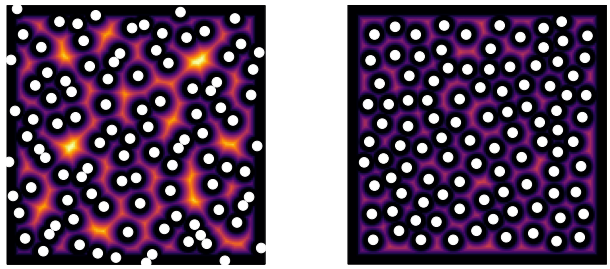
Dispertio: Optimal Sampling For Safe Deterministic Motion Planning

Luigi Palmieri, Leonard Bruns, Michael Meurer, and Kai O. Arras

***Abstract*—A key challenge in robotics is the efficient generation of optimal robot motion with safety guarantees in cluttered environments. Recently, deterministic optimal sampling-based motion planners have been shown to achieve good performance towards this end, in particular in terms of planning efficiency, final solution cost, quality guarantees as well as non-probabilistic completeness. Yet their application is still limited to relatively simple systems (i.e., linear, holonomic, Euclidean state spaces). In this work, we extend this technique to the class of symmetric and optimal driftless systems by presenting Dispertio, an offline dispersion optimization technique for computing sampling sets, aware of differential constraints, for sampling-based robot motion planning. We prove that the approach, when combined with PRM*, is deterministically complete and retains asymptotic optimality. Furthermore, in our experiments we show that the proposed deterministic sampling technique outperforms several baselines and alternative methods in terms of planning efficiency and solution cost.**

***Index Terms*—Motion and path planning, nonholonomic motion planning, reactive and sensor-based planning.**



(a) Halton (b) Dispertio

Fig. 1. Comparison of the coverage between l_2 low dispersion Halton samples and our optimized samples for a 2D Euclidean case. Bright color highlights uncovered areas. Our approach achieves better coverage than the baseline.

I. Introduction

MOTION planning is key to intelligent robot behavior. For motion planning in safety-critical applications, where self-driving cars, social or collaborative robots operate amidst and work with humans, safety guarantees, explainability and deterministic performance bounds are of particular interest. In the past, many motion planning approaches have been introduced to improve planning efficiency, path quality and applicability across classes of robotic systems. Probabilistic sampling-based motion planners [1]–[3] and their optimal variants [4], [5] have shown to outperform combinatorial approaches [6], especially for high-dimensional systems with complex differential constraints in cluttered environments. Sampling-based planners explore the configuration space by sampling states and connecting them to the roadmap, or tree, which keeps track of the state space connectivity. Typically samples are drawn from a uniform distribution over the state space by an independent and identically distributed (i.i.d.) random variable. Biasing techniques towards the goal region or promising areas of the configuration space may be used if available [7], [8]. The randomness of the samples set ensures good exploration of the configuration space, but comes at the expense of stochastic results which may strongly vary for each planning query in terms of planning efficiency and path quality. This stochasticity makes the formal verification and validation of such algorithms, needed for safety-critical applications, difficult to obtain.

To address this issue, several authors [9], [10] propose to use deterministic sets (or sequences). Contrarily to using i.i.d. random variables, this technique allows to achieve deterministic planning behaviors while still getting on par or even better performance. Moreover, as described also in [9], [10], deterministic sampling allows an easier certification process for the planners (e.g., in terms of final cost, clearance from the obstacles). Particularly, as we will see also in our case, those approaches have been shown to be complete (i.e., to find a solution) for planning queries for which a solution with certain clearance exists. However, current approaches limit their applicability to Euclidean spaces [9], systems with linear affine dynamics [10] and specific driftless ones [11].

With the goal to further enhance the usage of deterministic sampling to symmetric and optimal driftless systems, in this work we present *Dispertio*, an optimization-based approach to deterministic sampling. The method computes a sampling set which minimizes the actual dispersion of the samples. To compute the dispersion metric, we need access to a steer function [12], [13] that can compute an optimal path connecting two states. We focus our attention on uninformed batch-based

Manuscript received September 6, 2019; accepted November 11, 2019. Date of publication December 9, 2019; date of current version December 24, 2019. This letter was recommended for publication by Associate Editor J. O'Kane and Editor D. Song upon evaluation of the reviewers' comments. This work was supported by the European Union's Horizon 2020 Research and Innovation Programme under Grant 732737 (ILIAD). *(Luigi Palmieri and Leonard Bruns contributed equally to this work.) (Corresponding author: Luigi Palmieri.)*

L. Palmieri and K. O. Arras are with Corporate Research, Robert Bosch GmbH, 70049 Stuttgart, Germany (e-mail: luigi.palmieri@de.bosch.com; kaioliver.arras@de.bosch.com).

L. Bruns is with the KTH Royal Institute of Technology, RWTH Aachen University, 16931 Stockholm, Sweden (e-mail: leonardb@kth.se).

M. Meurer is with German Aerospace Center (DLR), RWTH Aachen University, 82234 Wessling, Germany (e-mail: michael.meurer@dlr.de).

Digital Object Identifier 10.1109/LRA.2019.2958525

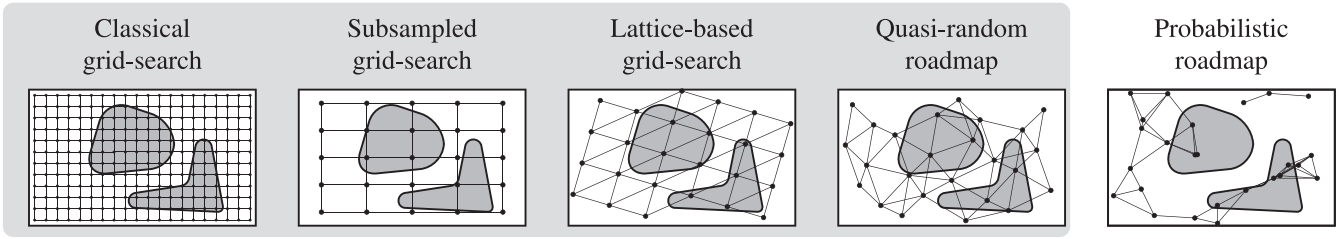


Fig. 2. Range of possible sampling and roadmap types as introduced by LaValle [9]. The highlighted ones are deterministic.

algorithms (e.g., PRM* [4]) where the set of samples can be precomputed offline. We prove that the approach, when combined with PRM*, is deterministically complete and retains asymptotic optimality. Furthermore, we systematically compare our approach to the existing baselines [10], [11]. The experiments demonstrate that *Dispertio* outperforms the baselines in terms of planning efficiency and overall final path quality.

