平成 10 年度　修士論文

視覚情報を用いた
状態・行動空間の自律的生成

State and action space segmentation
using visual information

指導教官　　新井 民夫　教授

精密機械工学専攻

学生証番号　　76272

小林 祐一

目 次

第1章　序論

1.1 研究の背景

ロボットが世間の注目を集めるようになってから数十年が経つ.「人工知能」を搭載したロボットが人間の社会に進出する日もそう遠くはないと言われた．同時にそれは現在でもしばしば耳にする言葉である．しかし現実には，産業用マニピュレータなど一定の作業に特化したロボットは人間社会にとって不可欠なものとなっているが，人間と活動する空間を共有するロボットはいまだに実用のレベルには達していない．

従来の人工知能に基づくロボットが，人間が行動する空間のような多様に状況が変化しうる世界で動くことの困難さは，さまざまな角度から指摘されている．その中には，人工知能の基本的な発想自体の限界を指摘するものもある [Dreyfus92]. 従来のロボティクスは，ロボットの制御理論と人工知能とを組み合わせるという考え方をとっていたため，実世界でロボットを動かすときには，このような人工知能の原理的な問題をそのまま背負い込むことになっていた．人工知能の抱える問題の代表的なものは，フレーム問題*，シンボル接地問題などであり，認知，知能といった心理学，哲学での議論と深い関わりを持っている．

近年，このような認知の問題と深く関係する問題を，ロボット研究の立場から従来の人工知能と異なるアプローチで取り組もうとする研究が提唱されるようになっている [銅谷 99]．このようなアプローチのもつ重要な考え方は，認知，学習といった問題をロボットの立場から考えることである．学習は，従来の人工知能でも研究されてきたが，記号処理の中で完結した学習は，先に述べたシンボル接地問題を免れず，すでに設計された記号的表象と実世界の対象の認識とを結びつける部分に多大な困難を伴う．

ロボット研究の立場では，知能そのものに対する議論とは別に，設計者の負担を軽くするために学習を導入するという考え方が強い．このような考え方から見たロボットの学習の目的は，設計者があらかじめ問題を予測して適切な設計を行

*フレーム問題の「フレーム」とは，行動する主体が，現在の状況を自身の持っている外界に関する知識と照合する，自身の知識の枠組みのことである．固定された問題設定の中では，適切なフレームを設計者が用意することでロボットは現在の状況を把握することができる．しかし，人間が行動する世界のように状況が多様に変化しうるような問題設定では，ロボット自身が何に注目し何を無視するか (どのようにフレームを設定するか) を決めなくてはならない．これをあらかじめ設計することが困難である，というのがフレーム問題のあらましである [松原 90].

うことが困難であるような問題に対し，学習によってロボットが自律的に対応できることである [宮崎 95]．しかし，学習を研究することは，先に述べた人工知能の問題点に対する解決法を模索するためにも重要な意味を持つ．それは，従来の人工知能研究における学習が概念学習などの形で完全に記号論的表象の中で完結していたのに対し，ロボットの学習は，ロボットのセンサ情報・運動情報などに基づいた情報表現を可能にするためである．この考えは「身体性」と呼ばれ，前述の設計者の負担を減らすための学習という立場からも重要性が認識されつつある．

「身体性」とは，ロボットの外界や自身の行動に関する知識は，設計者が設計者の視点で与えるのでなく，ロボット自身のセンサ情報や運動能力に即した形で記述されるべきであるという考え方である [國吉 99]．先に述べたシンボル接地問題の「シンボルの接地」とは，記号的表象によって表現された情報を実世界と結び付けることである．記号的表象が先に存在すると，実世界の情報とそれを完全に符合させることができない．知識は元来人間の感覚・行動・欲求などから発生したものであり，その過程を飛び越えて知識だけを定義することによって生じた問題だといえる．身体性に基づく情報表現は，ロボットの感覚・行動から出発している点でこのような問題を回避する重要な鍵になるといわれている [マクドーマン 99]．また，これは，より少ない問題依存の作りこみで実世界で適応できるロボットを実現しようとする立場からみても必要な考え方である．

一方，ロボット学習の方法論としては，多くの研究は

- TD-λ，Q-Lerning などの強化学習によるもの
- ニューラルネットワークの学習理論を用いるもの
- 遺伝的アルゴリズム (GA)，遺伝的プログラミング (GP) などの進化的計算手法

およびこれらを組み合わせたものに大別される．強化学習は，エージェントが外界から与えられる報酬および罰[†]をもとに適切な行動を生成するための理論である．強化学習の中で代表的な手法である Q-Learning は，状態と行動の組を Q 値として評価し，報酬を遅延信号として実際に経験した状態に伝播させるという方法をとっている．強化学習や，遺伝的手法による行動学習などは，予め用意された状態変数 (あるいは離散化された状態) と行動出力の関係を獲得するものであり，状

[†]一般に強化学習では罰は負の報酬として表現される．

態変数や行動要素が学習するシステムにとって適切な形で用意されていることを前提としている．実際の問題に適用する際には，センサ入力を適切な形で離散化する準備が必要であり，センサの入力が多次元，複雑であるほど事前の設計が困難であることが知られている [Touzet97]．

ニューラルネットワークの理論には，パーセプトロンに端を発し，エネルギー最小化問題を解くもの (ボルツマンマシン，Hopfield Network)，入出力の写像関係を教師信号に基づいて学習するもの (Back Propagation)，競合学習によりパターン分類のためのベクトル量子化を行うもの (Kohonen の自己組織化マップ，Fuzzy ART) など多岐にわたる．行動学習に適用される代表的な例は Feed forward 型の多層ニューラルネットワークであり，入出力関係を教師あり学習 (BP) により獲得し，強化学習に適用する方法である [Barto83]．これらの研究は数個の距離センサなどニューラルネットワークの入力として直接利用可能な問題に適用されているが，センサが多次元になればなるほど教師あり学習のためのデータは大量に必要になり，センサ入力に直接これらの学習理論を適用することは困難になる．

ロボットの学習能力を高めるためには，センサの性能を向上させる必要がある．具体的には，多くの行動学習研究における移動ロボットは自身の周囲に取り付けられた距離センサを仮定しているが，外界の状態の認識能力を向上させるためには，視覚センサの導入が不可欠である．距離センサは高々10^0 から 10^1 程度の次元の入力となるが，視覚センサを導入するとその次元は $10^3, 10^4$ の次元になる．このように，センサの能力の向上は入力センサの多次元化を意味し，従来の学習手法はその多次元入力を直接に利用することはできない．これは，行動学習の研究に共通した問題であり，多次元入力ベクトルを処理する過程は設計者の恣意にゆだねられてきた．

1.2 関連の研究

前節では，認知，知能の問題へのアプローチ，ロボット工学研究における設計負担の軽減という両面から，ロボットの身体性に即した情報表現が重要であること，ならびに種々の行動学習研究の中で状態変数を自律的に獲得することが未解決の問題であることを述べた．本節では，状態変数を獲得する過程に着目した「状態空間の自律的生成」という研究について述べる．

これまでのロボットの学習研究の多くは，すでに固定された状態変数と行動出力の間の関係を獲得する，行動学習に関するものであった．ロボットの身体性に即した内部表現を得る過程は，これまで主に研究されてきた行動学習の段階よりも前の，センサ情報をもとに状態変数 (あるいは離散的な状態表現) を設定する過程であるといえる．この過程を含めたロボットの行動学習を考えようとする研究が「状態空間の自律的生成」という表現で行われている．このような問題設定の目的は，より低いレベルからロボットの自律性を実現するということだけでなく，身体性に即した内部表現を獲得することによりより柔軟な学習能力をもつシステムを実現することにある．

Asada *et al.*[Asada96] は，移動ロボットのボール押し操作 (サッカーロボットのシュート動作の獲得) に関して，対象物の位置，大きさなど，画像処理により抽出されたパラメータにより構成される入力ベクトル空間に対して，目標状態に到達する行動要素という観点から状態空間を分割する方法を提案している (**図.1.1**)．まず行動を離散化・固定された基本要素 (Action primitives) とし，同一の行動をとってすでに生成されている状態に到達する状態をひとまとまりのものとみなす．最初は目標状態のみから開始し，順次目標の近辺から状態を分割している．

Ishiguro *et al.*[Ishiguro96] は，移動ロボットの経路探索問題において，生の画像情報から統計的手法により状態空間分割を行って強化学習に利用する方法を提案している．事前にランダムな動作を多数行い，得た入力情報を主成分分析など統計的手法によって膨大な次元になる画像情報を圧縮し，判別関数を用いて状態空間を分割している．

また，Murao *et al.*[Murao97] は，強化学習の代表的手法である Q-Learning における Q 値を用いた状態の分割手法を提案している．移動ロボットの経路探索の

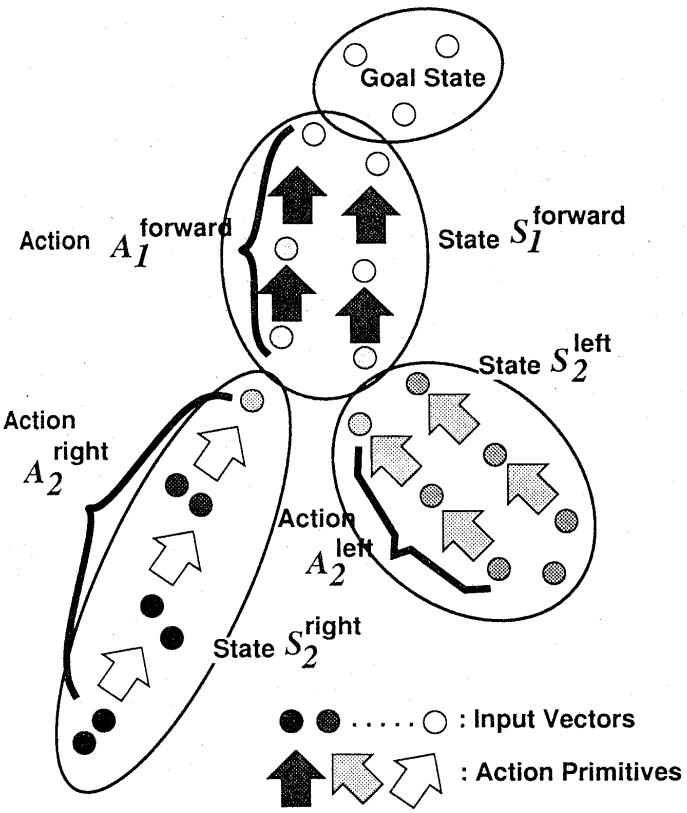


図. 1.1　目標から遡及する状態空間の生成方法 [Asada96]

中でも比較的容易なタスクを設定し，距離センサという低次元入力を仮定し，単純な経路の中での繰り返し試行において，Q 値を判別関数に利用して状態を分割している．

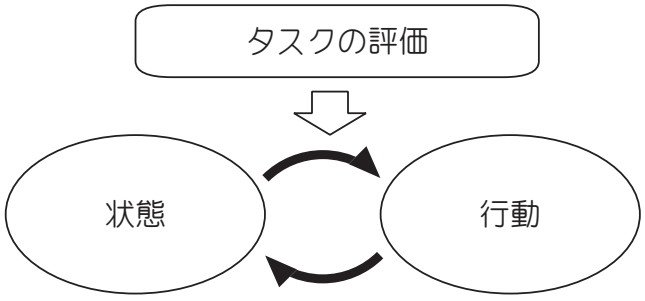


図. 1.2　状態・行動空間の生成と外的な評価

これらの研究に共通の問題の枠組みを**図.1.2** に示す．状態空間の分割を行うためにはあらかじめ行動が確定していなければならず，また，行動空間を生成するためには状態識別が成立していなければならない．それに加え，状態識別や行動が適切かを判別するには，外界からの評価が必要である．これらの研究は共通して，強化学習の特色である遅延信号としての報酬を利用して状態識別の適切さを判別している．

上記のように，最終的な報酬を遅延信号として用いて状態を生成する研究は，行動をあらかじめ離散化・固定しておく必要がある [Ishiguro96][Murao97]．また，画像入力はその画素数分の次元をもった入力であり，その次元を圧縮するために，研究 [Asada96] では画像処理によるパラメータ抽出を行い，研究 [Ishiguro96] では事前に得たデータを基に統計処理を行っている．

遅延報酬を用いて状態と行動とを同時に無の状態から生成することは原理的に困難であり，何らかの形で分割のための手がかりを多くしなければならない [Asada96]．また，強化学習は，最終的な報酬を遅れ信号として扱うことができる反面，膨大な試行を繰り返さなければならなず [Ishiguro96]，問題が大きくなればなるほど必要な試行数は膨大になる．

ロボットの身体性に即した情報表現を目指すためには，センサ情報を設計者の作りこみを介さない形で利用しなければならない．視覚センサを利用する場合は，センサ入力の次元が膨大になるため，何らかの画像処理を行うことが必要であるが，対象物の形状などを仮定するなどの画像処理における設計者の恣意による特徴抽出は身体性の表現のためには排されなければならない．また，獲得した知識を拡張可能にするためには，データを収集する段階とそれを学習に利用する段階とに分かれず，逐次的に構造を自己組織するシステムであることが望ましい．

以上をまとめると，

- 画像処理などの事前の特徴抽出を設計者が行う，あるいはセンサ情報処理の過程と学習の過程を分離させるということを回避し，多次元センサ入力から直接に状態空間を生成する
- 行動をあらかじめ離散化，固定するのでなく，状態空間の自律的生成とと同時に自己組織的に獲得する

ということが未解決の問題であるといえる.

1.3 研究の目的

前節の議論より，本研究では多次元センサとして関連の研究と同様に視覚センサを用いることとし，

- 視覚入力から事前の処理を経ずに状態空間を生成すること
- 状態だけでなく連続値の行動を同時に自己組織すること

ということを本研究の目的とする．

状態の生成方法として，パタン分類の代表的なアルゴリズムであるベクトル量子化アルゴリズムを利用する．逐次的に状態を生成していくための適用方法を提案し，量子化された状態同士の位相関係を利用することにより，位相近傍の状態同士を結びつけるように行動を自己組織的に修正する．

具体的な課題としてマニピュレータによる対象物の押し操作を想定し，位相構造を用いて逐次的に知識を拡張するシステムを提案する．

問題設定として，本研究では状態と行動とを同時に生成するための評価の枠組みとして，**即時的な 2 値の評価**を用いる．即時的評価とは，一連の行動の結果でなく，時間を微小ステップに区切ったときの各ステップ毎に与えられる評価ということであり，2 値の評価とは，連続な評価関数でなく良いか悪いかという情報のみを与える評価である．これは問題設定としては遅延報酬をもちいるという関連の研究と比較してやさしい問題を扱うことになるが，状態と行動とを同時に生成するためには，原理的に即時的な評価が必要であるという考えに基づく．評価が即時的に与えられるとしても，逐次的に与えられる評価をロボットのセンサ，行動の側から解釈しなおし，身体性に即した内部表現を獲得するという意味を持つ．

また，評価が 2 値で与えられるというのは強化学習において報酬や罰が与えられるのに対応しており，ある程度の幅を持った評価が与えられる．2 値の評価を厳しくすれば性能の良い行動が獲得されるが，その反面行動獲得のための試行数は多く必要になる．評価をゆるくすればより少ない試行で学習が可能になるが，獲得された行動には改善の余地がある．このような Trade off の問題を，本研究では行動を修正する方法を提案することによって解決を図る．

1.4 本論文の構成

第 1 章では，ロボット学習に関する問題点を示し，本研究の，視覚情報からの状態空間の生成，状態と行動の自己組織化という目的を示した．

第 2 章では，状態空間を構成するための基本的な考えを示し，そのために必要なベクトル量子化アルゴリズムについて概説し，本研究で必要とする位相構造を保持したベクトル量子化について述べる．

第 3 章では，2 章で示した状態空間の生成法に基づき，状態と行動とを自己組織的に生成・修正する方法について述べる．行動を自己組織するための考え方を示し，位相構造を利用した行動の自己組織アルゴリズムを提案する．

第 4 章では，シミュレーションにより状態・行動空間の自律的生成をおこなう提案手法の有効性・問題点について検証する．

第 5 章では，実験を行う．シミュレーション結果を用いて実機での押し操作実験を行い，シミュレーションと実機の環境とのずれについて論じる．また，提案手法をロボット自身のセンサ情報を評価信号として用いた問題に適用し，その有効性を検証する．

第 6 章では本研究の結論，および今後の展望を述べる．

第2章　ベクトル量子化アルゴリズムによる状態表現

2.1 はじめに

本章では，状態空間生成のための基本的な考え方を述べ，状態空間を生成するためのベクトル量子化アルゴリズムについて検討する．本研究の状態・行動空間の生成に必要な位相構造を保持する方法について述べ，状態と行動の生成アルゴリズムを提案する．

2.2 節では，本研究における状態の生成の基本的な考え方を示し，ベクトル量子化アルゴリズムに求められる要件を考察する．

2.3 節では，Kohonen の自己組織化マップとその応用としてのベクトル量子化の原理とその機能について述べ, 位相構造を保持するネットワークについて述べる．

2.4 節では，2.3 節で示した位相構造を保持するアルゴリズムに基づき，本研究における状態空間の生成方法について述べる．

2.2 状態空間生成の考え方

2.2.1 状態・行動の定義

まず，本研究における状態空間および行動空間の定義を述べる．ロボットはセンサから入力を得，最終的にアクチュエータに出力する (**図.2.1**)．センサ入力には様々なものがあるが，一般にセンサ能力を向上させることは入力次元が高次元 (多次元) になることを意味する．タスクの達成のためには，この多次元の入力を適切な行動に結びつけるための適切な表現に変換する必要がある．設計者がこの過程を扱う場合は，画像処理などの特徴抽出，離散化などがこれに相当する．このような，センサ入力空間から行動にとって必要な形へ射影された空間を状態空間と呼ぶ (**図.2.2**).

