

第5章

グラスプレス・マニピュレーション におけるロボット指の制御モードの 決定

5.1	はじめに	78
5.2	従来研究	79
5.3	ロボット指によるグラスプレス・マニピュレーションのモデル	80
5.3.1	モデル化上の仮定	80
5.3.2	力学モデル	80
5.4	指の制御モードの自動決定	83
5.4.1	基本方針	83
5.4.2	操作の確実性	83
5.4.3	過大な内力の発生可能性	85
5.4.4	指の制御モードの決定手順	85
5.5	数値例	88
5.5.1	例1：平面上での直方体の滑らせ操作	88
5.5.2	例2：直方体の転がし操作	90
5.5.3	例3：平面上での四角錐の滑らせ操作	91
5.5.4	パラメータの変動の効果	92
5.6	制御モード決定手法に対する考察	96
5.6.1	提案手法の性質	96
5.6.2	計算量	96
5.7	おわりに	98

5.1 はじめに

前章までに、グラスプレス・マニピュレーションの問題を扱う上で重要となる、操作の確実性および過大な内力の問題について力学解析を行った。本章以降では、その結果を利用して、複数のロボット指を用いたグラスプレス・マニピュレーションの問題を取り扱う。

ロボットによる環境との接触作業においては、過大な内力の発生を避けるために通常力制御が用いられる。しかし、グラスプレス・マニピュレーションにおいては、位置制御を用いた方が望ましい場合も存在する。平面上での、位置制御ロボットでの線接触による押し操作 (“stable push” [Lynch 1996b]) はその典型例である。ここから分かることは、さまざまなグラスプレス・マニピュレーションを実行するためには、状況に応じて位置制御と力制御を適切に使い分ける必要がある、ということである。

そこで、本章では、ロボット指の制御モード（位置制御もしくは力制御モード）を決定する（力制御の場合はその目標力も含めて決定する）問題について考察する。

5.2 従来研究

本章で扱う、グラスプレス・マニピュレーションのためのロボット指の制御モードを決定する問題については、関連研究も少ない。

沢崎らは、複数の移動ロボットによる協調物体操作を対象に、各ロボットの先端において制御をどのように行うべきかを決定する手法を提案している [沢崎 1993]。ここでは、各ロボットの先端は、位置と力のハイブリッド制御されており、具体的には、位置制御方向の力がゼロとなるように位置制御方向を回転させるようなフィードバック制御を行う。これにより、局所的な情報のみから各ロボットが自律的に動作を行うことを実現している。このことは、相互に密な通信や正確な位置決めが期待できない移動ロボットにとっては大きな利点である。しかし、多指ハンドのようなシステムでこのような制御を行うことは、操作の確実性が損われる点や、stable push のような位置制御のみによる簡便なマニピュレーションの実現が不可能になることから、適用範囲は限定される。

余らは、接触を維持可能な押し付け力の存在範囲の大きさを評価する指標を利用して、接触作業における最適な指先位置を決定する方法を示している [余 2000]。しかし、この指標には重力の効果を含めることができない（むしろ、重力を外乱と考えている）ため、重力を積極的に利用して物体操作を行うグラスプレス・マニピュレーションへの適用は適切でない。また、指力の方向のみから計算される指標であるため、指力の大きさをどうすべきかは分からない。指の制御モードについても考慮されていない。

また、渡辺らは、環境との接触作業を対象に、指定された外力の集合に対して、その中のいずれの外力にも対抗できるような指先位置と姿勢を決定する問題を扱っている [渡辺 2002]。ここではロボットへの負担を小さくすることを優先しており、最悪の場合での関節トルクが最小となるように、指先位置と姿勢を分枝限定法によって解いて決定している。ただし、指の制御モードについては考慮していないため、したがって外乱が加わった場合には、瞬時的に関節トルクを変化させてそれを打ち消すことができることが前提となっている。

5.3 ロボット指によるグラスプレス・マニピュレーションのモデル

5.3.1 モデル化上の仮定

本章では、以下の仮定をおく。

- 対象物、ロボットの指先、環境は剛体である。
- マニピュレーションは準静的に行われる。
- クーロン摩擦が、対象物と環境の間、および対象物とロボット指の間に存在する。
- 静止摩擦係数と動摩擦係数は等しい。
- 摩擦円錐は凸多面錐 [平井 1999] によって近似できる。
- ロボット指は対象物と摩擦あり点接触する。
- すべての接触は複数の点接触で近似できる (2.3 節参照)。
- 不完全な接触[張 1996]は存在しない。
- ロボットの指力の接触法線成分には上限がある。
- ロボットの各指は位置制御モードもしくは力制御モードのどちらかに設定される。
- 位置制御モードの場合、ロボットの指は、その摩擦円錐内の任意の力を (受動的に) 発生することができる。
- 力制御モードの場合、ロボットの指は位置・力のハイブリッド制御[Raibert 1981] される。ここでは、ロボット指は接触法線方向には指令された力を能動的に発生し、接線方向には摩擦円錐内の任意の力を受動的に発生できるとする。

下線部は 2.3 節における仮定に加えて新たに導入したものである。記号等は 2.3 節で定義されたものを用いる。

ここで扱うのは、実現したい対象物の運動と、そのときの指先位置が与えられたときに、そのためには各指を位置制御すべきか、力制御すべきかを定める問題である。さらに、力制御の場合は指力の指令値まで決定することとする。

5.3.2 力学モデル

Fig. 5.1 のようなグラスプレス・マニピュレーションを考える。対象物の重心を原点とする座標系をとり、 $\mathbf{p}_{\text{env } 1}, \dots, \mathbf{p}_{\text{env } m} \in \mathcal{R}^3$ を対象物と環境との接触点位置とする。同様に、

$\mathbf{p}_{\text{rob } 1}, \dots, \mathbf{p}_{\text{rob } n} \in \mathcal{R}^3$ を, $1, \dots, n$ 番目のロボット指と対象物との間の接触点位置とする. 接触点 \mathbf{p} での対象物側への単位接触法線ベクトルを $\mathbf{n}(\mathbf{p}) \in \mathcal{R}^3$ で表す.

実現したい対象物の運動が与えられれば, そのとき $\mathbf{p}_{\text{env } i} \in \mathcal{C}_{\text{slide}}$ か $\mathbf{p}_{\text{env } i} \in \mathcal{C}_{\text{stat}}$ かは判定することができる. 接触点 \mathbf{p} での摩擦円錐を, 単位稜ベクトル $\mathbf{c}_1(\mathbf{p}), \dots, \mathbf{c}_s(\mathbf{p}) \in \mathcal{R}^3$ を持つ凸多面錐で表す. $\mathbf{p}_{\text{env } i} \in \mathcal{C}_{\text{slide}}$ の場合は, $\mathbf{c}'(\mathbf{p}_{\text{env } i}) \in \mathcal{R}^3$ を, $\mathbf{p}_{\text{env } i}$ における滑りの方向と反対側の摩擦円錐のエッジとする.

