2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Concentration of Measure Phenomenon and its Implications for Sample-based Planning Algorithms in Very-High Dimensional Configuration Spaces

Joel M. Esposito

***Abstract*— In very high-dimensional ($\gg 10$) spaces, a collection of points generated uniformly at random will concentrate very tightly about its expected value – defying intuition developed in low-dimensional spaces. This paper explores the implications of this for two major classes of sample-based robot motion planning algorithms: Rapidly Exploring Random Trees (RRTs) and Probabilistic Road Maps (PRMs). First we show that the graph vertices concentrate in a thin-shelled hyper-sphere, with almost none near the origin nor at the edges of the workspace. Next we examine how varying one of the algorithms' parameters – the maximum edge length– can dramatically alter the algorithms' complexity and the connectivity of the resulting graph. Finally, we explore how the position of the initial node, often placed arbitrarily, can impact the shape of the graph. While the contributions of this paper are largely theoretical, many robotic applications of practical interest have extremely high-dimensional configuration spaces including humanoids, swarms and soft (a.k.a. continuum) robotics.**

I. Introduction

In the last few decades sample-based algorithms, such as Rapidly Exploring Random Trees (RRT) [1] and Probabilistic Roadmaps (PRM) [2], have become the dominant paradigm in robot motion planning. While there are great number of variants of these algorithms in the literature, they all sample a discrete set of points in the configuration space to efficiently build a connected and accessible graph which approximates the underlying free configuration space it is embedded in. This approach is in contrast to older computational geometry-type algorithms which try to construct exact representations of the space. Sampling based approaches are often touted as ways to combat the "curse of dimensionality".

Meanwhile the recent explosion of interest in so-called big data and deep learning has spurred an interest in the geometry of very, very high-dimensional spaces (see [3] for an overview). For example in the *word2vec* model for natural language processing, a text sample is represented as a point in a vector space of several hundred dimensions. Recommender systems, like those used to address the Netflix Challenge, try to find a low dimensional basis for high-dimensional data encoding the preferences of millions of users across a catalog of thousands of products. And training a deep convolutional neural network like *AlexNet* can be thought of as an optimization problem in a parameter space of millions of dimensions.

Robotics and Control Engineering, United States Naval Academy, Annapolis, Maryland 21402, USA, esposito@usna.edu

The geometry of such high-dimensional spaces often defies the intuition garnered from working with low-dimensional systems. For example, it is well know that the volume of a unit hyper-sphere peaks in 5-dimensions and then tends to zero as the number of dimension approaches infinity [4]. Our research program explores the implications of running some common sample-based planning algorithms in very high dimensional configuration spaces. In this paper we focus on one particular phenomenon – termed *concentration of measure* – which predicts that sums of many random variables (even uniformly distributed ones) concentrate exponentially tightly about their expected value. As we will see in Sect. IV, this effect occurs in as few as 2 dimensions, and by 20 dimensions it simply cannot be ignored. This has interesting implications for the various planning algorithm design choices, such as sampling strategy, initialization and parameter selection which, to our knowledge, which have not been discussed in the literature before.

The remainder of this paper is organized as follows. In Section II we review two major classes of sample-based planning algorithms. In Sect. III we review other works that have explicitly considered planning problems in very high dimensional spaces. In Sect. IV we formally introduce and prove the concentration of measure phenomenon. Section V illustrates a variety of implications for the RRT and PRM planning algorithms. Finally, we summarize our findings and our agenda for future work in Sect. VI.

II. Background: sample-based planning

Let a robot's configuration be defined as a point on a compact N-dimensional manifold $\mathbf{q} \in C$. The subset of the configuration space which does not include obstacles is called the *free* configuration space, C_{free}. The basic motion planning problem is as follows. Given initial and goal configurations, $\mathbf{q}_0$ and $\mathbf{q}_{goal}$, in the same connected components of C_{free}, find a continuous path connecting them lying entirely within C_{free}.

Sample-based planners such as RRT and PRM use a pre-processing phase to create a graph in C_{free} (specified by sets of vertices V and edges E), often termed a *roadmap*, which is both connected and easily accessible from any point in C_{free}, approximating the connectivity of C_{free}. A generic RRT pre-processing algorithm is shown in Fig. 1 while a simplified version of PRM is given in Fig. 2. The subroutines are defined as follows. `SAMPLE` generates a point in C_{free} (described in more detail later). `NEAREST` finds the

Rapidly Exploring Random Tree (RRT)

$V = \{\mathbf{q}_0\}$, $E = \{\}$
for $i = 1 \ldots n$ **do**
  $\mathbf{q}_{rand} =$ SAMPLE $\in C_{free}$
  $\mathbf{q}_{near} =$ NEAREST($\mathbf{q}_{rand}, V$)
  $\mathbf{q}_{new} =$ EXTEND($\mathbf{q}_{near}, \mathbf{q}_{rand}, d_{\max}$)
  if COLLISIONFREE($\mathbf{q}_{new}, \mathbf{q}_{near}$) **then**
    $V = V \bigcup \mathbf{q}_{new}$, $E = E \bigcup (\mathbf{q}_{new}, \mathbf{q}_{near})$
  end if
end for
RETURN $G = \{V, E\}$

Fig. 1. A generalization of the RRT pre-processing algorithm.

