cables to eight or even seven. Once the possibilities of directly transferring existing theorems from grasping have been exhausted, their underlying tools, such as Grassmann theory and convexity theory, can be used to derive customized theorems for other types of cable robots. Regardless of the method, significant analysis of cable robots can be derived from the abundant work on multifinger grasps.

VI. Conclusion

Although the close relationship between cable robots and grasping, which are both based on unidirectional constraints, has been pointed out in literature, it has not yet been fully exploited. In this paper, it was shown that the antipodal grasp theorem can be applied to cable robots. It is hoped that this paper will spur other new ideas to be taken from grasping and applied to cable robots and vice versa.

References

[1] R. Bostelman, A. Jacoff, F. Proctor, T. Kramer, and A. Wavering, "Cable-based reconfigurable machines for large scale manufacturing," in *Proc. Japan-USA Symp. Flexible Automation*, Ann Arbor, MI, Jul. 2000.

[2] T. Morizono, K. Kurahashi, and S. Kawamura, "Realization of a virtual sports training system with parallel wire mechanism," in *Proc IEEE Int. Conf. Robotics Automation*, Albuquerque, NM, Apr. 1997, pp. 3025–3030.

[3] S. Kawamura and K. Ito, "A new type of master robot for teleoperation using a radial wire drive system," in *Proc IEEE/RSJ Int. Conf. Intelligent Robots Systems*, Yokohama, Japan, Jul. 1993, pp. 55–60.

[4] A. Ming and T. Higuchi, "Study on multiple degree-of-freedom positioning mechanism using wires (part 1)," *Int. J. Jpn. Soc. Precision Eng.*, vol. 28, no. 2, pp. 131–138, 1994.

[5] S. Kawamura, W. Choe, S. Tanaka, and S. Pandian, "Development of an ultrahigh speed robot FALCON using wire drive system," in *Proc. IEEE/ICRA Int. Conf. Robotics Automation*, vol. 1, Nagoya, Japan, May 1995, pp. 215–220.

[6] R. Williams and P. Gallina, "Planar cable-direct-driven robots: Design for wrench exertion," *J. Intell. Robot. Syst.*, vol. 35, pp. 203–219, 2002.

[7] A. Fattah and S. Agrawal, "Workspace and design analysis of cable-suspended planar parallel robots," in *Proc. ASME Mechanisms Conf.*, Montreal, QC, Canada, Sep. 2002, DETC2002/MECH-34 330.

[8] R. Verhoeven, M. Hiller, and S. Tadokoro, "Workspace, stiffness, singularities and classification of tendon-driven stewart-platforms," in *Proc. 6th Int. Symp. Advances in Robot Kinematics*, Strobl, Austria, 1998, pp. 105–114.

[9] Y. Shen, H. Osumi, and T. Arai, "Manipulability measures for multi-wire driven parallel mechanisms," in *Proc. IEEE Int. Conf. Industrial Technology*, Guangzhou, China, Dec. 1994, pp. 550–554.

[10] R. Roberts, T. Graham, and J. Trumpower, "On the inverse kinematics and statics of cable-suspended robots," in *Proc. IEEE Int. Conf. Computational Cybernetics and Simulation*, vol. 5, Orlando, FL, Oct. 1997, pp. 4291–4296.

[11] V.-D. Nguyen, "Constructing force-closure grasps," *Int. J. Robot. Res.*, vol. 7, no. 3, pp. 3–16, Jun. 1988.

[12] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC, 1994.

[13] W. Meyer, "Seven fingers allow force-torque closure grasps on any convex polyhedron," *Algorithmica*, vol. 9, no. 3, pp. 278–292, Mar. 1993.

[14] T. Mason, *Mechanics of Robotic Manipulation*. Cambridge, MA: MIT Press, 2001.

[15] L. Han, J. Trinkle, and Z. Li, "Grasp analysis as linear matrix inequality problems," *IEEE Trans. Robot. Automat.*, vol. 16, no. 6, pp. 663–674, Dec. 2000.

[16] E. Rimon and J. Burdick, "A configuration space analysis of bodies in contact—Part I. 1st order mobility," *Mech. Mach. Theory*, vol. 30, no. 6, pp. 897–912, 1995.

[17] ——, "A configuration space analysis of bodies in contact—Part II. 2nd order mobility," *Mech. Mach. Theory*, vol. 30, no. 6, pp. 913–928, 1995.

[18] R. Rockafeller, *Convex Analysis*, 10th ed. Princeton, NJ: Princeton Univ. Press, 1970.

[19] B. Mishra, J. Schwartz, and M. Sharir, "On the existence and synthesis of multifinger positive grips," *Algorithmica*, vol. 2, pp. 541–558, 1987.

[20] V.-D. Nguyen, "The synthesis of stable force-closure grasps," M.S. thesis, Dept. Elect. Eng. Comp. Sci., Massachusetts Inst. Technol., Cambridge, 1986.

[21] S. Tadokoro, Y. Murao, M. Hiller, R. Murata, H. Kohkawa, and T. Matsushima, "A motion base with 6-DOF by parallel cable drive architecture," *IEEE/ASME Trans. Mechantronics*, vol. 7, no. 2, pp. 115–123, Jun. 2002.

[22] J. Ponce, S. Sullivan, J.-D. Boissonat, and J.-P. Merlet, "On characterizing and computing three- and four-finger force-closure grasps of polyhedral objects," in *Proc. IEEE Int. Conf. Robotics and Automation*, Atlanta, GA, May 1993, pp. 821–827.

Dynamic Force Distribution in Multifingered Grasping by Decomposition and Positive Combination

Yu Zheng and Wen-Han Qian

***Abstract*—This paper presents a general algorithm for computing the optimal dynamic force distribution in multifingered grasping. It consists of two phases. In the offline phase, we select a spanning set for the required dynamic resultant wrench and find a corresponding spanning set for the total contact force. Then, in the online phase, the total contact force is obtained by decomposition of the resultant wrench into the former spanning set and a coefficient vector followed by positive combination of the latter spanning set with the vector. To make the online computation as simple as possible, iterative operation is executed offline and only arithmetic operation is employed online. To improve the grasping quality, the two spanning sets are selected elaborately.**

***Index Terms*—Contact constraint, dexterous robot hand, dynamic force distribution (DFD), multifingered grasps, optimal contact force.**

Nomenclature

m_0	Number of frictionless point contacts (FPC).
m_f	Number of point contacts with friction (PCwF).
m_s	Number of soft finger contacts (SFC).
m	Total number of contacts, $m = m_0 + m_f + m_s$.
$\boldsymbol{f}_i \in \mathbb{R}^{q_i}$	Contact force at contact i, $i = 1, 2, \ldots, m$,.
q_i	Dimension of the ith contact force space, equal to one, three, and four for FPC, PCwF, and SFC, respectively.
$\boldsymbol{f} \in \mathbb{R}^q$	Total contact force, $\boldsymbol{f} = [\boldsymbol{f}_1^T \ \cdots \ \boldsymbol{f}_i^T \ \cdots \ \boldsymbol{f}_m^T]^T$.
q	Dimension of the total contact force space, $q = m_0 + 3m_f + 4m_s$.
$\boldsymbol{w} \in \mathbb{R}^6$	Resultant wrench, $\boldsymbol{w} = [\boldsymbol{p}^T \ \boldsymbol{m}^T]^T$.
$\boldsymbol{p} \in \mathbb{R}^3$	Resultant force.
$\boldsymbol{m} \in \mathbb{R}^3$	Resultant moment.
$\boldsymbol{G} \in \mathbb{R}^{6\times q}$	Grasp matrix.
$\boldsymbol{G}^+ \in \mathbb{R}^{q\times 6}$	Pseudoinverse of $\boldsymbol{G}$ or the right-inverse if $\boldsymbol{G}$ is full row rank.

Manuscript received May 7, 2004; revised November 15, 2004. This paper was recommended for publication by Associate Editor N. Sarkar and Editor I. Walker upon evaluation of the reviewers' comments. This work was supported by the National Natural Science Foundation of China under Grant 59685004.

