A Fast Procedure for Optimizing Dynamic Force Distribution in Multifingered Grasping

Yu Zheng and Wen-Han Qian

***Abstract*—This correspondence deals with the dynamic force distribution (DFD) problem, i.e., computing the contact forces to equilibrate a dynamic external wrench on the grasped object. The sum of the normal force components is minimized for enhancing safety and saving energy. By this optimality criterion, the DFD problem can be transformed into a linear programming (LP) problem. Its objective function is the inner product of the dynamic external wrench and a vector, and the constraints on the vector, given by a set of linear inequalities, define a polytope. The solution to the LP problem can always be attained at the vertex of the polytope called the solution vertex. We notice that the polytope is determined by the grasp configuration. Along with the direction change of the dynamic external wrench, only the solution vertex moves to an adjacent vertex sequentially, whereas the polytope with all its vertices remains unchanged. Therefore, the polytope and the adjacencies of each vertex can be computed in the offline phase. Then, in the online phase, simply search the adjacencies of the old solution vertex for the new one. Without lost of optimality, such a DFD algorithm runs a thousandfold faster than solving the LP problem by the simplex method in real time.**

***Index Terms*—Dexterous robot hand, duality, dynamic force distribution (DFD), multifingered grasping, optimal contact force.**

I. Introduction

Multifingered robot hands have been explored with great enthusiasm for over two decades for their capability to manipulate objects dexterously. A key issue in this area is dynamic force distribution (DFD) for finding the *optimal* contact forces *in real time* to equilibrate a *dynamic* external wrench on the grasped object subject to contact constraints. Since Salisbury and Roth [1] decomposed the contact force into a manipulation force and an internal force, DFD algorithms have appeared one after another [2]–[22]. They can be classified into two categories.

In the first category, the algorithms pursue optimal contact forces possibly fast. Kerr and Roth [2] proposed a linear programming (LP) algorithm based on linearized friction constraints. Cheng and Orin [3]–[5] developed a compact dual LP algorithm. Nahon and Angeles [6], [7] presented a quadratic programming (QP) algorithm. Based on a Lagrange multiplier method, Nakamura *et al.* [8] gave a nonlinear programming (NLP) algorithm. By transforming the contact constraints into the positive definiteness of a linearly constrained matrix, Buss *et al.* [9] formulated the problem of grasping force optimization as an optimization problem on the smooth manifold of linearly constrained symmetric positive definite matrices, for which efficient gradient flow algorithms [9]–[11] were raised. Han *et al.* [12] further cast the contact constraints into linear matrix inequalities (LMIs) and brought forth an interior point algorithm. Helmke *et al.* [13] suggested a Newton algorithm. Liu *et al.* [14], [15] demonstrated that some of the above algorithms [9]–[13] are quadratically convergent. These efforts contribute to the mainstream of DFD.

Manuscript received January 8, 2006; revised March 29, 2006. This work was supported by the National Natural Science Foundation of China under Grant 59685004. This paper was recommended by Associate Editor W. E. Dixon.

The authors are with the Robotics Institute, Shanghai Jiao Tong University, Shanghai 200030, China (e-mail: yuzheng007@sjtu.edu.cn; whqian@sh163.net).

Digital Object Identifier 10.1109/TSMCB.2006.879015

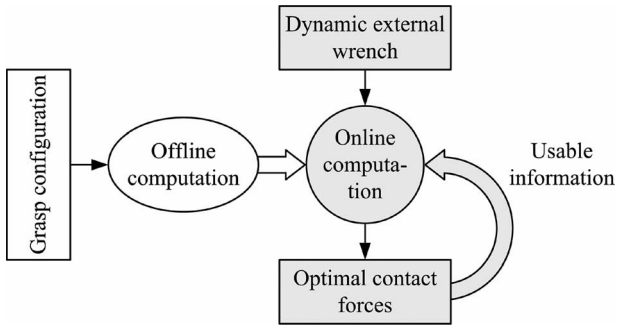

Fig. 1. Schema of the proposed DFD algorithm.

At the cost of some optimality, the algorithms of the second category give priority to computational efficiency. Scientists tried to divide the DFD algorithm into an offline phase and an online phase. Only the latter runs in real time and can speed up if the former covers more computation. To realize this good idea, however, is not easy, because the optimal contact forces are varying in accordance with the dynamic external wrench. Minimizing the online computation while keeping the solution possibly optimal is the crux of the two-phase approach. Park and Starr [16], Maekawa *et al.* [17], Zuo and Qian [18], and Zheng and Qian [19] put forward different techniques to solve the crux. Besides, some analytical or suboptimal methods can be found in the literature [20], [21].

There are primarily two criteria for optimizing contact forces, namely: 1) minimal inclination angle for increasing stability and 2) minimal force magnitude for enhancing material safety as well as saving energy [19]. As these criteria can hardly be fulfilled at the same time, and too small an inclination angle does not make any profit, the latter was mainly considered [2]–[23]. The mainstream [2]–[15] minimizes the contact forces by minimizing the sum of their normal force components. Other work relevant to DFD includes using neural networks [22], [23], decomposing the contact force [24]–[29], and establishing a general framework for DFD [30].

In this correspondence, we present a novel technique not only to reduce the online computation to an extreme but also to achieve truly optimal contact forces. Our work starts from Liu's achievement [31]. By linearizing friction cones and especially the duality of polytopes, Liu transformed contact force optimization into a new LP problem and solved it by the simplex method [31]. Its objective function to be maximized is the inner product of the external wrench and a vector, and the constraints on the vector are given by a set of linear inequalities, which represent a set of closed half spaces in the wrench space. Their intersection forms a polytope as the feasible region of the LP problem. Convex analysis [32] tells that the solution to the LP problem can always be attained at a vertex of the polytope. We are pleased to see that the polytope relates only to the grasp configuration and can be computed offline (Fig. 1). Its vertices remain fixed, and a sequence of them takes turns at being the solution in real time. As the external wrench changes in direction continuously, the solution shifts from one vertex to its adjacency. If the offline computation contains the adjacencies of each vertex, then the online computation is just seeking an adjacency having the maximum inner product rather than searching all the vertices by comparing their inner products for the maximum. Using the helpful information from the previous solution, the online computation becomes more rapid (Fig. 1).