II. Related Work

LaValle *et al.* [9] highlight the relationship between grid-based and probabilistic planning, see Fig. 2. The authors advocate that grid-based planners and probabilistic sampling-based planners all belong to the same class of sampling-based algorithms and are extremes of a broad spectrum of sampling strategies, ranging from deterministic to highly stochastic techniques. They highlight the benefits of deterministic sampling sets (or sequences) such as grids [14], [15], lattices [16], or Halton and Sukharev sequences [17], [18]. In particular, LaValle *et al.* [9], [19] show that *dispersion* (see Sec. III for the definition) is the deciding metric when it comes to resolution complete path planning. The reason is that dispersion provides lower bounds on the coverage of the space. The authors prove that low dispersion sampling, e.g. Halton and Sukharev sequences [17], [18], provides deterministic completeness guarantees on finding feasible paths, which i.i.d. sampling can only probabilistically provide, i.e., the planner will find a solution with a probability of 1 as the number of samples goes to infinity. Our approach follows the ideas presented by [9] and extends their results to motion planning with differential constraints. Khaksar *et al.* [20] propose a sampling-rejection mechanism to compute a set of PRM* low-dispersion samples. Differently from this approach, *Dispertio* computes the sampling set off-line and considers symmetric driftless systems with complex kinematic constraints.

While the authors in [9], [20] focus on feasibility in deterministic sampling-based motion planning, Janson *et al.* [10] extend the approach to address optimality. The authors show that with a particular choice of low dispersion sampling (l_2 dispersion of order $O(n^{-1/D})$, e.g., Halton sequence, with D being the state space dimension of the considered system), optimal sampling-based planners (i.e., PRM* [4], [21]) can use a lower connection radius compared to i.i.d. sampling thus requiring a lower computational complexity, i.e., $r_n \in \omega(n^{-1/D})$. Moreover, they show that the cost or suboptimality of the returned solution can be bounded, based on the dispersion. The latter work limits its applicability to Euclidean spaces and to systems having linear affine dynamics. In comparison our method can be applied also to symmetric and optimal driftless systems with differential constraints.

Poccia [11] proposes an approach for generating a set of deterministic samples for nonholonomic systems. The approach needs an explicit and careful analysis of the system equations to come up with a sampling scheme. Differently, our approach provides an algorithm that only needs the availability of an optimal *steer* function, a common assumption for optimal sampling-based planning [4], [21].

Unlike state-lattice approaches [16], which can be seen as part of the class of deterministic sampling-based planners, our approach does not rely on a regular grid or a set of pre-defined motion primitives. Instead, it optimizes the position of the samples based on the dispersion metric that accounts for the differential constraints of the system.

III. Our Approach

In this section, we introduce the problem formulation and the novel dispersion definition. We will then describe our algorithm and analyze its properties.

A. Problem Definition

Let $\mathcal{X} \subset \mathbb{R}^D$ be a manifold defining a configuration space, $\mathcal{U} \subset \mathbb{R}^M$ the symmetric control space, $\mathcal{X}_{\text{obs}} \subset \mathcal{X}$ the obstacle space and $\mathcal{X}_{\text{free}} = \mathcal{X} \setminus \mathcal{X}_{\text{obs}}$ the free space. A driftless control-affine system can be described by a differential equation as

$$\dot{\boldsymbol{x}}(t) = \sum_{j=1}^{M} g_j(\boldsymbol{x}(t))\mathbf{u}(t) \tag{1}$$

where $\boldsymbol{x}(t) \in \mathcal{X}$, $\mathbf{u}(t) \in \mathcal{U}$, for all t, and $g_1, \ldots, g_M$ being the system vector fields on $\mathcal{X}$. For the remainder of the letter we will focus on symmetric systems for which an optimal steer function exists.

Let γ denote a planning query, defined by its initial state $\boldsymbol{x}_{\text{start}} \in \mathcal{X}$ and goal state $\boldsymbol{x}_{\text{goal}} \in \mathcal{X}$. We define the set of all possible solution paths for a given query γ as Σ_γ, with $\sigma \in \Sigma_\gamma : [0,1] \to \mathcal{X}_{\text{free}}$ being one of the possible solution paths such that $\sigma(0) = \boldsymbol{x}_{\text{start}}$ and $\sigma(1) = \boldsymbol{x}_{\text{goal}}$. The arc-length of a path σ is defined by $l(\sigma) = \int_0^1 ||\dot{\sigma}(t)||_2 \, dt$. The arc-length induces a *sub-Riemannian* distance dist on $\mathcal{X}$: $\text{dist}(\boldsymbol{x}, \boldsymbol{z}) = \inf_\sigma l(\sigma)$, i.e., the length of the optimal path connecting $\boldsymbol{x}$ to $\boldsymbol{z}$, which due to our assumptions is also symmetric. Let σ^* denote the set of all points along a path σ. The dist-clearance of a path σ is defined

as

$$\delta_{\text{dist}}(\sigma) = \sup\left\{r \in \mathbb{R} \mid \mathcal{R}_{\text{dist}}(\boldsymbol{x}, r) \subseteq \mathcal{X}_{\text{free}}\ \forall \boldsymbol{x} \in \sigma^*\right\} \quad (2)$$

where $\mathcal{R}_{\text{dist}}(\boldsymbol{x}, r)$ is the cost-limited reachable set (closed if not otherwise stated) for the system in Eq. (1) centered at $\boldsymbol{x}$ within a path length of r (e.g., a sphere for Euclidean systems):

$$\mathcal{R}_{\text{dist}}(\boldsymbol{x}, r) = \left\{\boldsymbol{z} \in \mathcal{X} \mid \text{dist}(\boldsymbol{x}, \boldsymbol{z}) \leq r\right\}. \quad (3)$$