接触点 $\mathbf{p}_{\text{env } i}$ において可能な接触力 $\mathbf{f} \in \mathcal{R}^3$ の集合は以下のように表せる.

$$\begin{cases} \{\mathbf{f} | \mathbf{f} \in \text{span}\{\mathbf{c}_1(\mathbf{p}_{\text{env } i}), \dots, \mathbf{c}_s(\mathbf{p}_{\text{env } i})\}\} \subset \mathcal{R}^3 & (\mathbf{p}_{\text{env } i} \in \mathcal{C}_{\text{stat}} \text{ のとき}) \\ \{\mathbf{f} | \mathbf{f} \in \text{span}\{\mathbf{c}'(\mathbf{p}_{\text{env } i})\}\} \subset \mathcal{R}^3 & (\mathbf{p}_{\text{env } i} \in \mathcal{C}_{\text{slide}} \text{ のとき}) \end{cases} \quad (5.1)$$

ここで $\text{span}\{\dots\}$ は, その要素ベクトルで張られる凸多面錐である [平井 1999].

一方, $\mathbf{p}_{\text{rob } i}$ における可能な接触力 \mathbf{f} の集合は,

$$\begin{cases} \{\mathbf{f} | \mathbf{f} \in \text{span}\{\mathbf{c}_1(\mathbf{p}_{\text{rob } i}), \dots, \mathbf{c}_s(\mathbf{p}_{\text{rob } i})\}, \mathbf{n}(\mathbf{p}_{\text{rob } i})^T \mathbf{f} \leq f_{\text{max } i}\} \subset \mathcal{R}^3 & (i \text{ 番目のロボット指が位置制御のとき}) \\ \{\mathbf{f} | \mathbf{f} \in \text{span}\{\mathbf{c}_1(\mathbf{p}_{\text{rob } i}), \dots, \mathbf{c}_s(\mathbf{p}_{\text{rob } i})\}, \mathbf{n}(\mathbf{p}_{\text{rob } i})^T \mathbf{f} = f_{\text{com } i} \leq f_{\text{max } i}\} \subset \mathcal{R}^3 & (i \text{ 番目の指が力制御のとき}) \end{cases} \quad (5.2)$$

ここで $f_{\text{max } i}$ は i 番目のロボット指の法線方向の接触力の上限であり, $f_{\text{com } i}$ は i 番目のロボット指の (法線方向の) 指令指力である.

また, 以下の行列を定義する.

$$\mathbf{W}_{\text{env}} = \begin{bmatrix} \mathbf{I}_3 & \dots & \mathbf{I}_3 \\ \mathbf{p}_{\text{env } 1} \times \mathbf{I}_3 & \dots & \mathbf{p}_{\text{env } m} \times \mathbf{I}_3 \end{bmatrix} \in \mathcal{R}^{6 \times 3m}$$

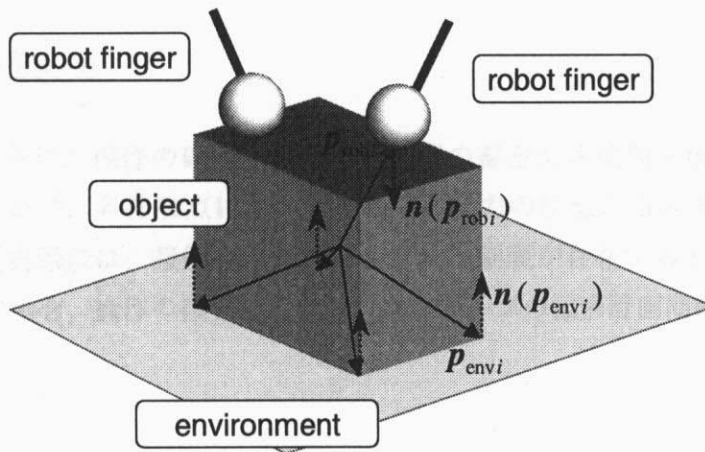


Fig. 5.1: Object in Grasplless Manipulation

$$\begin{aligned}
\mathbf{C}_{\text{env}} &= \text{diag}(\mathbf{C}_{\text{env } 1}, \dots, \mathbf{C}_{\text{env } m}) \\
\mathbf{C}_{\text{env } i} &= \begin{cases} [\mathbf{c}_1(\mathbf{p}_{\text{env } i}) \cdots \mathbf{c}_s(\mathbf{p}_{\text{env } i})] \in \mathbb{R}^{3 \times s} & (\mathbf{p}_{\text{env } i} \in \mathbf{C}_{\text{stat}} \text{ のとき}) \\ [\mathbf{c}'(\mathbf{p}_{\text{env } i})] \in \mathbb{R}^{3 \times 1} & (\mathbf{p}_{\text{env } i} \in \mathbf{C}_{\text{slide}} \text{ のとき}) \end{cases} \\
\mathbf{W}_{\text{rob}} &= \begin{bmatrix} \mathbf{I}_3 & \cdots & \mathbf{I}_3 \\ \mathbf{p}_{\text{rob } 1} \times \mathbf{I}_3 & \cdots & \mathbf{p}_{\text{rob } n} \times \mathbf{I}_3 \end{bmatrix} \in \mathbb{R}^{6 \times 3n} \\
\mathbf{C}_{\text{rob}} &= \text{diag}(\mathbf{C}_{\text{rob } 1}, \dots, \mathbf{C}_{\text{rob } n}) \in \mathbb{R}^{3n \times ns} \\
\mathbf{C}_{\text{rob } i} &= [\mathbf{c}_1(\mathbf{p}_{\text{rob } i}) \cdots \mathbf{c}_s(\mathbf{p}_{\text{rob } i})] \in \mathbb{R}^{3 \times s} \\
\mathbf{N}_{\text{rob}} &= \text{diag}(n(\mathbf{p}_{\text{rob } 1}), \dots, n(\mathbf{p}_{\text{rob } n})) \in \mathbb{R}^{3n \times n},
\end{aligned}$$

ここで \mathbf{I}_3 は 3×3 の単位行列, $\mathbf{p} \times \mathbf{I}_3 \in \mathbb{R}^{3 \times 3}$ は \mathbf{p} との外積と等価な歪対称行列である.

