

# Planning In-hand Object Manipulation with Multifingered Hands Considering Task Constraints

Katharina Hertkorn, Maximo A. Roa, and Christoph Borst

**Abstract**—In-hand manipulation with a multifinger hand is defined as changing the object pose from an initial to a final grasp configuration, while maintaining the fingertip contacts on the object surface. Given only the task constraints, represented as a desired motion of the object and an external force to be applied or resisted, the problem can be expressed as finding a good set of contact points on the object and a corresponding hand configuration compatible with the task to be executed. This paper presents a method for solving such problem, taking into account the kinematic structure and torque limits of the hand, the force closure condition (which must be guaranteed during the whole trajectory), and task compatibility. The feasibility of such method is tested in simulation of 2D and 3D examples.

## I. INTRODUCTION

Simple two-finger grippers are very common in industrial applications, mainly due to their great ability to grasp objects firmly with high robustness, and the low cost for implementation and maintenance. Multifinger hands, although more complex and expensive, allow the execution of robust grasps as well as dexterous manipulation tasks, mimicking in most cases the human hand capabilities [1].

The concept of dexterous manipulation has received several definitions, but it is generally accepted as the ability of changing the relative pose (position and orientation) of an object with respect to the hand, while keeping a stable grasp on the object (at least in the initial and final poses) [2]. A nice discussion on the definition of dexterity and the pros and cons of arm vs hand manipulation is presented in [3]. To summarize, a dexterous end effector is useful to increase the workspace of a manipulator. It also increases precision and speed and reduces the energy required to accomplish a task, when compared to an object manipulation using the whole arm with a simple end effector at the tip.

Several types of within-hand manipulations are recognized [3], [4]:

- Regrasping: the object is released and regrasped to change its pose with respect to the hand [5].
- Rolling: the object is manipulated while the fingertips roll over the object surface [6].
- Sliding: the slippage of the object inside the hand workspace is controlled to change the object pose [7].
- In-hand manipulation: the kinematic redundancy of the fingers is used to change the object from an initial to a final configuration, while maintaining fingertip contacts (local rolling is allowed) [8].

- Finger pivoting: the object is rotated while it is held by two contact points (commonly the thumb and index fingertips) that create the axis of rotation [9].
- Finger gaiting (relocation): one finger is lifted from the object surface and relocated to a more convenient position while the remaining fingers keep a stable grasp on the object [10].

This paper will focus on the problem of in-hand manipulation. Traditionally, this problem has been stated as finding the quasi-static motions for the fingertips required to move the object from a given initial grasp to a final predefined grasp. This can be solved for simple cases by a reachability analysis, which requires two steps: first, building the workspace of possible motions for a hand given the object and the grasp on the object, and second, verifying if the range of motion is achievable without changing the initial grasp [11].

The analysis of the manipulation workspace is an interesting topic on its own. Although a multifinger hand can be seen as a collection of small manipulators (fingers) with a common base (palm), the manipulation workspace for an object, i.e. the set of possible object locations that allow some degree of manipulation, is not simply the union of the workspaces for each one of the fingers, as this set does neither imply the existence of feasible contact locations on the object at any moment during the manipulation, nor are self-collisions or hand-object collisions considered. Note also that the definition of manipulability depends on a particular grasp on the object to be manipulated. The manipulability analysis can be performed analytically for simple cases with punctual contacts [12]. The study of the workspace for 3D manipulation of polyhedral objects has also been performed using optimization techniques considering the object geometry, hand joint limits, force constraints, and collision avoidance [13]. In this way, given an initial grasp on the object, the workspace that contains the possible rotations and translations of the object while keeping the grasp is obtained.

The problem tackled in this paper is the inverse one: given a desired motion in the space, one should obtain the grasp configuration (contact locations on the object surface and associated hand/object pose) that guarantees a successful execution of the manipulation task. Even more, the task can have additional constraints, for instance, of force resistance or force application in some directions (e.g. the hand should lift the own-weight of the object). A typical problem of this class is, for instance, removing the lid from a childproof bottle. A simple rotation of the lid will not open the bottle; it requires a simultaneous application of force in one axis and rotation around the same axis.

The rest of the paper is organized as follows. Section II formally defines the in-hand manipulation problem to be solved, and presents some previous related works. Section III describes the solution method proposed for this problem. Section IV discusses solutions obtained with the presented method. Finally, Section V concludes this work and presents some ideas for future research along this line.

## II. PROBLEM DEFINITION AND RELATED WORKS

The in-hand manipulation problem to be solved is formulated as follows. Given the kinematic and dynamic properties of a robotic hand, the geometry and weight of an object, and the task constraints associated to a manipulation task, the best grasp configuration for executing the desired manipulation must be computed. Note that the specification of the initial grasp configuration automatically renders the corresponding hand joints trajectories during the manipulation execution. The following assumptions are considered:

- The object is represented by a set  $\Phi_o$  of surface points  $\mathbf{p}_o$  with corresponding inward pointing normals  $\mathbf{n}_o$ . The set  $\Phi_o$  is large enough to describe the object surface accurately.
- The robotic hand has  $n$  non-redundant fingers, and there is one contact point per fingertip with the object.
- There is local rolling (no slippage) of the fingertips on the object surface, and the fingertip surface is finite.
- The friction between fingertips and the object is described with Coulomb's law.
- The motion is quasi-static, i.e. the dynamics of the manipulation is negligible.
- The complete trajectory specified for the task lies inside the manipulation workspace of the hand.

The manipulation task is defined by a desired trajectory  $\mathbf{t}_{\text{des}}$  of the center of mass (CM) of the object, and a force  $\mathbf{f}_{\text{ext}}$  applied at CM. This force represents for instance the own weight of the object, or a force that the object should apply for fulfilling a desired task.

