, and sampling methods.

\subsection{Coupled robotics systems}

Given a fleet of robots with state spaces $X_1, \ldots, X_n$, the joint state space for the fleet (ignoring, for the moment, interrobot collisions) is given by the Cartesian product $X_{\text{fleet}} = X_1 \times \cdots \times X_n$. The Random and Extend functions on each $X_i$ can be combined to obtain corresponding functions on $X_{\text{fleet}}$, but there is a choice that has to be made concerning the distance function. We now explore this choice in the approachable case where $X_{\text{fleet}}$ is two-dimensional. Such a system appears, for example, when controlling an arcade claw machine or 3D printer along independent rails.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig2.png}
\caption{Path planning in $\mathbb{R}^2$ with respect to the $\ell^1$, $\ell^2$, and $\ell^\infty$ metrics (top, middle, bottom, respectively) produces markedly different paths. Here, path planning occurs in the square $[0,3] \times [0,3]$ while avoiding the middle square $[1,2] \times [1,2]$. The tree produced by $RRT^*$ is shown, along with the best path from $(1.5,0.5)$ to $(1.5,2.5)$. Each model uses the same 1000 samples.}
\end{figure}

---

\footnote{This does not help summability, but one can show it doesn’t hurt it either.}
Other choices of $\ell^p$ distances, with $1 < p < \infty$, can also be used, with low values of $p$ favoring joint motion and high values of $p$ favoring individual motion. Because free-space optimal trajectories for $\ell^p$ distances with $1 < p < \infty$ are straight lines, in simple cases the long-term behavior of path planners will resemble the Euclidean solution Figure 3. However, in complex scenarios, different choices of $p$ will lead to different paths.

Fig. 3: Evolution of the RRT* tree for the $\ell^p$ metric with $p = 10$. The high value of $p$ causes $\ell^\infty$-like behavior at first, and eventual convergence to a streamlined solution.

Consider a robotic system consisting of two robots $R_x$ and $R_y$ moving along independent linear rails, whose positions are given by variables $x$ and $y$, respectively. The configuration space for the joint system is then $\mathbb{R}^2$, and there is some leeway on the choice of metric $d((x_0, y_0), (x_1, y_1))$. We consider three reasonable choices: the $\ell^1$ Manhattan metric given by $|x_0 - x_1| + |y_0 - y_1|$, the familiar $\ell^2$ Euclidean distance, and the $\ell^\infty$ maximum norm given by $\max(|x_0 - x_1|, |y_0 - y_1|)$. The $\ell^1$ metric can be interpreted as computing the total energy expenditure, while the $\ell^\infty$ metric computes the fastest transition time. The Euclidean metric does not have a natural interpretation, but provides a middle point between $\ell^1$ and $\ell^\infty$ among the $\ell^p$ norms given by $\sqrt[p]{|x_0 - x_1|^p + |y_0 - y_1|^p}$ for $1 \leq p < \infty$. More generally, in this context the variable $p$ can be thought of as a coupling parameter that calibrates the tradeoff between total path length minimization (when $p = 1$) and total time minimization (when $p = \infty$).

Path planning with respect to the $\ell^1$, $\ell^2$, and $\ell^\infty$ metrics produces markedly different results (Figure 2). The $\ell^1$ metric imposes no penalty for traveling along the coordinates one at a time since the corresponding cost is additive, and therefore results in the coordinates mostly changing in sequence. For multi-robot systems, this corresponds to the $\ell^1$ metric inadvertently incentivizing excessive sequential vehicle motion. Conversely, the $\ell^\infty$ metric inadvertently incentivizes excessive joint motion, since the slower robotic system can move without imposing an additional total cost. This results in unnecessary motion, as seen in Figure 2, where the optimal path moves further left along the $x$-axis than is strictly necessary, since such motion is not penalized by the $\ell^\infty$ metric. The Euclidean $\ell^2$ metric provides a middle ground between the two options: both excessive waiting and excessive joint motion are penalized.