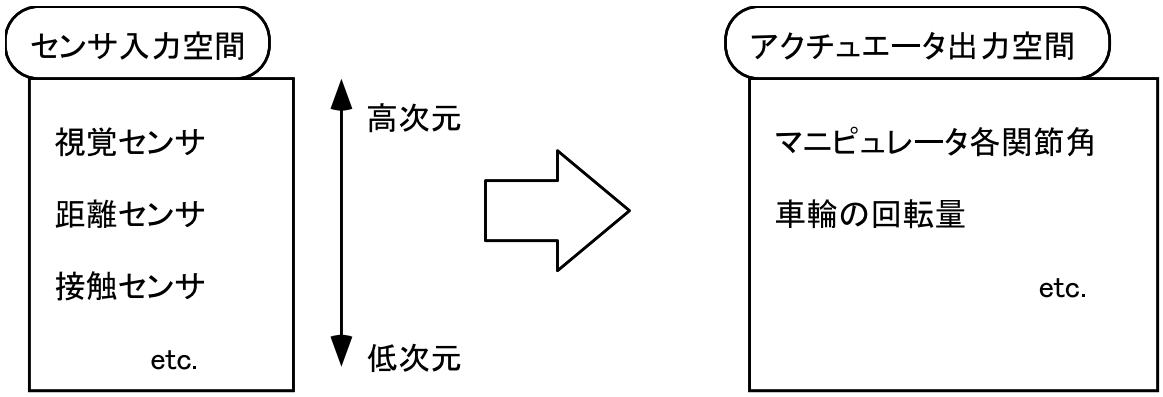


図. 2.1 ロボットのセンサ入力とアクチュエータ出力

一方，アクチュエータ出力はマニピュレータの場合は各関節角，移動ロボットの場合は車輪の回転量などであるが，タスクの達成のためにアクチュエータ出力の変換をあらかじめ得ておく場合も存在する．マニピュレータでは逆運動学の変換，移動ロボットでは絶対位置，姿勢への変換などである．このような，状態空間からアクチュエータ出力空間への出力を行う間の，アクチュエータ出力空間への射影を行う空間を行動空間と呼ぶ．視覚センサなどの多次元センサ入力を仮定した場合，センサ入力空間と状態空間の間の変換に比べ，行動空間とアクチュエータ出力空間の間の変換は情報量の変化が少なく，タスクに対する依存性も低い．

行動の表現方法としては，一連のアクチュエータ出力の時系列という考え方も

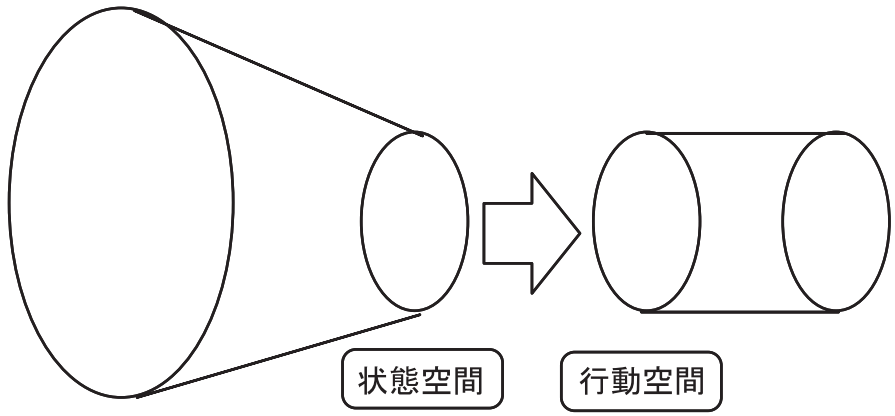


図. 2.2　状態空間と行動空間

あるが，本研究では，センサ入力をもとにアクチュエータ出力を決定するという過程をより原始的なレベルから考える．そのため，行動は，一連のアクチュエータ出力の時系列としてではなく，その結果がセンサによって把握できる程度，すなわちアクチュエータ出力によりセンサ入力に変化が検知される程度に微小の行動空間内の変数で表される．

以上の議論をまとめ，本研究では，センサ入力空間からアクチュエータ出力空間への射影を得る過程を状態空間，行動空間を介した変換としてとらえ，状態空間を

行動決定に利用するためにセンサ入力空間を射影した空間

と定義する．また，行動空間を

状態空間との関連を表現する，アクチュエータ出力空間に射影される空間

と定義する．

なお，この文脈に従い,「状態」という言葉をロボットのセンサ入力から行動を決定するのに適切な形で取り出された (あるいは離散化された) 状態変数という意

味で用い,「行動」という言葉は微小なアクチュエータ出力へ変換できる出力変数を表すものとする．また，時間は微小のステップに離散化され，センサ入力に対し状態を識別し，行動出力を行うという過程を繰り返すものとする．

2.2.2 状態の表現方法

第 1 章でも述べたように，状態と行動とは，評価に基づいて互いを分節しあう関係にある (**図.2.3**)．ここで評価とは，前述の各時刻ステップに対して行動の結果を一定の基準に基づいて良いか悪いかを決定するものとする．図左側のように，状態を固定して考えれば，行動が異なっても評価が変わらない限りは，その行動の違いにはその評価に関する限り大きな意味はない．つまり，同じ状態に対して評価が変わる行動が，同じ状態に対する識別するべき行動ということになる．

逆に行動を固定して考えれば，図右側に示すように，状態が異なっても評価が変わらない限りは，その状態の違いにはその評価に関する限り大きな意味はない．つまり，同じ行動に対して評価の変わる状態が，同じ行動に対して識別すべき状態ということになる．

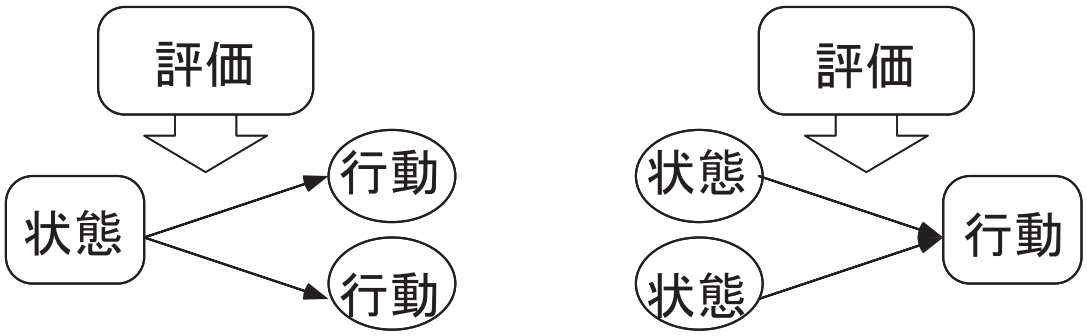


図. 2.3　状態と行動の関係

本研究で扱う問題は，センサ入力空間から状態空間への写像を得る過程であり，センサ入力空間の方がアクチュエータ出力空間に対して多次元であるような問題である．したがって，上述の考え方のうち状態を識別するほうに着目し，**同じ行動をとって評価が異なる状態同士を識別する**という考え方に基づいて状態の生成を考える.「識別」を入力ベクトルの離散化ととらえると，この考えを表現するためには，離散化された状態と，その状態に対応する行動という基本単位を考えれ

ばよい．以上より，離散化されたそれぞれの状態が行動出力値を保持するという状態・行動表現を基礎とする．

状態をただ離散化するだけでは*，離散化された状態同士の関係は記述されない．しかし，センサ入力空間から状態空間へ写像を形成する上での情報量の保存を考えると，状態は離散化されるだけでなく，センサ入力空間における関係を何らかの形で保存することが望まれる．この写像の前後において保存されるべき情報を，本研究では位相関係ととらえる．すなわち，位相関係を保存した状態の離散化を要件とする．位相関係の利用方法については 3 章でより詳しく論じる．

また，評価は即時的に与えられるため，行動の実行に際しては各ステップの行動に対して逐次的に与えられる．状態および行動の表現はこのようなオンラインの評価信号に基づき逐次的に生成・変更することを可能にするものでなくてはならない．

以上の議論より，状態・行動空間の自律的生成のための要件を以下に示す．

- 入力ベクトルが離散的な表現に変換されること
- 逐次的に状態表現を拡張，変更することができること
- 入力空間の位相関係を保持することができること

以下で，これらの条件を満足するアルゴリズムについて関連の研究を参照し，本研究における適用方法を議論する．次節では，入力ベクトルを離散的な表現に変換する代表的な方法であるベクトル量子化アルゴリズムについて述べる．

*強化学習の代表的手法である Q-Learning における状態と行動の関係を示したルックアップテーブルなどがこれにあたる．

2.3 ベクトル量子化アルゴリズム

本節では，自己組織化マップ (SOM) の基本原理と，それらを応用したベクトル量子化アルゴリズムを紹介する．最後に前章で述べた本研究での要求に応える方法について議論する．

2.3.1 自己組織化マップ (SOM) の基本原理

自己組織化マップ (Self Organizing Maps, 以下 SOM) とは，T. Kohonen により提案された競合学習によるベクトル量子化アルゴリズムである [Kohonen96]．SOM は入力ベクトルと同次元のベクトル (結合係数ベクトルと呼ぶ) を表現するノード (あるいはニューロン) を格子状に配置して構成される．入力ベクトルに対してノード群が一つの層をなし，ある入力に対して必ず一つのノードが勝者となって発火する．勝者となるノードは

$$\| \boldsymbol{x} - \boldsymbol{w}_c \| = \min_i \{ \| \boldsymbol{x} - \boldsymbol{w}_i \| \} \qquad (2.3.1)$$

なる c 番目のノードである．ここで，$\boldsymbol{x}$ は入力ベクトル，$\boldsymbol{w}_i$ はそれに対応する i 番目のノードの結合係数ベクトルである．勝者となるノードはユークリッド距離最小という意味で最整合ノードとも呼ぶ．結合係数ベクトルは，

$$\boldsymbol{w}_c(t+1) = \boldsymbol{w}_c(t) + \alpha(t)[\boldsymbol{x}(t) - \boldsymbol{w}_c(t)] \qquad (2.3.2)$$

によって更新する．最も単純なアルゴリズムでは，更新幅 $\alpha(t)$ は

$$\alpha(t) = \alpha_0(1 - \frac{t}{T}) \qquad (2.3.3)$$

で与えられる [Dayhoff92]．ここで t は学習の訓練回数，T は行われるべき訓練の全回数である．更新幅は徐々に減少し，最終的には 0 に収束する．

2 次元格子上に配置された SOM 層において，近傍を**図.2.4** のように定義する．勝者となったノードだけでなく近傍のノードにも (2.3.2) 式の同様の結合係数の更新をおこなうことにより，局所的な位置関係を保持したままマッピングをおこなう．近傍を決定する幅 d は最も単純なアルゴリズムでは以下のように設定される．

$$d(t) = d_0(1 - \frac{t}{T}) \qquad (2.3.4)$$

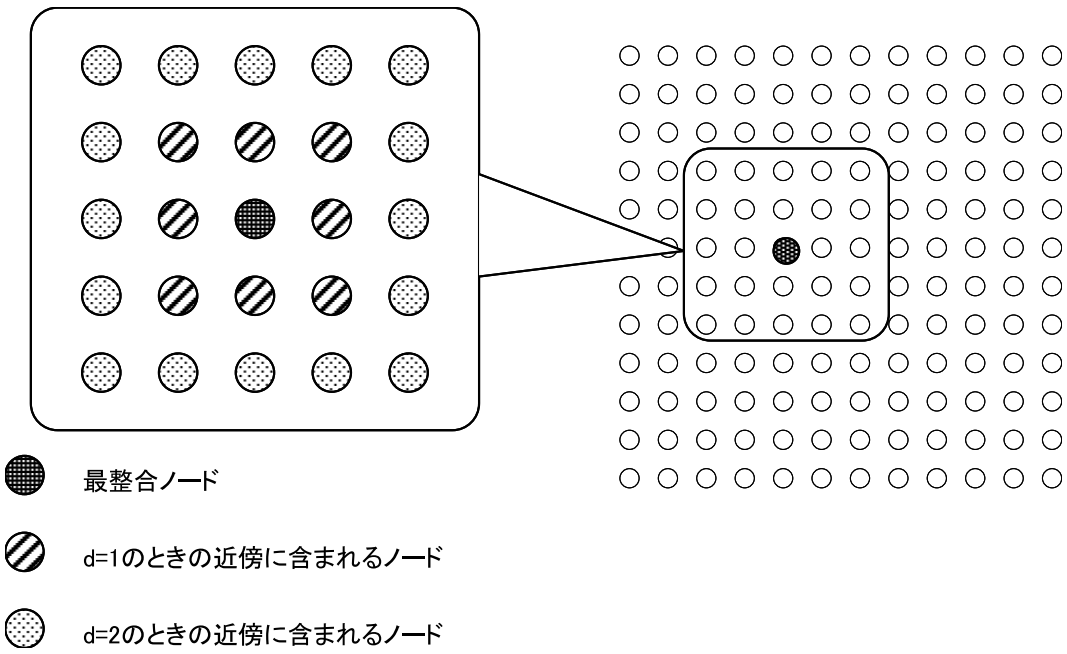


図. 2.4　2 次元格子状 SOM とその最整合ノードの近傍

以上のアルゴリズムを用いて 1 次元格子によって 2 次元平面上のベクトルを量子化した結果を**図.2.5〜2.8** に示す．入力空間は $M = \{x, y | 0 \leq x, y \leq 1\}$ であり，図の 2 次元平面に相当する空間から一様乱数を発生させて得る．各点はノードであり，その位置が結合係数ベクトルを表している．ノード同士を結ぶ線は 1 次元格子における隣接関係を表している．各ノードの結合の初期値 $\boldsymbol{w}_i$ は正方形中心付近の小さな乱数値をとる．計算ステップは 2000 回，ノード数は 30，$\alpha = 0.3$ である．計算ステップの初期状態から最終状態 (2000 回終了時) までの遷移状態をそれぞれの図で表す．計算ステップを経るにつれ，一次元格子が 2 次元平面に全体を覆うように広がり，格子上で隣接するノードは対応する入力ベクトル空間 (2 次元平面) でも隣接していることがわかる．

通常 SOM のノードは，2 次元または 3 次元の格子状に配列され，多次元の入力はその確率密度を反映して格子上各ノードの結合係数に写像される．SOM の特徴は，

- 多次元の入力が低次元 (多くは 2 次元) の格子上に写像される
- 入力ベクトルが位相関係を保持して写像される

ことである．位相関係を保持するとは，格子上に配置されたノードに対し，格子

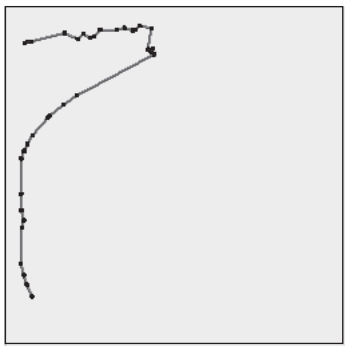


図. 2.5　1 次元格子の SOM(1)

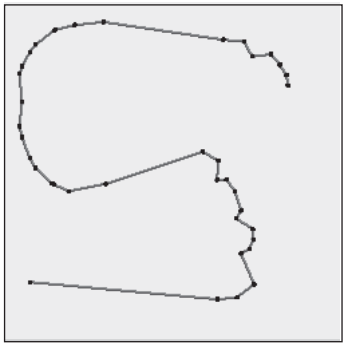


図. 2.6　1 次元格子の SOM(2)

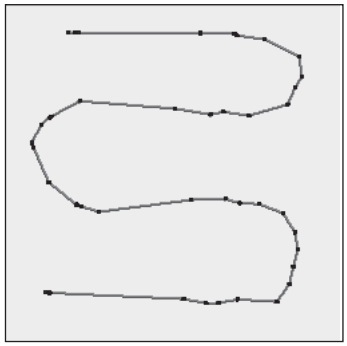


図. 2.7　1 次元格子の SOM(3)

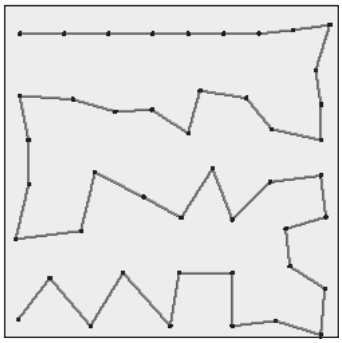


図. 2.8　1 次元格子の SOM(4)

上で隣接する† ノード同士の，それぞれ対応する入力ベクトル空間が隣接しているということを表す.

しかし，基本 SOM は前節で述べた本研究における必要に完全に応えるわけではない．基本的な SOM はノード数が最初から固定されており，設計者が試行錯誤により適当な個数を設計しなくてはならない．SOM の位相構造を保持する性質を残し，逐次的にノード数を変更可能なアルゴリズムについて述べる.

