Finding All Valid Hand Configurations for a Given Precision Grasp

Carlos Rosales, Josep M. Porta, Raúl Suarez, and Lluís Ros

Abstract— **Planning a precision grasp for a robot hand is usually decomposed into two main steps. First, a set of contact points over the object surface must be determined, ensuring they allow a stable grasp. Second, the inverse kinematics of the robot hand must be solved to verify whether the contact points can actually be reached. Whereas the first problem has been largely solved in a general posing, the second one has only been tackled with local convergence methods. These methods only provide one solution to the problem, even if many are possible, and depending on the initial estimation they use, they may fail to converge, which results in grasp re-planning in situations where it could be avoided. This paper overcomes both issues by providing a complete method to solve the kinematics of human-like hands. The method is able to find all possible configurations that reach the specified contact points, even when positive-dimensional sets of such configurations are possible.**

I. INTRODUCTION

Human-like mechanical hands are flexible end effectors capable of performing complex manipulation tasks [1]–[6]. A recurrent topic in this context is the so-called *grasp problem*, or, given an object, determine how to hold it. For an anthropomorphic hand, this amounts to determine where and how to place the hand fingers so that they firmly hold the object. This definition is general enough to include different types of grasps [7] [8], but we are particularly interested in *precision grasps*, i.e., those in which only the fingertips contact the object, with just one contact point per finger allowed, assuming that the fingertips are locally convex and non-deformable.

Several constraints must be fulfilled in grasp planning. First, the forces exerted on the object must be able to compensate for the application of external perturbation forces. This is accomplished by satisfying force- or form-closure conditions, depending on whether the object is fully constrained by the finger forces or by the finger positions, respectively [9]. Second, it must be ensured that the contact points on the object are reachable by the fingers, by solving the inverse kinematics of the hand. Third, the grasp must be planned avoiding collisions between the different solids involved (the fingers, the palm, the object, and the environment). Finally, the solutions must be optimized taking into account dexterity, isotropy, or energetic criteria. In sum, a grasp planning problem must address constraints of different nature using different techniques, forcing the planning problem to be solved in a sequential way. This implies that if any of the steps in the sequence fails, then the overall process has to be restarted, which makes completeness an important issue. In particular, if one solution exists in the inverse kinematics stage, then it must be found. If several solutions are possible, it would be desirable to find them all so that subsequent planning steps can take all possibilities into account, in order to minimize the chances of re-planning.

While the problem of finding grasps satisfying the force- or form-closure conditions has been largely solved in a general posing since the early works (see e.g. [10]–[13]), the inverse kinematics of robot hands has only been approached with local convergence methods. Usually, these methods return a single solution, which depends on an initial estimation given to the solver, but their convergence is not always guaranteed. In other words, they can fail to provide a solution, even if one exists. Examples of local methods include those of Borst *et al.* [14], who formulate the problem as a set of unconstrained optimization problems where the contact and kinematic constraints and the joint limits are introduced as penalty terms in the objective function, Gorce and Rezzoug [15], who rely on a neural network to learn the finger inverse kinematics, and later employ reinforcement learning to optimize the pose of the hand, and Rosell *et al.* [16], who propose an optimization method to iteratively compute joint movements that maximally reduce the distance from the fingertips to the contact points.

In contrast to local methods, this paper offers a complete method to solve the inverse kinematics of a robot hand, based on a recent technique for the position analysis of general linkages [17]–[19]. In other words, the method is able to find all possible configurations that reach the specified contact points, even when positive-dimensional sets of such configurations are possible. The method has been implemented and tested succesfully on the four-finger hand MA-I [3], [20], [21] mounted on a Stäubli robot arm (Fig. 1), but it remains applicable to any anthropomorphic hand.

The rest of the paper is organized as follows. Section II specifies the addressed problem. Section III describes the kinematic structure of the hand-object system. Section IV gives a proper algebraic formulation to the problem, suitable to the resolution method proposed in Section V. Section VI describes some experiments illustrating the method's performance and, finally, Section VII summarizes the conclusions and points deserving further attention.
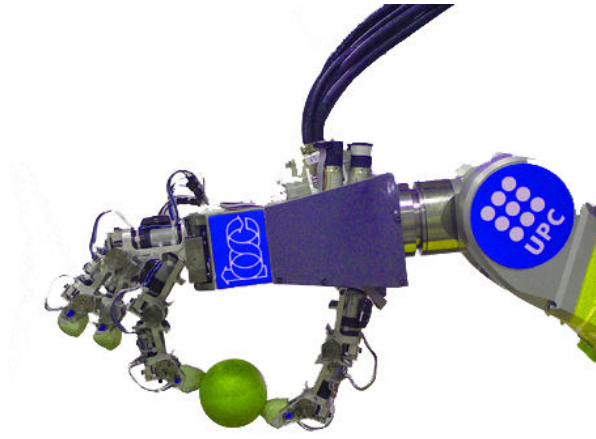
Fig. 1.   Mechanical Hand MA-I performing a two-finger precision grasp.

