

自律移動ロボットのための実時間神経修飾機構を用いた
コントローラの構築に関する研究

十倉 征司

目次

第1章 緒論	1
1.1 本研究の背景	1
1.2 ロボットコントローラの構築	3
1.2.1 ロボットコントローラの自律的構築手法	3
1.2.2 進化的計算手法によるコントローラ構築	5
1.2.3 状況のカテゴリ化と対応する行動発現の重要性	8
1.3 自律ロボットコントローラのハードウェア化	10
1.4 本研究の目的	12
1.5 本論文の構成	13
第2章 ニューラルコントローラの実時間神経修飾機構	15
2.1 はじめに	15
2.2 実時間神経修飾機構を有するニューラルコントローラ	18
2.2.1 生物における神経回路の動的な機能変更	18
2.2.2 神経回路の動的な改変機構のモデル化	19
2.2.3 実時間神経修飾機構による入出力特性の変化	26
2.2.4 実時間神経修飾機構のパラメータの自律的決定	27
2.3 ペグ運び問題を行うコントローラの自律的構築	29
2.3.1 ペグ運び問題	29
2.3.2 進化フェーズ	32
2.3.3 仮想環境から実環境への移行	33
2.3.4 提案手法の実環境変化への適応	35
2.4 まとめ	43

第 3 章	FPGA を用いた実時間神経修飾機構の実現	45
3.1	はじめに	45
3.2	ニューラルネットワークのハードウェア化	46
3.3	実時間神経修飾機構のハードウェア化	47
3.3.1	ニューロモジュレータ放出部のハードウェア化	49
3.3.2	ニューロモジュレータ反応部のハードウェア化	49
3.4	実機実験	49
3.4.1	実験環境	51
3.4.2	実機実験による頑健性の検証	53
3.4.3	初期環境変動に対する頑健性の検証	54
3.5	回路規模の比較	58
3.6	まとめ	59
第 4 章	実時間神経修飾機構の解析と改良	63
4.1	はじめに	63
4.2	ニューラルコントローラの入出力写像能力向上に向けた実時間神経修飾機構の改良	64
4.2.1	実時間神経修飾機構の改良すべき点	64
4.2.2	実時間神経修飾機構の改良	65
4.2.3	コントローラの入出力写像能力を改良したコントローラの確認	68
4.3	実時間神経修飾機構の行動解析	69
4.3.1	コントローラの行動特性を示す状態遷移ベクトル図	69
4.3.2	状況のカテゴリライズとそれに応じたコントローラの発現による未経験環境に対する頑健性	71
4.4	実時間神経修飾機構の改良による状況カテゴリライズ能力の向上	72
4.4.1	状態カテゴリライズを行うニューロモジュレータ放出部の改良	74
4.4.2	バックトレース問題への適用	76
4.4.3	改良による頑健性向上の確認	78

4.5	実時間神経修飾機構を有するニューラルネットワークのノイズに対する頑健性	79
4.6	まとめ	81
第5章	結論	83
5.1	本研究の成果	83
5.2	今後の課題	84
付録A		87
付録		87
A.1	ニューラルネットワークの確率的エンコーディング手法によるハードウェア化	87
A.1.1	ニューラルネットワークのハードウェア化	87
A.1.2	乗算のハードウェア化	88
A.1.3	シグモイド関数のハードウェア化	89
A.1.4	ニューラルネットワークのハードウェア化	91
参考文献		93
業績		101
謝辞		105

第1章 緒論

1.1 本研究の背景

近年、ロボットに関する技術革新はめざましく、愛知で開催された2005年日本国際博覧会においては、近未来での活躍を想定したロボットが多数発表されたことは記憶に新しい(図1.1)。また身近な所では、専門家ではない一般者向けのペットロボットや、リハビリ時に利用するロボットなど、日常生活で使用するロボットがいくつも発売、もしくは発表され、人々の期待と関心が高まっている[1, 2, 3, 4, 5, 6, 7]。そのため、様々な環境下でも、自律的に適切に動作しうるロボットシステム技術が求められている。しかし、ロボットシステムを構成する部品において、同じ品番であっても見られる個体差による特性の違い(図1.2参照)や非線形事象、不確定性など、現実世界にはモデル化困難な要因が数多く存在すると共に、これら様々な要因は、それぞれが錯綜して相互作用している。そのため、タスクの複雑化やロボットの自由度の増加に伴い、実時間で計算することを必要とするロボットのコントロールに対する適切なモデル化は非常に困難となっている[8]。つまり、コントローラ的设计段階において、制御対象を厳密にモデル化する必要がある従来の古典的制御手法[9, 10]を用いる場合、複雑なタスクを行う自律ロボット、あるいは、実験室や工場のような整備された特定環境ではない環境下において働くロボットシステムを構築することは未だに非常に困難である。

そこで、モデル化が困難な事象や錯綜する相互作用を考慮に入れたコントローラの自律的な構築手法を確立することが必要とされている。さらに、実用化に向けて自律ロボットを設計・構築する際には、コントローラや電源、駆動装置など、ロボットを構成するあらゆる部品をロボット内部に収納しつつ、性能の維持をも考慮する必要がある[11]。そのためには、消費電力の低

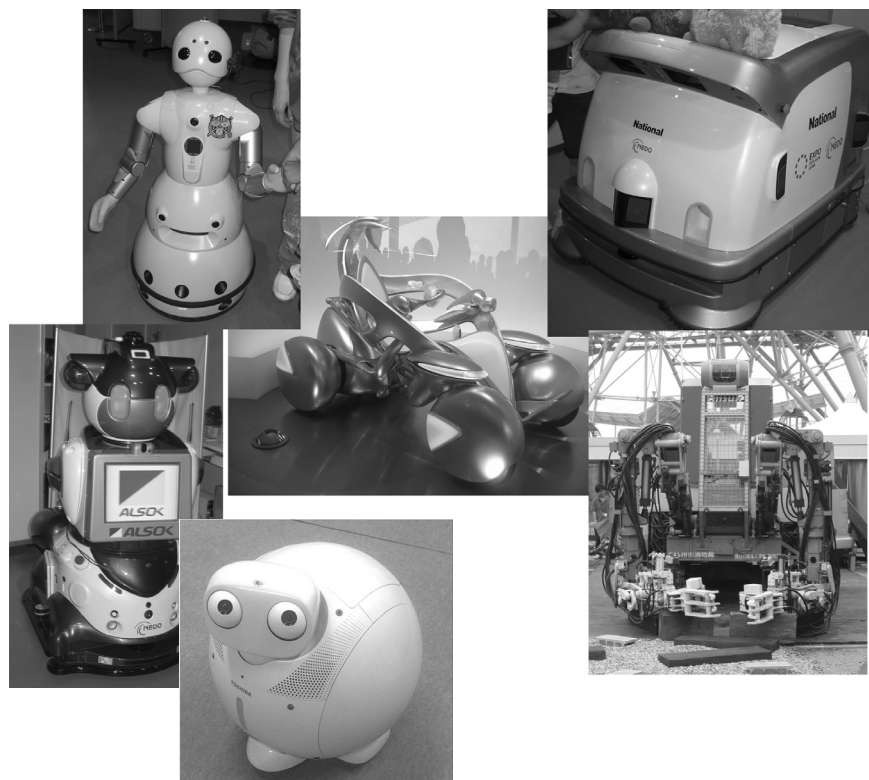


図 1.1: 2005 年日本国際博覧会で展示されたロボットの数々(コミュニケーションロボット Wakamaru(左上)、警備ロボット(左下)、パーソナルモビリティ i-unit(中央上)、聞き分けロボット ApriAlphaTM V3(中央下)、清掃ロボットの SuiPPi(右上)、大型レスキューロボット T52 援竜(右下))

減、構成部品の減少簡略化が必要となってくる。これらの要求に対し、コントローラのハードウェア化は、ロボットの小型化、コストダウンなどに大きく寄与すると考えられ、一つの有効な手段となることが期待される。

以下の節では、モデル化が困難な事象や錯綜する相互作用を考慮に入れたロボットコントローラの自律的構築手法とその一手法である進化ロボティクスについて述べ、問題点を明らかにする。加えて、頑健に動くロボットのコントローラの構築に関する研究例として、状況に応じて、行動を切り替えることで、すべてを記述することなく、頑健性を実現した研究例を取り上げ、

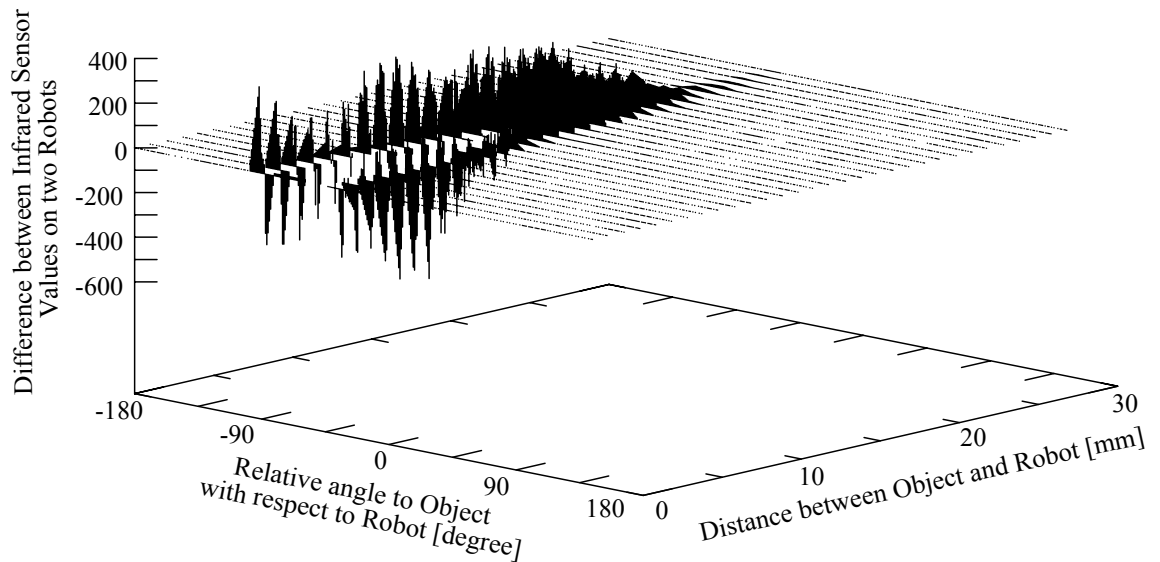


図 1.2: 2 台の *KheperaTM* の同じ部位に搭載されている赤外線センサの周りにペグ (直径 4cm) をそれぞれ置いたときの相対距離・相対角度に対するセンサ値の誤差 (センサ値はセンサの正面から見た近接物の方向と距離に依存した 0 ~ 1023 の整数値を取る)

状況のカテゴリ化と各状況に応じた行動発現の重要性と現状の問題点を述べる。その後、ロボットの体内という限られた空間に仕様を満たすシステムを構築する必要がある自律ロボット設計の観点から、考慮すべき点を取り上げ、システムのハードウェア化について述べる。最後に本研究の目的を明らかにする。

1.2 ロボットコントローラの構築

1.2.1 ロボットコントローラの自律的構築手法

ここでは、モデル化が困難な事象や錯綜する相互作用の影響を、設計時に陽に考慮せず構築できるロボットコントローラの自律的構築手法について述

べる。

先に挙げた、現実世界に存在する数限りない様々な要因の相互作用を設計者が厳密にモデル化することなく、ロボットコントローラ設計に反映させる手法として、ロボットの身体性を考慮し、環境とロボットの相互作用に着目した手法が注目を集めている [12, 13, 14]。なぜならば、ロボットは自らが動くことにより環境に働きかけることができ、同時に、環境が変化することにより自らの感覚入力を変更することができる。つまり、環境とロボットの間には、ロボットのアクチュエータとセンサを通して、相互作用が存在するからである。そこで、この環境とロボットの相互作用に着目し、提案されてきた様々な学習手法や適応手法について、例を次に述べる [15, 16, 17, 18, 19, 20, 21, 22, 23]。

強化学習によるコントローラ構築

ロボットのコントローラ学習手法の一つとして、強化学習を用いた様々な研究がなされてきている [24]。浅田ら [25] は、強化学習の手法を用いた自律的なロボットのコントローラ構築手法を提案した。浅田らの実験では、コントローラ出力が身体を通して物理的に引き起こされる事象である行動を、タスクに応じて適切に起こすために、まず、ロボットの直面する状態を示す状態空間と各状態空間で行うべき行動候補であるプリミティブな行動要素と、状態遷移が行われた際に行った行動の評価を行う評価関数を用意した。その後、複数回の試行を通じ、各状態における評価値を高めるプリミティブな行動を算出した。これにより、タスクを達成する一連の動作を確率的に算出し、ロボットコントローラの自律的構築が可能であることを示した。また、強化学習手法で欠点であった状態空間の分割の仕方について、高橋ら [26] は、自律的に状態空間を分割する手法を提案し、その手法の有効性を示した。

しかし、強化学習の手法はタスク達成にマルコフ過程を前提としている。そのため、一連の行動に対する報酬の与え方など、モデル化は行わないまでもタスクに対して適切な設計をコントローラ的设计者が行う必要がある。加えて、設計者である人と身体性の異なるロボットにとって、設計者が用意したプリミティブな行動は適切とは限らず、自律的に構築する方が望ましい。し

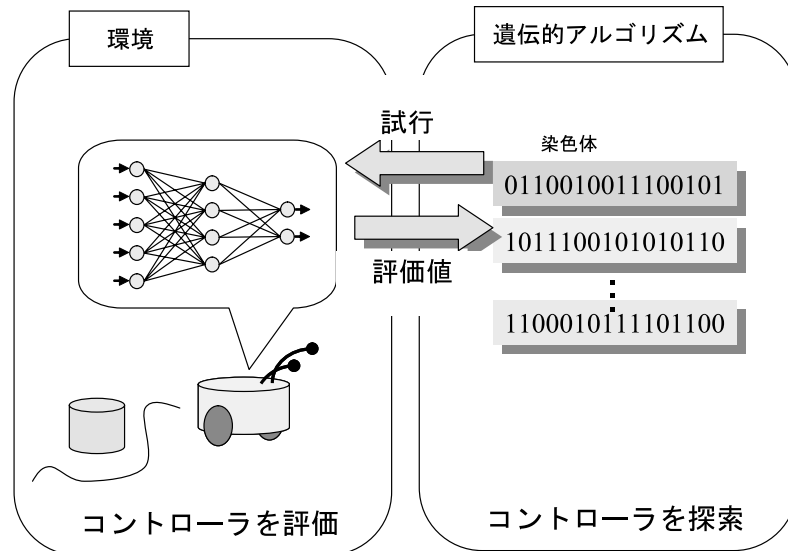


図 1.3: 進化ロボティクスのイメージ図

かし、選択すべき行動の最小単位であるプリミティブな行動の設計を行う必要があるといった問題が存在する。これらの問題に対し、遺伝的アルゴリズム (Genetic Algorithm: GA や Genetic Programming: GP)[27, 28, 29, 30, 31] などに代表される進化的計算手法とニューラルネットワークを用いて、プリミティブな行動をも自律的に獲得させ、ロボットのコントローラを構築する研究がある [32, 33]。

1.2.2 進化的計算手法によるコントローラ構築

ここでは、進化的計算手法を用いてロボットのコントローラを自律的に構築させる進化ロボティクスについて述べる。

図 1.3, 1.4 にそれぞれ、進化ロボティクスのイメージ図とフローチャートを示す。進化ロボティクスの手法は、ロボットのコントローラのパラメータを遺伝子に見立て、遺伝的アルゴリズム (GA や GP) を利用し、試行と評価を

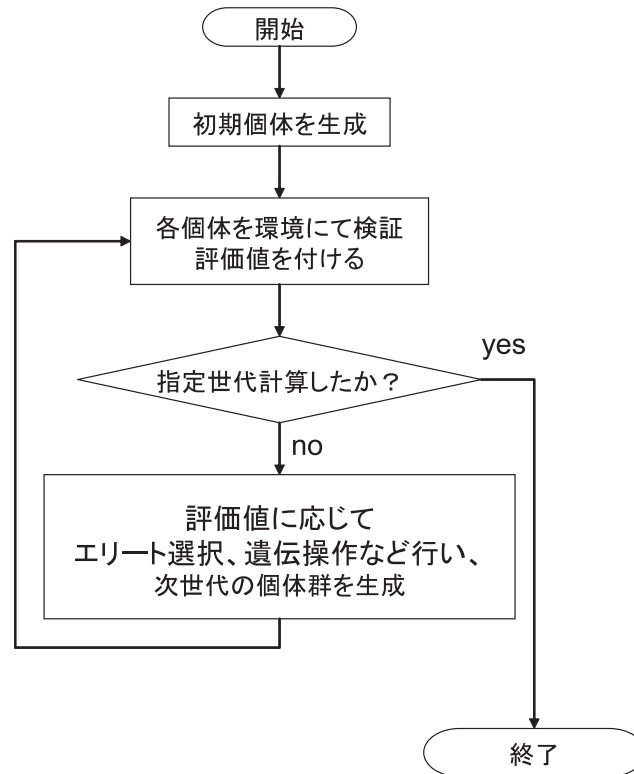


図 1.4: 進化ロボティクスによるコントローラの自律的構築のフローチャート

繰り返し行なうことで自律的にロボットのコントローラを構築する手法である。まず初めに、設計者はタスクに応じた評価値を設計すると共に、様々なパラメータをとることで、複数のロボットコントローラ (初期個体) を製作する。そして、製作したロボットコントローラを環境内で実行検証し、全てのコントローラに対して評価値を算出する (個体評価フェーズ)。その後、算出した評価値に基づいて個体選別や遺伝子操作などを行ない、次の世代の個体群を生成する (次世代生成フェーズ)。進化ロボティクスの手法は、この「個体評価フェーズ」と「次世代生成フェーズ」を繰り返すことにより、ロボットの身体性やロボットを取り巻く環境を、設計者が陽に意識しなくても、タスクに応じたコントローラのパラメータを自律的に獲得していくことができる利点を持つ。

また、進化ロボティクスの分野では、ロボットのコントローラとして、センサ入力とモータ出力を直接結合する低次元の行動記述ができるニューラルネットワークを用いた研究が多くなされている [33, 34, 35]。これは、ニューラルネットワークを用いることにより、プリミティブな行動要素までも身体性に応じて自律的に構築することが可能となり、設計者がロボットの立場に立った設計を行う必要がなくなる利点がある。

また、進化ロボティクスの多くの研究では、コントローラの構築をシミュレーション上にて進化させることで、下記にあげる利点を享受している：

- コントローラ構築時に絶対座標など容易に計測できない値をコントローラの評価・構築に反映できる
- 重力の大きさの変更やロボットの形状・大きさなど現実では変更が困難な要素をも考慮して構築できる
- 実機を用いた実験による実機の故障・劣化を防ぐ
- 多数回試行が容易である

一方で、進化ロボティクスには次にあげるいくつかの問題点の存在が指摘されている [36, 37]：

- 進化に多くの試行を必要とするため、実機を用いたコントローラ構築は困難である問題 (オンライン進化の困難性問題)
- オンライン進化の困難性問題のため、多くの研究では、進化環境としてシミュレーション環境を多く用いてきた。しかし、シミュレーション環境は実世界を完全に模擬することはできない。そのため、シミュレーション環境で育てたロボットコントローラを調整なしに実環境に直接移植しても、適切には動作し難い問題 (シミュレーションと実環境のギャップ問題)
- タスクの難易度が高い場合やコントローラの自由度が高すぎる場合、初期段階における個体同士の評価値の差が不十分なため、適切な淘汰が

発生せずに、コントローラ構築が失敗する問題 (進化のブートストラップ問題)

上記のうち、シミュレーションと実環境のギャップ問題については、従来型の進化の対象に問題があったと考えられる。多くの研究で用いられているニューラルネットワークは、非線形機能を有する閾値素子を多数用意し、それらを静的な結合荷重を持つシナプスで結合し、その結合荷重に情報を埋め込んでいる。そして、ニューラルネットワークを用いる従来の研究では、この情報を埋め込んだ静的な結合荷重を進化の対象として解探索を行う。これは、シミュレーション環境における、単なるセンサ入力に対する行動出力の関係を獲得しているに過ぎないことを意味する。つまり、進化環境であるシミュレーション環境に過度に適応してしまい、実環境の不確定性や、モデル化しきれなかった要素が存在する実環境、言い換えると、入出力関係が若干異なる環境には、従来手法のコントローラは適応できなかったものと考えられる。

上記のシミュレーションと実環境のギャップ問題に対し、Miglinoら [38] は、進化させた後の個体群を継続して実環境にて進化させる手法を提案した。しかし、Miglinoらの手法を用いた場合、実環境における試行のため、多大な時間と手間を必要とするばかりでなく、進化過程に用いる評価値として、「絶対座標など内界センサなどだけでは計測できない値」を用いることができなくなるといった問題がある。また、このシミュレーションとのギャップ問題の他の対処手法として、評価値の平均化 [36] や進化過程におけるノイズ添加 [39] などの手法が提案されている。しかし、適切なノイズ量が不明であるなどの問題があり根本的な解決には至っておらず、環境変化に対し頑健なコントローラの構築法が求められている。

1.2.3 状況のカテゴリライズと対応する行動発現の重要性

ここでは、コントローラへの入力履歴によって表される、ロボットが直面する状況のカテゴリライズ化について考慮した研究例を挙げる。そして、状況に応じて、行動を切り替えることで、すべてを記述することなく、頑健性を実現した例を紹介し、状況のカテゴリライズ化と各状況に応じた行動発現の重