2.3.2　SOM を応用したベクトル量子化アルゴリズム

基本的な SOM はあらかじめ用意された個数の格子状のノードにより表現されるものであり，適切なノードの個数，学習率などは設計者の試行錯誤や経験によって決定されなければならない．SOM を応用，変形したベクトル量子化のアルゴリ

†後に述べる TRN などのネットワーク状に表現されたノードにおいては，アークで結ばれたノード同士を隣接しているものと考える.

ズムは数多く提案されているが，ここでは，先に述べた位相関係を保持する能力，およびノードの個数を固定しないで必要に応じて拡張していく能力に注目して応用されたアルゴリズムについて述べる．

Self Creating and Organizing Neural Network(SCONN)

SCONN アルゴリズムは，基本 SOM の 2 次元に固定された位相構造のために生じる欠点を克服するために提案されたアルゴリズムである [Choi94]．基本 SOM では，位相構造が格子状に固定されているため，入力空間の確率分布によっては，実際に入力があっても発火しないノード (Dead node) が生じる可能性がある． 2 次

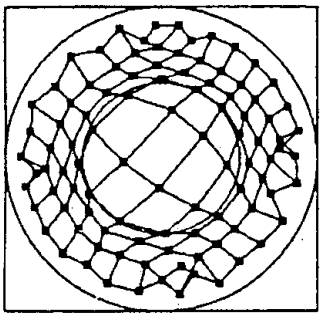


図. 2.9 Dead node が生じる例 (1)

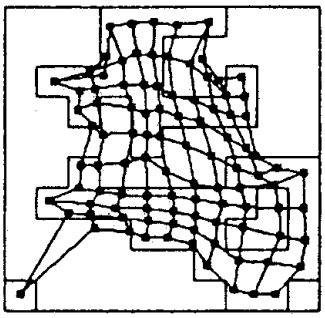


図. 2.10 Dead node が生じる例 (2)

元空間の部分空間を入力空間をに対する SOM の量子化で Dead node の生じる例を**図.2.10**，**図.2.9** に示す．**図.2.9** はドーナツ状の入力空間を 2 次元格子に写像した結果，中央の入力ベクトルのない領域に結合係数を持つノードが存在している．**図.2.10** は矩形の離散的な領域に 2 次元格子が充分に対応できない例である．

この問題を回避するために SCONN アルゴリズムでは木構造を導入することを提案している．以下にそのアルゴリズムを示す．各ノード i には，SOM と同様の入力と同次元 D の結合係数ベクトル $\boldsymbol{w}_i$ 以外に，他のノードに結合が存在するかどうかを表す C_{ij} および時間によってきまる　activation level $\theta(t)$ という変数が存在する．

(1) $\boldsymbol{w}_i \in R^D (i = 1, \cdots, N)$，各ノード同士の結合 C_{ij} を 0 に初期化する

(2) ある入力ベクトルを与える．

(3) ネットワーク上のそれぞれのノード i について，入力 $\boldsymbol{x}_i(t)$ と結合係数ベク

トル $\boldsymbol{w}_j$ の間のユークリッド距離を次式にしたがって計算する

$$d_j^2 = \| \boldsymbol{x}(t) - \boldsymbol{w}_j \| = \sum_{i=1}^{D} (x_i(t) - w_{ij}(t))^2 \qquad (2.3.5)$$

(4) ユークリッド距離最小の勝者ノードを決定する

(5) 勝者ノードが active ならば 6. へ，inactive ならば 7. へ行く

(6) active な勝者ノード (とその親，子ノード) に対して結合係数ベクトルの更新を行い，全ノードの activation を下げる

(7) inactive な勝者ノードに対して新たに子ノードを生成し，全ノードの activation level $\theta(t)$ を下げる

(8) 2. に戻る

結合係数ベクトルの更新式は

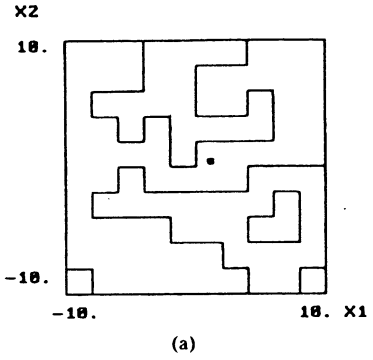
$$\boldsymbol{w}_i^{\mathrm{new}} = \boldsymbol{w}_i^{\mathrm{old}} + \alpha(t)(\boldsymbol{x}_i(t) - \boldsymbol{w}_i^{\mathrm{old}}) \qquad (2.3.6)$$

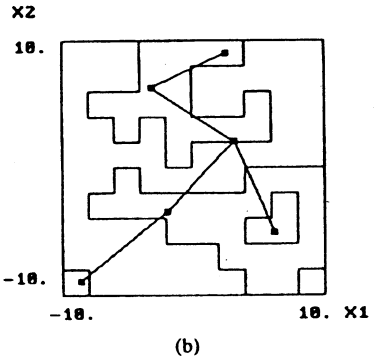
で，基本的な SOM アルゴリズムと同様であるが，この更新は基本 SOM の格子上の近傍ではなく，親子関係で表された近傍に対して行われる．また，各ノードの activation は勝者ノードの入力ベクトルとのユークリッド距離 d_{wj} によって決定される．

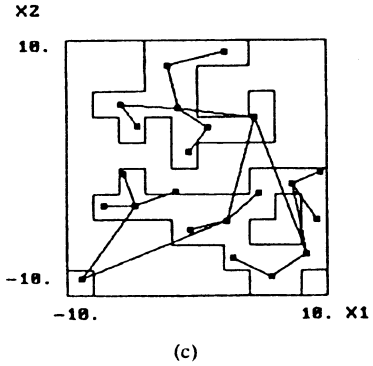
$$d_{wj} < \theta(t) \qquad (2.3.7)$$

ならば勝者ノード j は active となる．これはすなわち，勝者ノードの入力ベクトルに対するユークリッド距離がある閾値以上の時は木構造における新たなノードを生成し，その閾値 $\theta(t)$ を焼きなまし的に減少させていくということを意味している．**図.2.11**，**図.2.12** に 2 次元上の複雑な入力空間に対して SCONN アルゴリズムによりノードが生成されていく様子を示す．**図.2.10** の基本 SOM で Dead node の存在していた量子化と比較して，矩形の離散的な入力空間に適応した量子化が達成されていることがわかる．

SCONN アルゴリズムは，基本 SOM の持っているノード数が固定であること，位相構造が固定であること，という問題に対して木構造を逐次的に生成するという方法を提案している．これによりどの入力に対しても最整合とならないノードの発生を防いでいるが，







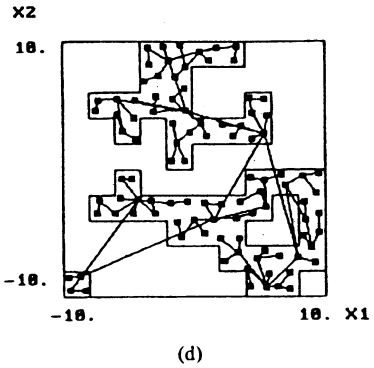


図. 2.11 アルゴリズムの適用例 (1)

図. 2.12 アルゴリズムの適用例 (2)

- 木構造が位相関係を充分に表現しているという保証がない
- 新たなノード (ここでは子ノード) を生成するルールへの疑問

という問題を持つ．ネットワークの表現として木構造を採用することは，入力ベクトル分類の上で相関計算 (最整合ノード発見の計算) の時間を短縮するという意味を持つが，この手法ではその利点が生かされていない．SOM の情報表現の効率化を生かす意味で木構造を導入する研究もなされている [Song98, Bursevski96] が，本研究では状態と行動の生成という状態分類の質的な問題を扱うため深く言及することはしない．

SCONN アルゴリズムでは Activation level という考えを用いて各ノードの担当する入力空間の範囲を設定しているが，2.2 節で述べたような状態同士の識別を行う上で，各ノードの担当する入力空間の範囲を設定する考えは識別の境界が明確にできないという問題が生じさせる．本研究では基本的にユークリッド距離最小のノードを最整合とするという考え方を用い，このような混乱を避ける．入力ベクトルに対するユークリッド距離をノードの出力の強さに反映させる考え方とし

ては Radial basis function [Fritzke94] があり，状態識別が充分に行われた局面では Radial basis function を用いた方法を提案する．この利用方法については位相関係の利用法とあわせて 3 章 で詳しく述べる．

位相構造に関して，木構造よりもより柔軟なネットワークの形で表現する方法が提案されており，次節でそのアルゴリズムについて説明する．ノードを生成していくルールに関して考察を行い，その上で本研究における状態の生成方法について述べる．

Topology Representing Network(TRN)

TRN とは Topology Representing Networks の略で，位相構造を保存して[‡]ベクトル量子化を行うことを目的として提案されたアルゴリズムである [Martinetz94]. 通常の SOM は位相近傍が 2 次元または 3 次元の格子上隣接するノードという形で固定されているため，入力空間の次元や入力ベクトルの疎密に対応することができない．これに対し，TRN アルゴリズムは，位相近傍を動的に変更していくことによってより入力空間の構造に柔軟に適応したマッピングを生成することを利点としている．以下に TRN アルゴリズムの概要を示す.

ネットワークには N 個のノードが存在し，各ノードが D 次元の入力に対応する結合係数 w_i をもっている．入力信号 x は R^D の部分空間 M の要素である.[§]

(1) $\boldsymbol{w}_i \in R^D (i = 1, \cdots, N)$，各ノード (ノード i とノード j) 同士の結合 C_{ij} を 0 に初期化する ($C_{ij} = 0$ は結合なし，$C_{ij} = 1$ は結合ありを意味する)

(2) ある入力ベクトル $\boldsymbol{x} \in M (\subset R^D))$ を与える．

(3) 入力 x に対し

$$\| \boldsymbol{x} - \boldsymbol{w}_j \| = \sum_{i=1}^{D} (x_i - w_{ij})^2 \qquad (2.3.8)$$

[‡]ここでの位相構造の保存とは，入力ベクトルの離散的表現であるノード同士の関係において，隣接する (アークで結ばれている) ノードに対応する入力ベクトルの部分空間同士が隣接していることを表す.

[§]ここでは TRN アルゴリズムのうち，位相構造を保存する部分を説明する．すなわち，本来はここに示すアルゴリズムに加え，結合係数 $\boldsymbol{w}$ の更新を，入力に対するユークリッド距離の小さいノードに対して行う.

によってユークリッド距離を計算し，この距離の最小のノード i_0 および 2 番目に小さいノード i_1 を発見する (ノード i_0 を最整合ノードと呼ぶ)

(4) $C_{i_0 i_1} = 0$ ならば $C_{i_0 i_1} = 1, t_{i_0 i_1} = 0$ とし (ノード i_0 とノード i_1 を結合させ), $C_{i_0 i_1} = 1$ ならば $t_{i_0 i_1} = 0$ とする (結合を refresh する)

(5) ノード i_0 にすでに存在するすべての結合に対して $t_{i_0 j} = t_{i_0 j} + 1 (C_{i_0 j} = 1)$ によって結合を古くする

(6) $C_{i_0 j} = 1$ かつ $t_{i_0 j} > T$ なる j について $C_{i_0 j} = 0$ とする (ノード i_0 の古くなった結合を削除する)

(7) 2. に戻る

ここで t_{ij} はノード i とノード j の間の結合の古さを表し，寿命 T に達すると結合は消滅する．すなわち，結合の生成・消滅を最整合ノードに対してもっとも近い位置にいるかどうかによって決定し，持続して最整合ノードの最近傍に存在するノードの結合を最終的に保存する．

図.2.13 に TRN によるベクトル量子化の様子を示す．図左上が初期状態であり，入力ベクトルは 3 次元直方体の内部，2 次元平面上，1 次元線分上，円周上に分布する．計算ステップを経るに従い，それぞれの入力空間の形状に従った位相構造を形成しながら量子化を行っていることがわかる．

TRN アルゴリズム自体は固定された個数のノードに対して適用される学習則であり，このままでは逐次的にノード数を変化させることができない．この問題に対処する方法として，TRN のアルゴリズムを利用しながらノードの数を必要に応じて増加させていく Dynamic cell structure(DCS) というアルゴリズムが提案されている [Bruske95]．これは，各ノードに発火頻度を記録しておき，発火頻度の多い (閾値より発火頻度が高くなった) ノードは分割される，という方法である．

以上の議論を表にまとめたものを**表 2.1** に示す．位相構造の保持性能の観点からは TRN アルゴリズムが有効であるが，ノード生成の方法は DCS の方法をそのまま適用することはできない．次節では，本研究における要件から，この応用方法を論じる．また，SCONN とは異なるユークリッド距離の利用方法については 3 章において論じる．

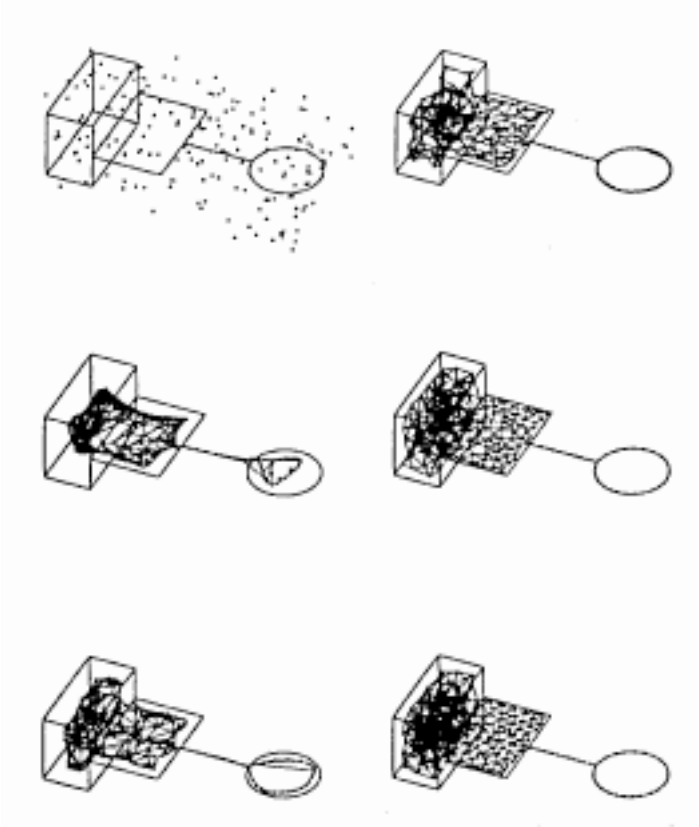


図. 2.13　TRN によるベクトル量子化の様子

表 2.1　各種 SOM アルゴリズムの比較

	ノード生成の方法	位相構造
基本 SOM	固定	格子状
SCONN	ユークリッド距離に基づく分割	木構造
TRN	固定	ネットワーク構造
DCS	発火頻度に基づく分割	ネットワーク構造

2.4 状態空間の構成方法

位相構造を保持してネットワークを構成するためにはTRNアルゴリズムが有効であり，本研究ではこれを基本的に採用する．ノードの個数を変更する方法として，Dynamic cell structureではノードの発火頻度によってノードを増加させている[Bruske95]．入力空間における入力の密度をネットワークに反映させるという考え方からはこの方法は妥当であるが，本研究で必要とする「同じ行動をとって評価の異なる状態同士を識別する」という考え方からは必ずしも妥当ではない．入力頻度が高いからといってその状態が細かく識別すべきとは言えないからである．

したがって，本研究ではノード同士の関係を記述する部分にはTRNアルゴリズムを用い，ノードの増加に関しては外界からの信号に基づいて行うものとする．すなわち，同じ行動をとって評価が異なるときに，その時点での入力ベクトルを結合係数とするノードを生成する．また，各ノードには出力を割り当て，最整合ノードに対応する行動出力として記憶する．また，状態の固定した表現を得るために，(2.3.2) 式のような結合係数 $\boldsymbol{w}$ の更新は行わない．それぞれのノードは入力次元 D と同次元の結合係数ベクトル $w \in R^D$ および行動出力ベクトル $\boldsymbol{o}$ をもつ．TRNアルゴリズムを用いた状態・行動生成のアルゴリズムは以下のプロセスからなる．

(1) 画像入力 $\boldsymbol{x}$ を得る

(2) 最整合ノード b を発見し，TRNアルゴリズムに基づきノード間の結合を更新する

(3) 行動出力 $\boldsymbol{o}_b$ を実行する

(4) 行動に対する 2 値の評価を得る

(5) 悪い評価ならば，$\boldsymbol{w}_j = \boldsymbol{x}$ なる新規状態ノードを生成し，新たな行動を探索，対応させる

(6) 1. に戻る

状態が位相構造を保持したネットワークの形で表現されたとき，これらのノードを状態ノードと呼ぶこととする．状態ノードは初期状態では 1 個であり，その結合係数ベクトル $\boldsymbol{w}_0$ および行動出力ベクトル $\boldsymbol{o}_0$ には乱数を割り当てる．また，

図.2.14 に，センサ入力から行動出力を与えるまでの過程を示す．状態ノードは先に述べた TRN アルゴリズムの一部に基づいて位相近傍を表すアークにより結ばれている．センサ入力ベクトルに対し，最整合ノードが発火する．それぞれの状態ノードは行動出力ベクトル $\boldsymbol{o}$ を保持する．

