Motion Planning in Environments with Dangerzones

Danielle Sent

Mark H. Overmars

UU-CS-2001-06

February 2001

Motion Planning in Environments with Dangerzones*

Danielle Sent Mark H. Overmars

Department of Information and Computing Sciences, Utrecht University
P.O. Box 80.089, 3508 TB Utrecht, the Netherlands
Email: {*danielle,markov*}*@cs.uu.nl*

Abstract

This paper addresses the problem of path planning for a free-flying object in a (three-dimensional) environment that contains both obstacles and so-called dangerzones. The path should (obviously) avoid collisions with the obstacles. The path is allowed to intersect with the dangerzones but this should be avoided as much as possible. We show that, under some mild conditions, a path always exists in which the moving object never completely penetrates the dangerzones. Based on this result we present a probabilistically complete roadmap method that finds such paths. The methods has been implemented and some experimental results are given.

1 Introduction

Motion planning has been extensively studied over the past two decades. Besides the traditional application in robotics, motion planning has become increasingly important in areas such as computer animation, computer aided design, and medical applications. In its basic form the motion planning problem asks for planning the path for a moving body from a given start to a given goal position in a workspace W containing a set $B = \{B_1, \ldots, B_n\}$ of obstacles. Different strategies for motion planning have been proposed. (See Latombe's book[9] for the situation up to 1991.) In recent years many people have concentrated on probabilistic roadmap methods (PRM), developed originally in Utrecht[11, 12] and Stanford[6, 5]. These methods use a local and a global planner. The global planner takes samples in the configuration space of the moving body and tries to connect samples using a local planner, in this way creating a roadmap of possible motions. Next the roadmap is searched for a path from start to goal configuration, which is then further improved using smoothing techniques. PRM's have been applied for many different types of motion planning (free-flying, car-like, robot arms, flexible, etc.) and many techniques have been suggested to improve the performance (e.g. visibility roadmaps, Gaussian sampling, and lazy PRM's). See [1, 2, 3, 4, 8, 7, 10, 13] for some of the many important results obtained recently.

Besides hard constraints on the resulting paths (no intersection with the obstacles and feasible for the moving body), in many applications there are also soft constraints, e.g. to preferably stay away from certain obstacles and to avoid certain areas in the workspace as much as possible. In this paper we give a possible approach to deal with such constraints. To this end we introduce the notion of a *dangerzone* that should be avoided as much as possible by the moving body. See Figure 1 for an

*This research was supported by the ESPRIT LTR project MOLOG, and by the Netherlands Organization for Scientific Research (NWO).

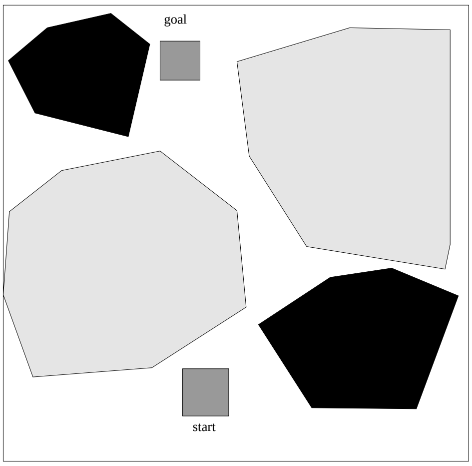

Figure 1: Two dangerzones (grey) and two obstacles (black) that the moving square should try to avoid.

example. The square body must move from the given start to the given goal position. It must avoid the black obstacles and should preferably avoid the grey dangerzones. This is not completely possible but the robot should try to stay as much as possible in the free space.

This paper is organized as follows. In Section 2 we formally define the notion of a dangerzone and the properties of the paths we want to compute. This leads to the notion of two new spaces: the undesirable space where the moving body lies completely in a dangerzone, and the semi-desirable space where the body partially lies in a dangerzone (besides the usual free and forbidden space). We next show that, under some mild conditions, whenever there exists a path, there also exists a path that completely avoids the undesirable space, so the body will always have at least one point in the free workspace. In Section 3 we show how we can use a probabilistic roadmap planner to find such paths, by giving an appropriate local planner and sampling method. We prove that the planner we obtain is probabilistically complete. In Section 4 we discuss some experiments with the developed planner that shows that the method indeed works and leads to the required paths. Finally, in Section 5 we state some conclusions and give some suggestions for further research.

