



ロボットマニピュレータの自律的動作計画に関する研究

比留川, 博久

(Degree)

博士 (学術)

(Date of Degree)

1987-03-31

(Date of Publication)

2009-03-04

(Resource Type)

doctoral thesis

(Report Number)

甲0673

(URL)

<https://hdl.handle.net/20.500.14094/D1000673>

※ 当コンテンツは神戸大学の学術成果です。無断複製・不正使用等を禁じます。著作権法で認められている範囲内で、適切にご利用ください。



博士論文

ロボットマニピュレータの自律的動作計画に関する研究

昭和 62 年 1 月

神戸大学大学院自然科学研究科

比留川 博久

概 要

ロボットマニピュレータは、フレキシブルなマニユファクチャリングツールとして大きな期待を集めているが、現在の産業用ロボットのほとんどは自律的に動作を計画する能力を持たないため、必要な動作は全て人間が教示しなければならない。この教示には非常に時間がかかり、大きな問題となっている。この問題に対処するためには、ロボットをよりインテリジェントにしなければならない。本研究は、この自律的な動作の計画の実現を目的として行った。尚、本研究の背景と目的については、第1章で詳しく論じた。

動作を自律的に計画するためには、まずそのためのアルゴリズムが必要である。マニピュレータの動作の中で作業対象との相互作用がない自由動作の場合、唯一の制約は障害物であるので、障害物回避アルゴリズムがあれば自由動作は自律的に計画可能である。そこで、本研究ではまずこの問題について検討を行った。これまでに提案されている障害物回避アルゴリズムは、解を発見的に探索するポテンシャル法と、マニピュレータと障害物が干渉しない空間をマニピュレータの関節空間において陽に求めてから解を探索する自由空間法に大きく分けられるが、何れも一長一短がある。ここでは作業空間の大局的な情報に基づいて解を発見的に探索する方法について検討した。まず、障害物回避のためのポテンシャル場を、各障害物と障害物の中間面上で0となるように定義した。次に、このような面上でエンドイフェクタの目標軌道を計画した。そして、これらに

基づいてマニピュレータの障害物回避のためのポテンシャル関数を定義し、それに最急降下法を適用することによって動作を計画した。本手法の有効性についてはシミュレーションスタディにより確認した。本手法の特徴をまとめると、次の様になる。まず、一般のポテンシャル法と比較すると、

- ・ 目標位置に到達するためのポテンシャルと障害物を回避するためのポテンシャルとの相互干渉が出来るだけ小さくなるように、大局的な見地からポテンシャル場を定義しているので、停留や障害物との衝突による解の探索の失敗が起りにくい。
- ・ エンドイフェクタの目標軌道を大局的に計画しているので、これに基づく解の探索の再試行が可能である。

また、自由空間法と比較すると、

- ・ 必要な計算量がマニピュレータの自由度数に対して線形にしか増加しない。
- ・ 対象とするマニピュレータの機構学的な特徴に依存せずに適用可能である。

この2つの特徴により、本手法は冗長自由度を持つような一般の多自由度マニピュレータに適用可能である。以上の方法については、第2章で述べた。

次に、このような自律動作計画に必要な演算処理が膨大になってしまうという問題点に対処するために、動作計画の結果を学習することにより、同じ動作あるいは似た動作を再び計画することなしに実行することを可能にし、これによって必要な演算処理を少なくし

ようという試みを行った。ここでは、知識ベースシステムの一つである適応プロダクションシステムをフレームワークとして用いて、そのための基礎的な検討をした。得られた結果を以下にまとめる。

- ・ 適応プロダクションシステムの特徴とその動作学習過程記述に対する適合性について論じた。
- ・ 適応プロダクションシステムに基づいて獲得知識を表現・利用する方法について示し、その際、弁別ネットが効率の良い知識の想起に対して有効に働くことを述べた。
- ・ 獲得知識の一般化と特化について述べ、これらによって獲得知識の有効範囲が学習されることを示した。
- ・ 動作学習過程全体について考え、ここで考えたモデリングにおいては、学習前の動作は演算処理に基づく閉ループ制御であったのが学習後は経験によって獲得した知識の想起に基づく閉ループ制御になることを述べた。
- ・ 適応プロダクションシステムをインプリメントし、学習過程のシミュレーションを行った。

以上の結果は、第3章で論じた。

最後に、実際にマニピュレータの動作を自律的に計画するために必要な全ての基本的な要素を備えたプロトタイプシステムを開発した。マニピュレータの作業を知的に計画するためには、まず、作業対象や障害物などから成る作業環境の情報を取得しなければならない。ここで問題となるのが、この作業環境に関する情報を如何にし

て計算機に入力するかである。現時点ではこれを計算機に全く自動的に認識させるのは極めて困難であり、逆に逐一人間が記述するのはあまりにも煩雑である。本研究では、グレイコード状の光パターンを用いた距離計測法と対話を組み合わせて、対象物体の3次元形状とその位置・姿勢の情報を得るシステムを試作した。この作業環境のモデルに基づいて、マニピュレータの動作を計画するのであるが、この際、計画された動作が所期の目的を達成しているかどうかを検証する必要がある。この検証を行うことなしに実機を動作させてしまうと、動作目的を達しないことがあるだけでなく、場合によっては作業環境やマニピュレータを破損する可能性もある。既に、幾つかのオフラインシミュレーションシステムが開発されているが、センサシステムと結合されているものは少ない。本システムでは、作業環境教示システムによって得た実環境のモデルに基づいて、マニピュレータ動作のグラフィックアニメーション、数値計算によるマニピュレータと障害物の間の干渉チェックを行えるオフラインシミュレータを開発した。以上の方法により計画された動作は、開発した7自由度多関節型のマニピュレータにより実行する。前述したオフラインシミュレータ中のマニピュレータのカイネマティクスは、この実機に合わせてある。ここでは、制御アルゴリズムとして各関節独立のPID制御を用いた。また、自由動作の自律的計画を対象として、実際に作業環境教示・動作計画・シミュレーション・実機による実行という全過程の運用をすることにより、本シ

システムの有効性を確認した。以上については、第4章で詳しく述べた。

本研究で得られた結果は、結論として第5章にまとめて示した。

目 次

	<i>page</i>
第1章 緒 論	
1.1 本研究の背景	1
1.2 本研究の目的	3
第2章 安全第一アルゴリズムとポテンシャル関数に基づく マニピュレータの障害物回避法	
2.1 はじめに	6
2.2 障害物回避のためのポテンシャル	8
2.2.1 障害物の境界評価関数と安全第一面	9
2.2.2 ポテンシャル関数	10
2.3 目標軌道の計画	17
2.3.1 安全第一グラフ	17
2.3.2 安全第一グラフ上での目標軌道の決定	20
2.4 マニピュレータの障害物回避動作の計画	21
2.4.1 マニピュレータの障害物回避動作の計画の ためのポテンシャル関数	21
2.4.2 最急降下法の適用	24
2.5 シミュレーション結果	26
2.5.1 システム構成	26
2.5.2 シミュレーション結果	27
2.6 まとめ	32

第3章	適応プロダクションシステムに基づく マニピュレータ動作学習過程のモデリング	
3.1	はじめに	34
3.2	適応プロダクションシステムと動作学習	35
3.2.1	適応プロダクションシステムとその特徴	35
3.2.2	PSの動作学習過程記述に対する適合性	37
3.3	獲得知識の表現と利用	38
3.3.1	プロダクションの形式と弁別ネット	38
3.3.2	獲得知識の一般化と特化	44
3.3.3	知識の構造の変化として捉えた動作学習 過程	50
3.4	インプリメンテーション	51
3.4.1	システム構成	51
3.4.2	プロダクションの例	51
3.5	まとめ	53
第4章	マニピュレータの自律的動作計画システム	
4.1	はじめに	54
4.2	システム構成	54
4.2.1	システムの構成要素	54

	<i>page</i>
4.2.2 システムの全体構成	56
4.3 作業環境教示システム	59
4.3.1 空間コード化による距離計測	59
4.3.2 対話入力とジオメトリックモデル	70
4.4 オフラインシミュレータ	77
4.4.1 グラフィックアニメーション	77
4.4.2 数値計算による干渉チェック	80
4.5 制御システム	81
4.5.1 軌道計画	82
4.5.2 制御アルゴリズム	85
4.6 システムの運用	90
4.7 まとめ	96
第5章 結 論	97
謝 辞	100
参考文献	101
著者文献	109

	<i>page</i>
付録 1. ポテンシャル関数を用いた運動分解の安定性	111
2. 対象としたマニピュレータのAマトリクスと 各関節の可動範囲	113

第1章 緒 論

1.1 本研究の背景

消費の多様化に伴い、工業生産は少品種大量生産の時代から多品種少量生産の時代へと移行しつつある。この様な状況の中で、フレキシブルなマニファクチャリングツールとして大きな期待を集めているのが、ロボットマニピュレータである。しかしながら、現在の産業用ロボットのほとんどは自律的に動作を計画する能力を持たないため、必要な動作は全て人間が教示しなければならない。この教示には非常に時間がかかり、マンパワーを要する点と教示の期間中生産ラインを休止しなければならない点の2点から大きな問題となっている。この問題に対処するためには、ロボットをよりインテリジェントにしなければならない。では、ロボットが備えるべき知的機能とは何であろうか。以下、マニピュレータによる機械部品の組み立て作業を対象を絞って議論を進める。

究極的には、目標組み上がり状態のみを与えれば、全ての作業を自律的に実行出来るマニピュレータの実現が望ましいと言える。このためには、まず第一に、組み立てシーケンスのプランニングが必要である。この種の問題は、従来より人工知能の分野でグラフ探索理論の適用対象の一つとして取り扱われてきた[Fahlman 1974,

Fikes 1971, Sacerdoti 1974]。しかし、これらの研究においては、各部品³の3次元形状は考慮されておらず、これを捨象した結果得られるトポロジーのみが議論の対象となってきた。最近、形状を考慮した研究[Sekiguchi 1985]も報告されてはいるが、未だ試みの段階であり、この問題は今後の精力的な研究に委ねられていると言える。

組み立てシーケンスが何らかの方法で与えられると仮定すると、次に必要となるのが、これを実行するマニピュレータ動作の計画である。言いかえると、これは作業対象の状態の変化により記述された作業をマニピュレータ自身の動作に変換するということである。この問題は、タスクレベルロボット言語[Lieberman 1977, Lozano-Perez 1977, Popplestone 1980]の開発という名のもとに精力的な研究が行われている。

この動作計画を実現するためにまず必要なのは、作業対象や障害物などの作業環境を視覚センサや触覚センサからの情報に基づいて認識することである。しかし、3次元の一般的なシーンを対象とした場合これは容易ではなく、現在実用的なレベルで認識可能な対象はかなり限定されている[Shirai 1982]と思われる。

この認識が可能であると仮定すると、次に必要になるのは得られた作業環境の情報に基づいて動作を自律的に計画するアルゴリズムである。マニピュレータの動作は、作業対象と物理的接触がない自由動作(free motion)と、これがある場合に要求されるコンプライアント動作(compliant motion)に大別される。自由動作に対する唯一の制約は障害物であるので、障害物回避アルゴリズムによって、これ

は自律的に計画出来る。しかしながら、一般的な機構を持つマニピュレータに適用可能な障害物回避アルゴリズムは、有効なものが未だ開発されていない。また、物体の把持動作やペグの挿入動作などのコンプライアント動作の自律的計画についても幾つかの研究が報告されている[Lozano-Perez 1981, Laugier 1981, Wolter 1985]が、現在のところ試みの段階であると思われる。

さらに、動作の自律計画法が確立したとしても、これらのアルゴリズムは、一般的に言って膨大な演算処理を必要とする。確かに近年のコンピュータの発達にはめざましいものがあるが、計算機資源の有効利用という観点から、これは大きな問題であると言える。

1.2 本研究の目的

本研究は自由動作の自律動作計画を例として、以上挙げた諸問題の解決を目的として行った。本研究は、次の3つの部分から成る。

- (1) 新しい障害物回避アルゴリズムの提案
- (2) 動作の帰納的学習に対する基礎的検討
- (3) 自律的動作計画システムの開発

これまでに提案されている障害物回避アルゴリズムは、解を発見的に探索するポテンシャル法と、マニピュレータと障害物が干渉しない空間をマニピュレータの関節空間において陽に求めてから解を

探索する自由空間法に大きく分けられるが、何れも一長一短がある。本研究では、ポテンシャル法の問題点を解決するという形で、障害物回避アルゴリズムの開発を行った。これについては、第2章で詳しく述べる。

また、(2)は自律的動作計画に必要な演算処理が膨大になってしまうという問題点に対処するために行ったものである。すなわち、動作計画の結果を学習することにより、同じ動作あるいは似た動作を再び計画することなしに実行することを可能にし、これによって必要な演算処理を少なくすることを目的としている。ここでは、知識ベースシステムの一つである適応プログラミングシステムをフレームワークとして用いて、そのための基礎的な検討を行った。これについては、第3章で詳しく述べる。

(3)の自律的動作計画システムは、知的なマニピュレータシステムの実現を目指して、そのプロトタイプシステムとして開発したものである。これは、作業環境教示システム・オフラインシミュレータ・7自由度多関節型マニピュレータとその制御システムから成る。作業環境教示システムは、CCDカメラ・液晶シャッター付スライドプロジェクタ・マイクロコンピュータというハードウェア構成のシステムで、三角測量の原理に基づく距離計測と対話を組み合わせて、作業環境に関する情報を取得し、その内部モデルを構築するシステムである。オフラインシミュレータは、作業環境教示システムにより得られた実環境のモデルに基づいて、動作の計画・検証を行うシステムで、グラフィックアニメーション・数値計算による干渉

チェックの機能を持つ。言うまでもなく、7自由度多関節型マニピュレータは、計画された動作を最終的に実行するためのものであり、オフラインシミュレータ中のマニピュレータは、これをシミュレートするように作成されている。以上のシステムについては第4章で詳細を述べるが、その際、このシステムを用いて実環境における実機の自由動作を計画出来る事を示す。

尚、本研究で得られた結果のまとめを第5章に結論として示す。

第2章 安全第一アルゴリズムとポテンシャル関数に基づくマニピュレータの障害物回避法

2.1 はじめに

マニピュレータにさせる作業を、マニピュレータ自身の動作ではなく作業対象の状態変化シーケンスによって記述することが出来れば、作業の計画に要する労力は非常に軽減されると考えられる。このためには、個々の要素動作を自律的に生成出来るアルゴリズムが必要である。マニピュレータの動作の中で作業対象との相互作用がない自由動作の場合、唯一の制約は障害物であるので、障害物回避アルゴリズムがあれば自由動作は自律的に生成可能である。以上の観点から、本章ではマニピュレータの障害物回避アルゴリズムについて検討した結果について述べる。

従来のマニピュレータの障害物回避法は、作業空間の局所的情報に基づいて発見的に解を探索する方法[Loeff 1975, Khatib 1978, Okutomi 1983, Kokaji 1985]と、マニピュレータの関節空間において障害物と干渉しない自由空間を陽に求めてから、その空間内で経路を探索する方法[Udupa 1977, Lozano-Perez 1979, 1981, Brooks 1983, Hasegawa 1986]に大別される。前者の方法は、ポテンシャル法と呼ばれているが、その手法が対象とするマニピュレータの機構

に依存しないことと、必要な計算量が自由度数に対して線形にしか増加しないという長所を持つ。しかしながら、この方法は作業空間の局所的情報のみに基づいて解を探索しているため、停留などの問題が生じたときに大局的な見地から探索をやり直すことは出来ない。従って、この方法では解が存在するときにそれを発見出来るのは、かなり限られた場合のみである。一方、自由空間法と呼ばれる後者の方法は、移動物体の一般化座標空間における自由空間において、大局的な情報に基づいて解を探索する方法で、この自由空間内に解が存在する場合は必ずそれを発見することが保証されている。この方法の問題点は、何の工夫もしないと必要な計算量がマニピュレータの自由度数の指数関数のオーダーで増加していくことである。このため、自由空間を求める際には対象となるマニピュレータの機構学的な特徴に着目して適当な近似や自由空間の低次元化を行い、計算量の低減が計られている。現在報告されている例では、探索の対象としている自由空間の次元は、3次元以下である。もし一般の多自由度のマニピュレータの関節空間において完全な自由空間を求めることが出来れば、障害物回避問題は解決するが、計算量の点から見て実際には不可能である。先程、自由空間法によると解が存在するときに必ずこれを発見すると述べたが、これは完全な自由空間を近似し低次元化した部分集合内で保証されているのであって、実際には解の見落としをする場合もある。また、可到達な作業座標の範囲を広げるために、冗長な自由度を利用すること [Hanabusa 1983] も考えられるが、この場合には探索の対象となる自由空間の

次元をより高くしなければならないので、自由空間法の適用は困難である。

以上の考察に基づいて、ここでは作業空間の大局的な情報に基づいて解を発見的に探索する方法について検討した。まず、障害物回避のためのポテンシャル場を、各障害物と障害物の中間面上で0となるように定義する。次に、このような面上でエンドイフェクタの目標軌道を計画する。このようにすると、目標に到達するためのポテンシャルによる効果と障害物を回避するためのポテンシャルによる効果との相互干渉が非常に小さくなり停留などの問題はおこりにくくなる。また、この目標軌道を用いて、大局的な見地から解の探索をやり直すことも可能になる[Hirukawa 1986a,1986d]。

目標軌道を求めるには、まずその候補となるグラフを求め、そのグラフ上で道を探査するという方法を採用するが、このグラフを求めるための基本的な考え方は三森ら[Mimori 1982]によって示された。しかし、彼らの考え方はアルゴリズムの形で明確に与えられていない。また、ここでは開始点・目標点とグラフの接続の仕方やグラフ上での道の決定法において異なる方法を採用している。

以下、障害物回避のためのポテンシャルの定義、目標軌道の計画法、これらに基づく障害物回避法、シミュレーション結果を述べる。

2.2 障害物回避のためのポテンシャル

2.2.1 障害物の境界評価関数と安全第一面

各障害物は、凸多面体からなると仮定する。このとき、それぞれの障害物 O_i は3次元空間内の点の集合として、

$$O_i = \cap_{j=1}^{M(i)} \{X | e_{ij}(X) \leq 0\} \quad (2-1)$$

と表される。ここに、 $M(i)$ は i 番目の障害物の境界面の総数、 $X^T=[x,y,z]$ は点の座標値、 $e_{ij}(X)$ は i 番目の障害物の j 番目の境界面の評価関数である。この関数は、障害物の内部で負、境界面上で0、外部で正の値を取るように定義する。さらに、その絶対値が点 X と境界面との距離を表すように決める。これは、

$$e_{ij}(X) = a_{ij}x + b_{ij}y + c_{ij}z + d_{ij} \quad (2-2)$$

と定義される。ここに、 a_{ij} 、 b_{ij} 、 c_{ij} 、 d_{ij} の符号は $e_{ij}(X)$ の値が障害物の内部で負になるように決め、かつ、 $a_{ij}^2 + b_{ij}^2 + c_{ij}^2 = 1$ となるように絶対値を決める。作業空間の外枠はマニピュレータの可動域により決まるが、ここでは簡単のためにこれを直方体であると仮定する。そして、この直方体の各面をただ一つの境界面からなる障害物として扱う。このとき、境界面の評価関数の符号は、作業空間内で正になるように決める。

次に、作業空間内の点 X と i 番目の障害物との一種の距離を表す関数を、

$$e_i(X) = \max_{j=1}^{M(i)} e_{ij}(X) \quad (2-3)$$

と定義する[Furukawa 1979]。このとき、点 X とその点に最も近い障害物との一種の距離を表す関数は、

$$e(X) = \min_{i=1}^N e_i(X) \quad (2-4)$$

で与えられる。ここに、Nは障害物の総数である。2つの最も近い障害物からの距離が等しい点は、周りの障害物から局所的には最も離れている点であると言える。従って、この点は障害物との衝突から最も安全な点であると考えられる。このような点の集合は、

$$SFP = \{X | e(X) = e_i(X) = e_j(X) > 0, \forall i \neq j\} \quad (2-5)$$

で与えられる。作業空間に対する仮定から、SFPは平面上の有界領域の集合となり、これらは障害物との衝突から最も安全な面となる。障害物回避経路を計画する際の指標としては、最短経路を求めるなど、幾つかの評価基準が考えられるが、ここでは「安全」を最優先して経路を計画する。そのために基本となるのがこれらの面であるので、我々はこれらの面のそれぞれを「安全第一面」(Safety First Plane)と呼ぶことにする。2次元空間では、安全第一面に相当するものは線分になる。この例をFig.2-1に、3次元空間での例をFig.2-2に示す。

2.2.2 ポテンシャル関数

以上の準備のもとに、作業空間内の各点Xに対して、障害物との接近度を評価するポテンシャル関数を次のような2次式によって定義する。

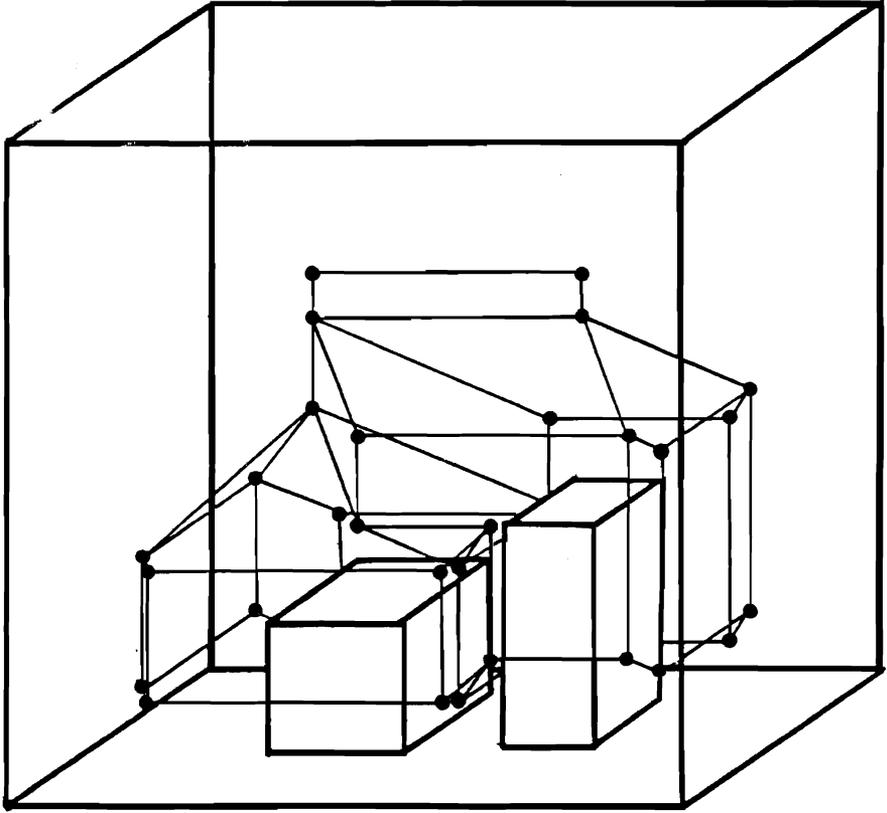


Fig.2-2 An Example of the Safety First Graph
in 3-dimensional Space

$$V_o(X) = \frac{1}{2} (e_{max}(X) - e(X))^2$$

(2-6)

ここで、 $e_{max}(X)$ は、点Xを通りその点に最も近い障害物の境界面に垂直な直線と、安全第一面の交点における、 $e(X)$ の値を示す関数で

ある (Fig.2-3 参照)。直感的に言うと、これは、他の障害物に近づく

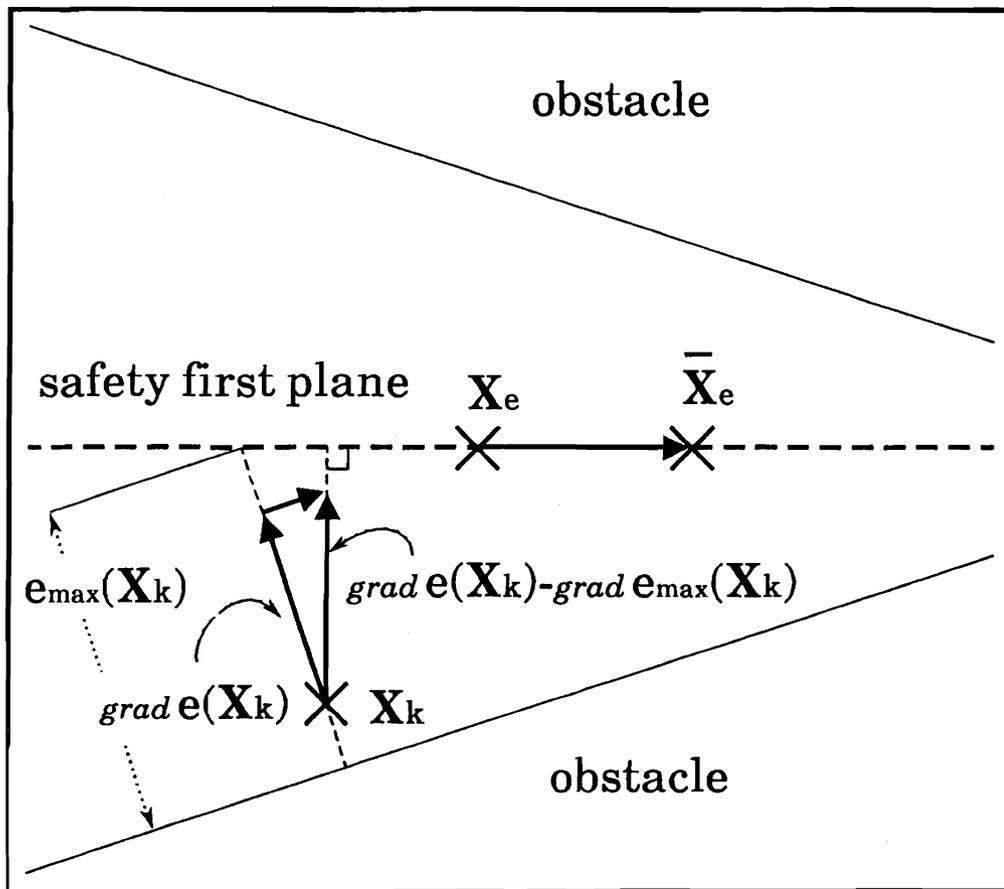


Fig.2-3 An Illustration of the Function $e_{max}(X)$ and $grad e(X_k) - grad e_{max}(X_k)$

