Real-Time Grasping-Force Optimization for Multifingered Manipulation: Theory and Experiments

Guanfeng Liu, *Student Member, IEEE,* and Zexiang Li, *Member, IEEE*

***Abstract*—Real-time grasping-force optimization is a vital problem in dextrous manipulation with multifingered robotic hands. In this paper, we review two prominent approaches that have recently been proposed for this problem, and identify their common need for an appropriate initial condition to start the recursive algorithms. Then, we propose an efficient algorithm that combines a linear-matrix-inequality method with a gradient method for automatic generation of initial conditions. By incorporating the initial conditions, optimal grasping forces can be generated efficiently in real time. Finally, we implement and evaluate all the algorithms for grasping-force control in the Hong Kong University of Science and Technology three-fingered hand platform. Experimental results demonstrate the efficiency of the proposed approach.**

***Index Terms*—Friction-cone constraint, grasping-force optimization, linear-matrix inequality, max–det problem.**

I. Introduction

INSPIRED by the dexterity of the human hand, robotics researchers have long been fascinated by the design and control of multifingered robotic hands [1]–[8]. Compared with current parallel jaw or special-purpose grippers that have dominated industrial applications, a general-purpose multifingered robotic hand has many distinct advantages, especially with robots moving from the traditional domains into household, elderly care, space applications, etc.

First, a much wider class of objects, sometimes with complicated geometries, can be stably grasped by a multifingered hand. Second, better knowledge about the geometry of the grasped object and the environment can be acquired through dextrous motions of the fingers and sensors integrated into the hand. Finally and more importantly, dextrous manipulation of a grasped object can be performed in a relatively restricted environment. Here, dextrous manipulation is achieved through a combination of four manipulation modes: 1) coordinated manipulation with fixed contacts [9], [10]; 2) rolling motion [11]–[15]; 3) sliding motion [16], [17]; 4) finger gaiting [18][19][20]. In any of these manipulation modes, the fingers have to impart forces on the grasped object to balance the gravity force and other possible external forces acting on the object, and are subject to friction cone constraints imposed by the physics of the contact. Determination of appropriate finger forces simultaneously satisfying these constraints is a fundamental problem in dextrous manipulation by multifingered robotic hands. This problem is made more difficult by the fact that friction-cone constraints are inherently nonlinear and optimal solutions in real time are required for even time-varying contacts (rolling and sliding). Sometimes, the dimension of the problem can become prohibitively large.

Despite these difficulties, there have been enormous progresses over the last twenty years in understanding the various issues that pave the way for an ultimate solution of this difficult problem. Salisbury and Craig [21] made the first attempt by introducing mathematic models of contacts and the grasp map that relates applied finger forces to object wrenches. Kerr and Roth [22] and Cutkosky [23] developed in detail the kinematic relations needed for fine manipulation in a multifingered hand system. Montana [24], [25] and Cai and Roth [26] studied the kinematics of contact and derived elegant formulas relating contact velocities to the change of contact coordinates. These fundamental relations pave the way for study of rolling and sliding motions, as in Li and Canny [27] and Murray *et al.* [28], where a general framework for nonholonomic motion planning and control was developed. Cole *et al.* [11], Paljug *et al.* [12], and Sarkar *et al.*[13] proposed algorithms for object manipulation under rolling contacts, whereas Cole *et al.* [16] and Zheng *et al.* [17] developed control algorithms for object manipulation under sliding contacts. Bicchi [29] and Okamura *et al.* [30] provided comprehensive surveys on status of dextrous manipulation research. Based on the kinematic relations of a multifingered robotic hand system, Kerr and Roth [22], Chen and Orin [31], and Jameson and Leifer [32] developed numerical algorithms for solving the grasping-force optimization problem with fixed points of contact. Since their algorithms are obtained by linearizing the friction-cone constraints, the resulting forces are relatively conservative and cannot be computed in real time. Nakamura *et al.* [9] proposed a nonlinear programming-based algorithm for off-line computation of optimal grasping forces. Similar works were also done by Sinha and Abel [33], Kumar and Waldron [34], and Kvrgic [35].

A breakthrough in the study of grasping-force optimization was made by Buss *et al.* [36]. Based on the important observation that the friction-cone constraints are equivalent to positive definiteness of certain symmetric matrices, they transformed the problem into a convex optimization problem in some properly defined Riemannian manifolds with linear constraints. Several gradient-flow type algorithms were developed for real-time

Manuscript received April 18, 2002; revised March 25, 2003. This work was supported by the Research Grants Council of Hong Kong under Grant HKUST 6187/01E, Grant HKUST 6221/99E, and Grant CRC98/01 EG02.

The authors are with the Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology, Kowloon, Hong Kong (e-mail: liugf@ee.ust.hk; eezxli@ust.hk).

Digital Object Identifier 10.1109/TMECH.2004.823879

computation of optimal grasping forces [37], [38]. However, when these algorithms were applied to time-varying contacts or to systems with a modest number of fingers, the computation task becomes formidable. Li and Qin [39] improved the problem by splitting the computation into an on-line and an off-line component and exploring block matrix inversion techniques with sparse matrices. With this improvement, the computation time for a two-fingered manipulation with rolling contact was reduced from 3 to 0.08 s on a Motorola 68 040 processor [39]. Han *et al.* [40], [41] further realized that the friction-cone constraints can be formulated as linear-matrix inequalities (or LMIs) and the grasping-force optimization problem as a convex optimization problem involving LMIs, with the max–det function as the objective function. The interior point algorithm [42] and [43] can be used to provide efficient solutions of the problem with either fixed points of contact, or time-varying contacts, and also for a modest number of fingers. A common problem encountered in all previous approaches [36], [39], [41] was, however, the need for an initial condition satisfying both the friction-cone constraints and the force balance equation to start the recursive algorithms. In [41], Han *et al.* provided a partial solution to this problem by solving another max–det problem involving LMIs. However, this method suffers from singularity issues.

Based on previous works by Han *et al.*[41], Li and Qin [39], and Buss *et al.* [36], this paper aims to provide a complete solution to the grasping-force optimization problem. First, through the use of a gradient algorithm at singular configurations and Han *et al.*'s LMI algorithm at nonsingular configurations, an appropriate initial condition is automatically generated in real-time. Second, by incorporating the initial condition, an algorithm is developed for efficient computation of optimal grasping forces. Efficiency of the method for manipulation tasks involving a large number of fingers and objects of complex geometries derives from the nature of the interior point algorithm [43], [42]. Finally, experimental studies with the Hong Kong University of Science and Technology (HKUST) three-fingered hand are performed to show convergence and effectiveness of the proposed method.

The paper is organized as follows. In Section II, we formulate the grasping-force optimization problem and review two important approaches for this problem. In Section III, we propose an algorithm for automatically generating initial conditions satisfying both the friction-cone constraints and the force balance equation. In Section IV, we develop quasi-static control laws for finger motions. In Section V, we report experimental results with the HKUST three-fingered hand. In Section VI, we end the paper with a brief discussion of future works.

II. Grasp Models and Problem Statement

In this section, we review the kinematic model of a multifingered hand manipulation system and introduce the grasping-force optimization problem. We discuss two promising approaches to this problem and the importance of automatic generation of initial conditions for the recursive algorithms.

Consider a k-fingered hand grasping an object as shown in Fig. 1. Assume that all fingers make contacts of constant types

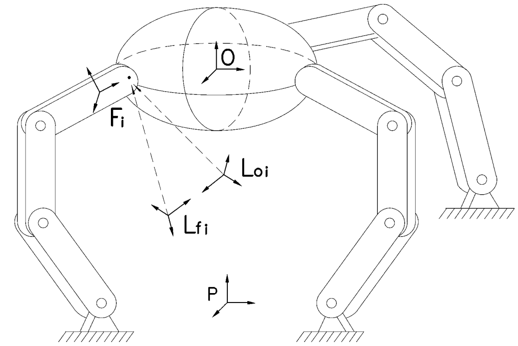


Fig. 1. k-fingered hand grasping an object.

with the object. Three contact models, frictionless point contact (FPC), point contact with friction (PCWF), and soft finger contact with elliptic approximation (SFCE) are considered in our analysis. Following the notations in [24] and [28], we attach an object frame O to the center of mass of the object, finger frame F_i $(i = 1, \ldots, k)$ to the fingertip of the ith finger, and local frames L_{oi} and L_{fi} $(i = 1, \ldots, k)$ to the object and finger i at the point of contact. A configuration of contact is described by contact coordinates $\eta_i = (\alpha_{oi}^T, \alpha_{fi}^T, \psi_i)^T \in \mathbb{R}^5$, where $\alpha_{oi} = (u_{oi}, v_{oi})^T \in \mathbb{R}^2$ are the local coordinates of contact relative to the object, $\alpha_{fi} = (u_{fi}, v_{fi})^T \in \mathbb{R}^2$ the coordinates of contact relative to the fingertip, and ψ_i the angle of contact. Collectively, a contact configuration of the system is described in local coordinates by $\eta = (\eta_1^T, \ldots, \eta_k^T)^T \in \mathbb{R}^{5k}$. The relation between the applied finger forces and the resulting object wrench is given by the grasp map $G \in \mathbb{R}^{6\times n}$

$$f_o = Gx \tag{1}$$

where $x = [x_1{}^T \ldots x_k{}^T]^T \in \mathbb{R}^n$, with $x_i \in \mathbb{R}^{n_i} (i = 1, \ldots, k)$ and $n = \sum_{i=1}^k n_i$, is the vector of finger forces. The dual map G^* takes the object velocity to the virtual contact velocity

$$v_c = G^* v_o$$

with $\langle v_c, x\rangle$ denoting the virtual work. The finger force is constrained to the friction cone

$$\mathcal{FC}_i = \{x_i \in \mathbb{R}^{n_i} \mid x_{i,n} \geq 0, \|x_{i,t}\|_s \leq x_{i,n}\}$$

or collectively to

$$\mathcal{FC} = \mathcal{FC}_1 \times \cdots \times \mathcal{FC}_k = \{x \in \mathbb{R}^n \mid x_i \in \mathcal{FC}_i\}$$

with $x_{i,n}$ and $x_{i,t}$ being, respectively, the normal and the tangential components of the finger forces at the ith point of contact. Here, $\|x_{i,t}\|_s$ denote vector norms described for each of the contact models by

$$\text{FPC: } \|x_{i,t}\|_s = 0 \tag{2}$$

$$\text{PCWF: } \|x_{i,t}\|_s = \frac{1}{\mu_i}\sqrt{x_{i,1}{}^2 + x_{i,2}{}^2} \tag{3}$$

$$\text{SFCE: } \|x_{i,t}\|_s = \sqrt{\frac{1}{\mu_i^2}\left(x_{i,1}^2 + x_{i,2}{}^2\right) + \frac{1}{\mu_{i,t}^2}x_{i,4}{}^2} \tag{4}$$

with $x_{i,1}$ and $x_{i,2}$ being the friction force components in the tangential plane, $x_{i,4}$ the moment along the contact normal, μ_i

the Coulomb friction coefficient, and $\mu_{i,t}$ the coefficient of torsional friction.

