Planning of Graspless Manipulation based on Rapidly-Exploring Random Trees

Kiyokazu MIYAZAWA* Yusuke MAEDA** Tamio ARAI***

*SONY Corporation
**Yokohama National University ***The University of Tokyo

1. Introduction
2. Problem Statement
3. Planning of Graspless Manipulation
4. Planned Results
5. Conclusion
1. Introduction

**Graspless Manipulation**

Manipulation without Grasping [Aiyama 93]

- manipulating
- sliding
- tumbling
- pivoting

Manipulation by Grasping (pick-and-place)
Merits of Graspless Manipulation

- **Manipulation by smaller force**
  
  No need to support all the weight of the object

- **Manipulation by simple mechanisms**
  
  Use of environment and gravity as virtual fingers

- **Manipulation when grasping is impossible**
  
  e.g. Existence of obstacles

---

Graspless manipulation is important as a complement of conventional pick-and-place
Planning of Graspless Manipulation

Problem

Manipulation planning: how to generate robot motion to move an object from initial to goal configuration by graspless manipulation

<table>
<thead>
<tr>
<th></th>
<th>Analysis required for Planning</th>
<th>Reversibility of Manipulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pick-and-Place</td>
<td>Geometry Level (collision avoidance)</td>
<td>Reversible</td>
</tr>
<tr>
<td>Graspless Manipulation</td>
<td>Geometry and Mechanics Level (contact forces and gravity)</td>
<td>Possibly Irreversible</td>
</tr>
</tbody>
</table>

Planning of graspless manipulation is difficult
Our Previous Planner

[Maeda ICRA04] [Maeda Adv. Rob. 05]

- Planning of graspless manipulation by multiple robot fingertips
- Uniform sampling of C-Space, graph representation of feasible manipulation, and A* search
- Local manipulation feasibility is checked by mechanical analysis [Maeda IROS03]
Problem of Our Previous Planner

- Very time-consuming
- Only 1 DOF for manipulated object

533 [CPU min] 203 [CPU min] 84 [CPU min] 990 [CPU min] (on Pentium 4–2.8GHz)
Objective

Accelerate planning of graspless manipulation

- For various graspless operations by multiple fingers
- With more DOF for manipulated object

Approach

- Give up optimality and find a feasible plan quickly
- Rapidly-exploring Random Trees (RRT)
  [LaValle 01]
2. Problem Statement

Assumptions

- Quasi-static manipulation of a rigid object
- Under gravity and Coulomb friction
- Each finger is modeled as a sphere
- Finger forces are upper-bounded
- Slipping and rolling of each finger is not allowed
- Each finger is in position- or force-control mode
Planning Problem

Input:
- Initial and goal configurations of object

Output:
- A series of finger control modes and commands
Configuration of manipulation system

\[ q = (X, F) \]

- \( X = (x, y, \theta) \): Configuration of object
- \( F = (F_1, \ldots, F_N) \): Configuration of robot fingertips

\[ F_i = (x_i, y_i, f_i) \]
Outline of Planning Algorithm

- Based on RRT-GoalBias [LaValle 01]

1. Sample a configuration
2. Find a nearest neighbor
3. Extend search tree
4. Check if goal reached?
   - Yes: END
   - No: Go back to Sample a configuration
Sampling Configurations

Sample a configuration

\[ q_{\text{sample}} = (X_{\text{sample}}, F_{\text{sample}}) \]

- With probability \( \varepsilon \), sample the goal configuration (RRT-GoalBias)
- Otherwise, sample randomly
Finding a Nearest Neighbor

Find the nearest neighbor in the search tree to the sampled configuration:

\[ q_{\text{near}} = (X_{\text{near}}, F_{\text{near}}) \]

Distance function:

\[
d(X_1, X_2) = \sqrt{w_x(x_1 - x_2)^2 + w_y(y_1 - y_2)^2 + w_\theta \left[ \min(|\theta_1 - \theta_2|, 2\pi - |\theta_1 - \theta_2|) \right]^2}
\]

Just consider the difference between \textit{object} configurations
Determine tree extension target

- With probability $\alpha$, change object configuration without changing fingertip locations
  \[ (X_{\text{near}}, F_{\text{near}}) \rightarrow (X_{\text{sample}}, F_{\text{near}}) \]

- Otherwise, change a fingertip location without changing object configuration (Regrasping)
  \[ (X_{\text{near}}, F_{\text{near}}) \rightarrow (X_{\text{near}}, F_{\text{sample}}) \]
Node Connection

- Changing object configuration
  - Collision check
  - Manipulation feasibility check
    [Maeda IROS03]

- Changing fingertip location
  - Collision check
  - Stability check
Fingertip Locations

- Restrict fingertip locations in a set of candidate points
  - Based on Halton sequence
  - When node-connection error rate grows, new candidate points are added
4. Planned Results

Graspless Manipulation of a Cuboid by Two Robot Fingers

Mass of object = 1
Friction coef. between environment and object = 0.5
Friction coef. between fingers and object = 0.7
Maximum finger forces = 10
Acceleration of gravity = 9.8
Plan A: Sliding on a Plane

\[ X = (x, y, \theta) \]

Start: (10, 10, 0) \quad \rightarrow \quad Goal: (50, 50, \pi/2)
Plan A: Result

- Fingertip in force-control mode
- Fingertip in position-control mode

Planning Time: 4.4 [CPU min]
(on Pentium4–2.8GHz)

Planning Time Range [CPU min] | 0.2 ~ 78.8
---|---
Avg. Planning Time [CPU min] | 13.0
Avg. # of Regrasp | 1.5
Avg. # of Sampling | 28.0

(in 100 trials)
Plan A: Other Results

# of Regrasping: 0
Planning Time: 4.1 [CPU min]

# of Regrasping: 1
Planning Time: 18.7 [CPU min]
Plan B: Sliding on a Plane

\[ X = (x, y, \theta) \]

Start: (50, 10, 0) → Goal: (50, 60, 0)
Plan B: Result

- Fingertip in force-control mode
- Fingertip in position-control mode

Planning Time: 29.8 [CPU min] (on Pentium4–2.8GHz)

<table>
<thead>
<tr>
<th>Planning Time Range [CPU min]</th>
<th>0.9 ~ 57.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Planning Time [CPU min]</td>
<td>15.1</td>
</tr>
<tr>
<td>Avg. # of Regrasp</td>
<td>2.3</td>
</tr>
<tr>
<td>Avg. # of Sampling</td>
<td>47.1</td>
</tr>
</tbody>
</table>
Plan B: Other Results

# of Regrasping: 2
Planning Time: 18.7 [CPU min]

# of Regrasping: 3
Planning Time: 36.0 [CPU min]
Plan C: Tumbling and Sliding on a Plane

Start (0, 0) \rightarrow \text{Goal } (20, \pi/2)

\[
X = (x, \theta)
\]
Plan C: Results

- Fingertip in force-control mode
- Fingertip in position-control mode

Planning Time: 3.7 [CPU min] (on Pentium4–2.8GHz)

<table>
<thead>
<tr>
<th>Planning Time Range [CPU min]</th>
<th>0.9 ~ 37.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Planning Time [CPU min]</td>
<td>5.8</td>
</tr>
<tr>
<td>Avg. # of Regrasp</td>
<td>3.1</td>
</tr>
<tr>
<td>Avg. # of Sampling</td>
<td>513</td>
</tr>
</tbody>
</table>

(in 100 trials)
A planner of graspless manipulation is developed.

- Manipulation planning where the object has two or three DOF
- Find a feasible manipulation quickly based on RRT

Future Work

- Incorporation of appropriate heuristics