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/TRO.2005.847609

$\boldsymbol{A} \in \mathbb{R}^{q\times h}$	Matrix whose columns constitute a basis for the null space of $\boldsymbol{G}$.
$\boldsymbol{x} \in \mathbb{R}^h$	An arbitrary vector.
h	Dimension of the null space of $\boldsymbol{G}$, $h = q - \mathrm{rank}(\boldsymbol{G})$.
f_{in}	Inward normal force at contact i.
f_{io}, f_{it}	Two tangential force components of $\boldsymbol{f}_i$.
f_{is}	Spin moment at SFC.
μ_i	Coulomb friction coefficient at contact i.
μ_{si}	Coefficient of spin moment for SFC with linear model (SFCl).
μ'_{si}	Coefficient of spin moment for SFC with elliptical model (SFCe).
f_i^U	Upper bound of the magnitude of $\boldsymbol{f}_i$.
S_w	A spanning set for $\boldsymbol{w}$.
$\boldsymbol{w}_j \in \mathbb{R}^6$	Element of $S_w (j = 1, 2, \ldots, l)$.
$\boldsymbol{c} \in \mathbb{R}^l$	Coefficient vector.
$c_j \in \mathbb{R}$	The jth element of $\boldsymbol{c}$.
S_f	A spanning set for $\boldsymbol{f}$ with respect to S_w.
$\boldsymbol{f}_j \in \mathbb{R}^q$	A feasible total contact force for $\boldsymbol{w}_j$, used as an element candidate of S_f.
$\boldsymbol{f}_j^- \in \mathbb{R}^q$	A feasible total contact force for $-\boldsymbol{w}_j$, used as an element candidate of S_f.
$\| \bullet \|_1$	1-norm of a vector.
$\| \bullet \|$	2-norm of a vector.

I. Introduction

The major function of multifingered robot hands is manipulating objects *dexterously*, which needs to compute dynamic force distribution (DFD) at contacts *in real time*.

DFD is a topic of finding the optimal $\boldsymbol{f}$ for a required $\boldsymbol{w}$, subject to static equivalence and contact constraints. There are many feasible solutions for $\boldsymbol{f}$ for a given $\boldsymbol{w}$. Among them, we look for an optimal one, whose inclination angle and magnitude are smaller for the sake of grasping stability and material strength (also reducing actuator power as a minor reason), respectively.

As a foundation of this problem (Fig. 1), Salisbury and Roth [1] decomposed the contact force exerted on the grasped object into two parts (a manipulation force and an internal force). Such decomposition was illustrated by Yoshikawa and Nagai [2] in more detail.

Kerr and Roth [3] proposed a linear programming (LP) algorithm. Later, Cheng and Orin [4]–[6] developed a compact-dual LP algorithm. Although the latter facilitates real-time optimization, still undesirable temporal discontinuities may be introduced into the solution, which is the most important deficiency in all LP algorithms, as shown in [7].

To overcome the deficiency, Nahon and Angeles [7], [8] presented a quadratic programming (QP) algorithm. Compared with LP schemes, it shows higher performance in computation speed and solution quality, but the linearized friction model is still adopted. As a result, both the QP and LP algorithms are inapplicable to soft finger grasping, where the friction constraint cannot be linearized up to now. Based on a Lagrange multiplier method, Nakamura *et al.* [9] brought forward a nonlinear programming (NLP) algorithm. Nevertheless, the applicability to real-time situations was not mentioned.

To speed up the online computation, Park and Starr [10], Maekawa *et al.* [11], and Zuo and Qian [12] fixed the direction of internal forces strictly inside the friction cones and left only the magnitude of internal forces to be adjusted in real time. Total computation of contact forces is divided into offline iterative search for the optimal internal force direction and online analytical computation of the total contact force. Moreover, the method [12] covers soft finger contact. However, the optimal direction of internal forces is fixed; hence, the final contact force is not so optimal, because the optimal direction usually changes along with the external wrench. Besides, using interaction force rather than internal force [10] may limit the search space of the optimal internal force direction, since the interaction force space is equal to or *less* than the internal force space [13].

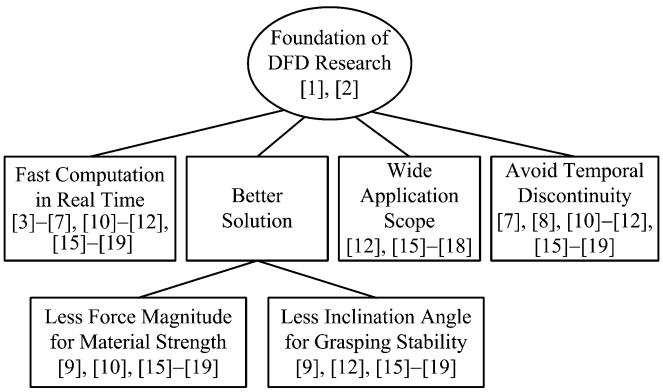

Fig. 1. Review and goals of DFD research.

Brockett [14] first introduced the algebra and group theory into robotic manipulation research. Buss *et al.* [15] transformed the contact constraints into the positive definiteness of a linearly constrained matrix. Then the task of grasping force optimization was formulated as an optimization problem on the smooth manifold of linearly constrained positive definite matrices, for which efficient projected gradient flow algorithms [15]–[17] were developed. Han *et al.* [18] further cast the contact constraints into linear matrix inequalities (LMIs) and formulated the force optimization problem as a convex optimization problem involving LMIs. Thereupon, highly efficient algorithms with polynomial time complexity can be directly applied. The theories based on manifolds and groups give a new approach to these problems and well-developed algorithms can be expediently used. However, the methods still require many iterations. To reduce online iterations, Remond [19] converted a constrained optimization problem into an unconstrained one.

Among the algorithms mentioned above, some [3]–[7], [10]–[12], [15]–[19] are claimed to be suitable for real-time control. Three [10]–[12] of them forcibly fix the direction of internal forces and restrict the contact force optimization. The others need large numbers of online iterations, so they are essentially numerical. Besides, some other analytical or suboptimal methods [20], [21] were proposed for real-time control, but they work only in some particular occasions.

Succeeding the idea in [12], in this paper, we also partition the total computation into two phases and shift as much computation as possible from online to offline, but the approach is new. In the offline phase (see Fig. 2), instead of searching the optimal direction of internal forces, we select an appropriate spanning set S_w for $\boldsymbol{w}$ and search the corresponding optimal spanning set S_f for $\boldsymbol{f}$, subject to stability and strength criteria. Then, in the online phase, $\boldsymbol{f}$ can be readily computed by decomposition of $\boldsymbol{w}$ into S_w and a coefficient vector $\boldsymbol{c}$, followed by a positive combination of $\boldsymbol{c}$ and S_f. $\boldsymbol{c}$ serves as an interconnector between $\boldsymbol{w}$ and $\boldsymbol{f}$ and varies according to the variation of $\boldsymbol{w}$. The interconnection through decomposition and positive combination keeps static equivalence and contact constraints for $\boldsymbol{w}$ and $\boldsymbol{f}$ if each element of S_w and its corresponding element of S_f satisfy these conditions. In addition, by proper selection of S_w, $\boldsymbol{f}$ can inherit the optimality from S_f. In this way, an algorithm with the simplest online computation until now as well as a superior solution quality is proposed. It also overcomes the temporal discontinuity due to continuous operation and can be applied to all three types of contact. Thus, all of the goals in Fig. 1 have been achieved.

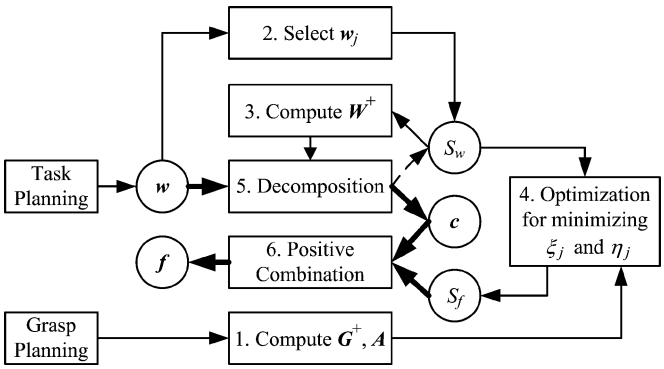

Fig. 2. Graph showing the algorithm procedure. Encircled are five principal data $\boldsymbol{w}$, S_w, $\boldsymbol{c}$, S_f, and $\boldsymbol{f}$, where $\boldsymbol{w}$—required resultant wrench; S_w—spanning set for $\boldsymbol{w}$; $\boldsymbol{c}$—coefficient vector; S_f—spanning set for $\boldsymbol{f}$; $\boldsymbol{f}$—final contact force. Only the thick arrows are performed online. All of the others are accomplished offline. Actually, the decomposition is carried out by $\boldsymbol{c} = \boldsymbol{W}^+\boldsymbol{w}$. Hence, the arrow from Step 5 to S_w is dashed.

In existing literature, there are decompositions of the contact force [1], [2], [22]–[26]. So far, however, none refers to the decomposition of the resultant wrench space for simplifying DFD in real time, as we are presenting. Unlike the numerical algorithms [3]–[9], [15]–[19], ours needs *no iteration during online computation*. Its computational complexity grows *linearly* with the contact number, while those with iteration grow very rapidly, such as *exponentially* [15]–[17] or *polynomially* [18]. Compared with the algorithms [10]–[12], we do not restrict the internal force in one direction and our online computation is much simpler. Many algorithms mentioned above do not cover soft finger contact, but ours is generally applicable.