2 Path planning

The problem we study in this paper is an extension of the traditional problem of motion planning. The workspace W not only contains a set of obstacles $B = \{B_1, \ldots, B_n\}$, but also a set of dangerzones $Z = \{Z_1, \ldots, Z_m\}$. Dangerzones are explicitly modelled in the environment (so they are e.g. not implicitly defined by an area around an obstacle). We assume that obstacles and dangerzones do not intersect each other (although most results carry through if a dangerzone can contain at most one obstacle). We consider obstacles and dangerzones to be open, that is, the robot is allowed to touch them.

Let R be the moving body, which we call the robot from here on. We assume that R is holonomic and can at least translate freely in the workspace W. (It can also have further rotational degrees of freedom.) A path τ for R is valid if it is feasible for the robot (satisfies constraints on the possible motions for the robot) and if the robot does not collide with any obstacle. We prefer the robot also to avoid the dangerzones as much as possible. This is difficult to define formally. But, as we will see below, we can at least avoid that the robot completely penetrates a dangerzone.

2.1 The configuration space

It is usual in motion planning to consider the configuration space C of the robot. This space can be subdivided in a number of subspaces:

- The free configuration space C_{free} that consists of those configurations in which the robot R does not intersect any obstacle or dangerzone.
- The forbidden configuration space C_{forb} that consists of those configurations in which the robot R does intersect an obstacle.
- The undesirable configuration space C_{undes} that consists of those configurations in which the robot R completely lies in a dangerzone.
- The semi-desirable configuration space $C_{semi-des}$ that consists of those configurations in which the robot R does not intersect an obstacle and at least one point of the robot lies in a dangerzone and at least one point does not.

It is easy to see that these four subspaces are disjoint and that their union is the full configuration space C.

2.2 Paths among dangerzones and obstacles

A path is a connected curve τ in the configuration space between a start configuration s and a goal configuration g. We assume that both s and g lie in C_{free}. We will now show that we never need to completely penetrate a dangerzone, or more precisely, we will show that when the problem has a solution, there always exists a path that completely lies in $C_{free} \cup C_{semi-des}$. The idea of the proof is as follows: Let τ be a path from s to g that lies in $C_{free} \cup C_{semi-des} \cup C_{undes}$. This path consists of pieces that lie in $C_{free} \cup C_{semi-des}$ and pieces that lie in C_{undes}. Now consider a maximal piece τ' of the path that lies in C_{undes}. Because the dangerzones do not intersect, while traversing τ' the robot lies completely within one danger zone Z_i. We will show that we can replace the piece τ' by another path τ'' in which the robot will always touch the boundary of Z_i. τ'' then lies in $C_{semi-des}$. Repeating this for all pieces that lie in C_{undes} we obtain a path in $C_{free} \cup C_{semi-des}$ as required.

Lemma 1 *Let τ be a path from a semi-desirable configuration s to a semi-desirable configuration g such that during the motion from s to g the robot is always contained in a dangerzone Z_i. Then there exists a path τ' from s to g for which the robot always touches the boundary of Z_i.*

Proof: We will first show that there exists a possibly forbidden path τ_i from s to g that always intersects the boundary of Z_i. Let p_s be a point of the robot that lies in the start configuration on the the boundary of Z_i and let p_g be a similar point for the goal configuration (such points must exist because the start and goal configuration are semi-desirable). Let p be the point on the boundary of Z_i at which p_s lies in the start configuration and p' the point at which p_g lies in the goal configuration. Now take a curve connecting p_s and p_g inside the robot R. We first translate the robot such that p always lies on the curve. This results in a new configuration m in which p_g lies on p. Next take a curve on the boundary of Z_i connecting p and p'. Translate the robot such that p_g always lies on the curve.