ことなしに、最も近い障害物から離れることの出来る最大の距離である。以下、任意の点 X_0 に対して $e_{max}(X_0)$ を求める方法を示す。 p 番目の障害物の q 番目の境界面が点 X_0 に最も近いとすると、

$$e(X_0) = e_{pq}(X_0) \quad (2-7)$$

となる。この関数の勾配は、

$$\nabla e(X) \Big|_{X=X_0} = \left(\frac{\partial e(X)}{\partial x}, \frac{\partial e(X)}{\partial y}, \frac{\partial e(X)}{\partial z} \right) \Big|_{X=X_0} = (a_{pq}, b_{pq}, c_{pq}) \quad (2-8)$$

となり、これは、最も近い障害物の境界面の法線ベクトルに相当し、大きさは1で、障害物の外へ向いている。ただし、点 $X=X_0$ において $e_{ij}(X)$ の値が相異なる (i, j) の組に対して等しいときは、適当な関数を選んでその勾配をとるものとする。 $A^T_{pq} = [a_{pq}, b_{pq}, c_{pq}]$ とおくと、点 X_0 を通り方向ベクトルが A_{pq} の直線は、

$$X = A_{pq} u + X_0 \quad (2-9)$$

となる。ここに、 u は媒介変数である。点 X_0 からこの直線に沿って p 番目の障害物から離れて行くと、安全第一面との交点において、他の障害物からの距離が p 番目の障害物からの距離に等しくなる。これが q 番目の障害物の s 番目の境界面であったとすると、この点においては、 $A^T_{rs} = [a_{rs}, b_{rs}, c_{rs}]$ として、

$$A^T_{pq} X + d_{pq} = A^T_{rs} X + d_{rs} = e(X) > 0 \quad (2-10)$$

という関係が成立する。(2-5)式の定義から分かるように、これは安全第一面を示す方程式になる。(2-9)式の直線と(2-10)式の安全第一面の交点は(2-9)式を(2-10)式に代入して得られる、

$$(A_{pq} - A_{rs})^T A_{pq} u = (A_{rs} - A_{pq})^T X_0 + (d_{rs} - d_{pq}) \quad (2-11)$$

という式から、媒介変数 u の値を決めることにより求められる。以

上の方法をまとめると、次に示す様になる。これを安全第一アルゴリズム(Safety First Algorithm)と呼ぶことにする。

安全第一アルゴリズム:

Fをp番目を除く障害物に属する全ての境界面から成る集合とする。

ステップ1. Fから一つの要素を任意に選び、その面の評価関数を $e_{rs}(X)$ とする。

ステップ2

$$(A_{pq} - A_{rs})^T A_{pq} \neq 0 \quad (2-12)$$

が成立するときは(2-11)式の解を求め、それを(2-9)式に代入して交点 X_{max} を求める。 $e_{rs}(X_{max})$ の値も求めておく。成立しないときは、ステップ6へ行く。

ステップ3

$$e_{rs}(X_{max}) > 0$$

ならステップ4へ行く。

$$e_{rs}(X_{max}) \leq 0$$

ならステップ6へ行く。

ステップ4.

$$e_{rs}(X_{max}) = \max_{j=1}^{M(r)} e_{rj}(X_{max}) \quad (2-13)$$

が成立するときは、ステップ5へ行く。

成立しないときは、ステップ6へ行く。

ステップ5.

$$e(X_{max}) > 0 \quad (2-14)$$

$$e_{rs}(X_{max}) = e(X_{max}) \quad (2-15)$$

がいずれも成立するときは、

$$e_{max}(X_0) = e_{rs}(X_{max}) \quad (2-16)$$

として、アルゴリズムを終了する。

どちらか一方でも成立しないときは、ステップ6へ行く。

ステップ6. その評価関数が $e_{rs}(X)$ である面を集合Fから削除する。

ステップ1へ行く。

ここで、ステップ2の(2-12)式は(2-11)式に唯一解が存在するための必要十分条件である。ステップ3の条件は冗長な条件であるが、この簡単な必要条件をチェックすることによってフィルタリングを行い、余計な候補を省くことが出来る。ステップ4の条件は、 $X=X_{max}$ において $e_{rs}(X)$ がr番目の障害物からの距離を示す関数に選ばれるかどうかをチェックする。ステップ5の(2-14)式により点 X_{max} が作業空間内の点であるかどうかを調べ、(2-15)式により

$e_{rs}(X)$ が $X=X_{\max}$ において最も近い障害物からの距離を表しているかどうかをチェックする。これらの条件を全て満たしたとき、点 X_{\max} は求める点であることが分かり、従って $e_{\max}(X_0)$ の値も(2-16)式のように求まる。このような点は必ず存在し、かつ、唯一である。以上によって、(2-6)式のポテンシャル関数の定義は完了した。この関数の値は、障害物から離れる程小さくなり、安全第一面では常に0となる。

2.3 目標軌道の計画

2.3.1 安全第一グラフ

本節では、安全第一面上でエンドイフェクタの目標軌道を計画する方法を示す。安全第一面間の交線をアークとし、これらの交線の交点をノードとするグラフを考えると、これはいうまでもなく安全第一面上にのっている。ここでは、これを目標軌道の候補として用いる。このグラフを『安全第一グラフ』(Safety First Graph)と呼ぶことにする。これは、2.2節で示した安全第一アルゴリズムを少し修正することによって求められる。基本的な考え方としては、まずノードを求め、これらのノードを接続することによってアークを求めるという方法を採用。これを以下に示す。

F を障害物に属する全ての境界面から成る集合とする。

ステップ1. Fから4つの要素より成る組を1つ選んできて、それぞれの面の評価関数を $e^k(X)$ ($k=1,2,3,4$)とする。これらは、次のようにかける。

$$e^k(X) = a^k x + b^k y + c^k z + d^k \quad (2-17)$$

ステップ2. 次の式の左辺の 3×3 行列の行列式が、0かどうかをチェックする。

$$\begin{bmatrix} a^1 - a^2 & b^1 - b^2 & c^1 - c^2 \\ a^1 - a^3 & b^1 - b^3 & c^1 - c^3 \\ a^1 - a^4 & b^1 - b^4 & c^1 - c^4 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} d^2 - d^1 \\ d^3 - d^1 \\ d^4 - d^1 \end{bmatrix} \quad (2-18)$$

0でないときは、(2-18)式の解 X_n を求め、この解を(2-17)式のどれかに代入して評価関数の値を求めておく。この値を $e^0(X_n)$ とおく。

0のときは、ステップ7へ行く。

ステップ3. $e^0(X_n) > 0$ ならステップ4へ行く。

$e^0(X_n) \leq 0$ ならステップ7へ行く。

ステップ4. 次の式が4つの面が属する各々の障害物について成り立つかどうかをチェックする。

$$e^0(X_n) = \max_{j=1}^{M(i)} e_{ij}(X_n) \quad (2-19)$$

4つの障害物全てについて成立するときは、ステップ5へ行く。それ以外の場合は、ステップ7へ行く。

ステップ5.

$$e(X_n) > 0 \quad (2-20)$$

$$e^0(X_n) = e(X_n) \quad (2-21)$$

がいずれも成立するときは、ステップ6へ行く。

それ以外の場合はステップ7へ行く。

ステップ6. X_n を安全第1グラフのノードの集合SFNに加える。そして、ステップ1で選んだ4つの面の組をそのノードに対応させて記憶しておく。ステップ7へ行く。

ステップ7. Fから選び得る4つの要素より成る組を全て調べたらステップ8へ行く。

それ以外の場合はステップ1へ行く。

ステップ8. それぞれのノードに対応して記憶されている4つの面のうち何れか3つが共通であるノードのペアを互いに継続ノードとして接続する。

アルゴリズムを終了する。

以上の考え方は、簡単な修正により2次元空間の場合にも適用可能である。2次元空間の場合の安全第1グラフはFig.2-1に示したグラフと一致する。3次元空間の場合は、Fig.2-2中の安全第1面の交線が安全第1グラフになる。

次の問題は、エンドイフェクタの初期位置あるいは目標位置とこのグラフを如何にして接続するかである。これには、2.2節の安全第1アルゴリズムをそのまま利用出来る。これを以下に示す。まずエンドイフェクタの初期位置をこのアルゴリズムにおける点 X_0 に

選んだときの点 X_{\max} をこの点の回避点と考える。そして、初期位置をスタートノードとし、点 X_{\max} をその継続ノードとする。次に、そのノードにおいて、 $e(X)=e_{pq}(X)=e_{rs}(X)$ という関係が成立するものを点 X_{\max} の継続ノードとする。以上の処理により、エンドイフェクタの初期位置をグラフと接続することが出来る。目標位置についても、点 X_{\max} をその接近点と考えることにより同様に処理することが出来る。ただし、ノード間の継続関係の向きは初期位置の場合と逆にしなければならない。また、目標位置の接近点が初期位置の回避点の継続ノードとなる場合があることに注意しなければならない。

2.3.2 安全第一グラフ上での目標軌道の決定

次に、安全第一グラフ上で、発見的グラフ探索法であるA*アルゴリズム[Hart 1968]を用いて、初期位置から目標位置までの経路を求める。ノード n を通る道の中で最小コストの道のコストを $f(n)$ とすると、これは、

$$f(n)=g(n)+h(n) \quad (2-22)$$

という形に書ける。ただし、 $g(n)$ はスタートノードからノード n までの最小コストの道のコスト、 $h(n)$ はノード n からゴールノードまでの最小コストの道のコストである。ここで各アークのコストには、そのアークの両端のノード間のユークリッド距離を用いる。 $g(n)$ の推定値 $\hat{g}(n)$ にはA*によって現在までに見つけられた道の中で最小コストを持つ道のコストを用い、 $h(n)$ の推定値 $\hat{h}(n)$ にはノード

nとゴールノードのユークリッド距離を選ぶ。このとき、明らかに

$$\hat{g}(n) \geq g(n) \quad (2-23)$$

$$\hat{h}(n) \leq h(n) \quad (2-24)$$

という関係が成立する。(2-24)式の条件下において、スタートノードからゴールノードまでの経路が存在するときには、A*は必ずコスト最小の道を見つけて終了することが理論的に保証されている [Hart 1968]。以上の方法により得られた初期位置から目標位置までの経路の例をFig.2-4に示す。この様に、本手法ではエンドイフェクタの目標軌道を大局的に計画しているので、従来手法では不可能だった解の再探索を行うことが出来る。

2.4 マニピュレータの障害物回避動作の計画

2.4.1 マニピュレータの障害物回避動作計画のためのポテンシャル関数

次に、マニピュレータの障害物回避動作を計画する方法を示す。ただし、ここでは各関節角の目標軌道を得る事を目的とし、マニピュレータの動力学については考慮しない。まず、マニピュレータの運動学方程式を次のように書く。

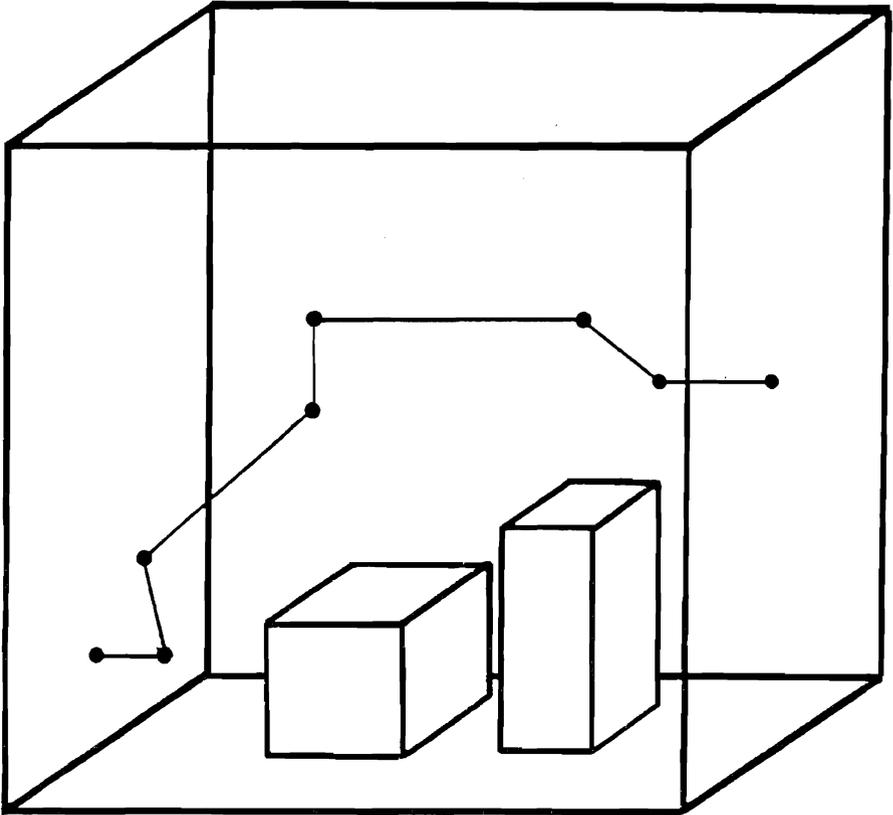


Fig.2-4 An Example of the Path on the Safety First Graph
in 3-dimensional Space

$$Y_e^T(\Theta) = [f_1(\Theta), \dots, f_m(\Theta)] \quad (2-25)$$

ここに、 Y_e はマニピュレータのエンドイフェクタの位置と姿勢を表す $m \times 1$ (m は作業座標の数)のベクトル、 Θ はその各関節変数を要素

とする $N_f \times 1$ (N_f はマニピュレータの自由度) のベクトル、 $f_j (j=1, \dots, m)$ は、関数名である。目標とするエンドイフェクタの位置と姿勢を \bar{Y}_e とすると、現在の位置と姿勢 Y_e との偏差を評価するポテンシャル関数は、2次形式を用いて次のように定義できる。

$$V(\Theta) = \frac{1}{2} (Y_e(\Theta) - \bar{Y}_e)^T W_e (Y_e(\Theta) - \bar{Y}_e) \quad (2-26)$$

ここに、 W_e は $m \times m$ の正定対称な重み行列である。次に、マニピュレータの腕全体と障害物の接近度を評価するポテンシャル関数を定義する。この評価を正確に行うためには (2-6) 式のポテンシャル関数をマニピュレータの全体積について積分しなければならない。しかし、(2-6) 式のポテンシャル関数は安全第一面を境界面として関数が切り替わるので、この積分を実行するには非常に多くの計算を必要とする。そこで、ここでは一様なポテンシャル場の場合と同じ様にポテンシャル関数の各リンク毎の積分値を、そのリンクの幾何学的中心点でのポテンシャル関数の値とそのリンクの体積との積で近似することにする。これは、次のように書ける。

$$V_0(\Theta) = \sum_{k=1}^L w_{ok} c_{ok} \left[\frac{1}{2} (e_{\max}(X_k(\Theta)) - e(X_k(\Theta)))^2 \right] \quad (2-27)$$

ここに、 $V_0(\Theta)$ はマニピュレータの腕全体に対する (2-6) 式のポテンシャル関数の近似積分値を示す Θ の関数で、 w_{ok} は k 番目のリンクに対する正の重みづけ定数、 c_{ok} は k 番目のリンクの体積、また $X^T_k(\Theta) = [g_{k1}(\Theta), g_{k2}(\Theta), g_{k3}(\Theta)]$ は k 番目のリンクの幾何学的中心点を示す Θ の関数である。

これら2つのポテンシャル関数 V_e と V_0 の和 V

$$V(\Theta) = V_e(\Theta) + V_o(\Theta) \quad (2-28)$$

をマニピュレータの障害物回避動作計画のために用いる。ポテンシャル関数 $V(\Theta)$ は、目標との偏差と障害物との接近度の和を意味し、これがより小さくなるように動作を計画する。

2.4.2 最急降下法の適用

(2-28)式のポテンシャル関数に最急降下法を適用すると、次のような Θ の漸化式が得られる。

$$\begin{aligned} & \Theta^{(r+1)} - \Theta^{(r)} \\ &= -h \left[J_e^T W_e (Y_e - \bar{Y}_e) + \sum_{k=1}^L w_{ok} c_{ok} J_k^T (\nabla e_{max}(X_k) - \nabla e(X_k)) (e_{max}(X_k) - e(X_k)) \right] \end{aligned} \quad (2-29)$$

ここに、 r はイテレーション数、 h は適当な正の定数、 J_e はその第 i 列が、

$$\frac{\partial Y_e^T}{\partial \theta_i} = \left[\frac{\partial f_1}{\partial \theta_i}, \frac{\partial f_2}{\partial \theta_i}, \dots, \frac{\partial f_m}{\partial \theta_i} \right] \quad (2-30)$$

である $m \times N_f$ のヤコビアン行列(θ_i は i 番目の関節)であり、 J_k はその第 i 列が、

$$\frac{\partial X_k^T}{\partial \theta_i} = \left[\frac{\partial g_{k1}}{\partial \theta_i}, \frac{\partial g_{k2}}{\partial \theta_i}, \frac{\partial g_{k3}}{\partial \theta_i} \right] \quad (2-31)$$

である $3 \times N_f$ のヤコビアン行列である。ただし、(2-29)式の \bar{Y}_e の要素のうちエンドイフェクタの目標位置については、2.3節で示した

方法によって求めた目標軌道上に適当な間隔で取っていき、 V_e の値がある一定値以下に収束したときに次の目標位置に切り替えるという方法を採用。従って Y_e 中の位置を表す部分を X_e とすると、ベクトル $(X_e - \bar{X}_e)$ は安全第一面と平行になるので、(2-29)式の第1項はエンドイフェクタを安全第一面と平行に現在位置から各中間目標位置に向かって動かすという効果を持ち、またベクトル $(\nabla e_{\max}(X_k) - \nabla e(X_k))$ は安全第一面と垂直になるので、第2項はマニピュレータ全体を安全第一面に近づけるという効果を持つ。これを図解すると、Fig.2-3の様になる。これらの効果は、幾何学的に考えて互いに相反しない。それゆえ、これらの項間の相互干渉は非常に小さくなると考えられる。これが本手法によると、停留の問題が起これにくくなる理由であり、従来の手法と比較した本手法の特徴の一つである。また e_{\max} は X の関数であるから Θ の関数でもあるが、最急降下法を適用するに際して点 X の回避点 X_{\max} を固定した目標点と考えて、これを定数とみなすと、

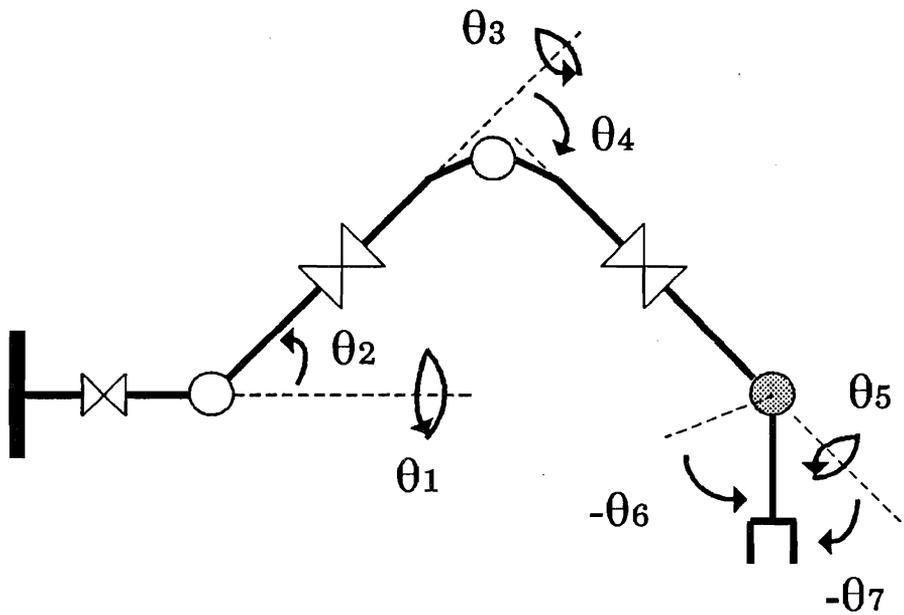
$$\Theta^{(r+1)} - \Theta^{(r)} = -h \left[J_e^T W_e (Y_e - \bar{Y}_e) - \sum_{k=1}^L w_{ok} c_{ok} J_k^T \nabla e(X_k) (e_{\max}(X_k) - e(X_k)) \right] \quad (2-32)$$

となる。ここで、 $\nabla e(X)$ は点 X に最も近い障害物に垂直なベクトルになるので、障害物との干渉による解の探索の失敗をなるべく避けるという観点からは(2-32)式を用いる方が良い(Fig.2-3参照)。以上の方法の有効性について、シミュレーションにより検討した結果を次節に示す。

2.5 シミュレーション結果

2.5.1 システム構成

対象としたマニピュレータは、7自由度多関節型で、関節は全て回転関節とした。この自由度配置をFig.2-5に示す。これを障害物と



⊗ : twist joint

○ : rotational joint

● : rotational joint with 2 d.o.f.

Fig.2-5 The configuration of the objective manipulator with 7 d.o.f.