The dist-clearance of a query γ is defined as

$$\delta_{\text{dist}}(\gamma) = \sup\left\{\delta_{\text{dist}}(\sigma) \mid \sigma \in \Sigma_\gamma\right\} \quad (4)$$

and denotes the maximum clearance that a solution path to a query can have. An optimal sampling-based algorithm solves the following $\hat{\delta}_{\text{dist}}$-robustly feasible motion planning problem $\mathcal{P}$: given a query $\hat{\gamma}$ with a dist-clearance of $\delta_{\text{dist}}(\hat{\gamma}) > \hat{\delta}_{\text{dist}}$, find a control $\mathbf{u}(t) \in \mathcal{U}$ with domain [0,1] such that the unique trajectory σ satisfies Eq. (1), is fully contained in the free space $\mathcal{X}_{\text{free}} \subseteq \mathcal{X}$ and goes from $\boldsymbol{x}_{\text{start}}$ to $\boldsymbol{x}_{\text{goal}}$. Moreover it minimizes, asymptotically, a defined cost function $c : \Sigma_\gamma \to \mathbb{R}_{\geq 0}$. Hereinafter, we will use the term *steer* function to indicate a function that generates a path in $\mathcal{X}$ connecting two specified states. In particular we will use steer functions that solve an optimal control problem, i.e., minimizing the cost c.

In the following sections, we describe the approach to solve $\mathcal{P}$ by using an optimization-based sampling technique that minimizes the actual dispersion of the sampling set used by batch-processing algorithms (e.g., PRM*.[1], see Algorithm 1).

Algorithm 1: PRM*. $\boldsymbol{x}_{\text{start}}$ is the Start State, $\boldsymbol{x}_{\text{goal}}$ the Goal State, n the Desired Number of Samples.

```
1:  procedure PRM*
2:    S ← SAMPLEFREE(n)
3:    V ← {x_start, x_goal} ∪ S
4:    for v in V do
5:      U ← NEAR(V, v, r_n)
6:      for u in U do
7:        if COLLISIONFREE(v, u) then
8:          E ← E ∪ {(v, u)}
9:        end if
10:     end for
11:   end for
12:   return SHORTESTPATH(x_start, x_goal, (V, E))
13: end procedure
```

Algorithm 2: Dispersion Optimization.

```
1:  procedure DISPERTIO
2:    D ← DISTANCETOBORDER
3:    while |S| < n do
4:      x_i ← arg max_c D_c
5:      UPDATEDISTANCEMATRIX(D, x_i)
6:      S ← S ∪ {x_i}
7:    end while
8:  end procedure
```

B. Dispersion for Differentially Constrained Systems

We use and modify the dispersion definition for a sampling set $\mathcal{S} = \{\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subset \mathcal{X}$, introduced by Niederreiter [22] and also adopted by [9], [10]:

$$d_{\text{dist}} = \sup\{r > 0 \mid \exists \boldsymbol{x} \in \mathcal{X} \text{ with } \mathcal{R}_{\text{dist}}(\boldsymbol{x}, r) \cap \mathcal{S} = \emptyset\}. \quad (5)$$

Intuitively the dispersion can be considered as the radius of the largest (open) ball (i.e., size of the reachable set) that does not contain an element of S. In the context of differentially constrained motion planning, we propose to adjust the dispersion metric to explicitly require the reachable sets $\mathcal{R}_{\text{dist}}(\boldsymbol{x}, r)$ to be fully contained in $\mathcal{X}$:

$$\tilde{d}_{\text{dist}} = \sup\{r > 0 \mid \exists \boldsymbol{x} \in \mathcal{X} \text{ with } \mathcal{R}_{\text{dist}}(\boldsymbol{x}, r) \cap \mathcal{S} = \emptyset \wedge \mathcal{R}_{\text{dist}}(\boldsymbol{x}, r) \subseteq \mathcal{X}\}. \quad (6)$$

We also require this metric to respect possible identifications of the configuration space. Differently from previous approaches [9]–[11], we will compute the dispersion metric by numerically generating offline the reachable sets $\mathcal{R}_{\text{dist}}(\boldsymbol{x}, r)$ where $r > 0$ is the path length obtained by an optimal controller.

C. The Dispersion Optimization Algorithm

As discussed by [2], [10] multi-query sampling-based planners, such as PRM* or FMT*, generate as initial step a set S of collision free samples, see line 2 of Algorithm 1. Instead of using i.i.d. random variables, or an existing deterministic technique to generate $\mathcal{S}$ (e.g., Halton sequence, [9]–[11]), we propose to compute the set by minimizing the dispersion of Eq. 6. Our algorithm named Dispertio is outlined in Algorithm 2. The general idea of the algorithm is to pick in each step the sample (up to $n < N_{CS}$) that maximizes the distance to both the defined border of the configuration space as well as to the next sample. In other words we want to greedily put the sample into the position that currently defines the dispersion.

We propose to make this task computationally feasible by discretizing the configuration space into a fine grid of N_{CS} equidistant (distance could be different per dimension) cells. The dispersion tensor $\boldsymbol{D}$ keeps track of the minimum distance to either the border or closest sample for each grid cell (in Algorithm 2 we denote the dispersion value at the cell or position $\boldsymbol{c}$ as $D_{\boldsymbol{c}}$), computed by solving Eq. 6 using an optimal steer function; see Fig. 1.

