Proceedings of the 2004 IEEE
International Conference on Robotics & Automation
New Orleans, LA • April 2004

Motion Planning of Robot Fingertips for Graspless Manipulation

Yusuke MAEDA
Dept. of Prec. Eng., School of Eng.,
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo
113-8656 JAPAN
maeda@prince.pe.u-tokyo.ac.jp

Tomohisa NAKAMURA
NTT Data Corporation

Tamio ARAI
Dept. of Prec. Eng., School of Eng.,
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo
113-8656 JAPAN
arai@prince.pe.u-tokyo.ac.jp

***Abstract*— In this paper, we present a method of motion planning of multiple robot fingertips for graspless (or nonprehensile) manipulation. The method can automatically generate various graspless operations, including pushing and tumbling. By considering whether each robot finger should be position- or force-controlled, we can obtain robust manipulation plans against external disturbances. Some examples of planned graspless manipulation of a cuboid by two robot fingers are presented. We also show an experimental result of execution of planned graspless manipulation by a robot with a multi-fingered hand.**

I. INTRODUCTION

Manipulation without grasping is referred to as graspless manipulation [1] or nonprehensile manipulation [2]. In this paper, we study graspless manipulation where the manipulated object is supported not only by robot fingers but also by the environment; it includes pushing, sliding and tumbling (Fig. 1). Graspless manipulation brings the following advantages to robots:

- Manipulation without supporting all the weight of the object
- Manipulation with simple mechanisms
- Manipulation when grasping is impossible

Thus graspless manipulation is important as a complement of conventional pick-and-place to enhance the dexterity of robots.

Planning of robot motions to move an object from an initial configuration to a goal is a fundamental problem in robotic manipulation. However, robot motion planning for graspless manipulation is much more difficult than that for pick-and-place [3]. In pick-and-place operation, once an object is grasped, the correspondence of its motion to the robot motion is trivial; therefore manipulation planning is reduced to a geometrical collision avoidance. On the other hand, the correspondence in graspless manipulation is nontrivial; thus manipulation planning requires mechanical analysis for consideration of the effect of gravity, contact forces, and so on. Moreover, graspless manipulation may be irreversible: For example, a robot can push an object but may not be able to pull it back. Therefore, most of related studies deal with planning of manipulation with a specific operation such as pushing (e.g., [4], [5]).

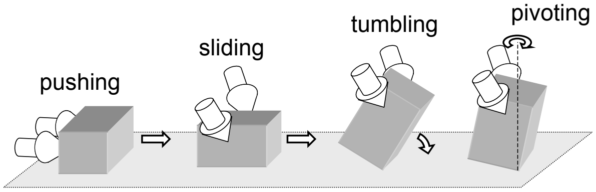


Fig. 1. Graspless Manipulation

The authors proposed a planning method for general graspless manipulation by multiple robot fingertips [3]. However, the method can deal with only planar graspless manipulation. In this paper we extend our previous method. The main improvements are as follows:

- Now it can deal with spatial graspless manipulation such as pushing and tumbling of a polyhedron by multiple robot fingertips.
- Control modes of robot fingers (force control/position control) are considered explicitly for realistic manipulation.
- A new stability measure [6] is adopted in planning to generate robuster graspless manipulation.
- A* algorithm [7] is used to accelerate planning.

This paper is organized as follows: Section II introduces a model of graspless manipulation. Section III describes a method to determine appropriate finger control modes (force control or position control) at an instant in graspless manipulation based on [8]. Section IV proposes a method of motion planning of robot fingertips for graspless manipulation. In Section V, some examples of planned graspless manipulation including pushing and tumbling are presented. We also show an experimental result of execution of planned manipulation by a robot with a multi-fingered hand. Finally, this paper is concluded in Section VI.

II. MODEL OF GRASPLESS MANIPULATION

A. Assumptions

In this paper, for graspless manipulation by multiple robot fingertips, we make the following assumptions:

1) The manipulated object, robot fingertips, and the environment are rigid.