Lastly, we demonstrate the fact that the choice of an Extend function that is compatible with Cost is critical. Working in $\mathbb{R}^2$ with the $\ell^1$ metric, we can choose to work with an Extend function that provided Manhattan paths, which adjust each coordinate one at a time, as shown in Figure 4. This would recover asymptotically-optimal $\ell^1$ paths. However, Euclidean shortest paths could not be recovered by using a Euclidean Cost functional, since diagonal paths would never be generated by Extend. This highlights the necessity of Assumption 4 requiring the continuity of the Cost functional.

Fig. 4: An implementation of RRT* using a Manhattan Extend function is asymptotically-optimal with respect to the $\ell^1$ cost (shown), but not with respect to the Euclidean cost, since it produces staircase paths whose Euclidean length corresponds to the Manhattan distance.

5.2 Sierpinski gasket

Our assumptions, while motivated by vehicle path planning, apply to a wide range of scenarios. We illustrate this by implementing RRT* in the Sierpinski gasket $\mathcal{S}$, a self-similar fractal set in $\mathbb{R}^2$ of Hausdorff dimension $Q = \log(4)/\log(3)$. We are not aware of any immediate applications to path planning inside $\mathcal{S}$, beyond providing a benchmark for our assumptions.

In order to apply RRT* to path planning on $\mathcal{S}$, we must specify a way to sample points, a way to make optimal connections. We start by describing the Sierpinski gasket, making use of the notation and results of [6]. A common description of $\mathcal{S}$ is as follows: start with a solid equilateral triangle, remove the central triangle defined by the midpoints of the sides, and repeat the process iteratively with the remaining three triangles. To specify a point $p \in \mathcal{S}$, we can specify a sequence of triangles that it resides in, e.g. the bottom-left corner of the top triangle is given by $p = (0, 1, 1, \ldots)$ where the 0 indicates that it is in the top triangle, and the 1s indicate that under further subdivisions
A random point in \( S \) can be selected by specifying an infinite digit sequence, or, equivalently by selecting a random point in the interval \([0, 1]\) and extracting its base-3 digits. The resulting probability distribution is uniformly distributed with respect to the \( Q \)-dimensional Hausdorff measure on \( S \), with respect to the natural metric we now describe.

Distances in \( S \) are computed abstractly by taking \( d(p, q) = \inf \ell(\gamma) \) where the infimum is taken over all finite-length curves \( \gamma \in S \subset \mathbb{R}^2 \) joining \( p \) and \( q \). One can construct optimal curves explicitly as follows. Start by zooming in and orienting \( S \) such that \( p \) is in the left triangle and \( q \) is in the right triangle, i.e. after normalization we have \( p = (1, p_2, p_3, \ldots) \) and \( q = (2, q_2, q_3, \ldots) \). While there are, in some cases, as many as five optimal paths between \( p \) and \( q \), we construct a specific one as follows. One possibility is that an optimal path passes through the mid-point indexed by \((1, \overline{2}) \) to \((2, \overline{3}) \) and \((1, \overline{3}) \) as before by replacing digits, then connect \((1, \overline{0}) \) to \((2, \overline{0}) \), and then interpolate towards \( q \) as before. In our implementation, we compute both of these paths and return the shorter one.

Because \( S \) is a compact self-similar subset of \( \mathbb{R}^2 \) with the induced metric and a symmetry-invariant sampling method, the assumptions for the convergence of RRT* are immediate to verify. The resulting search tree and near-optimal paths are shown in Figure 5.