For the organization of the rest of this paper, Section II introduces fundamentals and criteria of DFD. Section III explains the basic idea of the algorithm. Section IV discusses how to select the spanning sets. Section V depicts the algorithm procedure. A numerical example and conclusions are given in Sections VI and VII, respectively.

II. Formulations and Criteria

A. Basic Formulations

Consider an object grasped by a multifingered robot hand, which makes m contacts.

The general static equivalence equation for the grasped object is

$$\boldsymbol{w} = \boldsymbol{G}f. \tag{1}$$

The general solution to (1) can be expressed by

$$\boldsymbol{f} = \boldsymbol{G}^+\boldsymbol{w} + \boldsymbol{A}x. \tag{2}$$

The first term on the right-hand side of (2) is a particular solution called the manipulation force, which has an effect on $\boldsymbol{w}$, and the second is a homogeneous solution called the internal force, which does not affect $\boldsymbol{w}$ but can be changed to optimize the contact force.

In order to avoid separation and slip at contact, $\boldsymbol{f}$ must satisfy the following contact constraints [12], [15]–[18], [27], [28]:

$$\text{FPC}: f_{in} \geq 0, \quad f_{io} = f_{it} = 0 \tag{3}$$

$$\text{PCwF}: f_{in} \geq 0, \quad \sqrt{f_{io}^2 + f_{it}^2} \leq \mu_i f_{in} \tag{4}$$

$$\text{SFCl}: f_{in} \geq 0, \quad \frac{\sqrt{f_{io}^2 + f_{it}^2}}{\mu_i} + \frac{|f_{is}|}{\mu_{si}} \leq f_{in} \tag{5}$$

$$\text{SFCe}: f_{in} \geq 0, \quad \sqrt{\frac{f_{io}^2 + f_{it}^2}{\mu_i^2} + \frac{f_{is}^2}{\mu'^2_{si}}} \leq f_{in}. \tag{6}$$

This paper adopts SFCe, but the proposed algorithm can be applied to SFCl as well. A total contact force $\boldsymbol{f}$ is said to be *feasible* if it satisfies the static equivalence equation (1) as well as the contact constraints (3), (4), and (6).

In addition, the magnitude of $\boldsymbol{f}_i$ must be within the strength limit so that the grasping mechanism and the grasped object can bear

$$\text{FPC}: f_{in} \leq f_i^U \tag{7}$$

$$\text{PCwF}: \sqrt{f_{in}^2 + f_{io}^2 + f_{it}^2} \leq f_i^U \tag{8}$$

$$\text{SFC}: \sqrt{f_{in}^2 + f_{io}^2 + f_{it}^2 + \mu_i^2 f_{is}^2/\mu'^2_{si}} \leq f_i^U. \tag{9}$$

Noticing the dimensional difference between f_{is} and the other force components, we express the strength limit for SFC *equivalently*.

B. Optimization Criteria and Problems

Referring to Fig. 1, there are mainly two criteria for optimizing contact forces, which are different from those for planning optimal grasps such as [29].

1) Maximum Stability Criterion: To keep contact stability without separation and slip, it requires that each contact force be close to the inward normal, which can be evaluated by

$$\xi = \max_{1 \leq i \leq m} \xi_i \tag{10}$$

where ξ_i has one of the following forms:

$$\text{FPC}: \xi_i = \text{N.A.} \tag{11}$$

$$\text{PCwF}: \xi_i = \sqrt{f_{io}^2 + f_{it}^2}/\mu_i f_{in} \tag{12}$$

$$\text{SFC}: \xi_i = \sqrt{f_{io}^2 + f_{it}^2 + \mu_i^2 f_{is}^2/\mu'^2_{si}}/\mu_i f_{in}. \tag{13}$$

ξ_i is the ratio of the actual or equivalent tangential force to its maximum value before slip impending at contact i. At FPC, the fingertip exerts only a normal force and no tangential force is allowed, and therefore ξ_i is nonapplicable. In a sense, ξ is a measure of risk concerning grasping stability with its upper bound equal to unity. The less ξ is, the more stable it is.

2) Minimum Force Criterion: It often needs to consider that the magnitude of the contact force must be below the strength limit, which can be *practically* formulated as

$$\eta = \max_{1 \leq i \leq m} \eta_i \tag{14}$$

where η_i has one of the following forms:

$$\text{FPC}: \eta_i = f_{in}/f_i^U \tag{15}$$

$$\text{PCwF}: \eta_i = \sqrt{f_{in}^2 + f_{io}^2 + f_{it}^2}/f_i^U \tag{16}$$

$$\text{SFC}: \eta_i = \sqrt{f_{in}^2 + f_{io}^2 + f_{it}^2 + \mu_i^2 f_{is}^2/\mu'^2_{si}} \Big/ f_i^U. \tag{17}$$

Thus, η is a measure of risk concerning material strength with its upper bound equal to unity. A smaller η is more favorable.

With the aforesaid two criteria, we formulate the following.

Optimization Problem 1:

$$\text{Minimize } \eta \text{ subject to } (2),\ 0 \leq \eta \leq 1,\ \text{and } 0 \leq \xi \leq 1. \tag{18}$$

Optimization Problem 2:

$$\text{Minimize } \xi \text{ subject to } (2),\ 0 \leq \xi \leq 1,\ \text{and } 0 \leq \eta \leq 1. \tag{19}$$

Optimization Problem 3:

$$\text{Minimize } \eta \text{ subject to } (2),\ 0 \leq \xi \leq 1,\ \text{and } \eta/\xi = K \tag{20}$$

where K is a constant.

All of the above problems can be computed *directly* without difficulty, but their computation costs are too expensive for real-time control. The result so obtained is called the *optimal solution*. Particularly,

the solution of Optimization Problem 1 is called the *minimum contact force*. In what follows, we seek a decomposition and positive combination (DPC) algorithm which aims at a solution best approaching the optimal solution with a very low online computation cost.

III. Computing Feasible Contact Forces

A feasible $\boldsymbol{f}$ must satisfy two conditions: the static equivalence (1) and the contact constraints (3), (4), and (6). This section explores an easy way to fulfilling the conditions.

A. Attaining the Static Equivalence by Linear Combination

Suppose that S_w is a spanning set for $\boldsymbol{w}$, namely

$$S_w = \{\boldsymbol{w}_j \in \mathbb{R}^6 \,|\, j = 1, 2, \ldots, l\} \tag{21}$$

such that

$$\boldsymbol{w} = \sum_{j=1}^{l} c_j \boldsymbol{w}_j = \boldsymbol{W} c \tag{22}$$

where $\boldsymbol{W} = [\boldsymbol{w}_1\ \boldsymbol{w}_2\ \cdots\ \boldsymbol{w}_l] \in \mathbb{R}^{6\times l}, \boldsymbol{c} = [c_1\ c_2\ \cdots\ c_l]^T \in \mathbb{R}^l$. The solution to (22) may not be unique. We prefer

$$\boldsymbol{c} = \boldsymbol{W}^+ \boldsymbol{w} \tag{23}$$

where $\boldsymbol{W}^+$ is the pseudoinverse of $\boldsymbol{W}$. Since $\boldsymbol{W}^+$ can be computed offline, merely a matrix multiplication can compute $\boldsymbol{c}$ online.

Let $\boldsymbol{f}_j = [\boldsymbol{f}_{1,j}^T\ \cdots\ \boldsymbol{f}_{i,j}^T\ \cdots\ \boldsymbol{f}_{m,j}^T]^T$ be a total contact force for the resultant wrench $\boldsymbol{w}_j$, i.e.,

$$\boldsymbol{w}_j = \boldsymbol{G} f_j. \tag{24}$$

Combining (1), (22), and (24) yields

$$\boldsymbol{f} = \sum_{j=1}^{l} c_j \boldsymbol{f}_j \tag{25}$$

which is a total contact force for $\boldsymbol{w}$. Alternatively, if $\boldsymbol{w}_j$ and $\boldsymbol{f}_j$ satisfy the static equivalence equation (24), then their superpositions $\boldsymbol{w}$ and $\boldsymbol{f}$ by (22) and (25) also satisfy the static equivalence equation (1).

B. Preserving the Contact Constraints by Nonnegative Coefficients

As originally stated in [30], all of the total contact forces satisfying the contact constraints constitute a convex cone. Let $\boldsymbol{f}_j$ for $j = 1, 2, \ldots, l$ satisfy the contact constraints, and then $\boldsymbol{f}_j$ lies in the convex cone. From convex analysis, every positive combination (linear combination with nonnegative coefficients) of $\boldsymbol{f}_j, j = 1, 2, \ldots, l$ also lies in the convex cone. Thus, if $c_j \geq 0$ for all $j = 1, 2, \ldots, l$, $\boldsymbol{f}$ computed by (25) will satisfy the contact constraints as well.

