

# A Constraint-based Probabilistic Roadmap Planner <sup>★</sup>

A. Pérez , A. Rodríguez, J. Rosell and L. Basañez <sup>\*</sup>

*<sup>\*</sup> Institute of Industrial and Control Engineering  
Technical University of Catalonia, Barcelona, Spain.*

---

**Abstract:** The success of sampling-based methods to solve path-planning problems strongly relies in the sampling process, i.e. the performance in terms of number of samples needed and computational time spent strongly depends on the configurations sampled. This paper proposes an *importance sampling method* based on geometric constraints that only samples certain submanifolds of the configuration space where a mobile object is allowed to move, thus reducing the size of the search space and increasing the density of samples in the regions of interest. The proposed sampling method is used in a probabilistic roadmap planner, giving promising results.

---

## 1. 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 usually 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 samples of  $\mathcal{C}$ -space and in their interconnection with free paths, forming either roadmaps (PRM, Kavraki and Latombe [1994]) or trees (RRT, Kuffner and LaValle [2000]). PRM planners are conceived as multi-query planners, while RRT planners are developed to rapidly solve a single-query problem.

Sampling-based methods that use probabilistic sampling have been demonstrated to be probabilistic complete, e.g. for the basic PRM method the number of samples necessary to achieve a probability of failure below a given threshold has been determined by Kavraki et al. [1998]. Nevertheless, for performance purposes, this number should be reasonably low, and therefore care must be taken in the generation of samples and their interconnection.

In this line, importance sampling strategies are proposed to increase the density of samples in critical areas of the  $\mathcal{C}$ -space. These strategies have been classified by Hsu et al. [2006] into: a) those that bias samples using workspace information (e.g. van der Berg and Overmars [2005], Kurniawati and Hsu [2006]); b) those that over-sample the  $\mathcal{C}$ -space but quickly filter any not-promising configuration (e.g. Boor et al. [1999], Hsu et al. [2003]); c) those that

bias the sampling using the information gathered during the construction of the roadmap or tree (e.g. Kavraki et al. [1996], Hsu et al. [2005]); and d) those that deform (dilate) the free  $\mathcal{C}$ -space to make it more expansive to easily capture its connectivity (e.g. Saha et al. [2005], Cheng et al. [2006]).

In this paper we propose a novel importance sampling method for a PRM planner based on geometric constraints. Oftentimes, the positioning of an object with respect to its surroundings can be decomposed into a series of constrained movements which do not require using the six degrees of freedom (DOF) a rigid body has in free space. Geometric constraints provide a straightforward way of specifying these movements, by explicitly stating the relations (distances, angles, tangencies, and the like) that must hold between two or more geometric entities. If the constrained entities are rigid bodies, then the simultaneous satisfaction of a set of geometric constraints yields a submanifold of  $SE(3)$ <sup>1</sup> of allowed movements, commonly referred to as a configuration submanifold. Provided the availability of sets of user-defined constraints associated to a given task, a geometric constraint solver can be used to find the map between them and the configuration manifolds [Hoffmann and Joan-Arinyo, 2005]. Then, configurations belonging to these submanifolds can be sampled and used in a PRM, giving a better performance in terms of number of required samples and computation time.

The paper is structured as follows. Section 2 first describes the geometric constraint solver used. Then, section 3 presents the proposed approach, which is illustrated and evaluated in Section 4. Finally, section 5 gives the conclusions of the work.

## 2. PMF 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

---

<sup>★</sup> This work was partially supported by the CICYT projects DPI2007-63665 and DPI2008-02448. A. Pérez is also with the Escuela Colombiana de Ingeniería "Julio Garavito" placed in Bogotá D.C., Colombia.

<sup>1</sup>  $SE(3)$  stands for the group of rigid motions of  $\mathbb{R}^3$ , and  $SO(3)$  for the group of positive rotations of  $\mathbb{R}^3$ .