The rest of this correspondence is arranged as follows. Section II describes the DFD problem. Section III introduces Liu's transformation of the force optimization problem into an LP problem. Section IV

addresses our DFD algorithm. An example and the conclusion are given in Sections V and VI, respectively.

II. Problem Description

Consider an m-finger hand manipulating a three-dimensional (3-D) object, fixed with a right-handed coordinate frame. Assume that each finger contacts the object at a regular point with Coulomb friction. Let $\boldsymbol{n}_i$, $\boldsymbol{o}_i$, and $\boldsymbol{t}_i$ be the unit inward normal and two unit tangent vectors at contact i $(i = 1, 2, \ldots, m)$ in the object coordinate frame such that $\boldsymbol{n}_i = \boldsymbol{o}_i \times \boldsymbol{t}_i$. The contact force $\boldsymbol{f}_i$ can be expressed in the local coordinate frame $\{\boldsymbol{n}_i, \boldsymbol{o}_i, \boldsymbol{t}_i\}$ by

$$\boldsymbol{f}_i = [f_{in} \quad f_{io} \quad f_{it}]^T \tag{1}$$

where f_{in}, f_{io}, and f_{it} are the components of $\boldsymbol{f}_i$ along $\boldsymbol{n}_i$, $\boldsymbol{o}_i$, and $\boldsymbol{t}_i$, respectively. To avoid separation and slip at contact, $\boldsymbol{f}_i$ must satisfy the contact constraint

$$f_{in} \geq 0, \quad f_{io}^2 + f_{it}^2 \leq \mu_i^2 f_{in}^2 \tag{2}$$

where μ_i is the Coulomb friction coefficient at contact i. The nonlinear constraint (2) defines a circular cone named the friction cone. Following the mainstream [2]–[15], we also measure the force magnitude over all contacts by

$$\sigma = \sum_{i=1}^{m} \frac{f_{in}}{f_i^U} \tag{3}$$

where f_i^U is the force upper bound at contact i. The wrench in the object coordinate frame produced by $\boldsymbol{f}_i$ is

$$\boldsymbol{w}_i = \boldsymbol{G}_i \boldsymbol{f}_i$$

where $\boldsymbol{G}_i \in \mathbb{R}^{6\times 3}$ is the grasp matrix for contact i

$$\boldsymbol{G}_i = \begin{bmatrix} \boldsymbol{n}_i & \boldsymbol{o}_i & \boldsymbol{t}_i \\ \boldsymbol{r}_i \times \boldsymbol{n}_i & \boldsymbol{r}_i \times \boldsymbol{o}_i & \boldsymbol{r}_i \times \boldsymbol{t}_i \end{bmatrix}$$

where $\boldsymbol{r}_i$ is the position vector of contact i in the object coordinate frame.

Let $\boldsymbol{w}_{\text{ext}}$ denote the "dynamic" external wrench on the object. For equilibrium, the resultant wrench $\boldsymbol{w}$ applied by the hand should always conform to

$$\boldsymbol{w} = \sum_{i=1}^{m} \boldsymbol{w}_i = \sum_{i=1}^{m} \boldsymbol{G}_i \boldsymbol{f}_i = -\boldsymbol{w}_{\text{ext}}. \tag{4}$$

In real-time control of the hand, $\boldsymbol{w}_{\text{ext}}$ is sampled at a sequence of instants with sufficiently small intervals. Thus we encounter the DFD problem.

DFD Problem: Given $\boldsymbol{r}_i$ and $\boldsymbol{n}_i$, $i = 1, 2, \ldots, m$ and $\boldsymbol{w}_{\text{ext}}$, fast compute $\boldsymbol{f}_i$, $i = 1, 2, \ldots, m$, satisfying (2) and (4) with minimal σ at a sequence of sampling instants.

III. LP Formulation for the DFD Problem

In this section, we retell Liu's transformation of contact force optimization into an LP problem [31] using the terminology of convex analysis. A new formula for computing contact forces is derived in order to accomplish more computation offline.

A. Basic Equations

For linearization, substitute the friction cone with an n-side polyhedral cone ($n \geq 3$ since the friction cone is a 3-D cone). Its side edges in the frame $\{\boldsymbol{n}_i, \boldsymbol{o}_i, \boldsymbol{t}_i\}$ are

$$\boldsymbol{s}_{i,j} = f_i^U \begin{bmatrix} 1 & \mu_i \cos\dfrac{2j\pi}{n} & \mu_i \sin\dfrac{2j\pi}{n} \end{bmatrix}^T, \qquad j = 1, 2, \ldots, n. \tag{5}$$

Then $\boldsymbol{f}_i$ satisfying (2) can be approximately represented by

$$\boldsymbol{f}_i = \sum_{j=1}^{n} \lambda_{i,j} \boldsymbol{s}_{i,j} = \boldsymbol{S}_i \boldsymbol{\lambda}_i, \qquad \lambda_{i,j} \geq 0 \text{ for all } i \text{ and } j \tag{6}$$

where $\boldsymbol{S}_i = [\boldsymbol{s}_{i,1} \quad \boldsymbol{s}_{i,2} \quad \cdots \quad \boldsymbol{s}_{i,n}] \in \mathbb{R}^{3\times n}$, and $\boldsymbol{\lambda}_i = [\lambda_{i,1} \quad \lambda_{i,2} \quad \cdots \quad \lambda_{i,n}]^T \in \mathbb{R}^n$. From (1), (5), and (6), (3) can be rewritten as

$$\sigma = \sum_{i=1}^{m} \frac{f_{in}}{f_i^U} = \sum_{i=1}^{m} \sum_{j=1}^{n} \lambda_{i,j}. \tag{7}$$

Combining (4)–(6) leads to

$$\boldsymbol{w} = \sum_{i=1}^{m} \sum_{j=1}^{n} \lambda_{i,j} \boldsymbol{G}_i \boldsymbol{s}_{i,j} = \sum_{i=1}^{m} \sum_{j=1}^{n} \lambda_{i,j} \boldsymbol{w}_{i,j} = \boldsymbol{W}\boldsymbol{\lambda} = -\boldsymbol{w}_{\text{ext}} \tag{8}$$

where $\boldsymbol{w}_{i,j}$ is called a primitive wrench

$$\boldsymbol{w}_{i,j} = \boldsymbol{G}_i \boldsymbol{s}_{i,j} \tag{9}$$

and $\boldsymbol{W} = [\boldsymbol{w}_{1,1} \quad \boldsymbol{w}_{1,2} \quad \cdots \quad \boldsymbol{w}_{m,n}] \in \mathbb{R}^{6\times mn}$, and $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_1 \; \boldsymbol{\lambda}_2 \; \cdots \; \boldsymbol{\lambda}_m]^T \in \mathbb{R}^{mn}$.