Fig. 2.   Principal involved frames, attached to the hand (h), the object (o), and the world (w), and homogeneous transformations relating them.

II. PROBLEM STATEMENT AND ASSUMPTIONS

A *precision grasp* is a set of contact points on the object surface, $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, where n is the number of fingers used in the grasp. Given a robot hand and a precision grasp, our goal is to determine all hand configurations for which the fingers can reach the specified contact points. In this paper, we will consider that the hand adopts the so-called *anthropomorphic* structure, i.e., the one that closely mimics the human hand, but the approach could be applied to other hand structures, as long as the mathematical formulation adopted follows the guidelines provided in [17]–[19]. The following additional hypotheses will be made.

It will be assumed that the finger-to-point assignment is given, letting finger j be assigned to point $\mathbf{x}_j$, and that the given grasp already satisfies any force- or form-closure conditions required. The fingertip and the object will be smooth and rigid, with a local shape that allows the contact to be established on a point only. Moreover, the paper will be focused on dexterous manipulation tasks without sliding or rolling on the contact point and, hence, this point will be fixed both on the object and on the fingertip surfaces. Finally, collisions among bodies will be neglected, as they are supposed to be considered in later stages of the grasp planning.

Note that if, as usual, the hand is mounted on a six-degrees-of-freedom manipulator, then the kinematic structure of the manipulator does not constrain the set of possible solutions of the problem, as long as the hand-object system is generically located in the interior of the robot's workspace. In general, thus, the problem boils down to computing the possible object poses relative to the hand, together with the finger joint angles that make the specified contacts possible. The situation is depicted in Fig. 2. Since an homogeneous transformation $\mathbf{T}_{w,o}$ placing the object relative to the world is usually known, once the possible object poses $\mathbf{T}_{h,o}$ relative to the hand are determined, the possible configurations of the robot manipulator are obtained by solving its inverse kinematics using $\mathbf{T}_{w,h} = \mathbf{T}_{w,o} \cdot \mathbf{T}_{h,o}^{-1}$ as input.
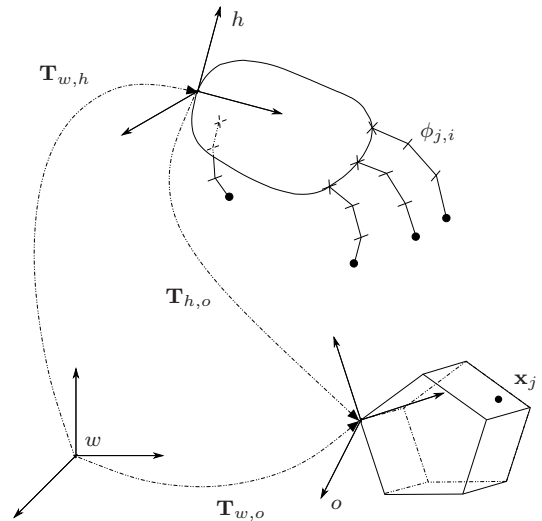
III. STRUCTURE OF THE HAND-OBJECT SYSTEM

Although every anthropomorphic hand has its own design, they are in general made up of a palm and several fingers, one of them acting as the thumb. Usually, non-thumb fingers are aligned with each other and with the palm, and the thumb is mounted asymmetrically, so that it can push against the other fingers to allow force-closure grasps. While some designs add an extra degree of freedom in the thumb and/or in the palm, a common trend is to design the fingers with the same kinematic structure, due to economic and modularity reasons.

An anthropomorphic finger is usually designed with four revolute joints, sequentially chained as follows (Fig. 3). The first and second joints are placed at the finger base, with their axes intersecting and orthogonal to each other, modeling the metacarpophalangeal joint. The third joint is placed in the middle of the finger, modeling the proximal interphalangeal joint. The fourth joint is placed near the fingertip, modeling the distal interphalangeal joint. While the first joint is responsible for abduction/adduction movements, the other three joints (whose axes are parallel) are used for flexion/extension movements. In sum, each finger has four degrees of freedom, which can either be independently actuated [3], some of them coupled [2], [4], [6], or even locked [5].

