Robotica (2000) volume 18, pp. 51–58. Printed in the United Kingdom © 2000 Cambridge University Press

The synthesis of 3-D form-closure grasps*

Dan Ding,** Yun-Hui Liu** and Shuguo Wang†

SUMMARY

This paper presents a new formulation of computing three-dimensional (3-D) frictional form-closure grasps of n robotic fingers. As 3-D form-closure grasps involve 6-D wrench space, we first propose a recursive reduction technique to transform the complicated problem in the 6-D space into a simpler 3-D one. Next, we rewrite the sufficient and necessary condition for form-closure grasps into its equivalent form of two sets of linear inequalities. Then, according to the linear inequality theory, the problem is transformed to searching for a set of points which ensure the inconsistency of each of the two linear inequality systems. To search for such points, we proposed two methods: The first one is based on testing whether the convex region formed by each linear inequality system is empty, while the second one relies on the potential field method. We have implemented the algorithm and confirmed their efficiency for the synthesis of 3-D form-closure grasps.

KEYWORDS: Form-closure grasps; Multifingered robot hand; Recursive reduction technique; Linear inequalities.

1. INTRODUCTION

Multifingered grasping has aroused remarkable interest over the last two decades for its potential in performing dextrous and fine manipulation tasks. Although its use is still confined to laboratory research due to the overall complexity, much work has been done in this area concerning mechanical design,[1,2] kinematics,[1,3,4] grasp planning[5,6] and dextrous manipulation.[7,8] In this paper, we address the problem of computing stable grasps of polyhedral objects, a fundamental issue of grasp planning.

Form-closure and force-closure are two essential properties concerning the stability of a grasp.[1,9] Under a form-closure grasp, any external wrench applied at the grasped object can be balanced by grasp forces of the robot hand. While in the analysis of force-closure property, kinematics of the robot hand must be taken into account.[9] So far, most previous research on form-closure grasps have been devoted to the following aspects:

- Tremendous efforts have been made to test whether a given grasp is form-closure. Salisbury and Roth[1] have shown that a necessary and sufficient condition for form-closure is that the primitive contact wrenches resulted by contact forces positively span the entire wrench space. This condition is equivalent to that the origin of the wrench space lies strictly inside the convex hull of the primitive contact wrenches.[10,11] Nguyen[12] proposed a simple test algorithm for 2-finger form-closure grasps. Trinkle[13] formulated the test of form-closure as a linear programming problem whose optimal objective value measures how far a grasp is from losing the closure. Recently, the author[14] has shown that the qualitative testing can be transformed to a ray-shooting problem of a convex hull. The developed algorithm is applicable to 2-D and 3-D frictional and frictionless grasps with any number of contacts.
- Compared with qualitative test, algorithms of grasp synthesis are more important. Much of the work about grasp synthesis concerns two phases. First, to solve for optimal contact force yielding a stable grasp given fixed finger positions. Buss and Hashimoto[15] made a key observation that the non-linear friction cone constraint is equivalent to positive definiteness of a certain matrix subject to linear constraints and then formulated it as an optimization problem on the smooth manifold of linearly constrained positive definite matrices via gradient flows. Second, to compute at least one (maybe optimal) finger contact point location that ensures a form-closure grasp. Nguyen[12] extended his test algorithm to compute all 2-finger form-closure grasps on a polygonal object. Ponce and Faverjon[5] also presented an algorithm for computing all grasps satisfying their sufficient conditions. Recently, Liu[16] showed that the non-form-closure region for n-finger planar grasp of polygonal object consists of two convex polytopes in the parameter space representing grasping points. However, only little mention in literature can be found on computing 3-D form-closure grasps due to complicated geometry and high dimension of the grasp space. Ponce et al.[6] made the first work on computing 3-D force-closure grasps of polyhedral objects. They showed that the force-closure synthesis on a polyhedron can be reduced to the problem of projecting a polytope onto some linear subspace, which led to a linear programming solution to compute maximum independent grasp regions. However, it only gives the detailed algorithm for 3-D concurrent grasps. Recently, the authors[17] developed an efficient algorithm for computing all grasping points of one finger to achieve a 3-D form-closure grasp with other $n-1$ fingers given. The authors[18] further extended this algorithm to general cases, i.e. we calculated grasping points of $n-m$, fingers, provided that grasping points of m fingers are given. However, those two algorithms both involve the calculation of a 6-D convex cone via Qhull algorithm,[19] which is not efficient.

In this paper, we proposed a novel formulation of 3-D form-closure grasp based on a recursive reduction technique

* This work is supported in part by the Hong Kong Research Grant Council under grants CUHK4151/97E and CUHK4166/98E.

** Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, Shatin, New Territories (Hong Kong)

† The Robotics Institute, Harbin Institute of Technology, Harbin (China)

https://doi.org/10.1017/S026357479900243X Published online by Cambridge University Press

and the theory of linear inequality system. Using the reduction technique, we recursively slice the convex hull of the primitive contact wrench in the 6-D wrench space by planes passing the point of origin and transforming the problem in the 6-D wrench space into one in a 3-D space. To ensure a form-closure grasp, all the 3-D points resulted from the reduction procedure have to envelop the neighbourhood of the origin. Then we rewrote this condition into its equivalent form of two linear inequality systems. We further showed that searching for form-closure grasping points is equivalent to searching for some points to make each of the two linear inequality systems inconsistent. Finally, two methods were developed to solve the searching problem. The first method involved an emptiness check of the convex region formed by each linear inequality system. The second method derived a non-form-closure condition and transformed the problem into a path-finding one while avoiding the intersection with either convex region and approaching a goal defined by a performance index.

