

7.5.0  
4

# 視覚と力覚の情報を用いたロボットの知的制御に関する研究

(A Study on Intelligent Control of Robots Using Visual and Force/Torque Information)

2004 年 3 月

横浜国立大学附属図書館



11472512

黄 健  
(Jian HUANG)



3.3.1	画像処理による特徴点の算出	30
3.3.2	画像処理によって検出された特徴点位置の補間	32
3.4	ニューラルネットワークのオンライン学習によるロボットの制御	33
3.4.1	ニューラルネットワークの構成	34
3.4.2	ニューラルネットワークの学習	35
3.4.3	ニューラルネットワークの学習を用いたロボットの制御	36
3.5	実験とその結果	39
3.5.1	鉄製棒状物体を用いたはめ合い実験	40
3.5.2	アクリル製の棒状物体を用いたはめ合い実験	41
3.6	要約	44

## 第4章 柔軟ツールを用いたロボットの制御

---

4.1	概説	45
4.2	作業システムの構成と作業の設定	45
4.3	画像処理で得られたツール先端位置の補間	46
4.4	ツール先端の位置によるロボットの目標軌道の生成	48
4.4.1	ロボット手先の目標位置の生成	49
4.4.2	ロボット手先の目標姿勢の生成	50
4.4.3	目標力の生成	52
4.5	位置と力のハイブリッド制御系の構成	52
4.5.1	ニューラルネットワークの構成	53
4.5.2	ニューラルネットワークの学習	53
4.5.3	オンライン学習型ニューラルネットワークを用いたハイブリッド制御	53
4.6	実験とその結果	56
4.6.1	ねじりモーメントによるツール先端の位置追従誤差の影響を確認する実験（姿勢制御なし，NNなし）	56
4.6.2	ツールの先端を目標軌道になぞらせる実験（NNなし）	58
4.6.3	ニューラルネットワークのオンライン学習を用いた追従実験	60
4.7	要約	62

## 第5章 冗長ロボットによる衝突の回避とその制御

---

5.1	概説	63
-----	----	----

5.2	立体視画像処理による衝突対象の位置の検出	64
5.3	仮想関節トルクによる冗長関節の目標軌道の生成	65
5.3.1	仮想ポテンシャル空間の設定	65
5.3.2	仮想関節トルクの生成	66
5.3.3	冗長関節の目標軌道の生成	67
5.4	冗長ロボットの制御	67
5.4.1	手先のコンプライアンス制御	68
5.4.2	評価関数を用いた冗長関節の選定	69
5.4.3	冗長ロボット制御系の構成	70
5.5	実験とその結果	72
5.5.1	実験1 (衝突物体を回避しながら手先での コンプライアンス特性を保つ実験)	73
5.5.2	実験2 (手先に半円軌道の倣い作業を させながら衝突を回避する実験)	76
5.6	要約	79

## 第6章 視覚情報を用いたシーリングロボットの目標軌道の実時間生成法と制御

6.1	概説	80
6.2	三次元視覚センサによるシーム位置の検出	81
6.2.1	三次元視覚センサの構成とシーム位置・姿勢の検出	81
6.2.2	画像処理アルゴリズム	83
6.3	シーリングロボットの目標軌道の生成法	84
6.3.1	ノズルの目標軌道の生成	84
6.3.2	視覚センサの目標軌道の生成	85
6.3.2.1	ファジィ推論	86
6.3.2.2	ファジィルールの学習則	87
6.3.2.3	ファジィルールの学習と視覚センサの目標軌道の生成	90
6.3.3	ロボットの目標軌道の生成	91
6.4	シーリングロボットの追従実験	94
6.4.1	シーリングロボットの視覚フィードバック制御	94
6.4.2	シーリングロボットシステム	94
6.4.3	追従実験と結果	96
6.5	要約	99

第7章 結論	100
<hr/>	
謝辞	102
<hr/>	
文献	103
<hr/>	
本論文を構成する公表論文	108
<hr/>	
付記 A 可搬式汎用多関節知能アームの機構と運動解析	110
<hr/>	
A・1 概説	110
A・2 マニピュレータの構成	110
A・3 座標系の定義	110
A・4 マニピュレータの運動学	112
A・4・1 $x$ - $y$ - $z$ 固定角法による手首姿勢の記述	112
A・4・2 同次変換行列による手首の位置・姿勢の記述	113
A・4・3 マニピュレータの順運動学	114
A・4・4 マニピュレータの速度解析	117
A・5 力覚センサ座標系と運動解析	122
A・6 ツール座標系と運動学	123
A・6・1 ツール座標系の順運動学	123
A・6・2 ツール座標系の速度解析	124
<hr/>	
付記 B 最小二乗法によるカメラのキャリブレーション	125
<hr/>	
B・1 概説	125
B・2 カメラモデル	125
B・3 カメラ座標系とロボットベース座標系	126
B・4 最小二乗法によるカメラのキャリブレーション	127

---

# 第1章

## 序論

---

### 1.1 概説

これまでのロボットのほとんどは、産業用ロボットのように人間から隔離された環境の下で用いられてきた。このようなロボットの作業対象は主に剛体である。また、大量生産という目的のため、パワーとスピードがロボットの制御に求められてきた。しかしながら、図1.1に示すように、これからのロボットは、福祉、介助、介護など人間の日常生活を支援する場で活躍することが期待される。このようにロボットの動作環境が変わるとともに、作業対象は金属などの剛体から布や紙などの柔軟物体に変化してきた。この分野においては、ロボットの制御について未解決の問題は多い。柔軟かく変形しやすい作業対象をロボットによって操作する場合、ロボットが作業対象を破壊し、人間に危害を加えてしまう可能性があるため、パワーとスピードよりロボットの安全性および環境に対する適応能力が、ロボットの制御において求められている。人間のように、ロボットがあらゆる環境において作業を行うためには、必要な環境情報を外界センサによって獲得し、自律的に作業を遂行することが望ましい。図1.2に示すように、本研究では、作業対象やロボットの手先ツールなど作業環境が変化する問題を取り上げ、柔軟性をもつロボットの新しい制御手法を導く。

ロボットの周囲の環境が未知である場合や対象物体の状態が変化する場合、力覚センサ、視覚センサなど外界センサによって環境の情報を獲得する必要がある。これまでの研究では、力覚情報はロボットの制御に広く用いられてきた。特にトルクや速度などを指令値に用いるロボットの制御系において、力覚センサによって測定した作業対象との接触力の情報を扱うことは容易である。一方、人間の場合、感覚器のうち視覚に80%以上を頼るといわれるため、視覚情報は環境の情報の獲得において重要な位置を占めることが分かる。対象物体の位置、形状の変化、ロボットの近くにいる人間および障害物の有無などロボットの周囲状況を視覚センサによって非接触で観測することができれば、環境が変化した場合に対しても、ロボットを柔軟に対応させることが可能である。したがって、このような視覚センサ、力覚センサなど外界センサを

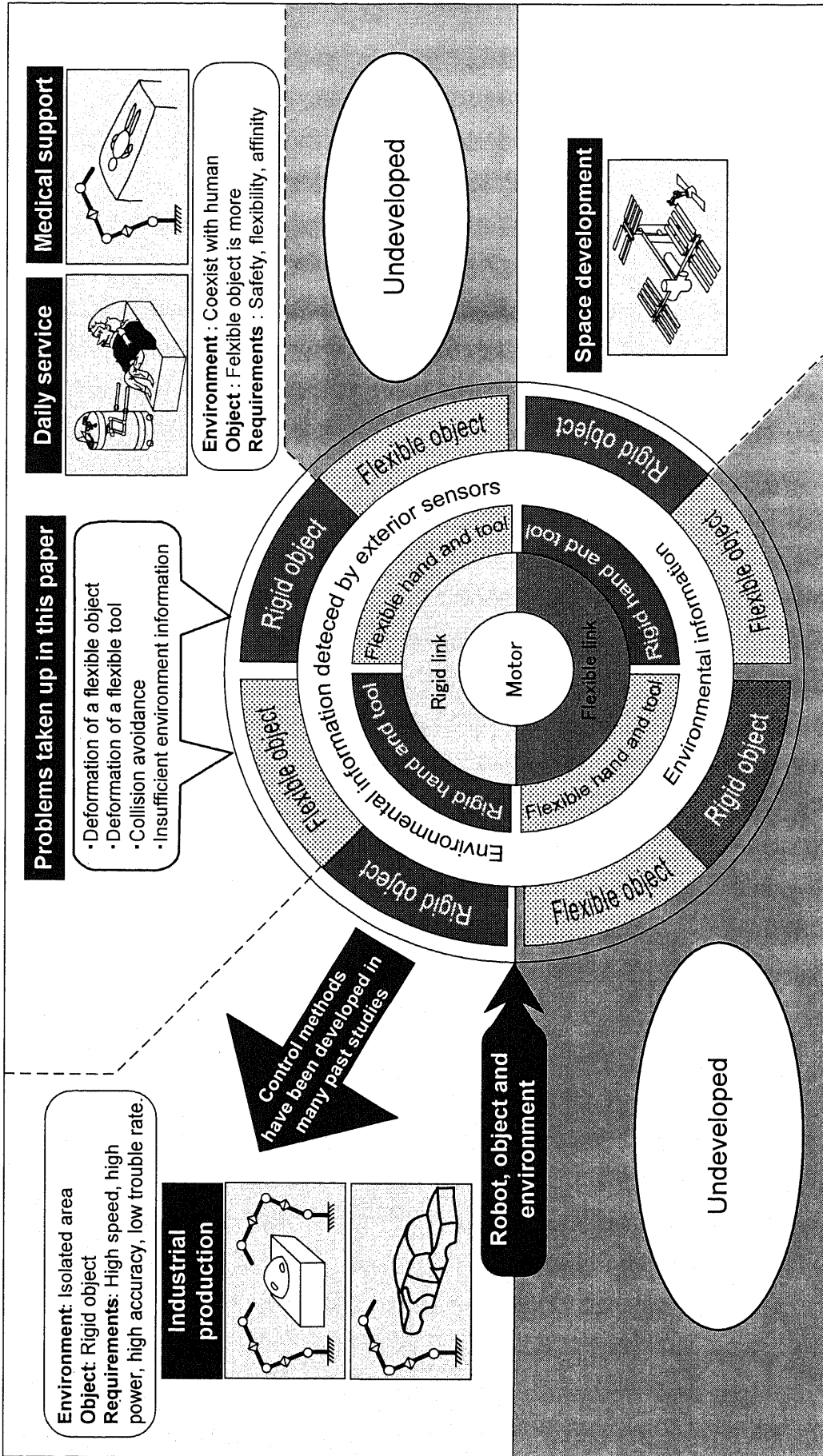


Fig. 1-1 New technology required for robot control with drastic change of social needs

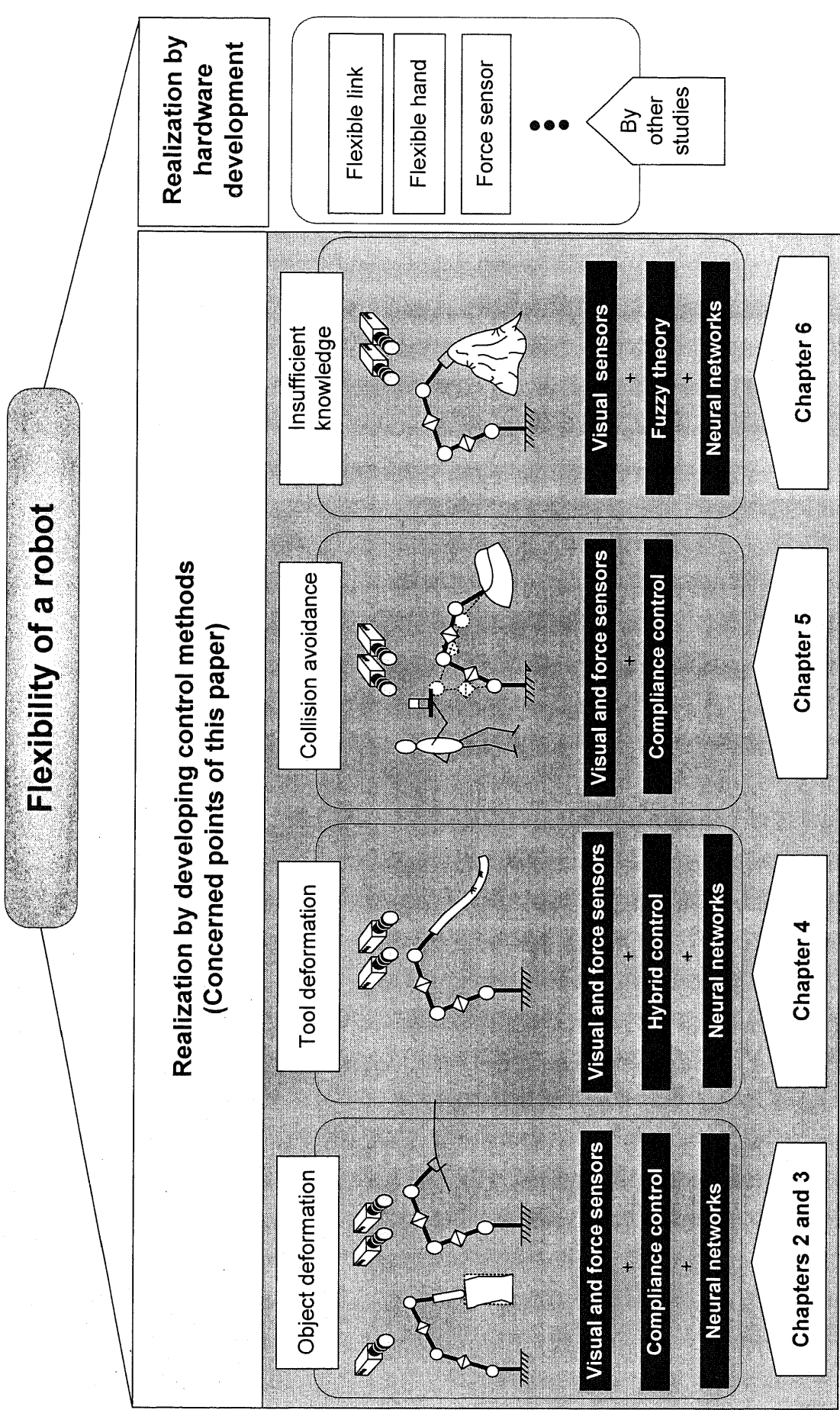


Fig. 1·2 Control method of robots discussed in the proposed study



用いることによって、実用性、汎用性の高いロボットの制御系を構成することが可能である。未知環境においてロボットに作業を遂行させるには、実用性、汎用性の高いロボットの制御系を構成する必要がある。一般に、ロボットの作業環境における情報としては、①対象物体の特性やロボット手先に取り付けられているツールの特性、②ロボットに衝突する人間や障害物の有無、③未知環境におけるロボットの目標軌道 などがある。したがって、これらの環境の情報に基づいてロボットの制御系を構成する場合、図 1・1 に示すように、次の問題を考えなければならない。

(1) 対象物体が変形する場合、これまで剛体対象を扱う作業のために開発したロボットの制御法を柔軟物体の操作にそのまま適用することはできない。

(2) モデリングによって対象物体の変形を算出する場合、曲げ剛性など対象物体の材質に関するパラメータの値が必要であるが、日常生活の環境における対象物体の材質に関するパラメータが不明の場合が多い。

(3) ロボットの手先に柔軟ツールを取り付ける場合、柔軟ツールの変形があるため、ロボットの運動学によってツール先端を制御するこれまでの手法をロボットの制御に適用することはできない。

(4) 既知の障害物との衝突にロボットを回避させる手法を、これまでの研究で開発してきたが、日常生活においてロボットの近くを通りかかる人間および近づいてくる障害物の位置を事前に予測できない場合が多い。

(5) 画像処理の時間が長い場合、ビジュアルフィードバック制御を用いたロボットの目標軌道を生成することは困難である。

これらの問題を解決するためには、柔軟性をもつロボットを構成する必要がある。図 1・2 に示すように、本研究では、力覚センサ、視覚センサなど外界センサによって環境の情報を獲得し、知的手法を用いてロボットを制御するという新しい手法を提案する。

ロボットによる接触作業を行うこれまでの研究に力覚センサが広く用いられてきた。しかしながら、問題(1)に示すようにロボットの手先に取り付けられる力覚センサを用いてロボットの手先と対象物体との接触力を検出することはできるが、接触によって生じる柔軟物体の変形を検出することはできない。このため、これまで剛体対象を扱う作業のために開発したロボットの制御法を柔軟物体の操作にそのまま適用しても作業を遂行できないことが多い。人間は柔軟物体を扱うとき、手先で検出した接触力と目で検出した視覚情報を脳で融合し、腕と手の目標位置を生成したのち速やかに行動を行う。したがって、ロボットによる柔軟物体を操作する場合、力覚情報だけではなく視覚情報を積極的にロボットの制御に利用することが大切であると考えられる。

しかしながら、視覚情報のデータ量は膨大であるため、画像処理に時間が掛かる。このため、視覚情報を用いたこれまでの研究において画像の取り込みと画像処理を並列に行うのではなく、画像を取り込んでから処理するというループを繰り返す手法を用いた。

その結果、画像情報を処理している最中にカメラから連続的に出力される画像を取り込むことができないため、画像情報の一部しか利用しなかった。一般に、入力された画像データを処理しながらカメラからの画像を連続的に取り込むという並列な処理手法を本論文では「画像のリアルタイム処理」と呼ぶ。画像のリアルタイム処理を実現するには、ハードとソフトの両面において工夫する必要がある。また、画像の取り込み・処理の周期はロボットの制御サンプリング周期より長いため、ロボットの制御サンプリング周期ごとに得られた画像処理の結果を補間する必要がある。

問題(2)に示すように、生活環境において柔軟物体の材質は多様であり、これらの柔軟物体の材質に関するパラメータは未知であることが多い。この問題を解決するには、視覚と力覚の情報を併用することが有効と考えられる。視覚と力覚の情報のセンサフュージョンを用いることによって、柔軟対象物体の変形モデルを求める必要がなくなるため、実用性の高いロボット制御系を構築することができる。さらに、ニューラルネットワークをロボット制御系に用いることによって、対象物体の変形特性の学習を行い、材質が類似している対象物体の作業を速やかに遂行することができる。

問題(3)に関しては、ロボットの手先に取り付けられた柔軟ツールの変形を何らかの方法によって解析し、柔軟ツールの先端位置を算出する必要がある。柔軟ツールの変形モデルによって変形解析を行うことは可能であるが、①材質は不明な柔軟ツールを用いる場合、②変形モデルを作成することは困難な場合 などには、新しい制御手法を提案することが必要である。

問題(4)に関しては、生活現場に作業をロボットに行わせるとき、作業対象とロボット手先の状態を外界センサによって検出し対応するのだけではなく、ロボットの周囲環境を常に観測し、変化があった場合には柔軟に対応しなければならない。したがって、安全かつ実用なロボット制御系を構成するには、ロボットの近辺に接近してくる人間や障害物に対し、手先の作業を行いながらできる限りの回避制御を行い、衝突を柔軟に回避することが望まれる。

問題(5)に関しては、視覚センサをロボットの手先に取り付け、ロボットのビジュアルフィードバック制御系を構成する場合、画像処理に長い時間を必要とすることは問題になる。視覚センサをロボットの手先に取り付ける場合、対象物体を撮影する範囲内に視覚センサが置かれることが必要であるため、ロボット手先の位置制御と同様に視覚センサの位置を制御しなければならない。この問題解決するため、すでに得られた情報を知的手法によって解析し、視覚センサとノズルの両方の目標軌道を生成するという新しい手法を提案することが期待される。

本論文では、これらの問題に着目し、その解決法についてまとめた。

## 1-2 従来の研究と本研究の目的

これまでのロボットのほとんどは剛体を扱う作業に用いられてきたが、生活環境、医療現場、生産現場においては柔らかく変形しやすい物体は数多く存在しており、柔軟物体を扱うロボットの新しい制御法がこれからの社会で求められている<sup>(1)</sup>。柔軟物体の種類は多く、力学的性質は多様であるため、剛体を対象とする作業に用いた制御法をそのまま柔軟物体の操作に適用して作業を遂行できないことは多い。この背景の中で、ロボット工学の分野において管状や線状の柔軟物体の操作<sup>(2)~(5)</sup>、柔軟曲面での扱い<sup>(6)</sup>、布のマニピュレーション<sup>(7)</sup>、柔軟物体のモデリング<sup>(5),(8)</sup>、画像情報を用いた柔軟物体の操作<sup>(3),(9)</sup>などの研究をはじめ、柔軟物体を扱うロボットの制御法に関する研究が行われている。しかしながら、これらの研究の多くは、対象物体の変形解析を行うことが必要であるため、未知の対象物体を用いた接触作業にこれまでの手法を適用できない。

柔軟物体のなかでは、コード、ケーブル、糸、ロープ、チューブなど細長い物体（以下、棒状物体と呼ぶ）が生産現場、生活環境、医療福祉など幅広い分野で多く扱われている。棒状物体の材質によって力学的特性が大きく異なるため、ロボットによる棒状物体の操作は困難である<sup>(1)~(5)</sup>。特に、小さい穴に棒状物体を挿入する場合、挿入力によって棒状物体に弾性座屈が生じるため、この作業を遂行する方法を開発する必要がある。棒状柔軟物体の操作において、ポテンシャルエネルギーが最小となるように柔軟物体のモデリングを行うという手法が提案された<sup>(5)</sup>。また、力覚情報を利用できない場合、画像情報のみを用いて棒状柔軟物体の先端に掛かる力を推定しながらロボットによる棒状柔軟物体の挿入作業を遂行するという手法<sup>(2),(5)</sup>も報告された。これらの研究に対し、管状柔軟物体の挿入作業を人間に行わせ、作業を行ったときの人間のスキルをマニピュレータに移植するという手法が提案され、それを用いて管状物体の挿入作業をマニピュレータに遂行させたことが報告された<sup>(3)</sup>。しかしながら、棒状物体の変形をモデリングすることによって棒状物体の先端に掛っている力を求める場合、曲げ剛性など棒状物体の材質に関するパラメータの値が必要である。また、開発された従来の方法は、決められた対象の作業に限定しているため、異なる対象での作業に適用することは困難である。

さらに、生活環境においては、柔らかく変形しやすい物体を扱う作業が多い。これに対し、壊れやすい剛体対象の場合には、ロボットの手先に柔軟ツールを取り付け、ツール先端の位置と力を制御する必要性がしばしばある。しかしながら、ツールの材質によって変形モデルを算出できない場合には、運動学に基づくこれまでの手法をツール先端の位置制御に適用することは難しい。

対象物体の変形問題を解決するため、第2、3、4章では、視覚と力覚の情報をニューラルネットワークの学習によってフュージョンさせ、その結果を用いてロボットによ

る柔軟物体を操作するという新しい制御手法を提案する。提案した手法を用いて二次元の変形を伴う“ペグインホール作業”をロボットに行わせる。また、提案した手法をさらに発展させ、三次元空間における棒状柔軟物体の“はめ合い作業”をロボットに遂行させる手法を提案し、実験によってその効果を確認する。ロボット手先に取り付けられた柔軟ツールが変形するという問題を解決するため、第4章では、立体視画像情報を用いてツールの先端位置を高速で検出することによって位置と力のハイブリッド制御系を構成するという手法を提案する。これらの手法では、視覚情報を用いるため、モデルによってツールの変形を解析する必要はないという利点がある。

一方、医療現場や日常生活においては、ロボットの周辺には人間や障害物の予測できない接近が多い。このため、ロボットが人間や障害物に衝突すると、事故につながる危険性は極めて高い<sup>(22)~(24)</sup>。これを回避するため、冗長ロボットを用いて柔軟に対応することが期待される。冗長ロボットは、汎用性の高い人間の腕と同じように作業環境に応じて関節の冗長自由度を利用できるため、特異点の回避<sup>(25)~(26)</sup>、障害物の回避<sup>(27)~(32)</sup>、インピーダンス制御<sup>(33)~(36)</sup>に広く用いられてきた。しかしながら、これまでの障害物の回避研究のほとんどは、障害物の位置が既知で静止しているという条件のもとで開発された手法である。実際には、この条件が成立しない場合が多い。このような場合においても衝突を回避しなければならない。それは、接触によって発生した衝撃力が装置を破壊する恐れのある作業では、接触力を検出してからインピーダンス制御を始めるのでは遅いからである。不特定な人間や障害物とロボットとの衝突を回避するため、未来の目標軌道及び形状情報をロボットの冗長制御に用いる予見制御法<sup>(37)</sup>が報告された。また、複数の移動ロボットを用いた作業の場合、移動ロボット同士またはロボットと障害物の間に仮想的なインピーダンスを設けて協調作業を実現する手法<sup>(38)</sup>も報告されている。

近年、障害物の回避制御に視覚センサを用いたアプローチが多くなってきた。これまでに、障害物の回避に必要な画像特徴の推定<sup>(39)</sup>、三次元再構成を行わない視覚サーボ系を用いた障害物の回避軌道の生成<sup>(40)</sup>などの報告があった。また、視覚情報を用いて対象物の状態によってビジュアルコンプライアンス制御を行うという手法<sup>(41)</sup>、カメラ1台を用いて平面冗長ロボットの手先と対象物間のインピーダンスを制御する手法<sup>(42)</sup>、ニューラルネットによる非接触型インピーダンス制御のパラメータの学習<sup>(43)</sup>が報告された。

作業中のロボットに人間や障害物など衝突対象が接近してきたとき、緊急の場合にはロボットの手先の作業を直ちに中止し衝突を回避しなければならないが、緊急の場合以外は、ロボットは手先の作業を行いながら近づいてくる人間や障害物との衝突を可能な限り柔軟に回避することが望ましい。また、作業を行う際に特異点の近辺での作業をできるだけ避けることが必要であるため、手先の作業と衝突の回避作業を両立させると同時にロボットの可操作性を考慮する必要がある。第5章では、人間や障害物など不特定な対象がロボットに近づいてくる問題を解決するため、ロボットの周辺に仮想ポテンシ

ャル場を設け、仮想ポテンシャル場に入った人間や障害物を立体視覚情報によって検出したとき仮想トルクを発生させるという手法を提案する。この手法を用いて倣い作業をロボットの手先に行わせ、作業中に近づいてくる人間や障害物との衝突を柔軟に回避することを確認する。

視覚情報は、環境の情報を獲得する有効な手段として幅広い分野に用いられてきた。しかしながら、画像情報の量は膨大であるため、処理するのに時間が掛かる。とくに、視覚センサをロボットの手先に取り付け、ロボットのビジュアルフィードバック制御系を構成する場合、ロボットの未来の目標軌道を生成すると同時に、視覚センサの未来位置を対象物体に外れないように考慮しなければならない。第6章では、ロボットを目標経路に追従させるというシーリング作業を取り上げ、視覚情報を用いたロボットの未来の目標軌道をどのように生成するかを検討する。

これまでに張・藤堂は、CCDカメラで取り込んだ薄鋼板の重ね合わせ部の画像に対して特徴をニューラルネットワークによって抽出する手法<sup>(50)</sup>、距離情報と濃淡情報の統合に基づく継目位置を検出することによってシーリングロボットを制御する手法<sup>(10)</sup>を提案した。すなわち、オフラインであらかじめ環境情報の教示を行い、作業中に視覚センサを用いて対象物体の位置を検出することによって、前もって教示された環境情報を再生中に修正しながらロボットに追従作業を行わせるという手法を提案した。また、形状が複雑な自動車鋼板の継目ラインでのロボットの教示作業に手間が掛かる問題に対して視覚センサによる教示軌道の自動生成法<sup>(12)</sup>、三次元視覚センサを用いて教示作業を必要としない目標軌道自動生成法<sup>(11),(13)</sup>が報告された。さらに、未知環境内で動作するロボットのビジュアルサーボ系の構成<sup>(54)</sup>、視覚情報と力覚情報のオンライン融合を用いたDDロボットのハイブリッド制御法<sup>(14)</sup>、視覚情報を用いたDDロボットの位置と力の制御法など研究を行った<sup>(15)</sup>。しかしながら、これまで開発された教示を必要としない手法では、ロボット手先の位置を制御したが、視覚センサの位置制御を考慮していない。第6章では、画像処理の時間が長い問題を解決するため、視覚センサからすでに得られた情報を知的手法によって解析し、視覚センサの目標位置とロボットの目標軌道を同時に生成する手法を提案する。

上記の問題を踏まえて、図1・2に示すようにいくつかの作業モデルを取り上げ、視覚センサと力覚センサを用いて環境の情報を獲得することによってロボットを制御するという新しい手法について研究を行う。したがって、本研究の目的は次のとおりである。

(1) 視覚と力覚の情報を用いたロボットの制御アルゴリズムを提案し、2次元変形を伴う“ペグインホール作業”に本手法を適用することによってその効果を確認する。

(2) 視覚と力覚の情報をニューラルネットワークに入力し、柔軟物体の変形特性を学習することによって、汎用性をもつロボットの制御系の構成法を提案する。提案した手法を三次元変形する棒状物体の操作に適応することによってその効果を確認する。

(3) ロボット手先に柔軟ツールを取り付ける場合、接触によってツールが変形する問題を解決するため、視覚情報を用いてツール先端の位置と力のハイブリッド制御を行うという手法を提案し、なぞる実験を行うことによってその効果を確かめる。

(4) 接近してくる不特定な人間や障害物による衝突を、ロボットの冗長関節によって柔軟に回避するという実用性・汎用性の高い制御アルゴリズムを提案し、実験によってその効果を確かめる。

(5) 視覚センサからすでに得られた情報を知的手法によって解析したのち、その結果を用いて視覚センサとロボットの未来の目標軌道を実時間で生成するという手法を提案し、追従実験によってその効果を確かめる。

### 1.3 本論文の構成

図 1.2 示すように、本論文の構成は、次のとおりである。

第 2 章では、視覚と力覚の情報を用いて変形を伴う“ペグインホール作業”をロボットに遂行させる手法を記述した。まず、高速カメラから出力される画像情報をリアルタイムで処理するため、デジタルシグナルプロセッサを用いて画像情報を並列に処理するという手法を開発した。そして、画像処理で得られた結果と力覚センサで得られた力覚情報をフュージョンさせることによってロボットを制御するという新しい手法を提案し、それによってロボットが二次元の変形を伴う“ペグインホール作業”を遂行した。

第 3 章では、第 2 章で提案した手法をさらに発展させ、三次元空間において変形する棒状変形物体の“はめあい作業”をロボットに遂行させる手法を導いた。具体的には、まず、2 台の CCD カメラを用いて三次元空間における対象物体の変形を観測する画像システムを構成した。制御サンプリング周期より画像処理時間が長いという問題点があるため、画像処理で得られた結果を補間してロボットの制御サンプリング周期ごとの値を求めるという手法を提案した。ついで、対象物体の変形特性を学習するニューラルネットワークを構成した。視覚と力覚の情報をニューラルネットワークの学習によってフュージョンさせ、その結果を用いてロボットを制御するという新しい手法を提案し、棒状物体の“はめあい作業”によって本手法の有効性を示した。

第 2, 3 章において対象物体は作業中に変形する問題を取り上げたのに対し、第 4 章では、ロボットの手先に取り付けられた柔軟ツールが変形する問題に着目する。接触作業における挿入力によって弾性座屈が棒状物体に生じるため、ロボットの運動学によって剛体ツールの先端を制御するこれまでの手法を柔軟ツールの制御には適用できない。この問題を解決するため、まず、ツール先端の位置を高速に検出する立体視観測システムを用いる。ついで、変形する柔軟ツールの先端位置とロボット手先位置の関係を解析し、ロボット手先の制御目標値を生成する。さらに、ツール先端の追従精度を向上させ

るため、ニューラルネットワークを制御系に取り入れるという手法を提案し、柔軟ツールを装着するロボットに“なぞる作業”を行わせることによってその効果を確認した。

本論文の後半（第5，6章）では、ロボットの作業環境の変化や環境情報の不足などの問題に視点をおき、ロボットを制御する手法を導く。第2，3，4章のように変形物体と柔軟ツールの状態に着目し、ロボットの制御を行った。これに対し、第5章では、接近してくる不特定な人間や障害物によって生じる衝突にロボットを柔軟に回避させる手法を導いた。まず、パーソナルコンピュータの中でロボットを囲む仮想ポテンシャル場を設け、仮想ポテンシャル場に入った人間や障害物を立体視覚センサによって検出したとき仮想トルクを発生させるという手法を提案した。ついで、衝突の回避とロボットの可操作度を考慮した評価関数を提案し、衝突を柔軟に回避しながら手先の倣い作業を遂行する制御系を構成した。提案した手法を用いて、ロボットに倣い作業をさせながら近づいてくる物体との衝突を回避させた。

第6章では、視覚センサを用いてロボットをシーリング軌道に追従させる作業を取り上げ、視覚センサとロボットの未来の目標軌道を生成する手法を検討する。シーリング作業とは、自動車車体の水漏れやさびを防ぐため、車体鋼板の継目部にシール剤を塗布する作業である。これまでに藤堂らは、視覚センサを用いてオフラインであらかじめ教示を行い、再生中に前もって教示されたシール軌道を修正しながらシーリングロボットに作業を行わせるという手法を報告した。また、教示を前もって必要とする方法をさらに発展させ、自動教示法、ならびに教示不要な制御手法を開発した。これまで開発した手法においては、画像処理で得られた視覚情報をノズルの目標軌道の生成に用いるため、手先の走行方向に対してノズルが視覚センサの後方に装着されている。しかしながら、従来の研究においてノズルの位置を制御しているが、ロボットの手先に取り付けられた視覚センサの位置を制御していない。特に、教示を必要としない手法の場合、視覚センサはシーリング軌道の曲がる部分において継目を見失ってロボットの手先が軌道から外れやすいと考えられる。

一般に、画像処理の最中でもロボットの手先が動いているため、視覚センサの移動先の目標位置を生成するのに情報が不足している。一方、作業対象が次々に流れてくる生産現場においては、車体鋼板の形状の多くは滑らかな連続曲線であることが事前に分かっており、これらの軌道を走るとき必要なロボットの速度など情報も決まっている。したがって本章では、これらの事前に分かっている情報を用いてニューラルネットワークとファジィ推論則を前もって構築する手法を提案した。作業中にすでに検出された視覚情報から視覚センサの未来位置を提案したファジィ推論によって予測できるため、視覚センサはシーリング軌道の曲がる部分においてもノズルと共に軌道から外れることなく追従作業を実現できることを確認した。

第7章では、本研究で得られた結論を記述した。

---

## 第2章

# 変形を伴うペグインホール

---

### 2.1 概説

これまでのロボットのほとんどは、産業用ロボットのように人間から隔離された環境の下で用いられてきた。しかしながら、これからのロボットの作業環境は、福祉、介助、介護など人間の日常生活を支援することに変化してきた。作業環境が変わるとともに、作業対象は金属などの剛体から布や紙などの柔軟物体に変化してきた<sup>(1)</sup>。柔軟かく変形しやすい作業対象をロボットによって操作する場合、ロボットが作業対象を破壊し、人間に危害を加えてしまう可能性があるため、パワーとスピードよりロボットの安全性および環境に対する適応能力がロボットの制御に求められている。ロボットがあらゆる環境において作業を行うためには、必要な環境情報を外界センサによって獲得し、自律的に作業を遂行することが望ましい。

ロボット工学の分野では、管状や線状の柔軟物体の操作<sup>(2)~(5)</sup>、柔軟曲面での扱い<sup>(6)</sup>、布のマニピュレーション<sup>(7)</sup>、柔軟物体のモデリング<sup>(5),(8)</sup>、画像情報を用いた柔軟物体の操作<sup>(3),(9)</sup>などの研究をはじめ、柔軟物体を扱うロボットのフレキシブル制御法に関する研究が盛んに行われている。しかし、柔軟物体の力学的性質は多様であり、剛体を対象とする作業に用いたロボットの制御法をそのまま柔軟物体の操作に適用することは困難である。また、作業が複雑になる場合、視覚情報と力覚情報のうちどちら一方のみを用いて作業を遂行することは困難と考えられる。

一般に、ロボットの手先に取り付けられる力覚センサを用いてロボットの手先と対象物体との接触力を検出することはできる。しかしながら、接触によって生じる柔軟物体の変形を力覚センサによって検出することができないため、これまで剛体対象を扱う作業に開発したロボットの制御法を柔軟物体の操作にそのまま適用して作業を遂行できないことは多い。人間は柔軟物体を扱うとき、手先で検出した接触力と目で検出した視覚情報を脳で融合し、腕と手の目標位置を生成したのち速やかに行動を行う。したがって、ロボットによる柔軟物体を操作する場合、力覚情報だけではなく視覚情報を積極的にロボットの制御に利用することが大切であると考えられる。



視覚情報を用いる場合、情報量は膨大であるため処理時間は長い。これまでに視覚情報を用いたシーリングロボットの制御<sup>(10)~(13)</sup>が報告された。また、立花・藤堂<sup>(14),(15)</sup>は、視覚情報を用いたDDロボットの目標経路の追従も問題について研究を行った。しかしながら、これらの研究では、画像の取り込みと画像処理を並列に行うことではなく、画像を取り込んでから処理するというループを繰り返す手法を用いた。その結果、画像情報を処理している最中カメラから連続的に出力される画像を取り込むことができないため、画像情報の一部をしか利用しなかった。

そこで本章では、まず、視覚情報の高速処理を実現するため、画像情報を並列に処理する手法を開発する。ついで、視覚と力覚の情報を用いて変形物体を扱うロボットを制御するという新しい手法を提案する。

## 2.2 作業システムと作業の設定

“ペグインホール”は、組立作業での力覚現象を表す典型的な作業であるため、多くの研究に用いられてきた。これまでの研究に用いた作業対象のほとんどは剛体のペグとホールであった。一方、生産現場、日常生活において変形する物体に作業対象、ツールなど棒状物体をはめる作業は多い。例えば、宇宙開発では、ロボットを用いて人工衛星に使用されている板状太陽電池の定期保守や故障の修理などの作業を行う場合、接触力によって太陽電池板の変形が生じる。太陽電池板の変形を許容範囲内に抑えるため、太陽電池板表面の垂直方向に力を制御することがロボットに要求される。

本章では、このような応用を想定し、“変形を伴うはめあい作業”を制御手法の検証実験として取り上げる。本章の実験に用いる作業システムの構成を図 2.1 に示す。ここでの作業は、ロボットの手先に取り付けられているアルミ製ペグ（直径  $\Phi$  19.76 mm）を片持ちアルミ板の上に置かれているアクリル製ホール（直径  $\Phi$  20.02 mm、深さ 40mm）に挿入することである。ペグとホールのクリアランスは約 0.13mm であり、挿入の開始位置を探しやすくするため、ペグの先を丸くした。また、ホール中心軸 <sup>$h_z$</sup> はロボットのベース作業系の <sup>$b_z$</sup> 軸と約 45° の方向に置かれている。

図 2.1 に示すように、はめあい作業中に生じる接触力の情報を検出するため、力覚センサをロボットの手先に取り付ける。また、図 2.2 に示すように、ペグをホールに挿入するとき、押し付け力により生じるアルミ板の変形のため、従来の力覚センサの情報のみを用いた仮想コンプライアンス制御法による作業の遂行は困難である。

ペグインホール作業に関する研究<sup>(16)~(18)</sup>では、対象物体の面取りをしないなど厳しい条件のもとで、作業をいかに遂行することが多く議論されてきた。本章では、これまで多く議論されてきたペグインホール作業の制御法とは異なり、“はめあい作業”

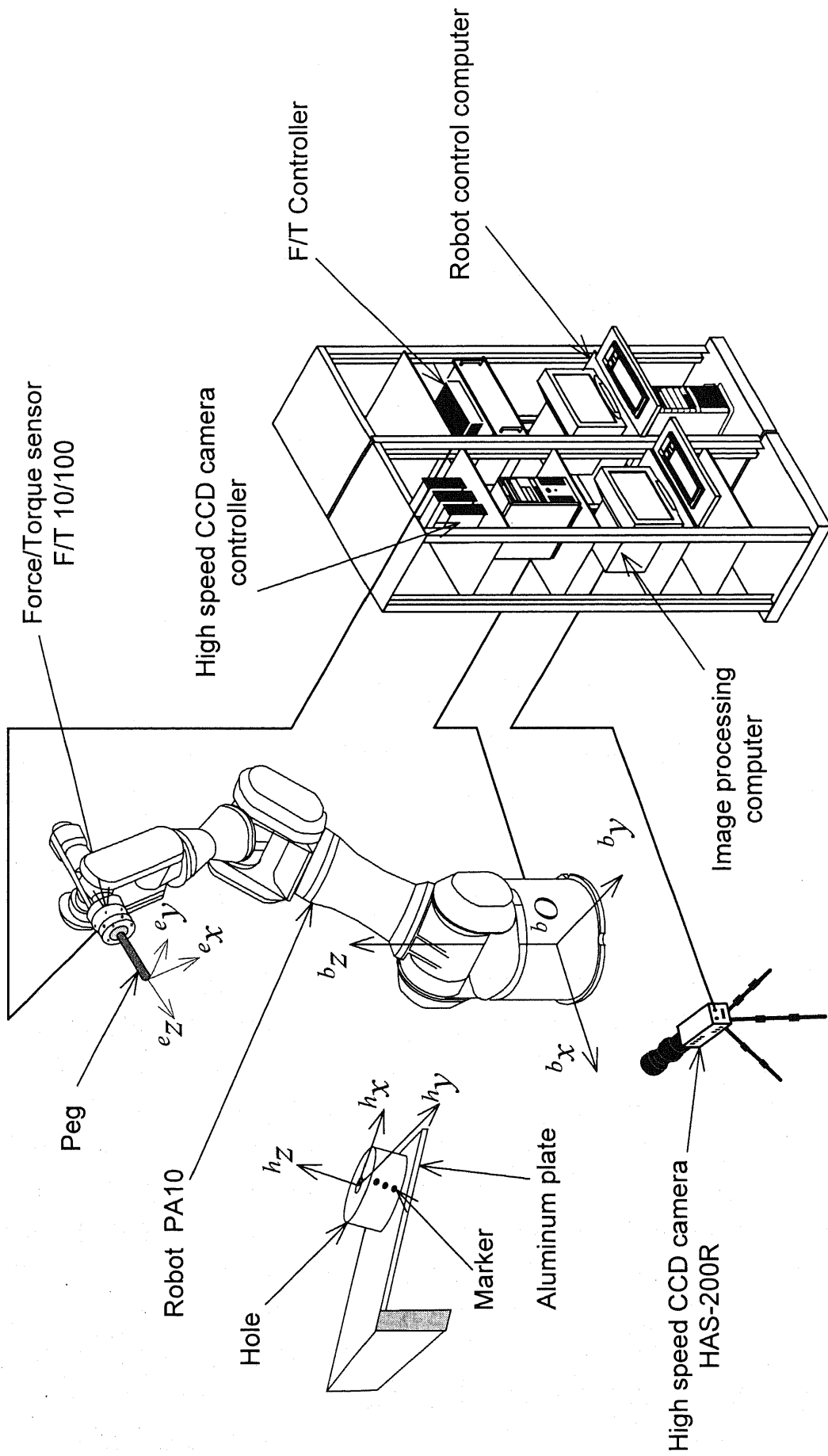


Fig. 2.1 A robot control system of completing peg-in-hole by using a F/T sensor and a visual sensor

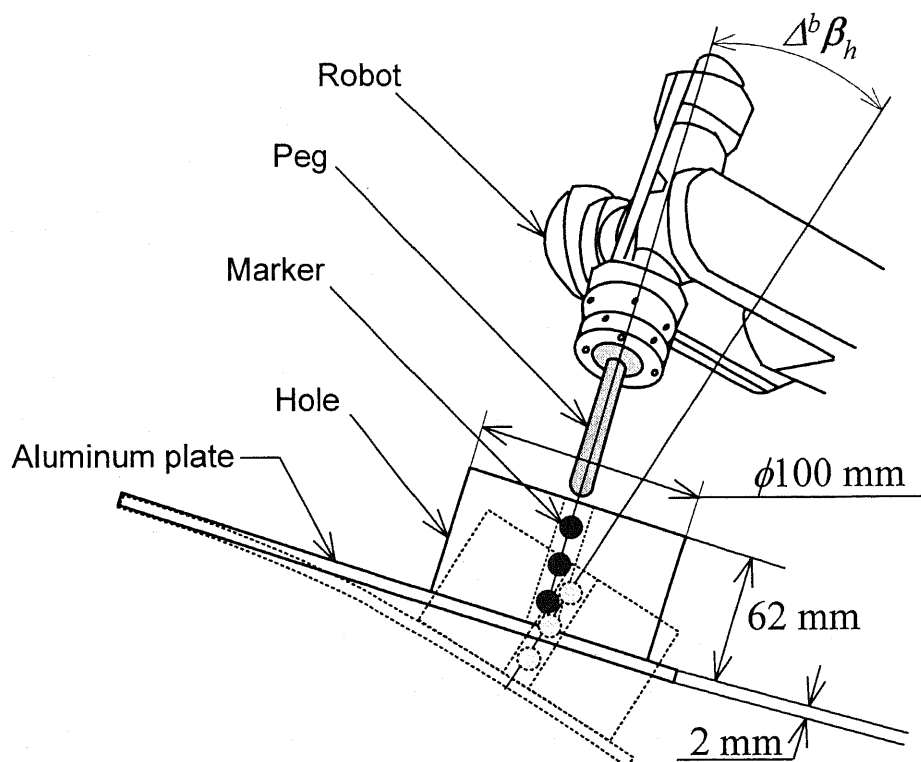


Fig. 2·2 Deformation generated in peg-in-hole task

の中に接触力によってホールの固定板（アルミ板）が変形する場合を取り上げ、視覚情報と力覚情報を融合することによって作業を実現するための制御法を検討する。

### 2·3 画像処理による変形情報の検出

図 2·1 に示すはめあい作業では、接触力により生じたホールの中心軸のずれを画像処理によって検出するため、本論文では毎秒 200 フレームの撮影能力をもつ小型高速 CCD カメラ (HAS-200R, 菱光社) が対象物体の横方向の床に置かれている。画像側のパーソナルコンピュータ (XPS B733r, デルコンピュータ) には画像入力ボード (GENESIS GEN/F/64/8 /STD, Matrox 社) が取り付けられている。画像ボード GENESIS 上に DSP (Digital signal processor) C80 が装着されているため、画像を取り込みながらボード上で画像のリアルタイム処理が可能となり、画像処理の高速化を実現できる。ただし、「画像のリアルタイム処理」とは画像のフレーム時間内に画像を入力し処理することをいう。本論文では以下、これと同じ意味で「リアルタイム」という言葉を用いる。また、画像処理用のパーソナルコンピュータとロボット制御用のパーソナルコンピュータの間でデジタル入出力ボード (PIO-48D(PCI), コンテック) を介してデータの転送と通信を行う。

### 2・3・1 画像情報の並列処理

これまでの研究に、画像入力してから処理し、処理してから入力という手順の繰り返すシリアル手法を用いた。このシリアル処理の流れを図 2・3(a)に示す。しかしながら、本章の実験に用いる CCD カメラ HAS-200R は、5 ms の時間間隔ごとに映像を高速で出力しているため、図 2・3 (a)に示すこれまでの手法では、連続的に出力される映像の同期信号にタイミングをうまく合わせることは困難であり、それによって映像を見落としてしまう可能性がある。

画像処理の並列化を実現するため、本研究において高速処理可能な画像入力・処理ボード GENESIS を用いる。この画像処理ボードに DSP を装着しており、ボード上で取り込んだ画像データを処理することができるため、膨大な画像データをパーソナルコンピュータに転送する必要はない。また、画像入力・処理ボード GENESIS はダブルバッファリング (Double buffering) という処理機能を有するため、画像の入力と処理を並行に行うことは可能となる。本章では、この機能を用いて画像の入力と処理の並列化を実現する方法を開発した。並列処理の流れを図 2・3(b)に示す。

図 2・3(b)に示すように、まず、画像データを格納するためのメモリバッファ  $ImgBuf1$ ,  $ImgBuf2$  の領域を確保する。画像の取り込みと処理が、確保されたデータバッファ  $ImgBuf1$ ,  $ImgBuf2$  で同時に行われるため、連続的に出力されている映像を見落とすことがなくスムーズに処理が行われる。図 2・3 (b)の方法を用いて画像を高速で入力しながらリアルタイム処理するプログラムを作成し、画像情報をリアルタイムに処理することを実現した。

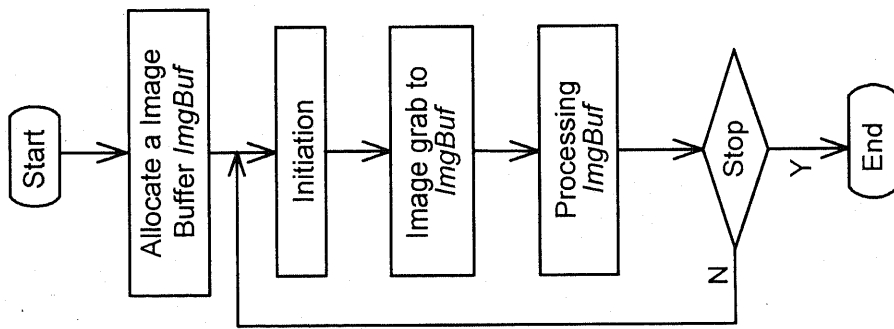
### 2・3・2 画像処理によるホール中心軸の角度の検出

図 2・2 に示すように、作業中に接触力により生じるホールの中心軸のずれ  $\Delta^b \beta_h$  を検出するため、ホールの側面に三つの黒いマーカーを付けた。時刻  $t=kT$  において、画像の 2 値化を行い、マーカーの重心位置  ${}^v s_i(k) = [{}^v X_i, {}^v Z_i]^T$  ( $i=1, 2, 3$ ) を算出したのち、

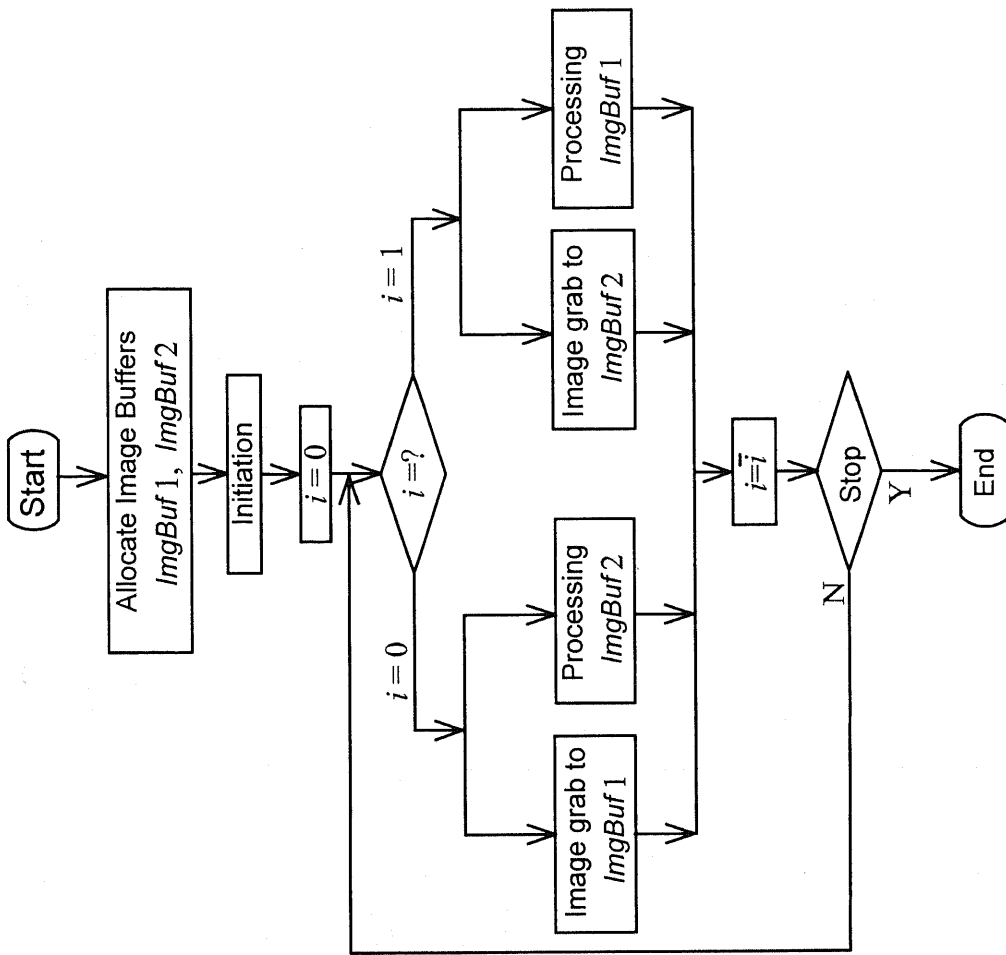
$${}^b s_i(k) = R_v \cdot {}^v s_i(k) + b_v \quad (i=1, 2, 3) \dots \dots \dots (2 \cdot 1)$$