B. Computation of the Minimum Contact Force

From (6)–(8), we see that contact force optimization can be transformed into the computation of nonnegative $\boldsymbol{\lambda}$ satisfying (8) with minimal $\sum_{i=1}^{m} \sum_{j=1}^{n} \lambda_{i,j}$.

For simplicity, replace $\boldsymbol{w}_{i,j}$ with $\boldsymbol{w}_k$ having the sole subscript k, where $k = 1, 2, \ldots, K$, and $K = mn$. Let $\mathcal{W}$ be the set of $\boldsymbol{w}_k$, $k = 1, 2, \ldots, K$, and $\mathcal{W}_c$ be the convex hull of $\mathcal{W}$. From (7) and (8), $\mathcal{W}_c$ consists of all the resultant wrenches that can be generated by the hand with $\sigma = 1$. For a force-closure grasp, the origin $\mathbf{0}$ of the wrench space lies in the interior of $\mathcal{W}_c$.

Let $\boldsymbol{w}_b = \boldsymbol{w}/\sigma$. Then $\sigma = \|\boldsymbol{w}\|/\|\boldsymbol{w}_b\|$, and $\boldsymbol{w}_b$ belongs to $\mathcal{W}_c$. Clearly, σ attains the minimum value when $\boldsymbol{w}_b$ falls on a face of $\mathcal{W}_c$, and $\boldsymbol{w}$ can be restricted to a positive combination of the elements of $\mathcal{W}$ on the face. Partition $\boldsymbol{W}$ into $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$, which comprise such elements and the others, respectively. Correspondingly, partition the identity matrix $\boldsymbol{I}$ of rank mn in columns into $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$. From [31], this nonnegative $\boldsymbol{\lambda}$ with minimal $\sum_{i=1}^{m} \sum_{j=1}^{n} \lambda_{i,j}$ can be calculated by

$$\boldsymbol{\lambda} = \boldsymbol{I}_1 \boldsymbol{W}_1^{+} \boldsymbol{w} = \boldsymbol{C}\boldsymbol{w} \tag{10}$$

where $\boldsymbol{W}_1^{+}$ is the pseudoinverse of $\boldsymbol{W}_1$, and $\boldsymbol{C} = \boldsymbol{I}_1 \boldsymbol{W}_1^{+} \in \mathbb{R}^{mn\times 6}$. Accordingly, the minimum contact forces $\boldsymbol{f}_i$, $i = 1, 2, \ldots, m$, can be computed by (6). Now we may partition $\boldsymbol{C}$ equally in rows into m submatrices $\boldsymbol{C}_i \in \mathbb{R}^{n\times 6}$, $i = 1, 2, \ldots, m$. Combining (6) and (10) yields

$$\boldsymbol{f}_i = \boldsymbol{S}_i \boldsymbol{C}_i \boldsymbol{w} = \boldsymbol{D}_i \boldsymbol{w} \tag{11}$$

where $\boldsymbol{D}_i = \boldsymbol{S}_i \boldsymbol{C}_i \in \mathbb{R}^{3\times 6}$.

The key to this approach is finding the aforementioned face of $\mathcal{W}_c$, which can be done as follows.

Let $\mathcal{W}_c^*$ denote the polar set of $\mathcal{W}_c$, which is given by

$$\mathcal{W}_c^* = \left\{ \boldsymbol{u} \in \mathbb{R}^6 | \boldsymbol{w}_k^T \boldsymbol{u} \leq 1 \text{ for all } k = 1, 2, \ldots, K \right\}. \tag{12}$$

Since $\mathcal{W}_c$ is a polytope containing the origin $\mathbf{0}$ as an interior point, $\mathcal{W}_c^*$ is a suchlike polytope [32]. Let p denote the support function of $\mathcal{W}_c^*$, which is the real-valued defined by

$$p(\boldsymbol{w}) = \sup_{\boldsymbol{u} \in \mathcal{W}_c^*} \boldsymbol{w}^T \boldsymbol{u} = \max_{\boldsymbol{u} \in \mathcal{W}_c^*} \boldsymbol{w}^T \boldsymbol{u}. \tag{13}$$

Proposition 1 [33]: Let $\mathrm{bd}\mathcal{W}_c$ denote the boundary of $\mathcal{W}_c$ and $\boldsymbol{u}_s$ a point of $\mathcal{W}_c^*$ such that $p(\boldsymbol{w}) = \boldsymbol{w}^T \boldsymbol{u}_s$. Then the following statements are true.

a) $p(\boldsymbol{w})^{-1} \boldsymbol{w} \in \mathrm{bd}\mathcal{W}_c$.
b) The hyperplane $H = \{ \boldsymbol{w} \in \mathbb{R}^6 | \boldsymbol{u}_s^T \boldsymbol{w} = 1 \}$ supports $\mathcal{W}_c$ at the point $p(\boldsymbol{w})^{-1} \boldsymbol{w}$.

Proposition 1(a) implies $\boldsymbol{w}_b = p(\boldsymbol{w})^{-1} \boldsymbol{w}$ and $\sigma_{\min} = p(\boldsymbol{w})$. Proposition 1(b) indicates that the hyperplane H contains the face we are seeking. From (12) and (13), $\boldsymbol{u}_s$ is the solution to the LP problem

$$\begin{cases} \text{maximize } \boldsymbol{w}^T \boldsymbol{u} \\ \text{subject to } \boldsymbol{w}_k^T \boldsymbol{u} \leq 1, \qquad k = 1, 2, \ldots, K. \end{cases} \tag{14}$$

Solving it in $O(K)$ time, the simplex method is the fastest way up to now [31]. However, its computation time is still considerable, especially when K is large. This motivates us to explore a more efficient approach.

IV. Innovated Approach to the LP Problem

The feasible region of the LP problem (14), i.e., the polytope $\mathcal{W}_c^*$, is determined by $\boldsymbol{w}_k$, $k = 1, 2, \ldots, K$, and related only to the grasp configuration. Since the objective function $\boldsymbol{w}^T \boldsymbol{u}$ is a linear function and bounded on $\mathcal{W}_c^*$, the maximum of $\boldsymbol{w}^T \boldsymbol{u}$ on $\mathcal{W}_c^*$, i.e., $p(\boldsymbol{w})$, can always be attained at a vertex of $\mathcal{W}_c^*$.

