

A sampling-based path planner for dual-arm manipulation

Mokthar Gharbi, Juan Cortés, Thierry Simeon

▶ **To cite this version:**

Mokthar Gharbi, Juan Cortés, Thierry Simeon. A sampling-based path planner for dual-arm manipulation. 2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Jul 2008, Xian, China. pp.383-388. hal-01986353

HAL Id: hal-01986353
https://laas.hal.science/hal-01986353

Submitted on 18 Jan 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A sampling-based path planner for dual-arm manipulation

M. Gharbi, J. Cortés, T. Siméon

***Abstract*— This paper presents a method to compute collision-free coordinated manipulation paths for multi-arm robot systems. The method extends previous work [8] by incorporating an explicit treatment of singular configurations, which permit to find solution paths requiring the robot reconfiguration. Such reconfiguration motions are neglected in most of related work. The performance of the planner is analyzed through experiments with academic examples. The generality of the approach is demonstrated by its application to a model of the complex DLR's Justin robot.**

I. INTRODUCTION

Multi-arm robot systems have been developed in diverse fields such as industrial manufacturing [28], medical robotics [1], and humanoid robotics [18]. Such systems must often perform coordinated manipulation tasks, in which several arms move simultaneously holding an object. Problems for coordinated multi-arm manipulation have mostly been formulated from the point of view of control [14], [20], [9], [6], while the path planning issue has been rarely addressed.

The difficulty for planning coordinated manipulation paths comes from the motion constraints induced by the multiple grasping. Indeed, a virtual closed-chain mechanism is formed when several manipulators grasp the same movable object. These kinematic loop-closure constraints have an important effect on the topology of the space that has to be explored for finding paths (see Section II). Generally, path planning methods have to be extended for treating such constraints.

Few approaches to closed-chain path planning have been proposed. Exact methods are able to provide a complete solution (i.e. they find a solution path if one exists, and report failure otherwise), but their applicability is either limited by their computational complexity [21], [5], [2], or because they are devised for particular classes of mechanisms [26], [11]. The resolution-complete method described in [19] presents more general skills, but it is still limited in practice to low-mobility closed-chain mechanism. Practical approaches have been developed using sampling-based path planning algorithms [15], [27], [10], [8], [24]. Sampling-based planners are able to solve complex problems in high-dimensional spaces with very low computational cost. The drawback of these planners with respect to the aforementioned methods is that they hold a weak completeness guarantee, namely probabilistic completeness [16]. Nevertheless, such a weakness is generally not a handicap in practice.

This paper builds on previous work described in [8]. It presents the extension to connect self-motion manifolds with sets of lower dimension corresponding to singular configurations. This extension permits to solve coordinated manipulation problems requiring the reconfiguration of one or several robots. To the best of our knowledge, this is the first time this reconfiguration issue is clearly addressed in a general sampling-based path planning framework.

Let us consider the academic example illustrated in Figure I. The system involves 2 planar 3R manipulators holding a movable object. Two path planning problems are represented in the figure. Problem A can be solved without reconfiguring any of the robots. In problem B however, the presence of the obstacle in the bottom left part of the workspace makes both arms need to go from "elbow down" to "elbow up" configuration.

The proposed approach (Section IV) is based on a decomposition of the (virtual) closed-chain mechanism into active and passive parts, which yields a particular parametrization of the configuration space described in Section III. The method can be applied to general multi-arm system. Nevertheless, for ease of explanation, notation and examples are limited to dual-arm systems.

Section V presents an empirical performance analysis using the academic problems illustrated in Figure I. Then, Section VI describes the application of the technique for planning coordinated manipulation paths with a model of the DLR's Justin robot [18]. The results obtained for this system show the scalability of the approach to complex mechanisms in a three-dimensional workspace.

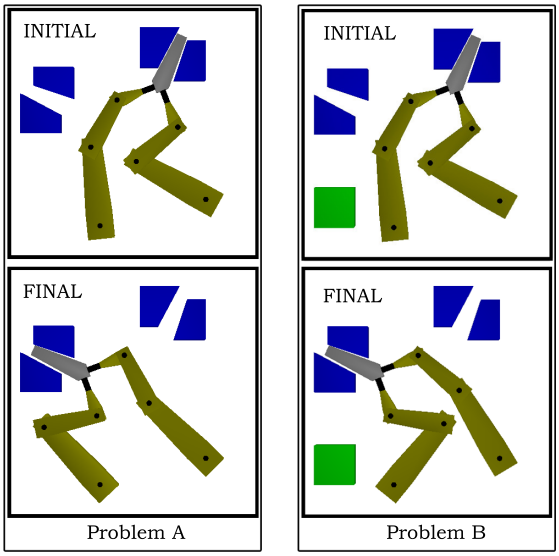


Fig. 1. Academic example of path planning problems for coordinated manipulation. The solution to problem B requires the reconfiguration (elbow up/elbow down) of both manipulators.