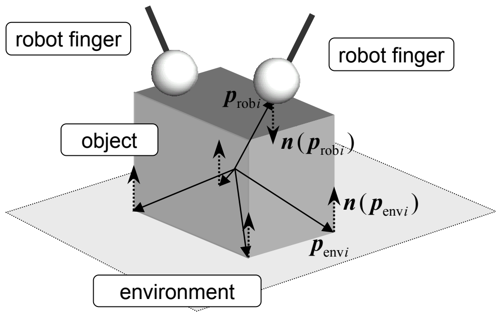


Fig. 2. Object in Graspless Manipulation

2) Manipulation is quasi-static.
3) Coulomb friction exists between the object and the environment (or robot fingertips). The friction coefficient on a contact surface is uniform.
4) Static and kinetic friction coefficients are equal.
5) All the contacts can be approximated by finite point contacts [6].
6) Each of friction cones can be approximated by a polyhedral convex cone [9].
7) Each robot finger is modeled as a rigid sphere and is in one-point non-sliding contact with the object; we consider only fingertips.
8) The normal force of each robot finger has an upper limit.
9) Each robot finger is either in position-control mode or in force-control mode.
10) Each of robot fingers in position-control mode can apply arbitrary force passively within its friction cone.
11) Each of robot fingers in force-control mode is in hybrid position/force control [10]; the finger can apply commanded normal force actively and arbitrary tangential force passively within its friction cone.
12) Sliding and rolling of robot fingers on object surfaces is not allowed. Regrasping is required to change the location of fingertips on the object.

The problem to be solved is to determine a sequence of fingertip positions and finger control modes to move an object from a given initial configuration to a given goal configuration by graspless manipulation. A sequence of desired normal forces is also to be obtained for force-controlled fingers.

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 $\boldsymbol{p}_{\text{env}\,1}, \dots, \boldsymbol{p}_{\text{env}\,m} \in \Re^3$ be positions of contact points between the object and the environment. Similarly, let $\boldsymbol{p}_{\text{rob}\,1}, \dots, \boldsymbol{p}_{\text{rob}\,n} \in \Re^3$ be positions of contact points between the object and the robot finger $1, \dots, n$. 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 $\boldsymbol{\mathcal{C}}_{\text{slide}}$ and $\boldsymbol{\mathcal{C}}_{\text{stat}}$, respectively. We can identify whether $\boldsymbol{p}_{\text{env}\,i} \in \boldsymbol{\mathcal{C}}_{\text{slide}}$ or $\boldsymbol{p}_{\text{env}\,i} \in \boldsymbol{\mathcal{C}}_{\text{stat}}$ when the object motion is specified. We approximate each friction cone at contact point $\boldsymbol{p}$ by a polyhedral convex cone with unit edge vectors, $\boldsymbol{c}_1(\boldsymbol{p}), \dots, \boldsymbol{c}_s(\boldsymbol{p}) \in \Re^3$. For $\boldsymbol{p}_{\text{env}\,i} \in \boldsymbol{\mathcal{C}}_{\text{slide}}$, let $\boldsymbol{c}'(\boldsymbol{p}_{\text{env}\,i}) \in \Re^3$ be a unit edge vector of the friction cone at contact point $\boldsymbol{p}_{\text{env}\,i}$ opposite to its sliding direction.

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

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

where $\operatorname{span}\{\dots\}$ is a polyhedral convex cone spanned by its element vectors [9]. On the other hand, the set of possible finger force $\boldsymbol{f}$ at $\boldsymbol{p}_{\text{rob}\,i}$ is:

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

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

Then we define the following matrices:

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

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 the cross product with $\boldsymbol{p}$.

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

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

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

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

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

III. Determination of Finger Control Modes

A. Overview

In planning of general graspless manipulation, we should consider control modes of robot fingers because appropriate use of both position control and force control according to situation is required to achieve robust manipulation. Thus we incorporate a method of determination of finger control modes at each instant [8] into our graspless manipulation planner. This method is used to test whether manipulation at an instant is feasible or not in our planner.

We use a stability index for graspless manipulation presented in [6] for the control mode determination, although our previous paper [3] adopted another index proposed in [11]. This is because we found that graspless manipulation even with a high value of the index in [11] can be easily perturbed by a small external disturbance; the index in [6] evaluates the robustness of manipulation against external disturbances directly, but the index in [11] does not.

The procedure of the control mode determination is to find the "optimal" combination of finger control modes (and desired normal forces for force-controlled fingers). Thus we search a combination of control modes for all the robot fingers that maximizes the manipulation stability, as far as excessive internal force could not be generated. A brief description of the method is given below. See [8] for the details.