Finally, if required, rotate the robot while keeping p_g on p' to configuration g. The resulting motion is feasible and will always intersect the boundary of Z_i. It might though intersect other dangerzones and even obstacles.

So we now have two paths from s to g. The original path τ that lies completely inside Z_i and the new path τ_i that always intersects the boundary of Z_i. Let CZ_i be the part of the configuration space that consists of all configuration in which the robot lies completely inside Z_i. s and g lie on the boundary of CZ_i. The original path τ lies inside CZ_i and the new path τ_i lies outside or on the boundary of CZ_i. From this it follows that s and g must lie on the same connected component of the boundary of CZ_i. Hence, there is a path τ' from s to g that lies completely on the boundary of CZ_i. This path will not intersect obstacles and the robot will always touch the boundary of Z_i. □

Up to now we considered a path intersecting just one dangerzone. We need to extend this to a path among multiple dangerzones with arbitrary start and goal configurations. This result is stated to the following theorem.

Theorem 1 *If there exists a path τ between two free configurations s and g, that possibly intersects one or more dangerzones Z, than there exists a path τ' that lies completely in $C_{free} \cup C_{semi-des}$.*

Proof: As indicated above we split τ in maximal pieces that alternatingly lie in $C_{free} \cup C_{semi-des}$ and in C_{undes}. During a maximal piece of the path that lies in C_{undes} the robot lies completely within one danger zone Z_i. From Lemma 1 it follows that we can replace these pieces by semi-desirable pieces. The resulting path will lie completely in $C_{free} \cup C_{semi-des}$ as required. □

Now we know that it suffices to look for paths from start to goal, where the robot will never be completely in the dangerzone. It will always have at least one point at the border of the zone. We will use this fact in our planner, in particular in the sampling approach used.

3 The planner

In this section we will show how to use the PRM approach to motion planning to obtain a probabilistically complete planner that can deal with both obstacles and dangerzones. Globally speaking, PRM works as follows: We sample the configuration space, throwing away forbidden and unwanted configurations. Each sample we keep we try to connect to samples found earlier. For this we use a simple local planner that should be fast for easy cases but is allowed to fail for more complicated situations. In this way we created a roadmap (a graph) in which the nodes correspond to the samples and the paths correspond to successful runs of the local planner. Once the roadmap is large enough (containing enough information about the possible paths), the start and goal configuration are added to it (again using the local planner), and a path is found in the graph, which can be turned into a motion from start to goal. Smoothing is often applied afterwards to improve the quality of the path.

To apply PRM in our environments with dangerzones we need two important ingredients: a sampling approach that samples $C_{free} \cup C_{semi-des}$, and a local planner that tries to stay out of dangerzones.

3.1 Sampling

Obviously, it is useful to first only sample C_{free}. If a path between start and goal can be found there than the dangerzones can be avoided altogether. But if such a path does not exist we also need to sample $C_{semi-des}$. Let us assume that obstacles and dangerzones do not touch (see the conclusions for some ideas on how to lift this restriction).

An easy approach would be to simply generate a random configuration and then test whether the robot at this configuration intersects an obstacle or lies completely within a dangerzone. Unfortunately, testing whether the robot lies completely within a dangerzone is a relatively expensive operation. We can avoid this as follows: We select a point p_W in the free part of the workspace (e.g. by selecting a random point and checking that it does not lie in an obstacle or dangerzone). Next we select a point p_R in the robot. Now we create a configuration as follows. If there are rotational degrees of freedom we pick them randomly. In the resulting orientation we place the robot R with point p_R on position p_W. Next we test whether the robot intersects any obstacles. Because p_W lies in the free part of the workspace and p_R lies inside the robot, the resulting configuration cannot lie completely inside a dangerzone.