A. Problem Statement

The friction-cone constraints (2)–(4) can be generalized as a *second-order cone* (SOC) constraint of the form

$$\|A_i x_i + b_i\| \leq c_i^T x_i + d_i \tag{5}$$

where $A_i = A_i^T \in \mathbb{R}^{n_i \times n_i}, b_i \in \mathbb{R}^{n_i}$, $c_i \in \mathbb{R}^{n_i}$, and $d_i \in \mathbb{R}$. The norm in (5) is the standard Euclidean norm, i.e., $\|u\| = (u^T u)^{(1/2)}$. An important discovery of Buss *et al.* [36] is that the quadratic inequalities (5) can be expressed as matrix inequalities (MI) of the form

$$Q_i(x_i) \geq 0 \tag{6}$$

where

$$Q_i(x_i) \in \mathbb{R}^{(n_i+1)\times(n_i+1)} := \begin{bmatrix} (c_i^T x_i + d_i)I & A_i x_i + b_i \\ (A_i x_i + b_i)^T & c_i^T x_i + d_i \end{bmatrix}.$$

The inequality in (6) means that the symmetric matrix $Q_i(x_i)$ is positive semi-definite. We have, for the FPC model (2), that

$$A_i = 0 \quad b_i = 0 \quad c_i = 1 \quad d_i = 0$$

for PCWF (3), such that

$$A_i = \begin{bmatrix} \frac{1}{\mu_i} & 0 & 0 \\ 0 & \frac{1}{\mu_i} & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad b_i = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad c_i = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

and $d_i = 0$, and for SFCE (4), such that

$$A_i = \begin{bmatrix} \frac{1}{\mu_i} & 0 & 0 & 0 \\ 0 & \frac{1}{\mu_i} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\mu_{it}} \end{bmatrix} \quad b_i = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad c_i = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

and $d_i = 0$. Collectively, the friction-cone constraints for the hand manipulation system are represented as

$$Q(x) \in \mathbb{R}^{l_0 \times l_0} = \mathrm{diag}(Q_1(x_1), \ldots, Q_k(x_k)) \geq 0 \tag{7}$$

where $l_0 = \sum_{i=1}^k (n_i + 1)$. It can be shown that the allowed contact force set

$$S = \{x \mid Q(x) \geq 0, Gx = f_o\} \tag{8}$$

is convex if it is not empty [41], [42].

Remark 1: Using MI (7), the *force closure problem* can be transformed into the requirement that [41]:

1) $\mathrm{rank(G)} = 6$;
2) $\exists x_{\mathrm{int}}$, such that $Q(x_{\mathrm{int}}) > 0$ and $Gx_{\mathrm{int}} = 0$.

Problem 1: Grasping-Force Optimization Problem: Find an optimal finger force x from the set S that minimizes a suitable cost function.

In general, there exists no unique or natural choice of cost functions. Several possibilities are discussed in the following sections.

B. Buss, Hashimoto, and Moore (BHM) Approach With Li and Qin's Improvement

A good treatment to the grasping-force optimization problem is first proposed by Buss *et al.* [36], [38]. They transformed the problem into a convex optimization problem on the Riemannian manifold of positive definite matrices with linear constraints

$$\min_Q \phi(Q) = \mathbf{Tr}\big(W_p Q + W_i Q^{-1}\big) \tag{9}$$

subject to

$$Q > 0 \tag{10}$$

$$B\mathrm{vec}(Q) = r \tag{11}$$

where $B \in \mathbb{R}^{m \times l}$, with $l = l_0^2$, is the constraint matrix obtained from the balance equation $f_o = Gx$ and the special requirement on elements of $Q, \mathrm{vec}(\cdot)$ denotes the vector operation, and $r = [r_1, \ldots, r_m]^T \in \mathbb{R}^m$ is a vector whose components are either 0 or the elements of f_o. $W_i \in \mathbb{R}^{l_0 \times l_0}$ and $W_p \in \mathbb{R}^{l_0 \times l_0}$ are two weighting matrices chosen for particular control tasks in real implementations.

The first term $\mathbf{Tr}(W_p Q)$ in (9) is used to weigh the normal component of x, while the second term $\mathbf{Tr}(W_i Q^{-1})$ is used as a barrier to ensure that the finger forces lie in the friction cone. In [38], the second term is replaced by a self-concordant term $\log \det Q^{-1}$. Problem (9) can be solved by the standard constrained gradient flow algorithm [44]

$$\mathrm{vec}(\dot{Q}) = \alpha(I - B^T(BB^T)^{-1}B)\mathrm{vec}(Q^{-1}W_i Q^{-1} - W_p) \tag{12}$$

with α an appropriate step size chosen to ensure convergence. The optimal solution for finger force x is recovered from the optimal solution for $\mathrm{vec}(Q)$. In algorithm (12), the gradient flow is based on the standard Riemannian metric. Different algorithms will be obtained if different Riemannian metrics are used [38]. In [38], a Dikin-type algorithm was developed along with a guideline for choosing the step size α for better convergence. A major difficulty in implementing (12) is that the computation time for $(BB^T)^{-1}$ will be formidable for systems with time-varying contacts and a modest number of fingers. Li and Qin [39] refined (12) by decomposing the computation of $(BB^T)^{-1}$ into the on-line and off-line components. They observed that

$$B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \in \mathbb{R}^{m \times l} \quad r = \begin{bmatrix} 0 \\ f_o \end{bmatrix} \in \mathbb{R}^m$$

with $B_1 \in \mathbb{R}^{(m-6)\times l}$ a constant submatrix representing the structural constraints, and $B_2 \in \mathbb{R}^{6 \times l}$ a time-varying submatrix describing the force balance constraints. Then

$$BB^T = \begin{bmatrix} B_1 B_1^T & B_1 B_2^T \\ B_2 B_1^T & B_2 B_2^T \end{bmatrix} := \begin{bmatrix} B_{11} & B_{12} \\ B_{12}^T & B_{22} \end{bmatrix}$$

and

$$(BB^T)^{-1} = \begin{bmatrix} B_{11}^{-1} + E\Delta^{-1}W & -E\Delta^{-1} \\ -\Delta^{-1}W & \Delta^{-1} \end{bmatrix}$$

where $\Delta = B_{22} B_{12}^T B_{11}^{-1} B_{12} \in \mathbb{R}^{6\times 6}$, $E = B_{11}^{-1} B_{12}$, and $W = B_{12}^T B_{11}^{-1}$. B_{11}^{-1} can be computed off-line as B_{11} is constant. The on-line component will be the inverse of the 6×6 matrix Δ, whose computation time is much smaller than that of $(BB^T)^{-1}$. Furthermore, B_1 and B_2 are, in general, sparse matrix. Thus, some numerical methods about sparse matrix computation can be used to further reduce the computation time.

Remark 2: In the BHM approach, the constrained convex set S (8) are imbedded into $\mathbb{R}^{l_0 \times l_0}$ via the positive semi-definiteness constraint (10) and the affine constraints (11). It is therefore necessary to find an initial point satisfying both (10) and (11) to start the recursive algorithm (12). This is, in fact, not an easy problem in real implementation.

C. Han, Trinkle, and Li (HTL) Approach

Han *et al.* [41] further observed that $Q(x)$ is linear in x and by reordering of the components of x, (7) can be expressed as an LMI of the form

$$Q(x) = G_0 + G_1 x_1 + \cdots + G_n x_n \geq 0 \tag{13}$$

where

$$G_0 = \operatorname{diag}\left(\begin{bmatrix} d_1 I & b_1 \\ b_1^T & d_1 \end{bmatrix}, \ldots, \begin{bmatrix} d_k I & b_k \\ b_k^T & d_k \end{bmatrix} \right)$$

$$G_j = \operatorname{diag}\left(0_1, \ldots, 0_{p-1}, \begin{bmatrix} c_{p,q} I & a_{p,q} \\ a_{p,q}^T & c_{p,q} \end{bmatrix}, 0_{p+1}, \ldots, 0_k \right)$$

$$j = \sum_{t=1}^{p-1} n_t + q, \qquad 0 < q \leq n_p$$

$c_p = [c_{p,1}, \ldots, c_{p,n_p}]^T$ and $A_p = [a_{p,1}, \ldots, a_{p,n_p}]$ with $a_{p,q} \in \mathbb{R}^{n_p}$ being the qth column vector of A_p. The affine constraints $Gx = f_o$ can be eliminated by substituting $x = G^+ f_o + Vy$ to (13)

$$\tilde{Q}(y) = \tilde{G}_0 + \tilde{G}_1 y_1 + \cdots + \tilde{G}_{n-6} y_{n-6} \geq 0 \tag{14}$$

where $x_0 = (x_{0,1}, \ldots, x_{0,n})^T = G^+ f_o = G^T (GG^T)^{-1} f_o$ is a special solution, $V \in \mathbb{R}^{n \times (n-6)}$ a matrix whose columns forming a basis for the null space of $G, \tilde{G}_0 = G_0 + \sum_{p=1}^{n} G_p x_{0,p}$, and $\tilde{G}_p = \sum_{q=1}^{n} G_q v_{q,p}, 1 \leq p \leq n-6$ with $v_{p,q}$ the (pq)th element of V. The cost function to be minimized consists of a linear term and a determinant term, and the problem is refined as a max–det problem (see the Appendix for a general discussion)

$$\min_y \phi(y) = e^T y + \log\ \det \tilde{Q}(y)^{-1} \tag{15}$$

subject to

$$\tilde{Q}(y) > 0 \tag{16}$$

$$F(y) = F_0 + \sum_{j=1}^{n-6} F_j y_j \geq 0. \tag{17}$$

Here, we have the freedom of arbitrarily selecting $F_j' s, j = 0, \ldots, n-6$. The only requirement from the max–det problem is that the matrices $\operatorname{diag}(\tilde{G}_j, F_j), j = 1, \ldots, n-6$, be linearly independent. One particular choice is that $F_0 = 1$, and $F_j = 0, j = 1, \ldots, n-6$, and the constraint (17) is automatically satisfied. The vector $e = (e_1, \ldots, e_{n-6})^T$ in (15) is chosen to weigh the normal component of the finger force x. In general, it varies with the contact configuration and can be specified as follows. First, select a constant vector $e_0 = (e_{0,1}, \ldots, e_{0,n})^T$ relative to the finger force x, e.g., with $e_0^T x$ giving the normal components of x. Then, transform e_0 to $e = (e_1, \ldots, e_{n-6})^T$ by the formula

$$e_p = \sum_{q=1}^{n} e_{0,q} V_{q,p}, \quad p = 1, \ldots, n-6.$$

Note that $\operatorname{logdet} \tilde{Q}(y)^{-1}$ in (15) is convex and self-concordant. With this term, the finger forces are confined in the interior of the friction cone [41]. Problem (15) can be solved by the interior point algorithm [42] with a given valid initial condition. The complexity of this algorithm grows slowly with the dimension of the system, a remarkable property that we choose the HTL approach as our focus. Furthermore, in the section that follows we will use the HTL approach to find a valid initial condition required by both the HTL approach and the BHM approach for the grasping-force optimization.