C. Making the Coefficients All Nonnegative

To sum up, a positive combination of feasible $\boldsymbol{f}_j, j = 1, 2, \ldots, l$ is a feasible $\boldsymbol{f}$. It remains to be dealt with how to always obtain a nonnegative $\boldsymbol{c}$ from (23).

This problem can be settled by a positizing process as follows:

$$\text{If } c_j < 0, \quad \text{then } c_j \leftarrow -c_j, \quad \boldsymbol{w}_j \leftarrow -\boldsymbol{w}_j, \quad \text{and} \quad \boldsymbol{f}_j \leftarrow \boldsymbol{f}_j^-, \quad \text{otherwise leave them unchanged}$$

where $\leftarrow$ means assignment. The process does not destroy the static equivalence between (22) and (25). Therefore, the feasible $\boldsymbol{f}$ can be computed by (25) as a positive combination indeed.

To complete this section, we construct

$$S_f = \{\boldsymbol{f}_j \text{ or } \boldsymbol{f}_j^- \in \mathbb{R}^q \,|\, j = 1, 2, \ldots, l\} \tag{26}$$

where $\boldsymbol{f}_j$ and $\boldsymbol{f}_j^-$ are alternatively used in (25). S_f is a spanning set with respect to S_w, which can positively span the feasible $\boldsymbol{f}$.

IV. Optimization of the Contact Force

As *feasibility* is transmitted from S_f (with respect to S_w) to $\boldsymbol{f}$ (with respect to $\boldsymbol{w}$) by positive combination, we would investigate transmission of *optimality* from S_f to $\boldsymbol{f}$ if $\boldsymbol{f}_j$ and $\boldsymbol{f}_j^-$ are optimized according to a certain criterion. Herein we assume the following.

1) The direction of $\boldsymbol{w}$ is time-varying. Otherwise, in a very simple case when $\boldsymbol{w}$ changes only in magnitude with a fixed direction, it can be spanned by a vector $\boldsymbol{w}_0$. Let $\boldsymbol{f}_0$ and $\boldsymbol{f}_0^-$ be the optimal contact forces for $\boldsymbol{w}_0$ and $-\boldsymbol{w}_0$, respectively. Then, the optimal $\boldsymbol{f}$ for $\boldsymbol{w}$ can be formulated directly as

$$\boldsymbol{f} = \begin{cases} \left(\boldsymbol{w}^T \boldsymbol{w}_0 / \boldsymbol{w}_0^T \boldsymbol{w}_0\right) \boldsymbol{f}_0, & \text{for } \boldsymbol{w}^T \boldsymbol{w}_0 \geq 0 \\ -\left(\boldsymbol{w}^T \boldsymbol{w}_0 / \boldsymbol{w}_0^T \boldsymbol{w}_0\right) \boldsymbol{f}_0^-, & \text{for } \boldsymbol{w}^T \boldsymbol{w}_0 < 0 \end{cases}.$$

2) $c_j \geq 0$ for $j = 1, 2, \ldots, l$, except otherwise indicated.

A. Upper Bounds of ξ and η

The upper bounds of ξ and η of $\boldsymbol{f}$ by (25) are evaluated, respectively, in (27)–(29) and (30)–(32), shown at the bottom of the next page.

Substituting (27)–(29) into (10), we have

$$\xi = \max_{1\leq i\leq m} \xi_i \leq \max_{1\leq i\leq m} \left\{ \max_{1\leq j\leq l} \xi_{i,j} \right\} = \max_{1\leq j\leq l} \xi_j \tag{33}$$

where we define $\xi_j = \max_{1\leq i\leq m} \xi_{i,j}$. This means that the overall ξ is bounded above by $\max_{1\leq i\leq m} \xi_i$, which in turn is not greater than $\max_{1\leq j\leq l} \xi_j$. Hence ξ can be reduced by reducing $\max_{1\leq j\leq l} \xi_j$.

Substituting (30)–(32) into (14), we have

$$\eta = \max_{1\leq i\leq m} \eta_i \leq \max_{1\leq i\leq m} \{\|\boldsymbol{c}\|_1 \max_{1\leq j\leq l} \eta_{i,j}\} = \|\boldsymbol{c}\|_1 \max_{1\leq j\leq l} \eta_j \tag{34}$$

where we define $\eta_j = \max_{1\leq i\leq m} \eta_{i,j}$. Different from ξ, the overall η is not bounded above by $\max_{1\leq j\leq l} \eta_j$ but by $\|\boldsymbol{c}\|_1 \max_{1\leq j\leq l} \eta_j$. Hence, η cannot be reduced by only reducing $\max_{1\leq j\leq l} \eta_j$.

B. Selecting S_w for Optimization

There could be various S_w for $\boldsymbol{w}$. After a number of numerical tests, we would like to recommend the following one:

$$S_w = \{\alpha_1 \boldsymbol{w}(t_L), \alpha_2 \boldsymbol{w}(t_M), \alpha_3 \boldsymbol{w}(t_R), \boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_6\} \tag{35}$$

where t_L and t_R are the endpoints of the time segment in which S_w is used, t_M is a point between t_L and $t_R (t_L < t_M < t_R)$, $\alpha_1,\ \alpha_2, \alpha_3$ are weights, and $\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_6$ are the elements of the standard basis for $\mathbb{R}^6$. This kind of S_w has several pleasing properties.

First, the parameters $\alpha_1, \alpha_2, \alpha_3$, and t_M can be selected by optimization so that $\boldsymbol{f}$ computed by (25) best approaches its optimal value. Noting that η from (25) is more difficult to be restricted than ξ, we propose a principle for optimizing these parameters as follows:

$$\begin{aligned} &\text{Minimize } \max_{1\leq i\leq m} \sum_{k=1}^{h} \left(\eta_{i,k}^* - \eta_i(\alpha_1, \alpha_2, \alpha_3, t_M, t_k)\right)^2 \\ &\text{subject to } \alpha_1, \alpha_2, \alpha_3 \geq 0, \text{ and } t_L < t_M < t_R \end{aligned} \tag{36}$$

where $\eta_{i,k}^* = \eta_i^*(t_k), k = 1, 2, \ldots, h$ are the values of η_i obtained by directly solving Optimization Problem 1 at $t_1, t_2, \ldots, t_h (t_L = t_1 < t_2 < \cdots < t_h = t_R)$. In solving (36), $\boldsymbol{f}_j$ or $\boldsymbol{f}_j^-$ of S_f is taken as the minimum contact forces. We seek $[\alpha_1^*, \alpha_2^*, \alpha_3^*, t_M^*]$ for which the maximum sum of squared deviations for $i = 1, 2, \ldots, m$ is minimal. As zero is a lower bound of the objective function, the solution exists.

Second, by properly selecting the weights $\alpha_1, \alpha_2, \alpha_3$, the values η and ξ of $\boldsymbol{f}$ from (25) can be very close to their optimal values at t_L, t_M, t_R.

Third, the smallest nonzero singular value of the matrix $\boldsymbol{W} = [\alpha_1 \boldsymbol{w}(t_L)\ \alpha_2 \boldsymbol{w}(t_M)\ \alpha_3 \boldsymbol{w}(t_R) \boldsymbol{e}_1 \boldsymbol{e}_2 \cdots \boldsymbol{e}_6]$ is unity (see the Appendix for details). This is helpful to slow down the variation of $\boldsymbol{c}$ and the increase of η. Indeed, suppose that $\boldsymbol{W}$ has singular value decomposition $\boldsymbol{U}D\boldsymbol{V}^T$. Substituting $\boldsymbol{W} = \boldsymbol{U}D\boldsymbol{V}^T$ into (23) yields

$$\begin{aligned} \|\boldsymbol{c}\| &= \|\boldsymbol{W}^+ \boldsymbol{w}\| = \|\boldsymbol{V}D^+\boldsymbol{U}^T\boldsymbol{w}\| = \|\boldsymbol{D}^+\boldsymbol{\nu}\| \\ &= \sqrt{\frac{v_1^2}{\sigma_1^2} + \frac{v_2^2}{\sigma_2^2} + \cdots + \frac{v_r^2}{\sigma_r^2}} \end{aligned} \tag{37}$$

where $\sigma_1, \sigma_2, \ldots, \sigma_r$ are the nonzero singular values of $\boldsymbol{W}$; $v_1, v_2, \ldots, v_r$ are the elements of $\boldsymbol{\nu} = \boldsymbol{U}^T\boldsymbol{w}$ and $\|\boldsymbol{\nu}\| = \|\boldsymbol{w}\|$. From (37), if $\sigma_1, \sigma_2, \ldots, \sigma_r$ are small, a small variation of $\boldsymbol{w}$ might result in a large variation of $\boldsymbol{c}$ and lead to a rapid increase of η at the points other than t_L, t_M, t_R.

