Synthesizing Grasp Configurations
with Specified Contact Regions

Carlos Rosales[1,2], Lluís Ros[1], Josep M. Porta[1], and Raúl Suárez[2]

(1) Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Barcelona, Spain

(2) Institut d'Organització i Control de Sistemes Industrials (UPC), Barcelona, Spain

Abstract

This paper presents a new method to solve the configuration problem on robotic hands: determine how a hand should be configured so as to grasp a given object in a specific way, characterized by a number of hand-object contacts to be satisfied. In contrast to previous algorithms given for the same purpose, the one presented here allows specifing such contacts between free-form regions on the hand and object surfaces, and always returns a solution whenever one exists. The method is based on formulating the problem as a system of polynomial equations of special form, and then exploiting this form to isolate the solutions, using a numerical technique based on linear relaxations. The approach is general, in the sense that it can be applied to any grasping mechanism involving lower-pair joints, and it can accommodate as many hand-object contacts as required. Experiments are included that illustrate the performance of the method in the particular case of the Schunk Anthropomorphic hand.

Keywords: Configuration problem, precision grasp, grasp planning, grasp synthesis, contact constraint, position analysis, inverse kinematics, anthropomorphic hand, prehension.

Figure 1: A typical grasp configuration for a scalpel can be specified by requiring the contact of regions $h_1, \ldots, h_4$ of the hand, with regions $o_1, \ldots, o_4$ on the object (top). The configuration problem is to determine how should the hand be configured relative to the object, in order to bring the hand regions into contact with their corresponding object regions (bottom).

1 Introduction

Substantial efforts have been done in Robotics thus far, to endow robots with the ability to grasp and dexterously manipulate objects with multifingered hands (Siciliano and Khatib 2008). Several aspects of this ability have been investigated, including 1) the determination of object contact points on which a form- or force-closure grasp (Bicchi 1995) is

guaranteed (Dizioğlu and Lakshiminarayana 1984; Markenscoff et al. 1990; Ferrari and Canny 1992; Ponce et al. 1997; Cornellà and Suárez 2009); 2) the delimitation of object regions such that an arbitrary contact on these regions assures a force/form closure grasp (Nguyen 1988; Trinkle et al. 1995; Pollard 2004; Roa and Suárez 2009); 3) the computation of finger forces required to equilibrate an external force applied on the object (Kerr and Roth 1986; Kumar and Waldron 1989; Buss et al. 1996; Cornellà et al. 2008); 4) the planning of joint motions that would allow a stable and manipulable displacement of the object (Li et al. 1989; Bicchi and Kumar 2001; Arimoto 2007; Saut et al. 2007); or 5) the synthesis of hand configurations satisfying a number of grasping conditions (Borst et al. 2002; Guan and Zhang 2003; Gorce and Rezzoug 2005; Miller and Allen 2004; Rosell et al. 2005; Ciocarlie and Allen 2009).

This paper addresses a problem within the latter aspect. Given a number of regions on the surface of the hand, and a number of corresponding regions on the surface of the object, determine how should the hand be configured relative to the object so that each hand region establishes contact on its corresponding object region. Fig. 1 illustrates the problem with an example. This problem, referred to as the *configuration problem* hereafter, arises when the object is to be grasped and manipulated in a specific way, characterized by a number of contact constraints to be satisfied (Borst et al. 2002; Guan and Zhang 2003; Gorce and Rezzoug 2005; Rosell et al. 2005). While the object regions may be the outcome of an algorithm for contact region delimitation (Nguyen 1988; Trinkle et al. 1995; Pollard 2004; Roa and Suárez 2009), those on the hand may derive from known patterns of static prehension (Kamakura et al. 1980), and the pairing between object and hand regions may be done using representative points on each region (Woelfl and Pfeiffer 1994; Fernández et al. 2005).

The configuration problem has mostly been addressed with local search methods to date. Examples of such methods include those proposed by Borst et al. (2002), who cast the problem into one of unconstrained optimization where the various constraints are introduced as penalty terms in an objective function, Gorce and Rezzoug (2005), 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. (2005), who propose an iterative method to compute joint displacements that maximally reduce the distance from the fingertips to the contact points. Although such methods are usually fast and return a solution in many cases, their convergence is not always guaranteed, even if a solution exists. Some of such methods, moreover, require a sufficiently-good initial estimation of the solution (Borst et al. 2002; Rosell et al. 2005), which might not always be available.

This work attempts to find a way around such limitations by proposing a new algorithm of guaranteed convergence; i.e., one that always provides a solution whenever one exists. This algorithm, which extends one introduced by Rosales et al. (2008), does not require an initial estimation of the solution and can, in fact, solve a superclass of the configuration problems dealt with by Borst et al. (2002), Gorce and Rezzoug (2005), Rosell et al. (2005), and Rosales et al. (2008), because all contact constraints considered in such works can be seen as particular cases of more general ones tractable herein.

The proposed algorithm is based on formulating the problem as a system of polynomial equations of special form, and then exploiting such form to solve the equations, using an extended version of a recent method based on linear relaxations (Porta et al. 2009). It must be noted that, whereas the algorithm in (Porta et al. 2009) can deal with lower-pair mechanisms of general structure, it can not be directly applied to the configuration problem of mechanical hands, because it is unable to cope with general contact constraints between free-form surfaces. Here, we extend that algorithm to be able to specify such constraints between Bézier patches defined anywhere on the object or on the hand, and to solve the corresponding equations.

The rest of this paper is organized as follows. To see which constraints come into play, Section 2 reviews the kinematic structure of existing anthropomorphic hands and describes how the hand-object contacts are specified. Section 3 shows that, reflecting such constraints, the configuration problem can be formulated as a system of polynomial equations. Section 4 presents a numerical method to isolate the solutions of this system. Section 5 illustrates the performance of the approach on the particular case of the Schunk Anthropomorphic (SA) hand—a

commercial version of the DLR II hand (Butterfass et al. 2004)—on various tasks requiring an object to be grasped in a special way. Finally, Section 6 gives the paper conclusions and highlights points deserving further attention.

2 The hand-object system

2.1 Structure of the hand

Although each anthropomorphic hand follows a particular design, all hands are in general made up of a palm and several fingers, one of them acting as the thumb. Usually, all fingers are aligned with each other and with the palm, except the thumb, which is mounted asymmetrically so that it can push against the other fingers. Each finger is composed of several phalanges, usually articulated through revolute (R) or universal (U) joints, whose degrees of freedom may be actuated, unactuated, or coupled to those of other joints. Mechanical limitations usually exist, that constrain these degrees of freedom to take values within prescribed ranges.

Many finger designs follow an URR structure or slight variations of it. This structure closely mimics that of the human finger (Napier 1993). It mounts a universal joint at the finger base, to model the metacarpophalangeal joint, and two additional revolute joints, to model the proximal and distal interphalangeal joints (Fig. 2). The axis of the U joint that is fixed to the palm is responsible for abduction/adduction movements, and the remaining axes, which are usually parallel, are responsible for flexion/extension movements of the finger.

The thumb structure is more diverse and controversial (Giurintano et al. 1995; Valero-Cuevas et al. 2003). Designs are found where the thumb adopts the same structure as that of the remaining fingers, which facilitates the construction of the hand. Other designs either decrease or increase the mobility of the thumb, by removing or adding joints with respect to the basic URR design. In all cases, however, the tip of the thumb is allowed to face all other fingertips, so as to be able to grasp and manipulate objects under stable prehensions. A summary of representative hand designs adopted during the last decade is provided in Table 1.

Note that, to reduce the number of motors necessary to actuate the hand, many hands have coupled
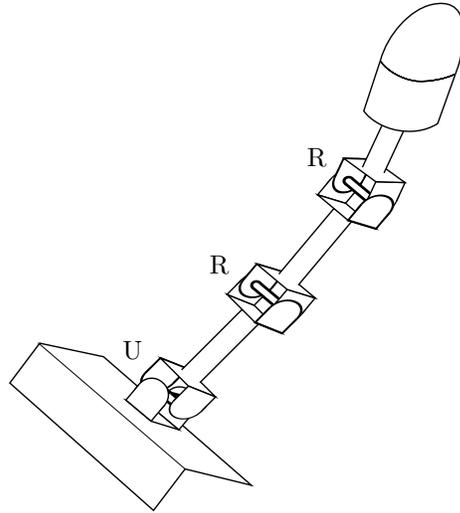
Figure 2: Common URR structure of an anthropomorphic finger.

degrees of freedom. The coupling of two joints A and B is indicated as $\widehat{AB}$ in Table 1, meaning that a rotation about an axis of A produces an identical rotation about an axis of B. On a coupling $\widehat{UR}$ only the parallel axes are coupled.

