Sampling-based path planning for geometrically-constrained objects

A. Rodríguez, A. Pérez, J. Rosell and L. Basañez

Abstract— **One of the key factors that affect the success and efficiency of sampling-based path planners is the obtention of samples in the more relevant regions of the workspace. This is known as *importance sampling*, and different approaches have already been proposed in this direction. This paper proposes a novel method to bias sampling by means of geometric constraints that reduces the sampling space to sets of lower dimensional submanifolds. These constraints may be imposed by the kinematic structure of the actuation system, by the task specification, or provided by a human user as an intuitive way to include problem knowledge to the planner. The method has been implemented and tested on a probabilistic roadmap planner giving promising results. A variant using a deterministic sampling source is also reported.**

I. INTRODUCTION

Tasks where an object has to be positioned with respect to its surroundings are ubiquitous in robotics, and therefore one of the main challenges in this field is the planning of collision-free paths for an object from a start to a goal configuration in a workspace containing obstacles. Path planning is often performed in the robot's Configuration Space ($\mathcal{C}$-space), where the robot is mapped to a point and the obstacles in the workspace are enlarged accordingly ($\mathcal{C}$-obstacles). The characterization of $\mathcal{C}$-obstacles is a difficult issue, that can be avoided by using sampling-based approaches. These methods consist in the generation of collision-free $\mathcal{C}$-space samples and their subsequent interconnection by free paths, forming either graphs or trees. Graph representations of the sample connectivity are convenient for algorithms that solve multiquery problems like probabilistic roadmap planners (PRM) [1], while tree representations are better suited for algorithms that solve single query problems, like the rapidly-exploring random trees (RRT) [2].

Planning algorithms that use probabilistic sampling have been demonstrated to be probabilistically complete. For a basic PRM, the minimum number of samples necessary to achieve a probability of failure below a certain threshold has been determined in [3]. However, the performance of these algorithms strongly depends on the total number of samples, so it should be kept as low as possible.

Importance sampling strategies have been proposed to increase the density of samples in the more relevant areas of the $\mathcal{C}$-space. These strategies have been classified by [4] into: a) those that bias samples using workspace information [5], [6]; b) those that over-sample the $\mathcal{C}$-space but quickly filter any non-promising configuration [7], [8]; c) those that bias the sampling using the information gathered during the construction of the roadmap or tree [9], [10]; and d) those that deform (dilate) the free $\mathcal{C}$-space to make it more expansive to easily capture its connectivity [11], [12].

In this paper a novel importance sampling method for sampling-based path planners is proposed. The method relies on the imposition of geometric constraints that must be satisfied between a mobile object (whose movements are being planned) and its surroundings. The main motivation for this work has been the observation that in many path planning problems, the mobile object is constrained to a submanifold with lower dimensionality than its embedding space. These constraints may be imposed by the kinematic structure of the actuation system (e.g., a planar manipulator embedded in a three-dimensional workspace), or by the task specification (e.g., to move a bucket filled with a liquid without spilling its contents, or to maintain two parts aligned during an assembly), to name just a few examples.

Geometric constraints provide a straightforward way of specifying constrained movements by explicitly stating the relations that must hold between two or more geometric entities (distances, angles, tangencies, and the like). If the constrained entities are rigid bodies, then the simultaneous satisfaction of a set of geometric constraints yields a submanifold of $SE(3)$ of allowed movements, commonly referred to as a configuration submanifold. Geometric constraint solvers are used to find the map between constraint sets and configuration submanifolds [13].

Planning collision-free movements of a constrained object does not require sampling the whole configuration space, as would be the case for a free-flying object, but rather the regions where the object is allowed to move: its configuration submanifold. If these regions define lower-dimensional subspaces, then higher sample densities and success rates can be achieved for similar sampling efforts, lowering the impact of the *curse of dimensionality*. Furthermore, it will be shown that geometric constraints also provide the human user with a mechanism for including knowledge about the relevant areas of the $\mathcal{C}$-space on an otherwise unconstrained problem.