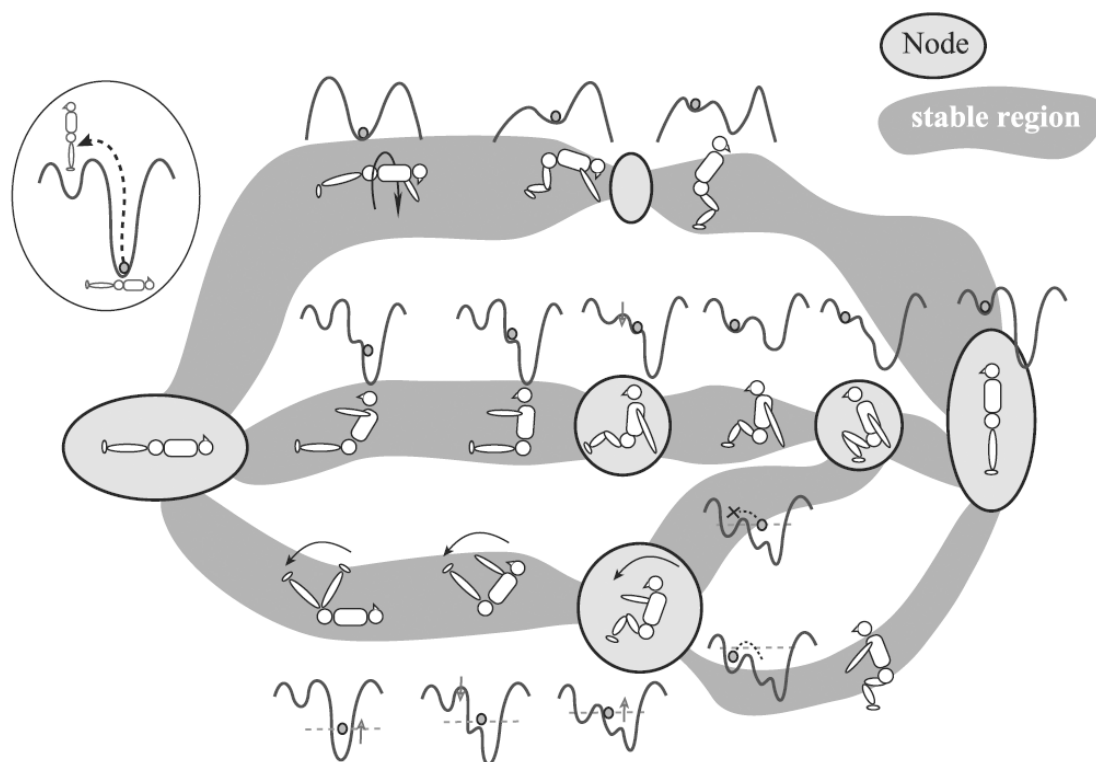


図 1.5: 起き上がり動作における "コツ" (図中の "Node") と呼ばれる力学的拘束条件の強い軌道収束点と拘束条件の緩い状態軌道束 (図中の "stable region") によって構成される起き上がり動作 (國吉ら [57] の文献から抜粋)

要性と現状の問題点を述べる。

國吉ら [56] は、人の起き上がり動作をキャプチャし解析することで、図 1.5 に示すように、タスク成功する一連の動作の中に "コツ" と呼ばれる、状態変化の系列中に存在する力学的拘束条件の強い特定の状態と、それらを繋ぐ拘束条件の緩い状態軌道束によって、起き上がり動作が成立することを明らかにした。同時に、その "コツ" と "コツ" の間は多様な行動を取りうることも示唆した。この知見を基に、観測した一連の力学的拘束条件の強い特定の状態に状態遷移するように、状態に応じた制御を行ない、等身大ロボットの起き上がり動作を実現した。

また、中村ら [21, 22, 23] は、閉曲線をアトラクタとする力学系を設計し、特定の運動パターンをその力学系に形成することができることを示した。そして、同じ力学系に示した複数のアトラクタ間を遷移させることで、一連の動作を行う際に、すべての行動を記述することなく、滑らかに行動することが可能であることを示した。

これらの研究からわかるように、一連の行動すべてを記述するのではなく、多数の行動をカテゴリ化し、そのカテゴリ化した行動を安定化させる、あるいは行動の切り替え時の拘束条件を厳しくすることで、頑健な行動が可能であることが知られている。しかし、タスク達成のために行動をどのようなタイミングで切り替えるか、つまり、身体を通したコントローラへの入力信号履歴によって表現される状況をどのようにカテゴリ化するかを自律的に獲得することは非常に困難である。

1.3 自律ロボットコントローラのハードウェア化

次にこの節では、自律ロボット構築にあたり、ロボットコントローラのハードウェア化について述べる。

実用の自律ロボットを設計する際には、生産コストや部品信頼性の向上、保守性などの様々な仕様を満たす必要がある。しかし、大きなシステムになればなるほど、ロボットの大きさは大きくなり、重量は増加し、ロボットのアクチュエータやバッテリーの増強が必要となり、さらにロボットが大きくなるという問題が生じてくる。そこで、自律ロボットの設計の際には、構成要素の小型化、軽量化、消費電力の削減、コントローラ改変の容易さなどを考慮する必要がある。これらの要求に対し、表 1.1 にコントローラを専用のハードウェアをすべて製作する場合と、FPGA などプログラマブルハードウェアでコントローラをハードウェア化した場合と、パソコンなどをコントローラとして搭載する場合との比較を示す。

表 1.1 に見られるように、専用ハードウェアコントローラを利用する場合は、他の手法に比べ利点が多いものの、コントローラの改変が容易ではない

表 1.1: 専用ハードウェアコントローラを利用する場合と FPGA など
でコントローラをハードウェア化する場合とパソコンなどを
コントローラとして搭載する場合の比較

	専用ハードウェア コントローラを 利用	FPGA など で コントローラを ハードウェア化	パソコンを コントローラとして 搭載
コントローラ 容積			
重量			
消費電力			×
変更の容易さ	×		
並列処理			×
多入力 多出力処理			
価格	×		

点や価格の面で大きな問題があり、改変を要しないシステムを大量生産する場合以外では、あまり適切ではない。一方、コントローラをFPGAなどを用いて構成した場合は、システムの改変が可能なおうえ、パソコンなどをコントローラとして搭載する場合に比べ、消費電力の低減化、容積の減少、重量の軽減が期待できる。これらは、電源系なども搭載した自律ロボットにおいては、大きな利点となる。

これらの長所を受け、進化ロボティクスの手法などを取り入れることにより、ハードウェアコントローラに適応機能を搭載させる研究 [40, 41] や、自律的にハードウェアコントローラを構築する手法も研究されてきた [42, 43, 44, 45]。しかし、樋口ら [46] の手法などに見られる専用のハードウェアを必要とする問題や、オフライン進化の場合は先の項で挙げたシミュレーションと実環境のギャップ問題などのために頑健性を高める必要があるなどの問題が存在している。

1.4 本研究の目的

これらを受け、本論文では、ロボットに影響を与える外界である環境が、コントローラを構築する際に想定した環境だけでなく、シミュレーション環境と実環境など、若干異なった場合においてもタスク達成できる、頑健なロボットコントローラの自律的構築を行う。同時に、獲得したコントローラの解析を行うことで、得られたコントローラが状況に応じてコントローラを発現させ、頑健に動くことを示す。

最初に、頑健にタスク達成を実現するコントローラの自律的構築のために、本論文では状況に応じてコントローラの入出力関係を変更できるニューラルネットワークと進化ロボティクスの手法を用いて、自律的に状況のカテゴリ化と、各状況における適した行動を獲得できるコントローラの自律的構築手法の提案を行う。そして、コントローラの調整することなく、シミュレーション環境から実環境への移行を行う実験と、未経験状態からのタスク成功率を例に示し、本手法の頑健性を示す。さらに、実行中のニューラルコント

ローラのシナプス荷重の時間推移の解析を通して、未経験環境における、状況に応じた行動の切り替え発現例を示し、本手法の有効性を示唆する。

次に、自律ロボットの設計の際に重要な、構成要素の小型化、軽量化、消費電力の削減、コントローラ改変の容易さなどに期待できるコントローラのハードウェア化を目指す。そのために、回路規模を考慮した手法を開発し、FPGAを用いたコントローラのハードウェア化を行い、実機にて実験することで提案するニューラルコントローラの有効性を示す。同時に、FPGA上にてハードウェア化したコントローラの回路規模を従来手法と比較することで、開発した回路規模縮小手法の有効性を示す。

最後に、コントローラの写像能力の向上のため、提案手法の改良を行う。同時に、得られたコントローラの機能の解析を行い、コントローラが獲得した状況のカテゴリズに応じて、的確に行動を切り替えることで、コントローラの経験していない環境下でもタスク達成が可能であることを示し、本アプローチの妥当性を示す。

1.5 本論文の構成

この節では、本論文の構成について説明する。

まず本章では本研究の背景を説明し、自律移動ロボットを作る上で問題となるコントローラの構築に関する困難性や、コントローラのハードウェア化を行うことによる利点などを示した。そして、本研究の目的を述べた。

第2章では、進化ロボティクスの問題点や特定の環境以外では動かないロボットの問題点に対し、自律的に状況を認識し、コントローラの入出力特性を実時間で変化させることで、状況にあったコントローラに再構成する実時間神経修飾機構を有するニューラルネットワークを提案する。そして、シミュレーションから実世界への移行が可能であることを示し、頑健性を示すと共に、実行時のニューラルコントローラのシナプス荷重の時間推移を解析することで、提案手法の有効性を示す。

第3章では、先に提案した実時間神経修飾機構を有するニューラルネット

ワークのハードウェア化手法を示す。その際に、従来のシグモイド関数のハードウェア化手法を用いた場合の回路規模増大問題に対処するため、回路規模の増大を緩和する手法を取り入れ、回路規模を抑えつつ、第2章で提案した実時間神経修飾機構のハードウェア化を行う。

第4章では、自律的に構築したコントローラを解析し、設計者が陽に介在せずとも、状況に応じたコントローラの獲得とタスクを行う際の状況のカテゴリライズ化ができることを示す。そして、コントローラが獲得した状況のカテゴリライズに応じて、的確に行動を切り替えることで、経験していない環境下でも行動を切り替えることができ、頑健性が高まることを示す。加えて、提案した実時間神経修飾機構に改良を加えることで表現能力が高まることを示す。

最後に、第5章では、本論文のまとめを行うと共に今後の課題について述べる。

第2章 ニューラルコントローラの実時間神経修飾機構

2.1 はじめに

第1章で述べたように、現在使われているほとんどのロボットは、環境を含む制御対象をモデル化し、そのモデルに対してコントローラを構築し、所望のタスクを行うように制御を行う。しかし、複雑なタスクを行うロボットを構築する際や、環境が複雑な場合、様々な要因が錯綜して相互作用するため、適切にモデル化することは難しく、所望のタスクを達成するロボットコントローラを製作することは難しくなる。

この問題に対し、低次元の行動記述ができるニューラルネットワークをロボットコントローラとし、自律的なコントローラの構築手法である進化ロボティクスの手法を利用することにより、設計者が陽に意識せずに、錯綜する環境の様々な要因を反映したコントローラを自律的に構築することが可能となる。しかし、進化ロボティクスの手法には、進化のブートストラップ問題や、オンライン進化の困難性問題と呼ばれる問題が存在する。加えて、従来のニューラルコントローラの静的な結合荷重を進化の対象とした進化ロボティクスの手法では、進化環境であるシミュレーション環境における入出力関係を求めているに過ぎない事に起因し、結果的に、実環境のような進化環境とは異なる環境において適切に動作しないという問題があった。

これらの問題を受けて本研究では、従来の多くの研究が行ってきた、図2.1に示す全ての状況に対応できる入出力関係を進化の対象にするのではなく、「どのような状況」で「どのような行動」を取るべきかというルールを進化の対象にする。具体的には、状況をカテゴライズ化するためのパラメータと、

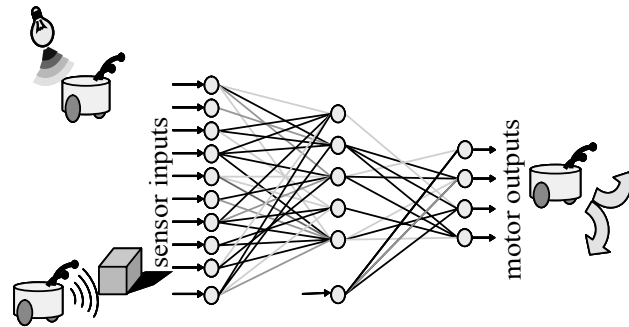


図 2.1: すべての状況に対応可能な従来のコントローラのイメージ図

カテゴライズ化された各状況に応じた行動を発現するコントローラの入出力関係を決定するパラメータを進化の対象とすることで、タスク達成に必要な「プリミティブな行動」を行う専門コントローラと、「同じプリミティブな行動を必要とする状況のカテゴライズ化」を同時に自律的に獲得させる。これにより、タスク達成に向けて、図 2.2 に示すように、状況に応じた専門のコントローラを発現させ、頑健性を高める。そのために、状況に応じてコントローラの入出力関係の変更が可能なコントローラとして、生物が具備している状況に応じて動的に神経回路の機能を変更する機構をモデル化し、ニューラルネットワークに動的な改編能力を持たせる。

本章では、この機構を用いた状況に合わせて行動を発現できるコントローラと、その自律的構築手法を提案する。そして、実環境や未経験状況の初期状態からの実行実験などを通じ、提案手法の頑健性を示す。同時に、実行中のニューラルコントローラのシナプス荷重の時間推移を解析することで、状況に応じて適切に行動を起こすことで頑健に動く様子を示し、本手法の有効性を確認する。

以下、本章の構成を示す。2.1 節においては、背景として進化ロボティクスの問題を挙げた。この問題に対し、2.2 節にて、状況に応じてコントローラの入出力関係を動的に変更する機能を実現するために参考にした、生物の動的な神経回路の変更機能の事例を紹介する。そして、提案手法である実時間神経修飾機構を有するニューラルコントローラについて説明する。加えて、コ

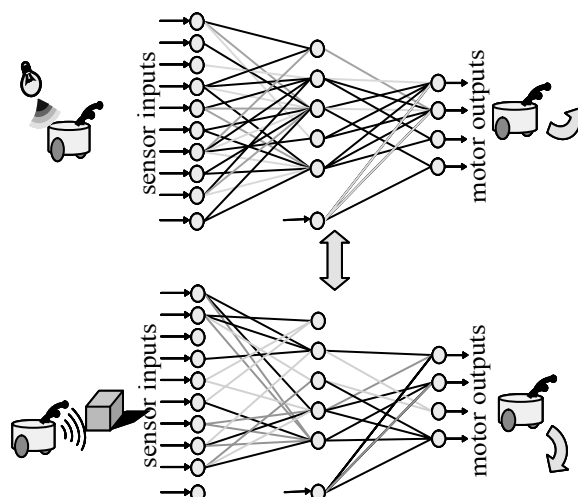


図 2.2: 状況に応じてニューラルコントローラにおけるシナプス荷重を変更し、入出力写像を変更することで、各状況専門のコントローラになる提案手法のコントローラのイメージ図

ントローラの入出力特性を進化の対象とする従来の多くの研究とは異なる、実時間神経修飾機構におけるコントローラの改編を要する状況と各状況に応じた行動を発現させるコントローラのパラメータの進化的構築について述べる。その後、2.3 節において、提案手法の有効性を見るべく、提案手法をペグ運び問題に適用し、シミュレーション環境から実機環境へのコントローラ移行実験を行う。そして、ロボットの行動軌跡と実行中のニューラルコントローラのシナプス荷重の時間推移を解析することで、状況に応じて適切に行動を起こすことで頑健に動く様子を示し、本手法の有効性を示唆する。最後に、2.4 節においてまとめを行う。

2.2 実時間神経修飾機構を有するニューラルコントローラ

2.2.1 生物における神経回路の動的な機能変更

この項では、状況に応じてコントローラの入出力関係を動的に変更する機能を実現するために参考にした、生物に見られる神経修飾機構を説明する。

近年の神経生理学の飛躍的な発展により、実際の神経系では、決して固定された神経回路に基づいて情報処理を行なっているのではなく、極めて動的かつ迅速に状況に応じて、その回路構成が変化していることが明らかになってきた [47]。ここでその一例を挙げる。

Meyrandら [48] は、甲殻類のロブスタの胃の神経回路が、食べ物の飲み込み時にどのように回路構成を変化させているのかに関して詳細な実験的検証を行なった。図 2.3 にその概念図を示す。同図左に示すように、通常時は食道 (oesophageal network)、胃 (gastric network)、幽門 (pyloric network) はそれぞれ独自の発振を行っており、互いに非干渉な関係にある。しかし、食べ物を飲み込んだ時には、これらの発振回路群が直ちに再編成され、同図右のような回路に即座に変化し、飲み込み動作を行う神経回路網が発現する。この、神経回路網の再編成により、飲み込み運動が適切に行なわれる。つまり、生物は状況に応じて神経回路網を再編成することにより、状況に応じた回路網を構築し、異なる機能を発現していることが明らかになってきた。この現象は、今まで存在しなかった結合を生成した（あるいは、存在していた結合を消去した）のではなく、顕在化していなかった結合を活性化することで顕在化した（あるいは、顕在化していた結合を抑制することで非顕在化した）ものと考えられている。このような、状況に応じた身体機能の動的な変化は、猫やアメフラシなど、幅広い生物において確認されている [49, 50]。これら生物の神経回路網の改編時に引き起こされているシナプス結合の抑制化・活性化やシナプス荷重の変更には、ニューロモジュレータと呼ばれる神経修飾物質が大きく関わっていると考えられている。

そこで本研究では、このニューロモジュレータによる神経回路網の動的な

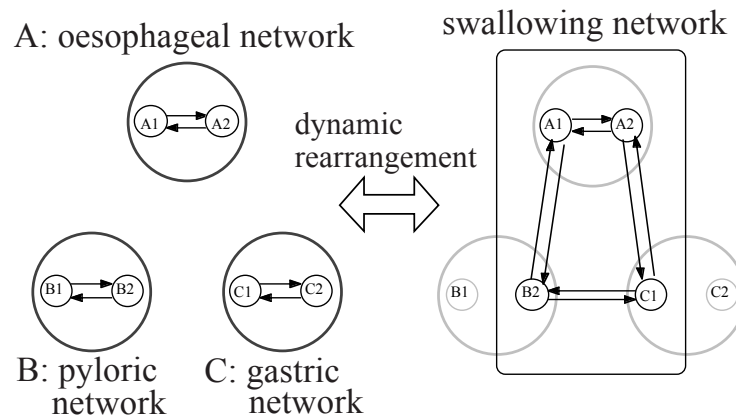


図 2.3: ロボスタに見られる神経修飾機能

変更を可能にする機能を参考に、状況に応じてニューロモジュレータの放出パターンを変化させ、その放出パターンに応じて実時間でコントローラの機能を変更するニューラルネットワークを構築した。以降、本論文では、この実時間でニューラルネットワークの機能変化を促す機構を実時間神経修飾機構と呼ぶこととする。

2.2.2 神経回路の動的な改変機構のモデル化

この項では、例に挙げた生物に見られる状況に応じて神経回路を動的に変更する機能のモデル化を行う。

ニューラルネットワークにおいて、コントローラが置かれた状況に直接関係する事象は、各ニューロンの活性度あるいは活性度の変化履歴である。そこで、本提案手法においては、ロボットの直面している状況をニューロモジュレータの放出パターンに反映させるために、各ニューロンの活性度履歴と関係づける。具体的には、各ニューロンはそれぞれの活性度と保持している条件に応じて、決められたニューロモジュレータを動的に放出するニューロモジュレータ放出部を持つものとした。加えて、ロボットの直面している状況に応じてコントローラの機能、つまり入出力関係を変更するために、各シナ

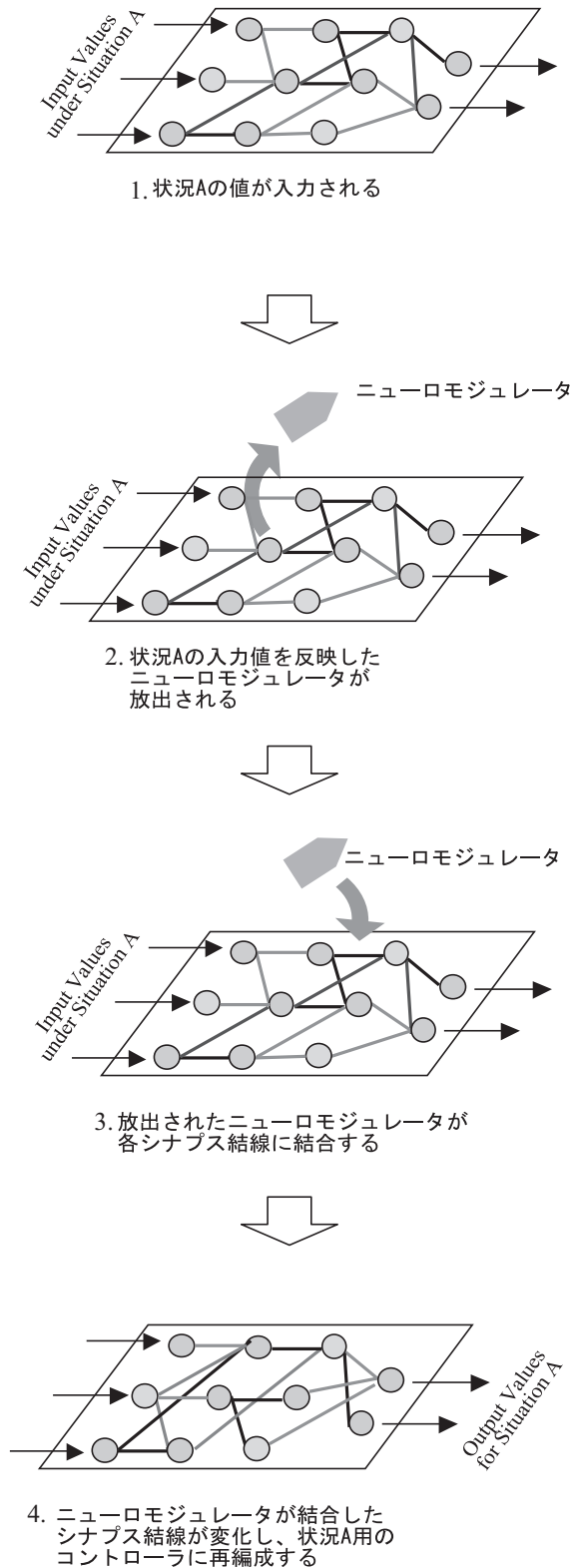


図 2.4: 実時間神経修飾機構付きニューラルネットワークのコントローラ再編成の流れ

プス結合は放出されたニューロモジュレータに反応するニューロモジュレータ反応部を有するものとした。そして、ニューロモジュレータ反応部において、それぞれが保持しているパラメータと放出されたニューロモジュレータの放出パターンに応じて、シナプス荷重を動的に変更できることとした。

これらニューロンにおけるニューロモジュレータ放出部とシナプス結合におけるニューロモジュレータ反応部を有することで、コントローラは状況に応じてニューロモジュレータの放出パターンを動的に変更することが可能となる。それと同時に、状況を反映し、動的に変化するニューロモジュレータの放出パターンに応じて、各シナプス荷重が変更することができることとなる。図 2.4 にモデル化した実時間神経修飾機構を有するニューラルネットワークのコントローラ再編成の流れを示す。図 2.4 において、ある状況 A における入力コントローラに入力される（図 2.4 中の 1）。各ニューロンから、その入力に応じたニューロモジュレータを放出する（図 2.4 中の 2）。状況 A を示す放出されたニューロモジュレータは、各シナプスに結合する（図 2.4 中の 3）。各シナプスは、結合したニューロモジュレータの状態により、荷重変更などの特定の影響を受ける（図 2.4 中の 4）。その結果、各状況に対応した専門のコントローラに変化し、行動が出力される。