外乱力がなければ, 対象物のつりあい方程式は

$$\mathbf{Q}_{\text{known}} + \mathbf{W}_{\text{env}} \mathbf{C}_{\text{env}} \mathbf{k}_{\text{env}} + \mathbf{W}_{\text{rob}} \mathbf{C}_{\text{rob}} \mathbf{k}_{\text{rob}} = \mathbf{0} \quad (5.3)$$

となる. ここで, $\mathbf{k}_{\text{env}} \geq \mathbf{0}$, $\mathbf{k}_{\text{rob}} \geq \mathbf{0}$ であり, $\mathbf{Q}_{\text{known}} \in \mathbb{R}^6$ は重力などの既知の一般化外力である. ロボット指の法線方向力の制限は, 以下のように書ける.

$$\mathbf{N}_{\text{rob}}^T \mathbf{C}_{\text{rob}} \mathbf{k}_{\text{rob}} \leq \mathbf{f}_{\text{max}} \quad (5.4)$$

ここで $\mathbf{f}_{\text{max}} = [f_{\text{max } 1}, \dots, f_{\text{max } n}]^T \in \mathbb{R}^n$ である.

5.4 指の制御モードの自動決定

5.4.1 基本方針

ロボット指の制御モードを「適切」に決めるためには、なんらかの指針が必要である。適切な評価関数が与えられれば、それに基づいて、「最適」な制御モードの組み合わせ（および、そのときの指力）を決めることができる。

本論文では、グラスプレス・マニピュレーションに適する指標として「操作の確実性」を3章で導入した。したがって、操作の確実性の指標を利用することによって、グラスプレス・マニピュレーションにおける最適な指の制御モードおよび指令指力を決定することができると考えられる。グラスプレス・マニピュレーションは、対象物を把持していないために、通常のピックアンドプレイス操作に比べて外乱に弱いことを考えると、操作の確実性ができるだけ高くなるようにすることは有効であると考えられる。

一般に、操作の確実性を高くする、という観点からは、力制御よりも位置制御のほうが望ましい。なぜなら、力制御された指に比べて、位置制御された指のほうが広い範囲の接触力を（受動的にはあるが）発生することができ、したがって力のつりあいが破られにくくなるからである。一方で、当然ながら不用意に位置制御を利用すると、対象物に過大な内力がかかる恐れがある (Fig. 4.1) のは4章で見たとおりである。

したがって、ここでは、「過大な内力が発生する恐れのない範囲で、操作の確実性の指標を最大化する」という方針で、各指の接触モードの組み合わせおよび（力制御指の）指力の指令値を決定することにする。

5.4.2 操作の確実性

3章で述べたように、操作の確実性の値は、一連の線形計画問題を解くことによって近似的に求まる。ここで、各接触点において可能な接触力の任意の組み合わせが可能であると仮定してみる（実際には、接触力の組み合わせには制限が存在する [小俣 1995, 余 2000, 小俣 2001]）。その場合、操作の確実性の値は、以下の一つの線形計画問題を解くことによっ

て求めることができる。

$$\begin{aligned}
 & \text{maximize } z \\
 & \text{subject to } \left\{ \begin{array}{l}
 z\mathbf{R}^{-1/2}\mathbf{l}_1 = \mathbf{Q}_{\text{known}} + \mathbf{W}_{\text{env}}\mathbf{C}_{\text{env}}\mathbf{k}_{\text{env } 1} + \mathbf{W}_{\text{rob}}\mathbf{C}_{\text{rob}}\mathbf{k}_{\text{rob } 1} \\
 \vdots \\
 z\mathbf{R}^{-1/2}\mathbf{l}_{N_{\text{vert}}} = \mathbf{Q}_{\text{known}} + \mathbf{W}_{\text{env}}\mathbf{C}_{\text{env}}\mathbf{k}_{\text{env } N_{\text{vert}}} + \mathbf{W}_{\text{rob}}\mathbf{C}_{\text{rob}}\mathbf{k}_{\text{rob } N_{\text{vert}}} \\
 \mathbf{N}_{\text{rob}}^T\mathbf{C}_{\text{rob}}\mathbf{k}_{\text{rob } 1} \leq \mathbf{f}_{\text{max}} \\
 \vdots \\
 \mathbf{N}_{\text{rob}}^T\mathbf{C}_{\text{rob}}\mathbf{k}_{\text{rob } N_{\text{vert}}} \leq \mathbf{f}_{\text{max}} \\
 \mathbf{N}_{\text{rob}}^T\mathbf{A}_{\text{force}}\mathbf{C}_{\text{rob}}\mathbf{k}_{\text{rob } 1} = \mathbf{f}_{\text{com}} \\
 \vdots \\
 \mathbf{N}_{\text{rob}}^T\mathbf{A}_{\text{force}}\mathbf{C}_{\text{rob}}\mathbf{k}_{\text{rob } N_{\text{vert}}} = \mathbf{f}_{\text{com}} \\
 \mathbf{k}_{\text{env } 1}, \dots, \mathbf{k}_{\text{env } N_{\text{vert}}} \geq \mathbf{0} \\
 \mathbf{k}_{\text{rob } 1}, \dots, \mathbf{k}_{\text{rob } N_{\text{vert}}} \geq \mathbf{0}
 \end{array} \right. \quad (5.5)
 \end{aligned}$$

ここで、 $\mathbf{f}_{\text{com}} = [f_{\text{com } 1}, \dots, f_{\text{com } n}]^T \in \mathcal{R}^n$ であり、 i 番目の指が位置制御されている場合は $f_{\text{com } i} = 0$ とする。 $\mathbf{A}_{\text{force}}$ は以下のような選択行列である。

$$\begin{aligned}
 \mathbf{A}_{\text{force}} &= \text{diag}(\text{diag}(a_1, a_1, a_1), \dots, \text{diag}(a_n, a_n, a_n)) \in \mathcal{R}^{3n \times 3n} \\
 a_i &= \begin{cases} 1 & (i \text{ 番目の指が力制御のとき}) \\ 0 & (i \text{ 番目の指が位置制御のとき}) \end{cases}
 \end{aligned}$$

$\mathbf{l}_1, \dots, \mathbf{l}_{N_{\text{vert}}} \in \mathcal{R}^6$ は 6 次元単位超球に外接する近似超多面体の頂点であり、 $\mathbf{R}^{-1/2}$ は適当な正定対称行列 $\mathbf{R} \in \mathcal{R}^{6 \times 6}$ の逆行列の Cholesky 分解である (式 (3.1) 参照)。

\mathbf{f}_{com} を定数ではなく変数と見なせば、線形計画問題 (5.5) を解くことで、操作の確実性を最大にする \mathbf{f}_{com} を求めることができる。つまり、各指の制御モードを適当に仮定すれば、その制御モードのパターンにおいて最適な \mathbf{f}_{com} が求まる。

ここで導入した「各接触点において可能な接触力の任意の組み合わせが可能である」という仮定は楽観的なもので、実際にはこれは成り立たない。このことは、2 章、3 章でも触れた。しかしながら、摩擦が比較的小さい場合は、この単純化は操作の確実性の値にあまり大きな影響を与えない。なぜなら、ある接触力の組み合わせはありえなくとも、それと同じ合力を、ありうる接触力の組み合わせで発生することが可能であることが多いためである。つまり、より正確には、「各接触点において可能な接触力の任意の組み合わせによる合力がすべて発生可能である」という条件さえ成り立てば、線形計画問題 (5.5) を利用することができる。そこで、本章では、この条件が成り立つとし、線形計画問題 (5.5) を使ってグラスプレス・マニピュレーションの操作の確実性を評価することにする。

