Submitted to IEEE International Conference on Robotics and Automation 2003

Probabilistic Motion Planning for Parallel Mechanisms

J. Cortés, T. Siméon

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

Abstract

Despite the increasing interest on parallel mechanisms during the last years, few researchers have addressed the motion planning problem for such systems. The few existing techniques lie onto a representation of the workspace of the mechanism (or its boundary). However, obtaining this representation is generally too difficult, only partial solutions exist for particular cases. In this paper we propose a general approach based onto probabilistic motion planning techniques. This approach does not need any modeling of the robot's workspace. It combines random sampling techniques with simple but general geometric algorithms that guide the sampling toward feasible configurations satisfying the closure constraints of the parallel mechanism. The efficiency and the generality of the method are demonstrated onto several complex mechanisms made up with serial or parallel associations of Stewart platforms, or created with several redundant robots manipulating an object.

1 Introduction

A parallel manipulator is a mechanism in which the end-effector is connected to the base by at least two independent kinematic chains [19]. The most representative parallel manipulator is the six-degrees-of-freedom (d.o.f.) mechanism known as Stewart platform [26, 8]. This definition can be also applied to more complex multi-closed-chain mechanisms formed by several manipulators handling an object. In this paper we address the problem of planning collision-free motions for such general parallel mechanisms.

The few existing techniques for trajectory validation and motion planning of parallel mechanisms [22] use a representation of the workspace of the mobile platform. The difficulty to compute such representations limits the generality of these approaches, like with deterministic motion planning techniques [16] that first relied on an exact model of the collision-free configuration-space CS_{free} of the mobile system. Probabilistic motion planning techniques [12] do not need to build a model of the space where they are applied. This property was the key of their success during the last decade. However, the closure constraints of parallel mechanisms (i.e. multiple closed-chains) remain an important challenge for probabilistic motion planning methods and the few existing approaches [17, 11, 6] are mostly limited to single-loop mechanisms. We present an extension of our work on motion planning for closed-kinematic chains [6] using the PRM framework, to efficiently deal with multi-closed-chain mechanisms such as general parallel robots.

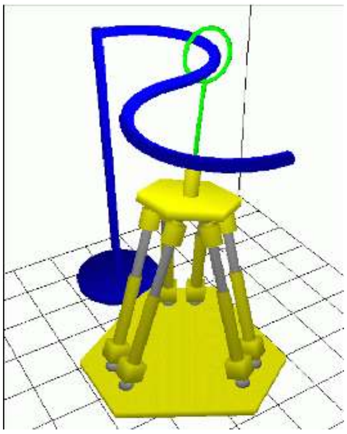

Figure 1: Example of motion planning problem for a Stewart platform.

A first application of the approach is to capture the self-collision-free workspace of the parallel mechanism into a small data structure (a random visibility roadmap [25]). Once computed for a given mechanism, this data structure can be used to generate in real-time valid motions avoiding self-collisions between the links of the mechanism. In presence of obstacles, the proposed approach also allows us to solve motion planning problems like the one illustrated in Figure 1 where the path to extract the ring mounted onto a Stewart platform from the "s-shaped" obstacle is computed in only a few seconds.

Section 2 first gives a brief overview of probabilistic motion planning techniques. In this same section, we discuss about the extension of these techniques to handle closed kinematic chains. Parallel mechanisms are presented in Section 3. Our approach for sampling random configurations of such systems is explained in Sections 4 and 5. Results in Section 6 show the generality of the method through different applications for various kinds of systems.

2 Probabilistic Motion Planning

Probabilistic motion planning techniques appeared in the last decade as an alternative to deterministic approaches. In particular, *Probabilistic RoadMap (PRM)* methods [12] have been mostly developed. These techniques have demonstrated to be efficient and general tools for motion computing.

PRM Principle. The general PRM principle is to construct a graph (roadmap) that captures the topology of CS_{free}. The nodes are randomly sampled configurations satisfying intrinsic conditions in this space (e.g. collision-free). The edges are short feasible paths (*local paths*) linking "nearby" nodes.