A task is successfully completed if four conditions during all the motion trajectory of the object are fulfilled: the trajectories of the contact points compatible with the desired motion must lie inside the workspace of the corresponding finger, the grasp is force closure (FC) at every instant (i.e. it can resist perturbations in all the directions), the fingers can apply the required forces on the object surface, and the joint and torque limits of the fingers are not exceeded.

Let  $\mathbf{x}$  be the object pose (position and orientation) relative to the robotic hand and  $g$  be a force closure grasp consisting of  $n$  contact points on the object surface. For simplicity, we will refer in the rest of the paper to the following three problems of in-hand manipulation:

- *Problem 1:* Find the optimal grasp configuration  $g_{\text{opt}}$ , given the task constraints and a fixed initial pose  $\mathbf{x}$  of the object with respect to the hand.
- *Problem 2:* Find an optimal starting pose  $\mathbf{x}_{\text{opt}}$  of the object relative to the hand, given the task constraints and an initial force closure grasp  $g$ .

- *Problem 3:* Find the optimal grasp configuration ( $\mathbf{x}_{\text{opt}}$  and  $g_{\text{opt}}$ ), given the task constraints.

The problem of finding an optimal pose for executing a trajectory is also relevant for robot manipulators. For this application, the positions and orientations that the Tool Center Point can reach within the workspace of the manipulator are represented as a capability map [14]. The best starting position that allows the complete execution of the task trajectory is searched within this capability map using pattern matching techniques. The same basic idea of searching motions within the workspace has been previously explored for in-hand manipulation with multifinger hands [15], using manipulation primitives (translations and rotations) that can be added to create more complex manipulation trajectories. Although that approach can deal with objects of different sizes and geometries, the trajectory for the in-hand manipulation task is defined based on pure geometrical reasoning, without taking into account important limitations like grasp stability or torque limits on the finger joints.

The problem of searching for the best grasp configuration for executing a predefined task has been recently tackled [16]. In this case, a grasp criterion is defined such that it takes into account the feasible set of positions, velocities and forces that can be imparted to the object. Although the approach defines the best grasp under the task constraints, it only provides the initial grasp on the object, but the obtained hand/object pose might not be suitable for the complete execution of the task; e.g. given an initial grasp configuration, it might happen that the desired trajectory cannot be completely followed by the hand. On the other hand, the solution method proposed in this work, presented in the next section, finds the best possible grasp on the object that guarantees the complete execution of the predefined task.

## III. SOLUTION METHOD

In this section, the considerations on the hand and its workspace are first presented in Section III-A. Next, the four conditions for a successful manipulation task are discussed: reachability (Section III-B), feasibility (Section III-C), force closure (Section III-D), and torque limits (Section III-E). Each one of the conditions is quantified with a penalty score  $P$ , and the total score for a considered trajectory is presented in Section III-F. The complete solution algorithms for Problems 1 to 3 are presented in Section III-G.

Each part is illustrated with a 2D example using a robotic hand with  $n = 2$  fingers, and each finger has  $m = 2$  joints (all the hand specifications are given in Table I). The task to be solved is moving a rectangle 0.5 units in the negative  $y$  direction. The rectangle has a length of 0.5 units in  $x$ -direction and a height of 1 in  $y$ -direction.

### A. Workspace of the Hand

The kinematics of the hand determines the set of reachable points for each finger  $f_i$ . A representative point  $\mathbf{p}_{f,i}$  with corresponding normal  $\mathbf{n}_{f,i}$  is defined for each fingertip. The points reachable by  $\mathbf{p}_{f,i}$  are non-ambiguous for non-redundant fingers, i.e. there is a unique solution for the

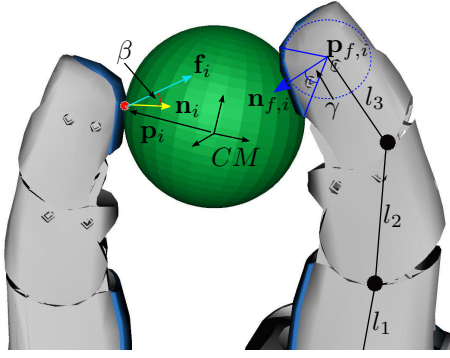


Fig. 1. Forces and normal directions at the fingertips and contact points.

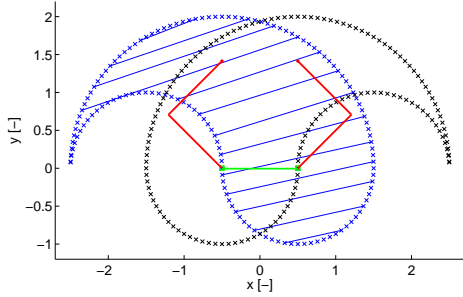


Fig. 2. The reachability map for the 2D hand is the union of the reachable sets of the left (blue shaded region) and the right finger (region framed by black crosses). A particular finger configuration is depicted in red; the green line represents the palm.

inverse kinematics of the fingers given a reachable point in the workspace. These reachable points define the set  $\Phi_{w,i}$ .

The finger can apply forces in all directions if a circular fingertip without a preferred contact surface is assumed. On the contrary, if the fingertip can only apply forces in certain directions, then the area of possible contact surface at each point in  $\Phi_{w,i}$  is described by an angle  $\gamma$  around the normal direction (Fig. 1).

The workspace properties are stored in two maps: a reachability and a feasibility map. The reachability map contains the set  $\Phi_w = \bigcup_i \Phi_{w,i}$ . This set is ordered in a k-d tree to allow a fast search for closest points in space. The boundary of all reachable points for the example in 2D is shown in Fig. 2. The blue shaded region represents the region  $\Phi_{w,1}$  (left finger), and black crosses frame the region  $\Phi_{w,2}$  (right finger).  $\Phi_w$  is the union of both sets. The hand fingers are depicted in red in a particular configuration, and the green line represents the palm of the hand.

