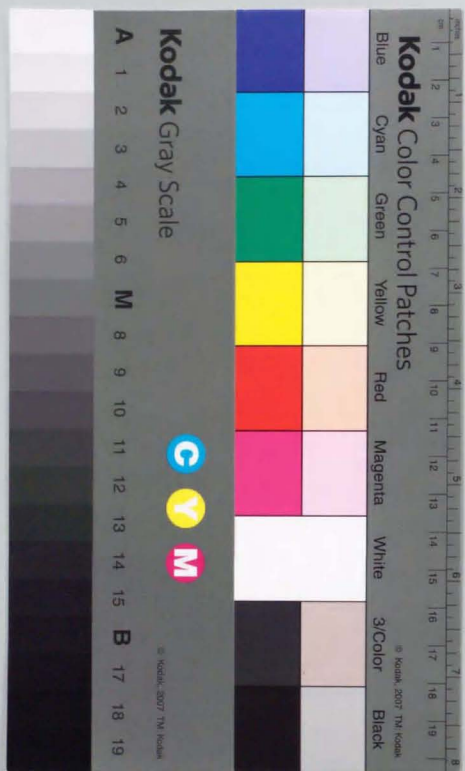


ニューラルネットワークによるロボットの
作業対象物の運動拘束認識に関する研究

関 啓 明



博士論文

ニューラルネットワークによる
ロボットの作業対象物の
運動拘束認識に関する研究

精密機械工学専攻

関 啓明

論文の概要

知能ロボットが、工場のような整備された場所ではなく、屋外や家庭、特殊環境など未知の情報が多い環境で作業を行うのに最も重要なことの一つは、センシングによる作業環境の認識である。本研究では、このようなセンシングの問題として、ロボットが把持した物体の環境による幾何学的拘束状態の検出・認識を目的とする。環境や対象物の形状が与えられたり、穴に棒を挿入する問題のように特定作業で状態が数種類に限定される場合には、その接触・拘束状態を検出したり、可動方向を計算する研究は数多くなされている。しかし、本研究ではロボットの汎用性の見地から、対象物の形状等が未知の場合に拘束状態を検出することを試みた。この検出ができれば、拘束状態に適した制御や組立・分解動作などの動作計画を行うことができる。

人間は暗闇の中でも手探りによる触知覚を用いて器用な作業ができる。そのような認識をロボットにもさせるために、様々な方向に微小な力やモーメントを加え、反力や変位を測定して、把持対象物が動くかどうか調べる能動的な「探り動作」を行わせた。このような拘束の認識は、対象物の形状をとらえる視覚の情報だけでは不十分である。拘束されている対象物の一部が隠されていたり、拘束の部分の形状が複雑であると対応できないからである。

そこで、まず実際にインピーダンス制御された多関節ロボットを用いて探り動作を行い、拘束の状態によって得られるガタや摩擦の影響を含んだ変位と力のパターンを観察した。そして、3次元1自由度の拘束（低次対偶）について解析的な検出手法を構成した。変位と力のパターンから特徴点により各部分の動きを検出し、探り方向の拘束の剛性を求める方法、ロボットの手先の位置姿勢の変化から対偶の種類を判別する方法、ねじ対偶のモデルのパラメータの推定方法等を検討し、3次元1自由度の拘束が検出できることを確認した。ただし、ねじ対偶と回転対偶の差は少ない。解析的手法は、検出手法での閾値等の設定が難しく、多自由度になると拘束のモデルの数も増える。

次に、ニューラルネットワークを用いた検出方法について検討した。主に平面3自由度の拘束を取り扱った。拘束状態は、基本的には、ある空間の方向に対象物が動くかどうかで表現できる。平面3自由度の拘束の場合には、可動空間は3次元空間 (x, y, θ) の部分となる。ロボットの「探り動作」で得られる情報は、その可動空間の離散的な方向の誤差を含む情報（その方向に対象物が動くかどうか）である。そのため、探り動作によって得られた摩擦やガタや柔らかさを含む離散的な方向の拘束の情報を統一的に表現し、その情報から多自由度な拘束を検出するために、場合分けやルールなどの知識ベースではなく、2個のネットワーク（前段・後段）から構成される2段の階層的ニューラルネットワークによる拘束の認識手法を提案する。前段ニューラルネットの入力は探り方向のベクトルであり、出力はその方向が可動(1)か不動(0)かである。拘束状態が与えられる度に前段ニューラルネットの忘却付きの学習を行うと、拘束状態（可動空間）がリンクの重みにより内部表現される。そのリンクの重みには可動空間が情報圧縮されて反映されていると考えられるので、その分布から拘束状態の種類や姿勢を予め学習済みの後段のニューラルネットワークが認識するようにした。このような考えに基づき、16種類に分類した平面3自由度の典型的な拘束を用いて、前段ニューラルネットワークの学習過程、リンクの重みの分布、中間層の働き、及び、後段ニューラルネットワークの認識等について、シミュレーションにより解析・考察を行い、2段のニューラルネットワークによる認識手法の有効性を確認した。実際の探り動作で得られた力と変位のデータから探り方向の拘束の剛性を計算

するアルゴリズムを構成し、離散的な可動空間を得た。また、実測値の場合の、可動空間の大きさやオフセットの影響の防止、対象物の大きさの考慮、学習結果のばらつきへの対処等のために、ニューラルネットワークの改良を行った。ニューラルネットワークを用いると統一的に処理でき、閾値等の微調整が不要であるが、実際の処理が明確でなく、また、ニューラルネットワークが認識可能であるように、入出力の構造の決定やデータの前処理等を行わなければならない。

目次

1 序論	1
1.1 研究の背景	2
1.2 接触・拘束状態の検出に関する過去の研究	6
1.2.1 接触点及び接触法線の検出	6
1.2.2 接触を用いた形状の検出	7
1.2.3 接触・拘束状態の遷移の検出	8
1.2.4 接触・拘束状態の検出	9
1.2.5 接触・拘束状態の表現と解析	11
1.2.6 マニピュレータの制御と拘束の表現	15
1.3 ニューラルネットワークのロボットへの応用に関する過去の研究	17
1.3.1 ニューラルネットワークとロボットの研究	17
1.3.2 ニューラルネットワークによる接触・拘束状態の検出	19
1.3.3 ニューラルネットワークの重みの解析	20
1.4 研究の目的と意義	22
1.5 本論文の構成	27
2 ロボットの探り動作と運動拘束	28
2.1 はじめに	29
2.2 探り動作による拘束の検出の方針	30
2.3 探り動作	31
2.3.1 探り動作に必要なロボットアームの制御	31
2.3.2 インピーダンス制御による探り動作の方法	32

2.3.3	インピーダンス制御と拘束による手先の変位と反力	32
2.3.4	探りの大きさの設定について	33
2.4	探り方向の拘束状態と変位と力の関係	34
2.4.1	探り方向の拘束状態と拘束の剛性	34
2.4.2	ガタや摩擦等がある時の拘束状態と変位と力の関係	34
2.5	探り動作の戦略の考察	37
2.6	まとめ	38
3	探り動作の実験システム	39
3.1	はじめに	40
3.2	実験システム	41
3.2.1	システム概要	41
3.2.2	5自由度多関節ロボットアーム	42
3.2.3	6軸力覚センサ	44
3.2.4	2本指ハンド	45
3.2.5	インターフェース	47
3.2.6	コンピュータと計算の分担	47
3.3	マニピュレータとハンドの制御	49
3.3.1	トルク制御によるインピーダンス制御	49
3.3.2	位置制御によるインピーダンス制御	50
3.3.3	インピーダンスの設定について	51
3.4	まとめ	52
4	実際の探り動作によって得られる力と変位のパターン	53
4.1	はじめに	54
4.2	探り動作の実験	55
4.2.1	探り動作時のインピーダンスの設定	55
4.2.2	探り動作の様子	56
4.3	探り動作で得られた変位と力のパターン	58

4.3.1	自由な方向、拘束された方向、弾性のある方向のパターン	58
4.3.2	ガタの影響	60
4.3.3	摩擦力の影響	61
4.3.4	摩擦力和重力の分離	62
4.3.5	摩擦のあるガタがある場合	64
4.3.6	可動方向に沿って動く場合のパターン	65
4.4	まとめ	67
5	解析的手法による3次元1自由度の拘束の検出	68
5.1	はじめに	69
5.2	3次元1自由度の拘束	70
5.2.1	3次元1自由度の対偶	70
5.2.2	3次元の位置と姿勢の表現	72
5.3	3次元1自由度の拘束の検出アルゴリズム	74
5.3.1	探り動作による3次元1自由度の拘束の検出の手順	74
5.3.2	3次元の探り動作と対偶の動き	75
5.3.3	探り方向の拘束の剛性の検出	76
5.3.4	ガタや一定力の大きさの検出	80
5.3.5	可動方向の検出	81
5.3.6	対偶の種類の判別	83
5.3.7	ねじ対偶のパラメータの推定	86
5.4	探り動作による3次元1自由度の拘束の検出結果	88
5.4.1	探り方向の剛性の検出結果	88
5.4.2	対偶の種類によって発生する変位と力の違い	91
5.4.3	ねじ対偶のパラメータの推定結果	96
5.5	まとめと考察	99
6	入出力画像の種類を認識する2段の階層的ニューラルネットワーク	100
6.1	はじめに	101

6.2 階層的ニューラルネットワークと誤差逆伝播学習	102
6.2.1 階層的ニューラルネットワークの構造	102
6.2.2 誤差逆伝播学習	103
6.3 忘却付き構造学習法	105
6.3.1 忘却付き学習	105
6.3.2 隠れユニット明確化学習	105
6.3.3 選択的忘却付き学習	106
6.3.4 忘却付き構造学習の手順	107
6.4 2 段の階層的ニューラルネットワーク	108
6.4.1 入出力画像の種類を検出する問題	108
6.4.2 従来提案されているニューラルネットワークによる実現法	108
6.4.3 2 段の階層的ニューラルネットワークの構造	112
6.4.4 2 段の階層的ニューラルネットワークの性質	113
6.4.5 画像の種類を認識の手順	113
6.4.6 前段ニューラルネットワークの学習	115
6.4.7 後段ニューラルネットワークの学習	116
6.4.8 2 段の階層的ニューラルネットワークの解釈	117
6.5 ニューラルネットワークのプログラム	118
6.6 2 段の階層的ニューラルネットワークの論理関数識別への応用	122
6.6.1 論理関数を識別する問題	122
6.6.2 論理関数を識別する 2 段のニューラルネットワークの構造	122
6.6.3 前段ニューラルネットワークの学習	125
6.6.4 後段ニューラルネットワークの学習	126
6.6.5 前段ニューラルネットワークの挙動	128
6.6.6 忘却付き学習の効果	128
6.6.7 忘却付き学習の収束の判定	131
6.6.8 選択的忘却付き学習の効果と収束の判定	137
6.6.9 中間層ユニットの並べ変えの指標の検討	140

6.6.10 学習後発生したリンクの重みの分布パターン	142
6.6.11 識別結果	144
6.7 まとめ	147
7 平面 3 自由度の拘束と 2 段の階層的ニューラルネットワークによる認識	148
7.1 はじめに	149
7.2 平面 3 自由度の拘束	150
7.2.1 典型的な拘束	150
7.2.2 拘束の分類	151
7.3 平面 3 自由度の可動空間	152
7.3.1 可動空間と拘束の姿勢	152
7.3.2 拘束の姿勢を表す代表ベクトル	153
7.4 拘束を検出する 2 段のニューラルネットワークの構造と学習法	155
7.4.1 平面 3 自由度の可動空間を識別する問題	155
7.4.2 2 段のニューラルネットワークの構造	155
7.4.3 拘束を識別する手順	158
7.4.4 前段ニューラルネットワークの学習	159
7.4.5 後段ニューラルネットワークの学習	162
7.5 前段ニューラルネットワークの挙動	164
7.5.1 学習データと学習のパラメータの設定	164
7.5.2 忘却付き学習の様子	165
7.5.3 可動空間の範囲とリンクの重みの関係	167
7.5.4 拘束の姿勢とリンクの重みの関係	169
7.5.5 中間層ユニットの活性度	172
7.5.6 中間層ユニットの解釈	172
7.5.7 拘束の種類とリンクの重みおよびユニットの活性度	174
7.5.8 拘束の姿勢と中間層ユニットの活性度	175
7.5.9 リンクの重みや中間層ユニットのパターンの発生確率	176
7.6 後段ニューラルネットワークの学習と識別結果	180

7.6.1	拘束の種類認識	180
7.6.2	拘束の姿勢の検出	181
7.6.3	拘束の種類認識例	181
7.7	まとめ	183
8	探り動作のデータを用いたニューラルネットワークによる拘束の認識	184
8.1	はじめに	185
8.2	探り動作による平面3自由度の可動空間の取得	186
8.2.1	探り動作による拘束状態の検出の方針	186
8.2.2	探り方向の拘束の剛性で表した可動空間	187
8.2.3	探り方向の拘束の剛性を求めるアルゴリズム	188
8.3	実際の探り動作の実験で得た平面3自由度の可動空間	192
8.3.1	探り動作時のインピーダンスの設定	192
8.3.2	探り動作で得られた変位と力のパターンの処理結果	194
8.3.3	探り動作で得た拘束の剛性で表した離散的な可動空間	195
8.4	2段の階層的ニューラルネットワークによる平面3自由度の拘束の認識	197
8.4.1	探り方向の拘束の剛性による拘束状態の認識	197
8.4.2	実際の探り動作で拘束を識別する手順	198
8.5	実際の可動空間に適した2段の階層的ニューラルネットワークの改良	200
8.5.1	改良方針	200
8.5.2	4層構造の前段ニューラルネットワーク	200
8.5.3	対象物の大きさを考慮した学習データ	204
8.5.4	拘束の姿勢を表す代表ベクトルの設定	206
8.5.5	後段ニューラルネットワークのグループ学習による学習誤差の抑制	207
8.6	2段の階層的ニューラルネットワークの改良の結果	208
8.6.1	オフセットや大きさの異なる可動空間の学習結果	208
8.6.2	拘束の姿勢を表す代表ベクトルの設定	210
8.6.3	後段ニューラルネットワークのグループ学習の結果	211
8.7	探り動作で得られた離散的な可動空間の認識	212

8.8	まとめと考察	216
9	結論と展望	217
9.1	本論文の結論	218
9.2	今後の展望	219
	謝辞	220
	参考文献	221
	研究業績	231

第1章 序論	1
第2章 基礎理論	10
第3章 基礎理論	20
第4章 基礎理論	30
第5章 基礎理論	40
第6章 基礎理論	50
第7章 基礎理論	60
第8章 基礎理論	70
第9章 基礎理論	80
第10章 基礎理論	90
第11章 基礎理論	100
第12章 基礎理論	110
第13章 基礎理論	120
第14章 基礎理論	130
第15章 基礎理論	140
第16章 基礎理論	150
第17章 基礎理論	160
第18章 基礎理論	170
第19章 基礎理論	180
第20章 基礎理論	190
第21章 基礎理論	200
第22章 基礎理論	210
第23章 基礎理論	220
第24章 基礎理論	230
第25章 基礎理論	240
第26章 基礎理論	250
第27章 基礎理論	260
第28章 基礎理論	270
第29章 基礎理論	280
第30章 基礎理論	290
第31章 基礎理論	300
第32章 基礎理論	310
第33章 基礎理論	320
第34章 基礎理論	330
第35章 基礎理論	340
第36章 基礎理論	350
第37章 基礎理論	360
第38章 基礎理論	370
第39章 基礎理論	380
第40章 基礎理論	390
第41章 基礎理論	400
第42章 基礎理論	410
第43章 基礎理論	420
第44章 基礎理論	430
第45章 基礎理論	440
第46章 基礎理論	450
第47章 基礎理論	460
第48章 基礎理論	470
第49章 基礎理論	480
第50章 基礎理論	490
第51章 基礎理論	500
第52章 基礎理論	510
第53章 基礎理論	520
第54章 基礎理論	530
第55章 基礎理論	540
第56章 基礎理論	550
第57章 基礎理論	560
第58章 基礎理論	570
第59章 基礎理論	580
第60章 基礎理論	590
第61章 基礎理論	600
第62章 基礎理論	610
第63章 基礎理論	620
第64章 基礎理論	630
第65章 基礎理論	640
第66章 基礎理論	650
第67章 基礎理論	660
第68章 基礎理論	670
第69章 基礎理論	680
第70章 基礎理論	690
第71章 基礎理論	700
第72章 基礎理論	710
第73章 基礎理論	720
第74章 基礎理論	730
第75章 基礎理論	740
第76章 基礎理論	750
第77章 基礎理論	760
第78章 基礎理論	770
第79章 基礎理論	780
第80章 基礎理論	790
第81章 基礎理論	800
第82章 基礎理論	810
第83章 基礎理論	820
第84章 基礎理論	830
第85章 基礎理論	840
第86章 基礎理論	850
第87章 基礎理論	860
第88章 基礎理論	870
第89章 基礎理論	880
第90章 基礎理論	890
第91章 基礎理論	900
第92章 基礎理論	910
第93章 基礎理論	920
第94章 基礎理論	930
第95章 基礎理論	940
第96章 基礎理論	950
第97章 基礎理論	960
第98章 基礎理論	970
第99章 基礎理論	980
第100章 基礎理論	990

第1章 序論

本書は、基礎理論の発展とその応用について、体系的に解説する。第1章では、序論として、本書の目的と範囲を明らかにする。第2章から第10章までは、基礎理論の各分野について、詳細に解説する。第11章から第20章までは、基礎理論の応用について、解説する。第21章から第30章までは、基礎理論の発展について、解説する。第31章から第40章までは、基礎理論の応用について、解説する。第41章から第50章までは、基礎理論の発展について、解説する。第51章から第60章までは、基礎理論の応用について、解説する。第61章から第70章までは、基礎理論の発展について、解説する。第71章から第80章までは、基礎理論の応用について、解説する。第81章から第90章までは、基礎理論の発展について、解説する。第91章から第100章までは、基礎理論の応用について、解説する。

1.1 研究の背景

「ロボット」は、いわゆる目的主導型の専用機械に対して、多種多様の目的を柔軟にこなすことができる汎用機械として位置づけられる。マニピュレータやハンドは、片持ち梁と関節が交互に連なったシリアルリンク構造が代表的である。シリアルリンクの先端の位置精度を犠牲にするかわりに、位置姿勢を自由に安えらるよう運動の自由度を大きくし、汎用的な動作が可能である。そして、その自由度の制御や動作の計画をするソフトウェアを巧みに記述することで多様な作業を実現できる。そこで、ロボットの汎用性は、いかにロボットの知覚や行動を記述・表現するかという問題にかかってくる。それらを作業毎に記述するのではなく、できるだけ広い作業に対して統一的に知覚や行動を表現できる効率的な枠組みを作っていく方が望ましいと考えられる。

日本では、1980年以降、産業用ロボットが急速に普及した。電気アクチュエータの性能が向上し、関節の位置や速度のセンサを使った制御法が改善され、安定した精度の良い動作が可能となり、なにより、マイクロエレクトロニクス発展によって、コンピュータの情報処理能力が急激に増大したからである。使用されているほとんどの産業用ロボットは、教示した動作を忠実に再生してくれる、ティーチングブレック方式のマニピュレータである。組立作業等において正確な軌道を教示すれば、特にセンサを用いなくても、かなり多くの種類の単純な繰り返し作業を行わせることができた。もちろん、ロボットの周囲の環境や部品等の対象物の状態は一定であることが要求され、工場では、部品等の精度や位置姿勢のばらつきを所定の範囲内に収める努力をしたり、部品の供給など作業環境をロボット用に整備したりしている。

しかしながら、これらの産業用ロボットは動作が汎用的であるだけで、よく言われるように、単なる定型繰り返し動作を行う自動機械に近い。ロボットが動作しやすいように対象物や環境を整備しにくい場合やそれらと相互作用を伴う作業を行わせる場合は、外界センサを装備して環境や対象物の状態を検出する能力をロボットに付与する必要がある。現在の工場では、ティーチングブレック方式では限界のある作業のために、センサ付きロボットが多く導入されている。コストの面から簡単に単純なセンサと処理を備えたロボットから、視覚や力覚を装備し複雑な検出を行うロボットまで、位置決め、製品識別、検査、組立、溶接、挿入、バリ取り、グラインダ作業など、かなり広範な作業に使われている。ここでも、センシングしやすいように部品などの対象物や照明などの作業環境を整える工夫が多く見受けられる。また、部品などの対象物の状態が簡単なモデルで記述できたり、予め、その状態が数パターンの中のどれかであると分かっている場合は、センシングも容易になる。

それに対して、乱雑に並んでいる形状の複雑な部品を取り上げる作業、柔軟物を扱う作業、熟練者の経験やコツに頼っている作業、接触や拘束が複雑な組立作業などは、その作業をモデル等で明示的に記述することは簡単にはできない。そのような作業をロボットで自動化することは難しく、研究段階である。

また、工場のような閉ざされた世界ではなく、開かれた世界で動く産業用以外のロボットに目を向けてみる。屋外や家庭、宇宙や特殊環境用のロボットにとっては、周囲の環境や取り扱う対象物を検出・認識する能力は、産業用ロボットにまして重要である。工場内と違い、環境をロボットにとって都合の良いように改変・整備することが難しいうえ、対象物や環境は複雑に変化することも多く、それ

らの状態を数種類に限定したりすることは不可能に近いからである。人間が不確かさを含む環境の情報全体をロボットに与えておくことは無理である。このような場合には、外界センサを用いて環境や対象物の状態を検出・認識するばかりでなく、その情報と内部のモデルや知識を照らし合わせて判断し、環境に適応して自律的に行動・動作するロボットが求められている。そのような汎用的なロボットは「知能ロボット」「自律ロボット」と呼ばれる。

ところで、代表的なロボットの感覚には、視覚、近接覚（距離覚）、力覚、触覚などがある。それぞれ次のような特徴があると考えられる。

● 視覚

実世界の3次元情報を一度に広範囲にわたって非接触で得ることができる。ロボットの周辺のもの配置や動きを大域的に知覚するのに適している。注視している部分については、かなり詳細な対象物の形状や動きを検出できる。両眼視からは奥行き情報も得られる。ロボットの視覚としては、通常、カメラが使用される。

● 近接覚（距離覚）

レーザや超音波センサなどが用いられ、3次元空間のある1点までの距離を計測するのが基本である。かなり正確な距離が得られるのが特徴であり、走査することにより3次元の距離画像も得られる。

● 力覚

力覚センサを用いて、ロボットが直接作用・操作している対象物との間で確定的に得られるものである。ロボットと対象物の相互作用に起因する極めて局所的な情報である。局所的な一部の情報だけで意味のある検出を行う場合もあるし、局所的な情報を複数収集して、それらを合成あるいは再構成して対象物の状態を検出する場合もある。

● 触覚

触覚も力覚と同様のことが言える。また、ロボットの触覚は圧覚や滑り覚が主であるが、それだけでは物体の質感などはとらえることができず、触覚の本質が何であるかは解明されていない。そのため、多種多様の触覚センサが研究開発され、1つの形には固定されていない。

このようなロボットの感覚すなわち外界センサの感度、分解能、精度等を向上させ、特性を良くすることは重要な基本課題であり、地道な研究が続けられている。また、センサはリアルタイムのロボットの制御に直接結びついている。関節の位置や速度を検出する内界センサだけでなく、視覚や力覚などもフィードバックされ、ロボットの手の運動性能を上げ、対象物を巧みに操縦するよう位置制御や力制御の改良が進んできた。さらに、外界センサの出力を反射的に行動に結び付ける行動様式の研究もある。

しかし、知能ロボットにとって、センサと同じくらいあるいはそれ以上に重要で本質的な問題は、

「如何にしてセンサの情報から、意味のある情報を処理・検出し、環境を知覚・認識するか」

ということ、つまり、センサの情報処理と環境の理解である。センサの情報処理に関しては、各々のセンサのハードウェアに大きく依存しており、様々な手法が考えられてきた。古典的な信号処理の技

術から、エキスパートシステム、ニューラルネットワークやファジィ推論などの方法も応用されている。各種の方法で処理された情報に基づいて環境を理解するということは、ロボットの内部に環境や対象物の構造を反映した何らかの外界のモデルが作成されると考えられる。そして、その獲得された内部モデルに基づいて、環境に適応した行動を行うと考えられ、こうした考え方を「モデルベーストロボティクス」と呼ぶ。例えば、移動ロボットの場合には、目的地に移動するという明確な作業が決まっており、ここでの内部モデルは地図にあたる。未知環境に移動ロボットが置かれたときに、まず最初しなければならないことは、外界センサを用いて地図を作ることである。予め地図が与えられている場合でも、障害物などの環境の変動も含めて全て与えることは不可能であるから、環境の変化をセンシングして地図を更新していく必要がある。一般に、視覚や距離覚を利用した内部モデルは、環境や対象物の形状や配置など幾何学情報であることが多い。ただし、このような内部モデルの具体的な形態は、作業の目的、センサの種類、情報処理により様々なものが提案されており、汎用的なものはない。

さらに、知能ロボットのためのセンサ情報処理に関して、2つの重要な概念がある。1つは「センサーフュージョン」[山崎 92]の考え方である。複数のセンサの信号や情報を合わせることで、新しい効果や情報を得るものであり、「センサ統・融合」とも呼ばれる。

- 複数のセンサ情報を並列的、相補的に組み合わせる「複合」
- 各々のセンサからの情報にある演算処理を施してまとめた情報を得る「統合」
- センサ情報間あるいは内部モデルとの間で相互の関係が新たな知覚情報を与える「融合」
- 認識、予測、学習、記憶等のためにセンサ情報間の関係を理解する「連合」

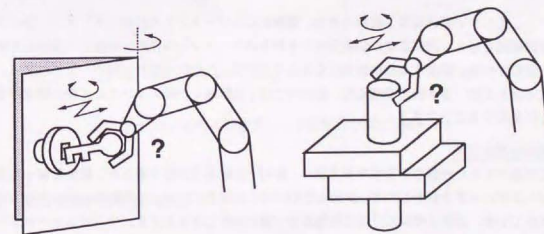
の4つに分類されている。例えば、ロボットの視覚でドアを検出することはできるが、実際にドアがどちらに動くかは、ドアのノブを把握して、力覚や運動の感覚を使って調べる方が自然である。対象物の形状の認識においても、視覚で辺や頂点などの特徴点をとらえたと共に、触覚で確認すると信頼性が上がる上、双方が補完しあって内部モデルを構築することも可能である。

もう1つは「アクティブセンシング」[石川 93][石川 88]である。エネルギーの面からは、センシングの実現のため、被測定系に対して何らかの形でエネルギーを供給するセンシング形態、情報の面からは、センサから得られる情報だけでは有用な情報が抽出できない場合、何らかの形で必要な情報を外部から供給するセンシング形態と定義されている。センサはアクチュエータと共存しており、お互いに密接な相互作用を持っている。人間の認識過程において、運動の指令が感覚系に伝えられること（遠心性コピー）は重要な意味を持つと指摘されている。例えば、視点を動かして、いろいろな角度から対象物を眺めることができるし、マニピュレータやハンドで対象物を触ったり動かしたりして力覚や触覚で感じることができる。対象物が動かなくても、センサを動かせばセンサの出力は変化する。アクチュエータを動かすためにセンサ情報をフィードバックするばかりでなく、有用なセンサ情報を得るために、アクチュエータが使われる。また、アクチュエータの動き自体も対象物を認識するための情報となり得る。単にセンサを走査するだけでなく、探索動作を行って、その結果得られた情報によって探索動作自体を変更していくこともある。このような運動を伴った能動的なセンシングにより、

- ある空間の中から対象物を探索したり、
- 局所的な情報を複数収集して大域的な情報を得たり、
- センサの分解能や特性を変化させたり、
- 力学的特性を検出したり

することができる。特に、力覚、触覚などは、根本的に局所情報であり、マニピュレータやハンドを動かして探索するセンシング形態がより本質的である。また、触覚や力覚機能を工学的に実現するためには、単なるセンサデバイスの問題だけでなく、アクチュエータ系の運動の機能も含めた触知覚(haptic)システム[Lederman 87][Campos 91][Rossi 91]として考えなければならない。

これまで述べてきたように、知能ロボットにとって最も重要なことは、センシングによる環境の認識である。外界センサの情報処理、能動的探索動作、環境の内部モデルの獲得は、相互に深く関連しており影響し合うことが必要である。ロボットの様々な感覚のうち、本研究では、ロボットの方覚や触覚を中心に考えていく。環境の認識という面では、対象物の接触や拘束状態を取り扱うことにした。これらは、ロボットが環境や対象物と相互作用するときに避けられない問題である。視覚情報だけでは検出できない物であり、どのようにしてセンシングし、内部表現した方がよいのか、興味のある問題である。また、内部モデルとしては、いわゆる幾何学情報ではなく、ニューラルネットワークの内部表現を有効に利用する方向で検討していく。ニューラルネットワーク以外の解析的方法についても比較検討し、ニューラルネットワークをこのような問題に適用した時の構成や挙動の特徴を明らかにしたい。



ドアの開閉
組立 / 分解
ドアの開閉や組立/分解作業などでは、探り動作で運動拘束を検出する

図 1.1: 接触・拘束状態の検出

1.2 接触・拘束状態の検出に関する過去の研究

1.2.1 接触点及び接触法線の検出

ロボットのマニピュレータやハンドが環境と接するとき、最も基本的なセンシングは接触点、接触法線および接触力の検出である。接触点が求められると、頂点やエッジ以外では、その点の近傍の形状から接触法線が同時に求められる。

分布型の触覚センサによる方法

ハンドと対象物間の接触点を検出するときには、指の部分に分布型の触覚センサを取り付けることがよく行われ、様々な方式の触覚センサが開発研究されている。接触点が直接その場所にあるセンサの出力となって得られるので確実であるが、信号線の増大の問題もあって、指の曲面全体に取り付けられるような分解能の高い触覚センサは少ない。また、把持した対象物と環境の間の接触点を検出する場合は、その間に触覚センサを取り付けておくことはできない。

力覚センサによる方法

力覚センサを使って接触点を計算する方法もある。この方法はセンサの数が少なくすむ。対象物と1点で点接触し、モーメントは作用せずカベクトルだけが作用している場合、根元の力覚センサでモーメントと力を測定して、接触点の位置を計算する方法 [Bicchi 91] が一般的である。接触点のカベクトルはそのまま並進力として検出でき、接触点の位置ベクトルとカベクトルの外積がモーメントとして力覚センサで検出できることを利用している。[Salisbury 83] は単純なリンクなどの場合の点接触、平面内の線・面接触の検出法を定式化し、[Brock 85] らや [Tsujiyama 89] らは3次元に拡張している。[Gordon 89] らは、複数の関節トルクセンサを用いて、センサの位置が複数あることを利用してリンクと対象物の接触点を検出する方法を提案しているが、基本的な考え方は同じである。

しかし、柔らかい物体による接触の場合、接触点にはモーメントも作用してしまう。[Bicchi 90] は接触を面接触と考え、接触点での法線方向のまわりのモーメントの作用も考慮して接触点を検出する方法を示している。接触する指先の形状を与えておけば、凸形状の場合は解が一意に決まる。これらの力覚センサを用いる方法の問題点は、基本的には1点接触しか扱えないことと検出精度が接触力の大きさに依存することである。

能動的な検出方法

ここまで述べてきた接触点を検出する方法は、静的で受動的な方法であるが、接触を保ったままマニピュレータやハンドを動かしたり、加える力の方向を変化させたりして能動的に検出する方法も多く提案されている。静的な情報だけでは接触点を一意に決定できなくても、マニピュレータやハンドの位置姿勢を変化させたときの情報をいれば接触点が検出できることは多い。

[永田 92][永田 90] らは、把持物体の頂点を外部環境と接触させ、回転中心の考え方を用いて、力覚からその接触している頂点の位置を検出している。そして、それを把持物体の間の座標変換パラメータを求めることに応用している。このとき接触力の作用線は接触している頂点を通るので、環境との接触動作を姿勢を変えながら複数回払い、異なる方向に力を発生させて、力の作用線の交点を求める

ことにより接触頂点の位置が推定される。[北垣 90][北垣 92] らや [澤田 90] らは、姿勢を変えた接触動作で情報を得るのではなく、摩擦円錐内の複数の向きの力情報を検出して、得られた複数の力の作用線から最も近い点を接触点としている。この方法は、把持物体の形状の情報を利用せず、力の情報のみから接触点を検出できる。しかし、接触点を一定に保つ必要があり、把持物体の頂点の検出にはよいが、曲面の接触点の検出は困難である。

[武藤 93] らは、力覚の情報だけでなく、把持物体（ツール）の幾何学情報を用い、マニピュレータを動かしたときの内界センサから得られる速度の情報を相補的に利用した実用的な接触点検出法を提案している。これは、力覚センサの情報は一般に誤差が大きくノイズや摩擦などの外乱に弱いため、それから計算される接触点も誤差が大きくなるという欠点を補うものである。把持している対象物の接触点が曲面であるときは、並進速度ベクトルと法線ベクトルが直交し、接触点が法線や頂点の時は、隣接する面の法線ベクトル群との内積の中に0または異符号のものが存在することを用いて、力情報と対象物の幾何情報より得られた接触点を補正している。

誤差の大きい力覚センサの情報を使用せず、関節の位置センサの情報を用いて接触点を高精度に算出している研究もある。[金子 90][金子 91][金子 93] らは、自己姿勢変形動作と呼ばれるコンプライアント動作によって得られるリンクの姿勢変化から接触点を検出している (図 1.2)。コンプライアントなリンク系が対象物と接触を維持しながら、姿勢を変化させていった時の前後のリンクの姿勢の交点を接触点とするものである。ハードウェアの変更を必要とせず、力センサよりノイズや摩擦の影響を受けにくいのが利点である。

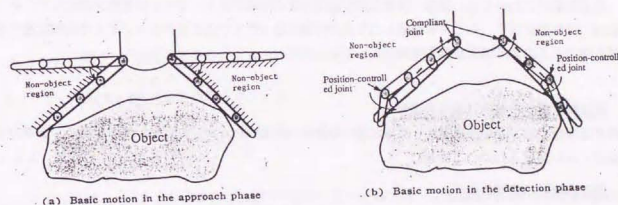


図 1.2: 自己姿勢変形動作による接触点の検出 [金子 90]

1.2.2 接触を用いた形状の検出

対象物が比較的小さい場合

触覚センサより対象物の大きさが小さい場合は、触覚センサを対象物に押し付けて圧力分布などを測定すれば、対象物の形や位置姿勢が推定できる。触覚イメージがそのまま形状の検出になる [Luo 86]。

対象物が比較的大きい場合

ところが、触覚センサより対象物が大きい場合や力覚センサの情報を利用するような場合には、対象

物を撫で回して頂点や稜線などの特徴的な部分を探索動作のような能動的探索動作が必要になる。ロボットの場合一般的には、マニピュレータやハンドを対象物に一定の力で押し付けたまま、動きながら接触点をセンシングしていくぞり動作あるいは微い動作を行ってセンシング領域を広げ、形状を検出することが多い[佐藤 89]。

先に述べた[武藤 93] とも、力覚センサの情報とマニピュレータの並進速度ベクトルを用いて接触点を連続的に検出して、対象物の輪郭形状を求めている。[Dario 87] らは、センサが対象物に接触した際に生じる熱量の流れを測定する触覚センサを用いて接触点を検出しながら、小さな四角形を描くように対象物の表面をなぞって形状を再現している。

ただ単に面や線をなぞるだけでなく、センシングの結果を見てなぞる方向を変えていくというより積極的な探索動作を行わせた研究もある。[Roberts 90] は、例えば、面をなぞって稜線を探し、稜線が見つければ稜線をなぞって頂点を探し、頂点が見つければさらにそこからつながる未知の稜線や面をなぞるというように探索動作を効率的に計画している。そして、それらの面や稜線や頂点の情報と予め内部に作成された対象物の3次元の幾何学情報とをマッチングすることによって、対象物の認識を行っている。このように、いくつかの対象物の幾何学形状を識別するのに効率的なあるいは最適なパスや動作を計画する研究もいろいろとなされている[Ellis 92][Berger 91]。

1.2.3 接触・拘束状態の遷移の検出

組立作業等においては、接触・拘束状態を変化させる作業が多く、それらの状態が変化したことを検出して作業が完了したことを認識したり次の作業に移ったりする必要がある。そこで接触状態の遷移を検出することは非常に重要である。

非接触状態から接触状態への遷移

簡単な問題では、非接触状態から接触状態への遷移の検出があるが、力覚センサ等を用いて衝撃力を測定するのが良く行われている。

異なる接触状態間の遷移

異なる接触状態間の遷移でも、作業が決まっておリ接触状態が数種類に限定される場合は、力センサの値にあるしきい値を設定して検出することも多い。[柳原 92] らは、散在した部品をハンドリングするとき、把持した部品以外の部品に接触して部品の状態が変化するのを、力覚センサの値にしきい値を設けて検出し、それをきっかけに視覚センサで部品の環境モデルを更新している。

しかし、力覚センサの値を処理して接触点の変化等を抽出する方が、対象物の接触・拘束状態よりはっきりとした形でとらえることができる。[北垣 90] らは、力覚センサの情報から接触点の位置を推定し、接触点が変わらないと仮定したときのモーメントを予想して、実際のモーメントとの差をモニタリングすることにより、対象物を回転させて迎合合わせ動作を行わせるときの1点接触から線接触などの他の接触状態への遷移を検出している。

[高瀬 84][高瀬 89][高瀬 85][高瀬 87]、[末広 90][末広 88]、[長谷川 91] らは、マニピュレータが行う作業をスキルとよぶ基本動作に分解抽出し、特に組立作業においては、突き当て、迎合合わせ、はめ合

いの3つのスキルで典型的な動作はほとんど達成できるとしている。これらの平面内の多角形対象物の組立作業において、マニピュレータの並進と回転速度から接触したままの対象物の瞬間回転中心を求めることによって、例えば、瞬間回転中心のステップ場の変化をとらえて、接触状態の遷移を検出し、作業の終了条件としている。各々の点接触による拘束状態に移行するための制御方策としては、接触点に基づいた仮想リンク機構の概念を用いて、力制御と位置制御を適用している。そして、接触による拘束を維持しながら運動する瞬間回転中心(拘束中心)の集合である接触法線と、その拘束から抜け出すことができるような左、右回りの瞬間回転中心(離脱中心)の集合である左、右離脱中心領域から、確実な接触を実現するための環境拘束による運動の強制停止などの接触運動を解析している(図1.3)。対象物を把持している場合には、作業の進行にともなって触覚パターンが変化するとい

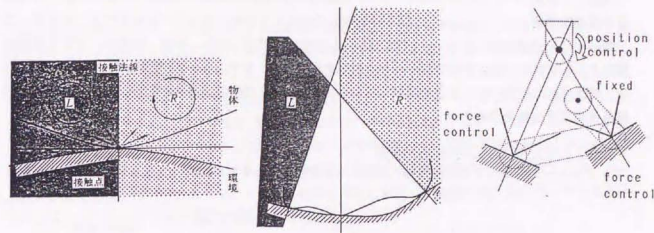


図 1.3: 瞬間回転中心による接触運動と仮想リンク機構の概念 [長谷川 91]

うことが、[Fearing 87] により報告されている。

1.2.4 接触・拘束状態の検出

ここでいう接触・拘束状態の検出は、個々の接触位置や法線方向、接触力などのような局所的な情報ではなく、例えば、点接触、多点接触、線接触、面接触といった大域的な接触や拘束の状態がどうなっているかという情報を得ることである。接触状態が遷移した後の状態がどうなったかを知ることでもある意味のあることである。

触覚センサによる検出

触覚センサを用いたものとしては、対象物の全体形状の検出もかねているが、[木下 91] の触覚アスベクト法がある。対象物体を多面体と仮定すると面、稜線、頂点が特徴であり、それらの部分に触覚センサが接触すると、それぞれの特徴に対応した触覚像が得られる。多面体の表面に対する触覚像は、その多面体の幾何学的な構造に基づいてグラフ化され、触覚像間が関係づけられる。逆に、それらの情報を用いて、検出された触覚像から、パターンマッチングを行って、接している多面体の位置や姿勢等を求めることが可能である。[Browse 87] は触覚像とモデルとのマッチングから把持姿勢を

探索した。また、[Fearing 88][Fearing 85]らは、人間の指のような弾性のある指に、剛体の辺や頂点や円筒が色々な角度と力で接触したときの弾性指に生じる応力や歪みを解析し、検出される応力分布から、接触の状態を識別することを提案している。また、円筒形の弾性指の接触部分の変形から、接触している物体の曲率や姿勢等を検出する研究も行っている。

多指ハンドの関節角の利用

特に多指ハンド等の場合には、指の配置の情報を有効に利用することができる。把握した物体の接触状態や位置姿勢を求めるには、空間的に配置されたセンサの局所情報と指配置と把持物体の形状モデルとの間の幾何学的なマッチングにより、把持姿勢の候補等を探すが一般的である [Salisbury 88]。Grimson らは多面体の対象物のモデルと指の配置の組み合わせから解釈木を定義して、指のセンサから接触点と法線ベクトルが得られたとして、接触点間の距離や法線間の角度などの幾何学的拘束条件から候補を探索した。[Schneider 86]らは効率よく探索するためにセンサや指を度のように配置したら良いかという議論をしている。[中村 90]らも同様に、頂点、稜線、平面、円柱面、球面を局所の特徴とし、それらの検出が可能なセンサの使用を想定して、モデルベースで解釈木により対象物の同定、位置姿勢の計算を行う手法を提案している。[Siegel 91]は指の関節角とトルクの情報を用いて把持物体の位置姿勢を求めている。

ロボットの手先の力と変位による検出

マニピュレータが操作する対象物体の接触・拘束状態を検出するには、ロボットの手先の力と変位

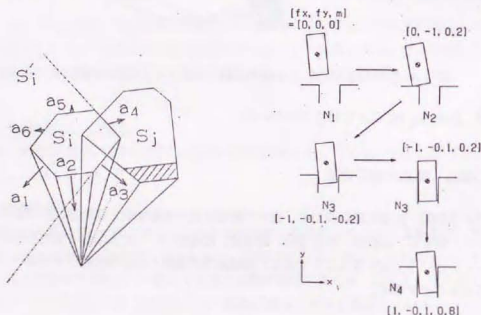


図 1.4: 凸多面体による観測可能信号集合と識別手続きの例 [平井 90]

の信号がよく用いられる。[Hou 89]らは、スクリーン理論を用いて位置姿勢と力の情報から数種類の拘束状態を同定する方法を示している。[平井 88][平井 90][平井 88]らは、力と変位の信号から接触状態そのものを識別・同定する手法を提案している (図 1.4)。組立部品の剛体の幾何モデルの面・稜線・頂点の組み合わせで表現された接触状態について、拘束された物体が可能な運動と物体に作用し得る外力が凸多面体集合を構成することを用い、各々の接触状態で観測可能なセンサの信号 (外力) の集

合を求めた。そして、与えられた作業時に生じる得る接触状態の間で、凸多面体理論を用いてそれらの観測可能信号集合の合併集合等を求めることにより、その作業における接触状態の識別手続きを定式化し、識別関数を自動生成した。また、接触状態間の遷移可能性などを利用して、識別手続きを単純化する手法も示している。そして、これらの手法を利用して組立作業をモニタリングすることを提案している。しかし、接触状態を一意に決定するのは困難な場合が多い。また、実験的な検討は行われておらず、今後の検証が待たれる。

探索動作の利用

大局的な接触・拘束状態の検出においても、アクティブセンシングの手法はしばしば用いられる。[Desai 89]らは、センサと幾何学情報の誤差を考慮し、静的な力の情報だけでは状態を決定できない時には、能動的に微小変位を与えたときの力の変化から状態を決定している。

具体的な作業において、探索動作によって接触状態を検出した報告も多い。作業を限定すれば、接触状態を検出する方法やそのためのセンシング動作の計画もその作業に特徴的な情報を用いて専用で記述することができ、比較的容易である。力覚センサの情報を用いて、穴を探索してペグを挿入したり、同じく、コネクタとプラグの接触姿勢を探索動作で検出し、平行に接するように姿勢を変えて位置合わせを行って挿入したりする作業 [澤田 88][石川 91]が実現されている。ペグの挿入作業において、挿入中の状態を、その前の状態と作業指令とセンサ情報からなる経験的なルールによって、認識する方法も考えられている。また、ペグの挿入のときに生じる各接触状態において挿入力やモーメント及びそれらの変化がどうなるかをファジィ集合で定性的に記述し、実際の挿入時のセンサ情報からファジィパターンマッチングによって接触状態を検出している。

ここに述べたほとんどの方法は、予め知識として対象物の幾何学情報を与えているか、作業を限定していることが多い。未知環境では把持した対象物の形状は不明であり、視覚を用いたとしても、対象物が接触・拘束状態にあるときには対象物が隠れて見えにくい部分も多く、完全な幾何学情報を得ることは相当困難である。そのような場合に、接触や拘束の状態を検出する研究は少ない。

1.2.5 接触・拘束状態の表現と解析

ここでは、これまで述べてきた接触や拘束状態を検出する研究以外で用いられている、接触や拘束状態の表現や解析に関する手法について記述する。一般には、対象物体の幾何学モデルに基づいて、運動や接触・拘束状態の解析や計算がなされる場合が多い。

peg in hole 作業の記述

精密はめ合い作業、いわゆる peg in hole 作業は、古くから解析検討されてきた問題である。問題は、力やモーメントを検出して peg の位置や姿勢をどのように修正していったら良いかである。[Whitney 82][Whitkey 81]らは、面と面がある hole への円柱の peg のはめ合い作業を対象として、組立過程で生じるそれぞれの接触状態について、摩擦力やくいづき (Wedging, Jamming) 等を考慮した詳細な静力学解析を行って、はめ合い作業のための作業用 R C C (Remote Compliance Center) ハン

ドを開発している。[内山 93]や[野崎 91][野崎 89]らはピン挿入シミュレータを開発している。それは peg と hole を剛体とし、分類された接触状態ごとに接触点におけるはねかえり係数や摩擦係数を含めて運動方程式を解き、運動方程式から得られるパラメータが幾何学的条件と力学的条件を満たすかどうかで状態遷移の判定を行っている。つまりやくいつきの状態も考慮して解析している。[Lozano 84]らは、作業状態の識別可能性などを含めて、はめ合い作業の動作計画問題を検討し、backprojectionの技法を応用している。[井出 92]らは、peg in hole 作業を hole を探す探索動作と peg を差し込む挿入動作に分けて解析し、センサからの力覚情報とロボットの修正動作の間の対応関係を表す写像を、作業を繰り返しながら、2 分木の自己組織化データベースを用いて学習・獲得していく組立作業システムを構築している。

グラフ理論による記述

接触状態の遷移の表現に関しては、状態の変化の関係をグラフ理論で表現することが多く行われている。[平井 88]らは、組立作業において、部品同士の接触状態に注目し、接触状態の変化を追跡することによって組立過程を表現し、その運動学的性質を調べている。部品と部品がどの面や頂点で接触しているかによって両者の接触状態とその変化をグラフによって表現し、接触状態ネットワークを構成する。平面運動する多面体の場合について、物体同士は点接触すると仮定して、接触状態ネットワークを自動生成するアルゴリズムを提案した。[末広 93][末広 91]らは、多面体組立作業を操作対象物体と環境との面接触関係の実現ととらえており、ある操作に依存して面接触関係が生じることを接触関係グラフで記述している(図 1.5)。そして、作業を行った後、視覚システムから幾何モデルを得たとき、その作業から得られる面接触条件を満たしていない場合には、逆に、面接触条件を満たすように幾何モデルの位置姿勢を修正することを行っている。

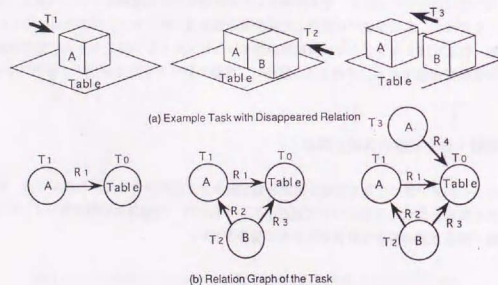


図 1.5: 接触関係グラフによるタスクの記述 [末広 91]

多面体同士の接触の記述

未知の環境ではなく、対象物の幾何学形状モデルや目的とする作業を与えておいて、そのときの拘束

状態を表現し、計算・解析する研究は数多くなされている。特に、対象物が多面体同士である場合には、幾何モデルから多面体同士の接触による拘束条件を導出する一般的なアルゴリズムが既に示されている。[Ohwovoriole 81]らは、スクリュ理論により多面体の接触を定式化し、拘束から離脱する動作、拘束を維持する動作、拘束物体に発生できる力を求める方法を示した。平井らは、ロボットが操作する対象物の運動を片側拘束に起因する一般的な不等式で記述した。この不等式条件を扱うために、凸多面体理論を導入し、演算や判定のアルゴリズムを構築している。多点支持の幾何学的な拘束によって剛体を把握したときに許容される物体の位置姿勢の微小変位や部品の組み付け、分解時に許容される微小変位等の計算に応用した。また、[平井 92]らは、把持した物体が環境と接触するときの反力を計測してマニピュレータの運動を制御するダンピング制御において、幾何学モデルと接触状態の遷移を表す作業モデルから、位置決め誤差関数と接触点における拘束ベクトルと非負 1 次結合形式で表された力ベクトルを使って、そのときのダンピング係数行列を導出する研究も行った。この方法は環境が複雑になると困難になる。[比留川 91][比留川 91][比留川 93]らは、作業対象と作業環境の形状は 3

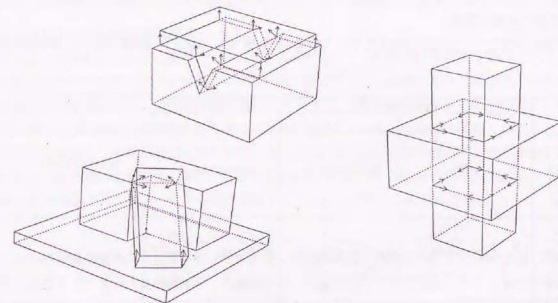


図 1.6: 作業対象の拘束条件の計算例 [比留川 91]

次元の多面体に限定し、作業環境は固定されていてそれらの間には摩擦がないとして、作業対象に許される微小変位を求める手法を提案している(図 1.6)。各接触点での微小並進の拘束条件から、作業物体の座標系での微小変位・回転に変換し、線接触は線分の両端点から受ける拘束と等価であることを利用して、両側拘束や片側拘束を区別することなく、多点の接触による一般的な拘束条件式とそれを求めるアルゴリズムを示した。また、拘束条件式の係数行列を特異値分解することにより凸多面体の頂点を求める問題に帰着させて計算量を少なくし、拘束された物体に許される任意の微小変位を、拘束を変化させる微小変位と維持する微小変位の非負 1 次結合の形で計算する手法を示し、拘束された物体の離脱動作に応用した。[加藤 91]らは、2つの部品が複数の接触点で拘束し合うとき、拘束を保ったまま動くことのできる方向と自由度をヤコビ行列を利用して求め、接触点でのガタの影響を誤差領域の導入によって解析している。そして、複数の部品間に複数の拘束が存在する場合に、全部