III. Initial Point Algorithms

As mentioned in Section II, both the BHM approach and the HTL approach require an initial point x that satisfies $Q(x) > 0$ (or an initial point y that satisfies $\tilde{Q}(y) > 0$). This problem can be solved using the following two methods.

A. HTL Method

The HTL method [41] makes use of the max–det formulation and the interior point algorithm [42] (see the Appendix). First, augment the second constraint in the standard max–det problem with an additional base matrix $\tilde{G}_{n-6+1} = I$, and use the interior point algorithm to solve the following problem:

$$\min_z \phi(z) = \hat{e}^T z \tag{18}$$

subject to

$$\tilde{Q} = I > 0 \tag{19}$$

$$F(z) := \tilde{G}_0 + \sum_{j=1}^{n-6} \tilde{G}_j z_j + I z_{n-6+1} \geq 0 \tag{20}$$

where $z = [z_1, \ldots, z_{n-6+1}]^T, \hat{e} = [0, \ldots, 0, 1]$. Since $\tilde{Q} = I > 0$, the first constraint in the max–det problem is automatically satisfied. An initial point to start the interior point algorithm for (18) and (20) is given by

$$z = [0, \ldots, 0, t], \qquad t > -\lambda_{\min}(\tilde{G}_0)$$

where $\lambda_{\min}(\tilde{G}_0)$ is the minimum eigenvalue of $\tilde{G}_0$. If the optimal value of $\phi(z) = z_{n-6+1} < 0$, then, the corresponding solution vector $z^* = [z_1, \ldots, z_{n-6}]$ is a valid initial point for the grasping-force optimization problems in (9) and (15). In practice, we can stop the computation once $z_{n-6+1} < 0$ is reached.

The HTL method works well in general, but suffers from some singularity problem in some cases, when $\tilde{G}_1, \ldots, \tilde{G}_{n-6+1} = I$ become linearly dependent. This can be verified by checking

$$\operatorname{Rank}(A) < n - 6 + 1$$

with

$$A = [\operatorname{vec}(\tilde{G}_1), \ldots, \operatorname{vec}(\tilde{G}_{n-6+1})] \in \mathbb{R}^{l_0^2 \times (n-6+1)}$$

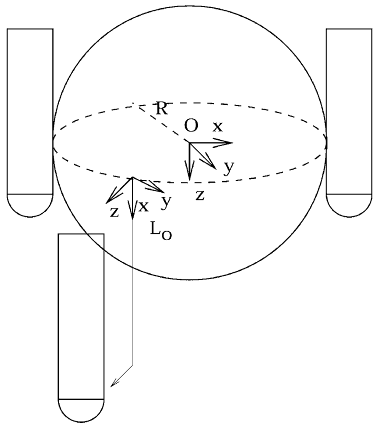


Fig. 2. Three-fingered hand grasping a sphere.

and $l_0^2 = (n+k)^2 >> n-6+1$ in general. Although the probability for the appearance of singular problem is low, it do exist and will make the HTL method disabled.

In the experiments of grasping a spherical object with a three-fingered hand, we find the following singular case.

Example 1: A Three-Fingered Hand Grasping a Spherical Object: Consider a three-fingered hand with cylindric fingertips grasping a spherical object of radius $R = 0.1$ m as shown in Fig. 2. The sphere is parameterized by the spherical coordinates

$$\psi : \mathbb{R}^2 \rightarrow \mathbb{R}^3 : \begin{bmatrix} u_o \\ v_o \end{bmatrix} \rightarrow \begin{bmatrix} R\cos u_o \cos v_o \\ -R\cos u_o \sin v_o \\ R\sin u_o \end{bmatrix}.$$

Let the local coordinates for the three contact points be $\alpha_{o1} = (0, (2\pi/3))^T, \alpha_{o2} = (0,0)^T$, and $\alpha_{o3} = (0, -(2\pi/3))^T$, respectively, and the weight of the object be 10 N. The contacts between the fingers and the object are PCWF. The grasp map and a basis for its null space are given by

$$G = \begin{bmatrix} 0 & \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 & 0 & -1 & 0 & -\frac{\sqrt{3}}{2} & \frac{1}{2} \\ 0 & -\frac{1}{2} & \frac{\sqrt{3}}{2} & 0 & 1 & 0 & 0 & -\frac{1}{2} & -\frac{\sqrt{3}}{2} \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ -\frac{\sqrt{3}}{20} & 0 & 0 & 0 & 0 & 0 & \frac{\sqrt{3}}{20} & 0 & 0 \\ \frac{1}{20} & 0 & 0 & -\frac{1}{10} & 0 & 0 & \frac{1}{20} & 0 & 0 \\ 0 & \frac{1}{10} & 0 & 0 & \frac{1}{10} & 0 & 0 & \frac{1}{10} & 0 \end{bmatrix}$$

and

$$V = \begin{bmatrix} 0 & 0 & 0 \\ 0.0866 & -0.0866 & 0 \\ 0.15 & 0.15 & 0 \\ 0 & 0 & 0 \\ -0.0866 & 0 & 0.0866 \\ 0.15 & 0 & 0.15 \\ 0 & 0 & 0 \\ 0 & 0.0866 & -0.0866 \\ 0 & 0.15 & 0.15 \end{bmatrix}.$$

The solution set for x is given by

$$x = x_0 + Vy \quad x_0 = [-3.3, 0, 0, -3.3, 0, 0, -3.3, 0, 0]^T.$$

Substituting the above result into (16) yields

$$\tilde{Q}(y) = Q(x_0 + Vy) = \tilde{G}_0 + \tilde{G}_1 y_1 + \tilde{G}_2 y_2 + \tilde{G}_3 y_3 \quad (21)$$

where

$$\tilde{G}_0 = \mathrm{Diag}(D_2, D_2, D_2) \quad \tilde{G}_1 = \mathrm{Diag}(D_0, D_1, 0)$$
$$\tilde{G}_2 = \mathrm{Diag}(D_1, 0, D_0) \quad \tilde{G}_3 = \mathrm{Diag}(0, D_0, D_1)$$

and

$$D_0 = \begin{bmatrix} 0.075 & 0 & 0 \\ 0 & 0.075 & 0.0866 \\ 0 & 0.0866 & 0.075 \end{bmatrix}$$
$$D_1 = \begin{bmatrix} 0.075 & 0 & 0 \\ 0 & 0.075 & -0.0866 \\ 0 & -0.0866 & 0.075 \end{bmatrix}$$
$$D_2 = \begin{bmatrix} 0 & 0 & -3.33 \\ 0 & 0 & 0 \\ -3.33 & 0 & 0 \end{bmatrix}.$$

Following the HTL method leads to a singular system as

$$\tilde{G}_1 + \tilde{G}_2 + \tilde{G}_3 = 0.15I.$$

We propose the following remedy to the HTL method.

B. Gradient Method

It is well known that positive definiteness of a given matrix is equivalent to the positiveness of all its eigenvalues, or its minimal eigenvalue. Based on this idea, we develop an algorithm to find a monotone increasing flow of the minimal eigenvalue so that it eventually becomes positive.

In (14), the matrix $\tilde{Q}(y)$ is a linear combination of constant base matrices, a distinct property based on which we can derive the gradient flow of its minimal eigenvalue. Let $\lambda_{l_0}(\tilde{Q}(y))$ be the minimal eigenvalue of $\tilde{Q}(y)$.

Theorem 1: Gradient Flow of $\lambda_{l_0}(\tilde{Q}(y))$: The monotone increasing flow of $\lambda_{l_0}(\tilde{Q}(y))$ is given by

$$\dot{y} = \nabla_y \lambda_{l_0}(\tilde{Q}(y)) = \begin{bmatrix} \omega_y^T \tilde{G}_1 \omega_y \\ \vdots \\ \omega_y^T \tilde{G}_{n-6} \omega_y \end{bmatrix}$$

where ω_y is the unit eigenvector of $\tilde{Q}(y)$ corresponding to the minimal eigenvalue.

Proof: Since $\lambda_{l_0}(\tilde{Q}(y))$ is the minimal eigenvalue of $\tilde{Q}(y)$, we have

$$(\tilde{Q}(y) - \lambda_{l_0} I)\omega_y = 0. \quad (22)$$

Taking derivatives on both sides with respect to y_i, yields

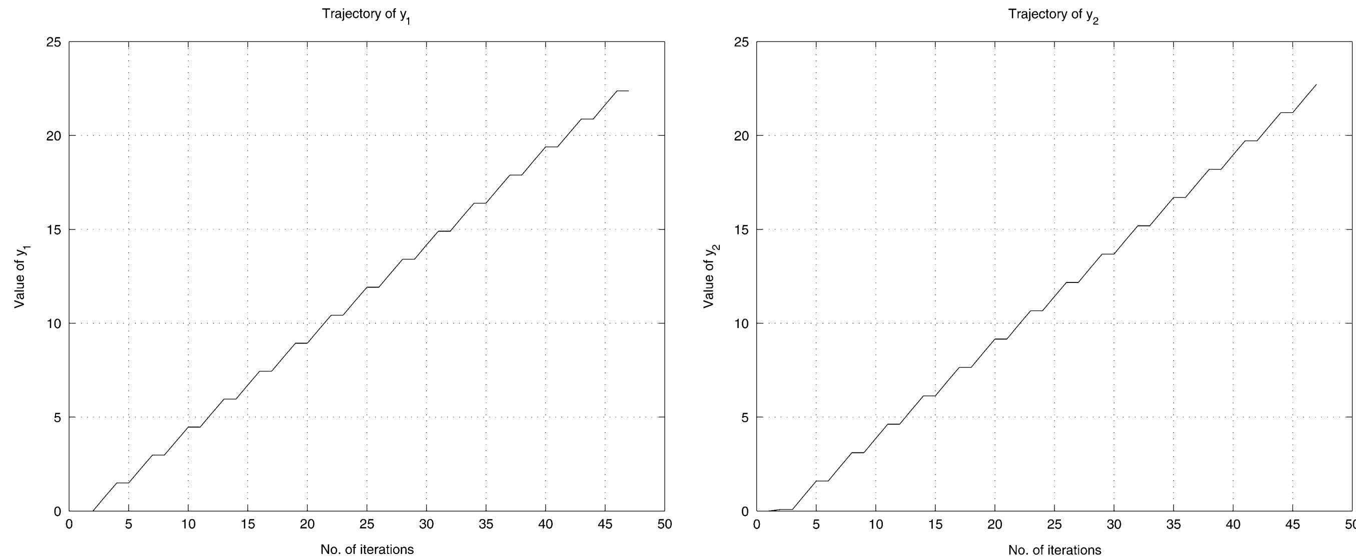
$$\left(\tilde{G}_i - \frac{\partial \lambda_{l_0}}{\partial y_i} I\right)\omega_y + (\tilde{Q}(y) - \lambda_{l_0} I)\frac{\partial \omega_y}{\partial y_i} = 0.$$