もちろん、特に摩擦が大きい場合には、ありうる接触力の組み合わせについて、より保守的な評価が必要となる。これには [小俣 1995, 余 2000, 小俣 2001] などの結果が利用できると考えられるが、今後の課題である。ただし、計算量がかなり大きくなることは避けられないであろう。

5.4.3 過大な内力の発生可能性

4章で述べたように、過大な内力の発生可能性は、以下のような線形計画問題を解くことによって判定できる。

$$\begin{aligned} & \text{maximize } \mathbf{b}_{\text{env}}^T \mathbf{k}_{\text{env}} + \mathbf{b}_{\text{rob}}^T \mathbf{k}_{\text{rob}} \\ & \text{subject to } \begin{cases} \mathbf{W}_{\text{env}} \mathbf{C}_{\text{env}} \mathbf{k}_{\text{env}} + \mathbf{W}_{\text{rob}} \mathbf{A}_{\text{pos}} \mathbf{C}_{\text{rob}} \mathbf{k}_{\text{rob}} = \mathbf{0} \\ \mathbf{k}_{\text{env}} \geq \mathbf{0} \\ \mathbf{k}_{\text{rob}} \geq \mathbf{0} \end{cases} \end{aligned} \quad (5.6)$$

ここで、

$$\begin{aligned} \mathbf{b}_{\text{env}} &= [1, \dots, 1]^T \\ \mathbf{b}_{\text{rob}} &= [\mathbf{b}_{\text{rob } 1}^T, \dots, \mathbf{b}_{\text{rob } n}^T]^T \in \mathcal{R}^{ns} \\ \mathbf{b}_{\text{rob } i} &= \begin{cases} [1, \dots, 1]^T \in \mathcal{R}^s & (i \text{ 番目の指が位置制御のとき}) \\ [0, \dots, 0]^T \in \mathcal{R}^s & (i \text{ 番目の指が力制御のとき}) \end{cases} \\ \mathbf{A}_{\text{pos}} &= \mathbf{I}_{3n} - \mathbf{A}_{\text{force}} \in \mathcal{R}^{3n \times 3n} \end{aligned}$$

である。この線形計画問題が有界であれば、内力も有界になり、過大な内力は発生しない。そうでなければ、過大な内力の発生する恐れがある。

4章で述べたように、この判定法はあくまで必要条件を調べているだけで、過大な内力の発生の十分条件ではない。より必要十分条件に近い判定法も考えられるが、計算量が大きくなることを避けるため、ここでは線形計画問題 (5.6) を用いることにする。必要条件を用いて判定するということは、結果として安全第一の方針（過大な内力の発生可能性が否定できない場合はそれを採用しない）ということになる。

5.4.4 指の制御モードの決定手順

以上の準備の下で、ロボット指の「最適」な制御モードの組み合わせ、および指令指力を、以下のような手順で決定することができる。

1. 各指に制御モード（位置制御／力制御）を適当に割り当てる。

2. その状態で過大な内力の発生可能性がないかチェックする（線形計画問題 (5.6)）. 過大な内力が発生する可能性がある場合はあきらめて4へ.
3. 操作の確実性が最大になるように指力 (f_{com}) を最適化する（線形計画問題 (5.5)）. 他の制御モードのパターンよりも操作の確実性が大きくなれば, これを記憶しておく.
4. まだ調べていない制御モードの組み合わせがある場合は1へ戻る. すべての組み合わせを調べ終わった場合はこれで終了.

すべての組み合わせを調べ終わったら, その中で操作の確実性の値を最大とする制御モードの組み合わせを採用する. もし, 操作の確実性が正になる組み合わせが一つも存在しないならば, このマニピュレーションを確実に実現することは不可能（無限小の外乱力によっても対象物の運動が乱されてしまう）ということになる.

ただし, 組み合わせの数 (n 本の指に対して 2^n 通り) だけの繰り返しがあるため, 上記の素朴な方針では少々計算量が大きい. しかし, 以下の性質を考えることで, 計算量を減らすことができる.

1. ある制御モードのパターンで, 過大な内力の発生可能性があるならば, そのパターンから位置制御の指を増やしたパターンでは必ず過大な内力の発生可能性がある.
2. ある制御モードのパターンにおける操作の確実性の値を, そのパターンから力制御の指を増やしたパターンでの操作の確実性の値が上回ることはない.

上記の2つの性質を利用すれば, 調べる必要のないパターンを省くことができる. 方針として,

- すべての指が位置制御モードのパターンから始め, 順に力制御の指を増やしていく.
- 操作の確実性の値が増加する可能性が明らかでないパターンは調べない. 例えば, すべての指が位置制御モードの場合に, 過大な内力の発生可能性がなく, ある操作の確実性の値が得られたならば, その他のパターンは一切調べない (調べる必要がない).

とする方法と,

- すべての指が力制御モードのパターンから始め, 順に位置制御の指を増やしていく.
- 過大な内力の発生可能性が明らかにあるパターンは調べない. 例えば, すべての指が力制御モードの場合に, 過大な内力の発生可能性があったならば, その他のパターンは一切調べない (調べる必要がない).

とする方法がある。ここでは前者を採用した。これは、問題 (5.5) が問題 (5.6) よりずっと複雑で時間がかかり、特に力制御モードと仮定した指が多い場合にそれが顕著になる（指力まで決定しないとイケないため）ので、前者の方針の方が早く最適な制御モードのパターンを見つけられることが多いためである。

5.5 数値例

前節で述べた手続きを、C言語を用いて実装した。実装にあたって、線形計画問題のソルバとしてフリーのGLPK (GNU Linear Programming Kit) [Makhorin] を用いた。GLPKは線形計画問題を解くのに、シンプレックス法と内点法（障壁法）を利用できるが、ここではシンプレックス法を用いている。

以降では、開発したプログラムを用いて、指の制御モードの決定問題の数値例を示す。

本節では、2本のロボット指によるグラスプレス・マニピュレーションを考える。対象物は直方体であり、その質量分布は均一であるとする。また、重力加速度を9.8、対象物と環境との間の摩擦係数を0.2、対象物とロボット指との間の摩擦係数を0.5とした。各接触点での摩擦円錐は、凸6面錐で近似 ($s = 6$) し、各指の力の上限は $\mathbf{f}_{\max} = [10, 10]^T$ とする。