The feasibility map contains the normals  $\mathbf{n}_{f,i}$  of the fingertip at each considered configuration, plus the unique solution to the inverse kinematics and the corresponding Jacobian. Fig. 3 shows the feasibility map for the left finger of the 2D example. A particular finger configuration is shown in red, and its corresponding normal is represented as a bold blue line. In this example, the finger can apply forces in all directions, so  $\gamma = \pi$ .

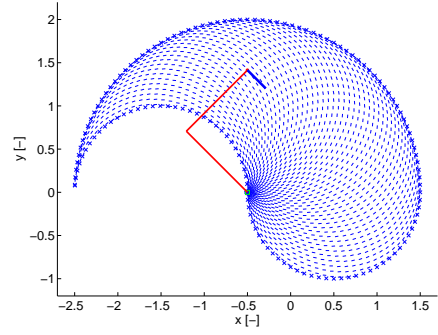


Fig. 3. Feasibility map for the left finger of the 2D hand with a particular finger configuration (red) and its corresponding normal (bold blue).

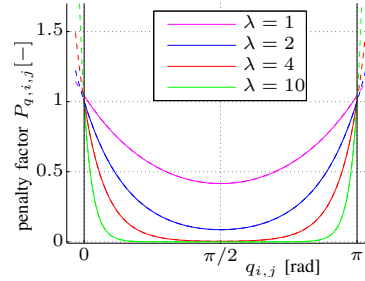


Fig. 4. Penalty function for joint limits (one joint  $j$ ).

## B. Reachability

The first condition that a valid trajectory needs to fulfill is that the fingers can reach points on the object surface. If the initial object pose is given (Problem 1), all the reachable contact points for that pose must be found. The set of reachable surface points  $\Psi_{\text{reach},i}$  for finger  $f_i$  is computed as the intersection of the reachable point set  $\Phi_{w,i}$  and the set of object points  $\Phi_o$ . If the contact points are given (Problem 2), the first thing to verify is that those points are reachable, i.e. the  $i$ -th contact point must belong to the reachable point set  $\Phi_{w,i}$ .

In a desirable situation, the position of each joint should be far from the joint limits. To consider this factor, each contact point  $\mathbf{p}_i \in \Psi_{\text{reach},i}$  that is reachable by the  $i$ -th finger receives a penalty score. The grasp penalty  $P_q$  is then the sum of all finger penalties:

$$P_q = \sum_{i=1}^n \left( \sum_{j=1}^m \left( e^{\lambda \cdot (-q_{i,j} + q_{\min,i,j})} + e^{\lambda \cdot (q_{i,j} - q_{\max,i,j})} \right) \right). \quad (1)$$

The parameter  $\lambda$  influences the interval in which the penalty for the joint limits is almost zero. It can be chosen arbitrarily depending on the relative importance of having the finger in the middle of its joint range. Fig. 4 shows the penalty function  $P_{q,i,j}$  for one joint  $j$  using several  $\lambda$ . The joint limits are  $q_{\min,i,j} = 0$  and  $q_{\max,i,j} = \pi$ .

For solving Problem 3, the set  $\mathcal{X}_{\text{reach},i}$  of reachable object poses is defined as the set of poses where the  $i$ -th finger has at least one reachable point on the object surface and the object is neither colliding with the palm nor the fingers. The set  $\mathcal{X}_{\text{reach}}$  contains all the object poses such that in each

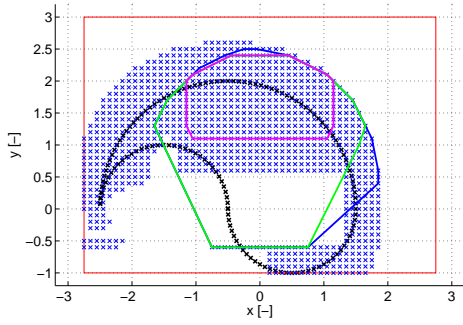


Fig. 5. Different sets of object positions:  $\mathcal{X}$  (red),  $\mathcal{X}_{\text{reach},1}$  (blue crosses),  $\mathcal{X}_{\text{reach}}$  (blue line),  $\mathcal{X}_{\text{feas}}$  (green line), and  $\mathcal{X}_{\text{final}}$  (magenta line). The boundary of the workspace  $\Phi_{w,1}$  is marked with black crosses.

pose the number of fingers that have some reachable points is at least the minimum required to get a force closure grasp (2 fingers for a 2D problem, 3 for 3D). Note that  $\mathcal{X}_{\text{reach}}$  is a subset of the union of the  $i$ -th sets:  $\mathcal{X}_{\text{reach}} \subset \cup_i \mathcal{X}_{\text{reach},i}$ .

In Fig. 5 the set  $\mathcal{X}$ , whose boundary is depicted as a red box, includes all the considered object positions (the orientation of the object is not considered in this example to simplify the representation). The set  $\mathcal{X}_{\text{reach},1}$  (finger 1) is illustrated with blue crosses. Note that  $\mathcal{X}_{\text{reach},1}$  has no points in  $x \in [-1.25, 1.25]$ ,  $y \in [-0.5, 0.5]$ , as the object collides with the palm of the hand when it is placed in this region. The crosses located within the boundary depicted in blue form the set  $\mathcal{X}_{\text{reach}}$ .

### C. Feasibility

The force that a finger can apply on the object surface at a point  $i$  depends on the friction coefficient  $\mu$ , the normal to the object surface  $\mathbf{n}_i$ , and the angle  $\gamma$  that describes the contact region on the fingertip. The feasible set of object points  $\Psi_{\text{feas},i} \subset \Psi_{\text{reach},i}$  contains the object point  $\mathbf{p}_i$  if the object normal  $\mathbf{n}_i$  is a feasible normal direction for the fingertip, i.e. if the angle between  $\mathbf{n}_i$  and  $\mathbf{n}_{f,i}$  is smaller than  $\gamma$ , and if Coulomb's friction law holds: the fingertip normal  $\mathbf{n}_{f,i}$  must lie inside the friction cone defined by  $\mathbf{n}_i^t \leq \mu \mathbf{n}_i^n$ , where  $\mathbf{n}_i^t$  and  $\mathbf{n}_i^n$  are the tangential and normal components of the object normal  $\mathbf{n}_i$ .

