A Random Loop Generator for Planning the Motions of Closed Kinematic Chains using PRM Methods

J. Cortés, T. Siméon, J.P. Laumond

LAAS-CNRS
7, avenue du Colonel-Roche
31077 Toulouse Cedex - France
{jcortes,nic,jpl}@laas.fr

Abstract

Closed kinematic chains in mechanical systems represent a challenge for their motion analysis, and therefore, for path planning. Closed mechanisms appear in different areas where path planning algorithms are applied. We propose a method to handle them within Probabilistic RoadMap (PRM) techniques. This method is an extension of the approach proposed in [5]. Our main contribution concerns the generation of random configurations. The structure of the mechanism is analyzed in a preprocessing step. Then, in the roadmap construction phase, an algorithm called Random Loop Generator uses data from this analysis. This algorithm increases the probability of randomly generating valid configurations of the closed mechanism. Experimental results demonstrate the efficiency of the approach.

1 Introduction

Many kinds of path planning algorithms have been proposed during the last 20 years [4, 11]. PRM techniques [6, 20], appeared in the 90's, have been well accepted by the research community because of their generality and good performance in high dimension configuration-spaces. These two properties allowed the development of methods capable to solve classes of problems that could not be handled with other techniques, and in particular problems containing complex closed chain mechanisms.

In the field of robotics, closed chains are formed when several manipulator arms coordinate to handle an object [7]. Parallel robots are closed loop mechanisms in which the end-effector is connected to the base by at least two independent kinematic chains [14]. In the area of mobile robots, legged robots also create loops whenever two or more legs contact with the ground [10].

The success of PRM path planners in robotics has led to their application in other areas such as computational chemistry or graphic animation. Examples containing systems with closed kinematic chains are easily found in all these domains. The molecular models used in computational chemistry [13] generally contain cyclic chains. In the application of motion planning to graphics animation [9], closed kinematic chains are formed, for example, when human-like characters act in contact with each other.

The analysis of closed kinematic chains can be expressed as an inverse kinematics problem. The formulation of this problem often leads to a nonlinear set of equations. A variety of techniques to solve such systems of equations is available [15]. However, none of them holds the suitable requirements to be used within probabilistic path planning methods. Required characteristics are generality, efficiency and robustness. Alternative solutions oriented to the treatment of closed mechanisms with PRM planners have to be studied.

The relative youth of PRM techniques or the difficulty of the problem may be the reasons why little work has been yet developed in relation with this subject (Section 2). Our approach shares ideas with one of them [5]. The main contribution in this paper is the method to generate random configurations of closed mechanisms (Sections 3 to 5). The *Random Loop Generator* drastically reduces the computing time of this process in relation to referenced works. The integration of the algorithm within a path planning software that uses several PRM techniques is presented in Section 6.

2 Related Work

The basic principle of PRM techniques [6, 20] consists in randomly sampling the configuration space (CS) and in connecting the samples in the free space (CS_{free}) by *local paths*. The result of this process is a graph (the roadmap) that encodes the topology of CS_{free}. Planning queries are then solved by connecting the start and goal configurations to the roadmap and searching a path in it. The final step is the smoothing of this resulting path.

Several variants [1, 2, 3, 19, 21] have been proposed to improve the performance of the basic-PRM algorithm. For all of them, the generation of random configurations is a trivial process in the case of open kinematic chains. On the contrary, the difficulty to randomly generate configurations of a mechanism containing closed chains significantly degrades the performance of PRM based techniques. Moreover, the existence of loops adds another difficulty to devise local methods. Closure constraints must be satisfied at every intermediate configuration along paths.

The first PRM based approach able to handle mechanisms with closed chains was presented in [12]. The method uses numerical optimization techniques in the roadmap construction. The nodes of the roadmap are obtained by sampling random configurations that ignore kinematic closure, and then a randomized gradient descent is performed to minimize error functions that express these constraints. The edges are computed by executing a randomized traversal of the constraint surface between two nodes, also using gradient descent. The drawback of this method is that, in practice, gradient descent algorithms present problems of convergence when the mechanical system is fairly complicated.

A more efficient method was presented later in [5]. It is based on the use of forward and inverse kinematics. Each loop in the mechanism is broken into two chains. For the node generation, standard PRM random sampling techniques and forward kinematics are applied to one of the chains (called *active*), and then inverse kinematics is used in the remaining (*passive*) part of the loop in order to force closure. In the computation of edges, the local planner is limited to act onto the active joints. The passive part of each loop follows the motion of the rest of the chain. Point to point inverse kinematics is computed along the path.