It can easily be shown that in this way $C_{free} \cup C_{semi-des}$ is completely sampled. The sampling though is not uniform. C_{free} is sampled uniformly but the chance that a semi-desirable position is chosen is inversely proportional to the area of overlap between the robot and the dangerzone(s). We consider this a useful property of the sampling approach because it favors configurations that have little overlap with dangerzones.

Clearly this sampling approach can be combined with other sampling techniques like Gaussian sampling[3] or visibility sampling[10]. We did not study such combinations yet.

3.2 Local Planner

The goal of the local planner is to efficiently connect two samples that are close together with a feasible path. Often a simple interpolation between the two samples is used (that is, we move along a straight line in configuration space). In our situation we want the path to lie completely in $C_{free} \cup C_{semi-des}$. The simple interpolation does not guarantee this. For example, when two samples lie on opposite sides of a dangerzone, the simple local planner will generate a path that goes through the dangerzone.

Instead we use the following approach. Remember that a sample was created by choosing a point in the robot and a point in the free part of the workspace. We have such a pair (p_W, p_R) for the first sample and a second pair (p'_W, p'_R) for the second sample. We first check whether the line segment $\overline{p_W p'_W}$ lies completely in the free workspace and whether $\overline{p_R p'_R}$ lies completely inside the robot. If not the local planner fails. Otherwise, we interpolate a point in the robot between p_R and p'_R and move this interpolated point over $\overline{p_W p'_W}$. At the same time we interpolate the remaining rotational degrees of freedom. This path we check for collisions with the obstacles in the usual way. It is easy to see that the resulting path lies in $C_{free} \cup C_{semi-des}$.

In the PRM approach one also has to decide which nodes to select for possible connection using the local planner. Many different methods have been proposed. We decided to use the most standard one, namely the nearest-k method, which means that we only choose the k nearest neighbors of a sample, given by a distance metric (using the Euclidean distance) on the configuration space.

3.3 Probabilistic completeness

The PRM method is probabilistically complete for many different types of motion planning problems. This means that, assuming a solution exists, when time goes to infinity the change that a valid path is found tends to 1. This is a desirable property of motion planners (see e.g. the thesis of Švestka[12]). Unfortunately, with the local planner described above there are situations in which a solution is never found. Consider the situation depicted in Figure 2. The rectangular robot must either move along

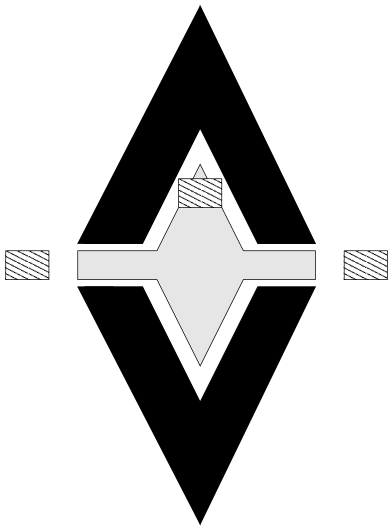

Figure 2: A scene with black obstacles and a grey dangerzone. With the standard local planner no path will be found for the translating robot.

the top or along the bottom triangle. This is obviously possible, but the problem is that there is no path through the free part of the workspace that can be followed by an interpolated point in the robot. Hence, it is impossible with the local planner described above to connect the left with the right.

To obtain a probabilistically complete planner we use a different local planner. We simply interpolate between the two configurations and check whether the robot remains in $C_{free} \cup C_{semi-des}$. (We do not use this local planner in our implementation because it is much slower.) To prove probabilistical completeness we can adapt a theorem from the thesis of Švestka[12] that states that a planner is probabilistically complete if the robot satisfies certain properties.

