Proceedings of the 2003 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Las Vegas, Nevada · October 2003

Automatic Determination of Finger Control Modes for Graspless Manipulation

Yusuke MAEDA *and* Tamio ARAI

Department of Precision Engineering, School of Engineering
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, JAPAN
maeda@prince.pe.u-tokyo.ac.jp

***Abstract*— To achieve various manipulation tasks by robot fingers, we have to use both position control and force control appropriately. In this paper, we propose a method to determine such control modes of robot fingers in graspless manipulation. Using the method, we can assign a position- or force-control mode to each robot finger so that the maximum manipulation stability will be achieved without excessive internal force. Desired finger forces are also determined for robot fingers to be force-controlled. We show some numerical examples of automatic determination of control modes in graspless manipulation of polyhedra on a plane by two robot fingers.**

I. INTRODUCTION

Graspless manipulation [1] (or nonprehensile manipulation [2]) is a method to manipulate objects without grasping. In this paper, we deal with graspless manipulation where the manipulated object is supported not only by robot fingers but also by the environment (Fig. 1). Such contact tasks are usually performed by force-controlled robots to avoid excessive internal force. In some cases, however, force control is inappropriate in terms of manipulation stability; even minute disturbance could perturb the path of the manipulated object. Pushing operation on a plane is a typical example and therefore usually performed by a position-controlled pusher (for example, "stable push" [3]). Thus we have to use both position control and force control appropriately to achieve various robotic graspless manipulation.

In this paper, graspless manipulation by multiple robot fingers is studied. We develop an algorithm to determine control modes of robot fingers automatically; we can choose whether each robot finger should be position-controlled or force-controlled with this algorithm. Desired finger forces are also determined for force-controlled fingers. The validity of this algorithm is assessed by numerical examples of typical graspless operations.

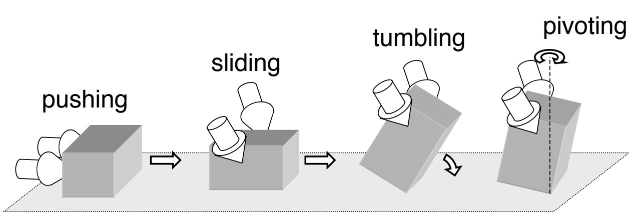

Fig. 1. Graspless Manipulation

II. MODEL OF GRASPLESS MANIPULATION

A. Assumptions

In this paper, we make the following assumptions:

- The manipulated object, robot fingertips, and the environment are rigid.
- Manipulation is quasi-static.
- Coulomb friction exists between the object and the environment (or robot fingers).
- Static and kinetic friction coefficients are equal.
- All the contacts can be approximated by finite point contacts [4].
- All the friction cones can be approximated by polyhedral convex cones [5].
- Each robot finger is in one-point non-sliding contact with the object.
- The normal component of each finger force has an upper limit.
- Each robot finger is either in position-control mode or in force-control mode.
- A robot finger in position-control mode can apply arbitrary force within its friction cone passively.
- A robot finger in force-control mode is in hybrid position/force control [6]; the finger can apply the commanded normal force actively and apply arbitrary tangential force within its friction cone passively.

The problem to be solved is to determine whether each robot finger should be position-controlled or force-controlled, and moreover, to determine the desired normal forces for force-controlled fingers, when the desired object motion and the positions of the robot fingertips on the object are specified.

In this paper, we deal with mechanical analysis of graspless manipulation at an instant; therefore commanded

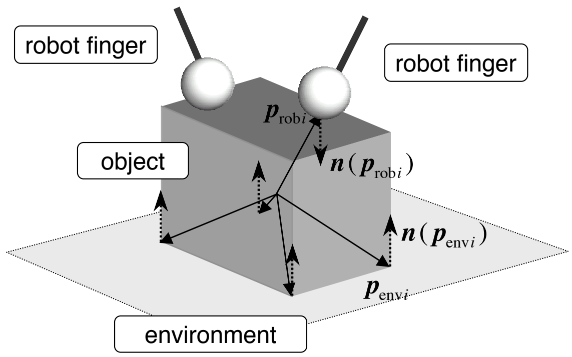

Fig. 2. Object in Graspless Manipulation

positions of robot fingers are regarded as constant. In real manipulation, however, commanded positions of robot fingers are updated step by step based on the desired object motion. Accordingly, even position-controlled robot fingers could manipulate the object.