As for the kinematic model of the fingertip-object contact, two choices are basically possible. Since the contact point must be fixed on the two surfaces (see Section II), the simplest possibility is to assume that the fingertip and the object are linked by a spherical joint placed at the contact point. This model is valid for wire-like fingers [22], but it does not accurately implement the fact that, being solid and rigid, the fingers cannot penetrate the object surface. To model penetration-free contacts, we will assume that the fingertip and the object are, instead, articulated through a revolute joint placed at the contact point, with its axis aligned with the common normal to both solids on that point (Fig. 4).
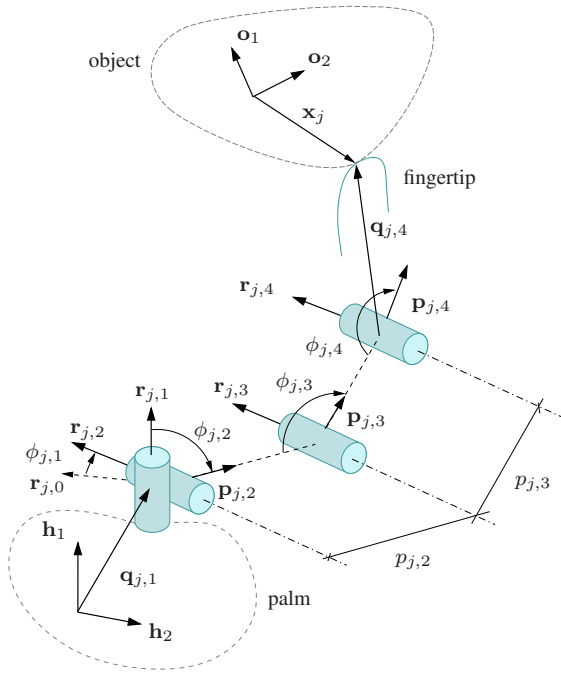
Fig. 3. Kinematic structure of an anthropomorphic finger. Revolute joints are represented as cylinders aligned with their axes. For clarity, the third vector of each reference frame is omitted.

Note that the hand-object system can be seen as a parallel robot. The object and the palm play the role of the "base" and "platform" links, respectively, and the fingers act as the robot "legs". In the usual notation [23], it is a n-URRR parallel platform, because each of its n legs involves a universal joint (the two orthogonal rotations) and three additional revolute joints (two for the phalanges, and another one for the contact). Finding the hand configurations that satisfy the given grasp is thus equivalent to the position analysis of such platform. Although this analogy could in principle be exploited, no technique seems to have been given yet to solve the specific n-URRR structure. Moreover, while the problem could be approached using elimination or homotopy techniques, the fact that the geometric parameters of the problem change every time a new grasp is presented frequently leads to singular mechanisms, so that relying on a general technique robust to singularities seems unavoidable. This is the reason that motivated the presented approach, which relies on an existing technique based on linear relaxations [17]–[19]. We next formulate the problem as required in [19], which entails forming a system of polynomial equations of special structure, characterizing the possible configurations of the hand-object system described above.

IV. MATHEMATICAL FORMULATION

A. Basic constraints

Let us define a reference frame for each one of the involved solids, namely the hand, the three phalanges of each finger and the object to be held (Fig. 3). The hand reference frame is defined by two orthonormal vectors $\mathbf{h}_1$ and $\mathbf{h}_2$, and their cross product $\mathbf{h}_3 = \mathbf{h}_1 \times \mathbf{h}_2$. We arbitrarily select the
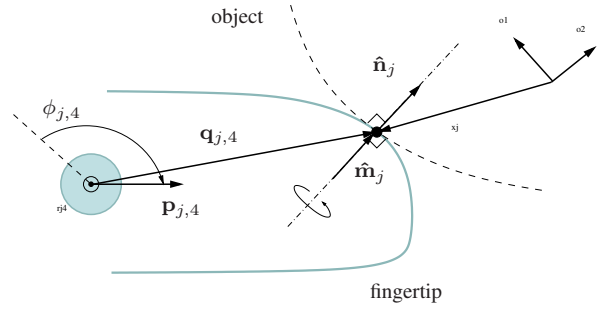

Fig. 4. Geometry of the fingertip-object contact.

hand frame to be the absolute frame and, hence, all vectors below will be described relative to this frame, except those marked with a hat that will be referred to other local frames. For each of the three phalanges of finger j, its corresponding frame is defined by two orthonormal vectors $\mathbf{r}_{j,i}$ and $\mathbf{p}_{j,i}$, and their cross product $\mathbf{t}_{j,i} = \mathbf{r}_{j,i} \times \mathbf{p}_{j,i}$, for $i = 2, 3, 4$. From the particular structure of the fingers, we can also assume that the origin of the reference frame of phalanx $i+1$ lies on the previous $\mathbf{p}_{j,i}$ axis. With this, the vector connecting the phalanx reference frames i and $i+1$ is $\mathbf{q}_{j,i} = p_{j,i} \cdot \mathbf{p}_{j,i}$, where $p_{j,i}$ are the distances shown in Fig. 3. Finally, the object reference frame is defined by two orthogonal vectors $\mathbf{o}_1$ and $\mathbf{o}_2$, and by $\mathbf{o}_3 = \mathbf{o}_1 \times \mathbf{o}_2$.