により、これをベース座標系の座標  ${}^b s_i(k) = [{}^b x_i, {}^b y_i, {}^b z_i]^T$  ( $i=1, 2, 3$ ) に変換する。ただし、 $R_v$  はベース座標系で記述されたカメラの姿勢行列であり、 $b_v$  はベース座標系で記述されたカメラの位置ベクトルである。ベース座標系におけるカメラの位置・姿勢を表すパラメータ  $R_v$ ,  $b_v$  をキャリブレーション (付録 B) によって求める。

また、本章での作業の場合、ホールを固定するアルミ製板の変形が  ${}^b y$  軸回りしか生じないため、ロボットベース座標系におけるホールの中心軸の回転量はベクトル  ${}^b \beta_h(k) = [0, \Delta^b \beta_{hy}(k), 0]^T$  で表される。ただし、回転角度  ${}^b \beta_{hy}(k)$  は



(a) Flow chart of serial image processing



(b) Flow chart of parallel image processing

Fig. 2.3 Flow charts of image processing

$${}^b\beta_{hy}(k) = \frac{1}{3} \left[ \tan^{-1} \left( \frac{{}^b z_2 - {}^b z_1}{{}^b x_2 - {}^b x_1} \right) + \tan^{-1} \left( \frac{{}^b z_3 - {}^b z_1}{{}^b x_3 - {}^b x_1} \right) + \tan^{-1} \left( \frac{{}^b z_3 - {}^b z_2}{{}^b x_3 - {}^b x_2} \right) \right] \dots (2.2)$$

であり、図 2.2 に示す  $\Delta {}^b\beta_h(k)$  は

$$\Delta {}^b\beta(k) = {}^b\beta(k) - {}^b\beta(k-1) = [0 \quad \Delta {}^b\beta_{hy}(k) \quad 0]^T \dots (2.3)$$

で与えられる。

画像 1 枚の入力時間が 5 ms であることに對し、DSP を用いて図 2.3(b) に示すような画像処理アルゴリズムを作成する。これによって、重心位置を検出するまでの時間を約 4 ms に短縮したため、画像を取り込みしながら処理することができた。

一方、実験に用いるロボットの制御サンプリング周期が 10 ms であるため、PIO ボードを介してロボット制御側のパーソナルコンピュータとの同期をとる必要がある。本章で提案するリアルタイム画像処理と同期転送のタイムチャートを図 2.4 に示す。

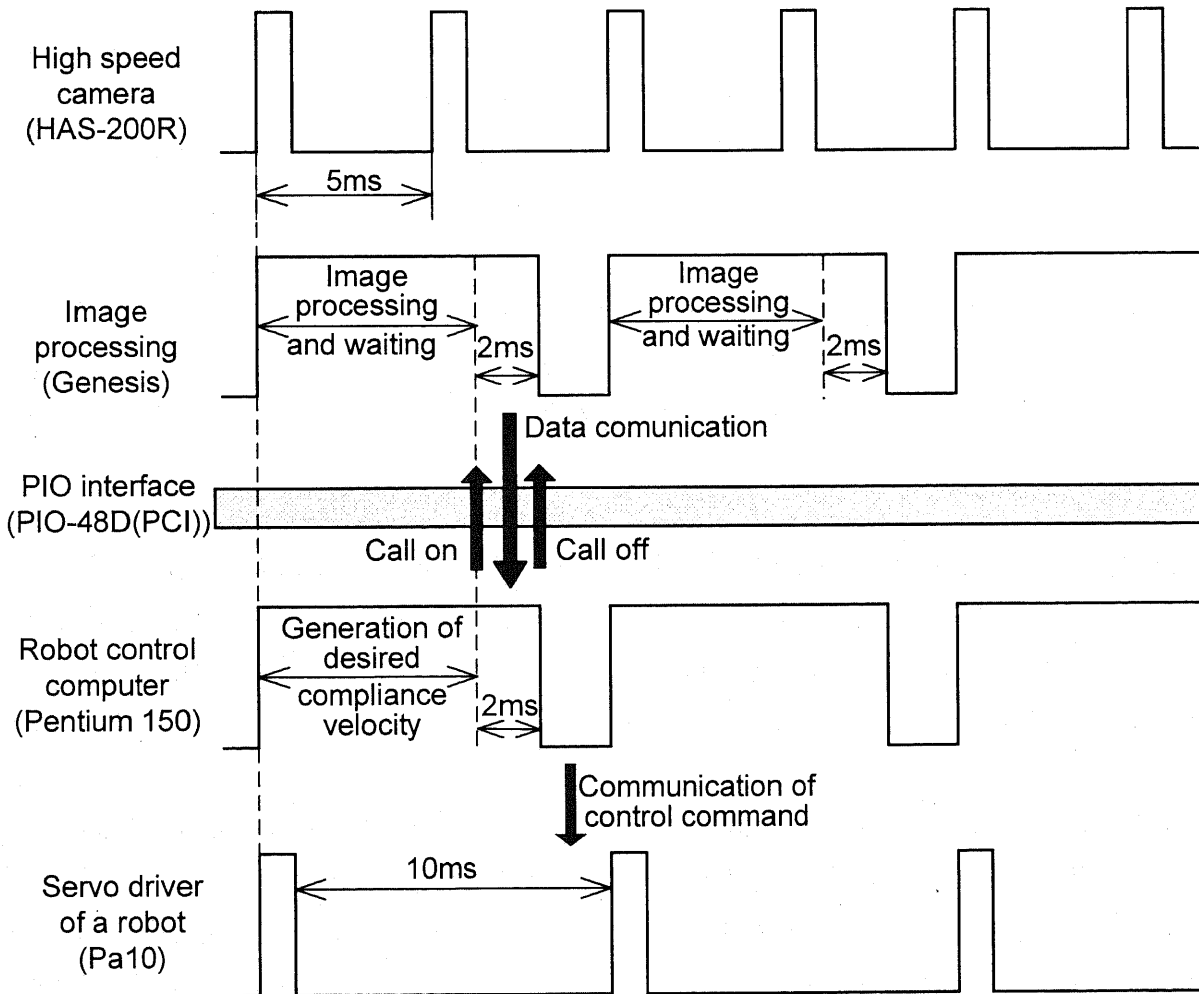


Fig. 2.4 Proposed time chart of robot control using sensor fusion

図 2.4 では、ロボットを制御するパーソナルコンピュータで制御の目標値を生成したのち、PIO インタフェースを介して画像側のパーソナルコンピュータと通信を行い、制御側に画像処理の結果を転送する。

## 2.4 視覚と力覚の情報のセンサフュージョンによるロボットの制御

本章で取り上げる“変形を伴うはめあい作業”では、押し付け力を許容範囲内に抑えるため、対象物体の表面の垂直方向における力の制御が必要となる。組立作業など外部環境との接触がある場合、設計者の要求によってロボットの手先効果器に仮想的な力学的特性を持たせることを容易にするため、仮想コンプライアンス制御法を用いることが多い。一般に、目標力のみを考慮する力制御の場合、仮想コンプライアンス制御則は

$$\dot{p}_r(k+1) = (I - TM_c^{-1} \cdot D_c) \cdot \dot{p}_r(k) + TM_c^{-1} \cdot \Delta f(k) \cdots \cdots \cdots (2.4)$$

で表される。ただし、

- $\dot{p}_r(k+1) \in R^{6 \times 1}$  : 時刻  $t=(k+1)T$  での手先の目標速度
- $\dot{p}_r(k) \in R^{6 \times 1}$  : 時刻  $t=kT$  での手先の速度
- $M_c \in R^{6 \times 6}$  : 仮想質量行列
- $D_c \in R^{6 \times 6}$  : 仮想粘性行列
- $\Delta f(k) \in R^{6 \times 1}$  : 時刻  $t=kT$  における目標の力  $f_d(k)$  と接触力  $f_e(k)$  の差であり、  
 $\Delta f(k)$  は  $f_d(k) - f_e(k)$  によって算出される
- $T$  : ロボットの制御サンプリング周期
- $I \in R^{6 \times 6}$  : 単位行列

“変形を伴うはめあい作業”では、ロボットの手先に取り付けられている力覚センサで作業中に生じる接触力を検出できるが、アルミ板の変形によって生じたホールの中心軸の変化を検出できない。そのため、式(2.4)に示す制御則を用いてロボットの手先の力を拘束しながら“変形を伴うはめあい作業”を遂行することは困難と考えられる。

一般に、人間がこのような作業を行う場合、視覚情報に基づいて対象物体の変位を特定しながら、望ましい手先の姿勢を調整し、作業を遂行する。これを参考にし、作業中接触によって生じたホールの中心軸のずれを補正するには、画像処理の結果から算出したホールの回転角度  $q(t)$  を、式(2.4)に示す従来の仮想コンプライアンス制御則に加え、ロボットの手先をさらに回転させることが必要であると考えた。したがって本章では、画像情報  $q(t)$  と力覚情報  $\Delta f(t)$  を融合した仮想コンプライアンス制御則とい

という新しい手法を提案する。この制御則は

$$M_c \ddot{p}(t) + D_c \dot{p}(t) = \Delta f(t) + K_q q(t) \cdots \cdots \cdots (2.5)$$

で表される。式(2.5)を離散時間系で表すと、

$$\dot{p}_r(k+1) = (I - M_c^{-1} D_c T) \dot{p}(k) + M_c^{-1} T \Delta f(k) + M_c^{-1} K_q T q(k) \cdots \cdots \cdots (2.6)$$

が得られる。 $K_q$  はねじり回転係数行列であり、従来のコンプライアンスによる力制御に画像処理の結果をフィージョンする度合いを表すため、本章では  $K_q$  を融合係数行列と呼ぶ。また、式(2.3)で表す時刻  $t=kT$  でのホールの中心軸の回転角度  ${}^b\beta_{hy}(k)$  と時刻  $t=(k-1)T$  でのホールの中心軸の回転角度  ${}^b\beta_{hy}(k-1)$  の差  $\Delta^b\beta_{hy}(k)$  を

$$\Delta^b\beta_{hy}(k) = {}^b\beta_{hy}(k) - {}^b\beta_{hy}(k-1) \cdots \cdots \cdots (2.7)$$

で表すと、 $q(k)$  は

$$q(k) = [0 \ 0 \ 0 \ 0 \ \Delta^b\beta_{hy}(k) \ 0]^T \cdots \cdots \cdots (2.8)$$

で表される。したがって、 $q(k)$  は、視覚情報の処理によって算出したホールの中心軸のずれに対し、ロボット手先の姿勢を調整するために必要な回転ベクトルである。

式(2.6)では、画像処理で得られたホールの中心軸のずれ  $\Delta^b\beta_{hy}(k)$  を補正するため、式(2.4)で表される従来の仮想コンプライアンス制御則に手先姿勢の回転量  $q(k)$  を加えたため、ロボット手先の姿勢をそのずれに対し適切に調整することができる。そのため、変形を伴う場合でも接触力を拘束しながら“はめあい作業”を遂行することは可能と考える。このアルゴリズムを実現するためのリアルタイム視覚フィードバック制御のブロック線図を図 2.5 に示す。

図 2.5 では、時刻  $t=kT$  において画像処理で得られたマーカーの画像座標系における座標  ${}^v s_i(k) = [{}^v X_i, {}^v Z_i]^T$  ( $i=1, 2, 3$ ) を式(2.1)によりベース座標系の座標  ${}^b s_i(k) = [{}^b x_i, {}^b y_i, {}^b z_i]^T$  ( $i=1, 2, 3$ ) に変換したのち、式(2.2), (2.3)によってホールの中心軸の回転角度ベクトル  ${}^b\beta_h(k)$  を求める。力覚情報に関しては、F/T センサで検出した接触力  $f_{em}(k)$  を座標変換行列  $R_f(k)$  によりロボットベース座標系の力覚情報  $f_e(k)$  に変換する。また、式(2.6)によって提案した制御則を用いてロボットの手先の仮想コンプライアンス速度  $\dot{p}_r(k+1)$  を算出し、逆ヤコビ行列  $J^{-1}$  でロボットの各目標関節角速度  $\dot{\theta}_r(k+1)$  を計算したのち、ロボットを制御する。



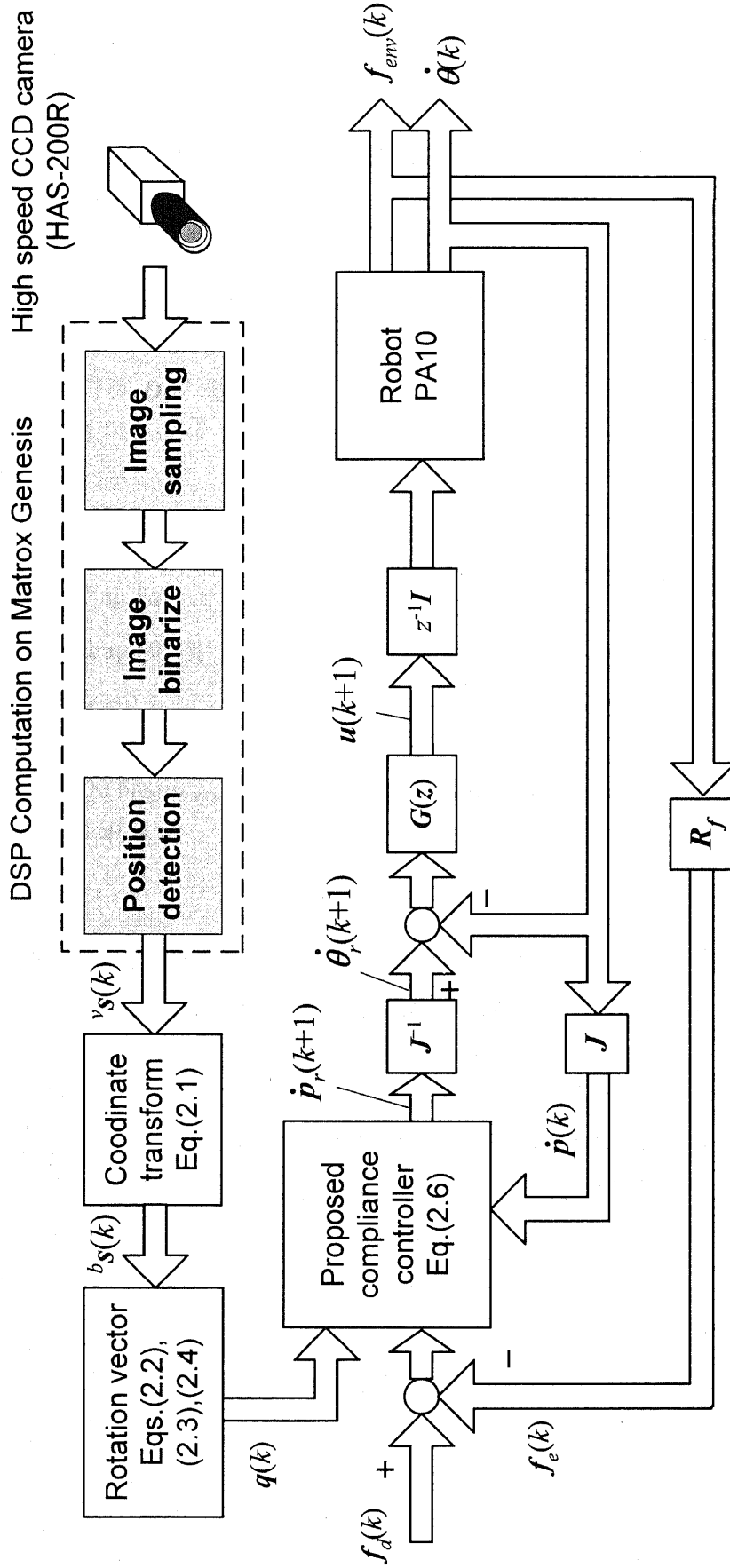


Fig. 2.5 The proposed visual based feedback control block diagram for a robot using sensor fusion

## 2.5 実験とその結果

“はめあい作業”に用いたロボットは7自由度可搬式知能アーム (PA10, 三菱重工) であるが, 第3関節 (S3 関節) を固定し, 6自由度マニピュレータとして実験を行った. 作業中の接触力を検出するため, 力覚センサ F/T (10/100, ビー・エル・オートテック社) を用いた. ロボットの制御サンプリング周期  $T$  は 10ms とする.

また, ロボットの手先が振動的にならないように, 式(2.5)で用いた仮想質量行列  $M_c$  と仮想粘性行列  $D_c$  の値は, 試行錯誤によって選び, 融合係数行列  $K_q$  の値はアルミ板の最大変形量とロボットの手先の追従速度を考慮して決定した. 実験に用いたこれらのパラメータは

$$M_c = \text{diag} [m_c \quad m_c \quad m_c \quad m_r \quad m_r \quad m_r] \cdots \cdots \cdots (2.9)$$

$$D_c = \text{diag} [d_c \quad d_c \quad d_c \quad d_r \quad d_r \quad d_r] \cdots \cdots \cdots (2.10)$$

$$K_q = \text{diag} [0 \quad 0 \quad 0 \quad 0 \quad k_{qy} \quad 0] \cdots \cdots \cdots (2.11)$$

である. ただし,

$$m_c = 20 \text{ kg},$$

$$m_r = 15 \text{ kg}\cdot\text{m}^2$$

$$d_c = 200 \text{ N}\cdot\text{s}\cdot\text{m}^{-1},$$

$$d_r = 150 \text{ N}\cdot\text{m}\cdot\text{s}\cdot\text{rad}^{-1}$$

$$k_{qy} = 150 \text{ N}\cdot\text{m}$$

とする.

### 2.5.1 実験 1: 力覚センサのみを用いたペグインホール(スポンジあり)

図 2.6 に示すように, アルミ板の塑性変形を避けるため, アルミ板の下にブロック状のスポンジを置いた. このスポンジは衝撃を吸収できるため広く用いられる. まず力覚センサのみを使用する場合, 式(2.4)に示す従来の仮想コンプライアンス制御法を用いて実験を行った. 作業中の接触力によって生じたホールの中心軸のずれに対し, ロボットの手先の姿勢をうまく合わせることができないため, ペグはホールの入口にとどまり, 作業をストップした.

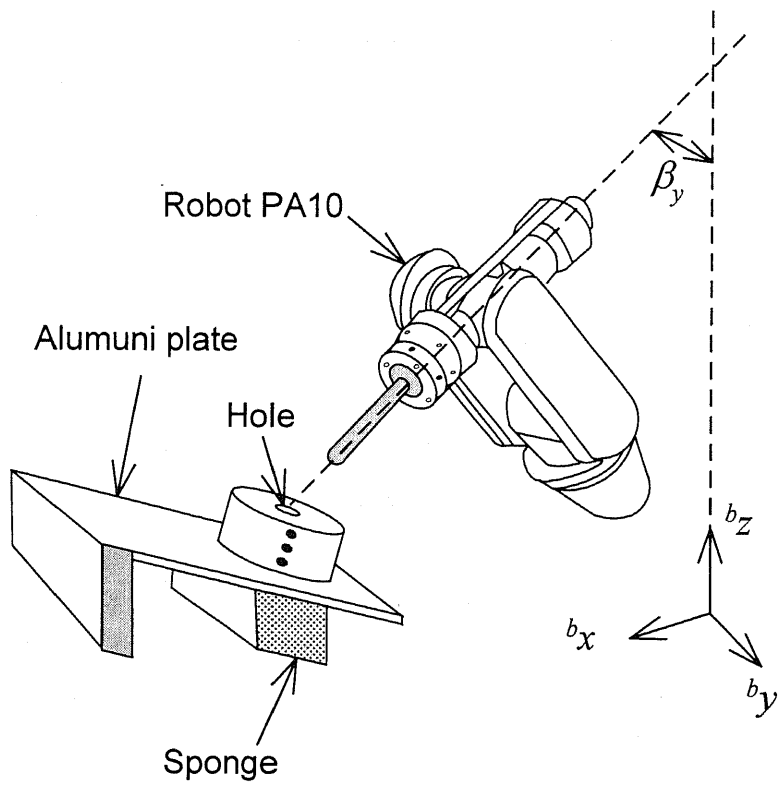


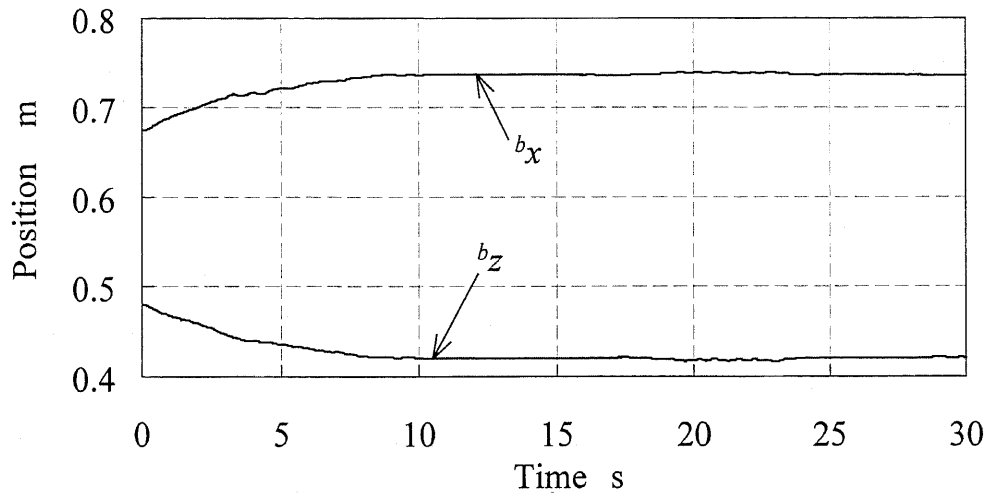
Fig. 2・6 Peg-in-hole experiment with a sponge set under the deform plate

**2・5・2 実験2:視覚センサと力覚センサを用いたペグインホール(スポンジあり)**

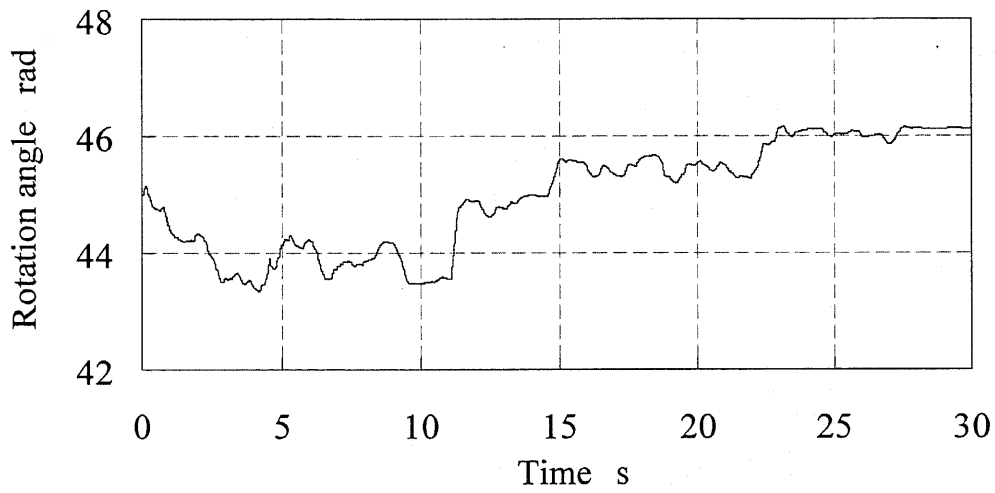
図2・6に示すように、アルミ板の下にブロック状のスポンジを置き、視覚と力覚センサを用いて実験を行った。接触力の目標値  $f_d$  を

$$f_d = [0 \ 0 \ -15 \text{N} \ 0 \ 0 \ 0]^T \dots\dots\dots (2\cdot12)$$

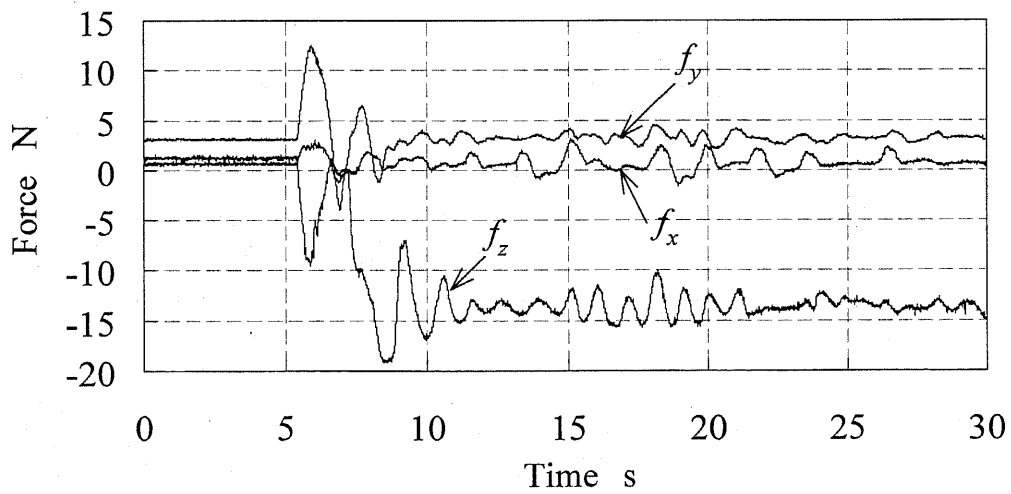
とした。実験の結果を図2・7に示す。実験2では、視覚センサを用いたため、ホールの中心軸のずれに対してロボット手先の姿勢を適切に調整することができた。そのため、接触力を目標値に拘束しながら作業を遂行した。作業を終了するまでの時間は約20sを要した。



(a) Position of the end effector of the robot



(b) Rotation angle  $\beta_y$



(c) Force detected at the robot hand

Fig. 2·7 Results of positions and forces obtained in the 2nd experiment

### 2.5.3 実験3: 視覚センサと力覚センサを用いたペグインホール(スポンジなし)

図2.8に示すように, 実験2でアルミ板の下に置かれていたスポンジを外して, はめあい作業を行った. このとき, 実験2に示したスポンジの反力がないため, 作業中の接触力によって生じるホールの中心軸のずれが大きくなり, 作業は一層困難になると予想される. この場合, 力の目標値  $f_d$  を

$$f_d = [0 \ 0 \ -10\text{N} \ 0 \ 0 \ 0]^T \dots\dots\dots (2.13)$$

とした. 実験の結果を図2.9に示す. 図2.9(c)に作業開始後4 s前後でペグとホールとの接触が始まり, 20 sまでペグの大半がホールに入ったことを示しており, 20 s前後から手先の姿勢をさらに調整し作業を最後に遂行したことを表す.

実験2と比べ, アルミ板が柔らかいため, 図2.9(b)に示すように視覚情報の処理によってロボット手先の姿勢  $\beta_y$  の調整量は大きかった. また, スポンジを用いた反力がある実験2の結果から, 図2.9(c)に示すホールの中心軸の変化に対して力の素早い調整が見られた. これは, 対象が柔らかいため, 人間と同じように, ロボットの手先の姿勢を変形に合わせて頻繁に調整することによるものと考えられる. 実験3では, 姿勢の調整により長い時間がかかったため, 作業の遂行は約30 sを要した.

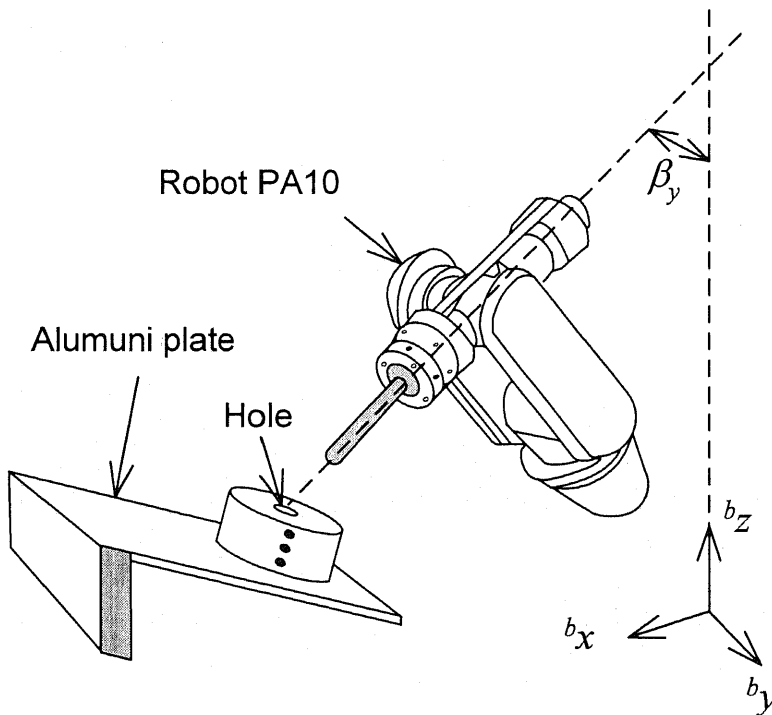
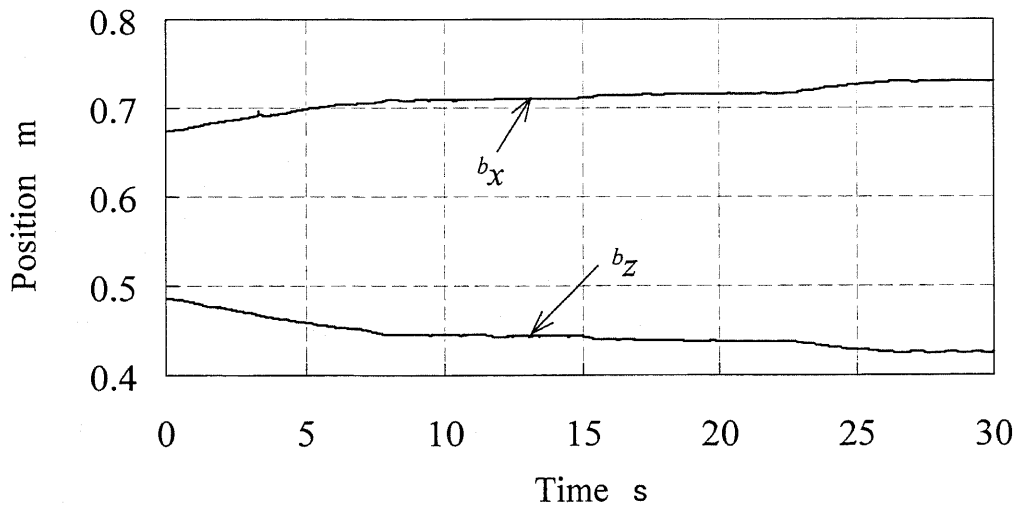
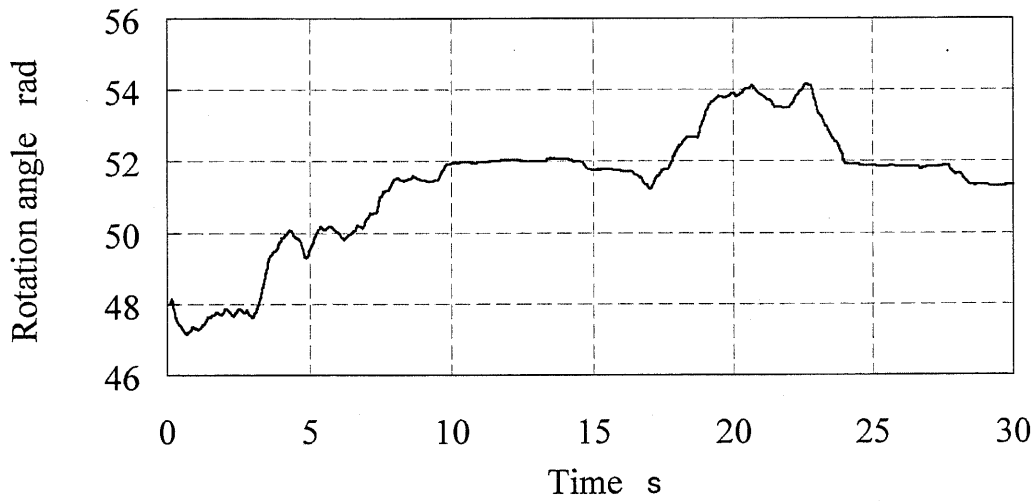


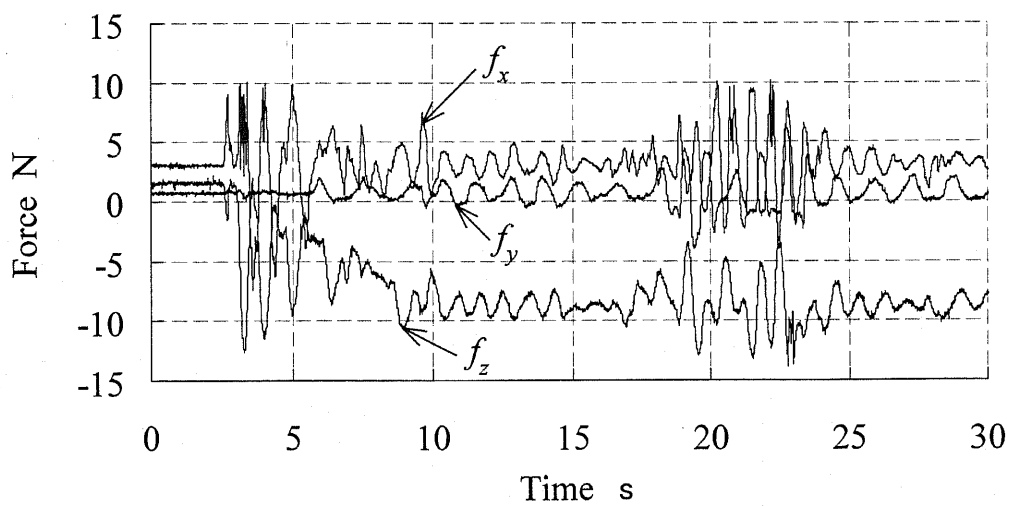
Fig. 2.8 Peg-in-hole experiment without any support



(a) Position of the end effector of the robot



(b) Rotation angle  $\beta_y$



(c) Force detected at the robot hand

Fig. 2·9 Results of positions and forces obtained in the 3rd experiment

## 2.6 要約

本章では、得られた主な結果は次のとおりである。

(1)変位物体の操作において、視覚と力覚情報のセンサフュージョンによるロボットの制御アルゴリズムを提案し、実験により本方法の有効性を確かめた。

(2)画像を高速で処理するため、DSP を用いた画像情報をリアルタイムで処理するアルゴリズムを提案し、実験によりその有効性を確かめた。

(3)リアルタイム視覚フードバック制御アルゴリズムを提案し、実験によりその有効性を確かめた。

---

## 第3章

# ロボットによる棒状変形物体の操作

---

### 3.1 概説

第2章では、視覚と力覚の情報を用いて2次元の変形を伴う“はめあい作業”をロボットに遂行させる手法を提案した。第3章では、3次元空間での変形を伴う棒状物体の“はめあい作業”を取り上げ、第2章で提案した手法をさらに発展することによって棒状物体の“はめあい作業”をロボットに遂行させる手法を検討する。

柔軟物体のなかでは、コード、ケーブル、糸、ロープ、チューブなど細長い物体（以下、棒状物体と呼ぶ）が生産現場、生活環境、医療福祉など幅広い分野で多く扱われている。棒状物体の材質によって力学的特性が大きく異なるため、ロボットによる棒状物体の操作は困難である<sup>(1)~(5)</sup>。特に、小さい穴に棒状物体を挿入する場合、挿入力によって棒状物体に弾性座屈が生じるため、この作業を遂行する方法を導き出す必要がある。

棒状柔軟物体を操作する研究において、ポテンシャルエネルギーが最小となるように柔軟物体のモデリングを行うという手法が提案された<sup>(5)</sup>。また、力覚情報を利用できない場合、画像情報のみを用いて棒状柔軟物体の先端に掛かる力を推定しながらロボットによる棒状柔軟物体の挿入作業を遂行するという手法<sup>(2),(4)</sup>も報告された。

これらの研究に対し、管状柔軟物体の挿入作業を人間に行わせ、作業を行ったときの人間のスキルをマニピュレータに移植するという手法が提案され、それを用いて管状物体の挿入作業をマニピュレータに遂行させたことが報告された<sup>(3)</sup>。しかしながら、棒状物体の変形をモデリングすることによって棒状物体の先端に掛っている力を求める場合、曲げ剛性など棒状物体の材質に関するパラメータの値が必要である。また、開発された従来の方法は、決められた対象の作業に限定しているため、異なる対象での作業に適用することは困難である。

一方、このような作業を行うとき、人間は目で得られた視覚情報と手で感じる力覚情報を有効に融合することによって弾性座屈を巧みに抑える方法を取りながら、作業を遂行する。そこで本章では、視覚と力覚の情報をニューラルネットワークの学習に



よってフュージョンさせ、その結果を用いてロボットを制御するという新しい手法を提案する。また、提案した手法を用いて棒状柔軟物体のはめ合い作業をロボットに行わせ、その効果を確かめる。

### 3.2 作業システムの構成と作業の設定

図3.1に示すように、本章の実験に用いるシステムは、可搬式汎用知能アーム (PA10, 三菱重工) 1台、力覚センサ (F/T 10/100, ビー・エル・オートテック) 1台、小型 CCD カメラ (CV-M40, JAI 社) 2台とパーソナルコンピュータ (Dimension XPS B733r, DELL 社) 1台から構成されている。

図3.1に示すロボットアームは7自由度を有するが、本章で取り上げる作業においてアームの冗長自由度を固定し、6自由度ロボットとして用いる。作業中に生じる接触力の情報を検出するため、力覚センサをロボットの手先に装着する。さらに、力覚センサを介して手先に内径8mmのホールを持つ円筒を取り付ける。

${}^b_y$ 軸に沿ってロボットの両側約0.3mの場所に2台のCCDカメラを取り付けることによって立体視画像観測システムを構成する。この画像システムを用いて三次元空間における棒状物体の特徴点の位置を検出し、その結果をロボットの制御に用いる。

また、図3.1に示すように、 ${}^b_x$ 軸方向に約0.85mの場所に作業台が置かれ、棒状物体が作業台に固定されている。本章の実験で用いる棒状物体は、長さ350mm、直径3mmのものとする。棒状物体の先端に直径8mm、長さ80mmのアルミ製ペグを取り付けており、ペグとホールとのクリアランスは約0.25mmである。作業は、ロボットの手先に取り付けているホールに棒状物体をはめることである。作業中に接触によって棒状物体の変形が生じるため、力覚情報のみによって作業を遂行することは困難である。

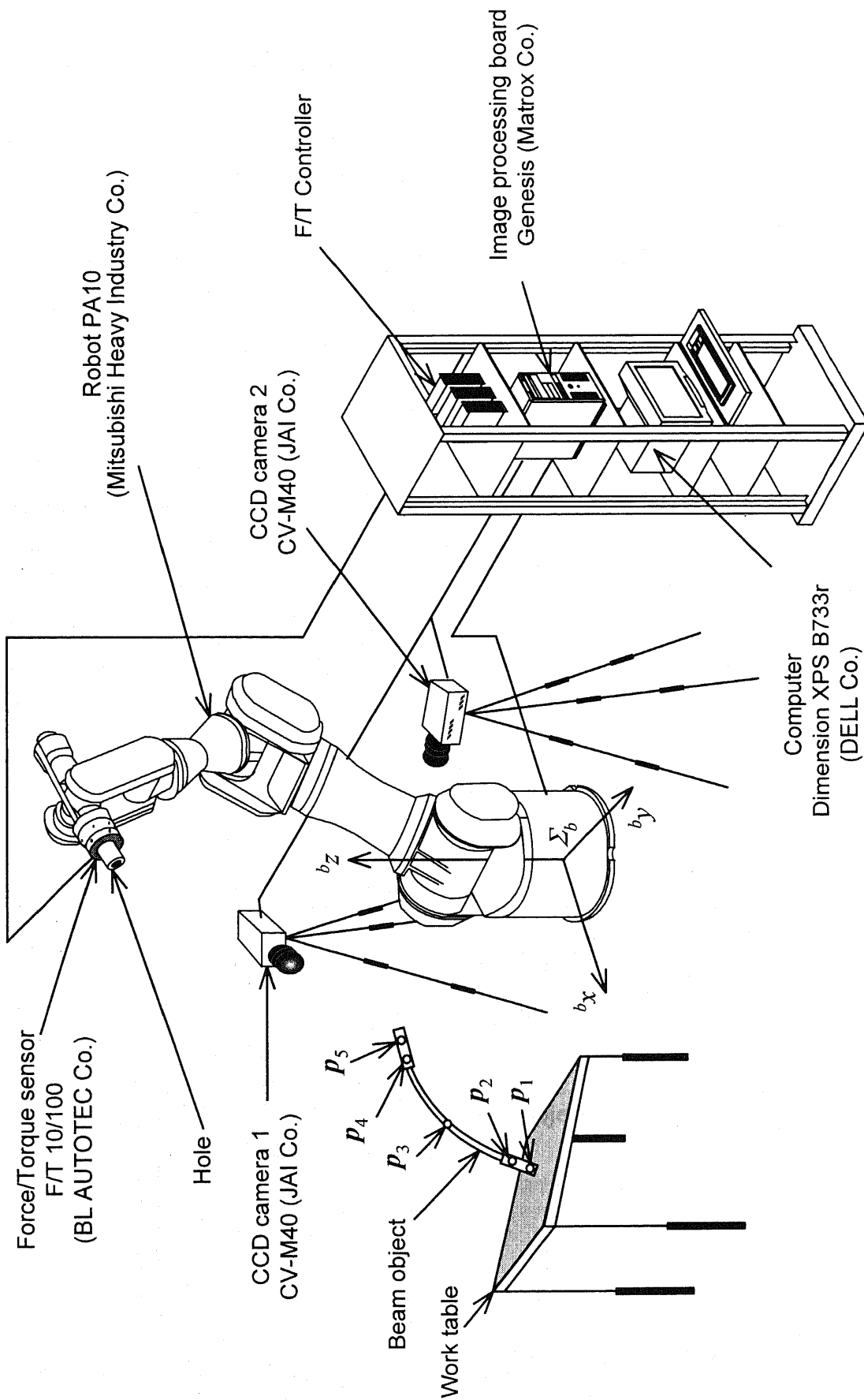


Fig. 3•1 Setup of a robot control system for inserting of a beam object into a hole

### 3-3 画像処理による目標値の生成

図 3-1 に示すように、2 台の小型 CCD カメラによって立体視画像システムを構成する。このカメラは 1 秒ごとに 60 フレームの映像を出力しているため、カメラのフレーム時間は約 16.67ms である。2 台のカメラからの映像信号を画像入力・処理ボード (GENESIS GEN/F/64/8 /STD, Matrox 社) によって入力し処理を行う。本章では、第 2 章の図 2-3(b) に示す画像の並列処理という手法を用いるため、カメラから連続的に出力されている画像を見落とすことなくリアルタイムに処理することができる。したがって、画像の入力・処理のサンプリング周期  $T_c$  は 16.67 ms である。

#### 3-3-1 画像処理による特徴点の算出

棒状物体の特徴点の配置を図 3-2 に示す。棒状物体の先端にアルミ製ペグを装着し、末端にアルミ製シリンダを取り付け、さらに、それを作業台に固定する。また、特徴点を画像処理によって検出しやすくなるため、マーカーを棒状物体の先端部に 2 個、末端部に 2 個、中間部に 1 個貼りつける。図 3-2 に示すように、末端にある 2 個のマーカーのそれぞれの位置を  $p_1, p_2$ 、先端にある 2 個のマーカーのそれぞれの

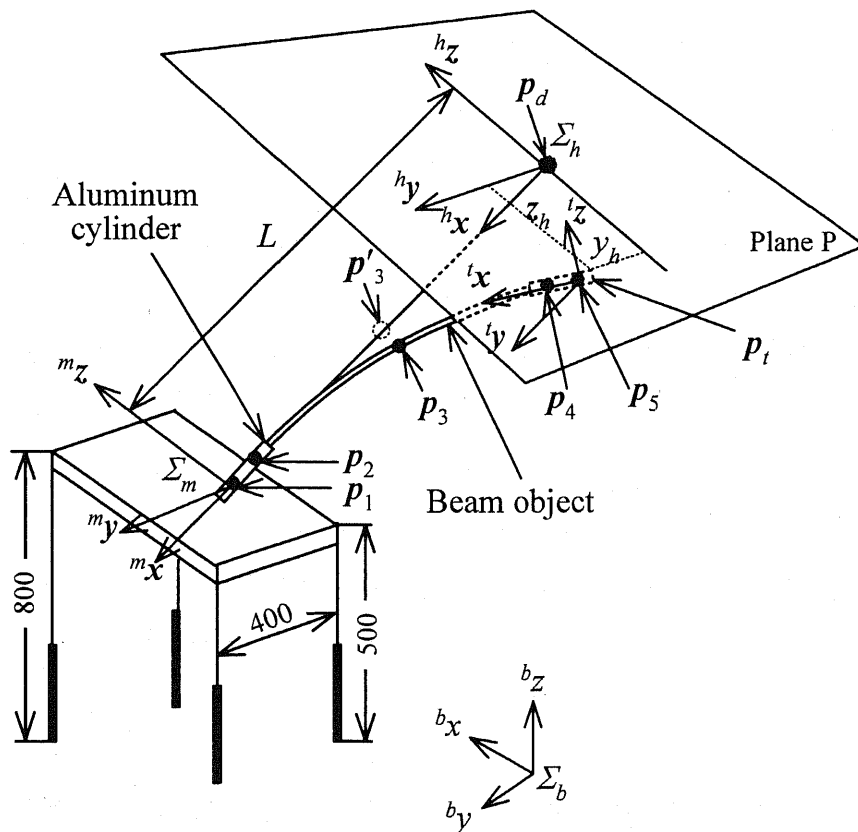


Fig. 3-2 Setup of a beam object and feature points

位置を  $p_4$ ,  $p_5$ , 中間部のマーカーの位置を  $p_3$  とする. ただし,  $p_3$  は弾性座屈の観察に用いられる. 画像処理を用いてこれらの特徴点を実時間で検出し, ベース座標系  $\Sigma_b$  における 3次元座標を算出する. ここで, 得られた特徴点の 3次元位置を  $p_i = [x_i, y_i, z_i]^T$  ( $i=1,2,\dots,5$ ) とする.

図 3.2 に示すように, 特徴点  $p_1$  を座標系  $\Sigma_m$  の原点とし, 棒状変形物体が真っすぐ伸びている状態での棒の先端位置  $p_d$  を座標系  $\Sigma_h$  の原点とする. 画像処理で得られた特徴点の位置によって座標系  $\Sigma_m$ ,  $\Sigma_h$  の方向ベクトルを計算できる. ここで,  $p_1$  と  $p_2$  を結ぶベクトルを  $p_{12}$  とし,  $p_4$  と  $p_5$  を結ぶベクトルを  $p_{45}$  とする. このとき, 方向ベクトル  ${}^m x$ ,  ${}^i x$  は,

$${}^m x = [x_m \ y_m \ z_m]^T = \frac{p_{12}}{\|p_{12}\|} \dots \dots \dots (3.1)$$

$${}^i x = [x_i \ y_i \ z_i]^T = \frac{p_{45}}{\|p_{45}\|} \dots \dots \dots (3.2)$$

で表される. ただし, ベクトル  $p \in R^{3 \times 1}$  のノルム  $\|p\|$  を  $\sqrt{p^T \cdot p}$  で定義する. また, ベース座標系  $\Sigma_b$  の  ${}^b x$ - ${}^b y$  平面へ  ${}^m x$ ,  ${}^i x$  のそれぞれの投影ベクトルを  ${}^m x'$ ,  ${}^i x'$  とする.

ここで,  ${}^b y$  軸の方向に向き, かつ  ${}^b x$ - ${}^b y$  の平面内で  ${}^m x'$  と直交するベクトル  ${}^m y'$ ,  ${}^i x'$  と直交するベクトル  ${}^i y'$  をとすれば, 方向ベクトル  ${}^m y$ ,  ${}^i y$  は

$${}^m y = \frac{{}^m y'}{\|{}^m y'\|} \dots \dots \dots (3.3)$$

$${}^i y = \frac{{}^i y'}{\|{}^i y'\|} \dots \dots \dots (3.4)$$

から求められる. また, 式(3.1)~(3.4)から  ${}^m x$  と  ${}^m y$  の外積  ${}^m z$ ,  ${}^i x$  と  ${}^i y$  外積  ${}^i z$  を算出できる.

棒状物体の長さを  $L$  とすれば, それが真っすぐ伸びているときの棒状物体の先端位置  $p'_d \in R^{3 \times 1}$ , 姿勢  $R_d \in R^{3 \times 3}$  は

$$p'_d = [x_d \ y_d \ z_d]^T = p_1 - L \cdot {}^m x \dots \dots \dots (3.5)$$

$$R_d = [{}^m x \quad {}^m y \quad {}^m z]^T \dots \dots \dots (3.6)$$

で表される. また,  $x$ - $y$ - $z$  固定角法により  ${}^b x$ ,  ${}^b y$ ,  ${}^b z$  軸回りの回転角  $\alpha_d$ ,  $\beta_d$ ,  $\gamma_d$  を算出できるため, 目標の位置・姿勢ベクトル  $p_d \in R^{6 \times 1}$  は

$$p_d = [x_d \ y_d \ z_d \ \alpha_d \ \beta_d \ \gamma_d]^T \dots \dots \dots (3.7)$$

により生成される.