次に、このニューロンにおけるニューロモジュレータ放出部と、シナプス結合におけるニューロモジュレータ反応部について詳細に述べる。その後、ロボットの自律的行動設計手法、つまり、「どのような状況」で「どのようなコントローラに変化させるか」を自律的に決定する手段について示す。

状況に応じた神経修飾物質ニューロモジュレータの放出

ここでは、ロボットの直面している状況を反映したニューロモジュレータの放出パターンを生成する、各ニューロンが持つニューロモジュレータ放出部について述べる。

各ニューロンは、ニューロモジュレータ放出部を持つものとし、それぞれの活性度と保持している条件に応じて、複数種あるニューロモジュレータのうち、決められた種類のニューロモジュレータを実時間で動的に放出（ある

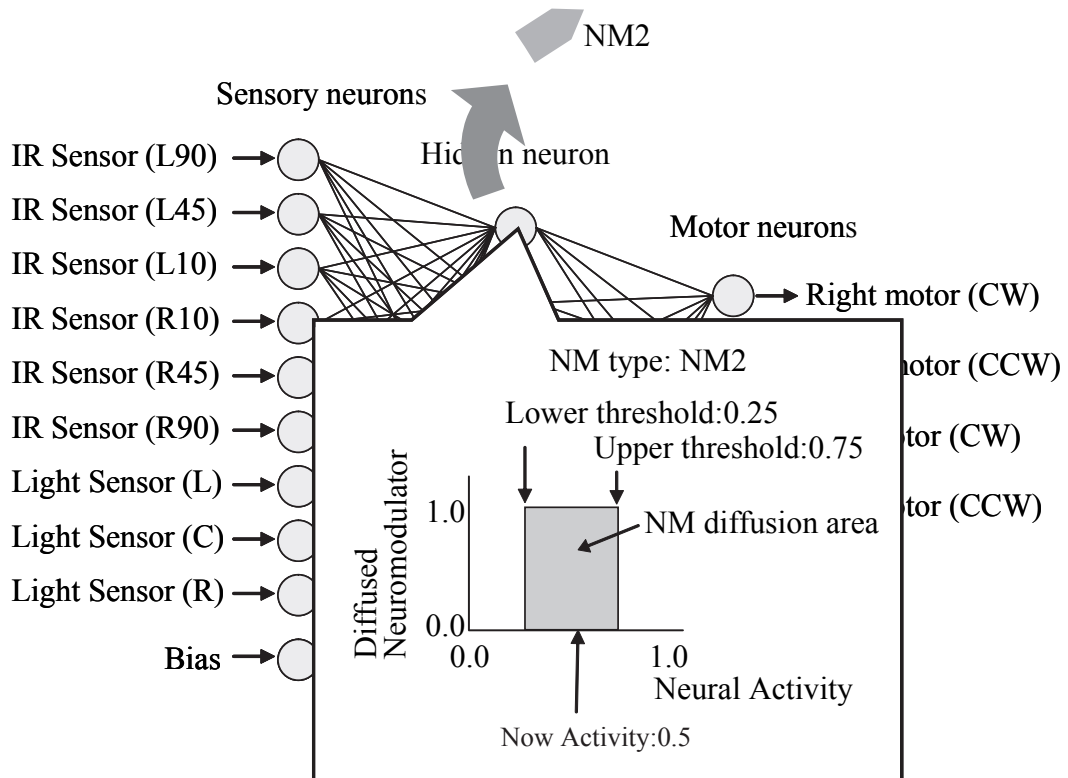


図 2.5: 2種類目のニューロモジュレータの放出権利を有し、放出するニューロン活性度下限条件と上限条件がそれぞれ、0.25 と 0.75 の設定のニューロンが活性度 0.50 を取った時のニューロモジュレータ放出の例

いは停止)するものとした。ここで、状況に応じてニューロモジュレータの放出パターンを変化させるために、ニューロモジュレータ放出条件のパラメータを各ニューロモジュレータ放出部が有するものとした。具体的には、ニューロモジュレータを放出する条件であるニューロンの活性度の上限値と下限値をパラメータとしてそれぞれの放出部毎に有するものとした。加えて、放出するニューロモジュレータの種類をもパラメータとして、それぞれの放出部毎に有するものとした。これら各ニューロンの放出部における、ニューロモ

ジュレータの放出に関する式は下記のとおりとした：

$$C_i^l(t) = \begin{cases} 1 & \left(\begin{array}{l} th_i^{under} < a_i(t) < th_i^{upper} \text{ かつ} \\ \text{ニューロン } i \text{ が } l \text{ 種類目の} \\ \text{ニューロモジュレータの放出権利を} \\ \text{有している場合} \end{array} \right) \\ 0 & \text{(上記以外)} \end{cases} \quad (2.1)$$

$$C^l(t) = \begin{cases} 1 & (\sum_{i=1}^{\text{ニューロン数}} C_i^l(t) \neq 0 \text{ の場合}) \\ 0 & \text{(上記以外)} \end{cases} \quad (2.2)$$

ここで、 $a_i(t)$ は、時間 t におけるニューロン i の活性度を示し、 th_i^{under} は、ニューロン i におけるニューロモジュレータを放出する活性度下限条件、 th_i^{upper} は、ニューロン i におけるニューロモジュレータを放出する活性度上限条件を示し、 $C_i^l(t)$ は、時間 t においてニューロン i が l 種類目のニューロモジュレータを放出したか否かを示す。そして、コントローラ全域における l 種類目のニューロモジュレータの放出の有無を示す $C^l(t)$ の算出は、式 (2.2) を用いた。

図 2.5 に、ニューロモジュレータ放出部におけるニューロモジュレータ放出例を示す。2 種類目のニューロモジュレータの放出権利を有し、放出するニューロン活性度下限条件と上限条件がそれぞれ、0.25 と 0.75 の設定のニューロンを考える。活性度 0.50 を取った時、ニューロン活性度が、それぞれの放出部が保持する条件 (活性度の上限と活性度の下限) を満たし、放出部は複数種類あるニューロモジュレータのうち、放出権利を有する 2 種類目のニューロモジュレータを放出することとなる。そして、このニューロモジュレータの放出判定を、式 (2.2) を用いて全てのニューロンに対し行うことで、その瞬間のネットワーク全域におけるニューロモジュレータの放出パターン (各種類のニューロモジュレータが放出されているか否か) が決定される。また、コントローラ全域におけるニューロモジュレータの放出パターンを見ることで、コントローラが置かれた状況を見ることが可能となる。

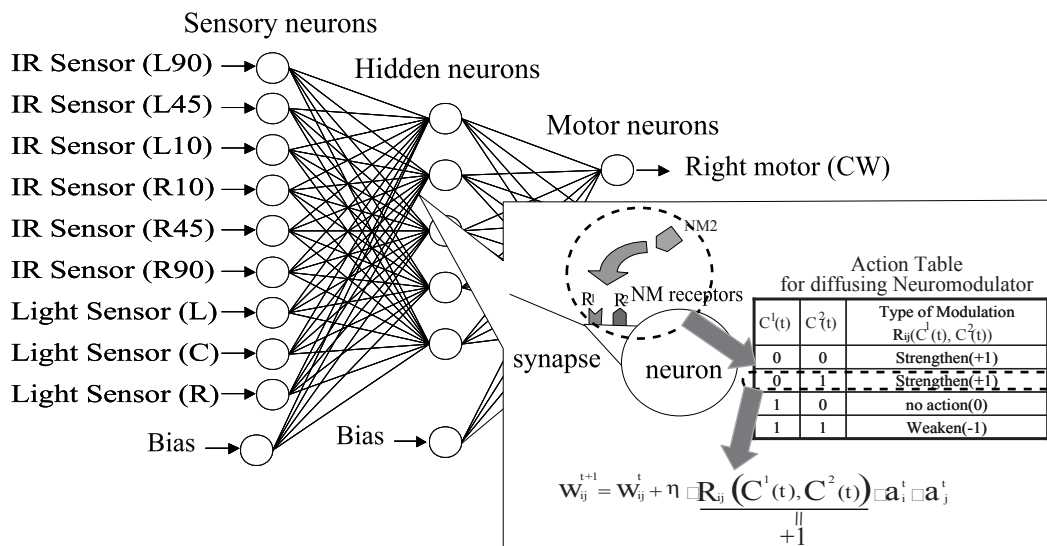


図 2.6: 2種類目のニューロモジュレータがだけ放出されている時、2種類目のニューロモジュレータだけ放出されている時のシナプス荷重の反応例が強化 ($R_{ij} = +1$) のルールを持っているニューロモジュレータ反応部の例

放出された神経修飾物質ニューロモジュレータに応じたコントローラの発現

次に、各シナプス結合それぞれが有するニューロモジュレータ反応部について説明する。

各シナプス結合は、状況に対応したコントローラを実時間で構成するために、コントローラの置かれた状況を反映したニューロモジュレータの放出パターンに応じて、シナプス荷重に変更を加えるニューロモジュレータ反応部を有するものとした。図 2.6 にニューロモジュレータの総種類数が 2 個の場合における、放出されているニューロモジュレータに対するニューロモジュレータ反応部の反応例を示す。加えて、各シナプス荷重に対する放出されたニューロモジュレータに対する反応式は下記の通りとした：

$$w_{ij}^{t+1} = w_{ij}^t + \eta \cdot R_{ij}(C^1(t), C^2(t) \dots C^n(t)) \cdot a_i^t \cdot a_j^t \quad (2.3)$$

この式において、 n はニューロモジュレータの種類数、 w_{ij}^t は時間 t におけるニューロン j からニューロン i へ繋がるシナプス結合の結線荷重を、 η は学習率を、

表 2.1: ニューロモジュレータの総種類数が 2 個の場合のニューロモジュレータに対するシナプス荷重の反応例

$C^1(t)$	$C^2(t)$	Type of modulation($R_{ij}(C^1(t), C^2(t))$)
0*	0	strengthen(+1)
0	1	strengthen(+1)
1**	0	no action(0)
1	1	weaken(-1)

* 0 means n -th neuromodulator is not diffused.

** 1 means n -th neuromodulator is diffused.

a_i^t は時間 t におけるニューロン i の活性度を意味する。また、 $R_{ij}(C^1(t), C^2(t))$ は放出されているニューロモジュレータの各放出パターンに対し、シナプス結合の強化、変化無し、弱화를それぞれ意味する $+1$ か 0 か -1 のいずれかをとるものとした。また、各シナプス結合におけるニューロモジュレータ反応部を持つ、ニューロモジュレータの解釈テーブルの例を表 2.1 に挙げる。

例えば、図 2.6 のように、表 2.1 と同じニューロモジュレータの解釈テーブルを持つニューロモジュレータ反応部を考える。各ニューロンが放出したニューロモジュレータの総和の結果、2 種類目のニューロモジュレータのみ放出している時 ($C^1 = 0, C^2 = 1$)、このニューロモジュレータ反応部は、解釈テーブルに基づき、 $R_{ij}(C^1(t), C^2(t)) = +1$ となる。これにより、式 (2.3) は:

$$w_{ij}^{t+1} = w_{ij}^t + \eta \cdot a_i^t \cdot a_j^t \quad (2.4)$$

となり、このシナプス荷重は $a_i^t \geq 0, a_j^t \geq 0$ なので、強化されることとなる。

この例に示したように、式 (2.3) と、反応部それぞれが保持しているパラメータと実時間で変化するニューロモジュレータの放出パターンにより、シナプス荷重を実時間で変更できることとした。これは、コントローラの入出力関係が状況に応じて実時間で変更できることを意味する。

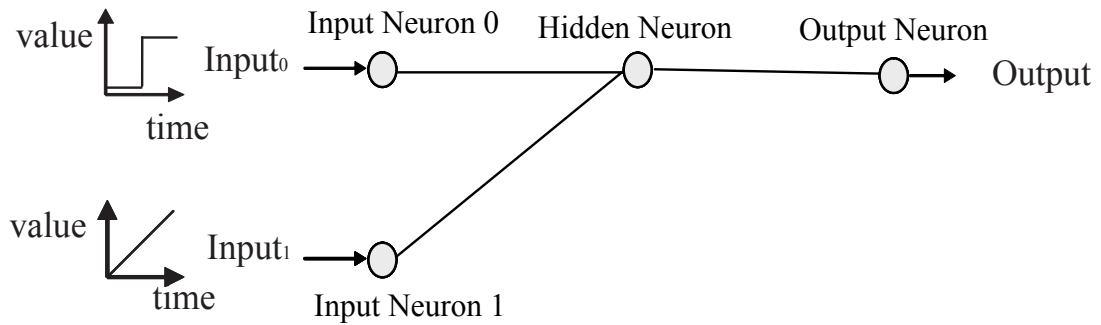


図 2.7: 実時間神経修飾機構を備えた 3 層フィードフォワード型ニューラルネットワークの例

2.2.3 実時間神経修飾機構による入出力特性の変化

ここで、実時間神経修飾機構を有するニューラルネットワークの入出力特性が入力に応じて大きく変化できることを確認する。

図 2.7 に示す 2 入力、1 出力の実時間神経修飾機構を有する 3 層フィードフォワード型ニューラルネットワークを考える。入力値 I_1, I_2 として次に示す値を入力することを想定する：

$$\begin{aligned}
 I_1 &= \begin{cases} 0.0 & (if \ t < 100) \\ 1.0 & (if \ t \geq 100) \end{cases} \\
 I_2 &= \begin{cases} \frac{t}{100} & (if \ t < 100) \\ \frac{t-100}{100} & (if \ t \geq 100) \end{cases}
 \end{aligned} \tag{2.5}$$

この式において、 t は 0 ~ 200 の整数値を取る。ここで簡単化のため、ニューロモジュレータの種類数は一種類、ニューロモジュレータの放出権利を有するニューロンは入力 I_1 を受ける入力ニューロンのみとした。

これらを含む実時間神経修飾機構を有するニューラルコントローラのパラメータ群を表 2.2 とした。その時のシナプス荷重の変化履歴とニューラルネットへの入出力履歴をそれぞれ図 2.8, 2.9 に示す。 I_1 の入力により、 I_1 の入力を受けるニューロンの活性度が 0.5625 を超える $t \geq 100$ では、ニューロモジュレータが放出されている。図 2.8 より、ニューロモジュレータが放出されて

表 2.2: 用いたの実時間神経修飾機構を有するニューラルネットワークのパラメータ

パラメータ	値
入力ニューロン個数	2 個
中間層ニューロン個数	1 個
出力層ニューロン個数	1 個
ニューロモジュレータの種類数	1 種類
ニューロモジュレータを放出できるニューロン	入力値 I_1 を受ける 入力ニューロンのみ
入力ニューロンがニューロモジュレータを放出するニューロンの活性度の下限値	0.5625
入力ニューロンがニューロモジュレータを放出するニューロンの活性度の上限値	1.000

いる時間であるグレー領域では、シナプス荷重がグレーでない領域と大きく異なる値を保持していることがわかる。ここで、ニューラルネットワークの入出力写像は、シナプス荷重の情報に依存しているため、シナプス荷重が図 2.8 に示される時間推移を行うことで、ニューラルネットワークの入出力写像は大きく変化することが、図 2.9 から確認できた。

2.2.4 実時間神経修飾機構のパラメータの自律的決定

次に、タスクに応じて「どのような状況」で「どのようなコントローラに変化させるか」を自律的に決定するために、先に述べたニューロモジュレータの放出条件と放出されたニューロモジュレータに対する反応の仕方についてのパラメータの決定方法に整理する。

先の項で、実時間神経修飾機構を有するニューラルコントローラを用いることで、入力履歴に応じて、入出力写像を大きく変化させることは確認でき

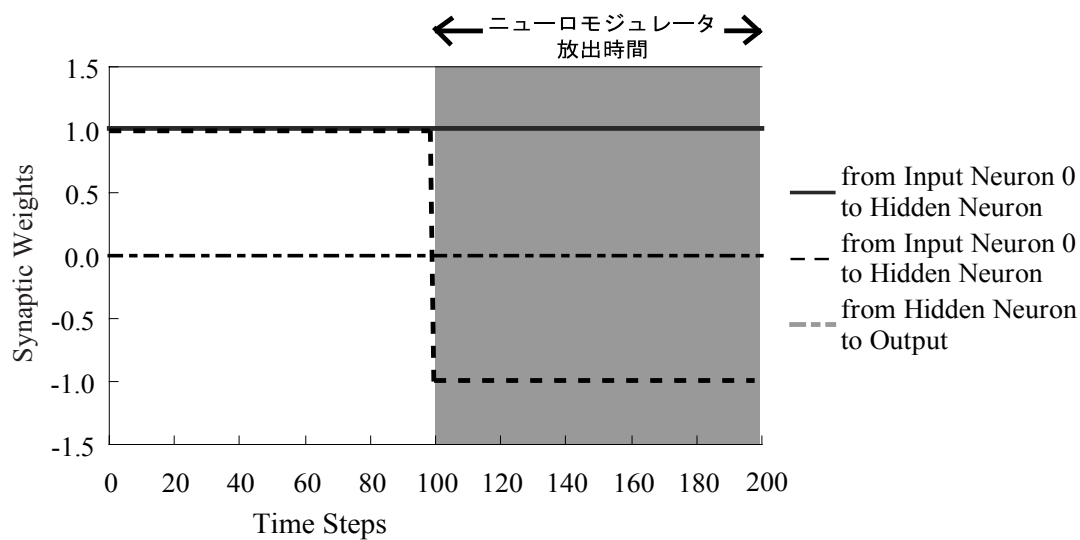


図 2.8: 実時間神経修飾機構を備えたニューラルネットワークのシナプス荷重の時間推移 (グレーの領域はニューロモジュレータが放出されている時を示す)

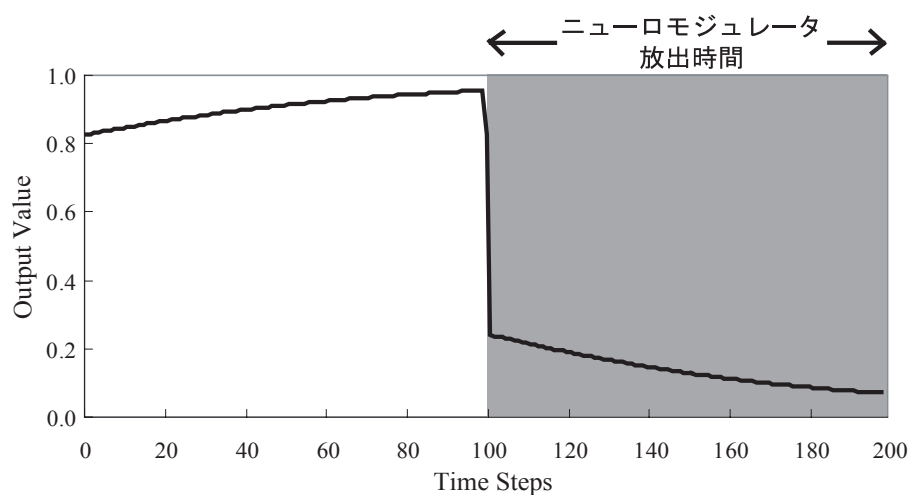


図 2.9: I_1, I_2 を入力値列として入力した時の実時間神経修飾機構を有するニューラルネットワークから出力された出力値 (グレーの領域はニューロモジュレータが放出されている時を示す)

た。しかし、これら入出力写像をタスクに応じて設計することは困難である。そこで、用いたパラメータを遺伝子に見立て、遺伝的アルゴリズムを用いて求めることとする。

求めるべきパラメータは下記の通りである：

- 各ニューロンにおけるニューロモジュレータ放出部において
 - － ニューロモジュレータを放出する権利を有するか？(0 または 1)
 - － 放出するニューロモジュレータの種類 (1, 2... n のいずれか。ただし n はニューロモジュレータの種類数)
 - － ニューロモジュレータを放出できる条件 (ニューロン活性度に対する下限値と上限値)
- 各シナプス結合におけるニューロモジュレータ反応部において
 - － 放出されたニューロモジュレータのパターンに応じたシナプス荷重の変更の仕方 (-1, 0, +1 のいずれか)

これらのパラメータを遺伝子に見立て、環境内において一定時間行動させたロボットの行動結果を評価することで、タスク達成するために必要な行動系列を算出するコントローラを進化的に獲得させる。これにより、タスク達成に必要な「行動を発現するコントローラ」と、「コントローラの形成を促すニューロモジュレータの放出条件」の組が進化を通して自律的に構築される。これはつまり、タスクに応じて「どのような状況」で「どのようなコントローラに変化させるか」を自律的に決定することとなる。

2.3 ペグ運び問題を行うコントローラの自律的構築

2.3.1 ペグ運び問題

次に、ここまで説明した提案手法を検証する例題として、図 2.10, 2.11 に示すようなペグ運び問題を取り上げる。環境にはロボット、ペグ、光源が存在す

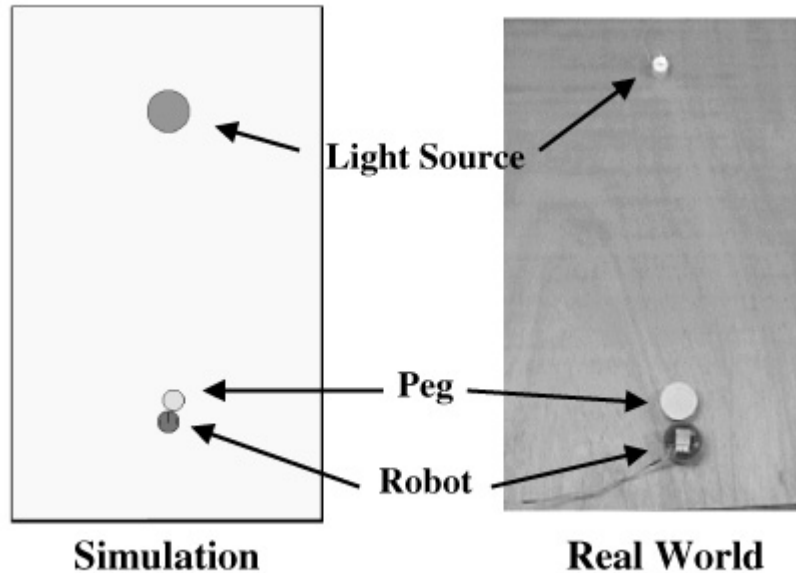


図 2.10: ペグ運び問題

る。タスクは、ペグと呼ぶ木材で作った円柱を、円筒形のロボットがゴールまで押し進めることである。ここで本研究では、初期位置において、ロボットとペグは接触しているものとした。これは進化過程を通して獲得された個体の環境変化に対する頑健性を示すことが、本研究の主たる目的としているため、ペグの探索行動の発現を必要としないためである。