Theorem 2 *The PRM method described above for environments with obstacles and dangerzones is probabilistically complete.*

Proof: We will only sketch the proof. Assume a path exists between the start and goal that satisfies our criteria. This path will correspond to some curve in the configuration space. Because of the properties of the space, this curve will have some positive clearance ϵ. We cover the path with small, partially overlapping balls with their center on the path. By choosing the size of the balls appropriately (depending on ϵ) we can guarantee that configuration in the same ball and in neighboring balls will be connected by the local planner. When time goes to infinity, the chance that every ball contains at least one sample goes to 1. As a result all these samples will get connected in the graph and a path from start to goal is found. $\square$

4 Experimental results

We implemented the planner described within the SAMPLE motion planning environment developed at Utrecht University. We tested it in a number of different environment. Here we report on two such experiments. In both cases we consider a robot with three translational degrees of freedom only. Realize that this actually complicates the problems because the robot has much less manouverability.

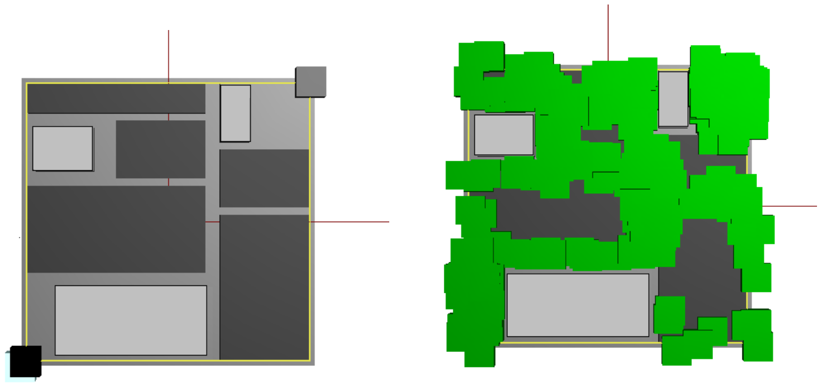

Figure 3: The workspace containing three obstacles and five dangerzone. The cube robot has to travel from bottom left to top right. The right figure shows the samples taken by the planner.

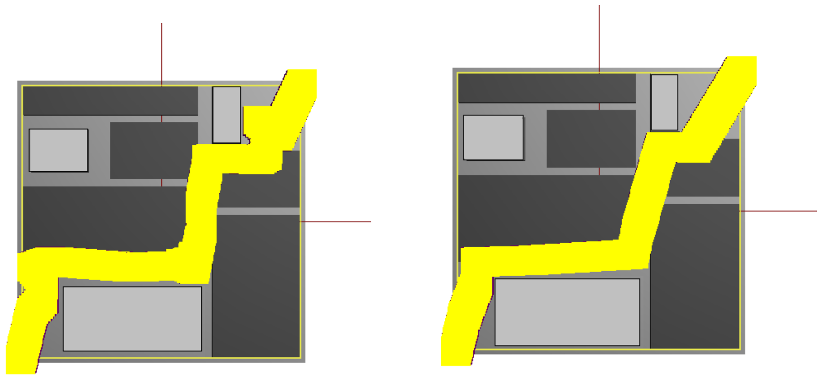

Figure 4: The path before and after smoothing.

4.1 A simple example

The first scene consists of three obstacles and five dangerzones (see the left picture in Figure 3). The workspace is ten by ten by one meter. The robot is a cube of one by one by one meter. Note that, even though the pictures seem two-dimensional because we look from above, this is actually a three-dimensional problem. There is no way the robot can move from start to goal without intersecting some dangerzones. As can be seen in the right picture in Figure 3 samples are created along the boundary of the dangerzones. The resulting path (before and after smoothing) is shown in Figure 4. As can be seen it does (and has to) intersect the dangerzones, but it tries to stay out of them as much as possible.

The following table shows some information about the number of samples and collision checks. These are rounded values, averaging over a number of runs. The running times are on a 500 MHz Pentium XEON processor.