B. Stability Measure for Graspless Manipulation

The performance index of manipulation stability defined in [6] evaluates how much the object can resist external disturbances without changing its motion. The value of the index, z, can be calculated approximately by solving the following linear programming problem:

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

where $\boldsymbol{f}_{\text{com}} := [f_{\text{com}\,1},\dots,f_{\text{com}\,n}]^T \in \Re^n$ 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:

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

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

$\boldsymbol{l}_1,\dots,\boldsymbol{l}_N \in \Re^6$ are position vectors of vertices of a hyperpolyhedron circumscribed to the six-dimensional unit hypersphere, which are used for approximate calculation [6]; $\boldsymbol{R} \in \Re^{6\times 6}$ 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, $\boldsymbol{Q} \in \Re^6$:

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

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

$$\boldsymbol{R} := \text{diag}(\boldsymbol{I}_3,\ M\boldsymbol{J}^{-1}) \in \Re^{6\times 6}, \tag{7}$$

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 Problem (5).

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

$$
\begin{aligned}
\boldsymbol{b}_{\text{env}} &= [1,\dots,1]^T \\
\boldsymbol{b}_{\text{rob}} &= [\boldsymbol{b}_{\text{rob}\,1}^T,\dots,\boldsymbol{b}_{\text{rob}\,n}^T]^T \in \Re^{ns} \\
\boldsymbol{b}_{\text{rob}\,i} &= \begin{cases} [1,\dots,1]^T \in \Re^s \\ \quad \text{if finger } i \text{ is position-controlled,} \\ [0,\dots,0]^T \in \Re^s \\ \quad \text{if finger } i \text{ is force-controlled,} \end{cases} \\
\boldsymbol{A}_{\text{pos}} &= \boldsymbol{I}_{3n} - \boldsymbol{A}_{\text{force}} \ \in \Re^{3n\times 3n}.
\end{aligned}
$$

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.

D. Procedure for Determining Finger Control Modes

The procedure to determine control modes of robot fingers (and desired normal forces for force-controlled fingers) is as follows:

1) Assume a combination of control modes (position control or force control) for each robot finger.
2) Test 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 ($\boldsymbol{f}_{\rm com}$) so that the value of manipulation stability, z, will be maximized (Problem (5)). If the maximized z is larger than the current maximum value, replace it.
4) If all the combinations of control modes have been already tested, stop. Otherwise, go back to step 1.

When all the procedure is completed, we have a combination of control modes that achieves the maximum manipulation stability, if any. 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.

IV. Planning of Graspless Manipulation

A. Configuration Space

Here we define a configuration space (C-Space) that represents the degrees of freedom for both the object and the robot fingertips. We represent positions of the robot fingertips in manipulation as their locations on the surfaces of the object [3]. Thus planning of graspless manipulation is transformed into finding a path in this C-Space.

However, we cannot search the C-Space in the same manner with conventional obstacle avoidance problems because graspless manipulation may be irreversible and regrasping causes discontinuous "jump" in this C-Space. Accordingly, we approximately represent this C-Space by a directed graph referred to as "manipulation-feasibility graph" [3]; we construct nodes of the graph by discretizing the C-Space, and connect the nodes with directed arcs. As a result, planning of graspless manipulation is reduced to searching this graph.

B. Generation of Manipulation-Feasibility Graph

Lattice points in the C-Space are sampled as possible nodes of the manipulation-feasibility graph. We accept each of the sampled points as a valid node if geometrical constraints are satisfied at the configuration (Fig. 3). We connect two nodes with a directed arc if the corresponding manipulation is "feasible"—that is, manipulation stability z is larger than a threshold, $z_{\rm min}$.

Manipulation-feasibility graphs have two kinds of arcs: for displacement of the object and for regrasping [3].

Arcs for displacement of the object correspond to moving the object without changing the locations of the robot fingertips on the surfaces of the object. These arcs connect adjacent nodes in the C-Space for the displacement of the object (Fig. 4(a)). We sample several points on each arc and calculate z at the points. Unless $z \geq z_{\rm min}$ at all the points, the arc is discarded.

Arcs for regrasping correspond to changing a location of one fingertip on the object with neither moving the object nor changing locations of the other fingertips. Note that the arcs for regrasping may be generated between "non-adjacent" nodes, because a location of a fingertip on the object changes discontinuously by regrasping. At each node in the manipulation-feasibility graph, we calculate z when a finger is removed. If $z \geq z_{\rm min}$, the finger can freely change its location on the object. In this case, we can generate bidirectional arcs for corresponding regrasping (Fig. 4(b)).

