

A Global Approach for Dexterous Manipulation Planning Using Paths in n-fingers Grasp Subspace

Jean-Philippe Saut^{*†}, Anis Sahbani^{*‡}, Véronique Perdereau^{*§}

^{*}Université Pierre et Marie Curie-Paris6, EA 2385,
LISIF, 3 rue Galilée, Ivry-sur-seine, 94200 France

[†] jean-philippe.saut@lisif.jussieu.fr

[‡] sahbani@ccr.jussieu.fr

[§] vperd@ccr.jussieu.fr

Abstract—This paper addresses the motion planning problem of the dexterous manipulation of 3D rigid objects by a robotic multi-fingered hand. We propose a novel approach based on probabilistic roadmap techniques. Inspired by the theory developed in [1], [17], the planner relies on a topological property that characterizes the existence of solutions in GS_n , a specific manifold of the configuration space. This property leads to reduce the problem by structuring the search-space. It allows us to design a manipulation planner that directly captures in a probabilistic roadmap the connectivity of sub-dimensional manifolds of the composite configuration space. The proposed method allows a global planning –both object and fingers trajectories are computed– that can cope with the obstacle presence in the environment. Collisions between different fingers or between object and fingers elsewhere than fingertips are avoided. Force closure constraints are taken into account to ensure the computed paths physical feasibility, under quasi-static motion assumption. First experiments demonstrate the feasibility and the efficiency of the approach.

I. INTRODUCTION

Robotic manipulation study has long been of important interest as it is a fundamental requirement for robots to be able to pick and move objects from their environment. A particular field in robotic manipulation concerns dexterous manipulation that is done with a multifingered hand, unlike ordinary manipulation that is realized with whole arm or wrist movement. Dexterous manipulation is effected with small finger movements used for both object displacement and grasp reconfiguration, allowing the grasped object manipulation to be done in a limited space. Multifingered manipulation involves many research domains such as mechanics (for hand design or contact study), control (finger force and/or position control) and, at a higher level, motion planning. This paper only deals with the latter topic. Dexterous manipulation planning's goal is to find trajectories for both grasped object and fingers that will lead the object from a start configuration to a goal configuration while avoiding unfeasible configurations as well as grasp instability. We present in next section existing work about dexterous manipulation planning.

II. RELATED WORK

A key point in dexterous manipulation is to determine collision-free finger trajectories as well as regrasping sequences. This has been the main topic of many research works in the past decades. A first problem formulation was proposed by Li and Canny in

the late 80's [11] but they did not give any resolution scheme (for another formulation see [7]).

Trinkle and Hunter [19] introduced a qualitative description of grasps, listing the contacts between elements of the grasped object and the hand like vertices or edges, called *contact formation* (CF). They build a set of CF and try to link them to grow a tree. Linking is done with a planning method working in joint space. The solution of the dexterous manipulation problem is found when start and goal configurations (CF) are linked to the tree. This work is restricted to a manipulation system with low degree of freedom. Montana [13] proposed a full configuration space description of the multi-fingered manipulation kinematics and presented the finger gaiting example for twirling a baton.

Zhang *et al.* [22], motivated by the observation that human beings do all their manipulation activities with a finite set of grasp, identify a set of *canonical grasps* (CG) and determined which of them can be connected. Then, a graph whose nodes are these CG is computed. Low level control laws are designed to realize the transition between the CG. Zhang *et al.* also experimented their method on a three-fingered 6 DOF hand.

Han and Trinkle [8] also get interested in finger gaiting and proposed a framework for the manipulation planning of a sphere with three fingers. A finger needs to be replaced if it is close to its workspace boundary or if it can not ensure a force closure grasp with any of the two others.

The same year, Rus [15] proposed a full dynamics algorithm called the *finger tracking* algorithm. The main idea of this algorithm is to use two fixed fingers that do not move (with respect to world frame) and a third one that moves to control the reorientation movement.

Cherif and Gupta [3], [4] used the same principle to plan the re-orientation of a convex object. Three fingertips are fixed and the motion of a fourth one is used to rotate the object.

More recently, Sudsang and Phoka [18] proposed a planning method for regrasping of a polygon with a 4-fingered hand. It is based on the construction of a *switching graph*. Each node of this graph is a set of particular grasps called *concurrent grasps*. This elegant technique allows to build a grasping configuration sequence to go from one grasp to another while preserving the force-closure property. However, it is only a regrasping planning method and does not regard object

motion planning.