Multiply both sides by ω_y^T and consider (22), we have

$$\omega_y^T \left(\tilde{G}_i - \frac{\partial \lambda_{l_0}}{\partial y_i} I\right)\omega_y = 0$$

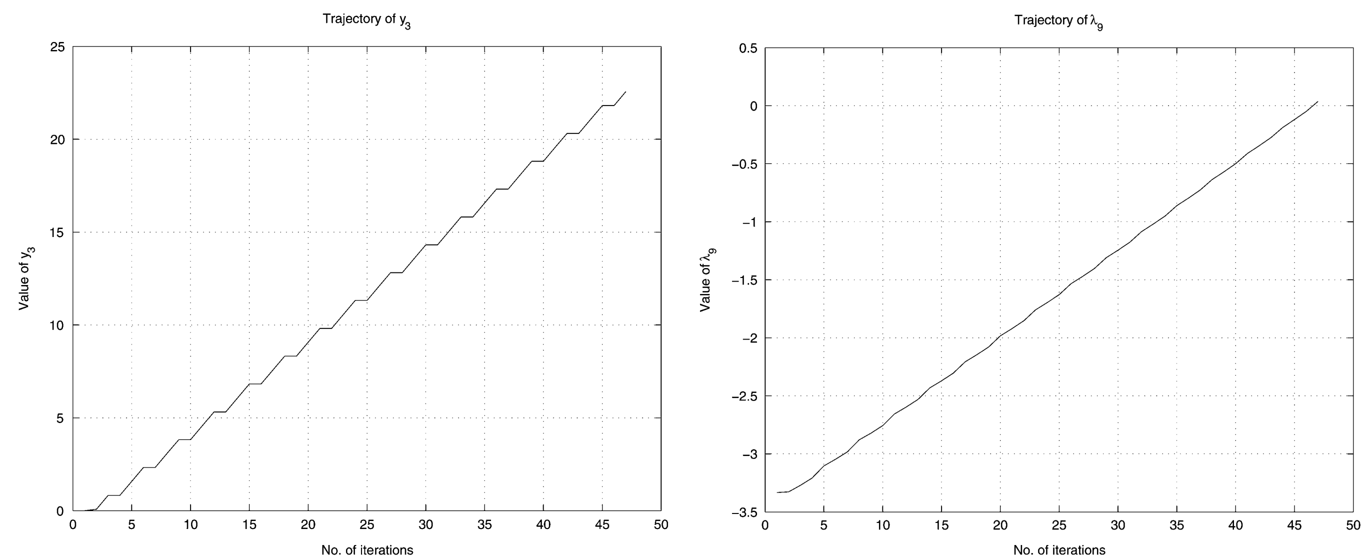
or

$$\frac{\partial \lambda_{l_0}}{\partial y_i} = \omega_y^T \tilde{G}_i \omega_y$$



(a) (b)

Fig. 3. (a) Trajectory of y_1. (b) Trajectory of y_2.



(a) (b)

Fig. 4. (a) Trajectory of y_3. (b) Trajectory of λ_9.

as $\omega_y^T \omega_y = 1$. The monotone increasing property of the flow follows from

$$\dot{\lambda}_{l_0} = (\nabla_y \lambda_{l_0})^T \dot{y} = (\nabla_y \lambda_{l_0})^T \nabla_y \lambda_{l_0} \geq 0.$$

Algorithm 1: Find a Valid Initial Condition
Input: A set of base matrices $\tilde{G}_0, \ldots, \tilde{G}_{n-6}$, tolerance $\delta < 0.5$, and the maximal step N;
Output: A vector $y^0 = (y_1^0, \ldots, y_{n-6}^0)^T$ such that $\tilde{Q}(y^0) = \tilde{G}_0 + \tilde{G}_1 y_1^0 + \cdots + \tilde{G}_{n-6} y_{n-6}^0 > 0$.

1) Set $k = 0$;
2) Initialize $y(0)$;
3) If $k > N$, quit from the algorithm and output failure;
4) Else compute $\tilde{Q}(k) = \tilde{Q}(y(k))$;
5) Finding the minimal eigenvector $\omega(k)$ of $\tilde{Q}(k)$;
6) Compute $\lambda(k) = \omega(k)^T \tilde{Q}(k) \omega(k)$;
7) If $\lambda(k) > 0$, quit from the algorithm and record the solution vector $y(k)$;

Fig. 5. HKUST three-fingered hand.

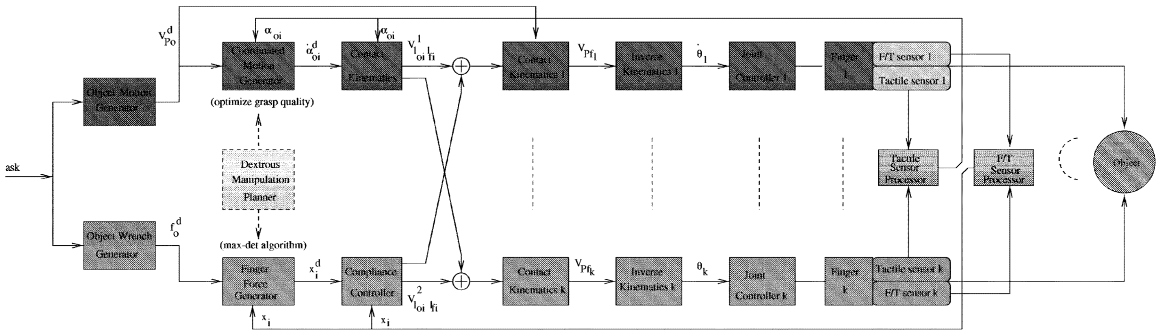


Fig. 6. Control diagram for a k-fingered hand manipulating an object.

8) Else calculate

$$\delta y(k) = \begin{bmatrix} \omega(k)^T \tilde{G}_1 \omega(k) \\ \vdots \\ \omega(k)^T \tilde{G}_{n-6} \omega(k) \end{bmatrix}$$

9) If $\|\delta y(k)\| < \delta$, quit from the algorithm and output failure; else update $y(k+1) = y(k) + h\delta y(k)$;

10) Set $k = k + 1$ and go to Step 3).

The update factor h needs to be tuned according to real conditions. □

Example 2: Finding an Initial Point: Example 1 Continued: In this example, Algorithm 1 will be used to search for an appropriate initial point y such that $\tilde{Q}(y) > 0$ in (21). The starting point $y(0)$ is given by

$$y_1(0) = 0 \quad y_2(0) = 0 \quad y_3(0) = 0$$

and $\lambda_9(\tilde{Q}(y(0))) = -3.33$. After applying the algorithm, the output is

$$y_1^0 = 22.3694 \quad y_2^0 = 22.7187 \quad y_3^0 = 22.5619$$

and the current minimal eigenvalue is given by

$$\lambda_9 = 0.0368$$

which is strictly positive. Hence, the solution vector $y^0 = [y_1^0, y_2^0, y_3^0]^T$ can be utilized as a starting point for the interior point algorithm. Fig. 3(a) and (b), and Fig. 4(a) shows the trajectory of y, and Fig. 4(b) the variance of the minimal eigenvalue of $\tilde{Q}(y)$. It is shown from these figures that the magnitude of each component of y varies in almost the same speed. This is because the grasp is symmetric. By using the LAPACK software to compute the eigenvalues and the eigenvectors, the computation time is about 20 ms on SUN ULTRA60. In the nonsingular case, the computation time is about 10 ms.

To conclude this section, we have the following algorithm for solving the grasping-force optimization.

Algorithm2: Grasping-Force Optimization
Input: Object wrench f_o and contact coordinates of the object $\alpha_{oi}, i = 1, \ldots, k$;
Output: Desired minimal contact force x^*.

1) Compute the grasp map G from $\alpha_{oi}, i = 1, \ldots, k$;

2) Compute a special solution $x_0 = G^+ f_o$ and the basis V of the null space of G;

3) Substitute $x = x_0 + Vy$ into $Q(x)$ to give a new basis matrices $\tilde{G}_0, \ldots, \tilde{G}_{n-6}$;

4) If $\tilde{G}_i, i = 1, \ldots, n-6$ and $\tilde{G}_{n-6+1} = I$ are linearly independent, use the max–det algorithm with $\tilde{G}_i, i = 1, \ldots, n-6+1$, as the input parameters. If $z_{n-6+1} < 0$, then $z^* = [z_1, \ldots, z_{n-6}]^T$ gives a desired initial point;
else call Algorithm 1 to obtain an initial point y^0;

5) Call the max–det algorithm with $\tilde{G}_i, i = 1, \ldots, n-6$ as the input parameters and y^0 or z^* as the initial point to obtain the desired optimal grasping force $x^* = x_0 + Vy^*$, where Vy^* is the optimal internal grasping force.

To compute $\alpha_{oi}, i = 1, \ldots, k$, in actual implementation, we use tactile data to first obtain $\alpha_{fi}, i = 1, \ldots, k$. Then, we solve for $\dot{\alpha}_{oi}$ from $\dot{\alpha}_{fi}$ with Montana's equation of contact [24]. Finally, $\alpha_{oi}, i = 1, \ldots, k$, is obtained by integrating $\dot{\alpha}_{oi}, i = 1, \ldots, k$.

IV. Hand Kinematics and Control

Based on the results of previous sections, we propose in this section a control algorithm for a multifingered robotic hand to manipulate an object from an initial configuration to a desired final configuration without dropping it. The control objectives include motion of the object, contact configurations and grasping forces. In the following, we will derive finger velocities from velocities of object and points of contact, and also finger forces.