品が全ての拘束関係を保ちながら微小変位できる空間を、グラフで表現された部品の座標系と接触点の座標系間の変換関係を用いて、多次元コンフィギュレーション空間上の超平面で区切られた領域として求めている。

[小方 93][小方 92] らは、並進自由度物体の組立作業において、作業空間を操作しない多面体物体の面を含む平面で領域に分割し、平面のどちら側に操作物体があるかどうか(作業状態)を操作しない物体の面の数だけある符号(＋)でシボリックに表現して、その符号列を作業状態毎に比較することによって作業状態の隣接条件や干渉を判定している。操作物体に大きさがある場合には、configuration space 法によって操作物体が点に変換される configuration space で同様の方法を用いている。また、作業状態の隣接関係を遷移グラフで表し、組立作業の教示システムに適用している。植之原は、ロボットの目標軌道の周りに位置誤差量の空間を設定して、その領域内部で生じる可能性のある接触状態を探索する方法を提案している。多面体を凸多面体モデルの和・差で表現し、多面体間の接触は点と面と稜線の組み合わせで生じるため、点と面、稜線と稜線の干渉判定を6次元のコンフィギュレーション空間で行っている。

拘束状態の分類と表現

拘束状態の分類とその表現の研究もなされている[横小路 93]。[高瀬 90][高瀬 93] は、多面体の典型

回転運動の束縛自由度数				
	0	1	2	3
0	無接触	該面はほとんどなし	該面はほとんどなし	該面はほとんどなし
1				
2				
3				

図 1.7: 物体の拘束状態の分類 [高瀬 90]

的な組立状態について、3次元多面体の拘束状態を並進運動と回転運動に対する自由度の束縛の数によって分類し、それらの拘束状態を実現するために必要な基本的なハンドリング作業を示した(図

1.7)。多自由度の結合が小さな自由度のシーケンスに分解できる場合は、それらの拘束状態の結合で表現した。そして、それぞれの組立状態に達するまでの動作を、付き当て動作、辺合わせ動作、はめあい動作およびそれらと力制御の組み合わせによって記述している。自由度の束縛の数は同じでも、かなり異なった拘束状態が存在するが、この方法では片側拘束と両側拘束などの区別を陽には行っていない。[池内 93] らは、3次元の微小な並進運動を扱い、多面体の面接触状態の分類を行っている。同様の分類は Krishnan によっても行われている。面接触が1つ生じると運動方向は法線方向の半空間に制限されることから、接触面をその法線方向に対応したガウス球面上の点で表現し、可能な運動方向もその方向に対応したガウス球面上の点で表現している。そうすると、可動方向はそれぞれの接触面に対応したガウス球面上の半球面の共通領域となり、典型的な面接触状態での自由度について考察している。そして、拘束からの離脱運動などによる各面接触状態の遷移により、組立作業時の作業モデルを構成している。[澤田 89] らは、環境に操作物体が接触して動く動作について、環境側からみた接触点位置と操作物体側からみた接触点位置が移動するか、停留するかで、接触動作を分類している。また、頂点・辺・平面の組み合わせによる点・線・面接触をそれに対応付け、分類ごとに、移動あるいは停留する接触位置での並進と回転動作によって接触動作の変換や拘束条件を記述した。

CADにおける接触状態の記述

そのほか、機械製品の設計を扱うCADシステム分野でも、接触状態は重要な問題である。部品形状の幾何学的な誤差などにより部品間の接触状態が変化すると、機械の持つ運動機能や位置決め機能などに大きな影響を与え、機械の機能が変わってしまう。[鈴木 92] らは、より一般的に2つの形状が最も接近する状態として接触状態を求める方法を提案している。部品形状に対してポテンシャルエネルギーを定義し、それが最小となる状態を Simulated Annealing 法で解いて、接触状態を求めている。また、部品間の接触関係を言語で直接記述することによって各部品の持つ拘束状態を表現し、自由度を評価関数として分解・組立手順を推論するシステムに関する研究もある。

これらの研究を眺めると、対象物が多面体で、幾何学情報や作業のモデルが与えられ、摩擦や物体の柔らかさなどを無視する場合には、各種の接触・拘束状態の情報が計算できることがわかる。未知環境においては、幾何学情報を全てセンシングするのは大変であるため、幾何学情報を介さずに、何らかのセンサの情報から直接、接触・拘束状態を表現する手法が求められる。

1.2.6 マニピュレータの制御と拘束の表現

拘束と制御は深い関係にあり、制御側からみた拘束の表現が研究されてきた。環境と接触を伴う場合の制御方法としては、いわゆる力制御、ステイフネス制御、ダンピング制御、コンプライアンス制御、インピーダンス制御、位置と力のハイブリッド制御などが代表的である[Raibert 81][Whitney 87][Salisbury 80]。

位置と力のハイブリッド制御

[Mason 81][Mason 81] は、自然拘束と人工拘束の考え方を導入し、位置制御と力制御が相対する概念であることを示した。自然拘束とは与えられた作業によって自動的に定まる拘束された速度や力であり、人工拘束は人が指令できる速度や力のことである。対象とする作業は、理想C曲面と呼ぶロボッ

トの手先の存在し得る点からなる滑らかな超曲面で表され、そのC曲面に対して接線方向には運動が自由であり、法線方向には力を自由に加えることができる。また、一般に機構の多くは部品同士の間接触から構成され、自由度が3以下の場合が多く、低次対偶と呼ばれるが、低次対偶における拘束について6次元の速度ベクトルと力ベクトルの拘束式を用いて論じ、特に両側拘束の場合、拘束物体に許される速度と力は直交補空間を形成することを示している。[吉川 85][吉川 85]は、アームの手先が対象物から受ける拘束を、手先位置を示すベクトル空間内のある拘束超曲面の組によって表現し、手先拘束下のマニピュレータの運動と拘束力を定式化している。[高瀬 86][高瀬 75]は、回転と直線運動を空間的に6個配置させることによりほとんどの束縛運動を記述できるとしている。そして、拘束された物体の動きを記述する変数として、物体の運動自由度構造と合致した位置変数、力を指定するための力変数、残りの自由度に対する補助変数を導入し、制御のための物体とマニピュレータの運動方程式を立てている。

インピーダンス制御

[Hogan 85]のインピーダンス制御は、ロボットがバネ・マス・ダンパ系として振る舞うよう制御をかけるものである。これらの制御によって、ロボットがかたい状態から柔らかい状態まで任意の柔らかさを持ちながら、環境に馴染んで行う動作のことをコンプライアント動作と呼んでおり、様々な動作を実現するための研究がなされてきている。問題は、環境の状態によって如何にインピーダンスを設定するかが重要である。[Peshkin 90]は、作業中に生じる全ての接触状態を考慮し、対象物の幾何情報を基にした作業環境の不確実性に対処でき、誤差修正のための動作を発生するようなコンプライアンスを設定する手法を提案した。[松尾 93]らは、一定のコンプライアンス行列と運動指令によって完了する動作を基本動作とした組立作業戦略を提案し、そのための幾何情報に基づくコンプライアンスの設定法として、安定性、誤差修正特性、実現可能性をコンプライアンス行列の制約条件する線形計画を用いた手法を考えている。

1.3 ニューラルネットワークのロボットへの応用に関する過去の研究

1.3.1 ニューラルネットワークとロボットの研究

ニューラルネットワークの発展

ニューラルネットワークは、1943年にMcCullochとPitts[McCulloch 43]が神経細胞(neuron)のモデルを提案して以来、生理学や情報処理、工学等の立場からさまざまな研究が行われてきた。彼らのモデルは、神経細胞は受け取った刺激の総和があるしきい値をこえると興奮し、それ以下では興奮せず、その信号を次に伝えていくものであった。Hebbは、神経細胞が興奮したとき、刺激を伝えた入力シナプス結合の強度が増加するという神経回路モデルの学習則を唱えた。[Rosenblatt 58]はその原理に基づいて、パターンを学習識別する機械であるパーセプトロンを構成した。これは神経細胞が層状の構造をとっている。ところが、[Minsky 54]らの研究によって、パーセプトロンの意味や限界が明らかにされると、神経回路網の研究は下火になった。しかし、地道な研究も行われ、[中野 91]の連想記憶モデルであるアソシエーションの研究、[尾崎 89]の自己組織的認識システムコグニロン、甘利の神経回路網の性質や限界に関する理論的研究などがある。

1980年代に入ると再びニューラルネットワークの研究が盛んになる。その原因となった3つのモデルは、[Hopfield 85]によるホップフィールドのモデルとFahmannとHiltonのボルツマンマシンと[Rumelhart 86][Rumelhart 86]とHiltonのバックプロパゲーション法である。バックプロパゲーションの2つの方法は相互ネットワークであり、最適解を求めるような問題に適用されている。バックプロパゲーション(誤差逆伝播)法は、階層的な多層のネットワークにおいて、望ましい出力と実際のネットワークの出力の差の2乗和を評価関数として、それを最小にするよう最急降下法に基づいて各層間の結合係数を修正していく学習アルゴリズムである。誤差を信号の流れと逆方向に伝えることによって、全ての階層で学習可能になり、線形分離不可能なパターンも中間層が特徴抽出を行うことで学習できるようになった。このため、ニューラルネットワークの応用が各分野で急速に進んだ。また、現在でも、脳や神経のモデル、工学的なニューラルネットワークの新しい構造の研究や、学習法の改良、高速化、理論的な解明、ニューロコンピュータなどの研究が行われている。

ニューラルネットワークの応用

工学的に最も多く応用されているのは、バックプロパゲーション法を用いた階層的な多層ニューラルネットワークである。3層以上の多層ネットワークで中間層ユニットの数が十分多ければ、任意の連続な写像を近似できることが証明されている。基本的な能力としては、特徴抽出を伴うカテゴリーの分類(パターン認識)、非線形写像のマッピングや補間、近似、情報圧縮などが挙げられる。また、ネットワークは並列処理であり冗長でロバストである。特に、ニューラルネットワークは、ある入力信号を解釈・変換して出力信号を出すとき、変換手続きを明示的に記述する必要がないため、変換規則が不明で入出力関係が分かっている問題に有効である。これらの特徴を利用して様々な工学的応用がなされているが、実際には従来の方法では解けない問題を上手く解決した例はほとんどない。しかし、ロボットが何らかの認識や作業を行うときに、入出力の関係や閾値を詳細に記述・指定することなく、自動的に入出力関係をネットワークの重みに反映させて学習してくれる自己組織性は非常に便

利である。また、ネットワークは入力に対して可塑的であり、モデル等の更新が容易である。

ロボットに關係の深いニューラルネットワークの研究には次のようなものが挙げられる。

● パターン認識

多層のあるパターンを持った信号を入力とし、それらをいくつかのカテゴリに分類するものである。どんな量を入力し、何を出力にするかという枠組みを問題にとって適切なものに設定するのが重要である。パターンの特徴抽出やしきい値の調整などが、学習によって自動的に行われるのが利点である。ロボットでは、各種のセンサの出力、あるいは、それから求められる特徴量をニューラルネットワークの入力とする場合が多い。

ロボットの視覚から得られた画像から、対象物体の種類、状態や姿勢を得たり、3次元の物体のモデルマッチングによる認識を行ったりする研究が行われている [中野 89][浜 92][角所 90]。これらは、空間知覚の問題 [阪口 92] である。同様に、触覚センサの圧力分布から、対象物との接触状態や方向、及び対象物の形状、運動の様子などを得たりすることも可能である。超音波センサの距離情報から向きによらずに対象物を認識した研究 [吉澤 92] もある。

● 最適化問題

この場合は、ボルツマンマシンやポプフィールドのモデルが使われる。問題をエネルギー関数に置き換えて、エネルギーを最小化して解を求めるものである。ロボットの動作計画において、障害物を回避する最適な軌道を生成するのに使われることが多い。また、ロボットの視覚において、両眼のステレオ画像の最適な対応付けを行うのに適用する研究や、対象物体と特徴点間の対応付けをし、3次元の画像解釈を行って対象物を認識する研究 [加藤 92] もある。

● 運動制御

ニューラルネットワークの非線形写像を学習する能力を利用して、制御対象の動特性を学習させ、ロボットのサーボレベルや上位レベルの制御に適用する研究は数多い [福田 91] [橋田 90] [福田 90] [五味 90] [藪田 91] [川人 86] [遠坂 92] [Kung 90] [Narendra 90] [Miller 87]。マニピュレータの逆動力学的計算、同定によく用いられている。ニューラルネットワークを用いた制御系にはいくつかの種類がある。1つは、制御対象の逆ダイナミクスを実現するべく直接コントローラとして使用される。通常のコントローラ以外に、ニューラルネットワークを制御対象のダイナミクスの同定器として用い、コントローラを調整するものもある。また、制御対象の補償器として用い、ダイナミクスを補償して望ましいものにするタイプもある。学習は、サンプリングタイム毎にリアルタイムに行うものと1回の試行毎に運動パターンを学習していくものがある。ニューラルネットワークの入力としては、ダイナミクスを学習するために、時間微分した値を含んだ形、例えば、位置、速度、加速度などが与えられる。同様に、現在のデータと過去のサンプリングタイムのデータを入力とするものもある。また、ニューラルネットワーク内部に時間遅延要素があり、信号をフィードバックするリカレントネットワークも用いられている。

● 逆運動学 (座標変換)

ロボットのマニピュレータの手先の位置姿勢を与えて、各関節角を求める逆運動学は、解析的

に解けない場合も多い。その非線形な座標変換のマッピングをニューラルネットワークで実現しようというものである [山本 93]。特に、冗長マニピュレータには計算量の面で有用である [田中 91]。また、視覚特徴量を直接マニピュレータの制御に結び付けるビジュアルサーボへのニューラルネットワークの適用も考えられている [横山 90][久場 90][堤 91]。これは、視覚系の作業座標系からマニピュレータの関節座標系への変換を実現するものである。

ネットワークの構造としては、多層のニューラルネットワーク以外に、[Albus 75][Albus 75] の考案したCMAC(Cerebellar Model Arithmetic Controller) も用いられる。これは、基本的には表参照方式のネットワークで学習時間が短い、分解能を上げようとすると素子数が増える。

● 行動の決定や運動の獲得

ニューラルネットワークによってマクロな行動や運動の枠組みを得ようとする試みであり、運動制御と関連が深い。運動スキルの自己組織化によって、制御則を獲得させた報告がある [藪田 91][橋本 90]。また、内部環境モデルと実環境に比較的大きな誤差があっても制御の目標値を変え動作目標を達成する能力を上位スキルと定義し、そのニューラルネットワークによる獲得について検討している報告や人間の行う作業をニューラルネットワークに学習させようとする研究がある。自律型の知能ロボットに関して、センサの入力に応じて行動を変化させる行動パターンの学習の研究やマクロな指令に基づいて運動系列を自動的に創生する自己学習モデルの研究もなされている。

1.3.2 ニューラルネットワークによる接触・拘束状態の検出

ニューラルネットワークのパターン認識能力や非線形写像の学習能力を利用して、接触や拘束状態の検出に適用した研究もいくつかある。触覚センサのパターンをニューラルネットワークに認識させることによって接触状態を検出する手法は、画像パターンを認識させるのと同様であり、典型的である。

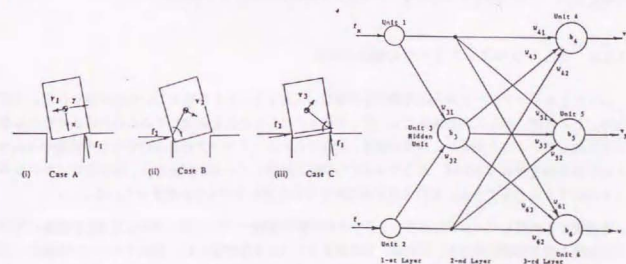


図 1.8: peg-in-hole 問題における非線形コンプライアンスとニューラルネットワーク [Asada 90]

[Asada 91][Asada 90][Asada 89]らは、組立作業においてニューラルネットワークを用いた非線形コンプライアンス制御法を提案している(図1.8)。ロボットの手先に加わる力ベクトルとそれに基づいて修正される手先の速度ベクトルの関係は、通常、ダンピング制御における粘性係数行列による線形形式で表されるが、面とりのない穴へのペダの挿入作業の例では、いくつかの接触状態を考慮すると線形関係は成り立たず、非線形なコンプライアンスが必要であることを示した。この非線形コンプライアンスを、力ベクトルを入力とし望ましい速度ベクトルを出力とする4層の階層型ニューラルネットワークで実現している。4層のうち、1番目の中間層は接触状態を検出する働きを行い、2番目の中間層は手先の動作の方向を決定し、出力層は線形関数により速度の大きさを決める役割をしている。学習は、実際の作業者の挿入作業時の力と速度を測定し、接触状態の変化点でのデータを抽出し、それを教師信号としている。

[Vukobratović 93][Borovac 93]らもほぼ同様の研究を行った。異なる点は、入力として、ロボットの指の柔らかな表面の下に取り付けられた分布型の圧力センサを用いていることである。操作物体の接触状態が変化すると、操作物体を柔軟に把持している指の表面の圧力分布が変化することを利用して、それぞれの点の圧力を直接ニューラルネットワークの入力にし、正しい操作方向を出力としている。また、3層のネットワークを用い、教師データは、人間の作業の計画やマスタースレープ型ハンドによる動作や作業のシミュレーションによって得ることができるとしている。[埴崎 89]らは、同じくピンを穴に挿入する作業を、2つの直列のニューラルネットワークで実現している。前半は、ピンの現在位置姿勢を入力してピンに加えるべき力・モーメントを出力し、後半はその力・モーメントを入力してピンが移動した後の位置姿勢を出力し、それらはまた前半の入力に戻るような仕組みである。このネットワークを人間が考えた教師信号で学習させることにより、挿入作業の世界モデルをネットワーク内に構築している。

これらの研究は、中間表現を設けて、各々の接触状態の検出とそれに対応する動作計画を独立して行うのではなく、認識から行動へ連続的に1つのニューラルネットワークで実現した点が大きな特徴である。しかし、予め作業を決めて、その作業に対する教師データを何らかの方法で作成する必要がある。また、中間表現をとっていないので教えた範疇の作業しかできない。

1.3.3 ニューラルネットワークの重みの解釈

ニューラルネットワークは入出力関係を学習し、入力を与えたとき望ましい出力が得られるようになる。ところが、ある入出力関係について、どのようにしてそれを実現しているのかはよくわからないまま使っていることが多い。学習の結果、形成されたネットワークの重みの分布や中間層のユニットの役割分担を解析できれば、どうやってその問題を解決しているかが分かり、明示的なアルゴリズムを抽出することもできる。そのような研究が少ないながらもいくつか始まっている。

本研究でも利用したニューラルネットワークの学習方法の一つに、[石川 92][石川 90]が提案している忘却付き構造学習法がある。これは、階層的ネットワークを対象とし、訓練パターンに内在する規則性を発見することを目的としている。ネットワークのリンクの重みに忘却を導入して不要なリンクを消滅させ、規則性を重みの骨格構造として出現させるため、中間層や重みの解釈が可能で、中間層のユニットの個数などを予め決めたりする必要がない。このような中間層のユニットの個数を自動的

に増加させたり、削減したりする学習法はその他にもいろいろと研究されている。また、そのようにしてリンクを単純化した小規模のモジュールを多数合わせ、学習済みのモジュールの結果を自動的に利用して、大規模なモジュール型ニューラルネットワークを構成することも研究している。また、忘却付き学習法に加えて、ネットワークの重みの削除が出力関数に与える影響を調べて、それが小さい重みを削除する方法をとって、ネットワークの汎化能力を向上させた、[小谷 93]らの研究もある。

[入江 90]らは、多層ネットワークに逆誤差伝播学習を適用したモデルの内部表現の解釈について、内部表現が本当に中間層に形成されたのか、単に学習例をコンパクトな形で記憶しているだけか、それとも、人間が信号の表現の意味を解釈したのか、という点を議論している。そして内部表現が中間層に形成されることを示すのに、入力分布が入力信号ベクトルの次元に比べてはるかに少ない自由度で表現でき、その自由度に対応する大規模な座標系が存在することを挙げ、それをシミュレーションにより示している。[渡辺 93]らは、内部表現として中間層ユニットの出力値に着目し、内部表現の特性を表す指標を導入している。それらは、中間層ユニットのロバスト性、学習パターン間の類似度、反応性を表す3つの指標であり、汎化能力と密接に関連していることを示した。

具体的なニューラルネットワークの応用例について、中間層の解釈を行った報告もある。[立石 89]らは、バックプロパゲーション法による階層型ニューラルネットワークを手書き数字認識に応用し、手書き数字のどの特徴を抽出し、出力層に伝えているかという中間層の機能を考察している。そのとき中間層の出力関数として、通常のシグモイド関数ではなく双曲線関数を用い、重みの絶対値の制限を導入して冗長度を減らして、中間層の働きの解釈を容易にした。

1.4 研究の目的と意義

知能ロボットが作業環境に適応した行動を行うのに最も重要なことの一つは、センシングによる作業環境や対象物の認識である。ここでは、工場のように作業環境がロボット用に整備され、対象物の状況がほとんど既知である場合ではなく、屋外や家庭、宇宙や特殊環境などの未知の情報が多い状況を想定する。変動する複雑な環境や未知の環境では、対象物のモデルを予めすべて与えておくことは不可能であり、ロボット自身が外界センサを能動的に使って、環境や対象物に関する情報を獲得する必要がある。

獲得した情報に基づいてロボットは何らかの認識を行って行動を起こすことになるが、2つの両極端の考え方として、センサ情報からロボットの行動計画まで、全てに渡ってモデルに基づいた処理を行うモデルベーストロボティクスの考え方と、明示的なモデルを持たないで、センサ情報から直接ロボットの反射的な行動を生成するサブサンプリングアーキテクチャに代表される考え方がある。前者は、実環境のモデル化ができれば、処理が明らかで様々な動作計画を行うのに有用であるが、センサの情報処理のような誤差が多く、モデルが実際の状況と合致しづらい場合にはうまく動作しない。後者は、比較的単純なアルゴリズムと基本的な反射行動である程度の一連の動作を実現できるが、途中の処理が明らかでないため、同じ状況（センサ入力）に対して目的により別の動作を行わせるような動作計画は困難である。そこで、本研究では、モデル化が複雑で誤差を含む下位レベルのセンサ処理はモデルによらず直接的な処理を行い、その結果を明示的に用いてモデルに基づく高次のロボットの動作計画を行う立場をとる。つまり、

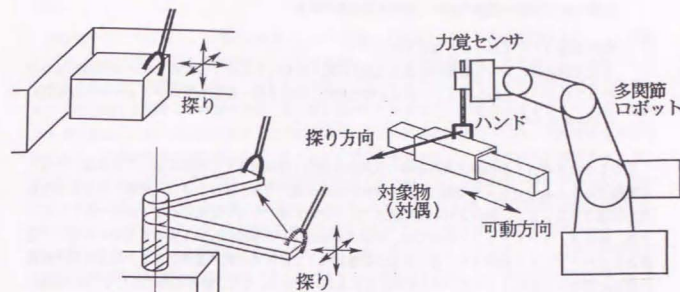
- (1) 環境や対象物の情報をどのような能動的探索によって収集するか
- (2) 獲得した外界センサの情報をどのように処理、認識するか
- (3) 認識結果をどのように表現し、動作計画と結び付けるか

が重要な問題となる。

本研究では、このような環境や対象物の情報をセンシングし、環境を認識する過程として、

ロボットが把持した作業対象物の運動拘束の認識

を考える。例えば、図 1.9 のように把持物体が環境と点、線、あるいは面接触していたり、複数の面で運動を拘束されていたり、クランクのようにある軸回りの運動しか許されていなかったりする状態を検出することである。この問題は、ロボットが環境や対象物と相互作用するときには避けられないものである。環境や対象物の幾何学形状が与えられたり、対象とする作業が特定されていて操作物体の状態が数種類に限定されたりする場合には、対象物の接触・拘束状態を検出・識別したり、可動方向などを計算する研究は数多くなされており、成果も上がっている。特に対象物を剛体で多面体とし、摩擦力を考慮しない場合は、一般的な解析や検出方法が提案されている。組立作業の一つとして、ペグを穴に挿入する作業については、様々な解析や各段階での接触状態の検出法などがある。しかし、ロボットが直接把持した未知の接触・拘束状態を検出する研究はほとんど見当たらない。



例えば、把持物体が環境と点、線、あるいは面接触していたり、複数の面で運動を拘束されていたり、クランクのようにある軸回りの運動しか許されていなかったりするような運動拘束を検出する。

図 1.9: 幾何学的な拘束状態の認識

未知の接触・拘束状態の検出が必要な作業の例には、次のようなものがある。

- ホームロボット
家庭でホームロボットが作業する場合、人間が使用している設備や道具を操作することが多い。それらの拘束状態を全て予め与えておくことは大変であり、変動することも多い。例えば、ロボットが通過するドアの開く方向（引き戸？回転？）やノブの回し方、物を収納する戸棚の戸や引き出しの開閉、水道やガスの栓の開閉、物の収納・箱詰め・取り出し、などが挙げられる。これらの拘束運動を伴う作業では、拘束の状態を検出して作業計画を立てる必要がある。従来の研究では、例えば、画像等を用いてドアを検出した場合、その拘束運動（動く方向）はそのドアの知識として予め与えてしまっていることが多い。
- 保守・点検作業、組立作業
プラントの保守・点検作業の自動化は重要であり、画像や超音波などのセンサによる監視等が行われている。それに加えて、可動部分が設計通りの運動をするかどうかの点検や、逆に、固定部分がしっかり固定されているかどうかの点検も重要である。異常がある場合には通常と拘束状態が異なる。また、組立作業や建築作業においても、作業の各段階でその組立が完全に行われたかどうかの確認として拘束状態を検出する必要がある。
- 分解作業
近年、環境やゴミの問題から資源の再利用の重要性が高まっている。廃棄された家電製品などの工業製品を分解・解体し、部品を分別し、再利用する技術が求められている。分解する製品の種類が決まっている場合には、それに応じた分解機械を作ればよいが、多種多様な製品を分解し、部品を回収する作業にはロボットが適している。分解作業では、ねじや止め具や部品の

位置や拘束状態の種類を検出・識別する必要がある。

- 拘束運動のコンピュータへの取り込み
3次元測定器によって物体の形状をCADに取り込むことは良く行われるが、仮想現実感などのコンピュータ上のシミュレーションのために、拘束状態（運動）をコンピュータに取り込むことが考えられる。

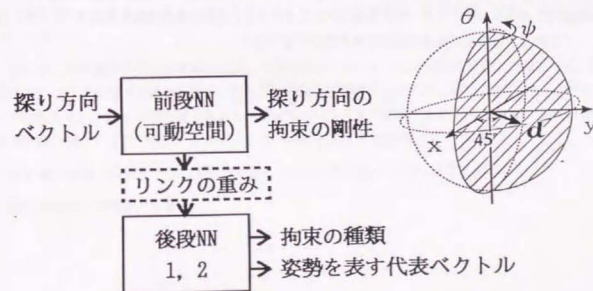
そのようなロボットが操作する対象物の大局的な接触・拘束状態を明示的に検出できれば、それらが既知であるときのロボットの動作計画の研究の成果が適用でき、様々な目的の接触や拘束を伴う作業を実現できる。まず、検出された拘束状態で行う作業に適した力や位置の制御方法をとることができる。例えば、インピーダンス制御では、仮想バネ定数などの各係数行列を定める基準になるし、位置と力のハイブリッド制御では、自由度毎の位置制御と力制御の選択の基準になる。拘束状態と作業に応じた記号レベルのフィードバックがかけられるわけである。また、動作目的に応じて、その接触・拘束を維持したり、離れたりするなどの動作を効率的に計画することが可能になる。さらに、操作物体と環境との間の点・線・面接触などの個別の情報を検出するセンシング計画が立てられ、個々のモデルに基づく詳細な接触情報を得ることもつながる。

さて、把持物体の拘束状態を検出するための情報源として、ロボットの視覚も1つの候補である。視覚により対象物や環境の幾何学的形状が得られれば、接触・拘束状態が直接計算できたり、それらをいくつかの状態に限定できたりする。しかし、把持物体は環境と接しており、物体と環境のそれぞれの形状の分離が困難であったり、形状の一部が隠されていたり、把持物体の下側や内側に拘束の機構がある場合も多く、視覚の情報だけでは不十分である。拘束状態の検出には視覚情報は本質的ではないといえる。物体が拘束されたら接触したりしている場合に直接生じている物理量は、力やモーメントおよび微小な変位量である。人間も、把持したものの動きを調べるときは、手に加わる力や物がずれる運動の感覚を利用して思うように思われる。例えば、ドアの開く方向や鍵を回す方向がよく分からないと、動きそうな方向に引っ張りたり押し回したりして動く方向を見つける。暗い部屋の中ではこれらの動作は顕著であり、無意識のうちに手を能動的に動かしてものに触り、それがどのように動くか調べたりする。人間は、視覚を補助するように、また、視覚がなくても、手探りによる触知覚を用いて器用な作業をすることができる。

このように考えると、ロボットが把持物体の拘束状態を検出するためには、手先の力や変位のセンシングが必須である。このとき、対象物を把持したまま受動的にセンシングしても有用な情報は取得できない。ある方向に力やモーメントを加え、対象物が動くかどうか調べるアクティブセンシング、いわゆる「探り動作」を繰り返す必要がある。集取された手先（把持物体）の動きや反力の情報から大局的な運動拘束を認識する。得られた実環境のデータには、本来の拘束の情報だけでなく、ガタや摩擦等の影響も含まれており、これらを取り除いた上で拘束の情報を取り扱う必要がある。また、拘束の情報には、拘束の種類の情報と姿勢の情報が含まれている。環境と対象物の形状や機構に起因して拘束状態が決定するので、センシング情報からモデルとして幾何学形状を構築していくことが考えられるが、力や変位の情報から直接に幾何学形状を求めるのは難しい。拘束状態は、対象物が空間的にどう変位できるか、どう力を加えられるかという可動空間の問題、広く言えば、空間の検出の問題である。そこで、変位や力から、ロボットの動作計画に利用しやすいように可動空間を表現する方が

よい。

本研究では、下位のセンサ情報の処理とモデルに基づく高次のロボットの動作計画をつなぐ手法として、拘束状態すなわち可動空間をニューラルネットワークで表現することを考える。拘束としては平面3自由度の運動拘束を扱い、それらを分類してニューラルネットワークに学習させることにより拘束の認識を実現する。各種の拘束状態を場合分けしてモデル化し、閾値や個々のルール等で識別するのではなく、統一的な表現をする方が効率的だと考えられる。理由の一つは、把持物体が多自由度であり、拘束が剛体どうしではなく、柔らかさやガタが存在するような場合にも対処できるようにすることである。従来の研究では、可動空間は剛体がある方向に動かないかによって決まる。許容される変位の集合をベクトル空間で表現していることが多い。ところが、同じ種類の拘束状態でも、弾性体によって拘束されている場合やガタがある場合にはそれらの方向へも変位が許される。同じ1mm動いたときでも、その意味は他の方向がどれくらい動かないかによって決まる。大局的な拘束状態の種類は絶対的な閾値などで区別されるものではなく、空間の方向に対して相対的なものであり、多分に主観的でもある。このようなものは閾値の設定は困難である。また、「探り動作」で得られる力や変位の情報は、空間の離散的な方向のデータであり、ニューラルネットワークの写像の補間能力を使えば、連続的なデータとして表せる。その連続的に表された可動空間は、ニューラルネットワークの有限個のリンクの重みによって実現されているわけであり、その重みに情報圧縮されている。そのリンクの重みから、拘束状態が検出できるはずである。逆に言えば、そのネットワークのリンクの重みを保存しておけば、可動空間を再生でき、ある程度、元の変位や力のデータを再現できる。再生した可動空間からロボットの動作計画を立てることも可能である。



可動空間を反映している学習後の前段ニューラルネットワークのリンクの重みの分布パターンを後段ニューラルネットワークで認識する点が大きな特徴である。

図 1.10: 平面 3 自由度の拘束を認識する 2 段の階層的ニューラルネットワーク

拘束をニューラルネットワークの重みから検出するときの問題は、通常のバックプロパゲーション法では入出力関係が同じでも重みの分布は1つには決まらないことである。しかし、本研究では、忘却付き構造学習法 [石川 92][石川 90] を用いて、同じ可動空間に対する重みの分布をできるだけ少ないパターンに固定する。さらに、重みの分布のパターンから、拘束状態を検出するのに、ニューラルネットワークのリンクの重みを入力とする、もう一つの階層ニューラルネットワークを使用して、2段階構造にする (図 1.10)。重みの分布のパターン認識に適していると考えられるからである。

このようにすると、探り動作時の変位と力の情報を入力して、動作計画等に有用な拘束の種類や姿勢を出力するようなニューラルネットワークの形態が期待される。発見した可動方向に動くような反射的な動作を直接出力しないのは、認識した拘束の種類や姿勢の情報をを用いて、行動目的によりロボットの動作を変える方が柔軟性が高いからである。また、運動拘束の検出は、必ずしもニューラルネットワークでなければ絶対に検出できない問題ではなく、解析的な手法でも実現できる。本研究では、比較として3次元1自由度の拘束の検出を解析的なアルゴリズムで行う。

以上、本研究では、

未知環境において、ロボットが把持している作業対象物の運動拘束を検出・認識する

ことを目的とする。

そのため、能動的な「探り動作」を行って、力と変位の情報を収集し、それらの情報からニューラルネットワークで可動空間を表現し、発生したニューラルネットワークのリンクの重みから拘束の種類・姿勢等を検出する。平面3自由度の拘束を扱い、それらを分類して、拘束の種類や姿勢をこの2段階の階層的ニューラルネットワークに学習させることにより、運動拘束の認識を実現する。また、比較として3次元1自由度の拘束の検出を解析的な手法で行う。

1.5 本論文の構成

次章以降の、本論文の構成について述べる。

第2章「ロボットの探り動作と運動拘束」では、作業対象物の運動拘束を検出・認識する方針を示す。探り動作で得られるロボットの手先の変位と力のパターンが拘束状態によってどのように変わるかをガタや摩擦力等の影響を含めて考察する。

第3章「探り動作の実験システム」では、拘束の認識のためのロボットの探り動作の実験システムの構成と探り動作時のロボットの制御について述べる。

第4章「実際の探り動作によって得られる力と変位のパターン」では、拘束状態を変えて探り動作の実験を行い、そのときに得られる力と変位のパターンを解析する。

第5章「解析的手法による3次元1自由度の拘束の検出」では、ニューラルネットワークを用いた拘束の認識を行う前に、比較として、解析的手法により3次元1自由度の拘束の検出を行う。検出手法を構築し、その過程で実用性や問題点を明らかにする。

第6章「入出力画像の種類を認識する2段階の階層的ニューラルネットワーク」においては、本研究で提案する、入出力画像の種類を検出する前後2個のネットワークからなる2段階の階層的ニューラルネットワークの構造と学習方法について述べる。また、基礎的な性質を明らかにするために、論理関数を識別する問題に適用し、学習過程、挙動、認識結果等について論じる。

第7章「平面3自由度の拘束と2段階の階層的ニューラルネットワークによる認識」では、本研究で扱った平面3自由度の拘束について分類し、可動空間を示した。また、それらの拘束を認識できる2段階の階層的ニューラルネットワークを提案する。そのときの学習法、学習過程、挙動、認識結果等について述べる。

第8章「探り動作のデータを用いたニューラルネットワークによる拘束の認識」では、可動空間を理想的に1/0で表現するのではなく、探り方向の拘束の剛性で表現して、実際の変位と力のパターンから取得して、それを2段階の階層的ニューラルネットワークで認識することを考える。実際の可動空間の学習に適した2段階のニューラルネットワークの改良とその結果について述べる。

第9章「結論と展望」では、本研究を通して得られた結論と今後の展望を述べる。

以上が本論文の構成である。

第2章 ロボットの探り動作と運動拘束

2.1 はじめに

未知環境において、ロボットが把持している作業対象物の運動拘束を検出・認識する方針を示す。そのためには、能動的なロボットの「探り動作」を行って拘束を検出するための情報を取得する必要がある。この探り動作の方法や制御について論じる。

探り動作で得る情報はロボットの手先の変位と力のパターンである。このパターンが運動拘束の状態によってどのように変わるかをガタや摩擦力等の影響を含めて考察する。

2.2 探り動作による拘束の検出の方針

未知環境において、ロボットが把持している作業対象物の運動拘束を検出・認識する方針を示す。そのために、能動的なロボットの「探り動作」を行って、拘束を検出するための情報を取得する。また、ガタや摩擦や柔らかさを含んだ実環境を対象とする。

人間の手探りの動作等を参考にすると検出の方針は次のようになる。

- (1) ロボットが対象物をしっかりと把握した状態で、ある方向に比較的小さい力やモーメントを加えて（探り動作）その時のロボットの手先の変位と力の情報を取得する。
- (2) その力と変位のパターンから、その方向が動く方向（可動）か、動かない方向（不動）か、判断する。あるいは、その「探り方向の拘束の剛性」（変位と力から求めたバネ定数）を計算する。このとき、ガタや摩擦などの影響を考えた処理をする。
- (3) このような探り動作を、様々な方向に対して行っていく。そうすると、探り動作を行った方向ベクトルとその方向が動くかどうか（あるいはバネ定数）の情報の組が複数個得られる。
- (4) 複数の方向の探り動作で得られた情報から、何らかの識別手法を用いて、作業対象物の全体の拘束の種類や姿勢等を検出する。本研究では、解析的に求める方法とニューラルネットワークによる方法を用いる。

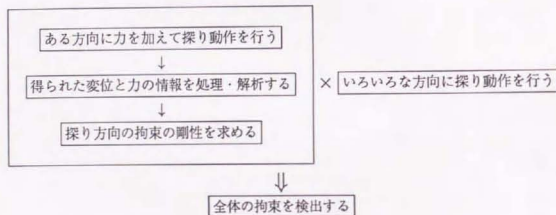


図 2.1: 探り動作の手順

2.3 探り動作

2.3.1 探り動作に必要なロボットアームの制御

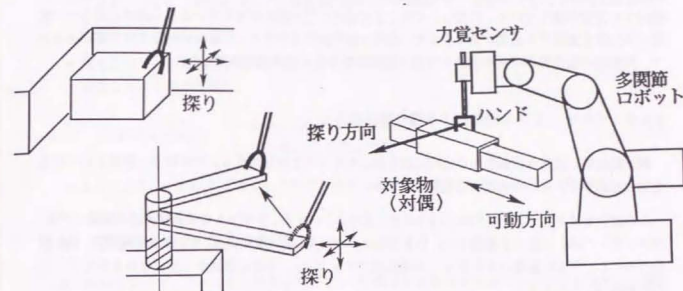


図 2.2: 探り動作

探り動作では、ロボットが把握した作業対象物にいろいろな方向の力やモーメントを加えて微小に動かす動作を行う。対象物の形状や幾何学的拘束に関する情報などが未知であるため、位置制御で動かそうとしても、方向によっては対象物やロボットに過大な力が生じる危険がある。剛性の低いロボットアームやハンドを用いると過大な力の発生は防げるが、内界センサだけで手先の位置を精度良く検出することできない。過大な力の発生を防ぎロボットの手先の位置や力を精度良く測定するためには、探り動作を行う時には力制御が本質的に必要である。

探り動作に適した力制御について検討すると次のようになる。

- 純粋な力制御
探り方向に対して、力の目標値を増減することにより、探り動作を行う。自由な方向である場合には、その方向に力を加えるようにしても反力が発生しないため暴走してしまう恐れがある。仮に動く範囲を制限したとしても、自由な方向に動く速度は制御できない。また、ゲインが大きいとガタがある場合には不安定になる恐れもある。
- 位置と力のハイブリッド制御
位置制御の方向と力制御の方向を選択しなければならないが、対象物の自由度やその拘束状態が分らないと位置と力の制御の方向の分配をすることができない。
- インピーダンス（コンプライアンス）制御
インピーダンス制御の場合には、仮想剛性行列を適当に設定することで仮想的に対象物がバネ

で支持されたような柔らかい動きができる。制御をかける段階では自由な運動と拘束された運動を区別する必要がある。また、力を加えたり、位置を動かしたりする動作は、仮想的なバネの平衡点（コンプライアンスセンタ）を移動することにより簡単に行うことができる。

拘束状態が未知の作業対象物に対して探り動作を行う場合には、インピーダンス制御により柔らかく把持する方法が適している。ただし、そのときに過大な力や変位が生じないように適当に柔らかい仮想バネ定数を選定する必要がある。また、動的に探り動作を行うと、対象物の慣性力の影響があるので、対象物の慣性等が不明の場合には探り動作は準静的に動かす必要がある。

2.3.2 インピーダンス制御による探り動作の方法

探り動作は、並進や回転のある方向に力を加えてみることから始まる。その結果、拘束されている方向には動かず、自由な方向には変位する。

この動作をインピーダンス制御で行うには、次のようにする。ロボットの手先は次式の関係（バネ・マス・ダンパ系）に従って動作する。作業座標系において、 K_d :剛性行列、 D_d :減衰係数行列、 M_d :慣性行列とし、手先の位置ベクトル x 、目標位置ベクトル x_d 、手先に環境から加えられる外力 F 、その目標値 F_d とすると、

$$M_d \ddot{x} + D_d \dot{x}_e + K_d x_e = F - F_d \quad (2.1)$$

$$x_e = x - x_d \quad (2.2)$$

ある方向に探り動作を行う時は力の目標値 F_d を探り方向に準静的に増減する。これは、仮想的なばねの平衡点 x_d を探り方向にずらしているのに等しい。準静的な動きを考えると、 $\ddot{x} = \dot{x}_e = 0$ として、

$$K_e x_e = F - F_d \quad (2.3)$$

に従ってロボットの手先が動くことになる。

2.3.3 インピーダンス制御と拘束による手先の変位と反力

3つに場合分けしてインピーダンス制御されたロボットの手先の変位と反力は次のようになる。

- 探り方向が拘束されている場合
変位は生じず、

$$x_e = 0, \quad F = F_d \quad (2.4)$$

となる。つまり、手先に加わる反力 F は目標値 F_d に追従して増減する。

- 探り方向が自由な方向である場合
反力がなく、

$$F = 0, \quad x_e = -K_e^{-1} F_d \quad (2.5)$$

となる。つまり、変位 x_e が力の目標値 F_d と逆向きに増減する。変位の大きさは設定した仮想バネ定数による。

- 探り方向がバネ定数 K_e の弾性を持っている場合
変位と力はそれに比例し、

$$F = -K_e x_e, \quad \begin{cases} x_e = -(K_d + K_e)^{-1} F_d \\ F = K_e (K_d + K_e)^{-1} F_d \end{cases} \quad (2.6)$$

となる。 $K_e = 0, \infty$ とすると、それぞれ自由な場合と拘束されている場合に対応する。

2.3.4 探りの大きさの設定について

探り動作において、どのくらいの力を加えて、どの程度の範囲を探るかという問題は非常に主観的である。100gfの力では動かなくても、1kgfの力を加えれば動く場合もあるし、1mm 動いたとしても、10mm 動くかどうかはわからない。これは、対象としている作業や環境の規模による。本研究では、日常の人間の比較的小さな対象に対する探り動作を目安とし、その程度の力と変位が実現できるような探り動作のパラメータを設定する。

2.4 探り方向の拘束状態と変位と力の関係

探り動作を行った時に観測する情報は、ロボットの手先の変位と力である。探り方向の拘束の状態によって、探り動作の時の変位と反力がどのように変化するか論じる。

2.4.1 探り方向の拘束状態と拘束の剛性

作業対象物が剛体である場合には、探り動作を行った時の把握した対象物の動き（探り方向の拘束状態）は、基本的には次の3つである。

- 自由に動く場合
変位が大きく生じ、反力は生じない。
- 拘束されて動かない場合
変位は生じず、逆向きに大きな反力を生じる。
- 可動方向に沿って探り方向とずれて動く場合
探り方向が可動方向に一致していないと対象物が動かないわけではなく、方向がずれていても力の方向が摩擦円錐の外側であれば、拘束（可動方向）に沿って移動する。移動方向と垂直方向に拘束から探り方向と移動方向の差に応じた反力を受ける。探り方向の情報だけでなく、沿って動く方向の情報得られる点が重要である。

対象物や環境が剛体ではなく柔らかさがあると、これらの3つの中間のような状態が存在する。ただし、ロボットの柔らかさと対象物や環境の柔らかさは区別できない。柔らかさを含めて連続的に扱うには、探り方向の拘束の剛性（バネ定数）を考えればよい。自由である場合はバネ定数が0であり、完全に拘束されている場合にはバネ定数は ∞ である。

2.4.2 ガタや摩擦等がある時の拘束状態と変位と力の関係

実際の環境では本来の拘束だけでなく、それらに影響を与え、検出を妨げる次のようなものがある（図2.3）。

- ガタ
ガタの中には、対象物と環境の間のガタもあれば、ロボットの機構や把握部分のガタもあり、それらの区別はできない。ガタによる動きと本来の拘束による動きを区別する必要がある。
- 摩擦力
動く方向と逆向きに摩擦力が生じる。摩擦力と拘束による反力を区別する必要がある。
- その他の一定力
重力やバネの予圧等の一定の力がかかっている場合がある。これらは本来の拘束ではない。摩

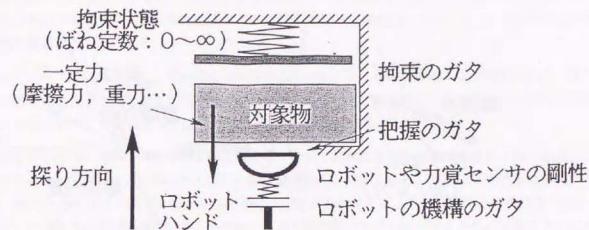


図 2.3: 探り方向の拘束の状態

摩擦力は動きを妨げる方向に発生するが、重力は動きによらず一定の方向に一定の力が生じる点で大きく異なる。そのため、探り方向の重力の成分を f_g 、一定の摩擦力を f_f とすると、探り方向に生じる力 f_1 は

$$f_1 = f_g + f_f \quad (2.7)$$

であるが、その逆向きに探り動作を行うと、生じる力 f_2 は

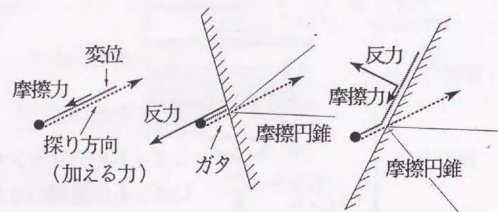
$$f_2 = f_g - f_f \quad (2.8)$$

となり、逆に重力と一定の摩擦力を分離することも可能である。なお、動的に動かす慣性力が生じるが、本研究では準静的な探り動作を扱って、慣性力は考慮しない。

ガタや摩擦が存在する場合に探り動作を行った時の力と変位の様子は次のようになる。また、それを図2.4に示す。

- 自由に動く場合
探りの力を加えても最大静止摩擦力に達するまでは動かないが、それを越えると、変位が生じ始める。そして逆向きに動摩擦力がかかる。摩擦力が一定でない場合には、Stick-Slip現象が起こりやすい。
- 拘束されて動かない場合
最初、ガタの範囲を動いた後、拘束に衝突し、加える力が摩擦円錐の内側にあるため動けず、逆向きに大きな反力を生じる。柔らかさがある場合には力とともに変位が増加する。
- 可動方向に沿って探り方向とずれて動く場合
最初、ガタの範囲を動いた後、拘束に衝突し、加える力が摩擦円錐の外側にあり、拘束に沿っ

て移動する。移動方向と逆向きに摩擦力、垂直方向に拘束から探り方向と移動方向の差に応じた反力を受ける。



a. 動ける方向 b. 動けない方向 c. 拘束に沿う方向

図 2.4: 探り方向の拘束の状態による変位と反力の関係

このように、探り方向の本来の拘束の情報（例えば、探り方向の拘束の剛性）を得るには、得られた変位と力の関係からガタや摩擦等の影響を除く必要がある。あるいは、それらを含んでいても、おおまかに拘束の種類が識別できるような手法が必要となる。

2.5 探り動作の戦略の考察

ある方向に力を加えてみる探り動作を1回行っただけでは、当然、拘束全体は検出できない。探り動作をいろいろな方向に関して行う必要がある。本研究では、基本的には均一にいくつかの方向に対して探り動作を行う。

しかし、アクティブセンシングの考え方を探り入れれば、より効率的な探り方が可能である。探り動作を行った方向が自由に動く方向や拘束されて動かない方向である場合にはその方向の拘束の情報だけしか得られず、それによって探り方向を効率的に変化させることはできない。

可動方向に沿って探り方向とずれて動く場合は、沿って動いた方向を利用して探る方向を動的に計画できる。沿って動いた方向は、動く方向と動かない方向の境界である。探り動作を行った方向から沿って動いた方向までの間は、拘束された方向と見做すことができ、その間は探る必要はなくなる。一般によく存在する拘束の場合には、他の動く方向と動かない方向の境界等もそれに垂直な方向に存在することが多い。その後は均一に探るのではなく、沿って動いた方向を基準に、例えば垂直な方向にいくつか探り動作を行っていくのがより効率的であると考えられる。そして、複数の沿って動いた方向が発見できれば、かなり探り方向を限定することができる。

2.6 まとめ

未知環境において、ロボットが把持している作業対象物の運動拘束を検出・認識する方針を示した。ある方向に力やモーメントを加えて「探り動作」を行ってそのときのロボットの手先の変位と力のパターンを収集する。そのパターンから探り方向の拘束の剛性等を計算する。このような探り動作を繰り返して、作業対象物全体の拘束の種類や姿勢を検出することになる。

そして、探り動作を行う時の力の制御法としてインピーダンス制御が適していることを論じた。また、そのときの拘束状態によるロボットの手先の動きについて説明した。

さらに、探り動作で得る手先の変位と力のパターンと運動拘束の関係を考察した。可動方向に沿って動く場合が重要であることを述べた。また、ガタや摩擦力等の影響について検討し、それらを検出して除去した上で、本来の拘束を識別する必要があることを述べた。

第3章 探り動作の実験システム

3.1 はじめに

探り動作の実験のために構成した実験システムについて述べる。基本的には、

- 探り動作をする多自由度ロボットアーム
- 対象物を把握するハンド
- 制御および拘束状態のセンシングに用いる力センサ類
- インターフェース
- 制御や動作計画等の計算を行うコンピュータ