The paper is structured as follows. Section II presents the geometric constraint solver used to map sets of constraints into configuration submanifolds. Then, the sampling methods for a single and for multiple submanifolds is presented in Section III. Finally, Section IV uses the proposed sampling strategy within a probabilistic roadmap planner and illustrates and evaluates it with some examples. A variant sampling strategy using deterministic sequences is also dis-

This work was partially supported by the CICYT projects DPI2007-63665 and DPI2008-02448.

The authors are with the Institute of Industrial and Control Engineering - Technical University of Catalonia, Barcelona, Spain. Contact: `adolfo.rodriguez@upc.edu`

cussed. Concluding remarks are presented in Section V.

II. THE GEOMETRIC CONSTRAINT SOLVER

Positioning Mobile with respect to Fixed (PMF), is a geometric constraint solver that takes on the problem of finding the configurations of a mobile rigid body that satisfy a set of geometric constraints defined between elements of the body and elements of its environment, which are considered fixed [14]. PMF accepts as input constraints distance and angle relations between points, lines, and planes. The solution methodology exploits the fact that in a set of geometric constraints, the rotational component can often be separated from the translational one and solved independently. This yields a solver that is computationally very efficient, with solution times within the millisecond order of magnitude.

By means of logic reasoning and constraint rewriting, the solver is able to map a broad family of input problems to a few rotational and translational scenarios with known closed-form solution. The solver can handle under-, well-, and overconstrained (redundant or incompatible) problems with multiple solutions. Figs. 1 and 2 depict the different configuration submanifolds to which an object can be restricted with PMF. All combinations between translational and rotational submanifolds are possible. Each configuration submanifold has a known parametric representation, so particular solutions can be represented in the form of a parameterized rigid transformation $T(\mathbf{z})$ that depends on as many parameters as available degrees of freedom (DOF), where $\mathbf{z}$ is the parametric coordinates vector. Consequently, a sweep across the parameter space will span the entire configuration submanifold.

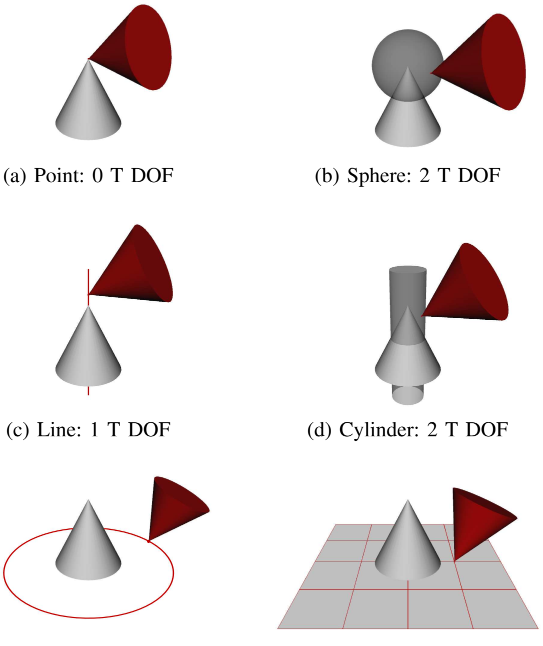
III. SAMPLING THE CONFIGURATION SUBMANIFOLDS

Multiquery planners, like the PRM, have a preprocessing stage that attempts to map the connectivity of the free $\mathcal{C}$-space ($\mathcal{C}_{free}$) onto a roadmap represented as a graph G. The graph vertices represent the configurations sampled from $\mathcal{C}_{free}$ and the edges represent collision-free paths that connect them. Then, in the query phase, the initial and the goal configurations are connected to the roadmap and a path is found using graph search algorithms.

Let N represent the total number of samples that are taken from the $\mathcal{C}$-space, (which includes both free and collision samples). It is well known that the volume of a space increases exponentially with a linear increase in its dimension (i.e., the *curse of dimensionality*), so choosing a value of N that captures sufficiently well the $\mathcal{C}$-space connectivity should take this into consideration. The average sample density c is a parameter that relates the total number of samples N to the dimension of the $\mathcal{C}$-space n according to $N = c^n$. For uniform sampling techniques, c represents the average number of samples per DOF.