2.2 Contact constraint specification

The contact constraints to be fulfilled are assumed to be given as a collection of pairs (h_c, o_c), $c = 1, \ldots, b$, where h_c and o_c are two-dimensional regions on the hand and object surfaces, respectively. The constraint (h_c, o_c) is meant to require the contact of h_c and o_c at some point, with the normals to h_c and o_c aligned at such point, to avoid the interpenetration of the regions.

By convention, h_c and o_c are assumed to be given as polynomial patches. That is, it is assumed that a polynomial function of the form

$$\mathbf{p} = \mathbf{p}(u, v), \qquad (1)$$

is given for each region, providing the parametric coordinates $\mathbf{p} = (p_x, p_y, p_z)$ of a point P in the region, in terms of some scalar parameters u and v, bound to lie within the interval $[0, 1]$. To properly align the normals of h_c and o_c, the parameterization $\mathbf{p}(u, v)$ is supposed to be nondegenerate, in the sense that, if $\mathbf{p}_u$ and $\mathbf{p}_v$ are the partial derivatives

3

Hand	#Actuated d.o.f.	Little	Ring	Middle	Index	Thumb
DIST hand (Caffaz and Cannata 1998)	16	-	URR			
Robonaut hand (Lovchik and Diftler 1999)	12	RR̂R̂R		UR̂R		ÛR
LMS hand (Gazeau et al. 2001)	16	-	URR			
Ultralight Anthropom. hand (Schulz et al. 2001)	10	RR̂R				ÛR̂R
GIFU II hand (Kawasaki et al. 2002)	16	UR̂R				URR
Shadow Robot hand (Shadow Robot Company 2003)	18	RUR̂R	UR̂R			UUR
DLR II hand (Butterfass et al. 2004)	13	-	UR̂R			RUR̂R
UBH 3 hand (Lotti et al. 2004)	20	URR				
MA-I hand (Suárez and Grosch 2005)	16	-	URR			
SA hand (Schunk GmbH & Co. KG 2006)	13	-	UR̂R			RUR̂R
Twendy-One hand (Iwata and Sugano 2009)	13	-	UR̂R			RUR

Table 1: Representative hand designs adopted during the last decade.

of $\mathbf{p}(u,v)$ with respect to u and v, then the normal vector to the patch, defined as

$$\mathbf{n} = \mathbf{p}_u \times \mathbf{p}_v, \qquad (2)$$

never vanishes for $(u,v) \in [0,1] \times [0,1]$.

For ease of explanation, $\mathbf{p}(u,v)$ will adopt the form of a standard Bézier patch of some given degree $M \times N$,

$$\mathbf{p}(u,v) = \sum_{i=0}^{M} \sum_{j=0}^{N} \mathbf{b}_{i,j} \cdot B_{i,M}(u) \cdot B_{j,N}(v), \qquad (3)$$

where $\mathbf{b}_{i,j}$ denote the Bézier control points of the patch, and $B_{i,j}(x) = \binom{j}{i} x^i (1-x)^{j-i}$ is the ith Bernstein polynomial of degree j. Note that any polynomial paramaterization $\mathbf{p}(u,v)$ can be converted into such form, by using an appropriate change of basis (Farin 2001).

3 Kinematic equations

The configuration problem can be formulated as a number of constraints that the poses of the hand and object links must fulfill. This section formulates such constraints mathematically, following the methodology proposed by Porta et al. (2009). Once gathered together, the constraints form a system of polynomial equations characterizing all possible solutions of the configuration problem. The special structure of this system will be beneficial to solve the problem numerically, as it will be shown in Section 4.

3.1 Link constraints

It will be convenient to label the hand and object links as $L_0, L_1, \ldots, L_n$, where L_0 is the palm link, $L_1, \ldots, L_{n-1}$ are the various phalange links, and L_n is the object link. The joints of the hand will also be labelled for reference, as $J_1, \ldots, J_m$.

Each link L_l, $l = 0, \ldots, n$, will be furnished with a local reference frame $\mathcal{F}_l$, and we will let the reference frame of the palm link, $\mathcal{F}_0$, to act as the absolute frame. Moreover, each frame will have an associated vector basis, and we will write $\mathbf{v}^{\mathcal{F}_l}$ to refer to the coordinates of vector $\mathbf{v}$, written in the basis of $\mathcal{F}_l$. Vectors with no superscript will either be expressed in the basis of the absolute frame, or in no particular frame, depending on the context.

With the previous notation, a *configuration* of the hand-object system will be an assignment of a *pose* $(\mathbf{r}_l, \mathbf{R}_l)$ to each link L_l, $l = 1, \ldots, n$, where $\mathbf{r}_l \in \mathbb{R}^3$ is the position of the origin of $\mathcal{F}_l$ with respect to $\mathcal{F}_0$, and $\mathbf{R}_l$ is a 3×3 rotation matrix giving the orientation of $\mathcal{F}_l$ relative to $\mathcal{F}_0$. The elements
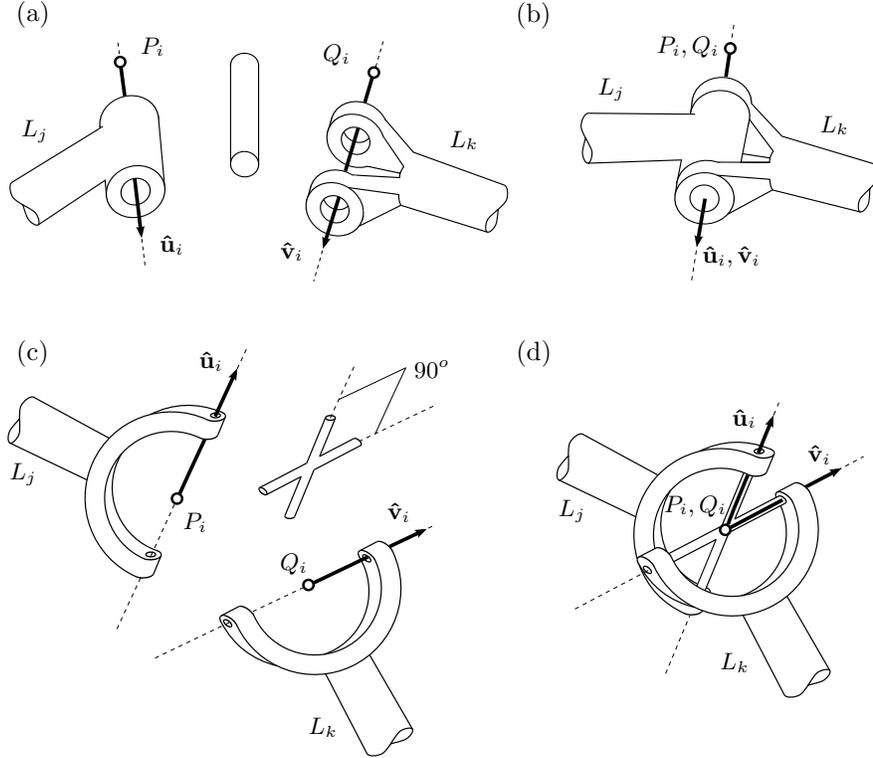
Figure 3: (a,b): The assembly of two links through a revolute joint is specified by imposing the coincidence of two points and the alignment of two vectors. (c,d): The assembly through a universal joint is specified by imposing the coincidence of two points and the orthogonality of two vectors. [Figure adapted from
Porta et al. (2009).]

of the rotation matrices are not independent, because if $\mathbf{R}_l$ has the form $(\hat{\mathbf{c}}_l, \hat{\mathbf{d}}_l, \hat{\mathbf{e}}_l)$, then it must be

$$
\begin{aligned}
\|\hat{\mathbf{c}}_l\|^2 &= 1, & (4) \\
\|\hat{\mathbf{d}}_l\|^2 &= 1, & (5) \\
\hat{\mathbf{c}}_l \cdot \hat{\mathbf{d}}_l &= 0, & (6) \\
\hat{\mathbf{c}}_l \times \hat{\mathbf{d}}_l &= \hat{\mathbf{e}}_l, & (7)
\end{aligned}
$$

for $l = 1, \ldots, n$, in order for $\mathbf{R}_l$ to represent a valid rotation. Note that the joints, the contacts, and the mechanical limits impose additional constraints on the link poses. These constraints are next formulated explicitly.