The authors are with LAAS-CNRS, Université de Toulouse, F-31077 Toulouse, France. mgharbi@laas.fr, jcortes@laas.fr, nic@laas.fr

II. PROBLEM FORMULATION

The path planning problem consists in finding a feasible path between two given configurations of a mobile system. Feasible paths have to satisfy intrinsic constraints of the system (e.g. mechanical design constraints, kinematic constraints), as well as constraints that arise from the environment (e.g. collision avoidance). Using the notion of configuration space $\mathcal{C}$ [17], the problem is reduced to explore the connectivity of the subset $\mathcal{C}_{feas}$ of the feasible configurations.

Coordinated manipulation requires to compute the motion of several robots simultaneously holding an object. Such a multiple grasping induces motion constraints that can be modeled as kinematic loop-closure constraints (see details in Section III). These constraints significantly increase the difficulty of path planning, since they make $\mathcal{C}_{feas}$ have a complex topology. Indeed, configurations of a closed-chain mechanism have to satisfy equations of the form $f(\mathbf{q})=0$, where $\mathbf{q}$ are the joint variables, and f is a system of non-linear equations. Configurations satisfying loop-closure equations form several manifolds $\mathcal{M}_i$, which can be connected through sets of lower dimension $\mathcal{S}_k$ [3], [25]. The $\mathcal{M}_i$ are called self-motion manifolds and the $\mathcal{S}_k$ are sets of singular configurations. Figure 2 illustrates a fictive example with three joint variables $\{\theta_1,\theta_2,\theta_3\}$. Let us consider a function of the form $f(\theta_1,\theta_2,\theta_3)=0$, representing loop-closure constraints. This function maps to several surfaces embedded in the joint-space $\mathcal{Q}$. Such surfaces are the different self-motion manifolds $\mathcal{M}_i$. In this example, $\mathcal{M}_1$ and $\mathcal{M}_2$ intersect at a singular set $\mathcal{S}$. $\mathcal{Q}_{obst}$ represent the obstacle regions in the joint space.

Planning coordinated manipulation motions therefore requires to find feasible collision-free paths that may lie on different manifolds, whose connection needs to explore lower-dimensional subsets of the configuration space.

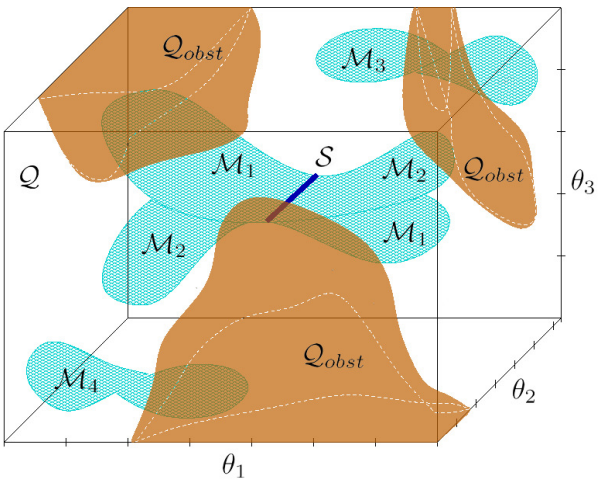


Fig. 2. Illustration of the topology of the configuration space under kinematic loop-closure constraints.

III. MODEL AND PARAMETRIZATION

A virtual closed-chain mechanism is formed when several manipulators simultaneously hold an object. The grasps of the movable object can be modeled either as a fixed attachment (i.e. the object and the gripper form a rigid body) or by a joint, if some degrees of freedom are allowed between the gripper and the object. Hereafter, explanations will be limited to the case of two manipulators holding an object with firm grasps. Note however that the proposed approach is general.

Figure 3 shows the model of two planar 3R manipulators grasping an object. The composed system (together with the ground link) can be considered as a single loop linkage. It can also be modeled as a multi-loop linkage like a parallel mechanism, with the movable object corresponding to the platform and the manipulators to the legs of the mechanism. In this work, we have chosen the latter option because metrics for path planning are generally easier to device in this case. The configuration $\mathbf{q}$ of the system is defined by the location of the object and the configuration of each manipulator: $\mathbf{q} = \{\mathbf{q}_{\text{Obj}},\mathbf{q}_{\text{Arm1}},\mathbf{q}_{\text{Arm2}}\}$. These parameters are involved in the loop-closure equations, and therefore, the values of some of them (the dependent parameters) depend on the values of the others (the independent parameters). In general, the number of independent parameters is 3 for planar mechanisms and 6 for spatial mechanisms. Following the nomenclature of related works [10], [8], we call *active* and *passive* variables respectively the independent and dependent parameters. The terms active and passive indicate their role within the path planning approach described in next section.