A. Sampling a single configuration submanifold

The average sample density c is provided as input to the proposed sampling method, which then uses it to compute the total number of samples N. For the unconstrained spatial scenario, that is, the sampling of six-dimensional $SE(3)$, the total number of samples is $N = c^6$. When constructing the $\mathcal{C}$-space representation of the workspace in the presence of a single constraint scenario, samples are taken only from the configuration submanifold of its associated constraint set, which is a subset of $SE(3)$. To simplify the discussion, it is assumed that both the initial and final configurations of the mobile object are contained in the configuration submanifold. Note that this assumption can be relaxed by adding an extra sampling effort that connects the configuration submanifold to the problem endpoints (e.g., by means of a RRT).



(a) Point: 0 T DOF

(b) Sphere: 2 T DOF

(c) Line: 1 T DOF

(d) Cylinder: 2 T DOF

(e) Circle/ellipse: 1 T DOF

(f) Plane: 2 T DOF

Fig. 1. Translational submanifolds to which the point of an object can be constrained, along with the associated number of translational degrees of freedom (T DOF). One of the two objects is mobile, while the other remains fixed, so for each of the above cases there are two different scenarios, depending on which object is fixed. The case of unconstrained translations (3 T DOF) is not depicted, but is also handled by the solver.

(a) Parallel vectors: 1 R DOF

(b) Angle vectors: 2 R DOF

Fig. 2. Rotational submanifolds to which an object can be constrained, along with the associated number of rotational degrees of freedom (R DOF). The case of unconstrained rotations (3 R DOF) and fully constrained rotations (0 R DOF) are not depicted, but are also handled by the solver.

The interface between the geometric constraint solver and the planner is very simple and generic, and can be extended to other more sophisticated sampling-based planners and importance sampling strategies [15]. When sampling a submanifold, the planner does not need to know the explicit parametric representation of the submanifold, but rather its number of DOF, and the range of values each DOF parameter can take (e.g., rotations about an axis are parameterized with a single variable with values in $[0, 2\pi)$). To obtain a sample in the configuration submanifold, the planner constructs the parametric coordinates vector $\mathbf{z}$ by generating for each $z_i \in \mathbf{z}$ a random value within its valid parameter range, and provides it to the geometric constraint solver that maps it to workspace coordinates via $T(\mathbf{z})$.

Changing either the planner or the geometric constraint solver by a different implementation is straightforward as long as the abovementioned interface is maintained. In fact, the solver could even be substituted by a hardcoded $T(\mathbf{z})$ map, like the direct kinematics of a robot manipulator.

If the configuration submanifold has a dimension $m < 6$, the number of samples needed to achieve a sample density similar to that of the unconstrained scenario is c^m, that is, $c^{(6-m)}$ times less samples.

As an example, consider the scenario depicted in Fig. 3. It is a simplified model of a laparoscopic surgery setup, in which a slender tool enters a cavity (the "patient's" body) through a small opening, and must perform movements while avoiding collisions with obstacles contained in the cavity. The contents of the cavity are assumed to be known (e.g., though some medical imaging technique). In this scenario, the tool is not free to move in any direction, but rather constrained by the opening so that the center of rotation of the tool coincides with the opening centerpoint. This constraint can be modelled as a *point-line* coincidence constraint between the centerpoint of the opening (fixed object) and the tool axis (mobile object). The configuration submanifold associated to this constraint has four DOF: three rotations about the fixed point, and translations along the current direction of the tool axis. The needle could be further constrained by other task-specific constraints, such as following a trajectory with its tip, which would further decrease the dimensionality of its configuration submanifold. This scenario, however, will not be discussed.

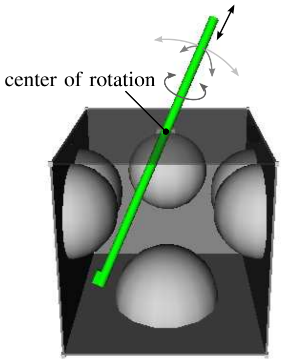
By sampling only on the configuration submanifold, each sample is guaranteed to comply with the constraints imposed by the task, and the available computational power can be used to improve sample density rather than performing a more wasteful blind search through $SE(3)$. For this particular example, the constrained PRM requires c^2 less samples than the basic PRM for a similar sample density (e.g., for $c = 10$, the constrained PRM requires 100 times less samples, or put otherwise, if $N = 10^6$, then $c = 10$ for the basic PRM, and $c = 31.62$ for the constrained PRM).

