# Derandomizing probabilistic roadmap methods

Abraham Sánchez L.[1] and René Zapata.[1]

LIRMM, UMR5506 CNRS, 166 rue Ada 34392,
Montpellier Cedex 5, France
{asanchez, zapata}@lirmm.fr

**Abstract.** We propose the use of quasi-random sampling techniques for motion planning in high dimensional configuration spaces, within the popular probabilistic roadmap (PRM) framework. We show through several experiments, some performance advantages of QPRM approach, in comparison to its randomized counterpart.

## 1 Introduction

Automated motion planning is rapidly gaining importance in various fields. Originally the problem was studied in robotics. But in the past few years many new applications arise in fields such as animation, computer games, virtual environments and maintenance planning and training in industrial CAD systems.

Motion planning algorithms attempt to find a sequence of actions to move a system from an initial to a goal configuration. The seminal paper by Schwarthz and Sharir on the *generalized piano mover's problem* sparked a great deal of research efforts aimed at finding efficient solutions for a wide class of problems [1].

Many different techniques for motion planning have been devised. See the book of Latombe [2] for an extensive overview of the situation up to 1991 and e.g. the proceedings of the yearly IEEE International Conference on Robotics and Automation for many more recent results.

Over two decades of motion planning research have led to two primary trends. In 80s, deterministic approaches provided both elegant, complete algorithms for solving the problem, and also useful approximate or incomplete algorithms. The curse of dimensionality due to high-dimensional configuration spaces motivated researchers from the 90s to present time to develop randomized approaches which are incomplete, but capable of efficiently solving many high-dimensional problems.

A similar pair of trends occurred many years ago with the Monte Carlo methods. These trends were followed by a third trend: the development of quasi-random approaches that use deterministic samples to achieve performance that is often superior to random sampling. In this paper, the term quasi-random can be considered synonymous with deterministic; the term exist to emphasize comparisons with random and pseudo-random concepts. Quasi-random sampling ideas have improved computational methods in many areas, including numerical integration, optimization, image processing, and computer graphics [3].

It is therefore natural to ask: Can quasi-random sampling ideas also improve motion planning methods designed for high degrees of freedom? Is randomization really the key to solving high-dimensional problems? We argue in this work that randomization is not necessarily advantageous in solving high-dimensional planning problems.

In this paper, we investigate the use of deterministic sampling in the context of the popular PRM framework developed independently at different sites [4], [5], [6], [7]. We present implemented, quasi-random variant of the classical PRM and indicate some advantages over its randomized counterpart.

## 2 Sampling methods

Deterministic sampling techniques have been developed by numerous mathematicians over the past century. Excellent overviews of the subject include [8], [3]. A brief treatment, specific to our problem is presented here.

Let $X = [0,1]^d \subset \Re^d$ define a space over which to generate samples. Consider designing a set, $P$, of $N$ $d$-dimensional sample points in a way that covers $X$ uniformly in some sense. In the one-dimensional case, the points may be evenly spaced in an obvious, uniform way. There are both the challenges of defining a useful criterion of uniformity and then designing a sample set that attempts to optimize the criterion. A good criterion should mesure whether the sample points appear reasonable in various region of $X$. For example, PRM algorithm repeatedly tries to connect to samples in a randomly-centered ball in $X$. It would be useful if the criterion measures how many samples will fall into these neighborhoods.

There are a variety of measures that can be used to characterize the nonuniformity of the sample points within the hypercube, however, the dominant methods as referred to as the *discrepancy* measures of the sample. We define the discrepancy to measure how far from ideal the point set $P$ as:

$$D_N(P, \mathcal{R}) = \sup_{R \in \mathcal{R}} \left| \frac{P \cap R}{N} - \mu(R) \right| \tag{1}$$

where $\mu(R)$ denotes the Lebesgue measure (or volume) on $X^d$ and the supremum is taken ove all axis-parallel boxes, $R$.

Whereas discrepancy is based on measure, a metric-based criterion, called *dispersion*, can be introduced:

$$d_N(P, \rho) = \sup_{x \in X} \min_{p \in P} \rho(x, p) \tag{2}$$

Above $\rho$ denotes any metric, such as Euclidean distance or $\ell^\infty$. If $\rho$ is a Euclidean metric, the dispersion yields the radius of the largest empty ball. If $\mathcal{R}$ represents the set of all balls, then $D_N$ is at least as large as this volume because $|P \cap R| = 0$. It is kwnon that $d_N(P, \rho) \leq D_N(P, \mathcal{R})^{\frac{1}{d}}$, in which $\rho$ is the $\ell^\infty$ metric and $\mathcal{R}$ is the set of all axis-aligned rectangular subsets (see [8]). Hence, low-discrepancy point sets lead to low dispersion.