The following constant vectors are also defined. The finger anchor point is defined relative to the hand reference frame by a vector $\mathbf{q}_{j,1}$. The abduction/adduction axis for finger j is defined by a vector $\mathbf{r}_{j,1}$. The vector $\mathbf{r}_{j,0}$ defines the reference position for the rotation around $\mathbf{r}_{j,1}$. The contact point at the fingertip is defined by $\hat{\mathbf{q}}_{j,4}$ relative to the $(\mathbf{r}_{j,4}, \mathbf{p}_{j,4}, \mathbf{t}_{j,4})$ frame. This same point is defined relative to the object frame by a vector $\hat{\mathbf{x}}_j$. Note that $\hat{\mathbf{q}}_{j,4}$ and $\hat{\mathbf{x}}_j$ can be transformed into absolute coordinates using, respectively,

$$\mathbf{x}_j = (\mathbf{o}_1, \mathbf{o}_2, \mathbf{o}_3) \cdot \hat{\mathbf{x}}_j, \tag{1}$$

$$\mathbf{q}_{j,4} = (\mathbf{r}_{j,4}, \mathbf{p}_{j,4}, \mathbf{t}_{j,4}) \cdot \hat{\mathbf{q}}_{j,4}. \tag{2}$$

Note now that the joints between the different solids forming the hand-object system introduce constraints in the relative position between the reference frames defined. In particular, the last three joints of each finger have parallel axes, which implies that

$$\mathbf{r}_{j,2} = \mathbf{r}_{j,3} = \mathbf{r}_{j,4}, \tag{3}$$

and the fact that $\mathbf{r}_{j,2}$ is orthogonal to $\mathbf{r}_{j,1}$ requires that

$$\mathbf{r}_{j,1} \cdot \mathbf{r}_{j,2} = 0. \tag{4}$$

Moreover, to make the finger-object contact behave as a revolute joint (Fig. 4), we force the alignment of the fingertip and object normals as follows

$$(\mathbf{r}_{j,4}, \mathbf{p}_{j,4}, \mathbf{t}_{j,4}) \, \hat{\mathbf{m}}_j = (\mathbf{o}_1, \mathbf{o}_2, \mathbf{o}_3) \, \hat{\mathbf{n}}_j, \tag{5}$$

where $\hat{\mathbf{m}}_j$ and $\hat{\mathbf{n}}_j$ represent the contact point normal expressed in the fingertip and object frames, respectively.

As it is well known [19], when an object is grasped with n fingers, $n-1$ independent kinematic loops arise. The

grasping in Fig. 1, for example, creates one such loop. The loop equation formed by fingers j and k can be written as

$$\mathbf{x}_j - \sum_{i=1}^{4} \mathbf{q}_{j,i} = \mathbf{x}_k - \sum_{i=1}^{4} \mathbf{q}_{k,i}. \tag{6}$$

Thus, the system of equations characterising the valid configurations of the hand-object system includes (1) to (5) gathered for each finger j, the orthonormality constraints

$$\|\mathbf{r}_{j,i}\| = 1, \ \|\mathbf{p}_{j,i}\| = 1 \ \text{and} \ \mathbf{r}_{j,i} \cdot \mathbf{p}_{j,i} = 0 \tag{7}$$

and

$$\|\mathbf{o}_1\| = 1, \ \|\mathbf{o}_2\| = 1 \ \text{and} \ \mathbf{o}_1 \cdot \mathbf{o}_2 = 0, \tag{8}$$

relative to the reference frames, and (6) for each independent kinematic loop. Note that, since the vectors $\mathbf{h}_1$, $\mathbf{h}_2$, $\mathbf{r}_{j,0}$, $\mathbf{r}_{j,1}$, $\mathbf{q}_{j,1}$, $\hat{\mathbf{q}}_{j,4}$, and $\hat{\mathbf{x}}_j$ are constant in this system, the only variables involved are the vectors $\mathbf{r}_{j,2}$ and $\mathbf{p}_{j,i}$, on each finger j, for $i = 2, 3, 4$, and the vectors $\mathbf{o}_1$ and $\mathbf{o}_2$.

Observe, finally, that a solution to the system directly provides the joint angles $\phi_{j,i}$, via dot- and cross-products of the involved vectors, and the sought $\mathbf{T}_{h,o}$ transformation.

B. Joint limits constraints

So far, the fact that each one of the revolute joints has a limited range of rotation has not been taken into consideration. To constrain the rotation angle $\phi_{j,i}$ to lie within some range we define the variables

$$c_{j,i} = \cos(\phi_{j,i}),$$
$$s_{j,i} = \sin(\phi_{j,i}),$$

which can be easily related with the vectors of the previous formulation noting that

$$c_{j,1} = \mathbf{r}_{j,0} \cdot \mathbf{r}_{j,2}, \tag{9}$$
$$s_{j,1} \, \mathbf{r}_{j,1} = \mathbf{r}_{j,0} \times \mathbf{r}_{j,2}, \tag{10}$$
$$c_{j,2} = \mathbf{r}_{j,1} \cdot \mathbf{p}_{j,2}, \tag{11}$$
$$s_{j,2} \, \mathbf{r}_{j,2} = \mathbf{r}_{j,1} \times \mathbf{p}_{j,2}, \tag{12}$$
$$c_{j,3} = \mathbf{p}_{j,2} \cdot \mathbf{p}_{j,3}, \tag{13}$$
$$s_{j,3} \, \mathbf{r}_{j,3} = \mathbf{p}_{j,2} \times \mathbf{p}_{j,3}, \tag{14}$$
$$c_{j,4} = \mathbf{p}_{j,3} \cdot \mathbf{p}_{j,4}, \tag{15}$$
$$s_{j,4} \, \mathbf{r}_{j,4} = \mathbf{p}_{j,3} \times \mathbf{p}_{j,4}. \tag{16}$$