同様に, 特徴点  $p_5$  から棒の先端までの距離  $l_i$  とすれば, 棒先端の位置  $p'_i \in R^{3 \times 1}$ , 姿勢  $R_i \in R^{3 \times 3}$  は

$$\mathbf{p}'_i = [x_i \quad y_i \quad z_i]^T = \mathbf{p}_s - l_i \cdot \mathbf{x} \quad \dots \quad (3 \cdot 8)$$

$$\mathbf{R}_i = [\mathbf{x} \quad \mathbf{y} \quad \mathbf{z}]^T \quad \dots \quad (3 \cdot 9)$$

で表される。また、得られた方向ベクトル $\mathbf{x}, \mathbf{y}, \mathbf{z}$ によってベース座標系 $\Sigma_b$ における棒先端の位置・姿勢ベクトルを $\mathbf{p}_i \in R^{6 \times 1}$ とすれば、 $\mathbf{p}_i$ を

$$\mathbf{p}_i = [x_i \quad y_i \quad z_i \quad \alpha_i \quad \beta_i \quad \gamma_i]^T \quad \dots \quad (3 \cdot 10)$$

から求める。

一方、座標系 $\Sigma_h$ における棒の先端位置・姿勢ベクトルを ${}^h\mathbf{p}_i \in R^{6 \times 1}$ とすれば、 $\mathbf{p}_i$ は

$$\mathbf{p}_i = \mathbf{p}_d + {}^b\mathbf{R}_h \cdot {}^h\mathbf{p}_i \quad \dots \quad (3 \cdot 11)$$

で表される。ただし、 ${}^b\mathbf{R}_h$ は座標系 $\Sigma_h$ から座標系 $\Sigma_b$ への変換行列である。式(3.11)

により座標系 $\Sigma_h$ における棒の先端位置・姿勢ベクトル ${}^h\mathbf{p}_i$ は

$${}^h\mathbf{p}_i = [x_h \quad y_h \quad z_h \quad \alpha_h \quad \beta_h \quad \gamma_h]^T = {}^b\mathbf{R}_h^{-1} \cdot (\mathbf{p}_i - \mathbf{p}_d) \quad \dots \quad (3 \cdot 12)$$

から算出できる。

### 3.3.2 画像処理によって検出された特徴点位置の補間

実験に用いるロボットの制御サンプリング周期を $T (=5 \text{ ms})$ とする。第2章で提案した画像の並列処理法を用いて、画像のリアルタイム処理を実現したとはいえ、画像入力・処理のサンプリング周期 $T_c (=16.67 \text{ ms})$ は $T$ より長い。このため、画像処理によって得られた目標位置・姿勢をロボットの制御サンプリング周期ごとに補間する必要がある。

これまでの著者らの研究<sup>(12)~(14)</sup>では、画像処理の負担を軽減するため、画像処理とロボット制御のためのコンピュータを別々に用意した。これらの研究では、画像の処理時間はカメラのフレーム時間 (NTSC カメラの場合 33ms) より長く、ロボットの制御サンプリング周期 $T$ の十倍以上になるのがほとんどであった。また、画像の内容によって処理時間が一定ではないため、視覚フィードバック制御系の設計は煩雑になる。制御アルゴリズムの作成便宜を図るため、これらの研究において $T_c$ を $T$ の整数倍に選んでいた。

図3.1に示すように、本章にて画像処理とロボット制御を1台のパーソナルコンピュータで行う。このため、図2.3(b)に示す画像のリアルタイム処理を実現するには、マルチスレッドの技術を用いてWindows NT4.0 (Microsoft 社)上で動作する制御ソフトウェアを開発した。これにより、図3.3に示すようにロボットの制御と画像処理を別々のスレッドで処理し、画像情報を最大限に活用した。

時刻 $t=kT-T_c$ にて入力された画像の処理を時刻 $t=kT$ において終了し、得られた特徴点の位置 $\mathbf{p}_j(kT-T_c) (j=1,2,\dots,5)$ に対し、

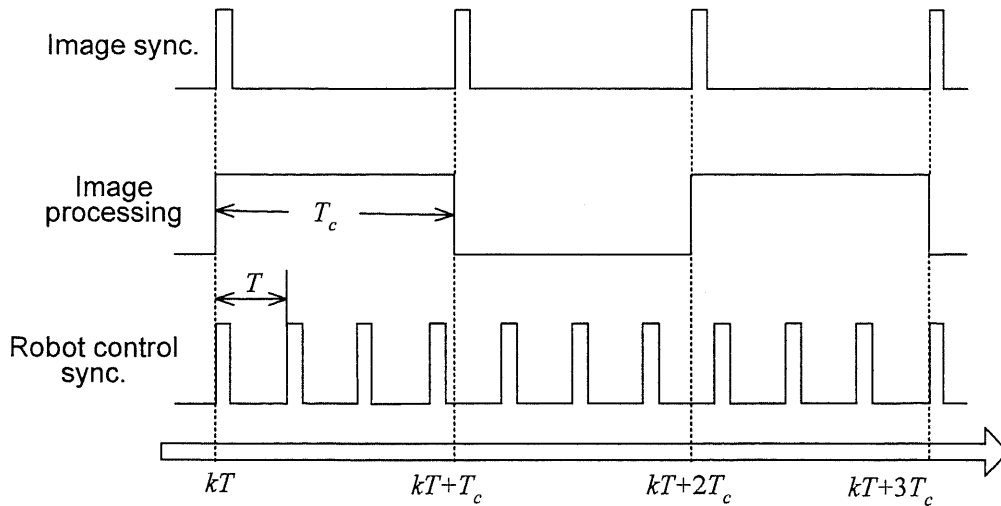


Fig. 3.3 Synchronization of image processing and robot control

$$p_j(iT) = \frac{p_j(kT - T_c) - p_j(kT - 2T_c)}{T_c} iT + p_j(kT - T_c) \quad kT \leq iT < kT + T_c \dots \dots \dots (3.13)$$

により補間を行う。同様に，式(3.7)より得られた目標位置・姿勢  $p_d(kT - T_c)$  の補間は

$$p_d(iT) = \frac{p_d(kT - T_c) - p_d(kT - 2T_c)}{T_c} iT + p_d(kT - T_c) \quad kT \leq iT < kT + T_c \dots \dots \dots (3.14)$$

により行われる。また， $p_i(k)$  の補間は

$$p_i(iT) = \frac{p_i(kT - T_c) - p_i(kT - 2T_c)}{T_c} iT + p_i(kT - T_c) \quad kT \leq iT < kT + T_c \dots \dots \dots (3.15)$$

から計算される。

### 3.4 ニューラルネットワークのオンライン学習によるロボットの制御

外部環境と接触がある作業に，ロボットの手先に仮想的な力学的特性を持たせる仮想コンプライアンス制御法（以下，CC手法と呼ぶ）を用いることが多い。一般に，仮想コンプライアンス制御則は

$$\dot{p}_r(k+1) = (I - TM_c^{-1} \cdot D_c) \cdot \dot{p}(k) + TM_c^{-1} \cdot \Delta f(k) + TM_c^{-1} \cdot K_c \cdot \Delta p(k) \dots \dots (3.16)$$

で表される。ただし，

$$\dot{p}_r(k+1) \in R^{6 \times 1} : \text{時刻 } t=(k+1)T \text{ での手先の目標速度,}$$

- $\dot{p}(k) \in R^{6 \times 1}$  : 時刻  $t=kT$  での手先の速,  
 $M_c \in R^{6 \times 6}$  : 仮想質量行列  
 $D_c \in R^{6 \times 6}$  : 仮想粘性行列  
 $K_c \in R^{6 \times 6}$  : 仮想ばね行列  
 $\Delta f(k) \in R^{6 \times 1}$  : 時刻  $t=kT$  での目標の力  $f_d(k) \in R^{6 \times 1}$  と接触力  $f_e(k) \in R^{6 \times 1}$  の差で  
 あり,  $\Delta f(k)$  は  $f_d(k) - f_e(k)$  によって算出される  
 $I \in R^{6 \times 6}$  : 単位行列  
 $\Delta p(k) \in R^{6 \times 1}$  : 時刻  $t=kT$  における手先の目標位置  $p_d(k)$  と手先の現在位置  
 $p(k)$  の差であり,  $\Delta p(k)$  は  $p_d(k) - p(k)$  によって算出される

式(3.16)に示すコンプライアンス制御は, 検出された位置偏差  $\Delta p(k)$  と力偏差  $\Delta f(k)$  が  $\mathbf{0}$  になるように手先の速度を調整する. 接触によって作業対象に変形が生じる場合には, その変形の情報を式(3.16)に反映することなく, CC手法をそのまま適用すると, 要求される力学的特性を実現できず, 作業を遂行できないことが多い. 一方, 人間が物体を操る場合, 目で見ている視覚情報と手で感じる力覚情報を脳で融合しながら望ましい手先の姿勢を素早く調整し作業を実現する. したがって, 柔軟物体を扱うとき対象物体の変形を式(3.16)に取り入れて考慮する必要がある.

ロボットの手先に取り付けられているホールが棒状柔軟物体に接触すると, 挿入力によって変形が生じる. このとき, 棒状柔軟物体が真っすぐになればなるほど, 挿入作業が行いやすくなると考えられる. したがって, 本章での作業方針としては, 座標系  $\Sigma_h$  の  ${}^h y$ - ${}^h z$  平面において位置誤差の情報によって位置制御を行い, 挿入動作を行うように座標系  $\Sigma_h$  の  ${}^h x$  軸方向における力の制御を行う. また, 棒状物体の変形を考慮するため, 視覚情報と力覚情報をニューラルネットワーク (以下, NNと呼ぶ) の学習によってフュージョンさせ, ロボットの姿勢を調整する.

### 3.4.1 ニューラルネットワークの構成

提案するNNの構成を図3.4に示す. 実験で用いるロボットの姿勢の3自由度に対して, それぞれ一つのサブNNを構成する必要がある.

図3.4に示すNNの学習においては, 座標系  $\Sigma_h$  における時刻  $t=kT$  での棒先端の位置  ${}^h p_t(k)$ , 手先の速度  ${}^h \dot{p}(k)$ , 力覚センサから検出された接触力の情報  ${}^h f_e(k)$  をNNに入力させて学習を行う. ここで, 選択行列  $S_n, S_q$  は

$$S_n = \begin{bmatrix} \mathbf{0}_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \dots \dots \dots (3.17)$$

$$S_q = \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ I_{3 \times 3} \end{bmatrix} \dots \dots \dots (3.18)$$

で表される.

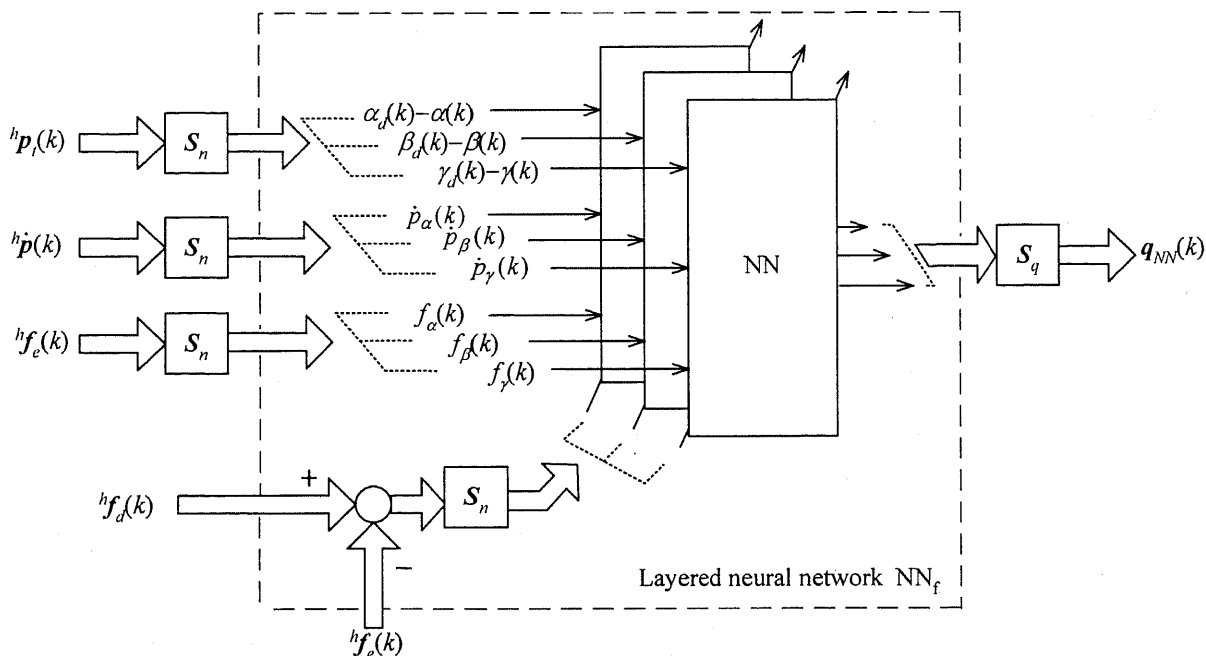


Fig. 3·4 Architecture of the proposed neural network for learning of  $\Delta f(k)$

### 3·4·2 ニューラルネットワークの学習

本章では、図 3·4 に示す各サブNNは3層型NNとし、その構成を図 3·5 に示す。入力層を A，中間層を B，出力層を C とし、各層のニューロン数を  $N_A, N_B, N_C$  とする。また、学習に用いるシグモイド関数  $F(x)$  は

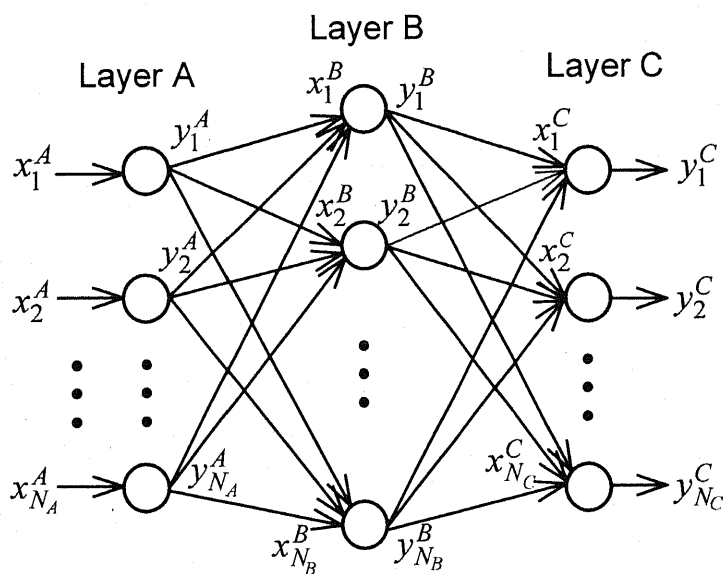


Fig. 3·5 A layered neural network



$$F(x) = \frac{1 - e^{-\frac{x}{\mu}}}{1 + e^{-\frac{x}{\mu}}} \dots \dots \dots (3 \cdot 19)$$

で表される。ただし、 $\mu$  はアニーリングパラメータである。結合荷重の学習は誤差逆伝ば法により行われる<sup>(20)</sup>。

### 3・4・3 ニューラルネットワークによる学習を用いたロボットの制御

座標系 $\Sigma_h$ において仮想コンプライアンス制御を行うための制御則は

$${}^h \dot{\mathbf{p}}_r(k+1) = \mathbf{A}^h \dot{\mathbf{p}}(k) + \mathbf{B} \Delta^h \mathbf{f}(k) + \mathbf{C} \Delta^h \mathbf{p}(k) \dots \dots \dots (3 \cdot 20)$$

で表される。ただし、

${}^h \dot{\mathbf{p}}_r(k+1) \in R^{6 \times 1}$ : 座標系 $\Sigma_h$ における時刻  $t=(k+1)T$  での手先の目標速度

${}^h \dot{\mathbf{p}}(k) \in R^{6 \times 1}$ : 座標系 $\Sigma_h$ における時刻  $t=kT$  での手先の速度

$\mathbf{A} \in R^{6 \times 6}$ : 係数行列  $\mathbf{A} = \mathbf{I} - \mathbf{M}_c^{-1} \mathbf{D}_c T$

$\mathbf{B} \in R^{6 \times 6}$ : 係数行列  $\mathbf{B} = \mathbf{M}_c^{-1} \mathbf{T}$

$\mathbf{C} \in R^{6 \times 6}$ : 係数行列  $\mathbf{C} = \mathbf{M}_c^{-1} \mathbf{K}_c T$

$\Delta^h \mathbf{f}(k) \in R^{6 \times 1}$ : 座標系 $\Sigma_h$ における時刻  $t=kT$  での目標の力  ${}^h \mathbf{f}_d(k)$  と接触力  ${}^h \mathbf{f}_e(k)$  の差であり、 $\Delta^h \mathbf{f}(k)$  は  ${}^h \mathbf{f}_d(k) - {}^h \mathbf{f}_e(k)$  より算出される

ただし、 ${}^h \mathbf{f}_d(k) = [{}^h f_{xd} \quad {}^h f_{yd} \quad {}^h f_{zd} \quad {}^h f_{\alpha d} \quad {}^h f_{\beta d} \quad {}^h f_{\gamma d}]^T$

${}^h \mathbf{f}_e(k) = [{}^h f_x \quad {}^h f_y \quad {}^h f_z \quad {}^h f_\alpha \quad {}^h f_\beta \quad {}^h f_\gamma]^T$

$\Delta^h \mathbf{p}(k) \in R^{6 \times 1}$ : 座標系 $\Sigma_h$ における時刻  $t=kT$  での目標位置  ${}^h \mathbf{p}_d(k)$  と現在位置  ${}^h \mathbf{p}(k)$  の差であり、 $\Delta^h \mathbf{p}(k)$  は  ${}^h \mathbf{p}_d(k) - {}^h \mathbf{p}(k)$  より算出される

図 3・2 に示すように、座標系 $\Sigma_h$ の  ${}^h y$ - ${}^h z$  平面において位置制御を行い、 ${}^h x$  軸方向において力の制御を行うため、選択行列を入れる。したがって、式(3・20)は

$${}^h \dot{\mathbf{p}}_r(k+1) = \mathbf{A} \cdot {}^h \dot{\mathbf{p}}(k) + \mathbf{B} \cdot \mathbf{S}_f \cdot \Delta^h \mathbf{f}(k) + \mathbf{C} \cdot \mathbf{S}_p \cdot \Delta^h \mathbf{p}(k) \dots \dots \dots (3 \cdot 21)$$

で表される。ただし、

選択行列  $\mathbf{S}_f = \text{diag}[1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$ ,

選択行列  $\mathbf{S}_p = \text{diag}[0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0]$ ,

また、棒状物体の変形を考慮するため、本章でニューラルネットワークのオンライン学習によって視覚と力覚のセンサフュージョンを行い、ロボットの姿勢を制御するという手法を提案する。したがって、式(3・21)にニューラルネットワークの出力を姿勢制御に加えると、本章で提案する制御アルゴリズムは

$${}^h \dot{\mathbf{p}}_r(k+1) = \mathbf{A} \cdot {}^h \dot{\mathbf{p}}(k) + \mathbf{B} \cdot \mathbf{S}_f \cdot \Delta^h \mathbf{f}(k) + \mathbf{C} \cdot \mathbf{S}_p \cdot \Delta^h \mathbf{p}(k) + \mathbf{H}_q \cdot \mathbf{q}_{NN} \dots \dots \dots (3 \cdot 22)$$

で表される。ただし、スケーリング係数行列  $H_n$  は

$$H_q = \text{diag}[0 \ 0 \ 0 \ h_1 \ h_2 \ h_3]$$

である。また、式(3.22)で得られた  ${}^h\dot{\mathbf{p}}_r(k+1)$  から

$$\dot{\mathbf{p}}_r(k+1) = {}^b\mathbf{R}_h \cdot {}^h\dot{\mathbf{p}}_r(k+1) \cdots \cdots \cdots (3.23)$$

によって座標系  $\Sigma_b$  におけるロボット手先の目標速度  $\dot{\mathbf{p}}_r(k+1)$  を算出できる。

式(3.22)に示すNNのオンライン学習を用いたロボット制御のブロック線図を図3.6に示す。作業開始のとき、ロボット手先をスタート位置から棒状物体の先端位置  $\mathbf{p}_i$  まで移動させる。ロボットの手先が棒の端と接触してから、式(3.22)で提案した手法によってロボットの制御を行う。

図3.6においては、時刻  $t=kT$  において GENESIS の画像処理によって棒状柔軟物体に貼り付けているマーカーの位置  $\mathbf{p}_i(kT-T_0)$  ( $i=1,2,\dots,5$ ) を検出し、式(3.13)により補間を行い、 $\mathbf{p}_i(k)$  ( $i=1,2,\dots,5$ ) が得られる。次に、生成される目標位置と姿勢  $\mathbf{p}_d(k)$  とロボットの手先の現在位置  $\mathbf{p}(k)$  の差  $\Delta\mathbf{p}(k)$  を求め、位置と力の制御を行う。また、時刻  $t=kT$  にて検出された力覚情報  ${}^h\mathbf{f}_e(k)$ 、手先の速度  ${}^h\dot{\mathbf{p}}(k)$ 、棒状物体の先端の位置  ${}^h\mathbf{p}_i(k)$  を NN に入力させる。

このようにして得られた位置と力覚の情報を式(3.22)、(3.23)で提案した制御則によってロボットの手先の目標速度  $\dot{\mathbf{p}}_r(k+1)$  を算出する。また、逆ヤコビ行列  $\mathbf{J}^{-1}$  によってロボットの各目標関節角速度  $\dot{\theta}_r(k+1)$  を計算し、 $\dot{\theta}_r(k+1)$  と現在の関節角速度  $\dot{\theta}(k)$  との差をサーボドライブに入力させ、ロボットを制御する。また、ロボットを制御するコンピュータでは、制御サンプリング周期  $T$  (5ms) の間に NN の学習を4回行う。

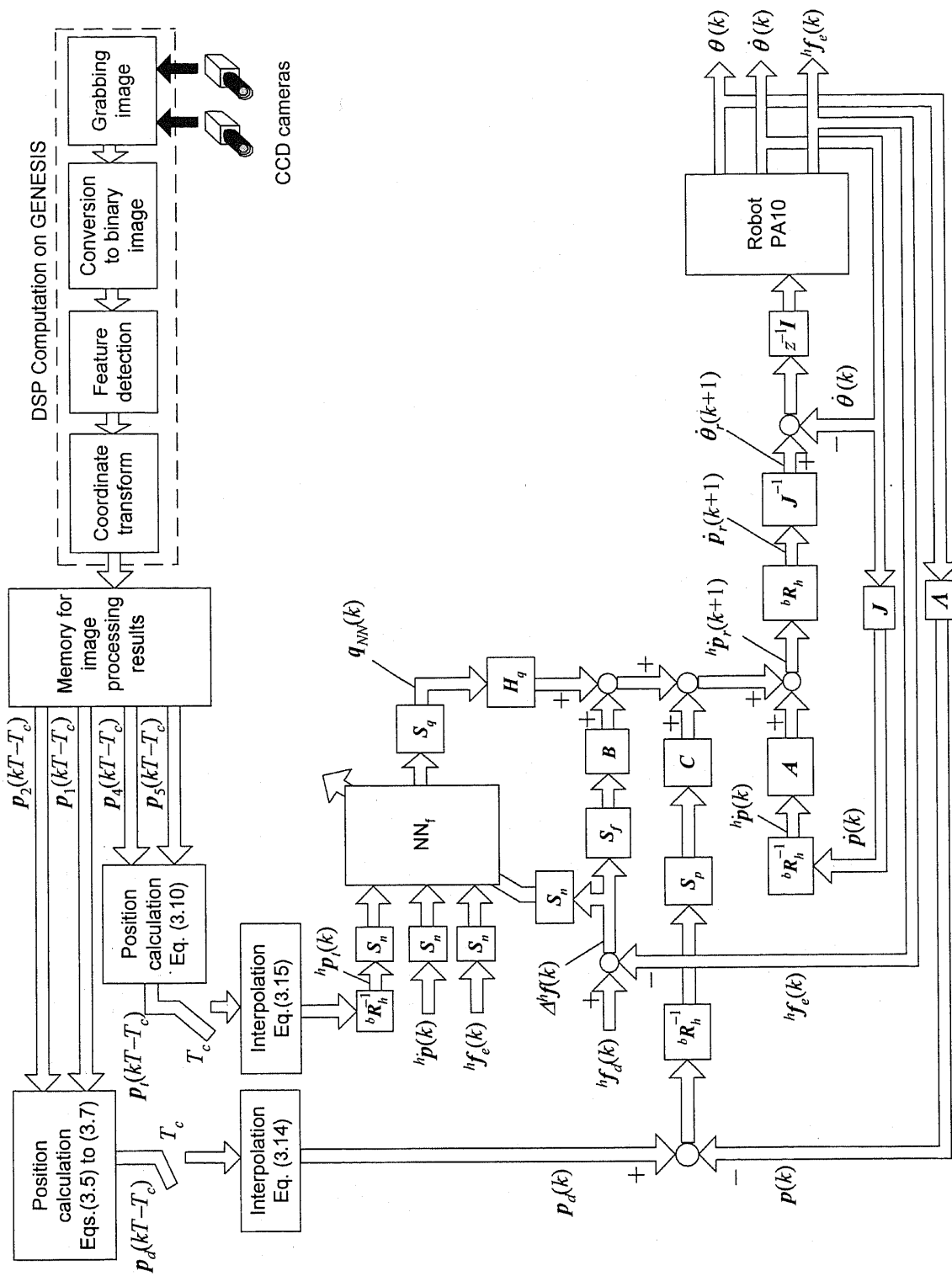


Fig. 3·6 The proposed neuro-control issue for a robot to insert a beam into a hole

### 3.5 実験とその結果

本章では，図 3.6 で提案する制御手法の有効性を検証するため，2 種類の材料で作った棒状物体を用いて“はめ合い実験”を行った．棒状物体の材料と寸法を表 3.1 に示す．

また，実験の結果を比較するため，式(3.16)に示す CC 手法を用いて表 3.1 に示す作業対象に対しはめ合い実験を行った．CC 手法を用いた制御は，図 3.6 から NN の部分を除いたブロック線図で表される．また，ロボットの手先が振動的にならないように，式(3.22)に示す制御パラメータ  $A$ ,  $B$ ,  $C$  を決定する．実験で用いたパラメータの値を表 3.2 に示す．また，式(3.22)に示すスケーリング係数行列  $H_q$  は試行錯誤により選ぶ．

Table 3.1 Beam objects used in the experiments

Materials	Iron	Acrylic
Diameter $\phi$ mm	3	3
Length mm	350	350

Table 3.2 Control parameters used in the experiments

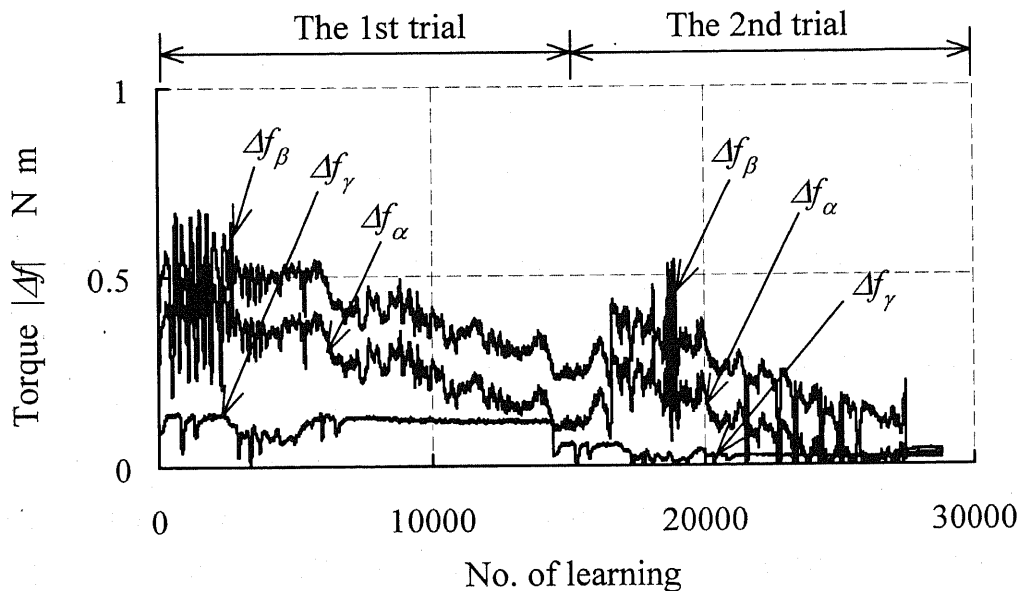
Compliance control	$M_c = \text{diag} [m_c \ m_c \ m_c \ m_r \ m_r \ m_r]$ $m_c = 0.7 \text{ kg}, \quad m_r = 0.5 \text{ kg} \cdot \text{m}^2$
	$D_c = \text{diag} [d_c \ d_c \ d_c \ d_r \ d_r \ d_r]$ $d_c = 114 \text{ N} \cdot \text{s} \cdot \text{m}^{-1}, \quad d_r = 8 \text{ N} \cdot \text{m} \cdot \text{s} \cdot \text{rad}^{-1}$
	$K_c = \text{diag} [K_1 \ K_1 \ 0 \ K_2 \ K_2 \ K_2]$ $K_1 = 14 \text{ N} \cdot \text{m}^{-1}, \quad K_2 = 1 \text{ N} \cdot \text{m} \cdot \text{rad}^{-1}$
	${}^h f_d = [0 \ 0 \ f_c \ 0 \ 0 \ 0]$ $f_c = 1 \text{ N} \quad (\text{for iron and acrylic}), \quad 2 \text{ N} \quad (\text{for iron})$
Neuro-control	$A = \text{diag} [a_1 \ a_1 \ a_1 \ a_2 \ a_2 \ a_2]$ $a_1 = 0.2, \quad a_2 = 0.2$
	$B = \text{diag} [b_1 \ b_1 \ b_1 \ b_2 \ b_2 \ b_2]$ $b_1 = 0.1 \text{ m} \cdot \text{s}^{-1} \cdot \text{N}^{-1}, \quad b_2 = 0.2 \text{ rad} \cdot \text{s}^{-1} \cdot \text{N}^{-1}$
	$C = \text{diag} [c_1 \ c_1 \ c_1 \ c_2 \ c_2 \ c_2]$ $c_1 = 0.02 \text{ s}^{-1}, \quad c_2 = 0.01 \text{ s}^{-1}$
	$H_q = \text{diag} [0 \ 0 \ 0 \ h_1 \ h_1 \ h_1]$ $h_1 = 0.02 \text{ m} \cdot \text{s}^{-1}$
NN learning	$N_a = 3, \quad N_b = 3, \quad N_c = 1,$ $\mu = 1.0, \quad \eta_w = \eta_z = 0.85, \quad \alpha_w = \alpha_z = 0.75$

### 3.5.1 鉄製棒状物体を用いたはめ合い実験

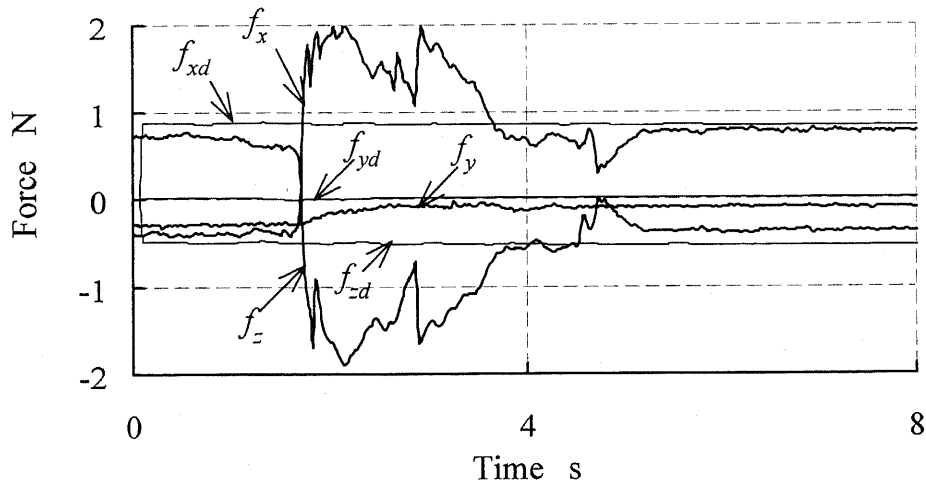
NNの結合荷重には学習開始時点で乱数を与えているため、作業を2回続けて試行した。NN学習の誤差信号の絶対値を図3.7(a)に示す。1回目の試行では、作業時間が長く、ホールはペグの中間部分までしか挿入しなかった。2回目の試行での学習誤差は1回目より減少していることが図3.7(a)から分かる。2回目の試行では、ホールは完全にペグをはめることができた。

このようにして得られたNNを用いて鉄製の棒状物体に対して続けて挿入実験を行った。3試行目以降の実験結果は、ほとんど同じパターンを示しており、図3.7(b)はその結果の一例である。また、実験の結果を比較するため、CC手法を用いて同じ対象に対して実験を行った。その実験結果を図3.7(c)に示す。

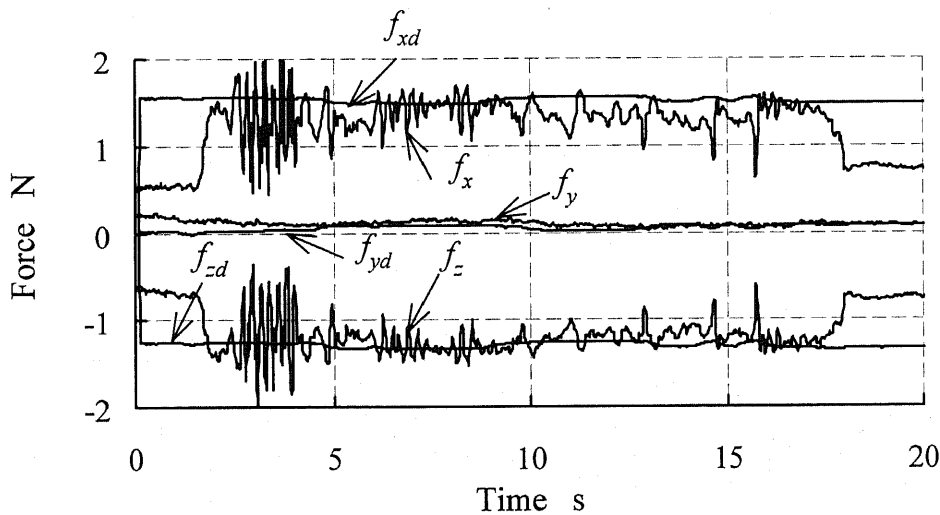
図3.7(b), (c)では、挿入作業を開始した直後に小振りの振動を示したが、棒状物体は鉄製で硬いため、接触力による変形はほとんどなかった。図3.7の(b)と(c)を比較すると、(1)両方とも接触が開始した時点で接触力は力目標値の付近にて小幅に変動する傾向があり、作業の進行につれて徐々に力目標値に近づいていくこと、(2)NN手法の作業時間はCC手法より大幅に短縮していることが分かる。



(a) Absolute error of NN learning ( $f_c=1N$ )



(b) Force results obtained by using Neuro-control method ( $f_c=1\text{N}$ )



(c) Force results obtained by using compliance control method ( $f_c=2\text{N}$ )

Fig.3.7 Results obtained in an experiment of inserting an iron beam into a hole

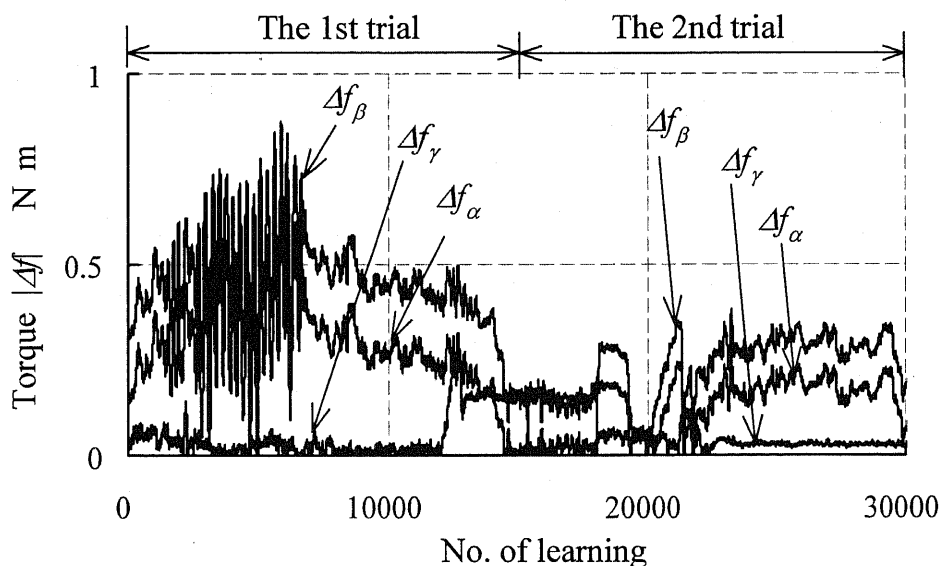
### 3.5.2 アクリル製の棒状物体を用いたはめ合い実験

鉄製棒状物体の実験と同じように、NNの結合荷重には学習開始時点で乱数を与えているため、作業を2回続けて試行した。NN学習の誤差信号の絶対値を図3.8(a)に示す。1回目の試行では、アクリル製の棒状物体は柔らかいため、作業時間が長く、ホールはペグの上部までしか挿入しなかった。2回目の試行での学習誤差は1回目より減少していることが図3.8(a)から分かる。2回目の試行では、ホールは完全にペグをはめることができた。

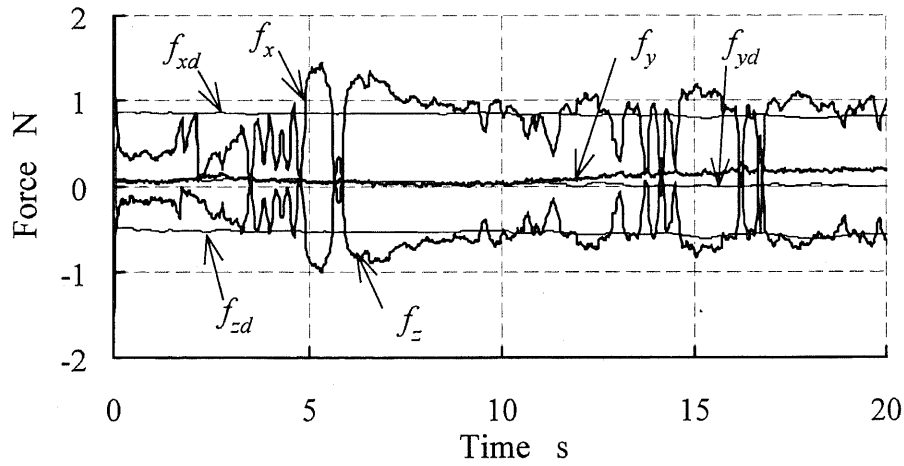
このようにして得られたNNを用いてアクリル製棒状物体に対して続けて実験を行った。3試行目以降の実験の結果はほとんど同じパターンを示しており、図3・8(b)は結果の一例である。実験の結果を比較するため、CC手法を用いて同じ対象に対してはめ合い実験を行った。

CC手法を用いた実験で得られた接触力の結果を図3・8(c)に示す。また、アクリル棒状物体の中間部に貼り付けられている特徴点  $p_3(k)$  の位置を図3・8(d),(e)に示す。CC手法を用いた場合には、図3・8(d),(e)に示すように接触力によって生じた変形が作業開始の状態から徐々に大きくなるのが分かった。これは、式(3・16)で示すCC手法では、アクリル棒の変形に対して手先の姿勢を柔軟に調整できないためと考えられる。

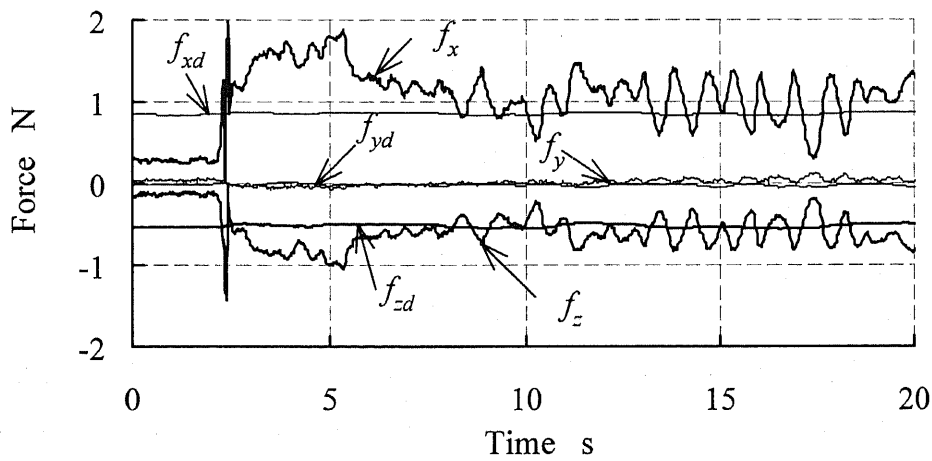
図3・8(d),(e)では、NNを用いた本手法は、特徴点  $p_3(k)$  の位置が作業開始のときの位置に戻るといった傾向を示している。これにより、本手法は対象物体の変形を一定の範囲内に抑えることができたといえる。これは、接触により生じた対象物体の変形に対してNNの学習によってロボットの手先の位置・姿勢を効率的に調整できたためである。



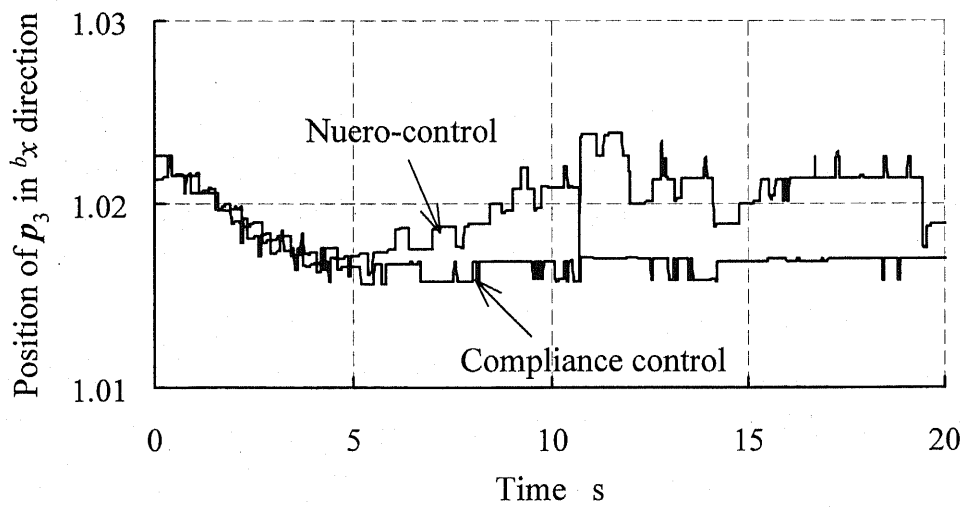
(a) Absolute error of NN learning ( $f_c=1N$ )



(b) Force results using neuro-control method

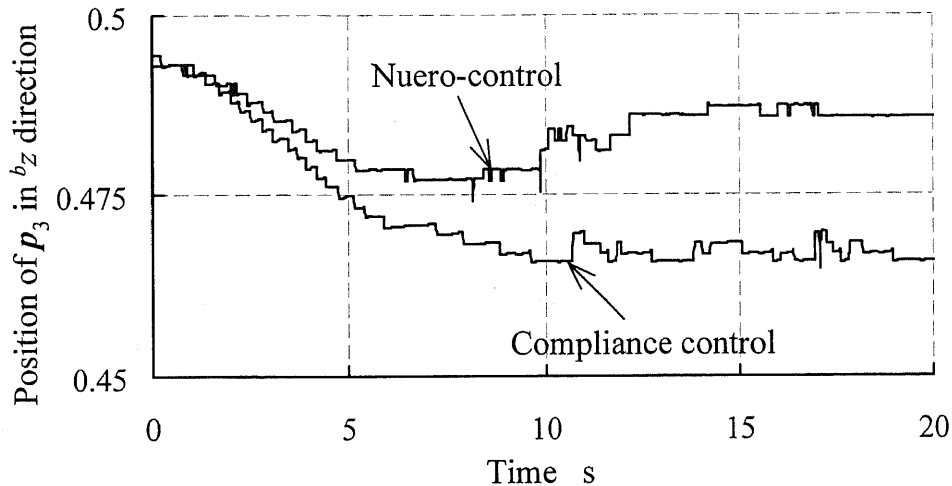


(c) Force results using compliance control method



(d) Position  $p_3$  in  $b_x$  direction





(e) Position  $p_3$  in  $b_z$  direction

Fig.3・8 Results obtained of inserting an acrylic beam into a hole ( $f_c=1N$ )

変形の少ない鉄棒の場合，CC手法と提案したNN手法とでは，同じように作業を遂行した．一方，柔らかいアクリル棒の場合，CC手法では作業を遂行できないのに対し，提案したNN手法により作業を実現できることが上記の実験結果から分かった．しかしながら，提案したNN手法では，①NNの学習誤差が0に収束するまでに一定の実験回数を必要とすること，②ニューラルネットワークのスケーリング係数行列の設定は経験による，など要因があるため，本実験で得られたNNの適用は，実験に用いた対象物体の変形特性に近いものに限定する．

### 3・6 要約

本章で得られた主な結果は，次のとおりである．

(1)視覚情報と力覚情報をニューラルネットワークの学習によってフュージョンするという手法を提案し，棒状柔軟物体の操作に適用することによって本手法の有効性を確かめた．

(2)棒状対象物体の変形量を推定するために作業対象の剛性パラメータの値をこれまでの研究で必要としていたのに対し，本章で提案した手法は，作業対象の剛性パラメータの値を必要としないという特長を持っている．本手法を用いて異なる材質の棒状物体に対し“はめ合い作業”を行い，その有効性を実証した．

---

## 第4章

# 柔軟ツールを用いたロボットの制御

---

### 4.1 概説

第2, 3章では, ロボットによる柔軟物体の操作に視点をおき, その解決法を導いた. 一方, 生活環境においては, 傷つけやすい物体や壊れやすい物体などの対象に対してはロボットの手先に柔軟ツールを取り付け, ツール先端の位置と力を制御する必要がしばしばある. しかしながら, ツールの材質によって変形モデルを算出できない場合には, 運動学に基づくこれまでの手法をツール先端の位置制御に適用することは難しい.

柔軟ツールをロボットの手先に取り付けてツール先端に接触作業を行わせる場合, 接触によってツールに変形が生じる. ツールの変形モデルを作成し変形を解析しないかぎり, ロボットの運動学からツール先端の位置を算出することはできない. このため, 本章では, 立体視情報を用いてツールの先端位置を高速で検出することによって構成する位置と力のハイブリッド制御手法を提案する. また, 柔軟ツールの先端の追従精度を向上させるため, ニューラルネットワークによるオンライン学習を用いる. 提案した手法を用いて柔軟ツールの先端を与えられた平面曲線軌道に押し付けながら追従させる実験を行う.

### 4.2 作業システムの構成と作業の設定

本章で用いる作業システムは, 可搬式汎用知能アーム (PA10, 三菱重工) 1台, 力覚センサ (F/T 10/100, ビー・エル・オートテック) 1台, 小型 CCD カメラ (XC-EI50, ソニー) 2台とパーソナルコンピュータ (Dimension XPS B733r, DELL 社) 1台から構成されている. その構成を図 4.1 に示す. 作業中に生じる接触力の情報を検出するため, 力覚センサをロボットの手先に装着する.

また, 長さ 0.3m の塩化ビニール製チューブを柔軟ツールとしてロボットの手先に取り付ける. ツールの先端位置を画像処理によって検出するため, カメラ 2台の前にリング型赤外照射器 (IRDR-110, 日進電子工業) を取り付け, チューブの先端に反射

マーカーを貼り付ける。また、図 4・1 に示すロボットアームは 7 自由度を有するが、本章では、アームの冗長自由度を固定し、6 自由度アームとして用いる。

図 4・1 に示すように、2 台の CCD カメラが、 $b_x$  軸の正方向に約 0.66m と 1.04m、 $b_y$  軸の負方向に約 0.55m, 0.45m、 $b_z$  軸の正方向に約 0.65m の場所に置かれている。2 台のカメラの距離は 0.45m である。2 台のカメラから構成される立体視画像システムによって 3 次元空間におけるツールの先端位置を検出し、ロボットの制御に用いる。また、 $b_x$  軸の正方向に約 0.7m の場所に作業台を置く。作業は、柔軟ツールの先端を  $b_z$  軸に一定の力で押し付けながら作業台に与えられている曲線軌道に追従させることである。

### 4・3 画像処理で得られたツール先端位置の補間

図 4・1 に示す 2 台の CCD カメラは毎秒 30 フレームの画像を出力する。2 台のカメラからの映像を画像入力・処理ボード (GENESIS GEN/F/64/8 /STD, Matrox 社) によって処理する。本章では、第 2 章の図 2・3(b) に示す画像の並列処理という手法を用いるため、カメラから連続的に出力されている画像を見落とすことなくリアルタイムに処理することができる。したがって、画像の入力・処理のサンプリング周期  $T_c$  は約 33.33ms である。一方、ロボットの制御サンプリング周期  $T$  は 5ms であるため、ロボットの制御サンプリング周期ごとに画像処理で得られた結果を補間することは必要である。

第 3 章の図 3・3 に示すように、時刻  $t=(n-1)T_c$  にて入力された画像の処理を時刻  $t=kT$  において終了し、2 台のカメラによって柔軟ツール先端の三次元空間位置  $p_i^c(n-1)$  が得られる。 $T_c$  ごとに得られた位置情報  $p_i^c(n-1)$ 、 $p_i^c(n-2)$  を用いて制御サンプリング周期  $T$  ごとの位置  $p_i(k) \in R^{3 \times 1}$  を

$$p_i(k) = \begin{cases} p_i^c(n-1) + (kT - (n-1)T_c) \frac{p_i^c(n-1) - p_i^c(n-2)}{T_c} & (nT_c \leq kT < (n+1)T_c), \\ p_i^c(n) + (kT - nT_c) \frac{p_i^c(n) - p_i^c(n-1)}{T_c} & ((n+1)T_c < kT < (n+2)T_c), \\ p_i^c(n+1) + (kT - (n+1)T_c) \frac{p_i^c(n+1) - p_i^c(n)}{T_c} & ((n+2)T_c < kT < (n+3)T_c), \\ \dots\dots\dots & \dots\dots\dots \end{cases}$$

$$(k=0,1,2,\dots), (n=0,1,2,\dots) \dots\dots\dots (4\cdot1)$$

によって補間する。ただし、 $n=0$  のとき、 $p_i^c(-1)=\mathbf{0}$ 、 $p_i^c(-2)=\mathbf{0}$  とする。

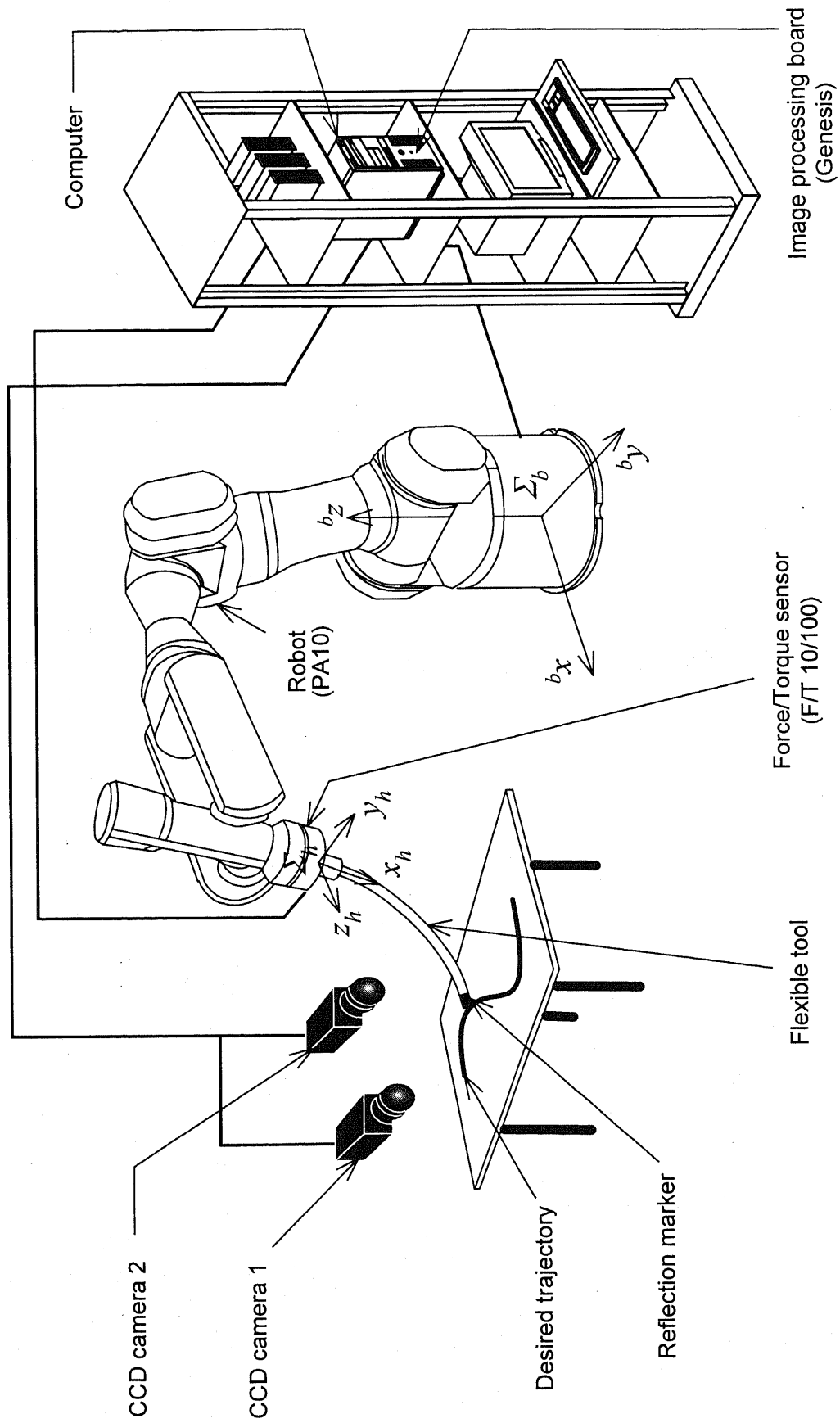


Fig. 4-1 Setup of a robot control system with a flexible tool held by the robot hand