Let us consider that the parameters defining the object location $\mathbf{q}_{\text{Obj}}$ are active variables. Then, if the arms are non-redundant manipulators, their joint variables are the passive variables. If the arms are redundant mechanisms, a number of joint variables corresponding to the degree of redundancy are also considered to be active variables. In summary, active and passive variables are defined as follows:

$$\mathbf{q}^{\text{a}} = \{\mathbf{q}_{\text{Obj}},\ \mathbf{q}^{\text{a}}_{\text{Arm1}},\ \mathbf{q}^{\text{a}}_{\text{Arm2}}\}\ ,\ \mathbf{q}^{\text{p}} = \{\mathbf{q}^{\text{p}}_{\text{Arm1}},\ \mathbf{q}^{\text{p}}_{\text{Arm2}}\}$$

Obtaining $\mathbf{q}^{\text{p}}$ from $\mathbf{q}^{\text{a}}$ requires solving two inverse kinematics (IK) problems, one for each arm, which we refer to as IK_{Arm1} and IK_{Arm2}. Since these IK problems involve non-redundant mechanisms, they have a finite number of solutions (for regular configurations). We assume that closed-form solutions are available for these IK problems, and that different solution classes (e.g. forward/backward, elbow

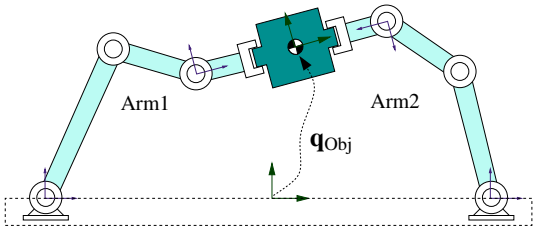


Fig. 3. Two planar 3R manipulators grasping an object. The composed system forms a closed kinematic chain.

up/elbow down, flip/no flip) can be characterized. We refer to each particular solution of IK_{Arm1} and IK_{Arm2} as $\mathbf{q}^{\text{p}}_{\text{Arm1},i}$ and $\mathbf{q}^{\text{p}}_{\text{Arm2},j}$ respectively. We also assume that *singular configurations* of the manipulators, permitting to change the IK solution class, are known and can be characterized. We call $\mathcal{S}_{i,i'}$ a singularity set between two IK solution classes i and i' of one of the arms.

IV. APPROACH

A. Overview

The method proposed in this paper for computing dual-arm coordinated manipulation motions is based on the extension of PRM-based planners to closed-chain mechanisms [10], [8]. The main contribution with respect to previous work concerns an explicit treatment of singular configurations, which permit to find solution paths requiring the robot reconfiguration.

The main idea of our approach is to divide the roadmap construction into two stages. In the first one, "parallel" roadmaps are constructed on different *layers* of $\mathcal{C}_{feas}$. Considering the parametrization described in the preceding section, each layer corresponds to a single combination of IK solution classes of the two manipulators. The number of layers depends on the maximum number of regular solutions of the IK problems. If n_{Arm1} and n_{Arm2} are the maximum number of IK solutions, then the number of layers is $n_{\text{Arm1}} \times n_{\text{Arm2}}$. We call $G_{i,j}$ the roadmap constructed on each layer, with $i = 1 \dots n_{\text{Arm1}}$ and $j = 1 \dots n_{\text{Arm2}}$. The second stage consists in connecting all those generated roadmaps $G_{i,j}$ through singular configurations. Note that these two stages (layer construction and connection) can be interleaved. They are described in Subsections IV-B and IV-C.

Once the multi-layer roadmap is constructed, path planning queries may involve searching for sub-paths on the different layers in order to find solutions requiring the reconfiguration of one or both manipulators. The process is explained in Subsection IV-D.

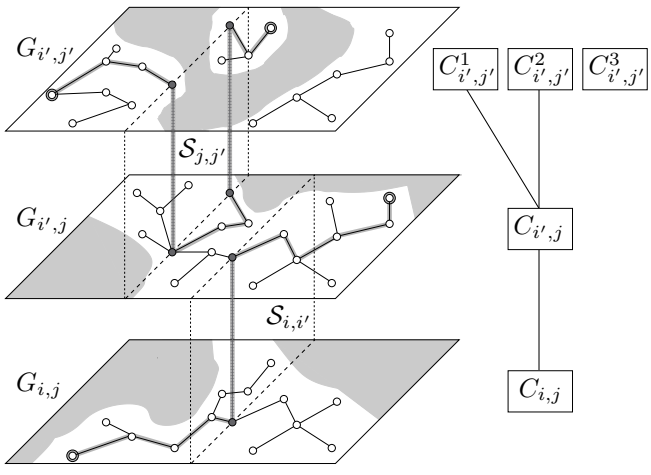


Fig. 4. PRMs constructed on three layers of $\mathcal{C}$ corresponding to different self-motion manifolds. These "parallel" roadmaps can be connected through singular nodes (in dark gray), here represented on perpendicular planes. The graph on the right side represents the connectivity of the connected components on different layers through singular configurations.