As shown in Fig. 1, the forward kinematics of the ith finger-object system can be expressed as

$$g_{po} = g_{pf_i} g_{f_i l_{f_i}} g_{l_{f_i} l_{o_i}} g_{l_{o_i} o}, \qquad i = 1, \ldots, k \tag{23}$$

where $g_{f_i l_{f_i}}, g_{l_{o_i} o} \in \mathrm{SE}(3)$ are constant transformations. Differentiating (23) yields

$$Ad_{g_{l_{o_i} o}} V_{po} = Ad_{g^{-1}_{f_i l_{o_i}}} V_{pf_i} - Ad_{g_{l_{o_i} l_{f_i}}} V_{l_{o_i} l_{f_i}}, \qquad i = 1, \ldots, k$$

where $V_{l_{o_i} l_{f_i}} \in \mathbb{R}^6$ is the contact velocity of finger i with respect to the object. To simultaneously control finger forces and contact locations, we split the contact velocity $V_{l_{o_i} l_{f_i}}$ into two

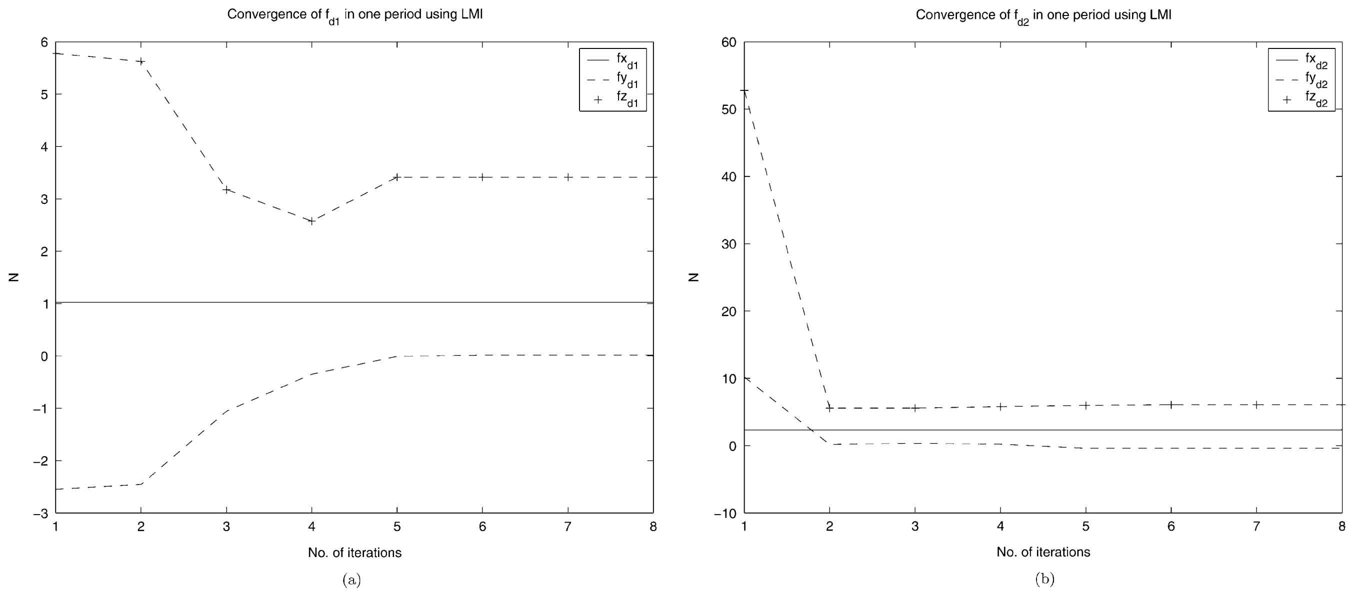


Fig. 7. (a) Convergence of the force of finger 1. (b) Convergence of the force of finger 2.

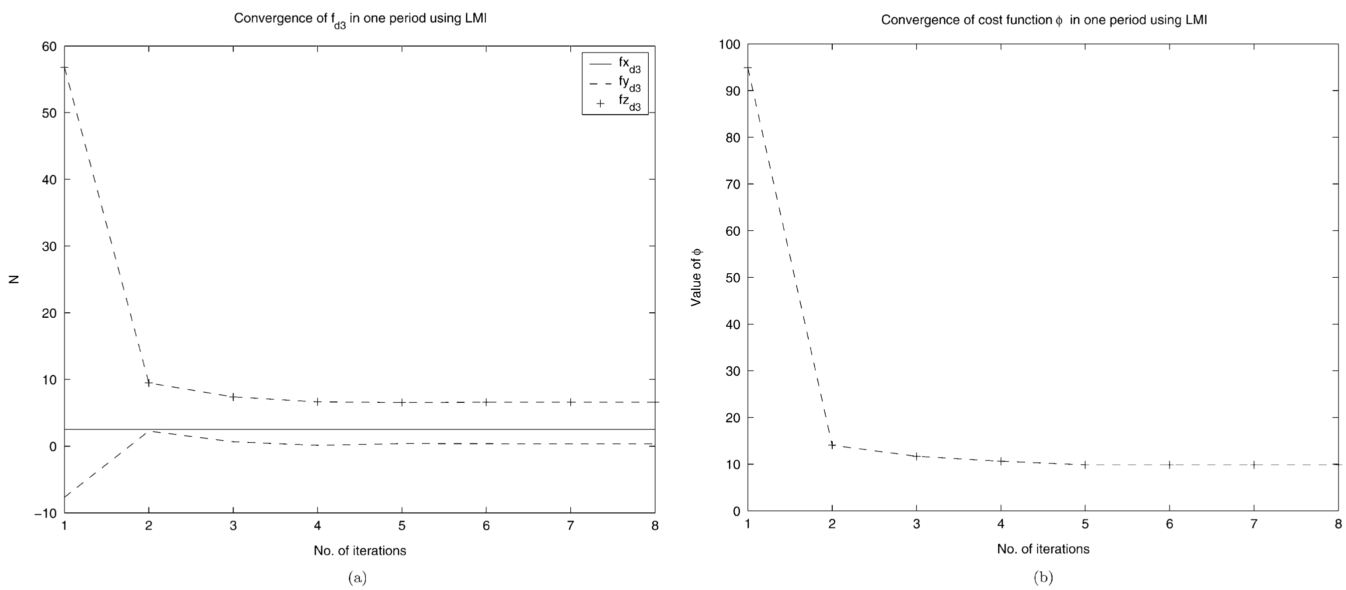


Fig. 8. (a) Convergence of the force of finger 3. (b) Convergence of the cost function ϕ.

components $V^1_{l_{o_i} l_{f_i}}$ and $V^2_{l_{o_i} l_{f_i}}$ as follows. First, $V^1_{l_{o_i} l_{f_i}}$ is specified along the direction of contact coordinates variation through Montana's kinematics of contact

$$V^1_{l_{o_i} l_{f_i}} = J_i(\eta_i)\dot{\eta}_i$$

where $J_i(\eta_i)$ is a function of the geometric parameters of the object and fingertip i, and related to the contact model used in manipulation [39]. Note that the rate of change of contact coordinates can be specified by minimizing some grasp quality functions as in [14], [45], [39]. The second component $V^2_{l_{o_i} l_{f_i}}$ of the contact velocity is chosen to be perpendicular to $V^1_{l_{o_i} l_{f_i}}$ and is used to regulate finger force $x_i \in \mathbb{R}^{n_i}$ through a compliance control scheme $V^2_{l_{o_i} l_{f_i}} = C_i x_i$. where $C_i \in \mathbb{R}^{6\times n_i}$ is a compliance matrix.

TABLE I
COMPARING COMPUTATIONAL TIME OF EXISTING ALGORITHMS

optimization algorithm	Computation time
BHM algorithm without Li and Qin's refinement	≥ 200 ms
BHM algorithm with Li and Qin's refinement	≤ 80 ms
LMI algorithm	≤ 70 ms

Summarizing our discussion, we propose the following control algorithm for the desired finger velocity:

$$V^d_{pf_i} = Ad_{g_{f_i o}}(\eta_i)V^d_{po} + Ad_{g_{f_i l_{f_i}}}(\eta_i)\left(V^1_{l_{o_i} l_{f_i}}(\eta_i, \dot{\eta}^d_i) + C_i\left(x^d_i - x_i\right)\right) \quad (24)$$

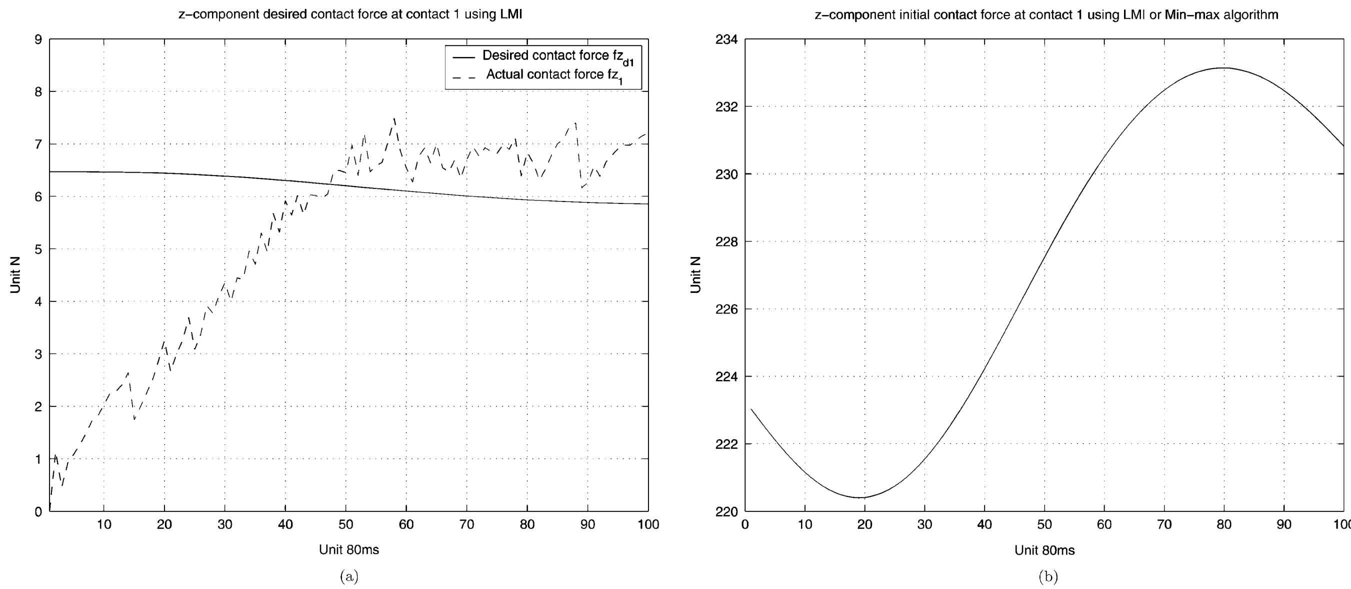


Fig. 9. Finger 1: (a) z-component contact force response. (b) z-component initial contact force.