また、操作の確実性の値の計算において、6次元単位超球を、それに外接する超多面体で近似する。ここでも、この超多面体の頂点としては、以下の76点をとった ($N_{\text{vert}} = 76$)。

$$l_i = \begin{cases} k[\pm 1, 0, 0, 0, 0, 0]^T \\ k[0, \pm 1, 0, 0, 0, 0]^T \\ k[0, 0, \pm 1, 0, 0, 0]^T \\ k[0, 0, 0, \pm 1, 0, 0]^T \\ k[0, 0, 0, 0, \pm 1, 0]^T \\ k[0, 0, 0, 0, 0, \pm 1]^T \\ k\left[\pm \frac{1}{\sqrt{6}}, \pm \frac{1}{\sqrt{6}}, \pm \frac{1}{\sqrt{6}}, \pm \frac{1}{\sqrt{6}}, \pm \frac{1}{\sqrt{6}}, \pm \frac{1}{\sqrt{6}}\right]^T, \end{cases} \quad (5.7)$$

ここで $k = 2\sqrt{3 - \sqrt{6}} \approx 1.48$ である (付録A.2参照)。式(3.1)の \mathbf{R} については、対象物の質量 M_o と慣性テンソル \mathbf{J}_o を使う、式(3.2)を適用した。

5.5.1 例1：平面上での直方体の滑らせ操作

平面上での $2 \times 2 \times 1$ の直方体の滑らせ操作を考える (Fig. 5.2)。対象物の質量を1とする ($M_o = 1$)。座標系は図に示されているように、対象物の質量中心が原点になるようにとる。この場合、 $\mathbf{J}_o = \text{diag}(5/12, 5/12, 2/3)$ および $\mathbf{Q}_{\text{known}} = [0, 0, -9.8, 0, 0, 0]^T$ である。

2本のロボット指でこの直方体を $[-1, 0, 0]^T$ の方向に動かすとしよう。Fig. 5.2 左のように、 $[1, \pm 1/2, 0]^T$ の位置で物体を押す場合、本章のアルゴリズムでは、両方の指を位置制御するのが最適となり、操作の確実性の値は0.6となる。これはLynchらの“stable push” [Lynch 1996b] に対応する。

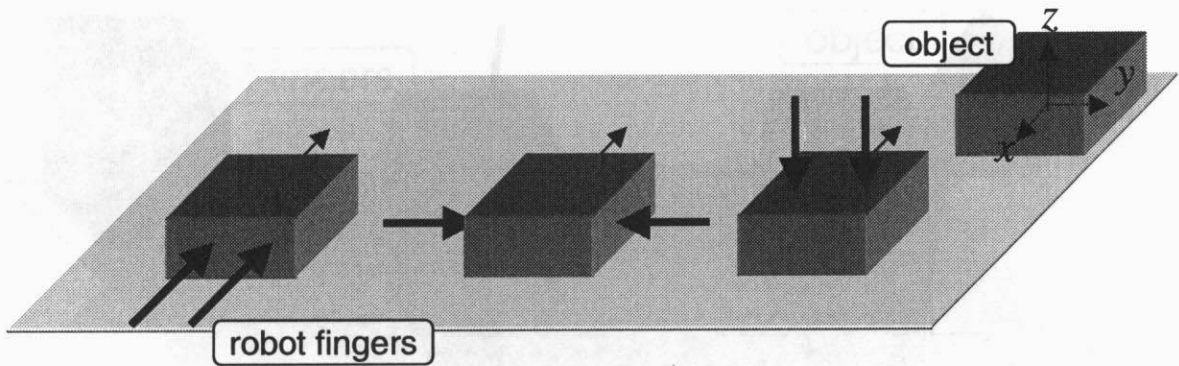


Fig. 5.2: Sliding a Cuboid

直方体を $[0, \pm 1, 0]^T$ の位置ではさんで滑らせる場合 (Fig. 5.2 中央), 一本の指を位置制御, もう片方の指を力制御するのが最適となる. この場合是对称なので, 位置制御/力制御を入れ替えても構わない. 力制御指の指令指力は $6.5 (< f_{\max i})$ となり, 操作确实性の値は 2.4 である. この場合, 力制御指は「ほどほど」の力で対象物を押すと操作の确实性を最大にできる, ということになる. なぜなら, 目一杯の力で押ししてしまうと, それと同じ方向に外乱力が加わった際に, 位置制御指の方が耐えられなくなってしまうからである.

Fig. 5.2 右のように, 直方体の上面の $[0, \pm 1/2, 1/2]^T$ の位置で押しながら引きずる場合, 両方の指を力制御するのが最適となる. 指令指力はともに $10 (= f_{\max i})$ となる. つまり, この場合は両方の指が目一杯の力で押すのが最適ということになる. この場合の操作の确实性の値は 1.7 である.

上記の数値例の場合, Pentium4-1.6GHz を搭載した Linux PC において, 0.02~0.7 CPU 秒を要した. もっとも時間のかかるのは, (5.5) の線形計画問題を解く部分である.

なお, 上の結果は $f_{\max} = [10, 10]^T$ とした場合のものであったが, 今度は $f_{\max} = [2.5, 2.5]^T$ だった場合の指の制御モードの決定を考える. このとき, Fig. 5.2 左の場合は同じく 2 本の指をともに位置制御とするのが最適となり, 操作の确实性は 0.6 と変わらない. Fig. 5.2 中央の場合は, 1 本の指を位置制御, もう片方の指を力制御する, という結果は変わらないが, 指力の制限により, 力制御指の指令指力は 2.3, そして操作の确实性は 0.2 と低下する. 一方 Fig. 5.2 右の場合は, 操作が不可能という結果が得られる. このように, 本章で提案する指の制御モードの決定法は, 指の能力 (この場合は発生可能な指力の上限值) によって, 適切な制御モード (もしあれば) を決定できることがわかる.