ともにグラフィックシミュレータを用いて表示し、生成された動作を目視によって確認出来るようにした。また、マニピュレータと障害物が干渉したかどうかを正確に判断するために数値計算による干渉チェックルーチン[Maruyama 1972]も用意した。シミュレーションは、全てマイクロコンピュータ(i8086+i8087,8MHz)上で行った。記述言語としては、Cを用いた。

2.5.2 シミュレーション結果

ここでは、(2-32)式に基づいてシミュレーションを行った結果を示す。これは、停留がほとんど起こらないことがシミュレーションにより分かったからである。(2-32)式中のパラメータは以下のようにして決めた。まず、 h は

$$J_e J_e^T > \frac{h}{2} J_e J_e^T J_e J_e^T \quad (2-33)$$

となるように選ぶと、障害物がない場合は目標位置へ安定に収束することが理論的に示せるが(付録1参照)、ここでは解軌道が振動的にならないという判断基準のもとに、試行的に $h = 1.0 \times 10^2$ と定めた。作業座標としてはエンドイフェクタの位置と姿勢を表現するものとして、

$$Y_e^T = [o_x, o_y, o_z, a_x, a_y, a_z, p_x, p_y, p_z] \quad (2-34)$$

を用いた。ここに、各要素は順に手先の方向ベクトル、接近ベクトル、位置ベクトルである。これに対応して、 $W_e = \text{diag}(w_{e1}, \dots, w_{e9})$ と選び、

$$W_{ei} = \alpha c_0 w_e \quad (i=1, \dots, 6) \quad (2-35)$$

$$W_{ei} = \beta c_0 w_e \quad (i=7, \dots, 9) \quad (2-36)$$

とする。ここに、 α 、 β は姿勢と位置の相対的な収束の速さを決める係数で、作業の性質によって決める。ここでは、 $\alpha=1.25 \times 10^{-2}$ または $\alpha=2.5 \times 10^{-2}$ 、 $\beta=1.0$ とした。 c_0 はマニピュレータの総体積で、

$$c_0 = \sum_{k=1}^L c_{ok} \quad (2-37)$$

である。以上のパラメータは、対象とするマニピュレータによって決まり、障害物とは無関係である。 w_e は、

$$w_e = 1 - \sum_{k=1}^L w_{ok} \quad (2-38)$$

と決める。この w_e 、 w_{ok} は、目標に到達するためのポテンシャルと障害物を回避するためのポテンシャルの間の重みを決めるパラメータで、解を探索する際にはこれを試行的に調節する。

以上の方法に基づくシミュレーション例をここで示す。まず、Fig.2-6のような作業空間を設定した。この図中の外枠は、マニピュレータの作業空間を考慮して決めた。マニピュレータのベースリンクは、点 $(x,y,z)=(0.6,0,0.6)$ [m]に固定されているとした。この2つの障害物を越していく動作の例を Fig.2-7に示す。これは、 $\Theta^T = [-1.89 \ -0.788 \ 2.64 \ 1.15 \ 0.41 \ -1.85 \ 1.05]$ [rad]なる初期状態から、エンドイフェクタの作業座標 $Y_e^T = [0 \ 1 \ 0 \ 0 \ -1 \ 1 \ 0.4 \ 0.15]$ となる状態

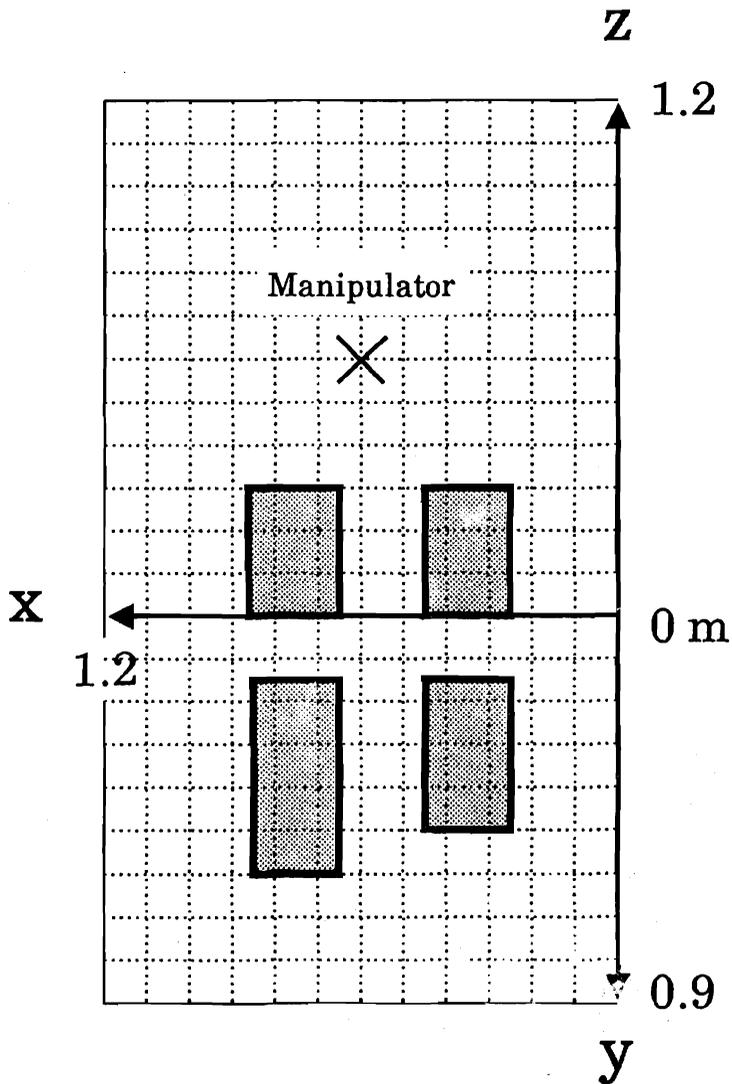


Fig.2-6 The working space of the manipulator

への動作である。エンドイフェクタの中間目標位置は、目標軌道上で0.1[m]間隔で切りかえている。ポテンシャルに対する重みは、 $w_{01}=0.125$ 、 $w_{02}=0.2$ 、 $w_{03}=0.25$ とし、目標位置が最終目標位置に

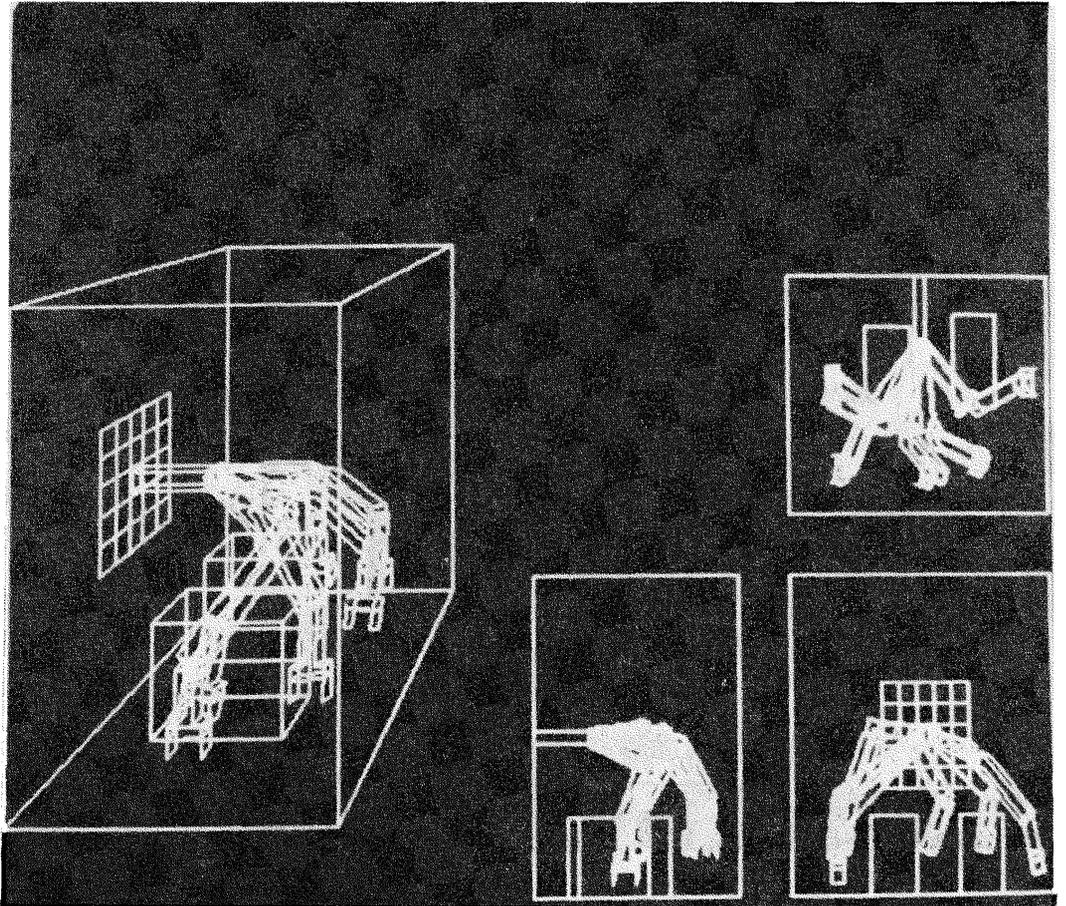


Fig.2-7 An Example of the Collision Avoidance Motion

切りかわった時点で全て0としている。この例では、3413回のイテレーションで目標状態に到達している。次に、右側の障害物を0.1[m]高くして、2つの障害物の間に手先を挿入する動作の例を

8に示す。初期状態は、 $\Theta^T = [-0.420 \ 0.107 \ -0.174 \ 1.07 \ 0.271 \ -1.44 \ 1.$

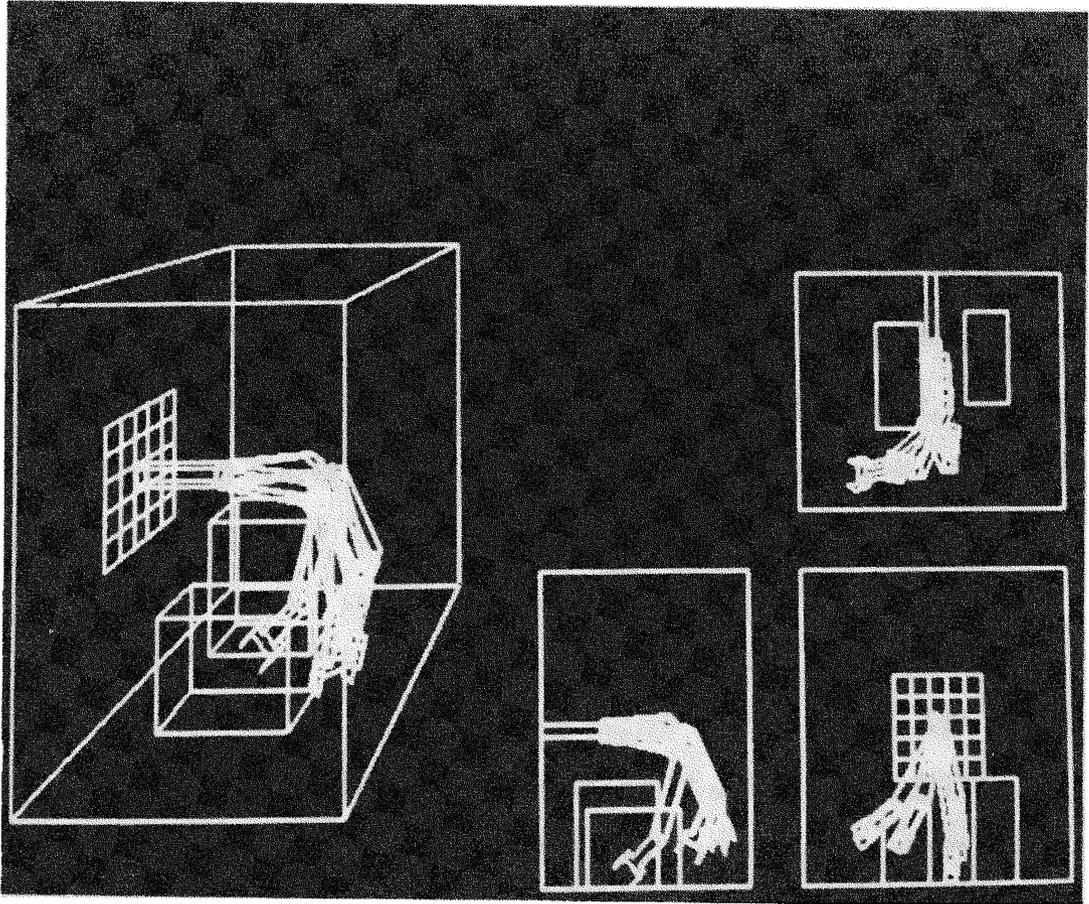


Fig.2-8 An Example of the Collision Avoidance Motion

14]で、目標作業座標は $Y_e^T = [0 \ 0.707 \ -0.707 \ 0 \ -0.707 \ -0.707 \ 0.550 \ 0.350 \ 0.500]$ である。ポテンシャルに対する重みは、先程の例と全

同じである。このことから分かる様に、この重みはかなりラフに決めても、実用上は問題ない。この例では、0.05[m]毎に目標位置を切りかえている。この間隔を小さくすると、手先が安全第一面から離れないので、一般に障害物との干渉は起こりにくい。イテレーション回数は、1112回であった。尚、計算にはマイクロコンピュータを使用しているので、1イテレーションあたり約1秒の演算時間を要している。

ここで示した例以外にも、無作為に選んだ多くの例において本手法の有効性を確認している。特に、余程 w_{ok} を大きくしない限り、停留の問題は起こらないことが確かめられている。

2.6 まとめ

本手法の特徴をまとめると、次の様になる。まず、一般のポテンシャル法と比較すると、

- (1) 目標位置に到達するためのポテンシャルと障害物を回避するためのポテンシャルとの相互干渉が出来るだけ小さくなるように、大局的な見地からポテンシャル場を定義しているので、停留や障害物との衝突による解の探索の失敗が起こりにくい。
- (2) エンドイフェクタの目標軌道を大局的に計画しているので、これに基づく解の探索の再試行が可能である。

また、自由空間法と比較すると、

- (3) 必要な計算量がマニピュレータの自由度数に対して線形にしか増加しない。
- (4) 対象とするマニピュレータの機構学的な特徴に依存せずに適用可能である。

この2つの特徴により、本手法は冗長自由度を持つような一般の多自由度マニピュレータに適用可能である。本手法の問題点は、解が存在するときにそれを必ず見つけることが理論的には保証されないことである。これは、本手法がやはり本質的には発見的な解の探索法であるという理由による。特に、現時点では、ポテンシャル間の重みづけの調節はヒューリスティックに行うしか方法がない。この重み係数の調整をシステマティックに行うことが出来れば、この問題点は幾分改善されると考えられる。この点については、今後の検討課題と考えている。

第3章 適応プロダクションシステムに基づくマニピュレータ動作学習過程のモデリング

3.1 はじめに

一般に、マニピュレータの自律動作計画はかなりの計算量を必要とする。これに対して、人間が腕を動かす場合を考えると、多種多様な動作をそのときの状況と目的に応じて「何となく」やってしまう。このような人間の運動制御機構の特徴の一つとして、それが学習機能を持っていることが挙げられる。最近の認知心理学の研究によって、学習は知識獲得・知識表現・知識利用の三つの側面から考えるべきであることが明らかになって来たが、動作の学習に関してはダイナミクスの学習[Arimoto 1984]や自律的な動作生成アルゴリズム[Lozano-Perez 1982]など、主として知識獲得法について精力的な研究が行なわれている。これに対して、獲得知識の表現・利用法についてはあまり検討されておらず、これら三つの問題を統一的に扱ったのは、倒立振子の制御を例とした研究[Ichikawa 1983]が挙げられるくらいである。以上の観点から、本章ではマニピュレータの障害物回避動作の学習を対象として、その知識獲得・知識表現・知識利用、さらには知識の一般化・特化について、知識表現法の一つである適応プロダクションシステム[Anzai 1974]に基づいて検討した結果について

て述べる[Hirukawa 1986b]。ただし、対象とするマニピュレータはその運動学のみを考える。

3.2 適応プロダクションシステムと動作学習

3.2.1 適応プロダクションシステムとその特徴

プロダクションシステム(Production System、略して、PS)は、「もし~ならば、 --- をする」という条件部と行為部から成るプロダクションルール(Production Rule)の集合であるプロダクションメモリ(Production Memory、略して、PM)、現在の内的・外的状況を示すワーキングメモリ(Working Memory、略して、WM)、全体の制御機構であるインタプリタ(Interpreter)の三者から成る。PSは、与えられたWMの初期状態とプロダクションルールの条件部のパターンマッチングを行ない、その条件部が真となったプロダクションの行為部を実行することによってその処理を開始する。これは、現在の内的・外的状況を示すWMの内容に従って次の行動を決定するのであるから、概念的にはこのパターンマッチングの段階が状況の認識に基づく知識の想起に相当する。この行為部の実行の結果WMの状態が変化し、新しい状態に適合したプロダクションが実行され、これを繰り返すことによって処理は進行する。このサイクルのことを認識・行動サイクルと呼ぶ。プロダクションの行為としてプロダクション自体の変更を導入すれば、PMの内容を逐次変化させていくことが出来る。このことによってPSに学習機能を持

たせたものを適応プロダクションシステム(Adaptive Production System)と呼ぶ。PSの構成をFig.3-1に示す。従来のプログラミング

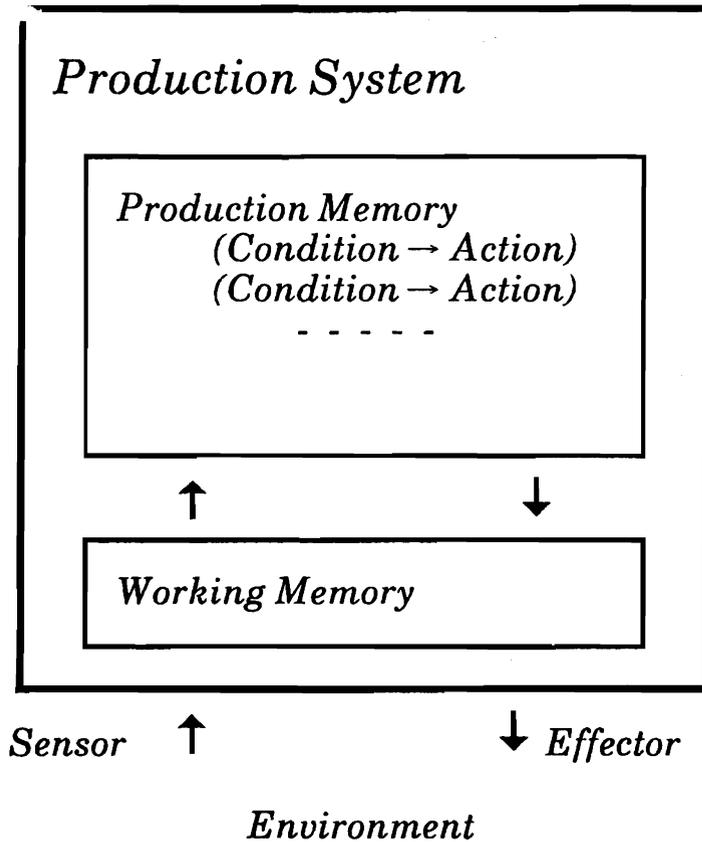


Fig.3-1 The Configuration of the Production System

システムはTuring Machineに基礎をおいていたのに対し、PSは数学的基礎をPost Machine[Post 1943]にしているところが異なり、これはPSの特徴を生み出す一つの要因になっている。その特徴とし

ては[Tsujii 1979].

- (1) 各プロダクションは意味的に独立である。
- (2) 各プロダクションは処理の流れと独立であり、処理の流れはそのときの状態によって決まる(このようなシステムを事象駆動型システムと呼ぶ)。
- (3) 認識・行動サイクルにオーバーヘッドがあるので処理効率が低くなる。

などが挙げられる。

3.2.2 PSの動作学習過程記述に対する適合性

前節で述べたPSの特徴に基づいて、ここでは動作学習過程の記述に対するPSの適合性について、二つの観点から論ずる。

まず、対象システムの性質という観点から考えてみる。対象システムが数式モデルなどの一つの集約的な知識によって記述され、しかもそれに基づく制御が容易であるときは、前節で述べたPSの特徴(3)の処理効率の悪さを考慮すると、従来の制御方式を手続き型プログラミング手法を用いてインプリメントする方が良好な結果が期待できる。しかし、非線形多変数システムや非数値情報で記述されるシステムでは、定式化やパラメータ同定が困難な場合があり、またこれらが可能であったとしてもその数式モデルに基づく制御は非常に困難な場合が多い。人間の手足はこのような対象の一つであると考えられる。このような対象に対して、試行によって獲得した断片的な知識によって制御を行なう方法が考えられる。この断片的な知

識を表現することに対して、前節で述べたPSの特徴(1)の各プロダクションの意味的独立性と、特徴(2)の事象駆動性が非常に適していると思われる。

次に、動作の意味(Semantics)という観点から考えてみる。従来、動作の学習は環境の認識や動作の目的などとは独立に考えられることが多かったと思われる。しかし、マニピュレータの動作でも人間の動作でもそれぞれの動作には意味があり、環境の認識や動作の目的と独立に考えるのはむしろ不自然であると考えられる。これに対して、PSはシンボリックな処理を行なうシステムであるので、視覚情報処理機構から得られる環境の認識結果や内部に持っている目的、あるいは効果器に出力する動作系列を全てシンボリックな情報に変換しておけば、これらを有機的に結合して統一的に処理することが可能である。これにより、動作の学習をその動作の持つ意味とともに扱うことが可能になる。これを実際に行なった例を次節で示す。

3.3 獲得知識の表現と利用

3.3.1 プロダクションの形式と弁別ネット

本節では、2章で述べた方法を知識獲得法として、試行によって獲得した断片的な知識を環境の認識や内部に持っている目的と有機的に結合して表現する方法について述べる。本研究のPSにおいては、知識獲得法と試行によって獲得される断片的な知識をどちらもプロ

ダクションルールとして記述している。知識獲得のためのプロダクションは、

P1)

If (C1) No other production matches WM

then (A1) Apply the potential method

(A2) Generate a new production

と表現される。ここに条件(C1)は現在のワーキングメモリに適合するプロダクションが他にないときに真となることを意味する。行為(A1)は、2章で述べた方法に従って試行を行なうことを意味し、行為(A2)はその試行経験を整理統合し、次の形式のプロダクションを生成するというものである。

P2)

If (C1) Environment is (E-list)

(C2) Goal state vector is \bar{Y}_e

(C3) Present angle vector is Θ^0

then (A1) Move the manipulator through

$\Theta^1, \Theta^2, \dots, \Theta^n$

ここに、(E-list) :現在の作業環境を表現するリスト

\bar{Y}_e :エンドイフェクタの目標位置と姿勢

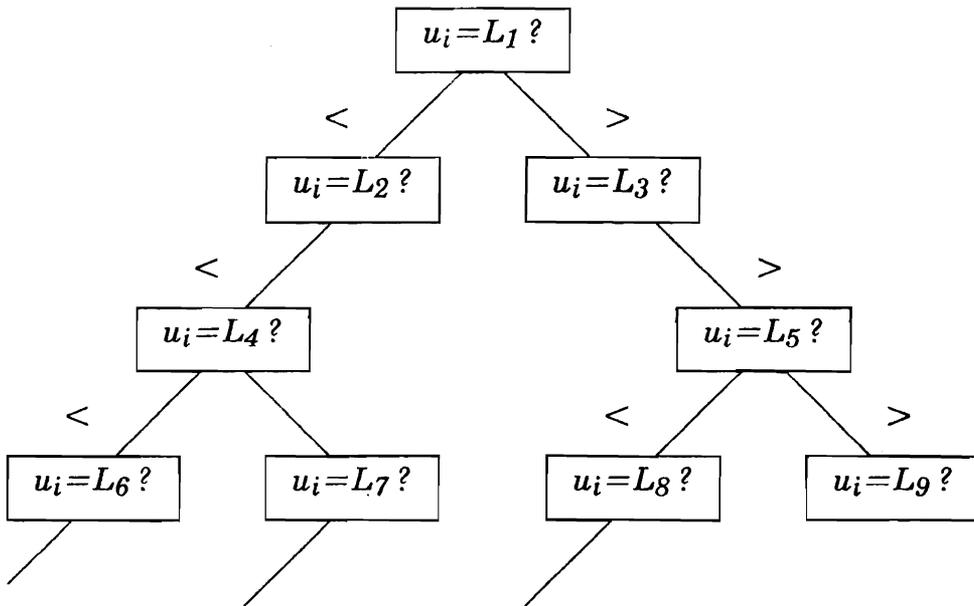
Θ^0 :初期関節角

①i :P1によって生成した関節角の系列

これは、現在の環境が(E-list)で、手先の目標値が \bar{Y}_e であって、マニピュレータの関節角が Θ_0 のときには、 $\Theta_1, \Theta_2, \dots, \Theta_n$ とマニピュレータを動かせば、手先が目標値に到達するということを意味する。そして、このプロダクションを生成するたびに条件部の弁別学習を行ない、弁別ネット[Anzai 1974]の形でプロダクションを記憶していく。この弁別ネットは、もともとはパターンの弁別学習のモデルとして考えられたもので、二分探索木を基礎にしている。これを模式的に示したのが Fig.3-2である。ここでは、これをLISPのデータベース機能を用いてインプリメントしている。獲得知識をこのような形で表現しておく、探索の手数は大まかに言ってプロダクションの数の対数に比例するから、プロダクションP2が多数獲得されたあとでも、ワーキングメモリとプロダクションの条件部のマッチングを効率良く行なうことが可能になる。これは、現在の環境と目的、マニピュレータの状態に適合した知識を効率良く想起できる事を意味する。以上のことを模式的に示したのが、Fig.3-3である。また、ここでのシステムでは、環境、目的、状態からなる階層的な弁別ネットを構成してある。この生成アルゴリズムの中心部分は次の様になる。

procedure GENERATE(条件、ノード)

if 弁別ネットが空 then



u_i : the Numerical Parameters in the Working Memory

L_i : the Numerical Parameters in the productions stored at the nodes

Fig.3-2 The Basic Concept of the Discrimination Net

(条件の第一要素 NIL NIL 下位ネットのノード名)を生成

if 次の条件がない then 行為をノードに記憶

else GENERATE(残りの条件 下位ネットのノード)

else if 条件の第一要素 = ノードの条件 then

if 次の条件がない then このプロダクションは既に学習

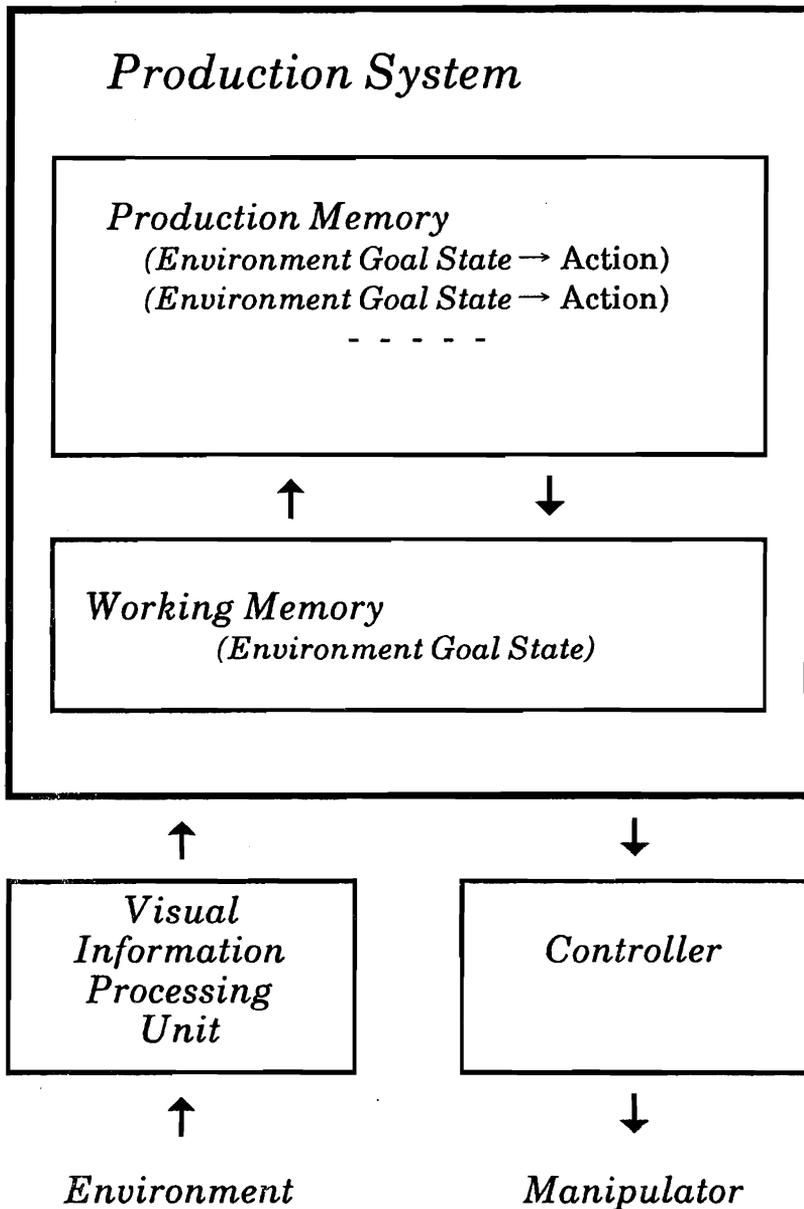


Fig.3-3 The Schematic Illustration of the Knowledge Retrieval

```

else GENERATE(残りの条件 下位ネットのノード)
else if 条件の第一要素 < ノードの条件 then
  if ノードが左の子を持つ
    then GENERATE(条件 左の子ノード)
    else (ノードの条件 新しい左の子 右の子
          下位ネットのノード)
          (条件の第一要素 NIL NIL
          下位ネットのノード) を生成
  if 次の条件がない
    then 行為をノードに記憶
    else GENERATE(残りの条件
                   下位ネットのノード)
else if 条件の第一要素 > ノードの条件 then
  if ノードが右の子を持つ
    then GENERATE(条件 右の子ノード)
    else (ノードの条件 左の子 新しい右の子
          下位ネットのノード)
          (条件の第一要素 NIL NIL
          下位ネットのノード) を生成
  if 次の条件がない
    then 行為をノードに記憶
    else GENERATE(残りの条件
                   下位ネットのノード)