B. Mechanical Model

Consider graspless manipulation of an object as in Fig. 2. We set an object reference frame whose origin coincides with the center of mass of the object. Let $\boldsymbol{p}_{\mathrm{env}\,1}, \ldots, \boldsymbol{p}_{\mathrm{env}\,m} \in \Re^3$ be positions of contact points between the object and the environment. Similarly, let $\boldsymbol{p}_{\mathrm{rob}\,1}, \ldots, \boldsymbol{p}_{\mathrm{rob}\,n} \in \Re^3$ be positions of contact points between the object and the robot finger $1, \ldots, n$. We denote inward unit normal vectors at contact point $\boldsymbol{p}$ by $\boldsymbol{n}(\boldsymbol{p}) \in \Re^3$.

Let us denote the sets of positions of sliding and non-sliding contacts by $\mathcal{C}_{\mathrm{slide}}$ and $\mathcal{C}_{\mathrm{stat}}$, respectively. We can identify whether $\boldsymbol{p}_{\mathrm{env}\,i} \in \mathcal{C}_{\mathrm{slide}}$ or $\boldsymbol{p}_{\mathrm{env}\,i} \in \mathcal{C}_{\mathrm{stat}}$, because the object motion is specified. We approximate each friction cone at contact point $\boldsymbol{p}$ by a polyhedral convex cone with unit edge vectors, $\boldsymbol{c}_1(\boldsymbol{p}), \ldots, \boldsymbol{c}_s(\boldsymbol{p}) \in \Re^3$. For $\boldsymbol{p}_{\mathrm{env}\,i} \in \mathcal{C}_{\mathrm{slide}}$, let $\boldsymbol{c}'(\boldsymbol{p}_{\mathrm{env}\,i}) \in \Re^3$ be a unit edge vector of the friction cone at contact point $\boldsymbol{p}_{\mathrm{env}\,i}$ opposite to its sliding direction.

The set of possible contact force $\boldsymbol{f} \in \Re^3$ at $\boldsymbol{p}_{\mathrm{env}\,i}$ can be written as follows:

$$\begin{cases} \{\boldsymbol{f} | \boldsymbol{f} \in \mathrm{span}\{\boldsymbol{c}_1(\boldsymbol{p}_{\mathrm{env}\,i}) \ldots \boldsymbol{c}_s(\boldsymbol{p}_{\mathrm{env}\,i})\}\} \\ \qquad\qquad\qquad \text{if } \boldsymbol{p}_{\mathrm{env}\,i} \in \mathcal{C}_{\mathrm{stat}}, \\ \{\boldsymbol{f} | \boldsymbol{f} \in \mathrm{span}\{\boldsymbol{c}'(\boldsymbol{p}_{\mathrm{env}\,i})\}\} \\ \qquad\qquad\qquad \text{if } \boldsymbol{p}_{\mathrm{env}\,i} \in \mathcal{C}_{\mathrm{slide}}, \end{cases} \tag{1}$$

where $\mathrm{span}\{\ldots\}$ is a polyhedral convex cone spanned by its element vectors [5]. On the other hand, the set of possible contact force $\boldsymbol{f}$ at $\boldsymbol{p}_{\mathrm{rob}\,i}$ is:

$$\begin{cases} \{\boldsymbol{f} | \boldsymbol{f} \in \mathrm{span}\{\boldsymbol{c}_1(\boldsymbol{p}_{\mathrm{rob}\,i}) \ldots \boldsymbol{c}_s(\boldsymbol{p}_{\mathrm{rob}\,i})\}, \\ \qquad\qquad \boldsymbol{n}(\boldsymbol{p}_{\mathrm{rob}\,i})^T \boldsymbol{f} \le f_{\max\,i}\} \\ \quad \text{if robot finger } i \text{ is position-controlled}, \\ \{\boldsymbol{f} | \boldsymbol{f} \in \mathrm{span}\{\boldsymbol{c}_1(\boldsymbol{p}_{\mathrm{rob}\,i}) \ldots \boldsymbol{c}_s(\boldsymbol{p}_{\mathrm{rob}\,i})\}, \\ \qquad\qquad \boldsymbol{n}(\boldsymbol{p}_{\mathrm{rob}\,i})^T \boldsymbol{f} = f_{\mathrm{com}\,i} \le f_{\max\,i}\} \\ \quad \text{if robot finger } i \text{ is force-controlled}, \end{cases} \tag{2}$$

where $f_{\max\,i}$ is the upper limit of normal force and $f_{\mathrm{com}\,i}$ is the commanded normal force for robot finger i.