The paper is organized as follows: Form-closure property and relevant issues are reviewed in Section 2. Section 3 describes the synthesis algorithm of 3-D form-closure grasp. Section 3.1 discusses the representation of grasping points. Section 3.2 gives the details of the recursive reduction technique. Section 3.3 describes the equivalent form of form-closure condition based on the theory of linear inequality system. In section 3.4 we first define the performance index and then describe the two methods. Section 4 implements the synthesis algorithm and examines its performance by numerical experiments. Section 5 concludes the paper.

2. FORM-CLOSURE GRASPS.

In order to discuss the 3-D form-closure grasps, we make the following assumptions:

- The object to be grasped is a polyhedron.
- A point contact with friction model is adopted.
- Kinematics and dynamical forces of the grasp are negligible.

Assume that the Coulomb friction with friction coefficient μ exists at the contact point. To ensure non-slipping at the contact point of finger i, the grasping force f_i must satisfy

$$f_{ix}^2+f_{iy}^2\leq\mu^2 f_{iz}^2 \tag{1}$$

where (f_{ix}, f_{iy}, f_{iz}) denotes x, y and z components of the grasp force f_i w.r.t. the object coordinate frame. The non-linear constraints in (1) geometrically represent a cone called friction cone (Figure 1).

To simplify the problem, we linearize the friction cone by a polyhedral convex cone with m sides. Under this approximation, the grasp force f_i can be represented as

$$f_i=\sum_{j=1}^{m}\alpha_{ij}\alpha_{ij}\vec{s}_{ij},\quad \alpha_{ij}\geq 0 \tag{2}$$

where $\vec{s}_{ij}$ represents the j-th edge vector of the polyhedral convex cone. Coefficients α_{ij} are non-negative constants. The force and torque, corresponding to the grasping force f_i applied at the center of mass of the object is given by

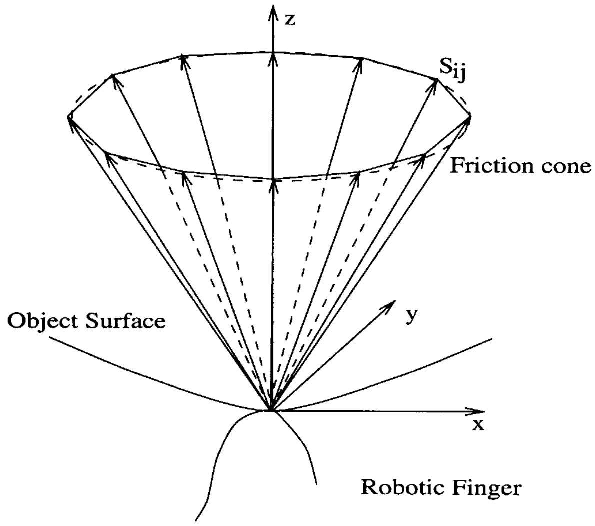

Fig. 1. The friction cone at a grasping point.

$$\underline{w}_i=\begin{pmatrix} f_i \\ \tau_i \end{pmatrix}=\begin{pmatrix} f_i \\ r_i\times f_i \end{pmatrix} \tag{3}$$

where r_i denotes the position vector of the i-th grasping point w.r.t. the object coordinate frame originated at the center of mass. Here we refer to the force f_i and moment τ_i pair as a wrench $\underline{w}_i$ Substituting eq. (2) into eq. (3) derives

$$\underline{w}_i=\sum_{j=1}^{m}\alpha_{ij}w_{ij} \tag{4}$$

where

$$w_{ij}=\begin{pmatrix} \vec{s}_{ij} \\ r_i\times\vec{s}_{ij} \end{pmatrix}$$

w_{ij} is called *primitive contact wrenches* of the finger. The net wrench applied at the object by the fingers is

$$\underline{w}_{net}=\sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_{ij}w_{ij}=W\alpha \tag{5}$$

where W and α are given by

$$W=(w_{11}, w_{12}, \ldots, w_{1m}, \ldots, w_{n1}, w_{n2}, \ldots, w_{nm})$$

$$\alpha=(\alpha_{11}, \alpha_{12}, \ldots, \alpha_{1m}, \ldots, \alpha_{n1}, \alpha_{n2}, \ldots, \alpha_{nm})^T$$

W is a $6\times nm$ matrix called *wrench matrix* and its column vectors are the primitive contact wrenches. For convenience, in the following we use w_i with a single subscript i, instead of w_{ij} to denote the i-th column vector of grasp matrix W, and use α_i to represent the i-th component of vector α.

Definition 1. *Suppose that an n-finger frictional grasp is given. For any external wrench* $\underline{w}_{ext}$ *applied at the object, if it is always possible to find an* α *with all* $\alpha_i\geq 0$ *such that*

$$W\alpha + \underline{w}_{ext} = 0,$$

the grasp is said to be form-closure.

It is well-known that a form-closure grasp is equivalent to a situation when the point of origin of the wrench space R^6 lies exactly inside the convex hull of the primitive contact wrenches w_i.

In order to properly balance any external disturbance (force and moment) applied on the object, it is of great significance to develop an efficient algorithm to calculate form-closure grasps. We address the following problem:

Problem 1. *Suppose that n-finger robot hand grasped a polyhedral object. Find the grasping points of the n fingers on the surface of the object such that the n-finger grasp* $\{g_1, g_2, \ldots, g_n\}$ *is form-closure.*

Since two surface coordinates are essential to represent fingertip positions on the surface of an object, the problem of computing a form-closure grasp, in general, is to find a solution in R^{2n}; n is the number of faces where the fingertip positions are to be located. However, during the process of grasp planning, it is often the case that some of the fingertips are guaranteed to slide along the object boundary towards locations admitting form-closure grasps while some others are fixed in position. Such a case would be the simple version of our problem.