PRM Variants. Several algorithms [1, 3, 27, 25, 2] have been proposed sharing this basic idea. These methods mostly differ from their sampling strategies. In particular, the *visibility-PRM* approach [25] is used in our solution. The algorithm building the graph only keeps the sampled 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 to compute a smaller roadmap which significantly decreases the number of calls to the local planner (the most expensive step of the roadmap construction) compared to other approaches. Figure 2 shows two roadmaps for the same 2D environment. The left one has been computed by a *basic*-PRM algorithm that keeps every valid sampled configuration. The right one, obtained by the visibility approach, encodes the same information in a much smaller structure.

The mentioned PRM techniques are called *multiple-query*. Once the roadmap is computed, motion planning queries are solved by connecting the start and goal configurations to the graph and searching a path in it. Other algorithms dedicated to solve *simple planning queries* have been developed from the same principles than PRM (e.g. *RRT* [15], *SBL-PRM* [23]).

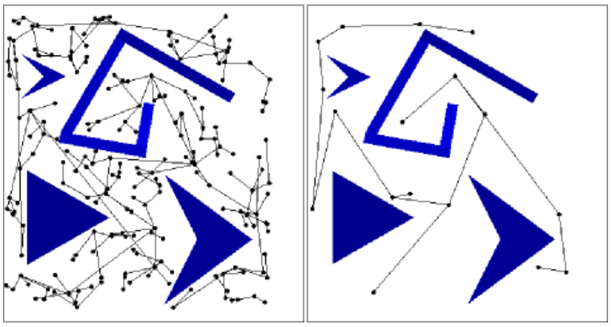

Figure 2: Basic-PRM and visibility-PRM in the same 2D environment.

All techniques above require the generation of random configurations of the mechanism. This is a trivial process in the case of open kinematic chains. On the contrary, when the mechanism contains closed chains, samples must be generated into a sub-dimensional manifold of CS. The difficulty to compute (and to connect) such configurations remains a challenge for the application of probabilistic motion planners.

PRM and Closure Constraints. Only a few works extending the PRM framework to deal with closed chain mechanisms can be found in the literature [17, 11, 6]. The approach in [6] demonstrates good performance onto complex 3D closed chains involving tenths of d.o.f.. Each single-loop in the mechanism is broken (as initially proposed in [11]) into two chains (*passive* and *active*). The random node generation combines a sampling technique called *Random Loop Generator (RLG)* with forward kinematics for the active chain and inverse kinematics for the remaining (passive) part of the loop in order to force the closure. When computing the 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 using point to point inverse kinematics.

The main interest of RLG is that it produces random samples for the active chain that have a high probability to be reachable by the passive part. The algorithm in [6] performs well on independent single-loops and was also applied to some cases of multi-loops. However, this approach requires an extension to efficiently handle more general closed-chain mechanisms. Parallel mechanisms are a more complex instance that presents particular interest.

3 Parallel Mechanisms

Description. A parallel mechanism is composed of a base $\mathcal{B}$, a platform $\mathcal{P}$ and n kinematic chains $\mathcal{M}_i$ linking them. We call A_{B_i} and A_{P_i} the frames corresponding to the connections of each $\mathcal{M}_i$ to $\mathcal{B}$ and $\mathcal{P}$ respectively. F_B and F_P are the frames associated to $\mathcal{B}$ and $\mathcal{P}$ (see Figure 3).

The situation of $\mathcal{P}$ is defined by a vector $q_{\mathcal{P}}$=$\{x, y, z, \theta_x, \theta_y, \theta_z\}$. The three firts elements represent the position of F_P with respect to F_B. The orientation is given by three consecutive rotations around the axes of F_P [1]. The platform is considered to be the end-effector of a parallel mechanism. Hence, an equivalence can be established between situations of $\mathcal{P}$ and points of the workspace of the system.

[1] The approach is valid for other representations of the orientation (e.g. Euler angles).