```

#ここに、括弧で囲まれた4要素からなるリストは弁別ネットのノードを表わし、

(判断条件 左の子ノード名 右の子ノード名

下位ネットのノード名 or プロダクションの行為)

という内容からなる。

この階層化によって、想起の対象となる知識の探索域の絞り込みが可能となる。例えば、ある動作中に環境と目的が変化しないとすると、ワーキングメモリにおいて変化するのはマニピュレータの状態だけであるから、これのみを環境と目的に対応する上位の弁別ネットによって制限された下位の弁別ネットに通せば、適合するプロダクションを選択することが出来る。この様子を示したのが、Fig.3-4である。

3.3.2 獲得知識の一般化と特化

前節で述べたプロダクションP2)は、ある特定の経験に基づいて生成したものであるから、再び全く同じ状態にならないと利用することは出来ない。すなわち、条件(C1)-(C3)が厳密に成立する状況にならないとプロダクションP2)は起動しない。これに対して、生物、特に人間は、前に経験したのと似た状況になれば、過去に獲得した知識を生かしてこれに対処することが出来る。これは、特定の経験に基づいて獲得された知識をより多くの場合に利用できるよう

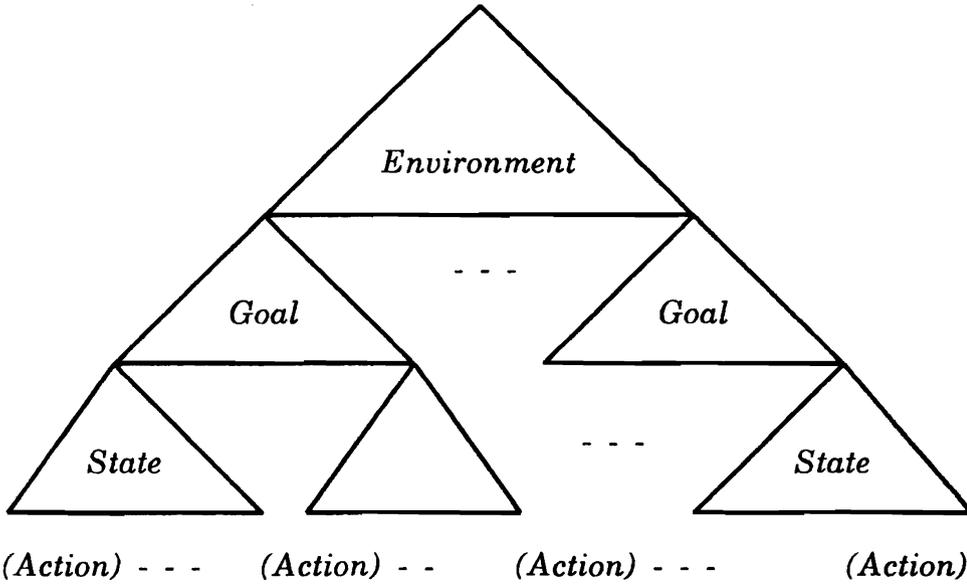


Fig.3-4 The Structure of the Hierarchical Discrimination Net

に、知識をより汎用的なものに変換しているからであると考えられる。これを知識の一般化と呼ぶ。そこで、本システムにおいても知識の一般化を考える。プロダクションシステムにおける知識の一般化について次のような定義[Vere 1978]がある。すなわち、

定義 プロダクションPAがプロダクションPBの一般化であるのは、

$$\forall \sigma, \sigma' ; PB(\sigma \rightarrow \sigma') \text{ ならば } PA(\sigma \rightarrow \sigma')$$

のときであり、かつ、そのときに限る。ここに、 σ 、 σ' はWMの状態を表わし、 $P(\sigma \rightarrow \sigma')$ はプロダクションPによって σ が σ' に書き換えられることを表わす。

これは、プロダクションPBによって引き起こせる任意のWMの変化は、PAによっても引き起こせるということで、言い換えれば、PBによって出来る処理は全てPAによっても処理可能であるという意味である。

以下、プロダクションの一般化はこの定義によるものとする。本システムについて考えると、前節のプロダクションP2)の条件部の数値パラメータはWM中の対応する数値に全く等しくないと、条件は真とはならない。これは、Fig.3-2の探索木のノードの条件に相当する。プロダクションP2)を一般化するためにこのノードの条件を次のように緩和する。

$$L - \varepsilon_i \leq u \leq L + \varepsilon_i \quad (3-1)$$

ここに、 u : WM中のi番目の数値パラメータ

L : ノードに記憶されている各プロダクション中の数値パラメータ

ε_i : 許容範囲を示すパラメータ

すなわち、条件の成立に対して許容範囲を設けるのである。このときの探索木を模式的に示したのがFig.3-5である。ここで、 ε_i の値

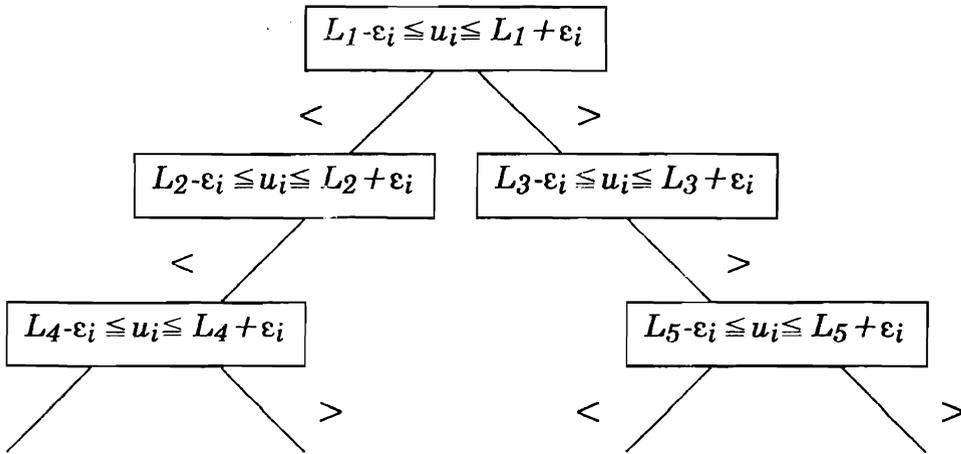


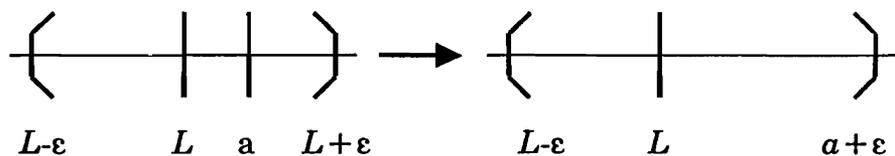
Fig.3-5 The Basic Concept of the Discrimination Net for the Generalized Productions

は対象領域に応じて適当に決めなければならない。このようにプロダクションの条件部を変換すると、変換後のプロダクションは、前述した定義に従って、変換前のプロダクションの一般化になっている。そして、この一般化されたプロダクションが後に想起され、かつこの想起に基づく試行が成功したときは、想起を始めたときの数値パラメータの値を a として、

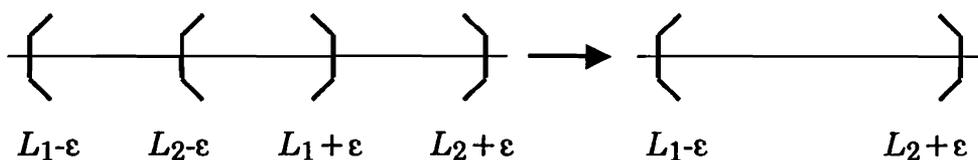
$$\min(L - \varepsilon_i, a - \varepsilon_i) \leq u \leq \max(L + \varepsilon_i, a + \varepsilon_i) \quad (3-2)$$

とさらにプロダクションを一般化する。これは、試行の成功によって許容範囲の境界をより広げるということである。これを Fig.3-6(A)に示す。また、他の試行の結果、条件の有効範囲が二つ以上のノードで重なり合う場合、すなわち、

(A)



(B)



(C)

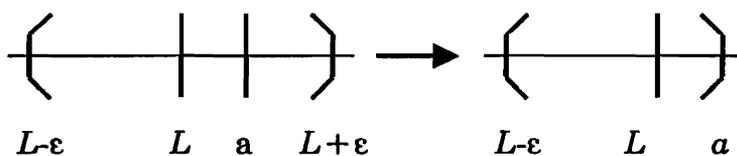


Fig.3-6 The Generalization and Specialization of Productions by the Adaptation of the Numerical Parameters

$$L_1-\epsilon_i \leq u \leq L_1+\epsilon_i, L_2-\epsilon_i \leq u \leq L_2+\epsilon_i \quad (3-3)$$

とするととき

$$\max(L_1-\epsilon_i, L_2-\epsilon_i) < \min(L_1+\epsilon_i, L_2+\epsilon_i) \quad (3-4)$$

となるときは、この二つのノードを統合し、

$$\min(L_1-\epsilon_i, L_2-\epsilon_i) \leq u \leq \max(L_1+\epsilon_i, L_2+\epsilon_i) \quad (3-5)$$

という条件に再構成する。これを Fig.3-6(B)に示す。そして、行為

部は適当な基準にどちらかのプロダクションのものを選択するかあるいは両者の中間的な行為を新しく生成する。このような再構成を行えば、想起の対象となるプロダクションの数を減らすことが出来るので、記憶容量の点からも知識の探索に要する時間の点からも望ましいと言える。

次に、プロダクションの特化について述べる。プロダクションの特化とは、一般化の逆の概念で、プロダクションを適用範囲のより狭いものに縮小するということである。プロダクションの適用結果が失敗に終わったときに、再び同じ失敗を繰り返さないように、このプロダクションの特化を行なうのである。本システムの場合について考えると、(3-1)式のようにプロダクションの条件を緩和することによって知識の一般化を行なうと、状況によっては経路の途中で障害物と干渉して、想起に基づく試行が失敗する場合は考えられる。このときは、想起を始めたときの数値パラメータの値を a として、(3-1)式の条件を

$$a < u \leq L + \varepsilon_i, \text{ when } a < L \quad (3-6)$$

$$L - \varepsilon_i \leq u < a, \text{ when } L \leq a \quad (3-7)$$

と厳しくすることによって、プロダクションを特化する。これを、Fig.3-6(C)に示す。このような一般化と特化を繰り返すことに

よって、プロダクションの有効な範囲を経験によって学習することが出来る。

3.3.2 知識の構造の変化として捉えた動作学習過程

ここでは、以上述べた動作学習過程全体について考えてみる。

最初、過去に試行経験のない状況でマニピュレータを動かすときは、プロダクションP1)を起動し、かなり複雑な演算処理に基づいて動作を行なわなければならない。しかも、このとき手先の位置・姿勢と関節角が内界あるいは外界センサーによって随時計測される必要があるから、これは閉ループ制御である。これに対して、プロダクションP2)が多数獲得された後、過去に経験のある状況でマニピュレータを動かす場合は、現在の環境・目的・状態に適合したプロダクションP2)を想起して来れば、目的の動作を行なうことが出来る。しかも、プロダクションP2)の集合を弁別ネットの形で記憶してあるので、この想起はかなり速く行なうことが可能である。また、動作中にはセンサーの値を必要としないのでこれは開ループ制御である。すなわち、学習前の動作は複雑な演算処理に基づく閉ループ制御によって行なわれているが、学習後の動作は過去の経験によって獲得した知識の選択に基づく開ループ制御になっている。人間の動作学習がどのようなメカニズムによって行なわれているかはあまり良く分かっていないが、以上述べたように知識の構造の変化として動作学習過程を捉えることも必要なのではないかと考えられる。

ここで、前節のプロダクションの一般化に際して、目標値に範囲を持たせたことについて考える。この場合は、目標値に完全には達しない。そこで、

「全動作の80~90%位までは知識の想起に基づく開ループ制御を行ない、最後の10~20%は閉ループ制御を行なう。」

とすれば、開ループ制御の間は演算処理もセンサーからの値も必要としないので過去の経験を生かすことが出来、なおかつ目標値に到達させることが出来る。ここで、開ループ制御と閉ループ制御を全動作の80~90%で切り替えるのは、切り替え点が目標点に近すぎるとそこで滑らかな動作が出来ないからである。

3.4 インプリメンテーション

3.4.1 システム構成

本研究では、コントローラであるプロダクションシステムは大型計算機(ACOS6-S__1000,NEC製)のLISP1.9でインプリメントしている。また、制御対象であるマニピュレータはマイクロコンピュータ(CPU-MC68000,OS-CP/M68K,language-C,SORD製)上のジオメトリックシミュレータで実現し、これを端末として接続している。このジオメトリックシミュレータには障害物も表示でき、これらの2次元表示には透視変換を用いている。

3.4.2 プロダクションの例

初期姿勢が $\Theta^T=[0.26,0.26,0.26,0.26,0.26,3.93,0.77](\text{rad})$ 、エンド

イフェクタの目標位置・姿勢が $Y_e = [-0.097, 0.33, -0.94, -0.99, -0.085, 0.072, -0.10, 0.050, 8.0]$ (m)で、一辺0.2(m)の立方体の障害物がある場合の動作の際に生成されるプロダクションは、

```
((CUBOID (ORIGIN 0 -0.030 1.0)
  (X_VECTOR 0.20 0.0 0.0)
  (Y_VECTOR 0.0 0.20 0.0)
  (Z_VECTOR 0.0 0.0 0.20))
(GOAL_STATE -0.097 0.33 -0.94 -0.99 -0.085 0.072
  -0.10 0.050 8.0)
(PRESENT_ANGLE 0.26 0.26 0.26 0.26 0.26 3.93 0.77)
'->'
(MOVE 0.66 -0.0027 0.47 0.43 0.34 3.67 0.39)
(MOVE 0.60 -0.091 0.41 0.55 0.43 3.56 0.073)

-----

(MOVE 0.65 -0.86 0.50 0.96 0.69 3.78 -0.66))
```

となる。ここに、プロダクション中の各アトムを持つ意味は、前節に示したプロダクションP2の定義から明らかであると思うので、参照されたい。

3.5 まとめ

障害物回避動作の学習を例にとり、マニピュレータの動作学習過程をモデリングしてみた。得られた結果を以下にまとめる。

- (1) 適応プロダクションシステムの特徴とその動作学習過程記述に対する適合性について論じた。
- (2) 適応プロダクションシステムに基づいて獲得知識を表現・利用する方法について示し、その際、弁別ネットが効率の良い知識の想起に対して有効に働くことを述べた。
- (3) 獲得知識の一般化と特化について述べ、これらによって獲得知識の有効範囲が学習されることを示した。
- (4) 動作学習過程全体について考え、ここで考えたモデリングにおいては、学習前の動作は演算処理に基づく閉ループ制御であったのが、学習後は経験によって獲得した知識の想起に基づく開ループ制御になることを述べた。
- (5) 適応プロダクションシステムをインプリメントし、学習過程のシミュレーションを行った。

今後は、ここで述べたような、動作の学習過程を知識の構造の変化として捉えるという考え方をさらに検討していきたいと考えている。

第4章 マニピュレータの自律的動作計画システム

4.1 はじめに

マニピュレータのための自律的な動作計画システムには、作業環境を認識する機能、それに基づいて動作を計画する機能、計画された動作をオフラインで検証する機能、実機の動作を制御する機能などが要求される。本研究で、これらの機能を持つプロトタイプシステムを開発した[Hirukawa 1986c]。本章では、このシステムについて述べる。

以下、4.2節においては本システムの構成要素と全体構成について述べ、4.3節では作業環境教示システム、4.4節ではオフラインシミュレータ、4.5節では制御システムについてそれぞれ詳しく説明する。さらに、4.6節では自由動作の計画を例とした本システムの運用例を示し、最後に4.7節では本章のまとめを行う。

4.2 システム構成

4.2.1 システムの構成要素

マニピュレータの作業を知的に計画するためには、まず、作業対象や障害物などから成る作業環境の情報を取得しなければならない

い。初期のマニピュレータのプログラミングシステム[Takase 1981, Mujtaba 1981]において用いられていた作業環境に関する情報は、作業対象物の特徴箇所の位置と姿勢を表す直交座標フレームのみであった。しかし、作業を自律的に計画するには、少なくとも作業環境を構成する各物体の3次元形状とその位置・姿勢の情報が必要である。

ここで問題となるのが、この作業環境に関する情報を如何にして計算機に入力するかである。現時点ではこれを計算機に全く自動的に認識させるのは極めて困難であり、逆に逐一人間が記述するのはあまりにも煩雑である。この問題に対する解決策として、三角測量の原理に基づく距離計測と対話を組み合わせて、特徴箇所の位置[Grossman 1978]や対象物体の3次元形状とその位置・姿勢[Hasegawa 1981]の情報を容易に得る方法が提案されている。この距離計測には様々な方法が提案されている[Jarvis 1983]が、本研究ではグレイコード状の光パターンを用いた距離計測法[Sato 1985]に基づいて、対象物体の3次元形状とその位置・姿勢の情報を得るシステムを試作した。

次の問題は、この3次元形状とその位置・姿勢の情報をどのような形で表現するかということである。3次元物体の形状モデリング技術は、CAD/CAMやコンピュータグラフィックスの分野において開発されてきた[Baer 1979, Requicha 1980, Okino 1982]が、近年ロボットシステム[Popplestone 1980, Wesley 1980, Matubara 1985]にも組み込まれつつある。本システムでは、作業環境の教示の際には

CSG(Constructive Solid Geometry)[Requicha 1980]タイプのデータ構造を用い、後の応用に備えてこれをB-reps(Boundary Representations)[Requicha 1980]タイプのデータ構造に内部で自動変換する方法を取った。

この作業環境のモデルに基づいて、マニピュレータの動作を計画するのであるが、この際、計画された動作が所期の目的を達成しているかどうかを検証する必要がある。この検証を行うことなしに実機を動作させてしまうと、動作目的を達しないことがあるだけでなく、場合によっては作業環境やマニピュレータを破損する可能性もある。既に、幾つかのオフラインシミュレーションシステムが開発されている[Matuya 1985]が、センサシステムと結合されているものは少ない。本システムでは、作業環境教示システムによって得た実環境のモデルに基づいて、マニピュレータ動作のグラフィックアニメーション、数値計算によるマニピュレータと障害物の間の干渉チェックを行えるオフラインシミュレータを開発した。

以上の方法により計画された動作は、開発した7自由度多関節型のマニピュレータにより実行する。前述したオフラインシミュレータ中のマニピュレータのカイネマティクスは、この実機に合わせてある。また、制御アルゴリズムには、各関節独立にPID制御を用いている。

4.2.2 システムの全体構成

本システムのブロックダイアグラムをFig.4-1に、全景をFig.4-2

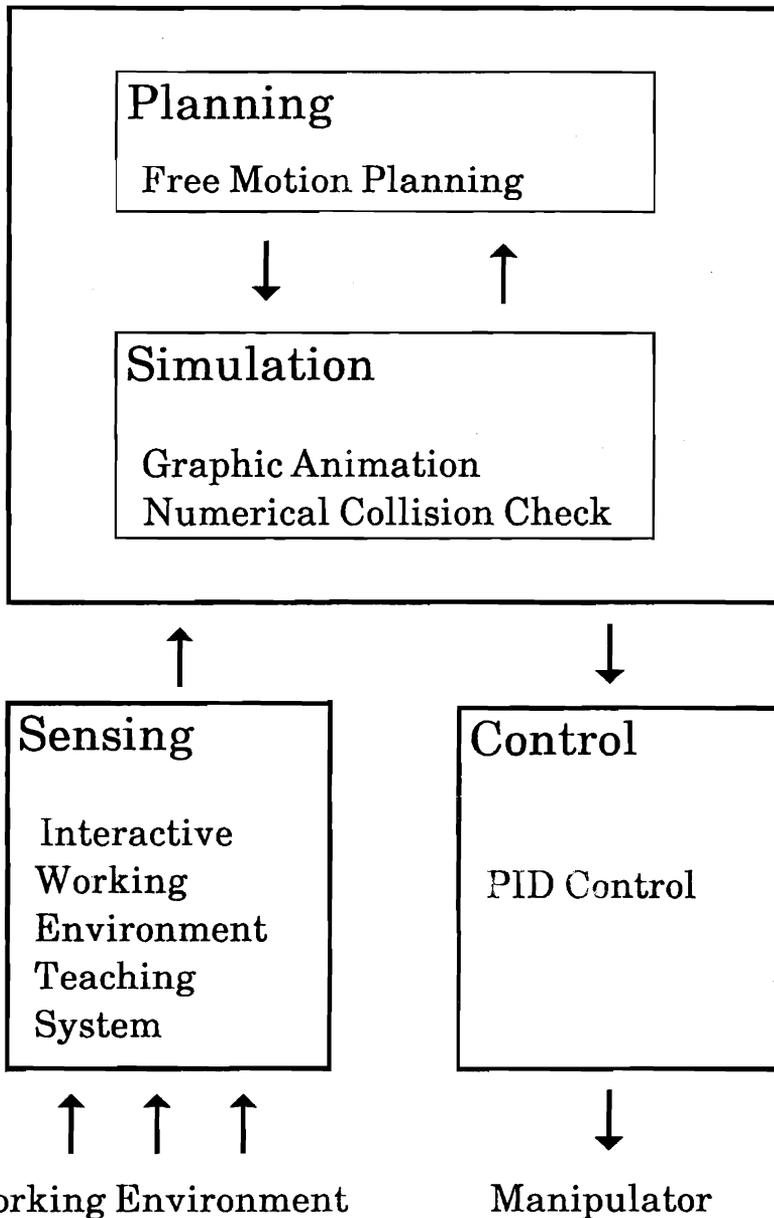


Fig.4-1 The Configuration of the Autonomous Motion Planning System for Robot Manipulators

に示す。本システムの主要部分は、マイクロコンピュータ上に構築

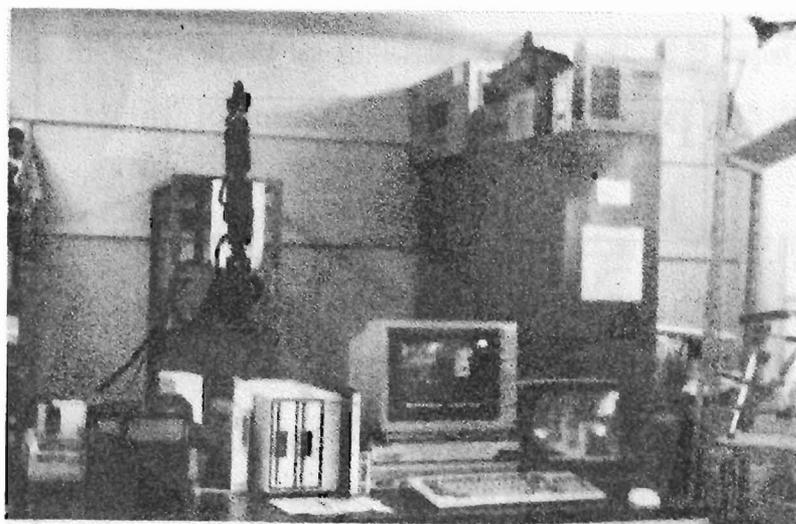


Fig.4-2 The Whole View of the System

している。この仕様を Table 4-1 に示す。開発言語には、Cを用い

	PC-9801
CPU	i8086-2(8MHz)
Co-processor	i8087-2(8MHz)
RAM	640kB
OS	MS-DOS
Language	Optimatizing C