Finally, the set S_w can span the entire resultant wrench space $\mathbb{R}^6$.

C. Calculating S_f for Optimization

As the optimal solution of $\boldsymbol{f}$ is computed by solving Optimization Problems 1, 2, or 3 with respect to $\boldsymbol{w}$, each element $\boldsymbol{f}_j$ of S_f can be obtained in the same way with respect to $\boldsymbol{w}_j$ of S_w. To specify ξ or η of $\boldsymbol{f}$ calculated online by (23) and (25), we can adjust ξ_j or η_j of $\boldsymbol{f}_j$ offline. Actually, only the maxima of ξ and η in the whole time segment between t_L and t_R determine the grasping quality

$$\bar{\xi} = \max_{t_L \le t \le t_R} \xi \quad \text{and} \quad \bar{\eta} = \max_{t_L \le t \le t_R} \eta. \tag{38}$$

Still, either value of them can be specified by adjusting ξ or η in computing S_f. Thus, by replacing ξ and η with $\bar{\xi}$ and $\bar{\eta}$ in (18)–(20), the optimization problems can be applied to DPC as well.

D. Segmentation

In general, $\boldsymbol{w}$ may vary significantly with time t, but S_w is static. For a better approach, we divide the $\boldsymbol{w} - t$ curve into a sufficient number of segments and apply different spanning sets to them.

If the magnitude of $\boldsymbol{w}$ can be measured by a single scalar $s \in \mathbb{R}$, then t can be segmented according to the $s - t$ curve. The natural choice of s seems to be $\|\boldsymbol{w}\|$. However, $\|\boldsymbol{w}\|$ does not give any physical meaning because the components $\boldsymbol{p}$ and $\boldsymbol{m}$ of $\boldsymbol{w}$ have different dimensions N and $N \cdot m$. As an alternative, we compute a $\eta^* - t$ curve where η^* is obtained by directly solving Optimization Problem 1. Then, we divide the $\eta^* - t$ curve at the points where $\eta^*(t)$ are local minima. If $\eta^*(t)$ varies slightly, the $\boldsymbol{w} - t$ curve can be divided equidistantly for convenience.

It should be pointed out that segmentation might bring on discontinuity at segment points. Nevertheless, the jump of $\eta(t)$ at segment points is negligible for $\eta(t)$ is approaching $\eta^*(t)$, and it does not affect the validity of the proposed algorithm.

V. Algorithm Procedure

Referring to Fig. 2, the algorithm is implemented in two phases.

In the offline phase:

Step 1) Compute the matrices $\boldsymbol{G}^+$ and $\boldsymbol{A}$.

Step 2) Select the spanning set S_w for $\boldsymbol{w}$.

Step 3) Construct the matrix $\boldsymbol{W}$ and compute $\boldsymbol{W}^+$.

Step 4) Calculate the spanning set S_f with respect to S_w under the desired optimization criterion.

In the online phase:

Step 5) *(Resultant wrench deomposition)*: Compute the coefficient vector $\boldsymbol{c}$ by (23).

Step 6) *(Positive combination)*: Positize the decomposition of $\boldsymbol{w}$ and compute the total contact force $\boldsymbol{f}$ by (25).

The online phase can be programmed as

$$\text{FPC}: \xi_i = \text{N.A.} \tag{27}$$

$$\begin{aligned} \text{PCwF}: \xi_i &= \sqrt{\left(\sum_{j=1}^{l} c_j f_{io,j}\right)^2 + \left(\sum_{j=1}^{l} c_j f_{it,j}\right)^2} \Big/ \mu_i \sum_{j=1}^{l} c_j f_{in,j} \le \sum_{j=1}^{l} c_j \sqrt{f_{io,j}^2 + f_{it,j}^2} \Big/ \mu_i \sum_{j=1}^{l} c_j f_{in,j} \\ &\le \max_{1 \le j \le l} \left(\sqrt{f_{io,j}^2 + f_{it,j}^2} \Big/ \mu_i f_{in,j} \right) \le \max_{1 \le j \le l} \xi_{i,j} \end{aligned} \tag{28}$$

$$\begin{aligned} \text{SFC}: \xi_i &= \frac{\sqrt{\left(\sum_{j=1}^{l} c_j f_{io,j}\right)^2 + \left(\sum_{j=1}^{l} c_j f_{it,j}\right)^2 + \frac{\mu_i^2}{\mu'^2_{si}} \left(\sum_{j=1}^{l} c_j f_{is,j}\right)^2}}{\mu_i \sum_{j=1}^{l} c_j f_{in,j}} \le \sum_{j=1}^{l} c_j \sqrt{f_{io,j}^2 + f_{it,j}^2 + \mu_i^2 f_{is,j}^2 / \mu'^2_{si}} \Big/ \mu_i \sum_{j=1}^{l} c_j f_{in,j} \\ &\le \max_{1 \le j \le l} \left(\sqrt{f_{io,j}^2 + f_{it,j}^2 + \mu_i^2 f_{is,j}^2 / \mu'^2_{si}} \Big/ \mu_i f_{in,j} \right) \\ &= \max_{1 \le j \le l} \xi_{i,j} \end{aligned} \tag{29}$$

$$\text{FPC}: \eta_i = \frac{1}{f_i^U} \sum_{j=1}^{l} c_j f_{in,j} \le \|\boldsymbol{c}\|_1 \max_{1 \le j \le l} \eta_{i,j} \tag{30}$$

$$\text{PCwF}: \eta_i = \frac{1}{f_i^U} \sqrt{\left(\sum_{j=1}^{l} c_j f_{in,j}\right)^2 + \left(\sum_{j=1}^{l} c_j f_{io,j}\right)^2 + \left(\sum_{j=1}^{l} c_j f_{it,j}\right)^2} \le \frac{1}{f_i^U} \sum_{j=1}^{l} c_j \sqrt{f_{in,j}^2 + f_{io,j}^2 + f_{it,j}^2} \le \|\boldsymbol{c}\|_1 \max_{1 \le j \le l} \eta_{i,j} \tag{31}$$

$$\begin{aligned} \text{SFC}: \eta_i &= \frac{1}{f_i^U} \sqrt{\left(\sum_{j=1}^{l} c_j f_{in,j}\right)^2 + \left(\sum_{j=1}^{l} c_j f_{io,j}\right)^2 + \left(\sum_{j=1}^{l} c_j f_{it,j}\right)^2 + \frac{\mu_i^2}{\mu'^2_{si}} \left(\sum_{j=1}^{l} c_j f_{is,j}\right)^2} \\ &\le \frac{1}{f_i^U} \sum_{j=1}^{l} c_j \sqrt{f_{in,j}^2 + f_{io,j}^2 + f_{it,j}^2 + \mu_i^2 f_{is,j}^2 / \mu'^2_{si}} \le \|\boldsymbol{c}\|_1 \max_{1 \le j \le l} \eta_{i,j}. \end{aligned} \tag{32}$$

TABLE I
OPERATION COUNT OF ONLINE COMPUTATION FOR THE DYNAMIC FORCE DISTRIBUTION ALGORITHM

operation	additions	multiplications	comparisons	expression
$\boldsymbol{c}$	$5l$	$6l$	0	(23)
$\boldsymbol{f}$	$(l\text{-}1)m_0+3(l\text{-}1)m_f+4(l\text{-}1)m_s$	$lm_0+3lm_f+4lm_s$	l	(25)
total	$(l\text{-}1)m_0+3(l\text{-}1)m_f+4(l\text{-}1)m_s+5l$	$lm_0+3lm_f+4lm_s+6l$	l	

TABLE II
COMPARISON OF THE COMPUTATIONAL EFFICIENCY WITH THE PREVIOUS ALGORITHMS WITHOUT ONLINE ITERATION

algorithm	additions	multiplications	comparisons	etc.
Park and Starr* [10]	215	232	17	1 sin, 1 cos, 4 sqrt. 1 arctan, 2 abs.
Zuo and Qian [12]	$7m_0+22m_f+30m_s$	$8m_0+28m_f+39m_s$	$m_0+m_f+m_s$	$2(m_f+m_s)$ sqrt.
Ours	$(l\text{-}1)m_0+3(l\text{-}1)m_f$ $+4(l\text{-}1)m_s+5l$	$lm_0+3lm_f+4lm_s+6l$	l	none

*In the algorithm, m_0=0, m_f=3, m_s=0.