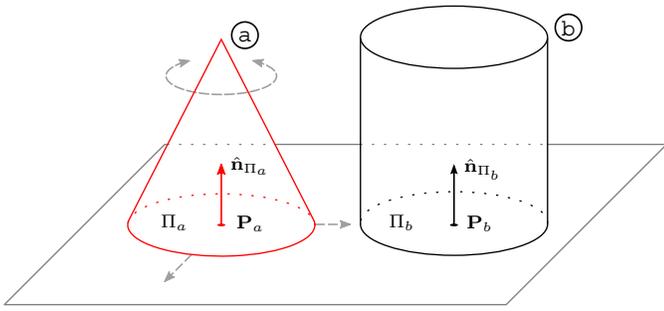


Fig. 1. The mobile object  $a$  is constrained with respect to the fixed object  $b$  according to the *plane-plane* coincidence constraint  $\Pi_a = \Pi_b$ . The available DOF of  $a$  (one rotation, two translations) are shown as dashed lines.

satisfy a set of geometric constraints defined between elements of the body and elements of its environment, which are considered fixed [Rodríguez et al., 2008]. PMF accepts as input constraints distance ( $d$ ) and angle ( $\angle$ ) 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 <sup>2</sup>.

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. Table 2 lists the different configuration submanifolds to which a mobile 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  $H(\mathbf{z})$  that depends on as many parameters as available DOF, where  $\mathbf{z}$  is the parametric coordinates vector. Consequently, a sweep across the parameter space will span the entire solution submanifold.

As a simple example, consider the objects  $a$  and  $b$  of Fig. 1. Constraining the mobile object  $a$  with respect to the fixed object  $b$  according to the *plane-plane* coincidence constraint  $\Pi_a = \Pi_b$  restricts three of  $a$ 's DOF. The remaining DOF (two translational, one rotational), are shown in Fig. 1 with dashed lines. The constraint  $\Pi_a = \Pi_b$  can be decomposed into the pure rotational *parallelism* constraint  $\hat{\mathbf{n}}_{\Pi_a} \parallel \hat{\mathbf{n}}_{\Pi_b}$  and the *point-plane* contained constraint  $\mathbf{P}_a \subset \Pi_b$ , where  $\hat{\mathbf{n}}$  represents a plane normal and  $\mathbf{P}_a$  is a point contained in  $\Pi_a$ . The transformation  $H(\mathbf{z})$ , when applied to the initial (and in general, non constraint-satisfying) configuration of  $a$ , takes it to any of the infinitely possible constraint-satisfying configurations. The particular solution is determined by the values of the parameter vector  $\mathbf{z}$ . The form of  $H(\mathbf{z})$ , as computed by the

<sup>2</sup> Computation times were measured on a Pentium 4 processor with a 3.4GHz CPU clock.

Table 1. Translational and rotational configuration submanifolds to which a mobile object can be restricted.

Translational submanifold	DOF
$\mathbb{R}^3$	3
Plane	2
Sphere	2
Cylinder	2
Line	1
Ellipse	1
Point	0
Rotational submanifold	DOF
$SO(3)$	3
Vectors at an angle	2
Parallel vectors	1
Fixed rotation	0

solver, and expressed as a homogeneous transformation, has the form:

$$H(\mathbf{z}) = \begin{bmatrix} \mathbf{R}(z_1) & \mathbf{T}(z_1, z_2, z_3) \\ 0 & 1 \end{bmatrix} \quad (1)$$

The rotational component is computed as

$$\mathbf{R}(z_1) = \mathbf{R}(\hat{\mathbf{n}}_{\Pi_b}, z_1) \mathbf{R}(\hat{\mathbf{u}}, \sigma) \quad (2)$$

where  $\mathbf{R}(\hat{\mathbf{n}}_{\Pi_b}, z_1)$  represents the free rotation by an angle of  $z_1$  about vector  $\hat{\mathbf{n}}_{\Pi_b}$ , and  $\mathbf{R}(\hat{\mathbf{u}}, \sigma)$  is a fixed rotation that enforces the rotational constraint, where  $\hat{\mathbf{u}} = \hat{\mathbf{n}}_{\Pi_a} \times \hat{\mathbf{n}}_{\Pi_b}$  and  $\sigma$  is the angle between  $\hat{\mathbf{n}}_{\Pi_a}$  and  $\hat{\mathbf{n}}_{\Pi_b}$ .