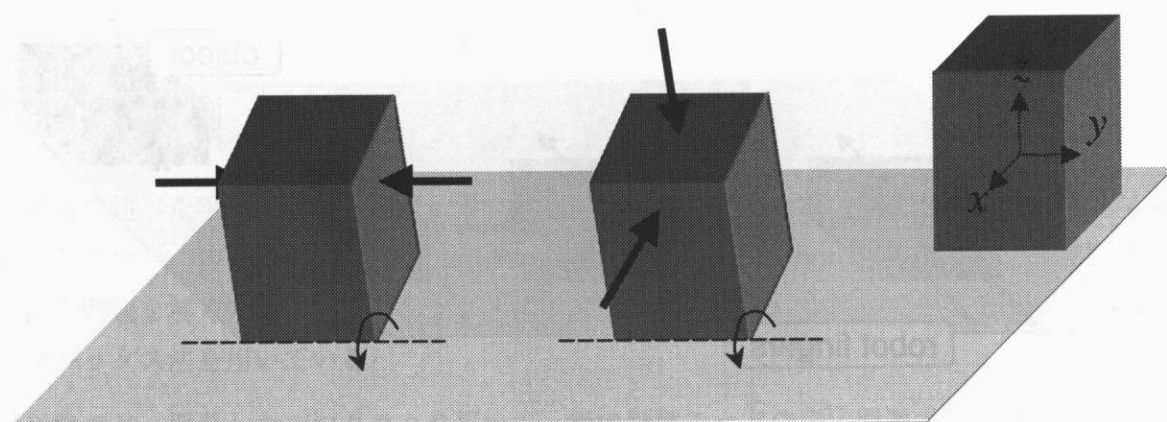


Fig. 5.3: Tumbling a Cuboid

5.5.2 例2：直方体の転がし操作

次に、Fig. 5.3 のような $1 \times 1 \times 2$ の大きさの直方体の転がし操作を考える。対象物の質量は 1 とする ($M_o = 1$)。座標系は図中に示すようにとり、その原点は対象物の質量中心である。この場合、 $J_o = \text{diag}(5/12, 5/12, 1/6)$ である。ここでは対象物の傾き角が $30[\text{deg}]$ の瞬間を考え、このとき、 $Q_{\text{known}} = [4.9, 0, -8.5, 0, 0, 0]^T$ となる。

この直方体を $[0, \pm 1/2, 1/2]^T$ の位置でつまんで動かす場合 (Fig. 5.3 左)、片方の指を位置制御、もう片方の指を力制御するのが最適となる (ただし、これはあくまで対象物の傾き角が $30[\text{deg}]$ の瞬間だけの「最適」であることに注意されたい)。この場合は対称性より位置制御と力制御が入れ替わっても構わない。力制御指の指令指力は 8.8、操作の确实性の値は 2.5 となる。

一方、Fig. 5.3 右のように、指先を $(0, 0, 1)$ および $(1/2, 0, 1/2)$ に置いて動かす場合は、両方の指を位置制御するのが最適となる。このとき操作确实性の値は 1.2 である。

同様に Pentium4-1.6GHz を搭載した Linux PC において、計算時間は前者の場合で 1.1 CPU 秒、後者の場合で 0.01 CPU 秒未満であった。後者の計算が早いのは、すべての指を位置制御にするパターンを調べた時点でこれが最適とわかってしまうからである。一方、前者では指力を最適化する計算をしなければならないため、計算時間が長くなる。

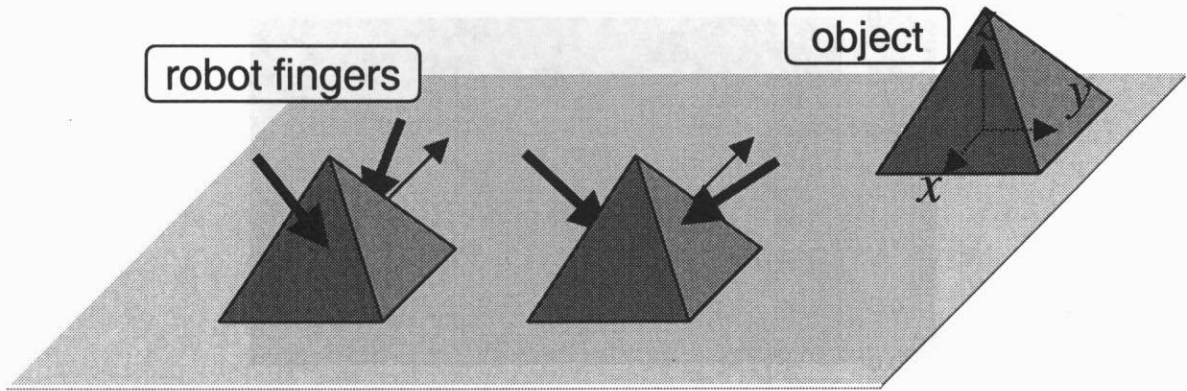


Fig. 5.4: Sliding a Pyramid

5.5.3 例3：平面上での四角錐の滑らせ操作

平面上での四角錐対象物の滑らせ操作を考える (Fig. 5.4). 四角錐の底面は 2×2 の正方形, 高さも 2 とする. また, 四角錐の質量は 1 とする ($M_o = 1$). 座標系は図のように, 対象物の質量中心が原点になるようにとる. この場合, $J_o = \text{diag}(7/20, 7/20, 2/5)$ および $Q_{\text{known}} = [0, 0, -9.8, 0, 0, 0]^T$ である.

2本のロボット指でこの対象物を $[-1, 0, 0]^T$ の方向に動かすとしよう. Fig. 5.4 左のように, $[\pm 1/2, 0, 1/2]^T$ の位置で物体を前後からはさんで動かす場合, 本章のアルゴリズムでは, 手前の指を力制御, 奥側の指を位置制御とするのが最適となった. 力制御指の指令指力は 10, 操作の確実性の値は 4.0 となる.

一方, 四角錐を $[0, \pm 1/2, 1/2]^T$ の位置で左右からはさんで滑らせる場合 (Fig. 5.4 右) は, 一本の指を位置制御, もう片方の指を力制御するのが最適となる. この場合是对称なので, 位置制御/力制御を入れ替えても構わない. 力制御指の指令指力は 10 となり, 操作確実性の値は 2.9 である.

上記の数値例の場合, Pentium4-1.6GHz を搭載した Linux PC において, 1 回の計算に 0.5 ~ 0.8 CPU 秒を要した.

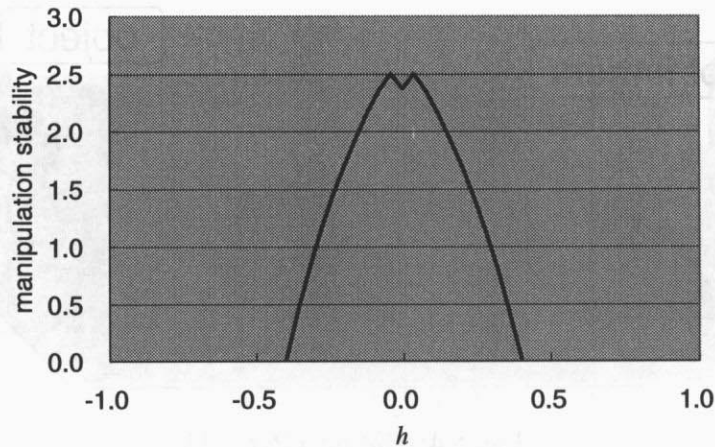


Fig. 5.5: Manipulation Stability of Two-Finger Sliding (1)

5.5.4 パラメータの変動の効果

ここでは、提案した手法によって指の制御モードを決定する際に、各種パラメータの変動によって操作の確実性の指標値がどのように変化するかを、Fig. 5.2 中央の、直方体を左右からつまんで滑らせる操作を対象に見てみる。なお、指標の値の変動については、近似解法による誤差の影響も含まれているので、細かい変動を見るときは注意が必要である。