$\boldsymbol{f}=\boldsymbol{0}$; //Initialization
$\boldsymbol{c}=\boldsymbol{W}^{+}\boldsymbol{w}$; //Computing the coefficient vector
for $j=1,2,\ldots,l$ //Computing the total contact force
  if $c_j<0$
    $\boldsymbol{f}=\boldsymbol{f}-c_j\boldsymbol{f}_j^{-}$;
  else
    $\boldsymbol{f}=\boldsymbol{f}+c_j\boldsymbol{f}_j$;
end

Note: Steps 2)–4) are taken for each segment, respectively.

Table I lists the online operation count of the algorithm. According to Table I, the algorithm owns *linear* computational complexity. Table II compares the computational efficiency of the algorithm with the previous ones without iteration during online computation. Although the online operation count of our algorithm is more than that of Zuo and Qian [12] when l is large, our solution has better quality.

VI. NUMERICAL EXAMPLE

Fig. 3 shows an equilateral tetrahedron with vertices $\boldsymbol{\nu}_1(0,0,2\sqrt{2}),\boldsymbol{\nu}_2(2,0,0),\boldsymbol{\nu}_3(-1,-\sqrt{3},0)$, and $\boldsymbol{\nu}_4(-1,\sqrt{3},0)$. It is grasped by a four-fingered robot hand, which makes a SFC $C_1(0,0,0)$, a FPC $C_2(-2/3,0,2\sqrt{2}/3)$, and two PCwFs $C_3(1/3,\sqrt{3}/3,2\sqrt{2}/3)$ and $C_4(1/3,-\sqrt{3}/3,2\sqrt{2}/3)$ with the object surface. The coefficients of friction and spin moment $\mu=0.4,\mu'_s=0.4$ mm. The upper bounds of contact forces $f_1^U=1000$ N and $f_i^U=500$ N for $i=2,3,4$. The required resultant force and moment

$$\boldsymbol{p}=\begin{bmatrix}(40+10\sin 12\pi t)\sin\pi t\cos(\pi/6)-10\cos 12\pi t\sin(\pi/6)\\(40+10\sin 12\pi t)\cos\pi t\\(40+10\sin 12\pi t)\sin\pi t\sin(\pi/6)+10\cos 12\pi t\cos(\pi/6)\end{bmatrix}$$

$$\boldsymbol{m}=\begin{bmatrix}\cos[\sin(2\pi t)]\sin[\cos(\pi t)]\\-\cos[\sin(2\pi t)]\cos[\cos(\pi t)]\\\sin[\sin(2\pi t)]\end{bmatrix}.$$

The periods of $\boldsymbol{p}$ and $\boldsymbol{m}$ are 1/6 and 2 s, respectively. As a whole, the period of $\boldsymbol{w}$ is 2 s.

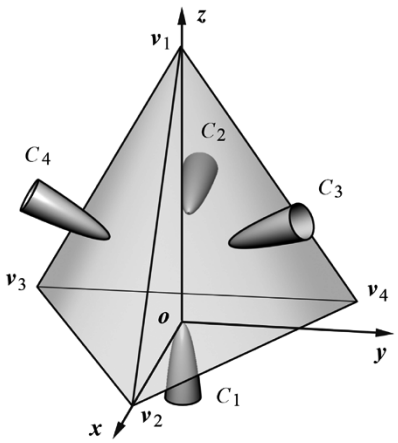

Fig. 3. Equilateral tetrahedron is grasped by a four-fingered gripper with a soft finger contact C_1, a frictionless contact C_2, and two frictional contacts C_3 and C_4. Each contact is at the center of a facet.

Due to the variation of $\boldsymbol{p}$, the η^*-t curve comprises 12 waves in 2 s, as depicted by the dashed line in Fig. 4. In accordance with its fluctuation, we divide the η^*-t curve into 12 segments at the points where η^* approaches local minima (the first column of Table III).

For each segment, we adopt the S_w given by (35) and determine the parameters $\alpha_1,\alpha_2,\alpha_3$, and t_M by (36) (the middle column of Table III), where the value η^* is taken for every 0.01 s. For instance, S_w for the first segment consists of the elements shown at the bottom of the page and $\boldsymbol{e}_1,\boldsymbol{e}_2,\ldots,\boldsymbol{e}_6$.

For material strength, we minimize $\bar{\eta}$ while calculating S_f. Fig. 4 depicts η and ξ of $\boldsymbol{f}$ in a period, where the optimal solution is obtained by directly solving Optimization Problem 1. As clearly reflected in Fig. 4(a), η computed by the algorithm is less than 0.72 and closely approaches optimal. The graceful result can be verified by the slight deviations listed in the last column of Table III. Fig. 4(b) shows that ξ of the optimal solution is equal to unity, while that computed by the DPC algorithm is below unity most often.

$$\alpha_1\boldsymbol{w}(t_L)=[-0.8848\quad 7.0788\quad 1.5326\quad 14.8915\quad -9.5617\quad 0]^T$$
$$\alpha_2\boldsymbol{w}(t_M)=[1.3114\quad 4.4816\quad -0.2737\quad 7.7209\quad -5.2228\quad 4.2147]^T$$
$$\alpha_3\boldsymbol{w}(t_R)=[22.1262\quad 64.8990\quad 17.8448\quad 138.8172\quad -106.1972\quad 156.0894]^T.$$

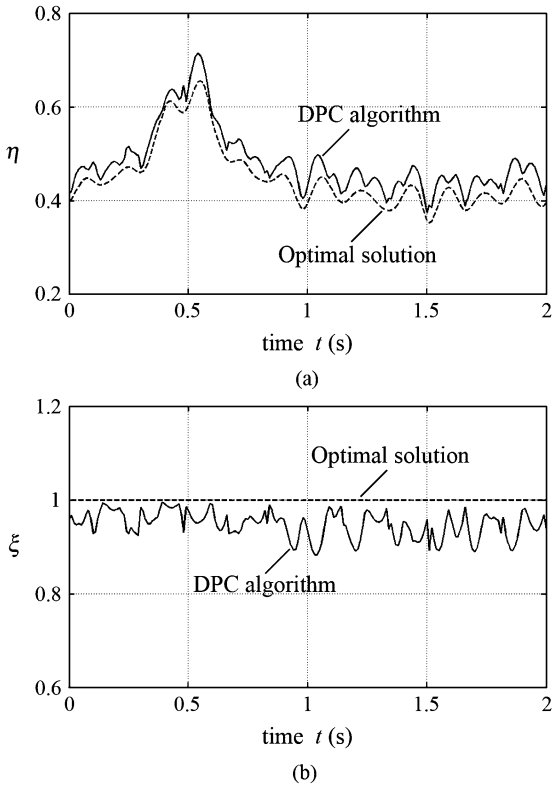

Fig. 4. Computed results of Optimization Problem 1 ($\bar{\xi} = 1$, minimize $\bar{\eta}$).

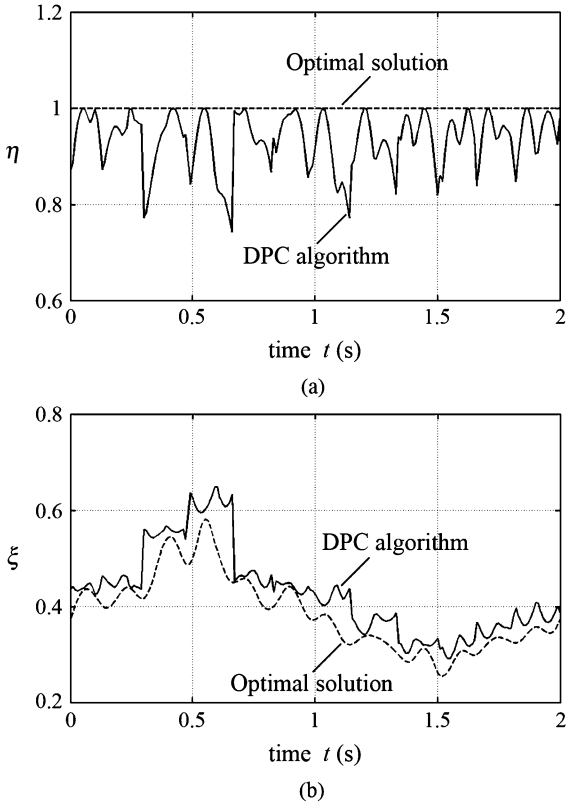

Fig. 5. Computed results of Optimization Problem 2 ($\bar{\eta} = 1$, minimize $\bar{\xi}$).