3.2   Joint assembly constraints

Since most hand designs only resort to revolute or universal joints (Table 1), we focus on formulating the constraints imposed by such joints, but other joint types would be formulated in a similar way (Porta et al. 2009).

In terms of spatial constraints, the assembly of two links L_j and L_k, through a revolute joint J_i, is equivalent to imposing the coincidence of two points, P_i and Q_i, and the alignment of two unit vectors, $\hat{\mathbf{u}}_i$ and $\hat{\mathbf{v}}_i$, respectively fixed to L_j and L_k (Fig. 3a). These two points and vectors are chosen on the axis of the joint, and they coalesce into a single point and vector when the two links get assembled (Fig. 3b). The coincidence and alignment

5

conditions can be written, respectively, as

$$\mathbf{r}_j + \mathbf{R}_j \, \mathbf{p}_i^{\mathcal{F}_j} = \mathbf{r}_k + \mathbf{R}_k \, \mathbf{q}_i^{\mathcal{F}_k}, \qquad (8)$$

$$\mathbf{R}_j \, \hat{\mathbf{u}}_i^{\mathcal{F}_j} = \mathbf{R}_k \, \hat{\mathbf{v}}_i^{\mathcal{F}_k}, \qquad (9)$$

where $\mathbf{p}_i^{\mathcal{F}_j}$ and $\mathbf{q}_i^{\mathcal{F}_k}$ refer to the position vectors of P_i and Q_i in frames $\mathcal{F}_j$ and $\mathcal{F}_k$, respectively. The valid poses of the two links, hence, are those that fulfill Eqs. (8) and (9) simultaneously.

Similarly, if J_i is a universal joint, the valid poses of L_j and L_k are those that fulfill

$$\mathbf{r}_j + \mathbf{R}_j \, \mathbf{p}_i^{\mathcal{F}_j} = \mathbf{r}_k + \mathbf{R}_k \, \mathbf{q}_i^{\mathcal{F}_k}, \qquad (10)$$

$$\mathbf{R}_j \, \hat{\mathbf{u}}_i^{\mathcal{F}_j} \cdot \mathbf{R}_k \, \hat{\mathbf{v}}_i^{\mathcal{F}_k} = 0 \qquad (11)$$

where Eqs. (10) and (11) impose the coincidence of two points P_i and Q_i, and the orthogonality of two unit vectors $\hat{\mathbf{u}}_i$ and $\hat{\mathbf{v}}_i$, respectively fixed on L_j and L_k. The points are located on the center of the universal joint, on $\mathcal{F}_j$ and $\mathcal{F}_k$. The vectors are aligned with the axes of the joint on such frames (Figs. 3c and 3d). Since vectors $\mathbf{p}_i^{\mathcal{F}_j}$, $\mathbf{q}_i^{\mathcal{F}_k}$, $\hat{\mathbf{u}}_i^{\mathcal{F}_j}$, and $\hat{\mathbf{v}}_i^{\mathcal{F}_k}$ are known a priori, the only unknowns in Eqs. (8)-(11) are the poses of the two links $(\mathbf{r}_j, \mathbf{R}_j)$ and $(\mathbf{r}_k, \mathbf{R}_k)$.

3.3 Joint limit constraints

For a revolute joint J_i incident to links L_j and L_k, the relative angle between L_j and L_k, denoted ϕ_i, is the angle between two unit vectors $\hat{\mathbf{a}}_i$ and $\hat{\mathbf{b}}_i$ orthogonal to the axis of J_i, fixed in L_j and L_k, respectively. Usually, due to the existence of mechanical limits, ϕ_i can only take values within a prescribed interval which, using a proper location for $\hat{\mathbf{a}}_i$ and $\hat{\mathbf{b}}_i$, can always be written in the form $[-\alpha_i, \alpha_i]$, with $\alpha_i \in [0, \pi]$. In our formulation, these limits can be taken into account by constraining the cosine of ϕ_i. For this, we define a new variable $c_i = \cos(\phi_i)$, and observe that the constraint $\phi_i \in [-\alpha_i, \alpha_i]$ is equivalent to the constraint $c_i \in [\cos \alpha_i, 1]$. Then we note that

$$c_i = \hat{\mathbf{a}}_i \cdot \hat{\mathbf{b}}_i. \qquad (12)$$

where

$$\hat{\mathbf{a}}_i = \mathbf{R}_j \, \hat{\mathbf{a}}_i^{\mathcal{F}_j}, \qquad (13)$$

$$\hat{\mathbf{b}}_i = \mathbf{R}_k \, \hat{\mathbf{b}}_i^{\mathcal{F}_k}. \qquad (14)$$

Thus, to constrain ϕ_i to the range $[-\alpha_i, \alpha_i]$ it is only necessary to add Eqs. (12)-(14) to the system to be solved, taking into account that c_i can only take values in the range $[\cos \alpha_i, 1]$. Joint limits for a universal joint can be imposed in a similar way.

3.4 Contact constraints

Let us suppose that in the required grasp some hand link L_k is required to be in contact with the object link L_n, where the contact has to be established between given regions h_c and o_c defined on L_k and L_n, respectively (Fig. 4). Let $H_c \in h_c$ and $O_c \in o_c$ be two points on such regions, with position vectors $\mathbf{h}_c^{\mathcal{F}_k}$ and $\mathbf{o}_c^{\mathcal{F}_n}$ relative to $\mathcal{F}_k$ and $\mathcal{F}_n$, respectively, and let $\hat{\mathbf{m}}_c$ and $\hat{\mathbf{n}}_c$ denote unit normal vectors to the link surface at such points. Then, the poses of L_k and L_n that bring the two regions in contact through H_c and O_c are those that fulfill

$$\mathbf{r}_k + \mathbf{R}_k \, \mathbf{h}_c^{\mathcal{F}_k} = \mathbf{r}_n + \mathbf{R}_n \, \mathbf{o}_c^{\mathcal{F}_n}, \qquad (15)$$

$$\mathbf{R}_k \, \hat{\mathbf{m}}_c^{\mathcal{F}_k} = -\mathbf{R}_n \, \hat{\mathbf{n}}_c^{\mathcal{F}_n}, \qquad (16)$$

where Eq. (15) imposes the coincidence of H_c and O_c, and Eq. (16) establishes the alignment of $\hat{\mathbf{m}}_c$ and $\hat{\mathbf{n}}_c$.

All vectors and matrices in Eq. (15) are unknowns. However, since H_c and O_c are bound to lie on h_c and o_c, the additional constraints

$$\mathbf{h}_c^{\mathcal{F}_k} = \mathbf{h}_c^{\mathcal{F}_k}(u_c, v_c), \qquad (17)$$

$$\mathbf{o}_c^{\mathcal{F}_n} = \mathbf{o}_c^{\mathcal{F}_n}(s_c, t_c), \qquad (18)$$

must be taken into account to properly formulate the contact, where $\mathbf{h}_c^{\mathcal{F}_k}(u_c, v_c)$ and $\mathbf{o}_c^{\mathcal{F}_n}(s_c, t_c)$ are parametric descriptions of regions h_c and o_c, given in the form of Eq. (3). Note that the Bézier control points of the patches $\mathbf{h}_c^{\mathcal{F}_k}(u_c, v_c)$ and $\mathbf{o}_c^{\mathcal{F}_n}(s_c, t_c)$ must be given in frames $\mathcal{F}_k$ and $\mathcal{F}_n$, respectively.