Then we define the following matrices:

$$\begin{aligned}
\boldsymbol{W}_{\mathrm{env}} &:= \begin{bmatrix} \boldsymbol{I}_3 & \ldots & \boldsymbol{I}_3 \\ \boldsymbol{p}_{\mathrm{env}\,1} \times \boldsymbol{I}_3 & \ldots & \boldsymbol{p}_{\mathrm{env}\,m} \times \boldsymbol{I}_3 \end{bmatrix} \in \Re^{6 \times 3m} \\
\boldsymbol{C}_{\mathrm{env}} &:= \mathrm{diag}(\boldsymbol{C}_{\mathrm{env}\,1}, \ldots, \boldsymbol{C}_{\mathrm{env}\,m}) \\
\boldsymbol{C}_{\mathrm{env}\,i} &:= \begin{cases} [\boldsymbol{c}_1(\boldsymbol{p}_{\mathrm{env}\,i}) \ldots \boldsymbol{c}_s(\boldsymbol{p}_{\mathrm{env}\,i})] \in \Re^{3 \times s} \\ \qquad\qquad \text{if } \boldsymbol{p}_{\mathrm{env}\,i} \in \mathcal{C}_{\mathrm{stat}} \\ [\boldsymbol{c}'(\boldsymbol{p}_{\mathrm{env}\,i})] \in \Re^{3 \times 1} \\ \qquad\qquad \text{if } \boldsymbol{p}_{\mathrm{env}\,i} \in \mathcal{C}_{\mathrm{slide}} \end{cases} \\
\boldsymbol{W}_{\mathrm{rob}} &:= \begin{bmatrix} \boldsymbol{I}_3 & \ldots & \boldsymbol{I}_3 \\ \boldsymbol{p}_{\mathrm{rob}\,1} \times \boldsymbol{I}_3 & \ldots & \boldsymbol{p}_{\mathrm{rob}\,n} \times \boldsymbol{I}_3 \end{bmatrix} \in \Re^{6 \times 3n} \\
\boldsymbol{C}_{\mathrm{rob}} &:= \mathrm{diag}(\boldsymbol{C}_{\mathrm{rob}\,1}, \ldots, \boldsymbol{C}_{\mathrm{rob}\,n}) \in \Re^{3n \times ns} \\
\boldsymbol{C}_{\mathrm{rob}\,i} &:= [\boldsymbol{c}_1(\boldsymbol{p}_{\mathrm{rob}\,i}) \ldots \boldsymbol{c}_s(\boldsymbol{p}_{\mathrm{rob}\,i})] \in \Re^{3 \times s} \\
\boldsymbol{N}_{\mathrm{rob}} &:= \mathrm{diag}(\boldsymbol{n}(\boldsymbol{p}_{\mathrm{rob}\,1}), \ldots, \boldsymbol{n}(\boldsymbol{p}_{\mathrm{rob}\,n})) \in \Re^{3n \times n},
\end{aligned}$$

where $\boldsymbol{I}_3$ is the 3×3 identity matrix, and $\boldsymbol{p} \times \boldsymbol{I}_3 \in \Re^{3 \times 3}$ is a linear transformation equivalent to cross product with $\boldsymbol{p}$.

Without external disturbances, the equilibrium equation of the object can be expressed as:

$$\boldsymbol{Q}_{\mathrm{known}} + \boldsymbol{W}_{\mathrm{env}} \boldsymbol{C}_{\mathrm{env}} \boldsymbol{k}_{\mathrm{env}} + \boldsymbol{W}_{\mathrm{rob}} \boldsymbol{C}_{\mathrm{rob}} \boldsymbol{k}_{\mathrm{rob}} = \boldsymbol{0}, \tag{3}$$

where $\boldsymbol{k}_{\mathrm{env}} (\ge \boldsymbol{0})$ and $\boldsymbol{k}_{\mathrm{rob}} (\ge \boldsymbol{0})$ are coefficient vectors to represent contact forces; $\boldsymbol{Q}_{\mathrm{known}} \in \Re^6$ is the known external (generalized) force applied to the object such as gravity. The limitation on the magnitude of normal finger forces can be written as:

$$\boldsymbol{N}_{\mathrm{rob}}^T \boldsymbol{C}_{\mathrm{rob}} \boldsymbol{k}_{\mathrm{rob}} \le \boldsymbol{f}_{\max}, \tag{4}$$

where $\boldsymbol{f}_{\max} = [f_{\max\,1}, \ldots, f_{\max\,n}]^T \in \Re^n$.