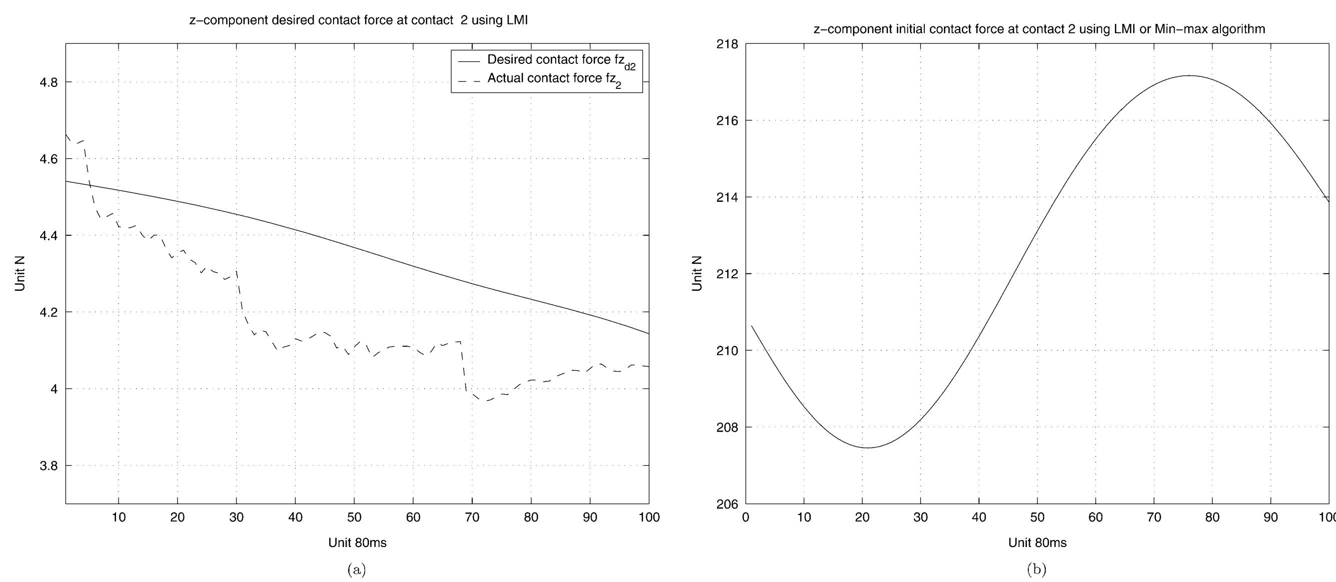


Fig. 10. Finger 2: (a) z-component contact force response. (b) z-component initial contact force.

where $\dot{\eta}_i^d$ is the desired rate of change of the contact coordinates, as specified by minimizing some grasp quality functions, x_i^d is the desired finger force computed from the grasping-force optimization algorithm. In real implementation, α_{fi} is directly measured through tactile sensors at ith fingertip, and the remaining components of η_i are computed by inverting Montana's kinematic equations of contact. x_i is obtained through force/torque sensors integrated with the fingers (see Fig. 5). Fig. 6 shows the block- diagram of this controller.

V. Experiments

The proposed control algorithms have been implemented and tested on the HKUST three-fingered hand platform as shown in Fig. 5. Each finger of the HKUST hand consists of a Motorman K-3S robot, equipped with a force/torque sensor and a 16×16 tactile array fingertip. A VME-based multiprocessor control system is utilized with three 8-axis digital signal processing motion control boards for joint-level control, and two Motorola 68 040 processors for object-level motion and grasping-force control. The two CPUs work in parallel, with one running the LMI algorithms for grasping-force generation and the other for planning and generation of contact coordinates and object motion. Synchronization of the tasks run in the two CPUs are realized through shared semaphores. This is an important function of the VxWorks operating system.

A. The Software

There are two software modules for grasping-force optimization. First, the maxdet-src module, which is downloaded from http://www.stanford.edu/~boyd/MAXDET.html, is used for grasping-force generation at a fixed contact location. Second, an *LMIcall* module, is used to compute the grasp map G, a special solution x_0, and the null matrix V of G. In this module,

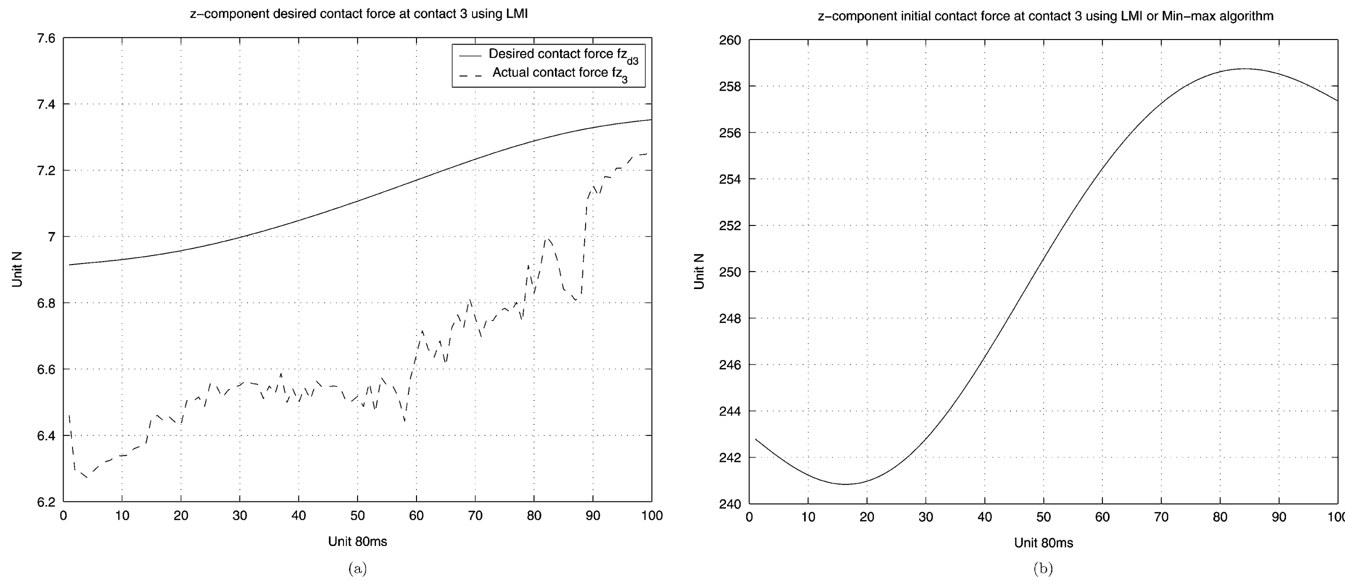


Fig. 11. Finger 3: (a) z-component contact force response. (b) z-component initial contact force.

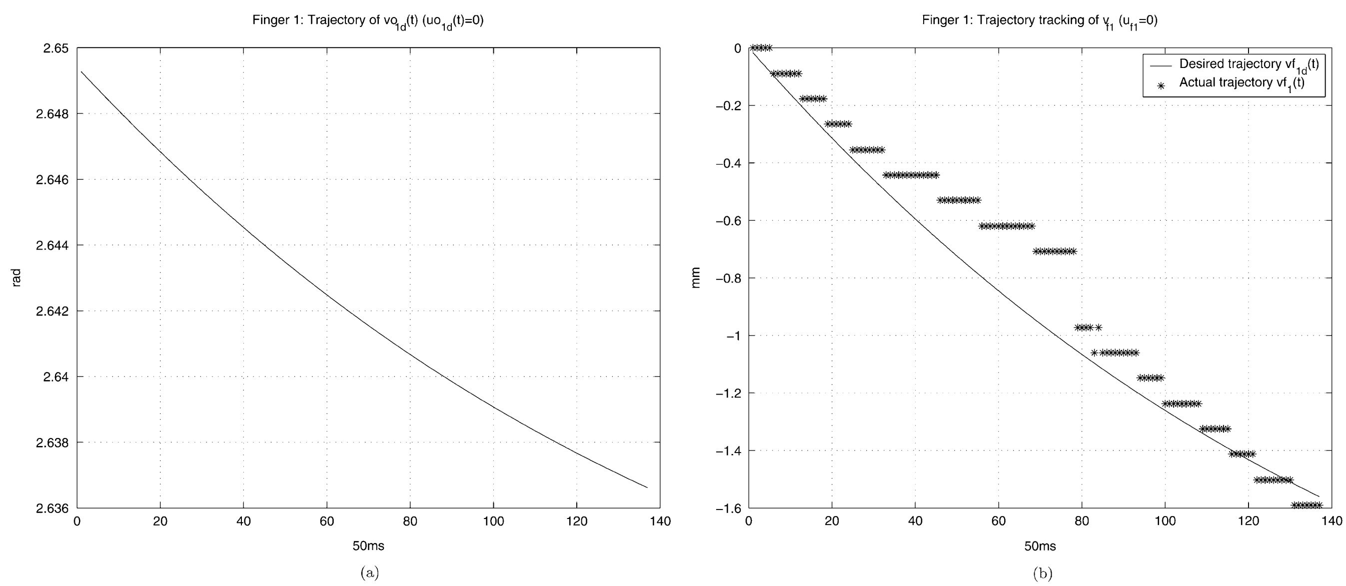


Fig. 12. (a) Finger 1: Desired trajectory of v_{o1}. (b) Finger 1: Trajectory tracking of v_{f1}.

we also compute a set of constant base matrices $\tilde{G}_0, \ldots, \tilde{G}_{n-6}$. *LMIcall* calls *maxdet-src* for yielding the optimal solutions. We make the final executable file by linking the math library *clapack* written in **C** with the object files of the above two modules using *ld68k*. We choose *ld68k* to generate executable codes which are compatible with Motorola 68040 processors.

B. Experimental Results

First, we test the convergence of the LMI algorithm for grasping-force optimization at a fixed contact location. The following parameters are used in the experiments. The weight and radius of the spherical object are 600 g and 110 mm, respectively. $e_0 = (0,0,1,0,0,1,0,0,1)^T$ (from which e can be calculated), and PCWF is assumed for all three contacts with $\mu_i = 0.5$. The local coordinates of three contact points are given by $\alpha_{o1} = (0, 2.775)$, $\alpha_{o2} = (0, -1.885)$, and $\alpha_{o3} = (0, 0.8377)$. The absolute tolerance is set to be $1e-5$, and the relative tolerance $1e-10$. Fig. 7(a) and (b), and Fig. 8(a) gives the trajectories of the three contact force vectors at the three contacts, and Fig. 8(b) depicts the cost function ϕ. The forces are shown to converge to their optimal values within six steps.

Second, we compare the computation time for an optimal grasping force by the BHM algorithm without Li and Qin's refinement, the BHM algorithm with Li and Qin's refinement, and the LMI algorithm. The results are shown in Table I. It should be noted that the computation time in Table I also includes the part on kinematics computation and trajectory generation. The computation time by the LMI algorithm is about 70 ms, the fastest among the three algorithms. It should be pointed out that the current system runs slowly even with the LMI algorithm. In future works, we will replace the old Motorola 68 040 processors by other more faster ones to reduce the computation time by the three algorithms.