Analogously, the unit vectors $\hat{\mathbf{m}}_c^{\mathcal{F}_k}$ and $\hat{\mathbf{n}}_c^{\mathcal{F}_n}$ in Eq. (16) must also be related to the patch parameters. This relationship can be established by taking into account that, for a parametric patch $\mathbf{p}(u, v)$ of the form of Eq. (3), the normal vector $\mathbf{n}(u, v)$ defined by Eq. (2) can be written as

$$\mathbf{n}(u, v) =$$

$$\sum_{i=0}^{2M-1} \sum_{j=0}^{2N-1} \mathbf{b}'_{i,j} \cdot B_{i,2M-1}(u) \cdot B_{j,2N-1}(v), \quad (19)$$

so that it can be thought of as a new Bézier patch, but now of degree $(2M - 1) \times (2N - 1)$. Explicit formulas for computing the control points $\mathbf{b}'_{i,j}$ in this expression, in terms of the control points $\mathbf{b}_{i,j}$ of $\mathbf{p}(u, v)$, are given by Yamaguchi (1997). Thus, $\hat{\mathbf{m}}_c^{\mathcal{F}_k}$ and $\hat{\mathbf{n}}_c^{\mathcal{F}_n}$ can be related to the patch parameters by defining two unnormalized vectors $\mathbf{m}_c^{\mathcal{F}_k}$ and $\mathbf{n}_c^{\mathcal{F}_n}$, and their norms μ_c and ν_c, placed in correspondence with $\hat{\mathbf{m}}_c^{\mathcal{F}_k}$ and $\hat{\mathbf{n}}_c^{\mathcal{F}_n}$ through the constraints

$$\mu_c^2 = \|\mathbf{m}_c^{\mathcal{F}_k}\|^2, \tag{20}$$

$$\nu_c^2 = \|\mathbf{n}_c^{\mathcal{F}_n}\|^2, \tag{21}$$

$$\mathbf{m}_c^{\mathcal{F}_k} = \mu_c \hat{\mathbf{m}}_c^{\mathcal{F}_k}, \tag{22}$$

$$\mathbf{n}_c^{\mathcal{F}_n} = \nu_c \hat{\mathbf{n}}_c^{\mathcal{F}_n}, \tag{23}$$

and setting the additional constraints

$$\mathbf{m}_c^{\mathcal{F}_k} = \mathbf{m}_c^{\mathcal{F}_k}(u_c, v_c), \tag{24}$$

$$\mathbf{n}_c^{\mathcal{F}_n} = \mathbf{n}_c^{\mathcal{F}_n}(s_c, t_c), \tag{25}$$

whose right-hand sides follow the form of Eq. (19).

3.5 Final system of equations

Summarizing, the final system of equations defining the possible grasp configurations will be formed by Eqs. (4)-(7) for each link, Eqs. (8) and (9) for each revolute joint, Eqs. (10) and (11) for each universal joint, equations of the form of (12)-(14) for each joint limit constraint, and Eqs. (15)-(18) and (20)-(25) for each contact constraint. Note that the variables involved in this system are:

- The pose variables $(\mathbf{r}_l, \mathbf{R}_l)$ corresponding to links L_l, $l = 1, \ldots, n$.

- The variables $\hat{\mathbf{a}}_i$, $\hat{\mathbf{b}}_i$, and c_i corresponding to the joint limit constraints on all joints J_i, $i = 1, \ldots, m$.

- The contact point coordinates $\mathbf{h}_c^{\mathcal{F}_k}$ and $\mathbf{o}_c^{\mathcal{F}_n}$, associated normal vectors $\mathbf{m}_c^{\mathcal{F}_k}$, $\mathbf{n}_c^{\mathcal{F}_n}$, $\hat{\mathbf{m}}_c^{\mathcal{F}_k}$, $\hat{\mathbf{n}}_c^{\mathcal{F}_n}$, vector norms μ_c and ν_c, and parameters u_c, v_c, s_c, and t_c, corresponding to all contact constraints (h_c, o_c), $c = 1, \ldots, b$.

It is worth mentioning that the $\mathbf{r}_l$ variables of this system can actually be eliminated, through a process explained in detail by Porta et al. (2009). The elimination is based on the fact that, for a loop of
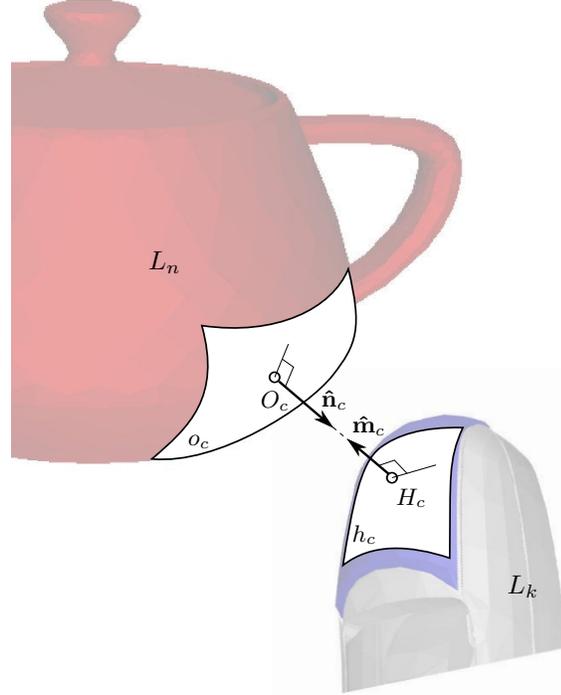
Figure 4: Elements intervening in a contact constraint (h_c, o_c). The constraint is satisfied when points $O_c \in o_c$ and $H_c \in h_c$ coincide, with the normals on such points aligned.

links pairwise constrained by joint or contact constraints, Eqs. (8), (10), and (15) occurring along the loop can be substituted by an equivalent "loop-closure" equation which is their sum, which does not contain any of the $\mathbf{r}_l$ variables. This process simplifies the system, and can always be invoked if desired, but the numerical method that follows is equally applicable to both the original and the simplified system.

4 Numerical Solution

Let n_e and n_v be, respectively, the number of equations and variables of the final system described in Section 3.5. This system can be compactly written as

$$\boldsymbol{\Phi}(\mathbf{q}) = 0, \tag{26}$$

where $\mathbf{q} = (q_1, \ldots, q_{n_v})$ refers to the vector of its variables, and $\boldsymbol{\Phi} : \mathbb{R}^{n_v} \to \mathbb{R}^{n_e}$ refers to the vector-

valued function describing its equations. A numerical method able to solve this system is next described, based on the approach proposed by Porta et al. (2009). The approach entails expanding the equations to a canonical form (Section 4.1) and then using a branch-and-prune method exploiting this form to isolate the solutions (Sections 4.2 and 4.3).

4.1 Equation expansion

We distinguish two groups of equations in the final system $\mathbf{\Phi}(\mathbf{q}) = 0$. A first group encompassing Eqs. (17), (18), (24), and (25), whose polynomials follow the Bézier form of Eqs. (3) and (19), and a second group encompassing the remaining equations, whose polynomials only contain monomials of the form q_i, q_i^2 and $q_i q_j$. Note that all equations of the second group can be easily converted into linear form by introducing the changes of variables

$$p_i = q_i^2 \qquad (27)$$
$$b_k = q_i q_j \qquad (28)$$

for all q_i^2 and $q_i q_j$ monomials occuring in them. After such changes, we obtain a new system of the form

$$\left. \begin{array}{l} \mathbf{\Lambda}(\mathbf{x}) = 0 \\ \mathbf{\Psi}(\mathbf{x}) = 0 \end{array} \right\}, \qquad (29)$$

where $\mathbf{x}$ is an n_x-dimensional vector encompassing all of the original q_i variables, and the newly-introduced p_i and b_k ones. Here, $\mathbf{\Lambda}(\mathbf{x}) = 0$ represents a collection of linear equations in $\mathbf{x}$, and $\mathbf{\Psi}(\mathbf{x}) = 0$ represents a collection of equations, each of which can only adopt one of these three forms:

$$x_k = x_i^2, \qquad (30)$$
$$x_k = x_i x_j, \qquad (31)$$
$$x_k = f(x_i, x_j). \qquad (32)$$

While the first two forms correspond to the changes of variables in Eqs. (27) and (28), the latter form corresponds to the scalar components of Eqs. (17), (18), (24), and (25), so that $f(x_i, x_j)$ refers to a Bernstein-form polynomial of degrees d_i and d_j in x_i and x_j, respectively.

4.2 Equation solving

It can be seen that, under the used formulation, each variable x_i of $\mathbf{x}$ can only take values within a prescribed interval (Porta et al. 2009), so that the Cartesian product of all such intervals defines an initial n_x-dimensional box $\mathcal{B} \subset \mathbb{R}^{n_x}$ which bounds all solutions of Eqs. (29). The algorithm to isolate such solutions recursively applies two operations on $\mathcal{B}$: *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 the box through its largest interval). To converge to all solutions, the whole process is recursively applied to the new sub-boxes, until one ends up with a collection of solution boxes, whose side lengths are below a given threshold σ.