Thus we have a manipulation-feasibility graph for planning. Note that we do not have to generate all the nodes and the arcs *before* graph searching; tests for generating them can be carried out *in* graph searching.

C. Cost Assignment for Graph Searching

We have to assign a cost, C, to each arc in the manipulation-feasibility graph to find the minimum-cost path from a start node to a goal node by graph searching. In this paper, we decide the assignment of costs considering the following demands in this order of priority:

- Minimize the number of regrasping.
- Minimize the displacement of fingertips.
- Maximize the manipulation stability.

The cost assigned to arcs for displacement of the object is:

$$C = \max_i \sum_{j=1}^{P} \left(1 + \frac{X_{\rm stab}}{z_j}\right) \|\Delta \boldsymbol{q}_{{\rm finger}\, i,j}\|, \tag{9}$$

where $\|\Delta \boldsymbol{q}_{{\rm finger}\, i,j}\|$ is absolute displacement of the i-th fingertip in the j-th segment of the arc that is divided into P segments; z_j is the manipulation stability in a representative point of the j-th segment; $X_{\rm stab}$ is a constant that is defined so that $X_{\rm stab}/z_{\rm min} \ll 1$.

On the other hand, the cost for arcs for regrasping is:

$$C = X_{\rm regr}, \tag{10}$$

where $X_{\rm regr}$ is a constant that is much larger than the value of (9).

D. Heuristic Function for A* Algorithm

We adopt A* algorithm [7] for fast graph searching. A* algorithm with an admissible heuristic function can find the minimum-cost path efficiently. The admissible heuristic function for manipulation planning, H, is designed as follows:

$$H = \begin{cases} \max_i \|\Delta \boldsymbol{q}^*_{{\rm finger}\, i}\| \\ \quad \text{if current fingertip locations are geometrically} \\ \quad \text{feasible even in the goal configuration,} \\ n_{\rm regr} X_{\rm regr} \\ \quad \text{if current fingertip locations are geometrically} \\ \quad \text{infeasible in the goal configuration,} \end{cases} \tag{11}$$

where $\|\Delta \boldsymbol{q}^*_{{\rm finger}\, i}\|$ is estimated displacement of i-th fingertip from the current configuration to the goal configuration without changing fingertip locations on the object; $n_{\rm regr}$ is the number of fingertips whose locations will be geometrically infeasible in the goal configuration; in other words, $n_{\rm regr}$ is the minimum number of fingertips that must change their location on the object by regrasping to achieve the goal configuration.

We assign very large cost to regrasping in our formulation (10). Therefore good estimation of the necessity of regrasping is important for the efficiency of graph searching. In order to accelerate graph searching, the heuristic function (11) utilizes

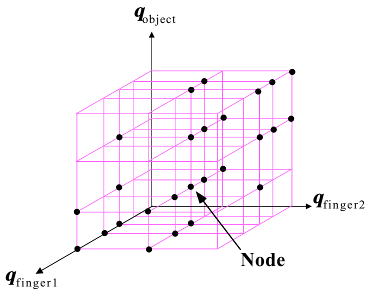
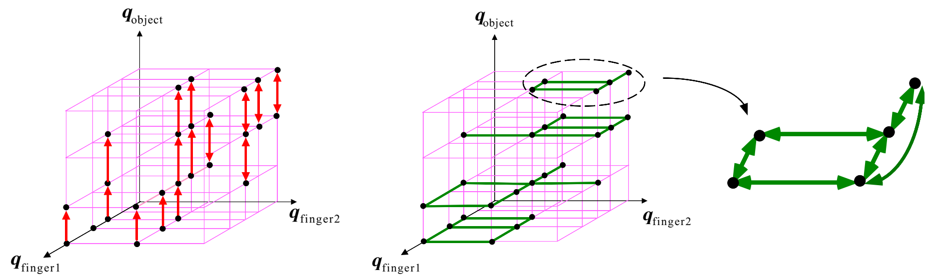


Fig. 3. Generation of Nodes



(a) arcs for displacement of object (b) arcs for regrasping

Fig. 4. Generation of Arcs

the fact that regrasping is necessary when a fingertip will collide with obstacles without changing its location.

V. Planned Results

In this section, we present some examples of planned manipulation by our proposed algorithm. They are graspless manipulation of a cuboid by two robot fingertips. The computation times for the examples below are measured on a Linux PC with Pentium 4 at 2.8 GHz.