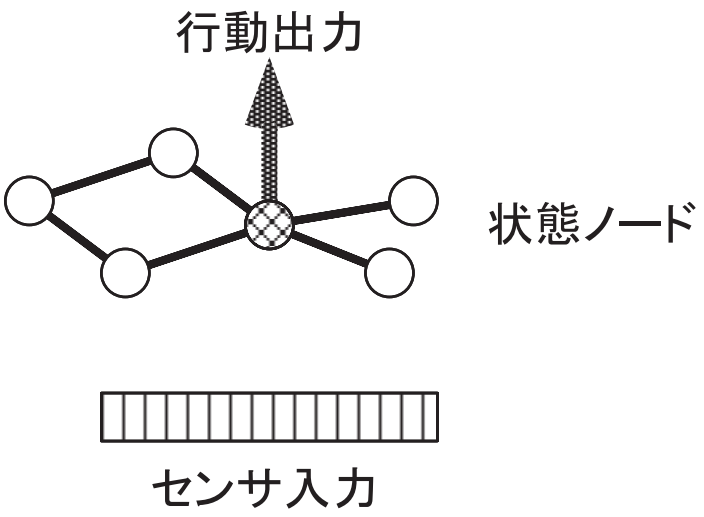


図. 2.14　状態ノードと行動出力

このような状態および行動の表現によって，センサ入力から行動出力を導出することが可能になった．しかし，このモデルでは状態同士の位相関係が利用されておらず，状態と行動とを単に離散化したにすぎない．次章ではこの方法の不充分な点を考察し，位相関係を利用し，連続値の行動出力を位相を利用して修正する方法を提案する．

2.5 おわりに

本章では，状態空間を生成するためのベクトル量子化アルゴリズムについて述べ，位相構造を保持して状態を表現し，状態・行動を生成する方法を提案した．

2.2 節では，本研究における状態空間・行動空間の定義を示し，評価信号のもとで状態と行動が互いを決定しあう関係にあることを述べた．本研究で扱う高次元のセンサ入力と低次元のアクチュエータ出力という問題設定を示し，同じ行動に対して評価が異なる状態を識別するという状態の生成のための基本的な考えを示した．

2.3 節では，Kohonen の自己組織化マップ (SOM) の基本原理を述べ，その位相保持性能に着目したアルゴリズムを概観した．基本 SOM は固定された位相構造とノード数という欠点を持つ．SCONN および TRN というアルゴリズムはこれらの欠点に対処するために提案されているが，本研究で必要とする状態空間の生成のためには TRN の位相構造の利用が有効であり，ノード数の変更には本研究に適した方法の提案が必要であることを示した．

2.4 節では，2.3 節でで示した位相構造を保持するアルゴリズムに基づき，ネットワーク状の位相構造を獲得する方法を示し，入力ベクトルの頻度ではなく評価信号を用いてノードを増加させることによる逐次的なネットワークの拡張方法を提案した．これにより，逐次的に与えられる 2 値の評価のもとでの状態および行動の表現が可能になった．

次章では，このようにして獲得された状態と行動の問題点を考察し，行動を修正する方法を提案する．

第3章　状態・行動生成のアルゴリズム

3.1 はじめに

本章では，2 章で述べた状態空間の生成方法に基づき，行動修正の方法を提案し，本研究での問題設定における状態と行動の自己組織化方法について述べる．

3.2 節では，本研究での問題設定の中で可能な行動の自己組織化方法の考え方について考える．

3.3 節では，3.2 節で述べた自己組織化の考えに基づき，行動修正を可能にする行動表現方法を述べる．

3.4 節では，3.2 節で述べた考え方に基づき，2 章で述べた状態空間の生成方法と行動の自己組織化アルゴリズムを組み合わせ，状態・行動を同時に自己組織するアルゴリズムを示す．

3.2 行動修正の考え方

2.4 節で述べた方法によって，離散化された状態とそれに対応する行動出力が獲得されたとする． このような問題設定において獲得された状態と行動は，ある一定の幅のある評価に基づいて生成されているため，その幅の中で可能な行動を生成することになる．つまり，生成された行動は最適である保証がない．

経路生成問題の例をあげると，幅を持った評価に基づいて**図.3.1** のような行動が生成されると，それぞれの境界で状態識別が起こり，状態ノードが生成される．しかし，仮に行動が境界に接触しないような形で生成されれば，これらの状態識別は不要になり，同時に行動の効率も改善される．

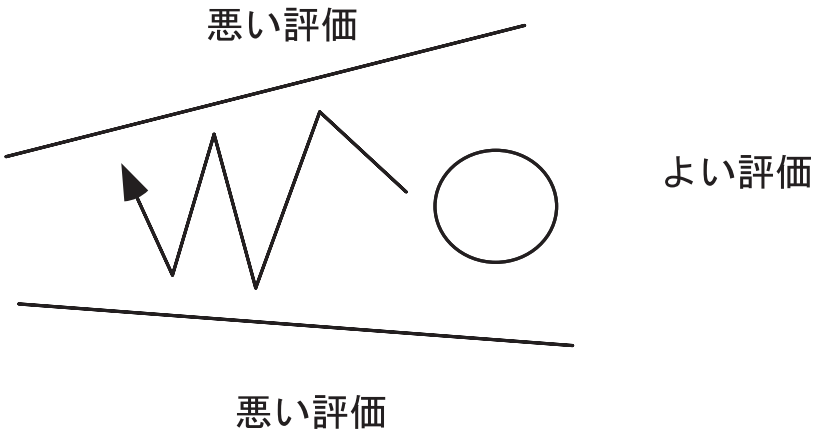


図. 3.1 ある幅を持って獲得された行動の例

2.2 節で述べたような本研究での基本的な考えにのっとると，同じ行動に対して評価が同じである限りは，その行動に対応する状態は識別しない．この考え方を用いると，すでに別のものとして分割された状態であっても，ある行動を対応させた結果同じ評価を得た場合は，同じものとみなすことが可能である．すなわち，すでに分割された状態に対して，**複数の状態に共通して良い評価を得る行動があれば，その行動によって状態が統合可能である**ということになる．ここで「統合」とは同じものとみなすという意味であり，提案モデルにおいて同じものとみなすことの具体的表現は 3.3 節で述べる．

2.4 節で述べた状態と行動の設定によって獲得された行動を修正する例を模式的

に**図.3.2** に示す．図の長方形領域は入力空間を表し，入力空間は線分によって状態に区切られる．各状態は行動出力を保持し，円内の矢印は保持している行動出力ベクトルを表している．**図.3.2** 左側に示すように，各状態が別々の行動を保持していたとしても，同図右側に示すような行動を試行して異なる状態同士でも同じ行動が良い評価を得れば，この行動を各状態に共通のものとして設定可能である．

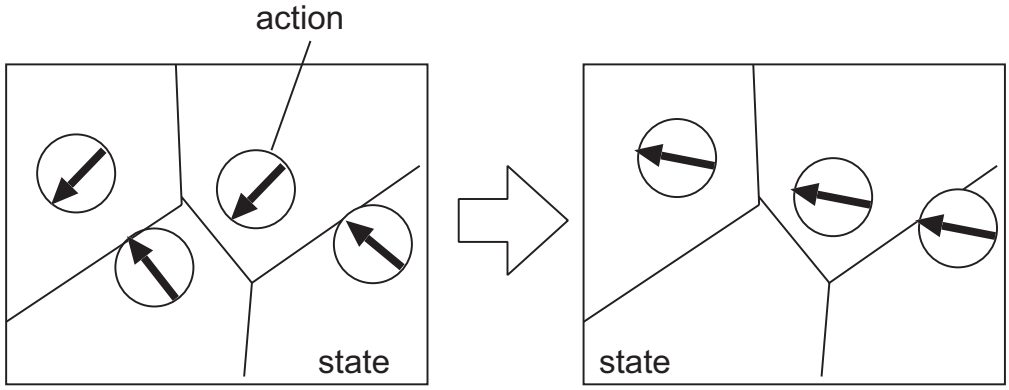


図. 3.2　複数の状態に共通する行動の探索

このような行動修正には，次のような利点がある．

- 行動を平滑化し出力の効率を改善する
- 位相近傍の状態同士を結びつける表現が可能になる

位相近傍の状態同士の結びつけは，識別不要な状態同士を統合し，状態表現の効率化を図るという考えにつながる．本研究では状態の統合は扱わないが，状態空間の生成に関する研究のほとんどが状態分割のみに着目しているのに対して，統合の考えは今後の重要な課題の一つであると考える．

このような行動修正が適用可能な例について述べる．**図.3.3** のような移動ロボットの経路生成などでは，壁にぶつからないという即時的な評価のみに基づいて行動が修正されれば，滑らかな行動が達成可能である．マニピュレータの障害物回避を含むリーチング動作などもこの例に当てはまる．また，本研究のシミュレーションおよび実験で扱う対象物に接触しつづけるという操作もこれに含まれる．

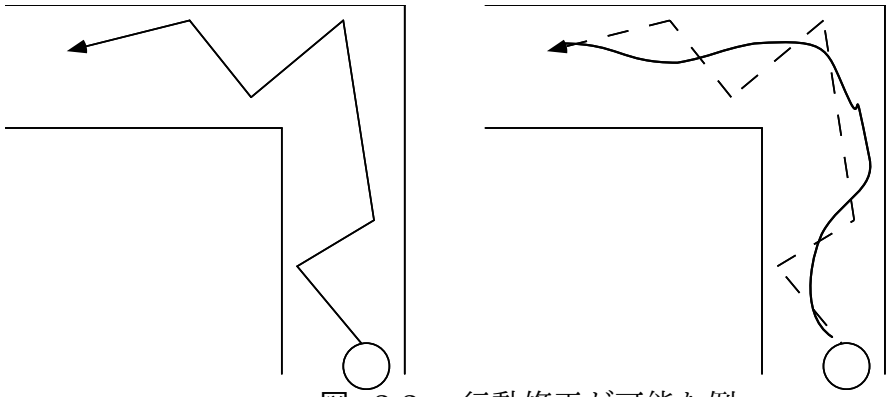


図. 3.3　行動修正が可能な例

逆に，移動ロボットのタスクとして設定される掃引作業のように，一定の行動ステップ毎に評価が判明するのでなく最終的な作業の達成状況によって評価が決まるような作業も存在する．このようなタスクはほかに領域探索，地図生成などが考えられる．このように，行動修正の考えはすべてのタスクに対して当てはまるわけではないが，比較的単純な行動に対しては適用可能であり，本研究のボトムアップな行動獲得の考え方に沿った意義を持っている．

3.3 行動の表現方法

各状態ノードは行動出力を保持しており，センサ入力に対して最整合となったノードの行動出力を出力するというのがこれまでに述べた枠組みである．2.3 節で述べたように，状態識別が充分に達成された上で行動の修正を行うためには，状態間の関係を位相関係に基づいて再構築しなければならない．最整合ノード以外に，その近傍ノードの行動出力の影響を与えるために Radial basis function[Fritzke94] を用いる．Radial basis function(以下 RBF と略す) は，ユークリッド距離に基づいて出力を決定する関数であり，本研究ににおいてはこれを位相近傍に存在するノード同士の出力の重ね合わせに用いる．

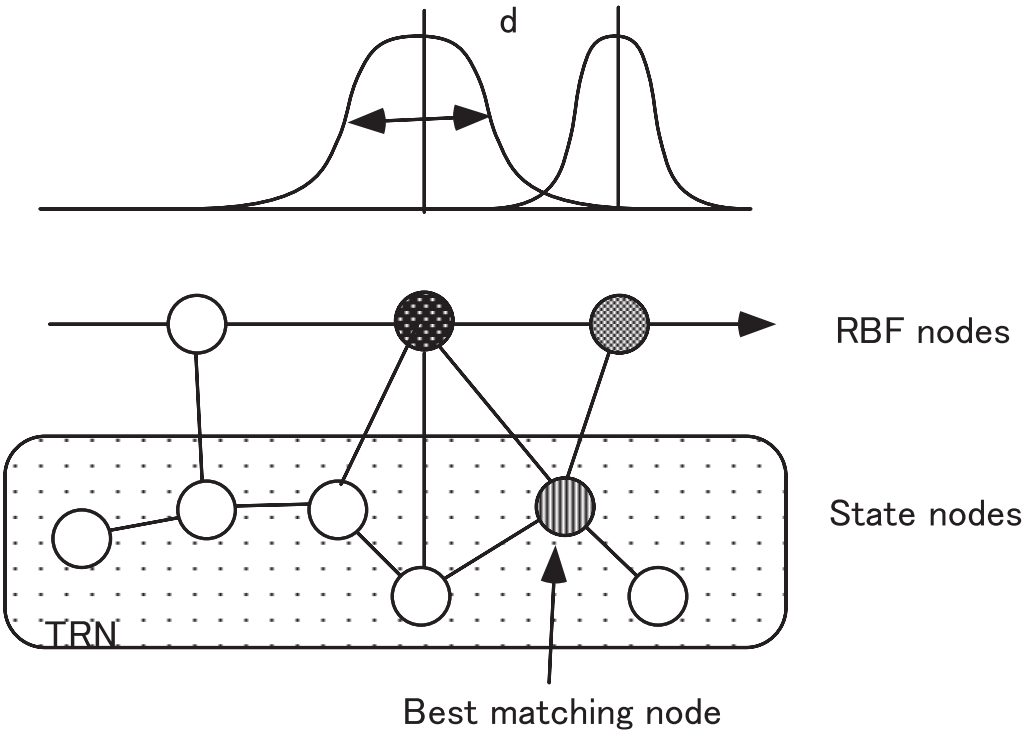


図. 3.4 RBF による出力の模式図

図.??に RBF による出力の模式図を示す．入力ベクトルとのユークリッド距離は，最整合ノードが最小であり，位相近傍のノードがそれについで小さい．この小ささは入力ベクトルに対する近さであり，これに基づいて近傍の影響も考慮するために RBF を用いる．図に示す d は近傍への影響の度合いを表す変数で，大きいほど位相近傍のノードに対して出力の影響を強く与える．数式上の表現を以下

に与える．

最整合ノードを bmu，ノード i の位相近傍を $\mathrm{Nh}(i)$ と表すと，現在の入力 $\boldsymbol{x}$ に対するノード i の影響の強さ

$$\boldsymbol{a}_i = \frac{\mathrm{rbf}_i(\boldsymbol{x})}{\sum_{j\in\mathrm{Nh(bmu)}} \mathrm{rbf}_j(\boldsymbol{x})} \tag{3.3.1}$$

が定義される．この影響の強さは RBF に基づき，

$$\mathrm{rbf}_i(\boldsymbol{x}) = \exp\left(\frac{-\parallel \boldsymbol{x}-\boldsymbol{w}_i \parallel}{d_i^2}\right) \forall i \in \mathrm{Nh(bmu)} \tag{3.3.2}$$

である． d_i はノード i の近傍への影響の強さを表す変数である．これを用いて，行動出力 $\boldsymbol{o}$ は

$$\boldsymbol{o} = \sum_{i\in\mathrm{Nh(bmu)}} \boldsymbol{a}_j \boldsymbol{o}_j \tag{3.3.3}$$

で表される．ただし $\boldsymbol{o}_i$ は それぞれのノード i に対応する行動出力である．

状態がその位相近傍において充分に識別されたと判断されたとき (この判断の方法については次節で述べる)，乱数による行動出力を行い，最整合の状態ノードの行動出力と無関係の行動を試行する．別の状態に遷移したときに良い評価を維持していれば，その行動は二つの状態に共通して良い評価を得るということになる．このような新たに発見された行動を RBF により近傍へ伝播させる．そのときの入力 $\boldsymbol{x}$ および行動 $\boldsymbol{o}$ を結合係数ベクトル，行動出力ベクトルとするノードを新規に生成し (これを RBF ノードと呼ぶ)，ネットワークに追加する．状態ノードと区別するのは，すでに得られている状態ノードの行動を保存し，悪い評価を得たときに元の状態に戻せるようにするためである．RBF ノード i は別の状態に遷移した瞬間の状態 $\boldsymbol{s}_i$ および行動出力 $\boldsymbol{o}_i$ を保持している．

2 章で述べた位相関係は，RBF ノードによる行動修正の伝播に利用される．生成された RBF ノードは，生成されたときに最整合だった状態ノードにリンクを張り，その状態ノードの位相近傍の状態ノードが最整合となったときにその RBF ノードの行動による試行を行う．そこで良い評価を得れば新たにその状態ノードとの間にもリンクを張り，さらにその位相近傍を探索する．このようにして行動の影響を拡大していくのに位相近傍を用いる．

次に，以上に述べた状態・行動の表現方法に基づき，逐次評価に基づいて状態・行動を生成するアルゴリズムについて述べる．

3.4 状態・行動空間生成のアルゴリズム

以上に述べた状態・行動の表現方法に基づき，逐次評価に基づいて状態・行動を生成するアルゴリズムについて述べる．状態・行動空間生成のモデルを**図.3.5** に示す．外部からの評価に基づいて行動を探索，修正するアルゴリズムを以下に示す．現行の最整合ノードをノード i とする．