If it is possible to compute the distance to the border quickly (e.g., Euclidean case), we initialize $\boldsymbol{D}$ with the distance to the border for each grid cell, otherwise $\boldsymbol{D}$ is initialized with ∞, line 1 of Algorithm 2. In this case, we check whether the update step to a potential sample would affect any border sample. If this is the case, we will not add the sample to $\mathcal{S}$, but instead run an update step on the border sample without adding it. At each algorithm iteration, we generate a sample $\boldsymbol{x}_i$ that maximizes the current dispersion tensor $\boldsymbol{D}$ and add it to $\mathcal{S}$, see lines 3–7 of Algorithm 2. For a given sample, $\boldsymbol{D}$ is updated (line 5 of Algorithm 2) with a flood-fill algorithm, by only expanding cells for which the dispersion has been updated. In this way we are

[1]Due to space limitations, we will not detail the algorithm PRM*. A reader interested to the properties of the algorithm can refer to [4].

exploiting the connectedness of time-limited reachable sets. The flood-fill algorithm sequence can be pre-computed to prevent double checking of already tested cells.

Despite having a time complexity exponential in dimensions due to the flood-fill algorithm (i.e., $\mathcal{O}(n\xi^D)$, with the constant $\xi > 0$ being related to discretization and complexity of dist), the algorithm is a feasible pre-computation step for many systems (e.g., Reeds-Shepp space, 6D kinematic chain using Euclidean distance). Once the set $\mathcal{S}$ has been generated, we can then use it in a motion planning algorithm such as PRM* (Algorithm 1). PRM*-edges are generated with the same steer function used to optimize the set $\mathcal{S}$.

D. Dispertio-PRM* Analysis

In this section we detail how PRM* [4], when using our deterministic sampling approach, retains the completeness and asymptotic optimality properties as in [9]–[11].

1) Completeness: We show that the approach deterministically returns a solution if it exists and returns failure otherwise [9]–[11]. Note that this is a stronger property than *probabilistic completeness* [1].

Theorem 1: Given a set of samples $\mathcal{S}$ with known dispersion $\tilde{d}_{\text{dist}}$ and considering general driftless systems for which we have steer functions that are optimal and symmetric, we can solve all planning queries γ with Algorithm 1 using a connection radius

$$r > 2\tilde{d}_{\text{dist}} \tag{7}$$

having clearance of

$$\delta_{\text{dist}}(\gamma) > 2\tilde{d}_{\text{dist}}. \tag{8}$$

Proof: To see this, first note that $\mathcal{R}_{\text{dist}}(\boldsymbol{x}, r)$ for optimal steering functions, is equivalent to time-limited reachable sets of the system. Hence, trajectories from $\boldsymbol{x}$ to any other point in $\mathcal{R}_{\text{dist}}(\boldsymbol{x}, r)$ will also be fully contained in $\mathcal{R}_{\text{dist}}(\boldsymbol{x}, r)$. Given a query γ with clearance $\delta(\gamma) > 2\tilde{d}_{\text{dist}}$, there exists a solution σ with $\mathcal{R}(\boldsymbol{s}_i, 2\tilde{d}_{\text{dist}}) \in \mathcal{X}_{\text{free}},\ \forall \boldsymbol{s}_i \in \sigma^*$. First note that due to the dispersion definition, there must be a sample of $\mathcal{S}$ in both $\mathcal{R}_{\text{dist}}(\boldsymbol{x}_{\text{start}}, \tilde{d}_{\text{dist}})$ and $\mathcal{R}_{\text{dist}}(\boldsymbol{x}_{\text{goal}}, \tilde{d}_{\text{dist}})$. Thus it is possible to connect the start and goal configuration to the roadmap. It remains to show that a dist-clearance of $\delta_{\text{dist}}(\gamma) > 2\tilde{d}_{\text{dist}}$ is sufficient to find a path from $\boldsymbol{x}_{\text{start}}$ to $\boldsymbol{x}_{\text{goal}}$. Let $\boldsymbol{q}_0$ and $\boldsymbol{q}_N$ denote the samples that $\boldsymbol{x}_{\text{start}}$ and $\boldsymbol{x}_{\text{goal}}$ are connected to, respectively. By taking $\boldsymbol{s}_1$ along a path σ and such that $\boldsymbol{q}_0$ lies on the border of $\mathcal{R}_{\text{dist}}(\boldsymbol{s}_1, \tilde{d}_{\text{dist}})$ we see, due to the dispersion definition in Eq. (6), that there must be another sample in the reachable set, denoted by $\boldsymbol{q}_1$. At this point we only know that the path from $\boldsymbol{q}_0$ over $\boldsymbol{s}_1$ to $\boldsymbol{q}_1$ must be collision free. Since only $\boldsymbol{q}_0$ and $\boldsymbol{q}_1$ are known, we require a factor of 2 in the clearance (i.e., $2\tilde{d}_{\text{dist}}$ in Eq. (8)), which ensures that the path from $\boldsymbol{q}_0$ to $\boldsymbol{q}_1$ must be collision-free. To see this, note that due the system symmetry both $\mathcal{R}_{\text{dist}}(\boldsymbol{q}_0, \tilde{d}_{\text{dist}})$ and $\mathcal{R}_{\text{dist}}(\boldsymbol{q}_1, \tilde{d}_{\text{dist}})$ must be contained in $\mathcal{R}(\boldsymbol{s}_1, 2\tilde{d}_{\text{dist}}) \subset \mathcal{X}_{\text{free}}$. We also know that the intersection $\mathcal{R}_{\text{dist}}(\boldsymbol{q}_0, \tilde{d}_{\text{dist}}) \cap \mathcal{R}_{\text{dist}}(\boldsymbol{q}_1, \tilde{d}_{\text{dist}})$ contains $\boldsymbol{s}_1$ and is thus nonempty. The trajectory from $\boldsymbol{q}_0$ to $\boldsymbol{q}_1$ must pass through this intersection and is hence collision-free. The same idea can now be repeated until the path to $\boldsymbol{q}_N$ is found. Fig. 3 visualizes the proof. ■