A. Algorithm for Seeking Vertices and Adjacencies

Herein we not only compute all the vertices of $\mathcal{W}_c^*$ but also clarify the adjacent vertices of each vertex in order to facilitate the online computation.

Equation (12) indicates that $\mathcal{W}_c^*$ is the intersection of the half-spaces bounded by the hyperplanes

$$H_k = \left\{ \boldsymbol{u} \in \mathbb{R}^6 | \boldsymbol{w}_k^T \boldsymbol{u} = 1 \right\}, \qquad k = 1, 2, \ldots, K. \tag{15}$$

The points $\boldsymbol{w}_k$, $k = 1, 2, \ldots, K$, can be classified as follows:

A1) a vertex of $\mathcal{W}_c$;
A2) a point on the boundary of $\mathcal{W}_c$ but not a vertex;
A3) an interior point of $\mathcal{W}_c$.

From the duality theory [32], d-faces (i.e., faces of dimension d) of $\mathcal{W}_c$ and $(5-d)$-faces of $\mathcal{W}_c^*$ are one-to-one correspondent. Because of this duality, the hyperplanes H_k, $k = 1, 2, \ldots, K$, can be classified according to the above classification of $\boldsymbol{w}_k$, $k = 1, 2, \ldots, K$, as follows:

B1) containing a facet (i.e., a 5-face) of $\mathcal{W}_c^*$;
B2) containing a face of $\mathcal{W}_c^*$ with its dimension below five;
B3) lying outside $\mathcal{W}_c^*$ and being redundant.

Clearly, a vertex of $\mathcal{W}_c^*$ is the intersection of six hyperplanes of type B1 or of some hyperplanes of types B1 and B2.

Proposition 2: Let $\boldsymbol{w}_{k_1}, \boldsymbol{w}_{k_2}, \ldots, \boldsymbol{w}_{k_6}$ be six elements of $\mathcal{W}$, and $\boldsymbol{M} = [\boldsymbol{w}_{k_1}\ \boldsymbol{w}_{k_2}\ \cdots\ \boldsymbol{w}_{k_6}]$. Then the solution to the system

$$\boldsymbol{M}^T \boldsymbol{u} = [1 \quad 1 \quad \cdots \quad 1]^T \tag{16}$$

denoted by $\hat{\boldsymbol{u}}$ is a vertex of $\mathcal{W}_c^*$ if the following conditions are both satisfied.

1) The reciprocal of the condition number of $\boldsymbol{M}$, denoted by $\kappa(\boldsymbol{M})$, is nonzero.
2) $\boldsymbol{w}_k^T \hat{\boldsymbol{u}} \leq 1$ for all $k = 1, 2, \ldots, K$.

Proof: Condition 1) implies that the square matrix $\boldsymbol{M}$ is nonsingular, and the solution $\hat{\boldsymbol{u}}$ to the linear system is unique. This means that the hyperplanes $H_{k_1}, H_{k_2}, \ldots, H_{k_6}$ given by (15) with respect to $\boldsymbol{w}_{k_1}, \boldsymbol{w}_{k_2}, \ldots, \boldsymbol{w}_{k_6}$ intersect at merely the point $\hat{\boldsymbol{u}}$. Condition 2) further ensures that $\hat{\boldsymbol{u}}$ is not outside of $\mathcal{W}_c^*$, which implies that none of $H_{k_1}, H_{k_2}, \ldots, H_{k_6}$ is of type B3. Therefore, $\hat{\boldsymbol{u}}$ is a vertex of $\mathcal{W}_c^*$. ■

Suppose that a vertex $\hat{\boldsymbol{u}}$ is found. Then we may divide $\mathcal{W}$ into $\mathcal{W}_1$ and $\mathcal{W}_2$ according to $\hat{\boldsymbol{u}}$, where $\mathcal{W}_1$ consists of $\boldsymbol{w}_k$, $k = 1, 2, \ldots, K$, on the facet of $\mathcal{W}_c$ dual to $\hat{\boldsymbol{u}}$, i.e., satisfying $\hat{\boldsymbol{u}}^T \boldsymbol{w}_k = 1$, and $\mathcal{W}_2$ consists of the others.

Proposition 3: Suppose that $\hat{\boldsymbol{u}}_1$ is a vertex of $\mathcal{W}_c^*$, and $\mathcal{W}$ is divided into $\mathcal{W}_1$ and $\mathcal{W}_2$ according to $\hat{\boldsymbol{u}}_1$. Let $\boldsymbol{w}_{k_1}, \boldsymbol{w}_{k_2}, \ldots, \boldsymbol{w}_{k_5}$ be five elements of $\mathcal{W}_1$, and $\boldsymbol{w}_{k_6}$ be an element of $\mathcal{W}_2$. If $\boldsymbol{w}_{k_1}, \boldsymbol{w}_{k_2}, \ldots, \boldsymbol{w}_{k_6}$ determine a vertex $\hat{\boldsymbol{u}}_2$ of $\mathcal{W}_c^*$, then the following statements are true.

1) $\hat{\boldsymbol{u}}_1$ and $\hat{\boldsymbol{u}}_2$ are different.
2) $\hat{\boldsymbol{u}}_1$ and $\hat{\boldsymbol{u}}_2$ are adjacent.
3) The intersection of the hyperplanes $H_{k_1}, H_{k_2}, \ldots, H_{k_5}$ given by (15) with respect to $\boldsymbol{w}_{k_1}, \boldsymbol{w}_{k_2}, \ldots, \boldsymbol{w}_{k_5}$ is the edge of $\mathcal{W}_c^*$ connecting $\hat{\boldsymbol{u}}_1$ and $\hat{\boldsymbol{u}}_2$.

Proof:

1) If $\hat{\boldsymbol{u}}_2 = \hat{\boldsymbol{u}}_1$, then from (16) we get $\hat{\boldsymbol{u}}_1^T \boldsymbol{w}_{k_6} = 1$, which implies that $\boldsymbol{w}_{k_6}$ is an element of $\mathcal{W}_1$ other than $\mathcal{W}_2$.