Figure 4 illustrates the approach. It shows PRMs constructed on three layers. Two singularity sets involving the reconfiguration of each arm are represented in form of perpendicular planes crossing the layers. The graph on the right side of the figure encodes the connection of connected components on different layers through singular configurations.

B. Constructing PRM layers

Following the principle described in [10], [8], the PRM layers $G_{i,j}$ are constructed acting on the active variables $\mathbf{q}^{\text{a}}$, and solving IK problems for obtaining the corresponding values of the passive variables $\mathbf{q}^{\text{p}}$. Details on the generation of nodes and edges are given below.

1) Nodes: The active variables $\mathbf{q}^{\text{a}}$, which involve the movable object and the redundant joints of each arm, are sampled using the extension of the RLG algorithm described in [7]. Instead of using uniform random sampling, this algorithm performs an iterative process using simple but general geometric operations that guide the sampling toward regions with a high probability of finding solutions for the passive variables $\mathbf{q}^{\text{p}}$ when solving the IK problems. A configuration can be obtained on each layer $G_{i,j}$ by combining the values of the active parameters an the several solutions to the IK problems: $\mathbf{q}_{i,j} = \{\mathbf{q}^{\text{a}}, \mathbf{q}^{\text{p}}_{i,j}\}$, with $\mathbf{q}^{\text{p}}_{i,j} = \{\mathbf{q}^{\text{p}}_{\text{Arm1},i}, \mathbf{q}^{\text{p}}_{\text{Arm2,j}}\}$.

Two strategies are then possible to insert nodes in the multi-layer roadmap. The first one is to systematically generate a node on each layer if the corresponding configurations is collision-free. The other is to (randomly) select one layer and to generate a node only on it. Results obtained with the two strategies are discussed in Section V.

2) Edges: Connections are tested between neighbor nodes on each layer using a local planner. This local planner is limited to act on the active variables. If we assume that the manipulators are not subject to differential constraints, a straightforward linear interpolation can be used as local planner. Given a small discretization step, the passive variables are computed for each intermediate configuration along the local path by solving the two IK problems and selecting the corresponding solutions $\{\mathbf{q}^{\text{p}}_{\text{Arm1},i}, \mathbf{q}^{\text{p}}_{\text{Arm2,j}}\}$.

Note that most of the heuristics for node insertion and connection proposed in the PRM literature can be adopted to improve the performance of the planner. For instance, Visib-PRM [23] and PDR [13] can be considered in order to generate compact roadmaps.

C. Connecting PRM layers

Moving between PRM layers implies changing the IK solution class of (at least) one of the arms. Accordingly to the parametrization introduced in Section II, such a reconfiguration requires to traverse a singular configuration of the manipulator. Our approach does not consider simultaneous reconfigurations of both arms. Note however that this is

not a limitation for the (probabilistic) completeness of the method[1].

Next we explain how to generate nodes in a sub-manifold involving singular configurations of one arm. Let us take for example $\mathcal{S}_{i,i'}$, which connects layers $G_{i,j}$ and $G_{i',j}$. All the configurations in this set have a fixed value of one or several joint variables of Arm1. We have assumed this (these) value(s) to be known. Singular configurations of Arm1 can therefore be generated by keeping such joint value(s) constant and randomly sampling the other ones. The location of the movable object $\mathbf{q}_{\text{Obj}}$ grasped by Arm1 can be then obtained by direct kinematics. Once $\mathbf{q}_{\text{Obj}}$ has been computed, $\mathbf{q}^{\text{a}}_{\text{Arm2}}$ (in the case of a redundant manipulator) is sampled using the RLG algorithm [8], and $\mathbf{q}^{\text{p}}_{\text{Arm2}}$ is obtained by solving the corresponding IK problem. Collision-free configurations sampled in $\mathcal{S}_{i,i'}$ are then connected to neighbor nodes in layers $G_{i,j}$ and $G_{i',j}$. Note that the aforementioned PRM heuristics for node insertion and connection can also be applied to the singular nodes.