Probabilistic Road Map (PRM)

$V = \{\mathbf{q}_0\}$, $E = \{\}$
for $i = 1, \ldots, n$ **do**
  $\mathbf{q}_i =$ SAMPLE $\in C_{free}$
  $V = V \bigcup \mathbf{q}_i$
end for
for $i = 1, \ldots, n$ **do**
  $U =$ NEAR $(\mathbf{q}_i, V, d_{max})$
  for $\mathbf{q}_j \in U$ **do**
    if COLLISIONFREE $(\mathbf{q}_j, \mathbf{q}_i)$ **then**
      $E = E \bigcup (\mathbf{q}_j, \mathbf{q}_i)$
    end if
  end for
end for
RETURN $G = \{V, E\}$

Fig. 2. A generalization of the PRM pre-processing algorithm.

closest existing vertex, $\mathbf{q}_{near} \in V$ to a query point $\mathbf{q}$, while NEAR returns a subset of vertices, $U \subset V$, that are within a distance $d_{\max}$ of the query point $\mathbf{q}$, using a preferred metric defined on the configuration space. EXTEND grows the tree from $\mathbf{q}_{near}$ toward a sample configuration $\mathbf{q}_{rand}$ by a small distance, $d_{\max}$, using a local planner. Any candidate edge, $(\mathbf{q}_i, \mathbf{q}_j)$ is collision checked by COLLISIONFREE before being added to the graph.

Note that both algorithms involve generating samples in the free configuration space via SAMPLE. While there are many possible approaches, including deterministic sequences [5], goal-biased [6] or medial-axis sampling [7], by far the most common approach in practice is to use a component-wise uniformly distribution with rejection sampling – *i.e.* to generate a point using N independent uniform distributions defined on intervals whose Cartesian product is a hyper-rectangular super set of C, and then check if the point is contained in C_{free}. If the point is rejected, the process is repeated until it is successful.

Also observe that the parameter $d_{\max}$ – an upper threshold on the length of a graph edge – appears in both algorithms. In the case of RRT, the subroutine EXTEND attempts to grow or steer the tree toward the sample by a distance not to exceed $d_{\max}$; while in PRM, rather than collision checking all possible pairs of vertices, connections are only attempted between vertex pairs returned by NEAR which are closer than $d_{\max}$. While this distance limit is not included in every variant of these algorithms, it is extremely common in practice. The motivation for this heuristic is rooted in the assumption that shorter edges are more likely to be collision free; that the cost of collision checking a potential edge is often proportional to its length; and that very dense graphs can be cumbersome to store and search.

In Section IV we show that when such a uniform sampling is used to generate a large collection of points in a high-dimensional space, the distribution of distances between them concentrates exponentially tightly around some expected value; and in Section V we discuss the implications of this for the PRM and RRT algorithms.

III. Literature review

The literature on sample-based motion planning is vast and there are many variants of the basic PRM and RRT algorithms. In this article we focus on the impact of uniform random sampling but many other sampling strategies exist including deterministic sequences [5], goal-biasing [6] or medial-axis sampling [7]. In [8] the authors discuss some properties of RRTs, used in this paper, and some open problems. In [9] the authors introduced PRM* and RRT* – optimal variants of the respective original algorithms. In doing so they present a thorough analysis and comparison of the time complexity which we discuss later. In [10] a model of the rate at which a planner fills the space is proposed and is shown to scale with the volume of a hyper-sphere. This concept reappears in this paper.

The contribution of this paper is largely theoretical, though we think it may guide the design of algorithms for emerging applications in robotics involving high-dimensional configuration spaces. For example, several industrial robots have 8 degrees of freedom (DOF) while researchers have built hyper-redundant robots with up to 30 DOF [11] – including snake [12] and eel-like robots [13]. Planning for humanoid robots requires considering 40+ DOF [14]. Continuum-style manipulators and soft robots can be modeled with an arbitrary number of DOF [15]. In swarm applications, experiments with over 30 differential drive-style robots have been conducted (90 DOF) [16]. Certainly the ultimate goal of modular robotics research is to far exceed that number [17]. Researchers have long since applied robot motion planning techniques to protein folding which can have hundreds or thousands of DOF [18].