For each grasp, the penalty  $P_a$  is the sum of the scores of each contact point  $\mathbf{p}_i \in \Psi_{\text{feas},i}$ . The angle of the friction cone is  $\theta$ , and  $\beta_i$  is the angle between the normal of the finger  $\mathbf{n}_{f,i}$  and the object normal  $\mathbf{n}_i$  (Fig. 1).

$$P_a = \sum_{i=1}^n (e^{(\lambda \cdot (\beta_i - \gamma - \theta))}). \quad (2)$$

In Fig. 6 the penalty function for one point is depicted. The red line is the limit of the fingertip angle  $\gamma$ , the black line is the final limit for the direction of the normal:  $\gamma + \theta$ . The parameter  $\lambda$  influences again the interval in which the penalty function is almost zero. For example, for  $\lambda = 10$  only the interval  $\beta_i \in [\gamma, \gamma + \theta]$  is penalized.

For Problem 3, the set  $\mathcal{X}_{\text{feas},i}$  of feasible object poses is defined as the set of poses where the  $i$ -th finger has at least one feasible point on the object. In analogy to  $\mathcal{X}_{\text{reach}}$ , the set

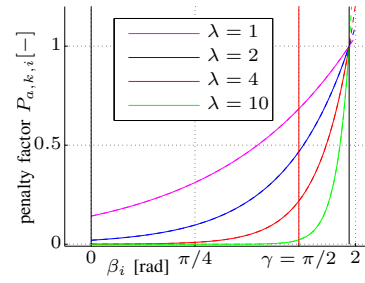


Fig. 6. Penalty function for normal direction (one contact at finger  $i$ ).

$\mathcal{X}_{\text{feas}}$  is defined as the set of object poses such that in each pose the number of fingers that have some feasible points is at least the minimum required to get a FC grasp. Note that  $\mathcal{X}_{\text{feas}} \subset \cup_i \mathcal{X}_{\text{feas},i}$ . The set  $\mathcal{X}_{\text{feas}}$  is framed in green in Fig. 5.

### D. Force Closure

Let  $\mathbf{f}_i$  be the force applied by the finger at a point  $\mathbf{p}_i$  on the object. The friction cone is linearized using a polyhedral convex cone with  $s$  sides. This leads to a grasping force

$$\mathbf{f}_i = \sum_{l=1}^s \alpha_{i,l} \mathbf{n}_{i,l}, \quad \alpha_{i,l} \leq 0, \quad (3)$$

and a corresponding wrench  $\mathbf{w}_i = [\mathbf{f}_i, \mathbf{p}_i \times \mathbf{f}_i]^T$ . The set  $\mathcal{W}$  of possible wrenches that can be applied on the object is

$$\mathcal{W} = \text{CH} \left( \bigcup_{i=1}^n (\mathbf{w}_{i,1}, \dots, \mathbf{w}_{i,s}) \right), \quad (4)$$

where  $\mathbf{w}_{i,l}$  is the primitive wrench generated by a unitary force  $\mathbf{f}_i$  along edge  $l$  of the linearized friction cone. A grasp  $g$  is FC if and only if the origin of the wrench space lies strictly inside  $\mathcal{W}$ , and its quality  $Q_g$  will be considered as the radius of the largest hypersphere fully contained in  $\mathcal{W}$  [17].

All the possible combination of contact points inside the feasible sets  $\Psi_{\text{feas},i}$  leads to candidate grasps that are later evaluated for the force closure property. All the FC grasps obtained with this evaluation are stored in the set  $\Psi_{\text{fc}}$ .

The penalty for each grasp is related to its quality  $Q_g$ :

$$P_{fc} = \max_r(Q_g) - Q_g, \quad (5)$$

where  $\max_r(Q_g)$  is the maximal grasp quality that occurs for all grasps at all object poses  $\mathbf{x}_r$ .

Note that for a specified force closure grasp (Problem 2) this step can be skipped.

### E. Force Constraint

An external wrench  $\mathbf{w}_{\text{ext}}$  acting on the center of mass of the object is related to the forces on the fingertip by the grasp matrix  $\mathbf{G}$  [17]:

$$\mathbf{w}_{\text{ext}} = \begin{bmatrix} \mathbf{f}_{\text{ext}} \\ \boldsymbol{\tau}_{\text{ext}} \end{bmatrix} = \mathbf{G} \mathbf{f}, \quad (6)$$

with  $\mathbf{f}$  containing the fingertip forces  $\mathbf{f}_i$  of all  $n$  fingers,  $\mathbf{f} = (\mathbf{f}_1 \dots \mathbf{f}_n)^T$ , and

$$\mathbf{G} = \begin{bmatrix} \mathbf{I}_3 & \dots & \mathbf{I}_3 \\ (\mathbf{p}_1 \times) & \dots & (\mathbf{p}_n \times) \end{bmatrix}, \quad (7)$$

where the symbol  $(\mathbf{a} \times)$  denotes the skew symmetric matrix of a vector  $\mathbf{a}$ . The external wrench  $\mathbf{w}_{\text{ext}}$  comes from a task constraint (for example, resisting the weight of the object), so (6) can be solved for the fingertip forces. Using the Jacobian  $\mathbf{J}_i$  of the finger  $f_i$ , the motor torques needed to counteract the external force can be computed.