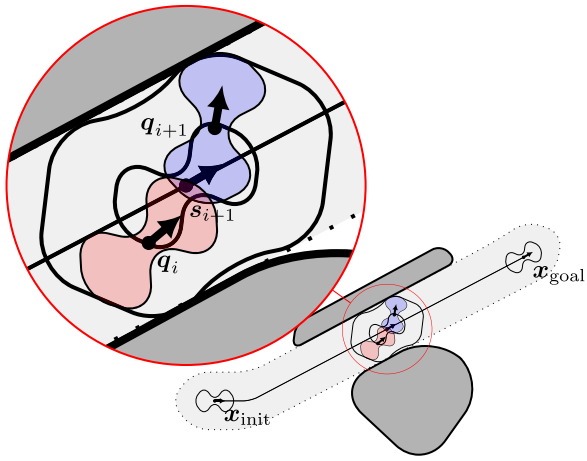


Fig. 3. Visualization of the proof of completeness. $\boldsymbol{s}_{i+1}$ is placed along the (unknown) path of maximum clearance σ such that $\boldsymbol{q}_i$ lies on the border of $\mathcal{R}_{\text{dist}}(\boldsymbol{s}_{i+1}, \tilde{d}_{\text{dist}})$. Since $\mathcal{R}_{\text{dist}}(\boldsymbol{q}_i, \tilde{d}_{\text{dist}})$ and $\mathcal{R}_{\text{dist}}(\boldsymbol{q}_{i+1}, \tilde{d}_{\text{dist}})$ overlap and are fully contained in $\mathcal{R}_{\text{dist}}(\boldsymbol{s}_{i+1}, 2\tilde{d}_{\text{dist}})$ the path from $\boldsymbol{q}_i$ to $\boldsymbol{q}_{i+1}$ is collision free.

2) Asymptotic Optimality: In this section, following [10] we will show that PRM* is asymptotically optimal when using the sampling sets generated by Dispertio. Particularly, Janson *et al.* [10] show that PRM* is asymptotically optimal when using deterministic sampling sets in D dimensions whose dispersion is upper-bounded by $\gamma_{PRM^*}\, n^{-1/D},\ \gamma_{PRM^*} > 0$. Next, we will show how the sampling sets generated by our approach reach the same asymptotic dispersion (see Theorem 2), i.e., lower l_2-dispersions, for all driftless control-affine systems, therefore retaining the PRM* asymptotic optimality. For the special Euclidean case, we show that the algorithm reaches the same asymptotically optimal dispersion as for example the Halton sequence. Note that for simplicity we are using a simplified version of the algorithm, without discretization and assuming the distance to the border is known. Throughout the discussion we again assume that the distance function dist is symmetric and optimal. Also note that $\mathcal{R}(\boldsymbol{x}, r)$ now denotes the *open* ball of radius r at $\boldsymbol{x}$ and $V(\cdot)$ the volume of a set.

Theorem 2: Under the assumption that the discretization of the space does not influence the placement of the samples, Algorithm 2's dispersion can be bounded by

$$nV(\mathcal{R}(\boldsymbol{x}, d_n/2)) \leq V(\mathcal{X}) \tag{9}$$

where d_n denotes the dispersion defined for a distance function dist as in Eq. (5), when n samples have been picked, i.e., $|\mathcal{S}| = n$. This yields for the D-dimensional Euclidean case an asymptotic behavior of

$$d_n \in \mathcal{O}\left(n^{-1/D}\right) \tag{10}$$

and the driftless control-affine case

$$d_n \in \mathcal{O}\left(n^{-1/\tilde{D}}\right) \tag{11}$$

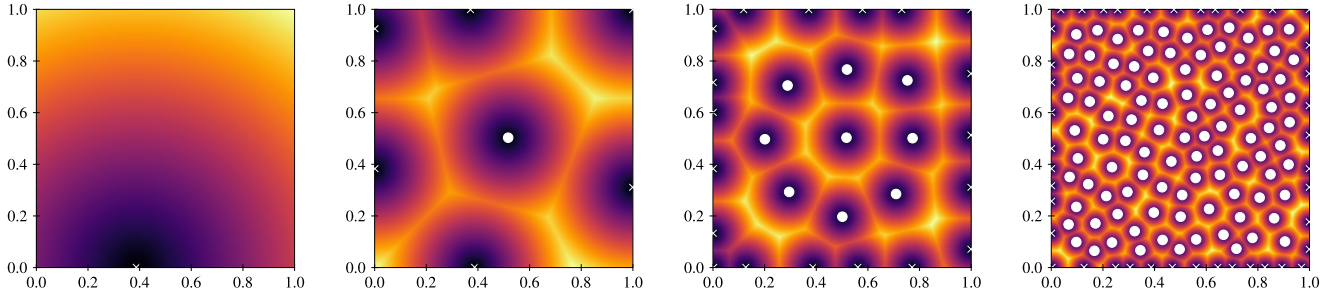


Fig. 4. Progression of the algorithm in 2D Euclidean space. The background color indicates the distance to the next sample (i.e., the distance matrix $\boldsymbol{D}$). The white crosses and dots show the processed border points and actual samples respectively.

with $\tilde{D} = \sum_{i=1}^{D} w_i$, where w_i are the weights of the boxes approximating the reachability space for driftless control-affine systems (see ball-box theorem [23], [24]).

Proof: To prove the asymptotic behavior of the algorithm, let us consider the case in which the discretization of the space has no effect on the placement of samples.[2] The key argument to analyze the asymptotic behavior of the algorithm is to realize that the nth sample is, by construction, placed such that its distance to the closest neighbor is d_{n-1}. Due to that, after n samples have been picked, we can note that

$$d_{n-1} \leq \min_{\boldsymbol{y} \in \mathcal{S} \setminus \boldsymbol{x}} \operatorname{dist}(\boldsymbol{x}, \boldsymbol{y}) \leq 2 d_{n-1} \quad \forall \boldsymbol{x} \in \mathcal{S}, \tag{12}$$

where the second inequality follows from the symmetry and optimality assumption of dist. From the first inequality it follows that (note that the ball is open)

$$\mathcal{R}(\boldsymbol{x}, d_{n-1}) \cap \mathcal{S} = \emptyset \quad \forall \boldsymbol{x} \in \mathcal{S}. \tag{13}$$

In addition, because of symmetry and optimality, the intersection of all open balls of radius $d_{n-1}/2$ must be empty, i.e.,

$$\bigcap_{\boldsymbol{x} \in \mathcal{S}} \mathcal{R}(\boldsymbol{x}, d_{n-1}/2) = \emptyset. \tag{14}$$