本論文では実機として自律移動ロボット *KheperaTM* を用いた。ロボットの構造を図 2.12 に示す。ロボットには、ペグを検出するための赤外線センサ 6 個と、ゴールである光源の方向を検知するための光センサ、2 個の独立した DC モータを想定している。ロボットのコントローラとしては、図 2.13 に示すように、3 層フィードフォワードニューラルネットワークを用いた。ここで、各出力ニューロン 2 個の差を取り、DC モータに渡すことで、それぞれの車輪が独立して、正転・逆転することを可能にした。また、シミュレーション環境においては、実機のセンサ特性に近いものを必要とするため、実際に

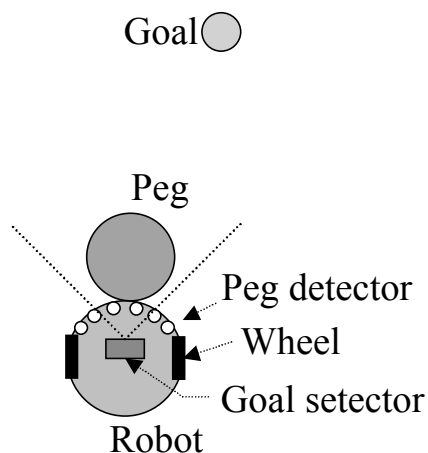
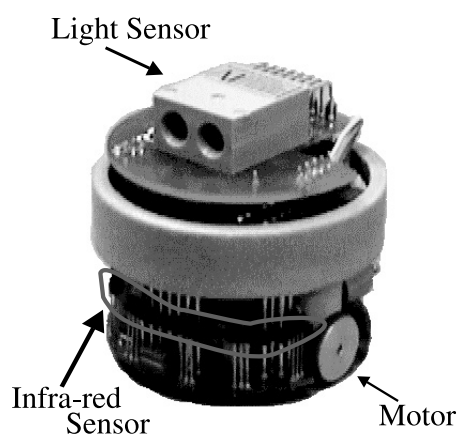


図 2.11: 実験で用いたロボットの仕様

図 2.12: *Khepera*TM

環境内に *Khepera*TM とペグを置き、各相対角度、各相対距離に対するセンサ値の平均値と分散値を測定し、シミュレーションに反映させた。一方、光センサは光源がロボットの正面 $\pm 18^\circ$ の範囲内に存在する場合、その方向（ロボット正面を 0° とし、右回りを正とする角度 θ ）を検知することができるものとした。加えて、光センサの値（すなわち光源の方向）をコントローラの3つの入力ニューロン（各々Left, Center, Right とラベルをつけた）の活性

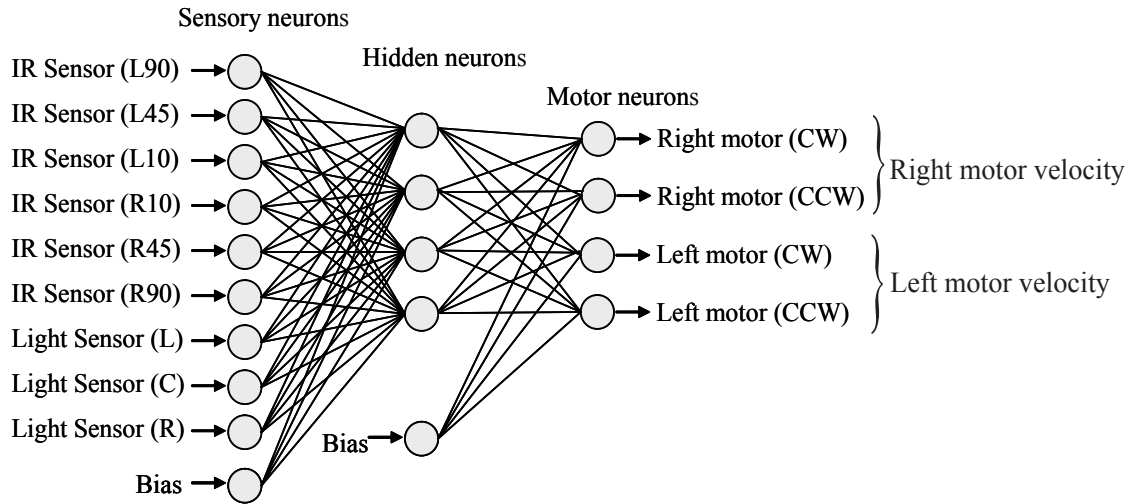


図 2.13: 実験で用いた 3 層フィードフォワードニューラルネットワーク

度に連続的に反映させるために、以下のような変換を行なった：

$$\begin{aligned}
 \text{LightLeft} &= \begin{cases} 1.0 & (\text{if } -18^\circ < \theta < -4^\circ) \\ 0.0 & (\text{otherwise}) \end{cases} \\
 \text{LightCenter} &= \begin{cases} 1.0 & (\text{if } -11^\circ < \theta < 11^\circ) \\ 0.0 & (\text{otherwise}) \end{cases} \\
 \text{LightRight} &= \begin{cases} 1.0 & (\text{if } 4^\circ < \theta < 18^\circ) \\ 0.0 & (\text{otherwise}) \end{cases}
 \end{aligned} \tag{2.6}$$

2.3.2 進化フェーズ

進化過程においては、2.2.3 で述べたように、各ニューロンの放出部におけるニューロモジュレータの放出権の有無、ニューロモジュレータの放出閾値、放出されるニューロモジュレータの種類、そして各シナプス結合のニューロモジュレータ受容体のニューロモジュレータの解釈テーブルを進化の対象とした。上記の情報を染色体上にコーディングし、遺伝的アルゴリズムを用いて適切な個体の生成を試みた。また、提案手法の妥当性を示すために、比較

対象として、ニューラルネットワークの静的なシナプス荷重とニューロンの閾値のみを進化の対象とした同様の3層フィードフォワード構造を有する従来手法も併せて検証した。なお比較する際に、進化におけるパラメータ探索空間の広さを同様にするために、本実験では、実数部分のbitコーディング長を調整することにより、両手法を用いたコントローラの遺伝子長をほぼ同じにした。

また、進化過程において各個体の評価値は、下記の式で表現される評価関数を用いて導出した：

$$fitness = \left\{ 1 - \frac{dist(light, peg_{end})}{dist(light, peg_{start})} \right\}^2 \times 100 \quad (2.7)$$

ここで、 $dist(light, peg_{start})$ は光源とペグの初期位置間の距離、 $dist(light, peg_{end})$ は、光源とペグの最終位置間の距離を表す。この式は、ロボットが最終的にペグを光源に近づければ近づけるほど評価値が得られる式となっている。この評価式(2.7)を用い、図2.10に示す環境において、ロボットとペグ、ロボットと光源の相対角度をランダムに変化させた初期位置からスタートさせ、500ステップ動くか、ペグが光源に到達するまでを1回とする試行を5回試行し、その平均評価値をその個体の評価値としてコントローラを進化をさせた。

2.3.3 仮想環境から実環境への移行

先に説明した環境において、両手法とも十分に評価値が上がり切るまでシミュレーション環境で進化させた(図2.14)。図2.15に、最終的に得られた両手法の最良個体(遺伝子)の、シミュレーション環境での行動軌跡を示す。図より、両手法ともコントローラ構築時に使った環境であるシミュレーション環境下では、ペグを光源まで押すことが確認できた。

次に、シミュレーション上で適切に動くことが確認できた両個体を実機にて検証した。その結果を図2.16と図2.17に示す。静的なシナプス荷重を進化の対象としている従来手法で得られた個体は、シミュレーション上では適切に動作した個体を用いても、実ロボット上に実装した場合、図2.16に示すように光源までペグを押すことができなかった。これは、シミュレーション環

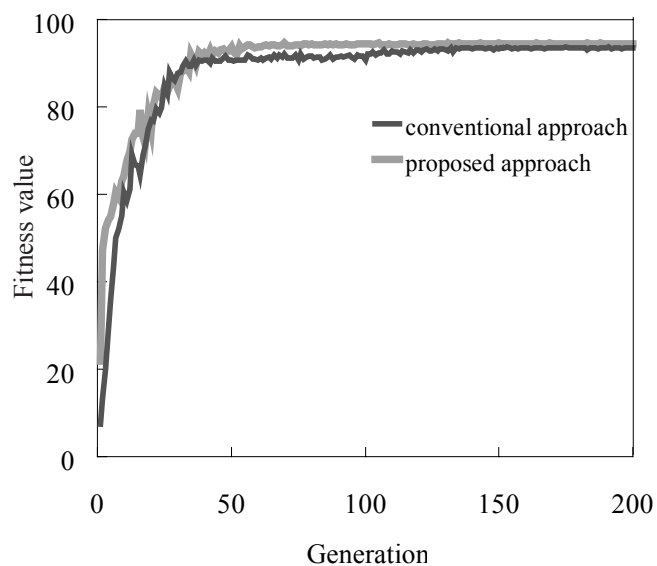


図 2.14: 進化時の評価値の推移

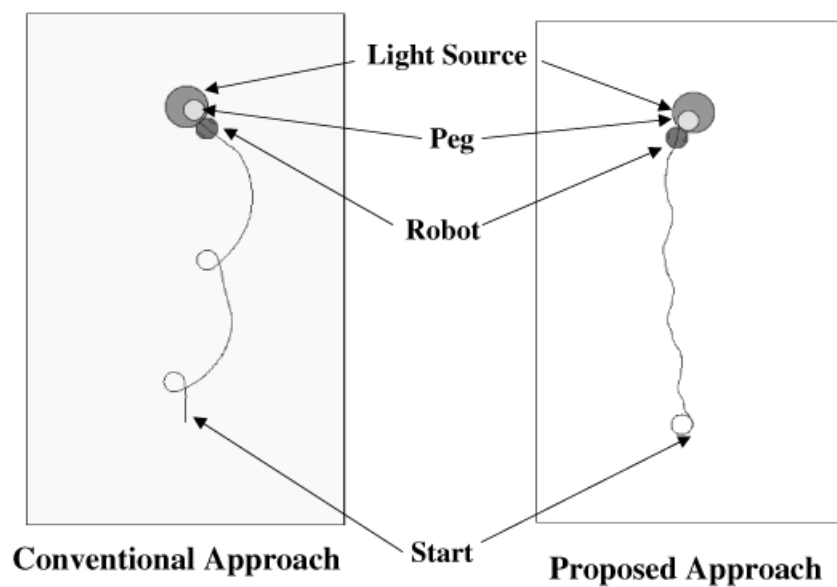


図 2.15: 従来手法と提案手法のシミュレーション環境での結果



図 2.16: 実環境での従来手法を用いたロボットの軌跡

境と実環境では、ペグと床の摩擦やペグとロボットの摩擦が異なり、ロボットがペグを押し進めた際のペグのずれや回転量が異なるなど、環境の差が原因である。それに対し提案手法では、図 2.17 に示すように、ロボット自身の体を小刻みに左右に揺らし、ペグが大きくずれないようにしつつ、光源まで適切にペグを押し進めることを確認した。

2.3.4 提案手法の実環境変化への適応

次にこの項では、図 2.17 に示した実環境において適切に動いた提案手法の個体を、進化段階では経験させていない環境下でも適切に動いた例を示す。加えて、この個体が置かれた状況に応じてコントローラを実時間で変更させることで、適切に動いた結果を示す。

図 2.18 は、ペグを進化段階で用いた直径 6cm のペグから未経験の直径 4cm のペグに変えた時のロボットの軌跡を示している。図に示すように、ペグの大きさを変えても適切に動くことが確認でき、提案手法の有効性が確認でき

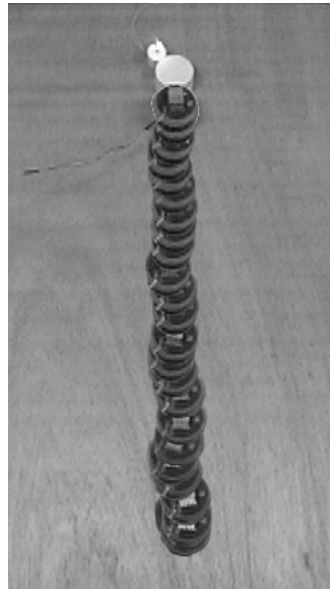


図 2.17: 実環境での、提案手法を用いたロボットの軌跡

た。また、図 2.19 にて、進化環境とはスタート位置が異なる時のロボットの軌跡を示す。図からわかるように、スタート位置が変わっても適切に動くことが確認できた。

ここで、提案手法が環境に応じてどの様に内部構造を変化させているかを見るために、初期状態でロボットの向いている方向が違う場合の提案手法におけるロボットの軌跡および、シナプス荷重、モータ出力、ニューロモジュレータの放出状況の時間推移の例を見せる。図 2.20,2.21,2.22 はそれぞれ、ペグとゴールに対し時計回りに約 60 度ロボットを回転させた状態からスタートした場合の行動軌跡と、シナプス荷重、モータ出力の時間推移を示す。そして、図 2.23,2.24 はそれぞれ、ペグとゴールに対し時計回りに約 60 度ロボットを回転させた状態からスタートした場合における、45Time Step のその場左回転行動を行っている時のネットワークの状態と 130Time Step の正面方向にあるペグを前進行動で押し進める行動を行っている時のネットワークの状態を示している。なお、この両図のシナプス結線の破線はマイナスの荷重を示し、結線の色の濃度は荷重の大きさを示す。加えて、図 2.25,2.26,2.27 はそ

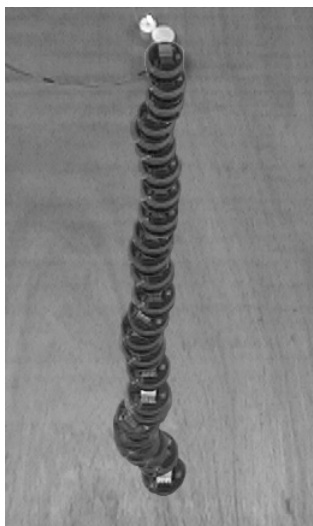


図 2.18: 進化段階で経験させなかった小さいペグ（直径 4cm）に変えたときの提案手法を用いたロボットの軌跡

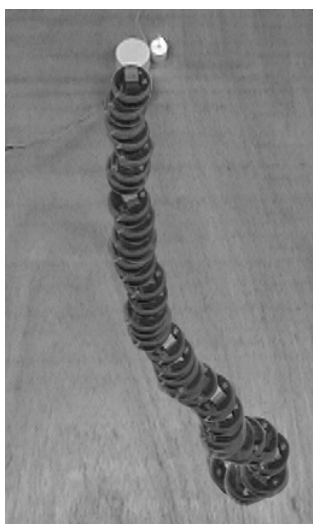


図 2.19: 進化環境とは異なる位置からスタートさせた時の提案手法を用いたロボットの軌跡

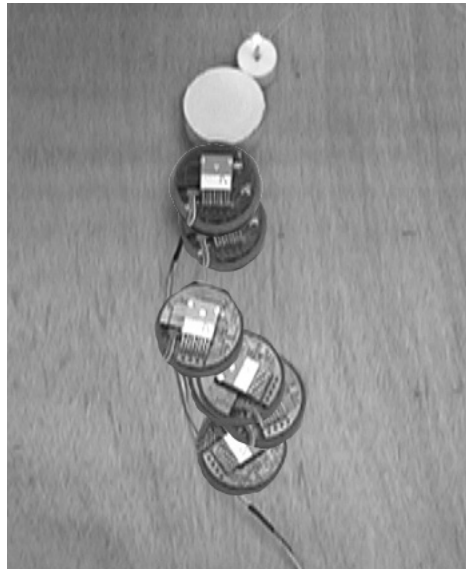


図 2.20: ペグとゴールに対し時計回りに約 60 度回転させた状態からスタートした場合のロボットの軌跡

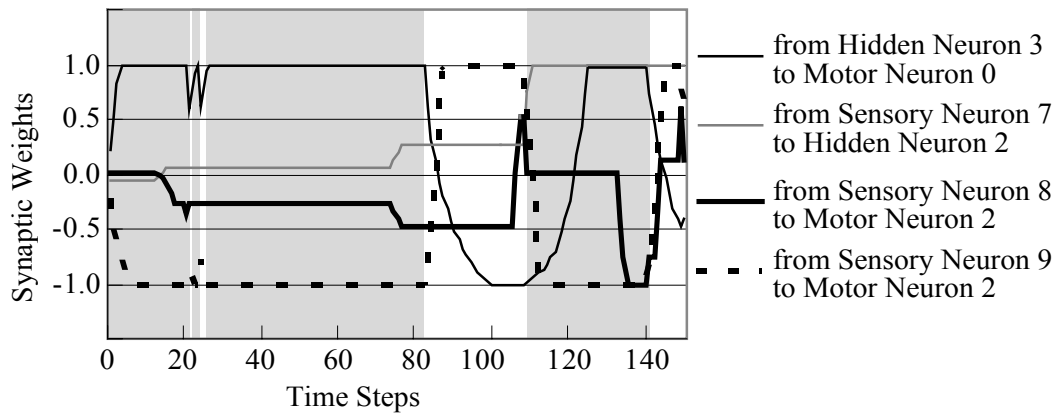


図 2.21: ペグとゴールに対し時計回りに約 60 度回転させた状態からスタートした場合の一部のシナプス荷重の時間的变化 (グレーの領域は 2 種類目のニューロモジュレータが放出されていた時を示す)

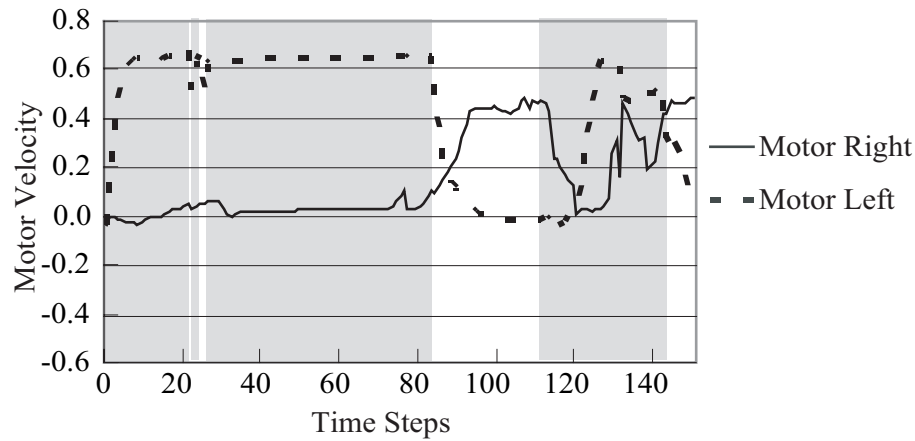


図 2.22: ペグとゴールに対し時計回りに約 60 度回転させた状態からスタートした場合のコントローラのモータ出力の時間的变化 (グレーの領域は 2 種類目のニューロモジュレータが放出されていた時を示す)

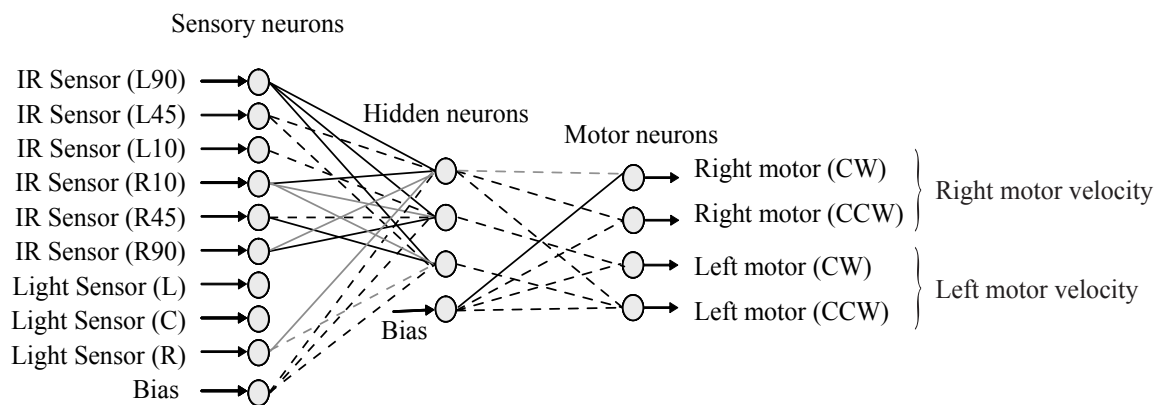


図 2.23: ペグとゴールに対し時計回りに約 60 度回転させた状態からスタートした場合の 45Time Step 後における、その場左回転行動を行っている時のネットワークの状態 (シナプス結線の破線はマイナスの荷重を示し、結線の色の濃度は荷重の大きさを示す)

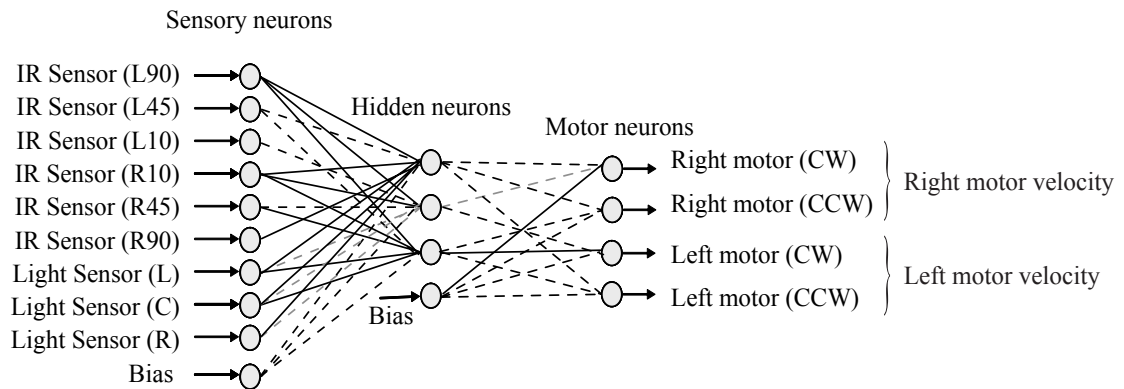


図 2.24: ペグとゴールに対し時計回りに約 60 度回転させた状態からスタートした場合の 130Time Step 後における、正面方向にあるペグを前進行動で押し進める行動を行っている時のネットワークの状態（シナプス結線の破線はマイナスの荷重を示し、結線の色の濃度は荷重の大きさを示す）