2), 3) If $\boldsymbol{w}_{k_1}, \boldsymbol{w}_{k_2}, \ldots, \boldsymbol{w}_{k_6}$ determine a vertex $\hat{\boldsymbol{u}}_2$, then $\boldsymbol{w}_{k_1}, \boldsymbol{w}_{k_2}, \ldots, \boldsymbol{w}_{k_5}$ are linearly independent, and the convex hull of them is a ridge (i.e., a 4-face) of $\mathcal{W}_c$. The set dual to the ridge, namely the intersection of $H_{k_1}, H_{k_2}, \ldots, H_{k_5}$, is the edge of $\mathcal{W}_c^*$ connecting $\hat{\boldsymbol{u}}_1$ and $\hat{\boldsymbol{u}}_2$. ■

Proposition 2 shows how to find a vertex, while Proposition 3 indicates how to find its adjacent vertices. By this, we have the following algorithm for seeking all the vertices of $\mathcal{W}_c^*$ and their adjacencies. First introduce the notations.

$\boldsymbol{u}_l$	lth vertex of $\mathcal{W}_c^*$.
l_1	Index of a vertex whose adjacency is explored.
l_2	Index of a vertex that has been found before.
L	Number of the vertices that have been found.
U_l	Index set of the vertices adjacent to vertex l.
N_l	Number of the vertices adjacent to vertex l.
$\hat{\boldsymbol{u}}$	Solution to the linear system (16).
$W_{l,1}$	Index set of $\boldsymbol{w}_k$, $k = 1, 2, \ldots, K$, satisfying $\boldsymbol{u}_l^T \boldsymbol{w}_k = 1$.
$W_{l,2}$	$W_{l,2} = \{1, 2, \ldots, K\} \setminus W_{l,1}$.
$\mathcal{F}_l$	Family of the edges of $\mathcal{W}_c^*$ that connect vertex l with its adjacent vertices. Each element of $\mathcal{F}_l$ is expressed by $\{k_1, k_2, \ldots, k_5\}$, where $k_1, k_2, \ldots, k_5$ belong to $W_{l,1}$.

Step 1) Find the first vertex $\boldsymbol{u}_1$. This can be done by searching $\mathcal{W}$ for $\boldsymbol{w}_{k_1}, \boldsymbol{w}_{k_2}, \ldots, \boldsymbol{w}_{k_6}$ satisfying Proposition 2 or solving the LP problem (14) with an arbitrary nonzero $\boldsymbol{w}$. Construct the index sets $W_{1,1}$ and $W_{1,2}$ according to $\boldsymbol{u}_1$. Set $L = 1$, $U_1 = \emptyset$, $N_1 = 0$, $\mathcal{F}_1 = \emptyset$, and $l_1 = 0$.

Step 2) If $l_1 < L$, then $l_1 = l_1 + 1$, and go to Step 3); otherwise, the procedure ends.
Step 3) Select five elements of $W_{l_1,1}$, say $k_1, k_2, \ldots, k_5$, which were not used together before. If all the combinations of the elements of $W_{l_1,1}$ have been examined, then the vertices adjacent to vertex l_1 are all found, and return to Step 2).
Step 4) If $\{k_1, k_2, \ldots, k_5\}$ exists in $\mathcal{F}_{l_1}$, then the vertex adjacent to vertex l_1 connected by the edge $\{k_1, k_2, \ldots, k_5\}$ has been found, and return to Step 3).
Step 5) Select elements of $W_{l_1,2}$, say k_6, which was not be used before. If all the elements of $W_{l_1,2}$ have been taken, then $\boldsymbol{w}_{k_1}, \boldsymbol{w}_{k_2}, \ldots, \boldsymbol{w}_{k_5}$ are not corresponding to an edge of $\mathcal{W}_c^*$, and return to Step 3).
Step 6) Construct the matrix $\boldsymbol{M} = [\boldsymbol{w}_{k_1}\ \boldsymbol{w}_{k_2}\ \cdots\ \boldsymbol{w}_{k_6}]$. If $\boldsymbol{M}$ is nonsingular, then solve the linear system (16), and go to Step 7); otherwise, return to Step 5).
Step 7) If the solution $\hat{\boldsymbol{u}}$ satisfies $\boldsymbol{w}_k^T \hat{\boldsymbol{u}} \leq 1$ for $k = 1, 2, \ldots, K$, then a vertex is found, and go to Step 8); otherwise, return to Step 5).
Step 8) If $\hat{\boldsymbol{u}}$ is a new vertex, then set $L = L + 1$, $\boldsymbol{u}_L = \hat{\boldsymbol{u}}$, $U_L = \{l_1\}$, $N_L = 1$, and $\mathcal{F}_L = \{\{k_1, k_2, \ldots, k_5\}\}$. Construct $W_{L,1}$ and $W_{L,2}$. Set $U_{l_1} = U_{l_1} \cup \{L\}$ and $N_{l_1} = N_{l_1} + 1$. Add $\{k_1, k_2, \ldots, k_5\}$ to $\mathcal{F}_{l_1}$. Return to Step 3). Otherwise, note down the index of the vertex, say l_2.
Step 9) If l_2 does not exist in U_{l_1}, then set $U_{l_1} = U_{l_1} \cup \{l_2\}$ and $N_{l_1} = N_{l_1} + 1$. Add $\{k_1, k_2, \ldots, k_5\}$ to $\mathcal{F}_{l_1}$. If l_1 does not exist in U_{l_2}, then set $U_{l_2} = U_{l_2} \cup \{l_1\}$ and $N_{l_2} = N_{l_2} + 1$. Add $\{k_1, k_2, \ldots, k_5\}$ to $\mathcal{F}_{l_2}$. Return to Step 3).

Because there exist points of type A2 in $\mathcal{W}$, the linear system (16) may be singular or the solution $\hat{\boldsymbol{u}}$ to (16) may be a vertex that has been found before. Thus, it needs to determine whether $\boldsymbol{M}$ is singular and $\hat{\boldsymbol{u}}$ is new in Steps 6) and 8), respectively. Due to the existence of points of type A3, some $\hat{\boldsymbol{u}}$ may not be a point of $\mathcal{W}_c^*$; hence, Step 7) is required.