B. Sampling multiple configuration submanifolds

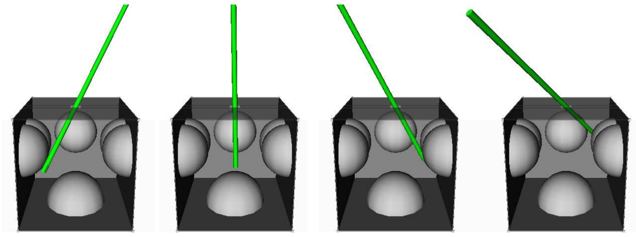
For problems where the movements of the mobile object are represented as a sequence of constrained movements, each taking place in a different configuration submanifold, a connectivity graph G_{con} is constructed between each of the configuration submanifolds (note that G_{con} is not the same graph as the roadmap G).

First, each input constraint set is represented as a vertex in G_{con} labeled with a unique identifier A, B, C, *etc.*, and is solved to obtain its associated configuration submanifold.

Then, the connectivity tests are performed. Two submanifolds are connected if their intersection is not null (otherwise they are disconnected). This occurs when the simultaneous satisfaction of two input constraint sets (e.g., A and B) yields a valid solution. When this is the case, a new vertex labeled as the union of the two input names is inserted in G_{con} (e.g., AB), and is connected with undirected edges to the corresponding vertices (e.g., (A, AB) and (AB, B)).



(a)



(b)

Fig. 3. Simplified model of a laparoscopic surgery setup. The slender tool enters a cavity through a small opening on its top. (a) The tool is constrained so that its axis coincides with the center of the opening. The constrained tool has four DOF (shown as arrows) and a fixed center of rotation. (b) Sequence from a constrained path returned by the planner.

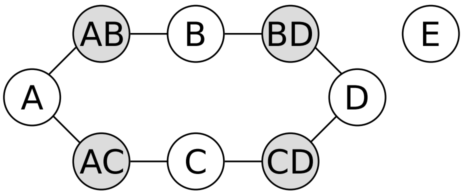


Fig. 4. Example of a configuration submanifold connectivity graph G_{con} constructed for a problem with five input constraint sets, labeled $\{A, B, C, D, E\}$, each represented with a graph vertex. The remaining vertices represent intersections between constraint scenarios. Note that there are two possible ways to connect configurations belonging to vertices $\{A, B, C, D\}$, and that a configuration belonging to E cannot be connected to any of the other vertices.

Finally, an optional test can be performed to verify that the start and goal configurations of the path planning problem are connected by the configuration submanifolds (without considering the obstacles). This is done by verifying that the submanifolds to which these configurations belong lie in the same connected component of G_{con}. Fig. 4 depicts an example graph constructed for a problem with five input constraint sets.

When sampling multiple configuration submanifolds, c^{m_i} samples are taken from each configuration submanifold in G_{con} (there is one submanifold per graph vertex), where m_i

Algorithm 1 Preprocessing stage of the constrained PRM.

```
Require: $c$, $C_{in}$
  $G.vertexSet \leftarrow \varnothing$, $G.edgeSet \leftarrow \varnothing$
  $G_{con}$ = SUBMANIFOLD-CONNECTIVITY($C_{in}$)
  for all $v \in G_{con}.vertexSet$ do
    $m$ = dimension of submanifold contained in $v$
    for all $i = 1$ to $c^m$ do
      $s$ =SAMPLE-SUBMANIFOLD($v$)
      if $s \in \mathcal{C}_{free}$ then
        INSERT($s, G.vertexSet$)
        for all $q \in G.vertexSet$ such that
        $s \neq q$ and $q \in$ NEIGHBORHOOD($s$) do
          if CONNECT($s, q$) then
            INSERT(($s, q$), $G.edgeSet$)
          end if
        end for
      end if
    end for
  end for
```

represents the dimension of the ith submanifold. The total number of samples N is then given by $\sum c^{m_i}$.