As it turns out, this algorithm explores a binary tree of boxes, whose internal nodes correspond to boxes that have been split at some time, and whose leaves are either solution or empty boxes. By properly implementing the bookkeeping of boxes awaiting to be processed, this tree can be explored either in depth- or breadth-first order, the choice of order depending on whether one wishes to isolate just one solution, or the entire solution set.

Note that the algorithm is complete, in the sense that the solution boxes it returns include all solution points of Eqs. (29). Thus, the algorithm will always succeed in isolating a solution, whenever one exists, provided that a small-enough value of the σ parameter is used. Detailed properties of the algorithm, together with examples of its output, are given by Porta et al. (2007, 2009).

4.3 Box shrinking

We next see how a given sub-box $\mathcal{B}_c \subseteq \mathcal{B}$ can be reduced, discarding portions of the box that contain no solution.

First, observe that the solutions of Eqs. (29) lying within $\mathcal{B}_c \subseteq \mathcal{B}$ must lie on the linear variety defined by $\mathbf{\Lambda}(\mathbf{x}) = 0$. Thus, in principle, we might shrink $\mathcal{B}_c$ to the smallest possible box bounding this variety inside $\mathcal{B}_c$. The lower and upper limits of the shrunk box along dimension x_i, $i = 1, \ldots, n_x$, would respectively be found by solving the two lin-
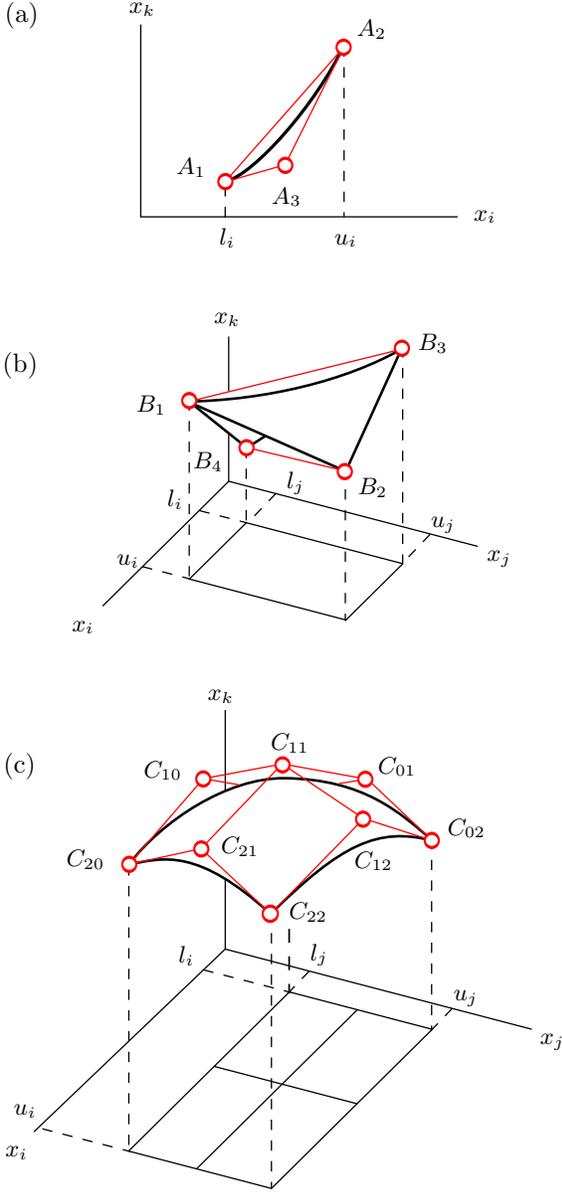
Figure 5: Polytope bounds within $\mathcal{B}_c$. (a) The points on $x_k = x_i^2$ are bound by the triangle $A_1 A_2 A_3$. (b) The points on $x_k = x_i x_j$ are bound by the tetrahedron $B_1 B_2 B_3 B_4$. (c) The points on $x_k = f(x_i, x_k)$ are bound by the convex hull of the points C_{pq}. In this example, $f(x_i, x_k)$ is a Bernstein-form polynomial of degree two in x_i and x_j, so that the control points C_{pq} form a grid of size 3×3.

ear programs

LP1: Minimize x_i, subject to: $\mathbf{\Lambda}(\mathbf{x}) = 0, \mathbf{x} \in \mathcal{B}_c$,
LP2: Maximize x_i, subject to: $\mathbf{\Lambda}(\mathbf{x}) = 0, \mathbf{x} \in \mathcal{B}_c$.

$\mathcal{B}_c$ may be further reduced, however, because the solutions must also satisfy the equations $\mathbf{\Psi}(\mathbf{x}) = 0$. These equations can be taken into account by noting that for each equation it is possible to define a convex polytope that bounds the equation solutions within $\mathcal{B}_c$. Thus, to better delimit the solutions of the system, $\mathcal{B}_c$ can be safely reduced to the smallest possible box enclosing the intersection of $\mathbf{\Lambda}(\mathbf{x}) = 0$ and the polytopes of all equations in $\mathbf{\Psi}(\mathbf{x}) = 0$. This reduction can be implemented by representing the individual polytopes with linear inequalities, and adding such inequalities to the constraint set of the linear programs **LP1** and **LP2**. We next see how such polytopes can be derived, for each one of Eqs. (30)-(32). The notation $[l_i, u_i]$ will refer to the interval of $\mathcal{B}_c$ relative to x_i.

To derive a polytope for $x_k = x_i^2$, note that the portion of the parabola $x_k = x_i^2$ lying within $\mathcal{B}_c$ is bound by the triangle $A_1 A_2 A_3$ in the x_i-x_k plane, where A_1 and A_2 are the points where the parabola intercepts the lines $x_i = l_i$ and $x_i = u_i$, and A_3 is the point where the tangent lines at A_1 and A_2 meet (Fig. 5a). Thus, the polytope of $x_k = x_i^2$ is defined by the triangle $A_1 A_2 A_3$, which can be represented by three inequalities that correspond to the three edges of this triangle.

To derive a polytope for $x_k = x_i x_j$, we realize that the portion of the surface $x_k = x_i x_j$ included in $\mathcal{B}_c$ is bound by a tetrahedron $B_1 B_2 B_3 B_4$ in the x_i-x_j-x_k subspace, whose vertices B_i are obtained by lifting the four corners of the rectangle $[l_i, u_i] \times [l_j, u_j]$ vertically to the surface $x_k = x_i x_j$ (Fig. 5b). Thus, the polytope of $x_k = x_i x_j$ is defined by the tetrahedron $B_1 B_2 B_3 B_4$, and can be represented by four inequalities, corresponding to the four faces of this tetrahedron.

Finally, to derive a polytope for $x_k = f(x_i, x_j)$, we resort to the subdivision and convex-hull properties of Bernstein polynomials (Farin 2001). Using the subdivision property, on the one hand, $f(x_i, x_j)$ is written in the form

$$f(x_i, x_j) = \sum_{p=0}^{d_i} \sum_{q=0}^{d_j} b_{p,q} \cdot B_{p,d_i}(x_i) \cdot B_{q,d_j}(x_j),$$