B. Fast Procedure for Solving the LP Problem

After all the vertices of $\mathcal{W}_c^*$, namely $\boldsymbol{u}_l$, $l = 1, 2, \ldots, L$, were sought out, (13) can be rewritten as

$$p(\boldsymbol{w}) = \max_{l=1,2,\ldots,L} \boldsymbol{w}^T \boldsymbol{u}_l. \tag{17}$$

Then the solution $\boldsymbol{u}_s$ to the LP problem (14) can be obtained just by computing and comparing L inner products. Since we know the adjacencies of each vertex, $\boldsymbol{u}_s$ can be found more quickly by the following procedure, which successively travels from one vertex to its adjacency until arriving at a vertex whose adjacencies cannot make an ascent of the inner product $\boldsymbol{w}^T \boldsymbol{u}_l$. Begin with some notations.

l_0 Index of the initial vertex.
l_1 Index of a vertex.
l_2 Index of a vertex adjacent to vertex l_1.

Step 1) Given l_0, set $l_1 = l_0$.
Step 2) Set l_2 to be an element of U_{l_1}.
Step 3) If $\boldsymbol{w}^T \boldsymbol{u}_{l_2} > \boldsymbol{w}^T \boldsymbol{u}_{l_1}$, then $l_1 = l_2$ and return to Step 2).
Step 4) If all the elements of U_{l_1} are checked, then return l_1 and terminate the procedure; otherwise, assign another element of U_{l_1} to l_2 and return to Step 3).

In real time, the procedure can speed up by taking the last solution vertex as the initial vertex. Generally, the succeeding solution is closed to the preceding or even the same temporally. Otherwise, the sampling interval should be shortened.

C. Algorithm for DFD

Combining the above arguments leads to a new two-phase DFD algorithm, which can be implemented as follows.

1) Offline phase: Given the contact positions and the inward normals, i.e., $\boldsymbol{r}_i$ and $\boldsymbol{n}_i$, $i = 1, 2, \ldots, m$.

Step 1) Compute $\boldsymbol{w}_k$, $k = 1, 2, \ldots, K$, by (9).
Step 2) Compute $\boldsymbol{u}_l$, U_l, $W_{l,1}$, $l = 1, 2, \ldots, L$, by the searching algorithm.
Step 3) Construct $\boldsymbol{W}_{l,1}$ and $\boldsymbol{I}_{l,1}$ for $l = 1, 2, \ldots, L$ by selecting the columns of $\boldsymbol{W}$ and $\boldsymbol{I}$ according to $W_{l,1}$, respectively. Set $\boldsymbol{C}_l = \boldsymbol{I}_{l,1} \boldsymbol{W}_{l,1}^+ \in \mathbb{R}^{mn \times 6}$ for $l = 1, 2, \ldots, L$. Partition $\boldsymbol{C}_l$ equally in rows into m submatrices $\boldsymbol{C}_{i,l}$, $i = 1, 2, \ldots, m$. Set $\boldsymbol{D}_{i,l} = \boldsymbol{S}_i \boldsymbol{C}_{i,l} \in \mathbb{R}^{3 \times 6}$ for $i = 1, 2, \ldots, m$ and $l = 1, 2, \ldots, L$.
Step 4) Set l_0 to be an arbitrary element of $\{1, 2, \ldots, L\}$ and $t = 0$.

2) Online phase: Given a dynamic external wrench $\boldsymbol{w}_{\text{ext}}$ with duration T and sampling interval Δt.

Step 5) Set $\boldsymbol{w} = -\boldsymbol{w}_{\text{ext}}(t)$. Find the index l_1 by calling the foregoing procedure with l_0.
Step 6) Compute $\boldsymbol{f}_i = \boldsymbol{D}_{i,l_1} \boldsymbol{w}$ for $i = 1, 2, \ldots, m$.
Step 7) Set $t = t + \Delta t$. If $t > T$, then the algorithm ends; otherwise, set $l_0 = l_1$ and return to Step 5).

The applicability of the DFD algorithm to real-time control of the robot hand is decided by the time complexity of its online phase, mainly of Steps 5) and 6). Step 5) finds the solution vertex at each instant. The worst situation may probably happen at the first instant, since the initial vertex is selected at random in Step 4) in the absence of prior information about the solution vertex. In spite of this, the computation time is much less than $O(K)$ of the simplex method and $O(L)$ of comparing all L inner products. After that, Step 5) runs in hopeful $O(1)$ time. Step 6) returns the optimal contact forces. The matrices $\boldsymbol{D}_{i,l_1}$, $i = 1, 2, \ldots, m$, have been calculated offline in Step 3) so that only m multiplications of a 3×6 matrix by a vector are required here. Therefore, for solving the DFD problem, this algorithm is more efficient than the previous ones with polynomial [12] or quadratic [13]–[15] complexities. Compared with [18] and [19], its online computation cost is not greatly reduced, but the computed contact forces are indeed optimal.

V. Numerical Example

Herein we demonstrate the efficiency of the proposed DFD algorithm using Matlab.

The object to be manipulated is a teapot, as shown in Fig. 2. Assume that the dynamic external wrench is given by

$$\boldsymbol{w}_{\text{ext}} = \begin{bmatrix} 0.5\cos 2.4\pi t \\ (2 + 0.5\sin 2.4\pi t)\sin 0.4\pi t \\ (2 + 0.5\sin 2.4\pi t)\cos 0.4\pi t - 5 \\ \cos(\sin 1.2\pi t)\sin(\cos 0.4\pi t)) \\ \sin(\sin 1.2\pi t) \\ \cos(\sin 1.2\pi t)\cos(\cos 0.4\pi t)) \end{bmatrix}.$$

The external wrench is periodic and its period $T = 5$ s. Assign the sampling interval $\Delta t = T/1000 = 5$ ms. We grasp the teapot with a four-finger [Fig. 2(a)] and a five-finger [Fig. 2(b)] robot hand. Their

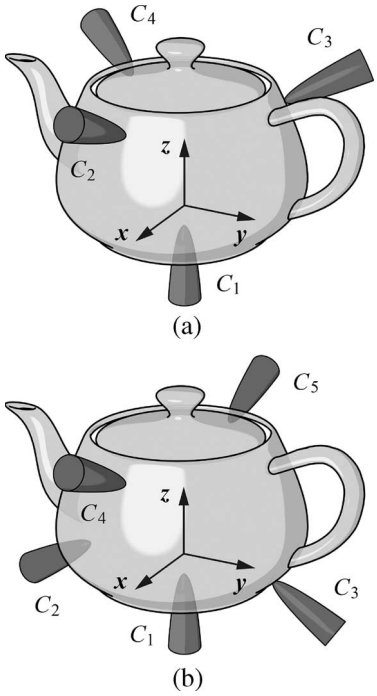

Fig. 2. Teapot grasped by (a) four-finger hand and (b) five-finger hand.