Note that $d_n \leq d_{n-1}$ and with n samples being in $\mathcal{S}$ we can state that

$$n V\big(\mathcal{R}(\boldsymbol{x}, d_n/2)\big) \leq V(\mathcal{X}) \tag{15}$$

must hold. To upper bound the dispersion for a number of samples n we would optimally use an explicit term for the volume $V\big(\mathcal{R}(\boldsymbol{x}, d_n/2)\big)$, but if no such term exists (as for general sub-Riemannian balls), we need to use a lower bound, for example by using the ball-box theorem. Let us first consider the case of a D-dimensional Euclidean space $\mathcal{X}$. In that case we obtain

$$n \alpha d_n^D \leq V(\mathcal{X}) \tag{16}$$

and thus

$$d_n \leq \frac{V(\mathcal{X})^{1/D}}{\alpha^{1/D} n^{1/D}} \in \mathcal{O}\left(n^{-1/D}\right) \tag{17}$$

[2]For brevity, we remove the explicit dist from the dispersion, but it is implied to be the distance function used in the algorithm and the reachable sets $\mathcal{R}$.

with $\alpha > 0$. This shows that in the Euclidean case, the achieved asymptotic dispersion is the same as for l_2 low dispersion sequences (e.g., Halton). For the driftless control-affine case we can use the same argumentation as in [11]. Under the assumption that the system is sufficiently regular we can find a parameter $A_{\max}$ such that

$$\operatorname{Box}^w\left(\boldsymbol{x}, \frac{d_n}{2A_{\max}}\right) \subseteq \mathcal{R}(\boldsymbol{x}, d_n/2) \tag{18}$$

and according to Lemma II.2 by Schmerling *et al.* [21] the volume is given by

$$V\left(\operatorname{Box}^w\left(\boldsymbol{x}, \frac{d_n}{2A_{\max}}\right)\right) = \left(\frac{d_n}{2A_{\max}}\right)^{\tilde{D}} \tag{19}$$

with $\tilde{D} = \sum_{i=1}^{D} w_i$. We can rewrite Eq. (15) as

$$n \left(\frac{d_n}{2A_{\max}}\right)^{\tilde{D}} \leq V(\mathcal{X}) \tag{20}$$

and thus

$$d_n \leq \frac{V(\mathcal{X})^{1/\tilde{D}} 2A_{\max}}{n^{1/\tilde{D}}} \in \mathcal{O}\left(n^{-1/\tilde{D}}\right). \tag{21}$$

■

Note that if the number of samples approaches the discretization of the space, they will actually converge to a Sukharev grid [18]. Hence, in the Euclidean case, the asymptotic dispersion is still $\mathcal{O}(n^{-1/D})$, but in the general case, we would need to inner-bound the reachable set with a Euclidean ball, which would lead to rather crude approximations as shown by Janson *et al.* [10] for the linear affine case. Thus, especially for nonholonomic systems, a grid of high resolution may be important to capture the shape of the reachable sets.

Corollary 1: Given that our set S has the same asymptotic dispersion as l_2 low dispersion sequences, our approach retains the asymptotic analysis carried out in [10], [11] and it allows the usage of a PRM* connection radius $r_n \in \omega(n^{-1/D})$.

IV. Evaluation

In this section we describe the experiments to evaluate how our approach performs in terms of planning efficiency and path quality compared to a set of baselines. To this end, we design two main experiments. In the first experiment, we compare our

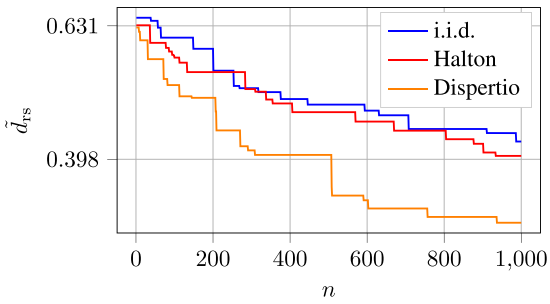


Fig. 5. Dispersion trend for the Reeds-Shepp case ($\eta = 1.0$, obstacle free environment). Our approach obtains a better dispersion than the baselines, thus achieving a better coverage of the state space as also shown in Fig. 1.

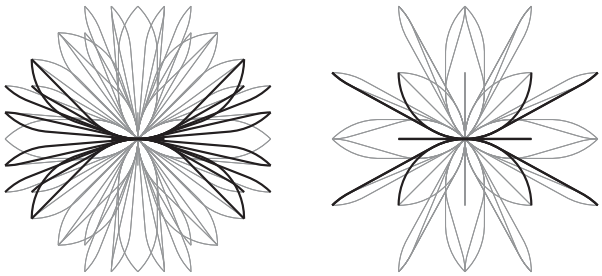


Fig. 6. The motion primitive sets, used for the state lattice approaches in Table I and I, respectively.

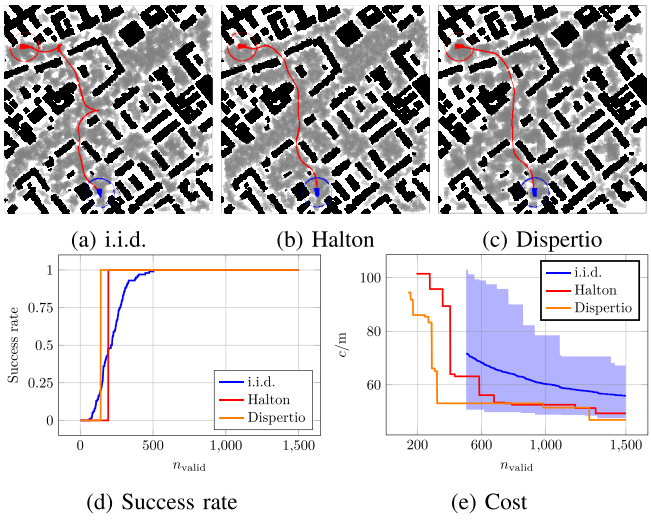


Fig. 7. Qualitative comparison of i.i.d., Halton and Dispertio. The top row shows example paths obtained after 1500 valid samples connecting starts (in **red**) with goals (in **blue**). The gray footprints represent the roadmap's vertices. The bottom row shows success rate and cost for this example.