III. Determination of Finger Control Modes

A. Basic Idea

To determine "appropriate" control modes of robot fingers, we need a principle for determination. For graspless manipulation, we have a quantitative stability measure [4]. The measure can be regarded as a performance index for "quasi-static closure" [7]; it evaluates the stability of graspless manipulation in terms of the magnitude of disturbing (generalized) force that the object can resist without changing its motion. Here, we determine the finger control modes to maximize the stability measure, because graspless manipulation is usually less robust than manipulation by grasping.

In order to increase the value of the stability measure, position control is preferable to force control. This is because a robot finger in position control can apply a wider variety of finger forces passively than that in force control. Position-controlled fingers, however, may generate excessive internal force in contact tasks (Fig. 3).

Thus, we select a combination of control modes for all the robot fingers that achieves the maximum manipulation stability, as far as excessive internal force could not be generated. Hereafter we develop a procedure to select the "optimal" combination of finger control modes (and desired normal forces for force-controlled fingers).

B. Stability Measure for Graspless Manipulation

The value of manipulation stability defined in [4] can be calculated approximately by solving a series of linear programming problems. If we assume that all the combinations of possible contact forces are also possible, the stability value z can be calculated by the following single linear programming problem:

$$
\begin{aligned}
&\text{maximize } z \\
&\text{subject to} \\
&\begin{cases}
z\boldsymbol{l}_1 = \boldsymbol{R}^{1/2}\left(\boldsymbol{Q}_{\text{known}} + \boldsymbol{W}_{\text{env}}\boldsymbol{C}_{\text{env}}\boldsymbol{k}_{\text{env}\,1} + \boldsymbol{W}_{\text{rob}}\boldsymbol{C}_{\text{rob}}\boldsymbol{k}_{\text{rob}\,1}\right) \\
\quad\vdots \\
z\boldsymbol{l}_N = \boldsymbol{R}^{1/2}\left(\boldsymbol{Q}_{\text{known}} + \boldsymbol{W}_{\text{env}}\boldsymbol{C}_{\text{env}}\boldsymbol{k}_{\text{env}\,N} + \boldsymbol{W}_{\text{rob}}\boldsymbol{C}_{\text{rob}}\boldsymbol{k}_{\text{rob}\,N}\right) \\
\boldsymbol{N}_{\text{rob}}^T\boldsymbol{C}_{\text{rob}}\boldsymbol{k}_{\text{rob}\,1} \le \boldsymbol{f}_{\max} \\
\quad\vdots \\
\boldsymbol{N}_{\text{rob}}^T\boldsymbol{C}_{\text{rob}}\boldsymbol{k}_{\text{rob}\,N} \le \boldsymbol{f}_{\max} \\
\boldsymbol{N}_{\text{rob}}^T\boldsymbol{A}_{\text{force}}\boldsymbol{C}_{\text{rob}}\boldsymbol{k}_{\text{rob}\,1} = \boldsymbol{f}_{\text{com}} \\
\quad\vdots \\
\boldsymbol{N}_{\text{rob}}^T\boldsymbol{A}_{\text{force}}\boldsymbol{C}_{\text{rob}}\boldsymbol{k}_{\text{rob}\,N} = \boldsymbol{f}_{\text{com}} \\
\boldsymbol{k}_{\text{env}\,1}, \ldots, \boldsymbol{k}_{\text{env}\,N} \ge \boldsymbol{0} \\
\boldsymbol{k}_{\text{rob}\,1}, \ldots, \boldsymbol{k}_{\text{rob}\,N} \ge \boldsymbol{0},
\end{cases}
\end{aligned}
\tag{5}
$$

where $\boldsymbol{f}_{\text{com}} = [f_{\text{com}\,1}, \ldots, f_{\text{com}\,n}]^T \in \Re^n$ and $f_{\text{com}\,i} = 0$ if finger i is position-controlled; $\boldsymbol{k}_{\text{env}\,i}$ and $\boldsymbol{k}_{\text{rob}\,i}$ are coefficient vectors to represent contact forces; $\boldsymbol{A}_{\text{force}}$ is a selection matrix defined as:

$$
\boldsymbol{A}_{\text{force}} := \operatorname{diag}(a_1, a_1, a_1, \ldots, a_n, a_n, a_n) \in \Re^{3n\times 3n}
$$

$$
a_i := \begin{cases} 1 & \text{if finger } i \text{ is force-controlled} \\ 0 & \text{if finger } i \text{ is position-controlled;} \end{cases}
$$