Here we discuss works explicitly aimed at planning in very high-dimensional configuration spaces. The provocatively titled article "Path Planning in 1,000+ Dimensions" [19], considers a strategy for very high-dimensional problems that have a very low-dimensional *task* space. By growing an RRT in the low-dimensional space and using a heuristic local

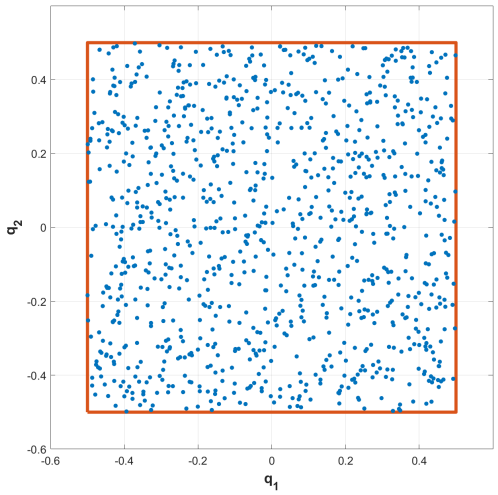

Fig. 3. An illustration of component-wise uniform sampling in 2-D.

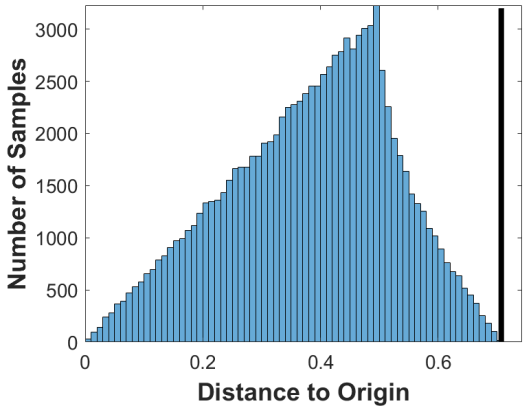

Fig. 4. The distribution of the distance of 100,000 points to the origin in 2 dimensions is not uniform.

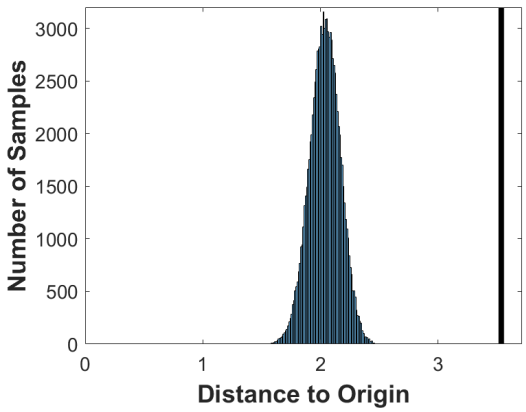

Fig. 5. The distribution of the distance of 100,000 points to the origin in 50 dimensions is highly concentrated.

planner to connect the samples in the higher dimensional space they are able to quickly solve very large problems. Unfortunately, they do not offer any analysis of the approach, which is unlikely to retain the probabilistic completeness of the original RRT algorithm. In a similar vein, [20] proposes an algorithm whose search begins in a low dimensional subspace of C_{free} and adaptively expands the search to include additional dimensions until a solution is found, preserving the completeness property of the original RRT. [21] seeks to leverage the Johnson-Lindenstrauss Lemma to estimate sample coverage in high-dimensional spaces using distance preserving random linear projections. [22] considers a similar approach but the projections are learned. In general we feel that the application of approximate nearest-neighbor techniques [23] and so-called matrix sketching techniques [3] to motion planning problems deserve more attention. [14] introduces XXL – a motion planner designed explicitly for high-dimensional spaces. That article includes a section that investigates an RRT variant called RLRT where the vertex selected for expansion is selected at random – instead of using the nearest neighbor algorithm. They show that on a 14 and 21-dimensional example problem the behaviour of these two algorithms is statistically in indistinguishable, suggesting that, in high dimensions, distance queries are less discriminating – a known fact related to the concentration of measure phenomena we discuss in this paper.

IV. Concentration Phenomena

We will use the term *component-wise uniform sampling* to refer to the following strategy for `SAMPLE`. Generate a point, $\mathbf{q}$ in the unit hypercube $[-1/2, 1/2] \times \cdots \times [-1/2, 1/2] \subset R^N$, where each component of the vector, q_i, is independently drawn from a continuous uniform distribution supported on the interval $[-1/2, 1/2]$.