Yashima and *al.* [21], [20] propose a randomized planning architecture based on contact mode switching. It considers all possible contact modes (sliding, sliding with roll, with spin, *etc.*). Based on RRT method [10], a global planner builds a random tree to explore the object configuration space (position and orientation) and a local planner tries to link the tree nodes. This local planner builds an object trajectory and randomly chooses a contact mode. Then, inverse problem is used to compute the joint torque trajectories that would lead to the desired object trajectory while satisfying the manipulation constraints. This approach can cope with all constraints, dynamic and stability constraints as well as collision constraints. The counterpart of such an approach is the heavy computing time. However, all these works compute a trajectory (path) for the object and resolve later the inverse kinematic model of fingers in order to validate the path so they do not consider the whole system *object+hand*. Therefore, they do not have any guaranty for the convergence of their planning algorithm.

The proposed method, in this paper, deals with quasi-static manipulation and guarantees to find, in a finite time, a solution if it exists (it uses a randomized planner and this planner type is probabilistically complete). In our planner, the whole system *object+hand* is considered. The main idea is to structure the composite configuration space in reduced sub-manifolds. We extend the approach proposed in [16], [17] to the context of dexterous manipulation planning.

A brief overview of this approach is presented in the next section.

III. MANIPULATION PLANNING THEORY: A BRIEF REVIEW

Let \mathcal{R} and \mathcal{O} denote a robot and a movable object in a 3D workspace. The configuration space of the two systems is $\mathcal{CS} = \mathcal{C}_{robot} \times \mathcal{C}_{object}$ where \mathcal{C}_{robot} is the robot configuration space and \mathcal{C}_{object} the object configuration space. The subset of all admissible configurations is noted \mathcal{CS}_{free} , i.e. configurations where the moving bodies do not intersect together or with the static obstacles.

Regarding the manipulation constraints, \mathcal{CS}_{free} is decomposed in two sub-spaces *PLACEMENT* (\mathcal{CP}) and *GRASP* (\mathcal{CG}). \mathcal{CP} is the sub-space of \mathcal{CS}_{free} defined as the set of free configurations where \mathcal{O} is placed at a stable configuration. Paths in \mathcal{CP} are called *transit paths*. \mathcal{CG} is the sub-space of \mathcal{CS}_{free} defined as the set of free configurations corresponding to all possible grasps of the object. Paths in \mathcal{CG} are called *transfer paths*.

$\mathcal{CG} \cap \mathcal{CP}$ is the set of all the configurations where the object is grasped by the robot and is in a stable position. A manipulation graph \mathcal{MG} is formed by the transit and transfer path connection of a number of configurations in $\mathcal{CG} \cap \mathcal{CP}$. The construction of \mathcal{MG} is done in two steps. First the connected components of $\mathcal{CG} \cap \mathcal{CP}$ are determined, then the connectivity of these sub-sets is computed using transit and transfer paths.

Let us denote \mathcal{G}_i a transformation matrix corresponding to the object grasp and \mathcal{P}_i a vector containing the object position and orientation. A configuration of the “robot grasping the object” system is completely determined by the knowledge of the set $(\mathcal{G}_i, \mathcal{P}_i)$ as long as the robot’s DOF are restricted so that its kinematic loop has no redundancy.

Two configurations $q_1 = (\mathcal{G}_1, \mathcal{P}_1)$ and $q_2 = (\mathcal{G}_2, \mathcal{P}_2)$ can be linked by a transfer path only if $\mathcal{G}_1 = \mathcal{G}_2$ as regrasping is not possible for the robotic arm while carrying the object. q_1 and q_2 can be linked by a transit path only if $\mathcal{P}_1 = \mathcal{P}_2$ as the object can not move alone. As the probability that any two randomly chosen configurations share the same \mathcal{P}_i or \mathcal{G}_i is null, these two configurations can not be linked by a single transit path or by a single transfer path. A trivial approach of this problem would have been to try to link the start and goal configurations with a sequence of transit-transfer or transfer-transit paths, that would correspond to sample the whole \mathcal{CS} . Such an approach works but leads to a very slow resolution and an excess of grasping and regrasping movements.