The penalty score is calculated per grasp  $g$  as the maximal ratio of each joint torque to its maximal torque:

$$P_\tau = \sum_{i=1}^n \max_j \left( \frac{|\tau_{i,j}|}{\tau_{\max,i,j}} \right) \quad (8)$$

### F. Evaluation of Grasps

To evaluate a desired trajectory  $\mathbf{t}_{\text{des}}$ , it is discretized with  $n_t$  via points. The initial object poses that can lead to a successful completion of the desired task are included in the set  $\mathcal{X}_{\text{final}}$ . Each pose in this set gets a total score  $P$  per trajectory obtained by adding the penalty scores  $P_t$  for each via point; the lower the  $P$  the better the trajectory.  $P_t$  is computed as the weighted sum of the scores previously introduced:

$$P_t = w_q \cdot P_q + w_a \cdot P_a + w_{fc} \cdot P_{fc} + w_\tau \cdot P_\tau. \quad (9)$$

The weightings are chosen such that each summand is smaller or equal to one:

$$\begin{aligned} w_q &= 1/(m \cdot n) \\ w_a &= 1/n \\ w_{fc} &= 1/\max_r(Q_g) \\ w_\tau &= 1/n \end{aligned} \quad (10)$$

### G. Algorithms

First, the algorithm to solve the problem of finding an optimal grasp given a certain object pose is presented. The calculation of the optimal object pose given a certain grasp is presented later. The algorithms can be combined to solve both problems at the same time.

Given an object pose (Problem 1), the optimal grasp is determined by first finding the feasible and reachable sets for each finger. Configurations with self-collision or collisions of the fingers/palm with the object are recognized and not taken into account. The combination of feasible points for all fingers gives the set of possible grasps. They are checked for force closure, and for the valid grasps the score for each via point in the trajectory is calculated considering reachability, feasibility, force closure and maximum torques of the fingers. Choosing the trajectory with the minimum penalty score automatically chooses the optimal initial grasp required to execute such trajectory.

To calculate an optimal object pose given a FC grasp (Problem 2), the set  $\mathcal{X}$  of all possible poses of the object where the desired trajectory can start is determined and uniformly sampled. In Fig. 5 the boundary of the set  $\mathcal{X}$  is depicted in red, and the boundary of the finger workspace is marked with black crosses. The size of the object is added

#### Algorithm 1: Calculate an Optimal Grasp (Problem 1)

**Given:** object pose  $\mathbf{x}$ , task constraints  
**Output:** optimal grasp  $g_{\text{opt}}$   
**foreach** finger  $f_i$  **do**  
    | obtain  $\Psi_{\text{reach},i}$  avoiding self-collision and collision  
    | with the object;  
    | obtain  $\Psi_{\text{feas},i}$ ;  
**end**  
combine the sets and check for FC to get  $\Psi_{fc}$ ;  
**foreach** grasp  $g$  in  $\Psi_{fc}$  **do**  
    | calculate  $P_g = \sum_{t=1}^{n_t} P_t$ ;  
**end**  
 $g_{\text{opt}} \hat{=} \min_g(P_g)$ ;

to the bounding box of the workspace of the hand and then shifted in the opposite direction of the desired trajectory. For each possible object start pose the grasp is checked for its validity along the trajectory, and its score is computed. The optimal object start pose is that one with the minimal score.

#### Algorithm 2: Calculate an Optimal Object Pose (Problem 2)

**Given:** force closure grasp  $g$ , task constraints  
**Output:** optimal object pose  $\mathbf{x}_{\text{opt}}$   
Sample the bounding box of the workspace of the hand to get  $\mathcal{X}$ ;  
**foreach** pose  $\mathbf{x}_r$  in  $\mathcal{X}$  **do**  
    | calculate  $P_r = \sum_{t=1}^{n_t} P_t$ ;  
**end**  
 $\mathbf{x}_{\text{opt}} \hat{=} \min_r P_r$ ;

Both problems, finding an optimal object pose and an optimal grasp, can be solved at the same time. Finding the optimal object pose (Algorithm 2) is an outer loop, as for each pose  $\mathbf{x}_r$  in  $\mathcal{X}$  the optimal grasp is found with Algorithm 1 in an inner loop. The object pose with the smallest penalty score  $P_r$  determines the optimal grasp.

## IV. RESULTS AND DISCUSSION

The proposed approach has been implemented and used for solving three experiments: First, the problem of finding an optimal grasp (Problem 1) is discussed in 2D, and the results are compared to the approach proposed in [16]. Then, the problem of finding an optimal pose of the object given the initial contact points (Problem 2) is presented with a 3D example. Finally, the combination of finding an optimal pose and an optimal grasp is solved and compared to [15]. The properties of the robotic hands used in the examples are summarized in Table I.

### A. Implementation Details

One of the main advantages of the proposed approach is the use of maps with information on the reachability and feasibility of contact points inside the finger workspace. These maps can be computed offline, and then used in

**Algorithm 3:** Calculate an Optimal Object Pose and an Optimal Grasp (Problem 3)

**Given:** task constraints

**Output:** optimal object pose  $\mathbf{x}_{\text{opt}}$  and optimal grasp  $g_{\text{opt}}$   
 Sample the bounding box of the workspace of the hand to get  $\mathcal{X}$ ;

**foreach** pose  $\mathbf{x}_r$  in  $\mathcal{X}$  **do**

**foreach** finger  $f_i$  **do**

    obtain  $\Psi_{\text{reach},i}$  avoiding self-collision and collision with the object;  
 obtain  $\Psi_{\text{feas},i}$ ;

**end**

  combine the sets and check for FC to get  $\Psi_{\text{fc}}$ ;

**foreach** grasp  $g_r$  in  $\Psi_{\text{fc}}$  **do**

    calculate  $P_{r,g} = \sum_{t=1}^{n_t} P_t$ ;

**end**

$P_r = \min_g(P_{r,g})$ ;

**end**

$\mathbf{x}_{\text{opt}} \hat{=} \min_r P_r$ ;

TABLE I

PARAMETERS OF THE HANDS FOR THE EXAMPLES

Parameter	2D hand	DLR/HIT hand [18]
number of fingers	2	5
link length [m]	1, 1	0.055, 0.025, 0.025
fingertip radius [m]	0.001	0.011
maximum joint torque [Nm]	1	2.4
minimum joint angles [°]	0, 0	-20, 0, 0
maximum joint angles [°]	180, 180	20, 85, 65

the execution of the algorithms, thus reducing the time required to solve expensive operations such as verifying the reachability of a point and verifying the feasibility of a required force at the fingertip.