Figure 3 illustrates a collection of 1,000 vertices generated in 2-D via such an approach. To the eye, Fig. 3 seems uniform, yet Fig. 4 shows that, even in 2-D, the distribution of the distance of the samples from the origin, $d = \|\mathbf{q}\|$, in *not* uniform – peaking around $d = 0.5$, with very few samples generated near the origin ($d \approx 0$) or at the vertices ($d \approx 0.707$ – the vertical line in Fig. 4). However in higher dimensions this disparity is *much* more extreme. Fig. 5 shows that in 50 dimensions virtually *all* of the samples are generated in a tight range $d \in [1.6, 2.4]$ with *none* being generated near the origin ($d \approx 0$) or near the vertices of the 50-D hypercube ($d \approx 3.5$ – the vertical line in Fig. 5).

In fact, concentration of measure predicts that, as N increases, with high probability *all* the samples will lie in the intersection of the unit hypercube and thin shelled hypersphere whose radius is the expected value of the distance, $E(d)$. While it is impossible to plot high-dimensional spaces, Fig. 6 provides a conceptual depiction of how the samples from Fig. 5 would be arranged in a 50-D hypercube. To derive this formally, first consider that the square of the L^2 norm of a random vector in R^N is the sum of N random variables, so via linearity:

$$E(d^2) = E(q_1^2 + \ldots q_N^2) = NE(q_i^2). \tag{1}$$

If q_i is drawn from a bounded uniform distribution defined

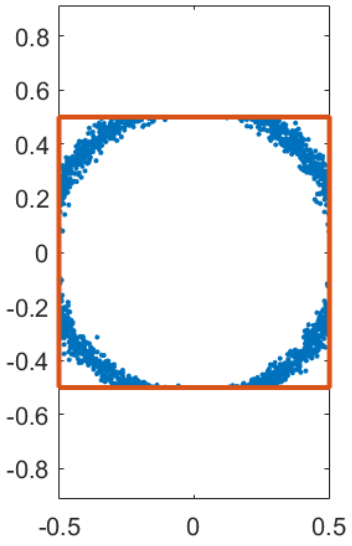

Fig. 6. A conceptual depiction of component-wise uniform sampling in 50 dimensions (compare to the 2-D case in Fig. 3). Despite being generated by a uniform distribution, points are concentrated at the intersection of the 50-D hypercube and a thin shelled hypersphere.

over $[-1/2, 1/2]$, the P.D.F. is simply 1. Then we may compute

$$E(q_i^2) = \int_{-1/2}^{1/2} t^2 dt = \frac{(1/2)^3}{3} - \frac{(-1/2)^3}{3} = 1/(12). \quad (2)$$

While there is no closed-form expression for $E(d)$ (also know as the hypercube point picking problem [24]), $E(d) \approx 1/2\sqrt{(N/3)}$. For example, when $N = 50$ $1/2\sqrt{(N/3)} = 2.04$ – which agrees with Fig. 5.

Because the terms are independent, the variance of the sum equals the sum of the variances (*i.e.* $Vard^2 = N \cdot Var(q_i^2)$), and

$$Var(q_i^2) = E(q_i^4) - E(q_i^2)^2, \quad (3)$$

where $E(q_i^2)^2 = (1/12)^2$ via Eq. (2) and

$$E(q_i^4) = \int_{-1/2}^{1/2} t^4 dt = \frac{(1/2)^5}{5} - \frac{(-1/2)^5}{5} = 1/80. \quad (4)$$

Therefore $Var(q_i^2) = 1/80 - (1/12)^2 \approx 0.0056$.

Now using Chernoff's bound, we can compute the bound probability a point is more than some distance c from the expected value

$$Pr(d^2 - E(d^2) \geq c^2) \leq e^{-\frac{(c^2 - N/12)^2}{N \cdot Var(q_i^2)}}, \quad (5)$$

which shows that the upper tail clearly decreases exponentially with both c and the dimension N (a similar result holds for the lower tail).

Next consider the distance between two uniformly randomly generated points inside the hypercube $d_{ij} = \|\mathbf{q}_i - \mathbf{q}_j\|$, known in the literature as the "hypercube line picking" problem. In this case the integral for the expected value, referred to as $\Delta(N)$, has not been solved analytically but can be easily numerically evaluated, for example in 3-D it is termed the Robbins Constant [25] $\Delta(3) \approx 0.661$ while in 8-D it is $\Delta(8) \approx 1.128$. It is known to be bounded as [24]:

$$\frac{1}{3}N^{\frac{1}{2}} \leq \Delta(N) \leq \left(\frac{1}{6}N\right)^{\frac{1}{2}} \sqrt{\frac{1}{3}\left[1 + 2\left(1 - \frac{3}{5N}\right)^{\frac{1}{2}}\right]}. \quad (6)$$

Again, since this quantity is the sum of N bounded random variables, it too concentrates exponentially in high dimensions via Chernoff's bound.