Instead, [1], and then [16], [17], proposed to use paths in $\mathcal{CG} \cap \mathcal{CP}$. The idea is to explore $\mathcal{CG} \cap \mathcal{CP}$ as such. $\mathcal{CG} \cap \mathcal{CP}$ is considered as the configuration space of a single system consisting of the robot together with the object placed at a stable position. Maintaining the placement while the object is grasped by the robot induces a closed chain for the global system. Paths in $\mathcal{CG} \cap \mathcal{CP}$ are of course not physically feasible but, thanks to the reduction property stated and proved in [1] (in 2D case but extension to 3D is straightforward), they always can be decomposed in a transfer and transit paths sequence.

A roadmap \mathcal{MG} can thus be computed that will link the different $\mathcal{CG} \cap \mathcal{CP}$ components with transfer or transit paths as shown in Figure 1.

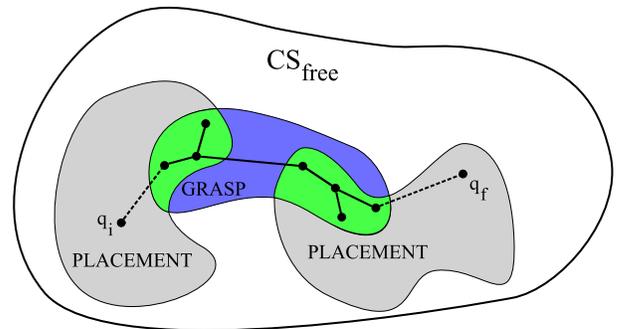


Fig. 1. Example of manipulation path linking two configurations q_i and q_f in \mathcal{CS}_{free}

Obviously, this manipulation planning theory can not be used in state for dexterous manipulation planning (manipulation by a multi-fingered hand). Indeed, during dexterous manipulation, the object is always grasped and subspace *PLACEMENT* becomes meaningless. However, we propose a reformulation of this theory for dexterous

manipulation planning. Our approach uses an analogy between regrasping and transit paths as well as a subspace analogous to $GRASP \cap PLACEMENT$. This is the paper contribution and next section topic.

IV. PROBLEM FORMULATION

Let us consider a n -fingered robotic hand \mathcal{H} manipulating a 6 DOF rigid object \mathcal{O} . The goal of the dexterous manipulation planning is to determine a finger movement and hand regrasping sequence in order to move the grasped object \mathcal{O} from an initial pose to a final one. An initial and a final grasps are associated with each of these poses. A pose is the combination of a position and an orientation i.e. a vector $\mathcal{P} = (x, y, z, \alpha, \beta, \gamma) \in \mathbb{R}^6$ where (x, y, z) is the object position and (α, β, γ) a parametrization of its orientation (e.g. Euler angles). A k -finger grasp is a vector $\mathcal{G}_k = (u_1, v_1, \dots, u_k, v_k) \in \mathbb{R}^{2k}$ where (u_i, v_i) are the coordinates of the i^{th} contact –assumed to be punctual– position on the object surface. The configuration vector of the i^{th} finger is $\mathcal{F}_i = (\theta_{i,1}, \dots, \theta_{i,m}) \in \mathbb{R}^m$ where $\theta_{i,j}$ is its j^{th} joint parameter and m its joint number. We assume all contacts to be point contact with friction obeying to Coulomb friction law. Let \vec{f} be the contact force that exerts a body S_1 on a body S_2 . There is no slipping at the contact as long as \vec{f} respects the following constraints:

$$\vec{f} \cdot \vec{n} > 0 \quad (1)$$

$$\sqrt{f_{t_1}^2 + f_{t_2}^2} \leq \mu f_n \quad (2)$$

where \vec{n} is the contact point normal, directed through S_2 , f_{t_1} and f_{t_2} the tangential components of \vec{f} , f_n its normal component and μ the contact friction coefficient.

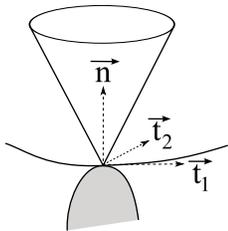


Fig. 2. Coulomb friction cone.