For a fixed $N$, one can use the *Sukharev sampling criterion* [9] for obtaining the best possible dispersion that can be obtained for any sample sets.

$$d_N(P) \geq \frac{1}{2\lfloor N^{\frac{1}{d}} \rfloor}, \tag{3}$$

which holds true for any point set $P$, when $d_N$ is the $\ell^\infty$ dispersion. Solving 3 for a prescribed dispersion yields, $N \geq (1/2d_N)^d$, which means that the number of samples is exponential in dimension.

### 2.1 Low-discrepancy sequences

Numerous low-discrepancy sample sets have been proposed. The choice of one set over the others depends on several concerns: 1) the desired range space, $\mathcal{R}$; 2) theoretical bounds on the discrepancy; 3) the quality of the samples as observed in applications; and 4) difficulty of constructing the samples. The most common range space is the set of axis-aligned rectangular subsets, which denote by $\mathcal{R}_{aar}$. In an appendix of [8], the current upper and lower bounds on the best possible discrepancy for an open sequence, attainable for different range spaces are summarized.

The best known upper bound for open sequences and $\mathcal{R}_{aar}$, $O(\frac{1}{N}(\log N)^d)$, is achieved by *Halton sequence* [10]. The Halton sequence is defined for a $d$-dimensional space by using $d$ prime bases to generate a sequence of $N$ quasi-random numbers as follows:

$$x_n = (\Phi_{b_1}(n), ..., \Phi_{b_d}(n)), \quad n = 1, 2, ..., N$$

The radix inverse function $\Phi_{b_j}(n)$ used in the above sequence is defined as:

$$\Phi_{b_j}(n) = 0.n_0 n_1 \cdots n_d = n_0 b_j^{-1} + n_1 b_j^{-2} + \ldots + n_d b_j^{-d-1} = \sum_{i=0}^{d} n_i b_j^{-i-1}$$

This finite sum has integer coefficients $n_i$ in which $0 \leq n_i < b_j$ and result from the digit expansion of the integer $n$ in base $b_j$:

$$n = n_d n_{d-1} \cdots n_2 n_1 n_0 = n_0 + n_1 b_j + n_2 b_j^2 + \cdots n_d b_j^d = \sum_{i=0}^{d} n_i b_j^i$$

In addition, for reference, a second low-discrepancy sequence, the Hammersley sequence [11], can be generated with a minor modification of the Halton sequence:

$$x_n = (n/N, \Phi_{b_1}(n), ..., \Phi_{b_{d-1}}(n)), \quad n = 1, 2, ..., N$$

For the case of $d = 2$, the Hammersley sequence is often referred to as the Van der Corput sequence, which was introduced much earlier.

Fig. 1 compares sample sets in $X = [0,1] \times [0,1]$ and also shows the Voronoi diagram of the point set (i.e., in each region the representative sample is the closest among all samples). Notice that for pseudo-random points there is a large variation in region size and shape, which illustrates the non-uniformity of random samples. For Hammersley points, the Voronoi regions appear more regular, the points are even better because it is closed sample set.