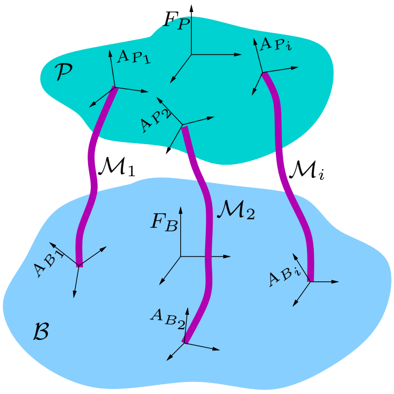

Figure 3: General description of a parallel mechanism.

Workspace. Normally, the workspace WS_P of a parallel mechanism is computed from the workspaces WS_{M_i} of the chains $\mathcal{M}_i$ and the dimensions related to $\mathcal{P}$. The difficulty is that WS_P can not be decoupled into two three-dimensional (graphically representable) sub-spaces because of the dependence between position and orientation of the end-effector. Therefore, only sub-sets of the workspace may be represented. Most of the existing works are limited to the determination of some particular sections of the positional workspace with constant orientation of the platform [10, 18]. Other techniques also compute the feasible rotations of the platform around a fixed point [21].

Configuration. The configuration of a parallel mechanism is defined by the joint values of the chains $\mathcal{M}_i$ (and the situation of $\mathcal{P}$). Configurations satisfying the closure constraints could be easily computed from a model of its workspace. Hence, a feasible situation $q_{\mathcal{P}}$ could be directly obtained from this representation. The situation of the connection-frames A_{P_i} w.r.t. the A_{B_i} would be given by $q_{\mathcal{P}}$, and the configuration of the chains $\mathcal{M}_i$ linking these frames could be then computed by inverse kinematics techniques.

However, modeling the workspace of a general parallel mechanism remains an open problem [20]. In next section, we describe an algorithm that generates random configurations of a general parallel mechanism without requiring the explicit computation of WS_P.

4 Random Configuration Sampling for Parallel Mechanisms

We propose a general approach that combines random sampling techniques with simple geometric operations for generating random configurations of parallel mechanisms. *Spherical shells* approximating the WS_{M_i} are used to progressively compute the situation $q_{\mathcal{P}}$ of $\mathcal{P}$. The algorithm first generates the position parameters of $q_{\mathcal{P}}$ and then it computes the rotation parameters. Such obtained situations of $\mathcal{P}$ correspond to random samples in a conservative approximation of WS_P. Then, the existence of a feasible configuration is checked for each chain $\mathcal{M}_i$ linking A_{B_i} and A_{P_i}. The configuration of the parallel mechanism is kept when all the $\mathcal{M}_i$ connect the base and the platform, else the process is iterated. Next paragraphs detail the main features of the approach.

WS_{M_i} Approximation. In a similar way than in [6], spherical shells bounding the reachable workspace (only in position) of the chains $\mathcal{M}_i$ are used. A trade-off between accuracy and computing time justifies this choice. A spherical shell is defined by the intersection of the volume between two concentric spheres and a cone whose vertex coincides with their common center. Spherical shells are used in collision detection as boundary volumes of objects because of the fast computation of their intersections [14]. Parameters characterising the spherical shell are derived from the features of the chains $\mathcal{M}_i$. The external and internal radii correspond with the maximun and minimum extension. The cone cutting the full shell can be computed from a further analysis of the joint limits and of the link lengths.

Platform Position. Given the fixed situation of the A_{B_i} in F_B, a plane $\Pi_{xy_{base}}$ can be computed by interpolating the position of the frame origins (when there are more than two). We next explain the method to generate the position of the platform w.r.t. the frame F_Π associated to this plane.

First, a rectangle C_{xy} approximating the orthogonal projection of WS_P on $\Pi_{xy_{base}}$ is computed as follows. The spherical shell approximating each WS_{M_i} is augmented by the distance from A_{P_i} to F_P. The projection of the external portion of sphere on $\Pi_{xy_{base}}$ corresponds in general to an ellipse. C_{xy} is the rectangle bounding the intersection of these ellipses. Figure 4 illustrates this process in the case of four chains $\mathcal{M}_i$. For clarity purpose, we have only represented the external surface of the augmented shell of the chain $\mathcal{M}_1$.