Clearly, the bounds for $\phi_{j,i}$ define corresponding bounds for $c_{j,i}$ and $s_{j,i}$. Thus, it is possible to constrain the range of the $\phi_{j,i}$ angles by incorporating (9)-(16) into the system of equations to be solved, together with the mentioned bounds on the $c_{j,i}$ and $s_{j,i}$ variables.

V. EQUATION SOLVING

The previous formulation has the particularity that all of its equations contain only linear, quadratic, or bilinear monomials. In other words, if x_i and x_j refer to any two of their variables, the monomials can only be of the form x_i, x_i^2 or $x_i x_j$. This structure fits particularly well with the relaxation technique presented in [17]–[19], which will be adopted here to obtain all problem solutions. To make the

paper self-contained, the technique is briefly outlined next. The reader is referred to [17]–[19] for further details.

To begin with, let us define the changes of variables $q_i = x_i^2$ and $b_k = x_i x_j$ for each quadratic and bilinear monomial, respectively. By substituting the q_i and b_k variables into (1)-(16), we obtain a new system of the form

$$L(\mathbf{x}) = 0, \tag{17}$$
$$Q(\mathbf{x}) = 0, \tag{18}$$
$$B(\mathbf{x}) = 0, \tag{19}$$

where $\mathbf{x}$ includes the original and newly defined variables, $L(\mathbf{x}) = 0$ is a block of linear equations, and $Q(\mathbf{x}) = 0$ and $B(\mathbf{x}) = 0$ are blocks of equations of the form $q_i = x_i^2$ and $b_k = x_i x_j$, respectively.

Note that since all unknowns in (1)-(16) are unit vectors, the x_i and b_i variables can only take values in the range $[-1, 1]$, and the q_i ones in $[0, 1]$. As a result, the search space where the solutions are to be sought is a rectangular box $\mathcal{B}$ formed by the Cartesian product of such ranges.

The algorithm isolates the solutions in $\mathcal{B}$ by iterating two operations, *box shrinking* and *box splitting*. Using box shrinking, portions of $\mathcal{B}$ containing no solution are eliminated by narrowing some of its defining intervals. This process is repeated until either (1) the box is reduced to an empty set, in which case it contains no solution, or (2) the box is "sufficiently" small, in which case it is considered a solution box, or (3) the box cannot be "significantly" reduced, in which case it is bisected into two sub-boxes via box splitting (which simply bisects its largest interval). To converge to all solutions, the whole process is repeated for the newly created sub-boxes, until one ends up with a collection of *solution boxes* whose side lengths are below a given threshold σ.

The crucial operation in this scheme is box shrinking, which [17]–[19] implement as follows. Note first that the solutions falling in some box $\mathcal{B}_c \subseteq \mathcal{B}$ must lie on the linear variety defined by $L(\mathbf{x}) = 0$. Thus, we may shrink $\mathcal{B}_c$ to the smallest possible box bounding this variety inside $\mathcal{B}_c$. The limits of this new box along, say, dimension x_i can be easily found by solving the two linear programs

LP1: Minimize x_i, subject to: $L(\mathbf{x}) = 0, \mathbf{x} \in \mathcal{B}_c$,

LP2: Maximize x_i, subject to: $L(\mathbf{x}) = 0, \mathbf{x} \in \mathcal{B}_c$.

However, note that the solutions also lie on the parabolas $q_i = x_i^2$ of $Q(\mathbf{x}) = 0$, and on the hyperbolic paraboloids $b_k = x_i x_j$ of $B(\mathbf{x}) = 0$. The two facts can be taken into account by noting that the portion of the parabola $q_i = x_i^2$ lying inside $\mathcal{B}_c$ is bounded by two half planes (Fig. 5(a)), and that the points of $\mathcal{B}_c$ verifying $b_k = x_i x_j$ necessarily lie inside a tetrahedron defined by four points, obtained by clipping $\mathcal{B}_c$ with $b_k = x_i x_j$ (Fig. 5(b)). Thus, the inequalities relative to such bounds can easily be added to **LP1** and **LP2** above, which usually produces a much larger reduction of $\mathcal{B}_c$, or even its complete elimination, if some of the linear programs is found unfeasible.