TABLE III
SELECTED POINTS, WEIGHTS, AND OTHER MAIN DATA FOR EACH SEGMENT

$[t_L, t_R]$	$[\alpha_1, \alpha_2, \alpha_3, t_M]$	minima of (36)
[0, 0.13]	[0.1770, 0.1023, 2.3433, 0.0698]	0.0078
[0.13, 0.30]	[0.1126, 0.1099, 0.0588, 0.2180]	0.0282
[0.30, 0.48]	[0.2961, 0.1511, 0.2636, 0.3836]	0.0109
[0.48, 0.67]	[0.0879, 0.0989, 0.1753, 0.6000]	0.0321
[0.67, 0.83]	[0.0481, 0.0903, 0.1981, 0.7461]	0.0274
[0.83, 0.98]	[0.1261, 0.0860, 0.1280, 0.8709]	0.0252
[0.98, 1.15]	[0.1365, 0.1321, 6.2554, 1.0853]	0.0248
[1.15, 1.34]	[0.0582, 0.1233, 0.1557, 1.2442]	0.0338
[1.34, 1.51]	[0.1032, 0.0882, 0.1024, 1.4100]	0.0213
[1.51, 1.66]	[0.0966, 0.1070, 0.1176, 1.5941]	0.0263
[1.66, 1.81]	[0.1488, 0.0883, 0.0737, 1.7399]	0.0200
[1.81, 2.0]	[0.1079, 0.1409, 0.1056, 1.9041]	0.0441

For grasping stability, we minimize $\bar{\xi}$. In this case, η and ξ are shown in Fig. 5, where the optimal solution comes from Optimization Problem 2 directly. Close to the optimal value, ξ obtained by the DPC algorithm is less than 0.6486 and η is not greater than that of the optimal solution. Compared with Fig. 4, Fig. 5 indicates that ξ is dramatically reduced and grasping stability is improved indeed; however, this improvement causes an increase in η.

Regarding grasping stability and material strength equally, we minimize $\bar{\eta}$ subject to $\bar{\eta} = \bar{\xi}$. Then, η and ξ are redrawn in Fig. 6, where the optimal solution is produced by Optimization Problem 3 with $K = 1$ directly. Fig. 6 shows that both η and ξ computed by the DPC algorithm are less than 0.85 and close to the optimal values.

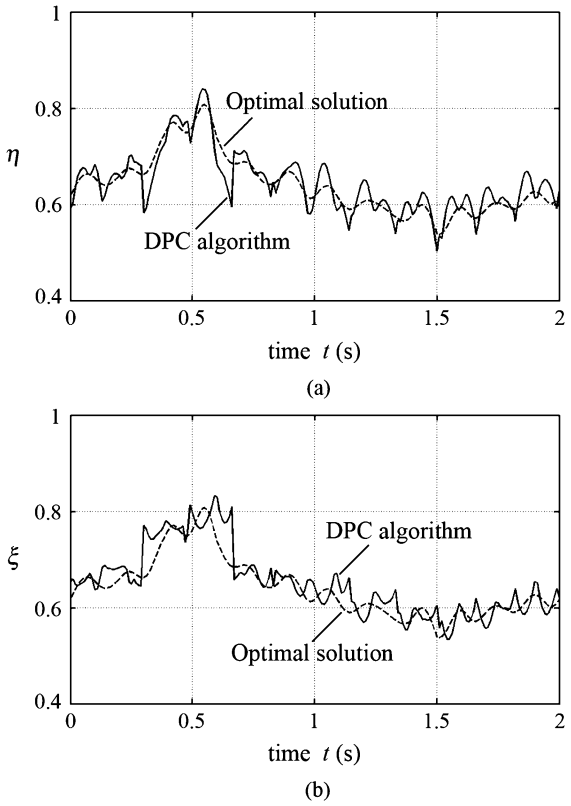

Fig. 6. Computed results of Optimization Problem 3 ($\bar{\eta} = \bar{\xi}$, minimize $\bar{\eta}$).

The total computation cost at each point is 136 additions, 153 multiplications, and nine comparisons. The algorithm has been implemented using Matlab and run on PCs with different CPUs. The execution time for a point is given in Table IV.

TABLE IV
EXECUTION TIME FOR A POINT ON PCS WITH DIFFERENT CPUS

CPU	Execution time of the DPC algorithm	Execution time of the optimal solution
Pentium III 800M	0.31 ms	480 ms
Pentium III 1G	0.24 ms	350 ms
Pentium IV 1.6G	0.17 ms	290 ms

VII. CONCLUSION AND FUTURE WORK

We present a new DFD algorithm for multifingered grasping (Fig. 2). Instead of online optimization, the technique of transmitting feasibility and optimality from the offline optimized S_f to $\boldsymbol{f}$ is addressed for the first time, which greatly reduces the online computation cost and keeps the superior solution quality. Making large strides toward all of the goals of DFD research (Fig. 1), hopefully the algorithm will be the first choice for the real-time control of multifingered robot hands.

Like other DFD algorithms and force-closure analysis, our algorithm also assumes that the friction coefficient is known and certain. The assumption is still a problem that deserves investigation in the future.

APPENDIX

$\boldsymbol{W}$ can be partitioned into two submatrices: $[\boldsymbol{W}_1 \mid \boldsymbol{I}]$ where the null space of $\boldsymbol{W}_1^T$ has nonzero elements and $\boldsymbol{I}$ is the identity matrix. This Appendix proves that the smallest nonzero singular value of $\boldsymbol{W}$ is irrelevant to $\boldsymbol{W}_1$ and equal to unity.

From matrix analysis, the nonzero singular values of $\boldsymbol{W}$ are the square roots of the nonzero eigenvalues of $\boldsymbol{W}^T\boldsymbol{W}$.

The nonzero eigenvalue λ of $\boldsymbol{W}^T\boldsymbol{W}$ must satisfy the equation $\boldsymbol{W}^T\boldsymbol{W}x = \lambda\boldsymbol{x}$ with a nonzero vector $\boldsymbol{x}$. From the partition of $\boldsymbol{W}$, we have

$$\left[\begin{array}{c|c} \boldsymbol{W}_1^T\boldsymbol{W}_1 & \boldsymbol{W}_1^T \\ \hline \boldsymbol{W}_1 & \boldsymbol{I} \end{array}\right]\left[\begin{array}{c} \boldsymbol{x}_1 \\ \hline \boldsymbol{x}_2 \end{array}\right] = \lambda\left[\begin{array}{c} \boldsymbol{x}_1 \\ \hline \boldsymbol{x}_2 \end{array}\right] \tag{39}$$

where $\boldsymbol{x}$ is partitioned into $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, i.e., $\boldsymbol{x} = [\boldsymbol{x}_1^T \mid \boldsymbol{x}_2^T]^T$.

Multiplying both sides of (39) on the left by $\left[\begin{array}{c|c} \boldsymbol{I} & -\boldsymbol{W}_1^T \\ \hline \boldsymbol{0} & \boldsymbol{I} \end{array}\right]$ yields

$$\left[\begin{array}{c|c} \boldsymbol{0} & \boldsymbol{0} \\ \hline \boldsymbol{W}_1 & \boldsymbol{I} \end{array}\right]\left[\begin{array}{c} \boldsymbol{x}_1 \\ \hline \boldsymbol{x}_2 \end{array}\right] = \lambda\left[\begin{array}{c} \boldsymbol{x}_1 - \boldsymbol{W}_1^T\boldsymbol{x}_2 \\ \hline \boldsymbol{x}_2 \end{array}\right]. \tag{40}$$

From $\lambda \neq 0$ and (40), it follows that

$$\boldsymbol{x}_1 = \boldsymbol{W}_1^T\boldsymbol{x}_2 \tag{41}$$

$$\boldsymbol{W}_1\boldsymbol{x}_1 = (\lambda - 1)\boldsymbol{x}_2. \tag{42}$$

From (41), it next follows that, if $\boldsymbol{x}_2 = \boldsymbol{0}$, then $\boldsymbol{x}_1 = \boldsymbol{0}$ and $\boldsymbol{x} = \boldsymbol{0}$. However, since $\boldsymbol{x} \neq \boldsymbol{0}$, we see that $\boldsymbol{x}_2 \neq \boldsymbol{0}$.

Multiplying both sides of (42) on the left by $\boldsymbol{x}_2^T$ and combining (41), we obtain

$$\|\boldsymbol{x}_1\| = (\lambda - 1)\|\boldsymbol{x}_2\|$$

which implies $\lambda \geq 1$.

On the other hand, it follows from (41) and (42) that $\lambda = 1$ is an eigenvalue of $\boldsymbol{W}^T\boldsymbol{W}$ and an eigenvector belonging to $\lambda = 1$ has the form of $\boldsymbol{x}_1 = \boldsymbol{0}$ and $\boldsymbol{x}_2$ being nonzero in the null space of $\boldsymbol{W}_1^T$.

Therefore, the smallest nonzero singular value of $\boldsymbol{W}$ is unity.

REFERENCES

[1] K. Salisbury and B. Roth, "Kinematic and force analysis of articulated mechanical hands," *ASME J. Mech. Transm. Autom. Des.*, vol. 105, pp. 35–41, Mar. 1983.