から成り立っている。

また、探り動作に用いる制御則として、インピーダンス制御の計算を記述する。

3.2 実験システム

3.2.1 システム概要

探り動作の実験のために構成したシステムの全体の写真と構成を図 3.2, 3.1 に示す。

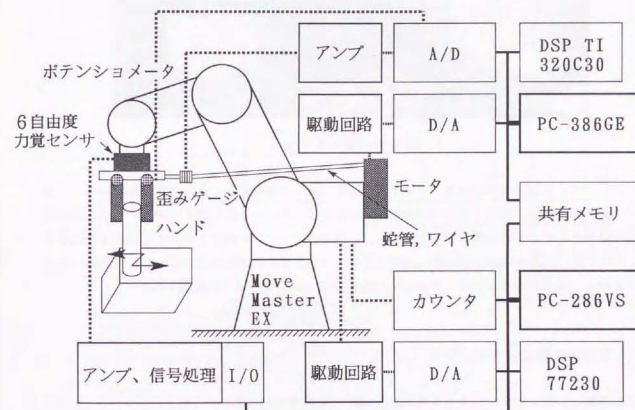


図 3.1: 実験システムの全体

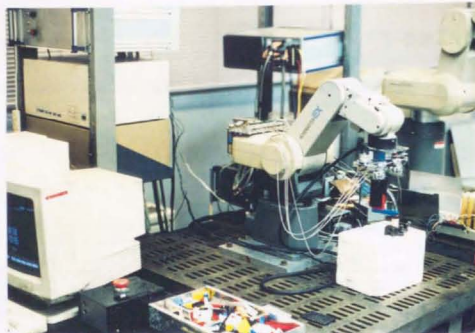


図 3.2: 実験システム

ハンドをつけた多関節のロボットアームを中心に、力センサ等を配し、それぞれ、駆動ユニットや各種インターフェースを通して2台のコンピュータと接続されている。1台はロボットアームの制御とそれに関係するセンサの処理等を行っている。もう1台はハンドの制御とロボットの動作計画等を行う。それぞれ、計算の高速化のため、DSPを用いている。また、2台のコンピュータの通信は共有メモリを介して高速に行っている。システムの各部分については、以下に説明する。

3.2.2 5自由度多関節ロボットアーム

物体を把握して探り動作を行わせるために、(株)三菱電機製の5自由度多関節型ロボット MOVE MASTER EX を使用した。仕様を表 3.1 に示す。

このロボットは5自由度であるため、そのままでは3次元の任意の位置姿勢をとることはできない。そこで、6自由度が必要な実験の場合には、2本指ハンドを取り付けずに、力覚センサの上に図 3.4 のようなアクチュエータ無し、エンコーダ (100p/r) 付きの回転の自由な関節を付加して6自由度を確保している。

ロボットの駆動ユニットは本来のものを使用せず、コンピュータから直接、電流制御できるように自作したものを使用した。コンピュータから D/A 変換ボードを通して指令電圧を出力するとそれに比例して関節の DC モータが電流制御されるようにした。DC モータの電流制御は、BURR-BROWN のパワーオペアンプ OPA541AM ($\pm 35V, 3A$) を用いている。また、アーム保持のための電磁ブレーキの開閉、安全のための異常検出や非常停止の機能をつけている。ロボットの関節角はエンコーダのパルスをカウンタボードでカウントしている。

表 3.1: 多関節ロボットアームの仕様 [マニュアルより]

項 目	仕 様	備 考
構 造	5 自由度 垂直多関節形	
動 作 範 囲	ウエイスト回転 300° (MAX 120°/sec)	J1 軸
	ショルダ回転 130° (MAX 72°/sec)	J2 軸
	エルボ回転 110° (MAX 109°/sec)	J3 軸
	リストピッチ ±90° (MAX 100°/sec)	J4 軸
	リストロール ±180° (MAX 165°/sec)	J5 軸
アーム	アッパーアーム 250mm	
長さ	フォアアーム 160mm	
可搬重量	max. 1.2kgf (ハンド重量を含む)	マニピュレータフェースより75mm位置(重心)
最大合成速度	1000mm/sec (リストツール面)	図1.3.4 P点の速度
位置繰り返し精度	±0.3mm (リストツール面のロール中心)	図1.3.4 P点の精度
駆動方式	DCサーボモータによる電気サーボ駆動	
本体重量	約 19kgf	
モータ容量	J1~J3 軸 30W, J4, J5 軸 11W	

項 目	単 位	J ₁	J ₂	J ₃	J ₄	J ₅
減 速 比	—	100	165.3	110	180	110
エンコーダ分解能	パルス/1回転	200	200	200	96	96

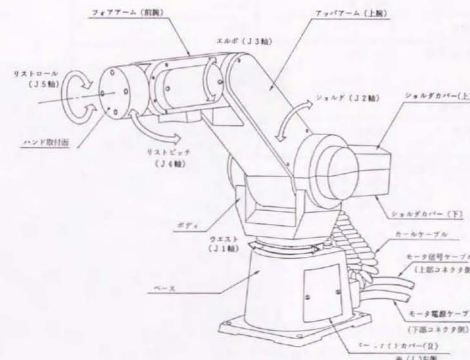
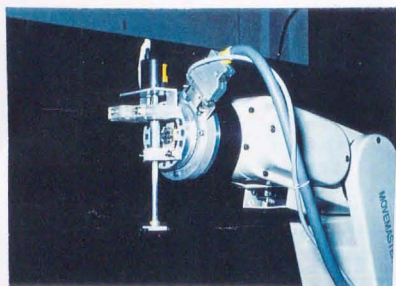


図 3.3: 多関節ロボットアームの概観



ロボットは5自由度であるため、6自由度が必要な実験の場合には、力覚センサの上にアクチュエータ無し、エンコーダ付きの回転関節を付加した。

図 3.4: 6自由度が必要な実験のための第6軸関節部

表 3.2: 6軸力覚センサの仕様 [マニュアルより]

外形寸法	外形 D 高さ H	54mm 31.8mm
質量		180g
定格荷重	F_x, F_y F_z H_x, H_y, H_z	2kgf 4kgf 10kgf.ccm
分解能		定格荷重の 1/2000
精度	直線性 ヒステリシス	定格荷重の 0.2% 定格荷重の 0.2%
データ入出力	シリアル入出力 パラレル出力 ディスクリート入出力 アナログ出力	23 ⁺ 1 ⁻ (RS232C, RS422) 16ビット 8入力8出力 6Fv24k (F _x , F _y , F _z , H _x , H _y , H _z)

3.2.3 6軸力覚センサ

ロボットのハンド（アーム先端）に加わる外力を測定するため、手首（アームとハンドの間）の部分に6軸力覚センサを装着した。センサは（株）ニツク製の UFS-2012A05-PAN を用いた。仕様を表 3.2 に示す。コンピュータとのリアルタイムのデータのやりとりはパラレル I/O を用いている。

また、場合によっては、1990 年に [長田 90] が製作した 6 軸力覚センサと（株）日本電気三栄製のストレーンアンプ 6MS2 を使用している。分解能は並進力 0.1N、モーメント 0.005Nm であった。

3.2.4 2本指ハンド

探り動作をする時に対象物を把握するために、2本指のハンドを用いた。これは、1993 年に [有竹 93] が製作した 2 本指ハンドの機構を使用した（図 3.5）。指の部分や力センサなど一部に変更を加えている。主要な部品の使用を表 3.3 に示す。

表 3.3: 2本指ハンドの主要部品

部品名	型	仕様
DC モータ	光進ミニモ製 2338S024S	定格 (24V, 6500rpm, 50gfcmm) 起動トルク 175gfcmm, 出力 3.79W
ワイヤ	朝日インテック製 TYPE-C(7×19)-60	直径 0.6mm, 切断荷重 27.5kgf
ポテンショメータ	緑測器製 CP-3M	抵抗 5kΩ, 分解能 0.1%, 電圧 10V
半導体歪ゲージ	共和電業製 KSN-2-120-E5-16	抵抗値 120Ω, ゲージ率-110 ゲージ長 2mm, 許容電流 20mA

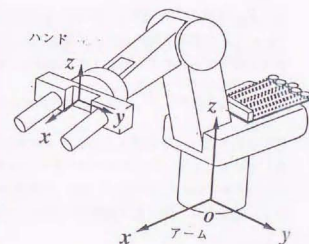
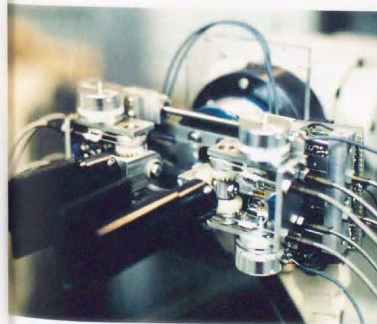


図 3.5: 2本指5自由度ハンド

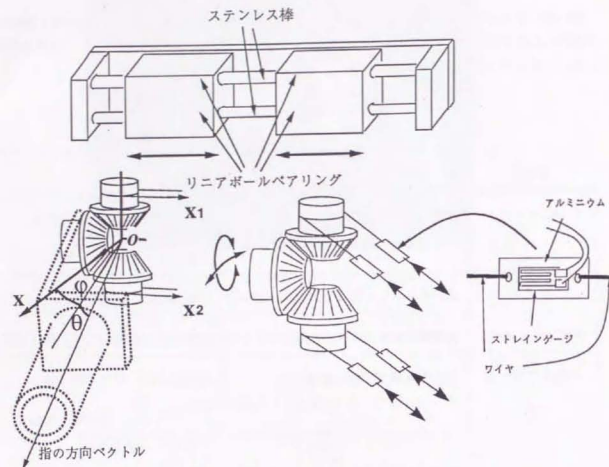


図 3.6: 指の自由度とワイヤ張力センサ [有竹 93 より]

機構

ハンドは根本の直動関節と2本の2自由度の指の計5自由度ある。根本はリニアボールベアリングで、ハンドの開閉方向のスライド機構になっている。これは、指を平行にしたまま任意の指の間隔で物体を把握するためである。1本の指はかさ歯車による差動機構を用いており、図3.6のように上下左右に動かすことができる。

駆動

駆動は蛇管とワイヤを通してDCモータで行っている。モータはロボットアームの1軸の上部に配置している。また、モータはパワーオペアンプを用いた駆動回路により、電流制御されており、コンピュータのD/Aの出力電圧に比例した電流が流れるようになっている。つまり、トルク制御されている。把握力は、指の先(関節から80mm)付近で、最大約400gfである。

センサ

指の関節角を測定するために、各々の指の関節にポテンショメータが直結されている。電圧出力であるため、A/D変換してコンピュータに取り込む。また、ハンドの把握力を測定するため、図3.6のように指の拮抗するワイヤに直列に力センサを配して、ワイヤの張力を測定している。力センサは、アルミニウムの小板に半導体ひずみゲージを張り付けたものである。拮抗する2枚のひずみゲージ

でハーフブリッジを組み、自作したアンプで電位差を約100~200倍に増幅し、A/D変換してコンピュータに取り込んでいる。ヒステリシス等の影響のため、分解能は指のトルクで約50~100g \cdot cmである。

3.2.5 インターフェース

センサや駆動ユニットとのインターフェースの仕様を表3.4に示す。

表 3.4: インターフェースボードの仕様

ボード	型	仕様	個数
カウンタ	(株)コンテック製 CNT24-4A(98)	4ch, 24bit, 4 通倍	2
A/D	(株)インターフェース製 AZI-209	16ch, 12bit, $\pm 2.5V$, 60 μ sec/ch	2
D/A	(株)インターフェース製 AZI-210	8ch, 12bit, $\pm 10V$	2
I/O	(株)インターフェース製 AZI-2710	32bit, フォトカプラ絶縁	1
駆動制御	自作	モード設定, エラー検出等	2

3.2.6 コンピュータと計算の分担

実験システムには、主に2台のパーソナルコンピュータを用いた。2台のコンピュータの通信は共有メモリを介して高速に行っている。また、計算の高速化のため、それぞれDSPのボードを使用している(表3.5)。

それぞれの計算の分担は次の通りである。また、それぞれの計算の同期は、タイマ割り込みと共有メモリによって行っている。

- PC-286VS (C言語, アセンブラ)
 - ロボットの関節角(カウンタ)や6軸力覚センサの値の入力(A/D)
 - モータの電流目標値の出力(D/A)
 - 駆動回路等の異常検出、安全管理
- MSP77230 (アセンブラ)
 - 各入出力値の変換

表 3.5: 使用したコンピュータ

メーカー	型式	CPU	その他
(株) エプソン製	PC-286VS	80286 16MHz	数値演算プロセッサつき
(株) エプソン製	PC-386GE	80386SX 16MHz	数値演算プロセッサつき
(株) マイテック製	MSP77230	NEC/ μ PD77230 33MHz	共有メモリでデータ通信
エムティティ(株) 製	DSP4200	T1/TMS320C30 33MHz	パラレル I/O でデータ通信

- ロボットアームの順運動学、逆運動学の計算
- 力覚センサの値の自重補償、座標変換
- アームのヤコビ行列等の計算
- アームのインピーダンス制御の計算
- PC-386GE (C 言語, アセンブラ)
 - ハンドの関節角や力センサの値の入力 (A/D)
 - ハンドのインピーダンス制御の計算
 - モータの電流目標値の出力 (D/A)
 - 駆動回路等の異常検出、安全管理
 - ロボットの動作計画
- DSP4200 (C 言語)
 - ニューラルネットワークの計算

アームとハンドの制御のサンプリング時間は、1.6msec としており、その時間内に制御に必要な計算を行っている。

3.3 マニピュレータとハンドの制御

本研究では、ロボットアームとハンドに対して、探り動作の時の制御法として、前章で述べたようにインピーダンス制御を用いた。それぞれ独立に制御している。その計算法を示す。トルクと位置による制御則の両方を用いている。

3.3.1 トルク制御によるインピーダンス制御

能動的インピーダンス制御は、ロボットがバネ・マス・ダンパ系として振舞うような制御をかけるものである。手先の望ましいインピーダンス (K_d :剛性行列, D_d :減衰係数行列, M_d :慣性行列) を次式で与える。作業座標系において、手先の位置ベクトル x 、目標位置ベクトル x_d 、手先に環境から加えられる外力 F 、その目標値 F_d とすると、

$$M_d \ddot{x} + D_d (\dot{x} - \dot{x}_d) + K_d (x - x_d) = F - F_d \quad (3.1)$$

ここで、

$$x_e = x - x_d \quad (3.2)$$

とおくと、

$$M_d \ddot{x}_e + D_d \dot{x}_e + K_d x_e = F - F_d \quad (3.3)$$

となる。外力 F が加わった時のマニピュレータの運動方程式は、

$$M(\theta) \ddot{\theta} + \hat{h}(\theta, \dot{\theta}) + g(\theta) = \tau + J_x^T(\theta) F \quad (3.4)$$

となる。ただし、

$$\dot{x} = J_x \dot{\theta}, \quad J_x(\theta) = \frac{\partial x}{\partial \theta^T}, \quad \ddot{x} = \dot{J}_x \dot{\theta} + J_x \ddot{\theta}, \quad \ddot{\theta} = J_x^{-1} \ddot{x} - \dot{J}_x^{-1} J_x^{-1} \dot{x} \quad (3.5)$$

- τ : 関節トルク
- θ : 関節角ベクトル
- $J_x(\theta)$: ヤコビ行列
- $M(\theta)$: 慣性行列
- $\hat{h}(\theta, \dot{\theta})$: 粘性摩擦、遠心力、コリオリ力
- $g(\theta)$: 重力

よって、非線形フィードバックは次式のようになる。

$$\tau = J_x^T(\hat{h}_x(\theta, \dot{\theta}) - M_x(\theta) M_x^{-1}(D_d \dot{x}_e + K_d x_e - (F - F_d)) - F) + g(\theta) \quad (3.6)$$

$$\text{ただし} \quad \hat{h}_x(\theta, \dot{\theta}) = J^{-T} \hat{h}(\theta, \dot{\theta}) - M_x(\theta) J \dot{\theta}, \quad M_x(\theta) = J^{-T} M J^{-1}$$

例えば、ダイレクトドライブモータを用いたマニピュレータのように、トルク制御できる場合には、この式を適用すれば良い。ところが、通常良く用いられている多関節型のマニピュレータには減速機が用いられており、摩擦やガタの影響でトルク制御を実現するのは難しい。そこで、探り動作のための柔らかい制御が実現できればよいと考え、減速機つきのロボットを制御する現実的な式に変形する。

まず、高速には動かないとして、遠心力、コリオリ力、 $\dot{\theta}$ に関する項を無視する。また、慣性を位置によらず一定ではなく、 M_d を $M_d M_e(\theta)$ に置き換える。

$$M_d M_e(\theta) \ddot{x} + D_d \dot{x}_e + K_d x_e = F - F_d \quad (3.7)$$

$$\tau = -J^T (M_d^{-1} (D_d \dot{x}_e + K_d x_e - (F - F_d)) - F) + g(\theta) \quad (3.8)$$

このままでは、減速機の摩擦により定常偏差生じるので、積分項を付加する。

$$\tau = -J^T (M_d^{-1} (D_d \dot{x}_e + K_d x_e - (F - F_d) + \frac{1}{T_i} \int (D_d \dot{x}_e + K_d x_e - (F - F_d)) dt) - F) + g(\theta) \quad (3.9)$$

この式により、トルクに比例したDCモータの電流制御を行う。

3.3.2 位置制御によるインピーダンス制御

ロボットの手の望ましいインピーダンス (K_d :剛性行列, D_d :減衰係数行列, M_d :慣性行列) は次式で与えられる。作業座標系において、手先の位置ベクトル x 、目標位置ベクトル x_d 、手先に環境から加えられる外力 F 、その目標値 F_d とすると、

$$M_d \ddot{x} + D_d (\dot{x} - \dot{x}_d) + K_d (x - x_d) = F - F_d \quad (3.10)$$

ここで、

$$x_e = x - x_d \quad (3.11)$$

とおくと、

$$M_d \ddot{x}_e + D_d \dot{x}_e + K_d x_e = F - F_d \quad (3.12)$$

となる。手先の（並進・回転）速度を $v = \dot{x}$ とおくと

$$M_d \dot{v} + D_d v + K_d x_e = F - F_d \quad (3.13)$$

となる。積分すると

$$v = M_d^{-1} \int ((F - F_d) - D_d v - K_d x_e) dt \quad (3.14)$$

となり、離散値系で表すと、サンプリング時間を Δt として

$$v_n = M_d^{-1} \sum_{i=1}^{n-1} ((F_i - F_d) - D_d v_i - K_d x_{e,i}) \Delta t \quad (3.15)$$

$$v_n = \Delta t M_d^{-1} ((F_{n-1} - F_d) - D_d v_{n-1} - K_d x_{e,n-1}) + v_{n-1} \quad (3.16)$$

と漸化式の形に書ける。ある時点の手先の速度 v_n が求まるので、これを関節の位置制御に直せばよい。関節角ベクトル θ 、ヤコビ行列 $J_e(\theta)$ とすると、 $v_n = J_e \dot{\theta}_n$ より

$$\theta_n = \theta_{n-1} + \dot{\theta}_{n-1} \Delta t = \theta_{n-1} + J_e^{-1} v_n \Delta t \quad (3.17)$$

である。現在の手先の位置姿勢と外力を測定して、この θ_n を目標値として位置制御を行えば良い。

3.3.3 インピーダンスの設定について

残りの問題は、作業に対してどのような仮想インピーダンスを設定するかである。仮想インピーダンス K_d, M_d, D_d は、作業座標系において対角成分で指定するのが通常である。仮想バネ定数が0と ∞ の場合はそれぞれ純粋な力制御と位置制御に対応する。さらに、仮想インピーダンスのコンプライアンスセンタ（作業座標系の原点）をどこにするかが重要である。ロボットの手先の基準点にとることが多いが、この点を基準に力や速度が計算されることになるため、コンプライアンスセンタの位置によりロボットの挙動がかなり異なる。

ロボットアームの場合は、コンプライアンスセンタはハンドの中心付近にとり、作業座標系でインピーダンスを設定している。ハンドは、作業座標系ではなく、関節毎に（関節座標系で）インピーダンスを設定している。差動機構の部分については、上下と左右の方向に対応させている。指を正面に向けた姿勢を仮想バネの平衡点とした。

3.4 まとめ

探り動作の実験のために構成した以下の部分からなる実験システムについて述べた。

- 探り動作をする5自由度多関節ロボットアーム
- 制御および拘束状態のセンシングに用いる手首の6軸力覚センサ
- 対象物を把握する5自由度2本指ハンド
- ハンドの力センサ、関節角センサ
- インターフェースボード
- 制御や動作計画等の計算を行うコンピュータとその役割分担

また、探り動作に用いる制御則として、減速機つきのロボットアーム用のインピーダンス制御の計算を示した。

第4章 実際の探り動作によって得られる 力と変位のパターン

4.1 はじめに

インピーダンス制御の下で、力の目標値を変えていくことにより、探り動作の実験を行い、そのときに得られる力と変位のパターンを調べる。次のような拘束状態について探り動作を行って、実際のパターンを解析する。

- 自由な方向、拘束された方向、弾性のある方向
- ガタのある方向
- 摩擦力のある場合
- 重力がかかる場合
- 可動方向に沿って動く場合

4.2 探り動作の実験

4.2.1 探り動作時のインピーダンスの設定

探り動作時に実際に設定した加える力の大きさやインピーダンスを表 4.1 に示す。人間が両手の上で扱える程度の作業を目安とし、その程度の力と変位が実現できるような値を考えている。ここでは、3次元6自由度の探り動作を対象としており、並進と回転のそれぞれについて均一なインピーダンスを設定した。

表 4.1: 探り動作で加えた力と設定したインピーダンス

	並進	回転
探りの最大力 $ F_d $	1.0N	0.05Nm
探りの最大変位 $ x_d $	0.01m	0.15rad
仮想バネ定数 K_d	100Nm	0.33Nm/rad
仮想質量 M_d	5.0kg	0.05kgm ²
仮想減衰係数 D_d	80Ns/m	0.3Nms/rad

4.2.2 探り動作の様子

探り動作の実験は、ロボットの手先を作業対象物に固定し、対象物や拘束状態を変えて行った。

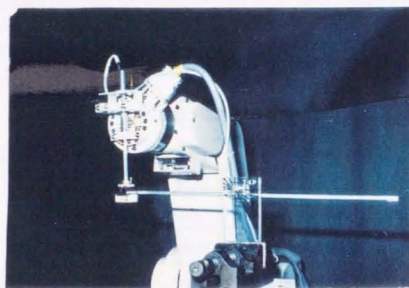
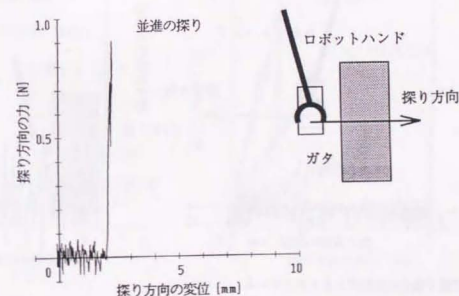
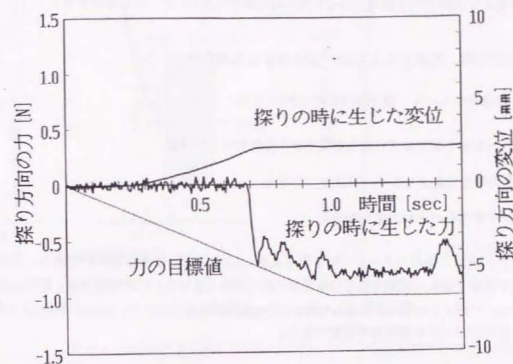


図 4.1: 滑り対偶や回転対偶に探り動作を行っている様子

準静的な探り動作を行わせるため、加える力は、1.0 秒間力の目標値を増加させ、その後 0.5 秒間探りの最大力を持続した。なお、力の目標値と探り方向とは逆向きになる。そして、その間に探り方向に生じた変位と力を 6.4msec 毎に記録した。変位の向きと反力の向きは逆向きであるが、グラフでは力の符号を探り方向の逆向きを正として表示してある。その様子を図 4.2 に示す。



1.0 秒間力の目標値を増加させ、その後 0.5 秒間探りの最大力を持続した。これはガタのある拘束された方向に探り動作を行った例である。

図 4.2: 探り動作の力の加え方と得られた変位と力のパターン例

4.3 探り動作で得られた変位と力のパターン

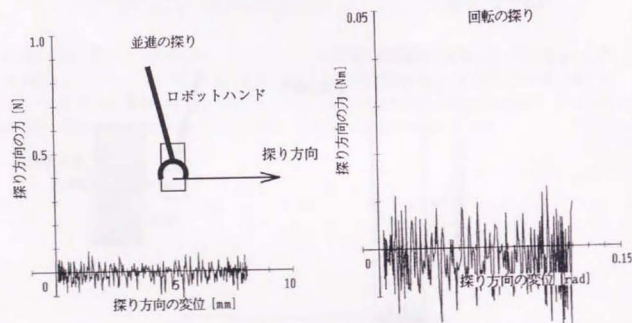
拘束状態を変えて実際に探り動作を行って得られた変位と力のパターンを解析する。

4.3.1 自由な方向、拘束された方向、弾性のある方向のパターン

次の3通りの場合について、探り動作を行った。

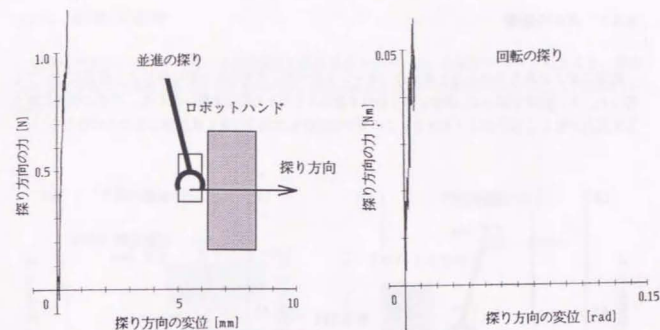
- ロボットの手先を固定せず自由な空間で探り動作を行った場合
- ロボットの手先を動かないように固定した場合
- ロボットの手先をバネに固定した場合

その結果得られた変位と力のパターンの例を図4.3, 4.4, 4.5に示す。結果は明らかであり、自由に動く場合は変位が大きく生じ、固定されている場合は反力が生じている。バネの場合は、変位と力がほぼ比例している。ロボットの手先を固定した場合、少し変位が生じているが、これはロボットの機構、力覚センサ、固定部のガタや剛性等の影響である。



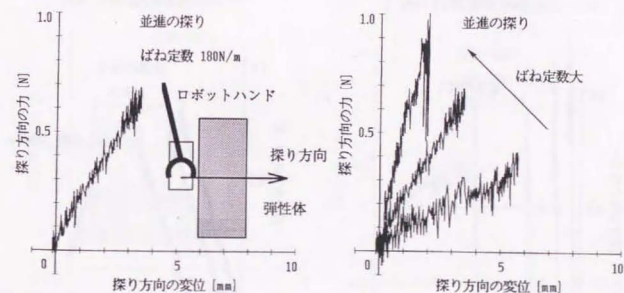
自由に動く場合は変位が大きく生じている。

図 4.3: 自由な方向に探り動作を行った時の変位と力のパターン例



固定されている場合は反力が生じている。ロボットの手先を固定した場合、少し変位が生じているが、これはロボットの機構、力覚センサ、固定部のガタや剛性等の影響である。

図 4.4: 拘束された方向に探り動作を行った時の変位と力のパターン例



バネの場合は、変位と力がほぼ比例している。

図 4.5: 弾性のある方向に探り動作を行った時の変位と力のパターン例

4.3.2 ガタの影響

拘束にガタがある場合の力と変位のパターンを調べた。ガタのある弾性体による拘束についても行った。その結果を図4.6に示す。探り動作を始めるとガタの範囲を動いたのち、ガタの限界に衝突して反力が生じる様子がよくわかる。このガタの部分は本来の拘束とは区別しなければならない。

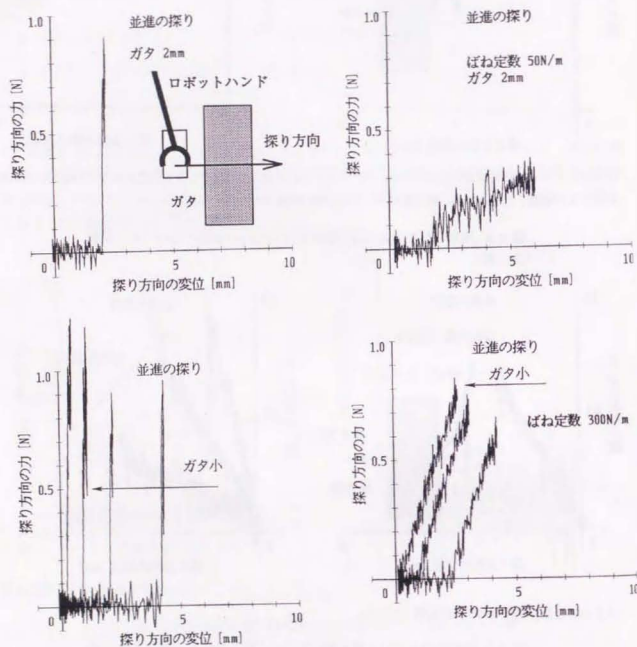


図 4.6: ガタのある拘束された方向に探り動作を行った時の変位と力のパターン例

4.3.3 摩擦力の影響

完全に拘束されている場合には当然摩擦力の影響はないが、動ける範囲が少しでもあると、摩擦力の影響が無視できなくなる。探り動作の時に摩擦力があつた場合の変位と力のパターンを図4.7に示す。

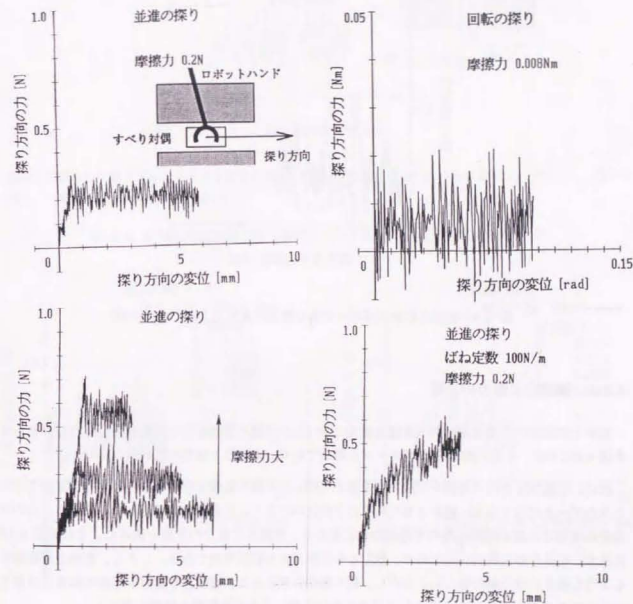


図 4.7: 摩擦のある自由な（弾性のある）方向に探り動作を行った時の変位と力のパターン例

これはロボットの手先に板と取り付け、摩擦のある面に押しつけて探り動作を行わせた。最大静止

摩擦力を越えて動摩擦力を受けながら変位が増えていく様子がわかる。探りの初期に変位が生じているのは、力覚センサや手先の固定部の剛性が小さいためと考えられる。弾性体の拘束についても同様の実験を行った。力にオフセットが生じている。また、面の状態を滑らかでなく、摩擦が大きい面の場合にも探り動作を行わせた。図 4.8 に示すように、摩擦力が大きく変動し、stick-slip が起きているのがわかる。

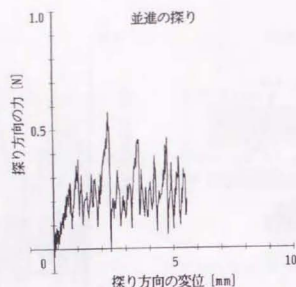
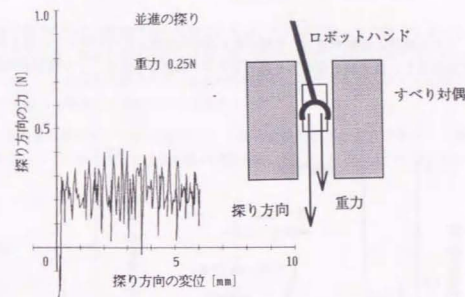


図 4.8: 摩擦の状態が不均一である場合の変位と力のパターン例

4.3.4 摩擦力和重力の分離

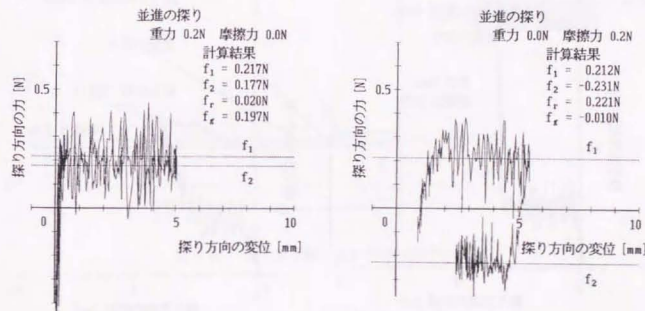
面の上に置かれた重さ 20g の対象物を重力方向 (上) に探り動作を行った時の変位と力のパターンを図 4.9 に示す。一定の摩擦力のパターンと同じであり、摩擦力和重力の区別はつかない。

次に、自重 20g の滑り対偶を可動方向を重力方向にして探り動作を行わせた。探る力を増加させたときのデータだけでなく、減少させた時の力と変位のパターンも合わせて取得した。そして、力の目標値の増加時と減少時の反力の平均値の和と差から、摩擦力和重力の分離を試みた。それを図 4.10 に示す。摩擦力が比較的一定であり、重力との分離がある程度可能である。しかし、摩擦力が変動するような場合には分離は難しい。ただし、探り動作を考えると、摩擦力も重力も本来の拘束とは異なるものであるから、同じ一定力として扱うことにする。パネの予圧等も同様に扱う。



面の上に置かれた重さ 20g の対象物を重力方向 (上) に探り動作を行った。一定の摩擦力のパターンと同じであり、摩擦力和重力の区別はつかない。

図 4.9: 重力方向に並進の探り動作を行った場合の変位と力のパターン例



自重 20g の滑り対偶を可動方向を重力方向にして探り動作を行わせた。探る力を増加させたときのデータだけでなく、減少させた時の力と変位のパターンも合わせて取得した。力の目標値の増加時と減少時の反力の平均値の和と差から、摩擦力和重力の分離を試みた。

図 4.10: 重力と摩擦がある場合に探り動作を行った時の変位と力のパターン例

4.3.5 摩擦のあるガタがある場合

ガタの部分に摩擦や重力の影響がある場合の変位と力のパターンの例を図 4.11 に示す。探り動作を始めると、最大静止摩擦力を越えるとガタの部分動き、その後本来の拘束に衝突する様子がわかる。

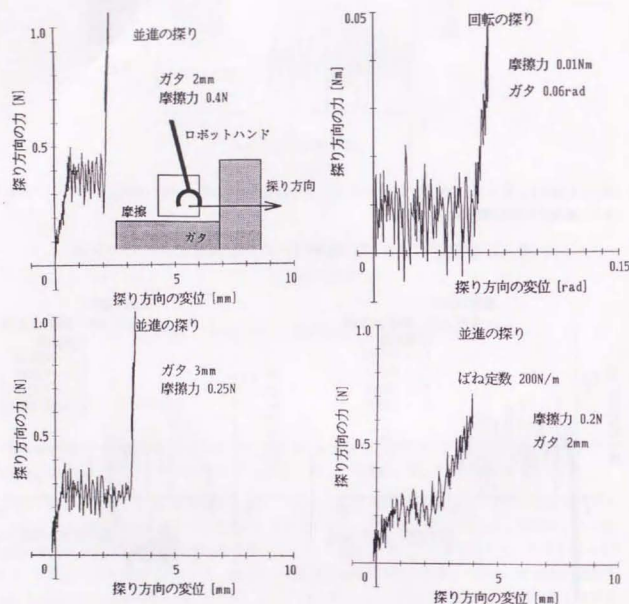


図 4.11: 摩擦とガタのある拘束された（弾性のある）方向に探り動作を行った時の変位と力のパターン例

4.3.6 可動方向に沿って動く場合のパターン

探り動作の力の方向が可動方向の面の摩擦円錐の外側にあれば、その面に沿って摩擦力を受けながら移動する。ガタがない場合には、摩擦を受けながら自由に動く場合の変位と力のパターンと同じである。ガタがある場合には、ガタの範囲を動いた後、本来の拘束の面に衝突するが、そこで拘束されるのではなく、可動方向に沿って摩擦力を受けながら移動していくことになる。

1mm 程度のガタのある滑り対偶を用いて探り動作を行った時の軌跡と探り方向の力と変位のパターンを図 4.12, 4.13 に示す。探り方向は滑り対偶の可動方向に対して $0^\circ \sim 60^\circ$ の 15° おきである。

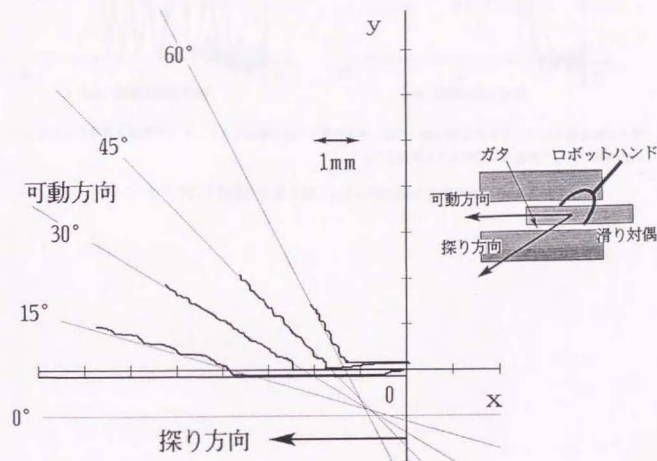
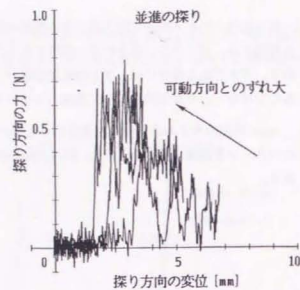
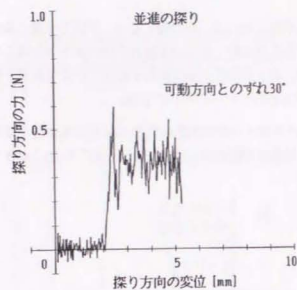


図 4.12: ガタのある滑り対偶での探り動作の様子



ガタがある場合には、ガタの範囲を動いた後、本来の拘束の面に衝突するが、そこで拘束されるのではなく、可動方向に沿って摩擦力を受けながら移動していく。

図 4.13: ガタのある可動方向に沿って動く場合の変位と力のパターン例

4.4 まとめ

インピーダンス制御の下で、力の目標値を変えていくことにより、探り動作の実験を行い、そのときに得られる力と変位のパターンを調べた。前章で考察したように、拘束の状態により特徴あるパターンが得られた。ガタや摩擦の影響等もパターンに現れており、本来の拘束を検出するためには、それらを検出して除去した上で認識を行う必要がある。

第5章 解析的手法による3次元1自由度の拘束の検出

5.1 はじめに

ニューラルネットワークを用いた拘束の認識を行う前に、比較として、解析的手法により拘束の検出を行う。3次元1自由度の拘束の検出を扱い、検出手法を構築し、その過程で実用性や問題点を明らかにする。

次のような項目について検討し、実際の探り動作のデータを用いて処理を行う。

- 3次元の探り動作
- 探り動作で得られるガタや摩擦の影響のある変位と力のパターンから探り方向の拘束の剛性を検出する手法
- 対偶の種類による、探り動作を行った時のロボットの手先の位置変化、姿勢変化、それらのなす角の違い、あるいは、それに相当する力やモーメント
- 3次元1自由度の拘束をねじ対偶のモデルで代表して表したときのねじ対偶のパラメータの推定

5.2 3次元1自由度の拘束

5.2.1 3次元1自由度の対偶

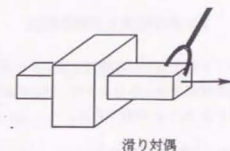
ここでは、3次元1自由度の拘束の検出を扱う。典型的な3次元1自由度の対偶には次の3つの種類がある。

- 滑り対偶
- 回転対偶
- ねじ対偶

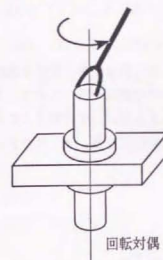
ロボットがこれらの対偶を把握して動かした時のロボットの手先の運動は、対偶のどこを把握しているかによって変わる。例えば、クランクの回転軸上を把握している場合には、ロボットの手先は姿勢の変化しか起こらないが、回転軸から離れた所を把握している場合には位置と姿勢がある一定の関係で変化する。つまり、ロボットの手先と対象物（対偶）の相対位置によって、手先の運動の性質が変わる。そこで、3次元1自由度の拘束を次の5種類に分類して取り扱う（図5.1）。

- 滑り対偶
- 回転対偶（回転軸上を把握した場合）
- 回転対偶（回転軸外を把握した場合）
- ねじ対偶（回転軸上を把握した場合）
- ねじ対偶（回転軸外を把握した場合）

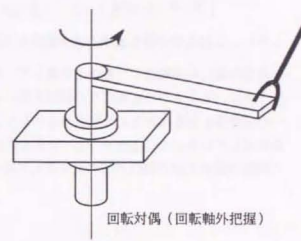
- ① 滑り対偶
- ② 回転対偶（回転軸上を把握した場合）
- ③ 回転対偶（回転軸外を把握した場合）
- ④ ねじ対偶（回転軸上を把握した場合）
- ⑤ ねじ対偶（回転軸外を把握した場合）



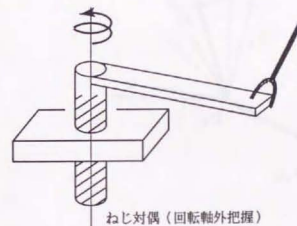
滑り対偶



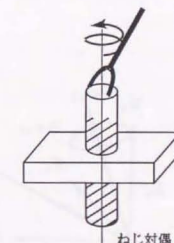
回転対偶（回転軸上把握）



回転対偶（回転軸外把握）



ねじ対偶（回転軸上把握）



ねじ対偶（回転軸外把握）

典型的な3次元1自由度の対偶には滑り対偶、回転対偶、ねじ対偶があり、ロボットがこれらの対偶を把握して動かした時のロボットの手先の運動は、対偶のどこを把握しているかによって変わる。

図 5.1: 3次元1自由度の拘束

5.2.2 3次元の位置と姿勢の表現

ロボットの先手や対象物の位置姿勢を表すには、通常、 4×4 の同次変換行列がよく用いられる。ある基準座標系からみたときの、対象に固定した直交座標系の原点位置ベクトル p と各座標軸の方向ベクトル x, y, z の組である。

$$T = \begin{bmatrix} x & y & z & p \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

しかし、3次元の姿勢を表すのに9変数あり冗長である。

角度の変化の連続性や一意性を考慮して、本研究では、2つの座標系の相対的な姿勢の変位を表現するのに、ある1つの主軸回りの回転を用いる。原点が一致している2つの座標系があったとき、1つの座標系をある軸回りに回転させるともう1つの座標系に重なるような軸が必ず存在することを利用している。この主軸の方向ベクトルに回転角をかけたものが姿勢の変位になる。インピーダンス制御の偏差や探り動作の時の変位として用いるのに便利である。

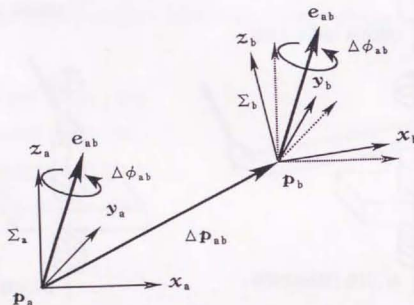


図 5.2: 相対的な位置姿勢の表現

基準座標系から見た2つの座標系の位置姿勢の同次変換行列を T_a, T_b とすると、

$$T_a = \begin{bmatrix} x_a & y_a & z_a & p_a \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_b = \begin{bmatrix} x_b & y_b & z_b & p_b \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.2)$$

このとき、 T_a から T_b に移る位置姿勢の変位 ΔX_{ab} は、位置変位 Δp_{ab} 、回転の軸 e_{ab} 、回転角 $\Delta \phi_{ab}$ とすると、次のようになる。

$$\Delta X_{ab} = \begin{bmatrix} \Delta p_{ab} \\ \Delta \phi_{ab} e_{ab} \end{bmatrix} \quad (5.3)$$

$$\Delta p_{ab} = p_b - p_a \quad (5.4)$$

$$e_{ab} = \frac{x_a \times x_b + y_a \times y_b + z_a \times z_b}{|x_a \times x_b + y_a \times y_b + z_a \times z_b|} \quad (5.5)$$

$$\Delta \phi_{ab} = 2 \tan^{-1} \frac{x_b (e_{ab} \times x_a)}{x_b x_a - 2(e_{ab} x_a)^2 + 1} \quad (5.6)$$

(ただし、 $x_a = x_b$ のときは、 y_a, y_b を用いる)

これは、 T_a を Δp_{ab} だけ平行移動すると原点は T_b の原点に一致し、それぞれの原点を一致させて T_a を単位ベクトル e_{ab} のまわりに $\Delta \phi_{ab}$ 回転させると T_b に姿勢が重なることを意味している。

逆に、 T_a を ΔX_{ab} 変位させたときの位置姿勢 T_b は次のように計算される。

$$p_b = p_a + \Delta p_{ab} \quad (5.7)$$

$$x_b = (x_a e_{ab}) e_{ab} + \cos \Delta \phi_{ab} (x_a - (x_a e_{ab}) e_{ab}) \quad (5.8)$$

(y_b, z_b も同様)

5.3 3次元1自由度の拘束の検出アルゴリズム

5.3.1 探り動作による3次元1自由度の拘束の検出の手順

ロボットの探り動作により3次元1自由度の拘束の検出を行う手順は次のようになる。

- (1) ロボットが対偶をしっかりと把握した状態で、直交した3方向の正負に、比較的微小な力やモーメントを加えて、並進と回転の探り動作を行う。
- (2) 探り動作時のロボットの手先の変位と力の情報を取得する。
- (3) その力と変位のパターンから、ガタや摩擦の影響を取り除いて、「探り方向の拘束の剛性」（変位と力から求めたバネ定数）を計算する。
- (4) このような探り動作を、様々な方向に対して行っていく。3次元1自由度の拘束では可動方向は1方向だけであるので、複数の探り方向の拘束の剛性の最小のものを選び、その時の変位方向を可動方向として検出する。
- (5) 可動方向に再び探り動作を行い、その時のロボットの手先の位置姿勢の変化から、拘束をねじ対偶と見做した時のパラメータを推定する。

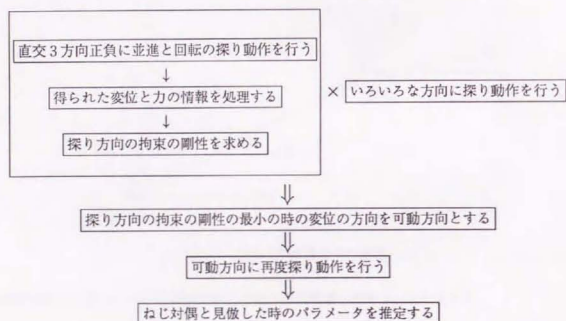


図 5.3: 3次元1自由度の拘束の検出の手順

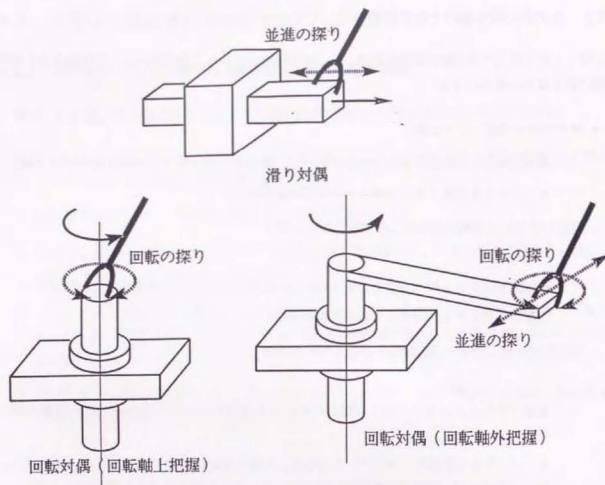
5.3.2 3次元の探り動作と対偶の動き

ロボットが3次元1自由度の対偶を把握して、探り動作を行った（力やモーメントを加えた）時の対偶の動きは次のようになる。

- 滑り対偶を把握している場合
 - ー 並進力を加えると探り方向と可動方向が近い場合はロボットの手先の位置が変化する。
 - ー モーメントを加えても、ロボットの手先は動かない。
- 回転対偶やねじ対偶の回転軸上を把握している場合
 - ー 回転力を加えると探り方向と可動方向が近い場合はロボットの手先の姿勢が変化する。
 - ー 並進力を加えても、ロボットの手先は動かない。
- 回転対偶やねじ対偶の回転軸外を把握している場合
 - ー 並進力を加えると探り方向と可動方向が近い場合はロボットの手先の位置と姿勢が変化する。
 - ー モーメントを加えると、モーメントの中心と対偶の回転中心がずれているが、モーメントを加える方向が回転軸の方向に近い場合はロボットの手先の位置と姿勢が変化する。
 - ー 回転軸に近い点を把握している場合には、対偶の姿勢変化が大きいのでもーメントを加える方が動きやすく、回転軸から遠い点の場合には、並進力を加える方が動きやすい。

3次元の探り動作を行う場合、以上のことを考えると、並進力かモーメントのいずれかを加えれば、拘束（可動方向）に沿って動く可能性がある。必ずしも、並進力とモーメントを同時にかける必要はないといえる。つまり、各探り方向について並進の探り動作と回転の探り動作を別々に行えば良い。

本研究では、直交した3軸の正負の方向に、並進と回転の探り動作を行うことにした。それらの間の空間の方向に可動方向がある場合は、加える力が摩擦円錐の外側にあつて可動方向に沿って移動することを期待している。



並進力かモーメントのいずれかを加えれば、拘束（可動方向）に沿って動く可能性がある。

図 5.4: 並進と回転の探り動作と対偶

5.3.3 探り方向の拘束の剛性の検出

可動方向を検出する前に、それぞれの探り方向について、得られた変位と力のパターンから、探り方向の拘束の剛性（バネ定数）を求める必要がある。前章で示したように、変位と力のパターンにはガタや摩擦の影響が含まれており、それらを除去しなければならない。

実際に探り動作を行って得られた変位と力のパターンには次のような特徴がある。

- 折れ線状であり、部分的には直線で近似できる。
- 探り方向の拘束の剛性は、パターンの最後の直線部分の傾きである。
- 摩擦などの一定力は、力にオフセットを生じさせる。
- ガタの部分など対象物が動いている時は、変位が増えながら力が増減している。
- 位置の変動に比べて力の変動が大きい。
- 動き始めのところでは、力の値が極大になる場合が多い。