While the specific numerical results in this section were derived using the L^2 norm, assuming the variables were identically and uniformly distributed and that C_{free} was a unit hypercube, it is important to note that the concentration of measure phenomena holds for a much broader set of conditions including non-uniform or non-identically distributed samples, and for any norm that can be expressed as the sum of components.

V. Implications for Sample-Based Planning Algorithms

Now let us consider the process of growing an RRT or PRM in an N-dimensional unit hypercube, using the component-wise uniform sampling strategy discussed above.

a) Vertex Distribution: In the case of PRM since the vertices are directly generated by `SAMPLE`, it is obvious that the vertices will obey the distribution used for `SAMPLE`. In the case of RRT, if no distance limit is imposed (i.e. $d_{\max} = \infty$) the same vertex distribution emerges. When $d_{\max} \neq \infty$ it has been shown [8] that, as n increases, the vertex set converges to `SAMPLE`'s distribution in the limit. Eqs. (2) and (5) suggest that in high dimensions nearly all the vertices will concentrate in a thin shell near the intersection of the unit hypercube and a hyper-sphere of radius $E(d) = 1/2\sqrt{N/3}$. Again, Fig. 5 illustrates that for a PRM in 50 dimensions with 100,000 vertices generated inside the unit hypercube, *none* of the vertices are within 1.4 units of the origin ($d = 0$) nor within 1 unit of the hypercube vertices ($d \approx 3.54$) and a conceptual 2-D illustration of this can be seen in Fig. 6.

b) Nearest Neighbors: It is well known that in high dimensions Euclidean distance provides poor *contrast* – the difference between the maximum and minimum value relative to the expected value. For example, in 50-D (Fig. 5), the contrast is on the order of 20%; whereas in 2-D (Fig. 4) the contrast is 100%. This suggests that for high-dimensional problems approximate nearest neighbor algorithms [23] could reduce run-time without sacrificing completeness or path quality. It also suggests that other distance metrics (i.e. fractional) may provide more discriminatory power. This is likely the reason [14] reports that, in high dimensions, replacing the `NEAREST` vertex selection with a randomly selected vertex can perform nearly as well in terms of completeness and path quality.

c) Selection of $d_{\max}$: The design parameter $d_{\max}$ appears in both the PRM and RRT algorithms and effectively defines the maximum edge length in the resulting graph. In this section we ignore $\mathbf{q}_0$, which may be placed arbitrarily

and will be discussed in the next sub-section, and assume all samples are generated using component-wise uniform sampling.

In the case of PRM, if $d_{\max} << E(d_{ij})$ then with high probability the set returned by NEAR will *always* be empty and the resulting edge set will also be empty! Fig. 9 shows the possible range of values is $d_{\max} \in (0, 7.07]$. Clearly care must be taken in selecting its value because a small change in $d_{\max}$ can radically change the number of samples returned by NEAR. Using $d_{\max} < 1.9$ results in an empty edge set. Such a graph will never represent a solution to the motion planning problem because it is not connected. On the other hand $d_{\max} > 3.5$ results in the set returned by NEAR to be the set of *all* vertices V resulting in $n(n-1)/2$ calls to COLLISIONFREE. This is inadvisable since the motivation behind $d_{\max}$ is to avoid the time and storage complexity of an all-pairs search. Its this sensitivity that gives rise to the so-called phase-change phenomena in random graph theory [9].

The PRM* algorithm introduced in [9] selects:

$$d^*_{\max} = \gamma_{PRM}(N)(\log(n)/n)^{1/N} \tag{7}$$

where

$$\gamma_{PRM}(N) \geq 2(1+1/N)^{1/N}(\mu(C_{free})/\chi_N)^{1/N}, \tag{8}$$

$\mu(C_{free})$ is the Lebesgue measure (i.e., volume) of the free space, and χ_N is the volume of the unit ball in N-dimensions. [9] shows that this selection results in $O(\log(n))$ time complexity (i.e. $O(\log(n))$ vertices returned by NEAR).

While that is true in the limit $n \to \infty$, [9] also notes that when the number of vertices in the graph is below some threshold, $n < n_0$, the number of calls to COLLISIONFREE is actually linear in the number of nodes n. The relationship between n_0 and the dimension N is non-obvious; but relates to how $d^*_{\max}$ scales with the dimension relative to the expected distance between samples in Eq. 2. Considering Eq.7, the $\log(n)/n$ term clearly varies with the number of samples, while the remaining terms are solely functions of the dimension N – including the volume of a unit hyper-sphere, which goes to zero at a factorial (super-exponential) rate as $N \to \infty$.