それぞれ、ロボットがペグとゴールに対し正対した状態からスタートした場合のロボットの軌跡とシナプス荷重、モータ出力、ニューロモジュレータの放出状況の時間推移である。そして、図 2.28 はこの場合の、73Time Step における正面方向にあるペグを前進行動で押し進める行動を行っている時のネットワークの状態を示している。なお、図 2.21,2.26 では、見やすさのため、一部のシナプス荷重推移のみ記している。

図 2.25,2.27 より、ロボットは後退することでペグやゴールとの位置関係の修正行動を行なった後、ペグを正面で捕らえてゴールまで押し進めていることが解る。また、図 2.20,2.22 の場合、ロボットはまずその場左回転行動により、ペグとゴールを正面にとらえなおし、ペグをゴールまで押し進める動作に移っているのが解る。また、図 2.23,2.24 より、同じコントローラにおいて、ニューラルコントローラのシナプス荷重の状態を変動させ、入出力関係を変更させることで、発現する行動を変化させていることが解る。

一方、シナプス荷重とモータ出力とニューロモジュレータの放出状況の推移を比較すると、図 2.21 の 80 Time Step 付近と図 2.26 の 35 Time Step 付近

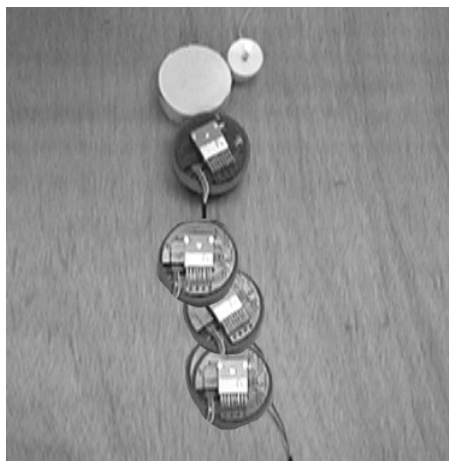


図 2.25: ペグとゴールに対し正対した状態からスタートした時のロボットの軌跡

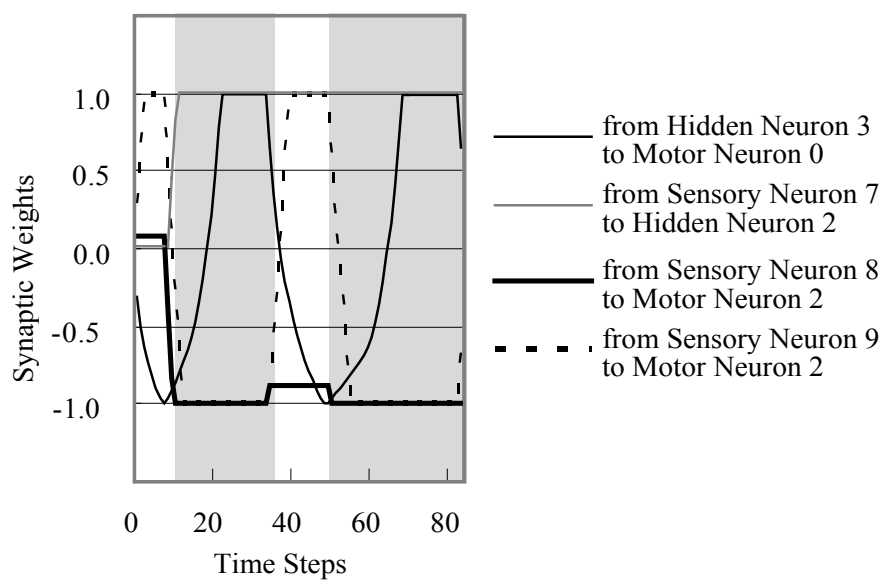


図 2.26: ペグとゴールに対し正対した状態からスタートした時の一部のシナプス荷重の時間的変化（グレーの領域は2種類目のニューロモジュールが放出されていた時を示す）

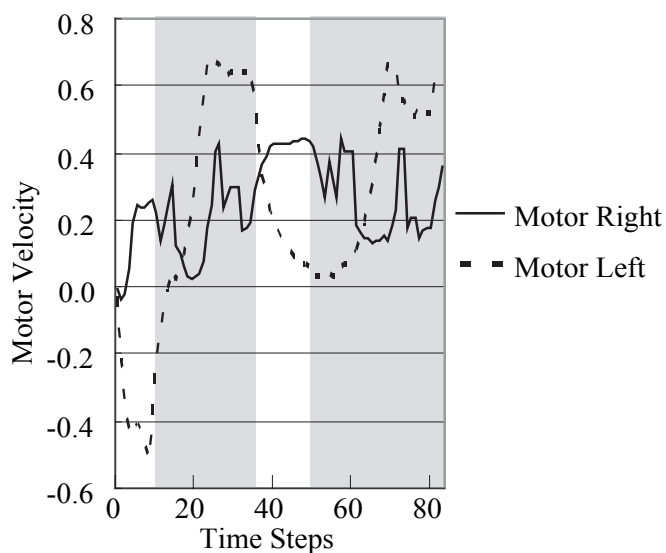


図 2.27: ペグとゴールに対し正対した状態からスタートした時のコントローラのモータ出力の時間的变化 (グレーの領域は2種類目のニューロモジュレータが放出されていた時を示す)

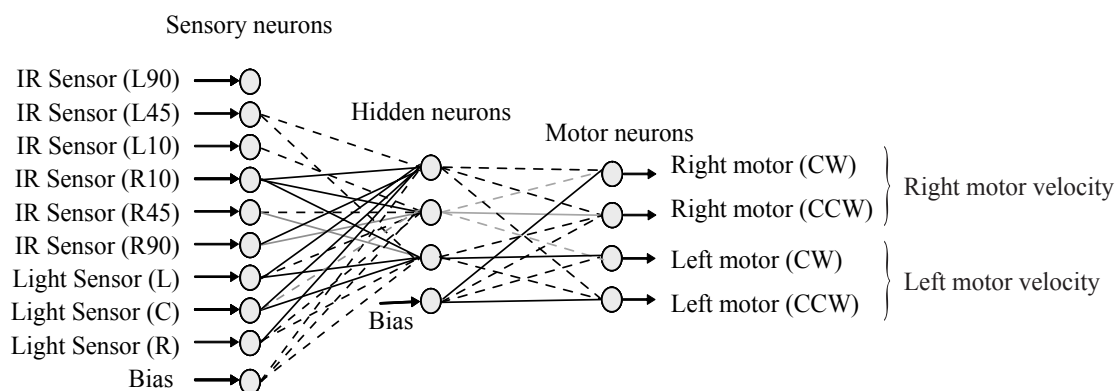


図 2.28: ペグとゴールに対し正対した状態からスタートした場合の 73Time Step 後における、正面方向にあるペグを前進行動で押し進める行動を行っている時のネットワークの状態 (シナプス結線の破線はマイナスの荷重を示し、結線の色の濃度は荷重の大きさを示す)

から、似た荷重変動が見られる。これは、図 2.20,2.25 においてそれぞれペグと光源に対し、ロボットが正対し、押し始めるといった似た行動を行なった時に対応する。これらの、正面にペグとゴールを捕らえつつ、似た行動である正面方向でペグを押し行動を起こした際のネットワークの状態が図 2.24,2.28 である。両図から見て解るように、初期状態が異なっている場合でも、同様のシナプス荷重で形成されたコントローラを発現し、ペグをロボット正面で押し進める行動を引き起こすことでタスク達成を行っていることが解った。このことから提案手法は、異なる環境（今回では異なる初期状態）に置かれても、それぞれ状況に応じたコントローラを適時発現させ行動することで、頑健にタスクが達成できたことが解った。

2.4 まとめ

この章では、実時間神経修飾機構を有するニューラルネットワークを提案し、ペグ運び問題に適用した。静的なネットワーク構造を進化の対象とする従来のニューラルネットワーク構築手法では、進化環境であるシミュレーション環境に適応できても、実環境にシームレスにコントローラを移行できなかったことを示した。一方、提案手法を用いた場合、シームレスに移行できることを確認した。また、提案手法は、状況に応じてニューラルコントローラのシナプス荷重を変更させることで、状況に応じて入出力関係を変更し、発現する行動を変化させることで、頑健にタスク達成していたことを確認した。

第3章 FPGAを用いた実時間神経修飾機構の実現

3.1 はじめに

前章では、生物の動的に神経回路を変更する機能をモデル化することで、状況に応じて、入出力関係を改変できるコントローラを提案すると共に、シミュレーションから実環境へシームレスに移行できることを示し、提案手法の有効性を示した。同時に、内部の解析を行ない、コントローラが適時切り替わり、状況に対応している様子を示した。

一方、実用化に向けて自律ロボットを設計・構築するためには、コントローラや電源、駆動装置などロボットを構成するあらゆる部品に対し、タスクに合わせて適切なものを選択し、ロボット内部にすべての部品を収納する必要がある [11]。そのためには、消費電力の削減、構成部品の減少簡略化が必要となってくる。また、実環境のように不確実性が存在する環境下で利用するロボットには、利用する環境やタスクに対応するため、コントローラ改変の容易さも重要となってくる。これらの要求に対し、プログラマブルなハードウェアでコントローラを構築することは、消費電力の低減、構成要素の小型化、部品点数の削減、コントローラ改変の容易さなどに大きく寄与すると考えられ、一つの有効な手段である。

そこで、本章では前章で提案した実時間神経修飾機構を有するニューラルネットワークに対し、Field Programmable Gate Array (以下、FPGA と略す) を用いることでハードウェア化を行った Dynamically Rearrangement Electronic Circuit (以下、DREC と略す) を提案し、CPU を用いないにも関わらず、状況に応じて実時間で入出力特性を変更し、頑健にタスクを達成

するロボットを製作する。加えて、本手法の有効性を検証するために、モータ出力を解析することにより、コントローラの有効性を確認する。そして、製作したFPGA上のコントローラの回路規模を従来手法でハードウェア化した場合と比較し、本手法の有用性を述べる。

以下、3.2節ではニューラルネットワークのハードウェア化について述べ、3.3節において、2章で提案した実時間神経修飾機構のハードウェア化について述べ、3.4節で検証実験である実機実験について説明する。その後、3.5節において、他のニューラルコントローラのハードウェア化手法との回路規模比較などを通して考察を行い、最後に3.6節で結びとする。

3.2 ニューラルネットワークのハードウェア化

ここでは、ニューラルネットワークのハードウェア化について述べる。

ニューラルコントローラは下記のような特徴がある：

- モータ出力を決定する連続値やセンサ値など、低レベルの情報を直接扱いやすい
- 多入力多出力処理が容易である
- 並列処理に向いている
- 非線形な処理ができる
- 計算アルゴリズムに繰り返し部分が多い

これらの特徴に対し、繰り返し計算や多入力多出力、並列計算に向いているハードウェア化を行う手法が提案されている。例えば、Haycock等[51]は、パルスニューラルネットワークのハードウェア化を提案している。従来の多くのニューラルネットワークの研究は、非線形機能を有する閾値素子を多数用意し、それらを静的な結合荷重で結合し、その結合素子に情報を埋め込む手法を用いてきた。これに対し、パルスニューラルネットワーク手法は、ニューロンの平均発火頻度に情報を埋め込むことで、情報処理を行う手法である。

この手法は、より生物の神経回路の計算処理に近く、確率的に情報処理を行うため、耐ノイズ性が高いなどの長所を有している。しかし、パルスニューラルネットワークの手法は、学習手法や解析手法の一般化が十分に行われていない。

一方、非線形機能を有する閾値素子を多数用意し、それらを静的な結合荷重で結合するニューラルコントローラのハードウェア化においては、多くの研究で用いられているシグモイド関数と乗算計算において、多大な回路が必要とする問題が存在する。この問題に対し、シグモイド関数への入出力のマッピングをハードウェアのメモリ上に構築する手法や、シグモイド関数を近似することで、回路規模を縮小する手法が提案されてきた [52, 53]。しかし、これらの手法では、関数精度を下げることで、メモリ空間の消費を抑えるに留まっていた。

これに対し、川島ら [54, 55] は、シグモイド関数の関数形状を正規分布曲線の積分曲線と近似することで、回路規模の増大を抑えたシグモイド関数のハードウェア化を提案した (図 3.1、付録参照)。そこで、本研究では、川島らの提唱するこの確率的エンコーディング手法に対して、親和性の高い実時間神経修飾機構のハードウェア化手法 DREC を開発する。

3.3 実時間神経修飾機構のハードウェア化

この節では、前章で提案した実時間神経修飾機構を有するニューラルネットワークに対し、シグモイド関数のハードウェア化問題を考慮したハードウェア化手法である DREC を提案する。

前節で挙げた川島らの確率的エンコーディング手法を用いたニューラルネットワークに、前章で提案した実時間変更可能な実時間神経修飾機構を有するニューラルネットワークのニューロモジュレータの放出・反応メカニズムを実現する回路を追加した (図 3.2)。以下、放出部と反応部のハードウェア化について詳細に示す。

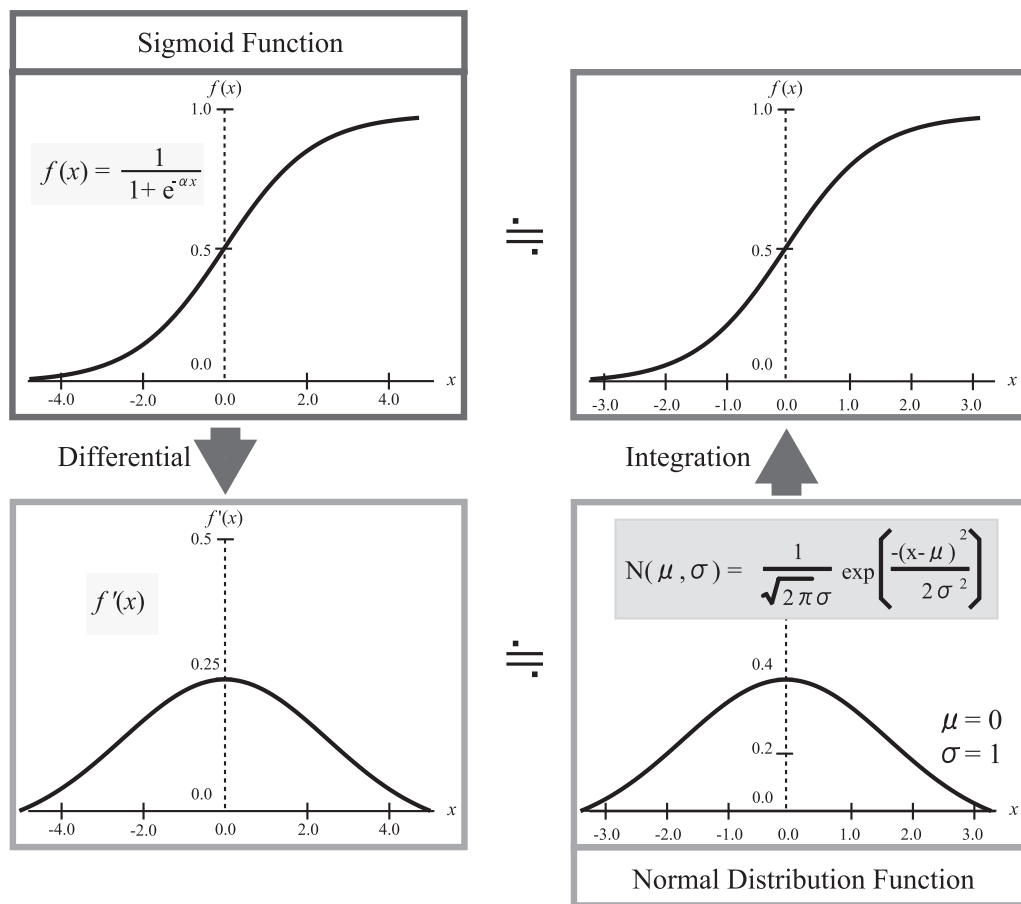


図 3.1: シグモイド関数の関数形状と正規分布曲線の積分曲線との関係

3.3.1 ニューロモジュレータ放出部のハードウェア化

具体的には、ニューロモジュレータの放出は、式(2.1),(2.2)に示すように、ニューロン j の活性度 a_j が放出閾値 $lower_j$ (下限値) および $upper_j$ (上限値) の間にあり、かつ放出権を有する時に行われる。したがって、これを回路で実現する際、活性度 a_j と $lower_j$ および a_j と $upper_j$ の比較は、それぞれ比較器を用いることで実装する。また、放出の有無は、2個の比較器の出力結果と放出フラグ $flag_j$ の論理積をとることで決定し、種類 n のニューロモジュレータの放出の有無は、 $type_j$ を入力に有する選択器を用いて c_j^n として出力する。そして、ネットワーク全体における n 番目のニューロモジュレータ NM_n の有無 C^n は、各ニューロンからの c_j^n の論理和をとることで決定する。

3.3.2 ニューロモジュレータ反応部のハードウェア化

ニューロモジュレータの反応は、式(2.3)に示すように、各シナプス結合が有するニューロモジュレータ解釈テーブル $R(C^1, \dots, C^n)$ と、シナプス結合前後に接続されたニューロンの活性度をもとにした積 $\tau_1 \cdot R(C^1, \dots, C^n) \cdot \frac{a_i+1}{2} \cdot \frac{a_j+1}{2}$ 、および現在のシナプス荷重と定数の積 $\tau_2 \cdot w_{ji}$ で表される。ここで、 τ_1 および τ_2 は $\frac{1}{2}$ の倍数を用いることで乗算器の回路規模を抑えて実装した。また、 $\frac{a_i+1}{2} \cdot \frac{a_j+1}{2}$ は川島らの確率的エンコーディング手法においても用いられているラプラスの定理を応用した乗算器を用いることで実現する。

3.4 実機実験

本節では、前節で述べたDREC(図3.2)を用いて、図2.13に示すコントローラを実装し、頑健性については先章と同様の手法で比較実験を行う。具体的には、比較対象として、実時間神経修飾機構を有さず、シナプス荷重が固定された従来型の3層フィードフォワードニューラルネットワークを用いた。そして、シミュレーション環境下において、各手法を用いたコントローラを進化的に構築した後に、実機におけるタスク成功率を比較する。また、行動

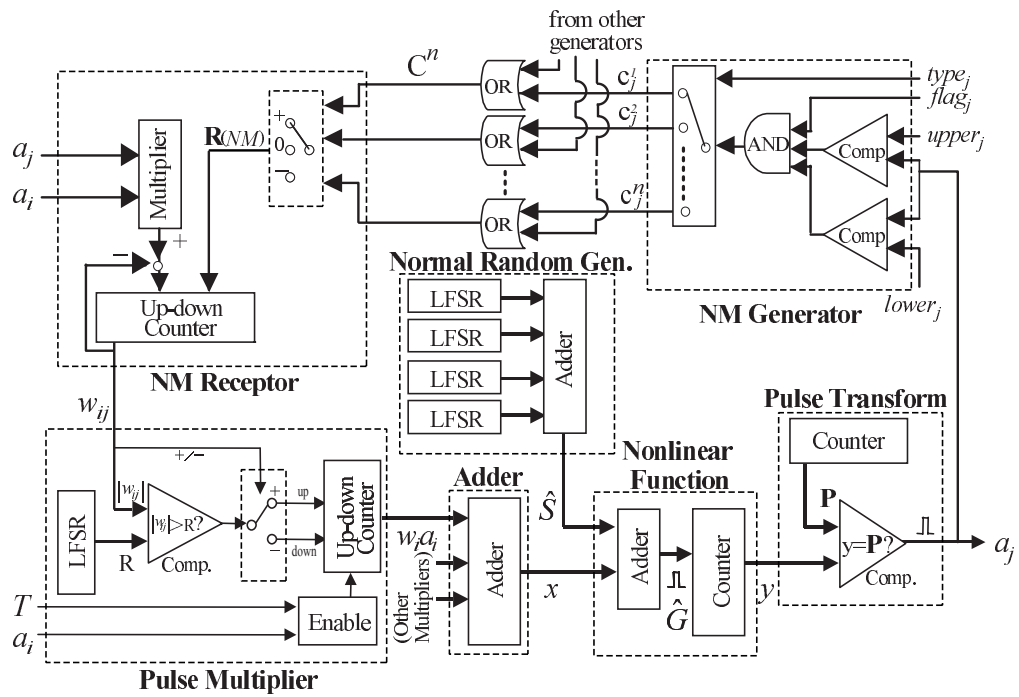


図 3.2: 構成した DREC によるニューロン一個分の回路ブロック図

表 3.1: 用いた遺伝的アルゴリズムのパラメータ

1 世代あたりの個体数	100
1 個体の 1 回の評価に行う試行回数	10
進化世代数	200
進化時に経験する初期状態におけるロボットとペグの相対角度	-2 ~ 2deg.
進化時に経験する初期状態におけるロボットとゴールの相対角度	-24 ~ 24deg.