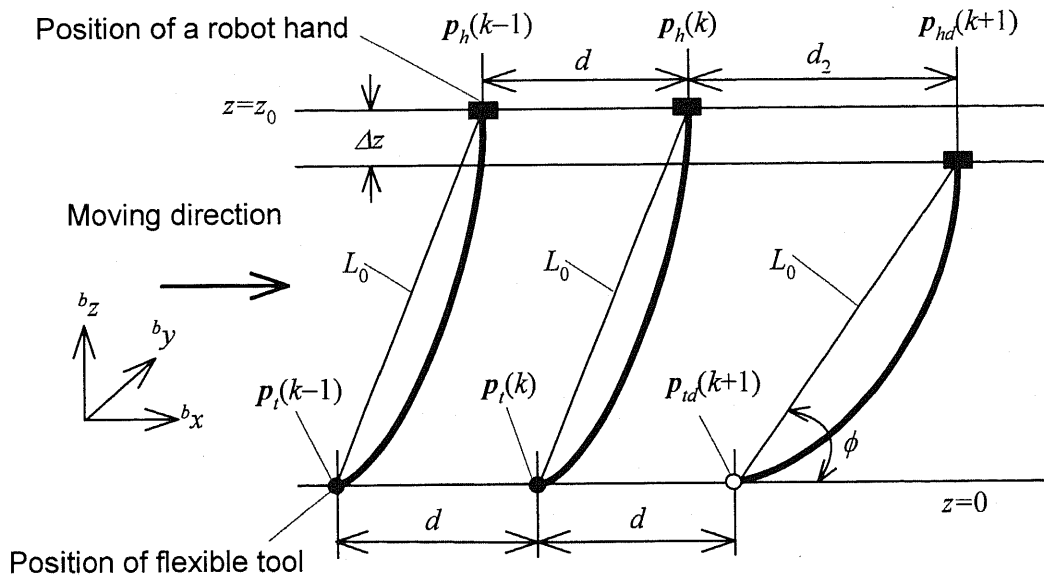


Fig. 4・3 Moving a flexible tool forward with a distance  $d$

#### 4・4・1 ロボット手先の目標位置の生成

図 4・2 に示すように、時刻  $t=kT$  のときロボットの手先を位置  $p_h(k)$  から間隔  $d$  だけ先にある目標位置  $p_{hd}(k+1)$  へ移動させるが、 $b_z$  軸方向における座標の変動  $\Delta z$  によってツール先端は位置  $p_t(k)$  から距離  $d_1$  だけ先の位置  $p_t(k+1)$  に到達し、目標位置  $p_{td}(k+1)$  には到着できないことが分かる。ロボットの手先を三次元空間内で移動させるため、ロボット手先の目標位置と現在位置のそれぞれを  $p_{hd}(k) \in R^{3 \times 1}$ ,  $p_h(k) \in R^{3 \times 1}$  とし、柔軟ツール先端の目標位置と現在位置のそれぞれを  $p_{td}(k) \in R^{3 \times 1}$ ,  $p_t(k) \in R^{3 \times 1}$  とする。

ツール先端を目標位置  $p_{td}(k+1)$  に到達させるためには、図 4・3 に示すように  $b_z$  軸方向の変動  $\Delta z$  に合わせてロボット手先の位置  $p_h(k)$  を距離  $d_2$  先の目標位置  $p_{hd}(k+1)$  に移動させる必要がある。

ここで、ツール先端は  $b_x$ - $b_y$  平面を一定の力  $f_{zd}$  で押し付けているとき、ツール先端とロボット手先との距離の初期値を  $L_0$  とする。時刻  $t=kT$  のとき、 $b_x$ - $b_y$  平面におけるツール先端の目標位置を  $p_{td}(k) = [x_{td}(k), y_{td}(k), z_{td}(k)]^T$  とし、角度  $\phi$  を直線  $L_0$  と  $b_x$ - $b_y$  平面となす角度とする。また、ツール先端の進行方向の方向ベクトル  $\tilde{p}_{td}(k) \in R^{3 \times 1}$  は

$$\tilde{p}_{td}(k) = \frac{p_{td}(k+1) - p_{td}(k)}{d} \dots \dots \dots (4 \cdot 2)$$

で表される。ただし、距離  $d$  は

$$d = \sqrt{(x_{td}(k+1) - x_{td}(k))^2 + (y_{td}(k+1) - y_{td}(k))^2} \dots \dots \dots (4 \cdot 3)$$

から求まる。また、 $b_x$ - $b_y$  平面における直線  $L_0$  の投影ベクトル  $\Delta p_L(k) \in R^{3 \times 1}$  は

$$\Delta p_L(k) = L_0 \cos \phi \cdot \tilde{p}_{td}(k) \dots \dots \dots (4.4)$$

で表される。ただし、角度  $\phi$  を

$$\phi = \sin^{-1} \left( \frac{z_0 - \Delta z}{L_0} \right) \dots \dots \dots (4.5)$$

から求める。さらに、時刻  $t=kT$  におけるツール先端の追従誤差  $\Delta p_t(k)$  を考慮すると、 ${}^b x$ - ${}^b y$  平面におけるロボット手先の目標位置  $p_{hd}^{xy}(k)$  は

$$p_{hd}^{xy}(k) = p_{td}(k+1) + \Delta p_L(k) + \Delta p_t(k) \dots \dots \dots (4.6)$$

から算出される。ただし、ツール先端の追従誤差  $\Delta p_t(k)$  は  $\Delta p_t(k) = p_{td}(k) - p_t(k)$  とする。また、時刻  $t=kT$  におけるロボット手先の目標位置  $p_{hd}(k)$  は

$$p_{hd}(k) = S_1 \cdot p_{hd}^{xy}(k) + z_d s_1 \dots \dots \dots (4.7)$$

から生成される。ただし、行列  $S_1$ 、ベクトル  $s_1$  は

$$S_1 = \text{diag}[1 \ 1 \ 0] \dots \dots \dots (4.8)$$

$$s_1 = [0 \ 0 \ 1]^T \dots \dots \dots (4.9)$$

で与えられる。また、 ${}^b z$  軸方向に力の制御を行うため、便宜上  $z_d$  は  $z_0$  とする。

#### 4.4.2 ロボット手先の目標姿勢の生成

図 4.1 に示すように、ベース座標系  $\Sigma_b$  におけるロボットの手先座標系  $\Sigma_h$  の姿勢は  $x$ - $y$ - $z$  固定角法によって定義される。ここで、 $t=kT$  においてロボット手先の姿勢を  $\alpha(k)$ 、 $\beta(k)$ 、 $\gamma(k)$  とする。このとき、ロボット手先の目標姿勢  $r_{hd}(k) \in R^{3 \times 1}$  は

$$r_{hd}(k) = [\alpha_d(k) \ \beta_d(k) \ \gamma_d(k)]^T \dots \dots \dots (4.10)$$

で表される。ツール先端の進行方向と  ${}^b z$  軸を含む平面の法線回りにロボット手先の回転がないようにするため、 $\alpha_d(k)$ 、 $\beta_d(k)$  はそれぞれ一定値  $\alpha_d$ 、 $\beta_d$  にする。しかしながら、図 4.4 に示すように進行方向に対してロボットの手先がツール先端の後方に置かれた場合、摩擦力の働きによってねじりモーメントが生じる。このため、柔軟ツール自身のねじり回転が発生し、ツール先端の位置を制御することはできない。したがって、ねじりモーメントが発生する原因は、押す力と摩擦力の方向が逆であるためと考えられる。

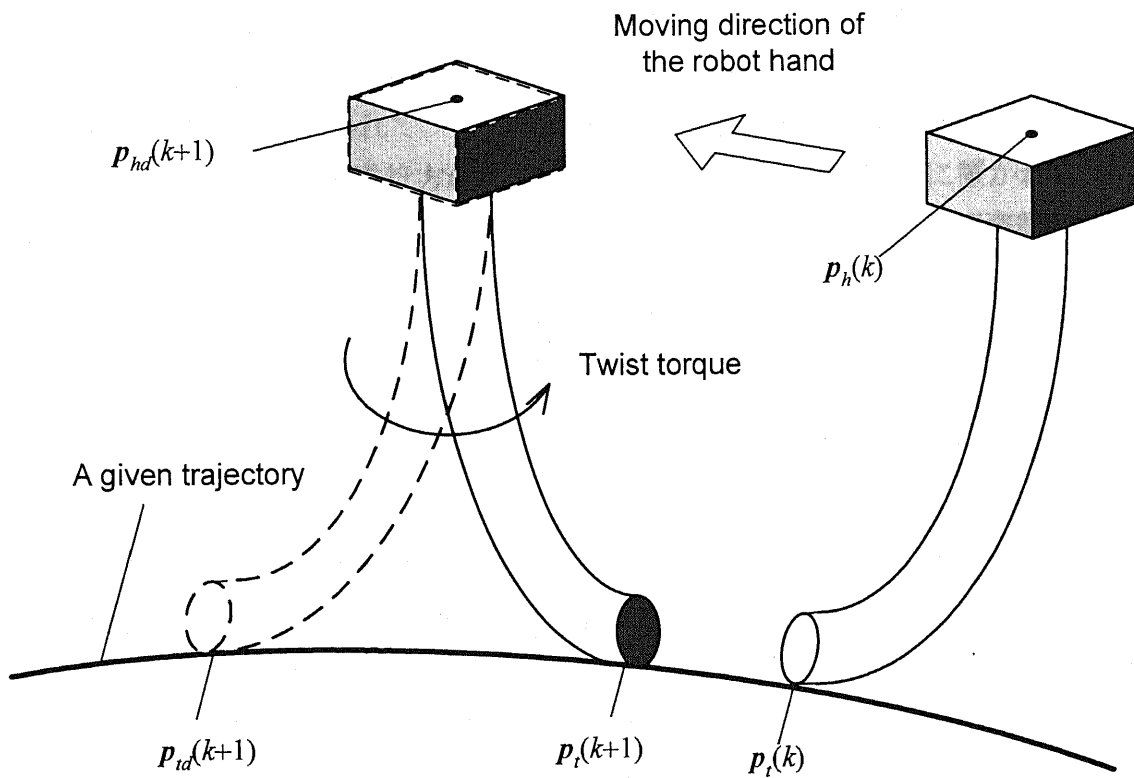


Fig. 4·4 Rotation of the flexible tool caused by twist torque

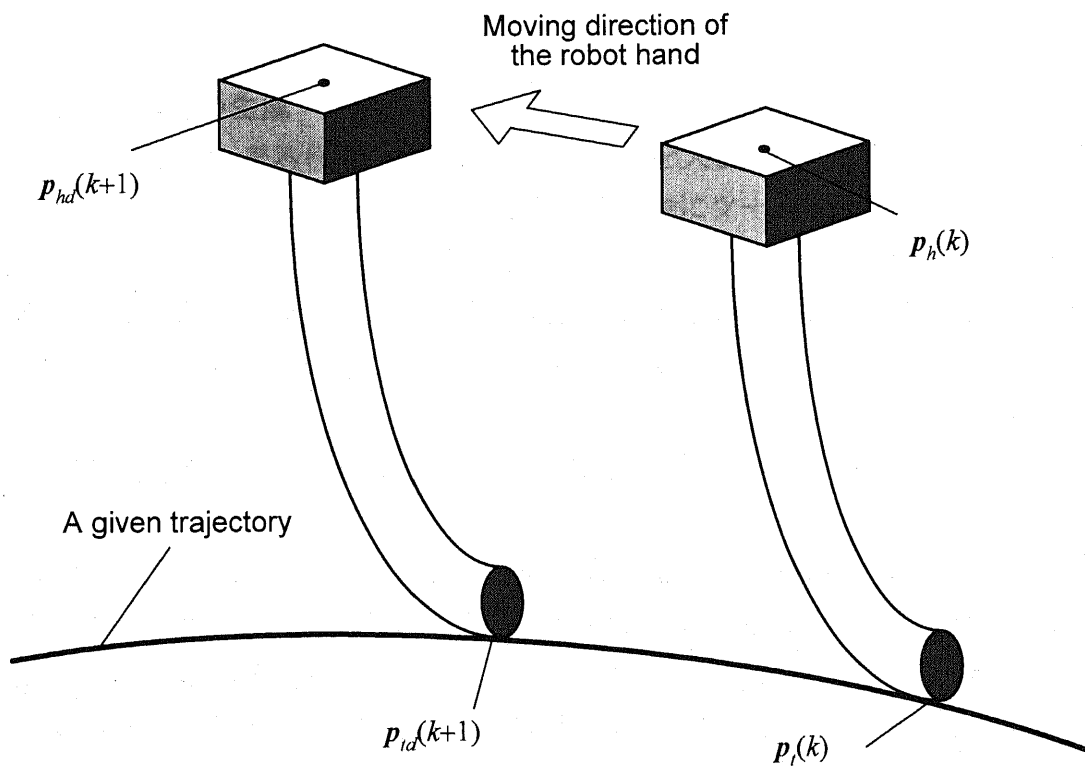


Fig. 4·5 Avoidance of rotation of the flexible tool caused by twist torque



本章では、ツール自身のねじり回転を回避するため、進行方向においてロボットの手先がツール先端を押しながら進めるのではなく、図 4・5 に示すようにロボットの手先がツール先端の前方に置かれてツール先端を引いて進めることにする。このため、本章では、ロボットの手先が進行方向に対してツール先端の前方に常に置かれるようにしなければならない。したがって、式(4・10)に示すロボット手先の目標姿勢を

$$r_{hd}(k) = \left[ \alpha_d \quad \beta_d \quad \tan^{-1}\left(\frac{y_{td}(k)}{x_{td}(k)}\right) \right]^T \dots\dots\dots (4\cdot11)$$

とする。

また、式(4・7)、(4・11)を合わせることによってロボット手先の目標位置・姿勢  $p_{rd}(k) \in R^{6 \times 1}$  は

$$p_{rd}(k) = S_p \cdot p_{hd}(k) + S_r \cdot r_{hd}(k) \dots\dots\dots (4\cdot12)$$

により生成される。ただし、選択行列  $S_p, S_r$  は

$$S_p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad S_r = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

である。

### 4・4・3 目標力の生成

図 4・1 に示す“なぞる作業”では、押し付け力を大きくすると、ツール先端に加える摩擦力も大きくなり、図 4・5 に示すように進行中にツール先端のねじり回転が発生しやすくなる。本章では、これを避けるためツール先端を  $z$  軸方向に一定の力で作業面に押し付けるように設定する。すなわち、 $z$  軸方向での目標力  $f_{zd}$  を一定にする。また、 $z$  軸方向における位置制御の目標値  $z_d$  を 0 とする。

## 4・5 位置と力のハイブリッド制御系の構成

柔軟ツールの先端を与えられた二次元曲線軌道に一定の力でなぞらせる作業を遂行するため、 ${}^b z$  軸方向に力覚センサを用いて押し付け力を制御し、 ${}^b x, {}^b y$  軸方向に視覚情報によって位置制御を行う。しかしながら、ツールの変形や手先の摩擦などの影響要因があるため、単に PID 制御を用いて位置の追従誤差を減少することは難しい。そこで、追従精度を向上させるため、 ${}^b x, {}^b y$  軸方向における位置制御にオンライン学習型ニューラルネットワークを本章で提案した制御系に用いる。

#### 4.5.1 ニューラルネットワークの構成

本章で提案するNNの構成を図4.6に示す。 $b_x, b_y$ 軸方向におけるロボットの位置制御に対して、それぞれのサブニューラルネットワークを構成する必要がある。

図4.6に示すサブニューラルネットワーク  $NN_x$  への入力は、 $b_x$ 軸方向における時刻  $t=kT$  でのロボット手先の目標位置  $x_{rd}(k)$ 、手先の目標速度  $\dot{x}_{rd}(k)$ 、手先の目標加速度  $\ddot{x}_{rd}(k)$  であり、学習するための修正量は手先の追従誤差  $\Delta x_r(k)$  とする。 $NN_x$  と同様に、サブニューラルネットワーク  $NN_y$  の入力は、 $b_y$ 軸方向における時刻  $t=kT$  でのロボット手先の目標位置  $y_{rd}(k)$ 、手先の目標速度  $\dot{y}_{rd}(k)$ 、手先の目標加速度  $\ddot{y}_{rd}(k)$  であり、学習するための修正量は手先の追従誤差  $\Delta y_r(k)$  とする。

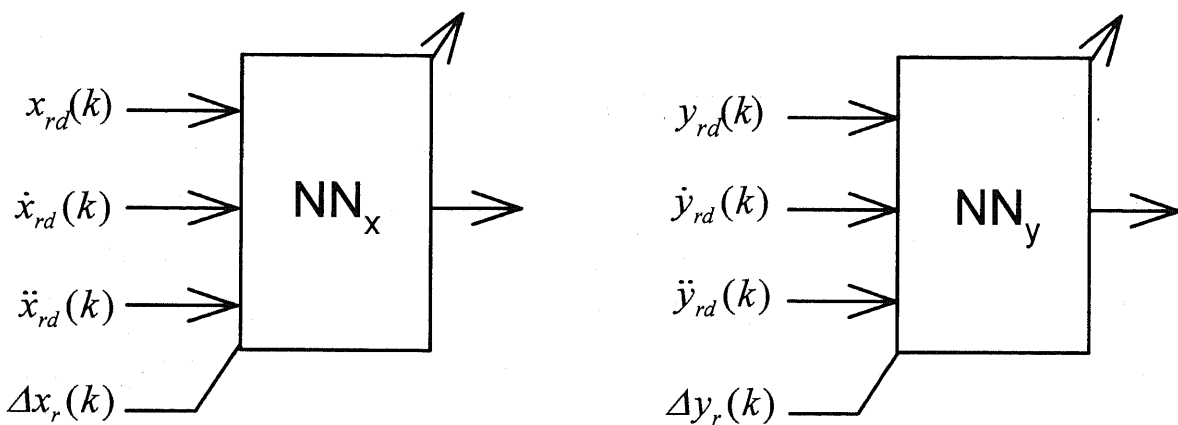


Fig. 4.6 Architecture of the proposed sub-neural networks

#### 4.5.2 ニューラルネットワークの学習

図4.6に示すサブニューラルネットワーク  $NN_x$  と  $NN_y$  は3層型NNとする。入力層をA、中間層をB、出力層をCとし、各層のニューロン数を  $N_A, N_B, N_C$  とする。また、学習に用いるシグモイド関数  $F(x)$  は

$$F(x) = \frac{1 - e^{-\frac{x}{\mu}}}{1 + e^{-\frac{x}{\mu}}} \dots \dots \dots (4.13)$$

で表される。ただし、 $\mu$  はアニーリングパラメータである。結合荷重の学習は誤差逆伝ば法により行われる<sup>(20)</sup>。

#### 4.5.3 オンライン学習型ニューラルネットワークを用いたハイブリッド制御

NNのオンライン学習を用いることによってフィードフォワード制御を実現し、繰り返し作業を行う場合追従精度を向上できることが知られている<sup>(21)</sup>。本章では、NN

のオンライン学習を用いて位置と力のハイブリッド制御を行うという手法を提案する。その制御ブロック線図を図4・7に示す。ただし、図4・7においては、

$$\text{選択行列 } \mathbf{S}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\text{選択ベクトル } \mathbf{s}_f = [0 \ 0 \ 1 \ 0 \ 0 \ 0]^T,$$

位置・姿勢制御ループのPIDコントローラ

$$\mathbf{G}_p(z) = \mathbf{K}_p^p + \mathbf{K}_I^p \frac{z}{z-1} + \mathbf{K}_D^p (1-z^{-1}),$$

力制御ループのPIDコントローラ

$$G_f(z) = K_p^f + K_I^f \frac{z}{z-1} + K_D^f (1-z^{-1}),$$

とする。

図4・7に示す制御系においては、ロボットの手先位置のフィードバック制御ループは、PIDコントローラ  $\mathbf{G}_p(z)$ 、ヤコビ逆行列、ロボットなど部分から構成されており、この制御ループが安定になるようPIDコントローラ  $\mathbf{G}_p(z)$  を設計する。また、図4・7に示すように、ツール先端の追従精度を向上させるために用いたオンライン学習型NNへの入力は、 ${}^b x$  軸と  ${}^b y$  軸方向における時刻  $t=kT$  でのロボット手先の目標位置  $x_{rd}(k), y_{rd}(k)$ 、手先の目標速度  $\dot{x}_{rd}(k), \dot{y}_{rd}(k)$ 、手先の目標加速度  $\ddot{x}_{rd}(k), \ddot{y}_{rd}(k)$  である。したがって、この制御系は、オンライン学習型NNによってフィードフォワード制御を行い、ロボット手先の追従精度を向上させている。

図4・7では、時刻  $t=kT$  において GENESIS の画像処理によってツールの先端位置  $\mathbf{p}_i^c(n-1)$  を検出し、式(4・1)により補間を行い、 $\mathbf{p}_i(k)$  を得る。ついで、式(4・2)～(4・4)から  ${}^b x$ - ${}^b y$  平面における直線  $L_0$  の投影ベクトル  $\Delta \mathbf{p}_L(k)$  を算出したのち、式(4・7)よりロボットの目標位置  $\mathbf{p}_{hd}(k)$  を生成する。また、式(4・12)のようにロボット手先の目標位置・姿勢の6自由度ベクトル  $\mathbf{p}_{rd}(k)$  を算出する。得られた  $\mathbf{p}_{rd}(k)$  と、ロボットの運動学  $\mathbf{A}$  によって求められたロボット手先の現在位置  $\mathbf{p}_r(k) \in R^{6 \times 1}$  との差  $\Delta \mathbf{p}_r(k)$  を計算したのち、 $\mathbf{S}_2 \cdot \Delta \mathbf{p}_r(k)$  から  ${}^b x$ 、 ${}^b y$  軸方向の位置偏差  $\Delta x_r(k), \Delta y_r(k)$  を求めてサブニューラルネットワーク  $\text{NN}_x, \text{NN}_y$  の修正量として用いる。

また、PIDコントローラ  $\mathbf{G}_p(z), G_f(z)$  によってロボット手先の目標速度  $\dot{\mathbf{p}}_{rd}(k)$  を算出する。逆ヤコビ行列  $\mathbf{J}^{-1}$  を用いてロボットの各関節の目標角速度  $\dot{\boldsymbol{\theta}}_r(k)$  を算出し、ロボットのサーボドライブユニットに入力させ、ロボットを制御する。また、図4・1に示すアームを制御するコンピュータでは、制御サンプリング周期  $T (=5\text{ms})$  の間にニューラルネットワークの学習を4回行うことが可能である。

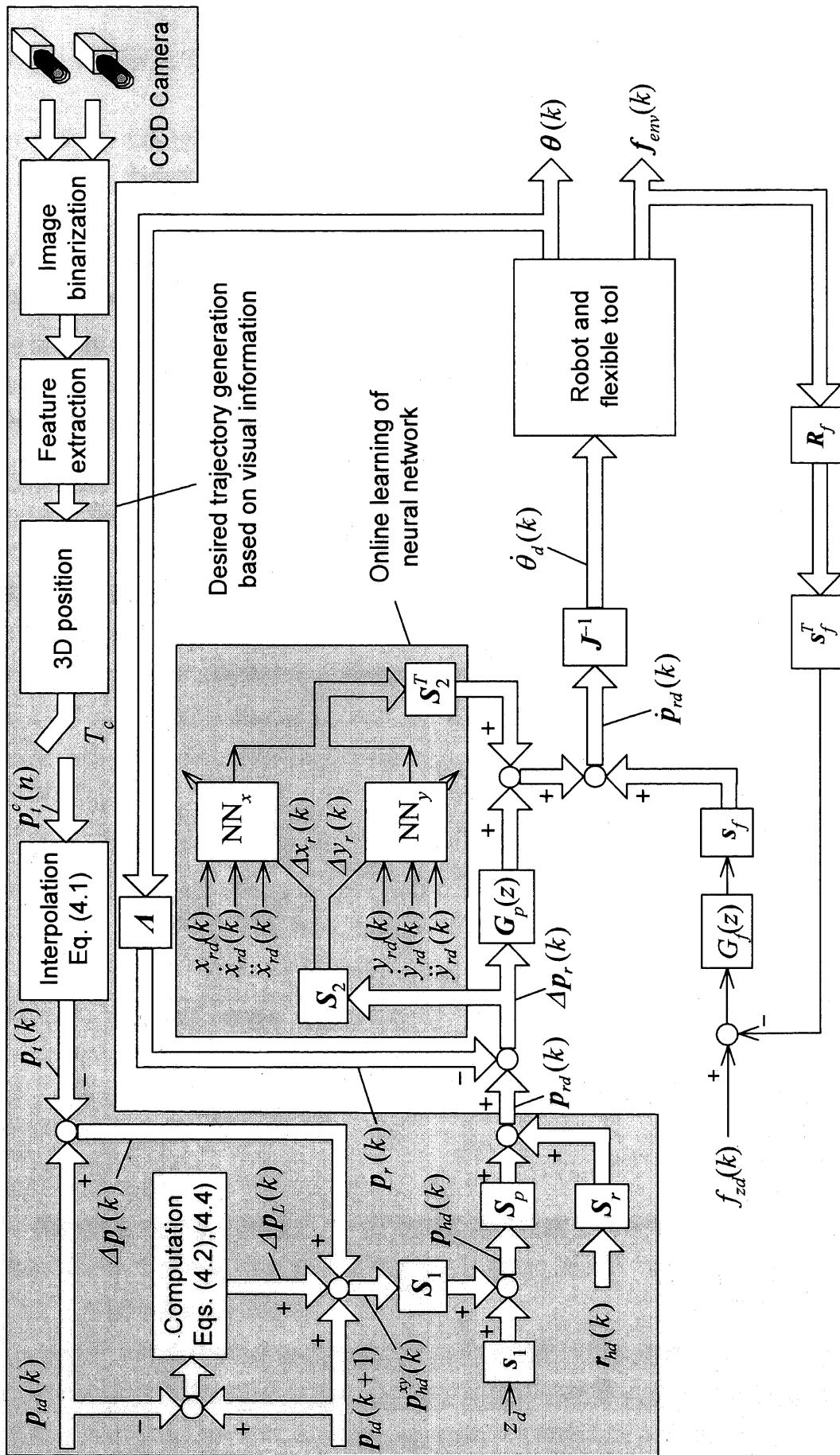


Fig. 4.7 The proposed position/force hybrid control issue with an online learning neural network

## 4.6 実験とその結果

### 4.6.1 ねじりモーメントによるツール先端の位置追従誤差の影響を確認する実験(姿勢制御なし, NNなし)

実験で与えられたロボット手先の目標軌道  $p_{rd}$  は図 4.8 に示すような半径  $R=100\text{mm}$  の円軌道である。摩擦力によって引き起こされたねじりモーメントがツール先端の追従精度に与える影響を確認するため、ロボット手先の姿勢を固定し、ロボット手先の位置制御を行う。この場合、剛体ツールを用いたときと同じように、ツールの長さをロボットの運動学で考慮することによってツール先端の目標軌道  $p_{td}$  を算出する。また、2台のカメラを用いて観測したツール先端の実際位置  $p_t$  と算出したツール先端の目標軌道  $p_{td}$  との誤差を計算し、ツール先端の追従効果を以下に検討する。

この実験で用いた制御手法は、図 4.7 の制御ブロック線図から、(1)視覚情報による目標軌道の生成、(2)NN学習の部分を除いたブロック線図で表される。この実験で用いた各パラメータを表 4.1 に示す。

Table 4.1 Control parameters used in the experiment

$K_p^p = \text{diag}[K_{p1}^p \quad K_{p1}^p \quad 0 \quad K_{p2}^p \quad K_{p2}^p \quad K_{p2}^p],$					
$K_{p1}^p = 5.0$		$1/\text{s},$	$K_{p2}^p = 2.0$		$1/\text{s},$
$K_I^p = \text{diag}[K_{I1}^p \quad K_{I1}^p \quad 0 \quad K_{I2}^p \quad K_{I2}^p \quad K_{I2}^p],$					
$K_{I1}^p = 0.5$		$1/\text{s},$	$K_{I2}^p = 0.2$		$1/\text{s},$
$K_D^p = \text{diag}[K_{D1}^p \quad K_{D1}^p \quad 0 \quad K_{D2}^p \quad K_{D2}^p \quad K_{D2}^p],$					
$K_{D1}^p = 0.1$		$1/\text{s},$	$K_{D2}^p = 0.1$		$1/\text{s}$
$K_p^f = 0.001 \quad \text{m}/(\text{s}\cdot\text{N}),$			$K_I^f = 0.0005 \quad \text{m}/(\text{s}\cdot\text{N}),$		
$K_D^f = 0.0001 \quad \text{m}/(\text{s}\cdot\text{N})$					
$z_d = 0,$	$\alpha_d = 0,$	$\beta_d = 1.222 \text{ rad},$	$\gamma_d = 0,$	$f_{zd} = 2 \text{ N}$	

得られたロボット手先とツール先端の追従軌跡を図 4.9 に示す。 $b_x$ - $b_y$  平面におけるロボット手先の位置  $p_h$  が目標軌道  $p_{hd}$  に追従したが、ツール先端の位置誤差が大きいことが図 4.9 から分かる。

スタート時点  $t=0$  において柔軟ツールの先端を作業台に押し付けていることによってツール自身の変形が生じるため、ツール先端の位置  $p_t$  が目標軌道  $p_{td}$  から外れる。また、ロボットの手先が点  $p_0 \rightarrow p_1 \rightarrow p_2$  に沿って移動しているとき、進行方向にツール

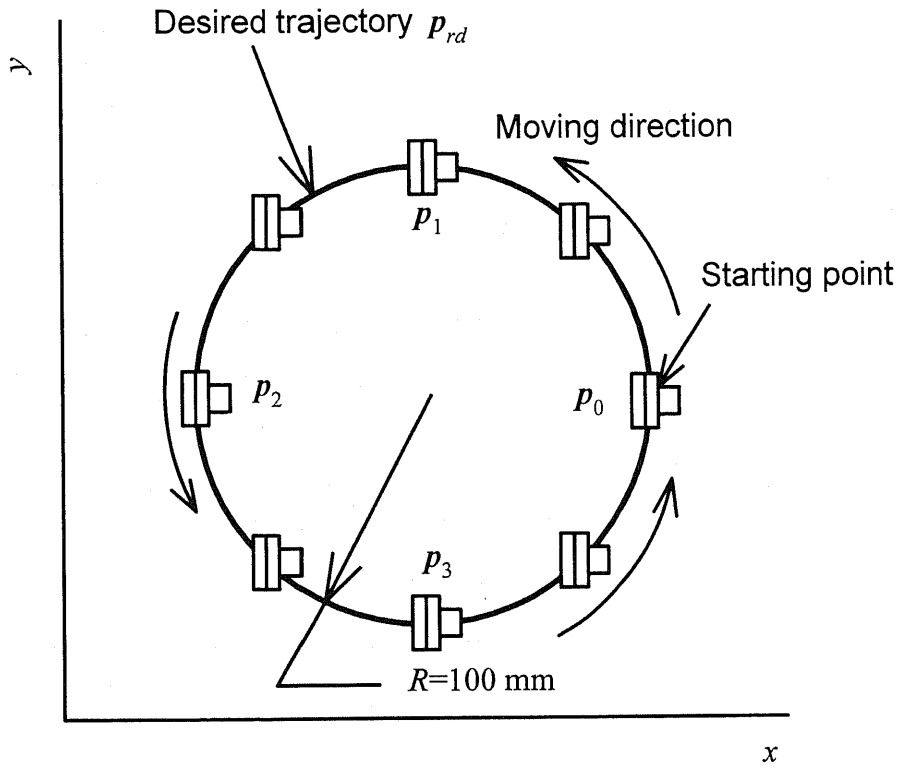


Fig.4·8 A circle trajectory in  $x$ - $y$  plane

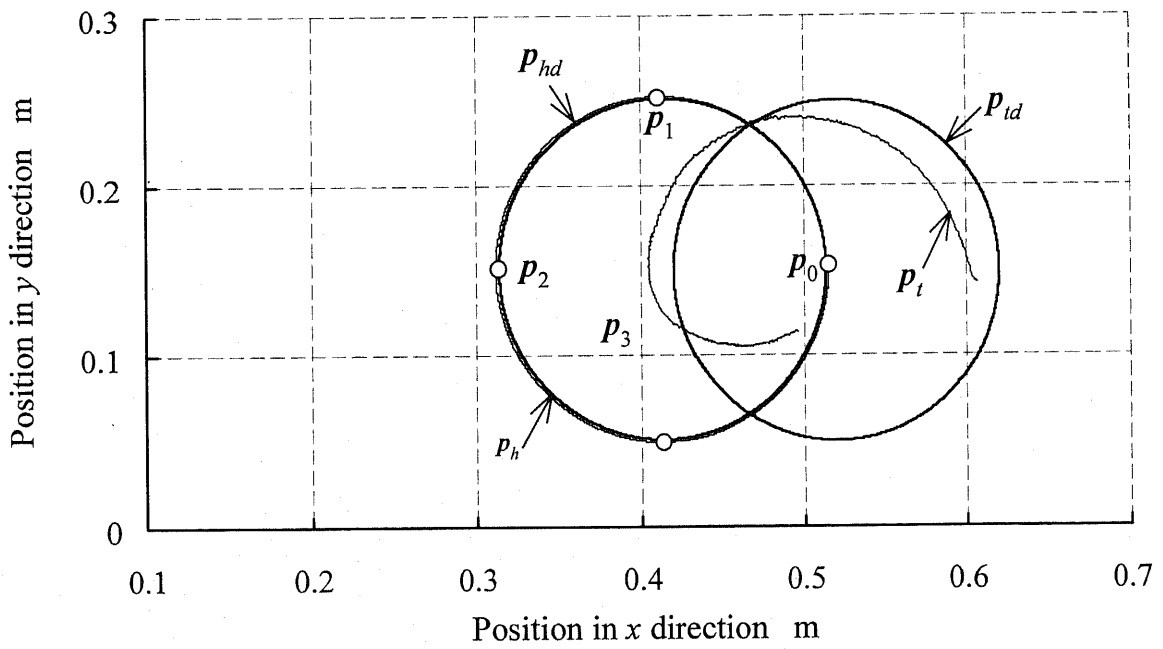


Fig.4·9 Tracking results in  $x$ - $y$  plane of the robot hand and the flexible tool

先端の先方に置かれたため、ツールを押し付けながら引くことができた。これに対し、ロボットの手先が点  $p_2 \rightarrow p_3 \rightarrow p_0$  に沿って移動しているとき、ロボットの手先は進行方向にツール先端の後方に置かれるため、ツールを押し付けて進める姿勢になる。この場合、摩擦力によってねじりモーメントが生じる。このねじりモーメントによってツール自身を回転させたため、作業の後半においてツール先端  $p_t$  は目標軌道  $p_{td}$  から大きく外れ、進行方向に進めることがほとんどできなかつたと考えられる。

#### 4.6.2 ツールの先端を目標軌道になぞらせる実験(NNなし)

図 4.5 に示すツール先端の摩擦による柔軟ツール自身のねじり回転を避けるため、図 4.6 に示すようにロボットの手先を進行方向において常にツール先端の先方に置く必要がある。しかしながら、この考え方を図 4.8 に示すような円軌道に適用すると、ロボット手先が  $z$  軸回り  $360^\circ$  回転させる必要があるが、マニピュレータの構造上から実現することは困難である。このため、図 4.10 に示す正弦波状曲線をツール先端の目標軌道とする。この実験に用いる制御手法は、図 4.7 の制御ブロック線図から NN のオンライン学習の部分を除いたブロック線図で表される。また、各パラメータの設定を表 4.2 に示す。

Table 4.2 Control parameters used in the experiment

$\mathbf{K}_p^p = \text{diag}[K_{p1}^p \quad K_{p1}^p \quad 0 \quad K_{p2}^p \quad K_{p2}^p \quad K_{p2}^p],$	
$K_{p1}^p = 5.0 \quad 1/s, \quad K_{p2}^p = 2.0 \quad 1/s,$	
$\mathbf{K}_I^p = \text{diag}[K_{I1}^p \quad K_{I1}^p \quad 0 \quad K_{I2}^p \quad K_{I2}^p \quad K_{I2}^p],$	
$K_{I1}^p = 0.5 \quad 1/s, \quad K_{I2}^p = 0.2 \quad 1/s,$	
$\mathbf{K}_D^p = \text{diag}[K_{D1}^p \quad K_{D1}^p \quad 0 \quad K_{D2}^p \quad K_{D2}^p \quad K_{D2}^p],$	
$K_{D1}^p = 0.1 \quad 1/s, \quad K_{D2}^p = 0.1 \quad 1/s$	
$K_p^f = 0.0010 \quad m/(s \cdot N),$	
$K_I^f = 0.0005 \quad m/(s \cdot N),$	
$K_D^f = 0.0001 \quad m/(s \cdot N)$	
$z_d = 0, \quad \alpha_d = 0, \quad \beta_d = 1.047 \text{ rad},$	
$f_{zd} = 0.5 \quad N$	

Table 4-3 Parameters of neural network learning

Neuron number of the input layer $N_a$	3
Neuron number of the hidden layer $N_b$	6
Neuron number of the output layer $N_c$	1
Annealing rate $\mu$	1.0
Learning rate $\eta$	0.85

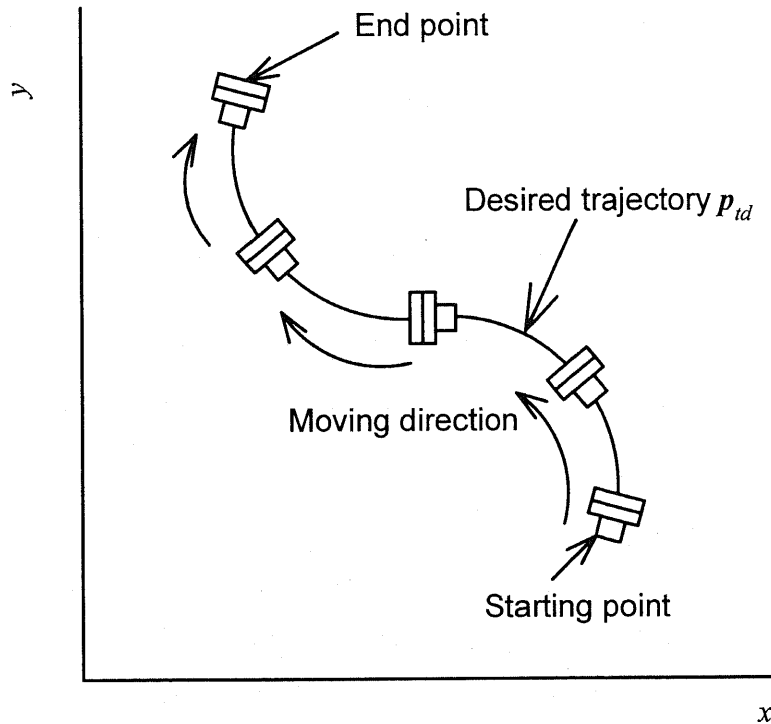


Fig.4-10 Desired trajectory of the flexible tool's tip

実験で得られた柔軟ツール先端の位置  $p_t$  とロボットハンドの位置  $p_h$  を図 4-11 に示す。図 4-11 では、進行方向に合わせてロボットの手先を  $b_z$  軸の回りに回転させたため、ねじりモーメントによるツール自身の回転という現象がなくなり、ツール先端に与えられた目標軌道  $p_{td}$  をなぞらせるという作業をほぼ遂行した。進行方向においてツール先端の前方にロボットハンドを置くという制御手法の有効性は、図 4-11 に示す結果により確かめられた。

一方、ロボット手先の姿勢を制御することによってツール先端  $p_t$  が目標軌道  $p_{td}$  に追従しているが、追従誤差は大きいということが図 4-11 から分かる。図 4-11 に示すように、 $p_t$  の矢印が指す部分において  $x, y$  軸方向にそれぞれ約 8mm, 12mm の追従誤差があった。



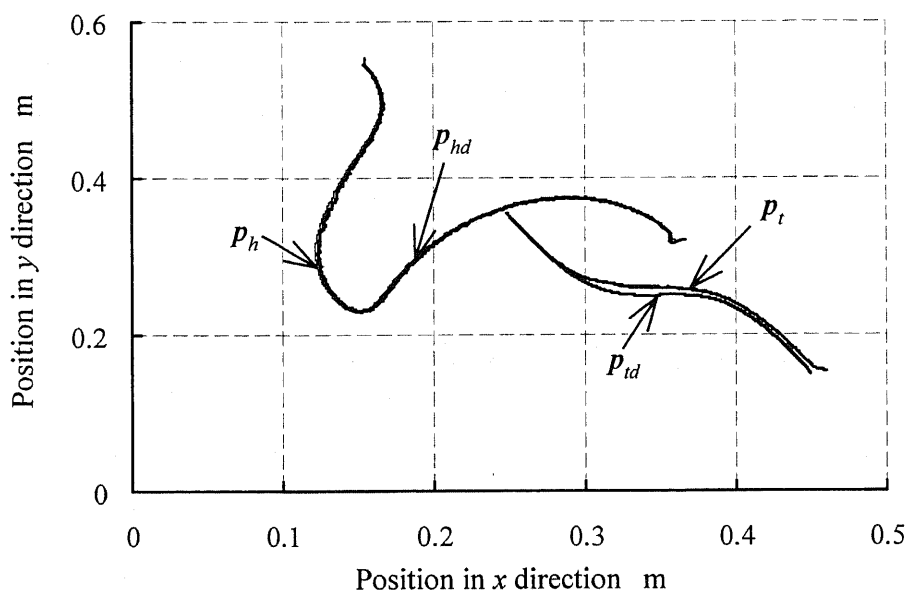


Fig.4.11 Tracking results in x-y plane of the robot hand and the flexible tool's tip

#### 4.6.3 ニューラルネットワークのオンライン学習を用いた追従実験

${}^b_x$ - ${}^b_y$  平面におけるツール先端の位置追従精度をさらに向上させるため、図 4.7 に提案したオンライン学習型 NN を用いた制御手法によって追従実験を行った。4.6.2 節の実験と同様に、図 4.10 に示す軌道をツール先端の目標軌道とする。このときの制御パラメータを表 4.2 に、NN のパラメータを表 4.3 に示す。

NN の結合荷重には、初回の作業の学習開始時点で乱数を与え、作業が終了したとき結合荷重のデータをファイルに保存する。2 回め以降の作業では、毎回作業が開始するとき前回に保存された結合荷重をファイルから読み込み、作業が終了するとき結合荷重の学習結果をファイルに保存する。このようにして学習の結果を蓄積することができ、繰り返し作業が進むにつれて追従精度が向上することを期待できる。

NN の学習に用いる誤差修正信号を図 4.12 に示す。一般に、学習誤差が収束するまで一定の学習回数が必要とされる<sup>(21)</sup>。この実験の場合、1 回めの作業開始の時点において  ${}^b_x$ - ${}^b_y$  軸方向の位置制御は主に PID 制御であり、位置追従誤差は大きい。約 10 000 回の学習が終了してから追従誤差は徐々に収束していることが図 4.12 から分かる。1 回めの実験で得られた結合荷重を 2 回めの実験に用いたため、追従誤差はより速く減少し、追従精度は向上した。

2 回めの実験で得られた  ${}^b_x$ - ${}^b_y$  平面の追従軌跡を図 4.13 に、 ${}^b_z$  軸方向の力を図 4.14

に示す。図 4・11 の追従軌跡と比べると、オンライン学習型 NN を用いた実験において追従誤差を大幅に減少したことが図 4・13 から分かる。特に図 4・13 では、 $p_t$  の矢印が指す部分において  $b_x, b_y$  軸方向の追従誤差の最大値は約 4mm であり、4・6・2 節での実験で得られた追従誤差の半分以下である。また、 $b_z$  軸方向での力は目標値 0.5 N の付近で変動したことが図 4・14 により分かる。提案した手法を用いて柔軟ツールの先端を目標軌道になぞらせる実験を遂行したことによってその有効性を実証した。

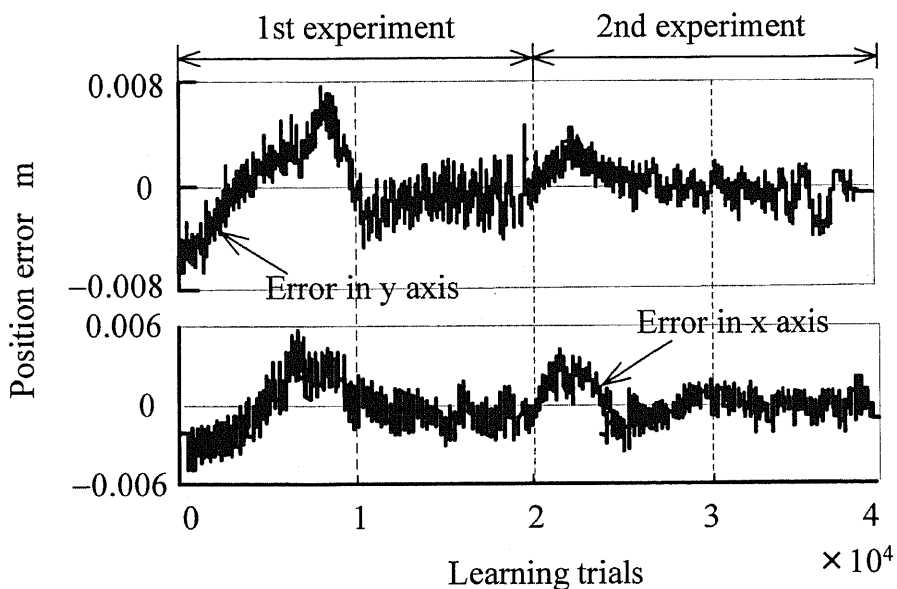


Fig.4・12 Error signals for NN learning

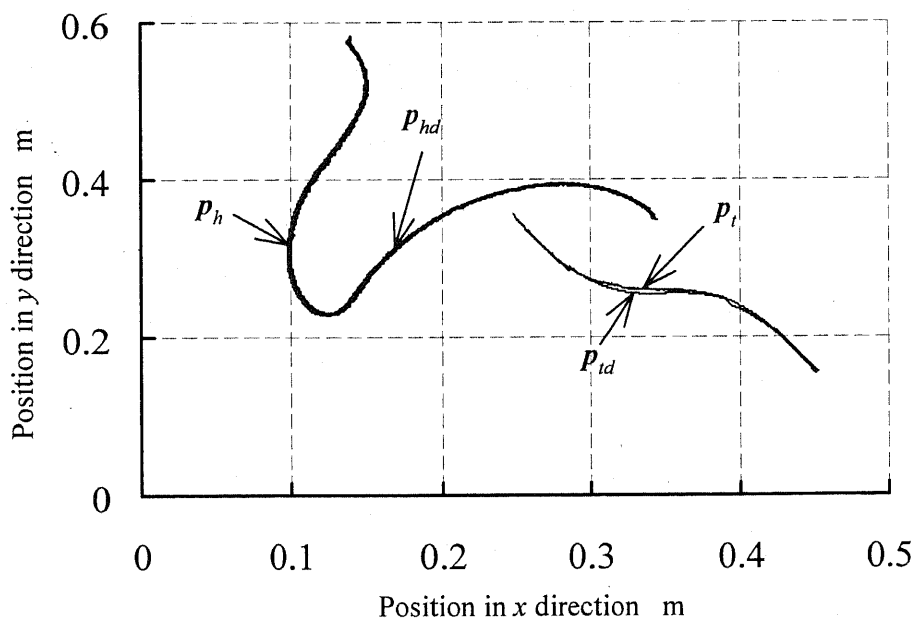


Fig.4・13 Tracking results in x-y plane of the robot hand and the flexible tool's tip

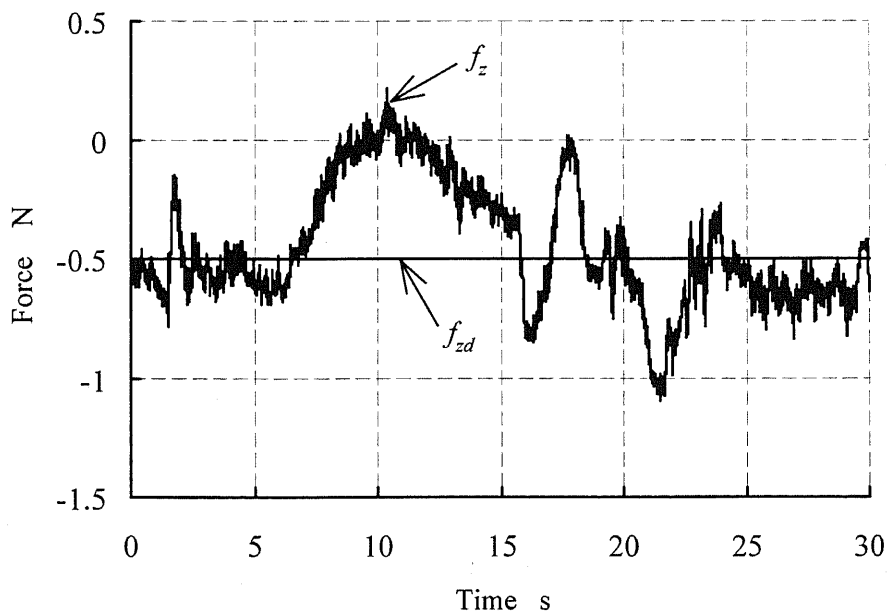


Fig.4·14 Results of force control in z axis

#### 4·7 要約

本章で得られた主な結果は、次のとおりである。

(1)ニューラルネットワークのオンライン学習を用いてマニピュレータがもつ柔軟ツールの先端の位置と力のハイブリッド制御を行うという手法を提案し、柔軟ツールの先端に曲線軌道をなぞらせる実験を行い、本手法の有効性を実証した。

(2)本章で提案した手法は、視覚情報を用いるため、柔軟ツールの変形モデルを立てることが困難な場合に適用できるという利点をもつ。実験によって本手法の有効性を確かめた。

---

## 第5章

# 冗長ロボットによる衝突の回避とその制御

---

### 5-1 概説

日常生活においてロボットの近くを通りかかる人間および近づいてくる障害物の位置を事前に予測できない場合が多い。このため、ロボットは人間や障害物と衝突し、事故につながる危険性は極めて高い<sup>(22)~(24)</sup>。このようなことを避けるため、冗長ロボットを用いて柔軟に対応することは期待される。汎用性の高い冗長ロボットは、人間の腕と同じように作業環境に応じて関節の冗長自由度を利用できるため、特異点の回避<sup>(25)~(26)</sup>、障害物の回避<sup>(27)~(32)</sup>、インピーダンス制御<sup>(33)~(36)</sup>に広く用いられてきた。

しかしながら、これまでの衝突の回避に関する研究のほとんどは、障害物の位置が既知で静止しているという条件のもとで開発された手法である。実際には、この条件が成立しない場合が多い。このような場合においても障害物との接触は回避しなければならない。それは、接触によって発生した衝撃力が装置を破壊する恐れのある作業では、接触力を検出してからインピーダンス制御を始めるのでは遅いからである。障害物とロボットとの接触を回避するため、未来の目標軌道とその形状情報をロボットの冗長制御に用いる予見制御法<sup>(37)</sup>が報告された。また、複数の移動ロボットを用いた作業の場合、移動ロボット同士またはロボットと障害物の間に仮想的なインピーダンスを設けて協調作業を実現する手法<sup>(38)</sup>も報告されている。

近年、障害物の回避制御に視覚センサを用いたアプローチが多くなってきた。これまでに、障害物の回避に必要な画像特徴の推定<sup>(39)</sup>、三次元再構成を行わない視覚サーボ系を用いた障害物の回避軌道の生成<sup>(40)</sup>などの報告があった。また、視覚情報を用いて対象物の状態によってビジュアルコンプライアンス制御を行うという手法<sup>(41)</sup>、カメラ1台を用いて平面冗長ロボットの手先と対象物とのインピーダンスを制御する手法<sup>(42)</sup>、ニューラルネットによる非接触型インピーダンス制御のパラメータの学習<sup>(43)</sup>が報告された。

作業中のロボットに接近してくる人間や障害物など衝突可能な対象が現れるとき、緊急の場合を除いてロボットの手先の作業を直ちに中止し回避制御を行うのではな

く、ロボットは手先の作業を行いながら近づいてくる人間や障害物との衝突を可能な限り柔軟に回避することが望ましい。また、作業を行う際に特異点の近辺での作業をできるだけ避けることが必要であるため、手先の作業と衝突の回避作業を両立させると同時にロボットの可操作性を考慮する必要がある。しかしながら、これまでの研究では、この問題についての十分な検討がなされたとはいえない。そこで本章では、上述の目的を実現するため、まずロボットの周辺に仮想ポテンシャル場を設け、仮想ポテンシャル場に入った衝突対象を立体視覚センサによって検出したとき、仮想トルクを発生させるという手法を提案する。また、衝突の回避とロボットの可操作性を考慮した評価関数を提案し、衝突を柔軟に回避しながら手先の倣い作業を遂行する制御系を構成する。提案した手法を用いて倣い作業をさせながらロボットに近づいてくる衝突物体を回避する実験を行い、その効果を実証する。

## 5.2 立体視画像処理による衝突対象の位置の検出

ロボットに近づいてくる衝突可能な対象の三次元空間の位置を検出するため、2台のCCDカメラ CV-M40 (JAI 社)によって立体視観測システムを構成する。また、この2台のカメラから毎秒 60 フレームの速度で映像信号を出力し、画像ボード GENESIS (GEN/F/64/8 /STD, Matrox 社)の入力端子に入力する。

本章では、第2章の図2.3(b)に示す画像の並列処理という手法を用いるため、カメラから連続的に出力されている画像を見落とすことなくリアルタイムに処理することができる。したがって、画像の入力・処理のサンプリング周期  $T_c$  は 16.67 ms である。一方、ロボットの制御サンプリング周期  $T$  は 5ms であるため、画像処理で検出した衝突対象の位置  $p_b^c$  を  $T$  ごとに補間することは必要である。

本章では、衝突対象は滑らかな曲線に沿って移動すると仮定したため、ロボットの制御サンプリング周期ごとの衝突対象の移動は直線と見なすことができる。ここで、 $T$  と  $T_c$  の最小公倍数 (Least common multiple, 以下 lcm と略称) を  $T_n$  とし、 $n_d = T_n/T$ ,  $n_c = T_n/T_c$  とする。  $t = in_d T$  において、図5.1に示すように画像処理から衝突対象の位置  $p_b^c(in_d T - T_c)$  をすでに得ているため、ロボットの制御サンプリング周期ごとに衝突対象の移動軌道の補間を

$$p_b(k) = \left. \begin{array}{l} p_b^c(in_d T + (j-1)T_c) \\ 0 \leq kT - (in_d T + jT_c) < T \\ \dot{p}_b(k-1) \cdot T + p_b(k-1) \\ kT - (in_d T + jT_c) > T \\ (j=0,1,\dots,n_c-1), (in_d \leq k < (i+1)n_d) \end{array} \right\} \dots\dots\dots (5.1)$$

によって行う。ただし、

$$T_n = \text{lcm}(T_c, T) \quad \dots \dots \dots (5.2)$$

$$\dot{p}_b(k-1) = \frac{p_b(k-1) - p_b(k-2)}{T} \quad \dots \dots \dots (5.3)$$

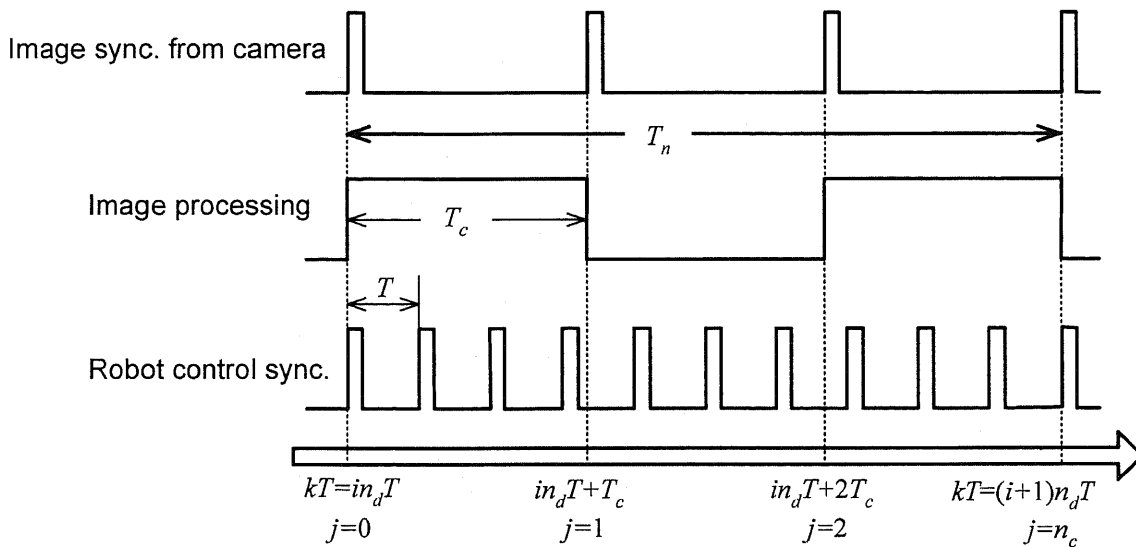


Fig.5.1 Synchronization of image processing and robot control ( $n_d=10, n_c=3$ )

### 5.3 仮想関節トルクによる冗長関節の目標軌道の生成

手先の作業に必要な自由度を  $m$ ，ロボットの関節数を  $n$  とする。一般に，手先の作業を行うために必要なロボットの関節を非冗長関節，余分となる関節を冗長関節と呼ぶ。すなわち，冗長関節数  $r$  は  $n-m$  になる。本章では，ロボットが人間や障害物と接触する前に，コンプライアンス制御を冗長関節に適用して衝突の回避を実現する。このため，まずマニピュレータの周囲に仮想ポテンシャル場を設ける。ついで，衝突対象が仮想ポテンシャル場に入ったとき衝突対象に最も近いリンクで仮想外力を発生させ，その力を各関節のトルクに振り分けることによって仮想関節トルクを生成させる。生成された仮想関節トルクを用いてコンプライアンス制御を行い，衝突の回避を柔軟に実現する。

#### 5.3.1 仮想ポテンシャル空間の設定

図 5.2 に示すように本章では，ロボットの周辺にリンクを中心軸とした半径  $r_v$  の円筒を仮想ポテンシャル空間として設定する。

また，時刻  $t=kT$  に仮想ポテンシャル空間に人間や障害物が入ったとき，その進入位置ベクトル  $e_v(k)$  を

$$e_v(k) = p_b(k) - p_v(k) \quad \dots \dots \dots (5.4)$$

により算出する. ここで, 図 5・2 に示す直線 AB と仮想ポテンシャル円筒表面との交点を位置ベクトル  $p_v(k)$  の先端とする (ベクトルの始点は  $\Sigma_b$  の原点). また,  $e_v(k)$  は常にリンクの中心に向かうベクトルである.