Figure 7 shows that for a fixed number of vertices, the value of $d^*_{\max}$ (displayed in red lines representing $n = 10^3$, 10^5, 10^7 and 10^9) grows faster than $E(d_{ij})$. Because d_{ij} concentrates tightly, when the red lines are well below the green region connections are only attempted between a few vertices in the lower tail resulting in the desired logarithmic complexity; however when the red lines are inside the green region connections will be attempted between a large fraction of the vertices resulting in a number of calls proportional to n. To illustrate this trend, consider second lowest red line in Fig. 7 representing $n = 10^7$. When $N = 10$ the histogram in Fig. 8 shows that $d^*_{\max}$ (vertical red line) selects the exponentially decreasing lower tail and $3,814$ vertices (0.035% of the total) are returned by NEAR for collision checking – hence $O(\log n)$; however when $N = 50$ Fig. 9 shows that $d_{\max}*$ selects $5,000,000$ vertices (50%) – suggesting a linear dependence on n. Figure 10 shows the

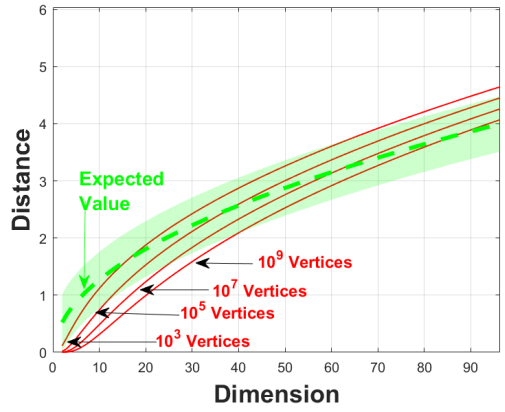

Fig. 7. The connection distance threshold suggested by PRM* for various numbers of vertices (10^3, 10^5, 10^7 and 10^9 shown in solid red lines) compared to the expected value of distance between nodes (green line, shaded area is ± 2 standard deviations).

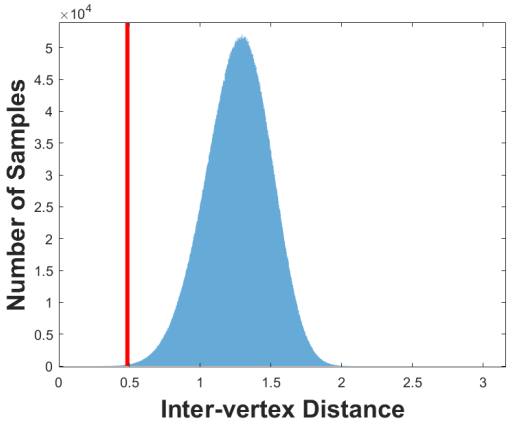

Fig. 8. For 10^7 vertices in 10-D the connection distance threshold suggested by PRM* (red vertical line) returns only 3,814 (0.038%) of the vertices for collision checking.

value of the threshold n_o as a function of the dimension, as computed numerically. Clearly it has exponential dependence on the dimension. In 10-D about 1.8×10^4 vertices are required to reach the $O(\log n)$ regime while are 1.6×10^{17} are need in 50-D. Naturally this "curse of dimensional" is a fundamental aspect of high dimensional problems; however it brings to light the fact that for very high dimensions, even with seemingly many nodes, $O(n)$ run time is to be expected.

d) Initial Configuration: The selection of $\mathbf{q}_0$ is often treated as arbitrary and inconsequential – frequently chosen at the origin $\mathbf{q}_0 = \mathbf{0}$. Keep in mind that the expected value of a distance of a sample configuration to the origin, $E(d)$, is much smaller than the expected value of the distance between two samples $E(d_{ij})$; and because these values concentrate tightly in higher dimensions we can say with high confidence that if $\mathbf{q}_0 = \mathbf{0}$, as $N \to \infty$

$$Pr(d_{0i} < d_{ij}) \to 1, \forall i, j. \tag{9}$$

In the case of PRM, this has the implication that if $E(d_{0i}) < d_{\max} < E(d_{ij})$ then with high probability, NEAR

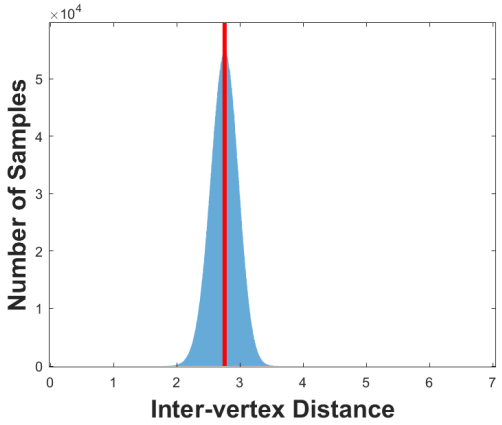

Fig. 9. For 10^7 vertices in 50-D the connection distance threshold suggested by PRM* (red vertical line) returns about 5,000,000 (50%) of the vertices for collision checking.