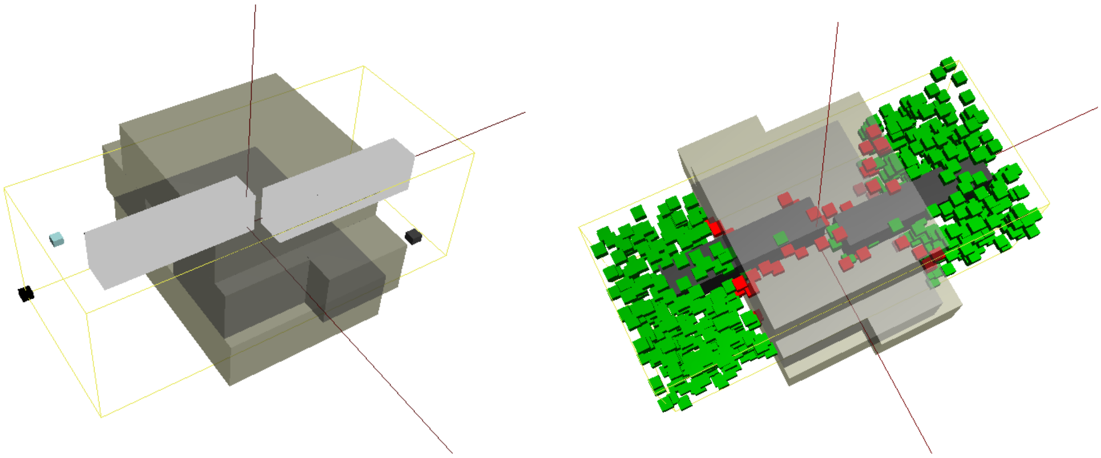

Figure 5: The workspace containing a block with a corridor with two dangerzones. The robot has to travel through the corridor, avoiding the dangerzones as much as possible. The bottom picture shows the samples taken by the planner.

Total time:	20 s
Number of graph nodes:	750
Number of collision checks:	20000
Number of local planner calls:	4000
Time for smoothing:	4 s

The number of failed samples is rather high. So is the amount of collision checks and local planner calls. This can be improved by using more sophisticated sampling techniques.

4.2 A narrow corridor

In the second experiment, a real three-dimensional scene has been used. The workspace has a big obstacle with a narrow corridor through it. The only way to move the robot from left to right is through the corridor. In this corridor two dangerzones are situated. The two dangerzones do not touch each other, nor the obstacle. See Figure 5 for a partially transparent view of the workspace with the red obstacles and blue dangerzones.

The implementation had no problem at all finding a path. The sampling method creates enough samples in the corridor and between the dangerzones. Rather a lot of samples were taken in the free space just in front of and behind the passage. This is due to the fact that the free space is sampled uniformly. This can though easily be avoided by using techniques like Gaussian sampling [3] or visibility sampling[10]. Figure 6 shows a path and a smoothed path for the robot.

The following table summerizes the performance of the planner.

Total time:	10 s
Number of nodes:	1000
Number of collision checks:	10000
Number of local planner calls:	2000
Time for smoothing:	2 s

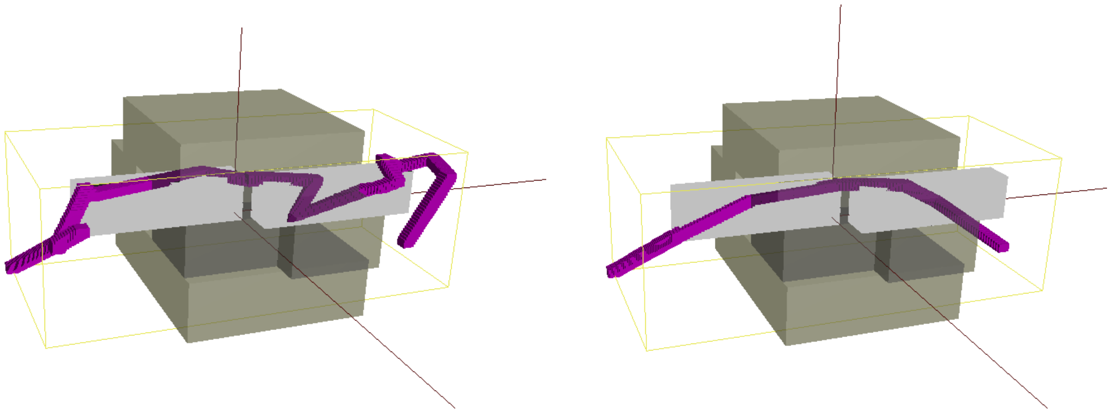

Figure 6: The path before and after smoothing.