6 Car Fleet Trajectory Planning

6.1 Model Description

6.1.1 Non-holonomic robotics: A differential constraint on robotic motion provides an infinitesimal restriction on the robot’s motion. That is, at every pose \( p \), it specifies a subspace \( H_pM \subset T_pM \) of the tangent space to the configuration space such that the robot can affect motion in along the vectors of \( H_pM \). While differential constraints can serve to restrict motion to a smaller-dimensional subspace, combining a sequence of controls can unlock additional motion directions (as seen in Figure 6), allowing motion throughout the configuration space. The two cases can be distinguished using Lie brackets of control vector fields: the Frobenius Theorem detects global restrictions and the Rashevsky-Chow Theorem detects global accessibility. The latter case is exemplified by the non-holonomic motion of car-like vehicles, which are modeled using the Roto-Translation Space.

Fig. 5: RRT* search tree (blue) and near-optimal paths (red) in the Sierpinski gasket \( S \), after 200 iterations. The top figure shows unobstructed path-finding. For the bottom figure, the grey region indexed by \((2, 0)\) was obstructed.

Fig. 6: Sub-Finsler geometry encodes non-holonomic controls by specifying a linear subspace of controllable directions at each point.

The extended Roto-Translation Space \( \overrightarrow{RT} \) encodes rigid motion in the plane, using coordinates \((x, y)\) and an angle coordinate \( \theta \). More precisely, \( \overrightarrow{RT} \) is the space \( \mathbb{R}^3 \) with coordinates \((x, y, \theta)\), and control vector fields \( \xi(x, y, \theta) = (\cos(\theta), \sin(\theta), 0) \) and \( \eta(x, y, \theta) = (0, 0, 1) \). One interprets \( \xi \) as motion in the forward direction, and \( \eta \) as a rotation. A trajectory in \( \overrightarrow{RT} \) is then a controllable trajectory (also known as permissible or horizontal) if \((x'(t), y'(t), \theta'(t)) \in \text{span}(\xi, \eta)\).

Taking the Lie bracket \([\xi, \eta]\) produces the sideways motion vector field \( \zeta = (-\sin(\theta), \cos(\theta), 0) \), which is not a linear combination of \( \xi \) and \( \eta \). While this means that \( \zeta \) is not a directly-controllable direction, the equation \([\xi, \eta] = \zeta\) means that motion in the \( \zeta \) direction can be obtained by...
composing motions in the $\xi$ and $\eta$ directions, as is familiar from parallel parking.

It is common to furthermore endow $\tilde{\mathcal{RT}}$ with the sub-Riemannian metric by using an inner product on the control subspace which makes $\xi$ and $\eta$ orthonormal directions. This allows computing the speed of a controllable path, and therefore its sub-Riemannian length. The sub-Riemannian distance on $\tilde{\mathcal{RT}}$ is then given by the infimum of lengths of controllable paths joining two given points. We will instead be interested in a sub-Finsler metric corresponding to the Reeds-Shepp vehicle model.

6.1.2 Reeds-Shepp car: The Reeds-Shepp car model is the standard representation of a vehicle for trajectory planning problems in low-speed environments. It represents a vehicle as a Cartesian product of individual vehicle configurations, not the direction of travel. Motion is modeled using the equations

$$\begin{align*}
\dot{x}_i &= v_i \cos \theta_i \\
\dot{y}_i &= v_i \sin \theta_i \\
\dot{\theta}_i &= u v_i \leq \left| \frac{v_i}{\rho_i} \right|
\end{align*}$$

where $\rho_i$ is the turning radius of vehicle $i$. The Reeds-Shepp model allows $v = \pm 1$, and is better adapted to low-speed motion planning tasks such as parallel parking. Figure 7 shows the trajectory that the Reeds-Shepp car can travel by taking advantage of the forward-backward motion to travel from the origin O to the destination C. According to Reeds-Shepp theory [34, 38], the trajectory traveling through OABC is the optimal path, whereas the other well-known method, Dubin’s car [10], results in OADEC trajectory as the optimal trajectory because of the constraint on the backward motion. In Figure 7 the arrows show the orientation of the vehicle, not the direction of travel.