3. SYNTHESIS OF FORM-CLOSURE GRASPS

3.1. Representing a grasping point.

The surface of a polyhedral object is not smooth at its edges and vertices, hence different coordinates must be introduced to represent the grasping point on different faces. To represent the grasping point of the finger i on a face, a local coordinate frame $\{\lambda_1^i, \lambda_2^i, \lambda_3^i\}$ is attached to the face (Figure 2). The origin of the coordinate frame is located at one vertex and the λ_3^i axis is parallel to the normal of the face. The other axes are defined according to the right-hand rule. The grasping point is represented by a local coordinate $(\lambda_1^i, \lambda_2^i)$. The coordinates of the grasping point g_i w.r.t. the object frame are calculated by

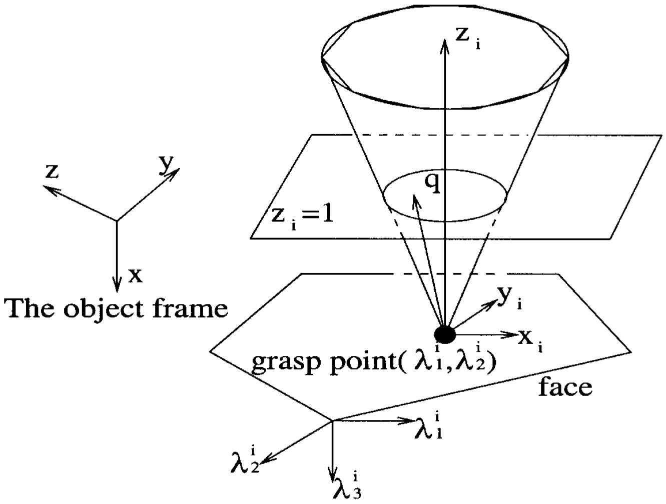

Fig. 2. The representation of a grasping point on an object face.

$$g_i = o_\lambda^i + R_\lambda^i \begin{pmatrix} \lambda_1^i \\ \lambda_1^i \\ 0 \end{pmatrix}$$

where o_λ^i and R_λ^i denote the origin and the rotation matrix of the local frame $\{\lambda_1^i, \lambda_2^i, \lambda_3^i\}$ w.r.t. the object frame, respectively. The components of the vector g_i are all affine in λ_1^i and λ_2^i

At the grasping point, another frame $\{x_i, y_i, z_i\}$ is introduced to represent the grasping force. Slice the friction cone by $z_i = 1$. The grasping force f_i w.r.t. the object frame is given by

$$f_{ij} = \beta_{ij} R_i \begin{pmatrix} x_{ij} \\ y_{ij} \\ 1 \end{pmatrix} \qquad j = 1, 2, \ldots, m$$

where R_i is the rotation matrix of the local frame, m is the number of sides of i-th polyhedral convex cone.

The resultant wrench on the object by the grasping force f_i is

$$\underline{w}_{ij} = \beta_{ij} \begin{pmatrix} R_i \begin{pmatrix} x_{ij} \\ y_{ij} \\ 1 \end{pmatrix} \\ g(\lambda_1^i, \lambda_2^i) - \left(R_i \begin{pmatrix} x_{ij} \\ y_{ij} \\ 1 \end{pmatrix}\right) \end{pmatrix} j = 1, 2, \ldots, m \qquad (6)$$

where vector q_{ii} is called q_{ii} the *primitive grasping wrench* of the finger i. It should be noted that the force vector in q_{ii} is constant while the moment vector is affine in λ_1^i, and λ_2^i

3.2. Recursive reduction technique

Theorem 1. *A convex hull $H(N)$ of N points in $R^d(x_1, x_2, \ldots, x_d)$ contains the origin point if and only if there is such an i that,*

(1) *in the given N points there are points with strictly positive x_i-coordinates and points with strictly negative x_i-coordinates as well;*

(2) *the intersection of the convex hull $H(N)$ with the hyperplane $x_i = 0$ contains the origin of the R^{d-1} defined by $x_i = 0$;*

The proof of this theorem is referred to in reference 15. The recursive reduction technique needs to calculate the slice of a convex hull by a hyperplane. Denote intersection points of the hyperplane with edges of the convex hull by set V. The slice is the convex hull $H(V)$ of points in set V. The following fact is well-known in Computational Geometry.

Proposition 1. *For any N points in R^d, the vertices of their convex hull belong to the N points and the edges belong to segments connecting them.*

From Proposition 1, we clearly obtain

Proposition 2. *For N points in R^d, denote intersection points of the segments connecting them with the hyperplane $x_i = 0$ by set E. The convex hull of points in set E is the intersection of the convex hull H(N) with the hyperplane.*

From Proposition 2, instead of explicitly calculating intersection of the convex hull with a hyperplane, we calculate the intersections of a hyperplane with segments connecting the points. Since the force vector (the first three coordinates) in q_{ij} is constant, we can reduce the dimension by successively calculating intersections between segments connecting the primitive contact wrenches and plane $x_1=0$ then plane $x_2=0$ and then plane $x_3=0$.

Algorithm 1.

- Step 1: Compute all the primitive contact wrenches w_{ij} in the representation of the parametric variable $(\lambda_1^i, \lambda_2^i)$, and denote them by set E.
- Step 2: According to signs of their x_1 coordinates, divide the primitive contact wrenches into groups E_+, E_o, E_-, which contain points with positive, zero, negative x_1 coordinates, respectively. If either E_+ or E_- is null, the form-closure grasp cannot be found and then the algorithm ends.
- Step 3: Calculate the intersections e_{new} of the plane $x_1=0$ with segments connecting points in set E_+ to those in E_-. Update the set E by all the intersection points and points in set E_0
- Step 4: According to signs of their x_2 coordinates, follow the same procedure as steps 2 and 3.
- Step 5: According to signs of their x_3 coordinates, follow the same procedure as steps 2 and 3.
- Step 6: The algorithm ends.