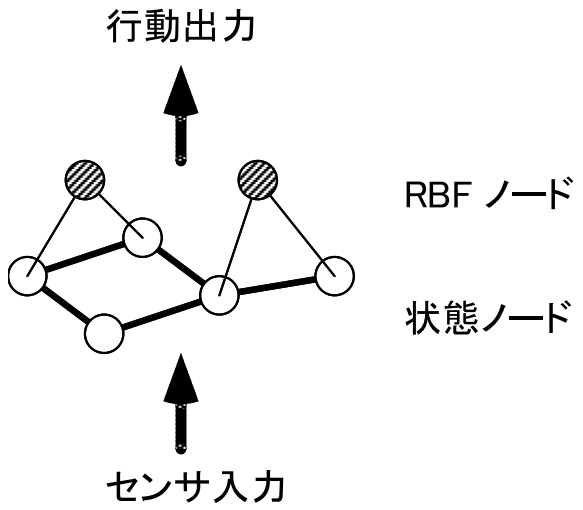


図. **3.5**　状態空間生成のモデル

(1) 良い評価を得たとき
st_i を減少させる

- 行動修正中のとき
  行動修正を行ったときの最整合ノードと別のノードが最整合であれば，現在の状態・行動に対して新規に RBF ノードを生成する
- 行動修正中でないとき
  RBF ノードの影響を更新する

(2) 悪い評価を得たとき
st_i を増加させ，位相近傍に伝播させる

- 行動修正中の時
  修正した行動出力を元に戻す
- 行動修正中でないとき
  ○位相近傍に RBF ノードが存在していなければ状態分割を行い，試行錯誤により行動を生成する
  ○位相近傍に RBF ノードが存在していれば RBF ノードの影響を更新する

乱数および RBF ノードによる試行を行うかどうかを決定するには各状態ノードに与えられた変数 st_i を用いる．これは各状態ノードに対応する状態の識別に関する不安定さを表す変数であり，状態分割が起こったときに増加させ，良い評価を維持しているときに減少させることによって状態分割の頻度を表現している．各ステップ毎に最整合ノードは $st_i < T_{st}$ ならば現在の出力に乱数を加えて行動修正を行う．ここで T_{st} は状態の安定さを判別するために設定する敷居値である．

(3.3.2) 式における d_i は RBF ノードの行動への寄与の大きさを表し，良い評価を得たときは増加させ，悪い評価を得たときは減少させる．その更新式は

$$\Delta d_i = \alpha \boldsymbol{a}_i \qquad (3.4.1)$$

で表される．α は学習定数であり，良い評価の時は $\alpha > 1$，悪い評価の時は $\alpha < 1$ とする．新規に生成した RBF ノードの影響によって悪い評価をえた場合にはその影響の度合いを減少させ，すでに獲得した状態ノードの行動出力に近づける．生成された状態に対し，ある程度その状態が安定であると判断されたとき，乱数に

より試行を行う (状態の安定の判別については前述)．別の状態に遷移したときに良い評価を得ていれば，そのときの状態および行動を用いて新たに RBF ノードをネットワークに追加する．RBF ノード i は別の状態に遷移した瞬間のセンサ入力値 $\boldsymbol{s}_i$ および行動出力 $\boldsymbol{o}_i$ を保持している．

RBF ノードは影響拡大 (d_i 増加) 中の状態とそうでない状態の二通りの状態を持つ．これは，RBF ノードの行動出力が状態ノードで分割された状態の一部に対して悪影響を与えているときにはそれ以上 その RBF ノードの影響を拡大させないためである．影響拡大中の RBF ノードの集合を R_a，影響を拡大しない状態の RBF ノードの集合を R_n　で表す．Nr はノード bmu とリンクで結ばれた RBF ノード　の集合を表す．行動出力は，最整合の状態ノード bmu とそのノードとリンクで結ばれた RBF ノード を用いて，

$$\boldsymbol{o} = \sum_{i \in \{\mathrm{Nr} \cap R_a\}} \boldsymbol{a}_i \boldsymbol{o}_i + (1 - A_a) \left(\sum_{j \in \{\mathrm{Nr} \cap R_n\}} \boldsymbol{a}_j \boldsymbol{o}_j + (1 - A_n) \boldsymbol{o}_{\mathrm{bmu}} \right) \qquad (3.4.2)$$

ただし

$$A_a = \sum_{i \in \{\mathrm{Nr} \cap R_a\}} \boldsymbol{a}_i, \qquad A_n = \sum_{j \in \{\mathrm{Nr} \cap R_n\}} \boldsymbol{a}_j \qquad (3.4.3)$$

によって得られるが，次式によって負の影響が出ることを防ぐ．

$$A = \begin{cases} 1 & \text{if } A \geq 1 \qquad (3.4.4) \\ A & \text{otherwise} \qquad (3.4.5) \end{cases}$$

(3.3.2) 式における d_i は，そのノードの行動出力対する影響の強さを決定するパラメータであり，影響拡大中の　RBF ノードは，よい評価を得たとき増加させる．影響を拡大しないノードは，よい評価を得ても d_i は変更せず，悪い評価を得たときのみ減少させる．これにより，すでに獲得した状態および行動を保存したまま，過度に細かく分割された状態同士をまとめる方向に行動を修正させることが可能になる．

3.5 おわりに

本章では，2 章で述べた状態空間の生成方法に基づき，本研究での状態と行動の自己組織化アルゴリズムを述べた．

3.2 節では，2.2 節で述べた「状態と行動が評価に基づき互いを決定する」という考えに基づき，異なる状態同士でも同じ良い評価を得るような行動が存在すれば，その行動を出力する限りはその状態は同じものとみなせるという考え方を示した．このような考え方が適用可能なタスクについて考察し，本研究で想定する多次元センサ入力，低次元アクチュエータ出力の問題設定の中でもそのようなタスクが存在することを示した．

3.3 節では，3.2 節で述べた考え方を行動表現に反映させる具体的方法を述べた．RBF(Radial basis function) を用いることによりノードの影響の強さを表現することが可能である．本研究ではこれをすでに生成された状態ノードとは別階層のノードという形でとして生成し，すでに獲得された状態および行動を壊すことなく逐次的に行動修正できる方法を提案した．

3.4 節では，3.3 節で述べた考え方に基づき，2 章で述べた状態空間の生成方法と本性で述べた行動の修正アルゴリズムを組み合わせた，状態分割・行動修正のアルゴリズムを説明した．

次章以降では，本提案アルゴリズムの有効性をシミュレーションおよび実験において検証する．

第4章　シミュレーション

4.1 はじめに

本章では，提案手法に対して 2 つのタスクを想定し，シミュレーションによる評価を行う．

4.2 節では，シミュレーションにおける問題設定，条件および評価方法を述べる．

4.3 節では，シミュレーション結果を示す．2 章で述べた基本的な状態と行動の生成アルゴリズムのみを用いる場合と 3 章で述べた RBF を用いて行動を修正する場合との比較を行い，提案手法の有効性を検討する．

4.2 シミュレーション方法

4.2.1 問題設定

図.4.1 に示すようなシステムを想定し，マニピュレータによる円形対象物の押し操作を行う．対象物は，手先とは静力学的な関係を保って運動する．システムは直上にカメラを持ち，手先，対象物，目標位置を示すマークを 2 値画像として捉える． 本研究では，視覚と最終的な動作を中心に扱うために，マニピュレータの運動学は既知とし (アクチュエータ出力空間と行動出力空間の射影は既得のものとし) ，2 次元平面上の出力を行動出力空間とする．

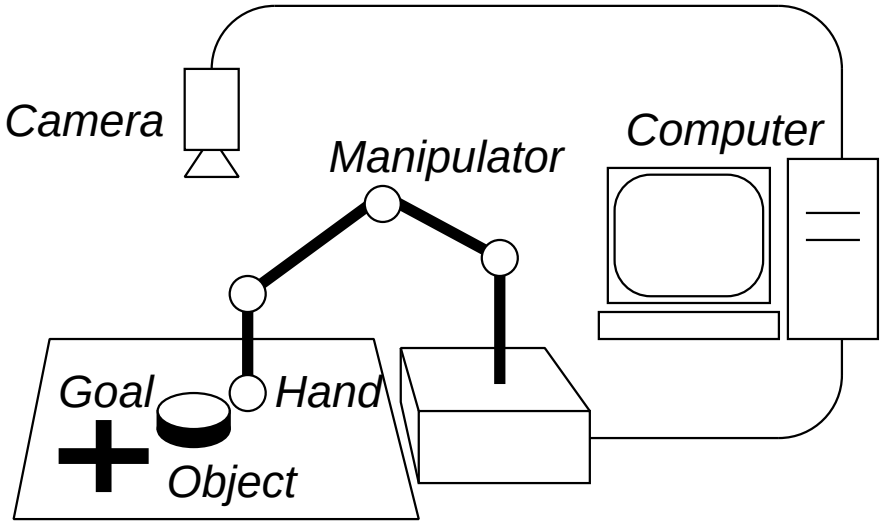


図. 4.1 問題設定

この問題設定は，視覚情報を直接用いることにより，入力画素数分の次元の入力空間から必要な情報を取り出すという意味をもつ．行動出力は 2 次元平面内の x 軸,y 軸方向で，ある一定範囲内のノルム一定以上の連続値を出力するものとする (**図.4.2**)．ノルム一定以上とするのは，行動出力が小さすぎるためにその行動の結果がセンサにより把握できないような事態を回避するためである．

マニピュレータによる押し操作を想定し，条件の設定を行う．入力画像にはマニピュレータの手先，対象物，目標位置を示すマークが映っており，しきい値処理により 30×30 の 2 値画像として入力を得る．対象物・目標・手先を画像入力に

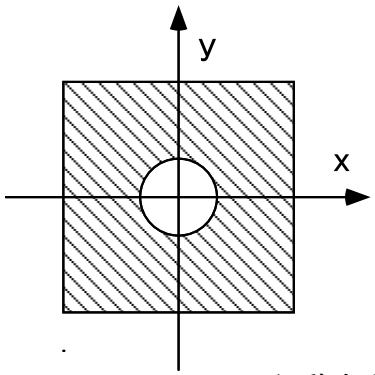


図. 4.2　行動出力空間

収め，2 値化する様子を**図.4.3**, **図.4.4** に示す．図左下十字のしるしが目標，その右上の円形が対象物，一番右側の円形が手先である．

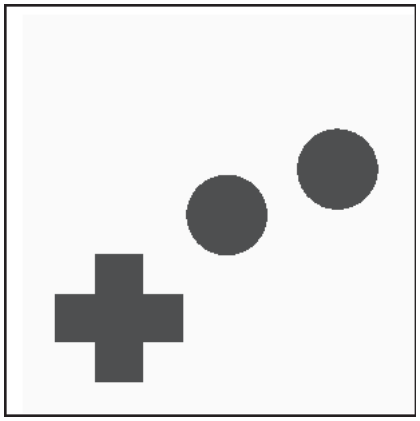


図. 4.3　対象物，手先，目標

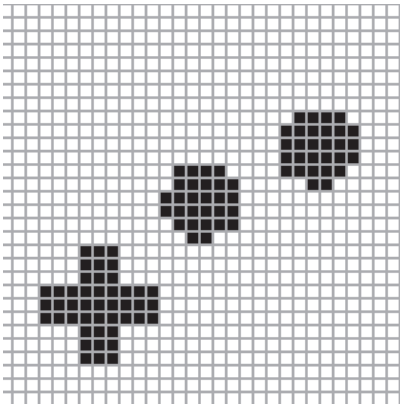


図. 4.4　2 値化された入力画像

シミュレーションには Sun Microsystems 社 Workstation(Ultra Sparc I 167MHz) を用いる．

目標に到達するかあるいは一定回数以上の行動ステップを繰り返すまでを一つの試行とし，試行を繰り返す．この問題設定で考えられる以下の 2 つのタスクを設定して評価を行う．提案する RBF を用いる手法と 2 章までの RBF を用いない手法を比較し，その効果を検討する．

(a) 手先が目標に近づくリーチングの動作

(b) 手先が対象物を目標に運ぶ動作

(a) はマニピュレータの手先と目標とを把握できるという問題設定においてもっとも容易なタスクであり，単純なタスクにおいて提案アルゴリズムの評価を行う．

(b) は実際に想定されるタスクであり，本手法が現段階で実世界に適用されるための方法とその問題点をこのタスクにおいて評価，議論する．状態の生成が行動とその評価に即して行われているかどうかを検証するために，センサ入力の次元を変化させ (画素数を変化させ)，センサ入力次元と状態数の関係を論じる．

4.2.2 運動モデルの記述

手先と接触したときの対象物の運動は準静的に起こるものとする．つまり，衝突により対象物に初速度が発生し，摩擦によりすべり運動が停止するという動的なモデルでなく，手先と対象物の間に干渉が生じるとき (手先と対象物が接触するとき) に対象物の位置変化に応じて対象物位置が変化する，という記述方法をとる．これは，実機環境における対象物の挙動にならったものである．また，シミュレーションにおける運動モデルと実機対象物の運動との間のずれの影響については次章で論じる．

図.4.5 に示すように，手先の微小変位の結果干渉が生じたとき，その幅を d として

$$\boldsymbol{v} = d\frac{\boldsymbol{v}_0}{|\boldsymbol{v}_0|} \tag{4.2.1}$$

なる $\boldsymbol{v}$ により手先の変位を決定する．

4.2.3 評価信号の生成方法

先に述べた 3 通りのタスクに対する評価信号の生成方法を示す．評価信号は，行動の学習を行う上で以下の要件を満たさなくてはならない．

(1) 無矛盾であること

(2) すべての状態に対し，良い評価を得られる行動が存在すること

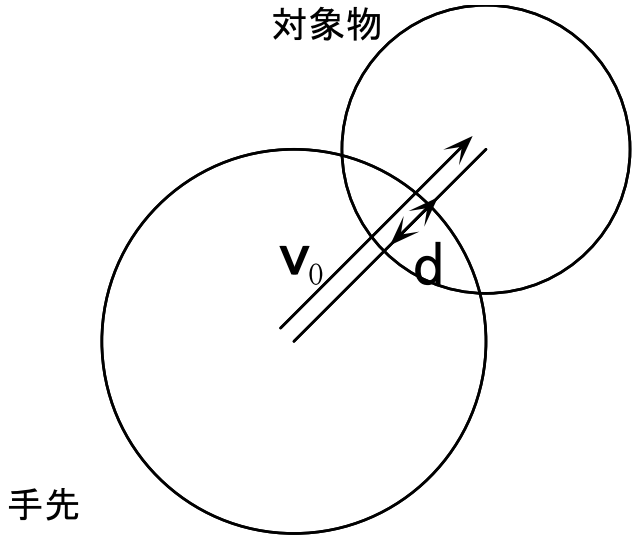


図. 4.5　対象物の運動モデル

(1) は，同じ状態に対して良い評価が与えられたり悪い評価が与えられたりすることがないということである．このような評価が与えられると，完全に同じ結合係数ベクトルを持つノードが複数生成される可能性がある．(2) は，どのような行動をとっても良い評価が得らないと，その時点での適切な行動が発見されず，ノードが生成できないためである．これは，遅延報酬を用いない本手法の限界である．

(a) 手先が目標に近づくリーチング動作

手先が目標に近づいた場合は良い評価，遠ざかった場合は悪い評価を与える．すなわち，手先の位置 $\boldsymbol{p}_h = (x_h, y_h)$，手先の移動量 $\boldsymbol{d} = (\Delta x, \Delta y)$ 目標の位置 $\boldsymbol{p}_g = (x_g, y_g)$ に対し，

$$\mathrm{E} = \begin{cases} 1 & \text{for } \| \boldsymbol{p}_h - \boldsymbol{p}_g \| - \| \boldsymbol{p}_h + \boldsymbol{d} - \boldsymbol{p}_g \| \geq 0 \quad (4.2.2) \\ -1 & \text{for } \| \boldsymbol{p}_h - \boldsymbol{p}_g \| - \| \boldsymbol{p}_h + \boldsymbol{d} - \boldsymbol{p}_g \| < 0 \quad (4.2.3) \end{cases}$$

により与える．

(b) 手先が対象物を目標に運ぶ動作

図.4.6 に示すように，対象物中心と目標とを結ぶ直線および対象物中心と手先中心とを結ぶ直線のなす角を θ とし，行動出力の結果としての θ の変化量を $\Delta\theta =$

$\theta(t+1) - \theta(t)$ とする．この θ および $\Delta\theta$ を用い，以下のような評価を与える．

- 手先と対象物が接触しているとき (干渉があるとき)
  - $\theta \leq \varepsilon$ ならば $E = 1$
  - $\theta \geq \varepsilon$ のとき
    - $\Delta\theta \leq 0$ ならば $E = 1$
    - $\Delta\theta \geq 0$ ならば $E = -1$
- 手先と対象物が接触していないとき
  $E = -1$

すなわち，手先と対象物が接触しないときは無条件に悪い評価で，接触しているときは，手先が目標から見た対象物の後方の位置を保っている限りは良い評価を与える．後方の範囲から外れている場合は，その範囲に入る方向に動いているときは良い評価を与え，そうでないときに悪い評価を与える．$\Delta\theta$ を評価に用いるのは，対象物に接触しかつ対象物の後方に位置するということが不可能な状況が存在するためである．先に述べたようにどのような行動をとっても良い評価が得られないような状況を回避するために導入されている．これにより，評価関数の定義域として，対象物と手先が接触するすべての状態を設定することが可能である．