Suppose a cuboid whose size is $5 \times 5 \times 10$. The mass of the object is 1 ($M = 1$) and its distribution is uniform; each robot finger is modeled as a sphere whose radius is 1; the gravitational acceleration is 9.8; the friction coefficient between the object and the environment is 0.5, and that between the object and each finger is 0.7; each friction cone is represented as a polyhedral convex cone with 6 unit edge vectors ($s = 6$); $\boldsymbol{f}_{\max} = [10, 10]^T$. Other parameters of the planning algorithm are as follows:

$$z_{\min} = 0.5; X_{\text{regr}} = 10^2; X_{\text{stab}} = 10^{-2}; N = 76; P = 3.$$

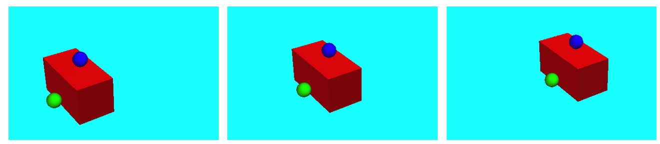
In this section we consider only one degree of freedom for the manipulated object. The degree of freedom is discretized into 31 segments. Fingertip locations are limited to 7×7 grid points on each face of the object. Thus, the maximum number of nodes in the manipulation-feasibility graph is ${}_{7\times7\times6}C_2 \times 31 = 1,335,201$.

A. Planning of Sliding Operations

Let us consider one-dimensional sliding of the cuboid on a horizontal plane. In this case, graspless manipulation as shown in Fig. 5 is generated; one force-controlled finger is located on the top of the object to press it down and another position-controlled finger is located behind the object to push it. It takes 533 CPU minutes for this planning with A*. When we do not use any heuristic functions, 1007 CPU minutes are required.

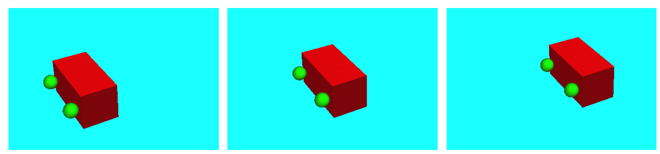
When we use "weaker" robot fingers by setting $\boldsymbol{f}_{\max} = [5, 5]^T$, different graspless manipulation is generated (Fig. 6); both fingers are position-controlled and push the object from behind. This corresponds to "stable push" [5]. The computation requires 203 CPU minutes (or 378 CPU minutes without heuristics).

These results indicates that our algorithm generates graspless manipulation with large internal force like grasping for "strong" robot fingers as in Fig. 5; on the other hand, graspless manipulation without internal force is generated for "weak" robot fingers as in Fig. 6.



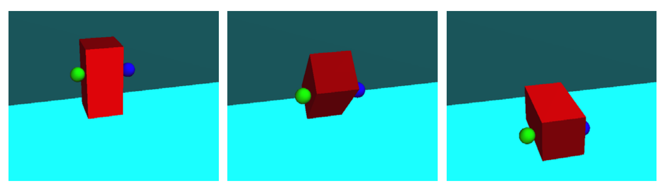
(a) start (b) intermediate (c) goal

Fig. 5. Planned Sliding Operation



(a) start (b) intermediate (c) goal

Fig. 6. Planned Pushing Operation



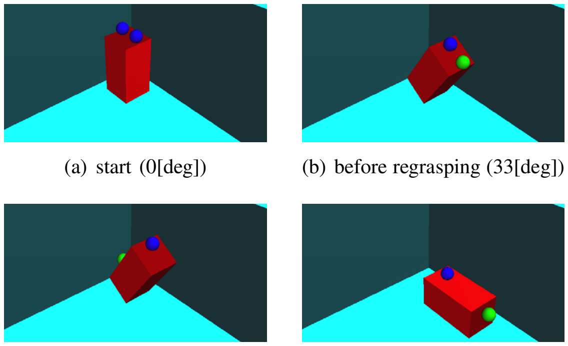
(a) start (0[deg]) (b) inter. (45[deg]) (c) goal (90[deg])

Fig. 7. Planned Tumbling Operation

B. Planning of Tumbling Operations

Let us consider tumbling of the object. When an obstacle exists behind the object, graspless manipulation as shown in Fig. 7 is generated; the robot fingers pinch the object to tumble it down. In this case, one finger is position-controlled and another finger is force-controlled. The computation time is 84 CPU minutes (or 1294 CPU minutes without heuristics).