The generation of a situation of $\mathcal{P}$ begins by randomly sampling a point p_{xy} in C_{xy}. Then, the in-

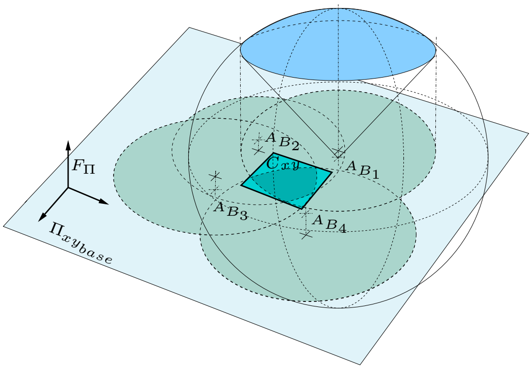

Figure 4: Illustration of the computation of C_{xy}.

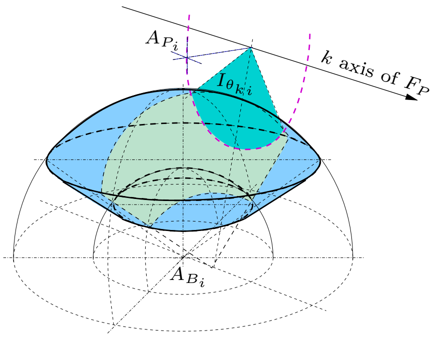

Figure 5: Illustration of the computation of $I_{\theta_{k_i}}$.

tersection of the line perpendicular to $\Pi_{xy_{base}}$ passing through p_{xy} with each one of the augmented spherical shells is computed. When one or several volumes are not intersected, a new point p_{xy} must be sampled. The result of this operation are one or several intervals in z (relative to $\Pi_{xy_{base}}$) for each $\mathcal{M}_i$. The intersection of such intervals represents a conservative approximation of the set of reachable positions of the platform for a given p_{xy}. The z coordinate of $\mathcal{P}$ is generated by randomly sampling in this set.

Platform Orientation. For a given position of $\mathcal{P}$, its orientation is generated by progressively computing the three elementary rotations. We next explain the process for one rotation.

The rotation of $\mathcal{P}$ around an axis k produces a circular motion of each A_{P_i}. The intersection of the spherical shell approximating each WS_{M_i} with the circle generated by its corresponding A_{P_i} is computed. The result is the set $I_{\theta_{k_i}}$ of values of the rotation parameter making A_{P_i} reachable (in position) by $\mathcal{M}_i$ (considering our approximation). Figure 5 illustrates this operation. When the circle intersects all the shells, the value of the rotation θ_k is obtained by randomly sampling in the interval(s) resulting of the intersection of the $I_{\theta_{k_i}}$ sets. The process is iterated when any of the computed intersections is null.

$\mathcal{M}_i$ Configuration & Validation. The conservativeness of the approach is essential in order to guarantee that WS_P will be completely sampled. However, the validity of each configuration must be tested. A configuration is valid when the situation of the platform induces feasible joint values of all the chains $\mathcal{M}_i$.

For parallel manipulators (Stewart platform type) obtaining the configuration of the $\mathcal{M}_i$ for a given situation of $\mathcal{P}$ is straightforward due to the simple nature of these chains. In the general context this process is more complex. Each triplet $\{A_{B_i},A_{P_i},\mathcal{M}_i\}$ is treated as a closed kinematic chain. When $\mathcal{M}_i$ is a non-redundant system, its configuration can be directly obtained by inverse kinematics. In case of redundancy, we use the RLG algorithm presented in [6] to generate it.

Note that this approach also allows to handle particular cases such as mechanisms where the position of the platform is fixed w.r.t. the base and/or the rotational mobility is limited (i.e. rotating only around one or two axes).