いま、指位置を $[\pm h, \pm 1, 0]^T$ (複号同順) とし、 h を -1 から 1 まで変化させたとき、すなわち 2 本の指を前後方向にずらした場合の操作の確実性の値を Fig. 5.5 に示す。幾何学的対称性より、操作の確実性のグラフも $h = 0$ に関して対称となる。 $h < -0.4$ もしくは $h > 0.4$ では、操作を実現することが力学的に不可能となることがわかる。

次に、指位置を $[h, \pm 1, 0]^T$ とし、 h を -1 から 1 まで変化させたとき、すなわち 2 本の指を前後方向に水平に平行移動させたときの操作の確実性の値を Fig. 5.6 に示す。この場合は操作の確実性の変動は比較的緩やかである。

続いて、指位置を鉛直方向にずらす場合を考える。指位置を $[0, \pm 1, \pm h]^T$ (複号同順) とした場合、および $[0, \pm 1, h]^T$ とした場合について、それぞれ h を -0.5 から 0.5 まで変化させたときの操作の確実性の値を Fig. 5.7, 5.8 に示す。Fig. 5.7 は、幾何学的対称性より $h = 0$ に関して対称となる。なお、 $h = \pm 0.45$ 付近で値が急に大きくなるのは、過大な内力の発生可能性が無くなって、両方の指を位置制御モードとできるようになるためである。Fig. 5.8 では、操作の確実性の値に大きな変化は見られない。

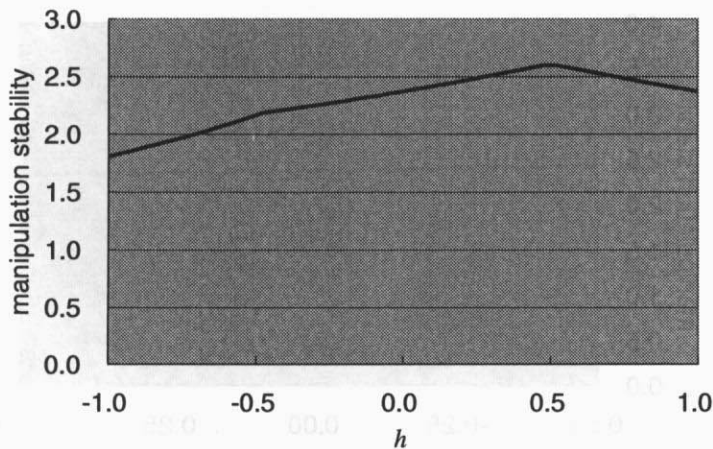


Fig. 5.6: Manipulation Stability of Two-Finger Sliding (2)

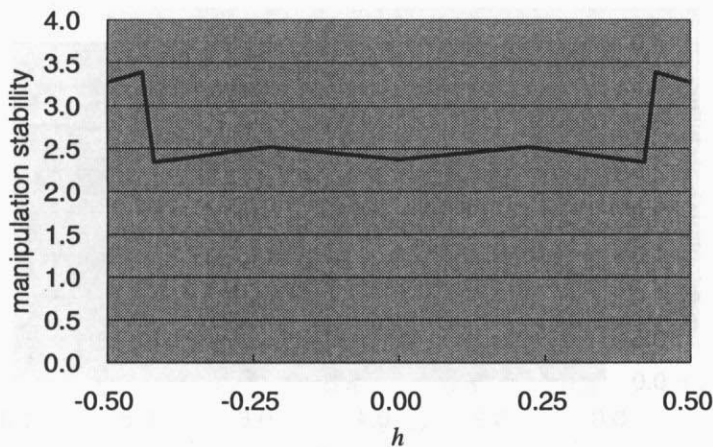


Fig. 5.7: Manipulation Stability of Two-Finger Sliding (3)

一方、指位置を $[0, \pm 1, 0]^T$ に固定し、摩擦係数を変えてみる。対象物と環境の間、およびロボット指と対象物との摩擦係数をそれぞれ μ_{env}, μ_{rob} とし、 $\mu_{env} = 0.2$ として μ_{rob} を 0 から 1 まで変化させたときの操作の确实性の値を Fig. 5.9 に示す。また、 $\mu_{rob} = 0.5$ として μ_{env} を 0 から 1 まで変化させたときの操作の确实性の値を Fig. 5.10 に示す。 μ_{rob} については、大きければ大きいほど操作の确实性は大きくなっていることがわかる。また、 $\mu_{rob} < 0.1$ では、力学的に操作が不可能であることが分かる。一方 μ_{env} については、大きければ大きいほど、この場合は摩擦に打ち勝って物体を押すことが困難になり、操作の确实性が減ることが見てとれる。

以上の例からは、提案した手法によって指の制御モードを決定するとき、多くの場合で

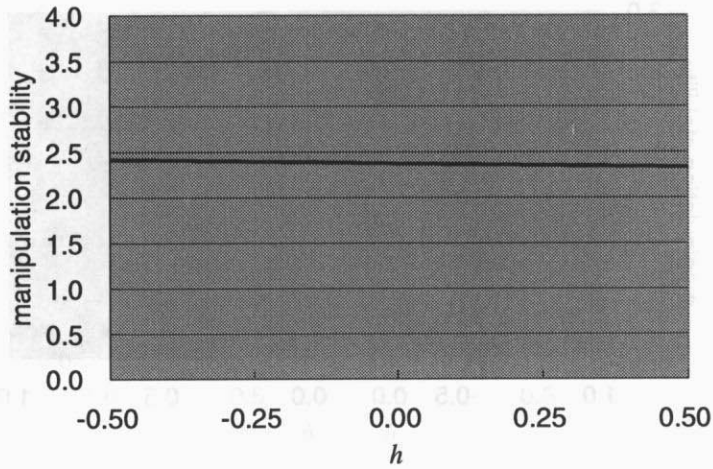


Fig. 5.8: Manipulation Stability of Two-Finger Sliding (4)

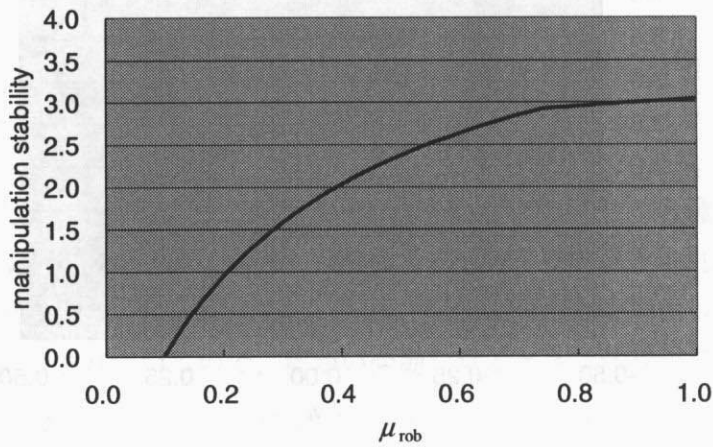


Fig. 5.9: Manipulation Stability of Two-Finger Sliding (5)