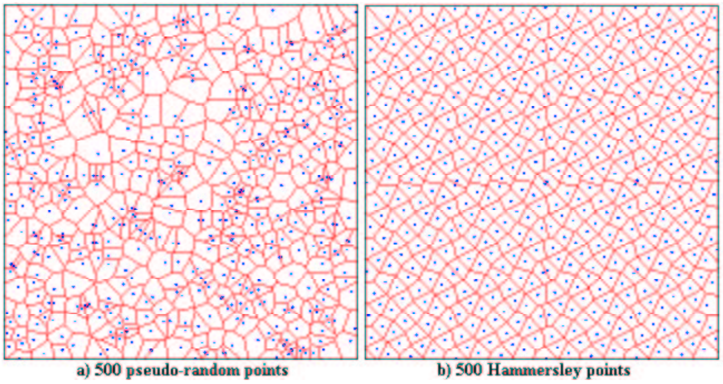

**Fig. 1.** Shown are 500 pseudo-random and Hammersley points, plus their associated Voronoi regions.

## 3 Quasi-random roadmap (QPRM)

We present a brief description of the motion planning problem and the probabilistic roadmap approach. Let $\mathcal{W}$ denote a subset of $\Re^2$ or $\Re^3$ in which a robot $\mathcal{A}$ is moving. The position of $\mathcal{A}$ is described by a configuration $q$ such that the position of every point on $\mathcal{A}$ can be determined relative to a fixed frame in $\mathcal{W}$. The set of all configurations is called the *configuration space* and is denoted by $\mathcal{C}$. For a configuration $q \in \mathcal{C}$, $\mathcal{A}(q)$ denotes the subset of $\mathcal{W}$ occupied by $\mathcal{A}$.

The cardinality of $\mathcal{C}$ is generally infinite since the robot is assumed to move continuously in $\mathcal{W}$. The aim of motion planning is to avoid a set of obstacles $\mathcal{O}$ in $\mathcal{W}$. If $\mathcal{A}$ intersects $\mathcal{O}$ we say that $\mathcal{A}$ *collides* with the obstacles and $\mathcal{C}_{free}$ is the set of collision-free (or *feasible*) configurations.

Given an initial configuration $q_s \in \mathcal{C}_{free}$ and a final configuration $q_g \in \mathcal{C}_{free}$, we define the motion planning problem as follows: Find a continuous path $\gamma : [0,1] \rightarrow \mathcal{C}$ such that $\gamma(0) = q_s$, $\gamma(1) = q_g$ and $\gamma(t) \in \mathcal{C}_{free}$ for all $t \in [0,1]$, or determine that no such path exists.

The probabilistic roadmap planner, is a relatively new approach to motion planning. It is a roadmap technique but rather than constructing the roadmap in a deterministic way, a probabilistic technique is used. A big advantage of PRM is that its complexity tends to be dependent on the difficulty of the path, and much less on the global complexity of the scene or the dimension of the configuration space.

The global idea of PRM is to pick a collection of (random) configurations in $\mathcal{C}_{free}$. These free configurations form the nodes of a graph $G = (V, E)$. Some pair of nodes are chosen and a simple local motion planner is used to attempt connections. When the local planner succeeds, an edge is added to the graph. The local planner must be very fast, but is allowed to fail on difficult instances. Once a graph reflects the connectivity of $\mathcal{C}_{free}$, it can be used to answer motion planning queries. To find a motion between a start configuration and a goal configuration, both are added to the graph using the local planner. Then a path in the graph is found which correponds to a motion for the robot. In a post-processing step this path is then smoothed to improve its quality. The pseudo-code for the algorithm for constructing the graph is shown now.

```
1. $V \leftarrow \emptyset$, $E \leftarrow \emptyset$
2. loop :
3.     $q \leftarrow$ a (random) configuration in $\mathcal{C}_{free}$
4.     $V \leftarrow V \cup \{q\}$
5.     $N_c \leftarrow$ a set of nodes chosen from $V$
6.     For all $q' \in N_c$, in order of increasing distance from $q$ do
7.       If $q'$ and $q$ are not connected in $G$ then
8.         If the local planner finds a path between $q'$ and $q$ then
9.           $E \leftarrow E \cup \{(q', q)\}$
```

The two time-consuming steps in this algorithm are line 3 and 8. The first generates a free sample and the other tests whether the local method can find a path between the new sample and a configuration in the graph. The geometric operations required for these steps dominate the work. To improve the efficiency of PRM we need to implement these steps very efficiently and we need to avoid calls to them as much as possible. That is, we need to place samples at "useful places" and need to compute only "useful edges". The problem is that it is not clear how to determine whether a node or edge is "useful". Many of the improvements described in [12], [13], [14], [15], [16] works this way.

Our approach in the PRM context is to simply replace the pseudo-random sample generator that appears in Line 3 of the roadmap generation algorithm above with a low-discrepancy, deterministic sampling method.

We can visualize the importance of the dispersion to the roadmap by placing a ball with radius equal to the dispersion at each sample point [17]; the entire sample space is now covered. Obviously, the fewer of these balls we have, the smaller our roadmap is (in terms of number of nodes); the smaller these balls are, the easier it is to connect a query to the roadmap. Of course, other methods (e.g., [13], [15]) can be used to address the narrow corridor problem; however, it is interesting to note that better performance can be obtained by merely replacing pseudo-random samples with deterministic samples. This seems to contradict the intuition that led to the PRM in the first place.