In this way, we finally obtain some new points in R^3 space whose coordinates are affine in λ_1^i, and λ_2^i. Thus the original problem in R^6 space is effectively simplified. However, the negative effect of the recursive reduction technique is the increase of the number of points to be processed.

3.3. *Equivalent form of form-closure condition*

Through the recursive reduction technique, we obtain some 3-D points resulting from 6-D primitive contact wrenches. As mentioned previously, to ensure a form-closure grasp, the convex hull of these 3-D points must contain the origin. In other words, a given system of wrenches achieves form-closure when the equation

$$\sum_{i=1}^{N} \alpha_i w_i(\lambda_1^j, \lambda_2^j) = 0 \quad \alpha_i > 0 \tag{7}$$

admits a non-trivial, strictly positive solution. N is the number of newly generated 3-D points.

We note that the equation above is equivalent to the following form:

$$\exists \zeta_i > 0, \quad \sum_{i=1}^{N} \zeta_i w_{ik}(\lambda_1^j, \lambda_2^j) = 0 \quad k = 1, 2 \tag{8}$$

$$\sum_{i=1}^{N} \zeta_i w_{i3}(\lambda_1^j, \lambda_2^j) \leq 0 \tag{9}$$

and

$$\exists \gamma_i > 0, \quad \sum_{i=1}^{N} \gamma_i w_{ik}(\lambda_1^j, \lambda_2^j) = 0 \quad k = 1, 2 \tag{10}$$

$$\sum_{i=1}^{N} \gamma_i w_{i3}(\lambda_1^j, \lambda_2^j) \leq 0 \tag{11}$$

Theorem 2. *The system of inequalities*

$$A(\lambda_1^j, \lambda_2^j) z < d(\lambda_1^j, \lambda_2^j)$$

is inconsistent if and only if there exists $\alpha > 0$ such that

$$\alpha^T A(\lambda_1^j, \lambda_2^j) = 0^T \tag{12}$$

$$\alpha^T d(\lambda_1^j, \lambda_2^j) \leq 0 \tag{13}$$

where α is an $n \times 1$ vector, A is an $n \times 2$ matrix, d is a $1 \times n$ vector.

The proof of this theorem can be seen in reference [15]. Here

$$A(\lambda_1^j, \lambda_2^j) \stackrel{\text{def}}{=} \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ \cdots & \cdots \\ w_{n1} & w_{n2} \end{pmatrix}$$

For the inequality system formed by (8) and (9)

$$d(\lambda_1^j, \lambda_2^j) \stackrel{\text{def}}{=} (w_{13}, w_{23}, \ldots, w_{n3})^T$$

For the inequality system formed by (10) and (11)

$$d(\lambda_1^j, \lambda_2^j) \stackrel{\text{def}}{=} (-w_{13}, -w_{23}, \ldots, -w_{n3})^T$$

From the sufficient condition of Theorem 2, the problem can be transformed into the inconsistency of each of the two linear inequality systems.

$$A(\lambda_1^j, \lambda_2^j) z < d(\lambda_1^j, \lambda_2^j) \tag{14}$$

$$A(\lambda_1^j, \lambda_2^j) z < -d(\lambda_1^j, \lambda_2^j) \tag{15}$$

Proposition 3. *On the face of the object, the grasping points $\{g_1, g_2, \ldots, g_n\}$ that result in an n-finger form-closure grasp are a set of parameters $(\lambda_1^1, \lambda_2^1, \ldots, \lambda_1^n, \lambda_2^n)$ that guarantee the inconsistency of both linear inequality systems* (14) *and* (15). *That is,*

$$\{(\lambda_1^1, \lambda_2^1, \ldots, \lambda_1^n, \lambda_2^n) \mid$$

$$A(\lambda_1^1, \lambda_2^1, \ldots, \lambda_1^n, \lambda_2^n) z < d(\lambda_1^1, \lambda_2^1, \ldots, \lambda_1^n, \lambda_2^n) \text{ inconsistent and}$$

$$A(\lambda_1^1, \lambda_2^1, \ldots, \lambda_1^n, \lambda_2^n) z < -d(\lambda_1^1, \lambda_2^1, \ldots, \lambda_1^n, \lambda_2^n) \text{ inconsistent}\}$$

https://doi.org/10.1017/S026357479900243X Published online by Cambridge University Press

Note that a geometric interpretation of Proposition 3 is that the convex regions bounded by inequality systems (14) and (15) are empty respectively.

Proposition 4. *An n-finger grasp* $(\lambda_1^1, \lambda_2^1, \ldots, \lambda_1^n, \lambda_2^n)$ *is non-form-closure if and only if either inequality system* (14) *or* (15) *is consistent. That is, the non-form-closure grasps are the following set of points*:

$$\{(\lambda_1^1, \lambda_2^1, \ldots, \lambda_1^n, \lambda_2^n) |$$
$$A(\lambda_1^1, \lambda_2^1, \ldots, \lambda_1^n, \lambda_2^n) z \le d(\lambda_1^1, \lambda_2^1, \ldots, \lambda_1^n, \lambda_2^n) \text{ consistent or}$$
$$A(\lambda_1^1, \lambda_2^1, \ldots, \lambda_1^n, \lambda_2^n) z \le -d(\lambda_1^1, \lambda_2^1, \ldots, \lambda_1^n, \lambda_2^n) \text{ consistent}\}$$

Here z is a 2×1 vector. Note that since matrix A and vector d contain the unknown variable $(\lambda_1^1, \lambda_2^1, \ldots, \lambda_1^n, \lambda_2^n)$, hence the convex regions in the 2-D space formed by the two inequality systems are not fixed. Therefore, we perform the following transformation:

$$y \stackrel{\text{def}}{=} (\lambda^{\mathrm{T}}, z_1, z_2, \lambda^T z_1, \lambda^T z_2)^T$$

where $\lambda = (\lambda_1^1, \lambda_2^1, \ldots, \lambda_1^n, \lambda_2^n)^T$ Then for each inequality system, we obtain a new form

$$C_y \le k$$

Here matrix C and vector k are constant and y is a $(6n+2)\times 1$ vector which defines a new state space. Thus in the new state space we obtain two fixed convex polytopes. Thus the geometrical meaning of Proposition 4 is that once a set of new state variables is inside either fixed convex polytope, the grasp $(\lambda_1^1, \lambda_2^1, \ldots, \lambda_1^n, \lambda_2^n)$ is non-form-closure.