If (1)-(16) have a finite number of solutions, the previous algorithm returns a collection of small boxes containing them
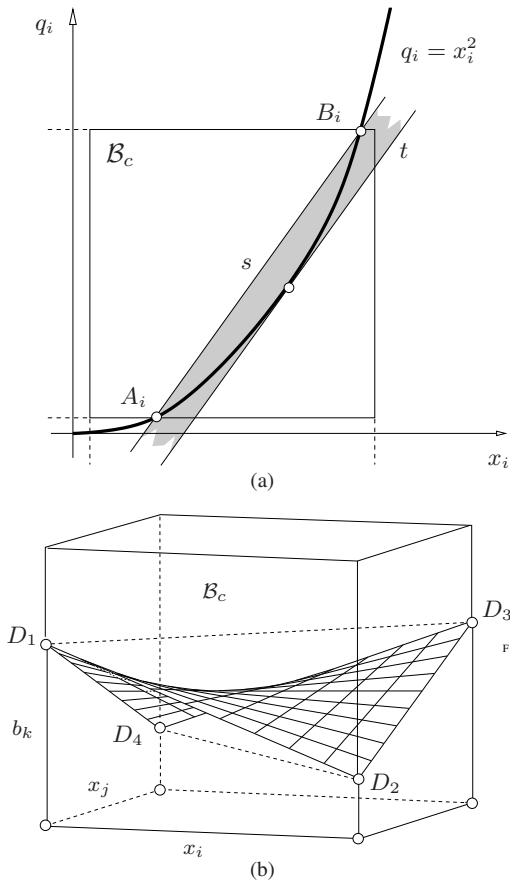
(a)

(b)

Fig. 5. Bounds used during box-shrinking. (a) Parabola bounds are obtained by clipping $q_i = x_i^2$ with $\mathcal{B}_c$, and getting the half-planes delimited by line s, through A_i and B_i, and its parallel line t, tangent to $q_i = x_i^2$. (b) Hyperbolic parabolid bounds are obtained by clipping $b_k = x_i x_j$ with $\mathcal{B}_c$, which defines the tetrahedron $D_1, \ldots, D_4$, and getting the half-spaces relative to the faces of this tetrahedron.

all, with each solution lying in one and only one box. If, on the contrary, the solution space is an algebraic variety of dimension one or higher, the returned boxes will form a discrete envelope of the variety. In any case, the algorithm is complete, and the accuracy of the output can be arbitrarily adjusted via the σ parameter, which, as said, limits the width of the widest interval on all solution boxes.

VI. TEST CASES

The approach has been tested succesfully for the particular case of the MA-I hand shown in Fig. 1. The Denavit-Hartenberg parameters of this hand are given in [20]. From them, it is easy to formulate the system of equations derived in Section IV, which has been solved in all cases with the CUIK package described in [17]–[19].

It is worth mentioning that the mathematical formulation adopted in Section IV is valid for a hand with any number of fingers. However, note that the dimension of the solution space, and hence the problem difficulty, directly depends on the number of fingers involved in the grasp. For a grasp using n fingers, the hand-object system has $f = 5n$ degrees of freedom (relative to the $5n$ revolute joints) but $r = 6(n-1)$

constraints (relative to the $n - 1$ kinematic loops created). Thus, by the Grübler-Kutzbach criterion, the dimension of the solution space will be $d = f - r = 6 - n$ in general. In our case, the dimension is two, since the MA-I hand is four-fingered. Two-dimensional solution spaces can certainly be isolated with the proposed technique, but the time to do so is excesively large on current desktop computers. In order to simplify the problem, however, one can introduce additional constraints to the system to be solved, as long as they are plausible for the adopted hand. We will do so in the test cases below, which illustrate, respectively, the behaviour of the solver on 0- and 1-dimensional solution spaces. Detailed data on these experiments, including the input/output files and some performance statistics of the technique can be found in [24].

A. Zero-dimensional spaces

In this experiment, the goal is to find all possible configurations of the hand holding a bottle, using four contact points on its surface. To make the solution space 0-dimensional, the proximal and distal joints of the ring and middle fingers have been coupled, letting $\phi_{j,3} = \phi_{j,4}$ for fingers $j = 1, 2$, which can be enforced by adding
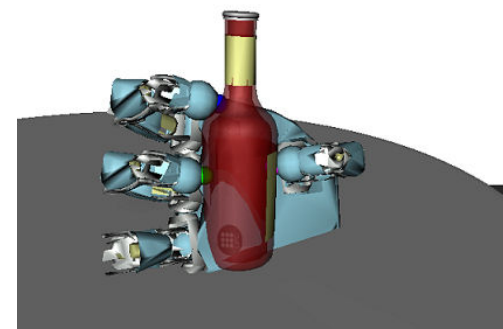
$$c_{j,3} = c_{j,4},$$
$$s_{j,3} = s_{j,4},$$

to the given formulation.

Fig. 6 shows two of the possible configurations found by the solver for a given precision grasp. Although both
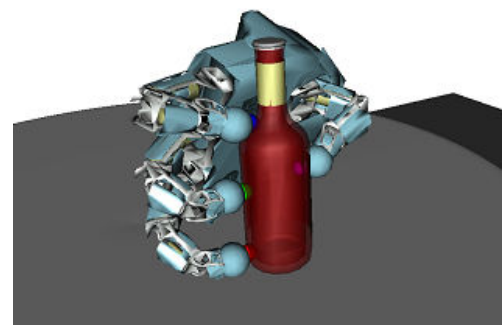
(a) A valid solution.