The translational component is given by

$$\mathbf{T}(z_1, z_2, z_3) = \mathbf{Q}_b(z_2, z_3) - \mathbf{R}(\mathbf{P}_a) \quad (3)$$

where  $\mathbf{Q}_b(z_2, z_3) = \mathbf{P}_b + z_2 \hat{\mathbf{d}}_2 + z_3 \hat{\mathbf{d}}_3$  represents any point contained in  $\Pi_b$ , and  $\mathbf{R}(\mathbf{P}_a)$  is the current rotational component solution, as computed in (2), applied to the mobile point. Vectors  $\hat{\mathbf{n}}_{\Pi_b}$ ,  $\hat{\mathbf{d}}_2$ , and  $\hat{\mathbf{d}}_3$  are orthogonal. Notice that the translational component is function of the rotational component.

### 3. THE PROPOSED APPROACH

#### 3.1 Sampling the configuration submanifold

When constructing the  $\mathcal{C}$ -space representation of the workspace, samples are taken only from the configuration submanifolds of the associated constraint sets. The proposed constraint-based PRM does not need to know the explicit parametric representation of a 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 PRM 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  $H(\mathbf{z})$ .

The total number of samples  $N$  generated by the PRM is set from the value  $c$  of the average sample density per DOF. For the unconstrained scenario, that is, the sampling of six-dimensional  $SE(3)$ ,  $N = c^6$ . Assuming that a path from the initial to the final configuration of the mobile

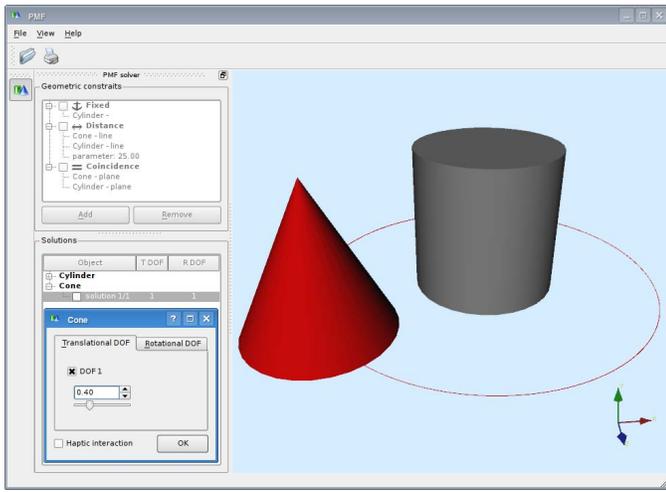


Fig. 2. Screenshot of the PMF geometric constraint solver user interface.

object exists for a single configuration submanifold with dimension  $m < 6$ , the number of samples needed to connect them is  $c^m$ , that is,  $c^{(6-m)}$  times lower than the unconstrained scenario. For instance, setting  $c = 10$  and  $m = 2$  yields  $c^{(6-m)} = 10^4$ , a difference of four orders of magnitude for the same sample density.

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, there exist as many constraint sets as configuration submanifolds. In these cases, the total number of samples  $N$  is given by  $\sum c^{m_i}$ , where  $m_i$  represents the dimension of the  $i$ th configuration submanifold.

Constraint scenarios can be intuitively created by means of the constraint solver user interface, depicted in Fig. 2. Once created, they can be encoded into *XML* format for persistent storage purposes, and retrieved at any later time by a decoder implemented in the path planning program. The following example represents an *XML* description of a constraint set, containing a single *point-plane* coincidence constraint:

```
<PMF>
  <ConstraintSet>
    <Constraint>
      <MobileElement type="point" X="2.0" Y="2.0" Z="2.0">
        Point coordinates
      </MobileElement>
      <FixedElement type="plane" DX="0.0" DY="0.0" DZ="1.0"
        X="0.0" Y="0.0" Z="0.0">
        Plane coordinates (normal vector + point)
      </FixedElement>
      <ConstraintType type="coincidence">
        Constraint type
      </ConstraintType>
    </Constraint>
  </ConstraintSet>
</PMF>
```

### 3.2 Connecting the samples

To test if a rectilinear path in  $\mathcal{C}$ -space that connects two configurations  $q_1$  and  $q_2$  is free or not, the local planner uses an iterative bisection method to collision-check the intermediate configurations up to a certain spatial resolution.