6.1.3 Fleet of Reeds-Shepp cars: We represent a fleet of $n$ cars as a Cartesian product of individual vehicle configurations. That is, a fleet is represented by $3n$ coordinates $(x_1, y_1, \theta_1), \ldots, (x_n, y_n, \theta_n)$, with each car satisfying the motion equations in 6.1.2.

Critically, we interpret all of the cars as part of a single system. In particular, we will model the motion of the fleet as a single $3n$-dimensional choreography function

$$P(t) = (x_1(t), y_1(t), \theta_1(t), \ldots, x_n(t), y_n(t), \theta_n(t))$$

(10)

that can be decomposed, via projection, into trajectories

$$P_i(t) = (x_i(t), y_i(t), \theta_i(t))$$

(11)

for each car $i$.

Analogously to the Reeds-Shepp theory for a single vehicle, the free-space fastest-trajectory problem asks for a choreography $P(t)$ starting at a specified time $t = 0$ and ending at an unknown time $t = T$ which satisfies the starting and stopping conditions

$$P(0) = (x_1^0, y_1^0, \theta_1^0, \ldots, x_n^0, y_n^0, \theta_n^0),$$

$$P(T) = (x_1^1, y_1^1, \theta_1^1, \ldots, x_n^1, y_n^1, \theta_n^1),$$

(12)

(13)

and furthermore minimizes the total travel time $T$ subject to the model’s constraints.

Under the Reeds-Shepp model, the velocity of each vehicle is constrained to $|v_i| \leq 1$. This can be expressed as the constraint $\|V\|_\infty \leq 1$, where $V = (v_1, \ldots, v_n)$ and

$$\|v_1, \ldots, v_n\|_p = \left( \sum_{i=1}^{n} |v_i|^p \right)^{1/p} \quad 1 \leq p < \infty, \quad p = \infty$$

(14)

(satisfying, in particular, $\lim_{p \to \infty} \|V\|_p = \|V\|_\infty$).

6.1.4 Optimal Choreographies: We work with the minimum-travel-time distance on the space $\mathbb{R}^{3n}$ of vehicle configurations, given by

$$d_p(a, b) = \min_{P} \text{Time}(P),$$

(15)

where $\text{Time}(P)$ is the time necessary to complete the choreography $P$ and the minimum is taken over all choreographies $P$ starting at $a = (a_1, \ldots, a_n) \in \mathbb{R}^{3n}$ and ending at $b = (b_1, \ldots, b_n) \in \mathbb{R}^{3n}$, subject to the model’s constraints. Note that here vehicles are allowed to move through each other to reach their destinations.

Shortest curves between $a$ and $b$, for use by the Extend function of RRT*, can be identified from Reeds-Shepp theory [34] (implemented in [20]) as follows. Let $p^{RS}_i(t)$ be the Reeds-Shepp geodesic for the $i^{th}$ vehicle starting at $a_i$ at $t = 0$ and arriving at $b_i$ at some time $T_i$, traveling at the maximum allowed velocity $|v_i| = 1$. An optimal choreography $P(t)$ between $a$ and $b$, with time parameter $t \in [0, \max_{i=1}^{n} T_i]$ is given by:

$$p_i(t) = p^{RS}_i \left( t \frac{T_i}{\|T_i\|_p} \right).$$

(16)

It is easy to see that the vehicles arrive at their destinations simultaneously, and $\|V\|_p = 1$. The proof of optimality relies on showing that (15) is equivalent to

$$d_p(a, b) = \|T_1, \ldots, T_n\|_p$$

(17)

by showing that (15) is the trajectory metric associated to the product-type metric (17), for which (16) provides optimal geodesics.