The choice of active and passive chains is a critical point in this approach. A valid configuration of a loop is generated if and only if the end-frame of the active chain is in the workspace of the passive chain ($W_{passive}$). Therefore, the probability to generate random configurations of an active chain achieving this condition depends on the workspaces of the two chains ($W_{passive}$ and W_{active}). Greater is the volume of $W_{active} \cap W_{passive}$ higher is this probability. However, closed-form inverse kinematics solutions of the passive chains are required for the efficiency of the roadmap computation. That means a limit for the number of joints in these chains. Thus, in many cases active chains have to be composed of a high number of joints and links, yielding to very small volumes of intersection. This situation drops off the above mentioned probability.

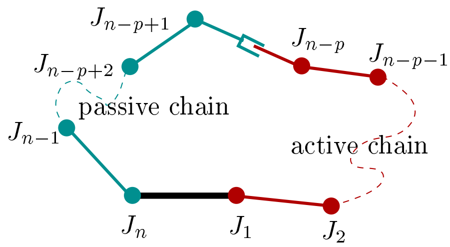


Figure 1: Decomposition of a single loop.

3 Contribution

The method that we propose is an extension of the approach in [5]. We share most of the ideas in that paper. Our contribution is to provide a solution to the above discussed drawback. We propose an algorithm that we call Random Loop Generator (RLG). This algorithm is used to generate configurations of each single loop in the mechanism. The aim is to make the end-frame of the active chain reachable by the passive chain at every random sample. The ideal solution should be to sample configurations of the active chain in the manifold corresponding to $W_{active} \cap W_{passive}$. However, computing this intersection is generally difficult.

The RLG algorithm performs a progressive generation (one parameter at one time) of the configuration of the active part of a loop. Contrary to the standard sampling method, where the value for a joint is randomly chosen in a constant interval that only depends on the joint limits, now this interval is recomputed for every configuration. The goal is to guide the end-frame of the active chain towards a region of the space. This region is a conservative approximation of the reachable workspace of the passive chain.

Another advantage of the algorithm is that the probability to generate valid (closed) configurations is slightly affected by the relative volume of $W_{passive}$ and W_{active}. Therefore, passive chains with efficient inverse kinematics solution can be chosen regardless of the rest of the chain.

The next sections describe our approach to generate configurations of closed mechanisms. First, in Section 4, we explain how the loops are decomposed in active and passive chains. Then, Section 5 details the principle of the RLG algorithm.

4 Mechanical System Decomposition

Closed chains in the mechanical system are handled separately. Each single loop is divided into two kinematic chains (passive and active).

Passive chains : when generating a closed configuration, the active chain is computed first, and then the configuration of the passive chain is imposed in order to close the loop. Thus, passive chains must be parts of the mechanism that are not affected by other constraints. Another condition for a passive chain is the full mobility of its end-effector, needed to reach a random situation of the end-frame of the corresponding active chain. As already mentioned, the efficiency of this approach depends on the inverse kinematics computation of passive chains. This is the other important criterion to select them.

Active chains : once identified the passive chain of a loop, the rest will be the active chain. When none of the joints in this chain is affected by constrains, the schema of the decomposition of the loop corresponds to Figure 1. The RLG algorithm is applied to the whole active chain $J_1 \rightarrow J_{n-p}$. However, RLG does not work on constrained joints (i.e. joints of which the configuration is imposed by characteristics of the mechanism). This situation occurs in the case of complex loops.

An order must be established for the generation of the configuration of loops. Joints included in two loops are treated only for the first defined one. When the other loop will be handled, the configuration of these joints has been already generated. Thus, this part of its active chain can be considered as a rigid body. RLG will be used to generate the configuration of the remaining active part of the loop. Hereafter, when we mention active chains, we refer to its non-constrained part (see Figure 2).

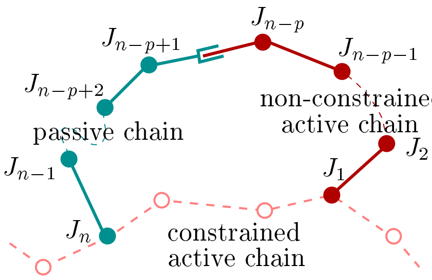


Figure 2: Decomposition of a loop with constrained joints.