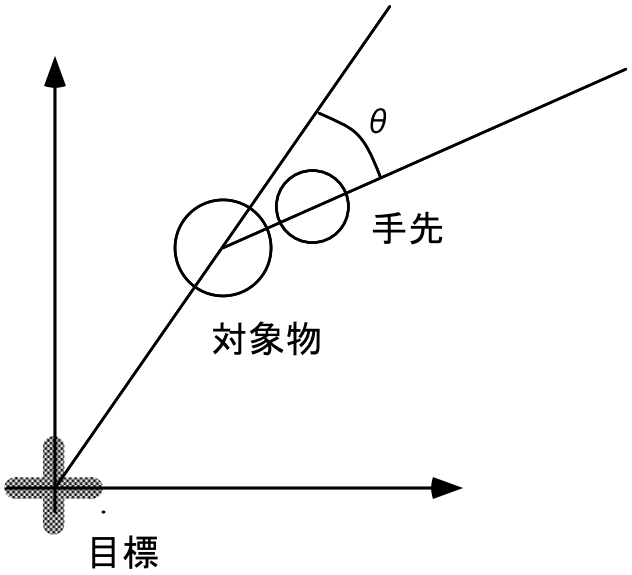


図. 4.6　押し操作における評価方法

対象物と手先が離れている状態から開始する場合の対象物の押し操作のためには，対象物の後方に向かうような行動に対して良い評価を与えるという評価方法を付加しなくてはならない．より厳密な評価関数を設定し，それに基づいた 2 値の評価を生成することも可能であり，その評価方法に基づく動作生成も達成されている [小林 99] が，本研究では，行動の結果から評価を生成する方法によって厳密な評価関数を設定しなくてもタスクの達成が可能であることを示す．

評価信号の生成方法に関する考察

対象物を目標まで押して運ぶ動作には，厳密に考えると対象物を目標に向かって押す動作と，対象物の後方に回り込む動作が含まれている．本研究のタスク (b) においては，対象物と手先が接触している初期条件から開始するために，回り込む動作そのものの (対象物を回避するなどの) 厳密な評価を導入していない．

対象物後方に回り込む動作をより明確に獲得させるためには，目標に向かって押すための評価と，対象物の後方に回り込むための評価を足し合わせる形で評価を与えなくてはならない．

ロボットの学習といえば経路探索や動作の獲得など一つの入出力の写像関係を獲得するものが多い [宮崎 95] が，実世界で動くロボットにとっては，複数の行動方針の重ね合わせの形で行動を決定しなくてはならないことも多い．本研究における押し操作についてもこのことは当てはまり，回り込む動作と押す操作は質的に異なるものである．最終的に一つの評価基準に統合したものを評価信号として利用するよりは，このような評価基準の統合の問題を含めた学習を扱うことは今後の重要な課題になると考えられる．

本研究ではこのような複数の評価基準を別々に扱うことなく，最終的な一つの評価基準として用い，シミュレーションおよび実験を行う．円形対象物の押し操作において，このようにして与えた評価により獲得した動作の実世界での有効性については，実験により検証することとする．

4.3 結果

(a) 手先が目標に到達するタスク
目標に向かう行動が生成されるようすを**図.4.7**, **図.4.8** にしめす．一回目の試行**図.4.7** と比較して，数回の試行で行動が適切な方向に修正され，5,6 回目の試行からは同じような行動の繰り返しになった．

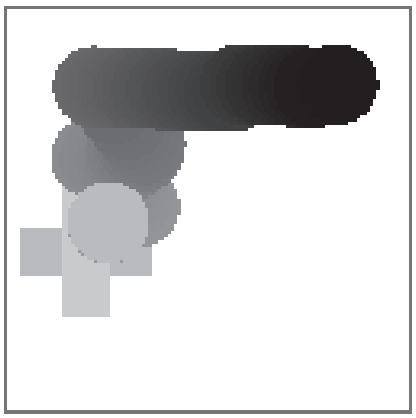


図. 4.7 1 回目の試行における手先の軌跡

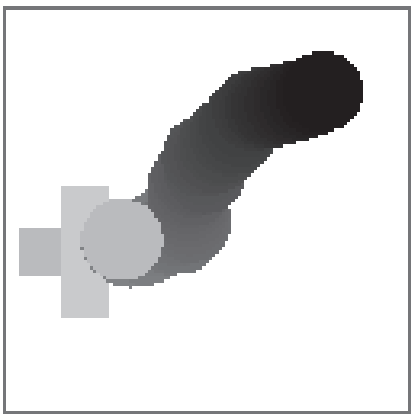


図. 4.8 10 回目の試行における手先の軌跡

行動修正を行わない場合はは，数回の試行で目標に到達できる状態と行動を獲得すると，それ以上の行動の変化はなく，手先の移動距離の総和は変化しない．一方，RBF ノードを用いる場合は，目標に到達できる行動が生成された後も行動の試行がなされ，RBF ノード数が増加するにしたがって行動が改善され，手先の移動距離も減少する．状態ノード数は，RBF あり，なしの場合ともに 10 前後で違いはなかった．

(b) 対象物を目標まで運ぶタスク
手先の初期位置を固定して与え，対象物を運ぶ操作を繰り返した時の試行回数とノード数の関係を**図.4.11** に，試行回数と手先の移動距離の関係を**図.4.12** にしめす．手先移動のタスクと同様，RBF ノードを利用する場合はしない場合と比較して手先の移動距離が減少することによって行動が適切な方向に修正されていることが確認された．

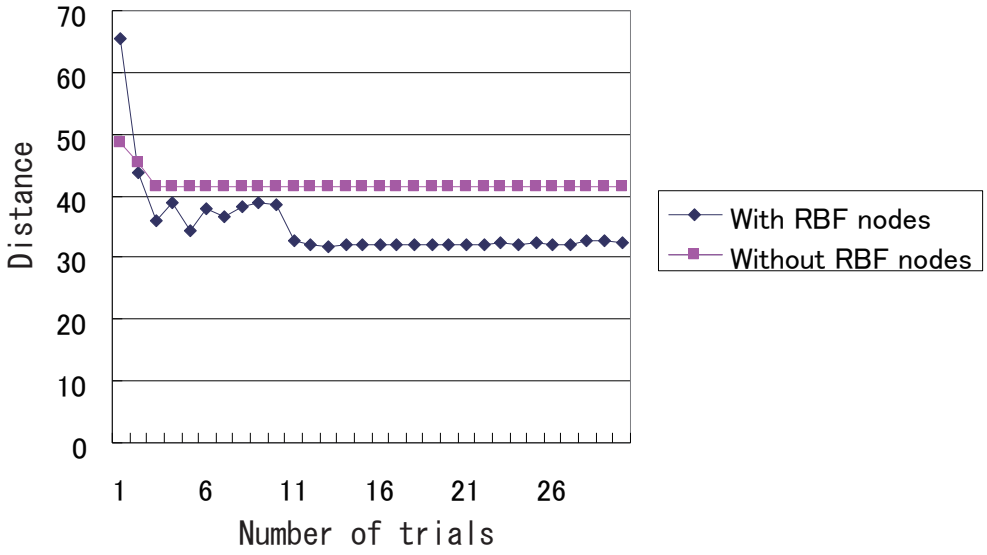


図. 4.9　試行回数と手先の移動距離の関係

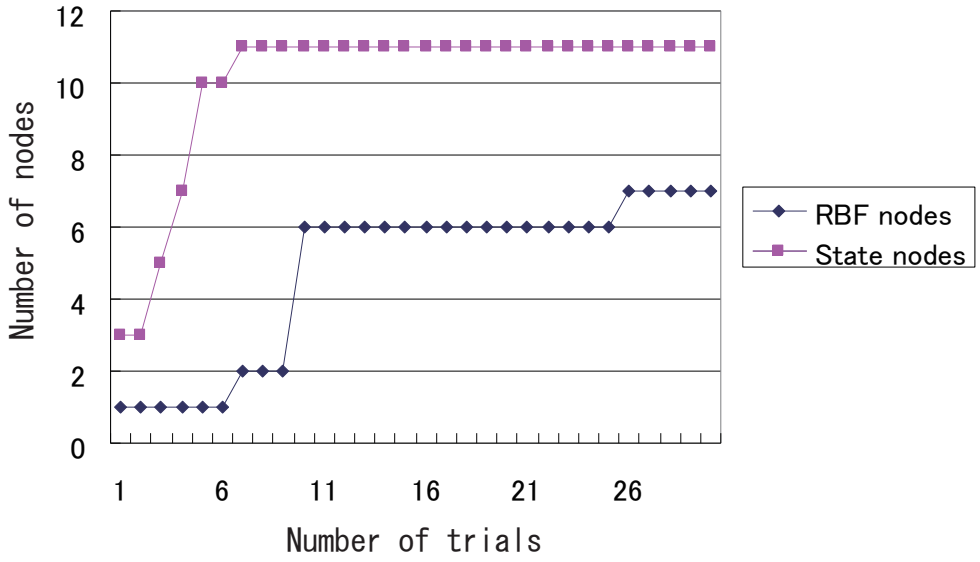


図. 4.10　試行回数とノード数の関係

また，行動出力の変化量の総和はRBF ノードを利用する場合はしない場合の 20 から 30% 程度であり，行動の平滑化が達成できていることがわかる．また，RBF ノードを使用する場合としない場合とでは状態ノード数はともに 50 個前後で大き

な違いはなかった.

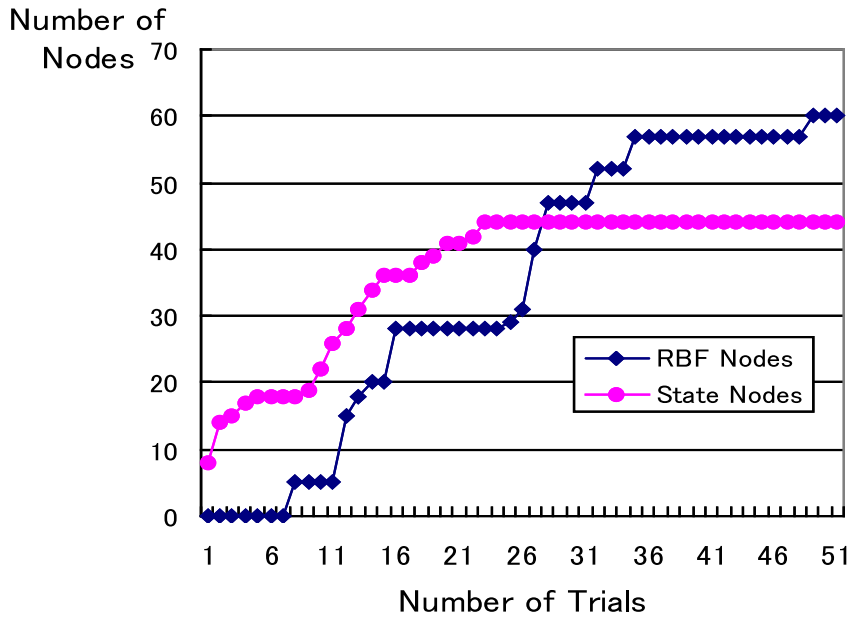


図. 4.11　試行回数とノード数の関係

図.4.13 に画素数と試行回数 50 回後の状態ノード数の関係を示す．画素数は正方形の一辺の格子数であり，状態ノード数は 5 回の平均値である．画素が一定数より少ないと行動学習は適切に行われず状態分割も過剰に起こる．これは本来識別すべき状態が同一のセンサ入力として認識され，同一のセンサ入力に対応する状態ノードが複数生成されてしまうためである．一定数以上では，画素数によらず一定の状態ノード数となる．これにより，入力次元に依存せず行動とその評価に即した適切な状態表現が獲得されていることが確認された.

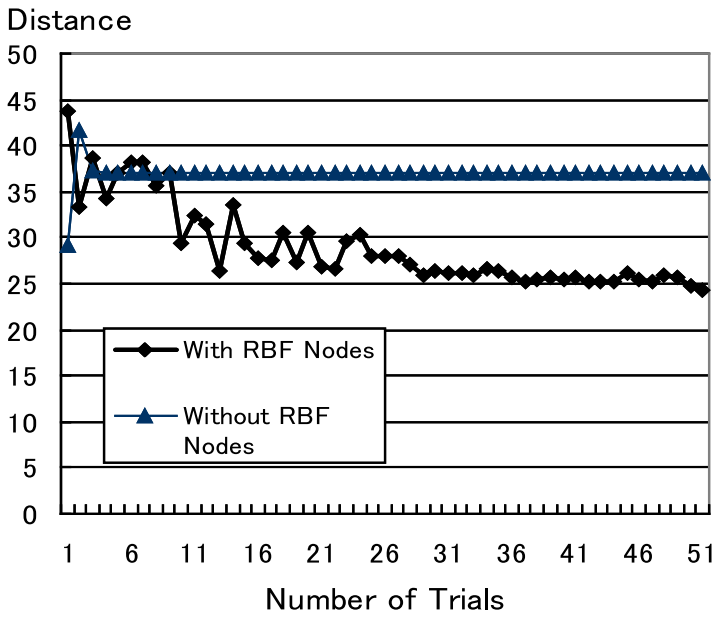


図. 4.12　試行回数と手先移動距離の関係

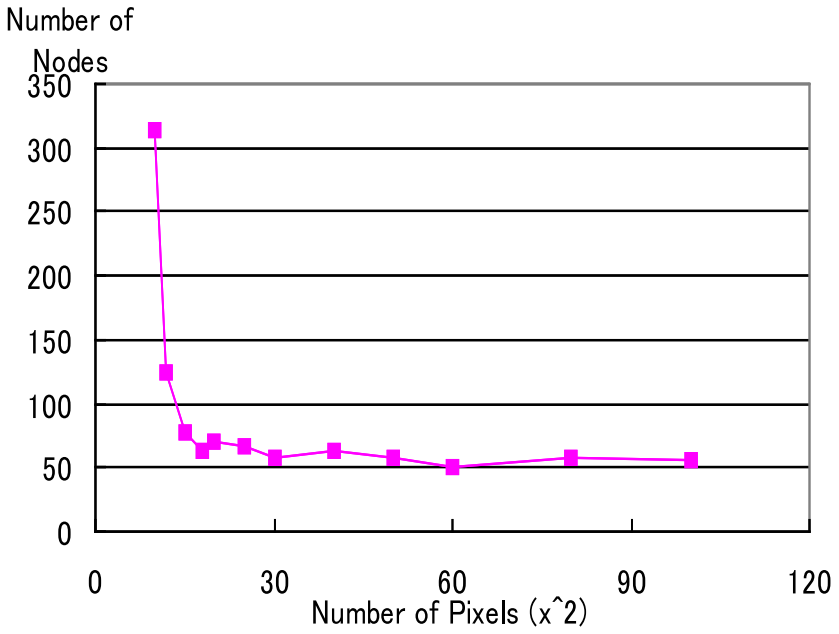


図. 4.13　入力画素数と生成された状態ノード数の関係

4.4 おわりに

本章では，提案手法に対して 2 つのタスクを想定し，シミュレーションによる評価を行った．

4.2 節では，シミュレーションにおける問題設定を述べ，マニピュレータによる円形対象物の押し操作を枠組みとして定めた．具体的なタスクとして手先の目標へのリーチング動作と対象物の押し操作を設定し，その評価方法を説明した． 対象物押し操作において 2 値の評価信号を生成するために，手先と対象物の接触情報と手先と対象物の位置関係を利用する方法を説明した．

4.3 節では，RBF を用いた行動修正アルゴリズムによって，目標へのリーチング動作において手先の軌跡が平滑化され，行動が適切な方向に修正されることを確認した．対象物の押し操作においても同様に RBF を用いる場合と用いない場合とを比較し，RBF ノードが，状態ノードの数に影響を与えずに行動の効率を向上させることが確認された．

また，多次元入力に対する適切な状態の生成という本提案手法の有効性を検証するために，入力画像の画素数を変化させたときの生成される状態ノード数を調べた．ある一定以上の画素数では生成される状態ノード数は変化せず，本提案手法が多次元の入力に対して適切な状態生成が可能であるということを確認した．

5 章では実験を行い，本章で行った押し操作の学習結果を用いた実験，ならびに本章で評価を与えるのに用いた「接触情報」に基づいてオンラインで学習を行う実験を行う．

第5章　実験

5.1 はじめに

本章では，シミュレーション結果を用いたオフライン学習実験とセンサ情報を用いたオンライン学習実験を行い，それぞれの立場から実世界における学習の問題点について考察する．

5.2 節では，4 章の議論を踏まえ，本研究における実験の位置付け，目的を二通りの立場から述べる．

5.3 節では，マニピュレータ，CCD カメラ，力覚センサなどからなる実験装置を示し，力覚センサから接触情報を得る実験方法について述べる．

5.4 節では，オフライン実験，オンライン実験それぞれの実験結果を示す．

5.5 節では，実験結果に対する考察を行い，提案手法の現状での限界，問題点などについて論じる．

5.2 目的

4 章において，本研究の目的である状態空間と行動空間の自律的生成を行う提案手法の検証を行った．シミュレーションは設計者がタスクに必要な評価方法を設計し，ロボット自身のセンサ，アクチュエータ情報によって再構成するという立場で行われた．本研究における実験の目的は，実世界でのロボットの挙動を 2 つの立場から論じることである．