$\boldsymbol{l}_1, \ldots, \boldsymbol{l}_N \in \Re^6$ are position vectors of vertices of a hyperpolyhedron circumscribed to the six-dimensional unit hypersphere, which are used for approximate calculation [4]; $\boldsymbol{R} \in \Re^{6\times 6}$ is a positive definite matrix for scaling of force and moment. $\boldsymbol{R}^{1/2} \in \Re^{6\times 6}$ is the Cholesky decomposition of $\boldsymbol{R}$, which is used to introduce the following norm for generalized forces, $\boldsymbol{Q} \in \Re^6$:

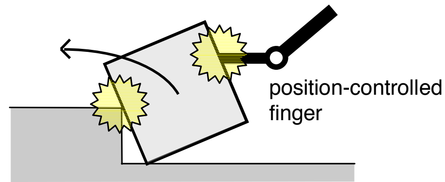

Fig. 3. Excessive Internal Force in Manipulation

$$
\|\boldsymbol{Q}\|_{\boldsymbol{R}} := \sqrt{\boldsymbol{Q}^T\boldsymbol{R}\boldsymbol{Q}} = \sqrt{\left(\boldsymbol{R}^{1/2}\boldsymbol{Q}\right)^T\left(\boldsymbol{R}^{1/2}\boldsymbol{Q}\right)}. \tag{6}
$$

We can have a coordinate-invariant norm by using, for example, the following scaling matrix:

$$
\boldsymbol{R} := \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{O} \\ \boldsymbol{O} & M\boldsymbol{J}^{-1} \end{bmatrix} \in \Re^{6\times 6}, \tag{7}
$$

where M is the mass of the object and $\boldsymbol{J} \in \Re^{3\times 3}$ is the inertia tensor of the object. Regarding $\boldsymbol{A}_{\text{force}}$ as constant and $\boldsymbol{f}_{\text{com}}$ as variable, we can find $\boldsymbol{f}_{\text{com}}$ that achieves the maximum manipulation stability for $\boldsymbol{A}_{\text{force}}$ by solving the linear programming problem (5).

As mentioned above, we have introduced an assumption that all the combinations of possible contact forces are also possible. Of course, this simplification is optimistic. When friction is small, however, the above simplification does not have big effect on the value of stability measure. This is because a resultant force of an impossible combination of contact forces is often equivalent to a resultant force of a possible combination of contact forces. Therefore, in this paper, we use the linear programming problem (5) to evaluate the stability of graspless manipulation. For cases with large friction, more conservative estimation of possible contact forces [8]–[11], which would require more complicated calculation, should be employed.

C. Possibility of Excessive Internal Force

The possibility of excessive internal force can be judged by the following linear programming problem [12]:

$$
\begin{aligned}
&\text{maximize } \boldsymbol{b}_{\text{env}}^T\boldsymbol{k}_{\text{env}} + \boldsymbol{b}_{\text{rob}}^T\boldsymbol{k}_{\text{rob}} \\
&\text{subject to} \\
&\begin{cases}
\boldsymbol{W}_{\text{env}}\boldsymbol{C}_{\text{env}}\boldsymbol{k}_{\text{env}} + \boldsymbol{W}_{\text{rob}}\boldsymbol{A}_{\text{pos}}\boldsymbol{C}_{\text{rob}}\boldsymbol{k}_{\text{rob}} = \boldsymbol{0} \\
\boldsymbol{k}_{\text{env}} \ge \boldsymbol{0},\ \boldsymbol{k}_{\text{rob}} \ge \boldsymbol{0},
\end{cases}
\end{aligned}
\tag{8}
$$

where

$$
\begin{aligned}
\boldsymbol{b}_{\text{env}} &= [1, \ldots, 1]^T \\
\boldsymbol{b}_{\text{rob}} &= [\boldsymbol{b}_{\text{rob}\,1}^T, \ldots, \boldsymbol{b}_{\text{rob}\,n}^T]^T \in \Re^{ns}
\end{aligned}
$$

$$\boldsymbol{b}_{\mathrm{rob}\,i} = \begin{cases} [1,\ldots,1]^T \in \Re^s \\ \quad \text{if finger } i \text{ is position-controlled} \\ [0,\ldots,0]^T \in \Re^s \\ \quad \text{if finger } i \text{ is force-controlled} \end{cases}$$

$$\boldsymbol{A}_{\mathrm{pos}} = \boldsymbol{I}_{3n} - \boldsymbol{A}_{\mathrm{force}} \ \in \Re^{3n\times 3n}.$$

When this linear programming problem is bounded, then the magnitude of internal force is also bounded; that is, excessive internal force could not be generated. Otherwise, excessive internal force might be generated.

Note that we can test just a necessary condition for excessive internal force using the linear programming problem (8). Stricter tests might be possible but time-consuming, therefore we use the problem (8) to determine control modes of robot fingers.