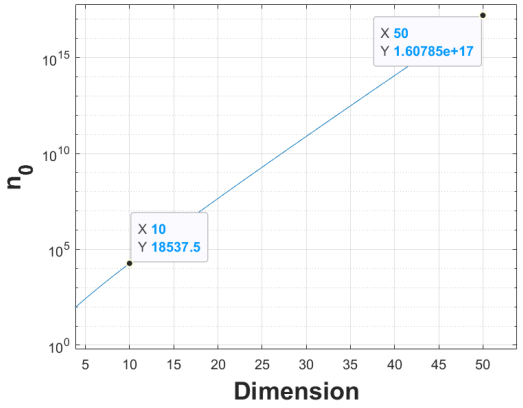

Fig. 10. The minimum number of vertices for PRM* to reach the $O(\log n)$ complexity regime, n_0, is exponential in dimension. Below that threshold the number of collision checks scales linearly with the number of nodes n. Note the log scale on the vertical axis.

will *only* return $\mathbf{q}_0$. This results in a roadmap resembling a star-shaped tree, rooted at $\mathbf{q}_0$ with every other node being a leaf (see Fig. 11). In the case of RRT an identical structure emerges if $d_{\max} \gg E(d)$, since the vertex distribution will match that of SAMPLE. For both cases, this behavior can be generalized beyond $\mathbf{q}_0 = \mathbf{0}$ to any initial configuration inside a ball of sufficiently small radius about the origin.

On the other hand in our RRT experiments where $d_{\max} \ll E(d)$, we consistently observed two trends. When the initial node is placed inside a hypersphere of radius $E(d)$ (i.e. $\|\mathbf{q}_0\| < E(d)$), at first it grows $N+1$ branches, quickly increasing the radius of the tree [8], until it touches the thin-shelled hypersphere at which point the radius levels off a few standard deviations above $E(d)$, and the remaining vertices approach the sample distribution as discussed above. On the other hand, when $\|\mathbf{q}_0\| \gg E(d)$ the tree rapidly grows a single branch toward the hypersphere, penetrating the interior before growing $N+1$ branches which eventually rejoin the thin-shelled hypersphere at which point the radius levels off a few standard deviations above $E(d)$.

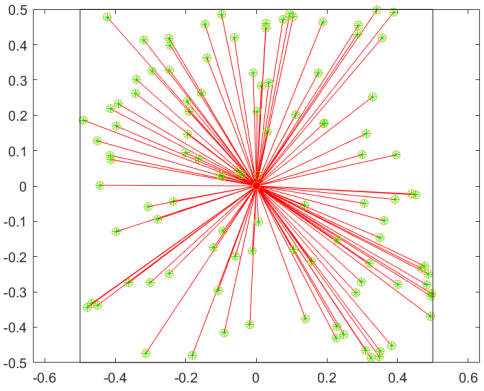

Fig. 11. A 2-D projection of a PRM in 100 dimensions where the selection of $d_{\max}$, caused $\mathbf{q}_0$ to *always* be the nearest neighbor resulting in a star-shaped tree graph.

VI. Conclusion and Future Work

In this paper we explored the behavior of sample-based motion planning algorithms, such as RRT and PRM, in *very* high dimensional hypercube-like configuration spaces. In particular, we saw that a commonly used sampling strategy (component-wise uniform distributions) gives rise to the so-called concentration of measure phenomena, where the expected value of the distance of the samples from the origin is $E(d) = 1/2\sqrt{N/3}$, with exponentially decreasing tail bounds. This suggests that the vertices generated by such algorithms tightly concentrate within a thin-shelled hypersphere of radius $1/2\sqrt{N/3}$, with effectively no vertices near the origin and none near the outer-most corners of the configuration space. Because the inter-vertex distances, $E(d_{ij})$, concentrate tightly as well, the nearest neighbor heuristic commonly used to create potential edges may be less discriminating in high dimensions because many vertices are essentially equidistant from each other – supporting similar observations in the literature [14] suggesting that selecting a vertex at random may be almost as effective. Care must be taken when setting the maximum edge length parameter which appears in many practical implementations of these algorithms – slight changes in the value can cause the graph to change from disconnected to densely connected. Even the value used by PRM* [9] can effectively require many collision checks in high enough dimensions resulting in $O(n)$ run-time. Furthermore, if the initial node is placed at the origin, instead of at random, it is with high probability the nearest neighbor of *every* vertex, creating a star-shaped graph where every vertex is a leaf except the origin.