5 Conclusions and future research

As can be seen from the experiments, the planner is able to find paths between dangerzones and obstacles. During the motion the robot always has at least one point in the free space. We have proven that there always exists such a path when dangerzones do not touch each other and do not touch any obstacles. We are currently working on methods to drop these assumptions. Dangerzones that overlap or intersect obstacles might be retracted to dangerzones that only touch. Touching dangerzones can be dealt with by explicitly sampling on the boundary of the dangerzones. And most of the results still hold when a dangerzone is allowed to contain one obstacle. (But not more than one, as can easily be verified.)

In our current implementation, the number of samples required, and hence, the number of collision checks and local planner calls, is rather high. The reason for this is that we sample the whole free configuration space without choosing ”good” areas to take samples in. Combination of our approach with existing techniques to reduce the number of samples and computation time required should remedy this problem.

Dealing with dangerzones is just one aspect of improving the quality of the paths. Many other “soft” constraints exist on paths, like staying away from obstacles, using as little degrees of freedom as possible, and avoiding sharp turns. We plan to study such quality issues further in the near future.

References

[1] N. Amato, O. Bayazit, L. Dale, C. Jones, D. Vallejo, OBPRM: An obstacle-based PRM for 3D workspaces, in: *Proceedings of the International Workshop on Algorithmic Foundations of Robotics (WAFR)*, 1998, pp. 155-168.

[2] R. Bohlin, L.E. Kavraki, Path Planning Using Lazy PRM, in: *Proceedings of the IEEE International Conference on Robotics and Automation*, 2000, pp. 521-528.

[3] V. Boor, M.H. Overmars, A.F. van der Stappen, The Gaussian Sampling Strategy for Probabilistic Roadmap Planners, in: *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, Detroit, Michigan, 1999, pp. 1018-1023.

[4] C. Holleman, L.E. Kavraki, J. Warren, Planning Paths for a Flexible Surface Path, in: *Proceedings of the IEEE International Conference on Robotics and Automation*, 1998, pp. 21-26.

[5] L.E. Kavraki, *Random networks in configuration space for fast path planning*, PhD thesis, Stanford University, 1995.

[6] L.E. Kavraki, J.-C. Latombe, Randomized Preprocessing of Configuration Space for Fast Path Planning, in *Proceedings of the IEEE International Conference on Robotics and Automation*, San Diego, 1994, pp. 2138-2139.

[7] J.J. Kuffner, S.M. LaValle, RRT-Connect: An Efficient Approach to Single-Query Path Planning, in: *IEEE International Conference on Robotics and Automation*, 2000, pp. 995-1001.

[8] F. Lamiraux, L.E. Kavraki, Path Planning for Elastic Plates Under Manipulation Constraints in: *Proceedings of the IEEE International Conference on Robotics and Automation*, 1999.

[9] J.-C. Latombe, *Robot motion planning*, Kluwer Academic Publishers, Boston, 1991.

[10] C. Nissoux, T. Siméon, J.-P. Laumond, Visibility Based Probabilistic Roadmaps, in: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Kyongjy, Korea, 1999, pp. 1316-1321.

[11] M.H. Overmars, A random approach to motion planning, Technical Report RUU-CS-92-32, Department of Computer Science, Utrecht University, The Netherlands, 1992.

[12] P. Švestka, *Robot motion planning using probabilistic road maps*, Ph.D thesis, Utrecht University, 1997.

[13] S.A. Wilmarth, N.M. Amato, P.F. Stiller, MAPRM: A Probabilistic Roadmap Planner with Sampling on the Medial Axis of the Free Space, in: *Proceedings of the IEEE International Conference on Robotics and Automation*, 1999, pp. 1024-1031.