これらの特徴を考慮して、図 5.5 のように、変位と力のパターンから探り方向の拘束の剛性（バネ定数）を求めるアルゴリズムを構成した。

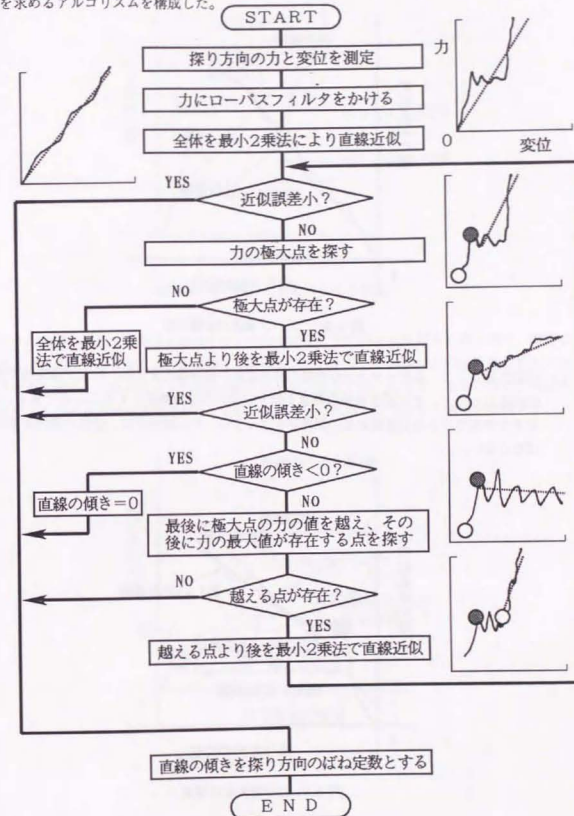


図 5.5: 探り方向のバネ定数を求めるアルゴリズム

(1) 力のデータの高い周波数の変動を取り除くためローパスフィルタ等の前処理を行う。

- (2) 変位と力のパターン全体に対して最小2乗法で直線近似を行う。

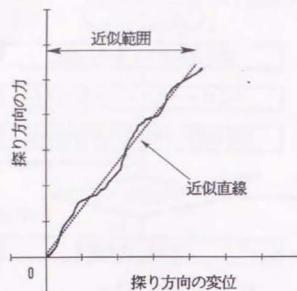


図 5.6: パターン全体の直線近似

- (3) 近似誤差として、各データと近似直線の間の距離の標準偏差をとる。拘束が理想的であれば近似誤差は小さく、その近似直線の傾きを探り方向の拘束の剛性とすることができる。しかし、ガタや摩擦力があれば直線近似の誤差は大きくなる。その場合には、近似の範囲を狭めなければならない。

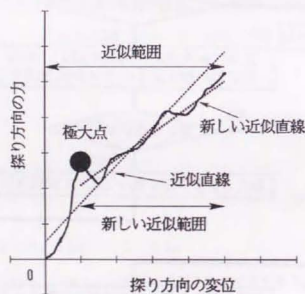


図 5.7: 力の極大点の探索

- (4) 探り動作を行っていった時に最初に現われる力の極大点を探すことにより、対象物が動き始める点を発見する。そして、力の極大点以降を直線近似する。摩擦力を受けながら動くような場合には、直線的なパターンであり近似誤差が小さく、その傾きを拘束の剛性とする。

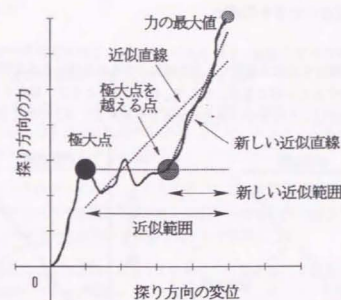


図 5.8: 力の極大点を越える点の探索

- (5) ガタの範囲に相当する動きを検出する。そのために、力の極大点を越える点を探す。厳密にいうと、力の値が力の極大点の値を越える最後の点で、かつ、その後に力の最大値が存在するような点を探す。この点はガタの範囲の動きの終端に相当する。ここで、本来の拘束に当たったと考え、この点以降を直線近似し、近似誤差が小さければ、その傾きを拘束の剛性とする。

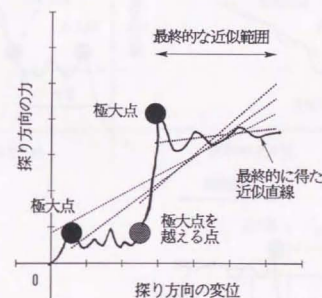


図 5.9: 第2の力の極大点の探索

- (6) さらに可動方向に沿って動いていったような場合を検出する。力の極大点を越える点以降の第2の力の極大点を探す。この点以降が摩擦力を受けながら拘束に沿って移動した部分に相当し、近似直線の傾きが拘束の剛性となる。

5.3.4 ガタや一定力の大きさの検出

探り方向の拘束の剛性を求める過程で、力の極大点やそれを越える点を利用して、ガタや摩擦係に相当する範囲を検出することができる。そこで、図 5.10 のように、相当する範囲の力の平均値や近似直線と軸との交点により、ガタや一定力の大きさを求めることが可能である。

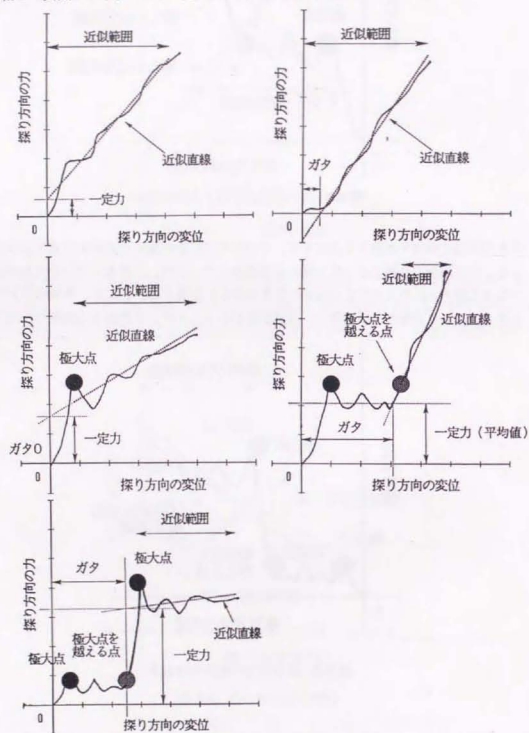


図 5.10: ガタや一定力の大きさの検出

5.3.5 可動方向の検出

3次元1自由度の拘束では可動方向は1方向だけであるので、複数の探り方向の拘束の剛性の最小のものを選び、その時の変位の方向を可動方向として検出する。

ただし、並進と回転の探り動作のときの剛性はそのままでは比較できない。つまり、位置と姿勢、力とモーメントの大きさはそのままでは比較できない。そこで、クランクの腕の長さに相当する係数を適当に定め、どちらかの次元に合わせて比較する。

また、可動方向と探り方向は少しずれていることが多いため、変位と力のパターンから探り方向の剛性を近似直線で求めた時のガタや摩擦の影響を除いた部分（可動方向に沿って動いたと見なせる部分）での実際のロボットの手先の変位方向を可動方向として計算した。

さらに、可動方向を確認し、より正確に推定するために、計算された可動方向に再び探り動作を行う。このときは並進と回転のいずれか大きい方の探り動作を行う。そして、同様に探り方向の拘束のパネ定数や可動方向を求める。

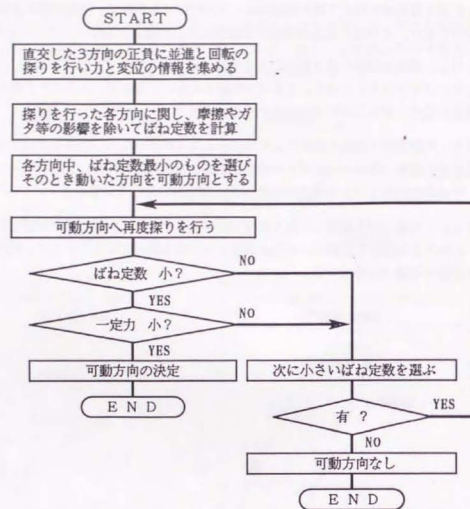


図 5.11: 可動方向の検出とその確認動作

5.3.6 対偶の種類判別

対偶の種類によって、可動方向に探り動作を行った時のロボットの手先の位置姿勢の変化に違いが生じる。これらの違いを利用すれば、対偶の種類や把握位置の判別が可能である。可動方向に探り動作を行った時、それぞれ

$$\text{位置の変化} \quad \lambda_H d_H \quad (|d_H| = 1) \quad (5.9)$$

$$\text{姿勢の変化} \quad \phi_H e_H \quad (|e_H| = 1) \quad (5.10)$$

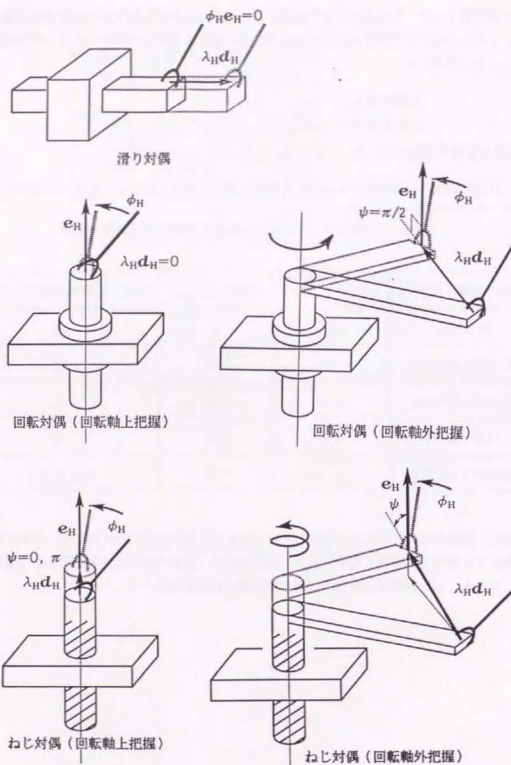
$$\text{位置と姿勢の変化のなす角} \quad \psi = \cos^{-1}(d_H e_H) \quad (0 \leq \psi \leq \frac{\pi}{2}) \quad (5.11)$$

とする。対偶の種類と把握位置による位置と姿勢の変化の関係を表 5.1、図 5.12 に示す。滑り対偶は

表 5.1: 対偶によるロボットの手先の位置姿勢の変化の違い

対偶の種類 (把握点)	位置変化 $\lambda_H d_H$	姿勢変化 $\phi_H e_H$	位置と姿勢の変化のなす角 ψ
滑り対偶	$\neq 0$	0	無
回転対偶 (回転軸上)	0	$\neq 0$	無
回転対偶 (回転軸外)	$\neq 0$	$\neq 0$	$\frac{\pi}{2}$
ねじ対偶 (回転軸上)	$\neq 0$	$\neq 0$	0
ねじ対偶 (回転軸外)	$\neq 0$	$\neq 0$	$0 \leq \psi \leq \frac{\pi}{2}$

位置の変化、回転対偶の回転軸上を把握している場合は姿勢の変化だけが生じる。回転対偶の回転軸外を把握している場合は位置と姿勢の変化の両方が生じ、位置の変化は回転軸に垂直な面内（なす角は垂直）である。ねじ対偶は回転軸方向の位置の変化の成分が生じる。



滑り対偶は位置の変化、回転対偶の回転軸上を把握している場合は姿勢の変化だけが生じる。回転対偶の回転軸外を把握している場合は位置と姿勢の変化の両方が生じ、位置の変化は回転軸に垂直な面内 (なす角は垂直) である。ねじ対偶は回転軸方向の位置の変化の成分が生じる。

図 5.12: 対偶によるロボットの手の位置姿勢の変化の違い

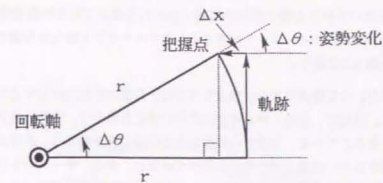


図 5.13: 回転軸回りに回転した時の位置姿勢の変化

回転軸外を把握している場合の位置と姿勢の変化は図 5.13 のようになる。微小な探り動作を行った時の軌跡は回転軸回りの円弧になる。軌跡が円弧であることによる直線からのずれは、微小な回転角に対して 2 次の微小量であるのに対し、変化する姿勢は 1 次の微小量であるため、姿勢の変化が顕著に現れる。

5.4 探り動作による3次元1自由度の拘束の検出結果

5.4.1 探り方向の剛性の検出結果

探り方向の拘束の剛性の検出アルゴリズムを実際の探り動作を行った時に得られた変位と力のパターンに適用した。力と位置のパターンは前章の実験で得られたものを用いた。力のデータにかけたローパスフィルタのカットオフ周波数は、並進力 5Hz、モーメント 3Hz である。近似直線の近似誤差の閾値は 0.020 とした。

その処理結果の例を図 5.15～図 5.20 に示す。力の極大点と極大点を越える点が有効に利用され、ガタや摩擦の影響を除去できていることが分かる。変位と位置のパターンの特徴点と近似直線によって、探り方向の拘束の剛性を検出するアルゴリズムが有効性が示された。

ただし、検出を失敗する例もある。パターンの変動に対して、近似直線の誤差の閾値が大きいとガタや摩擦力等を無視してしまうことになるし、小さ過ぎると力の変動に敏感になって検出を失敗する。適切な閾値を選ぶのは難しい。

また、直交3軸正負方向の並進と回転の探り動作（計12回）を行って得られたそれぞれの変位と力のパターンに対して、探り方向の剛性を計算し、最小の剛性のものを取り出して、その探り動作の時に動いた方向が可動方向であることを確認した。

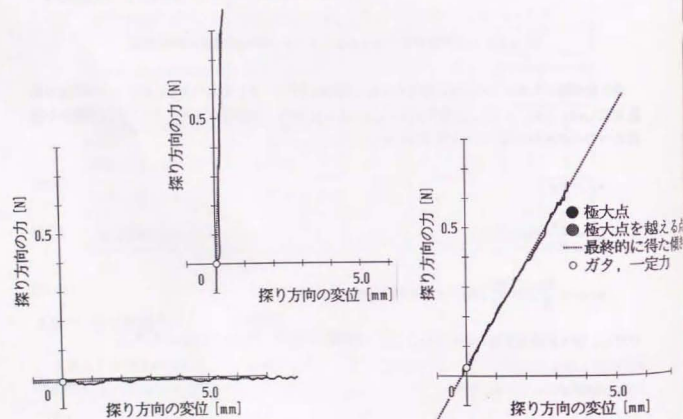
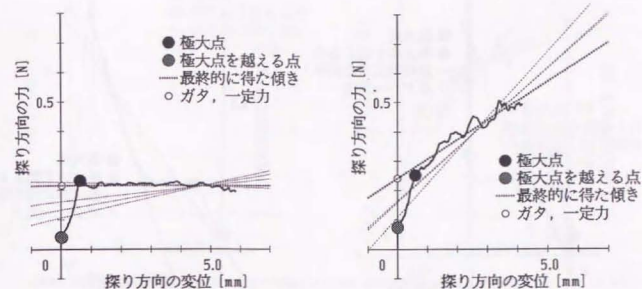
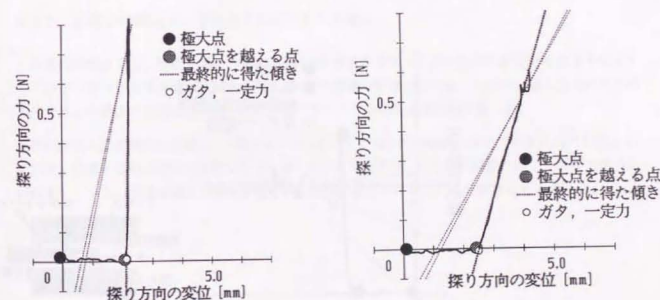


図 5.15: 自由な方向、拘束された方向、弾性のある方向の探り方向の拘束の剛性の検出結果



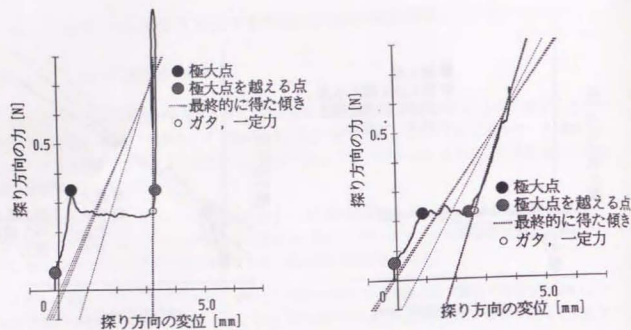
動き始める点が力の極大点により検出されている。その後は摩擦力を受けながら動いている。

図 5.16: 摩擦力がある場合の探り方向の拘束の剛性の検出結果



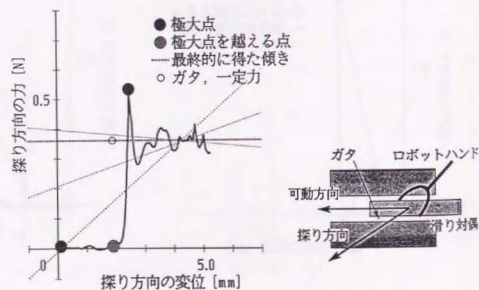
最初の平らな部分がガタの範囲の動きであり、両端の特徴点により検出できている。

図 5.17: ガタがある場合の探り方向の拘束の剛性の検出結果



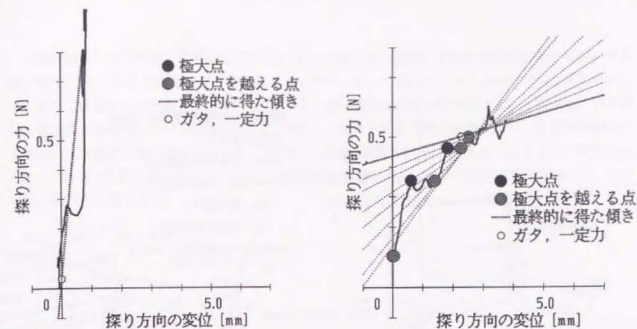
最初の平らな部分が摩擦のあるガタの範囲の動きであり、両端の特徴点により検出できている。

図 5.18: ガタと摩擦がある場合の探り方向の拘束の剛性の検出結果



最初の平らなパターン部分はガタの範囲の動きであり、次の平らな部分は可動方向に沿って摩擦力を受けながら移動している部分である。それぞれの部分の移り変わるところが極大点とそれを越える点により検出できている。

図 5.19: ガタのある可動方向に沿って動く場合の探り方向の拘束の剛性の検出結果



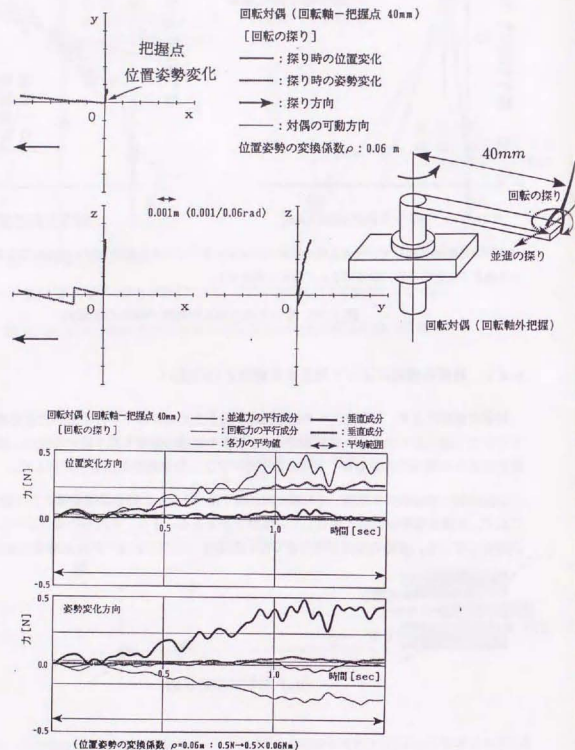
パターンの変動に対して、近似直線の誤差の閾値が大きいとガタや摩擦力等を無視してしまうことになるし、小さ過ぎると力の変動に敏感になって検出を失敗する。

図 5.20: 探り方向の拘束の剛性の検出の失敗例

5.4.2 対偶の種類によって発生する変位と力の違い

対偶の種類により、可動方向への探り動作の結果生じるロボットの手の位置姿勢の変化や発生する力がどう違うかを確かめる実験を行った。ガタや摩擦の影響を取り除いた後の、探り方向の拘束の剛性を求めた範囲の位置姿勢の変化と力やモーメントの平均値を測定・計算した。

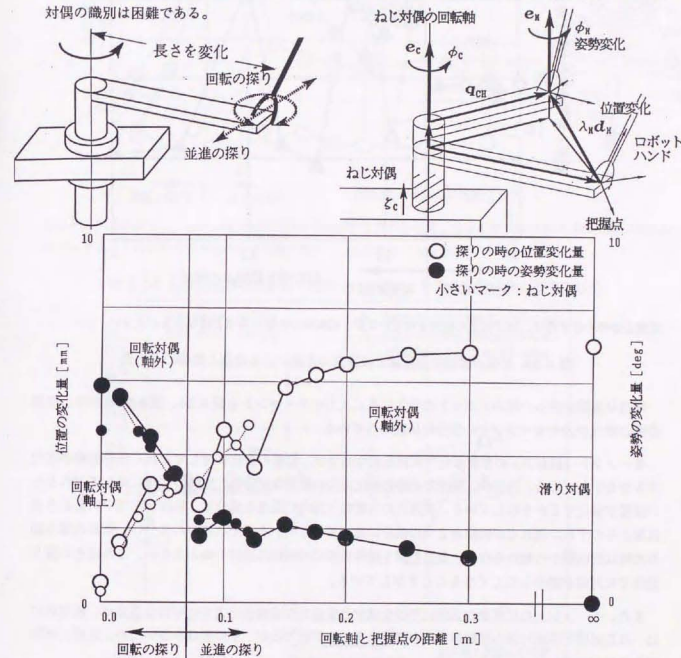
回転対偶の回転軸外を把握して可動方向に探り動作を行った時の位置姿勢と力の変化の例を図 5.21 に示す。位置と姿勢が同時に変化している様子が分かる。また、それらの変位に対して垂直方向の力が発生している。姿勢の変化が滑らかでないのは第 6 軸のエンコーダの分解能のせいである。



位置と姿勢が同時に変化している様子が見える。また、それらの変位に対して垂直方向の力が発生している。

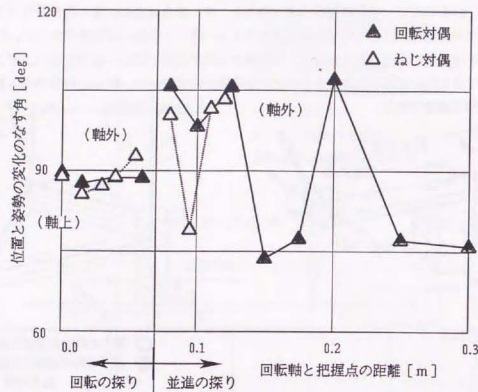
図 5.21: 回転対偶の回転軸から 40mm の点で探り動作を行った時の位置姿勢の変化と力

対偶の種類や回転軸から把握点までの距離を変えた時の探り動作による位置の変化量と姿勢の変化量を図 5.22 に、位置と姿勢のなす角を図 5.23 に示す。ねじ対偶はリード 3.5mm のものを用いている。また、回転軸に近い場合は回転の探り動作を、遠い場合は並進の探り動作を行っている。明らかに、把握点が回転軸に近いと姿勢の変化量が大きく、遠いと位置の変化量が大きい。位置と姿勢のなす角は、ほぼ垂直に検出されているが、回転軸から離れるほどばらつきが大きい。ガタ等の誤差により、ねじ対偶と回転対偶を区別できるほどの差は得られなかった。数 mm 程度の探り動作では、ねじ対偶の識別は困難である。



明らかに、把握点が回転軸に近いと姿勢の変化量が大きく、遠いと位置の変化量が大きい。

図 5.22: 対偶の種類や回転軸と把握点の距離による位置と姿勢の変化量



位置と姿勢のなす角は、ほぼ垂直に検出されているが、回転軸から離れるほどばらつきが大きい。

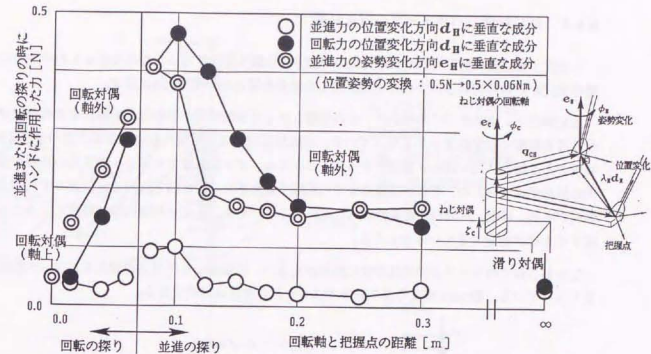
図 5.23: 対偶の種類や回転軸と把握点の距離による位置と姿勢のなす角

同様な実験を行った時のロボットの手に生じた力やモーメントを図 5.24、図 5.25 に示す。位置姿勢の変化が力やモーメントの成分に反映されている。

モーメント（回転力）の位置変化方向に垂直な成分は、位置の変化に対してどれくらい姿勢が変化するかを示している。同様に、並進力の姿勢変化方向に垂直な成分は、姿勢の変化に対してどれくらい位置が変化するかを示している。並進力の位置変化方向に垂直な成分は、軌跡が円弧であるときの直線からのずれが現れるが、前者 2 つの成分に比べて小さい。それぞれの力の成分は、回転の探り動作の時は回転軸から離れるほど、並進の探り動作の時は回転軸に近づくほど大きい。これはその探り動作では対偶が動かしにくくなることを示している。

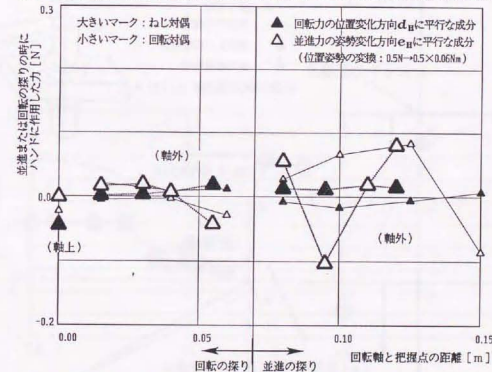
また、モーメントの位置変化方向に平行な成分と並進力の姿勢変化方向に平行な成分は、理想的には、ねじ対偶と回転対偶の回転軸方向の変位による差が生じるが、その差は小さいため、実際に判別できるほどの違いは得られなかった。

以上により、可動方向に探り動作を行った時のロボットの手の位置姿勢の変化や発生する力の違いにより対偶の種類や把握位置の判別が実際に可能であることがわかった。ただし、ねじ対偶と回転対偶の差は実際には得られず、判別は困難である。



それぞれの力の成分は、回転の探り動作の時は回転軸から離れるほど、並進の探り動作の時は回転軸に近づくほど大きい。これはその探り動作では対偶が動かしにくくなることを示している。

図 5.24: 対偶の種類や回転軸と把握点の距離による力の変位方向に垂直な成分



モーメントの位置変化方向に平行な成分と並進力の姿勢変化方向に平行な成分は、理想的には、ねじ対偶と回転対偶の回転軸方向の変位による差が生じるが、実際に判別できるほどの違いは得られなかった。

図 5.25: 対偶の種類や回転軸と把握点の距離による力の変位方向に平行な成分

5.4.3 ねじ対偶のパラメータの推定結果

3次元1自由度の回転対偶やねじ対偶の可動方向に探り動作を行った時のロボットの手先の位置姿勢の変化からねじ対偶のパラメータを推定した結果を図5.26～図5.28に示す。

推定誤差は、姿勢が 10° 程度あり、その影響もあって回転対偶の回転軸と把握点までの距離が長くなるほど位置の推定結果は悪くなっている。回転軸と把握点の距離の推定誤差は、真の値が100mm程度である場合、10～30mm程度となっている。また、ガタの影響でロボットの手先の姿勢の変位方向の誤差が大きいため、それが把握点までの距離で拡大され、回転軸方向の推定位置のずれが大きい傾向がある。以上、対偶の位置姿勢のパラメータについては、数mm程度の探り動作ということを考慮するとほぼ推定できているといえる。

しかし、ねじのリードの推定は非常に誤差が大きい。回転軸から把握点が離れるに従って誤差が大きくなっている。数mm程度の探り動作ではリードの推定は困難である。

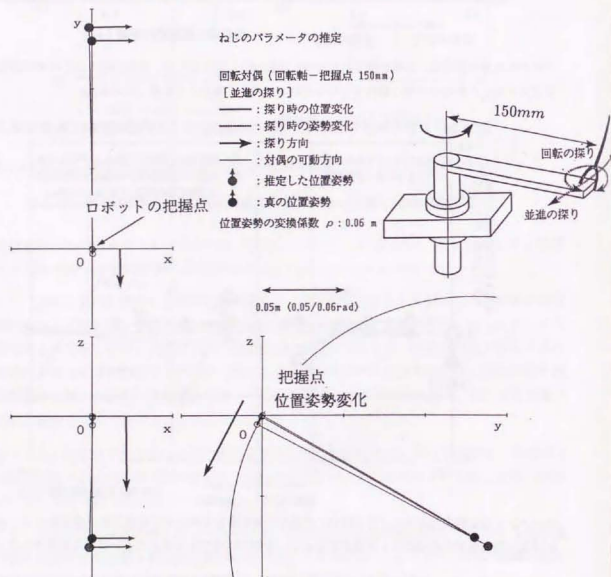
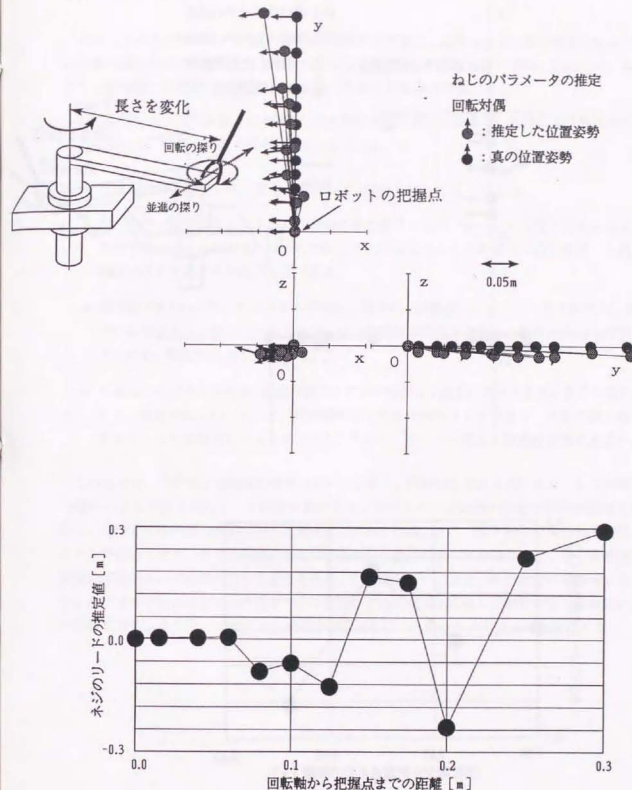
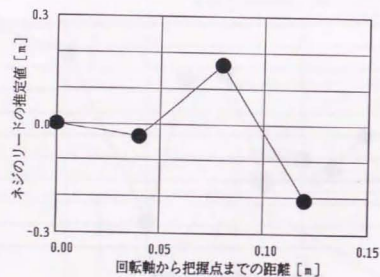
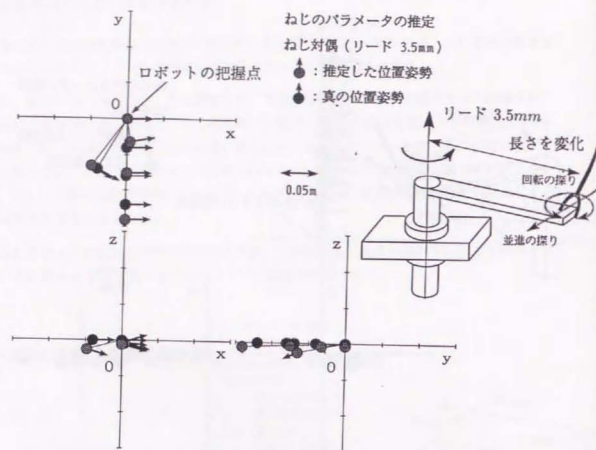


図 5.26: 回転対偶の回転軸から150mmの所を把握した時のねじ対偶のパラメータの推定例



回転軸と把握点の距離を変えて実験した。回転軸から離れた所を把握するほど推定誤差が大きくなっている。回転対偶であるため、ねじのリードのパラメータは0であるが、推定誤差が非常に大きい。

図 5.27: 回転対偶の可動方向に探り動作を行った時のねじ対偶のパラメータの推定例



回転軸と把握点の距離を変えて実験した。リード 3.5mm のねじ対偶であるが、リードのパラメータの推定誤差は非常に大きい。

図 5.28: ねじ対偶の可動方向に探り動作を行った時のねじ対偶のパラメータの推定例

5.5 まとめと考察

ニューラルネットワークを用いた拘束の認識を行う前に、比較として、解析的手法により 3 次元 1 自由度の拘束の検出を行った。3 次元 1 自由度の典型的な拘束は、滑り対偶、回転対偶、ねじ対偶があり、回転軸上を把握した場合と回転軸外を把握した場合で分類できる。

探り動作によって 3 次元 1 自由度の拘束の検出を行う手法を提案し、実験により検出が可能であることを示した。次のような項目が明らかになった。

- 3 次元の探り動作では、並進と回転の探り動作を別々に行ってよい。
- 探り動作で得られるガタや摩擦の影響のある変位と力のパターンから探り方向の拘束の剛性を検出するには、それぞれの動きに対応した力の極大点とそれを越える点を探索し、近似直線の誤差により判断する手法が有効である。
- 探り動作を行った時のロボットの手先の位置変化、姿勢変化、それらのなす角から、あるいは、それに相当する力やモーメントから、対偶の種類が判別できる。ねじ対偶と回転対偶の差は小さいため、実際の区別は無理である。
- 3 次元 1 自由度の拘束は、ねじ対偶のモデルで代表して表すことができる。ねじ対偶のパラメータは、回転軸の方向、リード、回転軸から把握点へのベクトルであり、実際の探り動作のデータからこれらを推定することができた。ただし、リードの推定の誤差は非常に大きい。

このように、3 次元 1 自由度の拘束のモデルを考え、解析的な手法を用いると、ねじ対偶とみなした時のパラメータなどによって拘束を識別することができ、各処理の結果や意味が明確である。しかし、1 つ 1 つの拘束の状態に対して別々にモデルを作成したり、探り動作で得られる変位と力のパターンを解析したり、かなり煩雑である。実際の探り動作のデータの場合には、その各処理における閾値の設定によって結果が異なる場合もあり、設定が難しい。また、多自由度の実環境にある拘束になると、それぞれのモデルの作成やガタや摩擦などの実環境を考慮した解析がさらに複雑になることが予想される。そこで、このような問題点の改良をニューラルネットワークに期待する。

第6章 入出力写像の種類を認識する2段の階層的ニューラルネットワーク

6.1 はじめに

本研究で用いた多層の階層的ニューラルネットワークの構造と動作について記述する。学習法としては、誤差逆伝播（バックプロパゲーション）法と忘却付き構造学習法の理論をまとめる。

そして、入出力の写像の種類を検出する問題を解決する一手法として、入出力関係の反映したニューラルネットワークのリンクの重みに着目した、2段の階層的ニューラルネットワークを提案し、学習アルゴリズム、解釈、他のニューラルネットワークとの比較等について述べる。

また、本研究で用いた各コンピュータにあわせて自作したプログラムについて説明する。

さらに、構築した2段の階層的ニューラルネットワークの基礎的な性質を調べるために、簡単な論理関数の識別の問題に対して適用する。

まず、論理関数の識別の問題を整理し、それに対応した2段の階層的ニューラルネットワークを構成する。前段のニューラルネットワークの学習の各過程の挙動を確認し、それぞれの収束を判定する指標について比較検討を行う。同様に、中間層ユニットの並べ変えの指標についても評価する。そして、最終的に構成した前段の学習により発生するリンクの重みの分布のパターンについて調べる。

後段のニューラルネットワークについては、前段のニューラルネットワークのリンクの重みの分布パターンから8種類の論理関数を識別する学習を行って、2段の階層的ニューラルネットワークが実際に動作することを確認する。

6.2 階層的ニューラルネットワークと誤差逆伝播学習

6.2.1 階層的ニューラルネットワークの構造

工学的に最も多く応用されているニューラルネットワークは、逆誤差伝播学習法を用いた階層的な多層のニューラルネットワークである。3層以上の多層ネットワークで中間層ユニットの数が十分に多ければ、任意の連続写像を近似できることが証明されている。

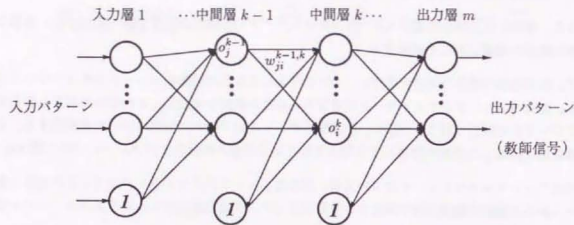


図 6.1: 階層的多層ニューラルネットワーク

図 6.1に示すように、入力層、中間層、出力層で構成され、中間層は1層以上である。同じ層のユニット（ニューロン）間には結合はなく、どのユニットも1つ前の層からのみ入力を受け、次の層にのみ出力を送る。 k 層のユニット i の出力 o_i^k は、1つ前の層 $k-1$ のユニット j からリンクの重み $w_{ji}^{k-1,k}$ を通じて出力を受けるとすると、

$$i_i^k = \sum_j w_{ji}^{k-1,k} o_j^{k-1} \quad (6.1)$$

$$o_i^k = f(i_i^k) \quad (6.2)$$

のように入力の重み付きの総和の単調増加関数関数 f で表せる。ただし、しきい値は出力が1のユニットとそのリンクの重みとして考える。出力関数 f は図 6.2のようなシグモイド関数がよく用いられる。

$$f(u) = \frac{1}{1 + e^{-u}} \quad (6.3)$$

本研究でもこの関数を用いた。この関数の微係数は次の性質がある。

$$f'(u) = \frac{e^{-u}}{(1 + e^{-u})^2} = f(u)(1 - f(u)) \quad (6.4)$$

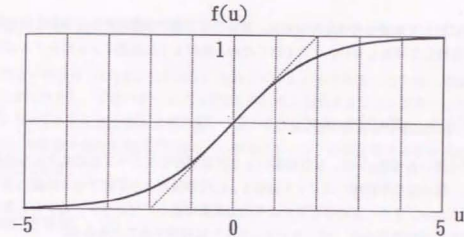


図 6.2: シグモイド関数

6.2.2 誤差逆伝播学習

多層ニューラルネットワークの中間層の学習は基本的にバックプロパゲーション（誤差逆伝播）法 [Rumelhart 86] が用いられる。学習の評価として次のような平均2乗出力誤差関数 E をとる。出力層 m のユニット j の望ましい出力（教師信号）を t_j とすると

$$E = \frac{1}{2} \sum_j (t_j - o_j^m)^2 \quad (6.5)$$

となる。ここで、この平均2乗出力誤差を最小にするため最急降下法を適用すると、

$$\Delta w_{ji}^{k-1,k} = -\epsilon \frac{\partial E}{\partial w_{ji}^{k-1,k}} \quad (\epsilon > 0) \quad (6.6)$$

ずつリンクの重みを変化させていけば、平均2乗出力誤差が極小の方向に向かう。これは、本来は入出力パターンが全て与えられた後にリンクの重みを修正することになるが、係数 ϵ が小さい場合は、各入出力が与えられる度に变化させても全体の变化量はほぼ等しくなる。

$$\frac{\partial E}{\partial w_{ji}^{k-1,k}} = \frac{\partial E}{\partial i_i^k} \frac{\partial i_i^k}{\partial w_{ji}^{k-1,k}} = \delta_i^k o_j^{k-1} \quad (6.7)$$

$$\delta_i^k = \frac{\partial E}{\partial o_i^k} \frac{\partial o_i^k}{\partial i_i^k} = \left(\sum_l \frac{\partial E}{\partial i_l^{k+1}} \frac{\partial i_l^{k+1}}{\partial o_i^k} \right) f'(i_i^k) = \left(\sum_l \delta_l^{k+1} w_{il}^{k,k+1} \right) f'(i_i^k) \quad (k \neq m) \quad (6.8)$$

$$\delta_i^m = (o_i^m - t_i) f'(i_i^m) \quad (k = m) \quad (6.9)$$

つまり、リンクの修正量は

$$\Delta w_{ji}^{k-1,k} = -\epsilon \delta_i^k o_j^{k-1} \quad (6.10)$$

となり、出力層より逆向きに誤差を送って、順に計算していくことになる。ここで、リンクの重みの初期値が全て同じ値であると、中間層のユニットの差がなくなるため、学習は行われない。よって初期値は小さなランダム値を設定する。

学習の収束に対して重要な意味を持つのは、重みの修正量の係数 ϵ である。小さいと学習速度が遅く、大きいと発散して収束しない。それを防ぐため、次のような速度による修正アルゴリズムが用いられる。

$$\Delta w_{ji}^{k-1,k}(t) = -\epsilon \delta_j^k o_j^{k-1} + \alpha \Delta w_{ji}^{k-1,k}(t-1) \quad (0 < \alpha < 1) \quad (6.11)$$

しかし、学習に用いる係数 ϵ, α は、試行錯誤により学習が速くスムーズに収束するものを選ぶ必要がある。さらに、層の数や中間層のユニット数等も、試行錯誤により学習させる問題に適したものを選ばなければならない。また、この学習法により出力誤差は極小に向かうが、必ずしも最小になるわけではなく途中の局所的な極小値（ローカルミニマム）に収束することもある。

6.3 忘却付き構造学習法

忘却付き構造学習法は[石川 92][石川 90]が提案しているものである。これは、階層的ニューラルネットワークを対象とし、学習パターンに内在する規則性を発見することを目的としている。ネットワークのリンクの重みに忘却を導入して不要なリンクを消滅させ、規則性を重みの骨格構造として出現させるため、中間層や重みの解釈が可能で、中間層のユニットの個数などを予め決めたりする必要がない点が特徴である。

6.3.1 忘却付き学習

忘却付き構造学習法の学習の評価 E_f として通常の誤差逆伝播学習での平均2乗出力誤差 E に加えてリンクの重みの絶対値の和の項を付加する。これはリンクの重みの大きさができるだけ小さいネットワークを生成することをねらっている。

$$E_f = E + E_1 = E + \epsilon_f \sum_{k,i,j} |w_{ji}^{k-1,k}| \quad (6.12)$$

となる。そこで最急降下法を適用した学習則は、誤差逆伝播学習の修正量 $\Delta w_{ji}^{k-1,k}$ に加えて次のようになる。

$$\Delta w_{ji}^{k-1,k} = \Delta w_{ji}^{k-1,k} - \epsilon_f \frac{\partial E_1}{\partial w_{ji}^{k-1,k}} \quad (\epsilon_f > 0) \quad (6.13)$$

$$\frac{\partial E_1}{\partial w_{ji}^{k-1,k}} = \text{sgn}(w_{ji}^{k-1,k}) \quad (6.14)$$

つまり、リンクの修正量は

$$\Delta w_{ji}^{k-1,k} = \Delta w_{ji}^{k-1,k} - \epsilon_f \text{sgn}(w_{ji}^{k-1,k}) \quad (6.15)$$

となる。重みが学習の繰り返し毎に一定量 ϵ_f ずつ減少することになり、成長しない不要なリンクの重みは次第に消滅する（0に近づく）。ここでも、一定の忘却量 ϵ_f の大きさは重要な意味を持っている。過大であれば、リンクの重みの成長が妨げられ学習速度が遅くなり、過小であれば、忘却の効果は少なく冗長な重みが残ってしまう。

問題点は2つある。1つは、1本のリンクの重みが重みの和が一定のままで何本かのリンクの重みに分散しても評価が変わらず、冗長表現になることである。もう1つは、学習が収束しても、平均2乗出力誤差がリンクの重みの絶対値の和の分だけ残ることである。

6.3.2 隠れユニット明確化学習

忘却付き学習の問題点を解決するために、学習の評価にもう一つの項を追加する。これは、ユニットの出力が0または1に近付いて、中間の値をとらないようにする項であり、0と1の近い方の値と

ユニットの出力 o_j^k の差 (距離関数) の和である。

$$E_c = E_f + E_2 = E + E_1 + \epsilon_c \sum_{k,j} d(o_j^k) \quad (6.16)$$

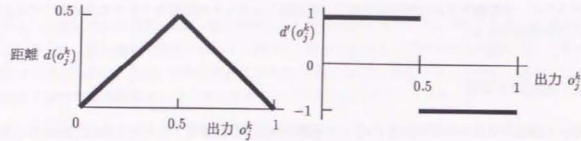


図 6.3: 距離関数とその微係数

距離関数 $d(o_j^k)$ は、図 6.3 のようになる。学習則は、忘却付学習の修正量 $\Delta w_{jif}^{k-1,k}$ に加えて次のようになる。

$$\Delta w_{jic}^{k-1,k} = \Delta w_{jif}^{k-1,k} - \epsilon_c \frac{\partial E_2}{\partial w_{ji}^{k-1,k}} \quad (\epsilon_c > 0) \quad (6.17)$$

$$\frac{\partial E_2}{\partial w_{ji}^{k-1,k}} = \frac{\partial E_2}{\partial i_i^k} o_j^{k-1} = \frac{\partial E_2}{\partial o_i^k} \frac{\partial o_i^k}{\partial i_i^k} o_j^{k-1} = d'(o_i^k) f'(i_i^k) o_j^{k-1} \quad (6.18)$$

つまり、リンクの修正量は

$$\Delta w_{jic}^{k-1,k} = \Delta w_{jif}^{k-1,k} - \epsilon_c d'(o_i^k) f'(i_i^k) o_j^{k-1} \quad (6.19)$$

となる。この明確化学習により、入力パターンに対して中間層のユニットが 1 または 0 に近い値を出力するようになり、冗長なユニットにつながるリンクの重みを消去する効果がある。

6.3.3 選択的忘却付き学習

忘却付き学習では、学習の出力誤差が残ってしまう。これを少なくするために、重みの絶対値がある閾値 w_t 以下のもののみを忘却付学習の対象とする。ある時点で閾値以下の重みでも必ずしも最終的に 0 になるわけではない。学習の評価関数とリンクの修正量は

$$\tilde{E}_f = E + \tilde{E}_1 = E + \epsilon_f' \sum_{|w_{ji}^{k-1,k}| < w_t} |w_{ji}^{k-1,k}| \quad (6.20)$$

$$\Delta w_{jif}^{k-1,k} = \Delta w_{ji}^{k-1,k} - \epsilon_f \operatorname{sgn}(w_{ji}^{k-1,k}) \quad (|w_{ji}^{k-1,k}| < w_t) \quad (6.21)$$

$$\Delta w_{jif}^{k-1,k} = \Delta w_{ji}^{k-1,k} \quad (|w_{ji}^{k-1,k}| \geq w_t) \quad (6.22)$$

となる。

6.3.4 忘却付き構造学習の手順

忘却付き構造学習は以下のような手順で行う。

- (1) 忘却付き学習
- (2) 忘却付き+隠れユニット明確化学習
- (3) 選択的忘却付き+隠れユニット明確化学習

6.4 2 段の階層的ニューラルネットワーク

6.4.1 入出力画像の種類を検出する問題

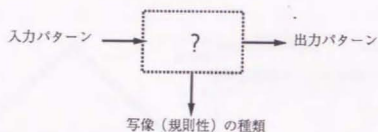


図 6.4: 入出力の関係を検出する問題

ここで、本研究で取り扱う、図 6.4 のように入力と出力の関係を検出する問題を考える。何かあるシステムに入力と出力があり、その間に何らかの関係や規則性があるとすると。そのような場合に、システムの入力と出力を観測して、それがどのような変換（写像）を行っているか、その種類を検出する問題を設定する。例えば、1 入力 1 出力の制御対象があった時に、その入出力関係を観測して制御対象の動特性（伝達関数）を同定するような問題である。ただし、ここでは入出力が複数の場合や入出力関係が陽に関数の形で記述できないような場合も含めて一般的に取り扱うことを目指す。

6.4.2 従来提案されているニューラルネットワークによる実現法

入出力の写像の種類を検出する問題に対して、2 段の階層的ニューラルネットワークを提案する前に、従来提案されている他の構造のニューラルネットワークでこの問題を実現する方法を考察する。

砂時計モデル

入出力ユニット数に比べて少ない中間層ユニットを持つ階層型ニューラルネットワークを利用して、入出力の恒等写像を学習させ、その特徴量を内部表現として中間層に特徴抽出させるものがある。この階層型ニューラルネットワークは「砂時計モデル」と呼ばれる。つまり、入力パターンを中間層で圧縮し、出力層で入力と同じ出力パターンに復元することになり、圧縮された中間層ユニットの表現を解析すれば、入力パターンの特徴が分かる。3 層ニューラルネットワークの場合は、圧縮と復元がそれぞれ 1 層ずつであるため、各層間の変換が線形になる ([Rumelhart 86])。そこで非線形な変換に対応するときは図 6.5 に示すような 5 層ニューラルネットワークが用いられる ([入江 90], [Cottrell 87])。

入出力の写像の種類を検出する問題に、この砂時計モデルを応用すると図 6.6 のような構成が考えられる。写像の入出力パターンの組をそのまま並べて学習する入力パターンにする。そのパターンの恒等写像を学習させると、中間層ユニットは入出力パターンに内在する特徴量を出力するようになる。その中間層の出力パターンのマッチングにより、写像の種類を検出する。この場合には、パターンに内在する特徴量をとらえるのに、リンクの重みではなく、中間層の出力パターンに着目している。

基本的には通常の階層的ニューラルネットワークと同じであるので、簡単な処理で済む点が良いが、

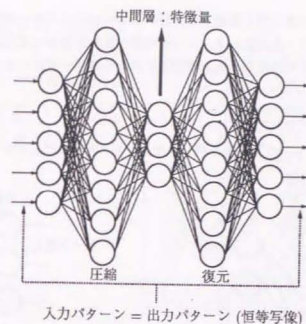


図 6.5: 5 層の砂時計モデル

入力 → 写像? → 出力

入出力パターンの組 k

$$\begin{pmatrix} \text{入力} \\ \text{出力} \end{pmatrix} : \begin{pmatrix} x_{k1} \\ y_{k1} \end{pmatrix}, \begin{pmatrix} x_{k2} \\ y_{k2} \end{pmatrix}, \dots$$

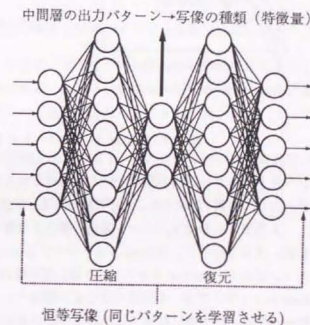


図 6.6: 入出力の写像の種類を検出する 5 層の砂時計モデル

以下のような欠点があり、そのような認識には向かない。

- 写像の種類により取り出す特徴量の次元が異なるため、中間層ユニットの個数を変化させる必要がある。
- 写像の種類が直接分かるのではなく、中間層ユニットの出力パターンの系列を予め用意されたものとマッチングして写像の種類を検出することになる。
- 学習後のリンクの重みのようにある 1 組の決まった個数のパラメータが得られるのではなく、

入出力パターンの組の数と同じ個数の中間層ユニットの出力パターンの系列が得られる。同じ画像でも、場合により、入出力パターンの組の数異なる場合や入出力パターンが変化する場合は、中間層ユニットの出力パターンが変化するため処理が複雑になる。