中のモータ出力などを解析することにより、第 2 章で示したように行動を大きく切り替えて、各状況に対応していることを示し、自律移動ロボットのコントローラ構築に有効であることを示す。

3.4.1 実験環境

第 2 章と同様に、環境には円筒形の車輪型ロボット (図 3.3)、ペグならびにゴール (到達目標) が存在する。ロボットのタスクは、第 2 章同様ペグを到達目標に向かって押し進めることである。ここでロボットはセンサ系として、半導体像位置検出素子 (Position Sensitive Device:PSD) であるシャープ (株) 製 GP2-D12 を使用した距離センサ、三菱電機 (株) 製 M64270AG (人工網膜チップ) を用いたゴール方向を検出する角度センサの 2 種類を前方にのみ有する。駆動系としては、左右に車輪を備えており、それぞれに独立のモータが実装されている。そして、これらの信号は直接、ロボット最上部に搭載されている書き換え可能デバイスである FPGA (ALTERA EP20K 400EBC652-3: 400kgates) 上に実装した DREC に直接入力、出力している。

コントローラ構築時の進化は表 3.1 に示す環境下で行った。また、進化過程における各個体の評価値は、複数回試行し、下記の式で得られた評価値の平均値を用いた：

$$fitness = \left(1 - \frac{dist_{nearest}(peg, goal)}{dist_{init}(peg, goal)} \right)^2 \times 100 \quad (3.1)$$

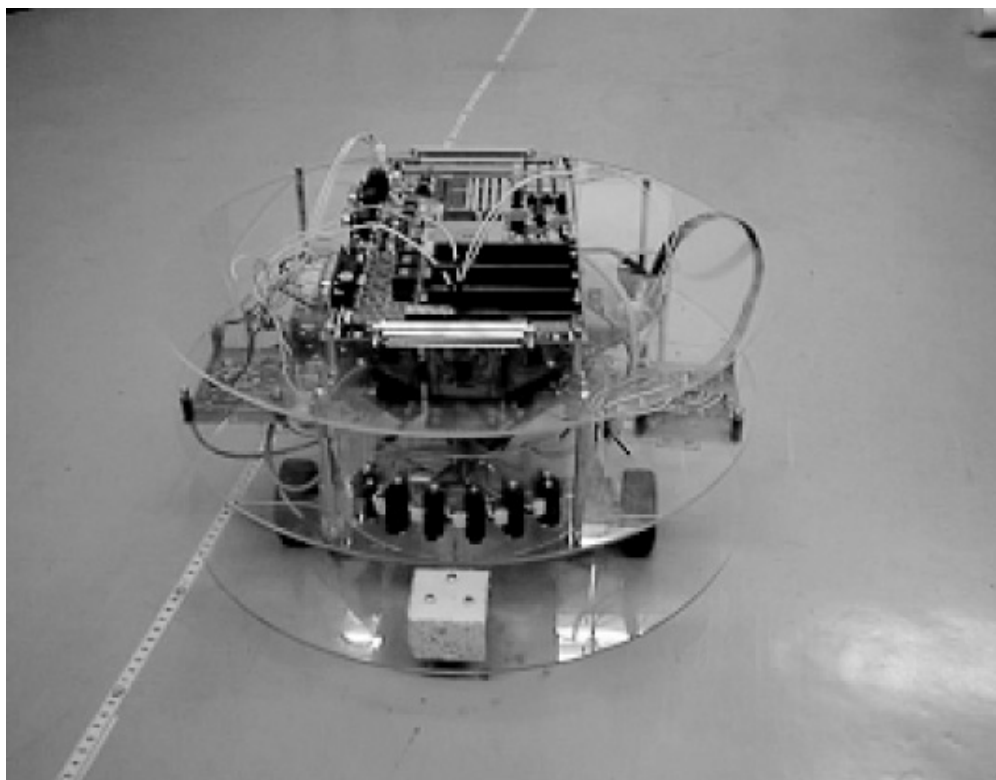


図 3.3: CPU レスロボット

ここで、 $dist_{init}(peg, goal)$ はタスクの初期状態におけるゴールの中心位置とペグの中心位置間の距離、 $dist_{nearest}(peg, goal)$ は、試行中におけるゴールとペグの最接近時のゴールの中心位置とペグの中心位置間の距離を示す。

加えて、従来の静的なシナプス荷重を進化の対象とした固定構造ニューラルネットワークを用いた従来手法のコントローラと提案手法のコントローラ、それぞれをシミュレーション環境において進化させた。その際の両手法の各世代に対する評価値の推移を図 3.4 に示す。また、図 3.5 に、実時間神経修飾機構を有するニューラルコントローラと、比較対象である固定構造のニューラルコントローラを進化させた最終個体のシミュレーション環境での行動結果をそれぞれ示す。両手法共に、進化環境であるシミュレーション環境にお

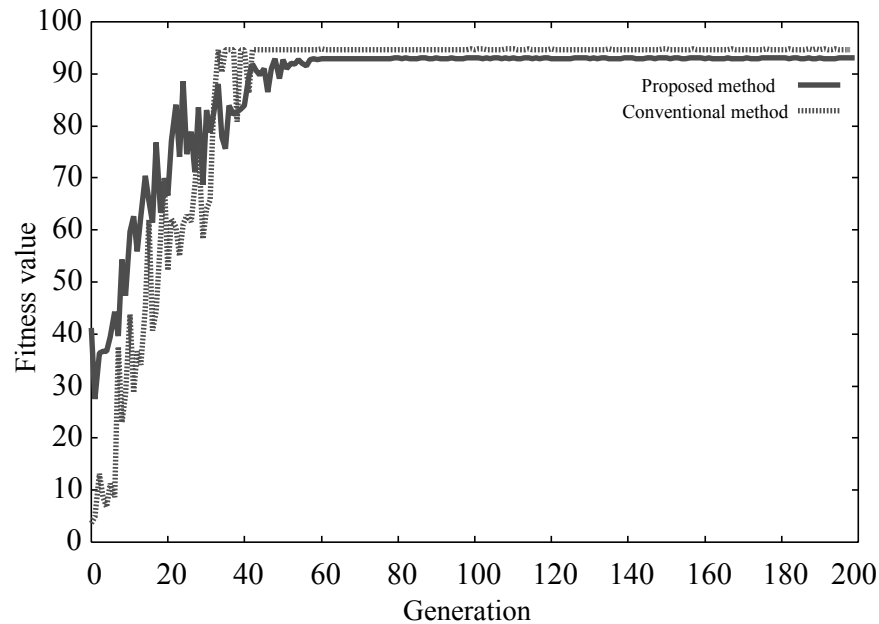


図 3.4: 進化時における評価値の推移

いては、適切にペグをゴールまで押し進めることができることを確認した。

3.4.2 実機実験による頑健性の検証

次に、シミュレーションを通じて得た進化最終個体データを制作した CPU レスロボットに移行し、実環境で実験した際の軌跡の一例を図 3.6, 3.7 にそれぞれ示す。

図 3.7 より、提案手法では、シミュレーション上で見られたような変曲点を複数持った軌跡を描きながら、タスクを達成できることがわかる。それに対し、図 3.6 の従来手法コントローラでは、シミュレーションとは曲率の異なる曲線を描き、曲がりきれずにゴールの横を通過し、タスク達成できなかったことが確認できた。

この行動中における、提案手法により実時間で変化した一部のシナプス荷重の時間推移を図 3.8 に、実環境上で試行した時の従来手法と提案手法のモー

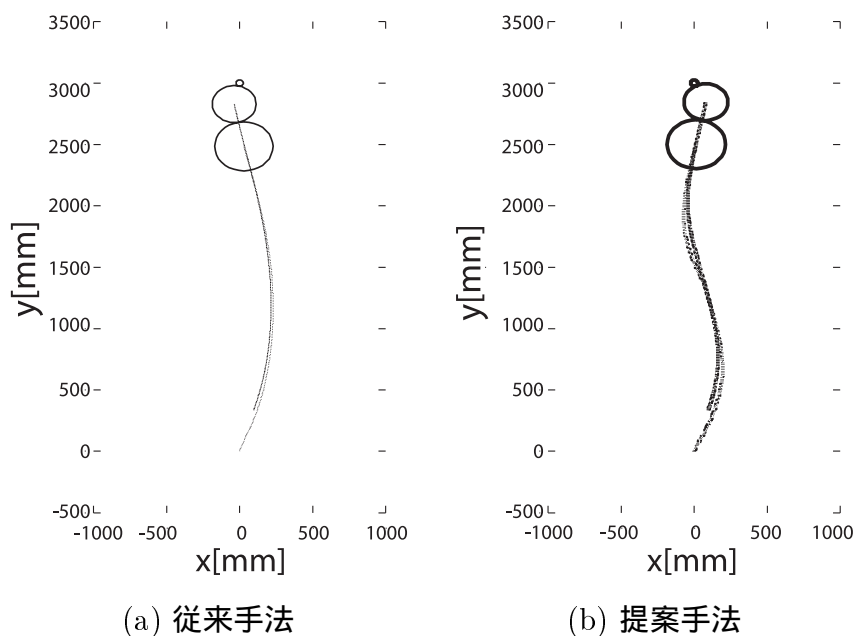


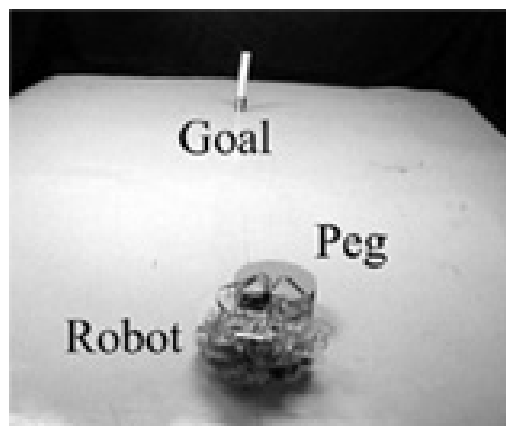
図 3.5: 進化後のシミュレーション環境における従来手法と提案手法それぞれのコントローラを用いたロボットの移動軌跡

タ出力推移を図 3.9,3.10 に示す。図 3.8 より、提案手法を用いる場合は、実環境においても 2 章と同様、シナプス荷重を大きく変化させていることが解る。これは、入出力関係を大きく変化させ、モータ出力の変化を促し、行動を切り替えていることを意味する。つまり、2 章での結果と合わせることで、FPGA でハードウェア化し、CPU を用いないコントローラにおいても、状況に応じてコントローラの入出力特性を状況に応じて大きく変化させ、環境に対処できていることを示唆している。

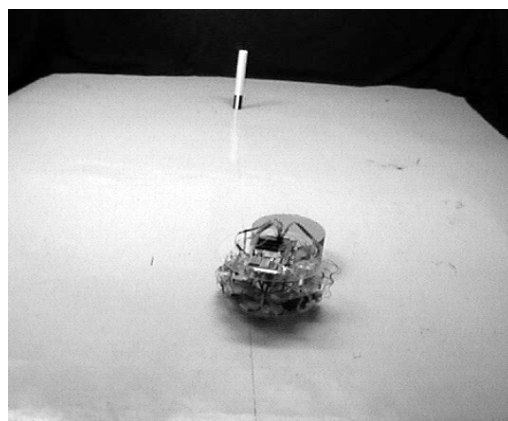
3.4.3 初期環境変動に対する頑健性の検証

この項では、タスク開始時の状況を変化させた時のタスク成功率を比較することで、提案手法の頑健性を検証する。

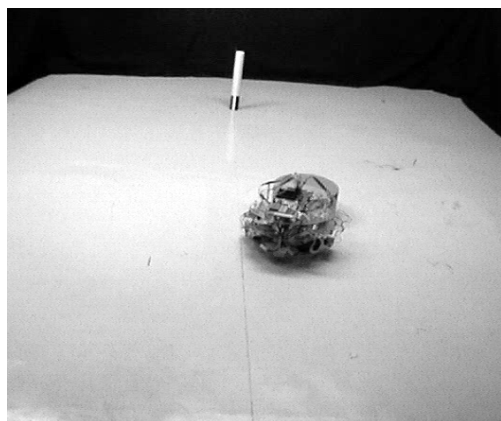
図 3.11、3.12 に、各手法を用いた CPU レスロボットにおいて、ロボットから見たゴールの初期角度を変化させた時のタスク成功率を測定した結果を



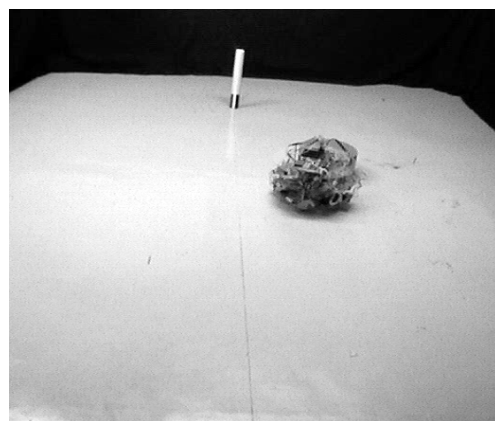
(1) 0 sec



(2) 16 sec



(3) 32 sec



(4) 48 sec



(5) 60 sec



(6) 80 sec

図 3.6: 静的なシナプス荷重を進化の対象とした従来手法のコントローラを搭載した実機の移動軌跡

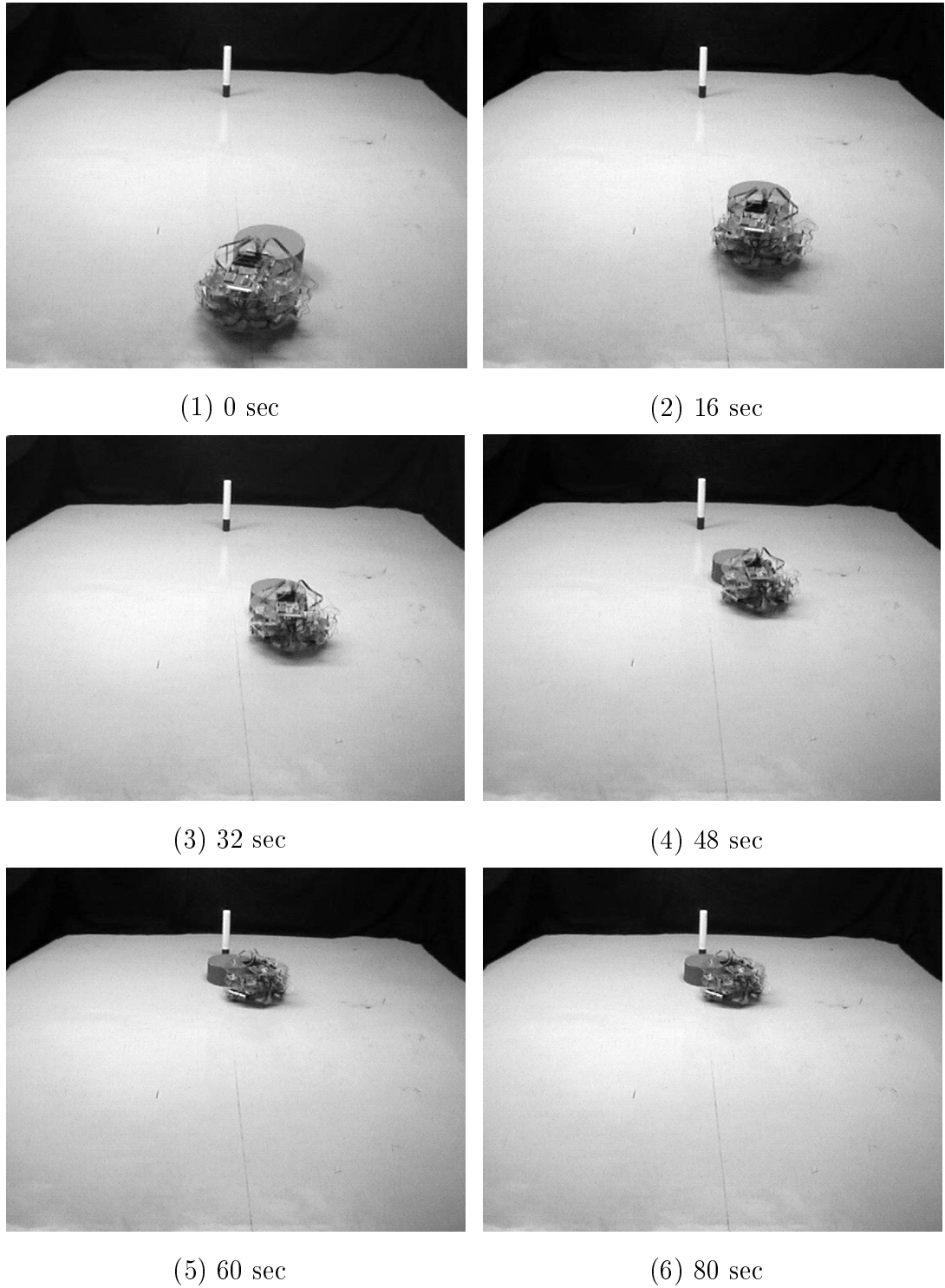


図 3.7: 提案手法を用いた実機の移動軌跡

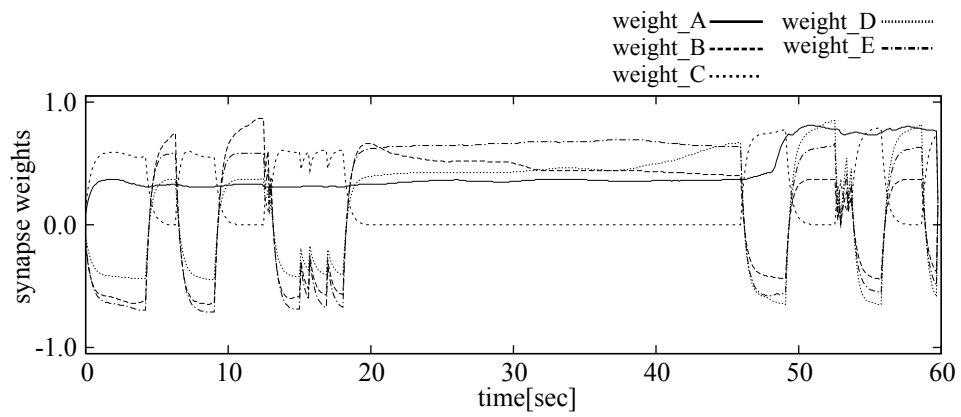


図 3.8: 提案手法コントローラの実環境行動時のシナプス荷重の時間推移例

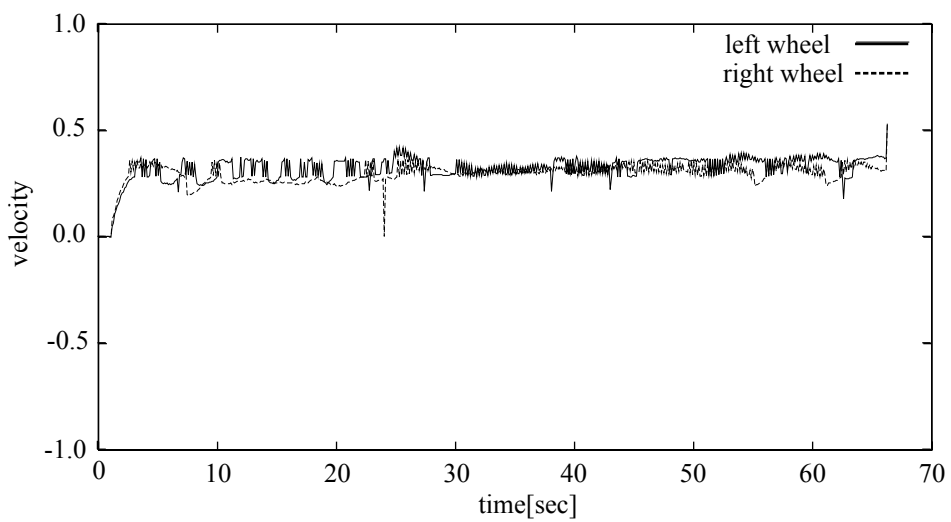


図 3.9: 従来手法コントローラの実環境行動時におけるモータ出力の時間推移

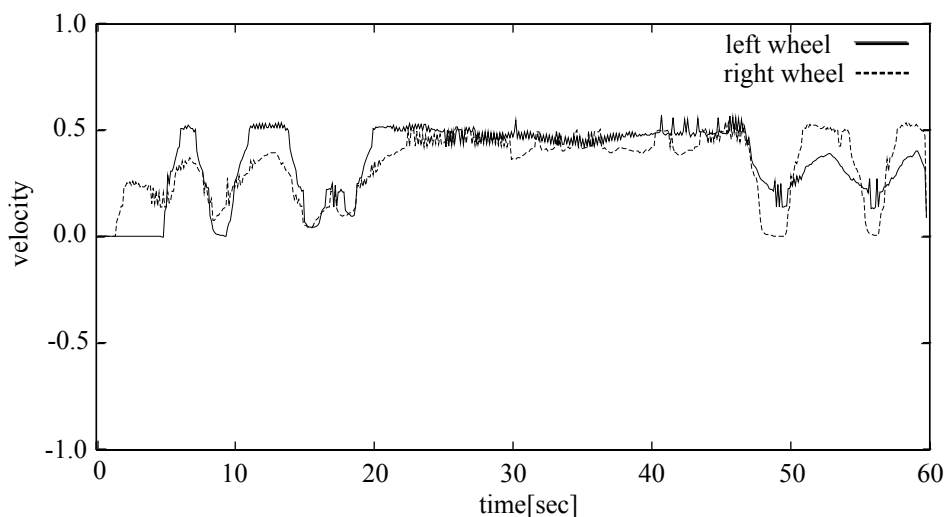


図 3.10: 提案手法コントローラの実環境行動時におけるモータ出力の時間推移

示す。ここで、両図におけるグレーの領域は、進化時に経験した初期角度を示している。従来手法では、シミュレーション環境下でタスク達成できた条件においても、実環境下では大きく成功率が下がることがわかる。これに対し、提案手法では、実環境下に移行してもタスク成功率の低下が小さく、従来手法に比べ頑健であることが示された。

3.5 回路規模の比較

本節では、実時間神経修飾機構を有するニューラルコントローラをハードウェア化するにあたり、確率的エンコーディング手法を用いた場合と、直接エンコーディングを行った場合の回路規模の比較について述べる。

表 3.2 に提案手法、従来手法を確率的エンコーディングで行った場合と直接的エンコーディング法（並列乗算器、メモリ回路などを用いる手法）で行った場合の回路規模の比較を示す。なお、見積もりの方法は、ニューロコントローラを構成する各モジュール（乗算器、シグモイド演算回路、加算器、ニュー

表 3.2: コントローラの回路規模の比較

	確率的 エンコーディング	直接 エンコーディング
実時間神経修飾機構を用いた ニューラルネットワーク	5221	32810
静的な結合荷重で構成した従来型 ニューラルネットワーク	2839	22510

(Logic Cells)

ロモジュレータ放出回路、ニューロモジュレータ反応回路)をRTL記述後論理合成し、FPGA上にマッピングすることで行った。

提案手法は、実時間神経修飾機構を有するニューロコントローラを、確率的エンコーディングを用いてハードウェア化した手法であるので、表中の左上の数値が実機実験で用いた提案手法の回路規模である。表から明らかなように、確率的エンコーディング法を用いることで直接的なエンコーディングを行った場合に比べ約1/6程度に回路を小規模化したことを確認できる(図3.13)。また、確率的エンコーディング手法を用いたシナプス荷重が静的である一般的なニューラルコントローラに比べ、提案手法は図3.8に見られるように複数のニューラルネットワークを構成できるにも関わらず、実時間神経修飾機構を付加した場合でも2倍以下の回路規模で実現されている。このことから、本手法はコントローラの小型化に有効な手段であることが確認できた。