A graph is constructed to represent, at a higher-level, the connectivity of the multi-layer roadmap. Nodes of this connectivity graph correspond to connected components on each layer, and edges are singular configurations connecting these components (see Figure 4 for an illustration).

D. Solving path planning queries

The initial and final points for a path planning query can be fully specified by configurations $\mathbf{q}_{\text{init}}$ and $\mathbf{q}_{\text{goal}}$, or they can be partially given by only specifying the active variables $\mathbf{q}^{\text{a}}_{\text{init}}$ and $\mathbf{q}^{\text{a}}_{\text{goal}}$. In the former case, $\mathbf{q}_{\text{init}}$ and $\mathbf{q}_{\text{goal}}$ are added to the corresponding PRM layers. In the latter case, the IK problems are solved to obtain the multiple possible configurations $\mathbf{q}^{\text{p}}_{\text{init}}$ and $\mathbf{q}^{\text{p}}_{\text{goal}}$, and the resulting nodes are inserted on all the layers where they are collision-free. Local paths are then computed to connect (the multiple) $\mathbf{q}_{\text{init}}$ and $\mathbf{q}_{\text{goal}}$ to neighbor nodes on the corresponding layers.

If $\mathbf{q}_{\text{init}}$ and $\mathbf{q}_{\text{goal}}$ can be connected to nodes in the same connected component on one PRM layer, the path planing query can be solved without crossing singularities. Otherwise, the high-level connectivity graph is explored (using an A* algorithm) to find the shortest path between any pair of connected components containing $\mathbf{q}_{\text{init}}$ and $\mathbf{q}_{\text{goal}}$. Thus, the resulting path involves the minimum number of arm reconfigurations required to solve the query.

E. Implementation details

Let us briefly mention some details about the implementation of the planner applied for tests described in next sections. The implemented planner applies the Visib-PRM approach [23], which yields compact roadmaps (i.e. with a small number of nodes). The planner interleaves PRM layer construction and connection stages. At each iteration, the stage type is randomly selected. A tuning parameter is used to favor the probability of selecting one of the stages. Tests have been made using a value of this parameter that makes the construction stage be selected with a probability of 80%. For each test, the roadmap construction is started from scratch, and the two-stage construction process is iterated until the solution to the path planning query is found.

[1]Probabilistic completeness proofs are not provided in this paper.

V. RESULTS ON AN ACADEMIC EXAMPLE

This section presents an empirical performance analysis of the proposed method on an academic example. The mobile system, represented in Figure I, involves two 3R planar manipulators simultaneously holding an object. In this case, the manipulators are non-redundant. Therefore, the active variables are those defining the object location: $\mathbf{q}^{\text{a}} = \mathbf{q}_{\text{Obj}}$. The maximum number of solutions to the IK problems IK_{Arm1} and IK_{Arm2} is 2. Thus, the number of PRM layers will be 4. For both arms, the 2 IK solution classes meet at a singularity set characterized by a joint value of the second rotation (the elbow) equal to zero.

The main goal of the tests was to compare the performance of the proposed algorithm when the path planning problem requires the manipulator(s) reconfiguration and when it does not. For this, the planner was applied to the two problems illustrated in Figure I. For problem A, which does not require the reconfiguration of any manipulator, the solution path can be found by constructing a roadmap on a single layer (i.e. only using one IK solution class of each arm). Numerical results[2] in the first row of Table I correspond to this test. However, since the need of reconfiguring manipulators for solving a path planning query is not known a priori, the algorithm generally needs to explore all the layers. The second row in the table shows results when the planner was applied to the same example and simultaneously constructed the 4 PRM layers. As we expected, the number of nodes and the computing time are approximately multiplied by 4. The solution of problem B requires the reconfiguration of both manipulators (see Figure 5). Thus, a solution path cannot be obtained by running a PRM algorithm on a single layer. The fifth row in the table corresponds to the performance of the planner for solving the problem by constructing the 4 PRM layers in parallel. The number of nodes is very similar to that in the previous case. Note however that, although the problem seems to be more difficult, the computing time is lower. The reason is that the presence of the obstacle in the bottom left part of the workspace reduces the volume of the

[2]All numerical results in the paper have been averaged over 20 runs of the planner. Computing time corresponds to an Intel Core2-Duo processor at 2.13 GHz.