5 Complex Mechanisms

The presented approach has been extended to handle more complex systems obtained by the associations of parallel mechanisms. It has been also adapted to a particular case of highly-redundant chains $\mathcal{M}_i$.

Associations. When n parallel mechanisms are connected in ***series*** (see left image in Figure 8), each platform $\mathcal{P}_i$, i=1..$n-1$, becomes the base for the next platform. The process of generation and validation of the configuration is progressively achieved for each platform starting from the base ($\mathcal{P}_1$) to the top ($\mathcal{P}_n$). When the sampled situation of a given $\mathcal{P}_i$ is not valid, the process does not re-start completely. It is only iterated from $\mathcal{P}_{i-1}$. When several parallel mechanisms are disposed in ***parallel*** (see right image in Figure 8) they form a "main" parallel system. Each mechanism can be considered as a chain $\mathcal{M}_i$ of the main system. Therefore, their platforms become passive elements of the whole mechanism.

Mobile $\mathcal{M}_i$ bases. Such case occurs for example when the $\mathcal{M}_i$ chains correspond to mobile manipulators (i.e. articulated system composed of an arm mounted on a mobile base [13], see Figure 10). Spherical shapes are suitable to bound the reachable workspace of the arm, but not for the whole mobile manipulator. We have extended the approach to the case where the A_{B_i} can freely move on parallel planes. While computing $q_{\mathcal{P}}$, these frames are considered to be placed at the position that maximizes the variation of each parameter. Then, feasible random configurations of the chains $\mathcal{M}_i$ are computed by RLG.

6 Results

The approach has been implemented into the motion planning software *Move3D* [24]. In this section we comment some of the obtained results for very different parallel mechanisms. Numerical results correspond to tests performed with a Sun Blade 100 workstation.

Self-collision-free motions. The first experiment aims to demonstrate the performance of the approach to compute self-collision-free motions of the Stewart platform. The roadmap computed for this mechanism can be used to generate such motions in real-time. The left image in Figure 6 shows an example of the self-collision configurations to be avoided. The graph illustrated in the other image of this figure was computed by the visibility-PRM [25] approach in 22

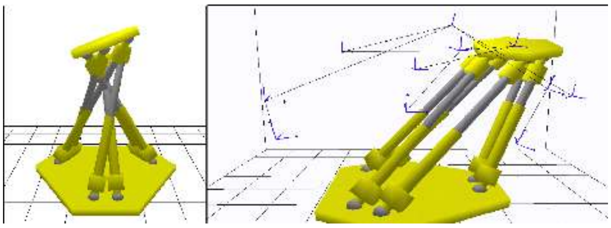

Figure 6: Illustration of the self-collision-free visibility-PRM computed for a Stewart platform.

seconds. It only contains one connected component made up with 11 configurations. This small graph covers more than the 99.99% of the robot workspace. The roadmap construction required the generation of 17442 configurations of the mechanism, of which 2328 were found to be collision-free. Using our sampling strategy, 30840 situations of the platform were tested for the generation of valid configurations (more than 50% of success). With similar tests performed using standard random sampling techniques to generate the situation of the platform [2], less than a 2% of the samples produced valid configurations of the mechanism. This illustrates the important gain (about 25 times faster) using the proposed sampling approach.

Motion planning for Stewart mechanisms. The second example illustrated by Figure 7 shows a constrained motion planning problem for the Stewart platform. The figure shows the start and goal configurations and the trace of the solution path. Note that the motion requires extreme deformations of the mechanism. A graph containing this solution was computed in 60 seconds. Once computed, it allows to process motion planning queries in some hundredths of second. The two images in Figure 8 correspond to motion planning problems involving associations of parallel mechanisms. The manipulator of the left image is a model of the Logabex-LX4 [5]. The arm is composed of four Stewart platforms connected in series. Motion planning queries solving problems where the manipulator with the grasped bar changes from one to another opening of the bridge were computed by RRT algorithm [15] in a few seconds. The right image illustrates an example where two sets of three Stewart platforms cooperate in a assembly task. This type of association as been proposed in [4] for the manipulation of large objects. The motion to assemble the two puzzle-like parts was computed in only 15 seconds.