It is also assumed that the rolling zone of the object/fingertip contact is small enough to be regarded as a point. Thus, changes of contact positions due to rolling are neglected. We choose the usual denomination *free finger* for a finger that is not participating to the grasp and *grasping finger* for a finger that is. A free finger is constrained to have no contact with the object whereas a grasping finger is constrained to maintain a contact with the object surface at its fingertip. So a free finger has m DOF and a grasping only two (the contact point position on the object surface). Let us denote \mathcal{CS}_{free} the subspace of all the free configurations. A configuration is free if there is no contact between \mathcal{O} and \mathcal{H} somewhere else

than the fingertips, no contact between \mathcal{O} or \mathcal{H} and obstacles in the environment and no contact between \mathcal{H} bodies that are not linked by a joint (e.g. there must be no contact between phalanges of different fingers).

The configuration of the system “ \mathcal{O} grasped by \mathcal{H} with k fingers” is fully determined by the knowledge of \mathcal{P} , \mathcal{G}_k and the \mathcal{F}_i such as finger i is a free finger. This configuration vector is enough to determine the whole system state as long as we consider that the finger inverse geometric models have only one solution (if not they can be restricted so as they do). We define the subspace of all the grasps with a given set of k grasping fingers and call it GS_k (for Grasp Subspace with k fingers). This subspace dimension is $2k + (n - k)m + 6$: for this subspace configurations, there are k grasping fingers with 2 DOF each and $(n - k)$ free fingers with m DOF each and the object adds its 6 DOF.

Two local path types are fundamental for any manipulation task, corresponding to elementary manipulation subtask: regrasping and object displacement. We will call the first one *regrasping path* and the second one *transfer path*.

During a regrasping path, the object is immobile and grasped by at least three fingers –the minimum contact number that can achieve grasp stability in case of point contact with friction in the 3D case. Unlike transit paths in manipulation planning, the robot can not move freely during a regrasping path. Only free fingers can move freely whereas grasping fingers have to stay fixed. A regrasping path allows a grasp change while the object pose remains constant and does not allow a change of the object pose.

During a transfer path the object moves while the grasp remains the same. The object poses between initial and final ones are chosen along a linear path linking them. The finger configurations are computed from their inverse geometric models as we know the fingertips positions from object pose and contact positions. A transfer path allows an object pose change while the grasp remains constant and does not allow a grasp change.

Figure 3 gives an illustration of the two kinds of paths for the particular case of a 4-fingered hand in the plane.

According to the two path definitions, two configurations $q_1 = (\mathcal{P}_1, \mathcal{G}_{k_1})$ and $q_2 = (\mathcal{P}_2, \mathcal{G}_{k_2})$, both belonging to GS_n , can thus be linked by a transfer path only if $\mathcal{G}_{k_1} = \mathcal{G}_{k_2}$ and by a regrasping path only if $\mathcal{P}_1 = \mathcal{P}_2$.

Paths in GS_n are generally not physically feasible. But, they can be decomposed in a sequence of regrasping-transfer or transfer-regrasping paths thanks to the reduction property [1]. The only case where paths in GS_n are feasible is when they are transfer paths (as transfer paths are a particular kind of paths in GS_n).

Another constraint on the feasibility of a path, concerns the grasp stability. Considering sufficiently slow movements of both fingers and object allows to ignore inertial forces, we will regard the stability problem as purely quasi-static. A grasp fundamental property is force closure and has been widely treated in literature (e.g. [2], [12]). A force closure

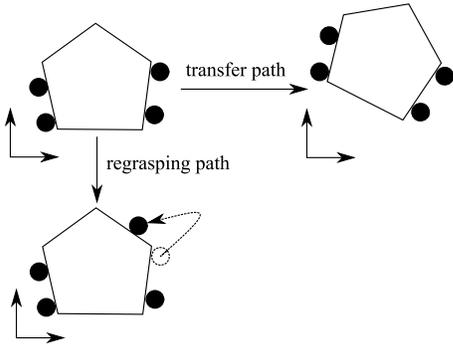


Fig. 3. Example of regrasping and transfer paths for the example of a 4-fingered hand in the plane.

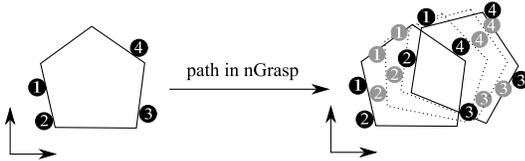


Fig. 4. A linear path in GS_n for a 4-fingered hand in the plane.

grasp is a grasp that can exert arbitrary force/torque wrench on the grasped object by applying appropriate contact forces. If the grasp is force closure all along a path, the path is physically feasible as long as the fingers can exert the forces needed to maintain the grasp and move the object. These forces can not be exerted if they are bigger than the limit forces that the fingers can apply to the object. To be sure they can be exerted, a quality index of the grasp can be added and tested along the path e.g. introducing an upper bound of the contact forces that can be exerted by the fingers in the force closure test ([2]). As a regrasping can occur during the decomposition of a path in GS_n , the force-closure property is tested for each $n - 1$ fingers grasp to give the possibility to each grasping finger to break its contact.