D. Procedure for Determining Finger Control Modes

Now we can devise a procedure to determine control modes of robot fingers (and desired normal forces for force-controlled fingers) as follows:

1) Assume a combination of control modes (position control / force control) for each robot finger.
2) Check the possibility of excessive internal force (Problem (8)). If excessive internal force may be generated, give up this combination and go to step 4.
3) Calculate desired normal finger forces ($\boldsymbol{f}_{\mathrm{com}}$) so that the value of manipulation stability, z, will be maximized (Problem (5)). If the maximized z is larger than the current maximum value, update the maximum value.
4) If all the combinations of control modes have been already checked, stop. Otherwise, go back to step 1.

When all the procedure is completed, we can select a combination of control modes that achieves the maximum manipulation stability. If there exist no combinations of control modes with a positive value of manipulation stability, the robot fingers cannot perform the desired object motion stably even against infinitesimal external disturbances.

In the above procedure, we have to solve linear programming problems repeatedly. A naive implementation of testing all the combinations of control modes (2^n patterns) is time-consuming. However, the following characteristics of this problem enable more effective calculation:

1) When changing control modes of one or more fingers from position control to force control, the value of manipulation stability is equal to or smaller than that of the original combination.
2) If the possibility of excessive internal force exists for a combination of finger control modes, there also exists the possibility of excessive internal force for other combinations in which control modes of one or more fingers are changed from force control to position control.

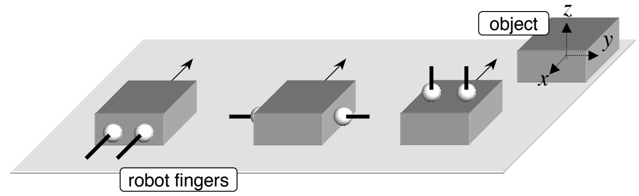

Fig. 4. Sliding a Cuboid

Taking these characteristics into consideration, we can omit to test some combinations of finger control modes. In our current implementation, first we test the case where all the fingers are position-controlled, and then increase the number of force-controlled fingers and check the new combination. This is to reduce the number of times of solving the problem (5), because the problem (5) is much more complex and time-consuming than the problem (8).

IV. Numerical Examples

We implemented the above procedure as a computer program, which uses GLPK (GNU Linear Programming Kit) [13] to solve linear programming problems. Here we show some numerical examples calculated by this program. The computation times for the examples are measured on a Linux PC with Pentium4–1.6GHz.

Let us consider graspless manipulation by two robot fingers. The manipulated object is a polyhedron, whose mass distribution is uniform; the gravitational acceleration is 9.8; the friction coefficient between the object and the environment is 0.2, and that between the object and each finger is 0.5; each friction cone is represented as a polyhedral convex cone with 6 unit edge vectors ($s = 6$); $\boldsymbol{f}_{\max} = [10, 10]^T$.

Here we adopt (7) to calculate the stability measure for graspless manipulation. We approximate the 6-dimensional unit hypersphere as a circumscribed hyperpolyhedron with the following 76 vertices ($N = 76$):

$$\begin{aligned} \{\boldsymbol{l}_i\} := \Big\{ & k[\pm1,0,0,0,0,0]^T,\ k[0,\pm1,0,0,0,0]^T, \\ & k[0,0,\pm1,0,0,0]^T,\ k[0,0,0,\pm1,0,0]^T, \\ & k[0,0,0,0,\pm1,0]^T,\ k[0,0,0,0,0,\pm1]^T, \\ & \frac{k}{\sqrt{6}}[\pm1,\pm1,\pm1,\pm1,\pm1,\pm1]^T \Big\}, \end{aligned} \quad (9)$$

where $k = 2\sqrt{3-\sqrt{6}}$ (≈ 1.48).

A. Example 1: Sliding a Cuboid on a Plane

Suppose sliding a cuboid whose size is $2 \times 2 \times 1$ on a horizontal plane (Fig. 4). The mass of the object is 1 ($M = 1$). The object reference frame is set as shown in the figure.