Based on Proposition 3 and Proposition 4, we proposed two methods to solve the problem of searching for a set of parameter $(\lambda_1^1, \lambda_2^1, \ldots, \lambda_1^n, \lambda_2^n)$ such that a form-closure grasp can be obtained.

3.4. *Determination of fingertip positions*

3.4.1. Definition of performance index. We have found empirically that there is not, in general, a unique solution to the form-closure problem. Thus it is an efficient and practical way to define a criterion to uniquely determine the fingertip position yielding a form-closure grasp. We note that many grasp metrics concerning the contact forces have been presented in published papers; however, few quality index relate to fingertip positions. In this section, a criterion similar to the one proposed by Ponce is adopted. We try to locate the fingertips in such a position that we can center as well as possible the center of mass of the object. This enables us to decrease the effect of gravitational and inertial forces during the motion of the robot.

In detail, the criterion introduced measures the L^2 distance between the center of mass $O_p = (x_p, y_p, z_p)$ of the grasped object and the center $O_d = (x_d, y_d, z_d)$ of the contacts corresponding to the grasping parameters λ_1^j, and λ_2^j

$$u = (x_d - x_p)^2 + (y_d - y_p)^2 + (z_d - z_p)^2$$

3.4.2. Method 1: Emptyness check of convex regions. Suppose we have a set of constraints as follows:

$$H_i = \{A_i^T x < b_i\} \quad i = 1, 2, \ldots, n.$$

Given a fixed vector v, we discuss the relation between the vector and the hyperplanes formed by the linear constraints. We define P_+, P_0, P_- to be the set of the subscripts of linear constraints that the scalar product of v, and the norm vector of the hyperplane of the corresponding linear constraint is greater than, equal to, or less that 0, respectively. Then three conditions are given to reduce the number of constraints to be processed. M denotes the convex space formed by all the linear constraints.

(i) if $P_+ = \emptyset$ or $P_- = \emptyset$, and $P_0 = \emptyset$, then $M \ne \emptyset$.
(ii) if one and only one set of P_+ and P_- are empty, and $P_0 \ne \emptyset$, then $M \ne \emptyset$ if and only if $\{x | A_i^T x < b_i, i \in P_0\} \ne \emptyset$,
(iii) if $M \ne \emptyset$, $P_- \ne \emptyset$, then there is a subscript $i_0 \in P_+$, such that $A_{i_0 1} x_1 + A_{i_0 2} x_2 + \ldots + A_{i_0 n} x_n < b_{i_0}$ is not redundant, that is irremovable in the decision whether M is empty. And if $M \ne \emptyset$, $P_+ \ne \emptyset$, then there is a subscript $i_1 \in P_-$, such that $A_{i_1 1} x_1 + A_{i_1 2} x_2 + \ldots + A_{i_1 n} x_n < b_{i_1}$ is not redundant.

Algorithm 2.

- Step 1: Set the initial value of $(\lambda_1^j, \lambda_2^j)$ so that the objective function achieves minimum 0.
- Step 2: Using Algorithm 2 to determine whether the two linear inequality systems are both inconsistent. If the condition is met, the algorithm ends.
- Step 3: Recursively modify the values of $(\lambda_1^j, \lambda_2^j)$ according to a heuristic search strategy until both convex regions are proved to be empty.

To update the value of $(\lambda_1^j, \lambda_2^j)$, we extend in $2^r \times 8$ directions with different radius on a face. Here $r = 0, 1, \ldots p$ and p is the number of layers we need to search. Suppose each face F_j where the fingertip positions are to be determined is bounded by n_i edges, the parameters λ_1^j and λ_2^j resulting from the searching procedure must also satisfy n_i linear constraints $f_{ji}(\lambda_1^j, \lambda_2^j) \le 0$ with $i = 1, 2, \ldots, n_i$.

3.4.3. Method 2: Potential field method. From Proposition 4, when a set of parameters $(\lambda_1^1, \lambda_2^1, \ldots, \lambda_1^n, \lambda_2^n)$ enables either the inequality system (14) or (15) to be consistent, the grasp obtained will be a non-form-closure. Through the transformation described in Section 3.3, we obtain two fixed convex polytopes in the new state space, which represent the non-form-closure region. Our aim is to find form-closure grasps, that is, we need to find a set of parameters $(\lambda_1^1, \lambda_2^1, \ldots, \lambda_1^n, \lambda_2^n)$ which corresponds to a point outside the two convex polytopes.