5 Random Loop Generator

The process to generate configurations of a mechanical system containing closed chains can be described as follows. First, the value of joints not implied in loops is randomly generated. Then, the RLG algorithm is applied to each loop following the order discussed in the previous section.

The configuration of an active chain $J_1 \rightarrow J_{n-p}$ is computed joint by joint, beginning at J_1. The interval of variation of each joint [1] is recomputed for each sample. The aim is to increase the probability to obtain a situation of the end-frame of this chain lying inside $W_{passive}$. When a configuration for the active chain is successfully generated, inverse kinematics is used to obtain the configuration of the passive chain $(J_n \rightarrow J_{n-p+1})$.

[1] We consider single joints. Complex joints can be decomposed. The value of a single joint is the equivalent to a parameter of configuration.

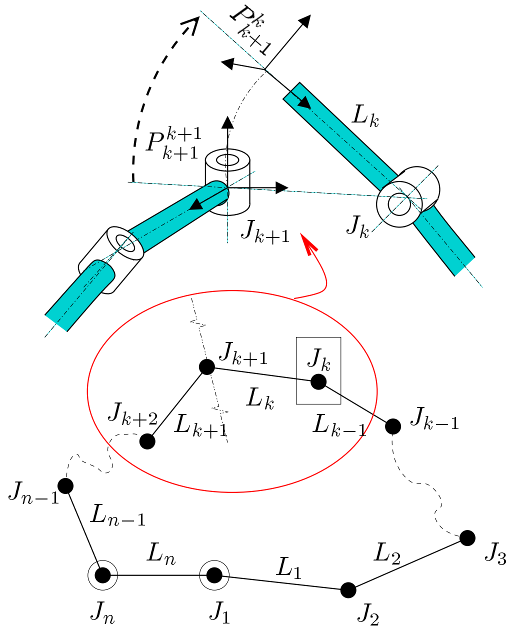


Figure 3: Illustration of the principle of RLG.

The configuration of the active chain is computed as follows. Consider the single loop illustrated by Figure 3, formed by n joints J_i and n links L_i, i=1...n. The value of a joint fixes the spatial situation of the next link in the kinematic chain. Let us consider that the configuration for the chain $J_1 \rightarrow J_{k-1}$ has been already generated, and therefore the spatial situation of L_{k-1} is fixed and known. Let P^k_{k+1} be the frame that corresponds to the physical placement of J_{k+1} at the end of the link L_k. The values of J_k that make possible to close the mechanism are the ones for which P^k_{k+1} is reachable (in position and orientation) by the chain $J_n \rightarrow J_{k+1}$. We call this set the *closure interval* of J_k.

Computing closure intervals involves geometrical operations in the working space. These operations require a representation of the workspace of every subchain: $J_n \rightarrow J_{k+1}$, k=1...n-p. Considering position and orientation increases the difficulty of this representation. Therefore, we will work with approximations of reachable workspaces (only in position). The choice of volumes to carry out this approximation must be made considering two conditions :

- the estimation of a closure interval obtained from these approximated representations must be conservative. This is required to guarantee that every possible configuration of the system is considered in the roadmap construction.

- the efficiency of RLG is based on a fast calculation of closure intervals. Therefore, intersections of volumes approximating workspaces must be easily computable.

5.1 Approximated Workspaces

We have chosen spherical shapes to approximate the reachable workspaces. This option answers the mentioned requirement for efficiency. They are often used in collision detection as boundary volumes of objects (e.g.[8]) because of the fast computation of their intersections.

The reachable workspace of any mechanism can be bounded by two concentric spheres centered at its base. The radii of these spheres correspond to the length of the chain in full extension and minimum extension. This length is defined by the distance between the origins of the base-frame and the end-frame.

The mechanism in Figure 4 illustrates in 2D the approach for a two-rotation (RR) planar chain. Values of the first joint θ_1 are limited between 0 and $\pi/2$ while θ_2 can variate between $-\pi/2$ and $\pi/2$. The darker region represents the reachable workspace of this chain. This region will be approximated by the ring between r_1 and r_2, with $r_2 = l_1 + l_2$ and $r_1 = |l_1 - l_2|$.

More accurate volumes could be considered for particular classes of mechanisms. We chose such representations to maintain generality and because of a trade-off between accuracy and computing time.