contact positions, unit inward normals, and force upper bounds are as follows:

Grasp (a)

$$\boldsymbol{r}_1=[0 \quad 0 \quad -20]^T \quad \boldsymbol{n}_1=[0 \quad 0 \quad 1]^T$$
$$\boldsymbol{r}_2=[50 \quad 0 \quad 40]^T \quad \boldsymbol{n}_2=[-\sqrt{3}/2 \quad 0 \quad -1/2]^T$$
$$\boldsymbol{r}_3=[-25 \quad 25\sqrt{3} \quad 40]^T \quad \boldsymbol{n}_3=[\sqrt{3}/4 \quad -3/4 \quad -1/2]^T$$
$$\boldsymbol{r}_4=[-25 \quad -25\sqrt{3} \quad 40]^T \quad \boldsymbol{n}_3=[\sqrt{3}/4 \quad 3/4 \quad -1/2]^T$$
$$f_1^U=20 \text{ N} \quad f_i^U=10 \text{ N for } i=2,3,4.$$

Grasp (b)

$$\boldsymbol{r}_1=[0 \quad 0 \quad -20]^T \quad \boldsymbol{n}_1=[0 \quad 0 \quad 1]^T$$
$$\boldsymbol{r}_2=[0 \quad -50 \quad -15]^T \quad \boldsymbol{n}_2=[0 \quad \sqrt{2}/2 \quad \sqrt{2}/2]^T$$
$$\boldsymbol{r}_3=[0 \quad 50 \quad -15]^T \quad \boldsymbol{n}_3=[0 \quad -\sqrt{2}/2 \quad \sqrt{2}/2]^T$$
$$\boldsymbol{r}_4=[50 \quad 0 \quad 40]^T \quad \boldsymbol{n}_4=[-\sqrt{3}/2 \quad 0 \quad -1/2]^T$$
$$\boldsymbol{r}_5=[-50 \quad 0 \quad 40]^T \quad \boldsymbol{n}_3=[\sqrt{3}/2 \quad 0 \quad -1/2]^T$$
$$f_1^U=20 \text{ N} \quad f_2^U=f_3^U=15 \text{ N} \quad f_4^U=f_5^U=10 \text{ N}.$$

The friction coefficient is 0.2 at these contacts. Each friction cone is substituted by a ten-side polyhedral cone, i.e., $n=10$ in (5). Running the searching algorithm, we see that $\mathcal{W}_c^*$ has 861 vertices for (a) and 1618 vertices for (b), i.e., $L=861$ for (a) and $L=1618$ for (b). Also, in both cases, a vertex has at most 13 adjacencies and most vertices have only six adjacencies.

We compute the optimal contact forces by three methods, namely: 1) the simplex method; 2) comparing L inner products as stated by (17) and using the function max in Matlab; and 3) the present approach. Their CPU times on various PCs are listed in Table I. The CPU times by 3) are approximately 1/1000 of those by 1). Reduction from 1) to 2) comes from the new two-phase technique. Further reduction to 3) is the contribution of utilizing prior information. More importantly, the CPU times by 1) are much greater than the sampling interval; therefore, the robot hand cannot be controlled in this way. Contrarily, the CPU times by 3) are far below the interval so that the real-time control can be fulfilled with ease.

TABLE I
EXECUTION TIME FOR AN INSTANT ON PCS WITH DIFFERENT CPUS

Grasp	CPU	1) Simplex method	2) Comparing all inner products	3) Our DFD approach
(a)	P4 1.7G	36.65 ms	0.2149 ms	0.0351 ms
	P4 2.8G	24.18 ms	0.1483 ms	0.0213 ms
	P4 3.4G	17.84 ms	0.1178 ms	0.0175 ms
(b)	P4 1.7G	48.02 ms	0.3094 ms	0.0372 ms
	P4 2.8G	25. 95 ms	0.1802 ms	0.0226 ms
	P4 3.4G	20.53 ms	0.1290 ms	0.0185 ms

VI. CONCLUSION

DFD in multifingered grasping can be formulated as an LP problem, of which the objective function is time varying along with the dynamic external wrench, and the feasible region is a polytope. As a crucial requirement, the problem must be solved in real time. Realizing that the polytope is fixed according to the grasp configuration, and the solution is moving successively from one vertex to its neighbor as the external wrench varies, we compute all the vertices and their adjacencies in the offline phase. Then in the online phase, the sequential solution can be found stepwise very quickly by looking in the adjacency of the previous solution vertex for a new one. Thus, an advanced DFD algorithm comes out with the following advantages.

- Dramatically faster than the simplex method, it runs at a speed so high as to richly satisfy the requirement of real-time control.
- The computed contact forces are really optimal. This means the foregoing two categories of DFD algorithms interact here eventually.
- The two-phase approach has a vivid geometrical meaning.
- Since the external wrench is not involved in the offline computation, it need not be known in advance and can be acquired through sensors in real time. Hence, this algorithm is also and particularly suitable for nonholonomic applications.
- By linearizing the soft finger contact constraint [34], this algorithm can be applied to such contact as well.

ACKNOWLEDGMENT

The authors would like to thank the reviewers and the editors for their careful review and valuable advice.

REFERENCES

[1] K. Salisbury and B. Roth, "Kinematic and force analysis of articulated mechanical hands," *ASME J. Mech., Transmiss., Autom. Design*, vol. 105, no. 1, pp. 35–41, Mar. 1983.
[2] J. Kerr and B. Roth, "Analysis of multifingered hands," *Int. J. Robot. Res.*, vol. 4, no. 4, pp. 3–17, 1985.
[3] F. T. Cheng and D. E. Orin, "Efficient algorithm for optimal force distribution—The compact-dual LP method," *IEEE Trans. Robot. Autom.*, vol. 6, no. 2, pp. 178–187, Apr. 1990.
[4] ——, "Optimal force distribution in multiple-chain robotic system," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 1, pp. 13–24, Jan./Feb. 1991.