9

Number of contacts	2 slightly constrained	3 moderately constrained	4 highly constrained
Object		**scalpel**	
Task name Task requirements	Upholding The task requires picking the scalpel up by means of a two-fingered grasp, using the index finger and the thumb.	Handling The task requires handling the scalpel delicately using the middle finger, the thumb, and the palm.	Incision The hand must contact the scalpel as in a usual grasping of a pencil, using the middle and index fingers, the thumb, and the palm.
n_v, n_e, d Computed solution	$219, 209, 17$	$243, 235, 16$	$331, 324, 18$
CPU time [s]	106	255	418
Object		**teapot**	
Task name Task requirements	Lid lifting The lid must be pulled up through its knob using a two-fingered grasp involving the index finger and the thumb.	Service The hand is required to hold the teapot by its handle, placing the thumb on top of the handle, while the index and middle fingers embrace the handle.	Transportation The palm has to contact the bottom of the teapot, while the index, middle, and ring fingers enclose the teapot so that it does not slide out of the hand.
n_v, n_e, d Computed solution	$219, 209, 17$	$288, 278, 19$	$312, 305, 17$
CPU time [s]	114	262	375
Object		**guitar**	
Task name Task requirements	Tunning The tunning task requires the hand to grasp a given key of the guitar with the index finger and the thumb, in order to tune the tension of the corresponding string.	Playing The fingertips must contact at specified strings and frets in order to perform a given chord, while the thumb contacts the guitar neck.	Holding The task requires an almost whole-hand grasp of the guitar, on a specific region where the guitar can not be damaged while being transported.
n_v, n_e, d Computed solution	$219, 209, 17$	$307, 298, 19$	$331, 324, 18$
CPU time [s]	68	229	664
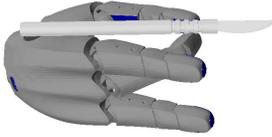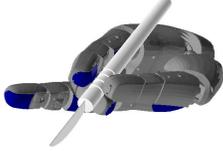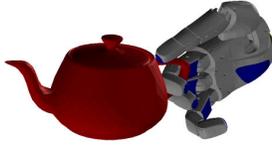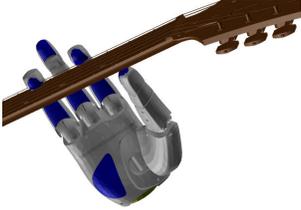
Table 2: Benchmark configuration problems and their computed solutions

where the scalars $b_{p,q}$ are the so-called control points of $f(x_i, x_j)$ relative to the interval $[l_i, u_i] \times [l_j, u_j]$. Using the convex-hull property, on the other hand, we know that the surface $x_k = f(x_i, x_j)$ must be contained inside the convex-hull of the 3D points C_{pq} with coordinates

$$\mathbf{c}_{pq} = \left(l_i + \frac{p}{d_i}(u_i - l_i), l_j + \frac{q}{d_j}(u_j - l_j), b_{p,q}\right), \quad (33)$$

for $p = 0, \ldots, d_i$, and $q = 0, \ldots, d_j$ (Fig. 5c). This convex hull defines a polytope for equation $x_k = f(x_i, x_j)$, which can be encoded as a set of inequalities by resorting to an algorithm for convex-hull computations (Barber et al. 1996).

5   Experiments

The presented method has been implemented in C, extending the libraries of the CUIK platform (Porta et al. 2009). This section illustrates the performance of the method under this implementation, on various experiments where an object needs to be grasped in a particular way, in order to fulfill a given task.

The experiments involve the solution of various configuration problems defined on a scalpel, a teapot, and a guitar, where each problem involves a number of regions to be placed in contact, imposed by the specific requirements of the task to be accomplished with the object (Table 2). In all experiments, the SA hand has been used to grasp the objects (Fig. 6), but the presented methodology is equally applicable to any other hand. While the area of all contact regions defined on the hand is approximately 40% of the fingertip area (the dark patches on the upper limbs in Fig. 6b), the area of the contact regions on the object varies from experiment to experiment, from 2% of the fingertip area on the teapot knob ("lid lifting" experiment), to 9000% of such area on the guitar neck ("playing" experiment).

We next explain how the equations of the hand can be set up, and later discuss the algorithm's performance on the mentioned experiments.

5.1   Equations for the SA hand

The SA hand is composed of four identical fingers that follow the anthropomorphic structure illustrated in Fig. 2. Three of these fingers are directly mounted on the palm, and act as ring, middle, and index fingers. The fourth finger is mounted on an intermediate link articulated with the palm through a revolute joint, which allows this finger to act as a thumb (Fig. 6). The hand has a total of fourteen links (one palm and thirteen phalanges) and thirteen joints (nine revolute joints and four universal joints).

To set up the equations, the links of the hand are labelled as $L_0, \ldots, L_{13}$, as shown in Fig. 6, and the joints as $J_1, \ldots, J_{13}$, letting J_i be the joint between L_{i-1} and L_i (for clarity, joint labels are not shown in Fig. 6). Twenty-six points and unit vectors are then defined, that provide the positions and orientations of all rotation axes of the hand relative to the involved links. The points correspond to the centers of the universal joints and to the midpoints of the revolute joints. The vectors correspond to unit vectors aligned with the rotation axes of the joints. These points and vectors are displayed in Fig. 6 and their coordinates are given in Table 3, in milimeters. All reference frames $\mathcal{F}_l$ are located with their origin in Q_l, so that $\mathbf{q}_l^{\mathcal{F}_l} = (0, 0, 0)$, for $l = 0, \ldots, 13$. The orientations of such frames can be deduced easily from the coordinates provided in Table 3. Taking into account these definitions, Eqs. (4)-(11) can readily be written for all links and joints involved.

To write down the equations of Section 3.3, the mechanical limits of the SA hand must be considered. Regarding the universal joints, the rotations about their $\hat{\mathbf{u}}_i$ and $\hat{\mathbf{v}}_i$ axes are limited to the ranges $[-15^o, 15^o]$ and $[-4^o, 75^o]$, respectively. Regarding the revolute joints, all of them can only rotate in the range $[4^o, 75^o]$, except for the revolute joint at the base of the thumb, which is restricted to the range $[0^o, 90^o]$. The reference configuration corresponding to setting all rotation angles to zero is shown in Fig. 6b.

Finally, it must be taken into account that not all joints of the SA hand are independently actuated. The two distal joints of each finger are coupled, so that when one of such joints is actuated, a rotation of the same angle about the other is produced. In the adopted formulation, the coupling of two rotation angles is simply imposed by equating the sine and cosine of such angles.

11
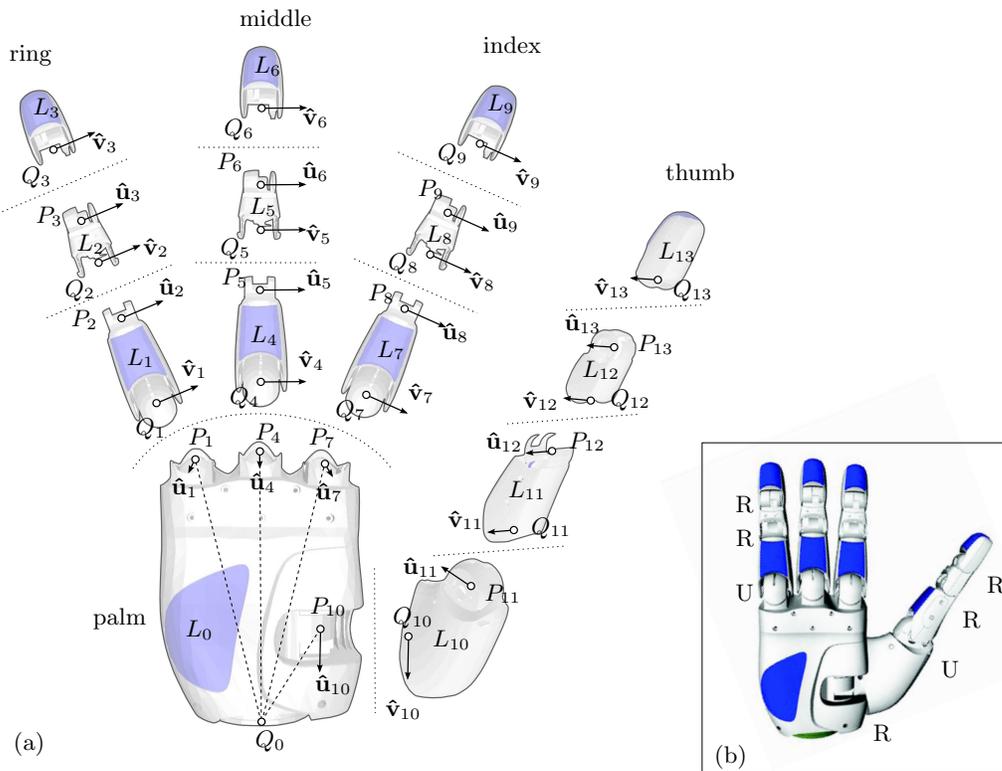
Figure 6: Geometric parameters (a) and reference configuration (b) of the Schunk Anthropomorphic Hand. The various joint types are indicated in (b).

5.2 Computed solutions

A system of equations has to be solved for each task of Table 2, encompassing the equations of the SA hand, together with the contact equations that impose the specific requirements of the task. It must be noted that Eqs. (4)-(11) relative to fingers not in contact with the object can actually be removed from this system, because such fingers do not intervene in any kinematic loop, and hence impose no loop-closure constraint on the overall system.