We can now define the dexterous manipulation planning problem as:

Problem : A dexterous manipulation planning problem is to find a manipulation path (i.e. an alternate sequence of regrasping and transfer paths) connecting two given configurations q_i and q_f in GS_n and ensuring the grasp stability.

V. PLANNING SCHEME DESCRIPTION

The main idea of our approach is to exploit the reduction property to decompose the construction of the manipulation graph. Therefore our planner is based on two modules:

- Computing the connected components of GS_n and capture its topology inside a probabilistic roadmap [9], [14].

- Determining the connectivity of these connected components using regrasping and transfer paths.

The first module builds a graph in GS_n using a probabilistic roadmap method. It aims to compute a roadmap that will capture the topology of GS_n so as to find a path between any two configurations belonging to the same connected component of GS_n . GS_n is the sub-manifold of the global configuration space with a lower dimension. The main difficulty using probabilistic roadmap approaches is to face sub-dimensional manifolds. Indeed the probability to choose a configuration at random on a given sub-dimensional manifold is null. Therefore we use the RLG algorithm (Random Loop Generator [5]). The idea here is to consider the robot and the object as a unique system forming a closed chain. Additional virtual degrees of freedom are added on each contact point to ensure the closure of the chain. This is illustrated by Figure 5, which shows how the additional virtual DOF are used to generate contact positions while the joints values are determined with inverse geometric models.

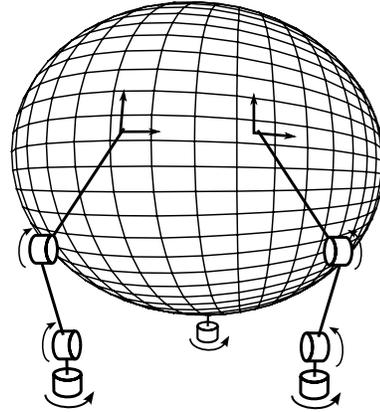


Fig. 5. The system "hand grasping object" contains closed kinematic chains. Virtual DOF (straight arrows on object surface) can be considered that allow to choose randomly configurations ensuring the chain closure.

We then propose to apply planning techniques for closed chain system to capture the topology of GS_n .

The second module tries to link the connected components of GS_n with a sequence of regrasping-transfer or transfer-regrasping paths. If such a sequence exists, the two connected components are merged.

Figure 6 shows an example of a roadmap in CS_{free} . A regrasping path can start from any configuration in GS_n and leads to an instantaneous exit from GS_n . Therefore, they connect two points on the figure plane while traveling in another subspace that has a higher dimension but that corresponds to different DOF (the DOF of the free finger). Because of obstacles and joint limits, GS_n is generally not one unique connected subset but composed of different connected sub-manifolds. These ones can possibly be linked by regrasping path.

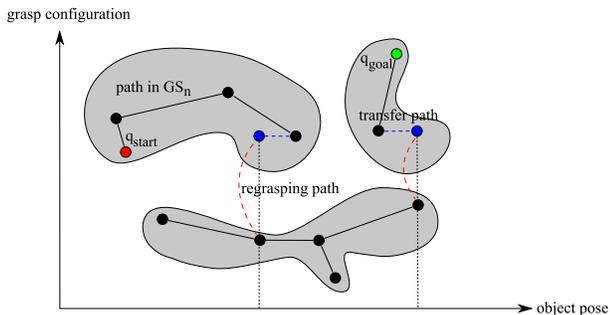


Fig. 6. A roadmap in CS_{free} . The figure plane is GS_n . Gray filled sets are collision-free configurations; red dashed lines are regrasping paths and are outside the figure plane.

The use of planning roadmap allows to build two possible versions of our planner: a single-query version and one multiple-queries one.

In the single-query version the start and goal configurations are initially added to the graph in GS_n (or linked to GS_n by a regrasping-transfer or transfer-regrasping path if they do not belong to GS_n). As soon as these configurations belong to the same connected component, the solution path between them is found across the graph.