Acknowledgment

The authors would like to thank the Office of Naval Research (Grant N0001422WX01227) for their financial support; and also Greg Chirikjian, Sertac Karaman and Steven LaValle for fruitful discussions on this topic.

REFERENCES

[1] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, May 2001.

[2] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, p. 566–580, June 1996.

[3] J. Wright and Y. Ma, *High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications*. Cambridge University Press, 2022.

[4] D. J. Smith and M. K. Vamanamurthy, "How small is a unit ball?" *Mathematics Magazine*, vol. 62, no. 2, p. 101, 1989.

[5] S. M. LaValle, M. S. Branicky, and S. R. Lindemann, "On the relationship between classical grid search and probabilistic roadmaps," *International Journal of Robotics Research*, vol. 23, no. (7/8), pp. 673–692, July/August 2004.

[6] J. Kim, J. Esposito, and V. Kumar, "RRT enhancements," *International Journal of Robotics Research*, vol. 15, no. 12, pp. 1257–1272, December 2006.

[7] J. Denny, D. Qin, and H. Zhou, "A fast and approximate medial axis sampling technique," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 10 213–10 219.

[8] S. M. LaValle and J. J. Kuffner, "Rapidly-exploring random trees: Progress and prospects," *Workshop on the Algorithmic Foundations of Robotics*, 2000.

[9] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *International Journal of Robotics Research*, vol. 30, no. 7, pp. 846—894, 2011.

[10] J. Esposito, "Conditional density growth (cdg) model: A simplified model of rapidly exploring random tree growth," *Robotica*, vol. 31, no. 5, pp. 733–746, August 2013.

[11] G. Chirikjian and J. Burdick, "A hyper-redundant manipulator," *IEEE Robotics Automation Magazine*, vol. 1, no. 4, pp. 22–29, 1994.

[12] S. Hirose and H. Yamada, "Snake-like robots [tutorial]," *IEEE Robotics Automation Magazine*, vol. 16, no. 1, pp. 88–98, 2009.

[13] R. Thandiackal, K. Melo, L. Paez, J. Herault, T. Kano, K. Akiyama, F. Boyer, D. Ryczko, A. Ishiguro, and A. J. Ijspeert, "Emergence of robust self-organized undulatory swimming based on local hydrodynamic force sensing," *Science Robotics*, vol. 6, no. 57, p. 6354, 2021.

[14] R. Luna, M. Moll, J. Badger, and L. Kavraki, "A scalable motion planner for high-dimensional kinematic systems," *The International Journal of Robotics Research*, vol. 39, p. 027836491989040, 12 2019.

[15] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.

[16] J. McLurkin, "Measuring the accuracy of distributed algorithms on Multi-Robot systems with dynamic network topologies," *9th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, 2008.

[17] K. Gilpin and D. Rus, "Modular robot systems," *IEEE Robotics Automation Magazine*, vol. 17, no. 3, pp. 38–55, 2010.

[18] J. R. Abella, M. Moll, and L. E. Kavraki, "Maintaining and enhancing diversity of sampled protein conformations in robotics-inspired methods," *Journal of Computational Biology*, Jan 2018.

[19] A. Shkolnik and R. Tedrake, "Path planning in 1000+ dimensions using a task-space voronoi bias," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 2061–2067.

[20] M. Xanthidis, J. M. Esposito, I. Rekleitis, and J. M. O'Kane, "Motion planning by sampling in subspaces of progressively increasing dimension," *Journal of Intelligent and Robotic Systems*, vol. 100, pp. 777–789, 2020.

[21] I. A. Şucan and L. E. Kavraki, "On the performance of random linear projections for sampling-based motion planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, USA, October 2009, pp. 2434–2439.

[22] J. Röwekämper, G. D. Tipaldi, and W. Burgard, "Learning to guide random tree planners in high dimensional spaces," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1752–1757.

[23] S. Arya, D. Mount, R. Silverman, and A. Wu, "An optimal algorithm for approximate nearest neighbor search in fixed dimensions," *Journal of the ACM*, vol. 45, no. 6, pp. 891–923, 1999.

[24] R. S. Anderssen, R. P. Brent, D. J. Daley, and A. Moran, "Concerning $\int_0^1 \cdots \int_0^1 \sqrt{(x_1^2 + ... + x_k^2)} dx_1 ... dx_k$ and a Taylor series method," *SIAM Journal of Applied Math*, vol. 30, pp. 22–30, 1976.

[25] D. Robbins, "Average distance between two points in a box," *American Mathematics Monthly*, vol. 85, no. 278, 1978.