(b) A non-valid solution due to collision.

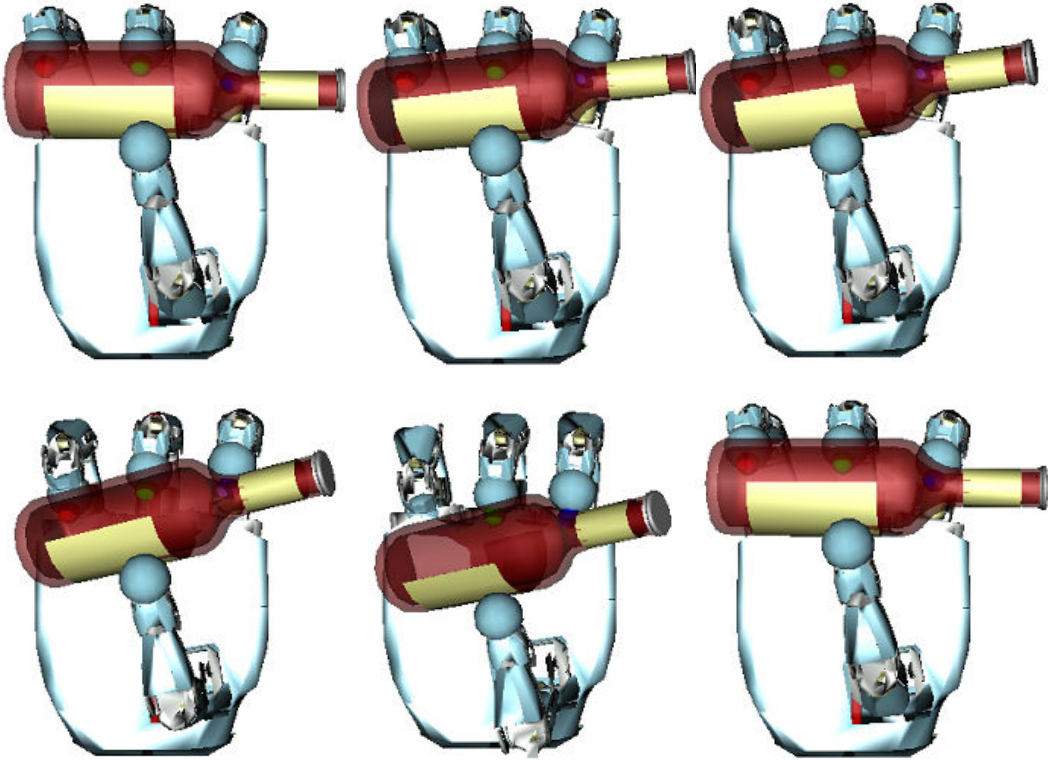Fig. 6. Different solutions for a given precision grasp.

Fig. 7.   Robot hand following a 1-dimensional solution subset to manipulate a bottle.

configurations are kinematically valid, note that the one in Fig. 6(a) is collision-free while that in Fig. 6(b) is not. On this specific case, a conventional solver would only return one of the kinematically valid solutions for this problem. If the returned solution happens to be the one in Fig. 6(b), note that the grasp would have to be re-planned. This is clearly avoidable using the proposed approach.

B. One-dimensional spaces

This second experiment is similar to the previous one, the only difference being that, now, only the proximal and distal joints of the ring finger are coupled, which yields a one-dimensional curve of solutions. In this case, the solver returns a box approximation of such curve, formed by a discrete continuum of aligned boxes. Note that the curve can be interpreted as giving a trajectory of the object relative to the hand, and it can thus be used as a roadmap to perform dexterous manipulation tasks. Fig. 7 shows several snapshots of the virtual model of the hand following one of the returned curves. The video accompanying this paper (also available in [24]) provides a continuous animation of the hand performing this motion. Up to our knowledge, none of the previously existing solvers for grasp planning can handle problems of this kind, where the assumed constraints allow for object motions relative to the hand.

VII. Conclusions and future work

This paper has presented a complete method to solve the inverse kinematics of an anthropomorphic hand. The

approach identifies *all* possible hand configurations reaching a given set of grasping points. This is an advantage over existing approaches, which can only provide a single solution, and fail to find a solution in some cases, even if one exits. Moreover, the proposed method can deal with problems where the assumed kinematic constraints allow for object motions relative to the hand.

In the future, we plan to enhance the solution techniques so as to be able to confront higher-dimensional problems in reasonable computation times, as well as to integrate other grasp planning stages into our approach. In this sense, we are currently working in the integration of collision constraints into the system of equations, to also make the solver return collision-free configurations. The ultimate solution would be to also include the force/form closure constraints into such system, which would eliminate the finger-to-point assignment problem, and the artificial decoupling of the grasp planning process into several stages. Simultaneously, the integration of contact models of higher complexity [25], accounting for rolling motions of the fingers over the object, is also being investigated.