6.1.5 Modeling Obstructions: The algorithm avoids three types of collisions: i) inter-vehicle collisions, ii) collisions by individual vehicles with environmental obstacles, and iii) paths that allow an individual vehicle to leave the model’s region. For collision-avoidance purposes, we will model the vehicles as a disk of radius $r$ centered at the coordinates $(x_i, y_i)$.
Inter-vehicle collisions are prohibited by restricting the distance between vehicles:
\[
\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \geq 2r, \text{ for } i \neq j. \tag{18}
\]

Environmental obstacles will be modeled as a collection of solid polygons \( O_1, \ldots, O_m \) in the plane, and we will restrict the range of motion to a \( W \times W \) region:
\[
0 \leq x_i, y_i \leq W.
\]

Because Reeds-Shepp geodesics are not linear, optimal free-space vehicle trajectories between points in this region may take the vehicle outside the region, and we must impose the spatial constraint explicitly.

For purposes of the implementation, type 2 and type 3 constraints are replaced with a list of edges \( E_j \) that may not be touched by any vehicle:
\[
d((x_i, y_i), E_j) \geq r, \tag{19}
\]
computed as the minimum of the distance to the endpoints of \( E_j \) and the distance to the perpendicular projection of \((x_i, y_i)\) to the line containing \( E_j \).

### 6.2 Simulation Results

In this section, we demonstrate the application of the method presented in §3 to the trajectory planning for a fleet of vehicles. Here, we assume that trajectory planning for a large number of vehicles is being achieved using decoupled motion planning and conflicts in trajectories of a small number of vehicles are resolved locally by increasing the dimension of the problem in the relevant region. Such results can be obtained via sub-dimensional expansion or a similar method, where the vehicles enter a box with specific poses and need to leave with desired poses while avoiding each other and other obstacles in the box. We assume that the problem can not be simplified any further by reassigning the tasks or similar methods, and joint planning is needed.

We developed our trajectory planning algorithm in C++, and in implementing Reeds-Shepp geodesics and the resulting Reeds-Shepp distance on the state space of a single vehicle, we made use of the Open Motion Planning Library (OMPL) [20]. The geometry of each car was approximated using two intersecting circles. In the following, we show two-car trajectory planning subject to different choices of the norm, and a three-car trajectory planning using \( \ell^2 \)-norm.

#### 6.2.1 Two-car trajectory planning:

In this section, we show the results of two-car trajectory planning and the effect of choosing different norms for the cost function.

In the scenario shown in Figure 8, two cars are navigating a small region with obstacles: the blue car is moving from the bottom-right corner of the region, with starting pose \([70, 20, \pi] \) to the upper-left corner of the region, with ending pose \([30, 80, 0] \); while the red car is positioned in the middle of the region with pose \([50, 57.5, \pi/2] \), and needs to return to the same pose.

We perform path planning with respect to the \( \ell^1 \), \( \ell^2 \), and \( \ell^\infty \) norms to define distances on the joint state space for the two vehicles. The three planning tasks result in different choreographies, shown in (a,c,e) of Figure 8, each of which is near-optimal for the respective metric.

The distinction between the three choreographies is apparent in the speed profiles\(^7\) seen in (b,d,f) in Figure 8. Observe that:

- the \( \ell^\infty \) planner provides the fastest choreography (12.3s), since the \( \ell^\infty \) cost of the choreography corresponds exactly to the time required to traverse it; while the \( \ell^2 \) and \( \ell^1 \) choreographies take longer to traverse (12.9s and 15s, respectively).
- the \( \ell^1 \) planner provides the smallest total amount of

\(^7\)Note that the choreographies are parametrized such that the faster car is always moving at the maximum allowed speed, which does not correspond to a unit-speed parametrization in the \( \ell^p \) metric.
motion (152.4 cm total), since the $\ell^1$ cost computes the total travel distance for both vehicles; while the $\ell^2$ and $\ell^\infty$ choreographies have longer total travel distance (158.5 cm and 158.6 cm, respectively).

- the speeds of the two cars are highly matched in the $\ell^\infty$ case, which prioritizes joint motion, and is highly varying in the $\ell^1$ case, which prioritizes individual motion; with $\ell^2$ motion providing a middle ground.