[5] ——, "Efficient formulation of the force distribution equations for simple closed-chain robotic mechanisms," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 1, pp. 25–32, Jan./Feb. 1991.
[6] M. Nahon and J. Angeles, "Real-time force optimization in parallel kinematic chains under inequality constraints," *IEEE Trans. Robot. Autom.*, vol. 8, no. 4, pp. 439–450, Aug. 1992.
[7] ——, "Optimization of dynamic forces in mechanical hands," *ASME J. Mech. Design*, vol. 113, no. 2, pp. 167–173, Jun. 1991.
[8] Y. Nakamura, K. Nagai, and T. Yoshikawa, "Dynamics and stability in coordination of multiple robotic mechanisms," *Int. J. Robot. Res.*, vol. 8, no. 2, pp. 44–61, 1989.
[9] M. Buss, H. Hashimoto, and J. B. Moore, "Dextrous hand grasping force optimization," *IEEE Trans. Robot. Autom.*, vol. 12, no. 3, pp. 406–418, Jun. 1996.
[10] M. Buss, L. Faybusovich, and J. Moore, "Recursive algorithms for real-time grasping force optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, Albuquerque, NM, Apr. 1997, pp. 682–687.
[11] ——, "Dikin-type algorithms for dextrous grasping force optimization," *Int. J. Robot. Res.*, vol. 17, no. 8, pp. 831–839, 1998.
[12] L. Han, J. C. Trinkle, and Z. X. Li, "Grasp analysis as linear matrix inequality problems," *IEEE Trans. Robot. Autom.*, vol. 16, no. 6, pp. 663–674, Dec. 2000.
[13] U. Helmke, K. Hüper, and J. B. Moore, "Quadratically convergent algorithms for optimal dextrous hand grasping," *IEEE Trans. Robot. Autom.*, vol. 18, no. 2, pp. 138–146, Apr. 2002.
[14] G. F. Liu and Z. X. Li, "Real-time grasping-force optimization for multifingered manipulation: Theory and experiments," *IEEE/ASME Trans. Mechatronics*, vol. 9, no. 1, pp. 65–77, Mar. 2004.
[15] G. F. Liu, J. J. Xu, and Z. X. Li, "On geometric algorithms for real-time grasping force optimization," *IEEE Trans. Control Syst. Technol.*, vol. 12, no. 6, pp. 843–859, Nov. 2004.
[16] Y. C. Park and G. P. Starr, "Finger force computation for manipulation of an object by a multifingered robot hand," in *Proc. IEEE Int. Conf. Robot. Autom.*, Scottsdale, AZ, May 1989, vol. 2, pp. 930–935.
[17] H. Maekawa, K. Tanie, and K. Komoriya, "Dynamic grasping force control using tactile feedback for grasp of multifingered hand," in *Proc. IEEE Int. Conf. Robot. Autom.*, Minneapolis, MN, Apr. 1996, pp. 2462–2469.
[18] B.-R. Zuo and W.-H. Qian, "A general dynamic force distribution algorithm for multifingered grasping," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 30, no. 1, pp. 185–192, Feb. 2000.
[19] Y. Zheng and W.-H. Qian, "Dynamic force distribution in multifingered grasping by decomposition and positive combination," *IEEE Trans. Robot.*, vol. 21, no. 4, pp. 718–726, Aug. 2005.
[20] Z. Ji and B. Roth, "Direct computation of grasping force for three-finger tip-prehension grasps," *ASME J. Mech., Transmiss. Autom. Design*, vol. 110, no. 4, pp. 405–413, Dec. 1988.
[21] V. R. Kumar and K. J. Waldron, "Suboptimal algorithms for force distribution in multifingered grippers," *IEEE Trans. Robot. Autom.*, vol. 5, no. 4, pp. 491–498, Aug. 1989.
[22] L.-M. Fok and J. Wang, "Two recurrent neural networks for grasping force optimization of multi-fingered robotic hands," in *Proc. Int. Joint Conf. Neural Netw.*, Honolulu, HI, May 2002, pp. 35–40.
[23] Y. S. Xia, J. Wang, and L.-M. Fok, "Grasping-force optimization for multifingered robotic hands using a recurrent neural network," *IEEE Trans. Robot. Autom.*, vol. 20, no. 3, pp. 549–554, Jun. 2004.
[24] T. Yoshikawa and K. Nagai, "Manipulating and grasping forces in manipulation by multifingered robot hands," *IEEE Trans. Robot. Autom.*, vol. 7, no. 1, pp. 67–77, Feb. 1991.
[25] A. Bicchi, "Force distribution in multiple whole-limb manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Atlanta, GA, May 1993, vol. 2, pp. 196–201.
[26] Y.-C. Chen, I. D. Walker, and J. B. Cheatham, "Visualization of force-closure grasps for objects through contact force decomposition," *Int. J. Robot. Res.*, vol. 14, no. 1, pp. 37–75, Feb. 1995.
[27] M. Aicardi, G. Casalino, and G. Cannata, "Contact force canonical decomposition and the role of internal forces in robust grasp planning problems," *Int. J. Robot. Res.*, vol. 15, no. 4, pp. 351–364, Aug. 1996.
[28] F. T. Cheng, "An efficient method for obtaining the general solution for the force balance equations with hard point contacts," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 27, no. 2, pp. 255–260, Apr. 1997.
[29] Y. Zhang, W. A. Gruver, J. Li, and Q. Zhang, "Classification of grasps by robot hands," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 31, no. 3, pp. 436–444, Jun. 2001.
[30] D. P. Chevallier and S. Payandeh, "On computation of grasping forces in dynamic manipulation using a three-fingered grasp," *Mech. Mach. Theory*, vol. 33, no. 3, pp. 225–244, Apr. 1998.
[31] Y.-H. Liu, "Qualitative test and force optimization of 3-D frictional form-closure grasps using linear programming," *IEEE Trans. Robot. Autom.*, vol. 15, no. 1, pp. 163–173, Feb. 1999.
[32] S. R. Lay, *Convex Sets and Their Application*. New York: Wiley, 1982.
[33] Y. Zheng and W.-H. Qian, "Simplification of the ray-shooting based algorithm for 3-D force-closure test," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 470–473, Jun. 2005.
[34] ——, "Linearizing the soft finger contact constraint with application to dynamic force distribution in multifingered grasping," *Sci. China,* Ser. E, vol. 48, no. 2, pp. 121–130, 2005.