Table 2 provides the size of the equation system (26) to be solved in each case, in terms of the number of variables (n_v) and equations (n_e) it involves, and the dimension of its solution space (d), predicted as the number of variables minus the number of non-redundant equations. Note in this regard that Eq. (9) introduces equations that are redundant in terms of predicting such dimension, because $\hat{\mathbf{u}}_i$ and $\hat{\mathbf{v}}_i$ are unit vectors, and it is

sufficient to establish the x and y components of Eq. (9) to determine the alignment of L_k relative to L_j. The third component of Eq. (9), however, is needed to remove a sign ambiguity in such alignment. Since a similar redundancy is introduced by Eq. (16), there will be as many redundant equations as the number of joints and contacts involved in the problem at hand.

As it can be observed from Table 2, typical configuration problems yield solution spaces of rather high dimension. To avoid the curse of dimensionality as much as possible, and converge to one solution rapidly, the proposed algorithm must be set to explore in depth-first order (Section 4.2). Running the algorithm in this order, we have obtained the hand-object configurations depicted, in the CPU times indicated in each case. All times reported correspond to a parallelized version of the algorithm, executed on a grid of 8 DELL Poweredge computers, equipped with two Intel Quadcore Xeon

Joint type	Ring		Middle		Index		Thumb	
	Par.	Value	Par.	Value	Par.	Value	Par.	Value
R	$\mathbf{p}_3^{\mathcal{F}_2}$ $\hat{\mathbf{u}}_3^{\mathcal{F}_2}$ $\hat{\mathbf{v}}_3^{\mathcal{F}_3}$	$(30,0,0)$ $(0,1,0)$ $(0,1,0)$	$\mathbf{p}_6^{\mathcal{F}_4}$ $\hat{\mathbf{u}}_6^{\mathcal{F}_5}$ $\hat{\mathbf{v}}_6^{\mathcal{F}_6}$	$(30,0,0)$ $(0,1,0)$ $(0,1,0)$	$\mathbf{p}_9^{\mathcal{F}_7}$ $\hat{\mathbf{u}}_9^{\mathcal{F}_8}$ $\hat{\mathbf{v}}_9^{\mathcal{F}_9}$	$(30,0,0)$ $(0,1,0)$ $(0,1,0)$	$\mathbf{p}_{13}^{\mathcal{F}_{12}}$ $\hat{\mathbf{u}}_{13}^{\mathcal{F}_{12}}$ $\hat{\mathbf{v}}_{13}^{\mathcal{F}_{13}}$	$(30,0,0)$ $(0,1,0)$ $(0,1,0)$
R	$\mathbf{p}_2^{\mathcal{F}_1}$ $\hat{\mathbf{u}}_2^{\mathcal{F}_1}$ $\hat{\mathbf{v}}_2^{\mathcal{F}_2}$	$(67.80,0,0)$ $(0,1,0)$ $(0,1,0)$	$\mathbf{p}_5^{\mathcal{F}_4}$ $\hat{\mathbf{u}}_5^{\mathcal{F}_4}$ $\hat{\mathbf{v}}_5^{\mathcal{F}_5}$	$(67.80,0,0)$ $(0,1,0)$ $(0,1,0)$	$\mathbf{p}_8^{\mathcal{F}_7}$ $\hat{\mathbf{u}}_8^{\mathcal{F}_7}$ $\hat{\mathbf{v}}_8^{\mathcal{F}_8}$	$(67.80,0,0)$ $(0,1,0)$ $(0,1,0)$	$\mathbf{p}_{12}^{\mathcal{F}_{11}}$ $\hat{\mathbf{u}}_{12}^{\mathcal{F}_{11}}$ $\hat{\mathbf{v}}_{12}^{\mathcal{F}_{12}}$	$(67.80,0,0)$ $(0,1,0)$ $(0,1,0)$
U	$\mathbf{p}_1^{\mathcal{F}_0}$ $\hat{\mathbf{u}}_1^{\mathcal{F}_0}$ $\hat{\mathbf{v}}_1^{\mathcal{F}_1}$	$(-4.30,-40.16,145.43)$ $(1,0,0)$ $(0,1,0)$	$\mathbf{p}_4^{\mathcal{F}_0}$ $\hat{\mathbf{u}}_4^{\mathcal{F}_0}$ $\hat{\mathbf{v}}_4^{\mathcal{F}_4}$	$(-4.30,0,145.43)$ $(1,0,0)$ $(0,1,0)$	$\mathbf{p}_7^{\mathcal{F}_0}$ $\hat{\mathbf{u}}_7^{\mathcal{F}_0}$ $\hat{\mathbf{v}}_7^{\mathcal{F}_7}$	$(-4.30,40.16,145.43)$ $(1,0,0)$ $(0,1,0)$	$\mathbf{p}_{11}^{\mathcal{F}_{10}}$ $\hat{\mathbf{u}}_{11}^{\mathcal{F}_{10}}$ $\hat{\mathbf{v}}_{11}^{\mathcal{F}_{11}}$	$(97,6,-87)$ $(\cos 55^o,0,\sin 55^o)$ $(0,1,0)$
R							$\mathbf{p}_{10}^{\mathcal{F}_0}$ $\hat{\mathbf{u}}_{10}^{\mathcal{F}_0}$ $\hat{\mathbf{v}}_{10}^{\mathcal{F}_{10}}$	$(-3,27.10,0)$ $(0,0,-1)$ $(1,0,0)$

Table 3: Parameters of the Schunk Anthropomorphic hand.

E5310 processors and a 4Gb RAM each one, using a threshold of $\sigma = 0.1$. Note that the cost of computing a solution increases with the number of contact constraints to be satisfied. This is because the size of the linear programs to be solved during box shrinking is proportional to the number of polytope inequalities introduced by such constraints, which increases the cost of each iteration of the algorithm (Section 4.3).

6 Conclusions

This paper has presented a new solution to the configuration problem of robotic hands. When compared to other approaches to this problem, the proposed method is always guaranteed to converge to a solution whenever one exists. An additional feature of the method is its ability to deal with general region-to-region contact constraints, as opposed to point-to-region (Borst et al. 2002; Rosell et al. 2005) or point-to-point (Gorce and Rezzoug 2005) constraints, and the possibility to define the involved regions as general Bézier patches, to better adapt the regions to the real surfaces of the hand and object considered.

The method performs sufficiently well and returns problem solutions in reasonable times. Such times are arguably large, however, so as to allow the execution of the method in real-time robotic platforms devoted to manipulation tasks. Instead, the approach is more suitable to off-line compu-

tations in the context of grasp planning, where efforts are being made to develop standard databases of graspable objects, along with corresponding sets of stable grasps for each object. Algorithms already exist that exploit such precomputed grasps to produce proper grasps of objects perceived online (Goldfeder et al. 2009).

Although the focus of the work has been on dealing with the kinematic and contact constraints inherent to the hand-object system, the approach seems to be versatile enough so as to accommodate the treatment of additional constraints arising in grasp synthesis, like stability, or dexterity constraints (Siciliano and Khatib 2008, Chapter 28). An interesting point for further research, thus, would be to explore the possibility of formulating such constraints in the form required by the method, to be able to synthesize grasp configurations satisfying all constraints simultaneously.

It is worth noting that, as defined, the configuration problem does not account for collision avoidance constraints. While such constraints might in principle be added to Eq. (26), this would considerably increase the size of the linear programs to be solved (Section 4.3), with the consequent increase in execution time. As an alternative, one can initially ignore collision constraints, and then use retraction techniques to try to eliminate object penetrations later on (Zhang and Manocha 2008), should such penetrations occur on the returned configuration. Clearly, this is another point

deserving further attention.

Finally, further work needs to be done, also, on automating the process of deciding which object regions should be placed in contact with which hand regions. While some heuristic methods have been proposed for the case in which such regions are isolated points (Woelfl and Pfeiffer 1994; Fernández et al. 2005), algorithms able to cope with general free-form regions are still to be developed.

Acknowledgements

References

Arimoto, S. (2007). *Control Theory of Multifingered Hands: A Modelling and Analytical Mechanics Approach for Dexterity and Intelligence.* Springer, 1st edition.

Barber, C., Dobkin, D., and Huhdanpaa, H. (1996). The Quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483.