When these configurations are free, the path is labeled with a cost computed as follows:

$$Cost = distance(q_1, q_2) * weight(q_1, q_2) \quad (4)$$

where  $distance(q_1, q_2)$  is a length measure between  $q_1$  and  $q_2$ , and  $weight(q_1, q_2)$  evaluates to 1 when  $q_1$  and  $q_2$  belong to the same configuration submanifold and evaluates to a big penalizing value otherwise. In this way, transitions between submanifolds occur in the neighborhood of their intersections, and are only pursued when they are necessary to reach the goal configuration.

The  $distance(q_1, q_2)$  function in (4) measures distances between configurations in  $SE(3)$ . There does not exist a bi-invariant (intrinsic) metric in  $SE(3)$ , however, both left- and right-invariant metrics can be defined [Belta and Kumar, 2002, Park, 1995, Zefran and Kumar, 1996, Kuffner, 2004]. Any metric in  $SE(3)$  ultimately depends on a choice of length scale. The left-invariant metric proposed in Park [1995] has been chosen, since its computation is fast and does not require an iterative procedure. According to it, the distance between two configurations  $q_1$  and  $q_2$  that share a common reference frame can be computed as

$$distance(q_1, q_2) = \sqrt{\phi^2 + \left(\frac{\Delta}{L}\right)^2} \quad (5)$$

where  $\phi$  and  $\Delta$  are respectively the rotational and translational distances between  $q_1$  and  $q_2$ , and  $L$  scales the contribution of the translational and rotational components. Let  $R$  and  $T$  represent the rotational and translational components of a sample configuration  $q$ , then

$$\Delta = |T_b - T_a| \quad (6)$$

$$\phi = \cos^{-1} \left( \frac{tr(R_a^{-1} R_b) - 1}{2} \right) \quad (7)$$

where  $tr(\cdot)$  represents the matrix trace, i.e., the sum of its diagonal terms.

### 3.3 The planning algorithm

A basic probabilistic roadmap planner is a multiquery planner that, in a preprocessing stage, attempts to map the connectivity of the free  $\mathcal{C}$ -space ( $\mathcal{C}_{free}$ ) onto a roadmap represented as a graph in which the vertices are configurations sampled from  $\mathcal{C}_{free}$  and the edges are collision-free paths that connect these configurations. 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.

Algorithm 1 details the preprocessing stage of a basic PRM, where a graph  $G$  representing the roadmap is constructed. The input is the number  $N$  of configurations to sample. The finding of collision-free paths that connect two edges (represented by the function `CONNECT` in the algorithm) is usually done by a simple and quick local planner.

The proposed constraint-based PRM uses the basic PRM modified as follows:

- Function `GET-RANDOM-SAMPLE()` now iteratively gets a sample from each of the configuration submanifolds

**Algorithm 1** Basic algorithm for the preprocessing phase of a basic PRM.

**BASIC PRM( $N$ )**

```

 $G.vertexSet \leftarrow \emptyset, G.edgeSet \leftarrow \emptyset, i \leftarrow 0$ 
For  $i = 1$  to  $N$  do:
   $s = \text{GET-RANDOM-SAMPLE}()$ 
  If  $s \in \mathcal{C}_{free}$  then
    INSERT( $s, G.vertexSet$ )
    ForAll  $q \in G.vertexSet \mid s \neq q$  and  $q \in \text{NEIGHBORHOOD}(s)$  do
      If CONNECT( $s, q$ ) then
        INSERT( $(s, q), G.edgeSet$ )
      End If
    End For
  End If
End For
RETURN  $G$ 

```

as explained in Section 3.1. This function maintains the sample density  $c$  value constant and equal in all the sampled submanifolds.

- Function CONNECT labels free edges as detailed in Section 3.2.

The implementation of the basic PRM and its extension proposed in the present paper has been done using the following tools: The *PQP* library for the collision detection process, the *Boost Graph Library* to manage the graphs (including the query phase using the A\* algorithm), and *Coin3D* and *Qt* for the graphic rendering and Graphical User Interface (GUI), respectively. An overview about the way to interact with these tools can be found in [Pérez and Rosell, 2008]. Objects are geometrically described by *VRML* models and the configuration of each object in the scene is handled by a custom *XML* file. An additional custom *XML* file contains the information regarding the sets of geometric constraints (Section 3.1).

#### 4. EXAMPLES AND EVALUATION

The proposed approach will be exemplified with the problem depicted in Fig. 3(a), 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.