Additionally, each sample is labelled with the identifier of its corresponding vertex in G_{con}, so when constructing the roadmap graph G, the neighbors of a free sample s are those free samples that lie within a certain predefined distance d_{neigh} from s and share a common identifier. The latter condition is enforced by means of the set intersection operation. For instance, two neighboring samples s and q with labels A and AB, and such that $d(s, q) < d_{neigh}$, will be connected in G because $A \cap AB = A$.

Section IV-C details an example problem containing multiple configuration submanifolds.

IV. A CONSTRAINT-BASED PROBABILISTIC ROADMAP PLANNER

A. The algorithm

Algorithm 1 details the preprocessing stage for the constrained PRM. It takes as input the average sample density c and the sets of input constraints C_{in}, and outputs a graph G representing the roadmap. Its main functions are:

- Function SUBMANIFOLD-CONNECTIVITY(C_{in}) computes the configuration submanifold connectivity graph G_{con} from the sets of input constraints in C_{in}. Each vertex in G_{con} is associated to a different submanifold.
- Function SAMPLE-SUBMANIFOLD(v) returns a random sample from the submanifold associated to v.
- Function NEIGHBORHOOD(s) returns a set of samples that lie within a predefined distance of sample s and that also share at least one of its identifying labels (i.e., neighboring samples from the same or adjacent submanifolds). Distances are not measured on $SE(3)$, but along the configuration submanifolds.
- Function CONNECT(s, q) tests if a path in $\mathcal{C}$-space that connects the configurations s and q is free or not. The local planner uses an iterative bisection method to collision-check intermediate configurations up to a certain spatial resolution [16]. These intermediate configurations are computed by interpolating in the parameter space of their associated configuration submanifold, rather than directly on $SE(3)$, and therefore satisfy the imposed geometric constraints. Free paths are labeled with a cost equal to the distance between the configurations.

The query phase of the planner uses the A* algorithm with the Euclidean distance heuristic.

In a previous work [17], G_{con} was not computed, so the submanifold intersections were unknown, hence their connectivity. Samples were taken only from the submanifolds associated to the input constraint sets, and labelled accordingly (they had only one identifier, whereas in the current approach they can have one or two). Transitions between submanifolds were handled by multiplying the cost function in CONNECT(s, q) by a large penalizing value when s and q belong to different submanifolds. Although good results were obtained, constraint satisfaction was not guaranteed at the transitions. The present approach guarantees constraint satisfaction all along the solution path.

B. Variant using deterministic sampling

Sampling-based methods usually rely on the use of a random number generation source, although the use of deterministic sampling sequences has been demonstrated to be a good alternative [18]. Deterministic sampling sequences provide an incremental and uniform coverage of $\mathcal{C}$-space, with a better dispersion than random sampling. Deterministic sampling has given slightly better results than random sampling in roadmap planners [19], although such an improvement may be limited to scenarios with few degrees of freedom [4]. Since the constrained PRM samples configuration submanifolds that commonly have less than six degrees of freedom, the use of a deterministic sampling sequence is an alternative that is worth exploring.