1 つは，シミュレーションで獲得した状態と行動とを用い，同じ条件の実機においてその挙動を検証することである．実機の環境と同じ条件をシミュレーションにおいて設定し，シミュレーション結果を用いて実機での対象物の押し操作を行う．この実験の目的は，本研究における問題設定において，モデルのずれがどの程度存在し，どのような影響を与えるかということについて検討することである．モデルのずれとは，シミュレーションで仮定した押し操作における対象物の運動モデル，ならびにシミュレーションで仮定したセンサ入力情報と実機の CCD カメラから得た画像情報のずれのことである．

もう 1 つは，実機においてオンラインで状態と行動との獲得を行うことである．手先の接触情報を評価信号として用い，接触を維持するような状態と行動を生成する実験を行う．この実験は設計者の評価を介さずロボット自身が自らの情報を利用して行動を利用するという意味を持つ．実世界で状態と行動をオンラインで生成する問題を検証し，このような方法の利用可能性を検討する．

5.3 方法

図.5.1 に実験装置の概要を示す．マニピュレータは川崎重工社製 Js-2，CPU は Pentium 200MHz で CCD カメラからの画像を利用してマニピュレータのコントローラと PC はシリアルおよび VME バスにより接続されている．PC から VME バスを介してコントローラの共有メモリに指令を送ることにより，16[msec] 毎に指令を変更することが可能である．CCD カメラは 34[msec] で画像を取り込むことが可能である．

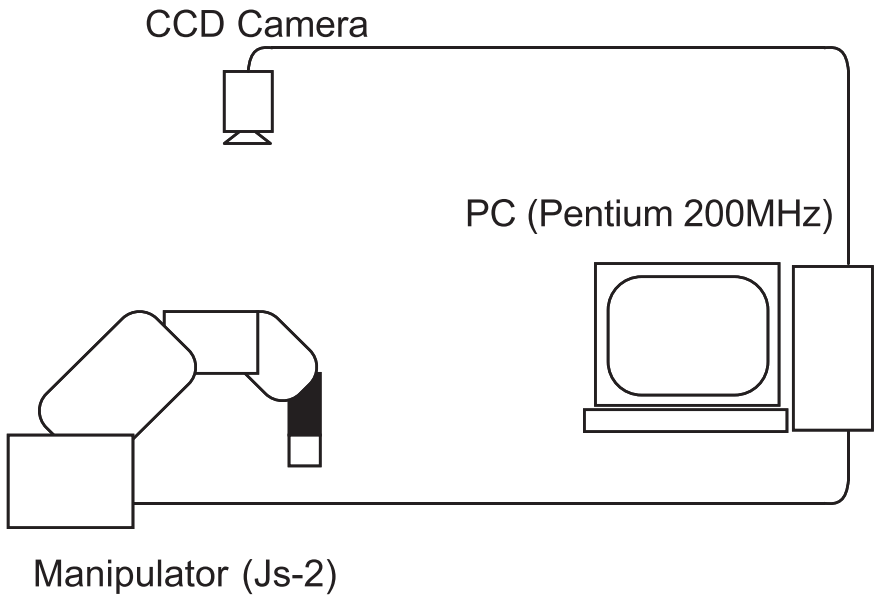


図. 5.1 実験装置の概要

図.5.2 に CCD カメラからとらえたマニピュレータ手先と対象物，目標の様子を示す．

実験は以下の二通りの方法で行った．

(1) オフラインでのシミュレーション結果を用いた押し操作

(2) 手先に力覚センサを取り付けたオンラインでの押し操作

後者の実験では，手先に力覚センサを取り付け，そのセンサ情報を評価に用いた．力覚センサは 6 軸の情報を持つが，そのうち 1 軸の並進とそれに垂直な 1 軸の回転

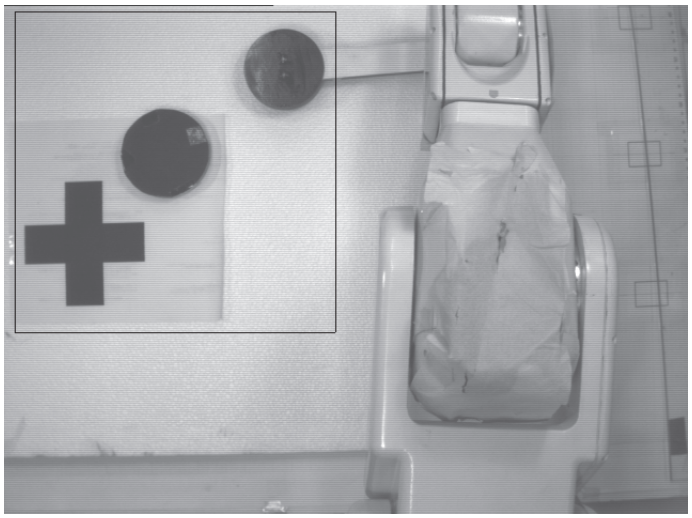


図. 5.2　カメラから見たマニピュレータ手先と対象物，目標

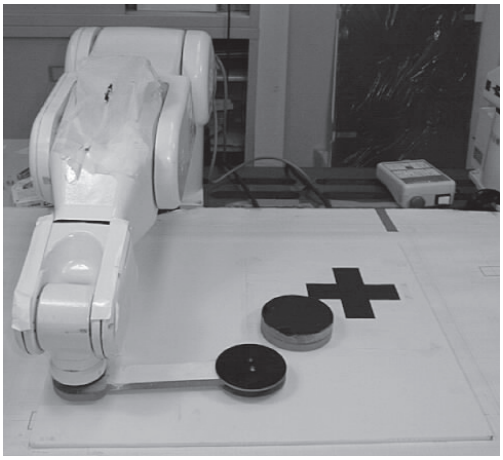


図. 5.3　押し操作の初期状態

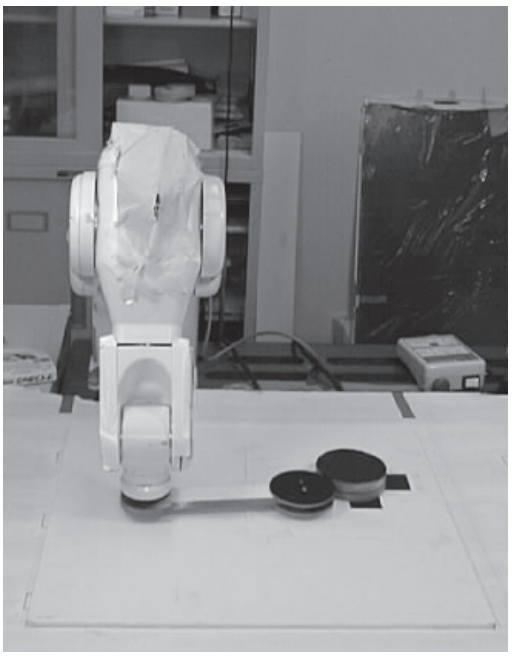


図. 5.4　操作終了時の状態

モーメント情報を用いる．取り付けられた手先の方向の力，作業する平面に垂直な軸の回転方向のモーメントが敷居値以上のときに良い評価を与えるという形で利用する (**図.5.5**)．これによって対象物に接触しつづける状態・行動をオンラインで獲得する．

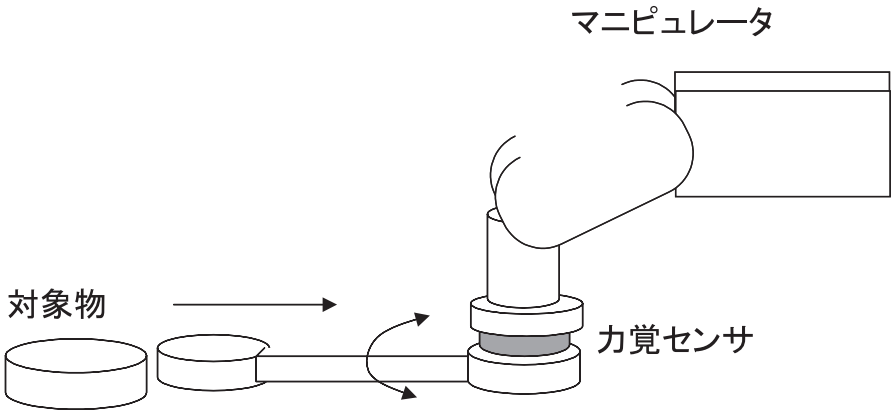


図. 5.5　力覚センサを用いた評価方法

5.4 実験結果

5.4.1 オフライン学習

30×30 画素でシミュレーション 50 回の試行を行った結果を用いて実機で押し操作実験を行った結果のマニピュレータ手先の軌跡を**図.5.7** に示す．図の右上が手先の初期位置であり，対象物の後方の位置を保ちながら目標の位置 10[mm] 以内まで運ぶことができた．

シミュレーション環境と実機環境の運動モデルのずれの影響を検討するため，対象物を**図.5.6** に示すような方法で重心を変え，同様の初期条件による押し操作を行った．押し操作の結果を**図.5.8** に示す．対象物の重心をグラフの向きで左側に偏らせたため，偏心なしの操作時と比較して，対象物は左側にずれる．そのずれに対応する形で手先は左側から対象物の後方に回り込む動きをしていることがわかる．

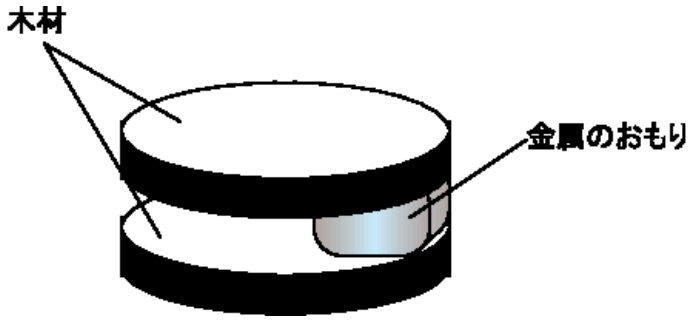


図. 5.6　対象物の重心変更

また，シミュレーションで 300×300 画素で学習を行い，その結果を利用した押し操作実験を行った．30×30 画素での実験結果とほぼ同様の押し操作が達成された．シミュレーションでの試行 50 回に，30×30 画素では 24(sec)，300×300 画素では 1270(sec) を要した．実機での押し操作では，1 回の試行に，30×30 画素では 45(sec)，300×300 画素では 135(sec) を要した．

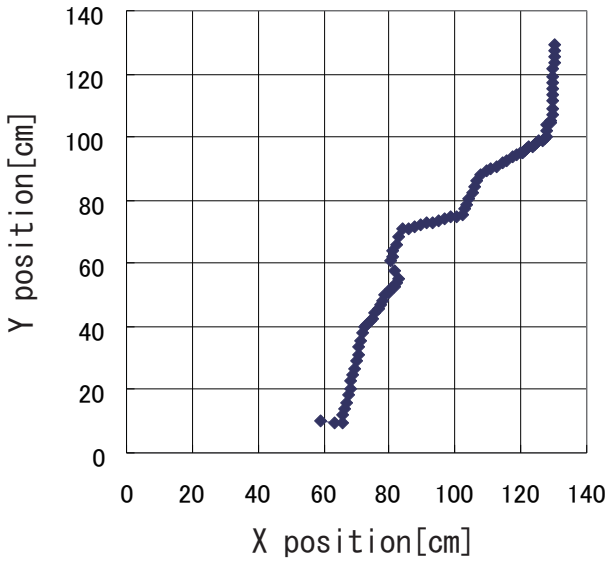


図. 5.7　手先の軌跡

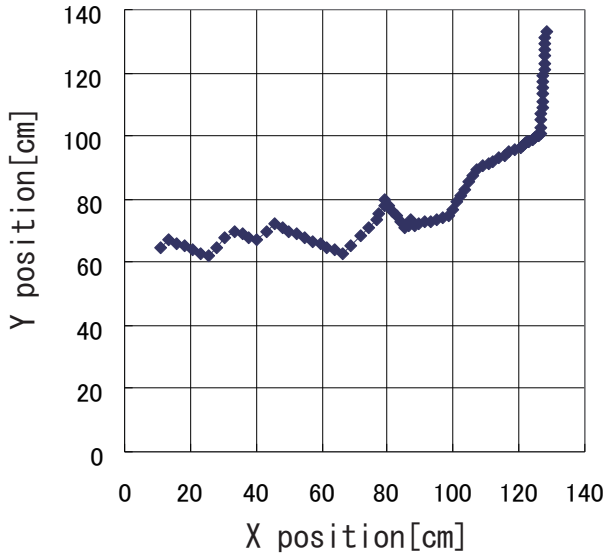


図. 5.8　手先の軌跡

5.4.2　オンライン学習

接触情報を用い，オンラインで接触を維持するように状態，行動を生成する実験を行った．1 回の試行の終了条件は，手先が初期位置から見て 150[mm] 離れた

位置までくることである．

図.5.9 に試行回数と行動ステップ数の関係を示す．最初は押す操作を十分に維持することができず，一定距離移動するのに比較的多くのステップ数を要するが，6 回目以降は比較的少ないステップ数で操作が終了しているということがわかる．5 回目の試行で RBF ノードが 2 つ生成され，この影響が出ているものと考えられる．**図.5.10** に試行回数と状態ノード数の関係を示す．6 回目以降はほぼ同様の行動が繰り返されるが，乱数による行動を試行する影響のために毎回多少軌跡は変わり，その都度状態ノードは増加する．

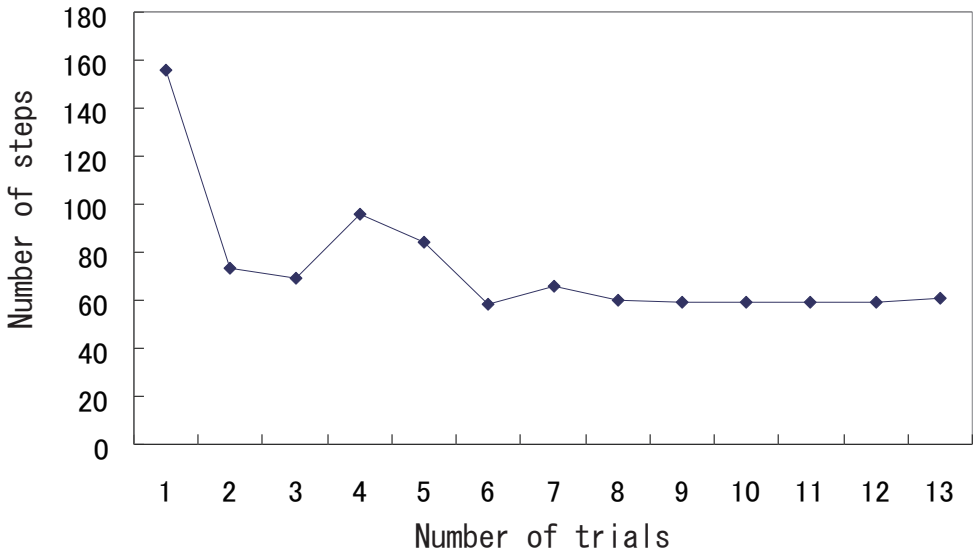


図. 5.9　試行回数と行動ステップ数の関係

図.5.11 に 1 回目の試行における押し操作の手先の軌跡，**図.5.12** に 10 回目の試行における軌跡を示す．オフライン実験と同様図右上が手先の初期位置である．1 回目の試行では良い評価を維持できずに行動が大きく変化しているが，10 回目の試行では対象物から大きくずれることなく，行動の小さい変化で対象物を押しつづけることができた．