The approximated reachable workspaces of sub-chains $J_n \rightarrow J_{k+1}$, k=1 . . . n-p, are processed in a pre-computing phase before the roadmap construction.

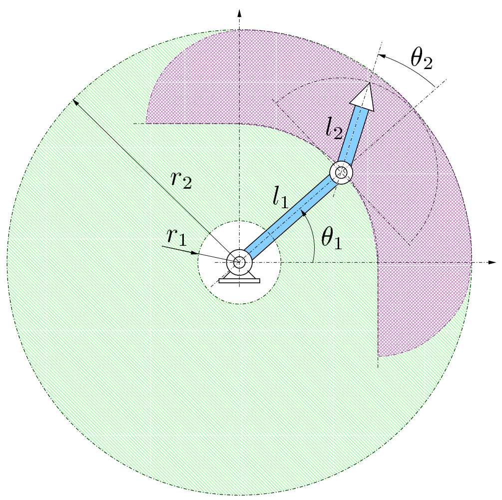


Figure 4: Conservative approximation of the reachable workspace of a RR planar chain.

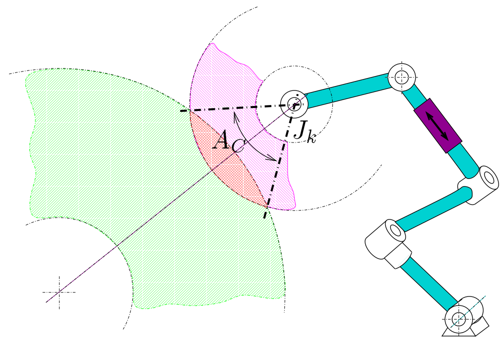


Figure 5: Illustration of the intersections computed for the closure interval estimation in the case of a revolute joint.

5.2 Closure Interval Computation

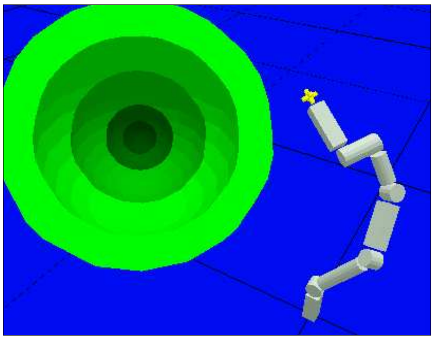
The configuration of the active chain treated by RLG is progressively generated from the base J_1 to the more distal joint J_{n-p}. The process is different for revolute and prismatic joints. The computation of closure intervals focuses on revolute joints.

Revolute Joints : let J_k be a revolute joint of the active chain currently processed by the algorithm (see Figure 5). Let J_{k+n_t+1} be the next revolute joint in the chain; with n_t the number of intercalated prismatic joints. The estimation of the closure interval of J_k is obtained from the intersection of the reachable workspaces of the chains $J_n \rightarrow J_{k+n_t+1}$ and $J_k \rightarrow J_{k+n_t+1}$. Both regions are approximated by the volume between two spheres as explained in Section 4. The intersection of these regions determines the estimated closure interval of J_k which we refer as A_C. If we call A_M the interval between the mechanical limits of J_k, then the random sampling must be made in the set $A_S = A_C \bigcap A_M$. If this $A_S = \emptyset$, the configuration is rejected.

Prismatic Joints : consider now that joint J_k is followed by a sequence of n_t prismatic joints. The conception of a general procedure to reduce the random sampling interval of such sequence is quite complicated. Thus, we sample in the whole interval between joint limits and we test the configuration. If after a given number of tries, the frame $P^{k+n_t}_{k+n_t+1}$ is not reachable by the chain $J_n \rightarrow J_{k+n_t+1}$ the configuration is rejected.

5.3 Results

We made several tests in order to analyze the performance of the RLG algorithm. Figure 6 shows one of the tests performed onto a mechanism composed of 6 rotary joints. It represents the active chain of a loop. The workspace of the passive chain as been modeled as a sphere. The test consists in generating 1000 configurations of the mechanism for which the end-effector (represented by a cross in the image) is inside the sphere. The difficulty of the problem depends on the radius of this sphere. The table compares for different radius values the performance of the RLG algorithm versus a standard sampling technique (as used in [5]). N is the number of sampled configurations. T corresponds to computing time in seconds.