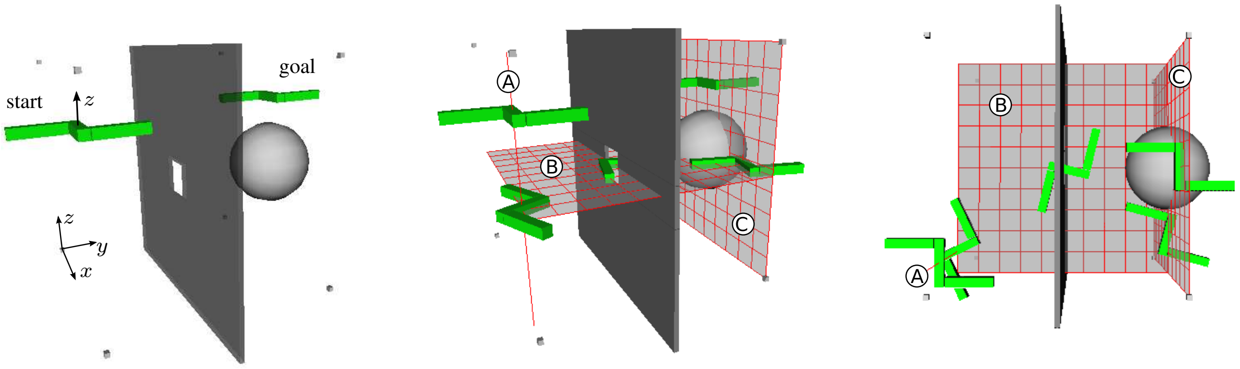
One of the most frequently used deterministic sequences is Halton's [20] due to its good dispersion. However, this sequence lacks a lattice structure for easily computing neighborhood relations, which is a very useful property for path planning purposes. For this reason, the $s_d(k)$ sequence [21], which is based on multiresolution grids was chosen as an alternative to random sampling.

In order to accommodate deterministic sampling into Algorithm 1, it is only necessary to reimplement function SAMPLE-SUBMANIFOLD(v) so that it generates samples according to the $s_d(k)$ sequence instead of randomly.

C. Example problem

The constrained PRM will be exemplified with the problem depicted in Fig. 5a, where an "S" shaped mobile object has to move from a start to a goal configuration by traversing a square hole in a wall and by avoiding a spherical obstacle.

Note that unlike the inherently constrained example of Fig. 3, this problem features a free-flying object that is unconstrained by the task definition. However, constraining the mobile object serves as a mechanism for including user knowledge about the problem into the sampling strategy. This example compares the solutions provided by the basic and



(a) (b) (c)

Fig. 5. (a) The "S" shaped object has to move from a start to a goal configuration by traversing a square hole in a wall and by avoiding a spherical obstacle. (b) and (c) show different instants of a particular solution sequence from a perspective and top view, respectively. The translational submanifolds to which the mobile object centerpoint is being constrained are explicitly shown with their corresponding labels.

constrained PRMs, and shows the tradeoff between a blind search and a search guided by a user-defined decomposition of the search space.

Inspection of the problem reveals that traversing the square hole in the wall is a typical narrow-passage problem that can be formulated and solved in a 3-DOF planar domain. Three input constraint sets are used to characterize the allowed movements of the mobile object and reduce the size of the sampling space: one set for the narrow passage, and the remaining two for connecting the planar problem to the initial and final configurations:

A) Fix the orientation of the mobile object to that of its initial configuration and constrain its centerpoint to a line that passes through the initial centerpoint position and is parallel to the z axis of the workspace (i.e. a 1-DOF submanifold).

B) Constrain a plane of the mobile object that passes though its centerpoint and is parallel to its z axis to a plane passing through the hole centerpoint and whose normal is parallel to the z axis of the workspace (i.e. a 3 DOF submanifold).

C) Maintain the z axis of the mobile object parallel to the z axis of the workspace, and constrain its centerpoint to a plane that is parallel to the wall and contains the goal configuration (i.e. a 3 DOF submanifold).

Fig. 6 depicts the connectivity graph G_{con} of the configuration submanifolds associated to the problem. Vertices A, B, and C correspond to the above constraint sets. Vertex AB represents the point intersection of constraint sets A and B, and has 0 DOF. Vertex BC represents the line intersection of constraint sets B and C, and has 2 DOF (translations along the line and rotations about the z axis). Figs. 5b and 5c depict different instants of a particular solution sequence from a perspective and top view, respectively. The accompanying video animates this solution sequence.