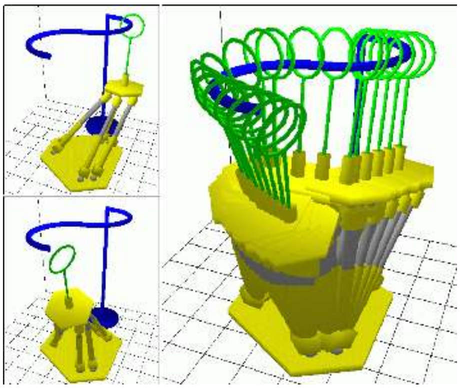

Figure 7: Sequence of the solution motion for the s-bar problem.

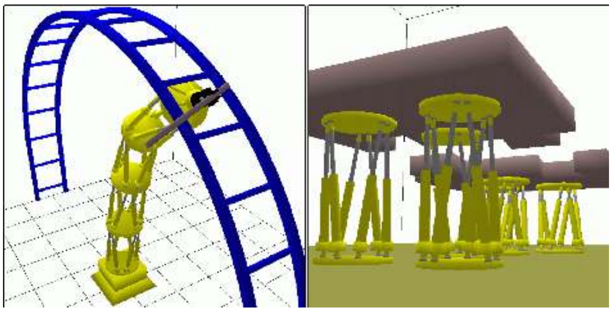

Figure 8: Examples of associations of Stewart platforms.

Parallel systems including manipulator arms. The two last examples show the generality of the approach. In both cases, the mechanism consists of several robotic arms grasping an object. The problem illustrated in Figure 9, where four *6R* manipulators have to unhook an object and to insert it into the cylindrical axis, was solved using RRT in less than 1 second. The last example (see Figure 10) combines two types of difficulty. First, the system composed by the three holonomic mobile manipulators and the piano is a very complex parallel mechanism (9 d.o.f. for each $\mathcal{M}_i$ chain). Also, the complexity of the scene makes the validation of collision-free configurations and local paths harder. A graph that permits to rapidly compute any feasible motion in this scene was computed using the visibility-PRM approach in about 5 minutes. In this example, the redundancy of the manipulators ($\mathcal{M}_i$ chains) is treated by the RLG algorithm as explained in Section 5.

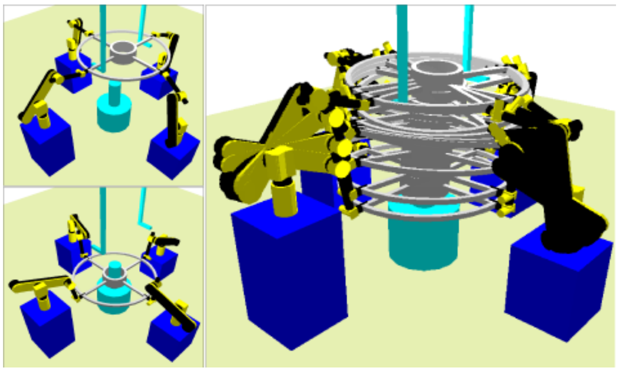

Figure 9: Four robotic arms manipulate an only object.

[2] Samples are taken in a six-dimensional-box bounding the space of feasible situations of $\mathcal{P}$.

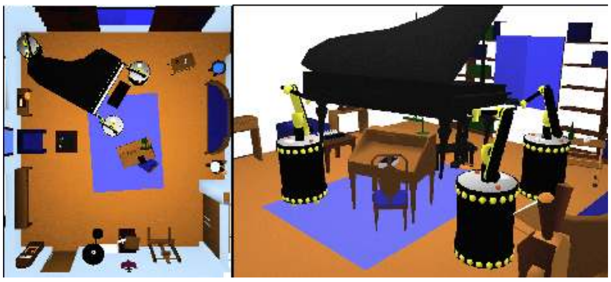

Figure 10: A piano mover problem with three cooperating mobile arms.