[2] 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.

[3] J. Kerr and B. Roth, "Analysis of multifingered hands," *Int. J. Robot. Res.*, vol. 4, no. 4, pp. 3–17, 1985.

[4] 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.

[5] ——, "Optimal force distribution in multiple-chain robotic system," *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 1, pp. 13–24, Jan./Feb. 1991.

[6] ——, "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.

[7] 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.

[8] ——, "Optimization of dynamic forces in mechanical hands," *ASME J. Mech. Des.*, vol. 113, no. 2, pp. 167–173, Jun. 1991.

[9] 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.

[10] 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.*, vol. 2, Scottsdale, AZ, May 1989, pp. 930–935.

[11] 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.

[12] B.-R. Zuo and W.-H. Qian, "A general dynamic force distribution algorithm for multifingered grasping," *IEEE Trans. Syst., Man, Cybern. B*, vol. 30, no. 1, pp. 185–192, Feb. 2000.

[13] ——, "On the equivalence of internal and interaction forces in multifingered grasping," *IEEE Trans. Robot. Autom.*, vol. 15, no. 5, pp. 934–941, Oct. 1999.

[14] R. W. Brockett, "Robotic manipulations and the product of exponentials formula," in *Proceedings of the International Symposium on Mathematical Theory of Networks and Systems*. Berlin, Germany: Springer-Verlag, 1984, pp. 120–127.

[15] 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.

[16] M. Buss and T. Schlegl, "Multi-fingered regrasping using online grasping force optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, Albuquerque, NM, Apr. 1997, pp. 998–1003.

[17] M. Buss, L. Faybusovich, and J. Moore, "Dikin-type algorithms for dextrous grasping force optimization," *Int. J. Robot. Res.*, vol. 17, no. 8, pp. 831–839, 1998.

[18] 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.

[19] C. Remond, V. Perdereau, and M. Drouin, "A multi-fingered hand control structure with online grasping force optimization," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatron.*, Como, Italy, Jul. 2001, pp. 804–809.

[20] Z. Ji and B. Roth, "Direct computation of grasping force for three-finger tip-prehension grasps," *ASME J. Mech., Transm., Autom. Des.*, vol. 110, 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] A. Bicchi, "Force distribution in multiple whole-limb manipulation," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, Atlanta, GA, May 1993, pp. 196–201.

[23] 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.

[24] 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.

[25] 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*, vol. 27, no. 2, pp. 255–260, Apr. 1997.

[26] Y. Zhang, W. A. Gruver, J. Li, and Q. Zhang, "Classification of grasps by robot hands," *IEEE Trans. Syst., Man, Cybern. B*, vol. 31, no. 3, pp. 436–444, Jun. 2001.

[27] R. D. Howe, I. Kao, and M. R. Cutkosky, "The sliding of robot fingers under combined torsion and shear loading," in *Proc. IEEE Int. Conf. Robot. Autom.*, Philadelphia, PA, Apr. 1988, pp. 103–105.
[28] B.-R. Zuo and W.-H. Qian, "A force-closure test for soft multi-fingered grasps," *Sci. China*, ser. E, vol. 41, no. 1, pp. 62–69, Feb. 1998.
[29] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3, Nice, France, May 1992, pp. 2290–2295.
[30] Z. X. Li and S. S. Sastry, "Task-oriented optimal grasping by multifingered robot hands," *IEEE J. Robot. Autom.*, vol. 4, no. 1, pp. 32–44, Feb. 1988.

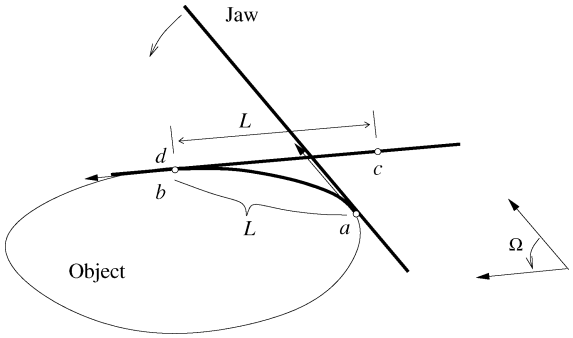

Fig. 1. Localizing a jaw on a motionless object through rolling. During the same period of rolling, the contact point moves from a to b on the object and from c to d on the jaw.

Localization of Curved Parts Through Continual Touch

Yan-Bin Jia

***Abstract*—We describe a simple system that localizes two-dimensional curved shapes through touch sensing, offering computational and experimental studies. The idea lies in determining the placement of a manipulator on a curved object during some special motion—rolling. A geometric algorithm is introduced to locate the boundary segment traced out by their contact using tactile data. Both completeness and local convergence have been established. The algorithm is asymptotically as efficient as evaluating the object's perimeter through numerical integration. For implementation, a two-axis force/torque sensor has been designed to realize contact sensing. Functioning like a "wrist," the sensor is calibrated over the ratio between the bending and twisting moments, eliminating the need for known weights. A simple geometry-based control strategy is devised to implement the rolling motion. Experiments have been conducted with an Adept Cobra 600 manipulator.**

***Index Terms*—Curves, kinematics of rolling, parts localization, solid mechanics, touch sensing.**

I. Introduction

Parts sensing and orienting involve determining the position and orientation of a part whose shape is often known. Not to disturb the part, parts sensing [27], [38], [44] invokes geometric algorithms to process sensor data which serve as constraints on the part. A practical drawback is that the robot is hardly reactive to sensing errors or minor disturbances on the part. A vision system, meanwhile, is unable to handle occlusions. Since a part is often machined according to some computer-aided-design (CAD) model, an image also contains redundant information that could become a source of errors and inefficiency in the process.

With performance guaranteed by mechanical analysis, parts orienting carries out operations, such as vibration [17], tray-tilting [11], parallel-jaw gripping [7], [14], pushing [2], [32], microelectromechanical systems (MEMS) actuation [5], or fixturing [8]. These methods trade sensing (and, thus, all sensor errors) for gained task robustness. Nevertheless, the tradeoff often requires special engineering of the task environment, which increases the cost while decreasing modularity and efficiency.

Manuscript received January 18, 2004; revised July 22, 2004. This work was supported in part by Iowa State University, and in part by the National Science Foundation under CAREER Award IIS-0133681. This paper was recommended for publication by Associate Editor Z. Li and Editor H. Arai upon evaluation of the reviewers' comments.

The author is with the Department of Computer Science, Iowa State University, Ames, IA 50011 USA (e-mail: jia@cs.iastate.edu).

Digital Object Identifier 10.1109/TRO.2005.844675

Knowledge about the geometry of a part facilitates its localization through the exploration of tactile information. In industrial automation, a workpiece is typically localized by finding the optimal registration of some measured points onto a given CAD model [19], [31], [33]. The general scheme iteratively improves on a transformation in order to minimize some least-squares error function and also on the registration of the measured data points. Though involved numerical routines have been developed, the local nature of nonlinear optimization guarantees neither completeness nor efficiency.

In most tasks, parts only need to be localized relative to the robot. In grasping, for instance, if the hand is already in contact with an object, then it needs to only know where the fingers are placed on the object rather than where the object is located (in the world coordinates). The human hand often calibrates itself by moving its fingers on the object's surface so as to "feel" the change of geometry.

To emulate such ability of "feeling," the robotic hand needs to be equipped with a force/torque or tactile array sensor. Such a sensor plays an important role in the dynamic integration of sensing into manipulation.

While a combination of tactile, force, and position sensing carries the promise of enhancing the flexibility and robustness of robotic manipulation [22], the integration of different control strategies for multiple sensor modalities can become very sophisticated and unreliable. From a minimalist point of view, one sensor modality should be preferred if it yields sufficient information needed for task execution.

The aim of this paper is to demonstrate that parts can be localized with very limited touch sensing plus a little action. By doing this, we hope to gain some insight into enabling the robot to "feel" the geometry and pose of an object. We believe that the retrieval and engineering of such knowledge will become important for skillful task execution in the long term.

Our investigation focuses on parts in curved shapes. A substantial amount of research has dealt with polygonal and polyhedral objects so far. These objects do not have local geometry (except at vertices). Nevertheless, actions and mechanics are inherently differential and subject to local geometric properties of bodies interacting with each other.

The specific problem studied in this paper is posed as follows. A jaw (as shown in Fig. 1) rolls from one location (a) to another (b) on an object. We would like to determine these two locations, thus locating the jaw on the object (i.e., determining the object's relative pose to the jaw). The idea is to measure the angle of rotation (Ω) by the jaw as well as the distance (L) of contact movement on the object boundary. We offer an algorithm that finds the segment traced out by the contact point in Section II.