Dexterous Manipulation Planning with a Multi-fingered Hand

Jean-Philippe Saut

Université Pierre et Marie Curie-Paris 6, FRANCE

presented at the ISR, Universidade de Coimbra
12th of December 2007
Contents

Introduction

Problem Formulation

Configuration Space Structure

Proposed Method

Simulation Results

Conclusion
Dexterous Manipulation

- Robotic hand
- Manipulation with fingertips
- Contact relocation (regrasping)
Control of the Manipulation Movement

- By a human → teleoperation
  
  NAIST hand + CyberGlove
  
  brain-computer interface (Johns Hopkins University)

- By an autonomous system → robotics

4/41
Architecture of a Manipulation System

measure of the environment (stereovision, 3D scanner, etc.)

motion planning

joint position references

grasping force computation

joint torque references

force/position control system

sensors

actuators

robot hand
Architecture of a Manipulation System

- measure of the environment (stereovision, 3D scanner, etc.)
- motion planning
  - joint position references
  - grasping force computation
  - joint torque references
- force/position control system
- sensors
- actuators
- robot hand
Dexterous Manipulation Planning

- Problem: How to manipulate an objet with a robotic hand?
- What movements must follow the fingers? and the object?
- How to adapt the usual motion planning techniques to dexterous manipulation planning?
Dexterous Manipulation Planning

An example of solution:

- start configuration
- goal configuration
Contents

Introduction

Problem Formulation

Configuration Space Structure

Proposed Method

Simulation Results

Conclusion
Problem Formulation

- Studied system and configuration space ($CS$) :
  - the manipulated object $\rightarrow CS_{object} = SE(3)$
  - a robot hand: $n$ fingers mounted on a fixed palm $\rightarrow CS_{hand}$
  - possible motionless obstacles

- System configuration space
  - $CS = CS_{object} \times CS_{hand}$

- Problem: starting from an initial system configuration $q_i$, find a feasible path to reach a final configuration $q_f$. 
Problem Formulation

Constraints:

- **collision**:
  - no collision allowed, except between fingertips and object
  → collision free configuration space $CS_{free}$

- **kinematics**:
  - the object movements are induced by the movements of the fingers

- **stability**:
  - the object grasp must be stable
  → what is a stable grasp?
Problem Formulation

Grasp Stability:

- Assumption: sufficiently slow movements → inertial effects are neglected → quasistatic study

- Two possible criteria:
  - form closure → not adapted to fingertip manipulation
  - force closure
    → we choose the force closure criterion
Elementary Paths

- What are the kinds of path that verify the constraints?
  - The motion of the object is induced by the motion of the fingertips.
  - Contact kinematic constraint: no slipping

- Two kinds of elementary paths:
  - transfer path=
    the object is displaced with a fixed grasp
  - regrasping path=
    the object is immobile while the grasp is modified
Solution of a dexterous manipulation planning problem: a sequence of transfer and regrasping paths.
## Contents

- Introduction
- Problem Formulation
- Configuration Space Structure
- Proposed Method
- Simulation Results
- Conclusion
Configuration Space Structure

- We introduce the $GS_k$ subspaces:

$$GS_k = \left\{ q = (q_{hand}, q_{object}) \in CS_{hand} \times CS_{object} / \text{the object is grasped by } k \text{ fingers} \right\}$$

The only configurations that can be valid are the grasp configurations.

- Dexterous Manipulation Planning Problem: for a $n$-fingered hand, find a path linking two configurations, both in $GS_n$. 

Configuration Space Structure
Configuration Space Topology

$G_{S_k}$s connectivity:

- $G_{S_k} \setminus G_{S_{k+1}}$ is composed of several connected components
  → using paths included in different subspaces will be necessary ↔ finger relocation
Proposed Approach

- All the solutions belong to \( \bigcup_{k \in [2; n]} GS_k \).

- Exploration of all the \( GS_k \)s \( \rightarrow \) very large computation time.

- Favor the search for a solution inside \( GS_n \):
  - favor the use of the most stable grasp configurations
  - \( GS_n \) is the grasp subspace with the smallest dimension (the most constrained)
Contents

Introduction

Problem Formulation

Configuration Space Structure

Proposed Method

Simulation Results

Conclusion
Proposed Method

The proposed method works with two alternating steps:
Proposed Method

The proposed method works with two alternating steps:

exploration of the different connected components of $GS_n$
The proposed method works with two alternating steps:

- **exploration** of the different connected components of $GS_n$
- **fusion of the components** of the graph via **transfer-regrasping paths**
Exploration of $GS_n$

- How to explore $GS_n$?

- $GS_n \equiv$ configurations with several closed kinematic loops.

- Motion planning problem for a system with several closed kinematic loops.