TABLE I
NUMERICAL RESULTS : ACADEMIC PROBLEMS

Problem	Strategy	n_{nodes}	T
A	single layer	11	0.93 s
A	parallel	42	4.28 s
A	one-by-one	34	2.49 s
B	single layer	–	–
B	parallel	39	1.96 s
B	one-by-one	35	2.17 s

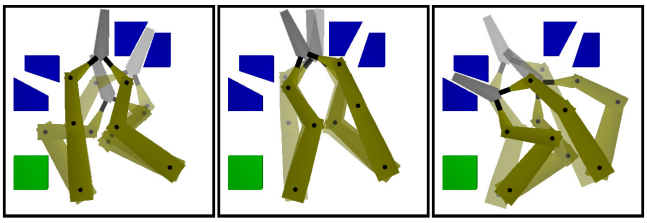


Fig. 5. Solution to problem B. The images show three parts of the solution path. The first part (left) corresponds to the sub-path between the initial configuration and the first singularity, which permits the reconfiguration of the left arm. The second one (middle) shows the sub-path to the second singularity (reconfiguration of right arm). The third sub-path (right) reaches the final configuration.

$\mathcal{C}_{feas}$, which, in this case, reduces the cost of computing local paths. These results show that, in the worst case, the computational cost increases linearly with the number of PRM layers.

Tests were also ran to analyze the performance of two strategies of the method: one simultaneously generating nodes on all the layers for each sampled value of $\mathbf{q}^a$, and the other generating only one node on one of the layers by randomly choosing the IK solution classes $\{i, j\}$. The aforementioned results correspond to the first strategy. The third and sixth rows of Table I show results for the second strategy. For problem A, numerical results show a better performance of the one-by-one PRM layer construction strategy. However, the number of nodes and the computing time obtained with both strategies are very similar for problem B. The reason of this different behavior is that, in the case of problem A, all the layers do not need to be explored for finding the solution path, whereas solving problem B requires to explore a wide portion of the configuration space, on the 4 layers, to find a path involving two reconfigurations. Finally, note that results obtained with the one-by-one PRM layer construction strategy for solving both problems are very similar. This shows that the need of reconfiguring robots for solving a path planning query does not affect the computational efficiency of the method.

VI. APPLICATION TO THE DLR JUSTIN ROBOT

Justin [18] (see Figure VI) is a humanoid robot composed of two 7 DoF DLR-Lightweight-Robot-III arms [12] mounted on a 3 DoF torso. The robot also include four-fingered DLRHand-II hands [4]. In the example of dual-arm coordinated manipulation presented below, the joints of the torso and the hands have been fixed. Thus, the active variables for path planning are those defining the object location and the value of the redundant joint of each arm. Accordingly to the inverse geometric model provided by the DLR, the redundant joint corresponds to the third rotation. For a given value of this redundant joint, the maximum number of IK solutions involving the other 6 rotations is 8. Therefore, the number of PRM layers will be 64. There are 4 singularity sets permitting to switch between IK solution classes of each arm. They are characterized by particular values (0 or 90 degrees) of one or two joints.

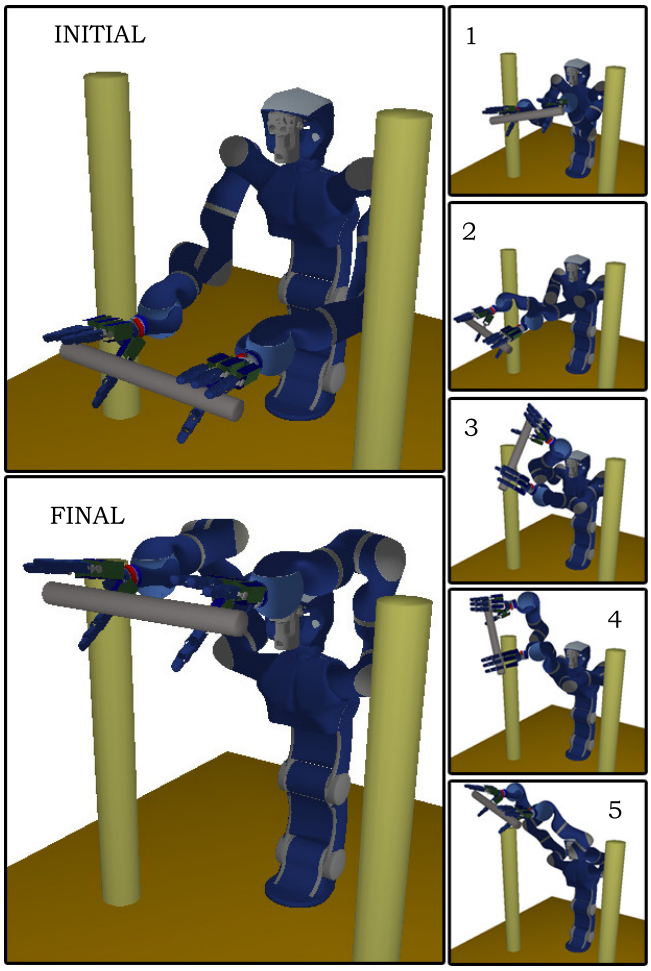


Fig. 6. Dual-arm manipulation planning problem for Justin requiring two reconfigurations. Singular configurations are represented in frames 2 and 4 of the solution path sequence.