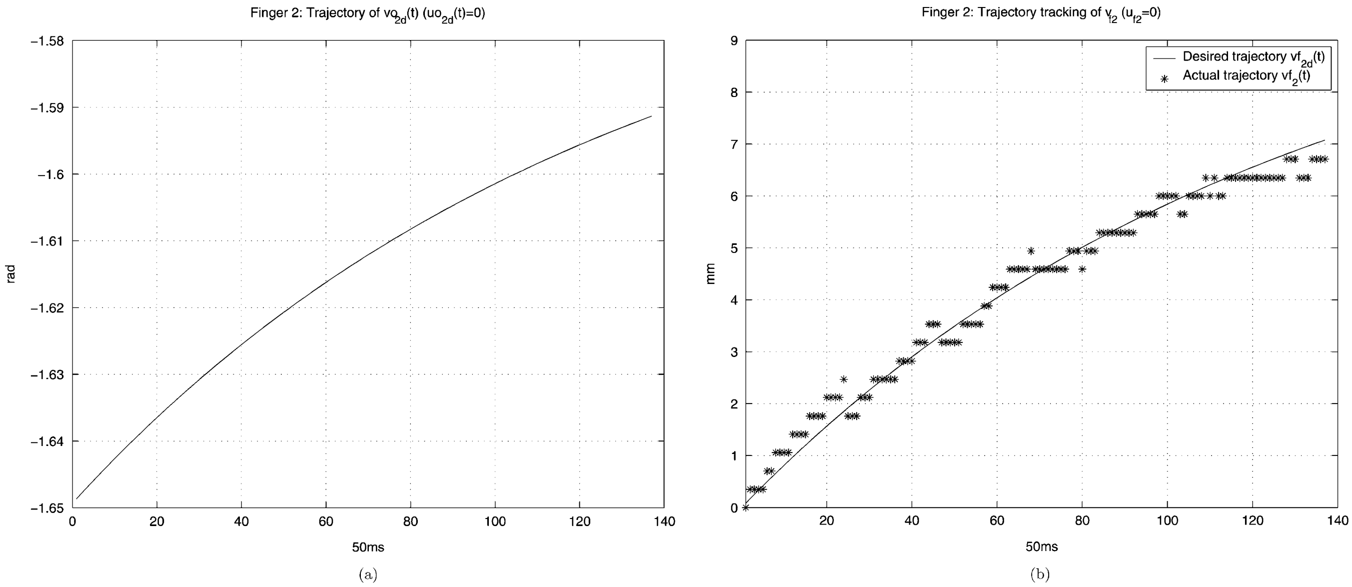


Fig. 13. (a) Finger 2: Desired trajectory of v_{o2}. (b) Finger 2: Trajectory tracking of v_{f2}.

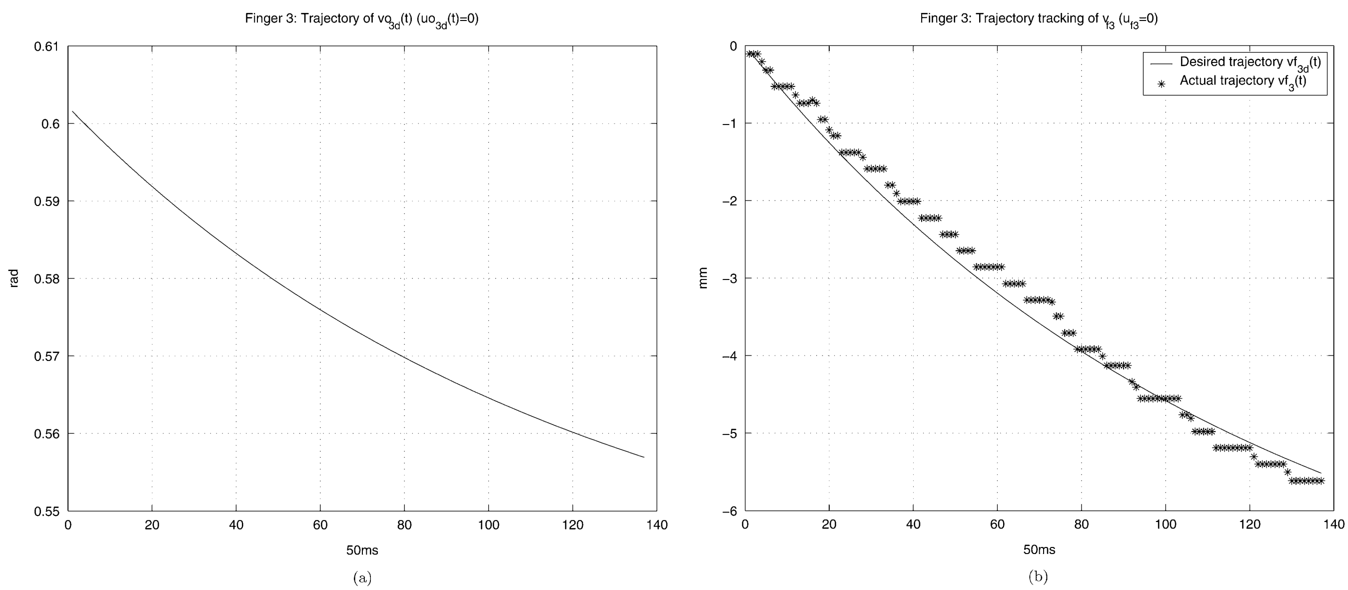
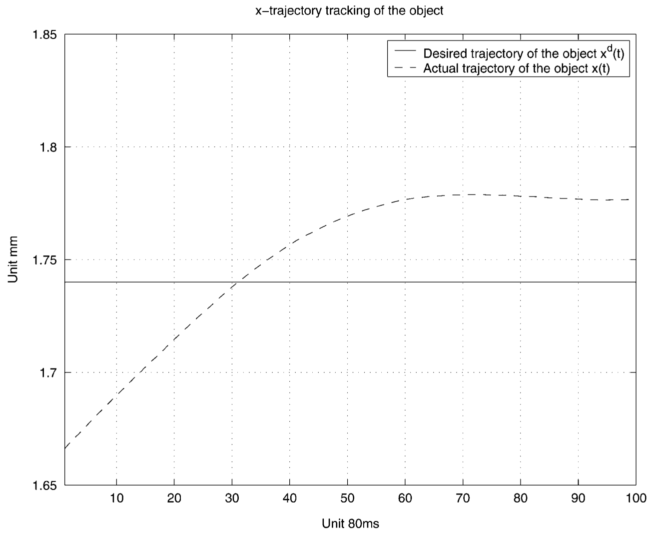


Fig. 14. (a) Finger 3: Desired trajectory of v_{o3}. (b) Finger 3: Trajectory tracking of v_{f3}.

Third, we verify the performance of the integrated algorithm in manipulating an object along a desired trajectory under rolling contact. In the experiment, the object is required to move 100 mm along the twist $(0, 0, 1, 0, 0, 0)$ in 8 s. Fig. 9(a), Fig. 10(a), and Fig. 11(a) show the z-component contact force response of the three contacts. It should be noted that the curves of the desired contact forces shown in these figures are given by optimal contact forces at the three contact curves. Fig. 9(b), Fig. 10(b), and Fig. 11(b) show the initial z-component contact forces of the three fingers, from either the LMI algorithm or the gradient algorithm. Note that the z-component contact force is significantly reduced through the optimization algorithm since we only consider its contribution in the cost function by choosing e_0. Figs. 12–14 show the contact coordinates response of the three contacts which tend to locate at three symmetric points (regarded as the optimal grasp configuration) of the great circle $u_o = 0$. $\psi_{id}, i = 1, \ldots, 3$, are planned to be constant in the experiments. Finally, the trajectory tracking results of the manipulated object are shown in Figs. 15 and 16. It is clearly shown from the experimental results that the tracking error of contact forces is less than 0.8 N, and the object displacement error is less than 0.9 mm.

VI. Conclusion

In this paper, an integrated algorithm for the real-time grasping-force optimization problem is developed based on the works of Buss *et al.*, Li and Qin, and Han *et al.*. The problem of automatically generating appropriate initial points in real-time encountered in these algorithms is solved through applying the LMI algorithm in combination with the gradient algorithm. Efficiency of this algorithm for manipulation tasks is verified by experiments with the HKUST three-fingered hand.



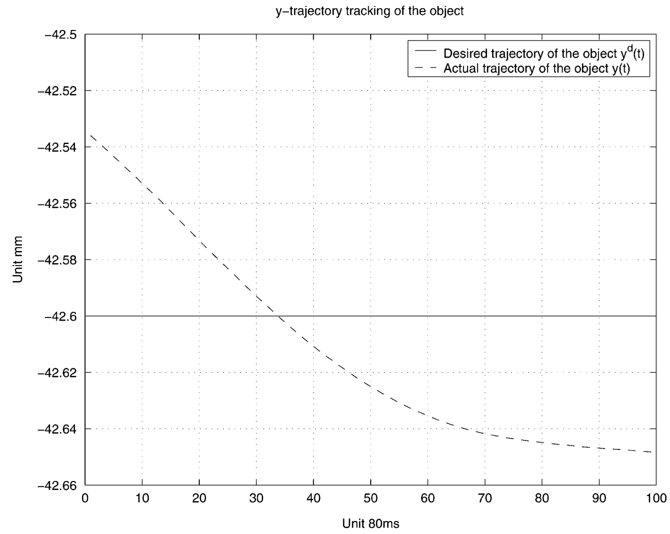
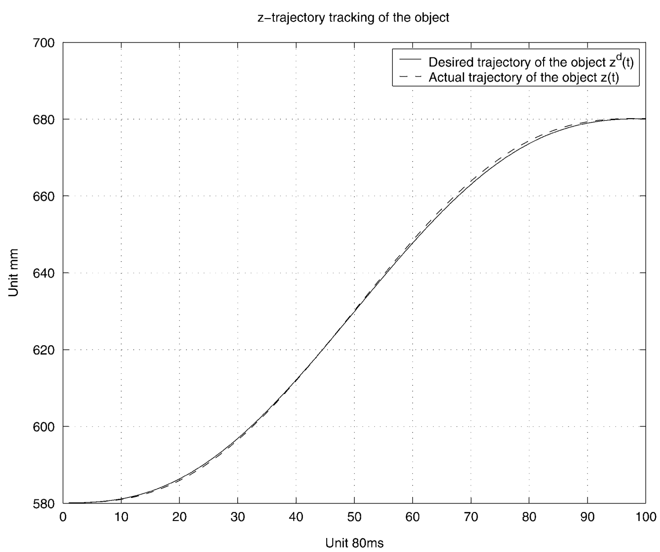


Fig. 15. (a) x-trajectory tracking of the object. (b) y-trajectory tracking of the object.