3.6 まとめ

本章では、前章で提案した実時間神経修飾機構を有するニューラルネットワークをFPGAを用いてハードウェア化した。具体的には、ハードウェア化

にあたり、回路規模の増大を緩和する川島らの提唱する確率的エンコーディング手法に対して、親和的な実時間神経修飾機構のハードウェア化を提案することで、回路規模の増加を抑えた。加えて、提案手法DREC搭載した実機を製作し、シミュレーションで獲得したコントローラデータをCPUを搭載していない実機へシームレスに移行させ、実環境で動くことを確認した。同時に、行動中のデータを解析することで、リアルタイムでコントローラ特性を変化させ、環境に適応した行動を発現していたことを示し、提案した手法の有効性を示した。さらに、確率的エンコーディング手法を用いた実時間神経修飾機構のハードウェアゲート数と、直接エンコーディング手法を用いた実時間神経修飾機構のハードウェアゲート数や従来手法である静的な結合荷重を持つニューラルネットワークのハードウェアゲート数とを比較することで、提案手法の有効性を示した。

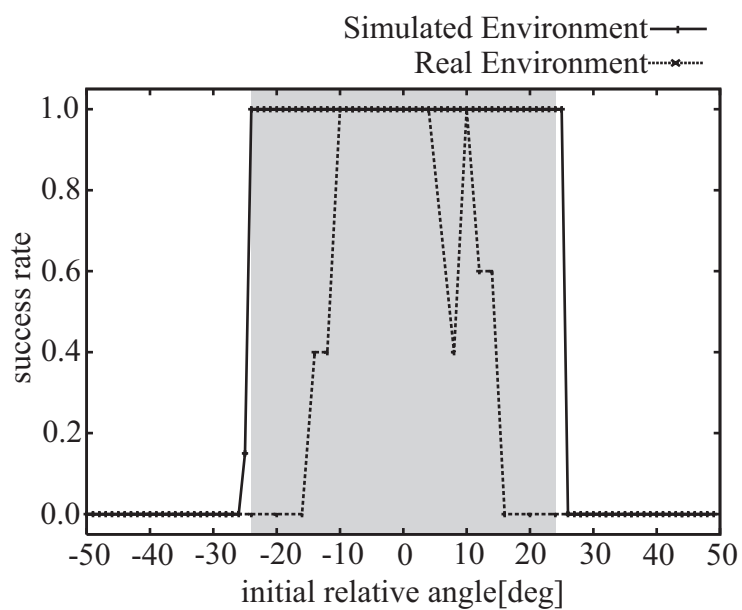


図 3.11: ロボットの初期方向を変動させた時の従来手法を用いたコントローラのタスク成功率の推移

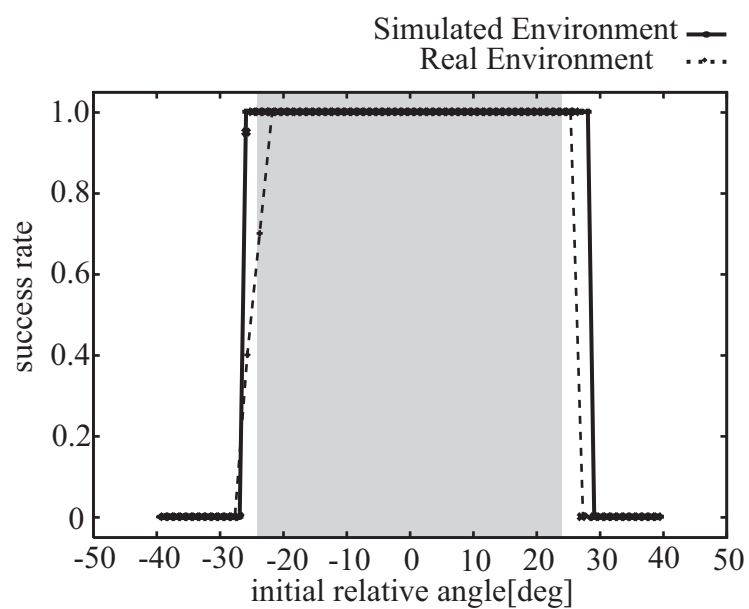


図 3.12: ロボットの初期方向を変動させた時の提案手法を用いたコントローラのタスク成功率の推移

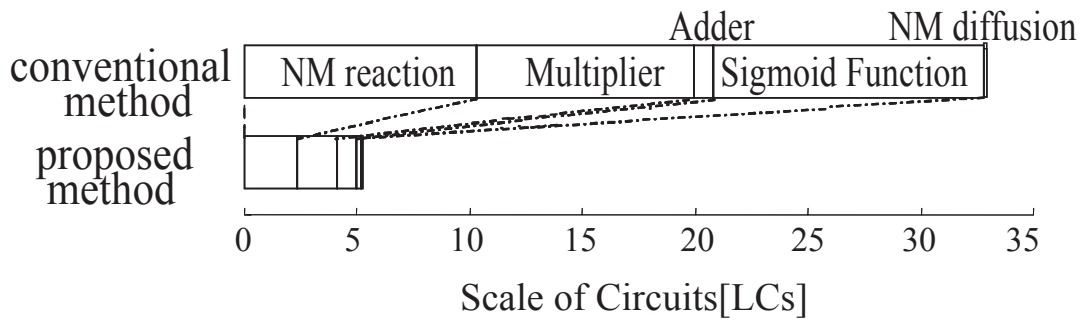


図 3.13: 確率的エンコーディング手法を用いた場合の提案手法と用いなかった場合の各要素の回路規模の比較

第4章 実時間神経修飾機構の解析と改良

4.1 はじめに

第2章では、状況に応じてコントローラの入出力特性を変更させることで頑健性を高めるニューラルコントローラとコントローラの自律的な構築方法について提案し、有効性を示した。第3章では提案したコントローラのハードウェア化手法を示し、CPUレスロボットの開発、実機検証、回路規模の比較検討を通して、提案手法の有用性を示した。

本章では、提案してきた実時間神経修飾機構を有するニューラルコントローラを解析し、従来手法に対して頑健性が高まった要因を明確にする。そのために、状況のカテゴリ化と入力履歴を利用した行動切り替えにより、本手法が未経験環境やセンサノイズが存在する環境下においても、対応できるニューラルコントローラを再編成し、頑健にタスク達成できることを示す。加えて、より複雑なタスクや頑健性の向上を目指し、実時間神経修飾機構の入出力写像の向上を行い、機能向上を行う。

以下、4.2節において、実時間神経修飾機構に改良を加えることで、ニューラルネットワークのシナプス荷重の取りうるバリエーションを増やす。同時に、入出力写像能力の向上をも目指す。4.3節においては、本手法で獲得したコントローラを解析することにより、従来手法に対して頑健性が高まった要因を明確にする。加えて、カテゴリ化した状況に応じてコントローラを再編成することで、経験していない環境下でも的確に行動を切り替え、頑健性が高まることを示す。4.4節では、実時間神経修飾機構のさらなる改良を行なうことで状況認識能力を高め、頑健性の向上を確認する。4.5節では、

本手法のセンサノイズに対する頑健性について示す。最後に4.6節で結びとする。

4.2 ニューラルコントローラの入出力写像能力向上に向けた実時間神経修飾機構の改良

この節では、2,3章で用いてきた実時間神経修飾機構の問題点を明らかにし、改良を加える。そのために、カテゴライズ化された状況に対する行動の多様性を目指し、ニューロモジュレータの放出部と反応部に改良を加え、入出力写像能力の向上を行う。

4.2.1 実時間神経修飾機構の改良すべき点

ここでは、実時間神経修飾機構を有するニューラルネットワークの問題点を挙げる。

2章では、実環境への移行が可能であることを示し、実時間神経修飾機構の有効性を示した。しかし、より複雑なタスクを実行できるコントローラを構築するためには、用いてきた実時間神経修飾機構を有するニューラルコントローラは、いくつかの問題点を有していた。

1つ目の問題として、状況のカテゴライズ数に対する遺伝子長増大の問題があった。ニューロモジュレータの種類数は区別する状況のカテゴライズ数とコントローラの表現能力に直結している。しかし、ニューロモジュレータの種類数を n とすると、 2^n 個の解釈テーブルを各シナプス結合の数だけ用意する必要があり、ニューロモジュレータの数を増やすと遺伝子長が爆発的に長くなる欠点があった。これは、進化アルゴリズムでの探索空間の増大に繋がりが、複雑なコントローラを作る上で進化に要する時間が長くなるため、大きな問題となる。(以下、「遺伝子長の増大問題」と呼ぶ)

2つ目の問題として、ニューラルコントローラの入出力写像の変更則に関して、問題があった。図4.1に、第2章で用いた実時間神経修飾機構を有す

4.2. ニューラルコントローラの入出力写像能力向上に向けた実時間神経修飾機構の改良65

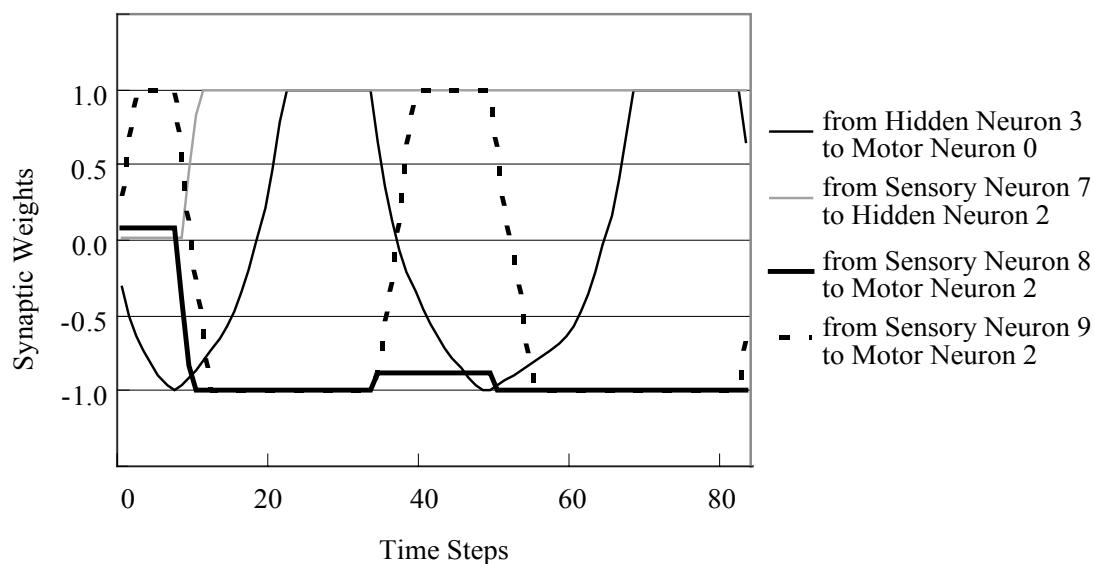


図 4.1: 改良前の実時間神経修飾機構を用いたニューラルネットワークに見られるシナプス荷重時間推移例

るニューラルコントローラを用いてペグ運び問題を行った際の、いくつかのシナプス荷重の時間推移例を示す。図より、シナプス荷重のとれる値が上限値、下限値、初期値（荷重変化を一切しなかった場合）の3種類の値に収束する傾向にあった。これは、ネットワークの表現能力を著しく低下させることに繋がる。また、シナプス荷重の変更式に前後のニューロン活性度の積を用いていたため、ロボットの直面している状況が変化し、ネットワークの再編が必要となった場合においても、前後どちらかのニューロンの活性度が低い状態では、シナプス荷重の変更が十分に行えないという問題があった。（以下、「入出力写像変更則の問題」と呼ぶ）

4.2.2 実時間神経修飾機構の改良

そこで、ニューロモジュレータ濃度の概念の導入とシナプス荷重の変更式に改良を加えることで、シナプス荷重が限界値に収束することなく、様々な

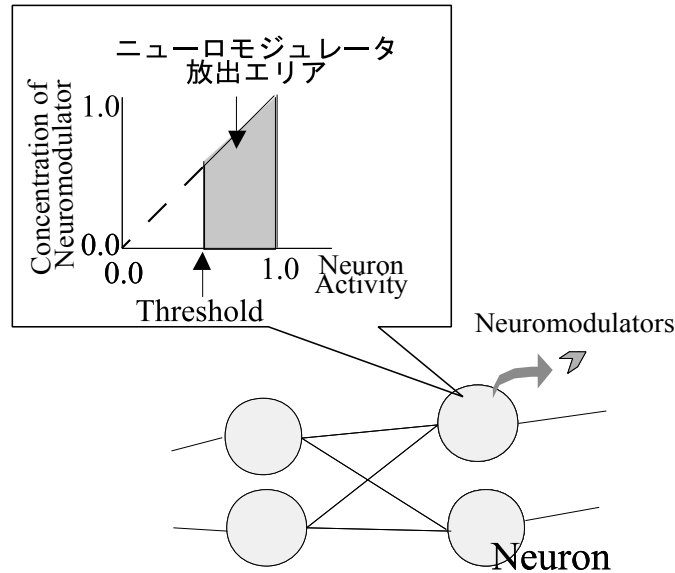


図 4.2: 実時間神経修飾機構の改良後のニューロンにおける活性度と放出ニューロモジュレータ濃度の関係

連続値で安定させ、ニューラルコントローラの入出力写像のバリエーションを増やす。同時に、シナプス荷重の変更式における、前後ニューロンの活性度の積の項を用いない式を提案する。具体的には、図 4.2 に示すように、各ニューロモジュレータの放出権を有するニューロンは、活性度が遺伝的に決定される閾値を越えている時に、遺伝的に決定された種類のニューロモジュレータを活性度に応じた濃度で放出できるものとした。ニューロン i から放出される各種ニューロモジュレータの濃度 Con_i^t を決定する式は下記の通りとした：

$$Con_i^t = \sum_{i=1}^{\text{ニューロン総数}} \begin{cases} a_i^t & (\text{if } a_i^t \geq \theta_i) \\ 0 & (\text{if } a_i^t < \theta_i) \end{cases} \quad (4.1)$$

ここで a_i^t は時間 t におけるニューロン i の活性度、 Con_i^t は時間 t における l 種類目のニューロモジュレータの濃度を示す。また、 θ_i は、各ニューロンが持つ遺伝的に決定する放出閾値である。

4.2. ニューラルコントローラの入出力写像能力向上に向けた実時間神経修飾機構の改良67

加えて、放出されたニューロモジュレータに対して、シナプス荷重の変更の仕方を決定するパラメータの遺伝子へのコーディング方法を改良する。具体的には、シナプス結合ごとに、 l 種類目のニューロモジュレータに対する反応の仕方 $Rule_{ij}^l$ を+1 (シナプス荷重の正方向へ反応)、-1 (シナプス荷重の負方向へ反応)、0 (反応無し) のいずれかをもつものとした。そして、下記の式に示すように全ての種類のニューロモジュレータに対し、反応の仕方 $Rule_{ij}^l$ と式 (4.1) に示すニューロモジュレータの濃度の積の総和を取ることとした：

$$W_{ij}^{t+1} = \sum_{l=1}^N Rule_{ij}^l \cdot Con_l^t \quad (4.2)$$

ここで、 N は用いるニューロモジュレータの種類数を意味する。

これらを用いることで、「入出力写像変更則の問題」に対しては、ニューロンの活性度の大きさは常に1以下の値を取るためシナプス荷重は発散することなく、それぞれ特定の値に収束するため解決する。また、ニューロモジュレータ濃度が連続値であるため、シナプス荷重も連続値を取ることができる。加えて、シナプス荷重の変更式を前後ニューロンに活性度に非依存にすることで、前後どちらのニューロンの活性度が低い状態においても、シナプス荷重の変更が可能となる。これらは、改良前に比べ、実時間修飾機構によるニューラルネットワークの写像バリエーションの増大、つまり行動の多様化を意味する。

ここで、進化の対象とするために遺伝子にコーディングする内容は、各ニューロンに存在するニューロモジュレータ放出部のニューロン放出閾値 θ 、ニューロモジュレータの放出権利の有無 F 、放出するニューロモジュレータの種類 l 。加えて、各シナプス結合に存在するニューロモジュレータ反応部の、 l 種類目のニューロモジュレータに対する反応の仕方 $Rule^l$ となる。図4.3に改良後の遺伝子コーディングの構成を示す。ここで、 I, J は想定するニューラルネットワークのニューロン数とシナプス結線数を意味する。改良前では、各シナプスにおいて n 種類のニューロモジュレータの放出パターン数 2^n 個の解釈テーブルを必要としていたのに対し、改良後は各シナプスに

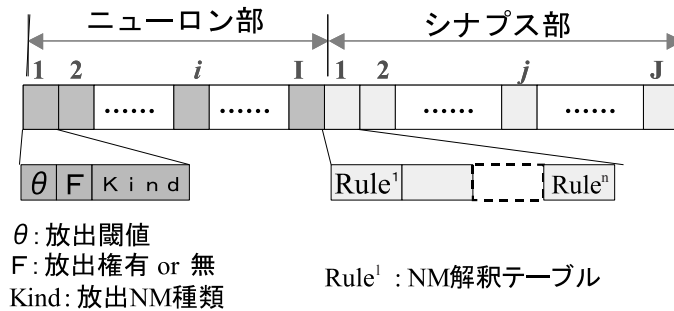


図 4.3: 改良した実時間神経修飾機構を有するニューラルコントローラの遺伝子の構成

おけるニューロモジュレータの解釈テーブルが n 種類それぞれのニューロモジュレータに対する反応の仕方をコーディングすることとなった。その結果、「遺伝子長の増大問題」は、改良手法を用いることで、ニューロモジュレータの種類数 n に対する遺伝子の増大が $order(2^n)$ から、 $order(n)$ に緩和され、進化時における探索空間の爆発を抑えることが可能となった。

4.2.3 コントローラの入出力写像能力を改良したコントローラの確認

ここでは改良により、多彩な行動バリエーションを導く実時間神経修飾機構を有するニューラルコントローラの入出力写像能力が改良前に比べ、改善された結果を示す。

改良後の実時間神経修飾機構を有するニューラルネットワークを用いて、ペグ運び問題のコントローラを構築した。図 4.4 に、構築したコントローラがタスク試行した際のシナプス荷重の時間推移の一例を示す。図 4.1 に比べ、上限値や下限値を設けることなく、シナプス荷重が様々な連続値で安定することが確認できる。これにより、ニューラルネットワークの入出力写像のバリエーションが増加し、実行できる行動のバリエーションが増加したことを確認できた。

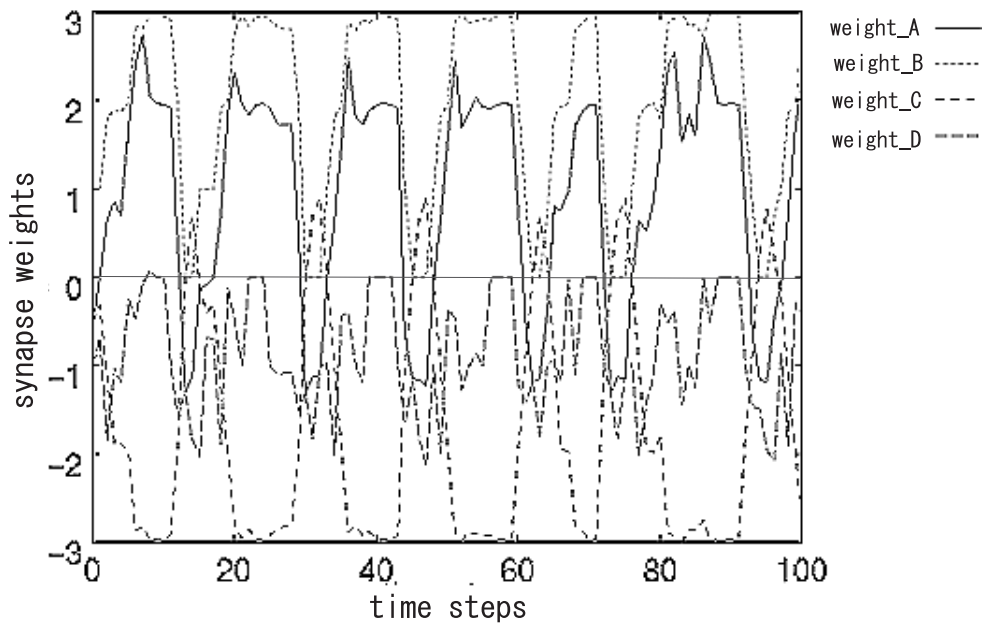


図 4.4: 行動の多様化に向けて改良された実時間神経修飾機構を有するニューラルネットワークのシナプス荷重の時間推移例

4.3 実時間神経修飾機構の行動解析

この節では、改良された実時間神経修飾機構を用いて自律的に構築したベグ運び問題を行うコントローラに対し、行動解析を行い、状況のカテゴリ化と各状態における行動が獲得できていることを確認する。加えて、静的なシナプス荷重を進化の対象とした従来手法に比べ、状況に応じたコントローラを実時間で再編することで、未経験状況においても状態遷移が可能となり頑健性を保持していることを示す。

4.3.1 コントローラの行動特性を示す状態遷移ベクトル図

ここでは、行動解析を行う際に用いた状態遷移ベクトル図について、図 4.5 を用いて説明する。

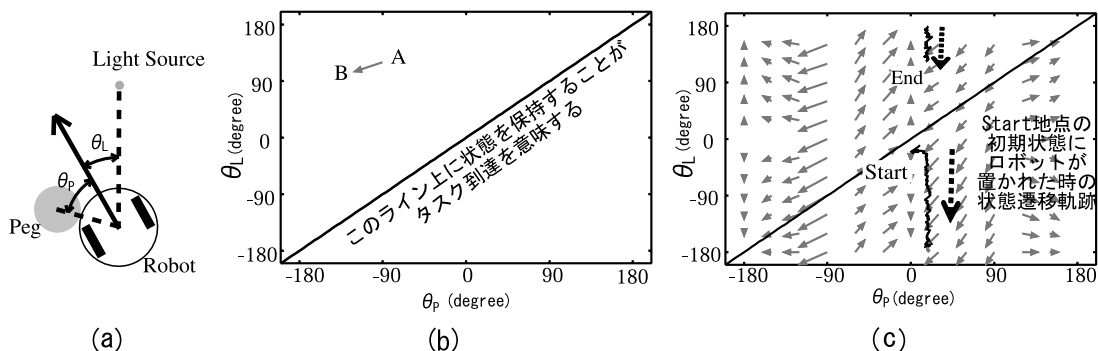


図 4.5: 状態遷移ベクトル図によるロボットの状態表現

ペグ運び問題において、図 4.5(a) に示すように、ロボットとペグ、ロボットとゴールそれぞれの相対角度を (θ_p, θ_L) とした時、図 4.5(b) は、 (θ_p, θ_L) が $A(-90, 125)$ から $B(-130, 100)$ に移行するような行動をした際のベクトル表現図である。この表現に基づき、あるコントローラに対し、ペグとロボットとゴールを任意の位置に置いた時のロボットの取る行動をベクトル表現した図が図 4.5(c) である。つまり、図 4.5(c) において各ベクトルは、ロボットの次状態への状態遷移を示している。また、ペグ運び問題は、ロボットに対してペグとゴールを同じ方向に保ち続けることがタスク達成には必要である。つまり、この状態遷移ベクトル図において、形成されたベクトルに応じて状態を遷移させ続け、 $(+180, +180)$ と $(-180, -180)$ を結ぶ直線上に状態を保ち続けることが、ペグ運びのタスクを達成することを意味する。反対に、ベクトルが存在しない、あるいは状態遷移が可能であっても、 $(+180, +180)$ と $(-180, -180)$ を結ぶ直線上に状態を遷移させることができない場合は、タスクを達成することができないことを意味する。また、そのような領域がこの状態遷移ベクトル図に多く存在すればするほど、そのコントローラには、タスク達成ができない状況が多く存在することを示している。