In the multiple-queries version, a learning phase is conducted in order to build a graph, composed of components in GS_n linked by regrasping-transfer and transfer-regrasping paths that will capture the topology of CS_{free} so that any configuration in CS_{free} can be linked to this graph.

Once a solution path composed of subpaths in GS_n and of regrasping or transfer path is found, the subpaths in GS_n are transformed into regrasping/transfer or *vice-versa* path in order to obtain a physically feasible path. Eventually, a smoothing step is conducted in order to filter the unavoidable jolts due to the probabilistic method adopted.

VI. CASE OF NON PARAMETERIZABLE SURFACES

The method we presented so far seemed only applicable to objects whose surfaces are parameterizable because it requires the computation of a continuous path in GS_n . However, it is possible to slightly adapt the problem formulation to work with objects that have non-parameterizable surfaces that is the more frequent case. To do so, the object surface has to be approximated by a triangulated polyhedron. This is not really a limitation as the surface can be approximated to any precision. A grasping finger contact position can then be determined by three parameters (tr_i, u, v) where tr_i is the index of the triangular facet on which lies the contact and u and v are parametric coordinates of the contact on the triangular facet. This representation uses a discretization of the object surface that can not directly be used to compute a continuous path in GS_n . Instead, a continuous path parametrization between two configurations in GS_n can be computed by finding the contact trajectories on the object surface between those two grasps.

These trajectories can be found using a geodesic computation algorithm and they will be line segments sequences since the object surface is polyhedral. From two configurations in GS_n , it is consequently easy to compute intermediate configurations between them that belong to GS_n too.

VII. EXPERIMENTAL RESULTS

We developed a planner to verify the effectiveness of our approach. It was written in C++ and uses the RAPID collision detection library [6]. The simulated hand has four fingers. Each finger has 3 degrees of freedom. Figure 7 shows an example of a dexterous manipulation planning problem.

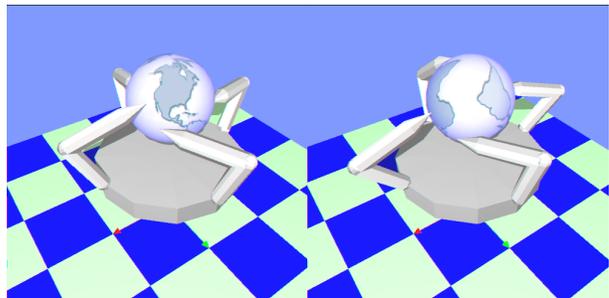


Fig. 7. Initial and final configurations of a dexterous manipulation planning problem.

The simulated example is the reorientation of a free-flying sphere (6 degrees of freedom). For this object GS_n has only one connected component. The start and goal configurations are linked by a sequence of paths in GS_n . Figure 8 shows some configurations of the manipulation path solution. The intermediate regrasping configurations are automatically determined by the planner.

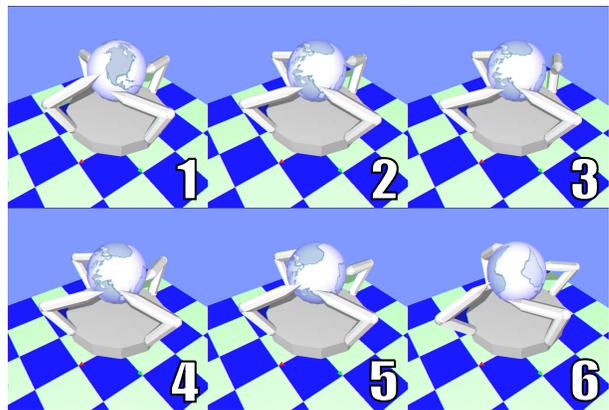


Fig. 8. Some configurations of the regrasping and object displacement sequence generated by our planner.