Here we employ the potential field method by considering the two convex polytopes as obstacles. We can set the desired direction according to the performance index. In order to make the point to be attracted towards that direction while being repulsed from the obstacles, the field of artificial forces $F(y)$ is introduced to denote the most promising direction in every iteration.

$$F(y) = -\nabla U(y) = \begin{pmatrix} \partial U/\partial y_1 \\ \partial U/\partial y_2 \\ \ldots \\ \partial U/\partial y_n \end{pmatrix}$$

Here $U(y)$ is the potential function composed of the attractive potential U_{att} and repulsive potential U_{rep} associated with the obstacle regions.

$$U(y) = U_{att}(y) + U_{rep}\,(y)$$

Here we define $U_{att}(y)$ using our performance index described in Section 3.4.1, and the repulsive function is defined as follows:

$$U_{rep}(y) = \begin{cases} 1/2\eta(1/\rho(y) - 1/\rho_0)^2 & \text{if } \rho(y) \le \rho_0, \\ 0 & \text{if } \rho(y) > \rho_0, \end{cases}$$

where η is positive scaling factors and $\rho(0)$ is a positive constant denoting the allowable distance to the obstacle. $\rho(y)$ denotes the distance from q to the obstacle polytopes, i.e.:

$$\rho(q) = \min\|y - y_i\|$$

The potential field method is an iterative one and in each step $-\nabla U(q)$, i.e. the gradient vector denotes the most promising direction along which we can update the previous parameter.

Note that the initial point we set lies inside the obstacle polytopes. In this case, we take a two-step measure. First, we find the shortest distance from the initial point to the boundary edges of the convex polytopes and along that direction we can move the point out of the obstacle regions quickly. Second, once we obtain a new point outside the obstacle region, we can turn to the potential field method described above.

4. IMPLEMENTATION

We have implemented the proposed algorithm on a Sun Ultra 5 workstation using the $C++$ programing language and verified its computation efficiency by two examples. Here we used method 1.

The first example concerns a four-finger grasp of a polyhedral object shown in Figure 3. Three fingertip positions are already fixed and the normal vectors nrm_i of the object at the grasp points are known.

In this example, parameters (λ_1, λ_2) concerning finger 4 on the left face of the polyhedron are calculated. The face is a 4×4 area. According to the center of mass of the object, that is, the center of the polyhedron in this example, we set the object function as follows:

$$u = (\lambda_1 - 2)^2 + (\lambda_2 - 2)^2$$

It's easy to find that the initial candidate for (λ_1, λ_2) should both be 2 to ensure that the object function achieves a minimum. In other words, we choose (2, 2) as the start point for searching the pair yielding form-closure grasps.

Here each friction cone is linearized by 8 segments and the friction coefficient $\mu = 0.3$. The normal vectors nrm_i of the object are given as follows:

$$nrm_1 = \begin{pmatrix} 0.0 \\ 0.0 \\ 1.0 \end{pmatrix}, \quad nrm_2 = \begin{pmatrix} 0.0 \\ 1.0 \\ 0.0 \end{pmatrix}$$

$$nrm_3 = \begin{pmatrix} 0.0 \\ 0.0 \\ 1.0 \end{pmatrix}, \quad nrm_4 = \begin{pmatrix} 0.0 \\ -1.0 \\ 0.0 \end{pmatrix}$$

Two groups of fixed positions are given to show the efficency of our algorithm.

Case I:

$$r_1 = \begin{pmatrix} 2.0 \\ 0.0 \\ 0.0 \end{pmatrix}, \quad r_2 = \begin{pmatrix} 0.0 \\ 1.5 \\ 0.0 \end{pmatrix}, \quad r_3 = \begin{pmatrix} 0.0 \\ 0.0 \\ 2.0 \end{pmatrix}$$

In this case, since the initial candidate of (λ_1, λ_2) is just the point we are searching, the algorithm ends quickly using only 0.362 sec. (Figure 4).

Case II:

$$r_1 = \begin{pmatrix} 2.0 \\ 1.5 \\ 1.5 \end{pmatrix}, \quad r_2 = \begin{pmatrix} 0.2 \\ 2.0 \\ -1.5 \end{pmatrix}, \quad r_3 = \begin{pmatrix} -0.5 \\ -0.5 \\ 2.0 \end{pmatrix}$$

In this case, the initial candidate of (λ_1, λ_2) cannot meet the form-closure condition, so the algorithm finds the right point $\lambda_1 = 2.0$ and $\lambda_2 = 1.5$ after 3 loops using 2.663 sec. (Figure 5).

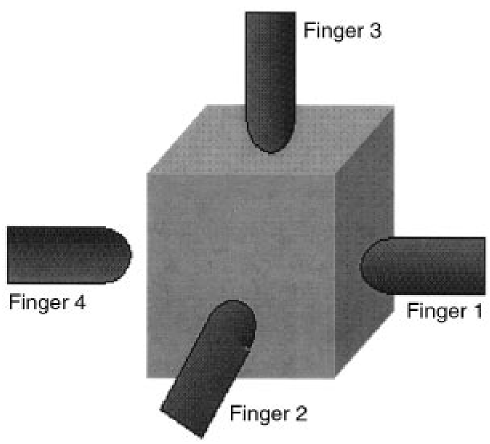

Fig. 3. Example 1.

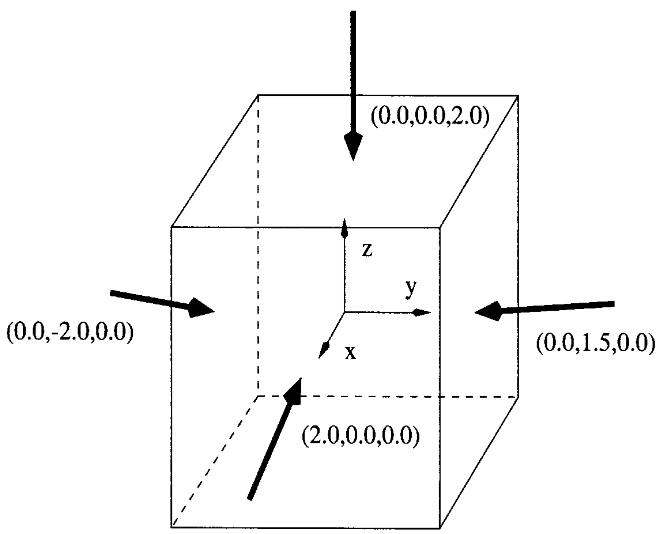

Fig. 4. Form-closure grasp found in Case I.