The set  $\Phi_w$  is represented as a voxelized volume structure, and the set  $\Phi_o$  of the object as a pointshell (pointcloud with normals). With this, a modified version of the Voxmap-Pointshell (VPS) algorithm [19] can be used to obtain  $\Psi_{\text{reach}}$ , which is then the intersection between a voxelmap and a pointshell. The algorithm has fast response times to collision queries ( $< 1\text{ms}$ ) even for arbitrarily complex scenarios.

The feasibility map is stored as a k-d tree, so for each object point in  $\Psi_{\text{reach}}$  the corresponding point in the workspace can be quickly found. For this point, the normals, Jacobians and joint configurations can be found in a lookup table.

### B. Problem 1: Find an Optimal Grasp Using a Fixed Initial Object Pose

The problem of finding an optimal grasp is discussed using a 2D example proposed by [16], and illustrated in Fig. 7. The 2D hand has  $n = 2$  fingers, and each finger has  $m = 2$  joints. The link lengths are 1 and the distance between the base joints of the fingers is also 1. The hand parameters are summarized in Table I. The task is to pull downwards a rectangle from a certain start pose  $\mathbf{x}$ :  $\mathbf{t}_{\text{des}} =$

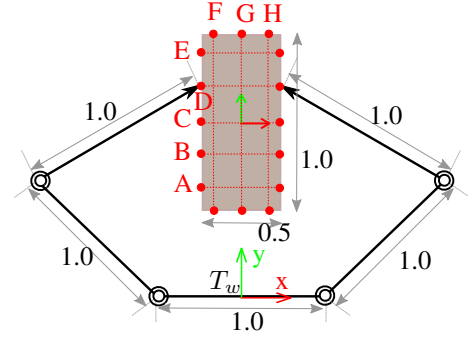


Fig. 7. Example 2D, finding an optimal grasp configuration given the object orientation.

$[\mathbf{x}, \dots, \mathbf{x} + \begin{pmatrix} 0 \\ -0.25 \end{pmatrix}]$ . During the task, the grasp needs to withstand an external force of  $\mathbf{f}_{\text{ext}} = \begin{pmatrix} 0 \\ -0.5 \end{pmatrix}$ . The rectangle has a length of 0.5 in  $x$ -direction and a height of 1 in  $y$ -direction. Its coordinate system is located in CM, in the middle of the object. In [16], three start positions of the object are discussed:  $\begin{pmatrix} 0 \\ 0.5 \end{pmatrix}$ ,  $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ ,  $\begin{pmatrix} 0 \\ 1.5 \end{pmatrix}$ . For each position, the object can have an orientation of  $(0, \frac{\pi}{4}, \frac{\pi}{2})$ . The goal is to find the optimal grasp among eight predefined grasps A-H at each start pose of the object, as shown in Fig. 7. The results are summarized in Table II.

TABLE II

COMPARISON FOR FINDING AN OPTIMAL GRASP

$r$	start pose $\mathbf{x}_r^T$	best grasp according to [16]	best grasp with the new approach
1,2,3	$(0, 0.5, \cdot)$	A	-
4	$(0, 1, 0)$	A - C	C
5	$(0, 1, \pi/4)$	A - D	B
6	$(0, 1, \pi/2)$	-	G
7	$(0, 1.5, 0)$	A - E	C
8	$(0, 1.5, \pi/4)$	-	B
9	$(0, 1.5, \pi/2)$	-	G

In [16], the initial poses  $\mathbf{x}_{1,2,3}^T = (0, 0.5, \cdot)$  (where  $\cdot$  indicates that there are three possible orientations) are taken into account. During a movement downwards, the object always collides with the palm of the hand, thus these are non feasible start positions for the task (their analysis only considers the initial grasp, not the full trajectory of the object). At the start object pose  $\mathbf{x}_4^T = (0, 1, 0)$  all grasps are feasible grasps. The best grasp candidates are A-C, as they have the same safety margins in velocity and force. Grasp A is considered to be the best grasp as its safety margin in position is the highest. With the proposed approach we found out that grasps F, G, and H do not have feasible movements as the fingers collide with the object along the trajectory. The best grasp is C ( $P_C = 2.941$ ), followed by D ( $P_D = 3.259$ ), B ( $P_B = 3.441$ ), E ( $P_E = 3.883$ ), and A ( $P_A = 4.334$ ). In Fig. 8 the distribution of the penalty score is shown: red is the summed penalty regarding the joint limits  $w_q \cdot P_q$ , orange reflects the penalty for the grasp quality  $w_{fc} \cdot P_{fc}$ . As grasp C has a very high grasp quality, its penalty factor is very low. Yellow depicts the penalty factor for the torque limits  $w_\tau \cdot P_\tau$ ,

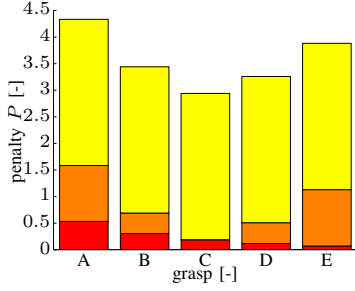


Fig. 8. Distribution of penalty scores over the complete trajectory of moving a 2D rectangle downwards. Red:  $w_q \cdot P_q$ , orange:  $w_{fc} \cdot P_{fc}$ , yellow:  $w_\tau \cdot P_\tau$ .