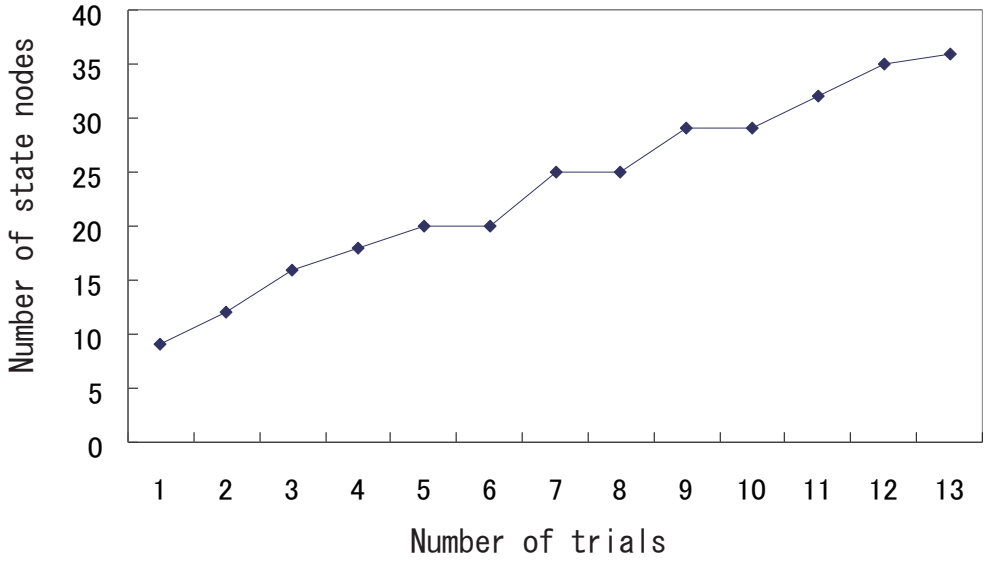


図. 5.10　試行回数と状態ノード数の関係

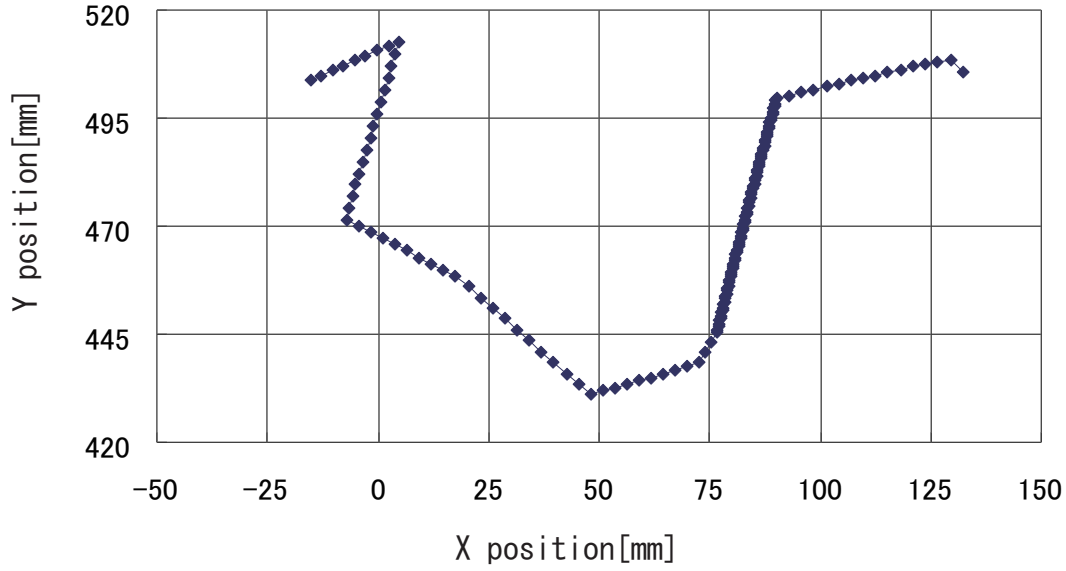


図. 5.11　1 回目の押し操作の軌跡

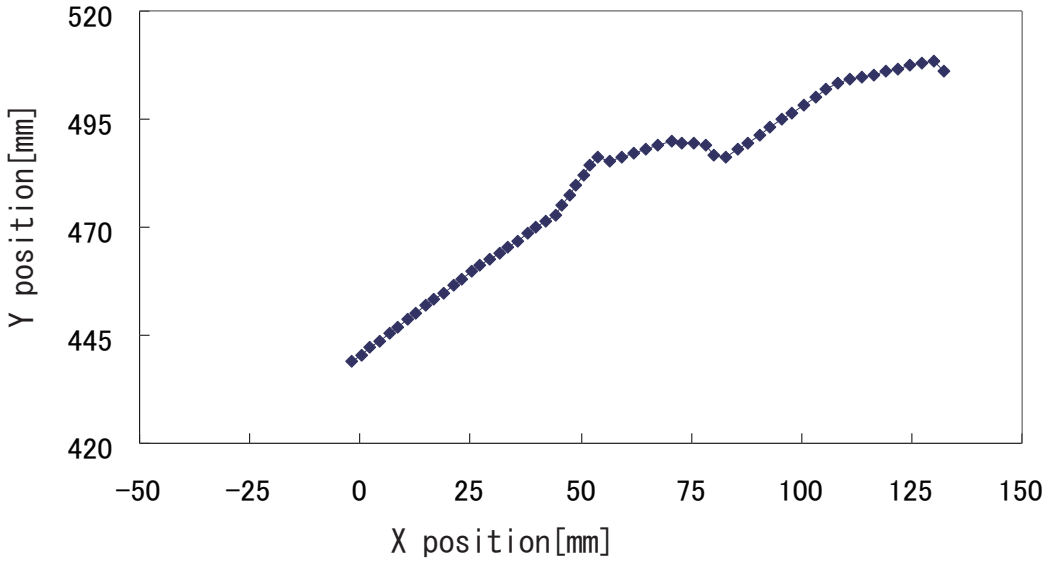


図. 5.12　10 回目の押し操作の軌跡

5.5 考察

シミュレーション結果を用いた押し操作実験については，運動モデルのずれを吸収するような操作が実現できるということが確認された．これは，学習手法が運動モデル誤差を吸収可能な方法であるというのではなく，シミュレーションで与えた評価の方法が，対象物の運動モデルのずれに対処できるものであったためである．すなわち，目標に対して対象物の後方に回り込むという評価と，対象物後方から目標に向かって動くという評価の二つを重ね合わせるということが，対象物の動きの変化に対してロバストな評価方法であるということができる．

また，シミュレーションで仮定したセンサ入力情報と実機の CCD カメラの 2 値化した画像入力の間のずれの影響は実際に押し操作を行うためには充分に小さいものであった．シミュレーションで用いた結果が充分に実環境で利用可能であるということがいえる．このことから，本手法によって，シミュレーションで実画像での見え方を再現できるような問題であれば，実画像での画像処理を経ずに直接画像情報から必要な動作指令を生成することが可能であるということができる．より一般的な表現を用いれば，センサ入力の予測可能な問題であれば，実機におけるセンサ入力からの特徴抽出などの過程を経ずに，シミュレーションの結果を利用可能であるということである．

また，力覚センサを用いたオンライン実験では，対象物との接触を維持するという評価指針に基づいてで状態と行動とを生成することが可能であることを示した．状態ノードが生成され続けるのは，乱数による試行を行うことによって新たな状態に移行し，新たに学習がなされるためである．手先と対象物の相対的位置関係が同じような場合を同じ状態として認識するためには，視覚入力を直接用いるだけでなく，部分的な注視領域を設定することによって相対的な位置関係が同じであることを認識できる機構を備えることが必要であると考えられる．

本実験で用いた力覚センサは 6 軸の情報を与えるものであり，このロボットの身体性に基づくという観点からは，それぞれのセンサ情報と状態・行動との関係を獲得することが望ましい．これはセンサ情報と行動の関係を獲得するという意味からも重要な意味を持っており，今後の重要な課題の一つであるということができる．

オンラインの押し操作実験の結果対象物に接触しつづける状態と行動とを獲得可能であることを確認したが，このようにして獲得した状態と行動を別の学習に適用することを考えると，行動は接触を持続するだけでなく接触するかしないかという境目に対して厳密な区別を行うように生成するということも考えられる．対象物操作の例では，対象物を押しつづけるだけでなく，回り込む，回避するなどの動作を考えると，評価信号の生成方法には拡張の余地が残されている．このような観点から議論を広げることは今後の重要な課題である．

5.6 おわりに

本章では，4 章で行ったシミュレーションの結果を用いた押し操作実験，ならびに手先に取り付けられた力覚センサから接触情報を得た接触情報を評価信号とするオンラインの実験を行い，本提案手法の実世界への適用に対する問題点を考察した．

5.2 節では，本研究における実験の位置付けを述べ，実験を行う二通りの立場を示した．1 つはシミュレーションで獲得した状態・行動の関係を直接用いて実機における押し操作に適用する立場であり，もう 1 つは実機のセンサ情報を用いたオンラインの学習に本提案手法を適用することである．

5.3 節では，実験装置，実験方法を示した．マニピュレータ，CCD カメラ，PC からなる実験システムを示し，対象物，手先などの様子を示した．センサ情報を用いた実験の方法として，力覚センサの情報を用いて対象物と手先の接触を検知する方法について述べた．

円形対象物の押し操作実験により，シミュレーション結果を用いた対象物操作がモデルのずれの大きな影響を受けずに実行可能であることを示した．対象物を偏心させることによって運動モデルに変化を与え，シミュレーションで獲得した操作がこのような運動モデルの変化に対応可能であることを示した．これは対象物の後ろ側に回りこむという評価を導入したためであり，2 値で与えた評価指針が実世界においても妥当であることを確認した．

力覚センサ情報から接触情報を得たオンラインでの押し操作実験により，実機においてセンサ情報を用いた状態と行動の生成が可能であることを示した．手先と対象物の接触を維持するタスクにとって，視点移動を行うことが重要であること，さらにこのような自身のセンサ情報を用いた状態生成を別の学習に利用するためには評価，行動の生成方法などを拡張する議論があることなど，今後の課題について議論した．

第6章　結論

6.1 結論

本研究では，視覚情報を用いて状態空間および行動空間を自律的生成するための方法を提案し，シミュレーションによってその有効性を示し，実験によって実世界に適用するための問題点について考察した．

ロボットの身体性を考慮した学習を実現する上で，多次元センサから情報を取り出す過程は重要な意味を持っている．これはロボットの行動とその評価に基づいて行われねばならず，本研究では即時的に 2 値の評価を与えられるという問題設定で，その過程を「状態・行動空間の自律的生成」という形で実現した．

本研究では状態を識別するためにベクトル量子化アルゴリズムを用い，TRN による位相関係を保持した状態の生成により離散的な状態表現に基づいて連続値の行動出力を扱うことを可能にした．同じ行動をとって評価の異なる状態を識別し，同じ行動をとって評価が同じである状態同士を結びつけるという考え方に基づいて状態を生成し，RBF によって離散的に獲得した状態と行動とを平滑化し，結びつけるアルゴリズムを提案した．

シミュレーションを行った結果，本研究で提案する手法が 2 値の評価を与えることのできる枠組みの中で，多次元センサに対応した適切な状態・行動生成が可能であることを確認した．また，また，RBF を用いた行動修正が行動の平滑化，効率化に有効であることを示した．従来の状態空間の自律的生成の研究においては，状態を分割することだけが注目されてきたが，このようにすでに分割された状態同士を結び付け，冗長な状態表現から情報をまとめる過程も今後注目されなければならないと考える．

実験を行った結果，シミュレーション結果を用いた押し操作がシミュレーションと実機のモデルのずれを吸収できることを確認した．また，実機における接触情報を用いた学習により，オンラインで状態と行動とを自己組織することが可能であることを示した．

6.2 今後の展望

本研究は，状態空間を生成するということを主目的としてなされたため，行動学習能力の観点からは充分な学習方法を提案しているとはいえない．遅延信号を用いる強化学習の枠組みなどを適用し，本手法によって得た状態および行動に基づいてどのように行動が獲得可能かということが，今後議論されねばならない．そのための重要な鍵の一つは，5 章でも述べた，自身のセンサ情報を評価信号として用いた状態・行動の生成であると考える．自身のセンサ情報を評価信号として用いることは状態・行動の関係獲得という意味を持っており，この評価信号が多次元あるいは連続値になったときの問題を考えることは非常に興味深い．

本研究は多次元センサ入力の代表的例として画像入力を扱ったが，画像情報を直接に利用することによる問題には本手法は充分に対処しているとはいえない．注視機構を導入するなど，視覚入力から状態空間を構成するより適切な方法について検討する必要がある．

本提案手法により状態同士を結びつける表現を達成したが，状態を結びつけることによる情報表現の効率化については充分に議論がなされなかった．より大局的に状態を統合するための位相関係の利用方法を検討する必要がある．

本研究の手法では，センサ入力空間は多次元，アクチュエータ出力空間は低次元の問題を扱った．しかし，マニピュレータの関節角から行動を記述しようとするならば，アクチュエータ出力も単なる乱数出力ではなくその運動学の構造を獲得しながら行動空間を自己組織するべきである．統計処理による次元圧縮 (SOM アルゴリズムでは Fitting の繰り返し計算に相当する) が必要になると考えられる．

謝辞

本修士論文を書くにあたり，多くの方々にお世話になりました．ここに深く感謝の意を表します．

新井教授には，研究の方向性について，自己主張をさせていただいたことを感謝するとともに，多くの迷惑をかけてしまったことを申し訳なく思っています．今後より一層研究に邁進し，学習の研究を画餅に終わらせぬよう努力することでご恩に応えたいと思っています．本当にありがとうございました．

太田助教授には，研究の上で普段から相談に乗っていただき，研究に対するモチベーションを大いに支えていただきました．また，空回りし，先が見えない状況で非常に救われました．頑迷な主張をする扱いにくい学生であったことと思われますが，その都度的確な助言をいただき，深く感謝しています．

相山助手には，研究の上でお世話になると同時に，研究の方針についてたびたび衝突し，多大な迷惑をかけてしまいました．僕と異なる視点から多くの厳しい指摘をいただいたことによって鍛えられました．どうもありがとうございました．

研究員の楊さん，廣木さん，博士課程の朱さん，佐々木さん，宮田さん，チャチャイさん，井上さん，中村さん，原さん，山下さんには，先輩の研究者として様々なことで助言をいただき，ありがとうございました．特に，博士 1 年の井上さんが学習をテーマに取り組んでいたことは大きな励みになりました．

修士 2 年の柿田君は，いろいろな点で見習うべき点を備えた同輩であり，研究でも研究以外の面でも，良い刺激を与えてもらいました．

修士 2 年の原田君には，技術的な面で非常に多くのことで世話になったのみならず，精神的に煮詰まったときにもしばしば助けてもらいました．

修士 2 年の平野君には，Free cell のやり方を教えてもらいました．研究中の気分転換に大いに役立ちました．

修士 1 年の池田君，今西君，木地本君，杉君，西林さん，福地君，顔君および学部生の岩田君，河野君，竹内君，千葉君，福田君，山本君，横井君には，様々な刺激をいただき，研究を持続させる活力を与えていただきました．

最後に，いろいろな形で僕を支え，励ましてくれた家族，友人，先輩方に感謝いたします．今後より良い研究を行い，胸を張って報告できるような成果を残せるよう，より一層努力します．

1999 年 2 月 10 日

小林 祐一

参考文献

[Asada96] S.Noda M.Asada and K.Hosoda: Action-based sensor space categorization for robot learning. *Proc. of IEEE/RSJ Int. Conf. IROS96*, pp. 1502–1506, 1996.

[Barto83] Richard S. Sutton Andrew G. Barto and Charles W. Anderson: Neurolike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on systems, man, and cybernetics*, Vol. SMC-13, No. 5, 1983.

[Bruske95] Gerald Sommer Joerg Bruske: Dynamic cell structure learns perfectly topology preserving map. *Neural Computation*, Vol. 7, pp. 845–865, 1995.

[Bursevski96] V. Burzevski and C. K. Mohan: Hierarchical growing cell structures. *ICNN96: Proc. Int'l. Conf. Neural Networks*, 1996.

[Choi94] Doo-II Choi and Sang-Hui Park: Self-creating and organizing neural networks. *IEEE Transactions on neural networks*, Vol. 5, No. 4, 1994.

[Dayhoff92] E. Dayhoff: ニューラルネットワークアーキテクチャ入門. 森北出版株式会社, 1992.

[Dreyfus92] H.L.Dreyfus: コンピュータには何ができないか：哲学的人工知能批判. 産業図書, 1992.

[Fritzke94] Bernd Fritzke: Fast learning with incremental rbf networks. *Neural Processing Letters*, Vol. 1, No. 1,2-5, 1994.

[Ishiguro96] R.Sato H.Ishiguro and T.Ishida: Robot oriented state space construction. *Proc. of IEEE/RSJ Int. Conf. IROS96,*, pp. 1496–1501, 1996.

[Kohonen96] T.Kohonen: 自己組織化マップ. シュプリンガー・フェアラーク東京, 1996.

[Martinetz94] Thomas Martinetz and Klaus Shulten: Topology representing networks. *Neural Networks*, Vol. 7, No. 3, pp. 507–522, 1994.

[Murao97] Hajime Murao and Shinzo Kitamura: Q-learning with adaptive state segmentation(qlass). *Proceedings of IEEE International*

Symposium on Computational Intelligence in Robotics and Automation(CIRA), Vol. 179-184, , 1997.

[Song98] Hee-Heon Song and Seong-Whan Lee: A self-organizing neural tree for large-set pattern classification. *IEEE Transactions on neural networks*, Vol. 9, No. 3, 1998.

[Touzet97] Claude F. Touzet: Neural reinforcement learning for behaviour synthesis. *Robotics and autonomous systems*, No. 22, pp. 251–281, 1997.

[小林 99] 小林祐一, 太田順, 井上康介, 新井民夫: 視覚・接触情報を用いた状態・行動空間の自律的生成. 第 11 回自律分散システム・シンポジウム, pp. 275–280, 1999.

[宮崎 95] 宮崎文夫: スキルと学習. 日本ロボット学会誌, Vol. 13, No. 1, pp. 20–24, 1995.

[マクドーマン 99] Karl F. MacDorman: 感覚—運動統合による記号接地. 日本ロボット学会誌, Vol. 17, No. 1, pp. 20–24, 1999.

[松原 90] 松原仁 J.McCarthy: 人工知能になぜ 哲学が必要か—フレーム問題の発端と展開. 哲学書房, 1990.

[銅谷 99] 銅谷賢治: 認知ロボティクスの目指すもの. 日本ロボット学会誌, Vol. 17, No. 1, pp. 2–6, 1999.

[國吉 99] 國吉康夫, ベルトゥーズリュク: 身体性に基づく相互作用の創発に向けて. 日本ロボット学会誌, Vol. 17, No. 1, pp. 29–33, 1999.