In the path planning problem illustrated by Figure VI, Justin has to move up a bar held by its two hands. The two vertical obstacles hinder the redundant joint of the manipulators play its role. Hence, the two arms need to go through singularities to reach the goal configuration. The small images on the right part of the figure show frames of the solution path. Frames 2 and 4 correspond to the two singularities enabling the reconfigurations.

Table II shows numerical results on the performance of the planner for solving the query illustrated by Figure VI without prior roadmap construction. These results show a similar performance between the parallel and one-by-one PRM layer construction strategies. However, the graph obtained by the one-by-one strategy is lighter (in terms of number of nodes) than the graph obtained from the parallel strategy.

TABLE II
NUMERICAL RESULTS : JUSTIN

Problem	Strategy	n_{nodes}	T
Justin	parallel	1220	155 s
Justin	one-by-one	844	139 s

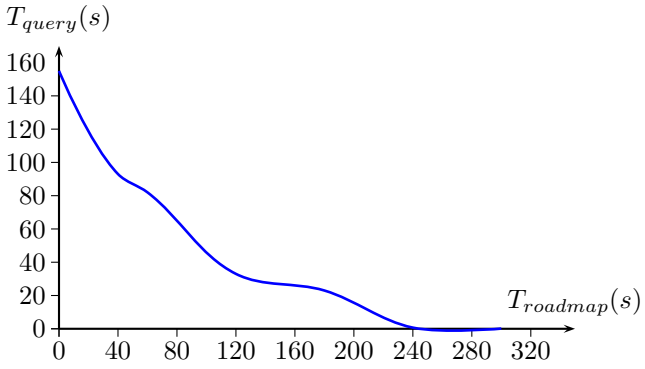


Fig. 7. Time for solving queries vs. time spent to construct the roadmap.

Nevertheless, when the goal is to construct a rich roadmap for multiple queries, the parallel construction strategy is more appropriate.

Figure VI shows the relationship between the roadmap construction time and the time needed to solve queries. The query time rapidly decreases when the size of the precomputed roadmap increases. Queries are solved instantaneously after 4 minutes of roadmaps construction.

VII. CONCLUSION AND FUTURE WORK

We have presented a method for planning dual-arm coordinated manipulation paths based on an extension of the PRM algorithm to closed-chain mechanisms. The approach is general, and can be directly applied to complex multi-arm systems. Results obtained with a model the DLR's Justin robot show the computational efficiency of the method.

The principle could be applied to other types of path planning algorithms. We are currently investigating the extension of RRT-like algorithms for rapidly solving single path planning queries. Other future work concerns the integration of the proposed method within a global approach to manipulation task planning [15], [22], including transit motions for regrasping operations.

VIII. ACKNOWLEDGMENTS

The authors thank Christoph Borst and Alin Albu-Schaeffer for providing Justin's models and related information. This work has been supported by the European Community under Contract IST 045359 "PHRIENDS", and by the Région Midi-Pyrénées under project "AMOBIO".

REFERENCES

[1] G. Ballantyne and F. Moll, "The da Vinci telerobotic surgical system: the virtual operative field and telepresence surgery," *Surgical Clinics of North America*, vol. 83, no. 6, pp. 1293–1304, Dec. 2003.

[2] S. Basu, R. Pollack, and M.-F. Roy, "Computing roadmaps of semi-algebraic sets on a variety," *Journal of the American Mathematical Society*, vol. 13(1), pp. 55–82, 2000.

[3] J. Burdick, "On the inverse kinematics of redundant manipulators: Characterization of the self-motion manifold," in *IEEE Int. Conf. Robot. & Autom.*, 1989, pp. 264–270.

[4] J. Butterfaß, M. Grebenstein, H. Liu, and G. Hirzinger, "DLR-Hand II next generation of a dextrous robot hand." in *IEEE Int. Conf. Robot. & Autom.*, 2001, pp. 109–120.

[5] J. Canny, *The Complexity of Robot Motion Planning*. MIT Press, 1988.

[6] K.-S. Chang, R. Holmberg, and O. Khatib, "The augmented object model: Cooperative manipulation and parallel mechanism dynamics," in *IEEE Int. Conf. Robot. & Autom.*, 2000, pp. 470–475.

[7] J. Cortés and T. Siméon, "Probabilistic motion planning for parallel mechanisms," in *IEEE Int. Conf. Robot. & Autom.*, 2003, pp. 4354–4359.