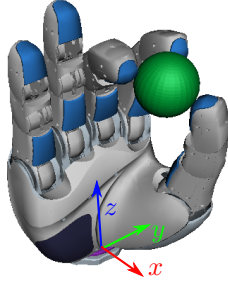


Fig. 9. Searching for an optimal grasp pose using predefined contact points (Problem 2).

which are almost the same for all grasp candidates.

For poses  $\mathbf{x}_5$  and  $\mathbf{x}_7$  there are different feasible grasp candidates, but no clear choice of an optimal one. We find grasp B and C, respectively, to be the optimal ones for these cases. Poses  $\mathbf{x}_6$ ,  $\mathbf{x}_8$ ,  $\mathbf{x}_9$  do not have grasp candidates according to [16], but our approach finds that grasps C, B, and G are respectively the best grasps for these start poses.

### C. Problem 2: Find an Optimal Object Pose Using a Pre-defined Force Closure Grasp

To illustrate the solution of Problem 2, a 3D example with the DLR/HIT Hand II is used. The hand parameters are given in Table I, and the specified grasp and coordinate axes are depicted in Fig. 9. The example task is rotating a sphere (radius 0.02m) around its  $y$ -direction for an angle of 11.5deg. The sphere weighs 0.08kg, so  $\mathbf{f}_{\text{ext}}^T = (0, 0, -0.8)$ . The rotation is obtained using three-finger grasps.

The bounding box of the set of all possible start positions  $\mathcal{X}$  is described with  $x \in [0.024, 0.074]$ ,  $y \in [0.04, 0.067]$ ,  $z \in [0.173, 0.19]$ . This cube is sampled with a step size of 0.005m. As the initial contact points on the object are fixed for the specified grasp, the object is rotated in each possible start pose around each one of its three axis from -30deg to 30deg with a step size of 10deg.

Fig. 10 depicts the valid start poses  $\mathcal{X}_{\text{final}}$  for the object. As the object is also rotated, some of the crosses stand for several object poses, e.g. position  $\mathbf{x}_r^T = [0.0550, 0.0250, 0.1835]$  has three valid orientations of the object associated. The best object configuration is at  $\mathbf{x}_{\text{opt}}^T = [0.055, 0.02, 0.1835]$  with a rotation of the object of -10deg around the  $z$ -direction.

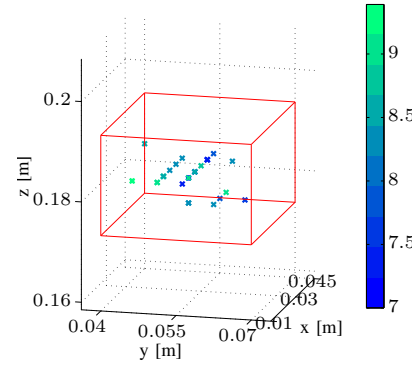


Fig. 10. Penalties  $P$  for each valid start object position.

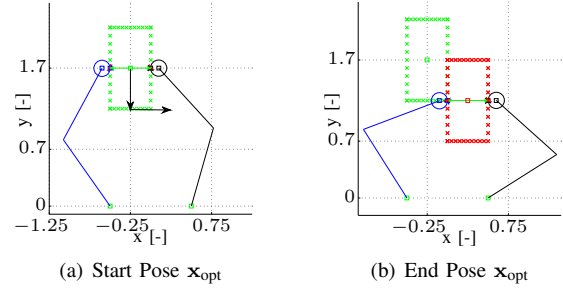


Fig. 11. Initial and final object poses and hand configurations for a down-side motion (Problem 3).

### D. Problem 3: Find an Optimal Grasp Configuration

The combination of both problems is analyzed for a down-side movement in 2D using the same hand and the same rectangle as in Section IV-B. The hand has now a realistic fingertip radius of  $r = 0.01$  with  $\gamma = \pi/2$ . Thus, only the frontal part of the fingertip can apply forces on the object. The defined trajectory (Fig. 11(a)) is

$$\mathbf{t}_{\text{des}} = \left[ \mathbf{x}, \dots, \mathbf{x} + \begin{pmatrix} 0 \\ -0.5 \\ 0 \end{pmatrix}, \dots, \mathbf{x} + \begin{pmatrix} 0.5 \\ -0.5 \\ 0 \end{pmatrix} \right], \quad (11)$$

and the external force is  $\mathbf{f}_{\text{ext}} = \begin{pmatrix} 0 \\ -0.75 \end{pmatrix}$ .

Fig. 12 depicts a penalty  $P_r$  for each valid start position of the object in  $\mathcal{X}_{\text{final}}$ . This penalty  $P_r$  is the minimal penalty for all possible grasps at that start position, including all possible orientations. The optimal object position is at  $\mathbf{x}_{\text{opt}} = \begin{pmatrix} -0.25 \\ 1.7 \end{pmatrix}$ . The optimal start and end position of the object with the corresponding hand configurations can be seen in Fig. 11(a) and Fig. 11(b).

Unlike [15], that solves the 2D problem relying on motion primitives, we can solve the problem for any trajectory in 2D space. The solution of Problem 3 in 3D is possible, but the combination of possible FC grasps explodes with the number of fingers and the number of feasible points for each finger.

## V. CONCLUSIONS

This paper presents a planning method to find a good set of contact points on the object and a corresponding hand

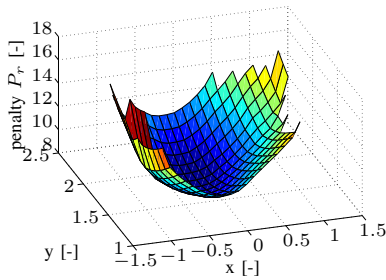


Fig. 12. Minimum penalty  $P$  for each object position for the down - side movement.

configuration, given the constraints of a manipulation task. These constraints are defined as a desired trajectory and an external force that needs to be applied or resisted. Due to the redundancy of the hand as a system, different possibilities for executing such task can be obtained. Previous methods start with predefined grasps on the object, and plan feasible locations for executing the desired trajectory considering only geometrical constraints in the workspace of the hand [15]. The computation of the initial grasp considering the task constraints has also been tackled before [16], but an initial grasp computed in this way does not guarantee that the task can be successfully executed.