REFERENCES

[1] W. Townsend, "The BarrettHand grasper - programmably flexible part handling and assembly," *Industrial Robot*, vol. 10(3), pp. 181–188, May 2000.

[2] J. Butterfass, M. Grebenstein, H. Liu, and G. Hirzinger, "DLR-Hand II: Next generation of dextrous robot hand," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 2001, pp. 109–114.

[3] R. Suárez and P. Grosch, "Mechanical hand MA-I as experimental system for grasping and manipulation," in *VideoProceedings of the 2005 IEEE International Conference on Robotics and Automation*, Barcelona, Spain, Apr. 2005.

[4] C. Lovchik and M. Diftler, "The Robonaut hand: A dexterous robot hand for space," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 1999, pp. 907–912.

[5] F. Lotti, P. Tiezzi, G. Vassura, L. Biagiotti, G. Palli, and C. Melchiorri, "Development of UB hand 3: Early results," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2005, pp. 4499–4504.

[6] Shadow Hand Homepage, http://www.shadowrobot.com/hand.

[7] M. Cutkosky, "On grasp choice, grasp models, and the design of hands for manufacturing tasks," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 269–279, 1989.

[8] N. Kamakura, M. Matsuo, H. Ishii, F. Mitsuboshi, and Y. Miura, "Patterns of static prehension in normal hands," *Americal Journal of Occupational Therapy*, vol. 34, pp. 437–445, 1980.

[9] A. Bicchi, "On the closure properties of robotic grasping," *International Journal of Robotic Research*, vol. 14, no. 4, pp. 319–334, 1995.

[10] X. Markenscoff, L. Ni, and C. Papadimitriou, "The geometry of grasping," *International Journal of Robotics Research*, vol. 9, no. 1, pp. 61–74, 1990.

[11] B. Mishra, J. Schwartz, and M. Sharir, "On the existence and synthesis of multifinger positive grips," *Algorithmica*, vol. 2, no. 4, pp. 541–558, 1987.

[12] V. Nguyen, "Constructing force-closure grasps," *International Journal of Robotics Research*, vol. 7, no. 3, pp. 3–16, 1988.

[13] C. Ferrari and J. Canny, "Planning optimal grasps," vol. 3(12-14), May 1992, pp. 2290–2295.

[14] C. Borst, M. Fischer, and G. Hirzinger, "Calculating hand configurations for precision and pinch grasps," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, Oct. 2002, pp. 1553–1559.

[15] P. Gorce and N. Rezzoug, "Grasping posture learning with noisy sensing information for a large scale of multifingered robotic systems," *Journal of Robotic Systems*, vol. 22(12), pp. 711–724, May 2005.

[16] J. Rosell, X. Sierra, L. Palomo, and R. Suárez, "Finding grasping configuration of a dextrous hand and an industrial robot," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain, Apr. 2005, pp. 1190–1195.

[17] J. M. Porta, L. Ros, and F. Thomas, "Multi-loop position analysis via iterated linear programming," in *Robotics: Science and Systems II*. MIT Press, 2006, pp. 169–178.

[18] J. M. Porta, L. Ros, T. Creemers, and F. Thomas, "Box approximations of planar linkage configuration spaces," *ASME Journal of Mechanical Design*, vol. 129, no. 4, pp. 397–405, 2007.

[19] J. M. Porta, L. Ros, and F. Thomas, "A linear relaxation technique for the position analysis of multi-loop linkages," Institut de Robòtica i Informàtica Industrial, Llorens Artigas 4-6, 08028 Barcelona, Tech. Rep., 2008, available through http://www-iri.upc.es.

[20] R. Suárez and P. Grosch, "Dexterous robotic hand MA-I, software and hardware architecture," in *International Conference on Intelligent Manipulation and Grasping*, Geneve, Italy, July 2004, pp. 91–96.

[21] J. A. Mora-Fillat and S. Domene, "Diseño de una mano de robot para manipulación diestra de objetos," Industrial Eng. thesis, Technical University of Catalonia, Barcelona, Spain, Sept. 1996.

[22] A. Srinath and A. Rao, "Kinematic chains for robot hands: Grasp and rigidity," *Mechanisms and Machine Theory*, vol. 42(6), pp. 691–697, June 2007.

[23] J.-P. Merlet, *Parallel Robots*. Springer, 2000.

[24] The CUIK Project web site, http://www-iri.upc.es/groups/gmr/cuikweb.

[25] E. Staffetti and F. Thomas, "Analysis of rigid-body interactions for compliant motion tasks using the Grassmann-Cayley algebra," in *Proceedings of the International Conference on Intelligent Robots and Systems*, vol. 3, nov 2000, pp. 2325–2332.