Bicchi, A. (1995). On the closure properties of robotic grasping. *The International Journal of Robotics Research*, 14(4):319–334.

Bicchi, A. and Kumar, V. (2001). Robotic grasping and manipulation. In Nicosia, S., Siciliano, B., Bicchi, A., and Valigi, P., editors, *Lecture Notes in Control and Information Sciences*, volume 270, chapter 3, pages 55–74. Springer Berlin Heidelberg, Berlin, Heidelberg.

Borst, C., Fischer, M., and Hirzinger, G. (2002). Calculating hand configurations for precision and pinch grasps. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and System*, volume 2, pages 1553–1559.

Buss, M., Hashimoto, H., and Moore, J. B. (1996). Dextrous hand grasping force optimization. *IEEE Transactions on Robotics and Automation*, 12(3):406–418.

Butterfass, J., Fischer, M., Grebenstein, M., Haidacher, S., and Hirzinger, G. (2004). Design and experiences with DLR hand II. In *Proceedings of the World Automation Congress*, volume 15, pages 105–110.

Caffaz, A. and Cannata, G. (1998). The design and development of the DIST-Hand dextrous gripper. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2075–2080.

Ciocarlie, M. T. and Allen, P. K. (2009). Hand posture subspaces for dexterous robotic grasping. *The International Journal of Robotics Research*, 28(7):851–867.

Cornellà, J. and Suárez, R. (2009). Efficient determination of four-point form-closure optimal constraints of polygonal objects. *IEEE Transactions on Automation Science and Engineering*, 6(1):121–130.

Cornellà, J., Suárez, R., Carloni, R., and Melchiorri, C. (2008). Dual programming based approach for optimal grasping force distribution. *Mechatronics*, 18(7):348–356.

Dizioğlu, B. and Lakshiminarayana, K. (1984). Mechanics of form closure. *Acta Mechanica*, 52(1):107–118.

Farin, G. (2001). *Curves and Surfaces for CAGD: A Practical Guide.* Morgan Kaufmann, 5th edition.

Fernández, C., Reinoso, O., Vicente, A., and Aracil, R. (2005). Kinematic Redundancy in Robot Grasp Synthesis. An Efficient Tree-based Representation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1203–1209.

Ferrari, C. and Canny, J. (1992). Planning optimal grasps. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2290–2295.

Gazeau, J. P., Zehloul, S., Arsicault, M., and Lallemand, J. P. (2001). The LMS hand: force and position controls in the aim of the fine manipulation of objects. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2642–2648.

Giurintano, D. J., Hollister, A. M., Buford, W. L., Thompson, D. E., and Myers, L. M. (1995). A virtual five-link model of the thumb. *Medical Engineering & Physics*, 17(4):297–303.

Goldfeder, C., Ciocarlie, M., Dang, H., and Allen, P. (2009). The Columbia grasp database. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1710–1716.

Gorce, P. and Rezzoug, N. (2005). Grasping posture learning with noisy sensing information for a large scale of multifingered robotic systems: Research articles. *Journal of Robotic Systems*, 22(12):711–724.

Guan, Y. and Zhang, H. (2003). Kinematic feasibility analysis of 3-D multifingered grasps. *IEEE Transactions on Robotics and Automation*, 19(3):507–513.

Iwata, H. and Sugano, S. (2009). Design of human symbiotic robot TWENDY-ONE. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 580–586.

Kamakura, N., Matsuo, M., Ishii, H., Mitsuboshi, F., and Miura, Y. (1980). Patterns of static prehension in normal hands. *The American Journal of Occupational Therapy*, 34(7):437–445.

Kawasaki, H., Komatsu, T., and Uchiyama, K. (2002). Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand II. *IEEE/ASME Transactions on Mechatronics*, 7(3):296–303.

Kerr, J. and Roth, B. (1986). Analysis of multifingered hands. *The International Journal of Robotics Research*, 4(4):3–17.

Kumar, V. and Waldron, K. J. (1989). Suboptimal algorithms for force distribution in multifingered grippers. *IEEE Transactions on Robotics and Automation*, 5(4):491–498.

Li, Z., Hsu, P., and Sastry, S. (1989). Grasping and coordinated manipulation by a multifingered robot hand. *The International Journal of Robotics Research*, 8(4):33–50.

Lotti, F., Tiezzi, P., Vassura, G., Biagiotti, L., and Melchiorri, C. (2004). UBH 3: An anthropomorphic hand with simplified endo-skeletal structure and soft continuous fingerpads. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4736–4741.

Lovchik, C. S. and Diftler, M. A. (1999). The Robonaut hand: A dexterous robot hand for space. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 907–912.

Markenscoff, X., Ni, L., and Papadimitriou, C. H. (1990). The geometry of grasping. *The International Journal of Robotics Research*, 9(1):61–74.

Miller, A. T. and Allen, P. K. (2004). Graspit! A versatile simulator for robotic grasping. *Robotics & Automation Magazine, IEEE*, 11(4):110–122.

Napier, J. (1993). *Hands*. Princeton University Press.

Nguyen, V.-D. (1988). Constructing force- closure grasps. *The International Journal of Robotics Research*, 7(3):3–16.

Pollard, N. S. (2004). Closure and quality equivalence for efficient synthesis of grasps from examples. *The International Journal of Robotics Research*, 23(6):595–613.

Ponce, J., Sullivan, S., Sudsang, A., Boissonnat, J.-D., and Merlet, J.-P. (1997). On computing four-finger equilibrium and force-closure grasps of polyhedral objects. *The International Journal of Robotics Research*, 16(1):11–35.

Porta, J. M., Ros, L., Creemers, T., and Thomas, F. (2007). Box approximations of planar linkage configuration spaces. *Journal of Mechanical Design*, 129(4):397–405.

Porta, J. M., Ros, L., and Thomas, F. (2009). A linear relaxation technique for the position analysis of multiloop linkages. *IEEE Transactions on Robotics*, 25(2):225–239.

Roa, M. and Suárez, R. (2009). Computation of independent contact regions for grasping 3-D objects. *IEEE Transactions on Robotics*, 25(4):839–850.

Rosales, C., Porta, J. M., Suárez, R., and Ros, L. (2008). Finding all valid hand configurations for a given precision grasp. In *Proceedings of the*

IEEE International Conference on Robotics and Automation, pages 1634–1640.

Rosell, J., Sierra, X., Palomo, L., and Suárez, R. (2005). Finding grasping configurations of a dexterous hand and an industrial robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1178–1183.

Saut, J.-P., Sahbani, A., El-Khoury, S., and Perdereau, V. (2007). Dexterous manipulation planning using probabilistic roadmaps in continuous grasp subspaces. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2907–2912.

Schulz, S., Pylatiuk, C., and Bretthauer, G. (2001). A new ultralight anthropomorphic hand. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2437–2441.

Schunk GmbH & Co. KG (2006). Schunk anthropomorphic hand. http://www.schunk.com/.

Shadow Robot Company (2003). Design of a dextrous hand for advanced CLAWAR applications. In *Climbing and Walking Robots and the Supporting Technologies for Mobile Machines*, pages 691–698.

Siciliano, B. and Khatib, O. (2008). *Springer Handbook of Robotics*. Springer-Verlag New York Inc. Chapters 15, 28, 22.

Suárez, R. and Grosch, P. (2005). Mechanical hand MA-I as experimental system for grasping and manipulation. In *VideoProceedings of the IEEE International Conference on Robotics and Automation*, Barcelona.

Trinkle, J. C., Farahat, A. O., and Stiller, P. F. (1995). First-order stability cells of active multi-rigid-body systems. *IEEE Transactions on Robotics and Automation*, 11(4):545–557.

Valero-Cuevas, F. J., Johanson, M. E., and Towles, J. D. (2003). Towards a realistic biomechanical model of the thumb: the choice of kinematic description may be more critical than the solution method or the variability/uncertainty of musculoskeletal parameters. *Journal of Biomechanics*, 36(7):1019–1030.

Woelfl, K. and Pfeiffer, F. (1994). Grasp strategies for a dextrous robotic hand. In *Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems*, volume 1, pages 366–373.

Yamaguchi, Y. (1997). Bézier normal vector surface and its applications. In *Proceedings of the International Conference on Shape Modeling and Applications*, pages 26–35.

Zhang, L. and Manocha, D. (2008). An efficient retraction-based RRT planner. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3743–3750.