approach against the baselines (uniform i.i.d. samples, Halton samples [9], Poccia's approach [11], state-lattice approach [16]) for a car-like kinematic systems (i.e., Reeds-Shepp (RS) [13]), over a subset of maps from the benchmark *moving-ai* [25], i.e., city maps, see Fig. 7 for example maps. The benchmark contains maps with several narrow corridors, and the planner needs to perform complex maneuvers (i.e., wholly exploiting

TABLE I
PATH QUALITY RESULTS OF ALL THE METHODS FOR THE CITY-MAPS BENCHMARK [25]. DISPERTIO OBTAINS ON AVERAGE BETTER SOLUTIONS AGAINST ALL THE BASELINES. SCORES ARE NORMALIZED BY $\sigma_{\text{rand}} = \sqrt{650}$

	i.i.d.	Halton	Poccia	Dispertio	Lattice
i.i.d.	·	-4.75	-5.41	-8.24	18.75
Halton	4.75	·	-1.84	-3.65	20.16
Poccia	5.41	1.84	·	-1.88	21.18
Dispertio	8.24	3.65	1.88	·	21.53
Lattice	-18.75	-20.16	-21.18	-21.53	·

(a) PRM*, Reeds-Shepp, $\eta = 0.1$, $n_{\text{all}} = 5000$, $r = 8.5\,\text{m}$

	i.i.d.	Halton	Poccia	Dispertio	Lattice
i.i.d.	·	-5.30	-6.86	-9.96	12.28
Halton	5.30	·	-0.51	-2.98	16.71
Poccia	6.86	0.51	·	-1.77	16.98
Dispertio	9.96	2.98	1.77	·	17.96
Lattice	-12.28	-16.71	-16.98	-17.96	·

(b) PRM*, Reeds-Shepp, $\eta = 0.05$, $n_{\text{all}} = 3200$, $r = 12.5\,\text{m}$

the full maneuverability), to let the car achieve its goal. We use a minimum turning radius $\rho = 5$ m and plan in environments of different size with w being the width of the map. For the state lattice, sets of motion primitives have been chosen after an informal validation and are shown in Fig. 6. The actual dispersion is reported as $\tilde{d}_{\text{rs}}$ and the number of drawn samples as n_{all}.

In the second experiment, we compare the approach to the baselines on a set of randomized maps and random planning queries. To show the general applicability of the algorithm we also benchmark it for a 6D kinematic chain in the 2D plane (comparing it to Halton sequence and i.i.d. samples). In this case, each joint either has an angle $\theta_i \in [-3, 3]$ with $i = 1, \ldots, 6$, or we plan in an identified space (i.e., the arms can wrap around) with $\theta_i \in [-\pi, \pi]/\sim$. We use as distance function the 2-norm in joint space (respecting the possible identification).

In both experiments we evaluate the approach in terms of cost (i.e., path length) and success rate, and show planning efficiency by plotting the cost progression. We use OMPL [26] and adopt its PRM* implementation (we made it deterministic by removing the random walk expansion step). As the connection strategy, we either use a fixed connection radius based on the computed dispersion such that Eq. (7) holds or the k-nearest connection strategy with $k_n = e(1 + 1/D) \log n$ (for the arm experiments), which ensures asymptotic optimality for all the samplers. All experiments run on a machine with Xeon E5-1620 CPU and 32 GB of RAM.

V. RESULTS AND DISCUSSION

We collect the evaluation's results in Tables I-II. The scores report how often an approach (on the table row) generates a better solution than another (on the table column). Whenever an approach is better, the score was changed by $+1$, by 0 for a draw (i.e., both fail to find a solution), and -1 if the approach yielded the higher cost. To account for the dependency of this score on the number of performed runs m, we normalize the scores by the standard deviation in case both samplers had a 50% chance of giving the better score (i.e., $\sigma_{\text{rand}} = \sqrt{m}$). In Table I, a green cell highlights that the approach on the table row

TABLE II
PATH QUALITY RESULTS OF DISPERTIO AGAINST THE BASELINES, ON RANDOMIZED MAPS AND QUERIES IN DIFFERENT SPACES (REED-SHEPP AND 6D KINEMATIC CHAIN). SCORES ARE NORMALIZED BY $\sigma_{\text{rand}} = \sqrt{1000}$

	i.i.d.	Halton	Poccia
RS $\eta = 1$, $n_{\text{all}} = 1500$, $r = 3.3\,\text{m}$	13.44	9.08	9.71
RS $\eta = 0.25$, $n_{\text{all}} = 1500$, $r = 6.5\,\text{m}$	8.73	5.79	3.32
RS $\eta = 0.1$, $n_{\text{all}} = 1500$, $r = 11\,\text{m}$	10.47	6.58	5.09
RS $\eta = 0.05$, $n_{\text{all}} = 1500$, $r = 16\,\text{m}$	15.27	7.59	8.10
KC $\{[-\pi, \pi]/\sim\}^6$, $n_{\text{all}} = 30000$, k-n	2.28	1.01	·
KC $[-3, 3]^6$, $n_{\text{all}} = 30000$, k-n	7.78	7.78	·

performs better than the one on the table column, red otherwise. Table II reports only the scores obtained by Dispertio against the baselines. Reeds-Shepp results are shown for different ratios $\eta = \rho/w$.

Experiment 1: Table I shows the results of the first experiment considering 13 maps with 50 queries each. Overall our approach achieves better costs and a higher success rate compared to the baselines. In general, Halton is the second best placed. State-lattice performs poorly, indicating that more effort is required for their motion primitives design (i.e., possibly also using a notion of dispersion in control space). Fig. 7 shows an example planning query for i.i.d., Halton sampling and our approach. It reports the obtained paths, the trend for the success rate and the cost progression. The blue range shows the minimum and maximum cost observed in these runs for the i.i.d. sampler. The cost results are only shown for success rates of 100%. Cost and success rate progressions of Fig. 7 highlight how our approach is faster in getting an initial good solution, and faster (as the number of samples increases) in converging to lower cost solutions in those cluttered and narrow scenarios. Furthermore, Fig. 5 reports a numerical comparison of the dispersion obtained for the Reeds-Shepp case after n samples for i.i.d., Halton and the proposed approach in an obstacle-free environment. Our approach achieves better dispersion than the baselines.