Radius	**Without RLG**		**With RLG**	
	N	**T**	**N**	**T**
200	29995	0.01	3302	<0.01
150	94959	0.02	4458	<0.01
100	441157	0.25	5604	<0.01
50	3549709	4.31	7267	<0.01
25	26080062	38.72	10516	<0.01

Figure 6: Results of one of the RLG tests.

These results show the good performance of RLG. The algorithm is not much affected by the difficulty to generate configurations of the active chain in $W_{active} \cap W_{passive}$. Hence, the better comparative performance of RLG increases with the complexity of the problem.

6 Path Planning

The presented approach has been implemented into the generic motion planning software *Move3D* [18]. Several variants of PRM techniques are integrated in this software. In particular, the visibility-PRM [19] is usually run because of the generation of small roadmaps to capture the connectivity of complex workspaces. The visibility-PRM algorithm keeps only configurations in two cases : when they link several connected components of the roadmap; or when they can not be connected to any of these components. The main advantage is that the number of calls to the local planner significantly decreases in relation to the basic approach. The most expensive step (in time) of the roadmap construction is the local path computation.

Two coordinated manipulation planning problems solved using visibility-PRM are commented next. In the example illustrated in Figure 7, a car (i.e. non-holonomic system) carries a pipe grasped at the other

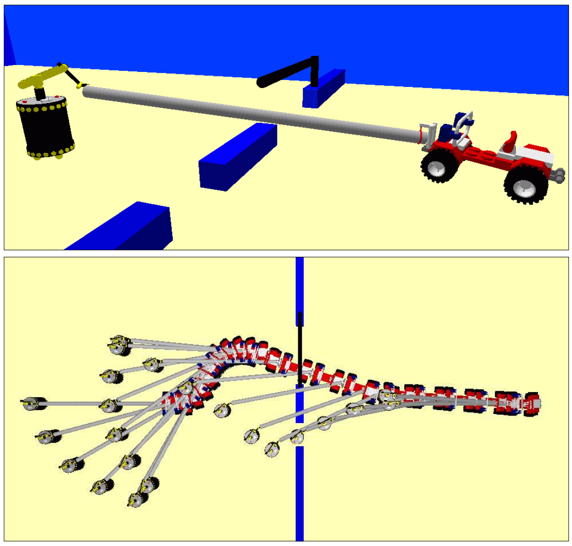


Figure 7: Non-holonomic car tracks a pipe coordinating with a mobile manipulator.

extremity by a mobile manipulator. The whole system (car, robot and pipe) forms a closed kinematic chain modeled by 12 joints. The local planner must consider constraints affecting the motion of the car and closure constraints. The problem is to pass from the one half to the other of an environment divided by a wall with two passages. The only possibility is that the car goes through the wider passage (limited in height by a bar) while the robot goes through the other. It was solved in about 5 seconds of computing time on a Sun Blade 100 workstation.

In the other example (Figure 8) two holonomic mobile manipulators coordinate to transport a plate object. The mechanical system contains 18 d.o.f.. Obstacles avoidance requires extreme deformations of the closed system. Consequently, narrow passages in the configuration-space must be traversed. To find such passages generally needs to try a very high number of random configurations. The solution was found in 25 seconds using RLG while about 10 minutes were necessary using standard random sampling.

7 Conclusions

We have presented a method that combined with existing approaches extends the variety of path planning problems that can be treated by PRM techniques. The implementation of this method has provided very good results treating complex closed mechanisms.

The efficiency of this approach has made possible to use it as one of the key components of a more complex planner for *pick and place* manipulation tasks [16].

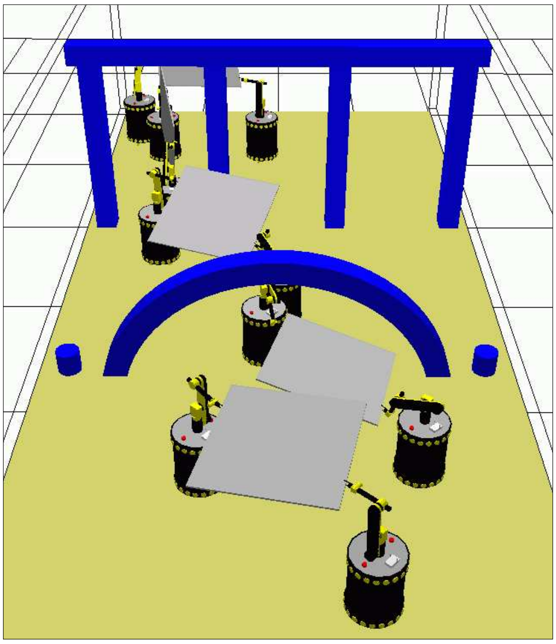


Figure 8: Two mobile manipulators coordinate through narrow passages.