- Which motion planning method can we use?
The Motion Planning Techniques

- Combinatorial motion planning.
  example: cell decomposition, Voronoï diagram, visibility graph

- Feedback motion planning.
  example: artificial potential field method

- Sampling-based motion planning.
The probabilistic Methods

- Probabilistic Roadmap Method (PRM) [Kavraki 1996]

- Rapidly-exploring Random Tree (RRT) [LaValle 1998]
GS_n Exploration

- How can we use the probabilistic methods to explore GS_n?
  - How can we sample GS_n?
  - How can we connect the samples?
**GSₙ Sampling**

- Passive-active chain decomposition (RLG [Cortés 2002]).
- Estimation of the accessibility domains of the fingers: intersection sphere/bounding volume hierarchy ([PQPli])

out of reach    partially reachable
Sample Connection

- Problem: paths in $GS_n$ are not kinematically feasible.

- Usual solution: using elementary paths like transfer-regrasping paths → these paths go outside $GS_n$.

Reduction Property

- Originally:

  ![Original Diagram](image)

  - [Alami 1994]
  - [Sahbani 2002]

- Extension: a collision-free path inside $GS_n$ can be decomposed into a finite sequence of transfer-regrasping paths.

  ![Simulation Diagram](image)

- $\rightarrow$ We can now define paths inside $GS_n$ and use them to search for a solution. They will be decomposed once a solution is found.
Paths inside $G_{S_n}$

- Linear path inside $G_{S_4}$:
Representation Change

- Configuration in $GS_n = \{ \text{object configuration} + \text{coordinates of the contact points} \}$.
- Joint parameters $\leftarrow$ inverse geometric models.
- Linear path inside $GS_n$ : linear change of both the pose parameters of the object and the coordinates of the contact points.
- Advantage : no need for grasp reconfiguration planning.
Proposed Method

Construction of a graph capturing the topology of $GS_n$

- Exploration of the different connected components of $GS_n$
- Fusion of the components of the graph via transfer-regrasping paths
Graph construction

- At each step, choose between:
  - extending the graph with the exploration of $GS_n$ 
  - trying to merge the connected components of the graph
  $\Rightarrow$ it is possible to favor one or the other choice

- Exploration of $GS_n$: visibility-PRM method [Siméon 00] to reduce the node number of the graph.

- Path computation:
  - transfer: a simple linear path for the trajectory of the object
  - regrasping: RRT method for the relocating fingers
Decomposition of the paths in $GS_n$

- The solution path is composed of transfer-regrasping paths and of paths in $GS_n$.
  → the paths in $GS_n$ must be decomposed.
- Decomposition of the paths into a transfer-regrasping path sequence:
Proposed Method

Sphere manipulation example:

- search for a solution

- path decomposition

- optimization
Contents

Introduction

Problem Formulation

Configuration Space Structure

Proposed Method

Simulation Results

Conclusion
Simulated Hand Model

- Geometry:

- Contact model: point contact with friction.
Implementation

- Test configuration: Intel Core2Duo 2.3GHz (one thread only), 2GB RAM.

- Collision detection: PQP C++ library [Gottschalk 1996]

- Comparison exploration of $GS_n$ vs dense sampling:
  - classic method:
    PRM + (sample connection=transfer-regrasping)
Example 2

Box Reorientation:

start configuration
goal configuration
Example 3

A more complicated example:

- reversing a long object (a pencil)

![start configuration](image1)

![goal configuration](image2)
Example 4

Another complicated example:

- changing an electric bulb $\rightarrow$ combination of two DMP problems (insertion and screwing)
Simulation Results

Results averaged on 200 tests

<table>
<thead>
<tr>
<th>example</th>
<th>sphere</th>
<th>box</th>
<th>pencil</th>
<th>bulb</th>
</tr>
</thead>
<tbody>
<tr>
<td>average resolution time (in seconds)</td>
<td>5</td>
<td>57</td>
<td>699</td>
<td>1019</td>
</tr>
<tr>
<td>average generated node number</td>
<td>41</td>
<td>155</td>
<td>1896</td>
<td>836</td>
</tr>
</tbody>
</table>

As a comparison, for the method using only transfer-regrasping paths:

<table>
<thead>
<tr>
<th>example</th>
<th>sphere</th>
<th>box</th>
<th>bulb</th>
<th>pencil</th>
</tr>
</thead>
<tbody>
<tr>
<td>average resolution time (in seconds)</td>
<td>580</td>
<td>763</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>average generated node number</td>
<td>858</td>
<td>1165</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>
Contents

Introduction

Problem Formulation

Configuration Space Structure

Proposed Method

Simulation Results

Conclusion
Conclusion

- We use paths inside $GS_n$ to avoid computation of regrasping movements when possible.
- We build a graph to capture $GS_n$ topology.
- We merge the graph different components via appropriate elementary paths.

Outlook
- Manipulation with two hands.
- Trying to introduce new contact models (deformable contacts).