リカレントニューラルネットワーク

時系列データを扱うフィードバック結合を持つリカレントタイプのニューラルネットワークの利用を

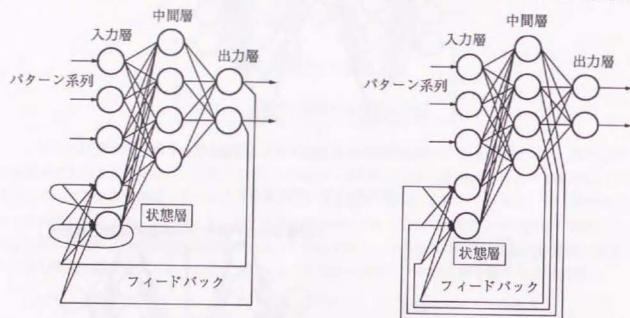


図 6.7: Jordan ネットワークと Elman ネットワーク

考える。リカレントタイプは状態層が新たに設けられており、順序のある複数個のパターンの系列をそのまま入力して学習・認識することが可能である。状態層は入力層の過去の履歴を減衰的に保存する働きをし、入力層と合わせてパターン系列の変化を学習することができる。この状態層を持つネットワークには、大きく分けて、Jordan ネットワーク [Jordan 86] と Elman ネットワーク [Elman 90] の2つのタイプがある。Jordan ネットワークは、出力層の出力を状態層へフィードバックする構造であり、Elman ネットワークは、中間層の出力を状態層へフィードバックする構造になっている。Jordan ネットワークでは、出力層に反映されない入力情報は学習できないが、Elman ネットワークでは、過去の入力情報の履歴が中間層に反映されるため、出力層に出力されない入力情報も学習できる。

入出力の画像の種類を検出する問題に、このリカレントニューラルネットワークを応用すると図 6.8 のような構成が考えられる。画像の入出力パターンの組をそのまま並べて学習する入力パターンにする。ネットワークの出力は識別した画像の種類にする。入出力パターンの組（入力パターンの系列）が時系列データであったり、ある順序がある場合にはこれらの関係を学習することができ、画像の種類検出に適している。

しかし、以下のような欠点がある。

- 基本的には順序のある時系列データの学習・認識を行う能力を持っている。しかし、画像の入

入力 → 画像? → 出力

入出力パターンの組 k (系列)

$$\begin{pmatrix} \text{入力} \\ \text{出力} \end{pmatrix} : \begin{pmatrix} x_{k1} \\ y_{k1} \end{pmatrix}, \begin{pmatrix} x_{k2} \\ y_{k2} \end{pmatrix}, \dots$$

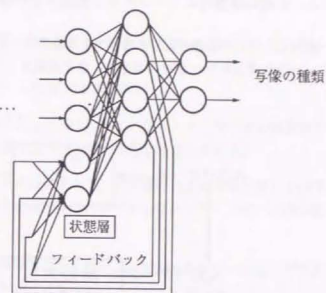


図 6.8: 入出力の画像の種類を検出するリカレントニューラルネットワーク

出力パターンの組の入力順序に意味がない場合にはそのままでは認識できない。

- 空間のパターンの系列を認識するような場合、そのパターンをある一定の順序で入力することができれば、時系列データとして認識できる。しかし、多次元の空間のパターンを分割して一定の順序で取り出すのは周りと隣接している情報を失わせることになる。

本研究では、特に順序のない入出力パターンの組を考えており、そのような場合には時系列パターンを扱うリカレントネットワークは適さない。

6.4.3 2段の階層的ニューラルネットワークの構造

一般的に入出力の画像の種類を検出する問題に対して、本研究では図 6.9 に示すような 2 段の階層的ニューラルネットワークを提案する。出力関数はシグモイド関数を用いている。前段のニューラ

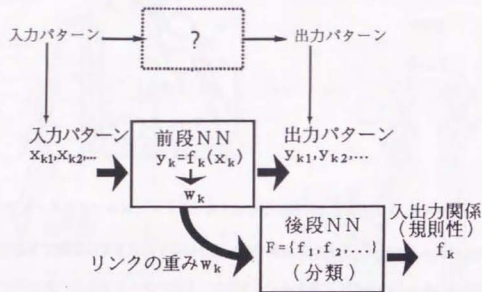


図 6.9: 入出力画像の種類を検出する 2 段のニューラルネットワークの構造

ルネットワークの入出力は、対象としている入出力パターンの組である。ある入出力パターンの組 k (入力 x_{ki} , 出力 y_{ki} , 写像 $y_{ki} = f_k(x_{ki})$ ($i = 1, \dots, n_k$)) が与えられる度に、それらを用いて前段のネットワークで学習を行うと、入出力関係 f_k が前段のネットワークのリンクの重み w_k で表現される。この学習後発生したリンクの重み w_k の分布は入出力関係 f_k の種類によって異なる。

そこで、逆に、前段のネットワークのリンクの重み w_k の分布パターンから、入出力間の写像の種類が何であるかを検出することができると考えられる。重みの分布のパターン認識の方法はいくつか考えられるが、ここでは後段のニューラルネットワークを用いる。つまり、入出力パターンを学習した後の前段のリンクの重みの分布 w_k を後段のニューラルネットワークの入力にする。これが、2 段のニューラルネットワークの大きな特徴である。

後段のニューラルネットワークの出力は写像の種類 f_k であり、様々な写像 f_1, f_2, \dots, f_n に対して発生した前段のニューラルネットワークのリンクの重みパターン w_k と写像の種類 f_k の関係を予め学習させておく。

6.4.4 2 段の階層的ニューラルネットワークの性質

2 段のニューラルネットワークの性質としては次のようなものが挙げられる。

- 与える入出力パターンの組の中のパターンの数には制限がない。
- 前段のネットワークは前処理的な役割を担っており、ニューラルネットワークの補間能力を用いて、離散的な入出力パターンから連続したパターンを学習する能力がある。
- 入出力パターンの組の情報を有限個の前段のネットワークの重みにより情報圧縮して表現しており、後段のネットワークは情報圧縮されたパラメータである重みのデータベースの役割をしている。
- 前段のネットワークの学習は毎回行わなければならない。通常のように、一回良く学習されたネットワークができればよいという問題にはならない。
- 前段のネットワークの重みの分布のパターン認識をするためには、与えられた入出力パターンの組に対して、学習後に同じ、あるいは、いくつかの限られたリンクの重みの分布パターンが発生する必要がある。

6.4.5 写像の種類認識の手順

ある入出力パターンの組が与えられた時、その写像の種類認識の手順をまとめると次のようになる。

- (1) 入出力パターンの組 (入力 x_{ki} , 出力 y_{ki}) を教師データとして前段のネットワークの学習を行う。
- (2) 学習後生じた前段のネットワークの重み w_k を後段のニューラルネットワークの入力に送る。
- (3) 予め学習済みの後段のネットワークは、入力された前段のネットワークの重み w_k の分布から、それに対応する写像の種類 f_k を検出し、出力する。

つまり、学習と認識の全体の流れは図 6.10 のようになる。認識の時にも前段のネットワークの学習を行う点の特徴である。学習時には、写像の種類の数分、前段のネットワークの学習を行って後段のネットワークの学習のための重みのパターンを用意するためかなり時間を要するが、予めオフラインで行っておける。認識時は、提示された入出力パターンに対して前段ネットワークの学習を 1 回行うだけである。

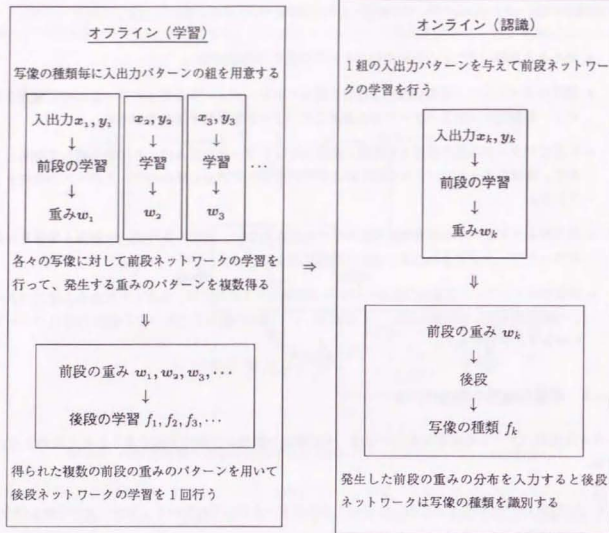


図 6.10: 画像の種類の認識手順

6.4.6 前段ニューラルネットワークの学習

前段のネットワークの重みの分布から後段のネットワークでパターン認識を行うためには、与えられたある入出力パターン組に対して、学習後に同じ、あるいは、いくつかの限られたリンクの重みの分布パターンが発生する必要がある。入出力関係 f_k と発生する前段ネットワークの重み w_k の分布との間に、1対1～数パターンの対応が要求される。多層の階層的ニューラルネットワークの学習で通常用いられる誤差逆伝播 (バックプロパゲーション) 法 [Rumelhart 86] では、学習前の重みの初期値 (小さな乱数で与える) や教師データを提示する順序 (学習順序) などにより、冗長で分散した様々なリンクの重みが発生してしまう。そこで、発生するリンクの重みの冗長性を少なくするために、基本的には、前段ニューラルネットワークの学習法として [石川 92] が提案している忘却付き構造学習法を採用した。学習の速さや後段のネットワークの負荷を考慮し、次のような順序で (図 6.11) 前段ネットワークの学習を行うことにした。

- (1) リンクの重みの初期値として小さな乱数を設定する。
- (2) 通常の誤差逆伝播学習により、リンクの重みを成長させる。
- (3) 忘却付き構造学習を行い、冗長なリンクの重みを消去する。適宜、隠れユニット明確化学習や選択的忘却付き学習も合わせて行う。速度修正型の学習則をまとめると、

$$\begin{aligned} \Delta w_{ji}^{k-1,k}(t) = & \alpha \Delta w_{ji}^{k-1,k}(t-1) \\ & - \epsilon \delta_j^k o_i^{k-1} \quad (\text{誤差逆伝播学習}) \\ & - \epsilon_f \text{sgn}(w_{ji}^{k-1,k}) \quad (\text{if } |w_{ji}^{k-1,k}| < w_i) \quad ((\text{選択的}) \text{ 忘却付き学習}) \\ & - \epsilon_c d^k (o_i^k)^2 f'(i_i^k) o_j^{k-1} \quad (\text{明確化学習}) \end{aligned} \quad (6.23)$$

となる。ただし、

$$\delta_i^k = \left(\sum_j \delta_j^{k+1} w_{ji}^{k,k+1} \right) f'(i_i^k) \quad (k \neq m) \quad (6.24)$$

$$\delta_i^m = (o_i^m - t_i) f'(i_i^m) \quad (k = m) \quad (6.25)$$

$w_{ji}^{k-1,k}$: $k-1$ 層 j ユニットから k 層 i ユニットへのリンクの重み

i_i^k, o_i^k : k 層 i ユニットの入力と出力

t_i : 出力層 m の i ユニットに対する望ましい出力 (教師データ)

f : シグモイド関数

d : 0 または 1 からの距離関数

そこで、それぞれの段階の学習の切替は、各項の係数 $\alpha, \epsilon, \epsilon_f, w_i, \epsilon_c$ の値を変化させることにより一括して容易に行うことができる。

- (4) 同じ機能や逆の機能を持った中間層ユニットがある場合には、どちらか一方の重みを再び小さな乱数にして学習をやり直す。
- (5) 後段のネットワークに入力するリンクの重みのパターンの場合の数を減らして、後段の認識の負担を少なくするため、中間層ユニットをユニットの出力やユニットに入力されるリンクの重みに着目した指標に基づいて、並べ変えを行う。

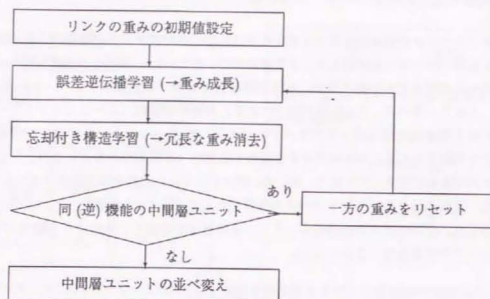


図 6.11: 前段ニューラルネットワークの学習の手順

学習（教師）データに関して、注意する点がある。出力関数にシグモイド関数を使っているせいで、出力層の望ましい出力を 0 ~ 1 にすると収束に時間がかかるため、望ましい出力値は 0.1 ~ 0.9 の範囲に変換して学習を行った (0.1 以下や 0.9 以上ではシグモイド関数がなだらかなのである)。

また、各学習過程の終了（収束）の判定は、

- 学習回数 (学習データ提示回数) t
- 重み $w_{ji}^{k-1,k}$ の絶対値の和の変化

$$Wc = \frac{1}{\Delta t} \sum_{|w_{ji}^{k-1,k}| < w_{ct}} |w_{ji}^{k-1,k}(t) - w_{ji}^{k-1,k}(t - \Delta t)| \quad (6.26)$$

- 平均出力 2 乗誤差

$$E = \frac{1}{2p_n} \sum_{p,j} (t_{pj} - o_{pj}^m)^2 \quad (6.27)$$

(p_n : 教師データの数, o_{pj}^m : 出力層 m のユニット j の出力, t_{pj} : 教師データ)

等の指標に基づいて行う。これは収束後の無駄な学習をしないためである。前段ニューラルネットワークは入出力パターンが与えられる度に学習を行うので、できるだけ速い学習が望まれる。

6.4.7 後段ニューラルネットワークの学習

後段のニューラルネットワークの学習は、予め 1 回だけ行っておく。そのためには、まず、前段のネットワークの重みの分布パターンを学習の教師データとして用意しておかなければならない。後段

のネットワークで認識させたい写像の種類 f_1, f_2, \dots, f_n に対して、入出力パターンの組を用意し、それぞれに対して学習後生じた複数の前段のネットワークの重みの分布のパターンを集める。そして、前段のネットワークの重みのパターンと写像の種類を学習の教師データとして、後段のニューラルネットワークの学習を行う。学習方法は通常の誤差逆伝播（バックプロパゲーション）法を用いる。

6.4.8 2 段の階層的ニューラルネットワークの解釈

ここで、2 段の階層的ニューラルネットワークの動作を分かりやすくするために、前段の入出力関係が線形であると仮定して解釈すると次のようになる。それを図 6.12 に示す。

ある線形関係にある入出力ベクトルの組 k (入力 x_{ki} , 出力 y_{ki} , 写像 $y_{ki} = A_k x_{ki}$ ($i = 1 \dots n_k$)) が与えられる度に、それらを用いて前段のネットワークで学習を行うことは、線形変換の行列 A_k を推定していることになる。ニューラルネットワークのリンクの重み W_k は、行列 A_k の各要素に対応する。予めいろいろな型の行列を学習しておいた後段のニューラルネットワークでは、その行列 A_k の各要素を観察して、線形変換（行列）の型（種類）を識別することになる。例えば、直交、上（下）三角、対角などである。

前段のニューラルネットワークで忘却付き構造学習や中間層の並べ換えを行っているが、それらは行列 A_k を基本的な性質を保ったまま簡単な形に変形していくことに対応している。例えば、行列 A_k の特異値分解に当たる。

このように線形システムと仮定したときの解釈は容易である。入出力パターンの写像の種類を識別する問題に対して、2 段のニューラルネットワークを構成することは自然な発想であるといえる。

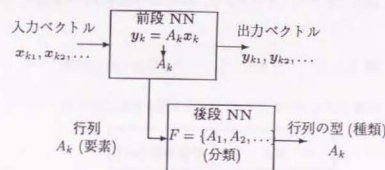


図 6.12: 線形システムと仮定した時の 2 段の階層的ニューラルネットワーク

6.5 ニューラルネットワークのプログラム

多層の階層的ニューラルネットワークのプログラムは、それぞれのコンピュータに合わせて自作した。用いたコンピュータは表 6.1 の通りである。

表 6.1: ニューラルネットワークの計算に用いたコンピュータ

コンピュータ			OS	目的
パソコン (PC98)	EPSON	PC386GE, PC486RS	MSDOS	小規模
DSP ボード	MTT	DSP4200(TI/TMS320C30)		オンライン, 高速
ワークステーション	Sun	Sparc Station 20	Unix	大規模, オフライン

使用したプログラミング言語はそれぞれの OS 上の C 言語である。また、パソコンのグラフィックスは MSC 附属の物を用いた。ワークステーションでは、X-Window 上で Xlib および Xtoolkit を用いた。パソコンとワークステーションのソフトの画面の例を図 6.15, 6.14 に示す。

ソフトは全て、メモリの制限の許す限り、任意の層数で、各層任意のユニット数の階層的ニューラルネットワークを扱えるようにした。学習法は、誤差逆伝播法、忘却付き構造学習法を取り扱い、各パラメータ等は随時変更できるようにしてある。学習の経過や結果も表示される。

それぞれのコンピュータでニューラルネットワークの学習を行った時の計算時間を表 6.2 に示す。3 層 (入力 3、中間 3、出力 1) のネットワークで、式 6.23 の忘却付き学習則を 10000 回実行した時の時間である。

表 6.2: ニューラルネットワークの学習の計算時間

(3 層 (3,3,1) ネットワーク、忘却付き構造学習 10000 回)

コンピュータ	計算時間 [sec]
PC486RS	2.5
DSP4200	1.9
Sparc Station 20	0.4

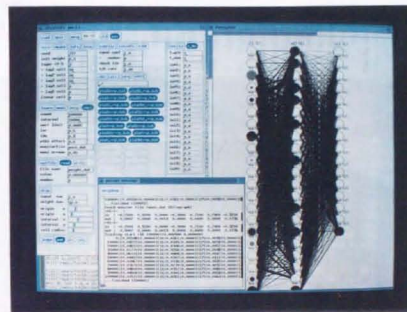


図 6.13: ワークステーション上のニューラルネットワークのソフト (写真)

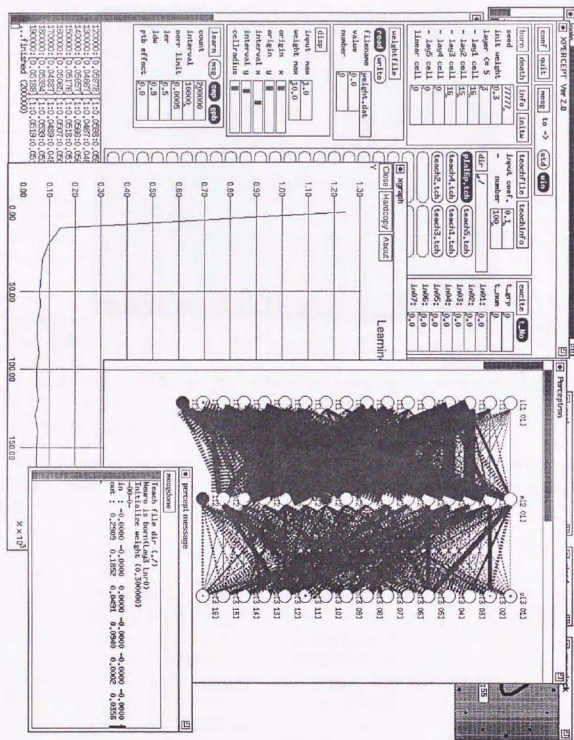


図 6.14: ワークステーション上のニューラルネットワークのソフト

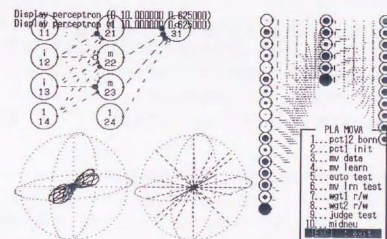
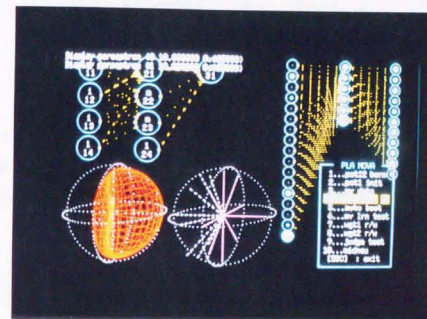


図 6.15: パソコン上のニューラルネットワークのソフト

6.6 2 段の階層的ニューラルネットワークの論理関数識別への応用

6.6.1 論理関数を識別する問題



図 6.16: 論理関数を識別する問題

2 入力 1 出力の論理関数の識別を考える。0 と 1 からなる 2 つ値が与えられた時、未知の論理関数を通して 0 か 1 が出力される。それをいくつか観測してその論理関数を識別しようとする問題である。ここでは、論理関数として、

AND, $\overline{\text{AND}}$, OR, $\overline{\text{OR}}$, XOR, $\overline{\text{XOR}}$, 1, 0

の 8 種類を扱った。入力パターンは

$$x_k = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\} \quad (6.28)$$

であり、1 つずつランダムに提示される。それに対する出力パターンは、

$$y_k = \begin{cases} \{0, 0, 0, 1\} & (\text{AND}) \\ \{1, 1, 1, 0\} & (\overline{\text{AND}}) \\ \{0, 1, 1, 1\} & (\text{OR}) \\ \{1, 0, 0, 0\} & (\overline{\text{OR}}) \\ \{1, 0, 0, 1\} & (\text{XOR}) \\ \{0, 1, 1, 0\} & (\overline{\text{XOR}}) \\ \{1, 1, 1, 1\} & (1) \\ \{0, 0, 0, 0\} & (0) \end{cases} \quad (6.29)$$

である。これらの入出力のパターンから 8 種類の論理関数のいずれかを識別する。

6.6.2 論理関数を識別する 2 段のニューラルネットワークの構造

2 入力 1 出力の論理値 (1/0) の入出力パターンから、8 種類の論理関数を識別する問題に、前章で述べた 2 段の階層的ニューラルネットワークを構成した。その構造を図 6.17 に示す。

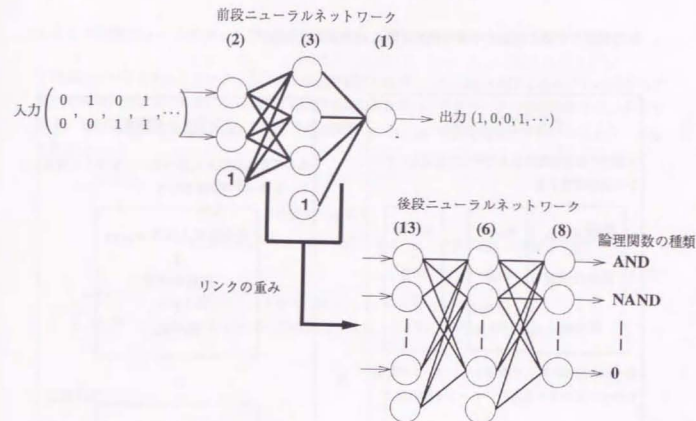


図 6.17: 8 種類の論理関数を識別する 2 段のニューラルネットワークの構造

前段ニューラルネットワークは 3 層構造で、ユニット数は入力層 2、中間層 3、出力層 1 (1 を出力する閾値に相当するユニットは含まない) である。中間層は 8 種類の論理関数を学習するのに十分な個数をとった。入力 1/0 からなる 2 入力のパターンであり、出力はある論理を通した後の 1/0 のパターンである。入出力パターンが与えられる度に、前段のニューラルネットワークの学習を行うと、入出力関係を反映したリンクの重みが生じる。このリンクの重みから、入出力の論理関数の種類を後段のニューラルネットワークで識別する。前段のリンクの重みは $13(3 \times 3 + 4)$ 個あり、これらを後段ニューラルネットワークの入力へ直接送る。ただし、後段のニューラルネットワークの学習のバランスを考慮して、リンクの重みの値を正規化する必要があるが、これについては後段の学習で言及する。

後段ニューラルネットワークも 3 層構造で、ユニット数は入力層 13、中間層 6、出力層 8 である。入力層のユニットは前段のリンクの重みが入力される。後段のニューラルネットワークの出力は 8 種類の論理関数の種類であり、それぞれ 1 つの論理関数に対して 1 つの出力ユニットが対応している。つまり、予め学習済みの後段のニューラルネットワークで理想的に識別が行われた場合には、識別された論理関数に対応したユニットだけが 1 を出力し、あとのユニットは 0 を出力する。

論理関数の学習と識別の手順の概要は図 6.18 のようになる。

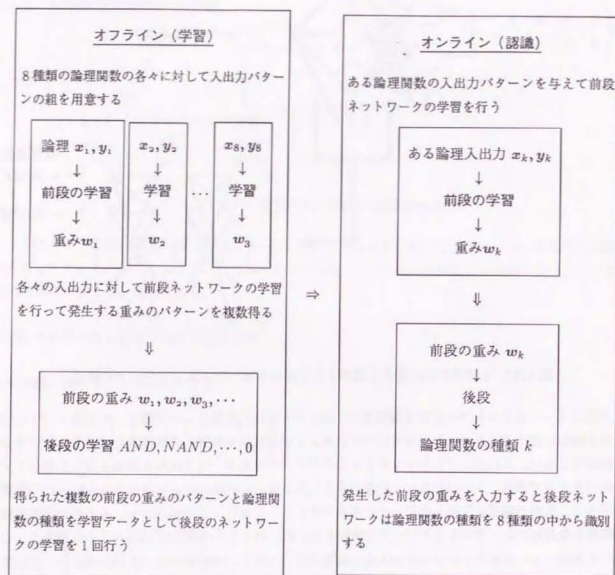


図 6.18: 論理関数の識別の手順

6.6.3 前段ニューラルネットワークの学習

前段ニューラルネットワークの忘却き構造学習法を基礎とした学習の順序とそれぞれの段階の学習の収束の判定は次のようになる。学習則は速度修正型を用い、学習データの提示順序はランダムである。各段階の学習の切替えは、次式の各項の係数 $\alpha, \epsilon, \epsilon_f, w_i$ の値を変化させることにより一括して行う。

$$\begin{aligned} \Delta w_{ji}^{k-1,k}(t) = & \alpha \Delta w_{ji}^{k-1,k}(t-1) \\ & - \epsilon \delta_j^k o_j^{k-1} \quad (\text{誤差逆伝播学習}) \\ & - \epsilon_f \text{sgn}(w_{ji}^{k-1,k}) \quad (\text{if } |w_{ji}^{k-1,k}| < w_t) \quad ((\text{選択的}) \text{ 忘却き学習}) \end{aligned} \quad (6.30)$$

$w_{ji}^{k-1,k}$: $k-1$ 層 j ユニットのから k 層 i ユニットのリンクの重み
 i_j^k, o_j^k, δ_j^k : k 層 j ユニットの入力、出力、および、逆伝播された出力誤差

学習順序

- (1) リンクの重みの初期値として小さな乱数を与える。
- (2) 忘却き学習を行い、リンクの重みを成長させると同時に、冗長なリンクを消去する。教師データの提示はランダムに行う。誤差逆伝播学習の係数 α, ϵ 、忘却学習の係数 ϵ_f を設定し、忘却する重みの閾値 w_t は大きな値にする。他の係数は 0 とする。そして、学習の収束の判定は、重みの絶対値の和の変化 W_c を指標として用いる。

$$W_c = \frac{1}{\Delta t} \sum_{|w_{ji}^{k-1,k}| < w_{ct}} |w_{ji}^{k-1,k}(t) - w_{ji}^{k-1,k}(t - \Delta t)| \quad (6.31)$$

ただし、 $w_{ji}^{k-1,k}$: リンクの重み、 t : 学習回数

この収束の指標がある閾値以下になれば、次の学習に移る。また、ある回数以上学習しても収束しない場合は、もう一度、リンクの重みを初期化し再学習する。

- (3) 選択的忘却き学習を行って、学習の収束を図る。学習を促進するため誤差逆伝播学習の係数 ϵ を大きくし、忘却する重みの閾値 w_t を適当な値に設定する。他の係数はそのままである。そして、学習の収束の判定は平均 2 乗出力誤差を用いる。

$$E = \frac{1}{2p_n} \sum_{p,i} (t_{pi} - o_{pi}^m)^2 \quad (6.32)$$

ただし、 p_n : 学習する教師データの数、 o_{pi}^m : 出力層の出力と望ましい出力

- (4) 後段のニューラルネットワークの識別の負担を減らすため、中間層の入れ換えを行う。入れ換えの指標は中間層ユニットの出力の変化 T_i を用いた。

$$T_i = \sum_p (o_{pi} - \bar{o}_i)^2 \quad (6.33)$$

$$\text{ただし} \quad \bar{o}_i = \frac{1}{p_n} \sum_{p=1}^{p_n} o_{pi} \quad (6.34)$$

p_n : 学習する教師データ, o_{pi} : ユニットの出力

中間層のユニット毎にこの指標を計算し、大きさの順に、同じ層の中で並べ換えを行う。その後、前段のネットワークの重みを後段へ送る。

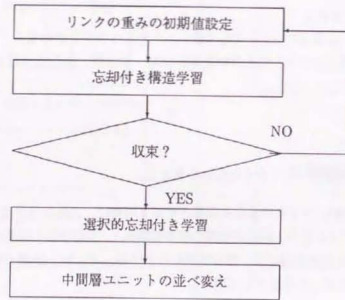


図 6.19: 前段ニューラルネットワークの学習の手順

6.6.4 後段ニューラルネットワークの学習

8種類の論理関数に対して、それぞれ前段のニューラルネットワークの学習を行って、前段のネットワークのリンクの重みのパターンを収集する。それらを用いて後段のニューラルネットワークの学習を行う。望ましい出力は、論理関数の種類に対応するユニットのみ1を出力し、他は0となる。学習法は通常の誤差逆伝播学習を使用する。この後段の学習は予め1回だけ行っておけば良い。

リンクの重み $w_{ji}^{k-1,k}$ の学習則は、次のように誤差逆伝播学習の速度による修正アルゴリズムを用いる。シグモイド関数を f 、ユニットの入力の和 i_i^k 、出力 o_i^k 、望ましい出力 t_i として

$$\Delta w_{ji}^{k-1,k}(t) = \alpha \Delta w_{ji}^{k-1,k}(t-1) - \epsilon \delta_i^k o_j^{k-1} \quad (6.35)$$

$$\delta_i^k = \begin{cases} (\sum_l \delta_l^{k+1} w_{li}^{k,k+1}) f'(i_i^k) & (k \neq m) \\ (o_i^m - t_i) f'(i_i^m) & (k = m) \end{cases} \quad (6.36)$$

ここで、後段のニューラルネットワークへ前段のネットワークのリンクの重みを入力する時に注意すべき点がある。中間層や出力層のユニットは出力関数がシグモイド関数であるため、出力値が0~1の範囲であるが、入力層への入力値もそれとバランスをとる必要がある。なぜならば、式 6.35 に示されるように、学習則のリンクの重みの修正量はリンクの先のユニットの出力値 o_j^{k-1} に比例するからである。つまり、入力層への入力値が大きいと、そこから中間層につながるリンクの重みの修正量が相対的に大きくなり、学習がアンバランスになって全体として学習が収束しないことがあるからである。そこで、8種類の論理関数の識別の場合は、前段のニューラルネットワークのリンクの重み $w_{ji}^{k-1,k}$ に対して

$$w_{ji}^{k-1,k}/10 \quad (6.37)$$

を後段ネットワークの入力層に送る。10 は、生じる重みの大きさから、経験的に決めた値である。

6.6.5 前段ニューラルネットワークの挙動

8種類の論理関数に対する2入力1出力の入出力パターンを用いて、前節に述べたような学習順序で、前段のニューラルネットワークの学習を行った。その時の標準的な係数等の設定を表6.3に示す。

表 6.3: 前段ニューラルネットワークの学習パラメータの設定値

係数		(1) 忘却付き学習時	(2) 選択的忘却付き学習時
リンクの重みの初期値	$w_{ji}^{k-1,k}$	-0.3~+0.3 乱数	
重みの修正速度の係数	α	0.9	0.9
誤差逆伝播学習の係数	ϵ	0.5	2.0
忘却の係数	ϵ_f	1×10^{-4}	1×10^{-4}
選択的忘却の重みの閾値	w_t	100	1.0

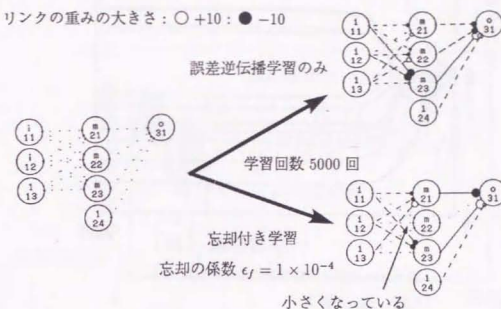
これらは、試行錯誤により定めたものである。選択的忘却付き学習の時の誤差逆伝播学習の係数が大きくなっているのは、その前の忘却付き学習により学習がほぼ収束しており、係数を大きくして学習速度を早めても、学習が発散する恐れが少ないからである。

6.6.6 忘却付き学習の効果

図6.20に忘却付き学習により生じた前段ニューラルネットワークを示す。比較として誤差逆伝播学習のみの場合も示してある。8種類の論理関数の入出力パターンの学習を試みた中で、これはXORの入出力パターンで学習した一例である。ネットワークのリンクの重みの大きさは入力端の円の直径及びリンクの点線の長さで、符合は●と○で図示した。本論文では今後この図示方法を用いる。また、図6.21,6.22にその時の前段ニューラルネットワークのリンクの重み $w_{ji}^{k-1,k}$ の成長と忘却の過程を示す。

通常の誤差逆伝播学習と比較すると、最初は双方ともリンクの重みが成長しているが、途中から、忘却付き学習では冗長な重みが消去されていくことがわかる。そして、最終的には冗長な中間層ユニットにつながるリンクは重みが0になってしまっている。これらのリンクの重みの成長や忘却の過程は、当然、重みの初期値や与える入出力パターンによって収束の速度等が毎回異なるが、他の論理関数の入出力パターンでも同様の傾向が見られる。

リンクの重みの大きさ：○+10：●-10



忘却付き学習では、冗長な中間層ユニットにつながるリンクの重みが消去されている。

$$XOR: \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix} \rightarrow 1, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \rightarrow 0, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \rightarrow 0, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \rightarrow 1 \right\} \quad (6.38)$$

図 6.20: 忘却付き学習の様子

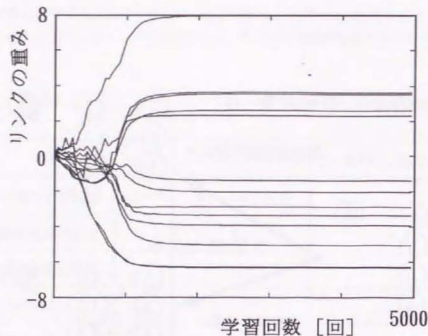
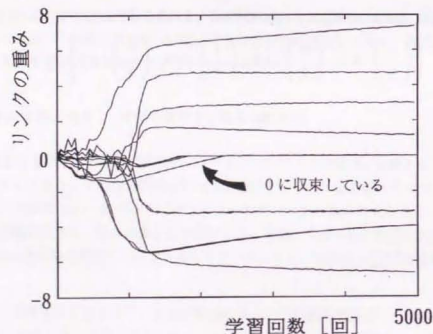


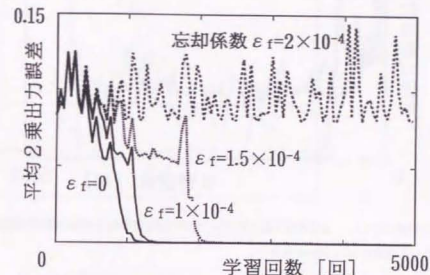
図 6.21: 逆誤差伝播学習の重みの成長過程



通常の誤差逆伝播学習と比較すると、最初は双方ともリンクの重みが成長しているが、途中から、忘却付き学習では冗長な重みが0になっていく。

図 6.22: 忘却付き学習の重みの成長と忘却過程

さらに、忘却の係数 ϵ_f の大きさの影響を確認した。図 6.23 に、平均 2 乗出力誤差 E_f の収束の様子を示す。この結果はすべて同じ重みの初期値から始めたものである。逆誤差伝播学習のみ $\epsilon_f = 0$ に比べて、忘却の係数が大きくなるほど学習の誤差の収束は遅くなっており、大き過ぎると収束しない。もちろん、忘却の係数が小さ過ぎると冗長な重みが消去されないので、誤差逆伝播学習の速度に対して適切な忘却の係数を設定する必要がある。



逆誤差伝播学習のみ $\epsilon_f = 0$ に比べて、忘却の係数が大きくなるほど学習の誤差の収束は遅くなっており、大き過ぎると収束しない。

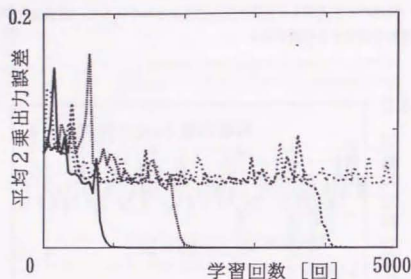
図 6.23: 忘却付き学習における忘却の係数 ϵ_f の影響

6.6.7 忘却付き学習の収束の判定

忘却付き学習の収束の判定を考える前に、学習の前のリンクの重みの初期値の違いによってどの程度収束の速さが変わるかを調べた。図 6.24 は、4 種類の重みの初期値に対して、XOR の入出力パターンでの学習時の平均 2 乗出力誤差の変化を示したものである。これを見ると、初期値によって収束のばらつきが大きい。つまり、学習回数を一定にすることは困難であり、何らかの指標により収束を判定する必要があると言える。

忘却付き学習の収束の判定をする指標の検討を行った。ニューラルネットワークのリンクの重みが十分収束した時点を判定でき、無駄な学習を行わない指標が要求される。検討した指標は以下の通りである。

- 平均 2 乗出力誤差
(p_n : 学習する教師データの数, o_{pn}, t_{pn} : 出力層の出力と望ましい出力)



4種類の重みの初期値に対して、XORの入出力パターンの学習時の平均2乗出力誤差の変化を示したものである。初期値によって収束のばらつきが大きい。

図 6.24: リンクの重みの初期値を変えた時の収束の様子

$$E = \frac{1}{2p_n} \sum_{p,i} (t_{pi} - o_{pi})^2 \quad (6.39)$$

- 重みの絶対値の和の変化
($w_{ji}^{k-1,k}$: リンクの重み, t : 学習回数)

$$W_a = \frac{1}{\Delta t} \left(\sum_{j,i} |w_{ji}^{k-1,k}(t)| - \sum_{j,i} |w_{ji}^{k-1,k}(t - \Delta t)| \right) \quad (6.40)$$

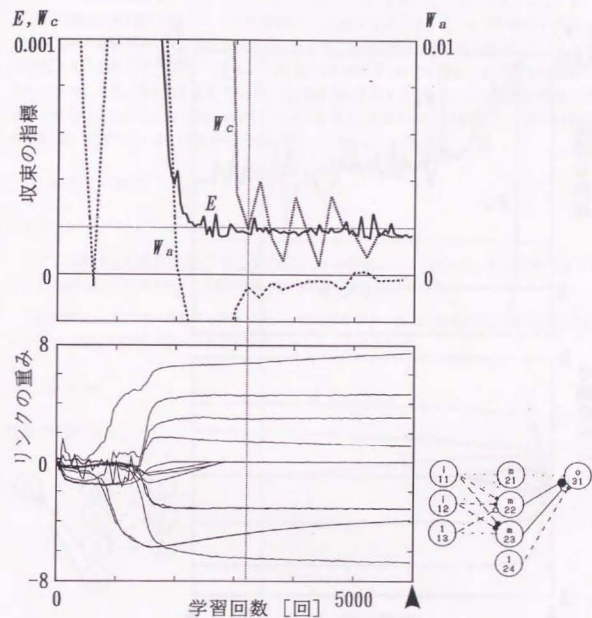
- ある大きさ以下の重みの変化の絶対値の和
($w_{ji}^{k-1,k}$: リンクの重み, t : 学習回数)

$$W_c = \frac{1}{\Delta t} \sum_{|w_{ji}^{k-1,k}| < w_{ct}} |w_{ji}^{k-1,k}(t) - w_{ji}^{k-1,k}(t - \Delta t)| \quad (6.41)$$

忘却付き学習を行った時のリンクの重みの収束の様子と各指標の変化を XOR の入出力パターンおよび、AND の入出力パターンの学習の一例について図 6.25, 6.26 に示す。各定数の設定は、

- 指標のサンプリング間隔 $\Delta t = 200$
- 重みの閾値 $w_{ct} = 2.0$

とした。



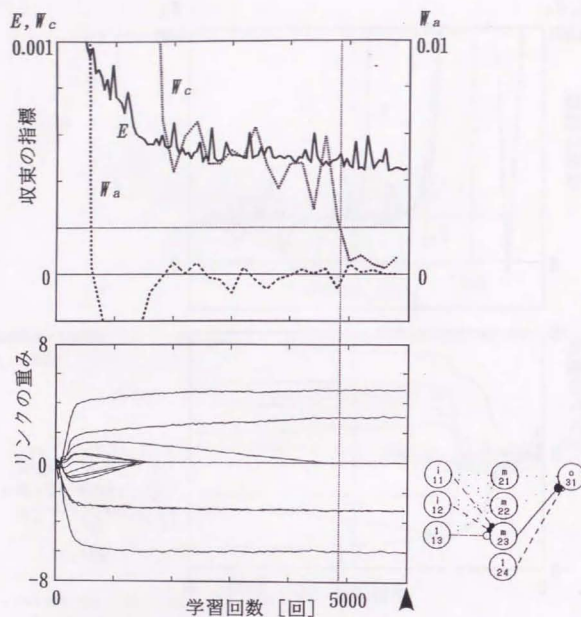
重みの絶対値の和の変化 W_a は全体の和を見ているだけなので、重みが収束する以前に小さくなっている。学習の出力誤差 E の収束に遅れて、リンクの重みが一定値に収束している。学習が収束した後も重みの忘却が続いているためである。重みの変化の絶対値の和 W_c が、リンクの重みの収束とよく連動している。

平均2乗出力誤差 E

重みの絶対値の和の変化 W_a

ある大きさ以下の重みの変化の絶対値の和 W_c

図 6.25: リンクの重みの収束と各指標の変化 (XOR の学習例)



平均2乗出力誤差 E
 重みの絶対値の和の変化 W_a
 ある大きさ以下の重みの変化の絶対値の和 W_c

$$\text{AND: } \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix} \rightarrow 0, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \rightarrow 0, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \rightarrow 0, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \rightarrow 1 \right\} \quad (6.42)$$

図 6.26: リンクの重みの収束と各指標の変化 (ANDの学習例)

リンクの重みの収束の様子と見比べると、重みの絶対値の和の変化 W_a は全体の和を見ているだけなので、重みが収束する以前に小さくなっており、収束の指標としては不適切である。また、学習の出力誤差 E の収束に遅れて、リンクの重みが一定値に収束していることがわかる。学習が収束した後も重みの忘却が続いているためである。そして、重みの変化の絶対値の和 W_c が、リンクの重みの収束とよく連動しているのがわかる。これは、この指標がリンクの各重みの変化を直接的に表したものであり、また、ある大きさ以下のリンクの重みのみを対象としているため忘却の過程が良く反映されるからであると考えられる。よって、忘却付き学習の収束の判定の指標には、重みの変化の絶対値の和 W_c が適している。収束の判定の閾値は、グラフより

$$W_c < 2 \times 10^{-4} \quad (6.43)$$

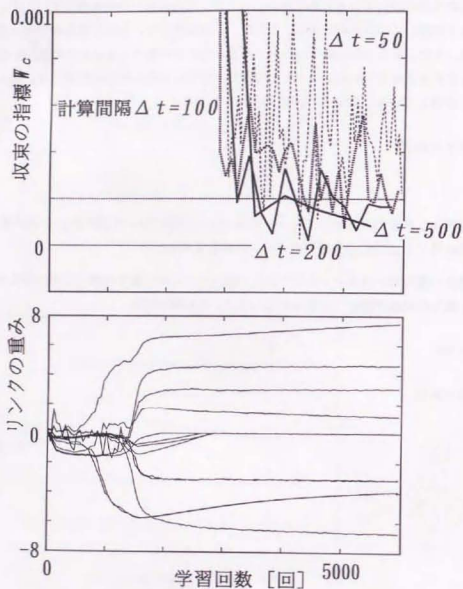
とした。

次に、指標 W_c を計算するサンプリング間隔 Δt の影響について調べた。その結果を図 6.27 に示す。この例は同じく XOR の入出力パターンの学習である。

計算間隔は一種のローパスフィルタであり、短いリンクの重みの細かな局所的な変動が出てしまい、長いと重みの収束が遅れて計算されてしまう。この場合には

$$\Delta t = 200 \quad (6.44)$$

程度が妥当である。



計算間隔は一種のローパスフィルタであり、短いとリンクの重みの細かな局所的な変動が出てしまい、長いと重みの収束が遅れて計算されてしまう。

図 6.27: 計算間隔を変えた時の重みの変化の絶対値の和 W_c の様子

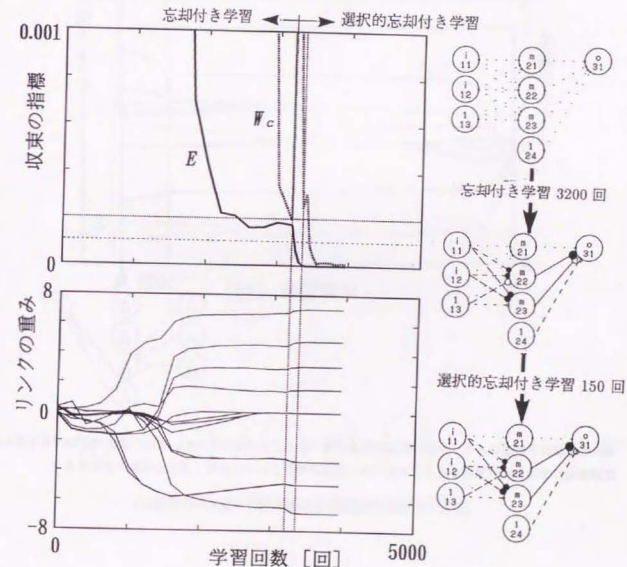
6.6.8 選択的忘却付き学習の効果と収束の判定

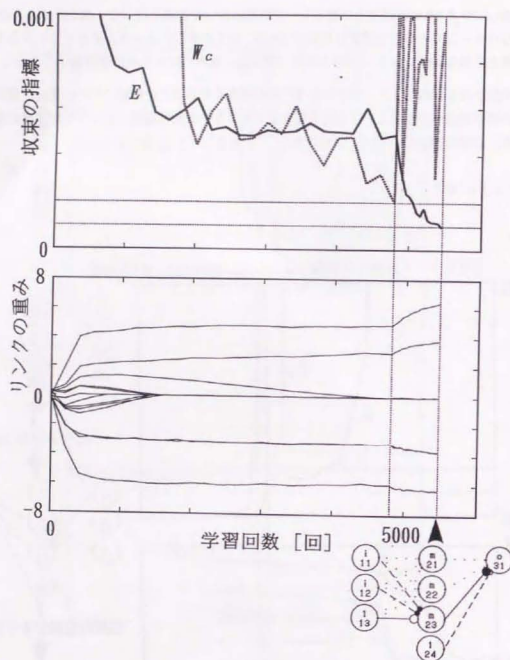
図 6.28, 6.29, 6.30 に忘却付き学習の後、選択的忘却付き学習を行った一例を示す。XOR, AND, 0 の入出力パターンでそれぞれ学習したものである。生じた前段ニューラルネットワークとリンクの重みの成長と忘却の過程、および平均 2 乗出力誤差 E 、重みの変化の絶対値の和 W_c を示してある。

選択的忘却付き学習により、学習の誤差の収束が良くなることが確認された。また、選択的忘却付き学習の学習到達度の改善という意味を考え合わせても、収束の指標としては平均 2 乗出力誤差が適当である。収束の判定の閾値は、グラフより

$$E < 1 \times 10^{-4} \quad (6.45)$$

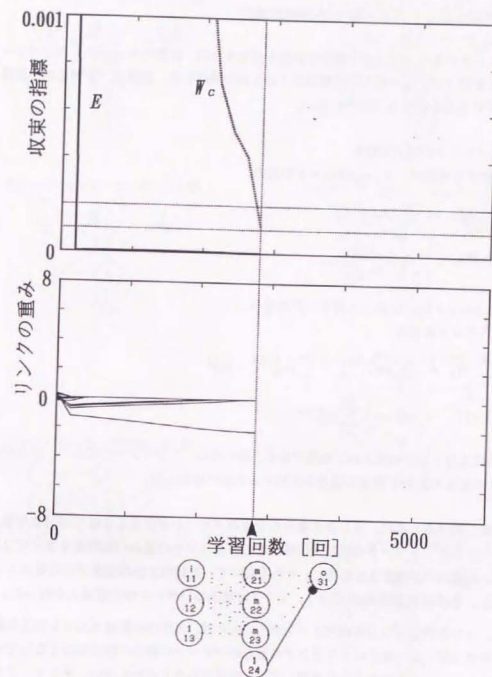
とした。





選択的忘却付き学習により、学習の誤差の収束が良くなることが確認された。また、選択的忘却付き学習の学習到達速度の改善という意味を考え合わせても、収束の指標としては平均2乗出力誤差が適当である。

図 6.29: 選択的忘却付き学習の様子 (ANDの学習例)



$$0: \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix} \rightarrow 0, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \rightarrow 0, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \rightarrow 0, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \rightarrow 0 \right\} \quad (6.46)$$

図 6.30: 選択的忘却付き学習の様子 (0の学習例)

6.6.9 中間層ユニットの並べ換えの指標の検討

後段のニューラルネットワークの識別の負担を減らすため、前段のニューラルネットワークの中間層の並べ換えを行うが、並べ換えの指標は以下のものを検討した。指標は、学習後の中間層ユニットの役割が区別できるものならば何でも良い。

- 中間層ユニットの出力の変化

(p_n :学習する教師データ, a_{pi} :ユニットの出力)

$$T_i = \sum_p (a_{pi} - \bar{a}_i)^2 \quad (6.47)$$

$$\text{ただし} \quad \bar{a}_i = \frac{1}{p_n} \sum_{p=1}^{p_n} a_{pi} \quad (6.48)$$

- 中間層ユニットにつながる入力リンクの重み

($w_{ji}^{k-1,k}$:リンクの重み)

$$W_i = \sum_j (w_{ji}^{k-1,k})^2 + \sum_j (w_{ji}^{k-1,k} - \bar{w}_i)^2 \quad (6.49)$$

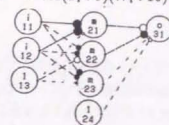
$$\text{ただし} \quad \bar{w}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} w_{ji}^{k-1,k} \quad (6.50)$$

前の項は入力リンクの重みの2乗和である。後の項は、入力リンクの重みの2乗和が同じでも、リンクの重みの分布が異なる場合を区別するために付加した。

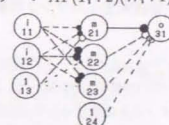
指標の効果を調べるために、リンクの重みの分布のパターンが少なくなる忘却付き学習ではなく、前段のニューラルネットワークの誤差逆伝播学習のみをリンクの重みの初期値を変えて20回行い、様々なリンクの重みの分布を20個得た。それを用いて、上記の2つの指標で中間層ユニットの並べ換えを行った。その結果を図6.31に示す。リンクの重みのパターンの分類は人が行った。