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 constraint sets are used to characterize the allowed movements of the mobile object and reduce the size of the search space: one set for the narrow passage, and the remaining two for connecting the planar problem to the initial and final configurations:

- Fix the orientation of the mobile object to that of its initial configuration and constrain its centerpoint to a line that is parallel to the  $z$  axis (i.e. a 1-DOF submanifold).
- Constrain the horizontal plane of the mobile object to coincide with a plane parallel to the  $z$  axis passing through the hole centerpoint (i.e. a 3 DOF submanifold).

$c$	$N$ for the constraint-based PRM $N = c + 2c^3$	$N$ for the basic PRM $N = c^6$
6	438	46656
7	693	117649
8	1032	262144
9	1467	531441
10	2010	1000000
11	2673	1771561
12	3468	2985984
13	4407	4826809

Table 2. Comparison between the number of samples  $N$  that must be taken by the constraint-based PRM and the basic PRM for the example depicted in Fig.3.

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

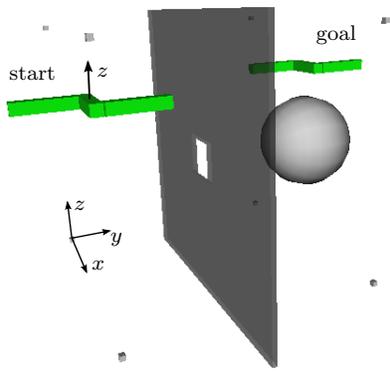
Figs.3(b) and 3(c) depict different instants of a particular solution sequence from a perspective and top view, respectively.

The number of samples that must be obtained for this problem evaluates to  $N = \sum c^{m_i} = c + 2c^3$ . Table 2 lists the number of samples  $N$  that must be taken for different sample density values  $c$  for the constraint-based PRM subject to the abovementioned constraint sets, and compares them to the basic (unconstrained) PRM. It can be seen that the constraint-based PRM requires a number of samples that is two orders of magnitude smaller compared to the basic PRM. This is very meaningful considering that the average success rate of a PRM without any kind of sampling bias strongly depends on the value of  $c$ . Fig. 4 shows the average success rate of the constraint-based PRM for different values of  $c$ , as well as a normalized measure of the average solution time. The convex shape of the solution time curve is due to the fact that the total number of samples  $N$  is a polynomial function of  $c$ . Also, for higher values of  $N$ , the connectivity of the graph representing the roadmap increases, as does the graph analysis phase needed to construct it. To contrast these results, the average success rate of an unconstrained PRM with  $N = 4407$  was lower than 2.5%. This number of samples yields a sample density  $c = 4.05$ , whereas in the constrained scenario the density value rises to  $c = 13$  and the success rate to over 98%, for equal  $N$ .

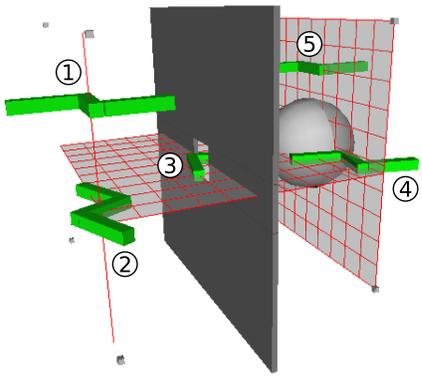
The time required by the geometric constraint solver to compute the map from constraint sets to configuration submanifolds was many orders of magnitude smaller than the time required by the PRM to find a collision-free path, and thus can be considered to add a negligible computational overhead.

#### 5. CONCLUSION

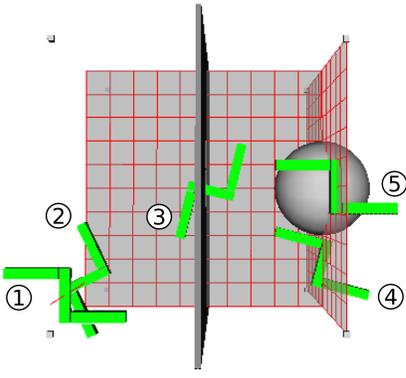
Sampling-based methods based on probabilistic sampling have the interesting property of being probabilistic complete. Nevertheless, importance sampling methods are required in order to solve a problem with as few samples as possible. In this line, this paper has proposed (for the path planning of an object) a simple, efficient, and intuitive method to bias the sampling using geometric constraints.