[8] ——, "Sampling-based motion planning under kinematic loop-closure constraints," in *Algorithmic Foundations of Roadmaps VI (WAFR2004)*, M. Erdmann and al., Eds. Springer-Verlag, Berlin, 2005, pp. 75–90.

[9] J. Desai and V. Kumar, "Motion planning for cooperating mobile manipulators," *Journal of Robotic Systems*, vol. 16(10), pp. 557–579, 1999.

[10] L. Han and N. M. Amato, "A kinematics-based probabilistic roadmap method for closed kinematic chains," in *Algorithmic and Computational Robotics: New Directions (WAFR2000)*, B. Donald, K. Lynch, and D. Rus, Eds. A.K. Peters, Boston, 2001, pp. 233–245.

[11] L. Han, L. Rudolph, J. Blumenthal, and I. Valodzin, "Stratified deformation space and path planning for a planar closed chain with revolute joints," in *Algorithmic Foundations of Robotics VII (WAFR2006)*, 2008, in press.

[12] G. Hirzinger, N. Sporer, A. Albu-Schäffer, M. Hähnle, R. Krenn, A. Pascucci, and M. Schedl, "DLR's torque-controlled Light Weight robot III - are we reaching the technological limits now?" in *IEEE Int. Conf. Robot. & Autom.* IEEE, 2002, pp. 1710–1716.

[13] L. Jaillet and T.Siméon, "Path deformation roadmaps," in *Algorithmic Foundations of Robotics VII (WAFR2006)*, 2008, in press.

[14] O. Khatib, "Object manipulation in a multi-effector robot system," in *Robotics Research 4*, R. Bolles and B. Roth, Eds. MIT Press, 1988, pp. 137–144.

[15] Y. Koga and J.-C. Latombe, "On multi-arm manipulation planning," in *IEEE Int. Conf. Robot. & Autom.*, 1994, pp. 945–952.

[16] A. Ladd and L. Kavraki, "Measure theoretic analysis of probabilistic path planning," in *IEEE Int. Conf. Robot. & Autom.*, vol. 20, no. 2, April 2004, pp. 229–242.

[17] T. Lozano-Pérez, "Spatial planning: A configuration space approach," *IEEE Transactions on Computers*, vol. 32(2), pp. 108–120, 1983.

[18] C. Ott, O. Eiberger, W. Friedl, B. Bauml, U. Hillenbrand, C. Borst, A. Albu-Schaffer, B. Brunner, H. Hirschmuller, S. Kielhofer, R. Konietschke, M. Suppa, T. Wimbock, F. Zacharias, and G. Hirzinger, "A humanoid two-arm system for dexterous manipulation," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 4-6 Dec. 2006, pp. 276–283.

[19] J. M. Porta, J. Cortés, L. Ros, and F. Thomas, "A space decomposition method for path planning of loop linkages," in *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems*, 2007, pp. 1882–1888.

[20] S. Schneider and R. Cannon, "Object impedance control for cooperative manipulation: Theory and experimental results," in *IEEE Int. Conf. Robot. & Autom.*, vol. 8(3), 1992, pp. 383–394.

[21] J. Schwartz and M. Sharir, "On the piano movers' problem II: General techniques for computing topological properties of real algebraic manifolds," *Advances in Applied Mathematics*, vol. 4, pp. 298–351, 1983.

[22] T. Siméon, J.-P. Laumond, J. Cortés, and A. Sahbani, "Manipulation planning with probabilistic roadmaps," *International Journal of Robotics Research*, vol. 23(7), pp. 729–746, 2003.

[23] T. Siméon, J.-P. Laumond., and C. Nissoux, "Visibility based probabilistic roadmaps for motion planning," *Advanced Robotics Journal*, vol. 14, no. 6, 2000.

[24] X. Tang, S. L. Thomas, and N. M. Amato, "Planning with reachable distances: Fast enforcement of closure constraints," in *IEEE Int. Conf. Robot. & Autom.*, 2007, pp. 2694–2699.

[25] F. Thomas, "The self-motion manifold of the n-bar mechanism," *Angeles J. et al. (Eds.) Computational Kinematics*, pp. 95–107, 1993.

[26] J. Trinkle and R. Milgram, "Complete path planning for closed kinematic chains with spherical joints," *International Journal of Robotics Research*, vol. 21(9), pp. 773–789, 2002.

[27] J. Yakey, S. LaValle, and L. Kavraki, "Randomized path planning for linkages with closed kinematic chains," *IEEE Transactions on Robotics and Automation*, vol. 17(6), pp. 951–958, 2001.

[28] Y. Yamada, S. Nagamatsu, and Y. Sato, "Development of multi-arm robots for automobile assembly," in *IEEE Int. Conf. Robot. & Autom.*, vol. 3, May 1995, pp. 2224–2229.