https://doi.org/10.1017/S026357479900243X Published online by Cambridge University Press

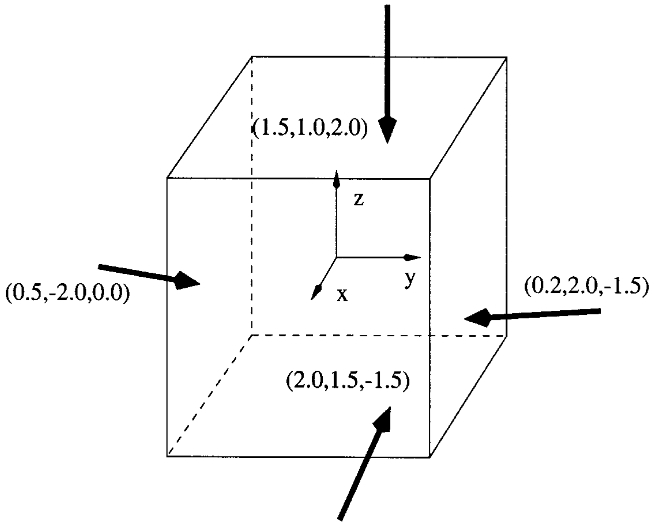

Fig. 5. Form-closure grasp found in Case II.

In the second example, we consider a four finger grasp of a tetrahedron which is not such a regular geometry as the cube in the first example (Figure 6).

Here each friction cone is linearized by 8 segments and the friction coefficient $\mu = 0.6$. The normal vectors nrm_i of the object are given as follows:

$$nrm_1 = \begin{pmatrix} 0.0 \\ 0.0 \\ 1.0 \end{pmatrix}, \quad nrm_2 = \begin{pmatrix} 1.0 \\ 0.0 \\ 0.0 \end{pmatrix}$$

$$nrm_3 = \begin{pmatrix} 1.0 \\ 1.0 \\ 1.0 \end{pmatrix}, \quad nrm_4 = \begin{pmatrix} 0.0 \\ -1.0 \\ 0.0 \end{pmatrix}$$

In this example, we set the gravity center of the tetrahedron as the origin point and the object function correspondingly becomes $u = (\lambda_1 - 3)^2 + (\lambda_2 - 4)^2$. So the initial candidate for λ_1 and λ_2 is (3, 4).

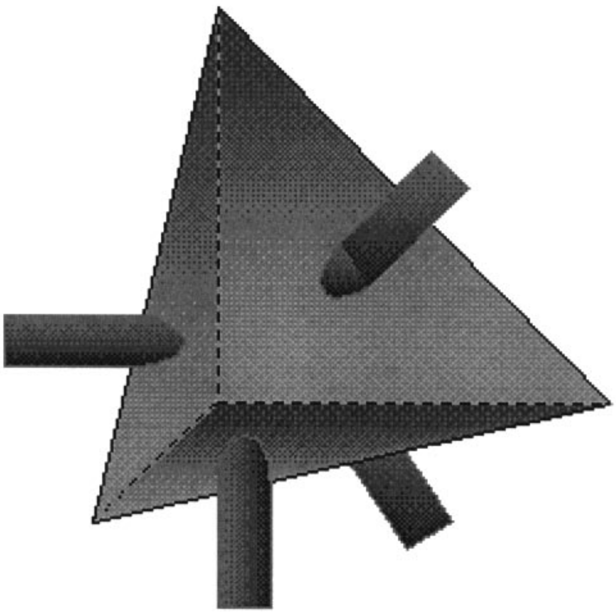

Fig. 6. Example 2.

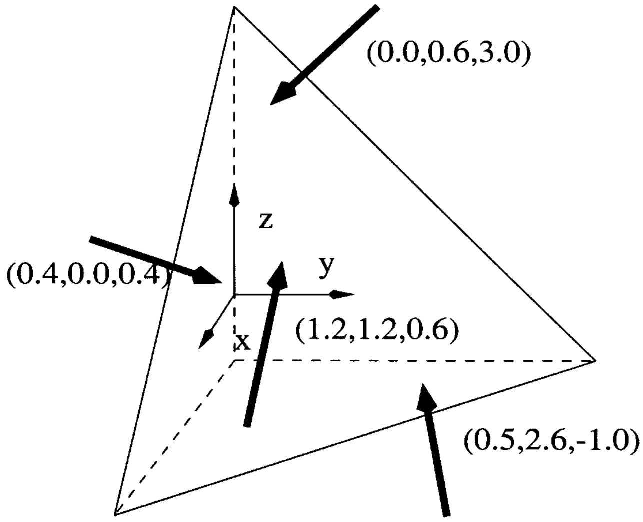

Fig. 7. Form-closure grasp found in Case I.