It may also be possible to improve the performance of other PRM sampling techniques [12], [6], [15], [14], [16] by using deterministic replacements for the samples.

## 4 Comparing PRM and QPRM experimentally

We have used Halton, Hammersley, and Sobol points as inputs to QPRM algorithms to solve a variety of planning problems in a range of dimensions. QPRM has performance better than or equal to its PRM counterpart. We have carried out several experiments with different values of connection radius on environments with narrow passages.

The time to generate the quasi-random samples in comparison to pseudo-random samples was never a significant factor. We begin here by exploring theoretical comparisons of discrepancy and dispersion from the literature. In the context of motion planning problems, discrepancy and dispersion are related to the uniform coverage of $\mathcal{C}$. Dispersion is also a direct measure of the maximum distance of any query to the roadmap and related to the ability to connect paths through narrow, twisting passages. Also, connection radii less than the dispersion of a point set may produce disconnected roadmap graphs. Next subsections present the experimental results for both holonomic and car-like robots.

### 4.1 Holonomic robots

The implementation for holonomic robots is in C++, using The Motion Strategy Library from the Univ. of Illinois, and the PQP collision package from the Univ. of North Carolina at Chapel Hill.

The number of nodes required to find a path that travels through the passage is shown in Table 1 and Table 2 for both QPRM and 50 averaged trials of the PRM. Fig. 2 shows two 6-dof planning problems. In the left case, the QPRM solved this problem with 11459 nodes, and the PRM averaged 12063 nodes over 50 trials. In the right case, the problem is solved with 22145 nodes for the QPRM, and the PRM averaged 25685 nodes over 50 trials.

Fig. 3 shows two planning problems performed on narrow corridors. In the first case (left picture), the QPRM solved the problem with 258 nodes, and the PRM averaged 474 nodes over 50 trials. Another case (right picture), QPRM with 996 nodes and PRM averaged over 50 trials 2843 nodes.

### 4.2 Car-like robots

The motion planning for car-like robots has received considerable attention in the past few years [7], [18]. These robots typically have three degrees of freedom, i.e., two positional dofs and one orientation dof, i.e. $(x, y, \theta)$, which are related through an intrinsic non-holonomic constraint:

$$\dot{y}\cos\theta - \dot{x}\sin\theta = 0$$

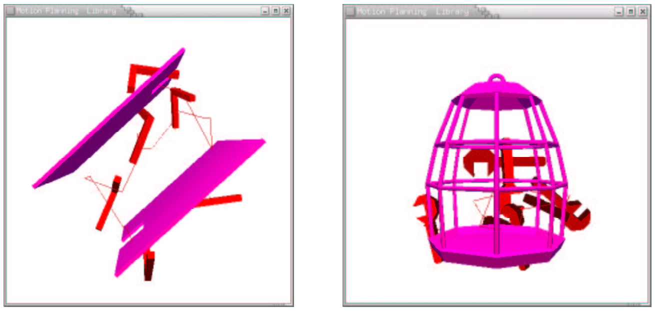

**Fig. 2.** In the left case, a 6-dof planning problem in which an elbow-shaped robot passes through two small opening; and in the right case, getting a wrench out of a cage.

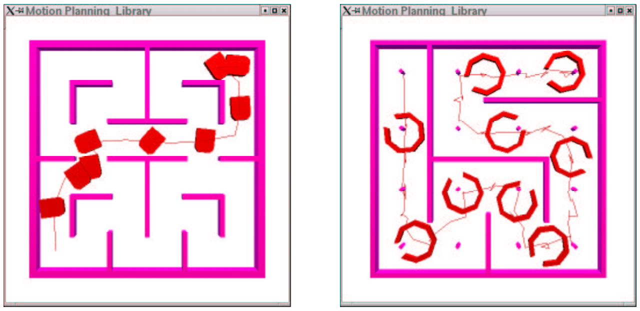

**Fig. 3.** Two examples of mobile robots in narrow corridors.

Since this equation is not integrable, there are constraints in the tangent space at each configuration (that is, on the allowable velocities) and these cannot be eliminated by defining a more restricted configuration space manifold in which the points can then move in any direction. The main consequence of this constraint is that an arbitrary path in the admissible configuration space does not necessarily correspond to a feasible trajectory for the robot. Therefore, the existence of a collision-free trajectory is not *a priori* characterized by the existence of a connected component in the admissible configuration space.