Table 4-1 The Specifications of the Micro-Computer

た。これは、実行が高速であること、ポートの入出力などの細かい処理も記述可能なこと、構造化プログラミングに適しているなどの理由による。また、各ソフトウェアは出来るだけモジュール化

し、ソフトウェア間のデータの伝達は、ファイルを介して行った。本システムにおける各ソフトウェアモジュールの関係をFig.4-3に示す。

この図において、CSGはCSGをデータ構造とする作業環境のモデルを示し、B-repsはこれをB-repsタイプのデータ構造に変換したものであることを意味している。Desired Trajectoryはプランニングの結果得られる関節空間における目標軌道であり、Trajectory Planning モジュールでこれを時間の関数に変換して、実機の目標軌道として用いる。

4.3 作業環境教示システム

4.3.1 空間コード化による距離計測

試作した距離計測システムは、CCDカメラ(SONY製、XC-37、Table 4-2 参照)、液晶シャッター付きスライドプロジェクタ、画像処理専用装置(ADS製、IP-4、Table 4-3 参照)、マイクロコンピュータから構成されている。この原理図をFig.4-4に示す。この図において、CCDカメラは点Oに、スライドプロジェクタは点Lに位置しているとする。まず、図中にA~Cで示してある様なマスクパターンを通してスライドプロジェクタの光を空間に投射し、それをCCDカメラで撮像して画像処理装置内のフレームメモリに入力する。そして、マスクパターンを反転させたスライドプロジェクタの光を投射した画像とそれぞれの画像の差を取り、それを適当なス

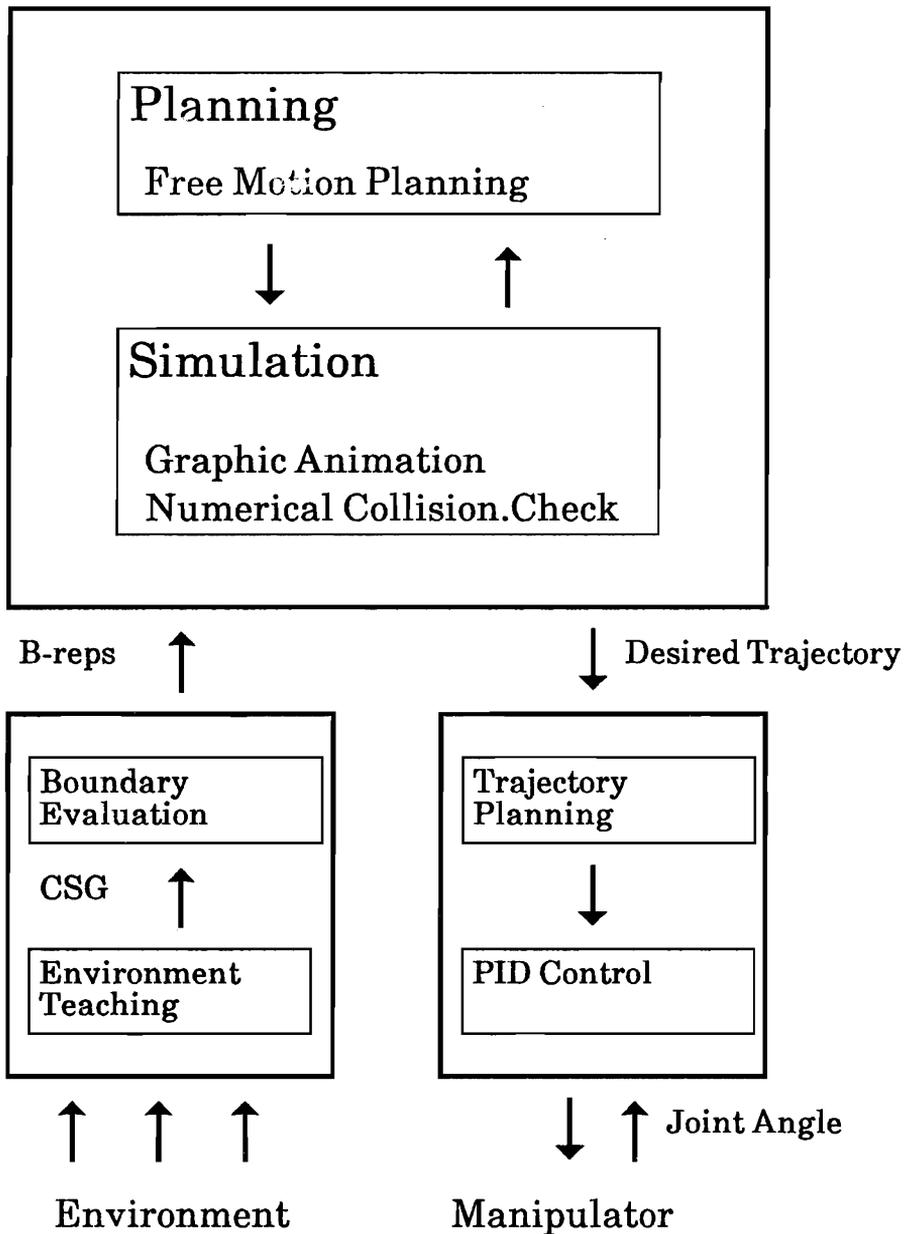


Fig.4-3 The Software Modules and their Relationships

レッシュヨルドで2値化する。最後に、Aのパターンから得られた2値

	XC-37
Efficient Pixes	384(H) x 491(V)
Imaging Area	8.8 x 6.6 mm
Scanning Type	525(H)60Hz(V) 2:1 Inter.
Resolution	280(H)350(V)
Sensitivity	400 lux F4
S/N	46dB

Table 4-2 The Specifications of the CCD Camera

	IP-4
Image Memory	512(H) x 512(V) x 16bit x 1
	512(H) x 512(V) x 8bit x 2
Graphic, Cursor Memory	512(H) x 512(V) x 1bit x 1
Character Memory	30(H) x 27(V)
Input Video Signal	Composite Video Signal (2:1 Inter.)
	1 Vpp/75Ω, 15.75kHz(H)60Hz(V)
Output Video Signal	ibid
A-D, D-A Converter	8bit, 10MHz
Interface	GP-IB, RS-232C

Table 4-3 The Specifications of Image Processor

画像をMSB、Cのパターンから得られた2値画像をLSBとする2進数3ビットの画像を作る。プロジェクトの画角が予め求められているとすると、このコード化画像から各々の画素のプロジェクトから

0 1 0 1 0 1 0 1 ---C
 0 0 1 1 0 0 1 1 ---B
 0 0 0 0 1 1 1 1 ---A

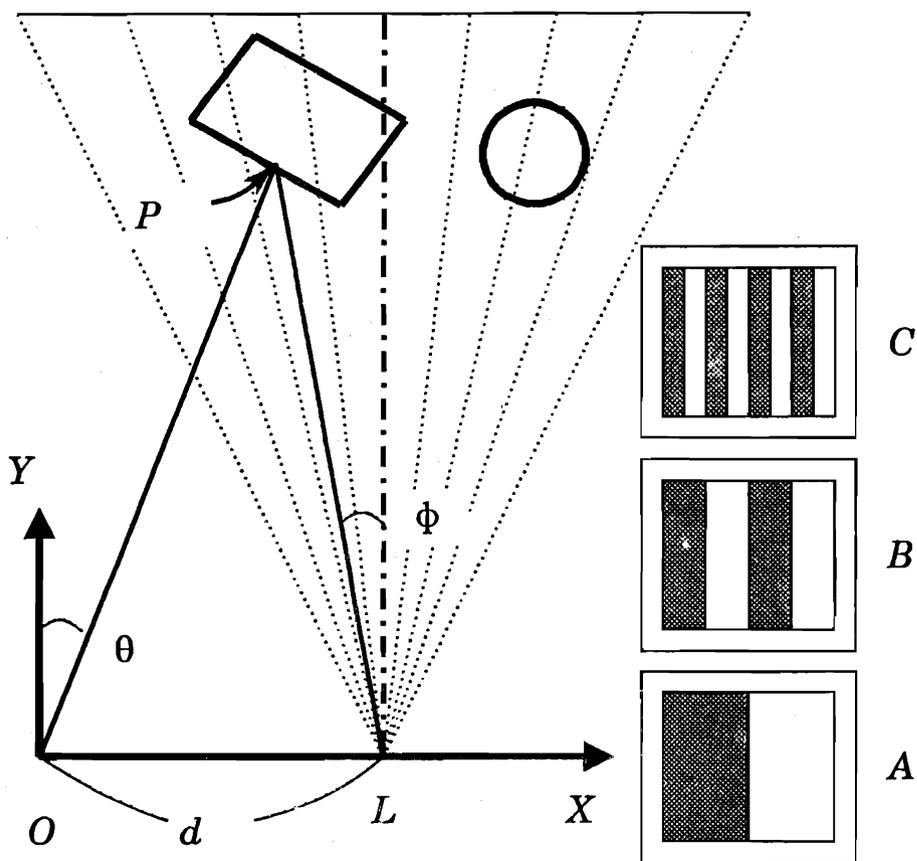


Fig.4-4 Principle of the Range Finding System based on the Spatial Coding

の角度 ϕ を知ることが出来る。例えば、図中の点PはマスクパターンAに対しては0に、Bに対しては1に、Cに対しては0にコード化さ

れ、これらから '010' とコード化される。また、カメラからの角度 θ も、カメラの画角が分かっているとすると、画素の位置から得られる。結局、三角測量の原理から、点Pの位置(x, y)は次の様に求められる。

$$x = y \tan\theta \quad (4-1)$$

$$y = \frac{d}{\tan\theta - \tan\phi} \quad (4-2)$$

本計測システムにおいては、マスクパターンは液晶シャッタにより実現し、画像間の差演算・2値化・合成の処理は画像処理専用装置を用いて実時間で実行している。実際の測定においては、7枚のマスクパターンを用いて7bitにコード化している。また、コード化エラーによる誤差を最小にするために、隣合うコード間のハミング距離が常に1であるグレイコードを用いている。マスクパターンを投射した作業環境の例を Fig.4-5に、これらから得られたコード化画像を Fig.4-6に示す。

以上の議論は、本計測システムの原理を示すためのもので、以下に3次元空間でのより一般的な距離計測原理について詳細に述べる。

まず、カメラの光学系を Fig.4-7 の様にモデル化する。ここに、 f はレンズの焦点距離、 $X_C Y_C Z_C$ -系はカメラの結像面の中心に原点を持つ座標系、 $X_L Y_L Z_L$ -系はレンズの中心に原点を持つ座標系で、図の様に各座標軸の向きを取る。このとき、 $X_L Y_L Z_L$ -系で表した点

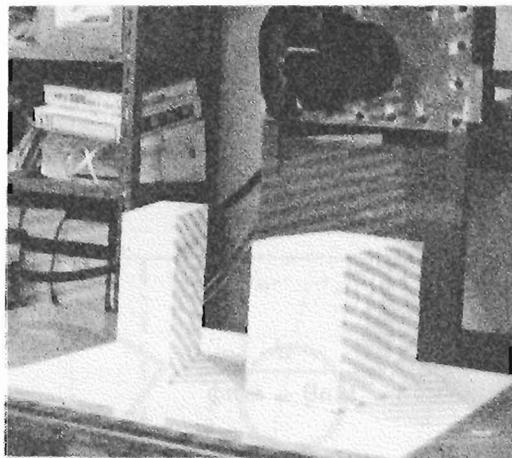


Fig.4-5 An Example of the Working Space with the Coded Light



Fig.4-6 An Example of the Coded Image

(X_L, Y_L, Z_L) は、 $X_C Y_C Z_C$ -系で表した点 (X_C, Y_C) に、

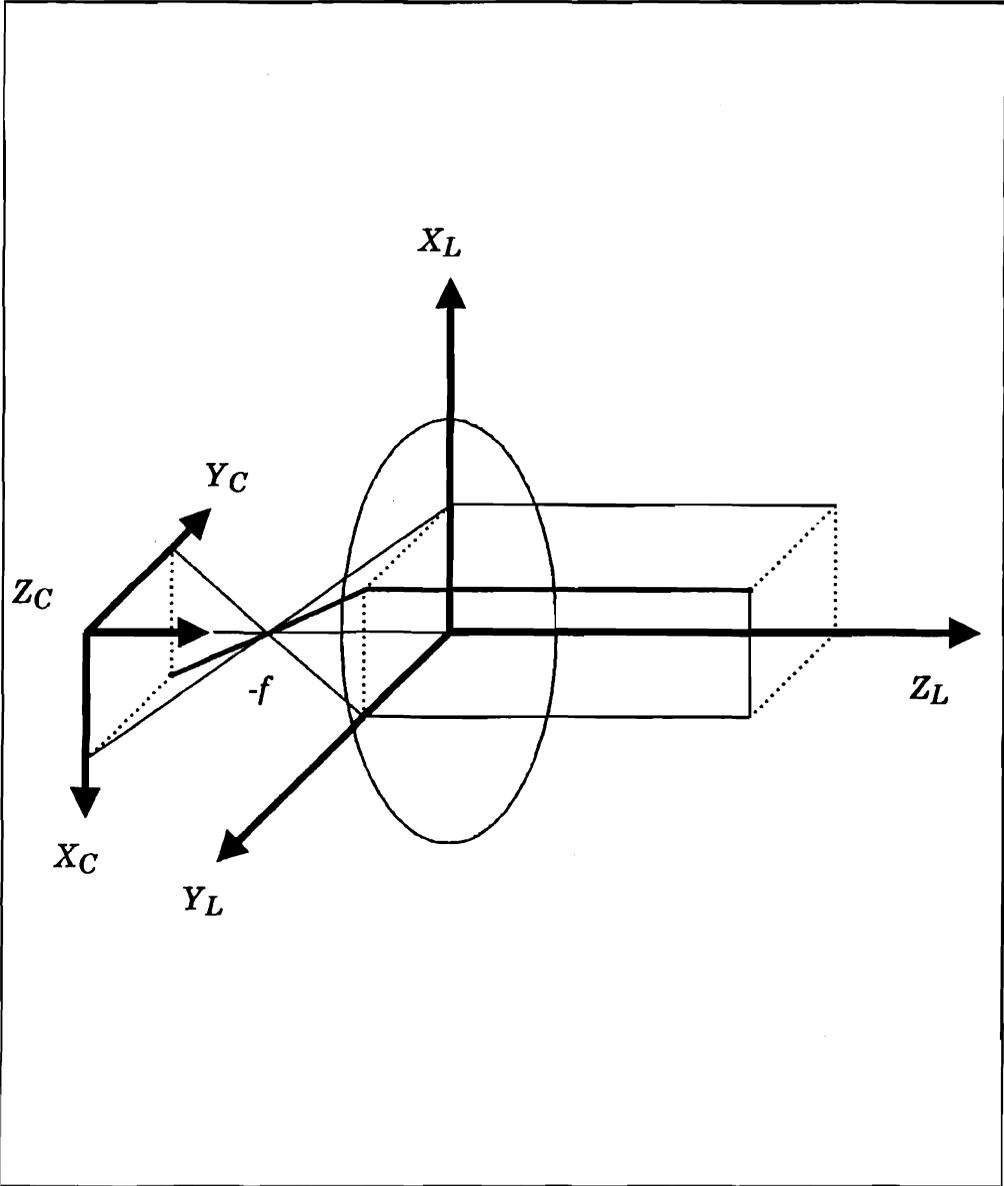


Fig.4-7 Camera Frame

$$\begin{bmatrix} s_C \cdot X_C \\ s_C \cdot Y_C \\ s_C \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1/f & 1 \end{bmatrix} \begin{bmatrix} X_L \\ Y_L \\ Z_L \\ 1 \end{bmatrix} \quad (4-3)$$

という関係で結像する。また、結像面とレンズとの距離 h_L は、

$$h_L = \frac{Z_L}{1 - Z_L/f} \quad (4-4)$$

となるが、これはレンズの焦点距離 f を調節することによって一定となる。結像点の位置は、簡単な変換によりフレームメモリ中の画素の位置(I,J)と対応づけられる。これは、次の様にかける。

$$\begin{bmatrix} s_{IJ} \cdot I \\ s_{IJ} \cdot J \\ s_{IJ} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_C \cdot X_C \\ s_C \cdot Y_C \\ s_C \end{bmatrix} \quad (4-5)$$

一方、 $X_W Y_W Z_W$ -系を作業空間のワールド座標系とすると、このワールド座標系で表された点 (X_W, Y_W, Z_W) は、

$$\begin{bmatrix} X_L \\ Y_L \\ Z_L \\ 1 \end{bmatrix} = \begin{bmatrix} B_{11} & B_{12} & B_{13} & B_{14} \\ B_{21} & B_{22} & B_{23} & B_{24} \\ B_{31} & B_{32} & B_{33} & B_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \quad (4-6)$$

と $X_L Y_L Z_L$ -系での座標値に変換できる(Fig.4-8参照)。(4-5),(4-3),(4-6)式より、

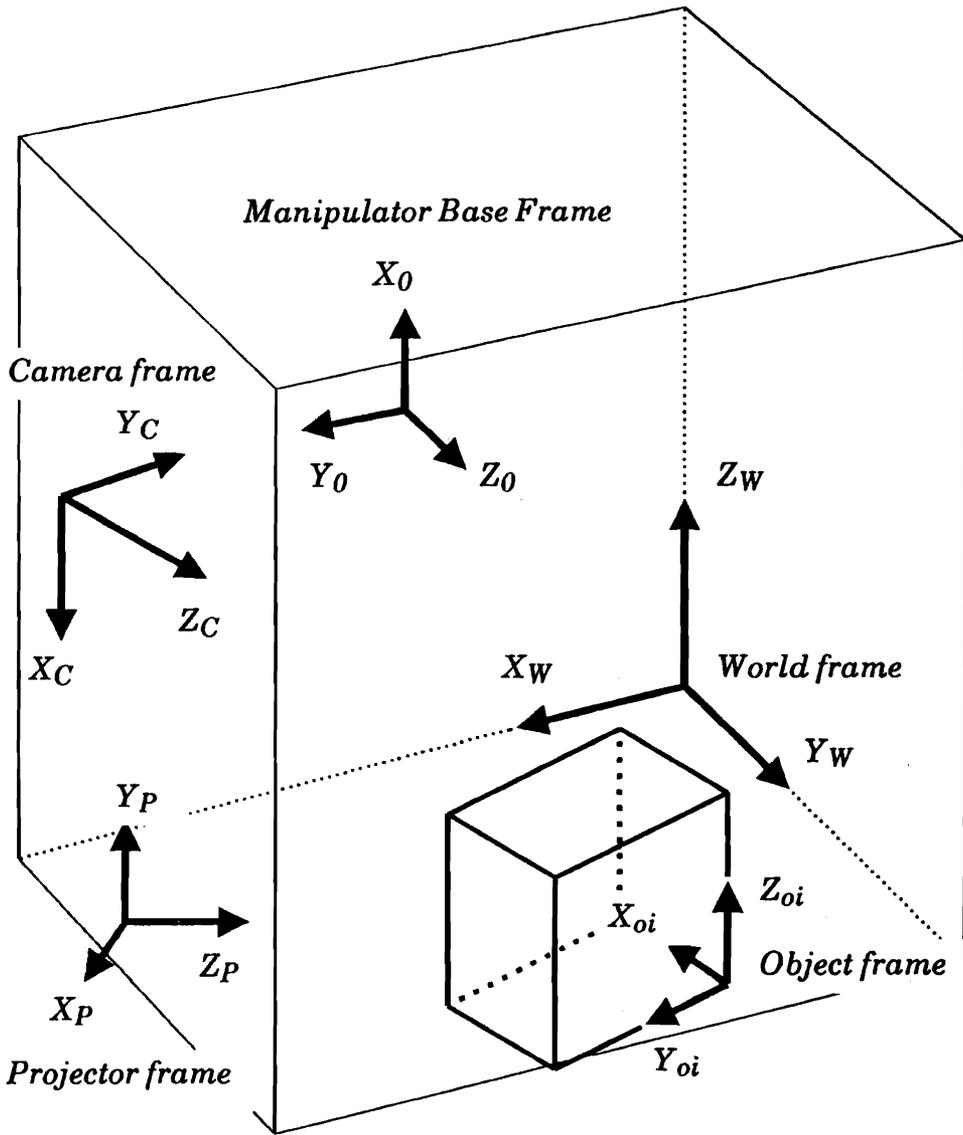


Fig.4-8 The Configuration of Frames

$$\begin{bmatrix} s_{IJ} \cdot I \\ s_{IJ} \cdot J \\ s_{IJ} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \quad (4-7)$$

という関係が得られる。

次に、プロジェクタの光学系を Fig.4-9の様にモデル化する。ここに、 $X_P Y_P Z_P$ -系はレンズの中心に原点を持つ図の様な座標系である。このとき、光源から出て $X_P Y_P Z_P$ -系で表した点 (X_P, Y_P, Z_P) へ投射される光が Y_L 軸を横切る座標 Y_{PL} は、

$$\begin{bmatrix} s_{PL} \cdot Y_{PL} \\ s_{PL} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1/h_p & 1 \end{bmatrix} \begin{bmatrix} X_P \\ Y_P \\ Z_P \\ 1 \end{bmatrix} \quad (4-8)$$

となる。 $X_P Y_P$ -平面にマスクパターンがあると考え、2進コード K と Y_{PL} の間には、

$$\begin{bmatrix} s_K \cdot Y_K \\ s_K \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_{PL} \cdot Y_{PL} \\ s_{PL} \end{bmatrix} \quad (4-9)$$

という線形関係が成立する。この変換マトリクスの要素は、コード化の方法により決まる。また、平面のマスクパターンを光学系において図の様にマウントすると、光源からの投射角と Y_{PL} の間には線形関係が成立しないことを注意しておく。カメラの場合と同様に、ワールド座標系で表された点 (X_W, Y_W, Z_W) は、

$$\begin{bmatrix} X_P \\ Y_P \\ Z_P \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} E_{11} & E_{12} & E_{13} & E_{14} \\ E_{21} & E_{22} & E_{23} & E_{24} \\ E_{31} & E_{32} & E_{33} & E_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \quad (4-10)$$

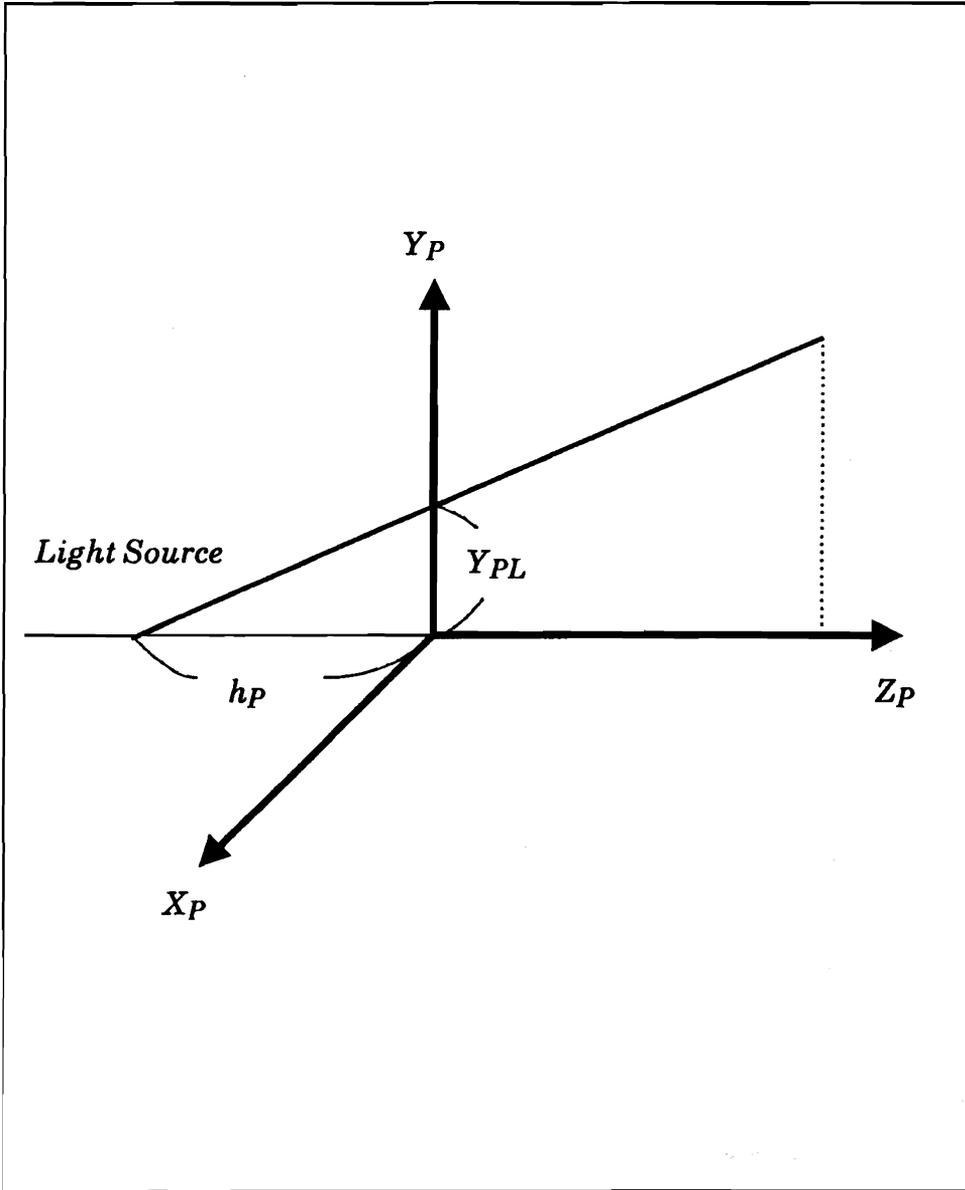


Fig.4-9 Projector Frame

と $X_p Y_p Z_p$ -系での座標値に変換できる。ただし、ワールド座標系は右手系、 $X_p Y_p Z_p$ -系は左手系なので、(4-10)式は右手系から左手系へ