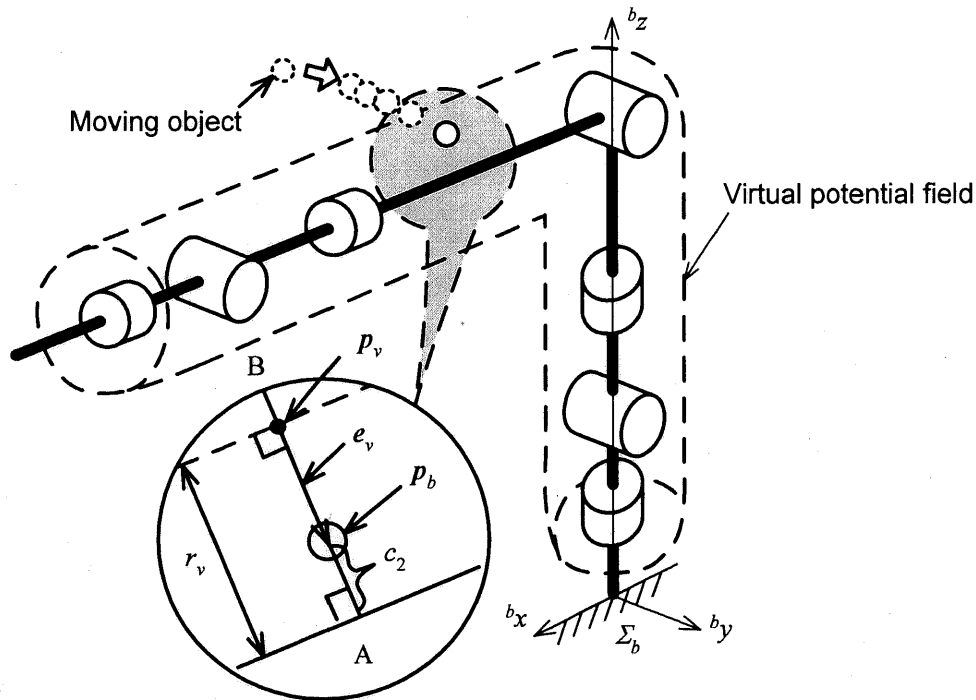


Fig. 5・2 Proposed virtual potential field for a 7' DOF redundant robot Pa10

### 5・3・2 仮想関節トルクの生成

図 5・3 に示すように, 仮想ポテンシャル空間内に進入する衝突対象を検出したこと  
 によって発生させる仮想外力  $f_v(k)$  は

$$f_v(k) = D_v \dot{e}_v(k) + K_v e_v(k) \dots \dots \dots (5.5)$$

で表される. ただし,

- $f_v(k) \in R^{3 \times 1}$  : 発生させる仮想外力
- $D_v \in R^{3 \times 3}$  : 仮想ポテンシャル空間の粘性行列
- $K_v \in R^{3 \times 3}$  : 仮想ポテンシャル空間の剛性行列

また, 式(5.5)で得られた仮想外力  $f_v(k)$  がマニピュレータに加わる時, 各関節へ  
 分配される仮想トルク  $\tau_v \in R^{n \times 1}$  は,

$$\tau_v = \begin{bmatrix} J_{tb}^T \\ J_{tf}^T \end{bmatrix} f_v \dots \dots \dots (5.6)$$

によって求められる<sup>(44)</sup>. ただし,

- $J_{tb}$  : 図 5・3 に示す仮想外力の掛かるリンクより根元側のヤコビ行列
- $J_{tf}$  : 図 5・3 に示す仮想外力の掛かるリンクより手先側のヤコビ行列

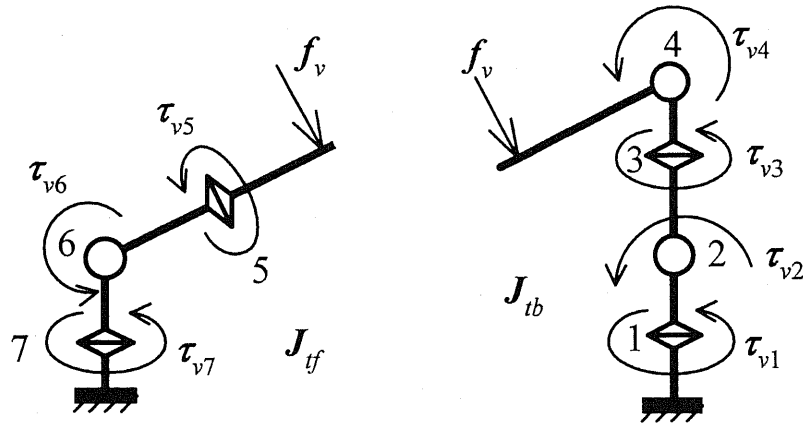


Fig. 5.3 Generation of a virtual torque  $\tau_v(k)$

### 5.3.3 冗長関節の目標軌道の生成

時刻  $t=kT$  にでポテンシャル場に進出した衝突対象の位置を検出することによって発生する仮想トルク  $\tau_v(k)$  を式(5.6)から求め、冗長関節に掛かるトルク  $\tau_{vr}(k) \in R^{r \times 1}$  を

$$\tau_{vr}(k) = S_r \tau_v(k) \dots \dots \dots (5.7)$$

によりを算出する。ここで、 $S_r \in R^{r \times n}$  は選択行列である。また、衝突を回避するため、冗長軸の目標軌道を関節コンプライアンス制御により求める。冗長関節における制御目標値  $\dot{\theta}_r(k+1) \in R^{r \times 1}$  は

$$\dot{\theta}_r(k+1) = (I - TM_r^{-1} D_r) \dot{\theta}_r(k) + TM_r^{-1} \tau_{vr}(k) + TM_r^{-1} K_r (\theta_{rd}(k) - \theta_r(k)) \dots (5.8)$$

で与えられる。ただし、

- $\dot{\theta}_r(k) \in R^{r \times 1}$  : 冗長関節の角速度
- $\theta_{rd}(k) \in R^{r \times 1}$  : 冗長関節の目標角度
- $\theta_r(k) \in R^{r \times 1}$  : 冗長関節の角度
- $M_r \in R^{r \times r}$  : 仮想質量行列
- $D_r \in R^{r \times r}$  : 仮想粘性行列
- $K_r \in R^{r \times r}$  : 仮想ばね行列
- $I \in R^{r \times r}$  : 単位行列

式(5.7)で  $S_r$  により選択された冗長関節に対してのみ仮想関節トルクを発生させ、式(5.8)に示すコンプライアンス制御を適用する。

## 5.4 冗長ロボットの制御

一般に、作業に必要な自由度以上の関節数を持つ冗長ロボットを制御する場合、ヤ



コピ擬似逆行列がよく用いられる。これに対して本章では、馬ら<sup>(31)(32)</sup>によって提案されたヤコビ行列を冗長関節と非冗長関節に分解するという概念を用いて、非冗長関節に手先 6 自由度の倣い作業を遂行させながら、冗長関節に衝突回避を行わせた。

#### 5・4・1 手先のコンプライアンス制御

時刻  $t=kT$  における手先の現在位置を  $p_h(k) \in R^{m \times 1}$ 、目標位置を  $p_{hd}(k) \in R^{m \times 1}$  とする。仮想コンプライアンス制御法を用いて手先に倣い作業を行わせる場合、手先速度の制御目標値  $\dot{p}_h(k+1) \in R^{m \times 1}$  は

$$\dot{p}_h(k+1) = (I - M^{-1}DT)\dot{p}_h(k) + TM^{-1}\Delta f(k) + TM^{-1}K\Delta p_h(k) \cdots \cdots \cdots (5 \cdot 9)$$

から求められる。ただし、

- $\dot{p}_h(k) \in R^{m \times 1}$  : 手先の速度
- $M \in R^{m \times m}$  : 仮想質量行列
- $D \in R^{m \times m}$  : 仮想粘性行列
- $K \in R^{m \times m}$  : 仮想剛性行列
- $\Delta f(k) \in R^{m \times 1}$  : 目標力  $f_d(k)$  と力センサの出力  $f_e(k)$  の差,  $\Delta f(k) = f_d(k) - f_e(k)$
- $\Delta p_h(k) \in R^{m \times 1}$  :  $\Delta p_h(k) = p_{hd}(k) - p_h(k)$
- $I \in R^{m \times m}$  : 単位行列

一般に、時刻  $t=kT$  における手先の速度  $\dot{p}_h(k)$  と関節角速度  $\dot{\theta}_n(k) \in R^{n \times 1}$  との関係は、

$$\dot{p}_h(k) = J_n \cdot \dot{\theta}_n(k) \cdots \cdots \cdots (5 \cdot 10)$$

で表される。ここで、 $J_n \in R^{m \times n}$  はヤコビ行列であり、冗長自由度  $r$  は

$$r = n - m \cdots \cdots \cdots (5 \cdot 11)$$

である。式(5・10)を冗長自由度と非冗長自由度に対応する項に分解すると、

$$\dot{p}_h(k) = J_m \cdot \dot{\theta}_m(k) + J_r \cdot \dot{\theta}_r(k) \cdots \cdots \cdots (5 \cdot 12)$$

が得られる。ただし、

- $J_m \in R^{m \times m}$  : 非冗長関節対応のヤコビ行列
- $\dot{\theta}_m(k) \in R^{m \times 1}$  : 非冗長関節の角速度
- $J_r \in R^{m \times r}$  : 冗長関節対応のヤコビ行列
- $\dot{\theta}_r(k) \in R^{r \times 1}$  : 冗長関節の角速度

である。ここで、冗長関節の角速度  $\dot{\theta}_r(k)$  を用いて式(5・12)により非冗長関節の角速度  $\dot{\theta}_m(k)$  は

$$\dot{\theta}_m = J_m^{-1} \cdot \dot{p}_h - J_m^{-1} \cdot J_r \cdot \dot{\theta}_r \cdots \cdots \cdots (5 \cdot 13)$$

で表される

式(5.13)は、 $\dot{\theta}_r(k+1)$ と $\dot{p}_h(k+1)$ が式(5.8), (5.9)から求められたとき、この両者を満たすことが可能な非冗長関節の関節角速度 $\dot{\theta}_m(k+1)$ を計算できることを示している。したがって、時刻 $t=(k+1)T$ における全関節の目標角速度 $\dot{\theta}_n(k+1)$ は、

$$\dot{\theta}_n(k+1) = I_m \cdot \dot{\theta}_m(k+1) + I_r \cdot \dot{\theta}_r(k+1) \cdots \cdots \cdots (5.14)$$

から求められる。ただし、

$I_m \in R^{n \times m}$ : 単位行列から冗長となる関節番号の列を除いた行列

$I_r \in R^{n \times r}$ : 単位行列から非冗長となる関節番号の列を除いた行列

### 5.4.2 評価関数を用いた冗長関節の選定

式(5.14)によって冗長関節と非冗長関節に数式上の振り分けができる。しかしながら、接近してくる衝突対象の方向と位置が不特定であるため、マニピュレータのどの関節を冗長関節として回避制御を行うかを定める必要がある。本章では、時刻 $t=kT$ における評価関数 $P(k)$ を

$$P = w_1 c_1(\theta(k)) + w_2 c_2(\theta(k)) \cdots \cdots \cdots (5.15)$$

で定義し、これを用いて冗長関節を決定するという手法を提案する。ただし、 $w_1, w_2$ は重み係数である。 $c_1(\theta(k))$ は時刻 $t=kT$ における可操作度<sup>(45)</sup>であり、

$$c_1(\theta(k)) = \sqrt{\det(J_n(\theta(k)) \cdot J_n^T(\theta(k)))} \cdots \cdots \cdots (5.16)$$

によって算出される。また図5.2に示すように、時刻 $t=kT$ における衝突対象の中心位置から着目している冗長リンクの中心軸までの距離を $c_2(k)$ とし、

$$c_2(\theta(k)) = r_v - |e_v| \cdots \cdots \cdots (5.17)$$

より求める。ただし、該当リンクが仮想ポテンシャル空間に入らない場合は、 $c_2=0$ とする。

また、どの作業を重視するかによって式(5.15)の重み係数 $w_1, w_2$ の値を決定する。本章では、取りうる冗長関節に対して式(5.15)の計算を行い、評価関数 $P$ が最も大きくなる関節を冗長関節として決定するという手法を用いる。

具体的には、冗長関節を決定するため、まず時刻 $t=kT$ にて式(5.14)で算出した目標関節角速度 $\dot{\theta}_n(k+1)$ を用いて関節角度の推定値 $\hat{\theta}_n(k+1)$ を

$$\hat{\theta}_n(k+1) = \theta_n(k) + \dot{\theta}_n(k+1) \cdot T \cdots \cdots \cdots (5.18)$$

から求める。したがって、時刻 $t=(k+1)T$ における評価関数の推定値 $\hat{P}(k+1)$ は、式(5.15), (5.16), (5.17)から

$$\hat{P}(k+1) = w_1 c_1(\hat{\theta}_n(k+1)) + w_2 c_2(k+1) \cdots \cdots \cdots (5.19)$$

となる。式(5.19)の計算は、着目している冗長リンクに対する $\theta_n(k+1)$ の予測値

$\hat{\theta}_n(k+1)$  を用いて式(5.16)から  $c_1(\hat{\theta}_n(k+1))$  を求め、衝突対象の予測位置  $\hat{P}(k+1)$  を用いて式(5.4),(5.17)により  $c_2(k+1)$  を算出して行う。複数の冗長軸を取りうる場合は  ${}_nC_m$  通りの組み合わせが考えられる。これらの組み合わせのすべてについて計算した評価関数の値が最大となる場合の  $\hat{\theta}_n(k+1)$  を制御系の目標関節角速度とする。以上の計算を各制御サンプリング時点で行う。このようにして手先の作業と衝突の回避作業を両立させる。

### 5.4.3 冗長ロボットの制御系の構成

本章で用いるマニピュレータ PA10 (三菱重工 (株)) は 7 関節を有するため、関節数  $n$  は 7, 作業自由度  $m$  は 6, 冗長関節数  $r$  は 1 となる。この場合、式(5.7)により得られる冗長関節に掛かるトルク  $\tau_{vr}(k)$  は

$$\tau_{vr}(k) = \mathbf{s}_r^T \boldsymbol{\tau}_v(k) \dots\dots\dots (5.20)$$

となる。ただし、 $\mathbf{s}_r \in R^{7 \times 1}$  は冗長軸を選択するベクトルであり、このベクトルにおいては、第  $r$  番目の要素のみ 1 で、それ以外の要素は 0 である。また、式(5.8)で表される冗長関節の制御目標値  $\dot{\theta}_r(k+1)$  は

$$\dot{\theta}_r(k+1) = (1 - TM_r^{-1} D_r) \dot{\theta}_r(k) + TM_r^{-1} \tau_{vr}(k) + TM_r^{-1} K_r (\theta_{rd}(k) - \theta_r(k)) \dots (5.21)$$

となる。手先での倣い作業と障害物の回避作業を実現するための制御ブロック線図を図 5.4 に示す。

図 5.4 においては、あらかじめ設定した仮想ポテンシャル場に衝突対象が進入したことを視覚情報の処理によって検出した場合、その進入ベクトル  $\mathbf{e}_v(k)$  を式(5.4)から計算し、式(5.5)から仮想外力  $\mathbf{f}_v(k)$  を、式(5.20)から仮想トルク  $\tau_{vr}(k)$  を算出する。また、手先のコンプライアンス制御の目標値  $\dot{\mathbf{p}}_h(k+1)$ 、衝突を回避するための関節コンプライアンスの制御目標値  $\dot{\theta}_r(k+1)$  をそれぞれ式(5.9), (5.21)によって生成したのち、式(5.19)によって評価関数の推定値  $\hat{P}(k+1)$  を計算し、冗長軸を選ぶ。ただし、ロボット PA10 を用いる場合、冗長関節数は 1 である。また、付録 A の図 A-1 に示すようにロボット PA10 の構造上では、S1 軸と S3 軸の回転方向は同じであるため、S3 軸を冗長軸とする。また、得られた目標角速度  $\dot{\theta}_n(k+1)$  をサーボドライバに内蔵されるジョイントサーボ  $\mathbf{G}_c(z)$  に入力させ、モータの制御を行う。

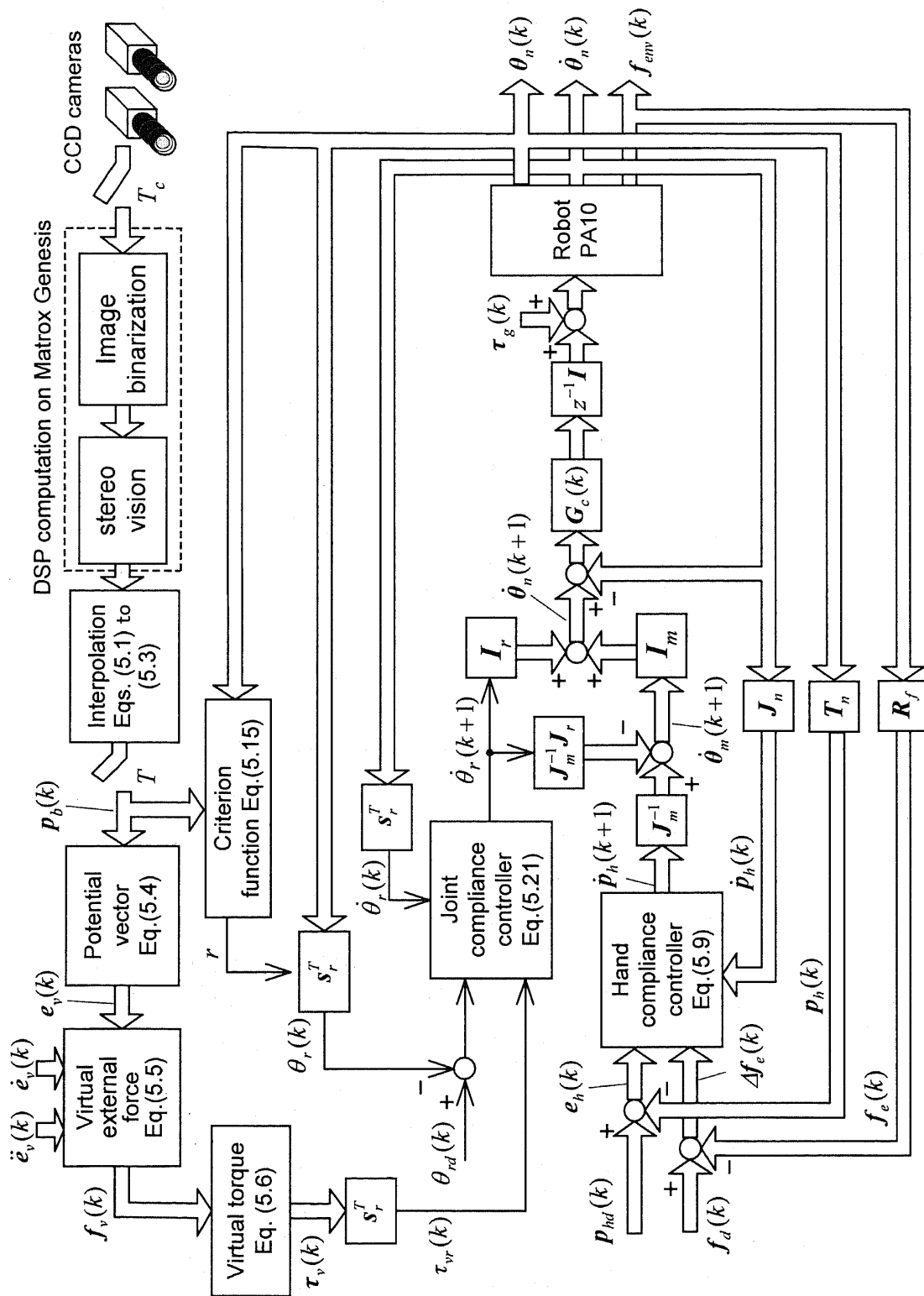


Fig. 5.4 The proposed control algorithm of a 7-DOF robot Pa10 ( $n=7$ ,  $m=6$ ,  $r=1$ )

## 5.5 実験とその結果

実験では半径  $r_b$  の発光球体を衝突対象として用いる。実験装置の配置を図 5.5 に示す。発光球体を予め用意した棒の先端に取り付け、実験実施者が棒を握って発光球体をロボットの S3 関節に接近していく。座標系  $\Sigma_b$  の原点から  ${}^b y$  軸方向に約 1 メートル離れた点の上方に 2 台のカメラを取り付ける。実験に用いた共通のパラメータを表 5.1 に示す。本来、現場の作業状況に応じて式(5.5)で用いた  $D_v$ ,  $K_v$  を決定するが、本実験では、回避制御が振動的にならないように  $D_v$ ,  $K_v$  の値を決定した。また、冗長関節のコンプライアンス制御とロボット手先のコンプライアンス制御の各パラメータについては、ロボット挙動が安定になるように試行錯誤によって選ぶ。

Table 5.1 Common parameters used in the experiments

$r_b = 0.05$ m,	$r_v = 0.3$ m	
$K_v = \text{diag}[2$	2	2] N/m
$D_v = \text{diag}[1$	1	1] N·s/m
$T = 0.005$ s,	$T_c = 0.0167$ s,	$T_n = 0.05$ s
$n_d = 10$ ,	$n_c = 3$	
Weight coefficients defined in Eq. (5.19) :		
$w_1 = 1.0$ ,	$w_2 = 0.0$	for no obstacle
$w_1 = 0.2$ ,	$w_2 = 0.8$	for obstacle detected

Table 5.2 Parameters used in the 1st experiment

$M = \text{diag}[m_{h1}$	$m_{h1}$	$m_{h1}$	$m_{h2}$	$m_{h2}$	$m_{h2}]$ ,
$m_{h1} = 0.5$ kg,	$m_{h2} = 0.5$ kg·m <sup>2</sup>				
$D = \text{diag}[d_{h1}$	$d_{h1}$	$d_{h1}$	$d_{h2}$	$d_{h2}$	$d_{h2}]$ ,
$d_{h1} = 110$ N·s/m,	$d_{h2} = 110$ N·m·s/rad				
$K = \text{diag}[K_{h1}$	$K_{h1}$	0	$K_{h2}$	$K_{h2}$	$K_{h2}]$ ,
$K_{h1} = 400$ N/m,	$K_{h2} = 400$ N·m /rad				
$M_r = \text{diag}[m_{j1}$	$m_{j1}$	$m_{j1}$	$m_{j2}$	$m_{j2}$	$m_{j2}]$ ,
$m_{j1} = 0.2$ kg,	$m_{j2} = 0.2$ kg·m <sup>2</sup>				
$D_r = \text{diag}[d_{j1}$	$d_{j1}$	$d_{j1}$	$d_{j2}$	$d_{j2}$	$d_{j2}]$ ,
$d_{j1} = 4$ N·s/m,	$d_{j2} = 4$ N·m·s/rad				
$K_r = \text{diag}[0$	0	0	0	0	0]
$p_{hd} = [p_x$	$p_y$	$p_z$	$R_x$	$R_y$	$R_z]^T$ ,
$p_x = 0.75$ m,	$p_y = 0$ m,	$p_z = 0.5$ m			
$R_x = 0$ rad,	$R_y = \pi/3$ rad,	$R_z = 0$ rad			
$f_d = [f_1$	$f_1$	$f_1$	$f_2$	$f_2$	$f_2]^T$ ,
$f_1 = 0$ N,	$f_2 = 0$ N·m				

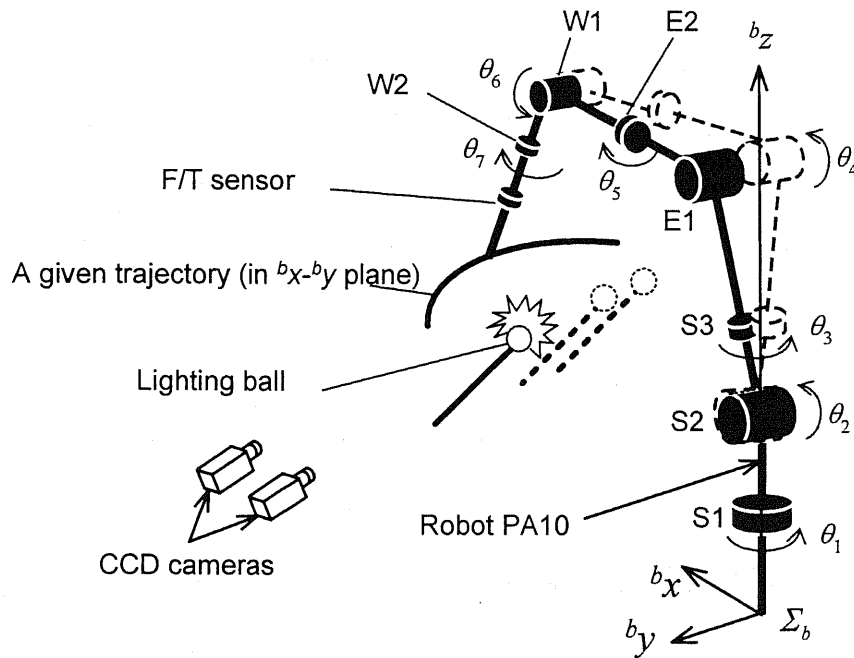


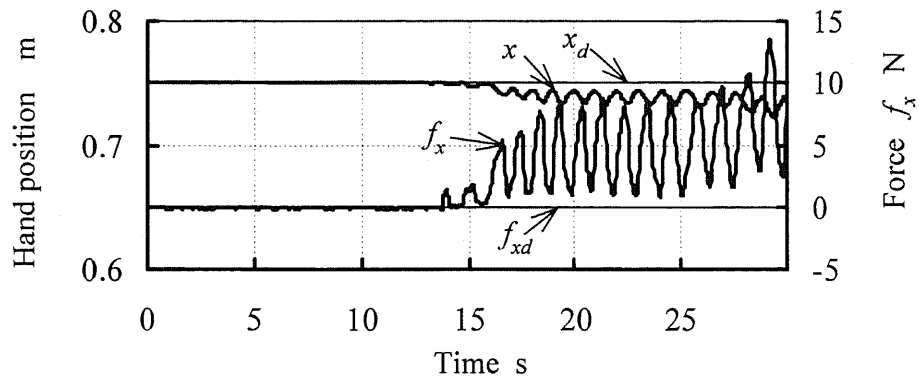
Fig.5.5 Setup of a collision avoidance experiment

### 5.5.1 実験1(衝突物体を回避しながら手先でのコンプライアンス特性を保つ実験)

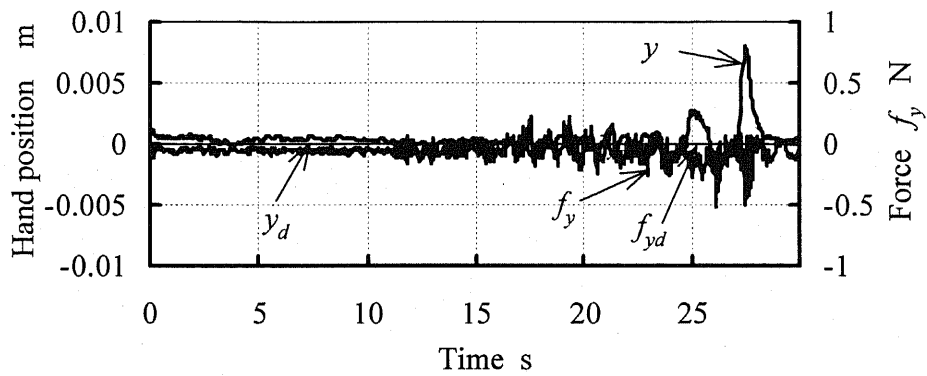
本実験では、衝突物体の回避を行うとともに、手先に力を加えたとき手先でのコンプライアンス特性が保たれていることを確認する、本実験で用いた制御パラメータ、目標位置、目標力を表5.2に、実験の結果を図5.6に示す。

図5.6(d)においては、式(5.19)の $c_2$ が仮想ポテンシャル場の境界線より下回ることを視覚センサによって5回ほど検出したため、衝突物体は仮想ポテンシャル空間に5回に進入したことが分かった。また、図5.6(e)から衝突物体の進入に合わせて各関節が素早く回避する方向に動いたことが分かる。また、提案した手法を用いて衝突を回避したことが図5.7に示すビデオ画面によって確認された。

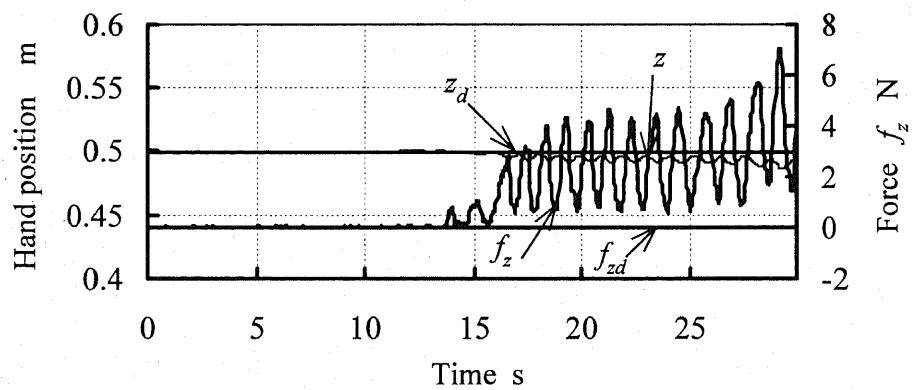
図5.6(a), (b), (c)から、外力を手先に加えると手先がそれに応じて柔軟に動いたため、コンプライアンス制御の特性を保っていることを確かめた。また、提案した本手法によって手先の運動と衝突の回避が独立に行われたことを図5.7により確認した。



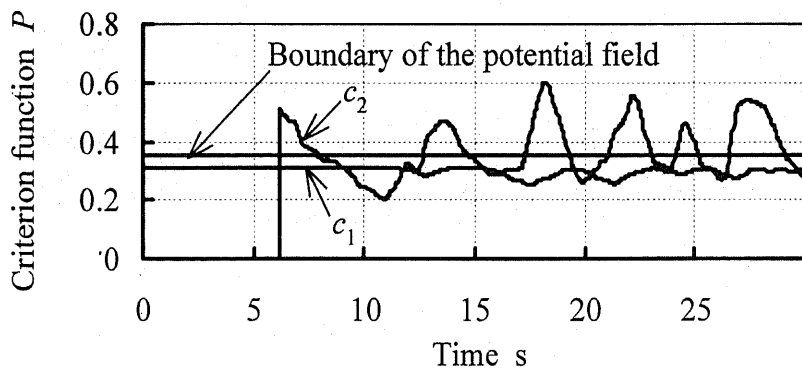
(a) Position and force in  $x$  direction



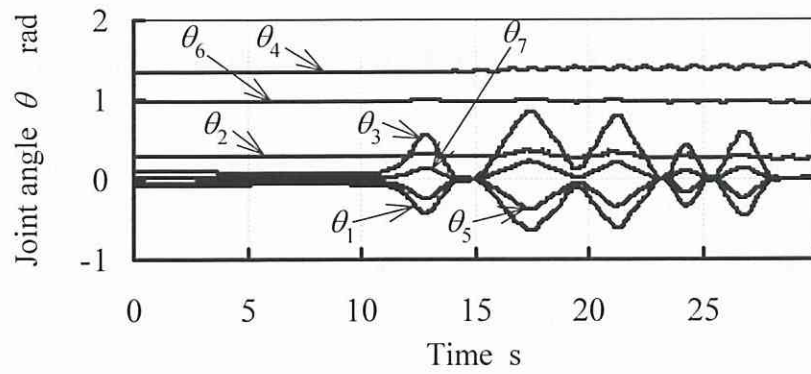
(b) Position and force in  $y$  direction



(c) Position and force in  $z$  direction

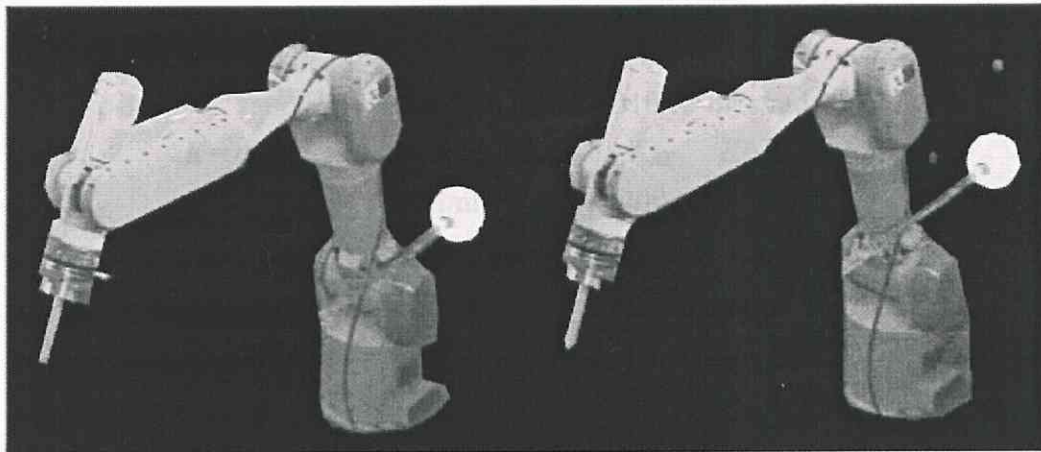


(d) Criterion function



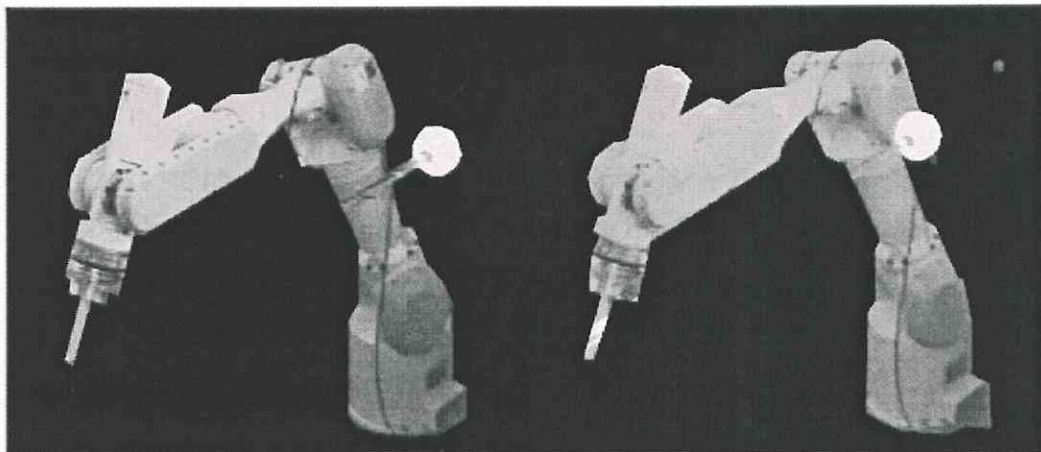
(e) Joint angle

Fig. 5·6 Results obtained in the 1st experiment



(a) At start position

(b) An obstacle is moving close



(c) Avoidance control starts

(d) During avoidance control

Fig. 5·7 Video images taken in the 1st experiment



### 5.5.2 実験 2(手先に半円軌道の倣い作業をさせながら衝突を回避する実験)

本実験では、衝突の回避を行いながらロボットが半径  $r_a$  の半円軌道の倣い作業を遂行できるかを確認する。各制御パラメータ、目標位置、目標力を表 5.3 に示す。冗長制御の効果を確認するため、まず、衝突物体の接近がない場合について倣い作業を行い、得られた  ${}^b z$  軸方向の結果を図 5.8(a)に示す。ついで、衝突対象の接近があり、かつ冗長制御を用いて倣い作業を行い、得られた  ${}^b z$  軸方向の結果を図 5.8(b)に、評価関数の計算結果を図 5.8(c)に、関節角度の追従結果を図 5.8(d)に示す。

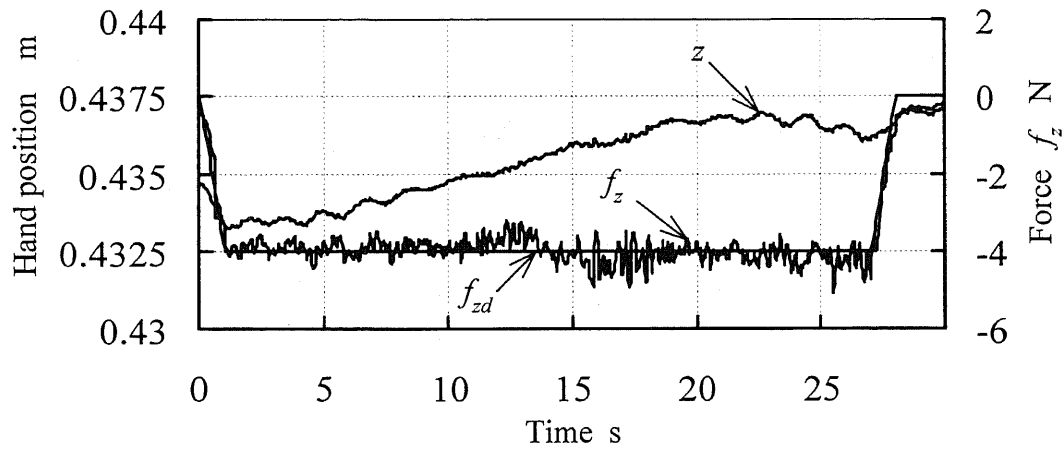
Table 5.3 Parameters used in the 2nd experiment

$M = \text{diag}[m_{h1} \quad m_{h1} \quad m_{h1} \quad m_{h2} \quad m_{h2} \quad m_{h2}]$ ,					
$m_{h1} = 1.2 \text{ kg}$ ,			$m_{h2} = 0.7 \text{ kg} \cdot \text{m}^2$		
$D = \text{diag}[d_{h1} \quad d_{h1} \quad d_{h1} \quad d_{h2} \quad d_{h2} \quad d_{h2}]$ ,					
$d_{h1} = 310 \text{ N} \cdot \text{s/m}$ ,			$d_{h2} = 110 \text{ N} \cdot \text{m} \cdot \text{s/rad}$		
$K = \text{diag}[K_{h1} \quad K_{h1} \quad 0 \quad K_{h2} \quad K_{h2} \quad K_{h2}]$ ,					
$K_{h1} = 670 \text{ N/m}$ ,			$K_{h2} = 400 \text{ N} \cdot \text{m} / \text{rad}$		
$M_r = \text{diag}[m_{j1} \quad m_{j1} \quad m_{j1} \quad m_{j2} \quad m_{j2} \quad m_{j2}]$ ,					
$m_{j1} = 0.3 \text{ kg}$ ,			$m_{j2} = 0.3 \text{ kg} \cdot \text{m}^2$		
$D_r = \text{diag}[d_{j1} \quad d_{j1} \quad d_{j1} \quad d_{j2} \quad d_{j2} \quad d_{j2}]$ ,					
$d_{j1} = 8 \text{ N} \cdot \text{s/m}$ ,			$d_{j2} = 8 \text{ N} \cdot \text{m} \cdot \text{s/rad}$		
$K_r = \text{diag}[0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$					
$f_d = [0 \quad 0 \quad 4.0 \text{ N} \quad 0 \quad 0 \quad 0]^T$					
Center of the circle : $p_{hd} = [0.70 \text{ m} \quad 0 \text{ m} \quad 0.435 \text{ m}]^T$					
Radius of the circle : $r_a = 0.15 \text{ m}$					

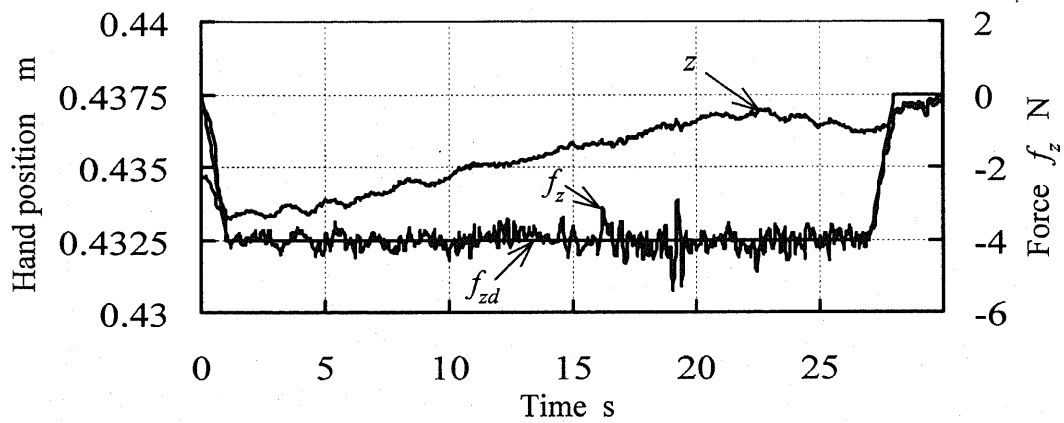
図 5.8(c)においては、式(5.19)の  $c_2$  が仮想ポテンシャル場の境界線より下回ることを視覚センサによって検出したとき、衝突物体の進入に合わせてロボットの各関節が素早く回避する方向に動いたことが図 5.8(d)から分かる。また、障害物が接近してくるときロボットが衝突を回避したことを図 5.9 に示すビデオ画面によって確かめた。

また、衝突を回避しながら倣い作業を行う実験で得られた図 5.8(b)の結果は、 ${}^b z$  軸方向の力制御では、図 5.8(a)の衝突物体がなかった実験結果とほぼ同程度の結果を得たため、本手法の有効性が確かめられた。

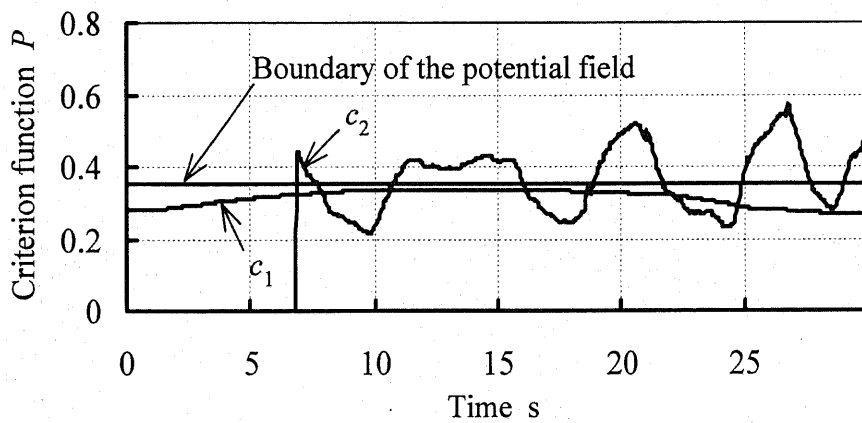
本実験に用いたロボットの冗長関節が 1 であるため、接近してくる衝突物体に対しての回避には限界があるが、冗長関節の数が多くなれば本手法によって完全回避の実現が可能であると考えられる。



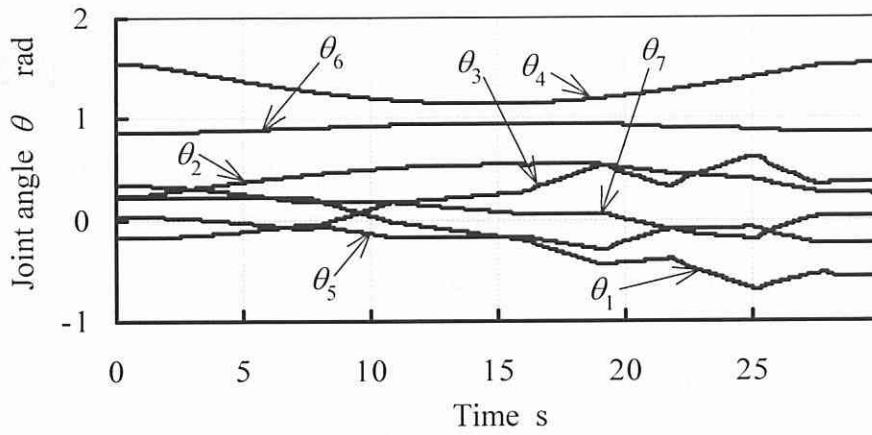
(a) Results of the experiment without any obstacle



(b) Results of the experiment using the proposed method for collision avoidance of a moving obstacle

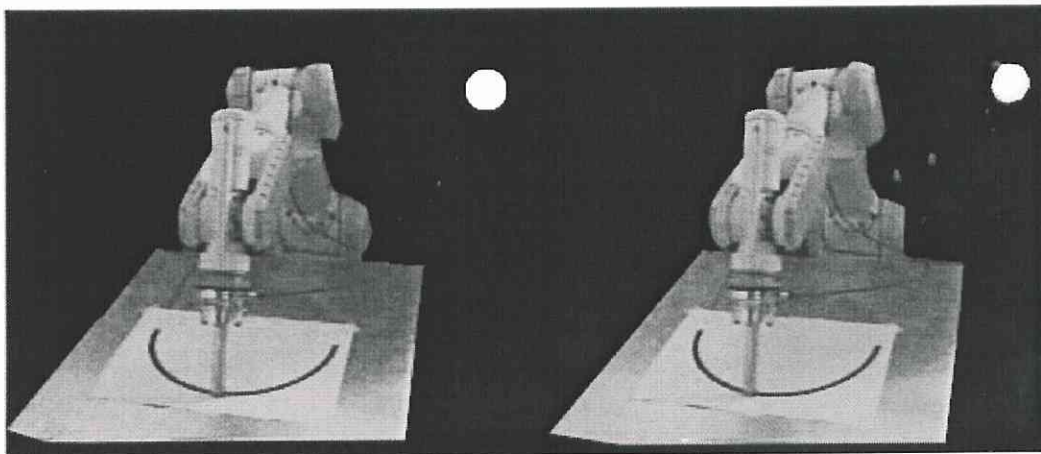


(c) Criterion function using the proposed method for avoidance of a moving obstacle



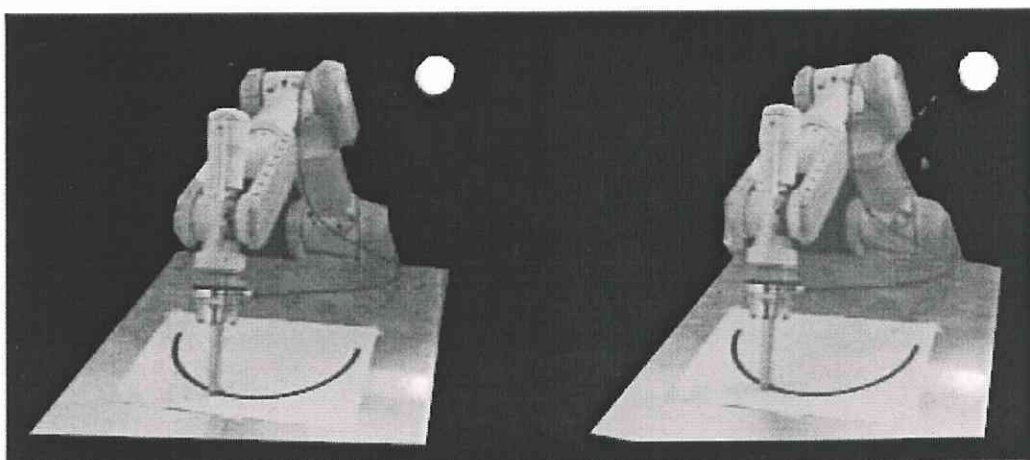
(d) Joint angles of the experiment using the proposed method for avoidance of a moving obstacle