The simulations thus illustrate the fact that the $p$ value in the $\ell^p$ metric provides a way to choose the extent to calibrate the planner to the designer’s preference in the tradeoff between fastest-motion or shortest-total-motion choreographies.

6.2.2 Three-car trajectory planning: Figure 9 shows the simulation results for three Reeds-Shepp cars maneuvering in an environment with polygonal obstacles with an $\ell^2$ cost function, which provides a compromise between the shortest-time and shortest-total-length options for choreography planning.

As in the previous example, we assume that we are working inside a bubble space of a higher-order planner using a sub-dimensional expansion approach, and that we need to identify an optimal choreography for the 3-car fleet with the specified starting and finishing poses.

In this example, the blue and magenta cars move from initial poses of $[70, 20, \pi]$ and $[80, 80, 0]$, as shown in Figure 9a, to their destinations at $[30, 80, 0]$ and $[70, 20, \pi]$, respectively, while the red gatekeeper car returns to its original pose $[50, 57.5, \pi/2]$ as shown in Figure 9d. Static environmental obstacles are shown as gray boxes.

We show two important intermediate poses for the fleet. First, in Figure 9b, we see that that the blue and magenta cars have change direction and started to move toward the center of the region, while the red car has started to make space for their paths. Next, in Figure 9c the red car is out of the way of the blue and red cars, and the blue and magenta cars move through the opened passage at the same time. Finally, the blue and magenta cars reach their destinations in Figure 9d, and the red car returns to its initial pose at the center of the region.

7 Summary and Conclusions

We provided a new, broader framework for path planning using sampling-based algorithms including RRT* and RPM*, and proved the probabilistic completeness and asymptotic optimality of these algorithms under our broader assumptions on the underlying geometry and the auxiliary functions. We extend and random providing free-space paths and random samples, respectively. The result recovered the original convergence guarantees of [17] in the broader framework, improving on the updated guarantees of [36].

We first demonstrated the algorithms in non-Euclidean metric spaces, the $\ell^p$ planar geometries, and the fractal Sierpinski gasket. We then applied them to the task of choreography planning for multiple car-like robots, by combining several Reeds-Shepp vehicle state spaces into a single metric space using an $\ell^p$ distance. We showed that the resulting choreographies depended on the parameter $p$, providing either lowest-total-length choreographies for small values of $p$ or fastest-time choreographies for high values of $p$, with $p = 2$ providing a convenient intermediate parameter. We demonstrated path planning for a 3-car scenario, which can be used as a localized planner inside a bubble space for a larger fleet of vehicles.

The next step in the development of the method is to apply path-shortening algorithms to further improve the final choreography and to accelerate the runtime (e.g., using non-Euclidean nearest-neighbor algorithms).

8 Acknowledgments

We thank Heath Camphire, Susan Tabulsi, and Maria-Pia Younger for their contributions in the early stages of the project. We thank Mason Experimental Geometry Lab for facilitating the research. This work was supported in part by a George Mason University COS-VSE Seed Grant.

References

[1] Aliakbar Agha-Mohammadi, Suman Chakravorty, and Nancy M. Amato. Sampling-based nonholonomic motion planning in belief space via Dynamic Feedback Linearization-based FIRM. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 4433–4440, Vilamoura-Algarve, Portugal, October 2012.

[2] Andrei Ardentov, Gil Bor, Enrico Le Donne, Richard Montgomery, and Yuri Sachkov. Bicycle paths, elasticae and sub-riemannian geometry. Nonlinearity, 34(7):4661–4683, Jun 2021.

[3] Andrey A. Ardentov, Yuri L. Karavaev, and Kirill S. Yefremov. Euler elasticae for optimal control of the motion of mobile wheeled robots: the problem of experimental realization. Regul. Chaotic Dyn., 24(3):312–328, 2019.