の変換と、回転・移動変換の積である。(4-9),(4-8),(4-10)式より、

$$\begin{bmatrix} s_K \cdot Y_K \\ s_K \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \quad (4-11)$$

という関係が得られる。(4-7),(4-11)式より、次の関係が成立する。

$$\begin{bmatrix} C_{11}-C_{31} \cdot I & C_{12}-C_{32} \cdot I & C_{13}-C_{33} \cdot I \\ C_{21}-C_{31} \cdot J & C_{22}-C_{32} \cdot J & C_{23}-C_{33} \cdot J \\ P_{11}-P_{21} \cdot K & P_{12}-P_{22} \cdot K & P_{13}-P_{23} \cdot K \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \end{bmatrix} = \begin{bmatrix} C_{34} \cdot I - C_{14} \\ C_{34} \cdot J - C_{24} \\ P_{24} \cdot K - P_{14} \end{bmatrix} \quad (4-12)$$

この式から、全てのパラメータが既知であるとする、左辺の係数行列の逆行列が存在するときは、測定値(I,J,K)から作業空間の点の位置(X_W,Y_W,Z_W)を求めることが出来る。尚、カメラパラメータC₁₁~C₃₄、プロジェクタパラメータP₁₁~P₂₄は、測定に先立ちキャリブレーションにより求めている。

4.3.2 対話入力とジオメトリックモデル

一般に、3次元のシーンを計算機に自動的に認識させるのは、現段階では非常に困難である。そこで、本システムでは前節で述べた距離計測と対話を組み合わせることにより、作業環境のモデルを得る方法を探っている。以下に、これを説明する。

ここで、対象物体の3次元形状のモデルにどんなデータ構造を用いるかという問題がある。ソリッドモデルのデータ構造は、CSGとB-repsに大別される [Requicha 1980]。CSGは、直方体や円柱など比較的単純な3次元の基準物体(プリミティブ)を集合演算で結合することにより複雑な物体を表現する方法で、データ量が少なく

人間にも分かり易いという長所を持つが、グラフィック表示や干渉チェックなどの応用には適さない。B-repsは、面・辺・頂点をポイントで結び、トポロジー構造によって物体を表現する方法で、CSGとは逆に、データ量が多く人間には分かりにくい反面、応用には適している。そこで、本システムでは対話入力にさいしてはCSGを用い、これに境界評価[Tivole 1980, Requicha 1985]を施すことによりB-repsに変換するという方法を採用した。CSGのプリミティブとしては、直方体と円柱を用意した。それぞれのプリミティブに対するパラメータをFig.4-10に示す。

対話入力は、以下の様な手順で行う。まず、これから教示するプリミティブの種類をキーボードから入力する。次に、カメラの像をモニタディスプレイで見ながら、そのプリミティブの特徴点を順にマウスを用いてポインティングする。このプロセスの例をFig.4-11に示す。そうすると、これらの特徴点の世界座標系における位置は距離計測システムによって既に求められているので、この値からプリミティブの大きさ・位置・姿勢を表すパラメータを全て求めることが出来る。この様なプロセスを繰り返すことにより、対象物体のCSGモデルを構築している。

本システムで用いているB-repsのデータ構造をFig.4-12に示す。このデータ構造により、複数個の任意の多面体(凹多面体を含む)から成る作業環境を表現することが出来る。ただし、
・Object listの0番目の項は、多面体の数を示す。

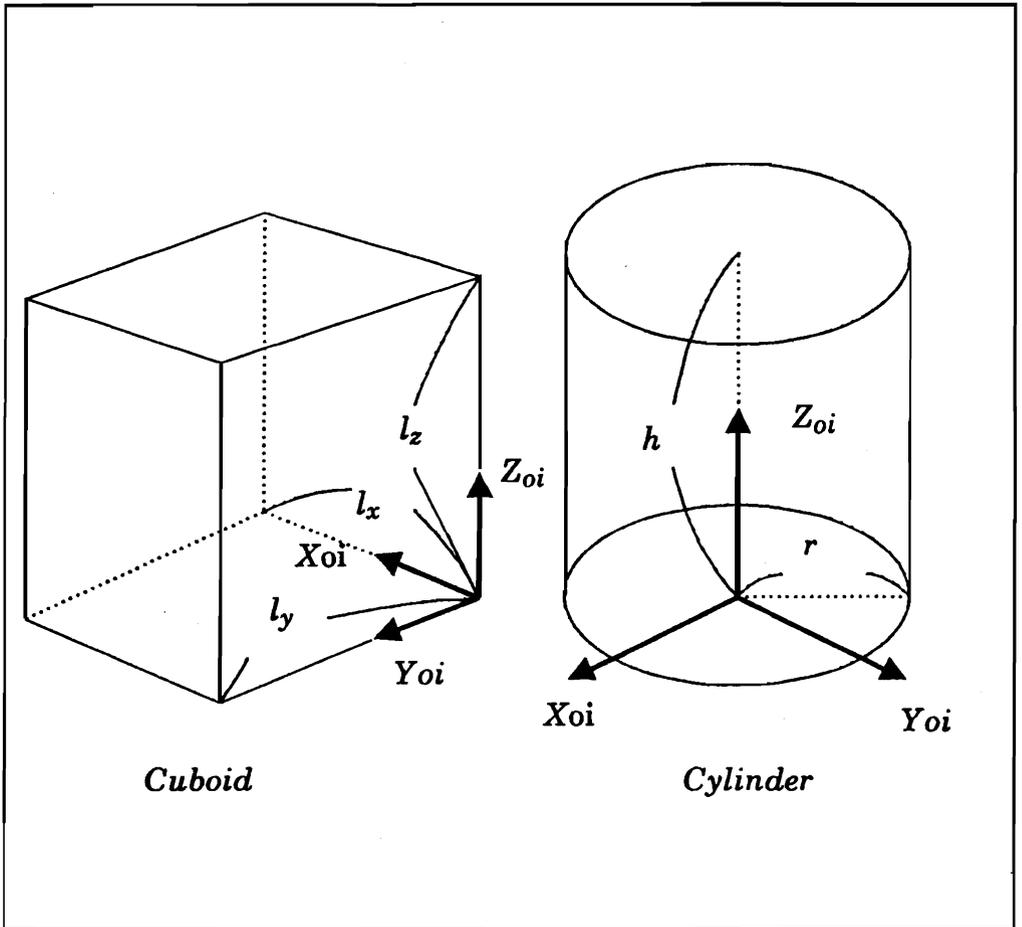


Fig.4-10 The Parameters of Primitives

- ・ Object frame list は、オブジェクト座標系(Fig.4-8 参照)で表された点(X_{oi}, Y_{oi}, Z_{oi})を

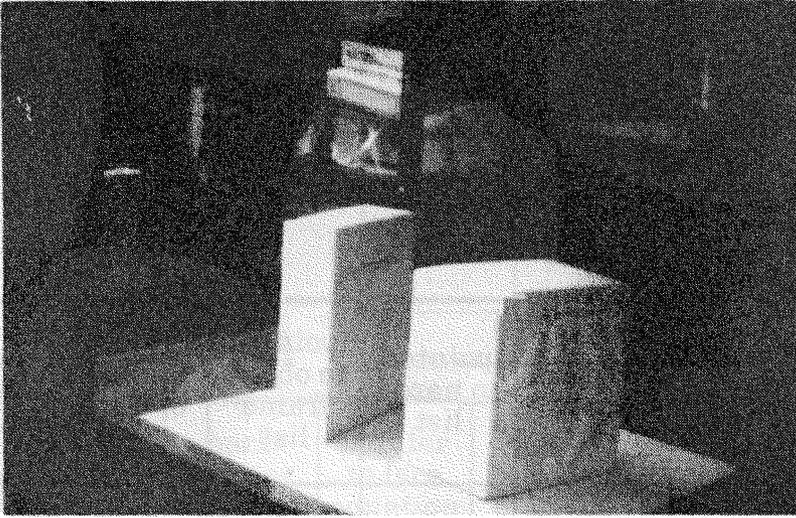


Fig.4-11 The Example of the Teaching Process

	Object list
0	
1	
2	

Object data block	length of name	pointer to name list	pointer to object frame list	number of faces	pointer to face list
1					
2					

	Name list
1	
2	
3	
4	

Object frame list	r_{11}	r_{12}	r_{13}	r_{21}	r_{22}	r_{23}	r_{31}	r_{32}	r_{33}	p_x	p_y	p_z
1												
2												

	Face list
1	
2	
3	
4	

Face equation list	a_{ij}	b_{ij}	c_{ij}	d_{ij}
1				
2				
3				
4				

Face data block	back pointer to object list	pointer to face equation list	number of loops	pointer to loop list
1				
2				
3				
4				

	Loop list
1	
2	
3	
4	
5	

Loop data block	back pointer to face list	number of edges	pointer to edge list
1			
2			
3			
4			
5			

	Edge list
1	
2	
3	
4	
5	
6	
7	

vertex list	X	Y	Z
1			
2			
3			
4			
5			
6			
7			
8			
9			

Edge data block	back pointer to face-1	back pointer to face-2	back pointer to loop	pointer to start vertex	pointer to end vertex
1					
2					
3					
4					
5					
6					
7					

Fig.4-12 B-reps type Data Structure

$$\begin{matrix} X_w & r_{11} & r_{12} & r_{13} & p_x & X_{oi} \\ \begin{bmatrix} Y_w \\ Z_w \end{bmatrix} & \begin{bmatrix} r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} & \begin{bmatrix} p_y \\ p_z \end{bmatrix} & \begin{bmatrix} Y_{oi} \\ Z_{oi} \end{bmatrix} \\ 1 & 0 & 0 & 0 & 1 & 1 \end{matrix} \quad (4-13)$$

とワールド座標系での値に変換する行列の要素である。

- ・Face equation listは、対応するオブジェクト座標系で表した、各境界面の載っている平面の方程式の係数で、(a_{ij},b_{ij},c_{ij})が各オブジェクトの外を向く、大きさ1のベクトルになる様に採る。
- ・Loopは、各境界面を囲むJordan曲線であり、境界面の内側を左に見る様にLoopを回ったときに、右ねじがオブジェクトの外側を向く様に、Loop中の各Edgeの順序と向きを決める。
- ・Vertex list中の各vertexの位置は、対応するオブジェクト座標系で表した値とする。

4.4 オフラインシミュレータ

4.4.1 グラフィックアニメーション

マニピュレータ動作のプランニング結果を視覚的に確認するために、作業環境とその中で動作するマニピュレータをグラフィックディスプレイに表示するシステムを試作した。このための透視変換の様子をFig.4-13に示す。この図において、X_vY_v-平面が投影面であり、Z_v軸が視線の方向である。このとき、X_vY_vZ_v-系で表された点(X_v,Y_v,Z_v)は、X_vY_v-平面上の点(X_G,Y_G)へ

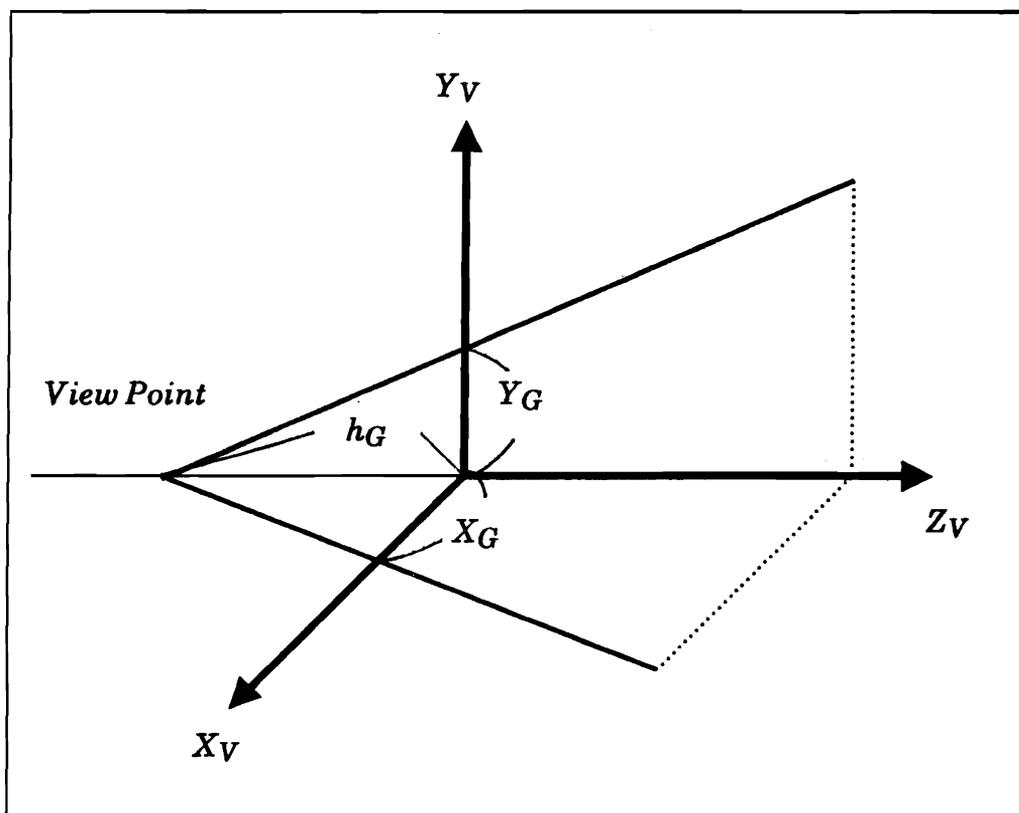


Fig.4-13 Perspective Transformation

$$\begin{bmatrix} s_G \cdot X_G \\ s_G \cdot Y_G \\ s_G \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/h_G & 1 \end{bmatrix} \begin{bmatrix} X_V \\ Y_V \\ Z_V \\ 1 \end{bmatrix} \quad (4-14)$$

と投影される。これをグラフィックディスプレイのスクリーン座標 $X_{SC}Y_{SC}Z_{SC}$ -系へ、

$$\begin{bmatrix} s_{SC} \cdot X_{SC} \\ s_{SC} \cdot Y_{SC} \\ s_{SC} \end{bmatrix} = \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_G \cdot X_G \\ s_G \cdot Y_G \\ s_G \end{bmatrix} \quad (4-15)$$

と変換することにより、透視図を表示することが出来る。また、ブ

ロジエクタの場合と同様にワールド座標系で表された点 (X_w, Y_w, Z_w) は、

$$\begin{bmatrix} X_v \\ Y_v \\ Z_v \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} G_{11} & G_{12} & G_{13} & G_{14} \\ G_{21} & G_{22} & G_{23} & G_{24} \\ G_{31} & G_{32} & G_{33} & G_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (4-16)$$

と $X_v Y_v Z_v$ -系での座標値に変換できる。ただし、ワールド座標系は右手系、 $X_v Y_v Z_v$ -系は左手系なので、(4-16)式は右手系から左手系への変換と、回転・移動変換の積である。(4-14)~(4-16)式から、ワールド座標系での点の位置とスクリーン座標系における位置との関係を得ることが出来る。

本システムにおいては、ワイヤーフレーム表示を用いているので、作業環境教示システムによって得られた作業環境のモデルの中でグラフィック表示に必要なのは、B-repsデータの辺と頂点についての部分のみである。同様に、マニピュレータの表示も各リンクの辺と頂点のデータに基づいて行っている。作業環境とマニピュレータのグラフィック表示例をFig.4-14に示す。この表示例では、透視図の他に、三面図も表示してある。

ここで、フレームメモリ中の画素の位置 (I, J) とスクリーン座標系での位置 (X_{sc}, Y_{sc}) の間の変換、

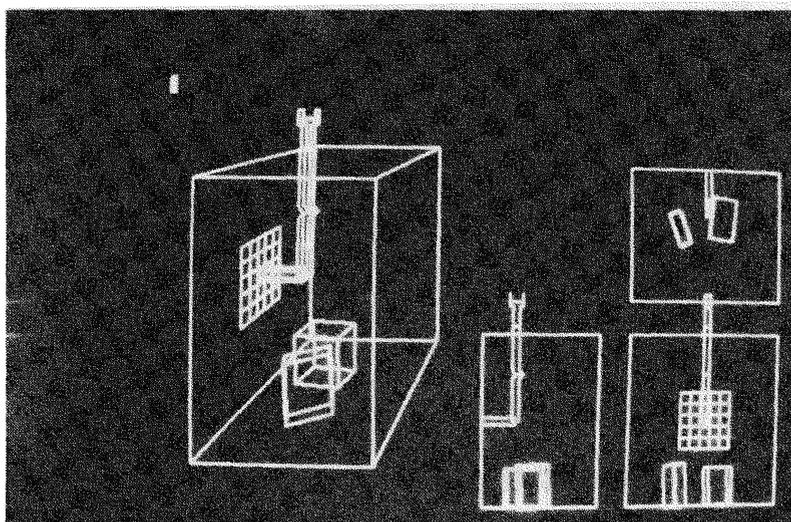


Fig.4-14 An Example of the Perspective View

$$\begin{bmatrix} s_{SC} \cdot X_{SC} \\ s_{SC} \cdot Y_{SC} \\ s_{SC} \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_{IJ} \cdot I \\ s_{IJ} \cdot J \\ s_{IJ} \end{bmatrix} \quad (4-17)$$

を求めておけば、(4-7)式から

$$\begin{bmatrix} s_{SC} \cdot X_{SC} \\ s_{SC} \cdot Y_{SC} \\ s_{SC} \end{bmatrix} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} \\ C_{21} & C_{22} & C_{23} & C_{24} \\ C_{31} & C_{32} & C_{33} & C_{34} \\ & & & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \quad (4-18)$$

と原理的にはカメラの像と同じ透視図を得ることが出来る。従って、これらを同じディスプレイにスーパーインポーズ[Hasegawa 1981]すれば、作業環境が正しく得られているかどうかを確認することが出来る。この例をFig.4-15に示す。

4.4.2 数値計算による干渉チェック

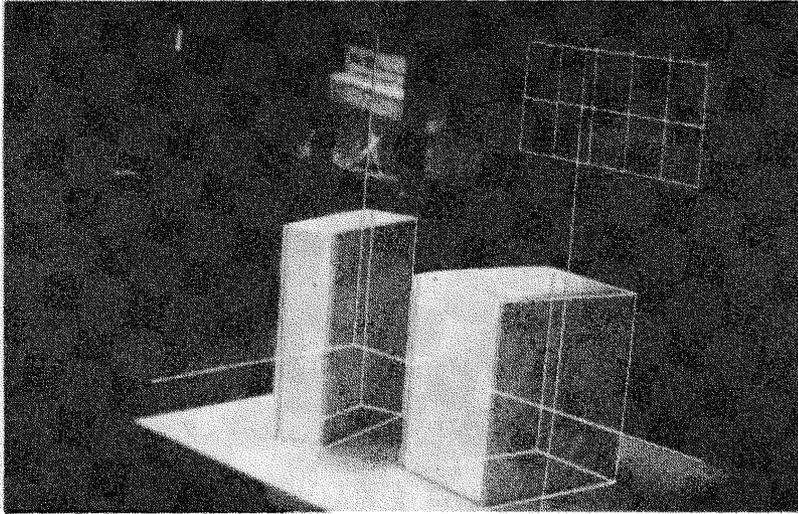


Fig.4-15 The Super-impose View of the Working Environment

グラフィックアニメーションによっても、マニピュレータと作業環境の干渉をある程度チェック出来るが、より正確な検証とプランニングの自動化のためには、数値計算による干渉チェック機能が必要である。本システムでは、干渉のための必要条件のチェックによるフィルタリング[Maruyama 1972]と詳細なチェック[Boyse 1979]を併用した方法により、これをインプリメントした。このアルゴリズムの実行には、作業環境とマニピュレータのB-repsモデルのデータの全てが必要である。尚、現在のシステムでは、直方体どうしの干渉チェックを1回行うのに、フィルタリングにかからなかった場合は、約450 [msec]を要する。

4.5 制御システム

4.5.1 軌道計画

カイネマティクスのみを考慮して計画された目標軌道を実機で実行するためには、実機の動特性を考慮して、これを時間の関数に変換しておく必要がある。本システムでは、以下に示す方法を採用した。

軌道データは、マニピュレータの関節空間において、

$$\theta_i^{(r)}: i=1, \dots, N \quad (N \text{はマニピュレータの自由度})$$

$$r=0, \dots, R \quad (r \text{はイテレーション回数})$$

と与えられているとする。各イテレーション毎のこの軌道の長さを次式で定義する。

$$l(r) = \max_{i=1}^N |\theta_i^{(r)} - \theta_i^{(r-1)}| \quad (4-19)$$

また、この累積和を、

$$L(r) = \sum_{j=1}^r \max_{i=1}^N |\theta_i^{(j)} - \theta_i^{(j-1)}| \quad (4-20)$$

とおいておく。サーボ系は、デジタル制御で実現することとし、そのサンプリング周期を T としたとき、速度の制限から各関節が単位周期に変位しうる最大値が $\Delta\theta_{\max}$ であると仮定する。また、加速度に対する制限から、この速度に達するのに必要な時間が NT であると仮定する。これらの値は、実験により概略値を求めておく。

まず、 r 回目のイテレーションデータに対する単位周期内の変位の最大値を次の様に定める。

(1) $L(R) \leq \Delta\theta_{\max} \cdot N$ のとき

$$\Delta\theta_{\max}(r) = \frac{k}{N} \Delta\theta_{\max} \quad (0 < k \leq M)$$

$$\Delta\theta_{\max}(r) = \frac{2M-k}{N} \Delta\theta_{\max} \quad (M < k < 2M)$$

(4-21)

ここに、 k は k 回目のサンプリングであることを示し、 M は、

$$M = \text{int}\left(\left(\frac{N \cdot L(R)}{\Delta\theta_{\max}}\right)^{1/2}\right) \quad (4-22)$$

である。

(2) $L(R) > \Delta\theta_{\max} \cdot N$ のとき

$$\Delta\theta_{\max}(r) = \frac{k}{N} \Delta\theta_{\max} \quad (0 < k \leq N)$$

$$\Delta\theta_{\max}(r) = \Delta\theta_{\max} \quad (N < k \leq N+M)$$

$$\Delta\theta_{\max}(r) = \frac{2N-M-k}{N} \Delta\theta_{\max} \quad (N+M < k < 2N+M)$$

(4-23)

ここに、

$$M = \frac{L - \Delta\theta_{\max} \cdot N}{\Delta\theta_{\max}} \quad (4-24)$$

である。(4-23)式を図示すると、Fig.4-16に示す様になる。

この加減速パターンに基づいて、 $\theta_i^{(r)}$ を時間の関数に変換する。

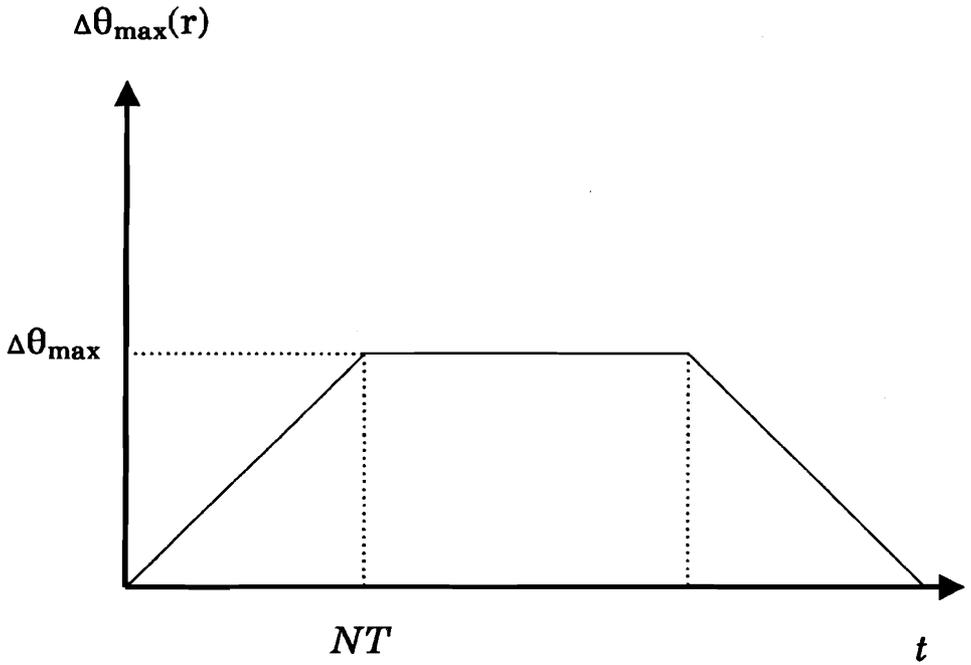


Fig.4-16 The Acceleration and Deceleration Pattern of the Trajectory in the Joint Space