The approach proposed in this paper considers at the same time the geometrical constraints of the manipulability workspace, and the conditions that need to be fulfilled during the task: joint and torque limits of the hand need to be observed, the fingers must apply forces on the object surface, and the grasp on the object during the movement required to fulfill the task must be force closure. By discretizing the desired trajectory and testing and scoring all four conditions in the via points of the desired motion, we are able to choose an optimal trajectory of the object inside the hand workspace, or equivalently, the initial grasp configuration (contact points and hand/object pose) that allows the optimal execution of the specified task.

Note that computational times for each problem type strongly depend on the number of points representing the object, the workspace of the hand, and the number of via points of the trajectory. In our case, Problems 1 took about 65min, Problem 2 (80min), and Problem 3 60min on a common desktop PC.

Several improvements can be made to the presented algorithm. For instance, to deal with more complex object geometries and to improve the computational time of the general 3D case, a previous analysis of the grasp space of the object can be realized [20], such that during the algorithm execution the possible combinations of contact points that lead to a FC grasp can be easily checked.

In the solution of the in-hand manipulation problems proposed in this paper, no joint velocity constraints have been taken into account. In a first approach, it could be considered that these constraints only limit the minimum time required to execute the desired task. However, dynamic

phenomena can become important, therefore demanding a measure of dynamic manipulability [21]. Using this measure in the solution of the problems presented here is also a possible extension of the proposed approach.

## VI. ACKNOWLEDGMENTS

This work was funded by the European Community's Seventh Framework Programme (FP7) under grant agreement No. ICT 248273, Project GeRT.

## REFERENCES

- [1] L. Biagiotti, F. Lotti, C. Melchiorri, and G. Vassura, "How far is the human hand? a review on anthropomorphic robotic end-effectors," *Internal Report - University of Bologna*, 2004.
- [2] A. Bicchi, "Hands for dexterous manipulation and robust grasping: A difficult road towards simplicity," *IEEE Trans. Robotics and Automation*, vol. 16, no. 6, pp. 652–662, 2000.
- [3] R. Ma and A. Dollar, "On dexterity and dexterous manipulation," in *Proc. Int. Conf. on Advanced Robotics - ICAR*, 2011, pp. 1–7.
- [4] Z. Li, J. Canny, and S. Sastry, "On motion planning for dexterous manipulation. part I: The problem formulation," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 1989, pp. 775–780.
- [5] P. Tournassoud, T. Lozano-Perez, and E. Mazer, "Regrasping," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 1987, pp. 1924–1928.
- [6] L. Han, Y. Guan, Z. Li, Q. Shi, and J. Trinkle, "Dextrous manipulation with rolling contacts," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 1997, pp. 992–997.
- [7] A. Cole, P. Hsu, and S. Sastry, "Dynamic control of sliding by robot hands for regrasping," *IEEE Trans. Robotics and Automation*, vol. 8, no. 1, pp. 42–52, 1992.
- [8] M. Cherif and K. Gupta, "3D in-hand manipulation planning," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1998, pp. 146–151.
- [9] D. Rus, "Dexterous rotations of polyhedra," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 1992, pp. 2758–2763.
- [10] J. Saut, A. Sahbani, S. El-Khoury, and V. Perdureau, "Dexterous manipulation planning using probabilistic roadmaps in continuous grasp subspaces," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2007, pp. 2907–2912.
- [11] A. Okamura, N. Smaby, and M. Cutkosky, "An overview of dexterous manipulation," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2000, pp. 255–262.
- [12] J. Kerr and B. Roth, "Analysis of multifingered robot hands," *Int. J. Robotics Research*, vol. 4, no. 4, pp. 3–17, 1986.
- [13] Y. Guan, H. Zhang, X. Zhang, and Z. Guan, "Workspace of 3-D multifingered manipulation," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008, pp. 2679–2684.
- [14] F. Zacharias, W. Sepp, C. Borst, and G. Hirzinger, "Using a model of the reachable workspace to position mobile manipulators for 3-D trajectories," in *Proc. IEEE-RAS Int. Conf. on Humanoid Robots*, 2009, pp. 55–61.
- [15] P. Michelman, "Precision object manipulation with a multifingered robot hand," *IEEE Trans. Robotics and Automation*, vol. 14, no. 1, pp. 105–113, 1998.
- [16] M. Sato and Y. Yoshikawa, "A grasp performance criterion for robot hands considering multiple aspects of tasks and hand configuration," in *IEEE Int. Conf. Robotics and Biomimetics*, 2011, pp. 1547–1554.
- [17] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, Florida: CRC Press, 1994.
- [18] H. Liu, K. Wu, P. Meusel, N. Seitz, G. Hirzinger, M. Jin, Y. Liu, S. Fan, T. Lan, and Z. Chen, "Multisensory five-finger dexterous hand: The DLR/HIT hand II," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008, pp. 3692–3697.
- [19] M. Sagardia, T. Hulin, C. Preusche, and G. Hirzinger, "Improvements of the voxmap-pointshell algorithm - fast generation of haptic data structures," in *Proc. 53rd Int. Wissenschaftliches Kolloquium*, 2008.
- [20] M. Roa, R. Suarez, and J. Rosell, "Grasp space generation using sampling and computation of independent regions," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2008, pp. 2258–2263.
- [21] Y. Yokokohji, J. Sanmartin, and M. Fujiwara, "Dynamic manipulability of multifingered grasping," *IEEE Trans. Robotics*, vol. 25, no. 4, pp. 947–954, 2009.