Case I:

$$r_1 = \begin{pmatrix} 0.5 \\ 2.6 \\ -1.0 \end{pmatrix}, \quad r_2 = \begin{pmatrix} 0.0 \\ 0.6 \\ 3.0 \end{pmatrix}, \quad r_3 = \begin{pmatrix} 1.2 \\ 1.2 \\ 0.6 \end{pmatrix}$$

In this case, the algorithm ends with $\lambda_1 = 2.6$ and $\lambda_2 = 3.6$ after two tries using 1.102 sec. The corrosponding position for finger 4 is (0.4,0.0,0.4). (Figure 7)

Case II:

$$r_1 = \begin{pmatrix} 1.1 \\ 2.0 \\ -1.0 \end{pmatrix}, \quad r_3 = \begin{pmatrix} 1.4 \\ 1.6 \\ 0.2 \end{pmatrix}, \quad r_4 = \begin{pmatrix} 2.0 \\ 0.0 \\ 0.0 \end{pmatrix}$$

In this case, we made r_4 fixed and calculate the fingertip position in Face 2. At last we got $\lambda_1 = 1.0$ and $\lambda_2 = 1.5$ after 1 loop using 0.923 sec. The corresponding position for finger 2 is (0.0, 1.0, 0.5). (Figure 8)

5. CONCLUSION

This paper presents a new formulation of 3-D form-closure grasp. Through the recursive reduction technique and based

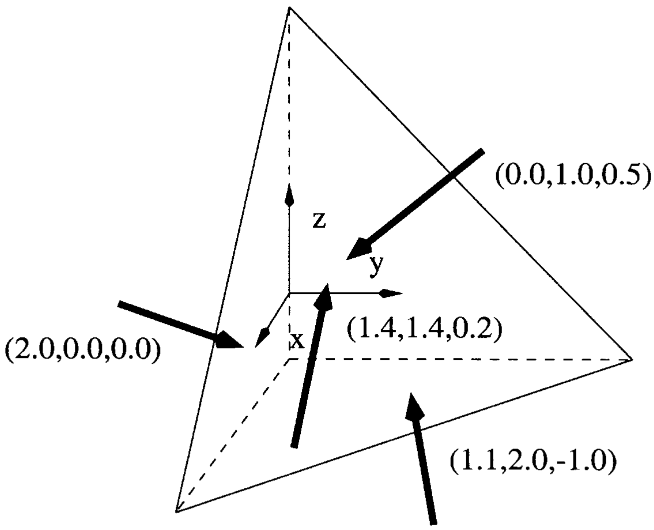

Fig. 8. Form-closure grasp found in Case II.

https://doi.org/10.1017/S026357479900243X Published online by Cambridge University Press

on the theory of linear inequality system, we successfully transformed the complicated problem in the 6-D wrench space to the inconsistency of each of the two 2-D linear inequality systems. Furthermore, we proposed two methods to solve the problem of searching proper grasping parameters based on the two propositions derived. Finally we have confirmed the efficiency of the proposed algorithm by two numerical experiments.

References

1. J.K. Salisbury and B. Roth, "Kinematic and force analysis of articulated hands", *ASME Journal of Mechanism, Transmissions, Automation, Design* **105**, 33–41 (1982).
2. M.R. Cutkosky, "On grasp choice, grasp model, and the design of hands for manufacturing task", *IEEE Trans. Robot. Automat*. **5**, 269–279 (June, 1989).
3. D.J. Montana, "The kinematics of contact and grasp", *Int. J. Robot. Res*. **7**, 3, 17–32 (1988).
4. H. Li, J.C. Trinkle and Z. Li, "Grasp Analysis as Linear Matrix Inequality Problems", *Proc. IEEE Int. Conf. Robot. Automat*. (1999) 1261–1268.
5. J. Ponce and B. Faverjon, "On computing three finger force-closure grasp of polygonal objects", *IEEE Trans. on Robotics and Automation* **11**, 6, 868–881 (1995).
6. J. Ponce, S. Sullivan, A. Sudsang, J.-D. Boissonnat and J.-P. Merlet, "On computing four-finger equilibrium and force-closure grasps of polyhedral objects", *Int. J. Rob. Res*. **16**(1), 11–35 (1997).
7. H. Zhang, K. Tanie and H. Maekawa, "Dextrous manipulation planning by graph transformation", *Proc. IEEE Int. Conf. Robot. Automat*. (1996) pp. 3055–3060.
8. L.A. Munoz, C. Bard and J. Najera, "Dextrous manipulation: A geometrical reasoning point of view", *Proc. IEEE Int. Conf. Robot. Automat*. (1995) pp. 458–463.
9. A. Bicchi, "On the closure properties of robotics grasping", *Int. Robotics Research* **14**, 4, 319–334 (1995).
10. B. Mishra, J.T. Schwartz and M. Sharir, "On the existence and synthesis of multifinger positive grips", *Algorithmica* (Special Issue: Robotics), **2**, 4, 541–558 (1987).
11. R.M. Murray, Z. Li and S.S. Sastry, *A Mathematical Introduction to Robotics Manipulation* (Orlando, FL: CRC, 1994) pp. 211–257.
12. V.-D. Nguyen, "Constructing force-closure grasps", *Int. J. Rob. Res.* **7**, 3, 3–16 (1988).
13. J.C. Trinkle, "A quantitative test for form closure grasps", *IROS* (1992) pp. 1670–1677.
14. Y.H. Liu, "Qualitative test and force optimization of 3-D frictional form-closure grasps using linear programming", *IEEE Trans. on Robotics and Automation* **15**, 1, 163–173 (1999).
15. M. Buss, H. Hashimoto and J.B. Moore, "Dextrous hand grasping force optimization", *IEEE Trans. Robot. Automat*. **12**, 406–418 (June, 1996).
16. Y.H. Liu, "Computing *n*-finger force-closure grasps on polygonal objects", *Proc. IEEE Int. Conf. on Robotics and Automation*, (1998) pp. 2734–2739.
17. Y.H. Liu, D. Ding, and S. Wang, "Constructing 3-D frictional form-closure grasps of polyhedral objects", *Proc. IEEE Int. Conf. on Robotics and Automation* (1999) pp. 1897–1903.
18. Y.H. Liu and D. Ding, "Towards construction of 3-D frictional form-closure grasps: a formulation", *IROS* (1999) pp. 279–284.
19. C.B. Barber, D.P. Dobkin and H. Huhdanpaa, "The quickhull algorithm for convex hulls", *Technical Report GCC53* (The Geometry Center, Minnesota, U.S.A., 1993).