基本的に、リンクの重みの分布のパターンが1つあれば、3個の中間層ユニットの入れ替えの場合の数は6通りあるが、並べ換えによりリンクの重みのパターンの場合の数は少なくなっていることがわかる。しかし、2つの指標のどちらを用いても、結果はあまり変わらない。そこで、ここでは中間層ユニットの出力の変化 T_i を並べ換えの指標として用いることにする。

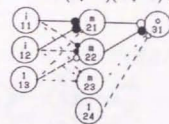
パターン A1($T_i:9$)($W_i:10$) 個



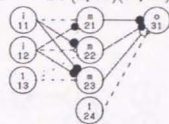
パターン A1'($T_i:2$)($W_i:1$) 個



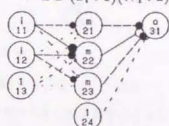
パターン A2($T_i:5$)($W_i:5$) 個



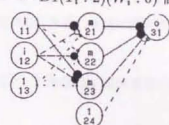
パターン B1'($T_i:1$)($W_i:0$) 個



パターン B1''($T_i:1$)($W_i:2$) 個



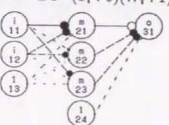
パターン B1($T_i:2$)($W_i:0$) 個



パターン B1'''($T_i:0$)($W_i:1$) 個



パターン B1''''($T_i:0$)($W_i:1$) 個



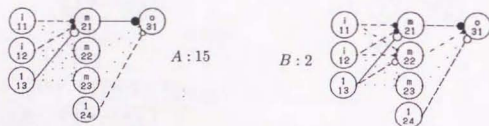
リンクの重みの大きさ: ○ +10 : ● -10

基本的に、リンクの重みの分布のパターンが1つあれば、3個の中間層ユニットの入れ替えの場合の数は6通りあるが、並べ換えによりリンクの重みのパターンの場合の数は少なくなっていることがわかる。しかし、2つの指標のどちらを用いても、結果はあまり変わらない。

図 6.31: 中間層の並べ換えの結果 (20 回) (T_i :出力による, W_i :入力リンクの重みによる)

6.6.10 学習後発生したリンクの重みの分布パターン

前段のニューラルネットワークの忘却付き構造学習後、発生したリンクの重みの分布のパターンを分類した。パターン分類は人が判断した。20の異なるリンクの重みの初期値に対して、AND, OR, XOR, 0の入出力パターンを学習させた時の結果を図6.32, 6.33, 6.34, 6.35に示す。与えられた入出力パターンの組に対して、学習後に発生するリンクの重みの分布のパターンは数種類に限られていることが分かる。よって、これらの限られたパターンは後段のニューラルネットワークで識別できると考えられる。



パターンBはさらに学習を続けるとパターンAになる。収束の速度が遅いため、途中で収束判定の閾値を越えたと考えられる。

図 6.32: 学習後のリンクの重みの分布 (AND) (20 試行中: 学習回数 10000 回までに収束 17)

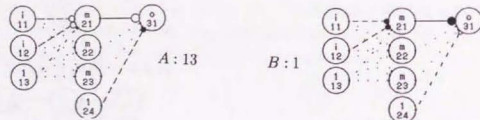
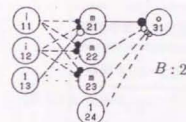
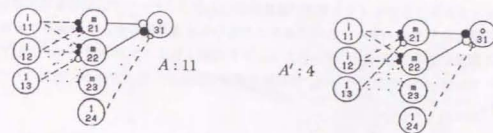


図 6.33: 学習後のリンクの重みの分布 (OR) (20 試行中: 学習回数 10000 回までに収束 14)



パターンA'は中間層ユニットの並べ替えの問題である。パターンBはほぼ同じ中間層ユニットが2つ生じており冗長であるが、差が少ないため、なくなるまでにはかなりかかる。

図 6.34: 学習後のリンクの重みの分布 (XOR) (20 試行中: 学習回数 10000 回までに収束 17)

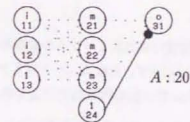


図 6.35: 学習後のリンクの重みの分布 (0) (20 試行中: 学習回数 10000 回までに収束 20)

6.6.11 識別結果

前段のニューラルネットワークを8種類の論理関数の入出力パターンで忘却付き構造学習し、発生したリンクの重みの分布のパターンの中で典型的なものを8種類選んだ。その重みの分布のパターンと対応する論理関数の種類（対応する出力ユニットのみ1他は0）を学習データとして、後段のニューラルネットワークの学習を行った。学習法は通常の誤差逆伝播法である。学習のパラメータを表6.4に示す。

表 6.4: 後段ニューラルネットワークの学習パラメータの設定値

パラメータ		誤差逆伝播学習時
リンクの重みの初期値	$w_{ji}^{k-1,k}$	-0.3~+0.3 乱数
重みの修正速度の係数	α	0.9
誤差逆伝播学習の係数	ϵ	0.5
学習回数	t	20000

その結果、学習の平均2乗出力誤差は

$$E = 0.0028 \quad (6.51)$$

まで収束し（一例）、学習に用いた8種類の典型的なリンクの重みの分布に対してはすべて識別した。その例を図6.37, 6.38, 6.39, 6.40, 6.41に示す。後段ニューラルネットワークのユニットの出力値は、黒の円の大きさに比例させて表示した。枠一杯の円は出力が1である。

前段のニューラルネットワークのリンクの重みの分布を後段のニューラルネットワークで識別可能であることが分かる。論理関数の場合には、発生する前段のネットワークのリンクの重みの分布が明らかに異なるため、このパターン認識は容易であるといえる。

図6.36は、前段のニューラルネットワークの学習前のリンクの重みの分布パターンに対する認識結果である。このように、学習に用いた典型的なパターンとかなり異なる学習していない重みのパターンを入力した場合は識別を誤る。

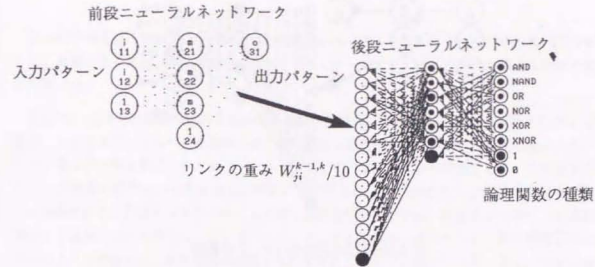


図 6.36: 前段のニューラルネットワークの学習前のリンクの重みの識別

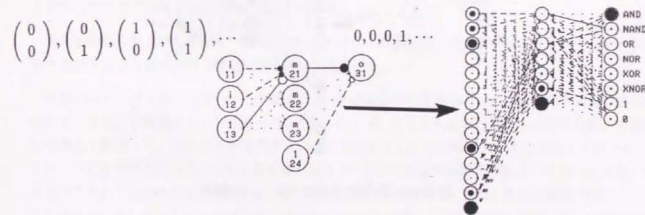


図 6.37: ANDの入出力パターンの識別

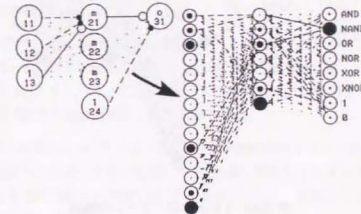


図 6.38: \overline{AND} の入出力パターンの識別

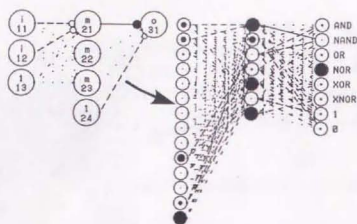


図 6.39: OR の入出力パターンの識別

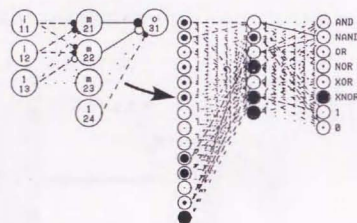


図 6.40: XOR の入出力パターンの識別

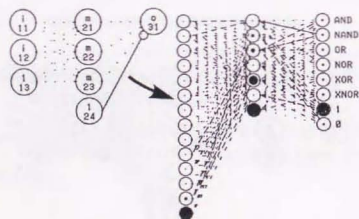


図 6.41: 1 の入出力パターンの識別

6.7 まとめ

工学的に最も良く用いられている多層の階層的ニューラルネットワークの構造と動作について記述した。学習法としては、誤差逆伝播（バックプロパゲーション）法と忘却付き構造学習法の理論について述べた。

そして、入出力の写像の種類を検出する問題を設定し、それを解決する一手法として、2段の階層的ニューラルネットワークを提案した。学習後の入出力関係が反映された前段のネットワークのリンクの重みの分布を後段のネットワークの入力としている点が大きな特徴である。この2段のネットワークの性質や動作について言及し、前段と後段のネットワークに分けて学習アルゴリズムを示した。基本的には、前段ネットワークには忘却付き構造学習法を毎回、後段ネットワークには誤差逆伝播法を1回用いる。他のニューラルネットワークによる実現法と比較すると、特に順序関係のない入出力パターンの組から、その写像の種類を検出するのに適しているといえる。また、線形写像を仮定すると行列の要素が前段ニューラルネットワークのリンクの重みに相当し、前後段の2段構造は自然な発想である。

さらに、本研究で用いた各コンピュータ（パソコン、DSP ボード、ワークステーション）にあわせて自作したプログラムや実行速度について述べた。

また、構築した2段の階層的ニューラルネットワークの基礎的な性質を明らかにするために、簡単な2入力1出力の論理関数の識別問題に対して適用した。

前段のニューラルネットワークの学習は、リンクの重みの初期値のセット、忘却付き学習、選択的忘却付き学習、中間層ユニットの並べ替えの順に行った。リンクの重みの成長・忘却の過程、学習の出力誤差を監視して、忘却付き構造学習法により冗長なリンクの重みが少なくなることを確かめた。また、むだな学習を行わないため、各学習の過程での収束の判定の指標を検討し、忘却付き学習ではある大きさ以下の重みの変化の絶対値の和、選択的忘却付き学習では平均2乗出力誤差が適していると分かった。その時のサンプリング間隔や閾値についても考察し、適切な値を設定した。中間層の並べ替えの指標は、ユニットの出力の変化を用いることにした。さらに、前段ニューラルネットワークの学習をリンクの重みの初期値を変えて複数回行い、発生するリンクの重みの分布パターンが数種類に限られていることを確認した。

後段のニューラルネットワークの学習は、前段のネットワークで生じた典型的な8種類のリンクの重みのパターンを用いて誤差逆伝播学習を行った。学習則の重みの修正量のバランスをとるため、前段のリンクの重みの1/10を入力とした。学習に用いた8種類のリンクの重みのパターンを識別し、それぞれに対する論理関数の種類を出力することができた。

以上より、2段の階層的ニューラルネットワークが実際に動作することが確かめられた。しかし、これは簡単な論理関数の入出力パターンを用いているため、学習によって生じた前段ニューラルネットワークのリンクの重みのパターンが明らかに異なるからであり、複雑な入出力パターンに対してはさらに検討が必要である。

第7章 平面3自由度の拘束と2段の階層的ニューラルネットワークによる認識

7.1 はじめに

まず、本研究で取り扱う平面3自由度の拘束とその分類について示す。それに対応する可動空間について述べ、拘束の姿勢の表現について検討する。

次に、平面3自由度の拘束(可動空間)を識別するために、本研究で提案する前段と後段のネットワークからなる2段の階層的ニューラルネットワークを適用し、学習方法と識別手順について述べる。

分類された16種類の拘束に対して、その可動空間を前段ニューラルネットワークに学習させて、学習後発生するリンクの重みの分布の関係や挙動を調べる。そのとき、中間層や出力層のユニットの出力(活性度)をそのユニットが反応している空間で表現し、中間層の解釈を試みる。

さらに、可動空間を学習した前段のニューラルネットワークのリンクの重みを用いて後段のニューラルネットワークの学習を行い、リンクの重みから拘束の種類や姿勢が認識できることを示す。

7.2 平面3自由度の拘束

7.2.1 典型的な拘束

本研究では、平面3自由度の拘束を取り扱う。図7.1に16種類の典型的な拘束を示す。

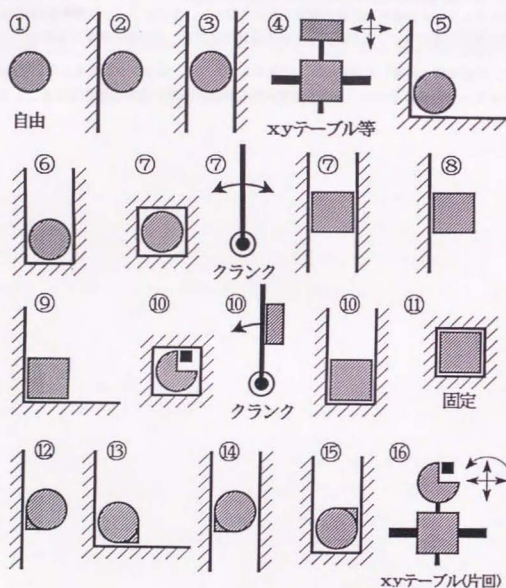


図 7.1: 平面3自由度の典型的な拘束の種類

これらは、許される微小な変位によって分けてある。例えば、(7)の拘束の場合、見かけは異なるが、クランクの回転軸上を把握している場合やクランクの無限遠方を把握している場合と考えると同一種類である。

7.2.2 拘束の分類

平面内運動は並進2自由度と回転1自由度である。それぞれの自由度に対して、片側の拘束と両側の拘束に着目して、分類を行うと表7.1のようになる。

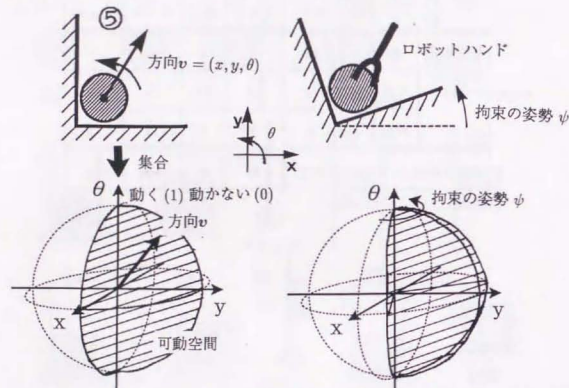
表 7.1: 平面3自由度の典型的な拘束の分類

		並進方向 (x, y)					
		無	片側	片+片	両側	片+両	両+両
回転方向 (θ)	無	1	2	5	3	6	7
	片側	16	12	13	14	15	10
	両側	4	8	9	7	10	11
並進+回転 (x, y, θ) (1自由度)		無	7	片側	10		

7.3 平面3自由度の可動空間

7.3.1 可動空間と拘束の姿勢

ロボットが把持する対象物の拘束状態は、空間のある方向 v に動くか(可動1) 動かないか(不動0)で表現できる。そして、動ける方向の集合が可動空間になる。可動空間は全体の空間の部分空間であり、拘束状態を表している。平面内運動は並進2、回転1自由度(x, y, θ)であるので、可動空間は3次元(x, y, θ)で表現できる。図7.2に典型的な拘束に対する可動空間の例を示す。

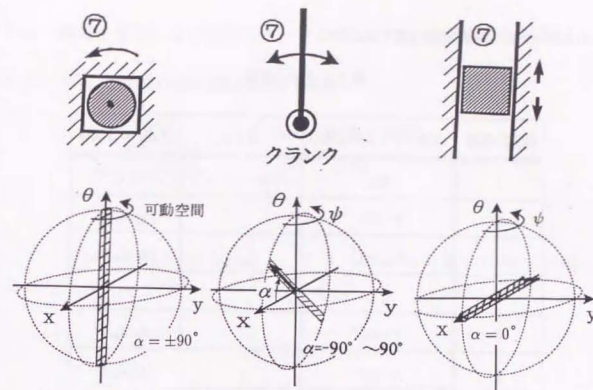


平面内運動の可動空間は、3次元(x, y, θ)の動くか動かないかの集合で表され、拘束の姿勢 ψ によって並進自由度 θ の回りに回転する

図 7.2: 平面3自由度の拘束に対する可動空間の例

ここで、注意しなければならないのは、拘束の姿勢(向き) ψ である。例えば、ロボットが把持している円柱に対して、当たっている角がどちらを向いているかということである。拘束の姿勢 ψ が変わると、図のように可動空間(の向き)は θ 軸回りに回転する。よって、拘束状態を識別するためには、拘束の種類とその姿勢を検出しなければならない。

図7.3は、可動空間を考えると同じ形状である拘束の例である。回転する拘束の軸上を把握した場合、回転 θ だけが許される。角柱が両側を壁で囲われている拘束の場合、動けるのはある並進の直線上だけである。クランクの拘束の場合は、微小な回転と並進が一定の割合で生じる。それぞれ、可動空間は空間の一直線となり、同じ種類の拘束として扱うことができる。また、その直線の傾きは図のように傾き α ($-90^\circ \sim 90^\circ$)で表せる。

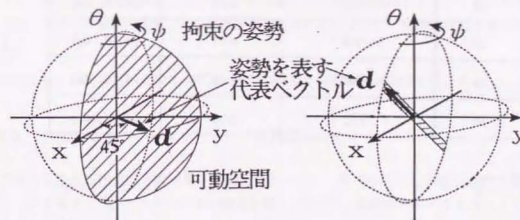


これらの拘束の可動空間は空間の一直線となり、同じ種類の拘束として扱う

図 7.3: 可動空間の形が同じである拘束の例

7.3.2 拘束の姿勢を表す代表ベクトル

拘束の姿勢(可動空間の向き)は、拘束のある部分の姿勢(角度) ψ で表すことができるが、値の連続性や同じ形状の可動空間の向き等を考慮すると、可動空間の向きを表すには不便である。そこで、可動空間に対してある一定の方向に、拘束の姿勢を表す代表ベクトル d を定め、このベクトルの向きで拘束(可動空間)の姿勢(向き)を表すことにした(図7.4)。



拘束の姿勢 ψ を表現する方法として、可動空間のある一定の方向に定めた代表ベクトル d を用いる

図 7.4: 拘束の姿勢(可動空間の向き)を表す代表ベクトル

そのときの拘束の姿勢の範囲を表 7.2 に示す。

表 7.2: 拘束の姿勢

拘束の種類	代表ベクトルの姿勢 ψ	傾き α	代表ベクトル d
1	なし		なし
2	$0 \sim 360^\circ$		$(d_x, d_y, 0)$
3	$0 \sim 180^\circ$		$(d_x, d_y, 0)$
4	なし		なし
5	$0 \sim 360^\circ$		$(d_x, d_y, 0)$
6	$0 \sim 360^\circ$		$(d_x, d_y, 0)$
7	$0 \sim 180^\circ$	$0, \pm 30, \pm 60, 90^\circ$	(d_x, d_y, d_θ)
8	$0 \sim 360^\circ$		$(d_x, d_y, 0)$
9	$0 \sim 360^\circ$		$(d_x, d_y, 0)$
10	$0 \sim 360^\circ$	$0, \pm 30, \pm 60, \pm 90^\circ$	(d_x, d_y, d_θ)
11	なし		なし
12	$0 \sim 360^\circ$	$\pm 45^\circ$	(d_x, d_y, d_θ)
13	$0 \sim 360^\circ$	$\pm 45^\circ$	(d_x, d_y, d_θ)
14	$0 \sim 180^\circ$	$\pm 45^\circ$	(d_x, d_y, d_θ)
15	$0 \sim 360^\circ$	$\pm 45^\circ$	(d_x, d_y, d_θ)
16	なし	$\pm 90^\circ$	(d_x, d_y, d_θ)

7.4 拘束を検出する 2 段のニューラルネットワークの構造と学習法

7.4.1 平面 3 自由度の可動空間を識別する問題

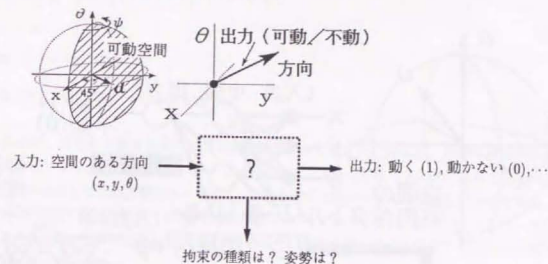


図 7.5: 可動空間 (拘束) を識別する問題

平面 3 自由度の可動空間 (拘束) の識別を考える。

ロボットの場合、環境の認識から直接それに基づいた動作を生成することも考えられる。例えば、探り動作により拘束の動く方向を発見したら、拘束の種類等を認識することなくそのまま動く方向に動かすような動作もある。つまり、未知の拘束の入力に対して直接ロボットの動く方向を出力するような構造をとることもできる。しかし、一旦、全体の拘束の種類を検出した上で、他の環境の状況や動作の目的等も検討して、複数の候補の中からそれに適した制御法や動作計画を選択する方がより柔軟なロボットの動作が実現できると考えられる。

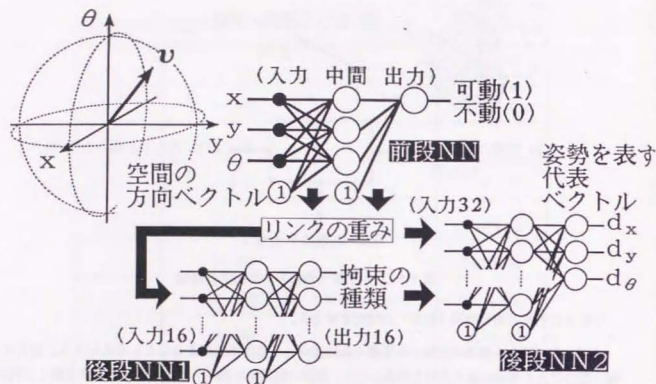
可動空間は空間のある方向 (x, y, θ) が動くかどうか (1/0) を表したものである。いろいろな方向について、動くかどうかを観測して、その可動空間の種類を識別しようとする問題になる。前節で述べたように、対象物の姿勢により可動空間の姿勢がかわるため、姿勢の検出も合わせて行う必要がある。

ここでは、前節で述べた図 7.1 の 16 種類の拘束 = 可動空間の識別を考える。

7.4.2 2 段のニューラルネットワークの構造

平面 3 自由度の拘束 (可動空間) を識別するために、前述の入出力画像の種類を識別する 2 段のニューラルネットワークを使う。その構造を図 7.6 に示す。前段のニューラルネットワークのリンクの重みに可動空間を反映させようとするものである。

前段のニューラルネットワークの構造は 3 層構造で、ユニット数は入力層 3、中間層 3、出力層 1 (1 を出力する閾値に相当するユニットは含まない) である。中間層は 16 種類の典型的な可動空間を学習するのに十分な個数をとった。入力、適当にサンプリングされる空間の方向ベクトル (x, y, θ)



前段のニューラルネットワークのリンクの重みに可動空間を反映させ、その分布パターンから後段ニューラルネットワークが拘束を識別する。

図 7.6: 平面 3 自由度の拘束を識別する 2 段のニューラルネットワークの構造

であり、出力はその方向が可動か (1) 不動か (0) である。ある拘束が与えられる度に前段ニューラルネットワークの学習を行うと、可動空間がリンクの重みで補間されて表現される。つまり、可動空間を反映したリンクの重みが生じる。逆に、このリンクの重みの分布のパターンから、可動空間の種類と姿勢を後段のニューラルネットワークで識別する。前段のリンクの重みは $16(3 \times 4 + 4)$ 個あり、これらを後段ニューラルネットワークの入力へ直接送る。実際には、後段ニューラルネットワークの学習のバランスを考慮し、重みの値を $1/10$ にして送っている。

後段のニューラルネットワークは 2 つのネットワークから成る。拘束の種類と姿勢の識別を 2 つのネットワークで分けて行っている。後段ニューラルネットワークの 1 つは 3 層構造で、入力層のユニットは前段のリンクの重みの数の 16 個、出力層のユニットは拘束の種類に対応して 16 個ある。入力は前段ニューラルネットワークのリンクの重みで、出力は拘束の種類である。前段のリンクの重みの分布が入力されると、理想的に拘束の種類が識別された場合には、出力層のその種類に対応するユニットのみが 1 を出力し、後のユニットは 0 を出力する。

もう 1 つの 3 層の後段ニューラルネットワークはその拘束の識別結果と前段ニューラルネットワークのリンクの重みを入力して、可動空間の姿勢を表す代表ベクトル $d = (d_x, d_y, d_\theta)$ を出力する。ニューラルネットワークの出力として、

- ベクトル
- 角度 ($0^\circ \sim 360^\circ$ 等)
- 方向ごとに 1 つのユニットを割り当てる

があげられるが、連続した値をとるベクトルが適していると考えた。従って、姿勢を検出する後段ニューラルネットワークの入力層のユニットは、合わせて 32 個 (前段のリンクの重み 16 + 拘束の種類 16) で、出力層のユニットは、姿勢を表す代表ベクトルに対応して 3 個である。代表ベクトル d を出力する出力層のユニットの値の範囲は $0 \sim 1$ であり、また、実際にネットワークが出力するベクトル d の大きさは必ずしも一定ではないので、出力に対して 0.5 をオフセットとして正規化を行っている。

2 つの後段ニューラルネットワークの学習は予め行っておく。

7.4.3 拘束を識別する手順

拘束を識別する手順は、まとめると図 7.7 のようになる。

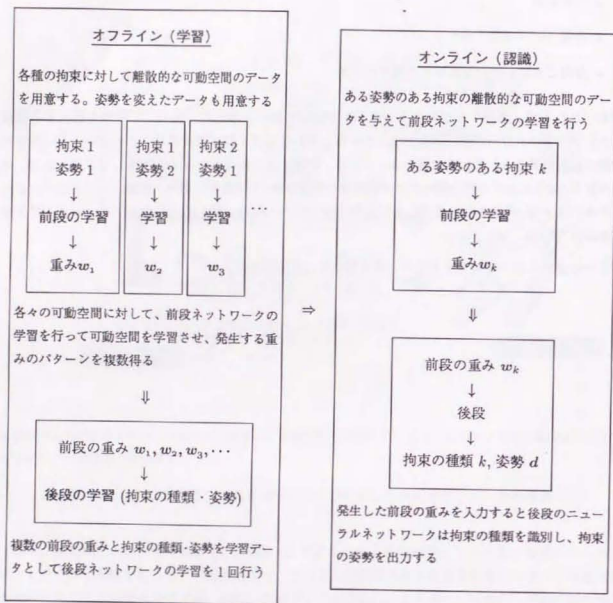


図 7.7: 拘束の識別の手順

7.4.4 前段ニューラルネットワークの学習

基本的には、同じ可動空間に対し、初期値や学習順序等による様々なリンクの重みの発生を減らすため、忘却付き構造学習を行い冗長な重みを消去する。学習データは図 7.8 に示すような適当にサンプリングした空間のいくつかの方向のデータを用いて行う。前段ニューラルネットワークの忘却付き構造学習法を基礎とした学習の順序とそれぞれの段階の学習の収束の判定は次のようになる。

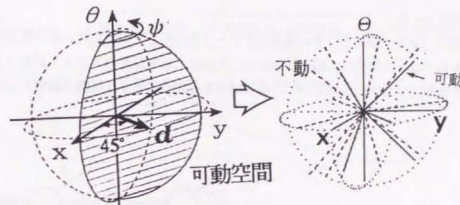


図 7.8: サンプリングされた可動空間の学習データの例

学習則

学習則は速度修正型を用い、学習データの提示順序はランダムである。各段階の学習の切替えは、次式の各項の係数 $\alpha, \epsilon, \epsilon_j, w_t$ の値を変化させることにより一括して行う。

$$\begin{aligned} \Delta w_{ji}^{k-1,k}(t) &= \alpha \Delta w_{ji}^{k-1,k}(t-1) \\ &\quad - \epsilon \delta_i^k o_j^{k-1} \quad (\text{誤差逆伝播学習}) \\ &\quad - \epsilon_j \operatorname{sgn}(w_{ji}^{k-1,k}) \quad (\text{if } |w_{ji}^{k-1,k}| < w_t) \quad (\text{選択的忘却付き学習}) \end{aligned} \quad (7.1)$$

$w_{ji}^{k-1,k}$: $k-1$ 層 j ユニートから k 層 i ユニートへのリンクの重み
 i_i^k, o_i^k, δ_i^k : k 層 i ユニートの入力、出力、および、逆伝播された出力誤差

収束判定の指標

各学習の収束の判定は次の指標を使い分ける。

- 学習回数 t
 各学習過程で一定の回数 t を越えて収束しない場合には、ニューラルネットワークのリンクの重みを初期化して学習をやり直す。
- 平均 2 乗出力誤差
 $(p_n: \text{学習する教師データの数}, o_{pi}^m, t_{pi}: \text{出力層の出力と望ましい出力})$

$$E = \frac{1}{2p_n} \sum_{p,i} (t_{pi} - o_{pi})^2 \quad (7.2)$$

- ある大きさ以下の重みの変化の絶対値の和
($w_{ji}^{k-1,k}$: リンクの重み, t : 学習回数)

$$Wc = \frac{1}{\Delta t} \sum_{|w_{ji}^{k-1,k}| < w_{cr}} |w_{ji}^{k-1,k}(t) - w_{ji}^{k-1,k}(t - \Delta t)| \quad (7.3)$$

リンクの重みのチェック

学習の効率化を図り、ある一定のリンクの重みのパターンを発生させるために、各学習が収束した段階でリンクの重みのチェックを行い、同じ層の中で、次のような中間層のユニットがあった場合には(図 7.9)、その中間層につながる入出力のリンクの重みを初期化(小さな乱数に設定)して、学習をやり直す。

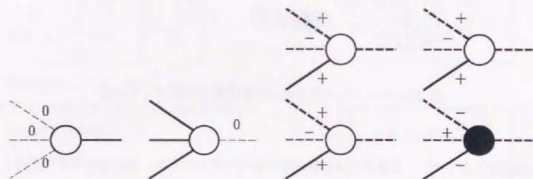


図 7.9: 中間層のユニットのリンクの重みのチェック

- 入力のリンクの重みが0に近いのに、出力のリンクの重みが0に近くない場合
- 出力のリンクの重みが0に近いのに、入力のリンクの重みが0に近くない場合

これらは忘却不十分の状態である。0に近いかどうかは閾値 W_c で判断する。

- 入力のリンクの重みがほぼ等しい中間層ユニットがある場合これは、同じ役割を果たしているユニットが2つ存在している状態である。
- 入力のリンクの重みの符号が逆で大きさがほぼ等しい中間層ユニットがある場合これは、補空間の役割を果たしているユニットが2つ存在している状態である。

これらの場合はこのまま学習を続けても収束が非常に遅くなる。そのため、どちらか一方のユニットに対して初期化を行って学習をやり直す。2つのユニット A, B の入力のリンクの重みを w_{jiA}, w_{jiB} とすると、閾値 W_c を用いて、それぞれ

$$|w_{jiA} \pm w_{jiB}| < W_c \quad (7.4)$$

で同じか、逆か、それ以外か、判断している。

学習順序

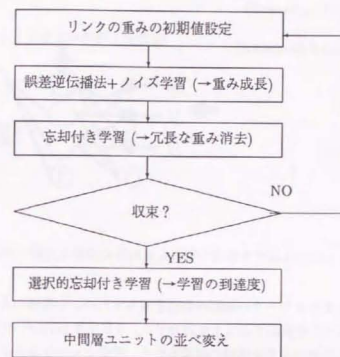


図 7.10: 前段ニューラルネットワークの学習の手順

- リンクの重みの初期値として小さな乱数を与える。
- 誤差逆伝播法により、リンクの重みを成長させる。このとき、学習データの空間の方向ベクトル (x, y, θ) の方向のみ偏った学習が進むのを防ぎ、ニューラルネットワークの補間の機能を高めるため、ノイズ学習を行う。つまり、出力の学習データは同じ状態で、入力ベクトルに微かなランダムノイズを加えて学習を行う。学習させる1つの方向ベクトルに対して、

$$(x + \delta_x, y + \delta_y, \theta + \delta_\theta) \quad |\delta_x| < \eta, |\delta_y| < \eta, |\delta_\theta| < \eta \quad (7.5)$$

を学習データとして用いる。なお、平均2乗出力誤差 E がある閾値以下になれば収束したと判断する。

- 忘却付き学習を行い、リンクの重みを成長させると同時に、冗長なリンクを消去する。誤差逆伝播学習の係数 α, ϵ 、忘却学習の係数 ϵ_f を設定し、忘却する重みの閾値 w_f は大きな値にする。他の係数は0とする。そして、学習の収束の判定は、重みの絶対値の和の変化 W_c を指標として用いる。この収束の指標がある閾値以下になれば、次の学習に移る。また、ある回数以上学習しても収束しない場合は、もう一度、リンクの重みを初期化し再学習する。

- (4) 選択的忘却付き学習を行って、学習の収束を図る。学習を促進するため誤差逆伝播学習の係数 ϵ を大きくし、忘却する重みの閾値 w_i を適当な値に設定する。他の係数はそのままである。そして、学習は一定回数行う。

$$\beta = \frac{\pi}{12} i \quad (i = -6, \dots, 6)$$

$$\alpha = \frac{\pi}{12 \cos \beta} j \quad (j = 0, \dots, 24 \cos \beta)$$

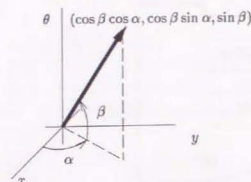


図 7.11: ユニットが反応する空間の割合を求めるための入力層への入力ベクトル

- (5) 後段のニューラルネットワークの識別の負担を減らすため、中間層の入れ換えを行う。入れ替えの指標は、ユニットが反応する（1 を出力する）3 自由度 (x, y, θ) の空間の大きさの割合 A_i を用いた。実際には次のような指標の計算を行う。前段ニューラルネットワークの入力層への入力ベクトル (x, y, θ) を図 7.11 のように、全入力空間に渡って、均一に変化させた時（全回数 n ）、ユニットが 1 に近い値を出力した回数を n_1 とすると

$$A_i = \frac{n_1}{n} \quad (7.6)$$

となる。これは、ニューラルネットワークへの入力空間のどれくらいの大きさの部分に対してそのユニットが反応しているかを示している。中間層のユニット毎にこの指標を計算し、大きさの順に、同じ層の中で並べ変えを行う。その後、前段のネットワークの重みを後段へ送る。

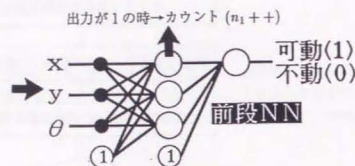
7.4.5 後段ニューラルネットワークの学習

16 種類の拘束（可動空間）に対して、それぞれ前段のニューラルネットワークの学習を行って、前段のネットワークのリンクの重みのパターンを収集する。それらを用いて後段の 2 つのニューラルネットワークの学習を行う。望ましい出力は、1 つの後段ニューラルネットワークについては、拘束の種類に対応するユニットのみ 1 を出力し、他は 0 となる。もう一つの後段ニューラルネットワークについては、拘束の姿勢を表す代表ベクトルである。学習法は通常の誤差逆伝播学習を使用する。この後段の学習は予め 1 回だけ行っておけば良い。

リンクの重み $w_{ji}^{k-1,k}$ の学習則は、次のように誤差逆伝播学習の速度による修正アルゴリズムを用いている。シグモイド関数を f 、ユニットの入力の和 \sum 、出力 o_i^k 、望ましい出力 t_i^k として

$$\Delta w_{ji}^{k-1,k}(t) = \alpha \Delta w_{ji}^{k-1,k}(t-1) - \epsilon \delta_i^k o_j^{k-1} \quad (7.7)$$

$$\delta_i^k = \begin{cases} (\sum_l \delta_l^{k+1} w_{il}^{k,k+1}) f'(i_i^k) & (k \neq m) \\ (o_i^m - t_i^m) f'(i_i^m) & (k = m) \end{cases} \quad (7.8)$$



7.5 前段ニューラルネットワークの挙動

7.5.1 学習データと学習のパラメータの設定

前段ニューラルネットワークの学習に用いたデータの種類と個数を表 7.3 に示す。拘束の種類番号は図 7.1 に対応している。

それぞれの可動空間に対して、いくつかの離散的な方向のデータを用いて学習を行っている。

表 7.3: 拘束の種類と学習データ

拘束の種類	代表ベクトルの姿勢 ϕ	傾き α	学習データ数
1	なし		20
2	$0 \sim 360^\circ$		240
3	$0 \sim 180^\circ$		120
4	なし		20
5	$0 \sim 360^\circ$		240
6	$0 \sim 360^\circ$		240
7	$0 \sim 180^\circ$	$0, \pm 30, \pm 60, 90^\circ$	$120 \times 5 + 20$
8	$0 \sim 360^\circ$		240
9	$0 \sim 360^\circ$		240
10	$0 \sim 360^\circ$	$0, \pm 30, \pm 60, \pm 90^\circ$	$240 \times 5 + 20 \times 2$
11	なし		20
12	$0 \sim 360^\circ$	$\pm 45^\circ$	240×2
13	$0 \sim 360^\circ$	$\pm 45^\circ$	240×2
14	$0 \sim 180^\circ$	$\pm 45^\circ$	120×2
15	$0 \sim 360^\circ$	$\pm 45^\circ$	240×2
16	なし	$\pm 90^\circ$	20×2
計			4960

これらの学習データを用いて前節に述べたような学習順序で、前段のニューラルネットワークの学習を行った時の標準的な係数等の設定を表 7.4 に示す。これらは、試行錯誤により定めたものである。

表 7.4: 前段ニューラルネットワークの学習パラメータの設定値

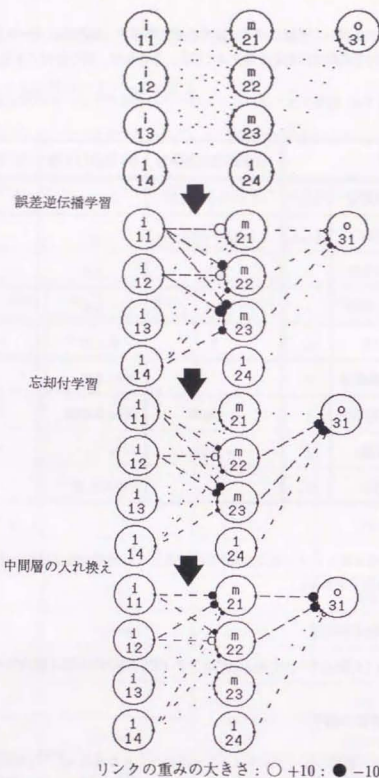
係数		(1) 誤差逆伝播学習	(2) 忘却付学習	(3) 選択的忘却付学習
リンクの重みの初期値	w_{ji}	$-0.3 \sim +0.3$ 乱数		
ノイズ学習の乱数	$\delta_{x,y,\theta}$	$-0.5 \sim +0.5$		
重みの修正速度の係数	α	0.9	0.9	0.9
誤差逆伝播学習の係数	ϵ	0.5	0.5	0.5
忘却の係数	ϵ_f	0	1×10^{-4}	1×10^{-4}
選択的忘却の重みの閾値	w_t		100	1.0
収束の判断 (学習回数)	t	< 10000	< 20000	2000
(平均 2 乗出力誤差)	E	< 0.01		
(重みの変化)	W_c		$< 3 \times 10^{-4}$	

また、前段ニューラルネットワークの全体の学習回数とその計算時間は収束の速度により異なるが、再学習を除くと次の通りである。

- 学習回数 3000 ~ 30000 回
- 計算時間 0.7 ~ 7.5 秒 (パーソナルコンピュータ EPSON PC486RS 使用時)

7.5.2 忘却付き学習の様子

図 7.12 に学習に伴う前段ニューラルネットワークのリンクの重み $w_{ji}^{k-1,k}$ の成長と忘却の様子を示す。ネットワークのリンクの重みの大きさは入力端の円の直径及びリンクの点線の長さで、符合は●と○で図示してある。誤差逆伝播学習でリンクの重みが成長した後、忘却付きの学習により、冗長な重みが消去されているのがわかる。



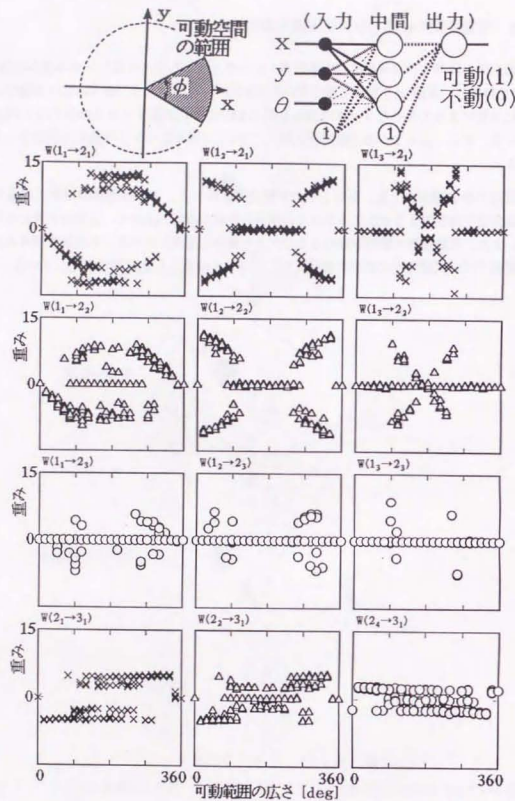
誤差逆伝播学習でリンクの重みが成長した後、忘却付きの学習により、冗長な重みが消去されているのがわかる。

図 7.12: 忘却付き学習の様子

7.5.3 可動空間の範囲とリンクの重みの関係

可動空間の範囲と学習後発生した前段のニューラルネットワークのリンクの重みの関係を調べた。回転 θ を無視し、並進 xy 平面で可動空間の範囲 ϕ を $0^\circ \sim 360^\circ$ 、 15° おきに变化させて、各々の範囲の時に学習によって生じたリンクの重みを図 7.13に示す。学習データは離散的な方向のデータを用いている。また、ある1つの可動空間に対して10個の異なるリンクの重みの初期値から学習を行っている。

初期値や学習順序により、ある1つの可動空間に対して、1～数種類のリンクの重みが生じている。誤差逆伝播学習よりは生じるリンクの重みの分布の種類が少なく、忘却付き学習の効果が現われている。また、可動空間の範囲が変わるとリンクの重みも変化している。不連続な所もあるが、ある程度の範囲では、可動空間の範囲の変化に対してリンクの重みも連続的に変化している。



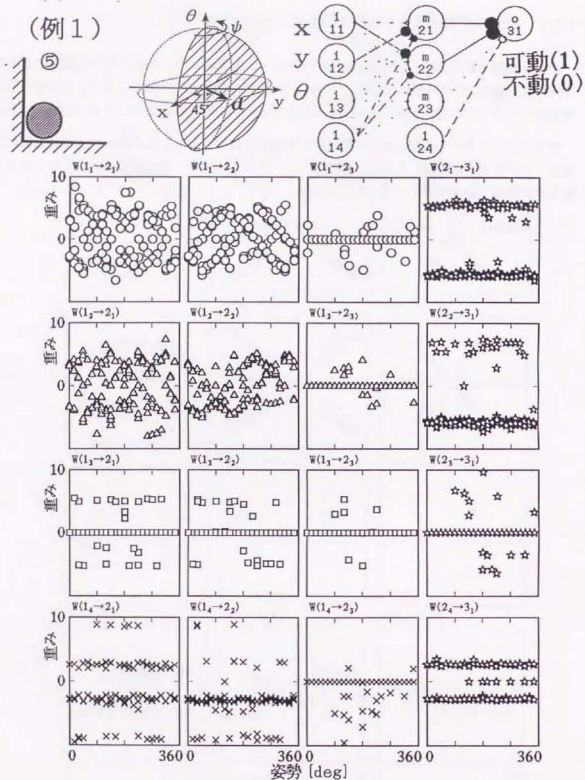
初期値や学習順序により、ある1つの可動空間に対して、1～数種類のリンクの重みが生じている。可動空間の範囲が変わるとリンクの重みも変化している。不連続な所もあるが、ある程度の範囲では、可動空間の範囲の変化に対してリンクの重みも連続的に変化している。

図 7.13: 可動空間の範囲とリンクの重みの関係

7.5.4 拘束の姿勢とリンクの重みの関係

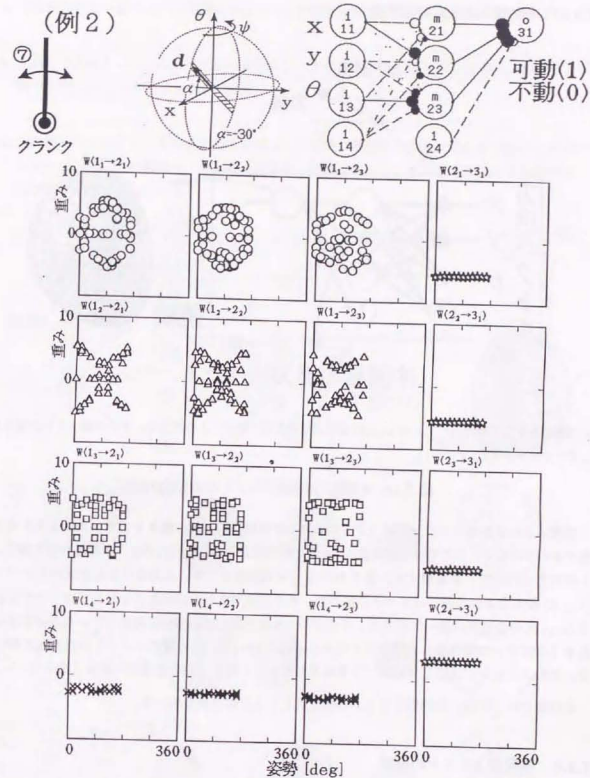
次に、拘束（可動空間）の姿勢φが前段のニューラルネットワークのリンクの重みにどう反映するか調べた。図 7.1 中の2つの拘束について、拘束の姿勢φを $0^\circ \sim 360^\circ$ 、 15° おきに变化させて、学習後生じたリンクの重みを図 7.14, 7.15 に示してある。1つの姿勢に対して10回学習を行っている。

かなりばらつきがあるが、大きく見れば、姿勢が変化するとリンクの重みが三角関数的に連続して変化している部分がある。これは、後段のニューラルネットワークで前段のニューラルネットワーク重みから姿勢を出力するのに良い性質である。



かなりばらつきがあるが、大きく見れば、姿勢が変化するとリンクの重みが三角関数的に連続して変化している部分がある。

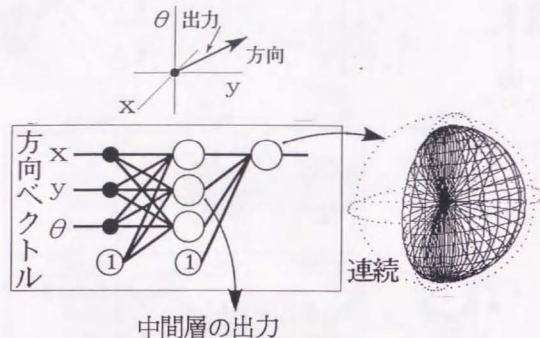
図 7.14: 拘束の姿勢とリンクの重みの関係



かなりばらつきがあるが、大きく見れば、姿勢が変化するとリンクの重みが三角関数的に連続して変化している部分がある。

図 7.15: 拘束の姿勢とリンクの重みの関係

7.5.5 中間層ユニットの活性度



入力層にある空間の方向ベクトル (x, y, θ) が入力された時、ユニットの出力を、その方向ベクトルの大きさとしてプロットする。

図 7.16: 中間層、出力層ユニットの活性度の表現

前段ニューラルネットワークのリンクの重みや中間層ユニットの働きをブラックボックスのまま使用するのではなく、ここでは、中間層ユニットの解釈を試みる。そのために、中間層や出力層のユニットの出力（活性度）を表現する。図 7.16 のような表現法をとる。入力層にある空間の方向ベクトル (x, y, θ) が入力された時、ユニットの出力を、その方向ベクトルの大きさとしてプロットする。それを (x, y, θ) の全空間に渡って行くと、そのユニットが空間のどの部分に反応しているかが分かる。これを「ユニットの活性度」と呼ぶことにする。図 7.16 には、出力層のユニットの活性度の例を示した。そのユニットが反応している（1 を出力している）部分（空間の方向）が良く分かる。

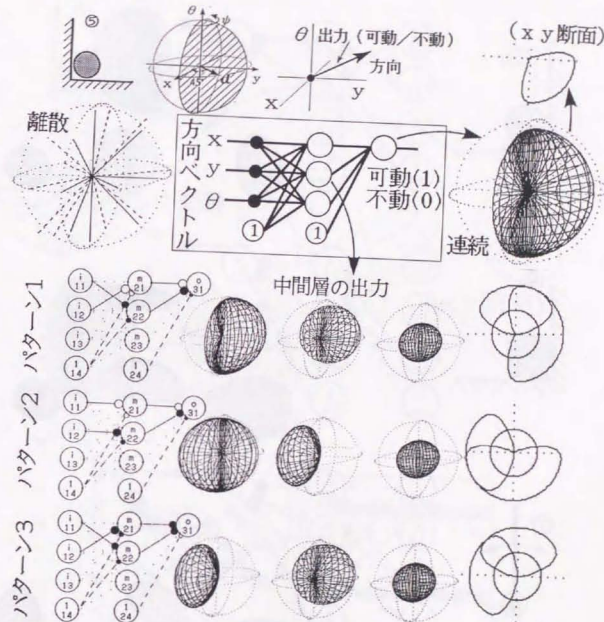
本研究では、今後、中間層ユニットの活性度としてこの表現を用いる。

7.5.6 中間層ユニットの解釈

角に円柱があるような拘束 (5) を学習した時の前段ニューラルネットワークのリンクの重みの分布のパターンとそのときの中間層や出力層のユニットの活性度（出力）を図 7.17 に示す。次のようなことが分かる。

- 同じ可動空間（拘束）に対して、主に 3 つのパターンのリンクの重みが生じた。リンクの重みの分布が異なれば、ユニットの活性度も各々となる。

- 明らかにことであるが、出力層のユニットの活性度は学習させた可動空間の形とほぼ等しい。
- 冗長な中間層ユニットは、それにつながるリンクの重みが 0 に近くっており、出力もどの空間の方向に対しても同じ大きさである。
- 中間層のユニットは、ある空間のまとまった 1 部分に反応しており、それらの形を大まかにとらえれば、それらの和集合・差集合・補集合・交わり等から、出力層のユニットの活性度（可動空間）が構成されている。



角に円柱があるような拘束 (5) を学習した時の前段ニューラルネットワークのリンクの重みの分布のパターンとそのときの中間層や出力層のユニットの活性度（出力）を示す。

図 7.17: 中間層、出力層ユニットの活性度

7.5.7 拘束の種類とリンクの重みおよびユニットの活性度

さらに、拘束の種類を変えて中間層ユニットの活性度の様子を調べた。それを図 7.18 に示す。発生した学習後の前段ニューラルネットワークのリンクの重みも示してある。それぞれ異なったリンクの重みの分布を発生している。可動空間が出力層ユニットの活性度として学習され、中間層ユニットの活性度がそれを構成しているようが見られる。中間層ユニットの活性度の形はまとまっているが、大きさはそれぞれ微妙に異なる。