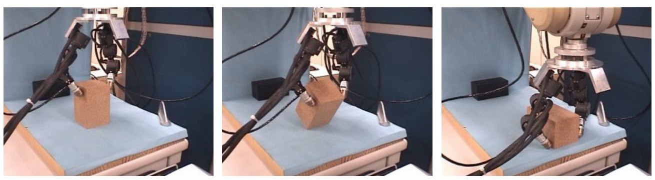
When additional obstacle exist by the side of the object, pinching the object is impossible. In this case, tumbling with regrasping is generated (Fig. 8). The computation time is 990 CPU minutes (or 1296 CPU minutes without heuristics).



(a) start (0[deg]) (b) before regrasping (33[deg])

(c) after regrasping (33[deg]) (d) goal (90[deg])

Fig. 8. Planned Tumbling Operation with Regrasping



(a) start (b) intermediate (c) goal

Fig. 9. Experiment of Tumbling Operation

C. Execution of Planned Manipulation

We conducted experiments with a robot with a multi-fingered hand to validate that our method can generate "decent" manipulation. The experimental system consists of a robot arm with five DOF (degrees of freedom) and a two-fingered hand attached at the end of the arm. Each finger has three DOF and a six-axis force sensor at its fingertip.

We used a cork cuboid of 0.092 [kg] as a manipulated object. Its dimension is 60[mm] $\times$ 60[mm] $\times$ 100[mm]. The friction coefficient between the object and fingertips and that between the object and the environment was identified as 0.15 and 1.2, respectively, from preliminary experiments. We planned graspless manipulation of the object using these values and played back the planned result with the robot.

Here we show a result of tumbling of the object on a plane. In this case, our algorithm generated pinching of the object by the fingers to tumble it like the case of Fig. 7; one finger is force-controlled and another finger is position-controlled throughout the tumbling operation. The planned operation is executed successfully by the experimental system (Fig. 9). Fig. 10 shows finger normal forces in the tumbling operation. The force-controlled finger ("finger1") succeeded in applying normal force roughly as planned.

VI. Conclusion

We developed a method of motion planning of robot fingertips for graspless manipulation. The method can generate a sequence of desired fingertip locations and control modes to achieve robust graspless manipulation from an initial configuration to a goal configuration. Some planned results by this method including pushing and tumbling operations were presented; this indicates our method can plan various graspless

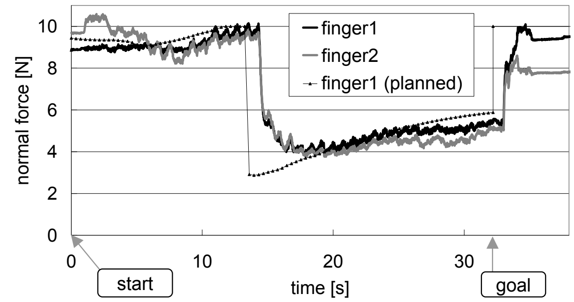


Fig. 10. Finger Forces in Tumbling

operations. An experimental result of successful execution of planned manipulation by a robot with a multi-fingered hand was also shown.

A major problem in our algorithm is that it requires much computation for planning. Currently we are trying to incorporate randomized motion planning techniques into our planner to reduce the computation time.

Acknowledgments

The authors would like to thank Yasumichi AIYAMA and Tsukasa KYOUSOU at University of Tsukuba for their great assistance in doing experiments. 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.

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] 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.

[4] M. Kurisu and T. Yoshikawa: "Trajectory Planning of an Object in Pushing Operation," Proc. of Japan-USA Symp. on Flexible Automation, pp. 1009–1016, 1994.

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

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

[7] J. Pearl: "Heuristics: Intelligent Search Strategies for Computer Problem Solving," Addison-Wesley, 1984.

[8] Y. Maeda and T. Arai: "Automatic Determination of Finger Control Modes for Graspless Manipulation," Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 2660–2665, 2003.

[9] 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.

[10] 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.

[11] H. Kijimoto, T. Arai, Y. Aiyama and T. Yamamoto: "Performance Analysis and Planning of Graspless Manipulation," Proc. of IEEE Int. Symp. on Assembly and Task Planning, pp.238–243, 1999.

[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.