This example was tested on a 1.8GHz Intel Pentium IV PC with 480MB of memory. Table I shows for this example numerical results of the algorithm performance. Most of the computation time is spent for checking connections

with transfer-regrasping paths; this shows the interest of our approach which limits the number of such connection tests by first computing connected components inside GS_n . This dexterous manipulation problem was solved in 31 seconds (average time).

average running time (s)	31
average generated node number	105
average graph node number	21

TABLE I
PLANNER PERFORMANCE

VIII. CONCLUSION

We presented a new global planning method to deal with the dexterous manipulation problem. It is global since it considers the whole system (*hand + object*) and so computes free trajectories for both fingers and object while respecting the constraints imposed by manipulation with point contacts with friction. It relies on a structuring of the search space allowing to directly capture into a probabilistic roadmap the connectivity of sub-manifolds that correspond to the places where regrasping and transfer paths are connected. It applies to every object shape. Preliminary results obtained with a first implementation of the approach demonstrates its efficacy to solve dexterous manipulation planning problems. There remain many possible improvements, in particular the way to optimize computed manipulation paths or to take into account contact rolling.

REFERENCES

- [1] R. Alami, J.-P. Laumond, and T. Siméon. Two manipulation planning algorithms. *Algorithmic Foundations of Robotics WAFR94*, 1994.
- [2] A. Bicchi. On the closure properties of robotic grasping. *The International Journal of Robotics Research*, 14(4):319–33, August 1995.
- [3] M. Cherif and K. Gupta. Planning quasi-static motions for reconfiguring objects with a multi-fingered robotic hand. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 146–151, October 1998.
- [4] M. Cherif and K. Gupta. Global planning for dextrous re-orientation of rigid objects: Finger tracking with rolling and sliding. *Research Report RR-3770 of the Institut National de Recherche en Informatique et en Automatique (INRIA)*, September 1999.
- [5] J. Cortés, T. Siméon, and J.-P. Laumond. A random loop generator for planning the motions of closed kinematic chains using PRM methods. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2141–2146, May 2002.
- [6] S. Gottschalk, M.C. Lin, and D. Manocha. OBBTree: A hierarchical structure for rapid interference detection. *Proceedings of ACM Siggraph'96*, 1996.
- [7] L. Han, Z. Li, J.C. Trinkle, Z. Qin, and S. Jiang. The planning and control of robot dextrous manipulation. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 263–269, 2000.
- [8] L. Han and J.C. Trinkle. Dextrous manipulation by rolling and finger gaiting. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 730–735, May 1998.
- [9] L. Kavraki and J.-C. Latombe. Randomized preprocessing of configuration space for fast path planning. *International Conference on Robotics and Automation*, pages 2138–2139, 1994.
- [10] S.M. LaValle and J.J. Kuffner. Randomized kinodynamic planning. *International Journal of Robotics Research*, 20(5):378–400, May 2001.
- [11] Z. Li, J. Canny, and S. Sastry. On motion planning for dexterous manipulation, part i: The problem formulation. *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)*, pages 775–780, 1989.
- [12] Y.-H. Liu. Qualitative test and force optimization of 3d frictional form-closure grasps using linear programming. *IEEE Transactions on Robotics and Automation*, 15(1):163–173, 1999.
- [13] D.J. Montana. The kinematics of multi-fingered manipulation. *IEEE Transactions on Robotics and Automation*, 11(4):491–503, August 1995.
- [14] M. Overmars and P. Svestka. A probabilistic learning approach to motion planning. In *Algorithmic Foundations of Robotics (WAFR94)*, 1994.
- [15] D. Rus. In-hand dexterous manipulation of 3D piecewise-smooth objects. *International Journal of Robotics Research*, 1997.
- [16] A. Sahbani, T. Siméon, and J. Cortés. A probabilistic algorithm for manipulation planning under continuous grasps and placements. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2002.
- [17] T. Siméon, J. Cortés, A. Sahbani, and J.-P. Laumond. A general manipulation task planner. *Algorithmic Foundations of Robotics*, V:311–328, 2003.
- [18] A. Sudsang and T. Phoka. Regrasp planning for a 4-fingered hand manipulating a polygon. *Proceedings of the IEEE International Conference on Robotics and Automation*, 2:2671–2676, September 2003.
- [19] J.C. Trinkle and J. Hunter. A framework for planning dexterous manipulation. *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)*, pages 775–780, April 1991.
- [20] M. Yashima, Y. Shiina, and H. Yamaguchi. Randomized manipulation planning for a multi-fingered hand by switching contact modes. *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, September 2003.
- [21] M. Yashima and H. Yamaguchi. Dynamic motion planning whole arm grasp systems based on switching contact modes. *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, May 2002.
- [22] H. Zhang, K. Tanie, and H. Maekawa. Dextrous manipulation planning by grasp transformation. *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)*, April 1996.