The total number of samples that must be obtained for this problem evaluates to:

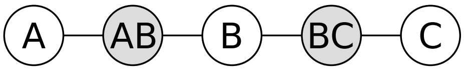


Fig. 6. Connectivity graph G_{con} of the configuration submanifolds for the problem depicted in Fig. 5.

$$\begin{aligned} N &= \sum c^{m_i} = \underbrace{c}_{A} + \underbrace{1}_{AB} + \underbrace{c^3}_{B} + \underbrace{c^2}_{BC} + \underbrace{c^3}_{C} \\ &= 2c^3 + c^2 + c + 1. \end{aligned}$$

Table I lists the number of samples N that must be taken for different sample density values c for the constrained PRM subject to the abovementioned user-defined constraint sets, and compares them to the basic (unconstrained) PRM. It can be seen that for the reported values of c, the constrained PRM requires a number of samples that is between two and three orders of magnitude smaller than what the basic PRM does. This is very meaningful considering that the average success rate of a PRM that uniformly samples the $\mathcal{C}$-space strongly depends on the value of c.

Fig. 7 shows the average success rate of the constrained PRM for different values of c, as well as a normalized measure of the average execution time. It can be seen that success rates in excess of 95% can be achieved with relatively modest sample counts ($N < 5000$). Both random and deterministic sampling strategies were tested, and the latter improved slightly the results of the planner while involving a very small extra computational effort. The improvements from the use of a deterministic sampling sequence become noticeable for $c > 8$, reaching a maximum success rate increase of 8.9% for $c = 9$.

To contrast these results, the average success rate of an unconstrained PRM with $N = 4577$ was 2.5%. This number of samples yields a sample density $c = 4.08$, whereas in the constrained scenario with deterministic sampling the density value rises to $c = 13$ and the success rate to over 98%, for equal N. The effect of reducing the sampling space size by constraining the mobile object has a great effect on the success rate that can even be slightly improved by resorting

TABLE I
COMPARISON BETWEEN THE CONSTRAINED PRM AND THE BASIC PRM FOR THE EXAMPLE DEPICTED IN FIG.5.

c	N for the constrained PRM	N for the basic PRM
	$N = 2c^3 + c^2 + c + 1$	$N = c^6$
7	743	117649
8	1097	262144
9	1549	531441
10	2111	1000000
11	2795	1771561
12	3613	2985984
13	4577	4826809

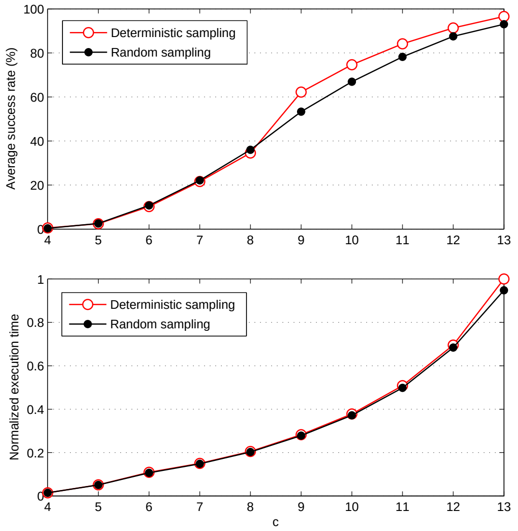


Fig. 7. Average success rate and average execution times as a function of the average sample density c, for solving the example of Fig.5 using the constrained PRM. The average execution time has been normalized to the $[0, 1]$ interval.

to deterministic sampling sequences.

It is worth mentioning that the time required by the geometric constraint solver to compute the map from constraint sets to configuration submanifolds is many orders of magnitude smaller than the time required by the PRM to perform the preprocessing stage, and thus can be considered to add a negligible computational overhead.

V. CONCLUSIONS

This paper proposes an *importance sampling* method for sampling-based path planners that bias the samples towards submanifolds of the configuration space that are relevant to the task. These submanifolds can be easily described in terms of geometric constraint sets that are imposed by the kinematics of the actuation system, by the task specification, or as a way of conveying user problem knowledge to the planner. The method has been shown to be simple and computationally efficient, and relies on reducing the dimensionality of the sampling space to substantially reduce the number of samples required for a given success rate. Experiments carried out on a probabilistic roadmap planner show that high success rates can be achieved with sample counts and execution times that are orders of magnitude less than what would be required by an unconstrained PRM. Both random and deterministic sampling strategies have been tested, showing that the latter can further increase the success rate of the planner on low-dimensional submanifolds.