7 Conclusions

The proposed approach allows to extend the PRM framework to efficiently handle complex mechanisms with multiple closed-chains. Our aim is to reach the highest level of generality. The approach can deal with the most general definition of parallel mechanisms and its efficacy was demonstrated onto complex examples (e.g. serial/parallel associations of Stewart platforms, parallel system with redundant chains). A possible improvement of this approach could be to integrate constraints for avoiding singular configurations along the trajectory [7, 20].

We are currently investigating the application of our closed-chain PRM approach to highly-articulated chains encountered in molecular models. Hence, tools for analyzing the motion of loops in protein structures [9] could help biologists to better understand the protein folding process.

Acknowledgments. This paper has greatly benefited from discussions with Jean-Pierre Merlet and Lluis Ros.

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*, 1998.

[2] R. Bohlin and L. Kavraki. Path Planning using Lazy PRM. In *IEEE Int. Conf. on Robotics and Automation*, 2000.

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

[4] K.S. Chai and K. Young. Designing a Stewart Platform-based Cooperative System for Large Component Assembly. In *IEEE Int. Conf. on Methods and Models in Automation and Robotics*, 2001.

[5] S. Charentus. *Modélisation et commande d'un robot manipulateur redondant composé de plusieurs plateformes de Stewart.* PhD thesis. Rapport LAAS n.90146, 1990.

[6] J. Cortés, T. Siméon and J.P. Laumond. A Random Loop Generator for planning the motions of closed kinematic chains with PRM methods. In *IEEE Int. Conf. on Robotics and Automation*, 2002.

[7] B. Dasgupta and T.S. Mruthyunjaya. Singularity-free path planning for the Stewart platform manipulator. In *Mechanism and Machine Theory*, vol. 33(6), pp. 711-725, 1998.

[8] B. Dasgupta and T.S. Mruthyunjaya. The Stewart platform manipulator: a review. In *Mechanism and Machine Theory*, vol. 35(1), pp. 15-40, 2000.

[9] A. Fiser, R.K. Do, and A. Sali. Modeling of Loops in Protein Structures. In *Protein Science* 9, 2000.

[10] C. Gosselin. Stiffness mapping for parallel manipulators. In *IEEE Transaction on Robotics and Automation*, vol. 6(3), pp. 377-382, 1990.

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

[12] 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(4), pp. 566-580, 1996.

[13] 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.

[14] 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.

[15] J. Kuffner and S. Lavalle. RRT-Connect: an efficient approach to single-query path planning. In *IEEE Int. Conf. on Robotics and Automation*, 2000.

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

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

[18] O. Masory and J. Wang. Workspace evaluation of Stewart platforms. In *Advanced Robotics Journal*, vol. 9(4), pp. 443-461, 1995.

[19] J.P. Merlet. *Parallel robots.* Kluwer, Dordrecht, 2000.

[20] J.P. Merlet. Parallel Robots: Open Problem. In *9th Int. Symp. of Robotics Research*, 1999.

[21] J.P. Merlet. Determination of the orientation workspace of parallel manipulators. In *Journal of Intelligent and Robotic Systems*, vol. 13, pp. 143-160, 1995.

[22] J.P. Merlet. *Manipulateurs parallèles, 7eme partie : Vérification et planification de trajectoire dans l'espace de travail.* Rapport de Recherche 1940, INRIA, 1993.

[23] G. Sanchez and J.C. Latombe. A Single-Query Bi-Directional Probabilistic Roadmap Planner with Lazy Collision Checking. In *Int. Symp. on Robotics Research*, 2001.

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

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

[26] D. Stewart. A platform with 6 degrees of freedom. In *Proc. of the Institution of mechanical engineers*, 180 (Part I, 15), pp. 371-386, 1965.

[27] S. Wilmarth, N.Amato and P.Stiller. MAPRM: A Probabilistic Roadmap Planner with Sampling on the Medial Axis of the Free Space. In *IEEE Int. Conf. on Robotics and Automation*, 1999.