Experiment 2: Table II reports the results for the second experiment. Also in this case, our approach achieves better performance in terms of final cost solution even in higher dimensional spaces. For the arm experiments we use a k-nearest neighbor connection strategy, which shows that the qualitative advantages hold even with different connection strategies (although not reported, similar results are obtained with a radius-based strategy that satisfies Eq. (7)). Furthermore, in very cluttered environments (with $\eta = 1.0$) our approach achieves on average a 10% higher success rate than the baselines, indicating how it can better exploit the knowledge of the nonholonomic constraints (i.e., maneuvering capabilities) in narrower scenarios.

VI. CONCLUSIONS

In this work we extend deterministic sampling-based motion planning to the class of symmetric and optimal driftless systems, by proposing Dispertio, an algorithm for optimized deterministic sampling set generation. When used in combination with PRM*, we prove that the approach is deterministically complete and retains asymptotic optimality. In the evaluation, we show that our sampling technique outperforms state-of-the-art methods in terms of solution cost and planning efficiency, while also converging faster to lower cost solutions.

As future work, we are interested in extending the approach towards non-uniform sampling schemes, for example to exploit learned priors, and to systems with drift.

REFERENCES

[1] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.
[2] S. M. LaValle and J. J. Kuffner, Jr, "Randomized kinodynamic planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378–400, 2001.
[3] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.
[4] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.
[5] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *Int. J. Robot. Res.*, vol. 34, pp. 883–921, 2015.
[6] T. Lozano-Pérez and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Commun. ACM*, vol. 22, no. 10, pp. 560–570, 1979.
[7] L. Palmieri, S. Koenig, and K. O. Arras, "RRT-based nonholonomic motion planning using any-angle path biasing," in *Proc. Int. Conf. Robot. Autom.*, Stockholm, Sweden, 2016, pp. 2775–2781.
[8] L. Palmieri, T. P. Kucner, M. Magnusson, A. J. Lilienthal, and K. O. Arras, "Kinodynamic motion planning on Gaussian mixture fields," in *Proc. Int. Conf. Robot. Autom.*, Singapore, 2017, pp. 6176–6181.
[9] S. M. LaValle, M. S. Branicky, and S. R. Lindemann, "On the relationship between classical grid search and probabilistic roadmaps," *Int. J. Robot. Res.*, vol. 23, no. 7/8, pp. 673–698, 2004.
[10] L. Janson, B. Ichter, and M. Pavone, "Deterministic sampling-based motion planning: Optimality, complexity, and performance," *Int. J. Robot. Res.*, vol. 37, no. 1, pp. 46–61, 2018.
[11] E. Poccia, "Deterministic sampling-based algorithms for motion planning under differential constraints," Master's thesis, Pisa Univ., Pisa, Italy, 2017.
[12] L. Palmieri and K. O. Arras, "A novel RRT extend function for efficient and smooth mobile robot motion planning," in *Proc. Int. Conf. Intell. Robots Syst.*, Chicago, IL, USA, 2014, pp. 205–211.
[13] J. Reeds and L. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific J. Math.*, vol. 145, no. 2, pp. 367–393, 1990.
[14] J. Lengyel, M. Reichert, B. R. Donald, and D. P. Greenberg, "Real-time robot motion planning using rasterizing computer graphics hardware," *ACM SIGGRAPH Comput. Graph.*, vol. 24, no. 4, pp. 327–335, 1990.
[15] A. Nash, K. Daniel, S. Koenig, and A. Felner, "Theta*: Any-angle path planning on grids," in *Proc. AAAI*, 2007, vol. 7, pp. 1177–1183.
[16] M. Pivtoraiko and A. Kelly, "Kinodynamic motion planning with state lattice motion primitives," in *Proc. Int. Conf. Intell. Robots Syst.*, San Francisco, CA, USA, 2011, pp. 2172–2179.
[17] J. Van der Corput, "Verteilungsfunktionen I & II," in *Proc. Nederl. Akad. Wetensch.*, vol. 38, 1935, pp. 1058–1066.
[18] A. G. Sukharev, "Optimal strategies of the search for an extremum," *USSR Comput. Math. Math. Phys.*, vol. 11, no. 4, pp. 119–137, 1971.
[19] M. S. Branicky, S. M. LaValle, K. Olson, and L. Yang, "Quasi-randomized path planning," in *Proc. Int. Conf. Robot. Autom.*, Seoul, South Korea, 2001, pp. 1481–1487.
[20] W. Khaksar, T. S. Hong, M. Khaksar, and O. Motlagh, "A low dispersion probabilistic roadmaps (LD-PRM) algorithm for fast and efficient sampling-based motion planning," *Int. J. Adv. Robot. Syst.*, vol. 10, 2013.
[21] E. Schmerling, L. Janson, and M. Pavone, "Optimal sampling-based motion planning under differential constraints: The driftless case," in *Proc. Int. Conf. Robot. Autom.*, Seattle, WA, USA, 2015, pp. 2368–2375.
[22] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, vol. 63. Philadelphia, PA, USA: SIAM, 1992.
[23] R. Montgomery, *A Tour of Subriemannian Geometries, Their Geodesics and Applications*. Providence, RI, USA: Amer. Math. Soc., 2002.
[24] W.-L. Chow, "Über systeme von linearen partiellen differential-gleichungen erster ordnung," in *The Collected Papers Of Wei-Liang Chow*. Singapore: World Scientific, 2002.
[25] N. Sturtevant, "Benchmarks for grid-based pathfinding," *IEEE Trans. Comput. Intell. AI Games*, vol. 4, no. 2, pp. 144–148, Jun. 2012.
[26] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robot. Autom. Mag.*, vol. 19, no. 4, pp. 72–82, Dec. 2012.