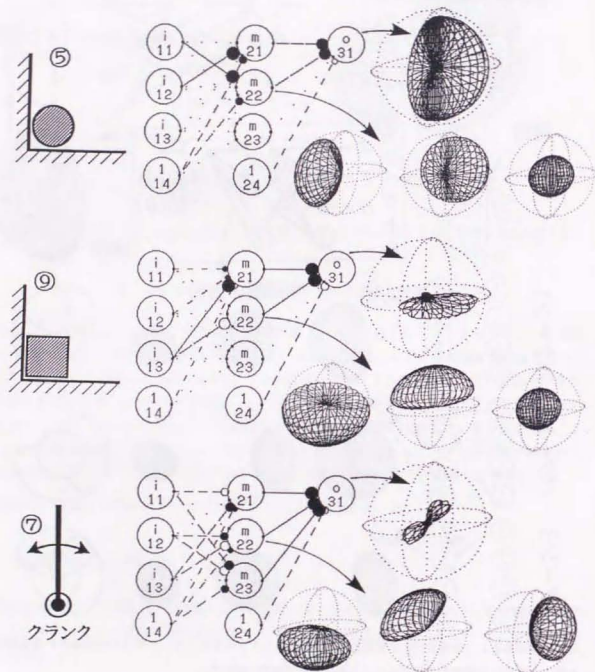


図 7.18: 拘束の種類とそれぞれの中間層、出力層ユニットの活性度

7.5.8 拘束の姿勢と中間層ユニットの活性度

拘束（可動空間）の姿勢を変化させた時に、中間層ユニットの出力がどのように変化するかを調べた。角に円柱があるような拘束 (5) を用いて、姿勢 ψ を変えて前段ニューラルネットワークの学習を行った。その結果を図 7.19 に示す。中間層と出力層のユニットの活性度の $x-y$ 断面を示してある。さらにそのときのリンクの重みの変化も付け加えて表示してある。可動空間の姿勢が変わっても、中間層の反応している空間（活性度）の形は変わらないが、中間層が反応している空間の方向は姿勢に伴って変化する（回転する）ことが分かる。

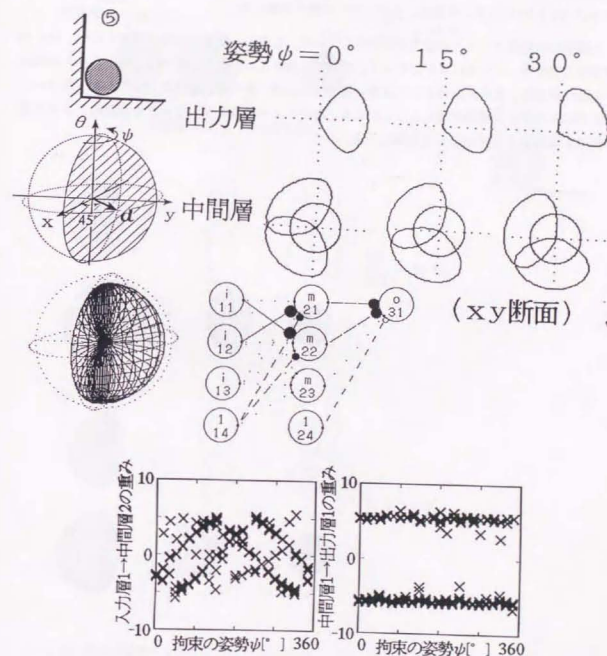


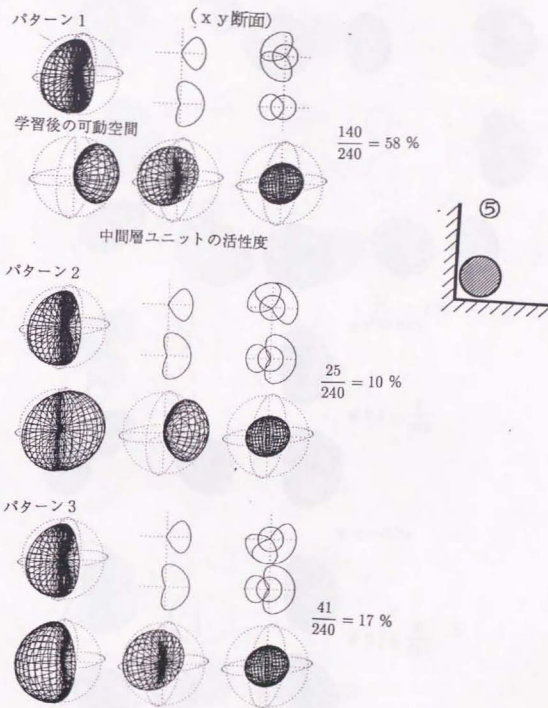
図 7.19: 拘束の姿勢と中間層、出力層ユニットの活性度

7.5.9 リンクの重みや中間層ユニットのパターンの発生確率

同じある可動空間に対して、発生する中間層ユニットの活性度（出力）のパターンを調べた。ユニットの活性度の種類はリンクの重みの分布の種類に対応している、これはリンクの重みの分布パターンを調べていることと同じである。

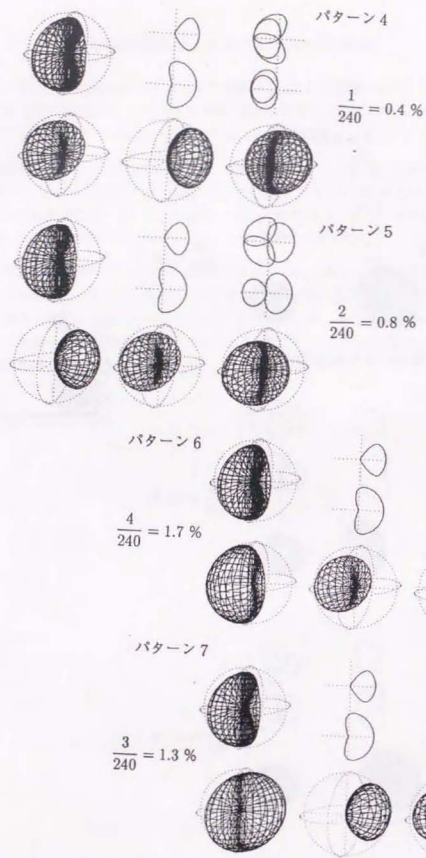
角に円柱があるような拘束 (5) について、姿勢を $0 \sim 360^\circ$ 、 15° おきに变化させて、各姿勢についてそれぞれ 10 回ずつ、合計 240 回前段ニューラルネットワークの学習を行った。学習前のリンクの重みの初期値は毎回異なる。学習後の前段ニューラルネットワークの活性度の種類と発生回数を図 7.20、7.21、7.22 に示す。分類は、人が一つ一つ見て判断した。

主に 3 種類の中間層ユニットの活性度が発生している。しかし、忘却付き構造学習を行い、いくつかの学習の工夫を行っているにもかかわらず、忘却不十分なユニットや学習が偏ったユニットが発生している場合がある。それは全体としては多い数ではないが、その分、前段ニューラルネットワークのリンクの重みの分布の種類が増えることになり、後段ニューラルネットワークで前段のリンクの重みから拘束を識別する学習にとって負担になる。



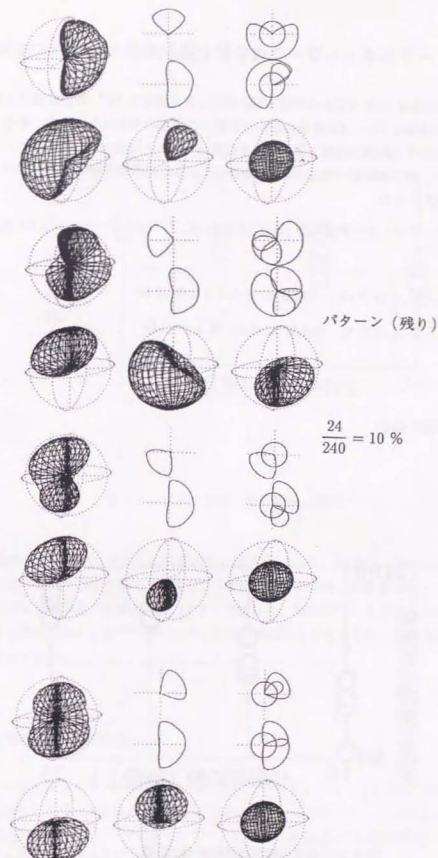
主にこの 3 種類が発生する

図 7.20: 中間層ユニットの活性度の種類と発生確率の例



対称性がよく稀に発生するものや忘却が不十分で収束していないものがある。

図 7.21: 中間層ユニットの活性度の種類と発生確率の例 (つづき)



学習した可動空間の形がいびつである。収束が不十分であったり、学習が偏っている (離散的な学習データのため) もがある。

図 7.22: 中間層ユニットの活性度の種類と発生確率の例 (つづき)

7.6 後段ニューラルネットワークの学習と識別結果

図 7.1 の 16 種類の拘束 (1/0 で表した可動空間) に対し各々姿勢を 15° おきに変えて前段ニューラルネットワークの学習を行い、4960 個のリンクの重みの分布のパターンを得た。そのリンクの重みの分布とそれに対応する拘束の種類と姿勢を表す代表ベクトルを後段ニューラルネットワークの学習データとして用い、誤差逆伝播学習を 30 万回を行った。誤差逆伝播学習の係数 $c = 0.5$ 、重みの修正速度の係数 $\alpha = 0.9$ とした。

後段ニューラルネットワークの計算時間 (学習 30 万回) は、ワークステーション SUN SparcStation 20 を使用して

- 拘束の種類 (入力 16 出力 16)、中間層 20 個のとき 4 分 50 秒
- 拘束の姿勢 (入力 32 出力 3)、中間層 30 個のとき 8 分 10 秒

であった。

7.6.1 拘束の種類の識別

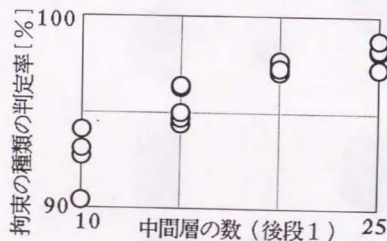


図 7.23: 拘束の種類の識別結果 (学習データ)

拘束の種類の学習結果を図 7.23 に示す。異なる中間層の個数で行ってある。学習に用いた前段ニューラルネットワークのリンクの重みのデータを入力した時の拘束の種類の判別は、中間層が 20 個の時 97.7% 正しかった。前段のリンクの重みの分布パターンを利用して拘束の種類が判別できることが示された。

7.6.2 拘束の姿勢の検出



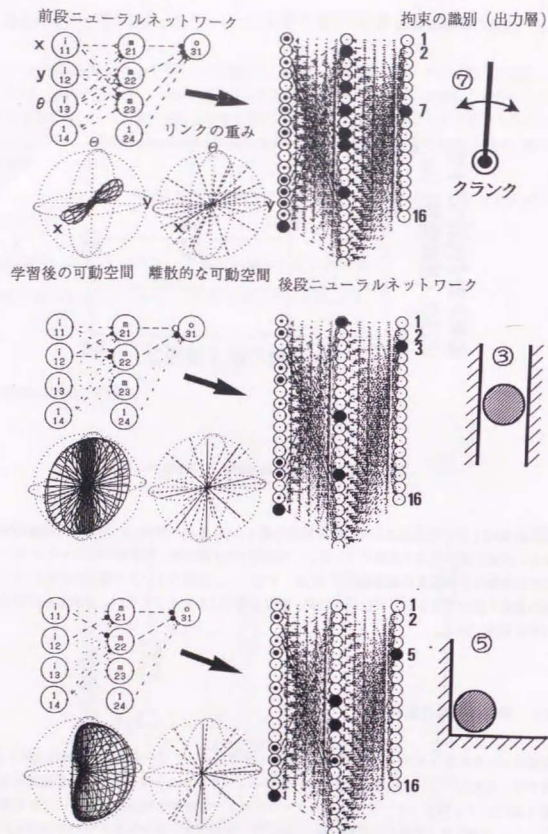
図 7.24: 拘束の姿勢の検出結果 (学習データ)

拘束の姿勢を表す代表ベクトルの学習結果を図 7.24 に示す。同様に、異なる中間層の個数で行ってある。拘束の姿勢を表す代表ベクトルは、中間層が 30 個の時、学習後のニューラルネットワークの出力と教師信号の誤差の標準偏差が 17.3° であった。前段のリンクの重みの分布のパターンから拘束の姿勢を検出できることがわかったが、検出の精度はあまりよくない。おおまかな拘束の方向を検出する程度である。

7.6.3 拘束の種類の識別例

後段ニューラルネットワークの学習後、前段ニューラルネットワークに拘束 (可動空間) を示して、学習させ、発生したリンクの重みの分布から後段ニューラルネットワークが拘束の種類を識別した例を図 7.25 に示す。前段ニューラルネットワークに与えた離散的な可動空間のデータと学習後の出力層ユニットの活性度 (学習した可動空間)、および、後段ニューラルネットワークの拘束の種類の識別の様子を示してある。後段ニューラルネットワークのユニットの出力は、ユニットの●の大きさで示した。

拘束の種類に対応する後段ニューラルネットワークの出力層のユニットが反応していることが分かる。前段ニューラルネットワークのリンクの重みの分布から、拘束の種類が識別できている。



後段ニューラルネットワークのユニットの出力は、ユニットの●の大きさで示した。拘束の種類に対応する後段ニューラルネットワークの出力層のユニットが反応していることが分かる。

図 7.25: 前段ニューラルネットワークのリンクの重みの分布から拘束の種類を識別した例

7.7 まとめ

本研究で取り扱う平面3自由度の拘束を許容される微小な変位に基づいて16種類に分類した。片側の拘束と両側の拘束がある。それに対応する可動空間を示した。そのとき、対象物(拘束)の姿勢によって、可動空間の姿勢が変化することを述べ、ニューラルネットワークが扱いやすいように拘束の姿勢を表す代表ベクトルを定めた。

次に、平面3自由度の拘束(可動空間)を識別するための前段と後段のネットワークからなる2段の階層的ニューラルネットワークを提案した。前段ニューラルネットワークは与えられる可動空間を学習し、発生したリンクの重みの分布を後段ニューラルネットワークに送る。後段ニューラルネットワークは2つのネットワークからなり、1つは拘束の種類を識別し、もう1つは拘束の姿勢を出力する。

分類された16種類の拘束に対して、その可動空間を前段ニューラルネットワークに学習させて、学習後発生するリンクの重みの分布の関係を挙動を調べた。忘却付き学習により発生するリンクの重みは数種類に抑えられている。可動空間の範囲や姿勢が変わるとリンクの重みも連続的に変化することが分かった。これはリンクの重みの分布から拘束の種類や姿勢を識別するのに良い性質である。

また、反応している空間を示すユニットの活性度から、中間層の解釈を行った。中間層ユニットはある空間のまとまった1部分に反応しており、それらの和・差・補・交わり集合等から出力層のユニットの活性度(可動空間)が構成されていることが分かった。また、それらは可動空間の姿勢が変わるとそれに伴って回転していくことが分かった。

さらに、可動空間を学習した前段のニューラルネットワークのリンクの重みをもちいて、後段のニューラルネットワークの学習を行った。その結果、提案する2段の階層的ニューラルネットワークを用いて、リンクの重みから拘束の種類や姿勢が認識できることが示された。

第8章 探り動作のデータを用いたニューラルネットワークによる拘束の認識

8.1 はじめに

前章まででは、拘束に対する可動空間は理想的に1/0で表現されていたが、ここでは、探り方向の拘束の剛性で表現して、それを2段の階層的ニューラルネットワークで認識することを考える。

実際の可動空間を得るためには、まず、探り動作を行ってロボットの手先の変位と力のパターンを得て、それを処理し、探り方向の拘束の剛性を求める必要がある。変位と力のパターンから探り方向の拘束の剛性を求めるアルゴリズムを示す。そして、実際の探り動作のデータを用いたその結果と得られた離散的な可動空間を示す。

また、実測値に適するように、2段の階層的ニューラルネットワークを改良する。前段ニューラルネットワークは、大きさやオフセットの異なる可動空間にも対応できるように出力層が線形ユニットである4層構造にし、学習方法を検討する。後段ニューラルネットワークについては、対象物の大きさを考慮した可動空間のデータを加える。さらに、グループ学習による学習誤差の抑制を図る。そして、実際の探り動作で得られた離散的な可動空間の認識を行う。

8.2 探り動作による平面3自由度の可動空間の取得

8.2.1 探り動作による拘束状態の検出の方針

ロボットの探り動作により、把握している対象物の拘束の状態（可動空間）を検出する方針は次のようになる。

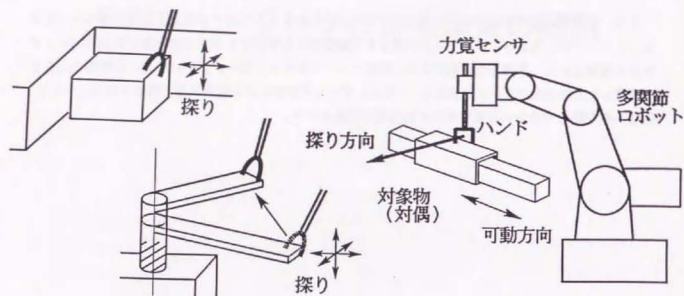


図 8.1: 探り動作と拘束状態

- (1) ロボットが対象物をしっかりと把握した状態で、ある方向に比較的小きな力やモーメントを加えてその時のロボットの手先の変位と力の情報を取得する。平面3自由度の場合は、 (x, y, θ) 空間のある方向ベクトルを探り方向として選ぶことになる。
- (2) その力と変位のパターンからその方向が動く方向（可動）か、動かない方向（不動）か、判断する。あるいは、その探り方向の拘束の柔らかさ（変位と力から求めたバネ定数）を計算しても良い。このとき、ガタや摩擦などの影響を考えた処理をする。
- (3) このような探り動作を、様々な方向に対して行っていく。そうすると、探り動作を行った方向ベクトルとその方向が動くかどうか（あるいはバネ定数）の情報の組が複数個得られる。つまり、離散的な可動空間が得られる。
- (4) 離散的な可動空間を2段の階層的ニューラルネットワークを用いて、それに対応する幾何学的拘束の種類と姿勢を認識させる。

重要な点は、探り動作は可動空間をサンプリングしているという点であり、直接的には、連続的な可動空間ではなく、離散的な可動空間が得られるということである。

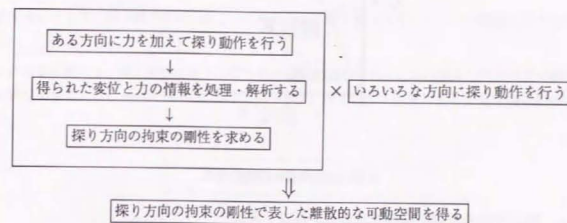


図 8.2: 探り動作の手順

8.2.2 探り方向の拘束の剛性で表した可動空間

平面3自由度 (x, y, θ) の探り動作の場合には、力とモーメントの成分 (F_x, F_y, M_z) を加える。

$$F_d = -[F_x, F_y, 0, 0, 0, M_z]^T \quad (8.1)$$

として、準静的に力の目標値を大きくしていくことになる。

その探り動作を行った時の把握した対象物の動き（拘束状態）は、前章でも述べたように、基本的には

- ・自由に動く場合
- ・拘束されて動かない場合
- ・可動方向に沿って探り方向とずれて動く場合

の3つである。対象物や環境に柔らかさがあると、これらの3つの中間のような状態が存在する。そこで、柔らかさを含めて扱うには、探り方向の拘束のバネ定数を考えればよい。シミュレーションでは、可動空間は $1/0$ で表現していたが、今後は、探り方向の拘束の剛性で表現する。

探り方向に対して、拘束の剛性をそのまま表示すると、剛性は $0 \sim \infty$ まで変化し、表現やその後の処理が困難である。そこで、各方向の剛性 K に対し、

$$K_\beta = 1 - \frac{2 \tan^{-1} K}{\pi} \quad (0 \geq K_\beta \geq 1) \quad (8.2)$$

をベクトルの大きさとして可動空間を表示する（図 8.3）。



図 8.3: 拘束の剛性の変換

8.2.3 探り方向の拘束の剛性を求めるアルゴリズム

拘束の剛性で表した可動空間を求めるためには、まず、さまざまな方向に探り動作を行ってそのときの探り方向の拘束の剛性を収集しなければならない。

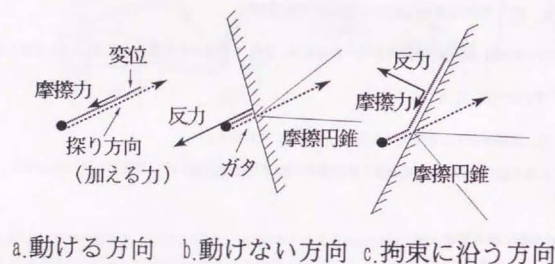


図 8.4: 探り方向の拘束の状態による変位と反力の関係

ある方向の探り動作に対する拘束の状態は、基本的には、自由に動く場合、拘束されて動かない場合、可動方向に沿ってずれて動く場合、の3つである。実際に得られた変位と力のパターンから、この3つの状態を(図8.4)検出し、それぞれガタの部分を除き、探り方向の拘束の剛性を求めることになる。拘束に沿って動いた場合には、1つの探り方向の情報のみならず、拘束に沿う方向と反力の方向の情報も含まれており、各方向の拘束の剛性が計算できると考えられる。すなわち、

- 自由に動く場合 → 変位の方向の剛性
- 拘束されて動かない場合 → 反力の方向の剛性

- 可動方向に沿って探り方向とずれて動く場合 → 沿って動いた変位の方向の剛性、反力の方向の剛性

が求まる。そして、探り動作を繰り返し、このような処理を繰り返すことにより離散的な可動空間を得る。

これらを考慮して、探り動作を行って得られた変位と力のパターンから探り方向の拘束の剛性を求める処理を構築した。それを図8.5に示す。図8.5は、例として、拘束に沿って動く場合の処理を示した。

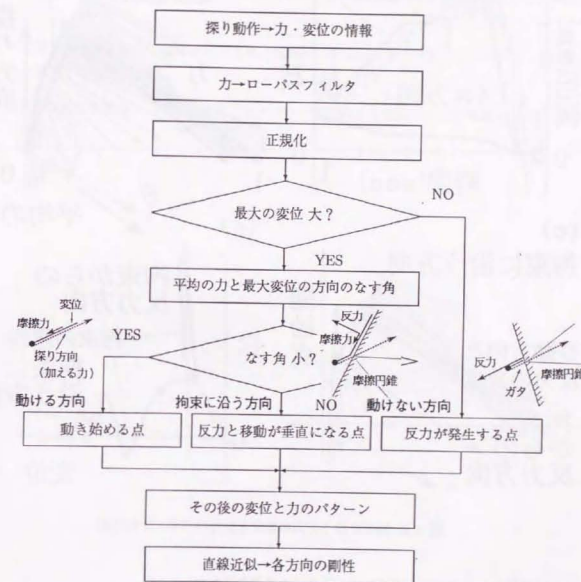


図 8.5: 変位と力のパターンから探り方向の拘束の剛性を求める処理

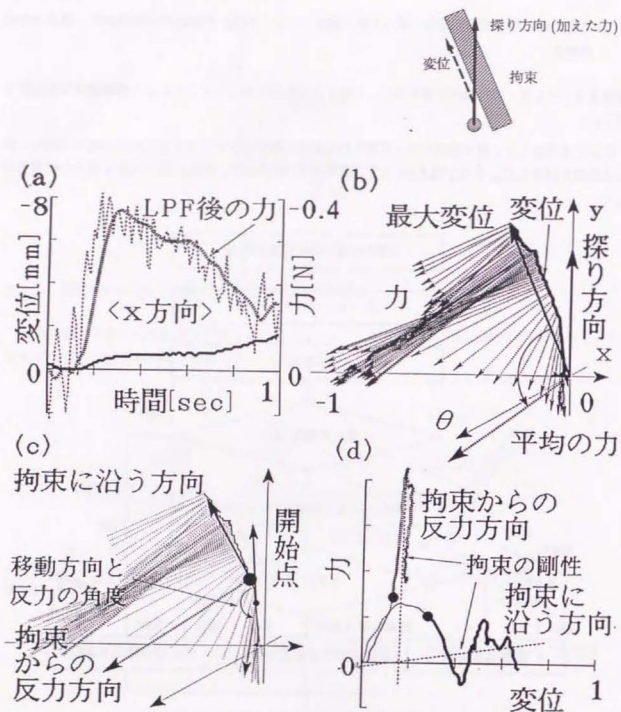


図 8.6: 拘束に沿う方向の変位と反力のパターンの処理

- (1) 前処理として力のデータにローパスフィルタをかける (a)。
- (2) 予め記録しておいた全く自由な時の変位と拘束した時の力の最大値で正規化する。
- (3) 変位の閾値で動けない場合を区別する。

- (4) 全体に渡って平均した力の方向と最大変位の方向のなす角の閾値で、動ける場合と拘束に沿って動く場合を区別する (b)。
- (5) 動ける場合は変位から動き始める点を、動けない場合は力の大きさを反力が発生する点を各々検出し、その後の変位と力から剛性を計算する。これはガタに相当する部分を除去している。
- (6) 拘束に沿って動く場合は、動いている部分で、移動方向と反力の角度が垂直に近くなる点を検出する (c)。これは、ガタの部分を動いた後、拘束に沿って動き始める点を探していることになる。
- (7) その後の変位と力から、拘束に沿う方向と拘束から反力を受ける方向 (c) について、拘束の剛性を計算する (d)。
- (8) 拘束の剛性の計算は、相当する変位と力のパターンを最小2乗法により直線近似してその傾きを求める。

8.3 実際の探り動作の実験で得た平面3自由度の可動空間

インピーダンス制御されたロボットを用いて探り動作を行って可動空間を得る実験を行った。その結果について説明する。

8.3.1 探り動作時のインピーダンスの設定

本研究では、日常の人間の比較的小さな対象に対する探り動作を目安とし、その程度の力と変位が実現できるような探り動作のパラメータを設定した。また、1回の探り動作（ある力を加える動作）は、2～3秒程度で準静的に行った。例えば、18方向に探り動作を行うと、探り方向の剛性を求める処理を含めて、40～60秒程度の時間を要する。

探り動作時に実際に設定した加える力の大きさやインピーダンスを表8.1, 8.2に示す。ここでは、平面3自由度 (F_x, F_y, M_z) の探り動作を対象としている。

表 8.1: 探り動作の設定

	並進	回転
探りの最大力 $ F_d $	0.5N	0.03Nm
探りの最大変位 $ x_e $	10mm	15deg

表 8.2: ロボットアームとハンドの仮想インピーダンスの設定

方向	M_d	D_d	K_d	T_i
アーム x 並進※	0.52 kg	15 Ns/m	50 N/m	0.018 sec
アーム y 並進※	0.58 kg	15 Ns/m	50 N/m	0.022 sec
アーム z 並進	0.55 kg	30 Ns/m	1000 N/m	0.020 sec
アーム x 回転	0.25 kgm^2	0.5 Nms/rad	0.5 Nm/rad	0.02 sec
アーム y 回転	0.25 kgm^2	0.5 Nms/rad	0.5 Nm/rad	0.02 sec
アーム z 回転※	0.25 kgm^2	0.05 Nms/rad	0.03 Nm/rad	0.02 sec
指根本関節	0.6 kg	0.6 Ns/m	40 N/m	0.2 sec
指上下方向	0.8 kgm^2	0.03 Nms/rad	1.0 Nm/rad	0.5 sec
指左右方向	0.8 kgm^2	0.03 Nms/rad	1.0 Nm/rad	0.5 sec

このようなパラメータを設定して、ロボットとハンドにインピーダンス制御をかけて、対象物を把握した状態で、探り動作を行った。実際の探り動作の様子を図8.7に示す。

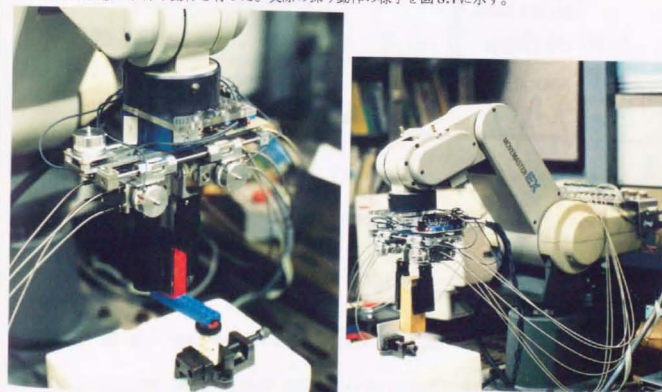


図 8.7: 実際の対象物を把握した探り動作の様子

8.3.2 探り動作で得られた変位と力のパターンの処理結果

インピーダンス制御されたロボットを用いて拘束に沿う方向 (図 8.4) に探り動作を行って得られた変位と力の関係とその処理を図 8.8 に示す。閾値などの処理の各設定値は 8.3 に示した。拘束に沿って動くまでの、最初のカタの部分の動きを検出しており、カタの影響を除去して拘束の剛性を計算することができた。

表 8.3: 変位と力のパターンの処理の各設定値

パラメータ		設定値
データ	数	100
	サンプリング	30msec
ローパスフィルタ	カットオフ周波数	5Hz
正規化	力	(0.5N, 0.5N, 0.03Nm)
	変位	(9.3mm, 10.5mm, 0.223rad)
変位の閾値		0.3
なす角の閾値		15°
動き始めの閾値		222/sec
反力の閾値		0.25
拘束に沿う場合	動きの閾値	200/sec
	垂直な反力の閾値	0.19

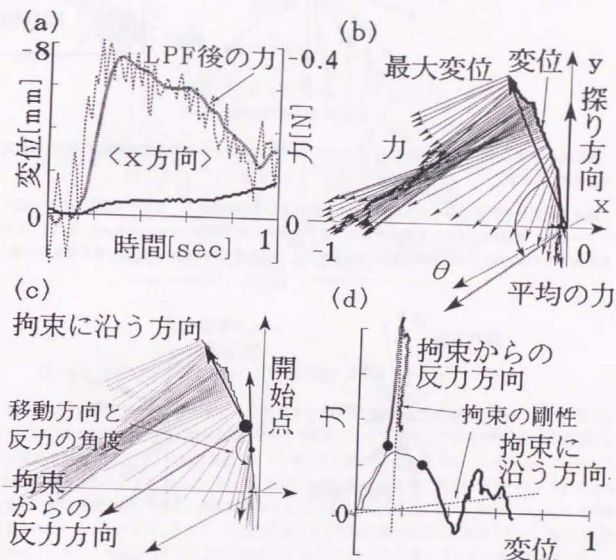


図 8.8: 実際に得られた変位と反力の情報の処理

8.3.3 探り動作で得た拘束の剛性で表した離散的な可動空間

いろいろな方向に探り動作を行ってそれぞれの方向の拘束の剛性を得る実験を行った。探り方向の数を一定にする必要はないが、図 8.9 のように、均一に 18 方位とした。

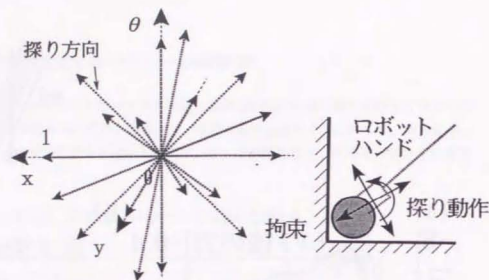
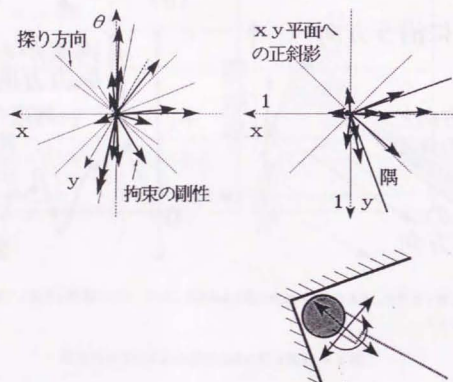


図 8.9: 18 の探り方向

隅に円柱が接している拘束 (図 7.1) について 18 の探り方向から得た拘束の剛性の分布 (離散的な可動空間) を示す。

探り方向以外の拘束に沿った方向の情報も得られているのが分かる。全体の可動空間の形は捉えられているが、部分的にはかなり誤差やばらつきが大きい。探り動作の効率化のためには、拘束に沿った方向やその反力の方向の情報が得られた場合は、その近辺の探り動作を止める事も考えられる。



探り方向以外の拘束に沿った方向の情報も得られているのが分かる。全体の可動空間の形は捉えられているが、部分的にはかなり誤差やばらつきが大きい。

図 8.10: 実際に得た拘束の剛性による可動空間の例

8.4 2 段の階層的ニューラルネットワークによる平面 3 自由度の拘束の認識

8.4.1 探り方向の拘束の剛性による拘束状態の認識

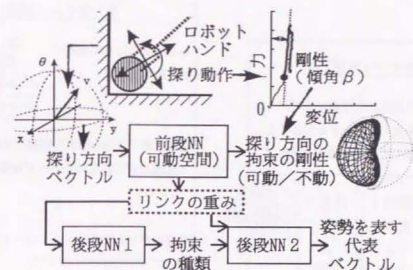


図 8.11: 探り方向の拘束の剛性により拘束状態の識別を行うニューラルネットワーク

前章では、拘束に対する可動空間を 1/0 ではなく、探り方向の拘束の剛性によって表現することを考えた。それによって、拘束状態の識別を行う 2 段のニューラルネットワークの入出力も変更される。それを図 8.11 に示す。可動空間をリンクの重みに反映させる前段ニューラルネットワークの入力は、探り方向のベクトル (x, y, θ) であり、出力は、探り動作時の変位と力のパターンから計算された探り方向の拘束の剛性となる。離散的な探り方向の情報 (可動空間) から、前段ネットワークの学習を行うと、ニューラルネットワークの補間機能により可動空間が連続的に表現される。後段は、前段のニューラルネットワークのリンクの重みの分布パターンから拘束の種類と姿勢を識別する。

8.4.2 実際の探り動作で拘束を識別する手順

探り動作による拘束を識別する手順は、まとめると図 8.12 のようになる。学習時には、拘束の種

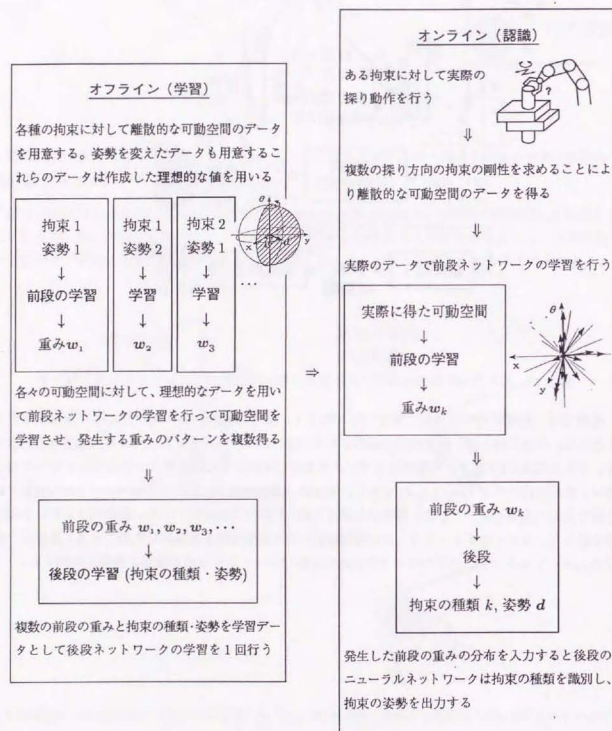


図 8.12: 探り動作と拘束の識別の手順

類や姿勢の数の分、理想的な可動空間のデータをコンピュータ上に用意し、それぞれに対して前段のネットワークの学習を行って後段のネットワークの学習のための重みのパターンを用意する必要がある。そのため、かなり時間を要するが、予めオフラインで行っておける。オンラインの認識時は、実際の探り動作で得られた可動空間のデータを用いて前段ネットワークの学習を 1 回行うだけである。

8.5 実際の可動空間に適した2段の階層的ニューラルネットワークの改良

8.5.1 改良方針

実際の探り動作で得られる値を扱うことができるように、2段の階層的ニューラルネットワークの構造と学習法の改良を行った。改良点とその方針を以下に示す。

- 実際に得られる可動空間は大きさやオフセットにばらつきがあるが、それによらず、前段のニューラルネットワークには一定の重みを発生したい。
→
線形ユニットを出力層に持つ4層構造にする。
1つのユニットにつながるリンクの重みの正規化したものとその大きさを後段ニューラルネットワークへ送る。
- 角に角柱が当たっているような拘束の場合、実際には回転の自由度が生じるが、そのような場合も認識したい。
→
回転の自由度の微小変位を考慮した学習データを作成する。
- 拘束の姿勢の認識精度を向上させたい。
→
拘束の姿勢を表す代表ベクトルを可動空間の部分の中心付近に統一的に設定することによって、後段のネットワークの学習の負担を減らす。
- 拘束の種類や姿勢の認識率の種類によるばらつきをなくしたい。
→
後段ニューラルネットワークでグループ学習を行う。

8.5.2 4層構造の前段ニューラルネットワーク

線形ユニットを出力層に持つ4層構造

可動空間を学習した前段ニューラルネットワークの重みの分布から拘束状態を検出するには、ある可動空間に対してネットワークの初期値や学習順序によらず一定の重みの分布を発生させる必要がある。基本的には忘却付き構造学習を行い冗長な重みを消去する。ところが、実現域から得られる可動空間の場合には、探り方向の拘束の剛性が連続値であるために、可動空間の大きさやオフセットが変化し、重みの分布に影響を与えたと考えられる。例えば柔らかい拘束では、拘束方向にも動くことができ、剛性は小さくなる。

この影響を防ぐため、図8.13のように前段ニューラルネットワークは出力層が線形出力ユニットである4層構造とした。これは、出力層のユニットに学習させる可動空間の大きさやオフセットが変

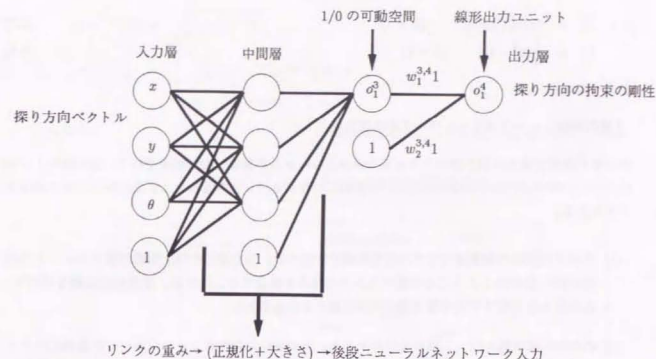


図 8.13: 実測値を扱う4層の前段ニューラルネットワーク

化しても、その1つ前のユニットには、一定の1/0の可動空間を出力させようとするものである。そうすると、3層目までのリンクの重みの分布は、4層の出力の変化の影響を受けないことが期待される。

出力層の線形出力ユニットの出力関数 f は、他の層のシグモイド関数の傾きを考慮して、次のようにした。

$$f(u) = \frac{u}{4}, \quad f'(u) = \frac{1}{4} \quad (8.3)$$

1つ前の層の出力を $o_1^3, 1$ とし、リンクの重みをそれぞれ $w_{11}^{3,4}, w_{21}^{3,4}$ とすると、出力層の線形出力ユニットの出力は、

$$i_1^4 = w_{11}^{3,4} o_1^3 + w_{21}^{3,4} 1 \quad (8.4)$$

$$o_1^4 = f(i_1^4) = \frac{1}{4} (w_{11}^{3,4} o_1^3 + w_{21}^{3,4}) \quad (8.5)$$

となる。ここで、

$$w_{11}^{3,4} o_1^3 = 4, \quad w_{21}^{3,4} = 0 \quad (8.6)$$

のとき、出力層とその1つ前のユニットの出力は同じになる ($o_1^4 = o_1^3$)。

つまり、出力層の1つ前のユニットの出力 o_1^3 の範囲が一定でも、出力層のユニットの出力 o_1^4 は、適当なリンクの重みにより、出力の範囲とオフセットを変化させることができる。後は学習により自動

的にリンクの重みを成長させればよい。逆方向の誤差の伝播は、望ましい出力を t_1 とすると

$$\delta_1^3 = (\delta_1^4 w_{11}^{3,4}) f'(t_1^3) \quad (k \neq 3) \quad (8.7)$$

$$\delta_1^4 = \frac{1}{4}(o_1^4 - t_1) \quad (k = 4) \quad (8.8)$$

のようになる。

4層の前段ニューラルネットワークの学習方法

出力層が線形である4層の前段ニューラルネットワークの学習は、次の順序で行う。出力層の1つ前のユニットが一定の出力になるように、出力層につながるリンクの重みをうまく成長させるのがポイントである。

- (1) リンクの重みの初期値として小さな乱数を設定する。出力層とその1つ前の層のユニットの出力が同じ値が出るようにその間のリンクの重みを設定する。これは、誤差の逆伝播を妨げず、入力層から3層までの学習を速く円滑に進めるためである。
- (2) 誤差逆伝播学習を行って重みを成長させる。出力層につながるリンクの重みの学習係数は小さく設定する。つまり、入力層から3層までの学習（リンクの重みの成長）が速く、出力層につながるリンクの重みの成長は遅いことになる。これらは、出力層の出力の範囲やオフセットによらず、1/0の可動空間を出力層の1つ前のユニットに発生させる効果がある。平均2乗出力誤差が小さくなれば次に移る。
- (3) 出力層につながるリンクの重みを減少し、他のリンクの重みに比べて小さな係数で学習を続ける。平均2乗出力誤差が小さくなれば次に移る。
- (4) 中間層は出力が中間値を取らず、1または0に近づくような明確化学習を併せて行う。明確化学習の学習誤差 E_2 が小さくなれば次に移る。
- (5) 忘却付き学習と選択的忘却付き学習を行って冗長な重みをなくす。ただし、出力層につながるリンクの重みの忘却は行わない。また、出力層の1つ前のユニットの明確化学習は引き続き行う。収束の判定は、平均2乗出力誤差と重みの変化により行う。
- (6) 同じ働きや補集合の働きをするリンクの重みがあれば一方の重みを消して学習をやり直す。
- (7) 各学習過程の途中で、ある一定の回数学習しても収束しなかった場合はもう一度最初から学習をやり直す。
- (8) 後段のニューラルネットワークの認識の負担を減らすため、中間層ユニットをユニットが1を出力する空間の割合によって並べ変える。
- (9) さらに、リンクの重みの変動の影響を抑えるために、出力層につながるリンクの重みを除く前段ニューラルネットワークのリンクの重みを直接後段ニューラルネットワークに渡すのではなく、1つのユニット毎にそれにつながるリンクの重みを正規化し、その大きさとともに渡すよ

うにした。ユニット i につながるリンクの重みを $w_{ji}^{k-1,k}$ とすると

$$|w_{ji}^{k-1,k}| = \sqrt{\sum_j (w_{ji}^{k-1,k})^2} \quad (8.9)$$

$$w_{ji}^{k-1,k} = \begin{cases} \frac{w_{ji}^{k-1,k}}{|w_{ji}^{k-1,k}|} & (|w_{ji}^{k-1,k}| \geq \lambda) \\ w_{ji}^{k-1,k} & (|w_{ji}^{k-1,k}| < \lambda) \end{cases} \quad (8.10)$$

となる。

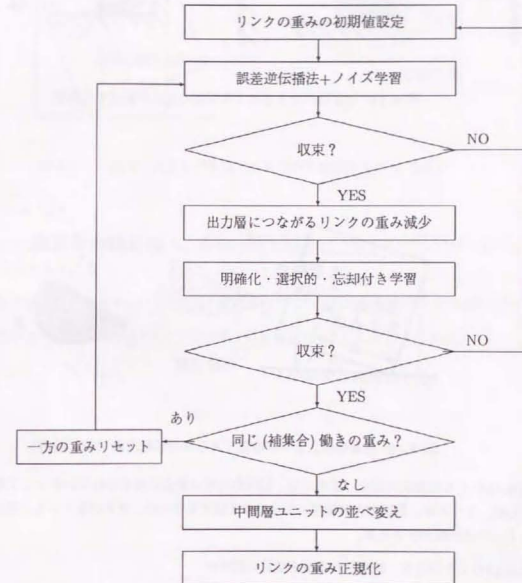


図 8.14: 前段ニューラルネットワークの学習の手順

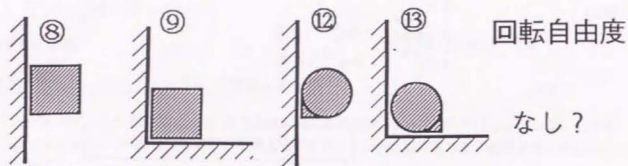


図 8.15: 対象物の大きさにより回転の変位が変化する拘束

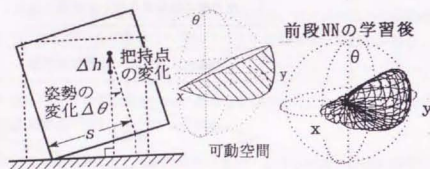


図 8.16: 対象物の大きさが有限である時の可動空間 ((8) の場合)

図 8.15 のような拘束の場合、前章では、回転自由度は完全に拘束されているとして学習データを作成した。ところが、実際の把持物体の大きさは有限であるため、ガタがなくても、把持点や探る方向によっては回転が許される。

図 8.16 のような場合、許容される微小変位の関係は

$$\Delta h \simeq s \Delta \theta \quad (8.11)$$

となる。可動空間は図 8.16 のようになる。これは、姿勢の変化は $\pm \Delta \theta$ の 2 方向あるので、上式の表す 2 つの平面の間の領域である。

図 8.17 のような場合、許容される微小変位の関係は、同様に、

$$\Delta h_1 \simeq s_2 \Delta \theta \quad (8.12)$$

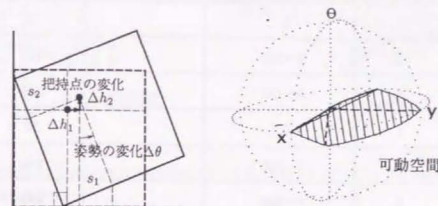


図 8.17: 対象物の大きさが有限である時の可動空間 ((9) の場合)

$$\Delta h_2 \simeq s_1 \Delta \theta \quad (8.13)$$

となる。可動空間は図 8.17 のようになる。姿勢の変化は $\pm \Delta \theta$ の 2 方向あるので、上式の表す 4 つの平面の間の領域となる。

片側の回転の自由度が許されている場合は、その方向については対象物の大きさの影響はない。

以上の考察により、図 8.16, 8.17 のような可動空間も学習データに加えた (表 8.4)。

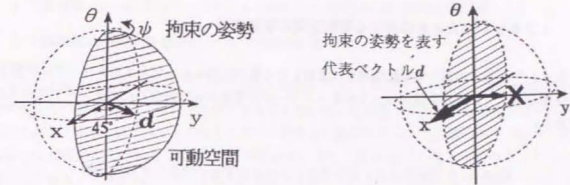
表 8.4: 拘束の種類と学習データ

拘束の種類	代表ベクトルの姿勢 ψ	傾き α	学習データ数
1	なし	-	50
2	$0 \sim 360^\circ$	-	240
3	$0 \sim 180^\circ$	-	120
4	なし	-	50
5	$0 \sim 360^\circ$	-	240
6	$0 \sim 360^\circ$	-	240
7	$0 \sim 180^\circ$	$0, \pm 30, \pm 60, 90^\circ$	$120 \times 5 + 50$
※ 8	$0 \sim 360^\circ$	$(1+2)$	240×3
※ 9	$0 \sim 360^\circ$	$(1+2)$	240×3
10	$0 \sim 360^\circ$	$0, \pm 30, \pm 60, \pm 90^\circ$	$240 \times 5 + 50 \times 2$
11	なし	-	50
※ 12	$0 \sim 360^\circ$	$\pm 45^\circ (1+2)$	$240 \times 2 \times 3$
※ 13	$0 \sim 360^\circ$	$\pm 45^\circ (1+2)$	$240 \times 2 \times 3$
14	$0 \sim 180^\circ$	$\pm 45^\circ$	120×2
15	$0 \sim 360^\circ$	$\pm 45^\circ$	240×2
16	なし	$\pm 90^\circ$	50×2
計			8080

※対象物の大きさが有限の場合にはある関係を持つ微小な回転が許される。それを可動空間の学習データとして加えた。

8.5.4 拘束の姿勢を表す代表ベクトルの設定

拘束の姿勢を表す代表ベクトルは可動空間に対してある一定の方向に定めれば、後段ニューラルネットワークはそれを学習できるが、可動な空間の部分の中心付近に設定するように統一すると後段ニューラルネットワークの負担が軽減される。図 8.18 のような拘束の場合、可動な空間の部分は 2 つの方向に分かれるが、一方の中心に設定すると学習結果が良くなることが期待される。



拘束の姿勢を表す代表ベクトルを可動な空間の部分の中心付近に設定するように統一すると後段ニューラルネットワークの負担が軽減される。

図 8.18: 拘束の姿勢を表す代表ベクトルの設定

8.5.5 後段ニューラルネットワークのグループ学習による学習誤差の抑制

前段ニューラルネットワークの重みのデータを均等に提示して後段ニューラルネットワークの学習を行うと、拘束の種類によって学習結果のばらつきが大きい。また、1つ1つの学習データに対して、学習の状態の悪いものの学習を強化すると、特異な学習データの学習が進んでしまう恐れがある。そこで、教師データの数による調整は面倒であるので、拘束の種類や姿勢を基準に学習データをグループ分けし、そのグループの学習誤差に応じて教師データの提示の割合を変えることにした。一定の学習回数おきに、グループ $i (i = 1, \dots, n)$ の提示率は、グループ毎の学習誤差 E_i 、その平均 \bar{E} と最小値 E_{\min} から

$$P_i = \frac{1}{n} \left(1 + \rho \frac{E_i - \bar{E}}{\bar{E} - E_{\min}} \right) \quad (\bar{E} = \frac{\sum_{i=1}^n E_i}{n}, \sum_{i=1}^n P_i = 1) \quad (8.14)$$

で決定した。係数 $\rho (0 \leq \rho \leq 1)$ により提示率への誤差の寄与を変えられる。このとき、

$$P_i \geq \frac{1-\rho}{n} \quad (E_i \geq E_{\min}) \quad (8.15)$$

である。つまり、基本的な提示率は $\frac{1}{n}$ であり、最低の提示率は $\frac{1-\rho}{n}$ が保証されている。その範囲で、学習誤差 E_i の平均からの差に比例して提示率 P_i を変化させている。学習が進んでいないグループほどそのグループの学習データが多く提示されるわけである。

8.6 2 段の階層的ニューラルネットワークの改良の結果

8.6.1 オフセットや大きさの異なる可動空間の学習結果

オフセットや大きさの異なる可動空間を、改良した4層の前段ニューラルネットワークに学習させた時の結果を示す。4層の前段ニューラルネットワークの学習時に設定したパラメータは表 8.5 の通りである。