Acknowledgments

We want to thank Marc Renaud for discussions on kinematics. This work has been developed in the framework of MOLOG project [17].

References

[1] N. Amato, O.B. Bayazit, L. Dale, C. Jones and D. Vallejo. OBPRM: An Obstacle-Based PRM for 3D Workspaces. In *Workshop on Algorithmic Foundations of Robotics*, pp. 155-168, 1998.

[2] R. Bohlin and L. Kavraki. Path Planning using Lazy PRM. In *IEEE International Conference on Robotics and Automation*, pp. 521-538, 2000.

[3] V. Boor, M. Overmars and A. van der Stappen. The Gaussian Sampling Strategy for Probabilistic Roadmap Planners. In *IEEE International Conference on Robotics and Automation*, 1999.

[4] K. Gupta and al.. *Practical Motion Planning*. J. Wiley, 1998.

[5] L. Han and N. Amato. A Kinematics-Based Probabilistic Roadmap Method for Closed Kinematic Chains. In *Workshop on the Algorithmic Foundations of Robotics*, 2000.

[6] L. Kavraki, P. Švestka, J.C. Latombe and M. Overmars. Probabilistic roadmaps for path planning in high dimensional configuration spaces. In *IEEE Transaction on Robotics and Automation*, vol. 12, no. 4, pp. 566-580, 1996.

[7] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg and A. Casal. Coordination and Decentralized Cooperation of Multiple Mobile Manipulators. In *Journal of Robotic Systems*, vol 13, pp. 755-64, 1996.

[8] S. Krishnan, A. Pattekar, M. Lin and D. Manocha. Spherical Shell: A Higher Order Bounding Volume for Fast Proximity Queries. In *Workshop on the Algorithmic Foundations of Robotics*, 1998.

[9] J. Kuffner. *Autonomous Agents for Real-time Animation.* PhD thesis, Stanford University, Stanford, CA, December 1999.

[10] J. Kuffner, S. Kagami, M. Inaba and H. Inoue. Graphical Simulation and High-Level Control of Humanoid Robots. In *IEEE International Conference on Intelligent Robots and Systems*, 2000.

[11] J.C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[12] S. LaValle, J.H. Yakey and L. Kavraki. A Probabilistic Roadmap Approach for Systems with Closed Kinematic Chains. In *IEEE International Conference on Robotics and Automation*, 1999.

[13] D. Manocha and Y. Zhu. Kinematic Manipulation of Molecular Chains Subject to Rigid Constraints. In *Proceedings of 2nd International Symposium on Molecular Biology*, pp. 285-294, 1994.

[14] J.P. Merlet. Parallel Robots: Open Problem. In *9th International Symposium of Robotics Research*, Snowbird, Utah, 1999.

[15] J. Nielsen and B. Roth. Formulation and Solution for the Direct and Inverse Kinematics Problem for Mechanisms and Mechatronic Systems. In *Proceedings of the NATO Advanced Study Institute on Computational Methods in Mechanisms*, vol 1, pp. 233-252, 1997.

[16] T. Siméon, J. Cortés, A. Sahbani and J.P. Laumond. A Manipulation Planner for Pick and Place Operations under Continuous Grasps and Placements. To appear in *IEEE International Conference on Robotics and Automation*, 2002.

[17] T. Siméon, J.P. Laumond, C. van Geem and J. Cortés. Computer Aided Motion: Move3D within MOLOG. In *IEEE International Conference on Robotics and Automation*, 2001.

[18] T. Siméon, J.P. Laumond and F. Lamiraux. Move3D: a generic platform for path planning. In *4th Int. Symposium on Assembly and Task Planning*, 2001.

[19] T. Siméon, J.P. Laumond and C. Nissoux. Visibility-based probabilistic roadmaps for motion planning. In *Advanced Robotics Journal*, vol 14(6), 2000.

[20] Petr Švestka. *Robot Motion Planning using Probabilistic Roadmaps.* Universiteit Utrecht, 1997.

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