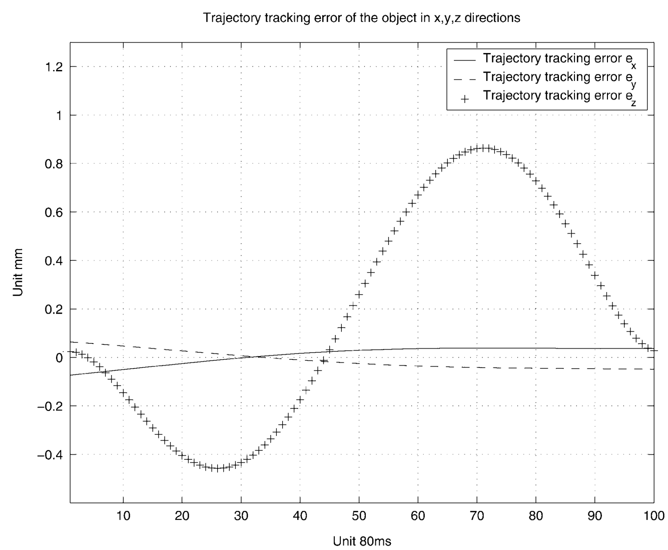


Fig. 16. (a) z-trajectory tracking of the object. (b) x, y, z-trajectory tracking errors of the object.

APPENDIX
MAX–DET PROBLEM: GENERAL DEFINITION

Consider the optimization problem

$$\min c^T x + \log\det G(x)^{-1} \tag{25}$$
$$\text{subject to}$$
$$G(x) > 0$$
$$F(x) \geq 0$$

where $x \in \mathbb{R}^m$ is the optimization variable. Functions $G : \mathbb{R}^m \rightarrow \mathbb{R}^{l\times l}$ and $F : \mathbb{R}^m \rightarrow \mathbb{R}^{n\times n}$ are affine

$$G(x) = G_0 + G_1x_1 + \cdots + G_mx_m$$
$$F(x) = F_0 + F_1x_1 + \cdots + F_mx_m$$

where $G_i = G_i^T$ and $F_i = F_i^T$ with $\mathrm{diag}(G_i, F_i), i = 1, \ldots, m$, being linearly independent. The inequalities in (25) denote matrix inequalities. We refer to problem (25) as a max–det problem subject to LMIs.

REFERENCES

[1] M. Mason and K. Salisbury, *Robot Hands and the Mechanics of Manipulation*. Cambridge, MA: MIT Press, 1985.

[2] J. Butterfass, M. Grebenstein, H. Liu, and G. Hirzinger, "DLR-hand II: Next generation of a dextrous robot hand," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2001, pp. 109–114.

[3] T. Okada, "Object handling system for manual industry," *IEEE Trans. Syst. Man, Cybern.*, vol. SMC-12, pp. 289–299, 1979.

[4] S. C. Jacobsen *et al.*, "Design of the Utah/MIT dextrous hand," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1986, pp. 1520–1532.

[5] C. S. Loucks *et al.*, "Modeling and control of the stanford/JPL hand," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1987, pp. 573–578.

[6] R. Tomovic *et al.*, "A strategy for grasp synthesis with multifingered robot hand," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1987, pp. 83–89.

[7] G. Woehlke, "A programming and simulation environment for the Karlsruhe dexterous hand," *J. Robot. Autonom. Syst.*, vol. 9, pp. 243–262, 1990.

[8] R. Menzel, K. Woelfl, and F. Pfeiffer, “Grasping with a dextrous robot hand,” in *Proc. IFAC Symp. Robot Control*, 1994, pp. 303–308.
[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] Z. X. Li, P. Hsu, and S. Sastry, “On grasping and coordinated manipulation by a multifingered robot hand,” *Int. J. Robot. Res.*, vol. 8, no. 4, pp. 33–50, 1989.
[11] A. Cole, J. Hauser, and S. Sastry, “Kinematics and control of multifingered robot hands with rolling contact,” *IEEE Trans. Automat. Contr.*, vol. 34, pp. 398–404, Apr. 1989.
[12] E. Paljug, X. Yun, and V. Kummar, “Control of rolling contacts in multi-arm manipulation,” *IEEE Trans. Robot. Automat.*, vol. 10, pp. 441–452, Aug. 1994.
[13] N. Sarkar, X. P. Yun, and V. Kumar, “Dynamic control of 3-D rolling contacts in two-arm manipulation,” *IEEE Trans. Robot. Automat.*, vol. 13, pp. 354–376, Aug. 1997.
[14] L. Han, Y. Guan, Z. X. Li, Q. Shi, and J. Trinkle, “Dextrous manipulation with rolling contacts,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 1997, pp. 992–997.
[15] A. Bicchi and R. Sorrentino, “Dexterous manipulation through rolling,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 1995, pp. 452–457.
[16] A. Cole, P. Hsu, and S. Sastry, “Dynamic control of sliding by robot hands for regrasping,” *IEEE Trans. Robot. Automat.*, vol. 8, pp. 42–52, Feb. 1992.
[17] X. Z. Zheng, R. Nakashima, and T. Yoshikawa, “On dynamic control of finger sliding and object motion in manipulation with multifingered hands,” *IEEE Trans. Robot. Automat.*, vol. 16, pp. 469–481, Oct. 2000.
[18] B. Goodwine and J. Burdick, “Controllability of kinematic control systems on stratified configuration spaces,” *IEEE Trans. Automat. Contr.*, vol. 46, pp. 358–368, Mar. 2001.
[19] T. Schlegl and M. Buss, “Hybrid closed-loop control of robotic hand regrasping,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 1998, pp. 3026–3032.
[20] H. Zhang, K. Tanie, and H. Maekawa, “Dextrous manipulation planning by graph transformation,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 1996, pp. 3055–3060.
[21] J. K. Salisbury and J. Craig, “Articulated hands: Force control and kinematic issues,” *Int. J. Robot. Res.*, vol. 1, no. 1, pp. 4–17, 1982.
[22] J. Kerr and B. Roth, “Analysis of multifingered hands,” *Int. J. Robot. Res.*, vol. 4, no. 4, pp. 3–17, 1986.
[23] M. Cutkosky, *Robotic Grasping and Fine Manipulation*. Boston, MA: Kluwer, 1985.
[24] D. Montana, “The kinematics of contact and grasp,” *Int. J.Robot. Res.*, vol. 7, no. 3, pp. 17–32, 1988.
[25] ——, “The kinematics of multi-fingered manipulation,” *IEEE Trans. Robot. Automat.*, vol. 11, pp. 491–503, Aug. 1995.
[26] C. Cai and B. Roth, “On the spatial motion of rigid bodies with point contact,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 1987, pp. 686–695.
[27] Z. Li and J. Canny, “Motion of two rigid bodies with rolling constraint,” *IEEE Trans. Robot. Automat.*, vol. 6, no. 1, pp. 62–72, Feb. 1990.
[28] R. Murray, Z. X. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC, 1994.
[29] A. Bicchi, “Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity,” *IEEE Trans. Robot. Automat.*, vol. 16, pp. 652–662, Dec. 2000.
[30] A. M. Okamura, N. Smaby, and M. R. Cutkosky, “An overview of dexterous manipulation,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 2000, pp. 255–262.
[31] F. T. Cheng and D. E. Orin, “Efficient algorithm for optimal force distribution—The compact-dual LP method,” *IEEE Trans. Robot. Automat.*, vol. 6, pp. 178–187, Apr. 1990.
[32] J. Jameson and L. Leifer, “Automatic grasping: An optimization approach,” *IEEE Trans. Syst., Man, Cybernet.*, vol. SMC-17, pp. 806–814, Sept./Oct. 1987.
[33] P. R. Sinha and J. M. Abel, “A contact stress model for multifingered grasps of rough objects,” *IEEE Trans. Robot. Automat.*, vol. 8, pp. 7–22, Feb. 1992.
[34] V. Kumar and K. Waldron, “Sub-optimal algorithms for force distribution in multifingered grippers,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 1987, pp. 252–257.
[35] V. M. Kvrgic, “Computing of the sub-optimal grasping forces for manipulation of a rough object by multifingered robot hand,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 1996, pp. 1801–1806.
[36] M. Buss, H. Hashimoto, and J. Moore, “Dextrous hand grasping force optimization,” *IEEE Trans. Robot. Automat.*, vol. 12, pp. 406–418, June 1996.
[37] M. Buss, L. Faybusovich, and J. Moore, “Recursive algorithms for real-time grasping force optimization,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 1997, pp. 682–687.
[38] ——, “Dikin-type algorithms for dextrous grasping force optimization,” *Int. J. Robot. Res.*, vol. 17, no. 8, pp. 831–839, 1998.
[39] Z. X. Li, Z. Qin, S. Jiang, and L. Han, “Coordinated motion generation and real-time grasping force control for multifingered manipulation,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 1998, pp. 3632–3638.
[40] L. Han, J. Trinkle, and Z. X. Li, “Grasp analysis as linear matrix inequalities,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 1999, pp. 1261–1268.
[41] L. Han, J. C. Trinkle, and Z. X. Li, “Grasp analysis as linear matrix inequality problems,” *IEEE Trans. Robot. Automat.*, vol. 16, pp. 663–674, Dec. 2000.
[42] L. Vandenberghe, S. Boyd, and S.-P. Wu, “Determinant maximization with linear matrix inequality constraints,” *SIAM J. Matrix Anal. Applicat.*, vol. 19, no. 2, pp. 499–533, 1998.
[43] Y. Nesterov and A. Nemirovsky, *Interior-Point Polynominal Methods in Convex Programming*. Philadelphia, PA: SIAM, 1994.
[44] U. Helmke and J. Moore, *Optimization of Dynamical Systems*. New York: Springer-Verlag, 1993.
[45] S. L. Jiang, K. K. Choi, and Z. X. Li, “Coordinated motion generation for multifingered manipulation using tactile feedback,” in *Proc. IEEE Int. Conf. Robotics and Automation*, 1999, pp. 3032–3037.

Guanfeng Liu received the B.E. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 1998, and the Ph.D. degree in electrical and electronic engineering from the Hong Kong University of Science and Technology, Hong Kong, in February, 2003.

He has been a Postdoctoral Researcher in the Computer Scienc Department, Rensselaer Polytechnic Institute, Troy, NY, since Autumn, 2003. His research interests include multifingered manipulation, parallel manipulators, hybrid and nonlinear system theory.

Zexiang Li received the B.S. degree in electrical engineering and economics (with honors) from Carnegie Mellon University, Pittsburgh, PA, in 1983, and the M.A. degree in mathematics and Ph.D. degree in electrical engineering and computer science, both from the University of California at Berkeley, in 1985 and 1989, respectively.

He is an Associate Professor in the Electrical and Electronics Engineerig Department, Hong Kong University of Science and Technology, Hong Kong. His research interests include robotics, nonlinear system theory, and manufacturing.