REFERENCES

[1] L. E. Kavraki and J.-C. Latombe, "Randomized preprocessing of configuration for fast path planning," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, vol. 3, 1994, pp. 2138–2145.

[2] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2000, pp. 995–1001.

[3] L. E. Kavraki, M. N. Kolountzakis, and J.-C. Latombe, "Analysis of probabilistic roadmaps for path planning," *IEEE Trans. on Robotics and Automation*, vol. 14, no. 1, pp. 166 –171, Feb. 1998.

[4] D. Hsu, J.-C. Latombe, and H. Kurniawati, "On the probabilistic foundations of probabilistic roadmap planning," *Int. Journal of Robotics Research*, vol. 25, no. 7, pp. 627 – 643, 2006.

[5] J. P. van der Berg and M. H. Overmars, "Using workspace information as a guide to non-uniform sampling in probabilistic roadmap planners," *Int. J. of Robotics Res.*, vol. 24 (12), pp. 1055–1071, 2005.

[6] H. Kurniawati and D. Hsu, "Workspace-based connectivity oracle: An adaptive sampling strategy for PRM planning," in *Algorithmic Foundations of Robotics VII*, S. Akella and et.al., Eds. Springer–Verlag, 2006.

[7] V. Boor, M. H. Overmars, and A. F. van der Stappen, "The Gaussian sampling strategy for probabilistic roadmap planners," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1999, pp. 1018–1023.

[8] D. Hsu, T. Jiang, J. Reif, and Z. Sun, "The bridge test for sampling narrow passages with probabilistic roadmap planners," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2003, pp. 4420–4426.

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

[10] D. Hsu, G. Sanchez-Ante, and Z. Sun, "Hybrid PRM sampling with a cost-sensitive adaptive strategy," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 3874 – 3880.

[11] M. Saha, J. C. Latombe, Y. C. Chang, and F. Prinz, "Finding narrow passages with probabilistic roadmaps: The small-step retraction method," *Autonomous robots*, vol. 19(3), pp. 301–319, 2005.

[12] H. L. Cheng, D. Hsu, J. C. Latombe, and G. Sanchez-Ante, "Multi-level free space dilation for sampling narrow passages in prm planning," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2006, pp. 1255– 1260.

[13] C. Hoffmann and R. Joan-Arinyo, "A brief on constraint solving," *CAD&A*, vol. 2, no. 5, pp. 655–664, 2005.

[14] A. Rodríguez, L. Basañez, and E. Celaya, "A relational positioning methodology for robot task specification and execution," *IEEE Trans. Robot.*, vol. 24, no. 3, pp. 600–611, 2008.

[15] X. T. Shawna Thomas, Marco Morales and N. M. Amato, "Biasing samplers to improve motion planning performance," in *Proc. of Int. Conf. on Robotics and Automation*, 2007, pp. 1625–1630.

[16] R. Geraerts and M. H. Overmars, "Sampling and node adding in probabilistic roadmap planners," *Robotics and Autonomous Systems*, vol. 54, no. 2, pp. 165–173, 2006.

[17] A. Pérez, A. Rodríguez, J. Rosell, and L. Basañez, "A constraint-based probabilistic roadmap planner," in *40th International Symposium on Robotics*, 2009.

[18] M. S. Branicky, S. M. LaValle, K. Olson, and L. Yang, "Quasi-randomized path planning," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2001, pp. 1481–1487.

[19] S. M. LaValle, M. S. Branicky, and S. R. Lindemann, "On the relationship between classical grid search and probabilistic roadmaps," *Int. Journal of Robotics Research*, vol. 23, no. 7-8, pp. 673–692, 2004.

[20] J. Halton, "On the effciency of certain quasi-random sequences of points in evaluating multi-dimensional integrals," *Numer. Math.*, vol. 2, pp. 84–90, 1960.

[21] J. Rosell, M. Roa, A. Pérez, and F. García, "A general deterministic sequence for sampling d-dimensional configuration spaces," *J. of Intelligent and Robotic Systems*, vol. 50, no. 4, pp. 361–374, 2007.