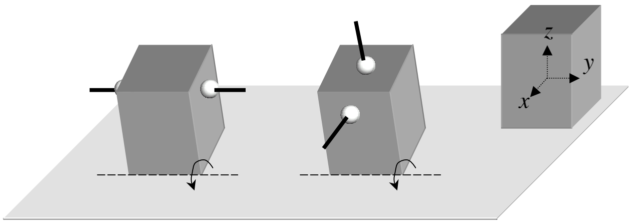

Fig. 5. Tumbling a Cuboid

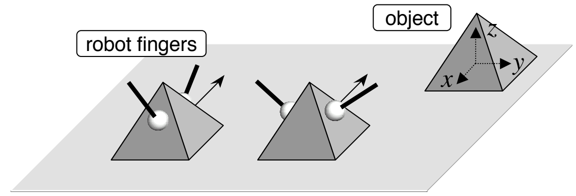

Fig. 6. Sliding a Pyramid

Two robot fingers translate the cuboid in the $(-1, 0, 0)^T$-direction. In this case, $\boldsymbol{J} = \mathrm{diag}(5/12,\ 5/12,\ 2/3)$ and $\boldsymbol{Q}_{\mathrm{known}} = [0,\ 0,\ -9.8,\ 0,\ 0,\ 0]^T$.

In the case of pushing the cuboid at $(1, \pm 1/2, 0)^T$ from behind (Fig. 4, left), our program says that both fingers should be position-controlled. The stability value is 0.6. This case corresponds to Lynch's "stable push" [3], [4].

In the case of pinching the cuboid at $(0, \pm 1, 0)^T$ (Fig. 4, middle), one finger should be position-controlled, and the other should be force-controlled. Note that these control modes can be swapped because of the symmetry. The desired normal force for the force-controlled finger is $6.5 (< f_{\max i})$, and the stability value is 2.4. In this case, the force-controlled finger should push the object at "moderate" force to achieve the maximum stability; we should leave a some margin for its counterpart finger.

In the case of dragging the cuboid at $(0, \pm 1/2, 1/2)^T$ (Fig. 4, right), both fingers should be force-controlled. The desired normal force for both fingers is $10 (= f_{\max i})$; that is, both fingers should push the object at their maximum force because the environment can apply unlimited reaction force. In this case, the stability value is 1.7.

It takes 0.02 through 0.7 CPU seconds for each of above calculation. Most of computation time is dedicated to solving the problem (5).

B. Example 2: Tumbling a Cuboid

Let us consider tumbling a cuboid whose size is $1 \times 1 \times 2$ (Fig. 5). The mass of the object is 1 ($M = 1$). The object reference frame is set as shown in the figure. In this case, $\boldsymbol{J} = \mathrm{diag}(5/12,\ 5/12,\ 1/6)$. When the tilt angle of the object is 30[deg], $\boldsymbol{Q}_{\mathrm{known}} = [4.9,\ 0,\ -8.5,\ 0,\ 0,\ 0]^T$.

In the case of pinching the cuboid at $(0, \pm 1/2, 1/2)^T$ (Fig. 5, left), one finger should be position-controlled, and the other should be force-controlled. These control modes can be swapped because of the symmetry. The desired normal force for the force-controlled finger is 8.8, and the stability value is 2.5. It takes 1.1 CPU seconds for this calculation.

On the other hand, when the fingertips are located at $(0, 0, 1)^T$ and $(1/2, 0, 1/2)^T$ (Fig. 5, right), both fingers should be position-controlled. The stability value is 1.2. It takes 0.01 CPU seconds for this calculation.

Note that these are results of instantaneous analysis; the optimal control modes depend on the tilt angle.

C. Example 3: Sliding a Pyramid

Let us consider sliding a square pyramid whose base edge is 2 and whose height is 2 (Fig. 6). This is an example that fingertips are located on faces that are neither parallel nor perpendicular to each other. The mass of the object is 1 ($M = 1$). The object reference frame is set as shown in the figure. Two robot fingers translate the pyramid in the $(-1, 0, 0)^T$-direction. In this case, $\boldsymbol{J} = \mathrm{diag}(7/20,\ 7/20,\ 2/5)$ and $\boldsymbol{Q}_{\mathrm{known}} = [0,\ 0,\ -9.8,\ 0,\ 0,\ 0]^T$.

In the case of pinching the pyramid at $(\pm 1/2,\ 0,\ 1/2)^T$ (Fig. 6, left), one finger located at $(-1/2,\ 0,\ 1/2)^T$ should be position-controlled, and the other located at $(1/2,\ 0,\ 1/2)^T$ should be force-controlled. The desired normal force for the force-controlled finger is $10 (= f_{\max i})$, and the stability value is 4.0. It takes 0.6 CPU seconds for this calculation.

In the case of pinching the pyramid at $(0,\ \pm 1/2,\ 1/2)^T$ (Fig. 6, right), one finger should be position-controlled, and the other should be force-controlled. These control modes can be swapped because of the symmetry. The desired normal force for the force-controlled finger is $10 (= f_{\max i})$, and the stability value is 2.9. It takes 0.8 CPU seconds for this calculation.