表 8.5: 4 層前段ニューラルネットワークの学習パラメータの設定値

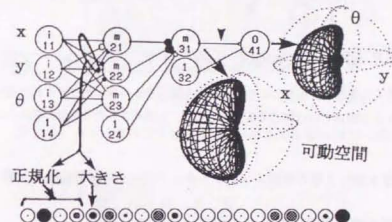
※ () 内は、出力層につながるリンクの重みの学習パラメータ (他と異なる場合)

係数		(1) 誤差逆伝播	(2) 明確化	(3) 忘却付き	(4) 選択的忘却
リンクの重みの初期値	w_{ji}	-0.3 ~ +0.3 乱数 (初 4.0 → 途中で $\frac{3}{10}$ 倍)			
ノイズ学習の乱数	δ	-0.5 ~ +0.5			
重みの修正速度の係数	α	0.9	0.9	0.9	0.9
誤差逆伝播学習の係数	ϵ	0.5 (0.01 → 0.1)	0.5	0.5	0.5
明確化学習の係数	ϵ_c	0	0.05	0	0
忘却の係数	ϵ_f	0	0	1×10^{-4} (0)	1×10^{-4} (0)
選択的忘却の重みの閾値	w_t			100	1.0
収束の判断 (学習回数)	t	< 8000	< 3000	< 19000	< 2000
(平均 2 乗出力誤差)	E	< 0.01		< 0.01	< 0.0002
(明確化学習誤差)	E_2		< 0.3		
(重みの変化)	W_c			< 0.001	< 0.001

また、前段ニューラルネットワークの全体の学習回数とその計算時間は収束の速度により異なるが、再学習を除くと次の通りであった。

- 学習回数 3000 ~ 30000 回
- 計算時間 0.8 ~ 8.1 秒 (パーソナルコンピュータ EPSON PC486RS 使用時)
- 計算時間 0.6 ~ 6.0 秒 (DSP ボード DSP4200 使用時)

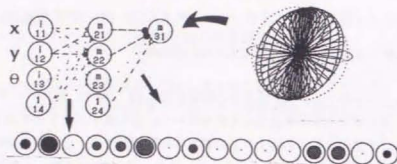
4 層の前段ニューラルネットワークの学習の結果を図 8.19 に示す。学習データの可動空間は、1/0 の可動空間の大きさとオフセットを変えたものを用いた。出力層ユニットとその 1 つ前のユニットの出力を示してある。オフセットのある大きさの異なる可動空間でも、出力層の前のユニットには理想的な 1/0 の可動空間が発生していることが分かる。また、図 8.20 に 3 層と 4 層のネットワークの比較をしてある。4 層のネットワークの入力層から 3 層までのリンクの重みは、3 層のネットワークに理想的な 1/0 の可動空間を学習させた時のリンクの重みとほぼ同じであり、さらに、正規化されたリンクの重みのパターンはほとんど同じである。



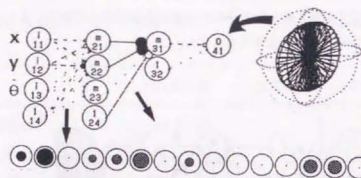
オフセットのある大きさの異なる可動空間でも、出力層の前のユニットには理想的な 1/0 の可動空間が発生していることが分かる。

図 8.19: 4 層の前段ニューラルネットワークの学習結果

1/0の可動空間を3層のニューラルネットワークで学習した時



大きさの異なる可動空間を4層のニューラルネットワークで学習した時



4層のネットワークの入力層から3層までのリンクの重みは、3層のネットワークに理想的な1/0の可動空間を学習させた時のリンクの重みとはほぼ同じであり、さらに、正規化されたリンクの重みのパターンはほとんど同じである。

図 8.20: 4層の前段ニューラルネットワークの学習結果の比較

8.6.2 拘束の姿勢を表す代表ベクトルの設定

拘束の姿勢を表す代表ベクトルを可動空間の部分の中心付近に設定するように統一したときの学習結果を調べた。

図 7.1の16種の拘束に対し、姿勢を 15° おきに変えて作成した前段ニューラルネットワークのリンクの重みのデータ8080個(対象物の大きさを考慮したデータも含む)を後段ニューラルネットワークの学習データとし、30万回学習を行った。その結果、代表ベクトルの誤差の平均は後段ニューラルネットワークの中間層が25個の時 11.8° であった。学習データの個数が4960個と異なるが、姿勢を表す代表ベクトルを可動空間の中心付近に一定に定めていなかった場合の結果は同じ中間層の数の時、 17.5° であったので、それに比べてかなり良くなっている。設定の統一化により、後段ニューラルネットワークの姿勢の学習の負担が軽減されたといえる。

8.6.3 後段ニューラルネットワークのグループ学習の結果

後段ニューラルネットワークにグループ学習を行った時の結果を示す。図 8.21, 8.22に教師データの提示率 ρ を変えた時の学習を示す。提示率の変更を行った間隔は、学習回数10000回おきである。グループ学習を行う($\rho = 0.8$)とばらつきが少なく収束も速く、効果がある。

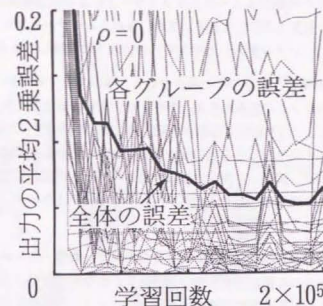
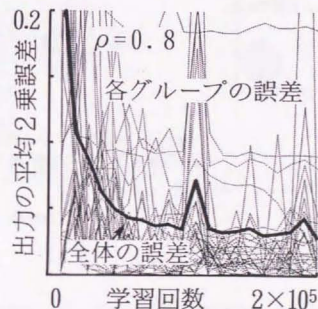


図 8.21: グループ学習を行わない時 $\rho = 0$ の学習の収束の様子



グループ学習を行うと各グループの学習誤差のばらつきが少なく、全体の学習の収束も速くなっている。

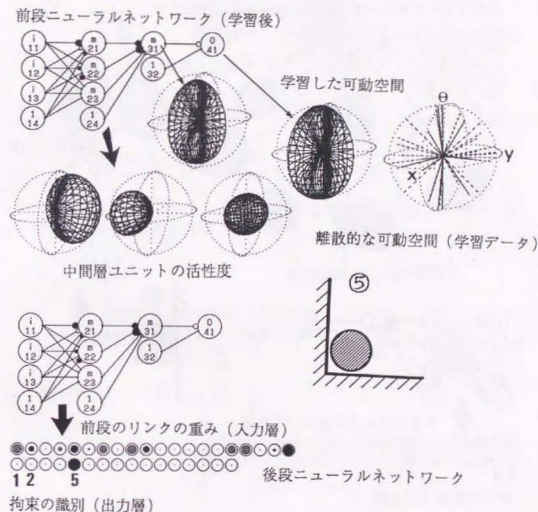
図 8.22: グループ学習を行った時 $\rho = 0.8$ の学習の収束の様子

8.7 探り動作で得られた離散的な可動空間の認識

実際の探り動作で得られた離散的な可動空間に対して、改良した4層の前段ニューラルネットワークと学習済みの後段ニューラルネットワークを用いた拘束の種類識別を試みた。図8.23のような探り動作で得られた可動空間を前段ニューラルネットワークで学習させた後のリンクの重みの分布と中間層ユニットの活性度とその後段ニューラルネットワークによる識別結果を図8.23, 8.24, 8.25に示す。リンクの重みは円の大ききで示してある。また、後段ニューラルネットワークへの入力（前段ニューラルネットワークのリンクの重み）と出力も円の大ききで表現してある。

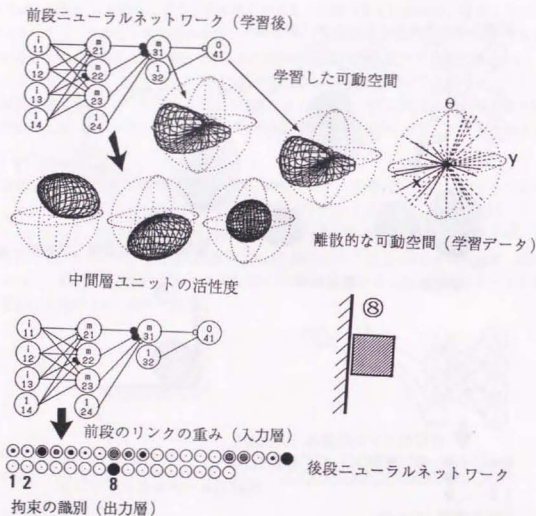
このように、識別できる場合とそうでない場合がある。識別できている時は、発生している前段ニューラルネットワークのリンクの重みの分布パターンや中間層の活性度が典型的なものである。識別できていないものを見ると、探り動作で得られた離散的な可動空間のデータの乱れが大きく、また、前段ニューラルネットワークが学習した可動空間が理想的なものとならずれている。ただし、拘束の種類識別は誤っているが、似ている拘束の種類出力ユニットが反応している。識別を失敗した場合に、類似性の高い拘束の種類出力ユニットが必ず反応するわけではないが、このような場合も多く見られた。

以上、対象とする拘束状態が理想的なものから、かなりガタ・摩擦・柔らかさが含まれているものまであり、それによって得られる離散的な可動空間の品質がかなりばらつく。4層の前段ニューラルネットワークによって、ある程度の可動空間の大きさやオフセットの変動には対応できるが、その性能にも限界がある。また、同じ可動空間に対しても、典型的なリンクの重みが発生せず、識別を誤ることもあった。離散的な可動空間から大きな構造だけを取り出すような、前段ニューラルネットワークの学習の改良がさらに必要である。



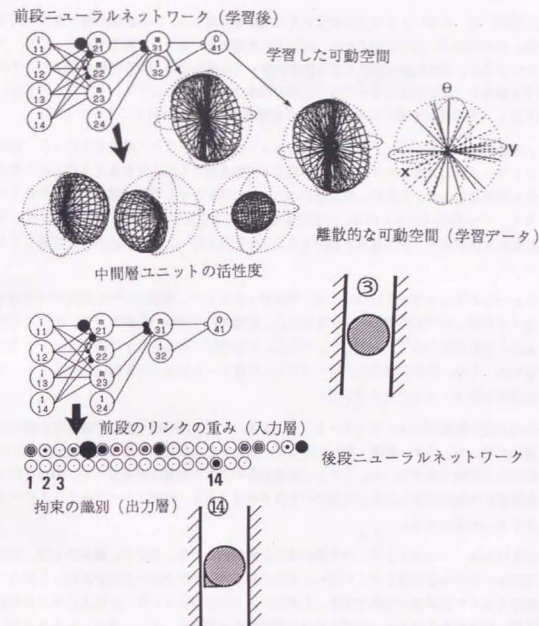
発生している前段ニューラルネットワークのリンクの重みの分布パターンや中間層の活性度が典型的なものであり、拘束の種類が識別されている。

図 8.23: 探り動作で得られた離散的な可動空間の識別結果の例 1



対象物の大きさを考慮しない理想的な可動空間では回転の自由度はないが、実際の対象物の大きさは有限であるため、離散的な可動空間には回転の変位が認められる。識別が成功しているのは、この大きさを考慮した可動空間も学習データに加えているからと考えられる。

図 8.24: 探り動作で得られた離散的な可動空間の識別結果の例 2



識別に失敗している。探り動作で得られた離散的な可動空間のデータの乱れが大きく、また、前段ニューラルネットワークが学習した可動空間が理想的なものとなりすぎている。ただし、拘束の種類は誤っているが、似ている拘束の種類は出力ユニットが反応している。

図 8.25: 探り動作で得られた離散的な可動空間の識別結果の例 3

8.8 まとめと考察

拘束に対する可動空間を理想的に1/0で表現するのではなく、対象物や環境の柔らかさ等を考慮して、探り方向の拘束の剛性で表現することを考えた。

探り動作時には、ロボットの手先の変位と力の情報を収集する。ある方向の探り動作に対する拘束の状態は、基本的には、自由に動く場合、拘束されて動かない場合、可動方向に沿ってずれて動く場合、の3つである。実際に得られた変位と力のパターンから、この3つの状態を検出し、それぞれガタの部分を除き、直線近似によって探り方向の拘束の剛性を求めるアルゴリズムを示した。探り動作を繰り返し、その処理を繰り返すことにより離散的な可動空間を得ることができた。

また、実測値に速するように、2段の階層的ニューラルネットワークの改良を行った。前段ニューラルネットワークは、大きさやオフセットの異なる可動空間にも対応できるよう出力層が線形ユニットである4層構造にした。これは、出力層のユニットに学習させる可動空間の大きさやオフセットが変化しても、1つ前のユニットには、一定の理想的な可動空間を出力させるものである。そのために、出力層につながるリンクの重みを制御するような学習を行い、4層構造が有効であることを確認した。

後段ニューラルネットワークについては、学習データとして、理想的な可動空間だけでなく、対象物の大きさを考慮した可動空間のデータを加えた。理想的には回転の自由度がない拘束でも大きさによって並進と回転が生じることを示した。さらに、可動空間の姿勢を表す代表ベクトルをその中心付近に設定することや、拘束の種類グループごとに学習データの提示率を変化させるグループ学習により学習誤差が小さくなることを示した。

改良した2段の階層的ニューラルネットワークを用いて、実際の探り動作で得られた離散的な可動空間の認識を行った。ガタ・摩擦・柔らかさなどの誤差を含んだ実際の離散的な可動空間に対して拘束の認識がある程度可能であった。しかし、典型的なリンクの重みが発生せず、識別を誤ることもあり、離散的な可動空間から大きな構造だけを取り出すような、前段ニューラルネットワークの学習の改良がさらに必要である。

このようにニューラルネットワークを用いると、解析的な手法と異なり、基本的には、拘束の種類に対して別々にモデルを作成しなくて良い。また、動く方向が動かない方向かといった場合に明示的に閾値を与えなくてよい点が有利である。しかし、ニューラルネットワークに入力する前処理の段階で、例えば、変位や力のパターンの場合分けの閾値等の設定があったり、有限な大きさをもつ拘束の学習データを加える必要があったりする。完全にモデルや閾値がないというわけではなく、ニューラルネットワークが学習できるように、ネットワークの入出力の構造等を人が与えなければならない。このような点や認識がブラックボックスになる点は、ニューラルネットワークを使用しにくくしている。探り動作で得られる誤差を含む力と位置のパターンを直接入力して、拘束の種類を出力するようなニューラルネットワークの構造が望まれるが、そのようなロバストで単純なネットワークの構築は難しい。

第9章 結論と展望

9.1 本論文の結論

本研究では、ロボットの探り動作とその時の位置と力の情報を解析し、16種類に分類した平面3自由度の拘束を2段の階層的ニューラルネットワークに学習させることにより、未知環境でロボットが把握している作業対象物の運動拘束を認識できることを示した。また、3次元1自由度の拘束の解析的な検出方法との比較を行って、モデル化が複雑な誤差を含む下位のセンサ情報の処理とモデルに基づく高次のロボットの動作計画を結び付ける手法として、本研究の手法が有効であることを示した。

本研究で得られた結論を以下に示す。

- (1) 3次元1自由度の拘束の解析的な検出手法を提案し、その有効性を示した。運動拘束を検出するには「探り動作」で得られる手先の変位と力の情報が有効である。ただし、ガタや摩擦等の影響を除去した上で本来の拘束を識別する必要がある。変位と力のパターン解析には、ガタの範囲の動きや摩擦力に対応した特徴点の探索が有効であるが、直線近似誤差の閾値設定が難しい。また、手先の位置姿勢変化や力/モーメントから対偶の種類が判別でき、ねじ対偶のパラメータで表現、推定できる。実際の最小変位では、ねじ対偶の区別は困難であった。解析的な手法は各処理の結果や意味が明確であるが、モデル作成、閾値設定等が複雑であり、多自由度ではさらに複雑化するといえる。
- (2) 平面3自由度の拘束を認識する2段の階層的ニューラルネットワークを提案した。入出力パターン列に対して忘却付き構造学習を行った前段のリンクの重みの分布を後段に入力してその写像の種類を識別するものであり、特に順序のない入出力パターン列に対して有効である。拘束の認識の場合には、可動空間を学習した前段のリンクの重みパターンから2つの後段が拘束の種類と姿勢を検出する構造になる。シミュレーションにより重みの分布や中間層ユニットの挙動を明らかにした。また、平面3自由度の拘束を16種類に分類し、各々姿勢を変化させて学習を行い、リンクの重みパターンから拘束の種類と姿勢の認識ができることを示した。
- (3) ニューラルネットワークを入力するための、探り動作の変位と力のパターンから拘束の剛性で表現される離散的な可動空間を求める手法を提案し、探り動作の実験により離散的な可動空間が得られることを確認した。また、実際の可動空間に適合するニューラルネットワークの改良を行い、その有効性を確認した。前段は、可動空間の大きさやオフセットの変化に対応できるように、線形の出力層を持つ4層にした。後段は、対象物の大きさを考慮した学習データの追加や学習のばらつきを小さくするグループ学習を行った。
- (4) 2段の階層的ニューラルネットワークによる、実際の探り動作で得られた可動空間の認識を行った。ガタ・摩擦・柔らかさなどの誤差のある可動空間の認識ができることが示された。離散的な可動空間から大きな構造だけを取り出すような学習の改良がさらに必要である。このように、ニューラルネットワークを用いると拘束毎に別々にモデルや閾値を明示的に与えなくてよい点が有利であり、モデル化が複雑なセンサ情報の処理と検出された拘束に基づくロボットの高次の動作計画をつなぐ手法として有効である。しかし、前処理の段階で、変位と力のパターンの場合分けの閾値等の設定や有限な大きさをもつ拘束の学習データの必要性など完全にモデルや閾値をなくすことはできなかった。

9.2 今後の展望

今後の展望としては、以下のようなテーマが挙げられる。

● 3次元6自由度の拘束状態の認識

本研究では、平面3自由度の運動拘束の認識を取り扱った。しかし、ロボットが作業する空間は3次元6自由度の空間である。物体の姿勢の表現など、2次元の問題と3次元の問題は本質的に異なることが多い。本研究で提案した2段のニューラルネットワークがそのまま適用できるとはいえない。位置と姿勢に分離して認識を行うとか、別の方法で拘束されている基準の平面を発見してから平面内の拘束の認識をするといった方法が考えられる。

● 実用的な分解作業のためのセンシング戦略

近年、環境やゴミの問題から資源の再利用の重要性が高まっているが、廃棄された家電製品などの工業製品を分解・解体する技術が重要になってきている。多種多様の製品を分解するためには、部品同士の運動拘束を検出する必要がある。一般的な手法を追及するだけでなく、使用するセンサの検討を含めて、分解作業のための実用的なセンシング法が求められる。

● ロボットの探り動作と空間認識

知能ロボットの根本的な問題として、自分の位置を認識したり、対象物体を器用に操ったりするための空間認識は重要である。位置関係や距離を全て数値や数式で記述するような認識ではなく、対象物や環境の大まかな位置や運動の関係を認識できるようになれば、ロボットの知能は向上しないと考える。非常に難しい問題であると思われるが、一般的な探り動作（アクティブセンシング）による検出を含めて、検討が必要である。

謝辞

本研究を進めるに当って、指導教官の佐々木健助教授には、学部4年・修士課程・博士課程・助手の長きにわたり多大な御指導を頂き、心から感謝致します。

高野政晴教授には、別の見地からの研究に対する御批評、御教示を頂き、また、本論文の審査をして頂き、深く感謝致します。

大園成夫教授、新井民夫教授、石川正俊助教授には、本論文の審査をして頂き、有用な御指摘、御助言を頂きました。心から感謝致します。

山川博司技官、井上健司助手（現大阪大学講師）、本郷素行技官（現宇宙科学研究所）には、いろいろと親身な御助言、御激励を頂き、大変お世話になりました。ありがとうございました。

長田利明氏（現三菱電機）、高橋英男氏（現日立）、有竹洋氏（現三菱電機）には、それぞれ卒論や修論の実験装置としてセンサやハンドの開発、ロボットの改良などを共に進めて頂きどうもありがとうございました。

精密機械工学科の諸先生方には、いろいろと御相談に載って頂き、御指導、御激励を頂きました。また、図書室、工作室、事務室の職員の皆様方にも、いろいろと研究生活をサポートして頂きました。心から感謝致します。

最後に、研究室で共に議論研究を進めてきた多くの高野・佐々木研究室の先輩、同輩、後輩の皆様方に深く感謝の意を表します。

1996年2月

関 啓 明

参考文献

- [Albus 75] J.S.Albus: A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller(CMAC): J. of Dynamic Systems, Measurement and Control, Trans. of the ASME, 97, pp.220-227, (1975)
- [Albus 75] J.S.Albus: Data Storage in the Cerebellar Model Articulation Controller(CMAC): J. of Dynamic Systems, Measurement and Control, Trans. of the ASME, 97, pp.228-233, (1975)
- [Asada 89] H.Asada, S.Hirai: Towards a Symbolic-Level Force Feedback: Recognition of Assembly Process States, Prep. 5th Int. Symp. of Robotics Research, (1989)
- [Asada 90] H.Asada: Teaching and Learning of Compliance Using Neural Nets:Representation and Generation of Nonlinear Compliance: Proc. IEEE Int. Conf. on Robotics and Automation, pp.1237-1244, (1990)
- [Asada 91] H.Asada, S.Liu: Transfer of Human Skills to Neural Net Robot Controllers: Proc. IEEE Int. Conf. on Robotics and Automation, pp.2442-2448, (1991)
- [Bajcsy 88] R.Bajcsy: Active Perception: Proc. IEEE, Vol.76, No.8, pp.996-1005, (1988)
- [Berger 91] A.D.Berger, P.K.Khosla: Using Tactile Data for Real-Time Feedback: Int. J. Robotics Research, Vol.10, No.2, pp.88-102, (1991)
- [Bicchi 90] Bicchi: Intrinsic Contact Sensing for Soft Fingers: IEEE Int. Conf. on Robotics and Automation, pp.968-973, (1990)
- [Bicchi 91] A.Bicchi et al.: Contact Sensing from Force Measurements: Int. J. Robotics Research, Vol.12, No.3, pp.249-262, (1991)
- [Borovać 93] B. Borovać, D. Katić: Connectionist Reactive Control for Robotic Assembly Tasks by Soft Sensored Grippers: 24th ISIR, pp.1-pp.8, (1993)
- [Brock 85] D.L.Brock, S.Chui: Environment Perception of an Articulated Robot Hand Using Contact Sensors: ASME Publication PED Vol.15 Robotics and Manufacturing Automation, pp.89-96, (1985)
- [Browse 87] R.A.Browse: Feature-Based Tactile Object Recognition: IEEE. Trans. Vol.PAMI-9-6, pp.779-786, (1987)
- [Campos 91] M.Campos and R.Bajcsy: A Robotic Haptic System Architecture: Proc. of the IEEE Int. Conf. on Robotics and Automation, pp.338, (1991)
- [Cottrell 87] G.W.Cottrell, P.Murano, D.Zipser: Image compression by back propagation: An example of external programing: ICS Report, pp.8702, (1987)

- [Dario 87] P.Dario, G.Buttazzo: An Anthropomorphic Robot Finger for Investigating Artificial Tactile Perception: *Int. J. of Robotics Research*, Vol.6, No.3, pp.25-, (1987)
- [Desai 89] R.S.Desai, R.A.Voltz: Identification and Verification of Termination Conditions in Fine Motion in Presence of Sensor Errors and Geometric Uncertainties: *Proc.IEEE Int. Conf. on Robotics and Automation*, pp.800-807, (1989)
- [Ellis 92] R.E.Ellis: Planning Tactile Recognition Paths in Two and Three Dimensions: *Int. J. Robotics Research*, Vol.11, No.2, pp.87-111, (1992)
- [Elman 90] J.L.Elman: Finding Structure in Time: *Cognitive Science*, Vol.14, pp.179-211, (1990)
- [Fearing 85] R.S.Fearing, J.M.Hollerbach: Basic Solid Mechanics for Tactile Sensing: *Int. J. Robotics Research*, Vol.4, No.3, pp.40-54, (1985)
- [Fearing 87] R.S.Fearing: Some Experiments with Tactile Sensing during Grasping: *Proc. IEEE Int. Conf. on Robotics and Automation*, pp.1500-1505, (1987)
- [Fearing 88] R.S.Fearing, T.O.Binford: Using a Cylindrical Tactile Sensor for Determining Curvature: *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp.765, (1988)
- [Gordon 89] S.Gordon, W.Townsend: Integration of tactile-force and joint-torque information in a Whole-Arm manipulation: *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp.464-, (1989)
- [Hogan 85] N.Hogan: Impedance Control: An Approach to Manipulation: Part I, II, III: *Trans. ASME Journal of Dynamic Systems, Measurement, and Control*, pp.1-24, (1985)
- [Hopfield 85] J.J.Hopfield: Neural Computation of Decisions in Optimization Problem: *Biological Cybernetics*, 52, pp.141-152, (1985)
- [Hou 89] Hou, Lee: Interpretation of Assembly Task Constraints from Position and Force Sensory Data: *SPIE Vol.1193 Intelligent Robots and Computer Vision System and Applications*, (1989)
- [Jordan 86] M.I.Jordan: Attractor Dynamics and Parallelism in a Connectionist Sequential Machine: In *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, pp.531-546, (1986)
- [Lederman 87] S.J.Lederman, R.L.Klatzky: Hand Movements: A Window into Haptic Object Recognition: *Cognitive Psychol.*, Vol.19, pp.342-368, (1987)
- [Lee 92] S.Lee et al.: A New Neural Net Approach to Robot 3D Perception and Visuo-Motor Coordination: *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, NC, 7-10, pp.554-559, (1992)
- [Lozano 84] T.Lozano-Perez et al.: Automatic Synthesis of Fine-Motion Strategies for Robots: *Int. J. Robotics Research*, Vol.3, No.1, pp.3-24, (1984)

- [Luo 86] R.C.Luo, W.H.Tsai: Object Recognition Using Tactile Image Array Sensors: *Proc.IEEE Int. Conf. on Robotics and Automation*, pp.1248-1253, (1986)
- [Kung 90] S.Kung, J.Hwang: Neural Network Architectures for Robotic Applications: *IEEE Trans. on Robotics and Automation*, Vol.5, No.5, pp.641-657, (1990)
- [Mason 81] M.T.Mason: Compliance and Force Control for Computer Controlled Manipulators: *IEEE Trans. on Systems, Man, and Cybernetics*, Vol.SMC-11, No.6, pp.418-432, (1981)
- [Mason 81] M.T.Mason: Compliance and Force Control for Computer Controlled Manipulator: *IEEE Trans. Systems, Man and Cybernetics SMC-11*, 6, pp.418-432, (1981)
- [McCulloch 43] W.S.McCulloch, W.Pitts: A Logical Calculus of the Ideas Immanent in Neural Nervous Activity: *Bulletin of Mathematical Biophysics*, Vol.5, pp.115-133, (1943)
- [Miller 87] W.T.Miller et al.: Application of a General Learning Algorithm to the Control of Robotic Manipulators: *Int. J. Robotics Research*, Vol.6, No.2, pp.84-98, (1987)
- [Minsky 54] M.Minsky: Neural Nets and the Brain Model Problem: Ph.D.Thesis, Princeton Univ, (1954)
- [Narendra 90] K.S.Narendra, K.Parthasarathy: Identification and Control of Dynamical Systems Using Neural Networks: *IEEE Trans. on Neural Networks*, Vol.1, No.1, pp.4-27, (1990)
- [Ohwovoriole 81] Ohwovoriole, Roth: An Extension of Screw Theory: *Trans. ASME Journal of Mechanical Design*, Vol.103, pp.725-735, (1981)
- [Peshkin 90] M.A.Peshkin: Programmed Compliance for Error Corrective Assembly: *IEEE Trans. on Robotics and Automation*, Vol.6, No.4, pp.473-482, (1990)
- [Raibert 81] M.H.Raibert, J.J.Craig: Hybrid Position/Force Control of Manipulators: *Trans. ASME Journal of Dynamic Systems, Measurement, and Control*, 102, pp.126-133, (1981)
- [Roberts 90] K.S.Roberts: Robot Active Touch Exploration: Constraints and Strategies: *IEEE*, pp.980-985, (1990)
- [Rosenblatt 58] F.Rosenblatt: The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain: *Psychological Review*, Vol.65, pp.386-408, (1958)
- [Rossi 91] D.D.Rossi: Artificial tactile sensing and haptic perception: *Meas. Sci. Technol.* 2, pp.1003-1016, (1991)
- [Rumelhart 86] D.E.Rumelhart et al.: Learning Internal Representations by Error Propagation, Parallel Distributed Processing: Explorations in the Microstructures of Cognition: Vol.1, Cambridge, MA, MIT Press, pp.318-362, (1986)

- [Rumelhart 86] D.E.Rumelhart et al.: Learning Representations by Back-propagating Errors: Nature 323, pp.533-536, (1986)
- [Salisbury 80] J.K.Salisbury: Active Stiffness Control of Manipulator in Cartesian Coordinates: Proc. IEEE int. Conf. Decision and Control, (1980)
- [Salisbury 83] J.K.Salisbury: Interpretation of Contact Geometries from Force Measurements: Proc. of the 1st Int. Symp. on Robotics Research, (1983)
- [Salisbury 88] J.K.Salisbury, W.Townsend, B.Eberman, D.Dipietro: Preliminary design of a Whole-Arm Manipulation System(WANS): Proc. of the IEEE Int. Conf. on Robotics and Automation, pp.254-, (1988)
- [Schneider 86] J.L.Schneider: An Objective Tactile Sensing Strategy for Object Recognition and Localization: IEEE Int. Conf. on Robotics and Automation, pp.1262-1267, (1986)
- [Shutter 88] J.D.Schutter, H.V.Brussel: Compliant Robot Motion I.A Formalism for Specifying Compliant Motion Tasks: The International Journal of Robotics Research, Vol.7, No.4, pp.3-17, (1988)
- [Siegel 91] D.M.Siegel: Finding the Pose of an Object in a Hand: Proc. IEEE Int. Conf. on Robotics and Automation, pp.406-411, (1991)
- [Simons 82] J.Simons et. al.: A Self-Learning Automation with Variable Resolution for High Precision Assembly by Industrial Robots: IEEE Trans. on Automatic Control, Vol.AC-27, No.5, pp.1109-1113, (1982)
- [Tsujimura 89] Tsujimura, Yabuta: Object Detection by Tactile Sensing Method Employing Force/Torque Information: IEEE Trans. on Robotics and Automation, pp.440-450, (1989)
- [Vukobratović 93] M. Vukobratović, D. Katić: Connectionist Learning Control Algorithms for Contact Tasks in Industrial Robotics: 24th ISIR, pp.643-650, (1993)
- [Whitney 81] D.E.Whitney: Force Feedback Control of Manipulator Fine Motions: Trans. ASME Journal of Dynamic Systems, Measurement, and Control, pp.91-97, (1981)
- [Whitney 82] D.E.Whitney: Quasi-Static Assembly of Compliantly Supported Rigid Parts: Trans. ASME, J. of Dynamic Systems, Measurement, and Control, Vol.104, pp.65-77, (1982)
- [Whitney 87] D.E.Whitney: Historical Perspective and State of the Art in Robot Force Control: The International Journal of Robotics Research, Vol.6, No.1, pp.3-13, (1987)
- [有竹 93] 有竹: 触覚センサ付き2本指ハンドによるハンドリング: 東京大学大学院工学系研究科精密機械工学専攻 1992 年度修士論文, (1993)
- [石川 88] 石川: 並列処理を用いた能動的センサシステム: 計測自動制御学会論文集, Vol.24, No.8, pp.860-866, (1988)
- [石川 90] 石川: 忘却を用いたコネクショニストモデルの構造学習アルゴリズム: 人工知能学会誌, Vol.5, No.5, pp.595-603, (1990)
- [石川 91] 石川, 木田: ソフトモーションロボットによるコネクタ挿入作業の自動化: 日本ロボット学会誌, Vol.9, No.3, pp.363-368, (1991)
- [石川 92] 石川: モジュール構造ニューラルネットワークの学習: 人工知能学会誌, Vol.7, No.2, pp.145-151, (1992)
- [石川 93] 石川: アクティブセンシングとロボットハンド: 日本ロボット学会誌, Vol.11, No.7, pp.938-942, (1993)
- [池内 93] 池内, 末広: 視覚による組立作業理解のための作業モデルとそれに基づく動作生成: 日本ロボット学会誌, Vol.11, No.2, pp.281-290, (1993)
- [井出 92] 井手他: 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, pp.118-128, (1992)
- [入江 90] 入江, 川人: 多層パーセプトロンによる内部表現の獲得: 電子情報通信学会論文誌, Vol.J73-D-II, No.8, pp.1173-1178, (1990)
- [逢坂 92] 逢坂他: ニューラルネットによるロボットマニピュレータの非線形特性の同定: 日本機械学会論文集 (C 編), Vol.58, No.555, pp.53-57, (1992)
- [小方 92] 小方, 高橋: 組立物体の状態記述に関する研究 (第2報, 並進自由度物体の場合と教示実行システムへの応用): 第2回ロボットシンポジウム, pp.133-138, (1992)
- [小方 93] 小方, 高橋: 幾何情報に基づく作業記述法を用いて異なる作業環境で教示内容を実行するロボットシステム: 日本ロボット学会誌, Vol.11, No.3, pp.444-452, (1993)
- [尾崎 89] 尾崎, 福島: 教師なし学習を用いたネオコグニトリオン型手書き文字認識システム: NC89-48, pp.1-6, (1989)
- [内山 93] 内山: 接触状態のシミュレーション: 日本ロボット学会誌, Vol.11, No.2, pp.201-205, (1993)
- [角所 90] 角所他: 画像の3次元解釈へのニューラルネットワークの適用: 電子情報通信学会論文誌, Vol.J73-D-II, No.8, pp.1297-1302, (1990)
- [加藤 91] 加藤他: 複数の幾何的拘束を受ける物体の可動空間の導出: 第9回日本ロボット学会学術講演会予稿集, pp.231-232, (1991)
- [加藤 92] 加藤他: ニューラルネットワークを用いたステレオ画像の対応探索法: 1992年度精密工学会春季大会学術講演会講演論文集, pp.717-712, (1992)
- [金子 90] 金子, 谷江: 自己姿勢変形性を利用した多指ロボットハンドと未知対象物との接触点検出: 日本ロボット学会誌, Vol.8, No.6, pp.12-, (1990)

- [金子 91] 金子, 谷江: 3Dリンク系の準静的自己姿勢変形性に関する基礎的研究: 日本ロボット学会誌, Vol.9, No.7, pp.139-144, (1991)
- [金子 93] 金子, 本川: 最少外界センサ実現に基づく能動的触覚センシング: 日本ロボット学会誌, Vol.11, No.2, pp.246-254, (1993)
- [川人 86] 川人: 随意運動制御における適応と学習: 日本ロボット学会誌, Vol.4, No.2, pp.184-193, (1986)
- [川人 88] 川人, 宇野: 随意運動制御における適応と学習: 日本ロボット学会誌, Vol.6, No.3, pp.222-230, (1988)
- [北垣 90] 北垣: 能動的力覚センシングによる物体の迎合せ動作: 第8回日本ロボット学会学術講演会予稿集, pp.395-398, (1990)
- [北垣 92] 北垣他: 力覚センシングによる接触状態遷移検出の一手法: 第3回ロボットセンサシンポジウム, pp.67-70, (1992)
- [木下 91] 木下他: 分布形触覚センサの開発とその動的検出: 第1回日本ロボット学会ロボットシンポジウム, pp.163-168, (1991)
- [小菅 87] 小菅, 古田, 横山: ロボットの仮想内部モデル追従制御系: 計測自動制御学会論文集, Vol.24, No.1, pp.55-62, (1987)
- [小谷 93] 小谷他: 階層型ネットワークの汎化能力向上について: SICE'93, pp.713-714, (1993)
- [五味 90] 五味, 川人: 神経回路モデルを用いた学習型インピーダンス制御: 第8回日本ロボット学会学術講演会予稿集, pp.17-20, (1990)
- [阪口 92] 阪口, 中野: 外界からの制約に基づく図形の内部表現と認識: 計測自動制御学会論文集, Vol.28, No.3, pp.392-400, (1992)
- [佐藤 86] 佐藤他: 触覚センサフィードバックによるなぞり動作: 第4回ロボット学会学術講演会予稿集, pp.147-148, (1986)
- [澤田 88] 澤田他: プラグ挿入作業における探索と挿入: 第6回日本ロボット学会学術講演会予稿集, pp.71-74, (1988)
- [澤田 89] 澤田他: 順応動作における動作経路の指定と生成: 第7回日本ロボット学会学術講演会予稿集, pp.203-204, (1989)
- [澤田 90] 澤田他: 力覚センサを用いた位置測定法: 第8回日本ロボット学会学術講演会予稿集, pp.399-400, (1990)
- [塩崎 89] 塩崎, 内山, 箱守: ニューラルネットワークを用いたピン挿入作業の制御: 第7回日本ロボット学会学術講演会予稿集, pp.733-734, (1989)

- [末広 88] 末広, 高瀬: 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, pp.31-37, (1988)
- [末広 90] 末広, 高瀬: スキルに基づくマニピュレーションシステム: 日本ロボット学会誌, Vol.8, No.5, pp.551-562, (1990)
- [末広 91] 末広, 池内: 作業モデルに基づく視覚による組立作業理解と動作生成: 第9回日本ロボット学会学術講演会予稿集, pp.929-932, (1991)
- [末広 93] 末広, 池内: 面接触拘束に基づく組立作業の位置情報の修正: 日本ロボット学会誌, Vol.11, No.4, pp.541-549, (1993)
- [杉本 86] 杉本: ロボットアームの力フィードバック制御: 計測と制御, Vol.25, No.1, pp.45-50, (1986)
- [鈴木 92] 鈴木他: ポテンシャル法による接触状態の計算法: 精密工学会誌, Vol.58, No.5, pp.817-822, (1992)
- [精密工学会編 89] 精密工学会編: メカトロニクス: オーム社, (1989)
- [高瀬 75] 高瀬: マニピュレータの運動成分の一般分解とその制御: 計測自動制御学会論文集, Vol.12, No.3, pp.300-306, (1975)
- [高瀬 84] 高瀬: 束縛運動の表現と束縛下にあるマニピュレータの動的制御: 計測自動制御学会論文集, Vol.21, No.5, pp.508-513, (1984)
- [高瀬 85] 高瀬, 末広: 器用な手さぐり動作のための回転中心の検出: 第3回日本ロボット学会学術講演会予稿集, pp.103-104, (1985)
- [高瀬 86] 高瀬: ロボットアームの運動とその定式化: 計測と制御, Vol.25, No.1, pp.15-21, (1986)
- [高瀬 87] 高瀬, 末広: 環境拘束による物体運動の強制停止に関する理論解析: 第5回日本ロボット学会学術講演会予稿集, pp.261-262, (1987)
- [高瀬 89] 高瀬, 末広, 比留川: 仮想パラレルマニピュレータによる多点接触運動の制御: 第7回日本ロボット学会学術講演会予稿集, pp.523-524, (1989)
- [高瀬 90] 高瀬: 手による作業: 計測と制御, Vol.29, No.3, pp.213-219, (1990)
- [高瀬 93] 高瀬, 比留川: 微細動作の実現: 日本ロボット学会誌, Vol.11, No.2, pp.206-211, (1993)
- [田中 91] 田中, 清水, 土屋: 神経回路網による冗長マニピュレータの逆キネマティクス問題の解法: 計測自動制御学会論文集, Vol.27, No., pp.815-821, (1991)
- [立石 89] 立石, 山崎: 手書き数字認識における階層型ニューラルネット: 情報処理学会論文集, Vol.30, No.10, pp.1281-1288, (1989)
- [堤 91] 堤: ニューロコンピュータによるロボットの軌道生成: 日本ロボット学会誌, Vol.9, No.2, pp.217-221, (1991)

- [電子通信学会編 85] 電子通信学会編: ロボット工学とその応用: コロナ社, (1985)
- [中野 89] 中野他: ニューラルネットを用いたモデルベースドビジョン: 第7回日本ロボット学会学術講演会予稿集, pp.743-744, (1989)
- [中野 91] 中野: ニューロコンピュータの基礎: コロナ社, (1991)
- [中村 90] 中村他: 局所の特徴に基づく対象物同定の一手法: 日本ロボット学会誌, Vol.8, No.2, pp.152-159, (1990)
- [永田 90] 永田, 小笠原, 高瀬: 接触情報による把握位置姿勢のセンシング: 第8回日本ロボット学会学術講演会予稿集, pp.835-838, (1990)
- [永田 92] 永田他: 接触情報を用いたロボットの把握位置姿勢推定: 計測自動制御学会論文集, Vol.27, No.7, pp.783-789, (1992)
- [長田 90] 長田: 力覚センサを利用した対象物の拘束関係検出法の開発: 東京大学工学部精密機械工学科 1990 年度卒業論文, (1990)
- [野崎 89] 野崎他: 組立作業における拘束運動のシミュレーション: 第7回日本ロボット学会学術講演会予稿集, pp.519-520, (1989)
- [野崎 91] 野崎, 内山, 豊福: 組立作業における拘束運動のシミュレーション: 第7回日本ロボット学会学術講演会予稿集, pp.519-520, (1991)
- [橋本 90] 橋本他: ニューラルロボットによる運動スキルの自己組織化: 第8回日本ロボット学会学術講演会予稿集, pp.9-16, (1990)
- [長谷川 91] 長谷川他: 環境モデルと作業スキルの統合によるロボット作業システム: 日本ロボット学会誌, Vol.9, No.1, pp.66-74, (1991)
- [浜 92] 浜, 嘉数: トポロジ保存マップを用いた3次元構造推定に関する基礎研究: ロボティクス・メカトロニクス講演会講演論文集, Vol.B, pp.561-564, (1992)
- [久場 90] 久場, 橋本: 神経回路網によるマニピュレータの動作方向の学習: 第8回日本ロボット学会学術講演会予稿集, pp.37-42, (1990)
- [平井 88] 平井他: 凸多面体理論を用いたマニピュレーションの運動学とその把握および組立作業への応用: 計測自動制御学会論文集, Vol.24, No.12, pp.1284-1291, (1988)
- [平井 88] 平井, 浅田, 得丸: 組立作業における物体の接触状態遷移に関する運動学的解析とそのネットワーク表現の自動生成: 計測自動制御学会論文集, Vol.24, No.4, pp.406-413, (1988)
- [平井 90] 平井他: 組立作業における位置・力信号の解釈による識別とモデルをもとにした識別手続きの自動生成: 計測自動制御学会論文集, Vol.26, No.2, pp.225-232, (1990)
- [平井 92] 平井, 岩田: 幾何モデルをもとにしたダンピング制御パラメータの導出: 第10回日本ロボット学会学術講演会予稿集, pp.635-636, (1992)

- [平井 86] 平井他: 多自由度ロボットの仮想コンプライアンス制御: 計測自動制御学会論文集, Vol.22, No.3, (1986)
- [比留川 91] 比留川, 松井, 高瀬: 多面体間の接触による拘束条件を幾何モデルから導出する一般的なアルゴリズム: 日本ロボット学会誌, Vol.9, No.4, pp.415-426, (1991)
- [比留川 91] 比留川, 松井, 高瀬: 多面体間の接触による拘束条件の高速解法とその離脱動作計画への応用: 日本ロボット学会誌, Vol.9, No.7, pp.841-848, (1991)
- [比留川 93] 比留川: 組立作業過程における拘束の表現と解析: 日本ロボット学会誌, Vol.11, No.2, pp.192-200, (1993)
- [福田 90] 福田他: 神経回路網によるロボット・マニピュレータの位置・力制御の研究: 日本機械学会論文集 (C 編), Vol.56, No.527, pp.210-216, (1990)
- [福田 91] 福田, 柴田: ニューラルネットワークを用いた位置・力の適応制御: 日本ロボット学会誌, Vol.9, No.6, pp.778-783, (1991)
- [牧野 78] 牧野, 高野: 機械運動学: コロナ社, (1978)
- [松尾 93] 松尾, 岩城: 線形計画を用いた組立作業に対するコンプライアンス設定法: 日本ロボット学会誌, Vol.11, No.4, pp.574-580, (1993)
- [錦田 90] 錦田他: 神経回路モデルによるロボットの力制御 (1 自由度マニピュレータの制御): 日本ロボット学会誌, Vol.7, No.1, pp.47-51, (1990)
- [錦田 90] 錦田他: 神経回路モデルによるロボットマニピュレータの位置と力のハイブリッド制御 (2 自由度マニピュレータの適応制御): 日本ロボット学会誌, Vol.9, No.1, pp.45-54, (1991)
- [武藤 93] 武藤, 下倉: 力・速度情報の相補の利用による接触点検出法: 日本ロボット学会誌, Vol.11, No.3, pp.429-435, (1993)
- [柳原 92] 柳原, 北: 視覚と力覚情報に基づいた部品操作の研究: 日本機械学会論文集, Vol.58, No.553, pp.92-99, (1992)
- [藪田 91] 藪田, 山田: ニューラルネットワークによるロボットの力制御: 日本ロボット学会誌, Vol.9, No.2, pp.224-231, (1991)
- [山田 90] 山田, 藪田: ニューラルネットワークによるスキルの獲得: 第8回日本ロボット学会学術講演会予稿集, pp.43-46, (1990)
- [山崎 92] 山崎, 石川編著: センサフュージョン-実世界の能動的認識と知的再構成-: コロナ社 (1992)
- [山本 93] 山本, 末松: マニピュレータ逆動力学計算のためのニューラルネットワークの一構成法: 日本機械学会論文集 (C 編), Vol.59, No.559, pp.211-216, (1993)

- [横小路 93] 横小路: 接触状態の分類と組立手順: 日本ロボット学会誌, Vol.11, No.2, pp.185-191, (1993)
- [横山 90] 横山他: ニューラルネットワークを用いた障害物回避のための最適軌道生成: 第8回日本ロボット学会学術講演会予稿集, pp.31-32, (1990)
- [吉川 85] 吉川: ロボットアームの位置と力の動的ハイブリッド制御—手先拘束の記述と関節駆動力の算出—: 日本ロボット学会誌, Vol.3, No.6, pp.25-31, (1985)
- [吉川 85] 吉川: ロボットアームの位置と力の動的ハイブリッド制御: 日本ロボット学会誌, Vol.3, No.6, pp.531-537, (1985)
- [吉川 88] 吉川: ロボット制御基礎論: コロナ社, (1988)
- [吉澤 92] 吉澤他: ニューラルネットワークと超音波センサを用いた奥行き情報に基づく3次元物体認識: 電子情報通信学会論文誌, Vol.J75-D-II, No.7, pp.1160-1169, (1992)
- [渡辺 93] 渡辺, 清水: 階層型ニューラルネットワークの内部表現と汎化能力の関係: SICE'93, pp.541-542, (1993)

研究業績

[学術論文]

- Hiroaki Seki, Ken Sasaki and Masaharu Takano, "Search Motion of a Robot for Detecting Mechanically Constrained Objects", International Journal of the Japan Society for Precision Engineering, Vol.26, No.1, pp.76-77, 1992.3
- 高野 政晴, 永島 利明, 関 啓明, "指の腹による物体の把握と操りの運動学", 日本ロボット学会誌, Vol.14, No.1, pp.83-90, 1996.1

[国際会議]

- Toshiro Nagashima, Hiroaki Seki and Masaharu Takano, "Analysis and Simulation of Grasping/Manipulation by Multi-Fingersurface", Proceedings of the Third International Conference on Automation Robotics and Computer Vision, Vol.2, pp.962-966, 1994.11
- Hiroaki Seki, Ken Sasaki and Masaharu Takano, "Detection of Kinematic Constraint from Search Motion of a Robot Using Link Weights of a Neural Network", Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol.3, pp.498-503, 1995.8
- Takao Keishi, Hiroaki Seki and Masaharu Takano, "Analysis and Simulation of Grasping/Manipulation Dynamics by Multi-Fingersurfaces", Proceedings of the IFToMM Ninth World Congress on the Theory of Machines and Mechanisms, pp.2272-2276, 1995.9

[口頭発表]

- 佐々木 健, 関 啓明, 高野 政晴, "力覚センサを利用した対象物の拘束関係の検出", 1990年度精密工学会春季大会学術講演会講演論文集, pp.365-366, 1990.3
- 関 啓明, 佐々木 健, 高野 政晴, "力覚センサを利用した対象物の拘束関係の検出 第2報 3次元自由度の拘束の検出", 1991年度精密工学会春季大会学術講演会講演論文集, pp.999-1000, 1991.3
- 関 啓明, 佐々木 健, 高野 政晴, "ロボットの探り動作による対象物の拘束状態の検出 第1報 検出用ニューラルネットの構造と挙動", 第10回日本ロボット学会学術講演会予稿集, No.1, pp.25-26, 1992.10
- 関 啓明, 佐々木 健, 高野 政晴, "力覚センサを利用した対象物の拘束関係の検出 第3報 ニューラルネットワークの適用", 1993年度精密工学会春季大会学術講演会講演論文集, pp.1109-1110, 1993.3

- 関 啓明, 佐々木 健, 高野 政晴, "力覚センサを利用した対象物の拘束関係の検出 第4報 実測値を扱うニューラルネットの構造と学習", 1994年度精密工学会春季大会学術講演会講演論文集, pp.697-698, 1994.3
- 加藤 俊, 関 啓明, 高野 政晴, 佐々木 健, "人間の指による操り動作の計測システムの開発", 1994年度精密工学会秋季大会学術講演会講演論文集, pp.259-260, 1994.9
- 高尾 啓士, 高野 政晴, 佐々木 健, 関 啓明, "指の腹による物体把握と操りの運動学と力学に関する研究 第3報 物体の操りの力学的解析", 1994年度精密工学会秋季大会学術講演会講演論文集, pp.565-566, 1994.9
- 真鍋 敬治, 高野 政晴, 佐々木 健, 中尾 成行, 関 啓明, "エレベータのガイドレールのたわみに起因するパラメータ励振の可能性に関する研究", 1994年度精密工学会秋季大会学術講演会講演論文集, pp.309-310, 1994.9
- 吉見 隆洋, 高野 政晴, 佐々木 健, 関 啓明, "RECSコンセプトに基づく移動ロボットのナビゲーションと位置決め", 1995年度精密工学会春季大会学術講演会講演論文集, pp.139-140, 1995.3
- 伊佐治 真, 高野 政晴, 佐々木 健, 関 啓明, "人間の指による操り動作の計測システムの開発 第2報 装置の改良及び精度評価", 1995年度精密工学会春季大会学術講演会講演論文集, pp.143-144, 1995.3
- 斎藤 滋規, 高野 政晴, 佐々木 健, 関 啓明, "移動するワークへの挿入作業におけるSTS制御", 1995年度精密工学会春季大会学術講演会講演論文集, pp.581-582, 1995.3
- 真鍋 敬治, 高野 政晴, 佐々木 健, 中尾 成行, 関 啓明, "エレベータのガイドレールのたわみに起因するパラメータ励振の可能性に関する研究 第2報", 1995年度精密工学会春季大会学術講演会講演論文集, pp.351-352, 1995.3

ヤマザキ