(a)



(b)



(c)

Fig. 3. (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 line and planes to which the mobile object centerpoint is being constrained are explicitly shown.

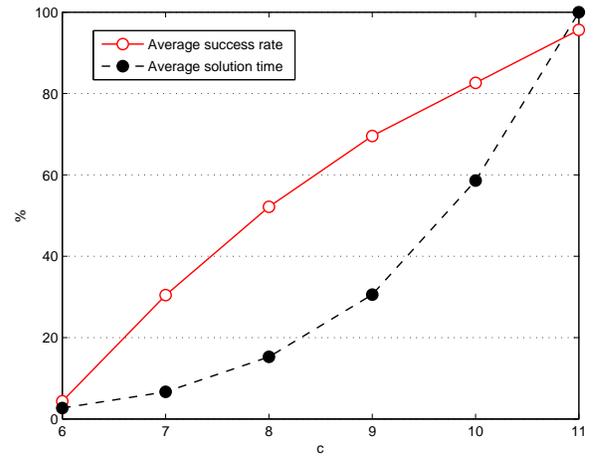


Fig. 4. Performance metrics for the solution of the example of Fig.3 using the constraint-based PRM. Parameter  $c$  stands for the average number of samples per DOF, and the average solution time has been normalized to the  $[0, 100]$  interval.

These constraints are defined between the mobile object and the obstacles in its environment, and reduce the sampling space to the submanifolds of the configuration space that satisfy the constraints. This results in a considerable reduction of the computational effort required to build the roadmap. The method has been tested on simulated examples yielding promising results.

Future work goes towards the obtaining of a solution path that guarantees the satisfaction of the constraints all along it. This requires: a) the interpolation of samples to be done in the parameter space; b) the distance be measured over the submanifolds; and c) the transition between submanifolds be done through configurations sampled over their intersection. Also the use of a deterministic sampling source is under consideration.

## REFERENCES

C. Belta and V. Kumar. Euclidean metrics for motion generation on  $SE(3)$ . *Proceedings of the I MECH E Part C Journal of Mechanical Engineering Science*, 216:47–60, January 2002.

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*, pages 1018–1023, 1999.

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*, pages 1255–1260, 2006.

C.M. Hoffmann and R. Joan-Arinyo. A brief on constraint solving. *CAD&A*, 2(5):655–664, 2005.

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*, pages 4420–4426, 2003.

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*, pages 3874 – 3880, 2005.
- D. Hsu, J.-C. Latombe, and H. Kurniawati. On the probabilistic foundations of probabilistic roadmap planning. *Int. Journal of Robotics Research*, 25(7):627 – 643, 2006.
- 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*, volume 3, pages 2138–2145, 1994.
- 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*, 12(4):566–580, August 1996.
- L. E. Kavraki, M. N. Kolountzakis, and J.-C. Latombe. Analysis of probabilistic roadmaps for path planning. *IEEE Trans. on Robotics and Automation*, 14(1):166 – 171, Feb. 1998.
- J. J. Kuffner. Effective Sampling and Distance Metric for 3D Rigid Body Path Planning. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2004.
- 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*, pages 995–1001, 2000.
- H. Kurniawati and D. Hsu. Workspace-based connectivity oracle: An adaptive sampling strategy for PRM planning. In S. Akella and et.al., editors, *Algorithmic Foundations of Robotics VII*. Springer-Verlag, 2006.
- F. C. Park. Distance metrics on the rigid-body motions with applications to mechanical design. *Journal of Mechanical Design*, 117:48–54, 1995.
- A. Pérez and J. Rosell. A roadmap to robot motion planning software development. *Accepted to Computer Applications in Engineering Education*, 2008.
- A. Rodríguez, L. Basañez, and E. Celaya. A relational positioning methodology for robot task specification and execution. *IEEE Trans. Robot.*, 24(3):600–611, 2008.
- M. Saha, J. C. Latombe, Y. C. Chang, and F. Prinz. Finding narrow passages with probabilistic roadmaps: The small-step retraction method. *Autonomous robots*, 19(3):301–319, 2005.
- 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.*, 24 (12):1055–1071, 2005.
- M. Zefran and V. Kumar. Planning of smooth motions on SE(3). *IEEE Int. Conf. Robot. Automat.*, pages 121–126, April 1996.