Fig. 5·8 Results of the 2nd experiment



(a) Tracking a half circle

(b) An obstacle is moving close



(c) Avoidance control starts

(d) During avoidance control

Fig. 5·9 Video images taken in the 2nd experiment

## 5・6 要約

本研究で得られた主な結果は次のとおりである。

(1) 接近してくる衝突対象が仮想ポテンシャル場に入ったことを視覚情報によって検出したとき、仮想関節トルクを生成させるという手法を提案し、実験によりその効果を実証した。

(2) 冗長ロボットの可操作性を考慮しながら手先の作業と衝突の回避作業を同時に行うという制御手法を提案し、実験によりその効果を実証した。

(3) 視覚情報の実時間処理を用いた冗長ロボットの制御系を構成し、実験によりその有効性を実証した。

---

## 第6章

# 視覚情報を用いたシーリングロボットの目標軌道の実時間生成法と制御

---

### 6.1 概説

本章では、シーリングロボットの追従作業を取り上げ、環境の情報が不足する場合、ロボットの未来の目標軌道をどのように生成するかを検討する。

シーリング作業とは、自動車体のスポット溶接と上塗り・下塗りなどの作業を行ったのち、水漏れやさびを防ぐため、車体鋼板の継目部にシール剤を塗布する作業である。シーリング作業は汚れと疲れの多い作業であるため、ロボットによるこの作業の自動化に関する研究が行われてきた<sup>(10)~(12),(47)~(50)</sup>。

これまでの研究では、まず、張・藤堂らはニューラルネットワークによる継目の位置を認識する方法を開発した<sup>(50)</sup>。ついで、視覚センサから得られた濃淡画像と距離情報を用いて、オフラインであらかじめ教示を行い、再生中に視覚センサを用いて継目位置を検出し、前もって教示されたシール軌道を再生中に修正しながらシーリングロボットが作業を行わせるという手法を開発した<sup>(10)</sup>。また、張・藤堂らは、継目ラインの形状が複雑な場合に対してロボットの作業軌道を生成し、教示を自動的に行うシステムの開発を試みた<sup>(11)</sup>。さらに、黄・藤堂らは、シーリング対象が次々に流れてくる実際のシーリング作業現場を想定し、鋼板の汚れや熱変形などワーク自身に生じる予測できない軌道の変化に対して、視覚情報を用いてシーリング目標軌道をオンラインで生成する方法を提案した<sup>(12)</sup>。

既報<sup>(10)</sup>においては、教示を前もって必要とする方法を用いたため、視覚センサの位置を制御せずに、ノズルの位置をシーリング軌道に追従させる制御を中心に行ってきた。しかしながら、自動教示法<sup>(11)</sup>、ならびに教示不要の制御手法<sup>(12)</sup>では、ノズルの位置を制御しているが、ロボットの手先に取り付けられた視覚センサの位置を制御していない。このため、視覚センサはシーリング軌道の曲がる部分において継目を見失ってロボットの手先が軌道から外れやすいと考えられる。本章では、視覚センサがシー

リング軌道を見失うことを防ぐため、ノズルとともに視覚センサをも実時間にシーリング軌道に追従させることを目的とする。

一般に、画像処理の最中でもロボットの手先が動いているため、視覚センサの移動先の目標位置を生成するのに必要な環境の情報が不足している。一方、作業対象が次々に流れてくる生産現場においては、車体鋼板の形状の多くは滑らかな連続曲線であることが事前に分かっている。それらの形状特徴を前もって学習させるため、作業中にあらゆる形状のシーリング軌道ごとの教示は不要となり、視覚センサの未来位置を予測できるという点に本手法は基づいている。

本章では、作業軌道の形状が連続曲線の場合、ニューラルネットワークとファジィ推論を用いて視覚センサの目標軌道を実時間で生成する手法を提案する。ついで、手先の走行方向に対してノズルが視覚センサの後方に装着されているため、視覚センサで既に検出された継目位置を用いることによってノズルの目標軌道を生成する。また、ノズルの目標軌道を視覚センサ座標系から見た回転量（すなわち、視覚センサの姿勢情報）に変換する。このように視覚センサの位置・姿勢を制御することによって、視覚センサとノズルを実時間でシーリング軌道に追従させる。本手法では、教示を必要としないため、シーリングの作業開始点と終了点の位置のみロボットに与えれば、自動的に作業軌道に追従することができる。

## 6.2 三次元視覚センサによるシーム位置の検出

### 6.2.1 三次元視覚センサの構成とシーム位置・姿勢の検出

既報<sup>(10)~(12)</sup>と同一の視覚センサを用いシームの6自由度空間情報（位置・姿勢）を検出する。図6.1に示すよう三次元視覚センサはCCDカメラ1台と2本のスリットレーザ光源から構成され、マニピュレータの手先に固定されている。

視覚センサ座標系からシームの空間位置を算出する原理を図6.2に示す。既報<sup>(12)</sup>により、CCDカメラから見たP平面に投影したスリット光1と2の不連続点の視覚座標系における空間位置 ${}^s s_{p1} = [{}^s S_{x1}, {}^s S_{y1}, {}^s S_{z1}]^T$ と ${}^s s_{p2} = [{}^s S_{x2}, {}^s S_{y2}, {}^s S_{z2}]^T$ は、

$${}^s S_{xi} = \frac{f - {}^s S_{zi}}{F} X_i \quad (i=1,2) \dots\dots\dots (6.1)$$

$${}^s S_{yi} = \frac{f - {}^s S_{zi}}{F} Y_i \quad (i=1,2) \dots\dots\dots (6.2)$$

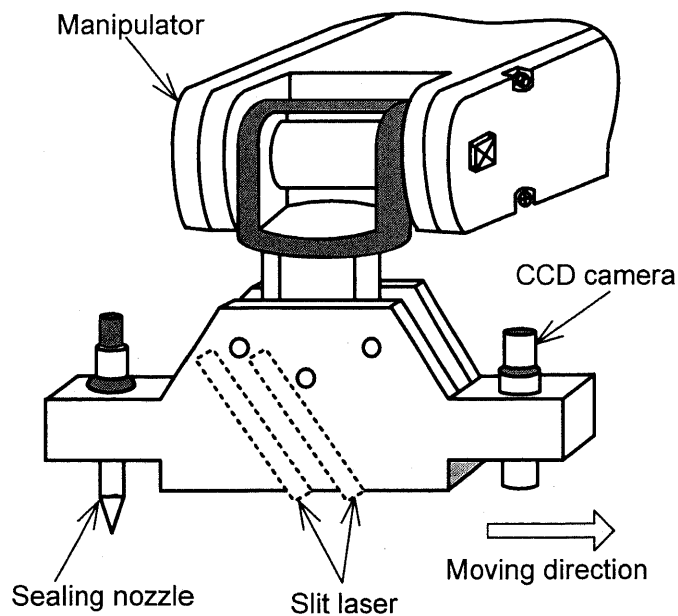


Fig. 6·1 A vision sensor and a sealing nozzle mounted on the end effector of a manipulator

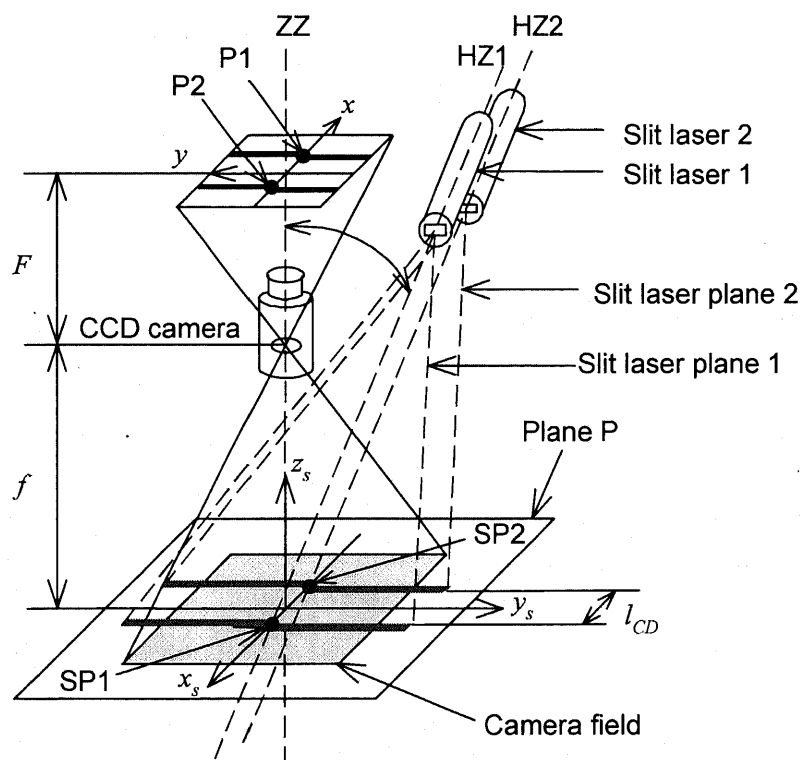


Fig.6·2 Optical principle of crank detection by a vision sensor

$${}^sS_{zi} = \frac{f \cdot X_i + 0.5 \cdot F \cdot l_{CD}}{X_i - F \cdot \tan \phi} \quad (i=1,2) \quad \dots\dots\dots (6.3)$$

で表される。ただし、 $(X_1, Y_1)$ はその点の像面での点  $P_1$  の座標、 $(X_2, Y_2)$ は点  $P_2$  の座標である。 $f$ と  $F$ は視覚センサ固有のパラメータ値である。また、既報<sup>(12)</sup>により、光切断イメージの上に現われるレーザスリットの投影の傾斜角と形状から、 $x_s$  軸、 $y_s$  軸、 $z_s$  軸回りの回転角  ${}^s\theta_x, {}^s\theta_y, {}^s\theta_z$  が求められる。

### 6.2.2 画像処理のアルゴリズム

図 6.3 に示すように、CCDカメラで得られた光切断イメージ上の継目位置に段差が現われる。この段差を検出するため、既報<sup>(10)~(12),(50)</sup>で提案された探索窓処理法を用いる。探索窓処理法は光切断イメージ上に投影されたスリットレーザの近傍に探索窓系列を生成し、探索窓系列の中心位置の二次微分の最大値を算出することによって継目位置を検出する方法である。この方法では、CCDカメラから得た膨大な画像データの全体を処理する必要がないため、画像の高速処理を図ることができる。

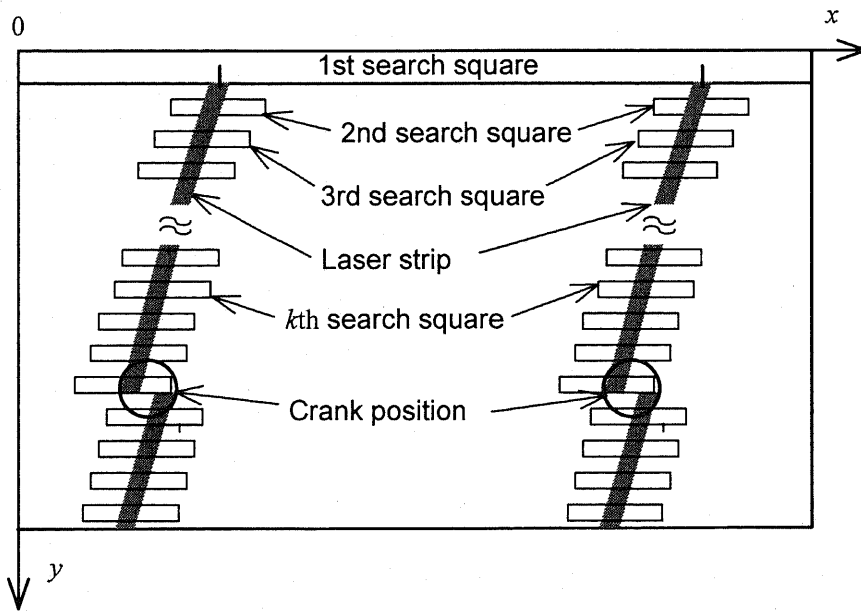


Fig.6.3 Detection of crank positions by applying the derivative computation to the centers of two searching block series



### 6.3 シーリングロボットの目標軌道の生成法

ロボット制御系のサンプリング周期を  $T$  とし、継目位置を検出するための画像処理のサンプリング周期を  $mT$  とする。また、図 6.1 に示すように視覚センサがノズルに先行して取り付けられているため、視覚センサの位置にノズルが来るまでに要する時間を  $nT$  とする。ただし、視覚処理とノズルの目標軌道を生成するための演算遅れがあるため、 $m$  と予見サンプリング数  $n$  の間には、 $n \geq 2m$  の条件が必要となる。本章では、 $n=2m$  とする。

#### 6.3.1 ノズルの目標軌道の生成

図 6.4 に示すように、時刻  $t=imT$  においてノズルの位置は  ${}^B p_N(im)$  であり、視覚センサの位置は  ${}^B p_S(im)$  である。また、ノズルの位置に先行して CCD カメラが取り付けられているため、この時点で継目の空間位置  ${}^B p(im-m)$  が画像処理により得られる。この情報は予見的な情報として利用し、ノズルの移動先の目標位置  ${}^B p_N(im+m)$  と姿勢  ${}^B R_N(im+m)$  を算出する。

ロボットの制御サンプリング周期ごとの目標軌道  $\theta_{Nd}(k)$  を生成するため、まずロボットの逆運動学によりノズルの位置  ${}^B p_N(im)$ ,  ${}^B p_N(im+m)$  と姿勢  ${}^B R_N(im)$ ,  ${}^B R_N(im+m)$  からロボットの関節角度  $\theta_N(im)$ ,  $\theta_N(im+m)$  を

$$\theta_N(im) = A_N^{-1}({}^B p_N(im), {}^B R_N(im)) \dots \dots \dots (6.4)$$

$$\theta_N(im+m) = A_N^{-1}({}^B p_N(im+m), {}^B R_N(im+m)) \dots \dots \dots (6.5)$$

より求める。ついで、関節角速度の急変が生じないようにするため、既報<sup>(11)</sup>で提案した二次ホルダを用いて、ノズルの目標軌道  $\theta_{Nd}(k)$  を

$$\theta_{Nd}(k) = \theta_N(im) + \frac{2(k-im)^2}{m^2} (\theta_N(im+m) - \theta_N(im))$$

$$(im \leq k < im+m/2) \dots \dots \dots (6.6)$$

$$\theta_{Nd}(k) = \theta_N(im) + (1 - \frac{2(k-im)^2}{m^2}) (\theta_N(im+m) - \theta_N(im))$$

$$(im+m/2 \leq k < (i+1)m) \dots \dots \dots (6.7)$$

より算出する。

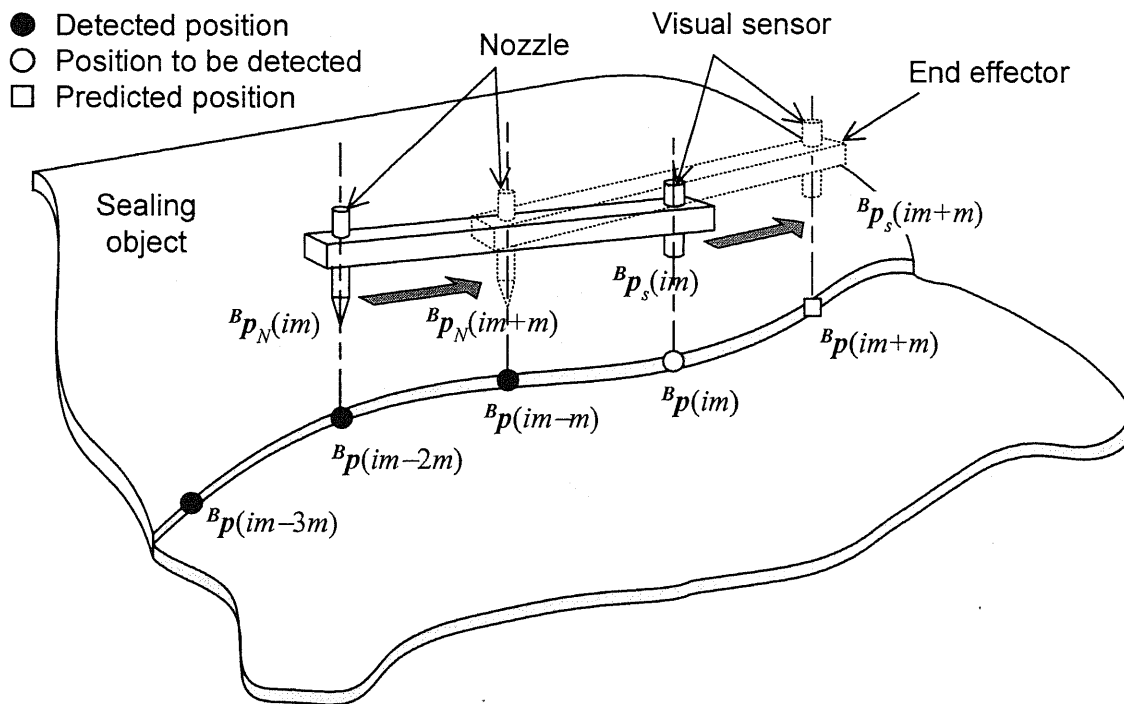


Fig.6.4 Moving the end-effector to desired position in tracking

### 6.3.2 視覚センサの目標軌道の生成

ロボットの先端に取り付けられた視覚センサの視野が狭いため、シール軌道の曲がる部分で視覚センサが軌道から外れやすいと考えられる。したがって、ノズルと同様に視覚センサの目標軌道を何らかの方法で生成し、その位置を制御する必要がある。6.3.1 節で述べたように画像処理によって検出した継目位置を予見情報としてノズルの目標軌道を生成することはできるが、画像処理の最中에서도手先が動いているため、視覚センサの目標位置を予測するのは困難である。

一般に、ファジィ推論は大局の判断が得意であり、未知の外部環境の変化に対して柔軟に対応できるため、知的制御によく用いられてきた<sup>(51)~(53)</sup>。一方、シーリングの生産現場で取り扱われる作業対象の形状の多くは連続曲線であるため、それらの形状の特徴を学習させたのち、あらゆる形状のシール軌道に対して何らかの方法によって視覚センサの移動先の位置を推定することは可能と考えられる。したがって、本章では、ファジィ推論によるCCDカメラの目標軌道の予測法を提案する。本手法では、作業環境とロボットの特性とを含む情報を学習したニューラルネットワークを用いてファジィ推論のルールを獲得した、このため、加工や熱変形など予想しにくい原因に対してより有効な対応を本手法に期待できる。

### 6.3.2.1 ファジィ推論

ここで、ファジィ推論の第  $k$  番目のルール  $R^k$  に対して前件部入力  $x_i$  の適応度  $X_{ik}$  が and で結合されたファジィ命題からなり、後件部は線形式である。

ルール  $R^k$  :

$$\left. \begin{aligned}
 &\text{If } x_1 \text{ is } X_{1k} \text{ and } x_2 \text{ is } X_{2k} \cdots \\
 &\quad \text{and } x_i \text{ is } X_{ik} \cdots \text{ and } x_n \text{ is } X_{nk} \text{ then} \\
 &y_{1k} = a_{11k}x_1 + a_{12k}x_2 + \cdots + a_{1nk}x_n + b_{1k} \\
 &y_{2k} = a_{21k}x_1 + a_{22k}x_2 + \cdots + a_{2nk}x_n + b_{2k} \\
 &\quad \cdots \cdots \cdots \\
 &y_{ik} = a_{i1k}x_1 + a_{i2k}x_2 + \cdots + a_{ink}x_n + b_{ik} \\
 &\quad \cdots \cdots \cdots \\
 &y_{mk} = a_{m1k}x_1 + a_{m2k}x_2 + \cdots + a_{mnk}x_n + b_{mk} \\
 &\quad k=1, 2, \cdots, r
 \end{aligned} \right\} \cdots \cdots \cdots (6.8)$$

ただし、入力  $\mathbf{x}=[x_1, x_2, \cdots, x_n]^T$  は  $n$  次元のベクトルであり、 $k$  番目のルールでの出力  $\mathbf{y}_k=[y_{1k}, y_{2k}, \cdots, y_{mk}]^T$  は  $m$  次元のベクトルである。ルール数は  $r$  個であり、添字  $i$  は第  $i$  番目の入力を示す。また、前件部で使われる入力の適合度は、図 6.5 に示すような釣鐘型のメンバーシップ関数である。その適合度  $X_{ik}(x_i)$  を

$$X_{ik}(x_i) = \exp\left(-\frac{(x_i - p_{ik})^2}{q_{ik}^2}\right) \quad (i=1, 2, \cdots, n) \cdots \cdots \cdots (6.9)$$

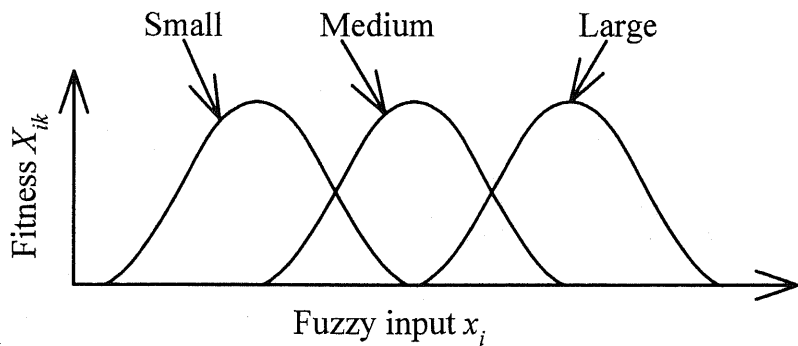


Fig.6.5 Proposed fuzzy membership function

で表す。ただし、 $p_{ik}$ ,  $q_{ik}$  はメンバーシップ関数の形状を表すパラメータである。また、適合度  $X_{ik}(x_i)$  は 0 から 1 までの実数を取り、最も適しているときに 1 となる。

$r$  個のルールを用いたときの出力  $Y$  を求めるための推論法は以下のようなになる。入力  $x=[x_1, x_2, \dots, x_n]^T$  が与えられたとき、ルール  $k$  の適合度  $\mu_k$  は

$$\mu_k = \prod_{i=1}^n X_{ik} = \exp\left(-\sum_{i=1}^n \frac{(x_i - p_{ik})^2}{q_{ik}^2}\right) \dots\dots\dots (6\cdot10)$$

で表される。個々のルールによる推論結果  $y_{ik}$  が式(6・8)で求められており、全体の推論結果  $Y=[Y_1, Y_2, \dots, Y_m]^T$  はこれらの重み付き平均値から求める。すなわち、第  $j$  番目の推論結果  $Y_j$  は

$$Y_j = \frac{\sum_{k=1}^r \mu_k y_{jk}}{\sum_{k=1}^r \mu_k} \quad (j=1, 2, \dots, m) \dots\dots\dots (6\cdot11)$$

となる。

### 6・3・2・2 ファジィルールの学習則

ファジィ推論によって入出力関係を導き出すとき、最も労力を要するのはファジィルールにおける諸パラメータの設定である。6・3・2・1 節で定義したファジィ推論の諸パラメータを導くには、誤差逆伝ば法に準じた方法で学習を行うことによってファジィ推論に用いる諸パラメータを同定するという手法を用いる。

ここで、ファジィ推論の入力を  $x$ 、第  $k$  番のルールの出力を  $y_k$  とし、式(6・9)で定義したメンバーシップ関数の形状を表すパラメータを  $p_{ik}, q_{ik}$ 、推論後件部のパラメータを  $a_{ijk}, b_{jk}$ 、ルールごとの適合値を  $\mu_i$  とする。また、出力  $Y_j$  に対する教師信号を  $Y_j^*$  とし評価関数  $E$  を次式で表す。

$$E = \frac{1}{2} \sum_{j=1}^m (Y_j - Y_j^*)^2 \dots\dots\dots (6\cdot12)$$

また、推論の誤差  $\delta_j$  は

$$\delta_j = \frac{\partial E}{\partial Y_j} = Y_j - Y_j^* \dots\dots\dots (6\cdot13)$$

で定義される。推論に用いる各パラメータの学習は下記のように行う。ただし、学習率を  $\varepsilon$  で表す。

#### (1) $a_{ijk}$ についての学習

$\Delta a_{ijk}$  は

$$\Delta a_{ijk} = -\varepsilon \frac{\partial E}{\partial a_{ijk}} \dots\dots\dots (6\cdot14)$$

で定義される。ここで、

$$\frac{\partial E}{\partial a_{ijk}} = \frac{\partial E}{\partial Y_j} \frac{\partial Y_j}{\partial y_{jk}} \frac{\partial y_{jk}}{\partial a_{ijk}} = \delta_j \frac{\partial Y_j}{\partial y_{jk}} \frac{\partial y_{jk}}{\partial a_{ijk}} \dots \dots \dots (6 \cdot 15)$$

$$\frac{\partial y_{jk}}{\partial a_{ijk}} = x_i \dots \dots \dots (6 \cdot 16)$$

の関係がある。式(6・11), (6・15), (6・16)を用いると式(6・14)から

$$\Delta a_{ijk} = -\varepsilon \frac{\mu_k}{\sum_{k=1}^r u_k} x_i \delta_i \dots \dots \dots (6 \cdot 17)$$

が得られる。  $a_{ijk}$  は

$${}^{new} a_{ijk} = {}^{old} a_{ijk} + \Delta a_{ijk} \dots \dots \dots (6 \cdot 18)$$

により古い値  ${}^{old} a_{ijk}$  から新しい値  ${}^{new} a_{ijk}$  へ更新させる。

### (2) $b_{jk}$ についての学習

$\Delta b_{jk}$  は

$$\Delta b_{jk} = -\varepsilon \frac{\partial E}{\partial b_{jk}} \dots \dots \dots (6 \cdot 19)$$

で定義される。ここで、

$$\frac{\partial E}{\partial b_{jk}} = \frac{\partial E}{\partial Y_j} \frac{\partial Y_j}{\partial y_{jk}} \frac{\partial y_{jk}}{\partial b_{jk}} = \frac{\mu_k}{\sum_{k=1}^r \mu_k} \delta_j \dots \dots \dots (6 \cdot 20)$$

の関係があるため、

$$\Delta b_{jk} = -\varepsilon \frac{\mu_k}{\sum_{k=1}^r u_k} \delta_i \dots \dots \dots (6 \cdot 21)$$

が得られ、  $b_{jk}$  は

$${}^{new} b_{jk} = {}^{old} b_{jk} + \Delta b_{jk} \dots \dots \dots (6 \cdot 22)$$

により古い値  ${}^{old} b_{jk}$  から新しい値  ${}^{new} b_{jk}$  へ更新させる。

### (3) $p_{i,k}$ についての学習

$\Delta p_{ik}$  は

$$\Delta p_{jk} = -\varepsilon \frac{\partial E}{\partial p_{jk}} \dots \dots \dots (6 \cdot 23)$$

で定義される。また、

$$\frac{\partial E}{\partial p_{jk}} = \frac{\partial E}{\partial Y_j} \frac{\partial Y_j}{\partial \mu_k} \frac{\partial \mu_k}{\partial p_{jk}} = \delta_j \frac{\partial Y_j}{\partial \mu_k} \frac{\partial \mu_k}{\partial p_{jk}} \dots \dots \dots (6 \cdot 24)$$

$$\frac{\partial Y_j}{\partial \mu_k} = -\frac{y_j \sum_{k=1}^r \mu_k - y_{jk} \sum_{k=1}^r \mu_k}{\left(\sum_{k=1}^r \mu_k\right)^2} = \frac{y_{jk} - y_j}{\sum_{k=1}^r \mu_k} \dots \dots \dots (6 \cdot 25)$$

$$\frac{\partial \mu_k}{\partial p_{ik}} = \frac{\partial}{\partial p_{ik}} \left[ \exp \left\{ -\sum_{i=1}^n \frac{(x_i - p_{ik})^2}{q_{ik}^2} \right\} \right] = \mu_k \left\{ \frac{2(x_i - p_{ik})}{q_{ik}^2} \right\} \dots \dots \dots (6 \cdot 26)$$

の関係があるため、

$$\Delta p_{ik} = -\varepsilon \frac{y_{jk} - y_j}{\sum_{k=1}^r \mu_k} \mu_k \left\{ \frac{2(x_i - p_{ik})}{q_{ik}^2} \right\} \delta_j \dots \dots \dots (6 \cdot 27)$$

が得られる。  $p_{ik}$  は

$${}^{new} p_{ik} = {}^{old} p_{ik} + \Delta p_{ik} \dots \dots \dots (6 \cdot 28)$$

により古い値  ${}^{old} p_{ik}$  から新しい値  ${}^{new} p_{ik}$  へ更新させる。

#### (4) $q_{ik}$ についての学習

$\Delta q_{ik}$  は

$$\Delta q_{jk} = -\varepsilon \frac{\partial E}{\partial q_{jk}} \dots \dots \dots (6 \cdot 29)$$

で定義される。ここで、

$$\frac{\partial E}{\partial q_{jk}} = \frac{\partial E}{\partial Y_j} \frac{\partial Y_j}{\partial \mu_k} \frac{\partial \mu_k}{\partial q_{jk}} = \delta_j \frac{\partial Y_j}{\partial \mu_k} \frac{\partial \mu_k}{\partial q_{jk}} \dots \dots \dots (6 \cdot 30)$$

$$\frac{\partial Y_j}{\partial \mu_k} = -\frac{y_j \sum_{k=1}^r \mu_k - y_{jk} \sum_{k=1}^r \mu_k}{\left(\sum_{k=1}^r \mu_k\right)^2} = \frac{y_{j,k} - y_j}{\sum_{k=1}^r \mu_k} \dots \dots \dots (6 \cdot 31)$$

$$\frac{\partial \mu_k}{\partial q_{ik}} = \frac{\partial}{\partial q_{i,k}} \left[ \exp \left\{ -\sum_{i=1}^n \frac{(x_i - p_{ik})^2}{q_{ik}^2} \right\} \right] = \mu_k \left\{ \frac{(x_i - p_{ik})^2}{q_{ik}^3} \right\} \dots \dots \dots (6 \cdot 32)$$

の関係があるため、

$$\Delta q_{ik} = -\varepsilon \frac{y_{jk} - y_j}{\sum_{k=1}^r \mu_k} \mu_k \left\{ \frac{(x_i - p_{ik})^2}{q_{ik}^3} \right\} \delta_j \dots \dots \dots (6 \cdot 33)$$

が得られる。  $q_{i,k}$  は

$${}^{new} q_{ik} = {}^{old} q_{ik} + \Delta q_{ik} \dots \dots \dots (6 \cdot 34)$$

により古い値  ${}^{old} p_{ik}$  から新しい値  ${}^{new} p_{ik}$  へ更新させる。

### 6.3.2.3 ファジィルールの学習と視覚センサの目標軌道の生成

時刻  $t=imT$  において、既に検出された継目の位置と速度情報を用いてファジィ推論で視覚センサの目標位置  ${}^B p_s(im+m)$  を推定し、目標軌道  $\theta_s(k)$  を生成する。6.3.2.2 節で述べたファジィルールの学習に用いたブロック線図を図 6.6 に示す。時刻  $t=imT$  において、ファジィ入力  $\mathbf{x}=[x_1, x_2, \dots, x_n]^T$  ( $n=12$ ) は、時刻  $t=(i-1)mT$  および  $t=(i-2)mT$  で検出された継目の位置情報  ${}^B p(im-m)$ ,  ${}^B p(im-2m)$ , 手先の移動速度  ${}^B \dot{p}(im-m)$ ,  ${}^B \dot{p}(im-2m)$  である。ファジィ出力  $\mathbf{Y}=[Y_1, Y_2, \dots, Y_m]^T$  ( $m=3$ ) は時刻  $(i+1)mT$  での視覚センサの位置  ${}^B \hat{p}_s(im+m)$  である。検証実験の結果に基づいてファジィ推論のルール数  $r$  を 7 とし、教示データは前もって用意した直線、折線、円弧曲線をロボットが追従した結果を学習に用いる。各教示データに対して誤差の二乗和が収束するまで約 4,000 回の学習を行った。それによりファジィ推論ルールのパラメータ  $[a_{ij,k}]_{12 \times 3 \times 7}$ ,  $[p_{i,k}]_{12 \times 7}$ ,  $[q_{i,k}]_{12 \times 7}$  が得られた。

学習済みのファジィルールを用いて直線、曲線など連続曲線形状をもつシール軌道に対して予測を行った。予測誤差は 1mm 以下であるため、学習で得られたパラメータ行列を用いて良好な推論結果を得た。予測結果の一例を図 6.7 に示す。また、図 6.6 に示すようなファジィ推論の構成では、継目の位置情報のみではなくロボットの手先速度も考慮しているため、速度の急激な変化がないという利点があると考えられる。

得られた時刻  $t=(i+1)mT$  の視覚センサの目標位置  ${}^B \hat{p}_s(im+m)$  から逆運動学により  $\theta_s(im+m)$  を算出し、視覚センサの目標軌道  $\theta_{sd}(k)$  を

$$\theta_{sd}(k) = \theta_s(im) + \frac{2(k-im)^2}{m^2} (\theta_s(im+m) - \theta_s(im)) \quad (im < k < im+m/2) \dots \dots \dots (6.35)$$

$$\theta_{sd}(k) = \theta_s(im) + (1 - \frac{2(k-im)^2}{m^2}) (\theta_s(im+m) - \theta_s(im)) \quad (im+m/2 < k < (i+1)m) \dots \dots \dots (6.36)$$

により生成する。

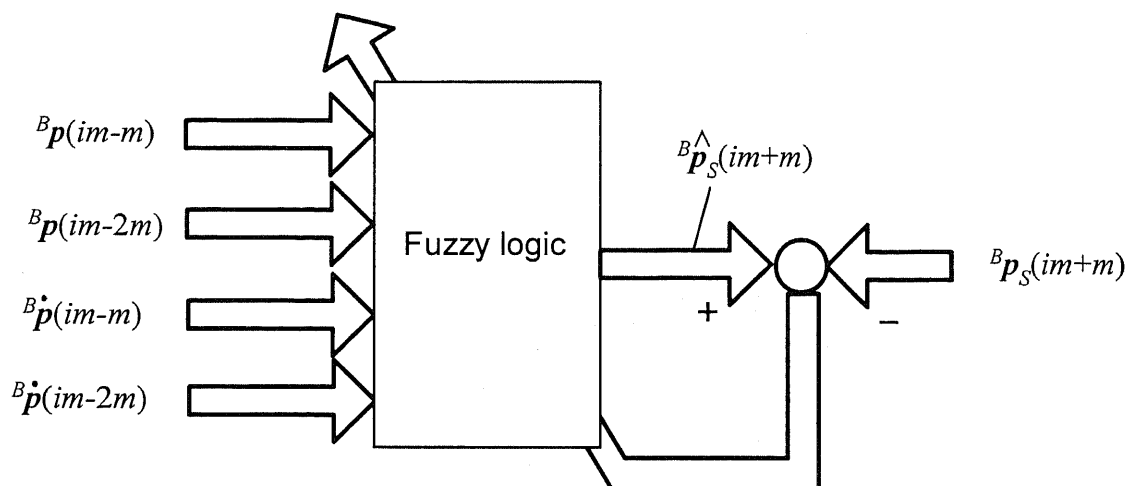


Fig.6.6 NN Learning for the proposed fuzzy rule

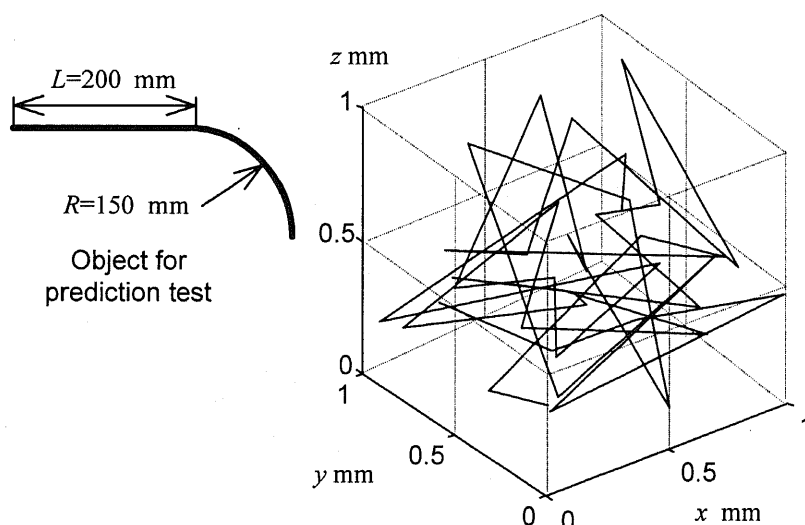


Fig.6.7 Prediction error using the obtained fuzzy model

### 6.3.3 ロボットの目標軌道の生成

実験に用いた6自由度のロボットによって手先に取り付けられたノズルと視覚センサのそれぞれの位置と姿勢の12自由度を同時に制御することはできないが、ノズルと視覚センサが作業中に果たす機能からノズルと視覚センサの位置制御のそれぞれに各3自由度ずつ振り分けることができる。したがって、本章では、ノズルの目標位置を視覚センサの目標姿勢に変換する方法を提案し、視覚センサの位置と姿勢を制御することによって、ノズルと視覚センサの位置制御を実時間で行う。

具体的には、時刻  $t=kT$  において、図 6.8 に示すようノズルの目標位置を  ${}^B p_{Nd}(k)=$



$[x_{Nd}(k), y_{Nd}(k), z_{Nd}(k)]^T$ , 視覚センサの位置を  ${}^B p_{Sd}(k)=[x_{Sd}(k), y_{Sd}(k), z_{Sd}(k)]^T$  とするとき, ノズルの位置を視覚センサから見た  $Y_S, Z_S$  軸回りの回転量  $\theta_{Sy}, \theta_{Sz}$  は, 変換式

$$\theta_{Sy}(k) = \tan^{-1}\left(-\frac{z_{Sd}(k) - z_{Nd}(k)}{l_s}\right) \dots\dots\dots (6\cdot37)$$

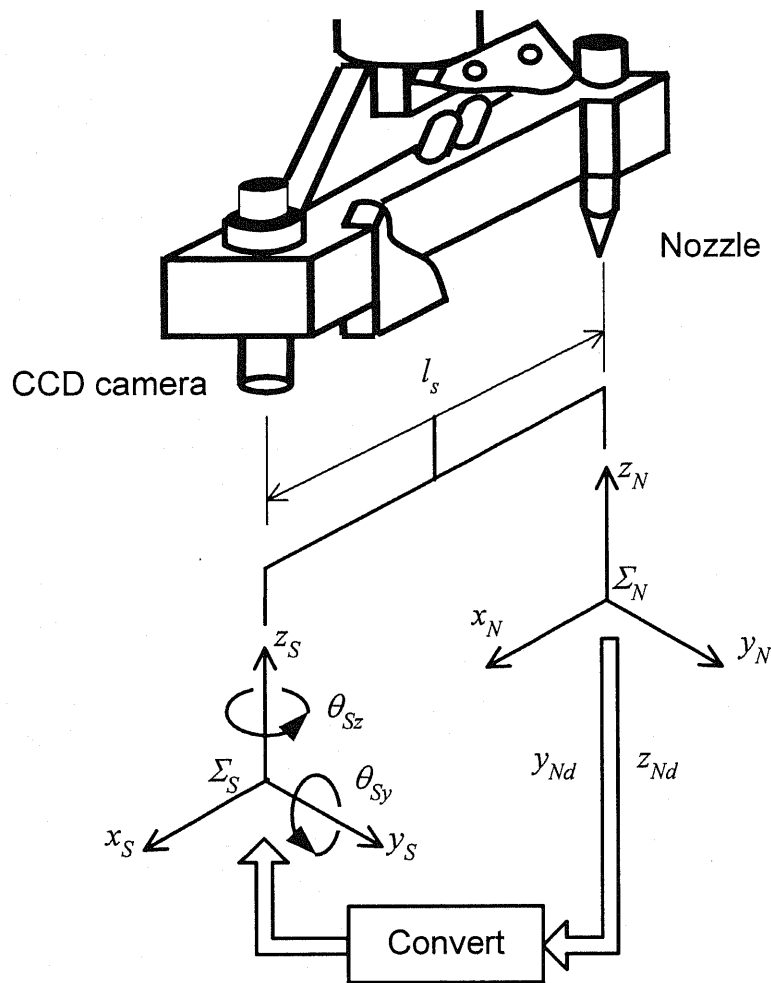


Fig.6・8 Method of converting a nozzle position to pose of a visual sensor

$$\theta_{Sz}(k) = \tan^{-1}\left(-\frac{y_{Sd}(k) - y_{Nd}(k)}{l_s}\right) \dots\dots\dots (6\cdot38)$$

から求められる。また, 視覚センサの姿勢の補正回転行列  $\Delta R_{Sd}(k)$  は

$$\Delta R_{Sd}(k) = \text{Rot}(\theta_{Sz}(k)) \times \text{Rot}(\theta_{Sy}(k)) \times \text{Rot}(\theta_{Sx}(k)) \dots\dots\dots (6\cdot39)$$

で表される。これにより視覚センサの姿勢  $R_{Sd}(k)$  から

$$R_{Sd}(k) = R_S(k) \Delta R_{Sd}(k) \dots\dots\dots (6\cdot40)$$

で目標姿勢  $R_{Sd}(k)$  を算出する。

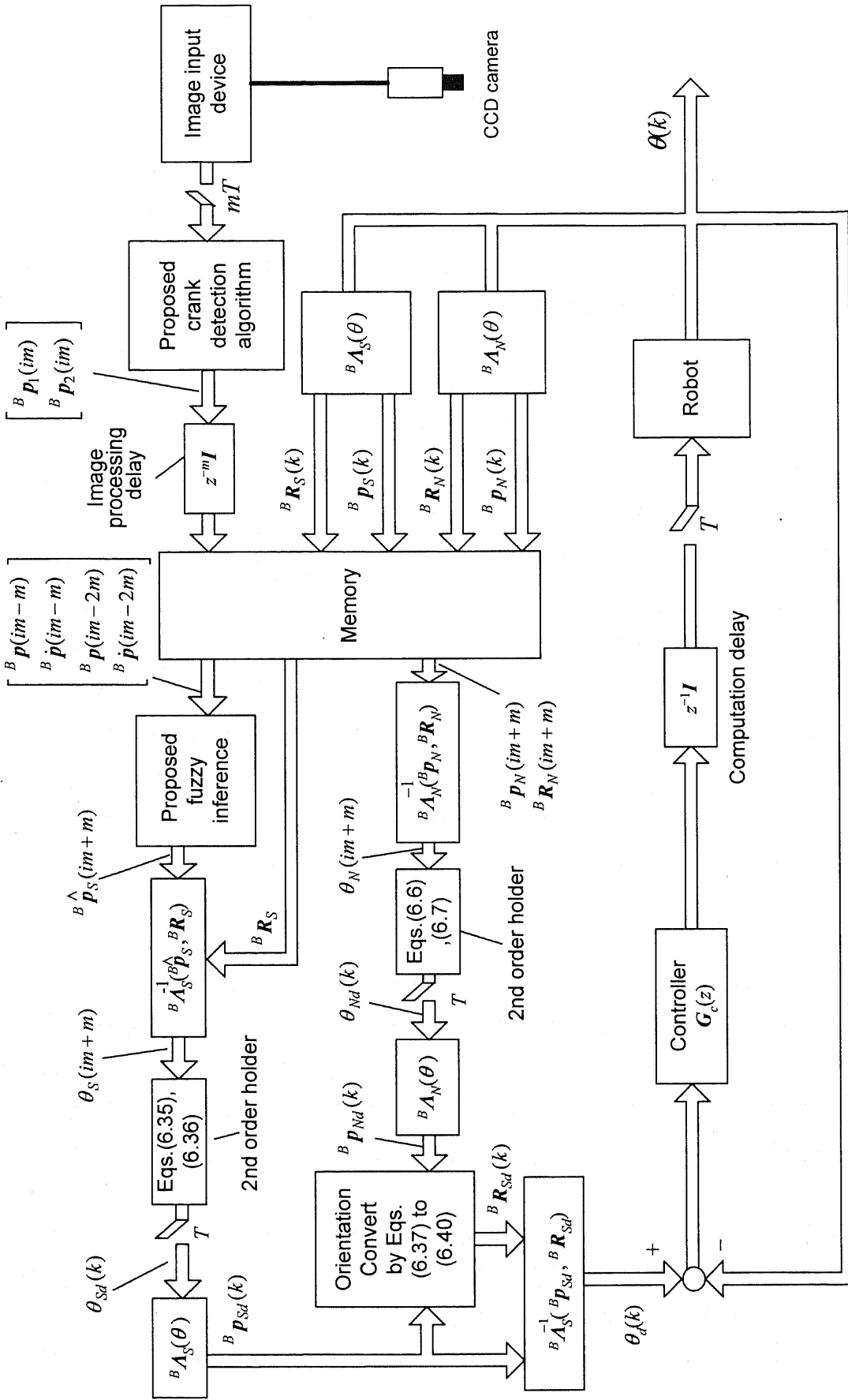


Fig. 6·9 Block diagram of the proposed visual feedback control of a sealing robot

## 6.4 シーリングロボットの追従実験

### 6.4.1 シーリングロボットの視覚フィードバック制御

追従実験に用いたシーリングロボットの制御ブロック線図を図 6.9 に示す. 図 6.9 において, シール軌道の画像時刻  $t=imT$  ( $i=0,1,2,\dots$ ) のとき, シール軌道の画像を取り込み, 6.2.2 節で述べた探索窓法によってスリット光の段差の位置を検出し, 継目の空間位置  ${}^B p(im)$  を算出する. また, 制御サンプリング周期ごとにロボットの関節角度  $\theta(k)$  から運動学によりノズルの位置  ${}^B p_N(k)$ , 姿勢  ${}^B R_N(k)$  と視覚センサの位置  ${}^B p_S(k)$ , 姿勢  ${}^B R_S(k)$  を計算し, メモリに保存する.

視覚センサの目標軌道の生成においては, 6.3.2 節で提案したファジィ推論法を用いて既に得られたシール軌道の位置と速度の情報から視覚センサの目標位置  ${}^B \hat{p}_S(im+m)$  を推定し, 式(6.35)~(6.36)で視覚センサの目標軌道  $\theta_{sd}(k)$  を生成する. ノズルの目標軌道の生成には, 検出されたシール軌道の位置  ${}^B p(im-m)$  から式(6.6), (6.7)によってノズルの目標軌道  $\theta_{Nd}(k)$  を生成する. また, ノズルの目標位置を視覚センサの回転量, すなわち式(6.37)~(6.40)で視覚センサの姿勢  ${}^B R_{sd}(k)$  に変換し, 逆運動学の変換行列  $A_S^{-1}({}^B p_{sd}(k), {}^B R_{sd}(k))$  によってロボットの目標関節角度  $\theta_d(k)$  を算出する. 図 6.8 に示す制御装置  $G_c(z)$  は

$$G_c(z) = K_P + K_D(1 - z^{-1}) \dots\dots\dots (6.41)$$

で表される. ただし,

$$K_P = \text{diag}[K_{P1} \ K_{P2} \ \dots \ K_{P6}] \dots\dots\dots (6.42)$$

$$K_D = \text{diag}[K_{D1} \ K_{D2} \ \dots \ K_{D6}] \dots\dots\dots (6.43)$$

### 6.4.2 シーリングロボットシステム

追従実験に用いたロボットは既報<sup>(10)~(12)</sup>にも使用した安川電機(株)製 6 自由度マニピュレータ Motoman K3S である. マニピュレータを駆動するためのモータをパーソナルコンピュータ PC9821Xv20 (日本電気 (株)) で制御する. また, CCDカメラで撮影した画像は画像入力ボード Meteor (Matrox (株)) を介して画像処理用パーソナルコンピュータ Dimension XPS R400 (DELL (株)) のメインメモリに転送され, 処理される.

ロボットの制御サンプリング周期  $T$  が 5 ms であるのに対し, 画像処理サンプリング周期と制御サンプリングとの比  $m$  を 11 に設定する. ビジュアルフィードバック制御を行うには, 制御側のコンピュータは画像処理側のコンピュータとの間の同期を取る必要がある. 同期通信の方法を図 6.10 に示す.