は、パラメータの変動に対して操作の确实性の値が不連続に変化してしまうことはないと思われる。しかし、指の制御モードが切り替わる境界付近では、切り替えの結果として操作の确实性が急に変化することもありうるので、注意が必要である。

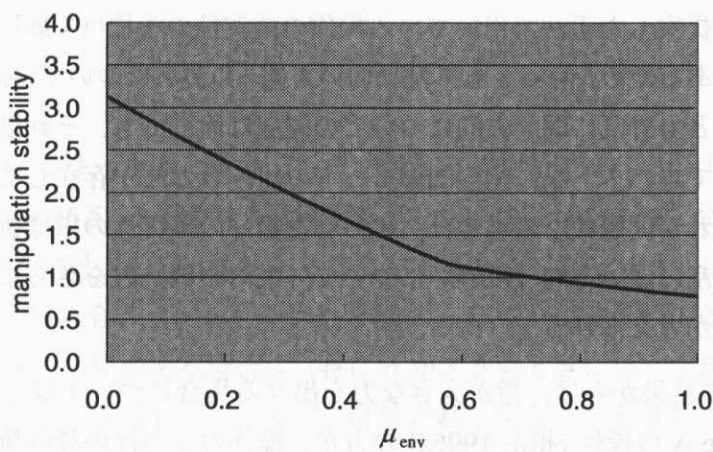


Fig. 5.10: Manipulation Stability of Two-Finger Sliding (6)

5.6 制御モード決定手法に対する考察

5.6.1 提案手法の性質

前節の数値例では、本章で提案したアルゴリズムは、おおむね適切な制御モードを決定できていると言ってよい。したがって、本章で提案した制御モード決定の方針、すなわち「過大な内力の発生の恐れがない限りにおいて、できるだけ操作の確実性の値を大きくする」という方法は、グラスプレス・マニピュレーションにおいては妥当であると考えられる。

また、この方針に従った場合、力制御よりも位置制御をできるだけ採用するようになる。このことは実機におけるマニピュレーションをより容易に実現する上でも有効である。

ただし、本章で提案した手法で用いている操作の確実性の指標（3章）は、ミニマックス基準によるものであるため、もっとも外乱に弱い方向「以外」については評価が行われない。したがって、もっとも外乱に弱い方向についての評価は同じでも、それ以外の方向についてはより外乱に対してロバストな、指の制御モードの組み合わせが存在しても、それを見落とすことがある。これが問題になる場合は、得られた操作の確実性の指標値を拘束条件として加え、もっとも外乱に弱い方向「以外」について外乱への抵抗力をさらに最適化する、などの方法をとることが考えられる。

また、5.5.1項の結果からは、指が大きな力を出せる場合については、内力を伴うあやつり、すなわち Type A の操作 [相山 1996a] の方が、操作の確実性の評価値が高くなる傾向が見て取れる。この結果自体は当然であるが、本章の手法を使ってマニピュレーションを選択する場合には、内力を大きくした状態でのマニピュレーションが選ばれやすくなる、ということには注意しておく必要がある。

なお、用途によって、本章で提案した制御モード決定のアルゴリズムに少し手を加えることも考えられる。例えば、ある外力仕様（すなわち、ある集合内の任意の外力に耐えられること）を満たすようにパワーグラスプの最適化を行った研究 ([余 1999, 小俣 2002]) があるが、これと同様な方法で制御モードを決定してもよい。この場合、操作確実性の値を最大化するのではなく、操作確実性の値が一定値以上になることを制約条件とし、他の評価関数（例えば指力の和など）を最小化する、といった方法をとることになる。

5.6.2 計算量

本節では、提案した指の制御モード決定アルゴリズムの計算量を考える。

提案手法では、操作の确实性を最大化する部分 (5.5) と過大な内力の発生可能性を判別する部分 (5.6) の 2 種類の線形計画問題を解く。しかし、それぞれの制約条件の数は、前者が最大で $(6 + 3n)N_{\text{vert}} = O(N_{\text{vert}}n)$ (標準形に変換した場合)、後者が 6 である。また、変数の数は前者が最大 $N_{\text{vert}}s(m+n) + n + 1 = O(N_{\text{vert}}s(m+n))$ (標準形に変換した場合) なのに対して、後者は $s(m+n)$ である。つまり、問題の規模は (5.5) の線形計画問題の方が圧倒的に大きく、したがってこちらの計算量が支配的になる。

シンプレックス法によって線形計画問題を解く場合、実際的には繰り返しの回数はほぼ拘束条件の数に比例し、また各繰り返しごとの演算量は、拘束条件の数と変数の数の積に比例するとされている [伊理 1995]。したがって、(5.5) をシンプレックス法で解いた場合、その計算量はおおよそ

$$\begin{aligned} &O(N_{\text{vert}}n) \times (O(N_{\text{vert}}n) \times O(N_{\text{vert}}s(m+n))) \\ &= O(N_{\text{vert}}^3 sn^2(m+n)) \end{aligned} \quad (5.8)$$

と見積もられる。

さらに、提案手法では、最悪の場合、指の位置制御/力制御モードのすべての組み合わせ (2^n 通り) について、(5.5) の線形計画問題を解く必要がある。したがって、提案手法の計算量は、おおよそ $O(2^n N_{\text{vert}}^3 sn^2(m+n))$ と見積もられる。この数字は決して小さいものではないが、実際には、組み合わせを調べる回数を調べる工夫 (5.4.4 項) の効果により、すべての組み合わせを調べる必要がある場合は (特に n が大きいときには) まれである。また過大な内力の発生可能性がある場合など、すぐに計算が終了することも多い。数値例の計算時間からも、実際的な計算量はより小さいことが予想される。

5.7 おわりに

本章では、グラスプレス・マニピュレーションでの指の制御モード決定問題を扱った。これは把持における指力決定問題を拡張したものに相当する。本章で提案した手法により、一連の線形計画問題を解くことで、各ロボット指の制御モード（位置制御／力制御）および、指令指力（ただし力制御の指についてのみ）を決定できることが示された。

本章の内容は、グラスプレス・マニピュレーションのある瞬間について、指の制御モード・指力を最適化する問題であった。これは、グラスプレス・マニピュレーションの計画問題の中の局所的な問題を解いたことになる。

次章では、この結果を踏まえ、より大域的な計画問題を解くことを考える。すなわち、操作対象物の初期位置・姿勢と目標位置・姿勢が与えられたときに、それをグラスプレス・マニピュレーションで実現するための、各ロボット指の位置・制御モード・指力の系列を決定する問題を扱う。