以下、 $\theta_i(kT)$ を単に $\theta_i(k)$ とかくことにする。まず、 $k=0$ に対しては、

$$\theta_i(0) = \theta_i(0) \quad (4-25)$$

とする。次に、 $k \geq 1$ に対しては、次の軌道データのイテレーション数を r として、

$$H(r) = \frac{\sum_{j=r}^n \max_{i=1}^N |\theta_i^{(r)} - \theta_i^{(r-1)}|}{\Delta \theta_{\max}(r)} \quad (4-26)$$

という値が、

$$H(r) \geq 1 \quad (4-27)$$

なる条件を満たすまで $n(\geq r)$ をインクリメントしていき、

$$\theta_i(k+m) = \theta_i(k-1) + \frac{m+1}{H(r)} (\theta_i^{(n)} - \theta_i^{(r)}) \quad (4-28)$$

$$(i=1, \dots, N)$$

という線形補間により次のサンプリング値を決める。ただし、

$$0 \leq m \leq \text{int}(H(r)) - 1 \quad (4-29)$$

である。以上の方法により、時間の関数である目標軌道を得ることが出来る。この方法の適用例を Fig.4-17 に示す。

4.5.2 制御アルゴリズム

一般に、マニピュレータの運動方程式は次の様にかける。

$$J(\Theta) \frac{d^2\Theta}{dt^2} + f(\Theta, \frac{d\Theta}{dt}) + g(\Theta) = \tau \quad (4-30)$$

ここに、 N をマニピュレータの自由度とするとき、

Θ : マニピュレータの一般化座標 ($N \times 1$)

$J(\Theta)$: 慣性テンソル ($N \times N$)

$f(\Theta, \dot{\Theta})$: 遠心力・コリオリ力 ($N \times 1$)

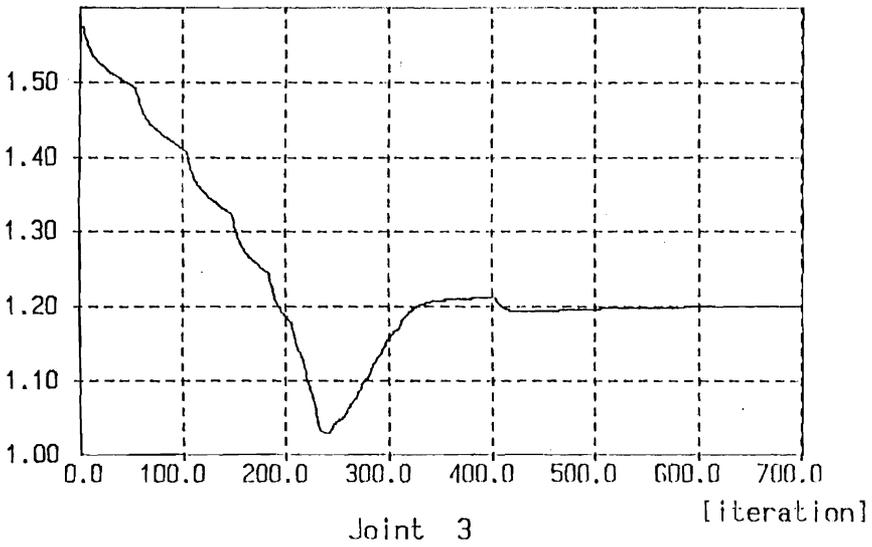
$g(\Theta)$: 重力 ($N \times 1$)

τ : 一般化力 ($N \times 1$)

である。 $J(\Theta)$ 、 $f(\Theta, \dot{\Theta})$ 、 $g(\Theta)$ は、一般に三角関数などを含む Θ の非線形関数になる。また、 $J(\Theta)$ は一般に非対角行列になるので、各自由度間には強い干渉が存在する。しかしながら、アクチュエータか

Original Trajectory

[rad]



Planned Trajectory

[rad]

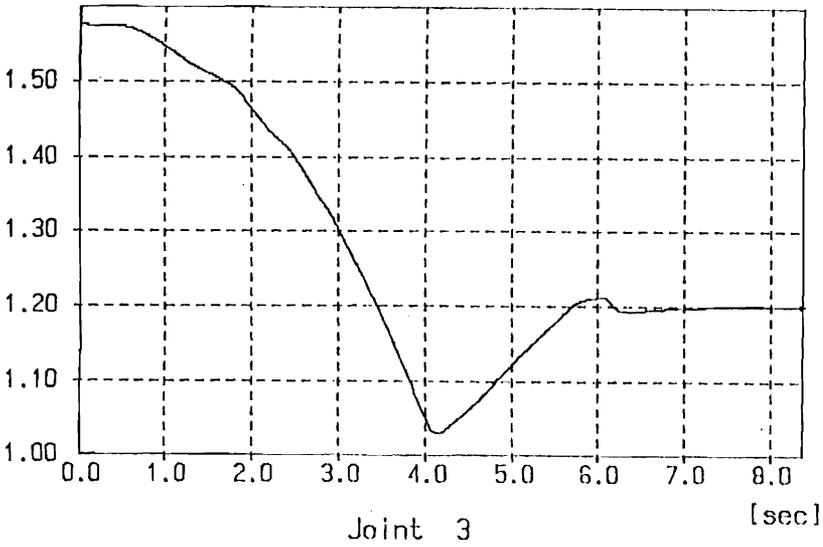


Fig.4-17 An Example of the Trajectory Planning

ら減速機を介してマニピュレータをドライブする場合、モータの慣性も考慮に入れて、(4-30)式は次の様書き換えられる。

$$(J_{Mi} + \frac{J_{ii}(\Theta)}{R_i^2}) \frac{d^2\theta_{Mi}}{dt^2} + B_i \frac{d\theta_{Mi}}{dt} = \tau_{Mi} - \frac{1}{R_i} h_i(\Theta, \frac{d\Theta}{dt}, \frac{d^2\Theta}{dt^2}) \quad (4-31)$$

ここに、

$\theta_{Mi} = R_i \theta_i$:モータの回転角

$\tau_{Mi} = \tau_i / R_i$:モータのトルク

J_{Mi} :i番目の関節のモータの慣性モーメント

B_i :モータの粘性摩擦係数

$R_i (\geq 1)$:i番目の関節の減速比

$J_{ij}(\Theta)$: $J(\Theta)$ のij要素

$h_i(\Theta, \dot{\Theta}, \ddot{\Theta})$:遠心力・コリオリ力、重力、慣性力から成る外乱項で、

次の式で表せられる。

$$h_i(\Theta, \frac{d\Theta}{dt}, \frac{d^2\Theta}{dt^2}) = f_i(\Theta, \frac{d\Theta}{dt}) + g_i(\Theta) + \sum_{j=1, j \neq i}^N J_{ij}(\Theta) \frac{d^2\theta_{Mj}/dt^2}{R_j} \quad (4-32)$$

尚、モータは電流制御するものとして、逆起電力の項を無視している。 (4-31),(4-32)式からマニピュレータの力学系の慣性モーメントによる影響は、モータから見ると減速比の2乗分の1になることが分かる。また、遠心力・コリオリ力の項は、

$$f_i(\Theta, \frac{d\Theta}{dt}) = \frac{1}{2} \left\{ \sum_{j=1}^N \frac{d\theta_{Mj}/dt}{R_j} \frac{\partial R(\Theta)}{\partial \theta_j} \right\} \frac{d\theta_{Mi}/dt}{R_i} \quad (4-33)$$

と(4-31)式の係数と合わせてギア比の3乗の分の1になる。本システ

ムで制御の対象としたマニピュレータの各関節のギア比は、110～2800/3 と非常に高いため、これらの項はほとんどアクチュエータの動特性に影響を及ぼさない。重力項をTaylor展開したときの0次の項のみが減速比分の1にしかならないが、開発したマニピュレータのフレームはCFRPで出来ており非常に軽量なので、この項も大きな影響はないと考えられる。そこで、本システムでは、PTP制御の場合にはその安定性が理論的に証明されている[Miyazaki 1985b]各関節独立のPIDフィードバック則により制御系を構成した。これは、次の様にかける。

$$\tau_i = k_{Pi}(\bar{\theta}_i - \theta_i) - k_{Di}\dot{\theta}_i + k_{Ii} \int (\bar{\theta}_i - \theta_i) dt \quad (4-34)$$

$$(i=1,,N)$$

実際、この制御則により、十分良好な制御結果を示した。実験結果の例をFig.4-18に示す。この図では、目標軌道を一点鎖線、実機の応答を実線で表してあるが、ほとんど両者は一致している。

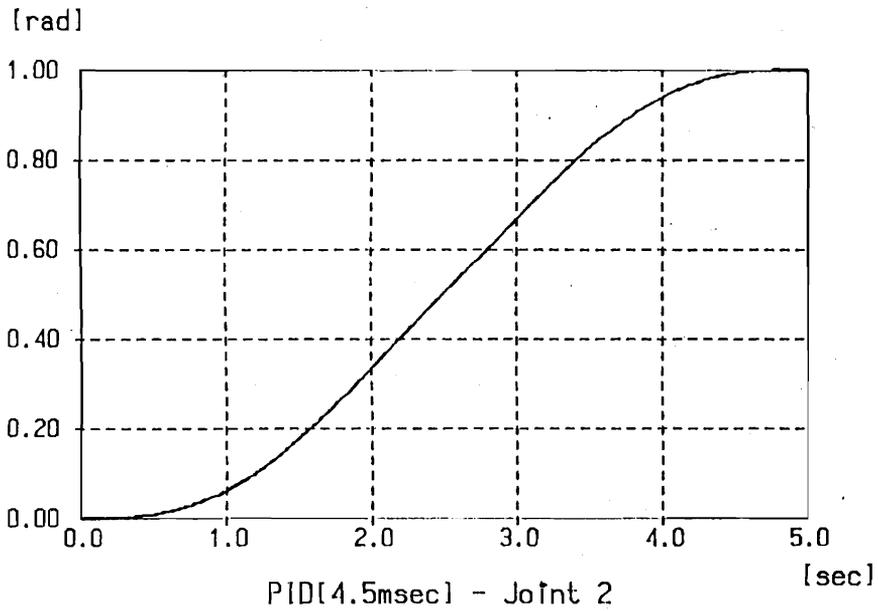


Fig.4-18 An Example of the Responce of Manipulator

4.6 システムの運用

本システムの有効性を確認するために、ここでは自由動作の自律的計画に適用した例を示す。

まず、Fig.4-19の様な作業環境を設定した。この2つの障害物の形

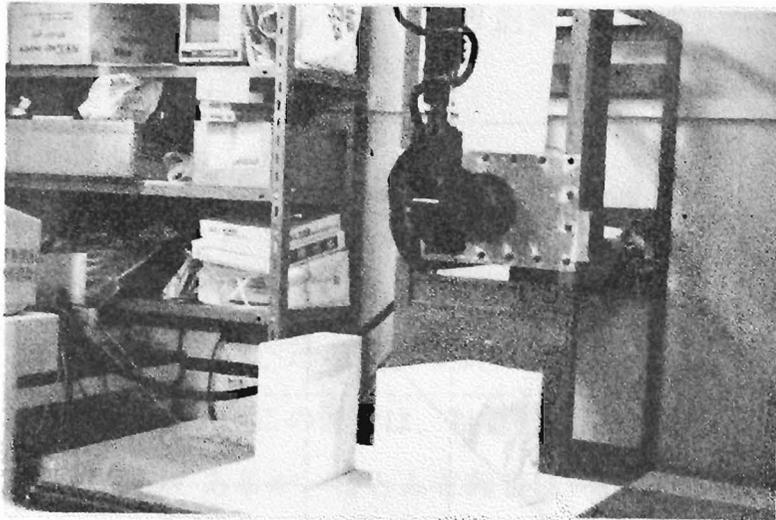


Fig.4-19 A Working Environment

状・大きさ・位置・姿勢を前述した作業環境教示システムによって入力し、得られた内部モデルのワイアフレーム表示とカメラの像とをスーパーインポーズしたのが、Fig.4-20である。そして、この作業環境のモデルに基づいて、2章で述べたアルゴリズムにより自由動作をオフラインシミュレータ上で計画した結果がFig.4-21である。最終的に、この動作を実機で実行したのがFig.4-22である。この様

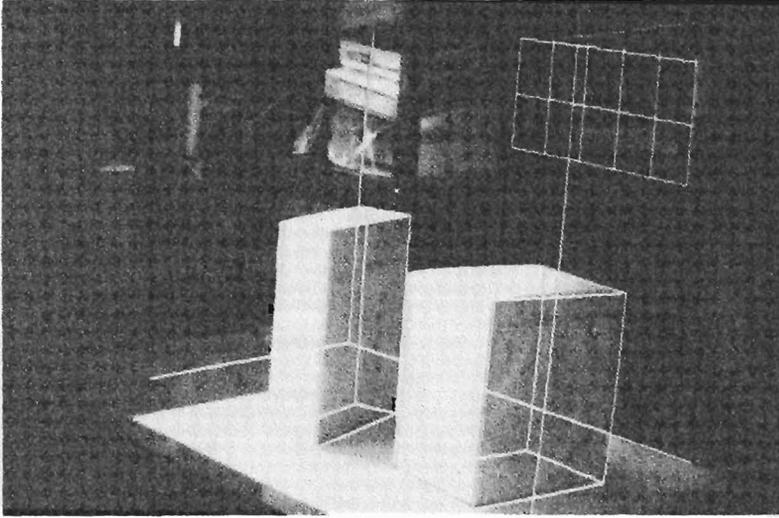


Fig.4-20 A Super-impose View of the Actual and Obtained Working Environment

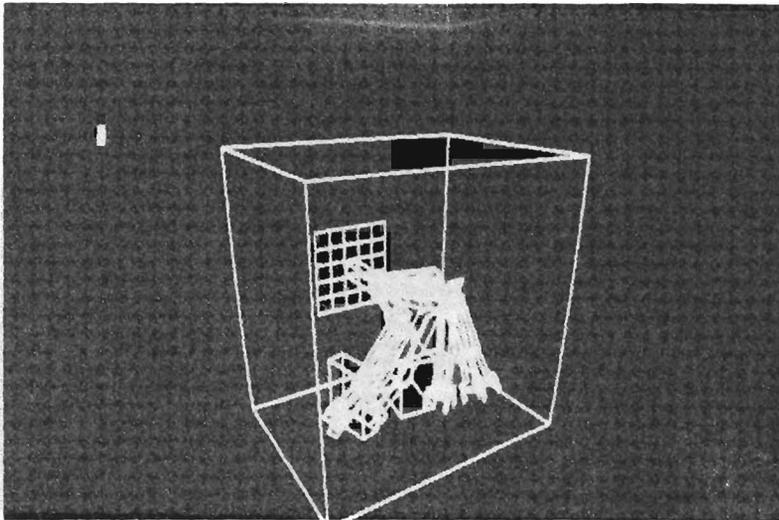


Fig.4-21 A Planned Motion on the Off-line Simulator

に、本システムによる自律的な動作計画が可能であることが確認さ

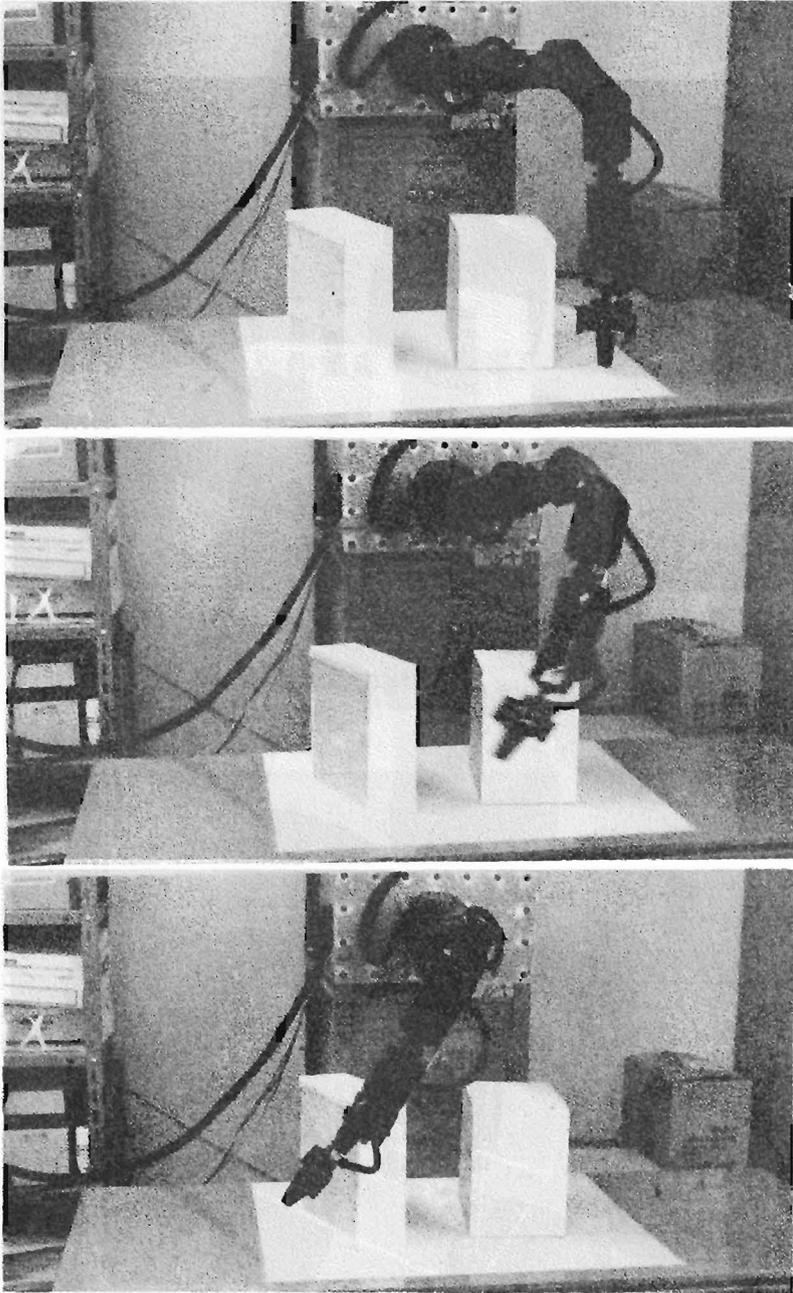


Fig.4-22 An Actual Motion of Manipulator

れた。

同様に、2つの直方体の間にマニピュレータのエンドイフェクタを挿入する動作を計画した例をFig.4-23~Fig.4-25に示す。

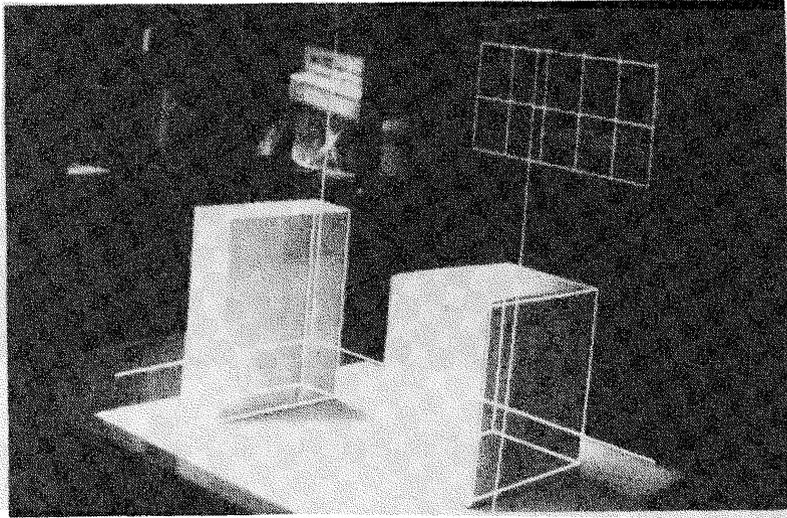


Fig.4-23 A Super-impose View of the Actual and Obtained Working Environment

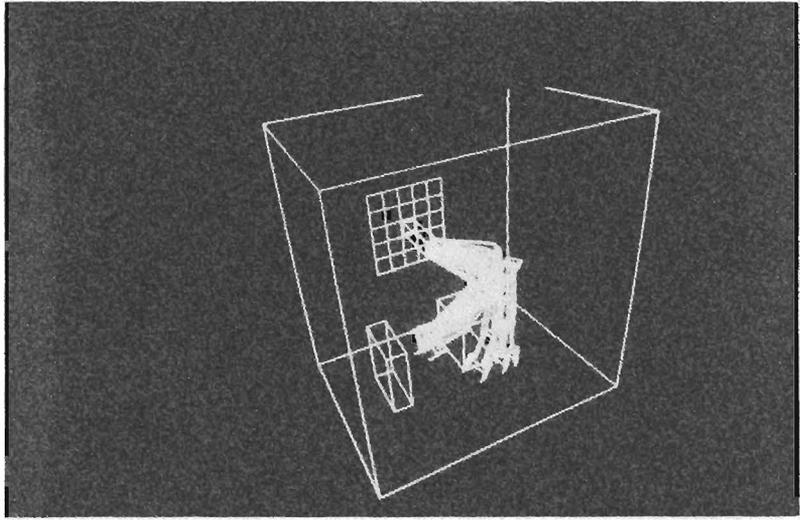


Fig.4-24 A Planned Motion on the Off-line Simulator

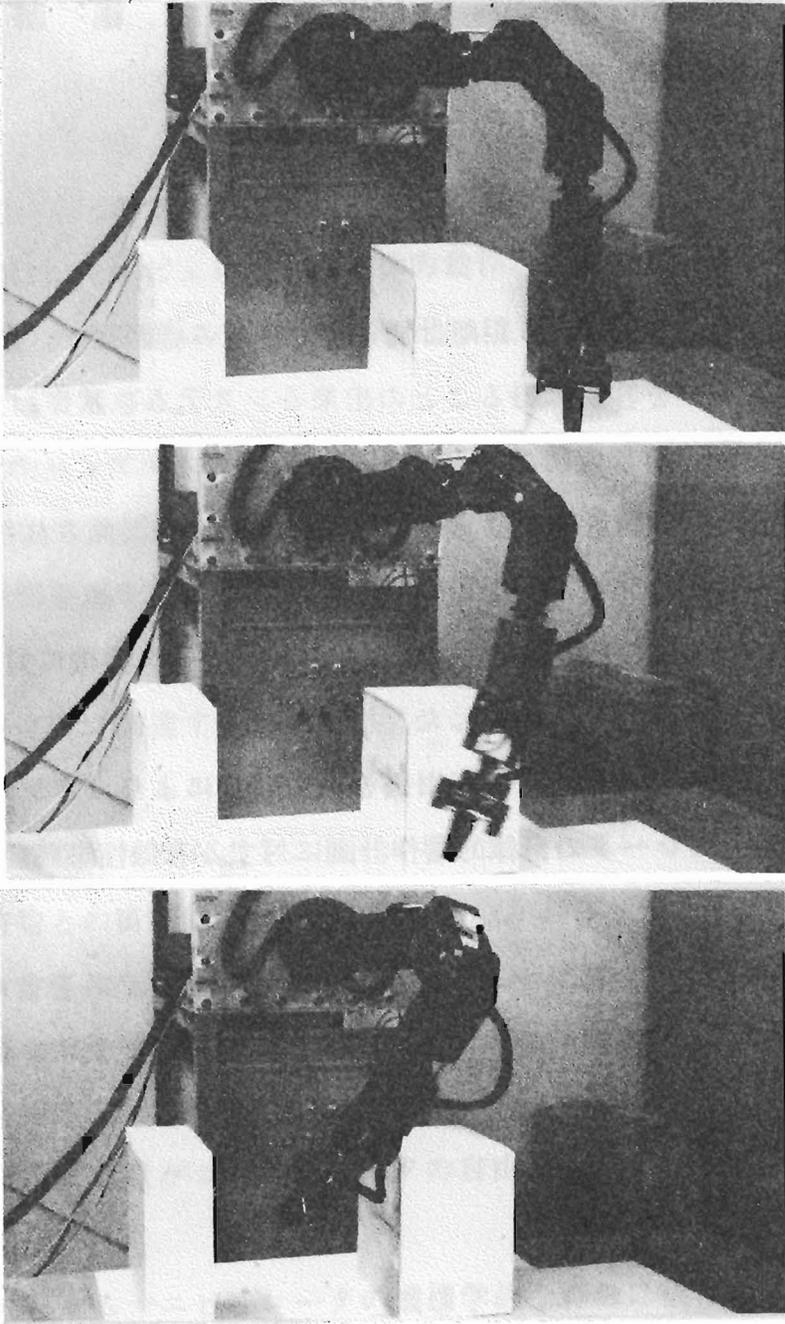


Fig.4-25 An Actual Motion of Manipulator

4.7 まとめ

本章で得られた結果をまとめると、次の様になる。

- (1) 空間コード化による距離計測と対話を組み合わせて、作業環境のモデルを容易に得ることの出来るシステムを試作した。
- (2) 得られた作業環境のモデルに基づいて、グラフィックアニメーション、数値計算による干渉チェックにより計画された動作を検証可能なオフラインシミュレーションシステムを開発した。
- (3) 各関節独立のPID制御則によりマニピュレータ実機に対する制御系をインプリメントした。
- (4) システム全体を用いた運用例を示すことにより、本システムのマニピュレータの自律的動作計画に対する有効性を確認した。