Much research has been done on motion planning for non-holonomic car-like robots (see [18] for a review). Within PRM framework, Švetska [7] use RTR paths as local method. An alternative to RTR local method is to use a local method that constructs the shortest car-like path connecting its argument configurations ([19]). Another strategy that has been used for non-holonomic planning is the RRT (Rapidly-Exploring Random Trees) approach [20].

The principle of the PRM framework, first, a graph of admissible configurations is built by connecting with a non-holonomic local planner (for instance, the Reeds & Shepp paths [19]) randomly generated configurations that are admissible and close to each other. Then, for solving any motion planning problem

**Table 1.** Performance data for PRM

| Prob. | dim | learning (sec.) | query (sec.) | nodes | edges |
|---|---|---|---|---|---|
| Elbow | 6 | 208.78 | 0.675 | 12063 | 55898 |
| Wrench | 6 | 724.19 | 2.855 | 25685 | 58684 |
| Maze 1 | 3 | 13.68 | 0.010 | 474 | 4200 |
| Maze 2 | 3 | 157.15 | 0.185 | 2843 | 42698 |

**Table 2.** Performance data for QPRM

| Prob. | dim | learning (sec.) | query (sec.) | nodes | edges |
|---|---|---|---|---|---|
| Elbow | 6 | 181.43 | 0.449 | 11459 | 49920 |
| Wrench | 6 | 517.75 | 1.839 | 22145 | 43940 |
| Maze 1 | 3 | 3.93 | 0.010 | 258 | 1242 |
| Maze 2 | 3 | 41.53 | 0.020 | 996 | 11654 |

initial and goal configurations are connected to configurations of the graph, and the graph is explored for a path between these configurations.

Fig. 4 shows two examples of planned paths for car-like robots: the workspace is modeled by a grid of 150 × 250 pixels. The number of nodes required to find a path that travels through the passage is shown in Table 3 for both QPRM and 100 averaged trials of the PRM. For the left picture, the QPRM solved the problem with 65 nodes, and right picture with the PRM averaged 98 nodes over 100 trials using the linear congruential generator.

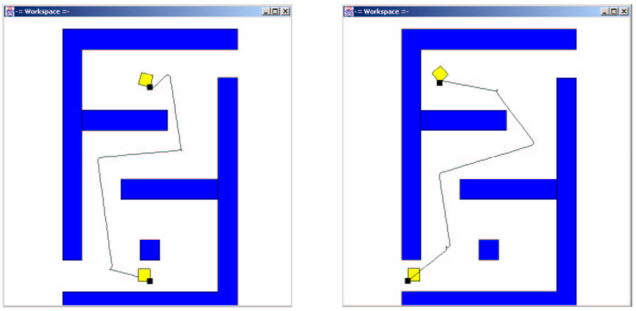

**Fig. 4.** Two examples of car-like robots navigating in narrow corridors.

**Table 3.** Performance data for both QPRM and PRM

| approach | learning (sec.) | query (sec.) | nodes | collision test calls | local method calls |
|---|---|---|---|---|---|
| QPRM | 36.773 | 0.50 | 62 | 397973 | 1386 |
| PRM | 70.652 | 0.70 | 98 | 925385 | 3381 |

## 5 Discussion

Based on our experiments, quasi-random samples appear to offer performance improvements similar to those observed in other fields where Monte Carlo methods were replaced with quasi-Monte Carlo methods. Another potential advantage of quasi-random sampling is that deterministic bounds on the performance of the planner can be derived.

With random sampling, performance guarantees and completeness are measured probabilistically [5], [7], [13]. When using deterministic samples, the planner is guaranteed to terminate in finite time with a solution, if a solution exists. Deterministic sampling enables the QPRM to be *resolution complete*, in the sense that if it possible to solve the query, it will eventually solve it. One can exploit dispersion bounds to characterize the set of configuration spaces that can be solved. This characterization is in terms of a parameter that measures the narrowest corridor width, in a manner similar to that of [21], [13]. Determining performance bounds remains a topic of further research.

The notion of resolution is comparable to dispersion, which guarantees each sample is within a prescribed displacement of other samples. The problem is that for *any* collection of $N$ $d-$dimensional points in $[0,1]^d$, the dispersion, $s$, is bounded as $s \geq \frac{1}{2}N^{-\frac{1}{d}}$ [3]. Therefore, an exponential number of samples is needed to maintain a fixed dispersion or resolution, regardless of whether random sampling, or quasi-random sampling are used.