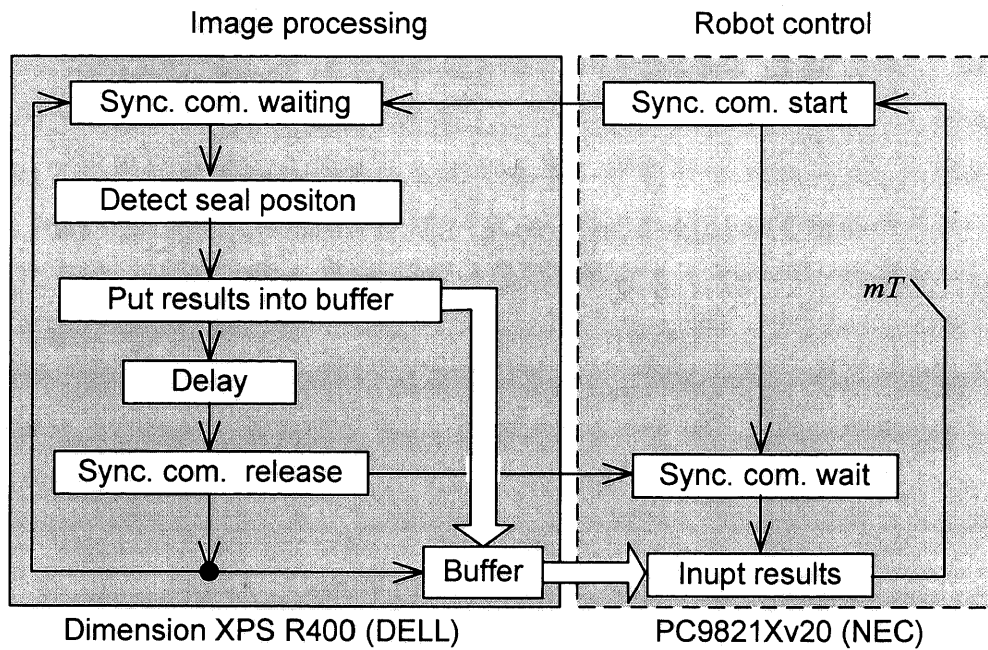


Fig.6·10 Synchronous communication (Sync. com.) between the computer for robot control and the computer for image processing

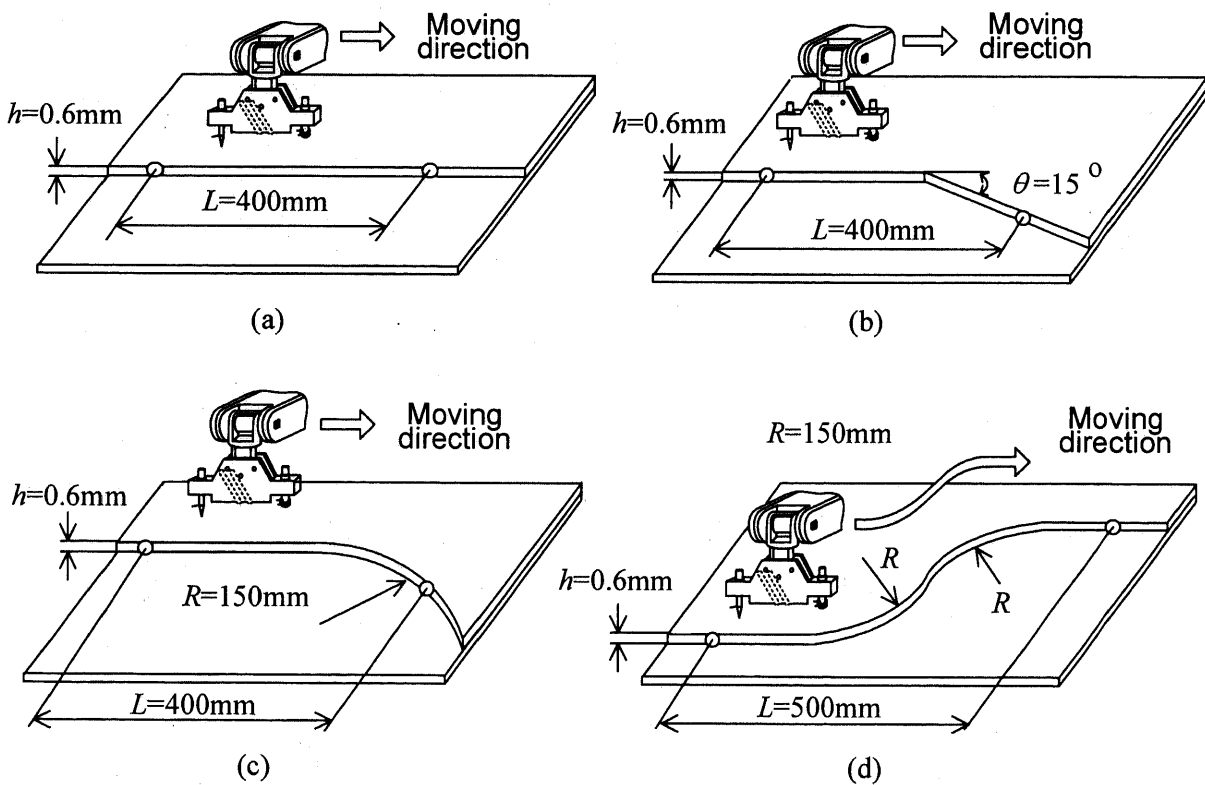


Fig. 6·11 Four types of sealing plates used in tracking tests

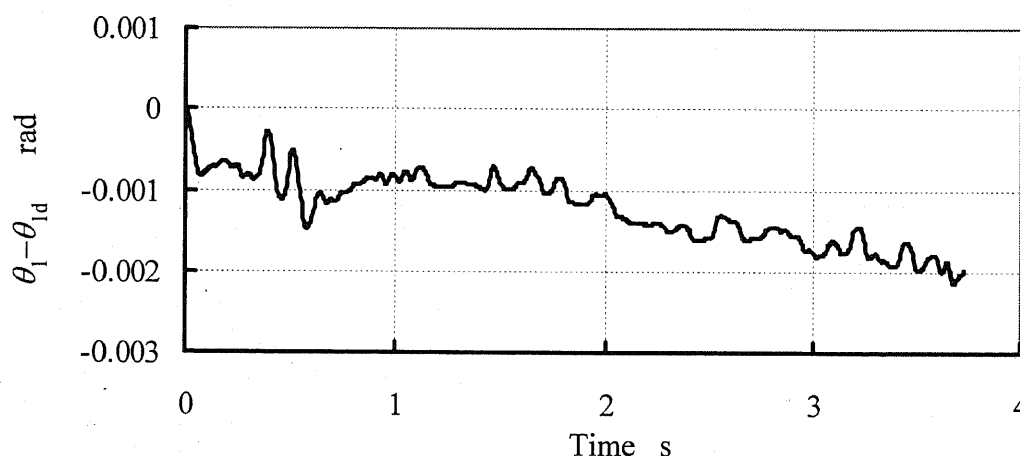
### 6.4.3 追従実験と結果

視覚センサの設置においては，スリットレーザの傾斜角度  $\phi$  を  $45^\circ$  に，図 6.2 に示す 2 本のスリットレーザ光源間の距離  $L_{CD}$  を 21 mm に設定した．シール対象は厚さ 0.6 mm のアルミ板を重ね合わせたものである．作業中ロボット手先の追従速度を 0.2m/s~0.4m/s に設定して実験を行った．式(6.42)~(6.43)に示す制御パラメータの値を表 6.1 に示す．また，図 6.11 に示す 4 種類の形状をもつシール軌道はシーリング対象として追従実験に用いた．

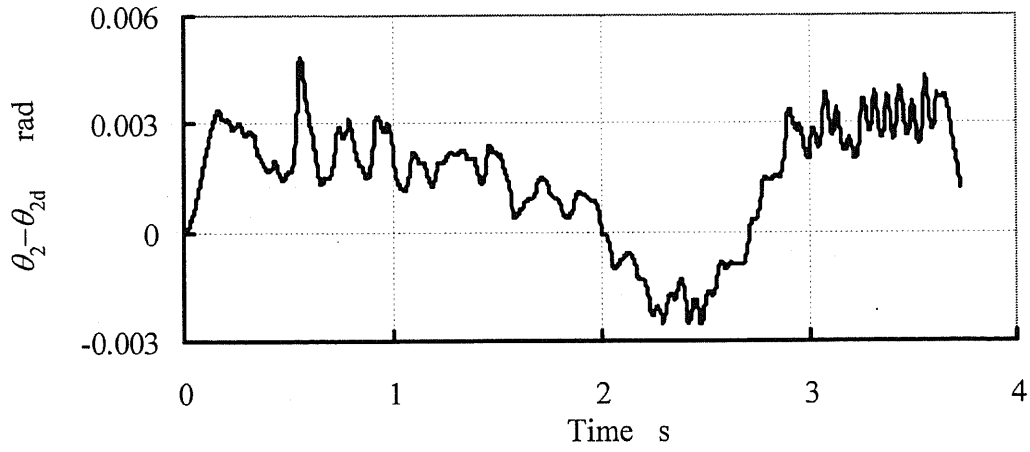
追従実験の結果の一例を図 6.12 に示す．また，これらのシーリング対象に対しての実験結果は追従誤差の最大値を 2mm 以下に抑えたため，追従作業がほぼ正確に行われたことを確認した．

Table 1 Control parameters used in the tracking tests

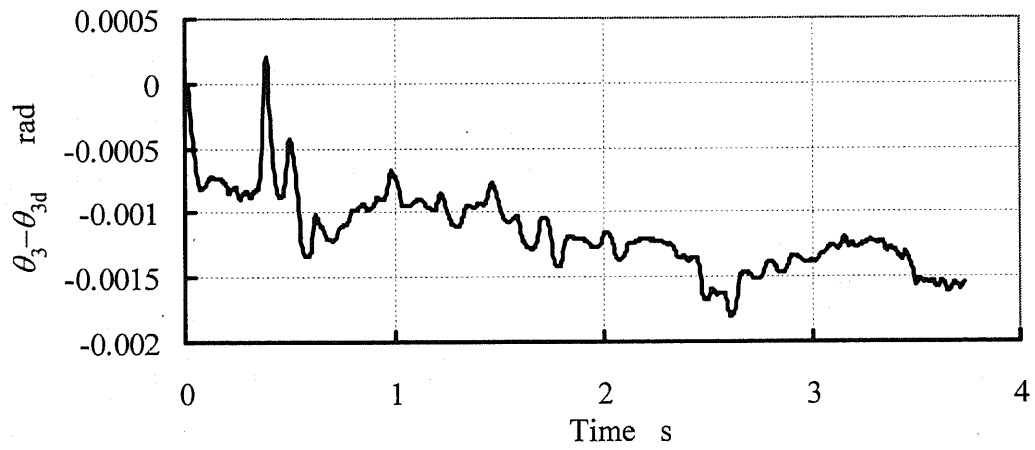
Joint $j$	$K_{Pj}$ V/rad	$K_{Dj}$ V/rad
1	7.362	2.964
2	5.790	1.987
3	6.912	2.100
4	0.194	0.032
5	1.809	0.230
6	0.617	0.075



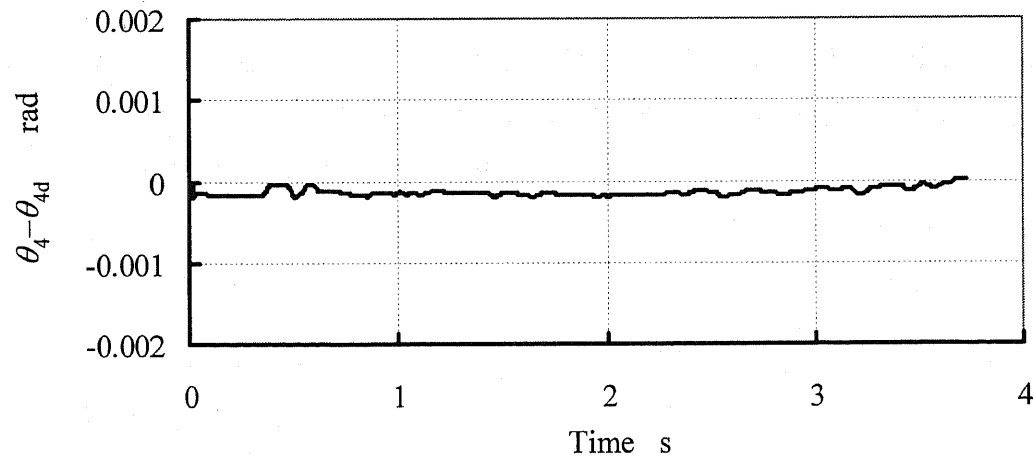
(a) Joint angle  $\theta_1 - \theta_{1d}$



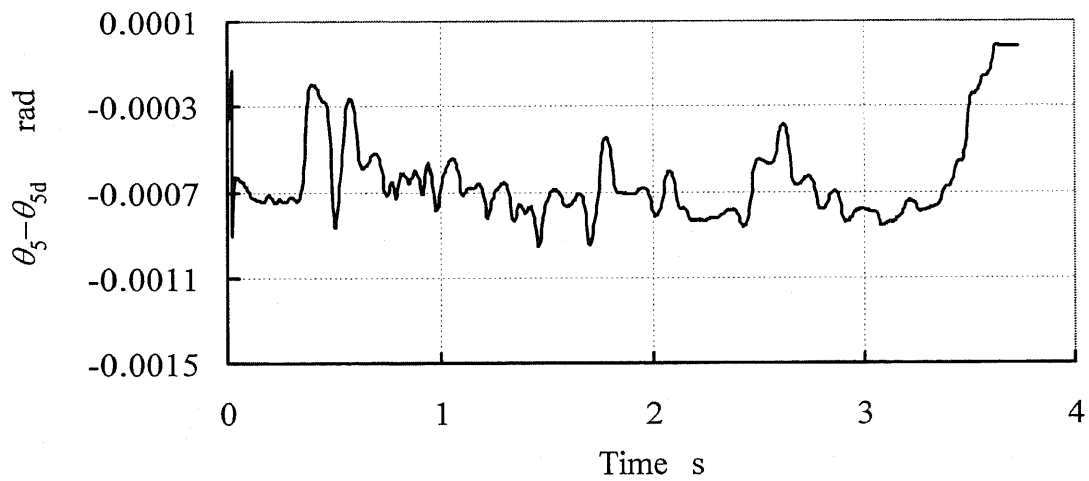
(b) Joint angle  $\theta_2 - \theta_{2d}$



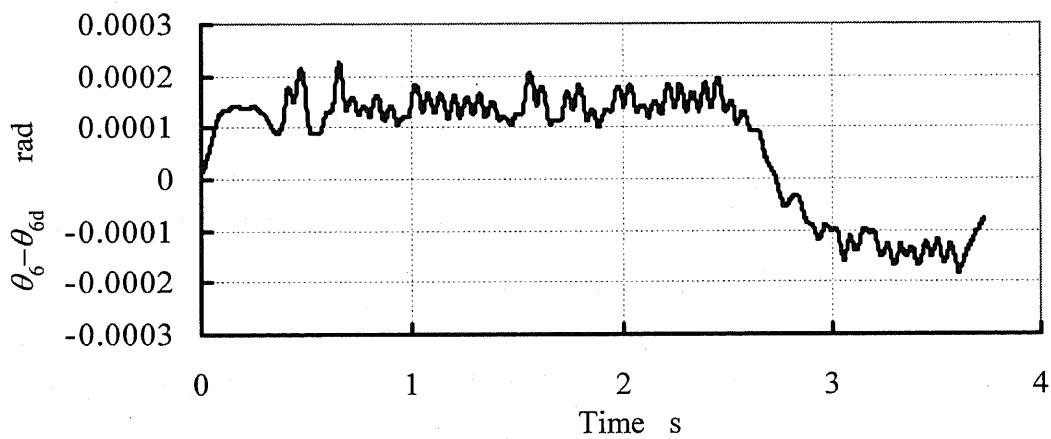
(c) Joint angle  $\theta_3 - \theta_{3d}$



(d) Joint angle  $\theta_4 - \theta_{4d}$



(e) Joint angle  $\theta_5 - \theta_{5d}$



(f) Joint angle  $\theta_6 - \theta_{6d}$

Fig. 6·12 Tracking error between actual joint angle and desired joint angle obtained in tracking a curved sealing model described in Fig.6·11(d) at a tracking speed of 0.25m/s

## 6.5 要約

本章で得られた主な結果は次のとおりである。

既報<sup>(10)</sup>で提案した教示不要という制御方式では、シール軌道が滑らかであるという前提に立っており、シール軌道が曲がる部分で視覚センサが軌道を見失うという問題があった。しかし、次々に流れてくる作業対象の形状の多くは事前に分かっている。それらの形状特徴を前もって学習させ、作業中にあらゆる形状のシール軌道を教示する必要はなく、視覚センサの未来位置を予測できるという点に本手法は基づいている。本章では、シーリングロボットの手先に取り付けられたノズルの位置と視覚センサ位置を同時に制御する方法を提案し、実験によってその有効性を実証した。本研究で得られた主な結果は次のとおりである。

(1) ノズルの目標軌道の生成においては、既に検出された継目の位置情報でノズルの目標軌道を生成する方法を用いた。視覚センサの目標軌道を生成するには、ニューラルネットワークによってファジィ推論のルールを学習させたのち、学習済みのファジィ推論を用いて視覚センサ目標軌道を生成する方法を提案し、実験により本方法の有効性を確かめた。

(2) ロボットの制御サンプリング周期ごとにノズルの目標軌道を視覚座標系から見た回転量に変換し、視覚センサの姿勢を補正することによってノズルの位置制御を実現する方法を提案した。これにより、(a)ノズルの位置と視覚センサの位置を制御サンプリング周期ごとに制御することができること、(b)ノズルと視覚センサはシール軌道から外れることがなく追従することを実験により確かめた。

(3) 教示を必要としない制御方式を実現するためのビジュアルフィードバック制御系を提案し、その効果を実験により確かめた。



---

## 第7章

### 結論

---

本研究では、ロボットによる柔軟物体のマニピュレーションという分野の基盤技術を開発した。剛体対象の場合には、ロボットと対象物体との接触力を力覚センサによって測定すれば、ロボットによる作業を容易に遂行できる。しかしながら、柔らかく変形しやすい物体の場合、接触によって対象物体が変形するため、力覚センサを用いただけでは作業の遂行は困難である。これらの問題を解決するため、本研究では、視覚と力覚の情報をフュージョンさせ、ロボットを知的制御することによって柔軟物体を操作する手法を提案した。また本研究で提案した手法は、視覚情報を用いるため、柔軟物体や柔軟ツールの材質に関するパラメータを必要としないという長所を持っている。さらに、本研究では、①接近してくる不特定な対象を柔軟に回避する手法、②ロボットの手先に取り付けられているノズルと視覚センサの目標軌道を知的手法によって生成し、同時にシール軌道に追従させる手法、を提案し実験によってその効果を実証した。これにより環境情報が不足している場合でもロボットの知的制御を行うことによって汎用性と実用性の高い制御系を構成することができた。

本論文により得られた主な結論は、下記のとおりである。

第2, 3, 4章においては、対象物体やロボットの手先ツールの変形問題を取り上げ、ロボットの制御手法を提案し、実験によって下記の結論が得られた。

(1) 変形物体の操作において、視覚と力覚の情報をを用いたロボットの制御アルゴリズムを提案し、二次元の変形を伴う“ペグインホール作業”に適用することによってその有効性を実証した。この手法をさらに発展させ、ニューラルネットワークの学習によって視覚と力覚の情報をフュージョンさせ、その結果をロボットの制御に用いるという手法を提案した。この手法を用いて三次元空間で変形する棒状物体の“はめあい作業”をロボットに行わせることによってその有効性を確かめた。

(2) ニューラルネットワークのオンライン学習を用い、マニピュレータがもつ柔軟ツール先端の位置と力のハイブリッド制御を行うという手法を提案し、柔軟ツールの先端を曲線軌道になぞらせる実験を行い、提案した手法の有効性を実証した。

(3) 柔らかく変形しやすい対象物体やツールの変形をモデルによって解析することは困難な場合、視覚情報を用いた本手法は、作業対象の剛性パラメータの値を必要としないため、異なる材質の対象やツールを扱う作業に提案した手法を適用できるという利点がある。

(4) 画像を高速で処理するためには、DSPを有する画像入力・処理ボードを用いて画像情報をリアルタイムで処理するアルゴリズムを開発した。画像情報のリアルタイム処理を用いることによってロボットの視覚フィードバック制御系を構成し、実験によりその有効性を確かめた。

第5, 6章においては、環境の情報が不足している場合に対しロボットの衝突回避や未来の目標軌道を生成する手法を提案し、実験によって下記の結論が得られた。

(1) 接近してくる人間や障害物が仮想ポテンシャル場に入ったことを視覚情報によって検出したとき、仮想関節トルクを生成させるという手法を提案し、実験によりその効果を実証した。

(2) 冗長ロボットの可操作性を考慮しながら手先の作業と衝突回避を同時に行うという制御手法を提案し、実験によりその効果を実証した。

(3) ノズルの目標軌道の生成においては、既に検出された継目の位置情報によってノズルの目標軌道を生成する方法を用いた。視覚センサの目標軌道を生成するには、ニューラルネットワークによってファジィ推論のルールを学習させたのち、学習済みのファジィ推論を用いて視覚センサの目標軌道を生成する方法を提案し、実験により本方法の有効性を確かめた。

(4) ロボットの制御サンプリング周期ごとにノズルの目標軌道を視覚座標系から見た回転量に変換し、視覚センサの姿勢を補正することによってノズルの位置制御を実現する方法を提案した。これにより、(a)ノズルの位置と視覚センサの位置を制御サンプリング周期ごとに制御することができること、(b)ノズルと視覚センサはシール軌道から外れることがなく追従することを実験により確かめた。

(5) 教示を必要としない制御方式を実現するためのビジュアルフィードバック制御系を提案し、その効果を実験により確かめた。

---

# 謝 辞

---

平成8年度から平成15年度まで8年間に本研究を進めるにあたり、終始懇切なご指導を賜った横浜国立大学大学院 藤堂 勇雄 教授に心から深く感謝の意を表し、厚くお礼を申し上げます。

機会あるたびごとに貴重な助言と懇切なご指摘をいただいた 藪田 哲郎 教授にも厚くお礼申し上げます。また、本論文をまとめるにあたり、貴重なご教示をいただいた 後藤 敏行 教授、高田 一 教授、眞田 一志 助教授に深く感謝申し上げます。

本研究および実験を行うにあたり、有益なご意見をいただいたトキコ株式会社 内海 典之 氏に謝意を表します。

また、工場の作業や研究室内でのサポートをしてくださった 桑原 弘 技官にも感謝します。また、実験の準備やソフトウェアの開発に協力してくださった 常盤 孝徳 氏、松浦 稔 氏、村松 勇夫 氏に謝意を表します。

おわりに、平成8年度～平成15年度まで藤堂研究室に在籍し、研究にご助力くださった諸氏に厚く感謝の意を表します。

---

# 文 献

---

- (1) 平井慎一, 柔軟物操作, 日本ロボット学会誌, Vol.16, No.2 (1998), pp.136-139.
- (2) 中垣博文, 平板や電線のような柔軟物体の挿入作業, 日本ロボット学会誌, Vol.16, No.2(1998), pp.159-162.
- (3) 平井慎一・野口英昭, 変形しやすい管状物体の挿入作業における人間の運動と作業状態識別則のロボットマニピュレータへの移植, 日本ロボット学会誌, Vol.15, No.8 (1997), pp.1172-1179.
- (4) 中垣博文・北垣高成・小笠原司・築根秀男, 画像トラッキングを用いた線状柔軟物体の先端力推定と挿入作業への応用, 日本ロボット学会誌, Vol.15, No.3 (1997), pp.422-430.
- (5) 若松栄史・和田隆広, 線状物体のモデリング, 日本ロボット学会誌, Vol.16, No.2 (1998), pp.145-148.
- (6) 呉建青・羅志偉・山北昌毅・伊藤宏司, 未知な柔軟曲面上での倣い作業におけるマニピュレータの動的位置/力制御, 日本機械学会論文集, C編, 63巻, 607号(1997), pp.937-944.
- (7) 小野栄一, 布のマニピュレーション, 日本ロボット学会誌, Vol.16, No.2 (1998), pp.149-153.
- (8) 久田俊明, 有限要素法によるモデリング, 日本ロボット学会誌, Vol.16, No.2 (1998), pp.140-144.
- (9) Chen, C. Y. and Zheng, Y. F. , Deformation Identification and Estimation of One Dimensional Objects by Vision Sensors, J. Robotic System, Vol.9, 5 (1992), pp.595-612.
- (10) 張曉林・藤堂勇雄, 距離情報と濃淡情報の統合に基づく継目位置検出法を用いたシーリングロボットの制御, 日本機械学会論文集, C編, 60巻, 577号(1994), pp.3130-3137.
- (11) 張曉林・藤堂勇雄, 視覚センサを用いたシーリングロボットの作業軌道の自動教

- 示法, 日本機械学会論文集, C編, 62巻, 593号(1996), pp.194-201.
- (12)黄 健・藤堂勇雄・牧謙一郎, 視覚情報を用いたシーリングロボットの目標軌道の実時間生成法と制御, 日本機械学会論文集, C編, 64巻, 624号(1998), pp.3047-3054.
- (13)黄 健・藤堂勇雄・常盤孝徳, シーリングロボットの目標軌道の実時間生成法と制御(オフライン学習を用いた視覚センサの未来位置の予測), 日本機械学会論文集, C編, 66巻, 648号(2000), pp.2669-2676.
- (14)タチバナ ルイス シロウ・藤堂勇雄, 視覚情報による並進回転分離形DDロボットの位置と力の制御, 日本機械学会論文集, C編, 65巻, 640号(1999), pp.4750-4758.
- (15)タチバナ ルイス シロウ・藤堂 勇雄, 視覚と力覚情報のオンライン融合を用いたDDロボットのハイブリッド制御, 日本機械学会論文集, C編, 67巻, 656号(2001-4), pp.1060-1068.
- (16)イヴァンゴドレール・山下忠・相良慎一・高橋佳之・和田和実・羅飛, はめあい作業のハイブリッド制御:シミュレーションと実験, 日本ロボット学会誌, Vol.12, No.1 (1994), pp.133-139.
- (17)井手浩二・宮崎文夫・安斗 星, 力覚情報を用いたはめあい作業の学習, 日本ロボット学会誌, Vol.10, No.1 (1992), pp.118-127.
- (18)末広尚士・高瀬国克, 接触運動の表現と制御およびその組立作業への応用, 日本ロボット学会誌, Vol.6, No.6 (1988), pp.31-38.
- (19)黄 健・藤堂勇雄, 視覚と力覚のセンサフュージョンによるロボットの制御(変形を伴う物体のマニピュレーション), 日本機械学会論文集, C編, 67巻, 660号(2001), pp.2616-2623.
- (20)Cichocki, A. and Unbehauen, R, Neural Networks for Optimization and Signal Processing, (1993), pp.122-135, John Wiley & Sons.
- (21)劉明輝・藤堂勇雄, ニューラルネットワークによるサーボ系のデジタル制御(オフライン学習の一手法), 日本機械学会論文集, C編, 57巻, 539号(1991), pp.2256-2262.
- (22)Traver, V. J., del PoBil, A. P. and Pérez-Francisco, M., Making Service Robots Human-Safe, Proc. 2000 IEEE/RSJ (2000), pp.696-701.
- (23)Martínez-Salvador, B., del PoBil, A. P. and Pérez- Francisco, M., A Hierarchy of Detail

for Fast Collision Detection, Proc. 2000 IEEE/RSJ (2000), pp.745-750 .

- (24)人間共存型ロボット研究専門委員会, 人間共存型ロボットシステムにおける技術課題, Vol.16, No.3(1998), pp.288-294.
- (25)Yoshikawa, T., Manipulability of Robotic Mechanisms, Int. J. of Robotics Research, Vol.4, No.2 (1985), pp.3-9.
- (26)Nakamura, Y. and Hanafusa, H., Inverse Kinematics Solutions with Singularity Robustness for Robot Manipulator Control, Trans. ASME, J. Dyn. Syst., Meas. and Control, Vol.108, No.4(1986), pp.163-171.
- (27)古荘純次・臼井広幸・佐野明人, 障害物の影響を考慮したマニピュレータの可操作度, 日本ロボット学会誌, Vol.6, No.3 (1988), pp.12-20.
- (28)古荘純次・臼井広幸, 冗長性を有するマニピュレータの一制御方式(障害物環境下における制御), 日本機械学会論文集, C編, 55巻, 514号(1989), pp.1391-1398.
- (29)彭志新・足立紀彦・中嶋新一, 冗長ロボットマニピュレータの拡張ハイブリッド制御, 日本機械学会論文集, C編, 57巻, 536号(1991), pp.1273-1278.
- (30)見浪護・内藤康弘・朝倉俊行, 冗長マニピュレータの回避可操作性, 日本ロボット学会誌, Vol.17, No.6 (1999), pp.887-895.
- (31)馬書根・ネンチェ, D., 冗長マニピュレータのための実時間動的冗長性制御法, 日本ロボット学会誌, Vol.13, No.5 (1995), pp.122-128.
- (32)馬書根・広瀬茂男・吉灘裕, マニピュレータの効率的な冗長制御法, 日本ロボット学会誌, Vol.14, No.5 (1996), pp.703-709.
- (33)辻敏夫・高橋利彰・伊藤宏司, 手先拘束を考慮したマニピュレータの多点コンプライアンス制御, 計測自動制御学会論文集, Vol.27, No.1 (1991), pp.85-92.
- (34)横井一仁・前川仁・谷江和雄, 冗長ロボットアームのためのコンプライアンス制御の一手法, 日本ロボット学会誌, Vol.11, No.1 (1993), pp.121-130.
- (35)辻敏夫・Achmad, J., 冗長マニピュレータのインピーダンス制御:冗長性を利用した関節インピーダンスの調節, 日本ロボット学会誌, Vol.12, No.4 (1994), pp.609-615.
- (36)永井清・吉川恒夫, 冗長マクロ・マイクロ・マニピュレータのインピーダンス制御, 日本ロボット学会誌, Vol.12, No.5 (1994), pp.122-128.
- (37)見浪護・野村佳弘・朝倉俊行, 冗長マニピュレータの未知対象物に対する軌道追従および回避制御のための予見/後見制御系, 日本ロボット学会誌, Vol.15, No.4

- (1997), pp.573-580.
- (38)新井民夫・太田順, 仮想的なインピーダンスを用いた複数移動ロボット系の動作計画, 日本ロボット学会誌, Vol.11, No.7 (1993), pp.1039-1046.
- (39)三河正彦・吉田耕一・丹野瑞紀・松本三千人, 障害物回避のための視覚に基づくマニピュレータ冗長制御と画像特徴推定, 日本ロボット学会誌, Vol.17, No.4 (1999), pp.534-539.
- (40)細田耕・坂本健二・浅田稔, 3次元再構成を行わない視覚サーボ系のための障害物回避軌道の生成, 日本ロボット学会誌, Vol.15, No.2 (1997), pp.290-295.
- (41)Castano, A. and Hutchinson, S., IEEE Trans. Robotics and Automation, Vol.10, No.3 (1994), pp.334-342.
- (42)辻敏夫・畑木道生・赤松寛優・金子真, マニピュレータの非接触型インピーダンス制御, 日本ロボット学会誌, Vol.15, No.4 (1997), pp.616-623.
- (43)野口裕史・辻敏夫・原田研介・金子真, ニューラルネットワークによる非接触型インピーダンス制御のパラメータ学習, 日本ロボット学会誌, Vol.18, No.4 (2000), pp.561-568.
- (44)Craig, J.J. (三浦宏文・下山勲 訳), ロボティクス (機構・力学・制御), (1991), pp.147-151, 共立出版.
- (45)内山勝・清水邦俊・箱守京次郎, ヤコビ行列式によるロボットアームの機構評価, 計測自動制御学会論文集, Vol.21, No.2 (1985), pp.190-196.
- (46)Sawano, S., Ikeda, J., Utsumi, N., Ohtani, Y., Kikuchi, A., and Ito, Y., Robotica, No.2 (1984), pp.41-46.
- (47)石黒一也, 視覚センサによるロボットシステム, ロボット, No.49(1985), pp.82-87.
- (48)国清巧・相澤憲輔・菅野清, 車体位置ずれ検知装置付シーリングロボットシステム, ロボット, No.77(1990), pp.86-95.
- (49)荒川賢一・柿崎隆夫・大名新治, トラッキングロボットによるシーリング作業のための目標点高速検出, 日本ロボット学会誌, Vol.16, No.3 (1998), pp.376-383.
- (50)張曉林・沢野進・富川英亮・藤堂勇雄, シーリングロボットのためのニューラルネットワークを用いた継目位置検出法, 日本機械学会論文集, C編, 59巻, 563号(1993), pp.2191-2197.
- (51)菅野道夫, ファジィ制御, (1988), 日刊工業新聞社.

- (52) 大島健司・山根敏, ニューロ・ファジィ制御によるロボット溶接の適応制御技術, 溶接学会誌, Vol.63, No.7 (1994), pp.38-44.
- (53) Guely, F. and Todo, I., A New Fuzzy Self-Organizing Feedforward Control Algorithm with Feedback and its Application to the Control of a Direct Drive Robot Arm, Proc. of 1990 JAPAN-USA Symposium on Flexible Automation, (1990-7), pp.661-668.
- (54) 細田耕・浅田稔, 構造やパラメータに関する先験的な知識を必要としないフィードフォワード補償器を持つ適応型ビジュアルサーボ系の構成, 日本ロボット学会誌, Vol.14, No.2 (1996), pp.313-319.



---

# 本論文を構成する公表論文

---

## 1. 研究論文 (全文査読)

- (1) 黄 健・藤堂勇雄・常盤孝徳：シーリングロボットの目標軌道の実時間生成法と制御 (オフライン学習を用いた視覚センサの未来位置の予測), 日本機械学会論文集, C編, 66 巻, 648 号(2000-8), pp.2669-2676.
- (2) 黄 健・藤堂勇雄：視覚と力覚のセンサフュージョンによるロボットの制御 (変形を伴う物体のマニピュレーション), 日本機械学会論文集, C編, 67 巻, 660 号(2001-8), pp.2616-2623.
- (3) 黄 健・藤堂勇雄・松浦 稔：視覚情報を用いた冗長ロボットの制御 (接近してくる障害物の回避制御), 日本機械学会論文集, C編, 68 巻, 674 号(2002-10), pp.2999-3006.
- (4) 黄 健・藤堂勇雄・村松勇夫：視覚と力覚のセンサフュージョンによるロボットのニューロ制御 (棒状柔軟物体の操作), 日本機械学会論文集, C編, 69 巻, 684 号(2003-8), pp.2085-2092.
- (5) 黄 健・藤堂勇雄：マニピュレータがもつ柔軟ツールの位置と力のハイブリッド制御, 日本機械学会論文集, C編, (掲載決定)

## 2. 国際会議論文 (全文査読)

- (1) Huang, J. and Todo, I., A Human-Safe Control for Collision Avoidance by a Redundant Robot Using Visual Information, Proceedings of the 1st International Symposium on Systems & Human Science – For Safety, Security, and Dependability –, Osaka, Japan (2003-11), pp.212-217.

## 3. 国内学会講演論文

(全文査読)

- (1) 黄 健・藤堂勇雄・長谷川 徹：立体視情報と力覚情報を用いたロボットによる動的作業とその制御, 第6回ロボティクス・シンポジウム予稿集, (2001-3), pp.211-216.

(査読なし)

- (1) 黄 健・藤堂勇雄・常盤孝徳：視覚情報を用いたカーブ軌道を有するシーリング作業のためのロボットの目標軌道の実時間生成法と制御,日本機械学会 1999 年度年次大会講演論文集(V), No.99-1(1999-7), pp.73-74.
- (2) 黄 健・藤堂勇雄・村松勇夫：視覚と力覚のセンサフュージョンによるロボットの制御 (変形物体の操作), 日本機械学会 2000 年度年次大会講演論文集(II), No.00-1, (2000-8), pp.585-586.
- (3) 黄 健・藤堂勇雄・松浦 稔：視覚情報を用いた冗長ロボットの動的障害物の回避制御, 日本機械学会 関東支部第 7 期総会講演会講演論文集, No.010-1,(2001-3),pp.285-286.
- (4) 黄 健・藤堂勇雄・岡 大蔵：視覚と力覚のセンサフュージョンによるロボットのニューロ制御 (変形物体の操作), 日本機械学会 2001 年度年次大会講演論文集(IV), No.01-1, (2001-8), pp211-212.
- (5) 黄 健・藤堂勇雄・村松勇夫：視覚情報と力覚情報を用いたロボットによる線状変形物体の操作, 日本機械学会ロボティクス・メカトロニクス講演会 (ROBOMECH'02), 1P1-I06, CD-ROM, (2002-6).
- (6) 黄 健・藤堂勇雄・尾崎毅史：柔軟ツールをもつマニピュレータの位置と力のハイブリッド制御, 日本機械学会ロボティクス・メカトロニクス講演会 (ROBOMECH'03), 1A1-2F-E2, CD-ROM, (2003-5).

---

# 付記 A

## 可搬式汎用知能アームの機構と運動解析

---

### A・1 概説

本論文で用いる可搬式汎用知能アーム型ロボットは、7自由度をもつマニピュレータである。本章では、このロボットの構成について説明し、運動学解析を行う。

### A・2 マニピュレータの構成

図 A・1 に示すように、本論文で用いる可搬式汎用知能アームは、各関節に減速機付モータを備えた7自由度マニピュレータである。このマニピュレータの手首部にセンサやツールなど作業道具を取り付けることは可能である。通常、マニピュレータの手首部とツールの間に力覚センサ（以下 F/T センサと呼ぶ）を取り付け、作業対象物からロボット手首に作用する力覚情報を測定する。また、本論文では、図 A・1 に示すマニピュレータの根元の関節から手首部までの各関節を  $S1, S2, S3, E1, E2, W1, W2$  のような記号で記述する。また、各関節の回転角度は  $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6, \theta_7$  によって表される。

### A・3 座標系の定義

本論文に用いるマニピュレータは各関節に応じて異なる座標系をもっており、これらの座標系を図 A・2 に示す。また、力覚センサやツールなど作業道具はそれぞれの座標系をもっているため、本論文でこれらの座標系を次のように記述する。

- ベース座標系  $\Sigma_b$  : マニピュレータの基準座標系。  $\Sigma_b$  はすべての座標系の基準であり、変化しない。
- 手首座標系  $\Sigma_m$  : マニピュレータ手首部の座標系。
- F/T センサ座標系  $\Sigma_f$  : 手首部に取り付けられる F/T センサの座標系。
- ツール座標系  $\Sigma_t$  : ロボット手首部に取り付けられるツールの座標系。
- 視覚センサ座標系  $\Sigma_c$  : 視覚センサの位置・姿勢を表す座標系。視覚センサ 2 台がある場合、それぞれの座標系を  $\Sigma_{c1}, \Sigma_{c2}$  とする。

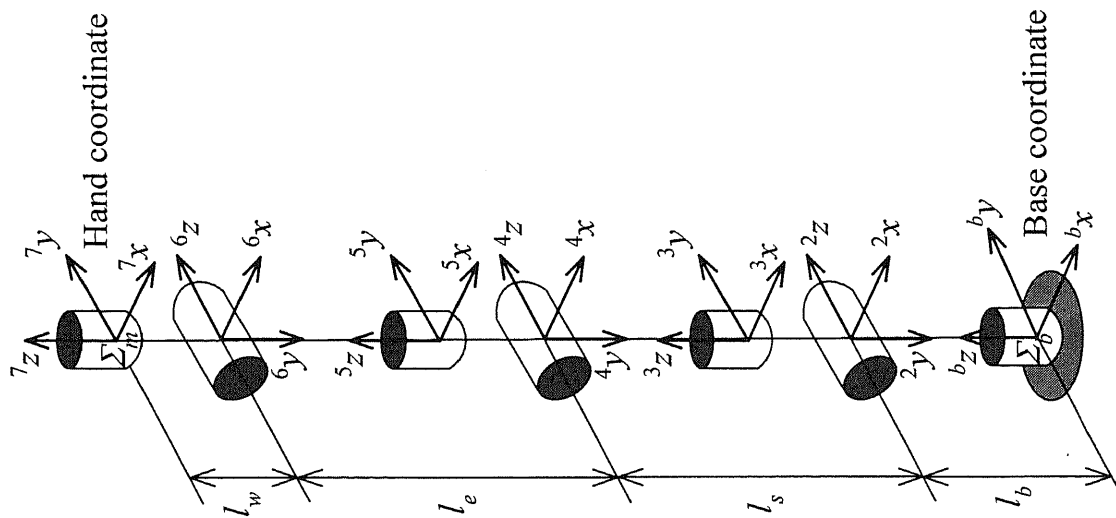


Fig.A•2 Coordinate definition of the 7 DOF manipulator PA10

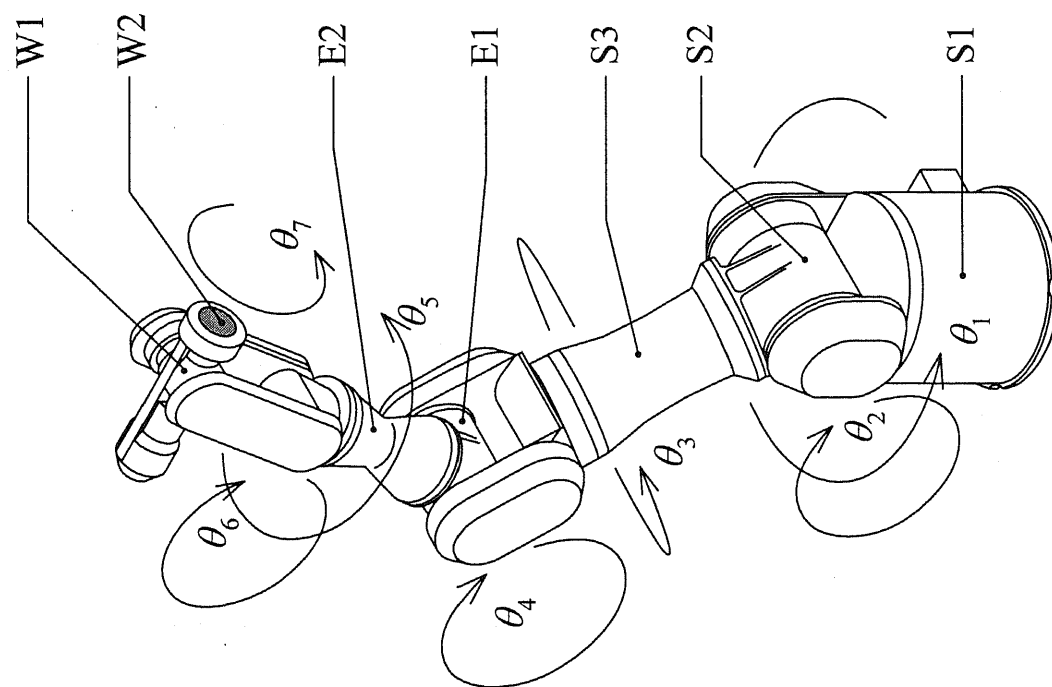


Fig.A•1 A 7 DOF manipulator

## A・4 マニピュレータの運動学

一般に、マニピュレータが三次元空間で任意の位置・姿勢を取るためには、6自由度を有することは必要である。6自由度以上の自由度をもつマニピュレータを冗長マニピュレータと分類する。図A・2に示すように本論文では、7自由度のマニピュレータを用いるため、このマニピュレータは1冗長自由度をもつマニピュレータである。

### A・4・1 x-y-z 固定角法による手首姿勢の記述

マニピュレータの姿勢を記述するとき、x-y-z 固定角法やx-y-z オイラー法など手法を用いた3個のパラメータによる記述が一般的である。これらの方法では、姿勢を位置要素と対等に扱えるため、制御系を構成しやすいという利点がある。本論文では、x-y-z 固定角法を用いて姿勢を表し、制御系を構成する。図A・3に示すように、x-y-z 固定角法によって姿勢を記述する。

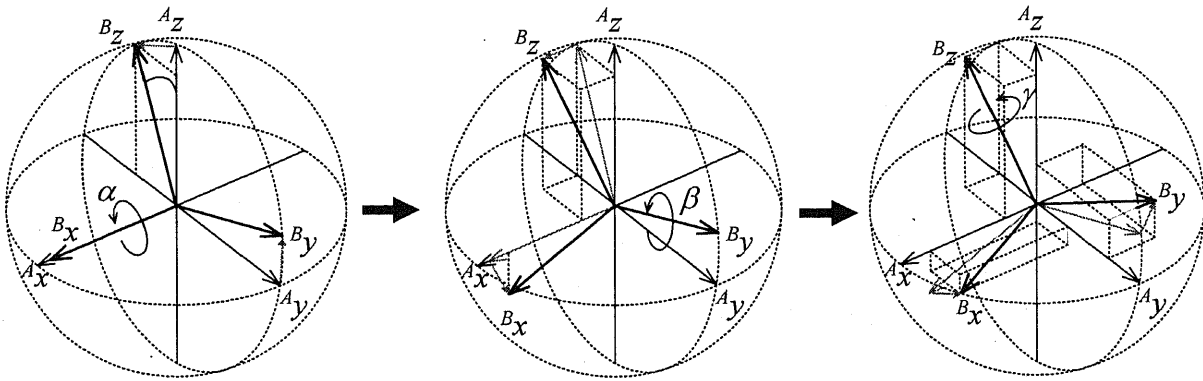


Fig.A・3 Orientation definition by x-y-z fixed angle method

x-y-z 固定角法では、求める姿勢を表す座標系  $\Sigma_B$  を基準座標系  $\Sigma_A$  に一致させた状態から  $A_x$  軸まわりに  $\alpha$ 、 $A_y$  軸まわりに  $\beta$ 、 $A_z$  軸まわりに  $\gamma$  という順に回転させ、これによって座標系  $\Sigma_A$  に対する座標系  $\Sigma_B$  の姿勢を表す。したがって、回転角  $\alpha, \beta, \gamma$  を用いて回転行列  ${}^A_B R(\alpha, \beta, \gamma)$  を算出することができる。

$${}^A_B R(\alpha, \beta, \gamma) = \begin{bmatrix} c_\gamma & -s_\gamma & 0 \\ s_\gamma & c_\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\beta & 0 & s_\beta \\ 0 & 1 & 0 \\ -s_\beta & 0 & c_\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha \\ 0 & s_\alpha & c_\alpha \end{bmatrix}$$

$$= \begin{bmatrix} c_\beta c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma & c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma - c_\alpha c_\gamma & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma \\ -s_\beta & s_\alpha c_\beta & c_\alpha c_\beta \end{bmatrix} \dots\dots\dots (A\cdot 1)$$

**A・4・2 同次変換行列による手首の位置・姿勢の記述**

図 A・4 に示すように、基準座標系  $\Sigma_A$  から見た座標系  $\Sigma_B$  原点の位置ベクトル  ${}^A p_B$  は

$${}^A p_B = [{}^A x_B \quad {}^A y_B \quad {}^A z_B]^T \dots\dots\dots (A\cdot 2)$$

であり、位置・姿勢の 6 次元ベクトルは

$${}^A x_B = [x_B \quad y_B \quad z_B \quad \alpha \quad \beta \quad \gamma]^T \dots\dots\dots (A\cdot 3)$$

によって表される。

また、回転行列  ${}^A R_B$  と位置ベクトル  ${}^A p_B$  を用いて、同次変換  ${}^A T_B$  による位置・姿勢を表記することができる。

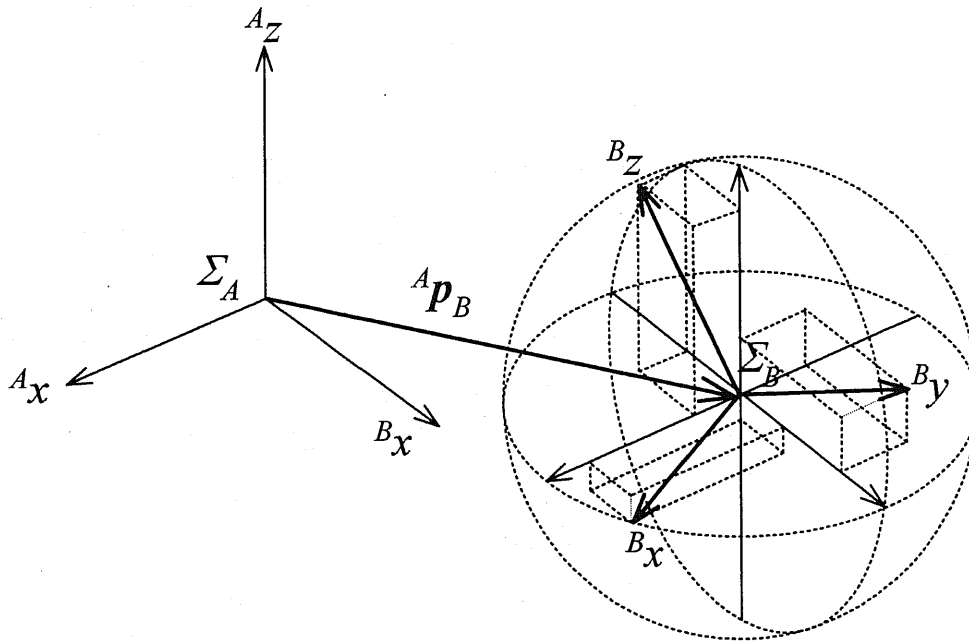


Fig.A・4 Expression of position vector and orientation matrix

同次変換行列  ${}^A T_B$  は、回転や平行移動など座標変換を表す。ここで、座標系  $\Sigma_A$  から座標系  $\Sigma_B$  へ変換する際、回転行列を  ${}^A R_B$ 、位置ベクトルを  ${}^A p_B$  としたとき、同次変換行列  ${}^A T_B$  は

$${}^A T_B = \left[ \begin{array}{ccc|c} & & & \\ & {}^A R_B & & {}^A P_B \\ & & & \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \dots\dots\dots (A\cdot4)$$

である。また、回転行列 ${}^A R_B$ は、主方向を示す単位ベクトルによって表されることができる。ここで、座標系 $\Sigma_A$ を基準にした座標系 $\Sigma_B$ の主方向を示す単位ベクトルは ${}^A X_B, {}^A Y_B, {}^A Z_B$ と表されると、回転行列 ${}^A R_B$ は

$${}^A R_B = \left[ \begin{array}{c|c|c} & & \\ \hline & & \\ \hline & & \\ \hline & & \\ \hline & & \\ \hline & & \\ \hline & & \\ \hline & & \end{array} \right] \dots\dots\dots (A\cdot5)$$

のように表すことができる。

### A・4・3 マニピュレータの順運動学

マニピュレータの運動学解析は、マニピュレータの関節角度から手首座標系 $\Sigma_m$ の位置・姿勢を求める。マニピュレータの関節角度ベクトル $\theta$ は

$$\theta = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4 \ \theta_5 \ \theta_6 \ \theta_7]^T \dots\dots\dots (A\cdot6)$$

とする。また、ベース座標系 $\Sigma_b$ における手首座標系 $\Sigma_m$ の位置・姿勢ベクトル ${}^b x_m$ を

$${}^b x_m = [x_m \ y_m \ z_m \ \alpha_m \ \beta_m \ \gamma_m]^T \dots\dots\dots (A\cdot7)$$

とする。

マニピュレータの個々のリンクに図 A・2 のような座標系を配置し、リンク間の同次変換行列を求める。ただし、本論文では、リンクパラメータを次の記号で表す。

- $l_i$ : 座標系 $\Sigma_i$ における座標系 $\Sigma_{i+1}$ の原点の $z$ 座標
- $d_i$ : 座標系 $\Sigma_i$ における座標系 $\Sigma_{i+1}$ の原点の $x$ 座標
- $\alpha_i$ :  ${}^i x$ 軸まわりに ${}^i z$ 軸と ${}^{i+1} z$ 軸がなす角
- $\theta_i$ :  ${}^i z$ 軸まわりに ${}^i x$ 軸と ${}^{i+1} x$ 軸がなす角

リンクパラメータを用いてリンク $i-1$ からリンク $i$ への同次変換行列 ${}^{i-1} T_i$ は

$${}^{i-1} T_i = \left[ \begin{array}{ccc|c} \cos\theta_i & -\sin\theta_i & 0 & d_{i-1} \\ \sin\theta_i \cdot \cos\alpha_{i-1} & \cos\theta_i \cdot \cos\alpha_{i-1} & -\sin\alpha_{i-1} & -\sin\alpha_{i-1} \cdot l_i \\ \sin\theta_i \cdot \sin\alpha_{i-1} & \cos\theta_i \cdot \sin\alpha_{i-1} & \cos\alpha_{i-1} & \cos\alpha_{i-1} \cdot l_i \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \dots\dots\dots (A\cdot8)$$

で表される。ここで、マニピュレータの各リンクのパラメータを運動学モデルによって求めて、表 A・1 に示す。

Table A·1 Parameters of the manipulator links

$i$	Link $i$	$l_i$	$d_i$	$\alpha_i$	$\theta_i$
1	Link 1	$l_b$	0	0	$\theta_1$
2	Link 2	0	0	$-\pi/2$	$\theta_2$
3	Link 3	$l_s$	0	$\pi/2$	$\theta_3$
4	Link 4	0	0	$-\pi/2$	$\theta_4$
5	Link 5	$l_e$	0	$\pi/2$	$\theta_5$
6	Link 6	0	0	$-\pi/2$	$\theta_6$
7	Link 7	$l_w$	0	$\pi/2$	$\theta_7$

表 A·1 に示すリンクパラメータを式(A·8)に代入し、各座標系間の同次変換行列は

$${}^bT_1 = \left[ \begin{array}{ccc|c} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & l_b \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \dots\dots\dots (A\cdot9)$$

$${}^1T_2 = \left[ \begin{array}{ccc|c} c_2 & -s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_2 & -c_2 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \dots\dots\dots (A\cdot10)$$

$${}^2T_3 = \left[ \begin{array}{ccc|c} c_3 & -s_3 & 0 & 0 \\ 0 & 0 & -1 & -l_s \\ s_3 & c_3 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \dots\dots\dots (A\cdot11)$$

$${}^3T_4 = \left[ \begin{array}{ccc|c} c_4 & -s_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_4 & -c_4 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \dots\dots\dots (A\cdot12)$$

$${}^4T_5 = \left[ \begin{array}{ccc|c} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & -1 & -l_e \\ s_5 & c_5 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \dots\dots\dots (A\cdot13)$$

$${}^5T_6 = \left[ \begin{array}{ccc|c} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_6 & -c_6 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \dots\dots\dots (A\cdot14)$$



$${}^6T_7 = \left[ \begin{array}{ccc|c} c_7 & -s_7 & 0 & 0 \\ 0 & 0 & -1 & -l_w \\ s_7 & c_7 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \dots \dots \dots (A \cdot 15)$$

で表される。ただし、

$$c_i = \cos \theta_i, \quad s_i = \sin \theta_i \quad (i=1,2,\dots,7) \dots \dots \dots (A \cdot 16)$$

である。

以上により、ベース座標系  $\Sigma_b$  から手首座標系  $\Sigma_m$  への変換行列  ${}^bT_m$  は

$$\begin{aligned} {}^bT_m &= {}^bT_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \cdot {}^4T_5 \cdot {}^5T_6 \cdot {}^6T_7 \\ &= \left[ \begin{array}{ccc|c} & & & x_m \\ & R_m & & y_m \\ & & & z_m \\ \hline 0 & 0 & 0 & 1 \end{array} \right] = \left[ \begin{array}{cccc|c} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \dots \dots \dots (A \cdot 17) \end{aligned}$$