今後の課題としては、マニピュレータのダイナミクスを含めたシミュレーション機能の付加、ユーザーインターフェクスの充実が挙げられる。

第5章 結 論

本研究は、マニピュレータの自律的動作計画を実現するために必要不可欠の問題について検討を行ったものである。得られた結果を以下にまとめる。

第2章では、マニピュレータの自由動作を自律的に計画することを目的として、新しい障害物回避アルゴリズムを提案した。本アルゴリズムの特徴を次に示す。まず、一般のポテンシャル法と比較すると、

- (1) 目標位置に到達するためのポテンシャルと障害物を回避するためのポテンシャルとの相互干渉が出来るだけ小さくなるように大局的な見地からポテンシャル場を定義しているので、停留や障害物との衝突による解の探索の失敗が起こりにくい。
- (2) エンドイフェクタの目標軌道を大局的に計画しているので、これに基づく解の探索の再試行が可能である。

また、自由空間法と比較すると、

- (3) 必要な計算量がマニピュレータの自由度数に対して線形にしか増加しない。
- (4) 対象とするマニピュレータの機構学的な特徴に依存せずに適用可能である。

この2つの特徴により、本手法は冗長自由度を持つような一般の多自由度マニピュレータに適用可能である。

第3章では、自律的動作計画に必要な演算処理が膨大であるという問題に対する一つの対策を与えるために行った。すなわち、動作計画の結果を学習することにより、同じ動作あるいは似た動作を再び計画することなしに実行することを可能にし、これによって必要な演算処理を少なくするという方法について、基礎的な検討を行った。得られた結果を以下にまとめる。

- (1) 適応プロダクションシステムの特徴とその動作学習過程記述に対する適合性について論じた。
- (2) 適応プロダクションシステムに基づいて獲得知識を表現・利用する方法について示し、その際、弁別ネットが効率の良い知識の想起に対して有効に働くことを述べた。
- (3) 獲得知識の一般化と特化について述べ、これらによって獲得知識の有効範囲が学習されることを示した。
- (4) 動作学習過程全体について考え、ここで考えたモデリングにおいては、学習前の動作は演算処理に基づく閉ループ制御であったのが、学習後は経験によって獲得した知識の想起に基づく開ループ制御になることを述べた。
- (5) 適応プロダクションシステムをインプリメントし、学習過程のシミュレーションを行った。

第4章では、実際にマニピュレータの自律的動作計画を行うための全ての要素を備えたシステムを開発した結果について述べた。得られた結果をまとめると次の様になる。

- (1) 空間コード化による距離計測と対話を組み合わせて、作業環境のモデルを容易に得ることの出来るシステムを試作した。
- (2) 得られた作業環境のモデルに基づいて、グラフィックアニメーション、数値計算による干渉チェックにより計画された動作を検証可能なオフラインシミュレーションシステムを開発した。
- (3) 各関節独立のPID制御則によりマニピュレータ実機に対する制御系をインプリメントした。
- (4) システム全体を用いた運用例を示すことにより、本システムのマニピュレータの自律的動作計画に対する有効性を確認した。

本研究は、自由動作の自律的計画に対しては一応の成果を挙げたと考えて良いと思う。今後は、把持動作やペグ挿入動作など、マニピュレータに求められる多種多様な動作の自律的計画アルゴリズムとそのアルゴリズムの実行に必要な諸要素を開発することが望まれる。ロボットの真の知能化は、人工知能の分野で発展してきた技術の安易な適用によってではなく、こうしたロボット固有の問題を一つ一つ解決していくことにより実現されると筆者は考えている。

謝 辞

本研究は、筆者が神戸大学大学院自然科学研究科システム科学専攻博士課程に在学中に行ったものである。研究を進めるに際して、指導教官として細部にわたり御指導を賜った北村新三教授に深甚なる謝意を表します。また、研究前半における指導教官として御指導して頂いた黒田一之神戸大学名誉教授に深く感謝します。

さらに、履修指導委員として御指導、御助言頂いた松本治弥教授、平井一正教授、瀬口靖幸教授(現大阪大学教授)に深く感謝します。また、講義などを通じて御指導、御助言頂いた高森年教授、池田雅夫助教授、木村一郎助教授に深く感謝します。

また、日頃から討論の相手になって頂き、的確な御助言を賜った田中克己助教授、土屋嘉章助手、呉松保男技官を始めとする工学部計測工学科自動制御研究室の諸氏、堤一義氏を始めとするシステム科学専攻情報・計測研究室の諸氏に感謝します。特に、本研究を進めるにあたり、何かと御協力頂いた人見信君、榊原聡君、神邑剛君、角谷和俊君、上田章君に感謝します。

最後に、筆者がロボティクスの研究に携わるに際し、多くの貴重な御示唆、御助言を受けた関西(若手)ロボット研究会のメンバーに深い感謝の意を表します。

参考文献

1. [Anzai 1974] 安西祐一郎;プロダクションシステムによる学習・適応機能のモデル、計測と制御、vol.18,no.4,pp.303-311,1974.
2. [Arimoto 1984] Arimoto,S.,Kawamura,S.,and Miyazaki,F; Bettering Operation of Robots by Learning, J.of Robotic Systems, vol.1,no.2, pp.123-140, 1984.
3. [Baer 1979] Baer,A., Eastman,C., and Henrion,M.; Geometric Modelling:a Survey, Computer-aided Design, vol.11, no.5, pp.253-272, 1979.
4. [Boyse 1979] Boyse,J.W.; Interference Detection Among Solids and Surfaces, Comm. ACM,vol.22,no.1,pp.3-9, 1979.
5. [Brooks 1983] Brooks,R.A.; Planning Collision-Free Motions for Pick-and-Place Operations, Int. Journal of Robotics Research, vol.2, no.4, pp.19-40,1983.
6. [Denavit 1955] Denavit,J.,Hartenberg,R.S.; A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices, J. of Applied Mechanics, pp.215-221, 1955.
7. [Fahlman 1974] Fahlman,S.E.; A Planning system for Robot Construction Tasks, Artificial Intelligence, vol.5,no.1, pp.1-49,1974.

8. [Fikes 1971] Fikes,R.E. and Nilsson,N.J.; STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving, Artificial Intelligence, vol.2, no.3/4, pp.189-208, 1971.
9. [Furukawa 1979] 古川正志、他; object modelingにおける境界評価関数の研究、精密機械、 vol.45, no.6, pp.1036-1042, 1979.
- 10.[Goldstein 1980] Goldstein,H.; Classical Mechanics, Addison-Wesley Pub., Massachusetts, 1980.
- 11.[Grossman 1976] Grossman,D.D.; Procedural Representation of Three-dimensional Objects, IBM J. Research and Development, pp.582-589, 1976.
- 12.[Grossman 1978] Grossman,D.D. and Taylor,R.H.; Interactive Generation of Object Models with a Manipulator, IEEE Trans. Sys. Man and Cyber., vol.SMC-8, no.9, pp.667-679, 1978.
- 13.[Hanabusa 1983] 花房秀郎、吉川恒夫、中村仁彦; 関節型ロボットアームの冗長性の解析とその優先順位を有する作業への応用、計測自動制御学会論文集、 vol.19, no.5, pp.421-426, 1983.
- 14.[Hart 1968] Hart,P.E., Nilsson,N.J. and Raphael,B.; A Formal Basis for the Heuristic Determination of Minimum Cost Paths, IEEE Trans. on Sys. Sci. Cyber., vol.ssc-4, no.2, pp.100-107, 1968.
- 15.[Hasegawa 1981] 長谷川勉; ロボットの作業環境のモデリングとモニタリング、計測自動制御学会論文集、 vol.17, no.5, pp.589-595, 1981.

- 16.[Hasegawa 1986] 長谷川勉;自由空間分類表現法によるマニピュレータの衝突回避動作の計画、計測自動制御学会論文集、vol.22, no.6, pp.616-622、1986.
- 17.[Ichikawa 1982] 市川惇信、小林重信;事象駆動型システムの表現と制御、計測と制御、vol.21,no.10,pp.929-938,1982.
- 18.[Ichikawa 1983] 市川惇信;適応プロダクションシステムによる汎用制御系、計測自動制御学会論文集、vol.19,no.1,pp.88-90,1983.
- 19.[Jarvis 1983] Jarvis,R.A.; A Perspective on Range Finding Techniques for Computer Vision, IEEE Trans. Pattern Analysis and Machine Intelligence, vol.PAMI-5, no.2, pp.122-139, 1983.
- 20.[Khatib 1978] Khatib,O.,Maitre,J.F.; Dynamic Control of Manipulators Operating in a Complex Environment, 3rd CISM-IFTToMM Symp., pp.267-282, 1978.
- 21.[Kodama 1978] 児玉慎三、須田信英;システム制御のためのマトリクス理論、計測自動制御学会、1978.
- 22.[Kokaji 1985] 小鍛冶繁;並列制御機構によるマニピュレータの制御(第2報)—実時間障害物回避制御—、精密機械、vol.51, no.12, pp.2328-2333, 1985.
- 23.[Laugier 1981] Laugier,C.; A Program for Automatic Grasping of Objects with a Robot Arm, 11th Int'l Symp. on Industrial Robots, October, 1981.
- 24.[Lieberman 1977] Lieberman,L.I. and Wesley,M.A.; AUTOPASS: an Automatic Programming System for Computer

- Controlled Mechanical Assembly, IBM J. Research and Development, vol.21, no.4, pp.321-333, 1977.
- 25.[Loeff 1975] Loeff,L.A. and Soni,A.H.; An Algorithm for Computer Guidance of a Manipulator in Between Obstacles, Trans. of ASME, Journal of Engineering for Industry,vol.97-3, pp.836-842, 1975.
- 26.[Lozano-Perez 1977] Lozano-Perez,T. and Winston,P.H.; LAMA: a Language for Automatic Mechanical Assembly, 5th Int'l Joint Conference Artificial Intelligence, pp.79-107, MIT, Cambridge, MA, August, 1977.
- 27.[Lozano-Perez 1979] Lozano-Perez,T.; An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles, Com. of the ACM, vol.22, no.10, pp.560-570, 1979.
- 28.[Lozano-Perez 1981] Lozano-Perez,T.; Automatic Planning of Manipulator Transfer Movements, IEEE Trans. Sys. Man Cyb., vol.SMC-11, no.10, pp.681-689, 1981.
- 29.[Lozano-Perez 1982] Lozano-Perez,T.; in Robot Motion, pp.473-498, MIT Press, 1982.
- 30.[Maruyama 1972] Maruyama,K.; A Procedure to Determine Intersections between Polyhedral Objects, Int'l J. Computer and Information Sciences, vol.1,no.3, pp.255-266,1972.

- 31.[Matubara 1985] 松原仁、岡野彰、井上博;作業目標レベルのロボット言語の設計と試作、日本ロボット学会誌、vol.3, no.3, pp.220-228, 1985.
- 32.[Matuya 1985] 松家英雄(編)、ロボットプログラミング特集、日本ロボット学会誌、vol.3, no.2, pp.107-157, 1985.
- 33.[Mimori 1982] 三森幸治、嘉数昇、沖野教郎;ソフトウェアド・ロボットの基礎研究—3次元空間内の物体移動問題、精機学会春季大会学術講演会論文集、pp.575-577, 1982.
- 34.[Miyazaki 1985a] 宮崎文夫他;人工ポテンシャルに基づいたロボットマニピュレータのセンサフィードバック制御、計測自動制御学会論文集、vol.21, no.1, pp.71-77, 1985.
- 35.[Miyazaki 1985b] 宮崎文夫、有本卓;ロボットマニピュレータのためのフィードバック制御則の安定性とロバスト性に関する考察、計測自動制御学会論文集、vol.21, no.1, pp.78-83, 1985.
- 36.[Mujtaba 1981] Mujtaba,S. and Goldman,R.; AL User's Manual, Stanford AI Lab., Memo AIM-344, Dec.,1981.
- 37.[Okino 1982] 沖野教郎;自動設計の方法論、養賢堂、東京、1982.
- 38.[Okutomi 1983] 奥富正敏、森政弘;ポテンシャル場を用いたロボットの動作決定、日本ロボット学会誌、vol.1, no.3, pp.226-232, 1983.
- 39.[Ozaki 1982] 尾崎弘明、毛利彰、高田勝;マニピュレータの占有空間を考慮した障害物回避動作の決定法、計測自動制御学会論文集、vol.18, no.9, pp.942-949, 1982.

- 40.[Paul 1981] Paul,R.P.; ROBOT MANIPULATORS: MATHEMATICS,PROGRAMMING, AND CONTROL, The MIT Press,Cambridge,1981.
- 41.[Poplestone 1980] Poplestone,R.J., Ambler,A.P., and Bellos,I.M.;An Interpreter for a Language for Describing Assemblies,Artificial Intelligence, vol.14, no.1, pp.79-107,1980.
- 42.[Post 1943] Post,E.L.; Formal Reduction of the General Combinatorial Decision Problem, American J. of Mathematics, vol.65,pp.197-215,1943.
- 43.[Requicha 1980] Requicha,A.A.G.; Representations for Rigid Solids: Theory, Methods, and Systems, Computing Surveys, vol.12, no.4, pp.437-464, 1980.
- 44.[Requicha 1985] Requicha,A.A. and Voelcher,H.B.; Boolean Operation in Solid Modeling: Boundary Evaluation and Merging Algorithm, Proceedings of the IEEE, vol.73, no.1,pp.30-44, 1985.
- 45.[Sacerdoti 1974] Sacerdoti,E.D.; Planning in a Hierarchy of Abstraction Spaces, Artificial Intelligence,vol.5, no.2,pp.115-135, 1974.
- 46.[Sato 1985] 佐藤宏介、井口征士; 空間コード化による距離画像入力、電子通信学会論文誌、vol.J68-D,no.3,pp.369-375, 1985.
- 47.[Sekiguchi 1985] 関口博、小島俊雄、井上久仁子、本多庸悟;回転機能部品の部品展開手法に関する研究、精密機械、vol.51,no.2, pp.359-365,1985.

- 48.[Shimizu 1970] 清水清孝;制約条件のないシステムの最適化手法、計測と制御、 vol.9,no.6, pp.449-459,1970.
- 49.[Shirai 1982] 白井良明;ロボットの物体認識技術、計測と制御、 vol.21, no.12, pp.1115-1121, 1982.
- 50.[Takegaki 1981] 竹垣盛一、有本卓;マニピュレータの作業座標フィードバック制御、計測自動制御学会論文集、 vol.17,no.5, pp.582-588, 1981.
- 51.[Takase 1981] Takase,K.,Paul,R.P.,and Berg,E.J.; A Structured Approach to Robot Programming and Teaching, IEEE Trans. Sys. Man Cyber., vol.SMC-11, no.4, pp.274-289,1981.
- 52.[Tivole 1980] Tilove,R.B.; Set Membership Classification: A Unified Approach to Geometric Intersection Problems, IEEE Trans.Computers, vol.C-29, no.10, pp.874-883, 1980.
- 53.[Tujii 1979] 辻井潤一;プロダクションとその応用、情報処理、 vol.20, no.8, pp.735-743,1979.
- 54.[Udupa 1977] Udupa,S.M.; Collision Detection and Avoidance in Computer Controlled Manipulators, 5th Int.Joint Conf. on Artificial Intelligence, pp.737-748, 1977.
- 55.[Vere 1977] Vere,S.A.; Relational Production Systems, Artificial Intelligence, vol.8, pp.47-68, 1977.
- 56.[Vere 1978] Vere,S.A.; Inductive Learning of Relational Productions, in PATTERN-DIRECTED INFERENCE SYSTEMS

(Waterman,D.A.,Hayes-Roth,F. eds.), pp.281-295, Academic Press, 1978.

- 57.[Wesley 1980] Wesley,M.A., Lozano-Perez,T., Lieberman,L.I., Lavin,M.A., and Grossman,D.D.; A Geometric Modeling System for Automated Mechanical Assembly, IBM J. Research and Development, vol.24, no.1, pp.64-74, 1980.
- 58.[Wolter 1985] Wolter,J.D., Volz,R.A. and Woo,A.C.; Automatic Generation of Gripping Positions, IEEE Trans. Sys. Man Cyb.,vol.SMC-15, no.2, pp.204-213, 1985.

著者文献

学術論文等

1. [Hirukawa 1986a] Hirukawa,H., Kitamura,S.: A Collision Avoidance Algorithm for Robot Manipulators using the Potential Method and Safety First Graph, Proc. of JAPAN-USA Symposium on Flexible Automation, pp.99-102, JAACE-ASME, Osaka, July,1986.
2. [Hirukawa 1986b] 比留川 博久、北村 新三、適応プロダクション・システムに基づくマニピュレータ動作学習過程のモデリング、バイオメカニズム8、pp.67-76、東京大学出版会、1986。
3. [Hirukawa 1986c] Hirukawa,H., Kitamura,S.: A Knowledge-Based Autonomous Robot Manipulation System, Proc. of International Symposium on Robot Manipulators: Modeling, Control and Education, pp.189-198 Albuquerque, Nov.,1986, (Jamshidi,M., Luh,J.Y.S.,and Shahinpoor,M. Eds. ; RECENT TRENDS IN ROBOTICS, North-Holland).
4. [Hirukawa 1986d] 比留川 博久、北村 新三、安全第一アルゴリズムとポテンシャル関数に基づくマニピュレータの障害物回避法、日本ロボット学会誌、(投稿中)。

口頭発表

1. 比留川 博久、北村 新三、学習制御系と適応プロダクション・システム、第27回自動制御連合講演会前刷、pp.59-60、1984。

2. 比留川 博久、北村 新三、適応プロダクション・システムを用いたマニピュレータ動作の学習、計測自動制御学会第3回知識工学シンポジウム資料、pp.45-50、1985。
3. 比留川 博久、北村 新三、知識ベースシステムによるマニピュレータ制御知識の表現と利用、第29回システムと制御研究発表講演会講演論文集、pp.205-206、1985。
4. 土居 勇、呉松 保男、北村 新三、比留川 博久、モデル参照形適応制御によるマニピュレータの制御、同上、pp.37-38、1985。
5. 比留川 博久、北村 新三、マニピュレータの学習とその獲得知識の利用、第2回フレキシブルオートメーションシンポジウム講演論文集、pp.53-56、1985。
6. 比留川 博久、北村 新三、適応プロダクション・システムに基づくマニピュレータ動作学習過程のモデリング、第9回バイオメカニズムシンポジウム前刷、pp.123-135、1985。
7. 比留川 博久、北村 新三、人見 信、知識ベースロボットシステムKarlの基本構成、第3回ロボット学会学術講演会講演論文集、pp.45-48、1985。
8. 比留川 博久、北村 新三、組み立てロボットのための知的ソフトウェアシステムの開発、計測自動制御学会第4回知識工学シンポジウム資料、pp.25-28、1986。
9. 比留川 博久、北村 新三、ロボットインテリジェンスのためのソフトウェアの表現、第30回システムと制御研究発表講演会講演論文集、pp.187-188、1986。
10. 人見 信、上田 章、北村 新三、比留川 博久、環境教示に基づくロボット動作シミュレーションシステム、同上、pp.85-86、1986。
11. 比留川 博久、北村 新三、角谷 和俊、ポテンシャルと安全第一グラフを用いたマニピュレータの障害物回避法、同上、pp.87-88、1986。

付録1. ポテンシャル関数による運動分解の安定性

マニピュレータの運動学方程式を

$$Y=f(\Theta) \quad (\text{A-1})$$

とする。ここに、 Y はエンドイフェクタの位置と姿勢を示す $M \times 1$ のベクトル、 Θ は関節角からなる $N \times 1$ のベクトル、 f はこれらの関係を表す関数とする。ここで、次の様な2次形式のポテンシャル関数を定義する。

$$V(\Theta) = \frac{1}{2} (Y(\Theta) - \bar{Y})^T W (Y(\Theta) - \bar{Y}) \quad (\text{A-2})$$

ここに、 \bar{Y} はエンドイフェクタの位置・姿勢の目標値、 W は適当な正定対称行列である。ここでは、このポテンシャル関数に基づいて Y を目標値へ動かす Θ の列を得る方法について考える。この方法は、第2章の障害物回避法の基礎になっている。 V に最急降下法を適用すると、

$$\Theta^{(r+1)} = \Theta^{(r)} - h J^T W (Y(\Theta) - \bar{Y}) \quad (\text{A-3})$$

という差分方程式が得られる。ここに、 J は $\partial f / \partial \theta_i$ を第 i 列とするヤコビアンであり、 h は適当な正の定数である。この差分方程式の解の安定性をリアプノフ直接法により調べる。そのまえに、

$$Z(\Theta) = Y(\Theta) - \bar{Y} \quad (\text{A-4})$$

とおいておく。 Θ の集合 K を、

$$K = \{\Theta \mid Z(\Theta) = 0\} \quad (\text{A-5})$$

とすると、Wを正定に選んであるから、

$$V(\Theta) = 0 \quad \Theta \in K \quad (\text{A-6})$$

$$V(\Theta) > 0 \quad \Theta \notin K \quad (\text{A-7})$$

となり、また

$$\begin{aligned} \Delta V &= \frac{1}{2} (Z(\Theta^{(r+1)})^T W Z(\Theta^{(r+1)}) - Z(\Theta^{(r)})^T W Z(\Theta^{(r)})) \\ &= -h (Z(\Theta^{(r)})^T W (J J^T - \frac{h}{2} J J^T W J J^T) W Z(\Theta^{(r)})) \end{aligned} \quad (\text{A-8})$$

となるので、

$$J J^T > \frac{h}{2} J J^T W J J^T \quad (\text{A-9})$$

が常に成立すれば、

$$\Delta V < 0 \quad \Theta \notin K \quad (\text{A-10})$$

となって、集合Kの漸近安定性が成立する。hを十分小さくとるとすると、(A-9)式が成立しないのは、

$$\det(J(\Theta)J(\Theta)^T) = 0 \quad (\text{A-11})$$

のときであって、これはいわゆる「特異姿勢」に相当する。ただし、 ΔV が0になって動けないのは、

$$Z(\Theta^{(r)})^T W (J J^T - \frac{h}{2} J J^T W J J^T) W Z(\Theta^{(r)}) = 0 \quad (\text{A-12})$$

となる方向のみで、これは「特異方向」と呼ばれている。

付録2. 対象としたマニピュレータのAマトリクスと各関節の可動範囲

本研究では、マニピュレータのカイネマティクスを最も良く使われる表記法である Denavit-Hartenberg記法[Denavit 1955]で表している。本研究で例として用いたマニピュレータのAマトリクスを以下に示す。

$$A_1 = \begin{bmatrix} \cos \theta_1 & 0 & -\sin \theta_1 & 0 \\ \sin \theta_1 & 0 & \cos \theta_1 & 0 \\ 0 & -1 & 0 & l_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} \cos \theta_2 & 0 & \sin \theta_2 & 0 \\ \sin \theta_2 & 0 & -\cos \theta_2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} \cos \theta_3 & 0 & -\sin \theta_3 & l_4 \cos \theta_3 \\ \sin \theta_3 & 0 & \cos \theta_3 & l_4 \sin \theta_3 \\ 0 & -1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4 = \begin{bmatrix} \cos \theta_4 & 0 & -\sin \theta_4 & l_5 \cos \theta_4 \\ \sin \theta_4 & 0 & \cos \theta_4 & l_5 \sin \theta_4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5 = \begin{bmatrix} \cos \theta_5 & 0 & -\sin \theta_5 & 0 \\ \sin \theta_5 & 0 & \cos \theta_5 & 0 \\ 0 & -1 & 0 & l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_6 = \begin{bmatrix} \cos \theta_6 & 0 & \sin \theta_6 & 0 \\ \sin \theta_6 & 0 & -\cos \theta_6 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_7 = \begin{bmatrix} \cos \theta_7 & 0 & -\sin \theta_7 & 0 \\ \sin \theta_7 & 0 & \cos \theta_7 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_8 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

ただし、 θ_i は i 番目の関節角、 $l_0 \sim l_5$ の長さのパラメータは、

$$l_0 = 0.265 \quad [\text{m}]$$

$$l_1 = 0.326 \quad [\text{m}]$$

$$l_2 = 0.326 \text{ [m]}$$

$$l_3 = 0.215 \text{ [m]}$$

$$l_4 = 0.030 \text{ [m]}$$

$$l_5 = 0.030 \text{ [m]}$$

であり、 A_8 はエンドイフェクタに関する変換である。

また、各関節の可動範囲は、

$$\theta_1 \quad -180 \sim +180 \text{ [deg]}$$

$$\theta_2 \quad -90 \sim +45 \text{ [deg]}$$

$$\theta_3 \quad 0 \sim +360 \text{ [deg]}$$

$$\theta_4 \quad +180 \sim +350 \text{ [deg]}$$

$$\theta_5 \quad -180 \sim +180 \text{ [deg]}$$

$$\theta_6 \quad +240 \sim +315 \text{ [deg]}$$

$$\theta_7 \quad 0 \sim +180 \text{ [deg]}$$

である。