D. Discussion

In the above numerical examples, our algorithm gives "reasonable" results on the determination of control modes of robot fingers. Therefore, we can say that our strategy—selecting a combination of finger control modes that achieves the maximum manipulation stability without excessive internal force—is effective in graspless manipulation. Our strategy uses position control preferably and uses force control as little as possible. This property is also desirable in terms of the ease of manipulation execution by real robot fingers.

We can make some variations of our algorithm according to purposes. For example, as found in power grasp optimization [14], we can use the stability measure not as the objective function but as a constraint. In this case, a different objective function (e.g., sum of finger forces) is optimized, and graspless manipulation with at least a specified stability value can be obtained.

V. Conclusion

We developed a method to determine control modes of multiple robot fingers for graspless manipulation. The method can automatically choose whether each robot finger should be position-controlled or force-controlled. Moreover, desired normal forces can be also determined for fingers to be force-controlled.

The basic strategy of the determination algorithm is to select a combination of finger control modes that achieves the maximum manipulation stability, as far as excessive internal force could not be generated. The algorithm is mainly composed of a series of linear programming problems. The proposed method was applied to some numerical examples of typical graspless manipulation such as sliding and tumbling, where control modes of robot fingers were determined successfully.

We presented a method to determine finger control modes for manipulation but note that it does not automatically mean switching of finger control modes during manipulation. Avoiding frequent switching of control modes is very important but it is another problem.

Now we are trying to incorporate the presented algorithm into our planner of graspless manipulation [15] so that more practical manipulation plans for spatial cases can be generated.

VI. Acknowledgments

This research was partly supported by Grant-in-Aid for Encouragement of Young Scientists (A) / Grant-in-Aid for Young Scientists (B), KAKENHI (13750210), 2001–2002.

VII. REFERENCES

[1] Y. Aiyama, M. Inaba and H. Inoue: "Pivoting: A New Method of Graspless Manipulation of Object by Robot Fingers," Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 136–143, 1993.

[2] M. T. Mason: "Progress in Nonprehensile Manipulation," Int. J. of Robotics Research, Vol.18, No.1, pp. 1129–1141, 1999.

[3] K. Lynch and M. Mason: "Stable Pushing: Mechanics, Controllability, and Planning," Int. J. of Robotics Research, Vol. 15, No. 6, pp. 533–556, 1996.

[4] Y. Maeda and T. Arai: "A Quantitative Stability Measure for Graspless Manipulation," Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 2473–2478, 2002.

[5] S. Hirai and H. Asada: "Kinematics and Statics of Manipulation Using the Theory of Polyhedral Convex Cones," Int. J. of Robotics Research, Vol. 12, No. 5, pp. 434–447, 1993.

[6] M. H. Raibert and J. J. Craig: "Hybrid Position/Force Control of Manipulators," ASME J. of Dynamic Systems, Measurement, and Control, Vol. 102, No. 2, pp. 126–133, 1981.

[7] M. Mason and K. Lynch: "Dynamic Manipulation," Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 152–159, 1993.

[8] Y. Yu and T. Yoshikawa: "Evaluation of Contact Stability between Objects," Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 695–702, 1997.

[9] Y. Yu and T. Yoshikawa: "Evaluation of Contact Stability between Objects," J. of Robotics Soc. of Japan, Vol. 18, No. 7, pp. 1026–1033, 2000 (in Japanese).

[10] T. Omata and K. Nagata: "Rigid Body Analysis of the Interdeterminate Grasp Force in Power Grasps," IEEE Trans. on Robotics and Automation, Vol. 16, No. 1, pp. 46–54, 2000.

[11] T. Omata: "Rigid Body Analysis of Power Grasps: Interdeterminate Grasp Force," Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 2203–2209, 2001.

[12] Y. Maeda, Y. Aiyama, T. Arai and T. Ozawa: "Analysis of Object-Stability and Internal Force in Robotic Contact Tasks," Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 751–756, 1996.

[13] A. Makhorin: GLPK (GNU Linear Programming Kit), http://www.gnu.org/software/glpk/glpk.html.

[14] Y. Yu, K. Takeuchi and T. Yoshikawa: "Optimization of Robot Hand Power Grasps," Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 3341–3347, 1998.

[15] Y. Maeda, H. Kijimoto, Y. Aiyama and T. Arai: "Planning of Graspless Manipulation by Multiple Robot Fingers," Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 2474–2479, 2001.