によって求められる。ただし、

$$R_m = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \dots \dots \dots (A \cdot 18)$$

$$x_m = t_{14}, \quad y_m = t_{24}, \quad z_m = t_{34} \dots \dots \dots (A \cdot 19)$$

$$\begin{aligned} t_{11} &= ([\{(c_1c_2c_3 - s_1s_3)c_4 - c_1s_2s_4\}c_5 - (c_1c_2s_3 + s_1c_3)s_5]c_6 - \\ &\quad \{(c_1c_2c_3 - s_1s_3)s_4 + c_1s_2c_4\}s_6)c_7 + \dots \dots \dots (A \cdot 20) \\ &\quad [-\{(c_1c_2c_3 - s_1s_3)c_4 - c_1s_2s_4\}s_5 - (c_1c_2s_3 + s_1c_3)c_5]s_7 \end{aligned}$$

$$\begin{aligned} t_{12} &= -([\{(c_1c_2c_3 - s_1s_3)c_4 - c_1s_2s_4\}c_5 - (c_1c_2s_3 + s_1c_3)s_5]c_6 - \\ &\quad \{(c_1c_2c_3 - s_1s_3)s_4 + c_1s_2c_4\}s_6)s_7 + \dots \dots \dots (A \cdot 21) \\ &\quad [-\{(c_1c_2c_3 - s_1s_3)c_4 - c_1s_2s_4\}s_5 - (c_1c_2s_3 + s_1c_3)c_5]c_7 \end{aligned}$$

$$\begin{aligned} t_{13} &= [\{(c_1c_2c_3 - s_1s_3)c_4 - c_1s_2s_4\}c_5 - (c_1c_2s_3 + s_1c_3)s_5]s_6 + \dots \dots \dots (A \cdot 22) \\ &\quad \{(c_1c_2c_3 - s_1s_3)s_4 + c_1s_2c_4\}c_6 \end{aligned}$$

$$\begin{aligned} t_{14} &= l_w([\{(c_1c_2c_3 - s_1s_3)c_4 - c_1s_2s_4\}c_5 - (c_1c_2s_3 + s_1c_3)s_5]s_6 + \\ &\quad \{(c_1c_2c_3 - s_1s_3)s_4 + c_1s_2c_4\}c_6) + \dots \dots \dots (A \cdot 23) \\ &\quad l_e\{(c_1c_2c_3 - s_1s_3)s_4 + c_1s_2c_4\} + l_s c_1 s_2 \end{aligned}$$

$$t_{21} = ([\{(s_1c_2c_3 + c_1s_3)c_4 - s_1s_2s_4\}c_5 - (s_1c_2s_3 - c_1c_3)s_5]c_6 - \{(s_1c_2c_3 + c_1s_3)s_4 + s_1s_2c_4\}s_6)c_7 + \dots \quad (\text{A} \cdot 24)$$

$$[-\{(s_1c_2c_3 + c_1s_3)c_4 - s_1s_2s_4\}s_5 - (s_1c_2s_3 - c_1c_3)c_5]s_7$$

$$t_{22} = -([\{(s_1c_2c_3 + c_1s_3)c_4 - s_1s_2s_4\}c_5 - (s_1c_2s_3 - c_1c_3)s_5]c_6 - \{(s_1c_2c_3 + c_1s_3)s_4 + s_1s_2c_4\}s_6)c_7 + \dots \quad (\text{A} \cdot 25)$$

$$[-\{(s_1c_2c_3 + c_1s_3)c_4 - s_1s_2s_4\}s_5 - (s_1c_2s_3 - c_1c_3)c_5]c_7$$

$$t_{23} = [\{(s_1c_2c_3 + c_1s_3)c_4 - s_1s_2s_4\}c_5 - (s_1c_2s_3 - c_1c_3)s_5]s_6 + \dots \quad (\text{A} \cdot 26)$$

$$\{(s_1c_2c_3 + c_1s_3)s_4 + s_1s_2c_4\}c_6$$

$$t_{24} = l_w([\{(s_1c_2c_3 + c_1s_3)c_4 - s_1s_2s_4\}c_5 - (s_1c_2s_3 - c_1c_3)s_5]s_6 + \{(s_1c_2c_3 + c_1s_3)s_4 + s_1s_2c_4\}c_6) + \dots \quad (\text{A} \cdot 27)$$

$$l_e\{(s_1c_2c_3 + c_1s_3)s_4 + s_1s_2c_4\} + l_s s_1 s_2$$

$$t_{31} = [ \{(-s_2c_3c_4 - c_2s_4)c_5 + s_2s_3s_5\}c_6 - (-s_2c_3s_4 + c_2c_4)s_6 ]c_7 \dots \quad (\text{A} \cdot 28)$$

$$+ \{(s_2c_3c_4 + c_2s_4)s_5 + s_2s_3c_5\}s_7$$

$$t_{32} = -[\{(-s_2c_3c_4 - c_2s_4)c_5 + s_2s_3s_5\}c_6 - (-s_2c_3s_4 + c_2c_4)s_6]s_7 \dots \quad (\text{A} \cdot 29)$$

$$+ \{(s_2c_3c_4 + c_2s_4)s_5 + s_2s_3c_5\}c_7$$

$$t_{33} = \{(-s_2c_3c_4 - c_2s_4)c_5 + s_2s_3s_5\}s_6 + (-s_2c_3s_4 + c_2c_4)c_6 \dots \quad (\text{A} \cdot 30)$$

$$t_{34} = l_w[\{(-s_2c_3c_4 - c_2s_4)c_5 + s_2s_3s_5\}s_6 + \dots \quad (\text{A} \cdot 31)$$

$$(-s_2c_3s_4 + c_2c_4)c_6] - l_e(s_2c_3s_4 - c_2c_4) + l_s c_2 + l_b$$

#### A・4・4 マニピュレータの速度解析

本節では、マニピュレータの手首座標系  $\Sigma_m$  の速度解析を行い、マニピュレータ手先速度と関節角速度の関係を求める。また、手先速度と関節角速度の関係からヤコビ行列を導出する。

手首座標系  $\Sigma_m$  の速度ベクトルを  ${}^b \dot{\mathbf{x}}_m$ 、マニピュレータの関節角速度ベクトルを  $\dot{\boldsymbol{\theta}}$  とし、

$${}^b \dot{\mathbf{x}}_m = [\dot{x}_m \quad \dot{y}_m \quad \dot{z}_m \quad \dot{\alpha}_m \quad \dot{\beta}_m \quad \dot{\gamma}_m]^T \dots \quad (\text{A} \cdot 32)$$

$$\dot{\boldsymbol{\theta}} = [\dot{\theta}_1 \quad \dot{\theta}_2 \quad \dot{\theta}_3 \quad \dot{\theta}_4 \quad \dot{\theta}_5 \quad \dot{\theta}_6 \quad \dot{\theta}_7]^T \dots \quad (\text{A} \cdot 33)$$

によって表す。また、マニピュレータ手首のヤコビ行列  ${}^b \mathbf{J}_m(\boldsymbol{\theta})$  は、

$${}^b \dot{\mathbf{x}}_m = {}^b \mathbf{J}_m(\boldsymbol{\theta}) \cdot \dot{\boldsymbol{\theta}} = \begin{bmatrix} {}^b \mathbf{J}_{m1}(\boldsymbol{\theta}) \\ {}^b \mathbf{J}_{m2}(\boldsymbol{\theta}) \end{bmatrix} \cdot \dot{\boldsymbol{\theta}} \dots \quad (\text{A} \cdot 34)$$

で表される。ここで、ヤコビ行列  ${}^b \mathbf{J}_m(\boldsymbol{\theta})$  は  $6 \times 7$  の行列であり、 ${}^b \mathbf{J}_{m1}(\boldsymbol{\theta})$  と  ${}^b \mathbf{J}_{m2}(\boldsymbol{\theta})$  は 3

×7の行列である。また、 ${}^b\mathbf{x}_m$ の位置成分を関節角度 $\theta_i$ に対して偏微分することによって ${}^b\mathbf{J}_{m1}(\boldsymbol{\theta})$ を求める。すなわち、式(A・23)、(A・27)、(A・31)より、偏微分

$$x_m = l_w \{ [(c_1c_2c_3 - s_1s_3)c_4 - c_1s_2s_4]c_5 - (c_1c_2s_3 + s_1c_3)s_5 \} s_6 + \dots \dots \dots (A\cdot35)$$

$$[(c_1c_2c_3 - s_1s_3)s_4 + c_1s_2c_4]c_6 \} + l_e \{ (c_1c_2c_3 - s_1s_3)s_4 + c_1s_2c_4 \} + l_s c_1 s_2$$

$$y_m = l_w \{ [(s_1c_2c_3 + c_1s_3)c_4 - s_1s_2s_4]c_5 - (s_1c_2s_3 - c_1c_3)s_5 \} s_6 + \dots \dots (A\cdot36)$$

$$[(s_1c_2c_3 + c_1s_3)s_4 + s_1s_2c_4]c_6 \} + l_e \{ (s_1c_2c_3 + c_1s_3)s_4 + s_1s_2c_4 \} + l_s s_1 s_2$$

$$z_m = l_w \{ [(-s_2c_3c_4 - c_2s_4)c_5 + s_2s_3s_5]s_6 + \dots \dots \dots (A\cdot37)$$

$$(-s_2c_3s_4 + c_2c_4)c_6 \} - l_e (s_2c_3s_4 - c_2c_4) + l_s c_2 + l_b$$

を行う。そして ${}^b\mathbf{J}_{m1}(\boldsymbol{\theta})$ は、

$${}^b\mathbf{J}_{m1}(\boldsymbol{\theta}) = \begin{bmatrix} \partial x_m / \partial \theta_1 & \partial x_m / \partial \theta_2 & \partial x_m / \partial \theta_3 & \partial x_m / \partial \theta_4 & \partial x_m / \partial \theta_5 & \partial x_m / \partial \theta_6 & \partial x_m / \partial \theta_7 \\ \partial y_m / \partial \theta_1 & \partial y_m / \partial \theta_2 & \partial y_m / \partial \theta_3 & \partial y_m / \partial \theta_4 & \partial y_m / \partial \theta_5 & \partial y_m / \partial \theta_6 & \partial y_m / \partial \theta_7 \\ \partial z_m / \partial \theta_1 & \partial z_m / \partial \theta_2 & \partial z_m / \partial \theta_3 & \partial z_m / \partial \theta_4 & \partial z_m / \partial \theta_5 & \partial z_m / \partial \theta_6 & \partial z_m / \partial \theta_7 \end{bmatrix} \dots \dots \dots (A\cdot38)$$

となる。

次に、各モータの角速度を用いて行列 ${}^b\mathbf{J}_{m2}(\boldsymbol{\theta})$ を求める。ベース座標系 $\Sigma_b$ における手首座標系 $\Sigma_m$ の回転角速度 ${}^b\boldsymbol{\omega}_m$ を

$${}^b\boldsymbol{\omega}_m = \begin{bmatrix} {}^b\omega_{mx} \\ {}^b\omega_{my} \\ {}^b\omega_{mz} \end{bmatrix} \dots \dots \dots (A\cdot39)$$

で表す。ただし、 ${}^b\omega_{mx}$ 、 ${}^b\omega_{my}$ 、 ${}^b\omega_{mz}$ は ${}^b x$ 、 ${}^b y$ 、 ${}^b z$ 軸に対する角速度である。

一方、各モータの角速度を ${}^b\dot{\theta}_i$ とすれば、角速度ベクトル ${}^b\boldsymbol{\omega}_m$ は

$${}^b\boldsymbol{\omega}_m = {}^b\boldsymbol{\omega}_1 + {}^b\boldsymbol{\omega}_2 + {}^b\boldsymbol{\omega}_3 + {}^b\boldsymbol{\omega}_4 + {}^b\boldsymbol{\omega}_5 + {}^b\boldsymbol{\omega}_6 + {}^b\boldsymbol{\omega}_7 \dots \dots \dots (A\cdot40)$$

$$= {}^b\dot{\theta}_1 + {}^b\dot{\theta}_2 + {}^b\dot{\theta}_3 + {}^b\dot{\theta}_4 + {}^b\dot{\theta}_5 + {}^b\dot{\theta}_6 + {}^b\dot{\theta}_7$$

によって表すことができる。ただし、 ${}^b\dot{\theta}_i$ はベース座標系におけるモータ $i$ の角速度ベクトルである。

$${}^b\dot{\theta}_1 = {}^b\mathbf{R} \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \dots \dots \dots (A\cdot41)$$

$${}^b\dot{\theta}_2 = {}^b\mathbf{R} \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_2 \end{bmatrix} = {}^b\mathbf{R} \cdot {}^1\mathbf{R} \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_2 \end{bmatrix} \dots \dots \dots (A\cdot42)$$

$${}^b\dot{\theta}_3 = {}^bR_3 \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_3 \end{bmatrix} = {}^bR_1 \cdot {}^1R_2 \cdot {}^2R_3 R \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_3 \end{bmatrix} \dots\dots\dots (A \cdot 43)$$

$${}^b\dot{\theta}_4 = {}^bR_4 \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_4 \end{bmatrix} = {}^bR_1 \cdot {}^1R_2 \cdot {}^2R_3 \cdot {}^3R_4 R \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_4 \end{bmatrix} \dots\dots\dots (A \cdot 44)$$

$${}^b\dot{\theta}_5 = {}^bR_5 \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_5 \end{bmatrix} = {}^bR_1 \cdot {}^1R_2 \cdot {}^2R_3 \cdot {}^3R_4 \cdot {}^4R_5 R \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_5 \end{bmatrix} \dots\dots\dots (A \cdot 45)$$

$${}^b\dot{\theta}_6 = {}^bR_6 \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_6 \end{bmatrix} = {}^bR_1 \cdot {}^1R_2 \cdot {}^2R_3 \cdot {}^3R_4 \cdot {}^4R_5 \cdot {}^5R_6 R \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_6 \end{bmatrix} \dots\dots\dots (A \cdot 46)$$

$${}^b\dot{\theta}_7 = {}^bR_7 \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_7 \end{bmatrix} = {}^bR_1 \cdot {}^1R_2 \cdot {}^2R_3 \cdot {}^3R_4 \cdot {}^4R_5 \cdot {}^5R_6 \cdot {}^6R_7 R \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_7 \end{bmatrix} \dots\dots\dots (A \cdot 47)$$

角速度ベクトル ${}^b\omega_m$ は、図A・5に示すようにベース座標系に対する瞬間的な回転軸であることが式(A・40)～式(A・47)によって分かる。したがって、角速度ベクトル ${}^b\omega_m$ と各モータの角速度 ${}^b\dot{\theta}$ の関係を

$${}^b\omega_m = {}^bJ_{m2}(\theta) \cdot \dot{\theta} \dots\dots\dots (A \cdot 48)$$

で表すことができる。

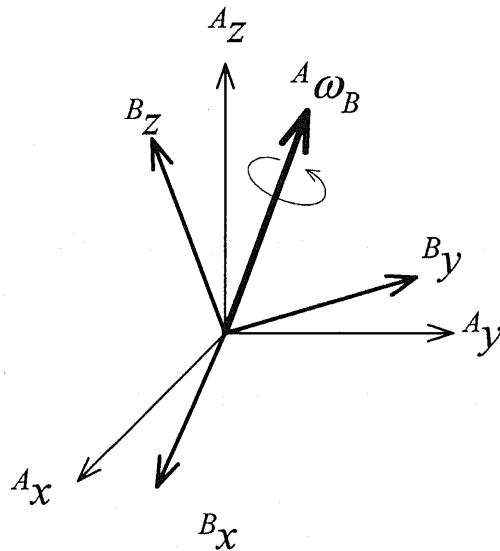


Fig.A・5 Rotating velocity of a coordinate

本論文では,  $x$ - $y$ - $z$  固定角法を用いて姿勢を表すため, 式(A・48)によって求めた角速度ベクトル ${}^b\boldsymbol{\omega}_m$ を  $x$ - $y$ - $z$  固定角法に合わせる必要がある.  $x$ - $y$ - $z$  固定角法における姿勢の回転速度 $\dot{\alpha}, \dot{\beta}, \dot{\gamma}$ と ${}^b\boldsymbol{\omega}_m$ の間に式

$$\begin{bmatrix} \dot{\alpha} \\ \dot{\beta} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} c_\gamma/c_\beta & s_\gamma/c_\beta & 0 \\ -s_\gamma & c_\gamma & 0 \\ s_\beta c_\gamma/c_\beta & s_\beta s_\gamma/c_\beta & 1 \end{bmatrix} \begin{bmatrix} {}^b\omega_{mx} \\ {}^b\omega_{my} \\ {}^b\omega_{mz} \end{bmatrix} = \boldsymbol{\Omega} \begin{bmatrix} {}^b\omega_{mx} \\ {}^b\omega_{my} \\ {}^b\omega_{mz} \end{bmatrix} = \boldsymbol{\Omega} \cdot {}^b\boldsymbol{J}_{m2}(\boldsymbol{\theta}_M) \cdot \dot{\boldsymbol{\theta}}_M \cdots \cdots \quad (\text{A} \cdot 49)$$

は成り立つ. したがって, 手先速度と関節角速度の関係からヤコビ行列を式(A・49)より導出することができる.

$${}^b\boldsymbol{J}_{m1}(\boldsymbol{\theta}) = \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} & J_{15} & J_{16} & J_{17} \\ J_{21} & J_{22} & J_{23} & J_{24} & J_{25} & J_{26} & J_{27} \\ J_{31} & J_{32} & J_{33} & J_{34} & J_{35} & J_{36} & J_{37} \end{bmatrix} \cdots \cdots \quad (\text{A} \cdot 50)$$

$${}^b\boldsymbol{J}_{m2}(\boldsymbol{\theta}) = \begin{bmatrix} J_{41} & J_{42} & J_{43} & J_{44} & J_{45} & J_{46} & J_{47} \\ J_{51} & J_{52} & J_{53} & J_{54} & J_{55} & J_{56} & J_{57} \\ J_{61} & J_{62} & J_{63} & J_{64} & J_{65} & J_{66} & J_{67} \end{bmatrix} \cdots \cdots \quad (\text{A} \cdot 51)$$

$$\begin{aligned} J_{11} = & l_w \{ [((-s_1c_2c_3 - c_1s_3)c_4 + s_1s_2s_4)c_5 - (-s_1c_2s_3 + c_1c_3)s_5]s_6 \\ & + [(-s_1c_2c_3 - c_1s_3)s_4 - s_1s_2c_4]c_6 \} \cdots \cdots \quad (\text{A} \cdot 52) \\ & + \{(-s_1c_2c_3 - c_1s_3)s_4 - s_1s_2c_4\}l_e - l_s s_1 s_2 \end{aligned}$$

$$\begin{aligned} J_{12} = & l_w [ \{(-c_1s_2c_3c_4 - c_1c_2s_4)c_5 + c_1s_2s_3s_5\}s_6 + (-c_1s_2c_3s_4 + c_1c_2c_4)c_6 ] \cdots \cdots \quad (\text{A} \cdot 53) \\ & + l_e (-c_1s_2c_3s_4 + c_1c_2c_4) + l_s c_1 c_2 \end{aligned}$$

$$\begin{aligned} J_{13} = & l_w [ \{(-c_1c_2s_3 - s_1c_3)c_4c_5 - (c_1c_2c_3 - s_1s_3)s_5\}s_6 + (-c_1c_2s_3 - s_1c_3)s_4c_6 ] \cdots \cdots \quad (\text{A} \cdot 54) \\ & + l_e (-c_1c_2s_3 - s_1c_3)s_4 \end{aligned}$$

$$\begin{aligned} J_{14} = & l_w [ \{(-c_1c_2c_3 - s_1s_3)s_4 - c_1s_2c_4\}c_5s_6 + \{(c_1c_2c_3 - s_1s_3)c_4 - c_1s_2s_4\}c_6 ] \cdots \cdots \quad (\text{A} \cdot 55) \\ & + l_e \{c_1c_2c_3 - s_1s_3\}c_4 - c_1s_2s_4 \end{aligned}$$

$$J_{15} = l_w [ -\{(c_1c_2c_3 - s_1s_3)c_4 - c_1s_2s_4\}s_5 - (c_1c_2s_3 + s_1c_3)c_5 ]s_6 \cdots \cdots \quad (\text{A} \cdot 56)$$

$$\begin{aligned} J_{16} = & l_w [ \{(c_1c_2c_3 - s_1s_3)c_4 - c_1s_2s_4\}c_5 - (c_1c_2s_3 + s_1c_3)s_5 ]c_6 \cdots \cdots \quad (\text{A} \cdot 57) \\ & - \{(c_1c_2c_3 - s_1s_3)s_4 + c_1s_2c_4\}s_6 \end{aligned}$$

$$J_{17} = 0 \cdots \cdots \quad (\text{A} \cdot 58)$$

$$J_{21} = l_w \{ [(c_1 c_2 c_3 - s_1 s_3) c_4 - c_1 s_2 s_4] c_5 - (c_1 c_2 s_3 + s_1 c_3) s_5 \} s_6 \\ + [(c_1 c_2 c_3 - s_1 s_3) s_4 - c_1 s_2 c_4] c_6 \} + l_e \{ (c_1 c_2 c_3 - s_1 s_3) s_4 + c_1 s_2 c_4 \} + l_s c_1 s_2 \quad \dots \quad (\text{A} \cdot 59)$$

$$J_{12} = l_w \{ [(-s_1 s_2 c_3 c_4 - s_1 c_2 s_4) c_5 + s_1 s_2 s_3 s_5 \} s_6 + (-s_1 s_2 c_3 s_4 + s_1 c_2 c_4) c_6 ] \\ + l_e (-s_1 s_2 c_3 s_4 + s_1 c_2 c_4) + l_s s_1 c_2 \quad \dots \quad (\text{A} \cdot 60)$$

$$J_{23} = l_w \{ [-(s_1 c_2 s_3 + c_1 c_3) c_4 c_5 - (s_1 c_2 c_3 + c_1 s_3) s_5 \} s_6 \\ + (-s_1 c_2 s_3 + c_1 c_3) s_4 c_6 ] + l_e (-s_1 c_2 s_3 + c_1 c_3) s_4 \quad \dots \quad (\text{A} \cdot 61)$$

$$J_{24} = l_w \{ [-(s_1 c_2 c_3 + c_1 s_3) s_4 - s_1 s_2 c_4 \} c_5 s_6 + \{ (s_1 c_2 c_3 + c_1 s_3) c_4 - s_1 s_2 s_4 \} c_6 ] \\ + l_e \{ s_1 c_2 c_3 + c_1 s_3 \} c_4 - s_1 s_2 s_4 \} \quad \dots \quad (\text{A} \cdot 62)$$

$$J_{25} = l_w [ -(s_1 c_2 c_3 + c_1 s_3) c_4 - s_1 s_2 s_4 \} s_5 - (s_1 c_2 s_3 - c_1 c_3) c_5 \} s_6 \quad \dots \quad (\text{A} \cdot 63)$$

$$J_{26} = l_w \{ [(s_1 c_2 c_3 + c_1 s_3) c_4 - s_1 s_2 s_4 \} c_5 - (s_1 c_2 s_3 + c_1 c_3) s_5 \} c_6 \\ - \{ (s_1 c_2 c_3 + c_1 s_3) s_4 + s_1 s_2 c_4 \} s_6 \} \quad \dots \quad (\text{A} \cdot 64)$$

$$J_{27} = 0 \quad \dots \quad (\text{A} \cdot 65)$$

$$J_{31} = 0 \quad \dots \quad (\text{A} \cdot 66)$$

$$J_{32} = l_w \{ [(-c_2 c_3 c_4 + s_2 s_4) c_5 + c_2 s_3 s_5 \} s_6 + (-c_2 c_3 s_4 - s_2 c_4) c_6 ] \\ - l_e (c_2 c_3 s_4 + s_2 c_4) l_e - l_s s_2 \quad \dots \quad (\text{A} \cdot 67)$$

$$J_{33} = l_w \{ (s_2 s_3 c_4 c_5 + s_2 c_3 s_5) s_6 + s_2 s_3 s_4 c_6 \} + l_w s_2 s_3 s_4 \quad \dots \quad (\text{A} \cdot 68)$$

$$J_{34} = l_w \{ (s_2 c_3 s_4 - c_2 c_4) c_5 s_6 + (-s_2 c_3 c_4 - c_2 s_4) c_6 \} - l_e (s_2 c_3 c_4 + c_2 s_4) \quad \dots \quad (\text{A} \cdot 69)$$

$$J_{35} = l_w \{ -(s_2 c_3 c_4 - c_2 s_4) s_5 + s_2 s_3 c_5 \} s_6 \quad \dots \quad (\text{A} \cdot 70)$$

$$J_{36} = l_w \{ [(-s_2 c_3 c_4 - c_2 s_4) c_5 + s_2 s_3 s_5 \} c_6 - (-s_2 c_3 s_4 + c_2 c_4) s_6 ] \quad \dots \quad (\text{A} \cdot 71)$$

$$J_{37} = 0 \quad \dots \quad (\text{A} \cdot 72)$$

$$J_{41} = 0 \quad \dots \quad (\text{A} \cdot 73)$$

$$J_{42} = -s_1 \quad \dots \quad (\text{A} \cdot 74)$$

$$J_{43} = c_1 s_2 \quad \dots \quad (\text{A} \cdot 75)$$

$$J_{44} = -c_1 c_2 s_3 - s_1 c_3 \quad \dots \quad (\text{A} \cdot 76)$$

$$J_{45} = (c_1 c_2 c_3 - s_1 s_3) s_4 + c_1 s_2 c_4 \quad \dots \quad (\text{A} \cdot 77)$$

$$J_{46} = -\{ (c_1 c_2 c_3 - s_1 s_3) c_4 - c_1 s_2 s_4 \} s_5 - (c_1 c_2 s_3 + s_1 c_3) c_5 \quad \dots \quad (\text{A} \cdot 78)$$

$$J_{47} = [ \{ (c_1 c_2 c_3 - s_1 s_3) c_4 - c_1 s_2 s_4 \} c_5 - (c_1 c_2 s_3 + s_1 c_3) s_5 \} s_6 \\ + \{ (c_1 c_2 c_3 - s_1 s_3) s_4 + c_1 s_2 c_4 \} c_6 \quad \dots \quad (\text{A} \cdot 79)$$

$$J_{51} = 0 \cdots \cdots \cdots (A \cdot 80)$$

$$J_{52} = c_1 \cdots \cdots \cdots (A \cdot 81)$$

$$J_{53} = s_1 s_2 \cdots \cdots \cdots (A \cdot 82)$$

$$J_{54} = -s_1 c_2 c_3 + c_1 c_3 \cdots \cdots \cdots (A \cdot 83)$$

$$J_{55} = (s_1 c_2 c_3 + c_1 s_3) s_4 + s_1 s_2 c_4 \cdots \cdots \cdots (A \cdot 84)$$

$$J_{56} = -\{(s_1 c_2 c_3 + c_1 s_3) c_4 - s_1 s_2 s_4\} s_5 - (s_1 c_2 s_3 - c_1 c_3) c_5 \cdots \cdots \cdots (A \cdot 85)$$

$$J_{57} = [\{(s_1 c_2 c_3 + c_1 s_3) c_4 - s_1 s_2 s_4\} c_5 - (s_1 c_2 s_3 - c_1 c_3) s_5] s_6 \\ + \{(s_1 c_2 c_3 + c_1 s_3) s_4 + s_1 s_2 c_4\} c_6 \cdots \cdots \cdots (A \cdot 86)$$

$$J_{61} = 1 \cdots \cdots \cdots (A \cdot 87)$$

$$J_{62} = 0 \cdots \cdots \cdots (A \cdot 88)$$

$$J_{63} = c_2 \cdots \cdots \cdots (A \cdot 89)$$

$$J_{64} = s_2 s_3 \cdots \cdots \cdots (A \cdot 90)$$

$$J_{65} = -s_2 c_3 s_4 + c_2 c_4 \cdots \cdots \cdots (A \cdot 91)$$

$$J_{66} = (s_2 c_3 c_4 + c_2 s_4) s_5 + s_2 s_3 c_5 \cdots \cdots \cdots (A \cdot 92)$$

$$J_{67} = \{(-s_2 c_3 c_4 - c_2 s_4) c_5 + s_2 s_3 s_5\} s_6 + (-s_2 c_3 s_4 + c_2 c_4) c_6 \cdots \cdots \cdots (A \cdot 93)$$

## A・5 力覚センサ座標系と運動解析

マニピュレータに取り付けられる F/T センサによって、マニピュレータの手先に作用する力とトルクを測定することができる。図 A・6 のように、本論文で用いる F/T センサは、 $\Sigma_f$  の各軸方向の力  $f_x, f_y, f_z$  と各軸まわりのトルク  $\tau_x, \tau_y, \tau_z$  を測定できる 6 軸力センサである。ここで、F/T センサ座標系  $\Sigma_f$  における力覚情報  ${}^f f$  は

$${}^f f = [f_x \quad f_y \quad f_z \quad \tau_x \quad \tau_y \quad \tau_z]^T \cdots \cdots \cdots (A \cdot 94)$$

のように定義される。

作業中にマニピュレータの手首部は回転するため、F/T センサによって検出した力覚情報は、F/T センサ座標系  $\Sigma_f$  からベース座標系  $\Sigma_b$  に変換する必要がある。図 A・6 に示すように、手首座標系  $\Sigma_m$  から F/T センサ座標系  $\Sigma_f$  への同次変換行列  ${}^m T_f$  は

$${}^m T_f = \left[ \begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_f \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \cdots \cdots \cdots (A \cdot 95)$$

で表される。また、式(A・17)を用いてベース座標系から F/T センサ座標系への同時変

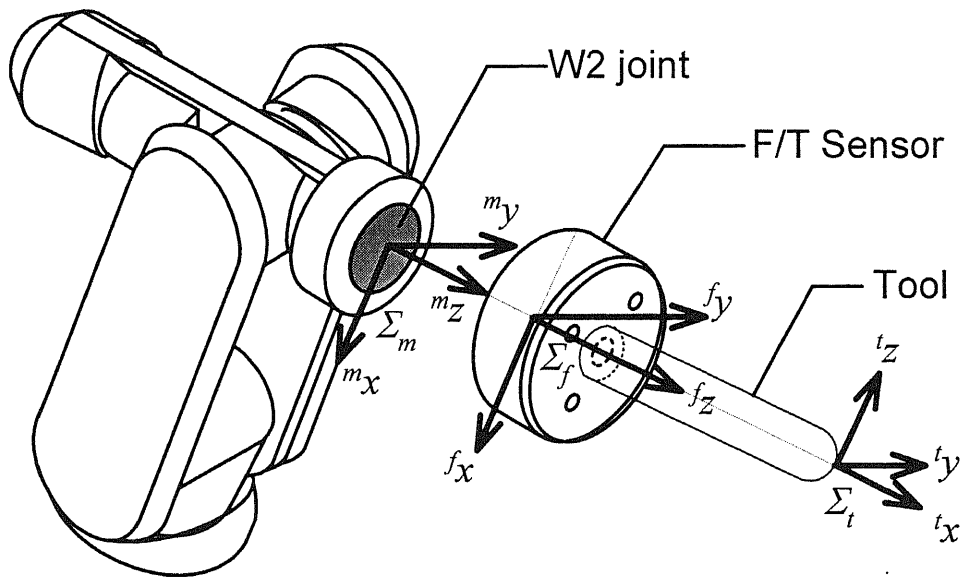


Fig.A・6 Definition of F/T sensor coordinate  $\Sigma_f$  and tool coordinate  $\Sigma_t$

換行列  ${}^bT_f$  を

$${}^bT_f = {}^bT_m \cdot {}^mT_f \dots\dots\dots (A\cdot96)$$

によって算出できる。よって、ベース座標系における 6 自由度の力覚情報  ${}^bf$  を

$${}^bf = {}^bT_f \cdot {}^ff \dots\dots\dots (A\cdot97)$$

から求める。

### A・6 ツール座標系と運動学

図 A・6 に示すように、ツールは F/T センサの前方に取り付けられるので、F/T センサ座標系  $\Sigma_f$  を介して同次変換行列を求める。ここで、ツール座標系  $\Sigma_t$  の位置・姿勢を表す順運動学関数を  $A_t$  とし、ベース座標系  $\Sigma_b$  におけるツール先端の位置ベクトル  ${}^bx_t$  を

$${}^bx_t = [x_t \ y_t \ z_t \ \alpha_t \ \beta_t \ \gamma_t]^T \dots\dots\dots (A\cdot98)$$

とする。

#### A・6・1 ツール座標系の順運動学

図 A・6 に示すようにツール座標系  $\Sigma_t$  は、F/T センサ座標系  $\Sigma_f$  に対して、 $f_z$  軸方向に  $l_t$  移動し、 $f_y$  軸まわりに  $-\pi/2$  rad 回転した座標系である。このため、F/T センサ座



標系  $\Sigma_f$  からツール座標系  $\Sigma_t$  への変換行列  ${}^fT_t$  は,

$${}^fT_t = \left[ \begin{array}{ccc|c} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & l_t \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \dots \dots \dots (A \cdot 99)$$

となる. したがって, ベース座標系  $\Sigma_b$  におけるツールの位置・姿勢の同時変換行列  ${}^bT_t$  を

$${}^bT_t = {}^bT_f \cdot {}^fT_t \dots \dots \dots (A \cdot 100)$$

によって算出できる.

### A・6・2 ツール座標系の速度解析

ツール座標系  $\Sigma_t$  の速度ベクトル  ${}^b\dot{x}_t$  は

$${}^b\dot{x}_t = [\dot{x}_t \quad \dot{y}_t \quad \dot{z}_t \quad \dot{\alpha}_t \quad \dot{\beta}_t \quad \dot{\gamma}_t]^T \dots \dots \dots (A \cdot 101)$$

のように定義する. 手首座標系  $\Sigma_m$  のヤコビ行列と同様に, ツール座標系  $\Sigma_t$  のヤコビ行列  ${}^bJ_t(\theta)$  を

$${}^b\dot{x}_t = {}^bJ_t(\theta) \cdot \dot{\theta} = \begin{bmatrix} {}^bJ_{t1}(\theta) \\ {}^bJ_{t2}(\theta) \end{bmatrix} \cdot \dot{\theta} \dots \dots \dots (A \cdot 102)$$

と表すことができる.  ${}^bJ_{m1}(\theta)$  と同様に,  ${}^b\dot{x}_t$  の位置成分を  $\theta_1$  から  $\theta_1$  で偏微分することによって  ${}^bJ_{t1}(\theta)$  を求める. また,  ${}^bJ_{m2}(\theta)$  と同様に, ツール座標系の姿勢回転角速度から  ${}^bJ_{t2}(\theta)$  を算出できる.

# 付記 B

## 最小二乗法によるカメラのキャリブレーション

### B・1 概説

付録 B では、1 台のカメラをロボットのベース座標系  $\Sigma_b$  に固定する場合について、最小二乗法によるカメラのキャリブレーションの手法を述べる。

### B・2 カメラモデル

本研究では用いるカメラのモデルをピンホールカメラモデルとする。ここで、CCD カメラの焦点距離を  $f$ 、 $x_c, y_c$  軸方向にそれぞれの解像度（1 ピクセルあたりの長さ）を  $r_u, r_v$  とする。一般に、カメラの焦点距離  $f$  や解像度  $r_u, r_v$  をカメラの内部パラメータと呼ぶ。図 B・1 に示すように、カメラ座標系  $\Sigma_c$  における点  ${}^c p = [{}^c X, {}^c Y, {}^c Z]^T$  と画像平面での投影点  ${}^l p = [u, v]^T$  との関係

$${}^c X = {}^c Z \cdot u \frac{r_u}{f}, \quad {}^c Y = {}^c Z \cdot v \frac{r_v}{f} \dots\dots\dots (B \cdot 1)$$

で表す。ここで、

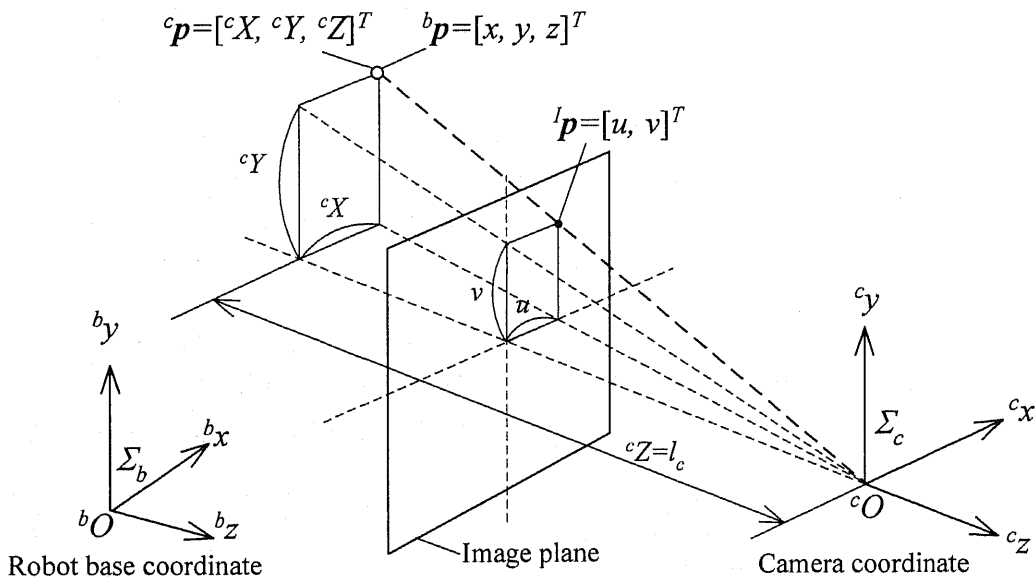


Fig. B・1 Camera model

$$K_u = \frac{r_u}{f}, \quad K_v = \frac{r_v}{f} \dots\dots\dots (B\cdot2)$$

とすれば, 式(B・1)は

$${}^cX = K_u \cdot {}^cZ \cdot u, \quad {}^cY = K_v \cdot {}^cZ \cdot v \dots\dots\dots (B\cdot3)$$

で表される. 対象物体は  $x_c - y_c$  平面にだけ移動する場合,  ${}^cZ$  は一定である. ここで,  ${}^cZ$  を  $l_c$  とする. したがって, 点  ${}^l p$  と点  ${}^c p$  の関係を

$$\begin{bmatrix} {}^cX \\ {}^cY \\ {}^cZ \end{bmatrix} = \begin{bmatrix} l_c K_u & 0 & 0 \\ 0 & l_c K_v & 0 \\ 0 & 0 & l_c \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \dots\dots\dots (B\cdot4)$$

で表すことができる.

### B・3 カメラ座標系とロボットベース座標系

カメラ座標系  $\Sigma_c$  とロボットベース座標系  $\Sigma_b$  との位置関係は図 B・1 に示されている. 座標系  $\Sigma_b$  における座標系  $\Sigma_c$  の原点の座標を  $p_0 = [x_0, y_0, z_0]^T$  とする. ロボットベース座標系  $\Sigma_b$  における 3次元空間での点  $p = [x, y, z]^T$  とカメラ座標系  $\Sigma_c$  における点  ${}^c p$  との関係は

$$p = p_0 + R_c \cdot {}^c p \dots\dots\dots (B\cdot5)$$

で表される. ただし,  $R_c$  は回転行列である. 一般に, 回転行列  $R_c$  とベクトル  $p_0$  をカメラの外部パラメータと呼ぶ. ここで, 座標系  $\Sigma_b$  の  ${}^b z$  軸,  ${}^b y$  軸,  ${}^b x$  軸まわりのそれぞれ回転を  $\gamma, \beta, \alpha$  とすれば, 回転行列  $R_c$  を

$$\begin{aligned} R_c &= \begin{bmatrix} c_\gamma & -s_\gamma & 0 \\ s_\gamma & c_\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\beta & 0 & s_\beta \\ 0 & 1 & 0 \\ -s_\beta & 0 & c_\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha \\ 0 & s_\alpha & c_\alpha \end{bmatrix} \\ &= \begin{bmatrix} c_\beta c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma & c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma - c_\alpha c_\gamma & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma \\ -s_\beta & s_\alpha c_\beta & c_\alpha c_\beta \end{bmatrix} \dots\dots\dots (B\cdot6) \end{aligned}$$

で求める. ただし,

$$c_\gamma = \cos(\gamma), \quad s_\gamma = \sin(\gamma), \quad c_\beta = \cos(\beta), \quad s_\beta = \sin(\beta), \quad c_\alpha = \cos(\alpha), \quad s_\alpha = \sin(\alpha),$$

である. 式(B・4), (B・6)を式(B・5)に代入すれば,

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} c_\beta c_\gamma & s_\alpha s_\beta c_\gamma - c_\alpha s_\gamma & c_\alpha s_\beta c_\gamma - s_\alpha s_\gamma \\ s_\alpha c_\beta & s_\alpha s_\beta s_\gamma - c_\alpha c_\gamma & c_\alpha s_\beta s_\gamma - s_\alpha c_\gamma \\ -s_\beta & s_\alpha c_\beta & c_\alpha c_\beta \end{bmatrix} \begin{bmatrix} {}^cX \\ {}^cY \\ {}^cZ \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}$$

$$\begin{aligned}
&= \begin{bmatrix} c_{\beta}c_{\gamma} & s_{\alpha}s_{\beta}c_{\gamma} - c_{\alpha}s_{\gamma} & c_{\alpha}s_{\beta}c_{\gamma} - s_{\alpha}s_{\gamma} \\ s_{\alpha}c_{\beta} & s_{\alpha}s_{\beta}s_{\gamma} - c_{\alpha}c_{\gamma} & c_{\alpha}s_{\beta}s_{\gamma} - s_{\alpha}c_{\gamma} \\ -s_{\beta} & s_{\alpha}c_{\beta} & c_{\alpha}c_{\beta} \end{bmatrix} \begin{bmatrix} l_c K_u & 0 & 0 \\ 0 & l_c K_v & 0 \\ 0 & 0 & l_c \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \\
&= \begin{bmatrix} l_c K_u c_{\beta} c_{\gamma} & l_c K_v (s_{\alpha} s_{\beta} c_{\gamma} - c_{\alpha} s_{\gamma}) & l_c (c_{\alpha} s_{\beta} c_{\gamma} - s_{\alpha} s_{\gamma}) \\ l_c K_u s_{\alpha} c_{\beta} & l_c K_v (s_{\alpha} s_{\beta} s_{\gamma} - c_{\alpha} c_{\gamma}) & l_c (c_{\alpha} s_{\beta} s_{\gamma} - s_{\alpha} c_{\gamma}) \\ -l_c K_u s_{\beta} & l_c K_v s_{\alpha} c_{\beta} & l_c c_{\alpha} c_{\beta} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \cdots \cdots \quad (\text{B}\cdot 7)
\end{aligned}$$

により座標系  $\Sigma_b$  における 3次元空間での点  $\mathbf{p}=[x, y, z]^T$  の座標を算出できる。

座標系  $\Sigma_b$  の  $b_x$  軸,  $b_y$  軸方向のみにロボットの位置制御を行う場合, 式(B・7)を

$$\begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{R}_t \begin{bmatrix} u \\ v \end{bmatrix} + \mathbf{b}_t \cdots \cdots \quad (\text{B}\cdot 8)$$

で表すことができる。ただし,

$$\mathbf{R}_t = \begin{bmatrix} l_c K_u c_{\beta} c_{\gamma} & l_c K_v (s_{\alpha} s_{\beta} c_{\gamma} - c_{\alpha} s_{\gamma}) \\ l_c K_u s_{\alpha} c_{\beta} & l_c K_v (s_{\alpha} s_{\beta} s_{\gamma} - c_{\alpha} c_{\gamma}) \end{bmatrix}, \quad \mathbf{b}_t = \begin{bmatrix} x_0 + l_c (c_{\alpha} s_{\beta} c_{\gamma} - s_{\alpha} s_{\gamma}) \\ y_0 + l_c (c_{\alpha} s_{\beta} s_{\gamma} - s_{\alpha} c_{\gamma}) \end{bmatrix}$$

である。また, 式(B・8)から行列  $\mathbf{R}_t$  の各要素はカメラの内部パラメータと外部パラメータを含むことが分かる。

座標系  $\Sigma_b$  ではカメラを一度取り付けたらその場所に固定するため, 行列  $\mathbf{R}_t$  とベクトル  $\mathbf{b}_t$  の各要素は定数と見なすことができる。したがって, 式(B・8)は

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix} = \mathbf{R}_t \begin{bmatrix} u \\ v \end{bmatrix} + \mathbf{b}_t \cdots \cdots \quad (\text{B}\cdot 9)$$

で表される。さらに, 画像処理で得られた結果  $[u, v]^T$  を  $[X, Y]^T$  に置き換えれば, 式(B・9)は

$$\begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{R}_t \begin{bmatrix} X \\ Y \end{bmatrix} + \mathbf{b}_t \cdots \cdots \quad (\text{B}\cdot 10)$$

で表される。式(B・10)は, 画像処理で得られた特徴点の位置  $[X, Y]^T$  を座標系  $\Sigma_b$  に変換する方法を示す。画像処理の結果をロボットの制御に用いるには, 行列  $\mathbf{R}_t$  とベクトル  $\mathbf{b}_t$  の各要素を何らかの方法によって同定する必要がある。

#### B・4 最小二乗法によるカメラのキャリブレーション

式(B・10)に示す行列  $\mathbf{R}_t$  とベクトル  $\mathbf{b}_t$  の各要素を同定するため, 最小二乗法を用いる。1回めの画像処理の結果に対し, 式(B・10)によって得られた  $\mathbf{c}_p$  と  $\mathbf{p}$  の間の関係を

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} + \begin{bmatrix} a_{13} \\ a_{23} \end{bmatrix} \cdots \cdots \quad (\text{B}\cdot 11)$$

で表すことはできる。式(B・11)を展開すると,

$$\begin{aligned} x_1 &= a_{11}X_1 + a_{12}Y_1 + a_{13} \dots\dots\dots (B\cdot 12) \\ y_1 &= a_{21}X_1 + a_{22}Y_1 + a_{23} \end{aligned}$$

のようになる。画像処理を  $n$  回に行った場合、

$$\begin{cases} x_1 = a_{11}X_1 + a_{12}Y_1 + a_{13} \\ y_1 = a_{21}X_1 + a_{22}Y_1 + a_{23} \\ \dots\dots\dots \\ x_n = a_{11}X_n + a_{12}Y_n + a_{13} \\ y_n = a_{21}X_n + a_{22}Y_n + a_{23} \end{cases} \dots\dots\dots (B\cdot 13)$$

のように、式(B・12)を  $n$  回に書くことができる。式(B・13)をまとめると、

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} X_1 & Y_1 & 1 \\ X_2 & Y_2 & 1 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & 1 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \end{bmatrix} \dots\dots\dots (B\cdot 14)$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} X_1 & Y_1 & 1 \\ X_2 & Y_2 & 1 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & 1 \end{bmatrix} \begin{bmatrix} a_{21} \\ a_{22} \\ a_{23} \end{bmatrix} \dots\dots\dots (B\cdot 15)$$

のようになる。

画像処理を  $n$  回に行ったことによって座標系  $\Sigma_c$  における  $n$  組みの特徴点の座標  $\{X_i, Y_i\}$  ( $i=1,2,\dots,n$ ) が得られ、式(B・14)、(B・15)によって座標系  $\Sigma_b$  における特徴点の座標  $\{x_i, y_i\}$  ( $i=1,2,\dots,n$ ) を算出できる。ここで、実際に測定した特徴点の空間位置を  $\{x_i^T, y_i^T\}$  ( $i=1,2,\dots,n$ ) とし、それと回帰結果  $\{x_i, y_i\}$  ( $i=1,2,\dots,n$ ) との誤差  $E=[E_x, E_y]^T$  は

$$E_x = \sum_{i=1}^n (x_i - x_i^T)^2, \quad E_y = \sum_{i=1}^n (y_i - y_i^T)^2 \dots\dots\dots (B\cdot 16)$$

のように定義される。

また、式(B・13)を式(B・16)に代入すると誤差  $E_x, E_y$  のそれぞれは

$$E_x = \sum_{i=1}^n (a_{11}X_i + a_{12}Y_i + a_{13} - x_i^T)^2 \dots\dots\dots (B\cdot 17)$$

$$E_y = \sum_{i=1}^n (a_{21}X_i + a_{22}Y_i + a_{23} - y_i^T)^2 \dots\dots\dots (B\cdot 18)$$

のようになる。 $E_x, E_y$  を最小にするため、 $E_x$  が  $a_{11}, a_{12}, a_{13}$  のそれぞれに対する微分を 0 に、 $E_y$  が  $a_{21}, a_{22}, a_{23}$  のそれぞれに対する微分を 0 にすればよい。したがって、式(B・17)、

(B・18)から

$$\begin{cases} \frac{\partial E_x}{\partial a_{11}} = \sum_{i=1}^n (a_{11}X_i + a_{12}Y_i + a_{13} - x_i^T)X_i = 0 \\ \frac{\partial E_x}{\partial a_{12}} = \sum_{i=1}^n (a_{11}X_i + a_{12}Y_i + a_{13} - x_i^T)Y_i = 0 \dots\dots\dots (B\cdot 19) \\ \frac{\partial E_x}{\partial a_{13}} = \sum_{i=1}^n (a_{11}X_i + a_{12}Y_i + a_{13} - x_i^T) = 0 \end{cases}$$

$$\begin{cases} \frac{\partial E_y}{\partial a_{21}} = \sum_{i=1}^n (a_{21}X_i + a_{22}Y_i + a_{23} - y_i^T)X_i = 0 \\ \frac{\partial E_y}{\partial a_{22}} = \sum_{i=1}^n (a_{21}X_i + a_{22}Y_i + a_{23} - y_i^T)Y_i = 0 \dots\dots\dots (B\cdot 20) \\ \frac{\partial E_y}{\partial a_{23}} = \sum_{i=1}^n (a_{21}X_i + a_{22}Y_i + a_{23} - y_i^T) = 0 \end{cases}$$

が得られる。また、式(B・19), (B・20)を展開すると、

$$\begin{cases} \left( \sum_{i=1}^n X_i X_i \right) a_{11} + \left( \sum_{i=1}^n X_i Y_i \right) a_{12} + \left( \sum_{i=1}^n X_i \right) a_{13} = \left( \sum_{i=1}^n X_i x_i^T \right) \\ \left( \sum_{i=1}^n Y_i X_i \right) a_{11} + \left( \sum_{i=1}^n Y_i Y_i \right) a_{12} + \left( \sum_{i=1}^n Y_i \right) a_{13} = \left( \sum_{i=1}^n Y_i x_i^T \right) \dots\dots\dots (B\cdot 21) \\ \left( \sum_{i=1}^n X_i \right) a_{11} + \left( \sum_{i=1}^n Y_i \right) a_{12} + n a_{13} = \left( \sum_{i=1}^n x_i^T \right) \end{cases}$$

$$\begin{cases} \left( \sum_{i=1}^n X_i X_i \right) a_{21} + \left( \sum_{i=1}^n X_i Y_i \right) a_{22} + \left( \sum_{i=1}^n X_i \right) a_{23} = \left( \sum_{i=1}^n X_i y_i^T \right) \\ \left( \sum_{i=1}^n Y_i X_i \right) a_{21} + \left( \sum_{i=1}^n Y_i Y_i \right) a_{22} + \left( \sum_{i=1}^n Y_i \right) a_{23} = \left( \sum_{i=1}^n Y_i y_i^T \right) \dots\dots\dots (B\cdot 22) \\ \left( \sum_{i=1}^n X_i \right) a_{21} + \left( \sum_{i=1}^n Y_i \right) a_{22} + n a_{23} = \left( \sum_{i=1}^n y_i^T \right) \end{cases}$$

のようになる。式(B・21)に  $a_{11}, a_{12}, a_{13}$  のそれぞれの係数と式(B・22)に  $a_{21}, a_{22}, a_{23}$  のそれぞれの係数をまとめると、式(B・21), 式(B・22)は

$$\begin{cases} t_{11}a_{11} + t_{12}a_{12} + t_{13}a_{13} = t_1^x \\ t_{21}a_{11} + t_{22}a_{12} + t_{23}a_{13} = t_2^x \dots\dots\dots (B\cdot 23) \\ t_{31}a_{11} + t_{32}a_{12} + t_{33}a_{13} = t_3^x \end{cases}$$

$$\begin{cases} t_{11}a_{21} + t_{12}a_{22} + t_{13}a_{23} = t_1^y \\ t_{21}a_{21} + t_{22}a_{22} + t_{23}a_{23} = t_2^y \dots\dots\dots (B\cdot 24) \\ t_{31}a_{21} + t_{32}a_{22} + t_{33}a_{23} = t_3^y \end{cases}$$

のように表される。ただし、

$$\begin{aligned}
 t_{11} &= \sum_{i=1}^n X_i X_i & t_{12} &= \sum_{i=1}^n X_i Y_i & t_{13} &= \sum_{i=1}^n X_i \\
 t_{21} &= \sum_{i=1}^n Y_i X_i & t_{22} &= \sum_{i=1}^n Y_i Y_i & t_{23} &= \sum_{i=1}^n Y_i \\
 t_{31} &= \sum_{i=1}^n X_i & t_{32} &= \sum_{i=1}^n Y_i & t_{33} &= n \\
 t_1^x &= \sum_{i=1}^n (X_i x_i^T) & t_2^x &= \sum_{i=1}^n (Y_i x_i^T) & t_3^x &= \sum_{i=1}^n x_i^T \\
 t_1^y &= \sum_{i=1}^n (X_i y_i^T) & t_2^y &= \sum_{i=1}^n (Y_i y_i^T) & t_3^y &= \sum_{i=1}^n y_i^T
 \end{aligned}$$

である。したがって、式(B・23)に示す方程式連立し、パラメータ  $a_{11}, a_{12}, a_{13}$  を

$$\begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix}^{-1} \begin{bmatrix} t_1^x \\ t_2^x \\ t_3^x \end{bmatrix} \dots\dots\dots (B \cdot 25)$$

によって算出できる。また、式(B・24)に示す方程を連立し、パラメータ  $a_{21}, a_{22}, a_{23}$  を

$$\begin{bmatrix} a_{21} \\ a_{22} \\ a_{23} \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix}^{-1} \begin{bmatrix} t_1^y \\ t_2^y \\ t_3^y \end{bmatrix} \dots\dots\dots (B \cdot 26)$$

によって算出できる。