Also, it is hard to compare deterministic, predictable methods to randomized methods, which yield varying results in multiple executions. We speculate that randomization appearing in other sampling techniques could be safely replaced and possibly improved with deterministic sampling; however, case-by-case comparisons would be necessary.

Derandomization of algorithms can be seen as a part of an effort to map the power of randomness and explain its role. Randomized algorithms are also related to probabilistic proofs and constructions in combinatorics by the effort to replace them by explicit, non-random constructions whenever possible.

A comparison with each variant of the PRM ([12], [14], [15], [16]), using quasi-random sampling would be interesting, so we propose it as a topic of future study.

## References

1. Schwarthz, J., Sharir, M. On the "piano movers" problem II: Techniques for computing topological properties of real algebraic manifolds. Advances in Applied Mathematics. New York: Academic (1983) 298-351
2. Latombe, J-C. Robot Motion Planning. Kluwer Academic Publishers (1991)
3. Niederreiter, H. Random number generation and quasi-Monte Carlo methods. Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania (1992)
4. Overmars, M. H.: A random approach to motion planning. Technical Report RUU-CS-92-32. Department of Computer Science, Utrecht University (1992)
5. Kavraki, L. E.: Random networks in configuration space for fast path planning. Ph.D thesis, Stanford University (1995)
6. Amato, N., Wu, Y. A randomized roadmap method for path and manipulation planning. Proc. IEEE Int. Conf. on Robotics and Automation (1996) 113-120
7. Švestka, P.: Robot motion planning using probabilistic roadmaps. Ph.D thesis, Utrecht University (1997)
8. Matoušek, J. Geometric discrepancy: An illustrated guide. Springer-Verlag (1999)
9. Sukharev, A. G. Optimal strategies of the search for an extremum. Translated from Russian, Zh. Vychisl. Mat. i Mat. Fiz., 11(4) (1971) 910-924
10. Halton, J. H. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. Numer. Math., Vol 2 (1960) 84-90
11. Hammersley, J. M. Monte-Carlo methods for solving multivariable problems. Ann. New York Acad. Sci., vol 86 (1960) 844-874
12. Amato, N., Bayazit, O., Dale, L., Jones, C., Vallejo, D. OBPRM: An obstacle-based PRM for 3D workspaces. Robotics: The algorithmic perspective, P. K. Agarwal, L. E. Kavraki, and M.T. Mason (eds). A.K. Peters (1998) 155-168
13. Hsu, D., Kavraki. L., Latombe, J-C., Motwani, R., Sorkin, S. On finding narrow passages with probabilistic roadmap planners. Robotics: The algorithmic perspective, P. K. Agarwal, L. E. Kavraki, and M.T. Mason (eds). A.K. Peters (1998) 141-154
14. Nissoux, C., Siméon, T., Laumond, J-P. Visibility based probabilistic roadmaps. Proc. IEEE Int. Conf. on Intelligent Robots and Systems (1999) 1316-1321
15. Boor, V., Overmars, M. H., Van der Stappen, A. F. The gaussian sampling strategy for probabilistic roadmap planners. Proc. IEEE Int. Conf. on Robotics and Automation (1999) 1018-1023
16. Wilmarth, S. A., Amato, N., Stiller, P.F. MAPRM: A probabilistic roadmap planner with sampling on the medial axis of the free space. Proc. IEEE Int. Conf. on Robotics and Automation (1999) 1024-1031
17. Branicky, M., LaValle, S. M., Olson, K., Yang, L. Quasi-randomized path planning. Proc. IEEE Int. Conf. on Robotics and Automation (2001) 1481-1487
18. Laumond, J-P(Editor). Robot motion planning and control. Lecture notes in control and information sciences 229. Springer (1998)
19. Reeds, J. A., Shepp, R. A.: Optimal paths for a car that goes both forward and backwards. Pacific Journal of Mathematics. 145(2) (1990) 367-393
20. LaValle, S. M., Kuffner, J. J. Randomized kinodynamic planning. Proc. IEEE Int. Conf. on Robotics and Automation (1999) 473-479
21. Barraquand, J., Kavraki, L., Latombe, J-C., Motwani, R., Li, Tsai-Yen, Raghavan, P. A random sampling scheme for path planning. The International Journal of Robotics Research, Vol. 16, No. 6 (1997) 759-774