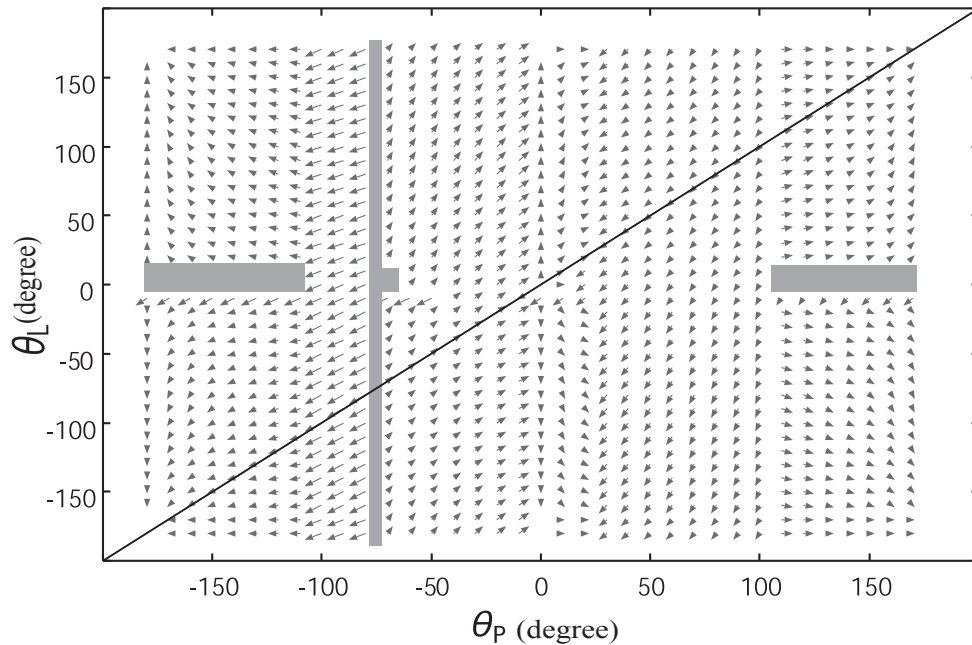


図 4.6: 従来手法を用いた時の各状態における状態遷移ベクトルパターン（グレー領域は状態遷移が行えない状況を示す）

4.3.2 状況のカテゴリイズとそれに応じたコントローラの発現による未経験環境に対する頑健性

ここでは、先に説明した状態遷移ベクトル図を用いて、状況のカテゴリイズとそれに応じたコントローラが発現することによる、提案手法の未経験経験環境への頑健性について述べる。

図 4.6 に静的なシナプス荷重を進化の対象とした従来手法を用いたコントローラの状態遷移ベクトル図を示す。図 4.6 には、図中のグレー領域に示す、相反する状態遷移を行う狭間においてベクトルが存在しない区域が存在している。これはつまり、このベクトルが存在しない状態に移行してしまうと他の状態に移行することができず、タスクが失敗することを意味している。また、図中のベクトルが存在している状態においても、 $(+180, +180)$ と $(-180, -180)$ を結ぶ直線上に移行することができないベクトル空間が存在しており、タス

ク達成ができる状態が限られていることが確認できた。

これに対し、図4.7にロボットの直面した状況の軌跡と、各状況における実時間神経修飾機構を有するニューラルコントローラによる状態遷移ベクトル図を示す。実時間神経修飾機構を有するニューラルネットワークの場合、実時間でニューラルネットワークの特性が変化するため、図4.7(b)と(c)に示すように、ロボットの置かれている状況が異なる場合、ベクトルの大きさと方向が大きく変化している。そして、状況に応じたベクトル遷移図の変更を複数回行うことで、ロボットの置かれた状態を $(+180, +180)$ と $(-180, -180)$ を結ぶ直線上に移行させ、保持できることが確認できた。

また、従来手法の状態遷移ベクトル図と異なり、実時間神経修飾機構を有するニューラルネットワークの場合、各状況において、ベクトルの方向がほぼ同じ方向を向いていることが多い。これは、それぞれの状況において、任意のコントローラ入力に対して、ほぼ同様の特定の行動を引き起こす専門のコントローラを発現していることを意味する。そのため、一つのコントローラで相反する行動同士が切り替わる状況に対処する必要がなくなり、従来手法による状態遷移ベクトル図に見られた、相反する行動の狭間の状態遷移が行えない空間が少ない。その結果、提案手法の状態遷移ベクトル図は、従来手法の状態遷移ベクトル図に比べ、空白の区域が少ない。これは、状態遷移できない状況が少なく、未経験状況下においても状態遷移ができる、つまり、未経験環境に対する頑健性が従来手法に比べ高いことを意味する。

4.4 実時間神経修飾機構の改良による状況カテゴライズ能力の向上

この節では、より詳細に状態のカテゴライズを行うために、状態カテゴライズを行うニューロモジュレータ放出部の改良を検討する。そして、非線形タスクであるバックトレラ問題に適用し、本手法によりノイズに対する頑健性が高まったことを示す。

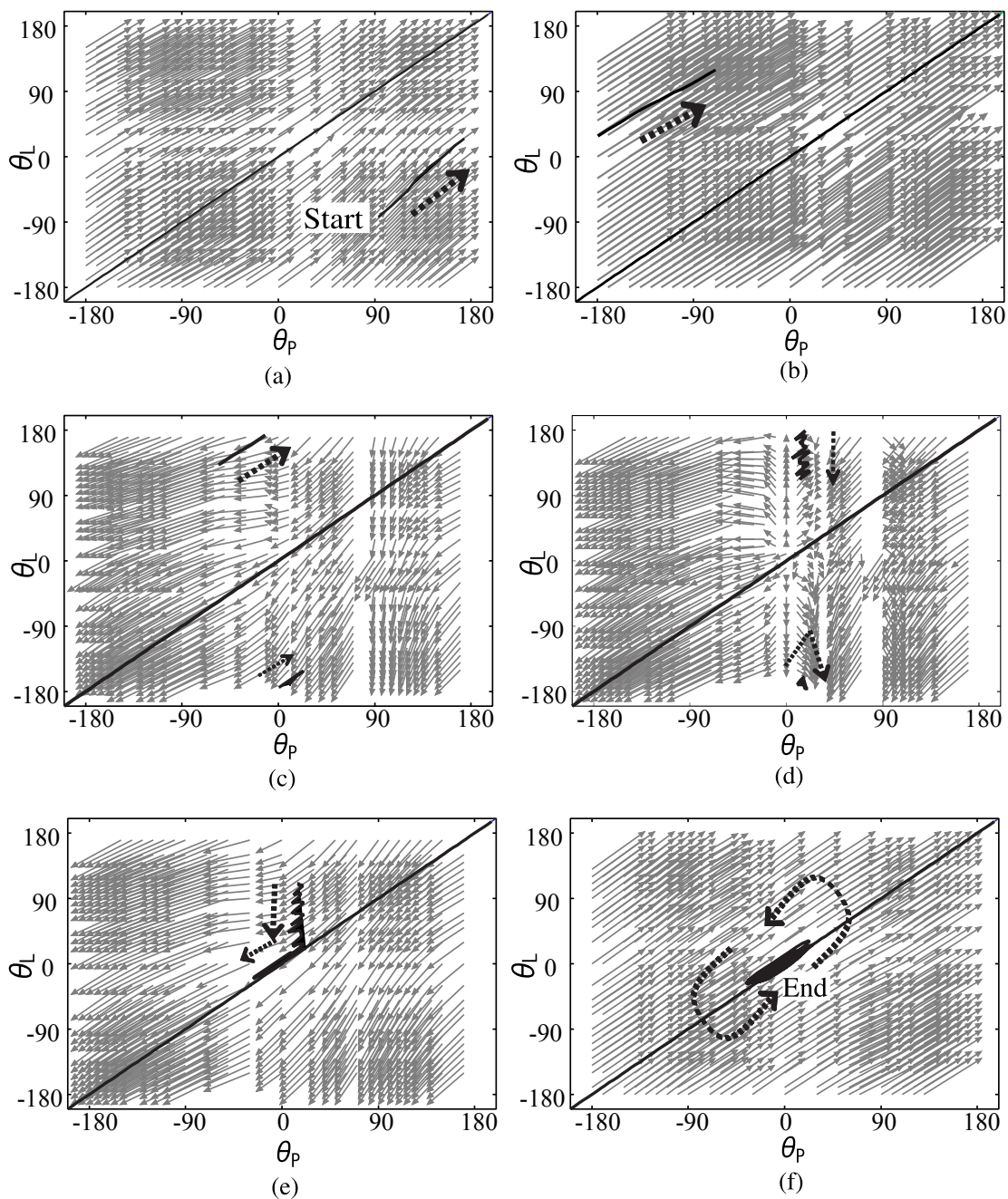


図 4.7: 提案手法を用いたコントローラで、Start の状態から実線で描いた状態遷移が起きた時の、実時間で変更した状態遷移ベクトルパターンの変更履歴

4.4.1 状態カテゴライズを行うニューロモジュレータ放出部の改良

4.2節の改良においてニューロモジュレータ濃度の概念を取り入れ、ニューロモジュレータの反応式を改良することで、ニューラルコントローラの写像能力を高めた。これに対しこの項では、ニューロモジュレータの放出に関する条件を改良すると共に、濃度の計算に時定数を持たせることで、応答特性をニューロモジュレータ毎に持たせることを可能にし、より詳細な状況のカテゴライズ化を可能にする。

具体的には、ニューロン i における l 種類目のニューロモジュレータ放出に関する式を、正規分布関数を利用した下記の通りとした：

$$S_{i,l}(t) = \frac{1}{\sqrt{2\pi}\sigma_{i,l}} \exp\left(-\frac{(a_i(t) - \mu_{i,l})^2}{2\sigma_{i,l}^2}\right) \quad (4.3)$$

ここで、 $\mu_{i,l}$ は l 種類目のニューロモジュレータの放出濃度が最大となるときのニューロン i の活性度、 $\sigma_{i,l}$ はニューロン i における l 種類目のニューロモジュレータの放出を規定する正規分布関数の広がりの度合いをそれぞれ表す。この式は、ニューロモジュレータの放出権を持つニューロンに対して、放出可能なニューロモジュレータの数だけ割り当てられる。このニューロモジュレータ放出部のイメージ図を図4.8に示す。また、式(4.3)で利用される変数において、各ニューロンは放出するニューロモジュレータの種類 l と、放出するニューロモジュレータごとのニューロモジュレータの放出量が最大となる値(正規分布関数の平均値と等価)を与える変数 $\mu_{i,l}$ 、および、ニューロモジュレータの放出量を決定する変数 $\sigma_{i,l}$ (正規分布関数での標準偏差と等価)を進化的に獲得するものとした。

また、時間 t におけるニューロン i が放出するニューロモジュレータ濃度 $C_i^l(t)$ に関しては、下記の通りとした：

$$C_i^l(t) = (1 - \tau_{i,l})C_i^l(t-1) + \tau_{i,l}S_{i,l} \quad (4.4)$$

この式(4.4)において、ニューロン i から放出される l 種類目のニューロモジュレータの濃度 $C_{i,l}(t)$ に対する時定数 $\tau_{i,l}$ が設定され、この変数 $\tau_{i,l}$ も進化の対象となる。

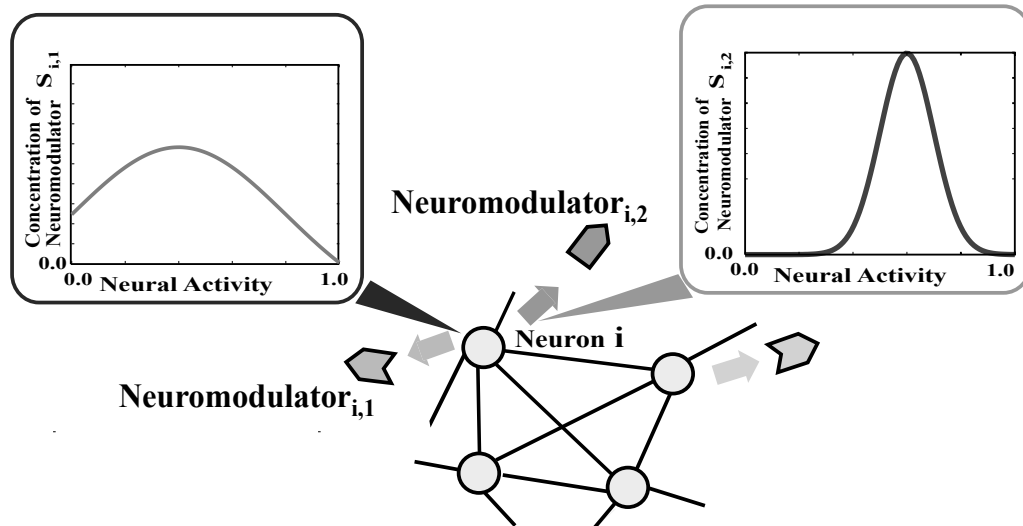


図 4.8: 改良したニューロモジュレータ放出部のイメージ図

これらを受けて、シナプス荷重の変更を行う式は、先の改良と同様の下記の式を用いた：

$$w_{ij}^{t+1} = \sum_{l=1}^N Rule_{i,j}^l \cdot Con_i^l(t) \quad (4.5)$$

ここで、 N は用いるニューロモジュレータの種類数を意味する。また、 $R_{i,j,l}$ は、ニューロン i からニューロン j へのシナプス結合において、 l 種類目のニューロモジュレータに対して、反応する権利を持っている場合、ニューロモジュレータに対応する変更則 $+1$ か -1 をとるものとし、反応する権利を有していなければ、 0 を取るものとした。

これらの改良により、実時間神経修飾機構を有するニューラルコントローラはコントローラ応答性をも、自律的に構築が可能となる。加えて、式(4.3)、(4.4)における $\mu_{i,l}$ と $\sigma_{i,l}$ と $\tau_{i,l}$ の効果により、ニューロンの活性度における情報の注目度や反応すべき応答速度に応じた的確にニューラルコントローラの改変を促すことを可能とした。

また、これらの改良を受けて、進化の対象となるパラメータは下記の通りとなった：

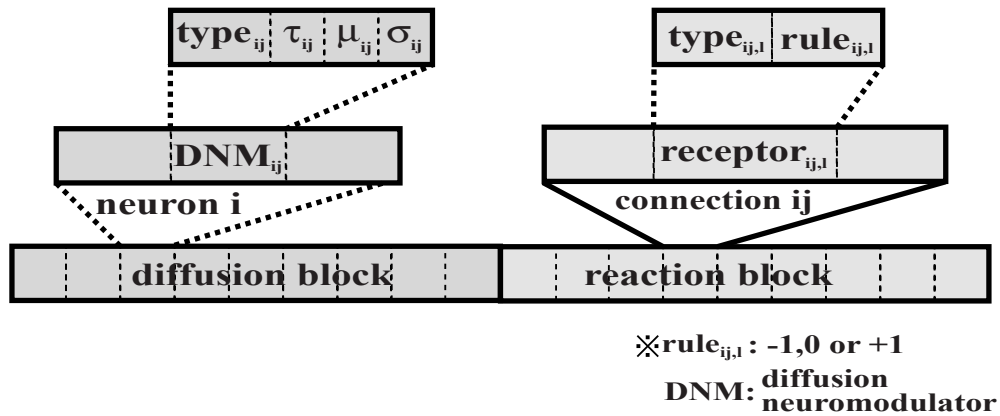


図 4.9: 改良版によるコーディング

- 各ニューロンにおけるニューロモジュレータ放出部において
 - 放出するニューロモジュレータの種類 (1, 2...n を放出できる個数分。ただし n はニューロモジュレータの種類数)
 - ニューロモジュレータを放出できる条件 $\sigma_{i,l}$ と $\mu_{i,l}$
 - 放出するニューロモジュレータの時定数 $\tau_{i,l}$ (ただし、 $-1 < \tau_{i,l} < 1$)
- 各シナプス結合におけるニューロモジュレータ反応部において
 - 反応するニューロモジュレータの種類 (1, 2...n を反応する種類数分。ただし n はニューロモジュレータの種類数)
 - 反応の仕方 $Rule_{i,j}^l$ ($-1, 0, +1$ のいずれか)

これらを遺伝子にコーディングした時の遺伝子の構造図は図 4.9 となる。

4.4.2 バックトレラ問題への適用

この項では、改良した実時間神経修飾機構を有するニューラルコントローラを非線形問題であるバックトレラ問題 [58],[60],[61],[63] に適用した例を示す。

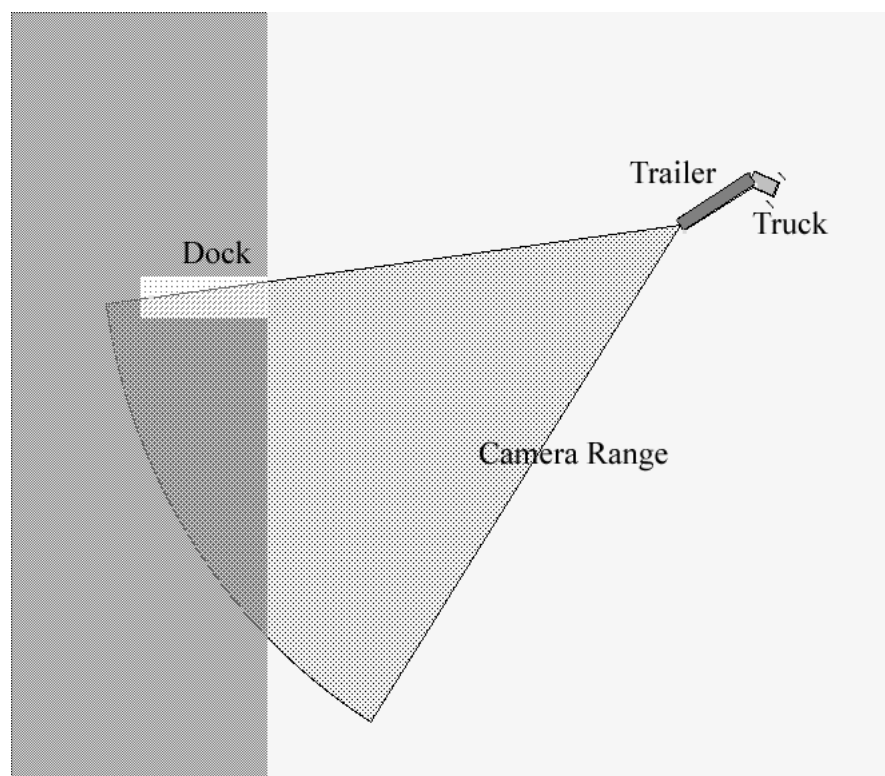


図 4.10: バックトレーラ問題

バックトレーラ問題は、図 4.10 に示す環境において車庫に後進でトレーラを入れることを目的とする。本タスクはトレーラ部とトラック部が直角になることで、後進できなくなるジャックナイフ現象を防ぐ必要がある非線形問題である。さらに、ペグ運び問題に比べ、制御対象であるトラック部とトレーラ部の拘束条件が複雑で、より困難なタスクである。

改良版実時間神経修飾機構を組み込んだコントローラとして、3層フィードフォワードニューラルネットワークを用いた。そして、コントローラの出力としては、トラックのステアリング角の大きさと速度を用いた。一方、入力には、トレーラ部の最後尾にカメラが搭載されているものと仮定し、トレーラ最後部から車庫までの距離とトレーラと車庫の成す相対角度、加えてトレーラとトラックの相対角の値を用いた。そして、放出できるニューロモジュレー

タの種類は最大で8種類とした。

次に、進化時に用いた評価関数 $fitness$ を次に示す：

$$fitness = k_1 \cdot \frac{dist(dock, trailer)_{end}}{dist(dock, trailer)_{start}} + k_2 \cdot S + k_3 \left\{ \frac{\theta(trailer, dock)}{\pi} + \frac{\theta(trailer, truck)}{\pi} \right\}$$

ここで、 $dist(dock, trailer)_{start}$ 、 $dist(dock, trailer)_{end}$ はそれぞれトレーラの最後尾と車庫の初期状態での距離と最終状態での距離を示している。また、 S は、最終状態での車両と車庫との重なっている面積、 $\theta(trailer, dock)$ 、 $\theta(trailer, truck)$ はそれぞれ、トレーラと車庫との角度、トレーラとトラック間の相対角を意味する。これはつまり、最終的にトラックとトレーラをより直線状にしつつ、車庫に入れることで、高い評価が得られる式になっている。これらの条件の下、静的なシナプス荷重を進化の対象とした従来手法によるニューラルコントローラと、改良前と改良後の実時間神経修飾機構を有するニューラルコントローラを、それぞれ複数回試行することで評価を行ない、進化的にコントローラを構築した。

4.4.3 改良による頑健性向上の確認

この項では、実時間神経修飾機構の改良を行った結果、頑健性が向上したことを示す。

頑健性評価として、初期状態におけるトレーラと車庫缶の距離を変化させた場合の評価値の変化を検証した。図4.11,4.12は、進化を通じて得られた改良前と改良後の実時間神経修飾機構を有するニューラルコントローラ各5個体に対し、初期状態におけるトレーラと車庫までの距離を変えた時の評価値の変化を示している。複数個体を比較することで、改良を加えることで以前の実時間神経修飾機構を用いた場合に比べ、評価値がいずれも高く、頑健性を高める手法として有効であることを確認した。

4.5. 実時間神経修飾機構を有するニューラルネットワークのノイズに対する頑健性79

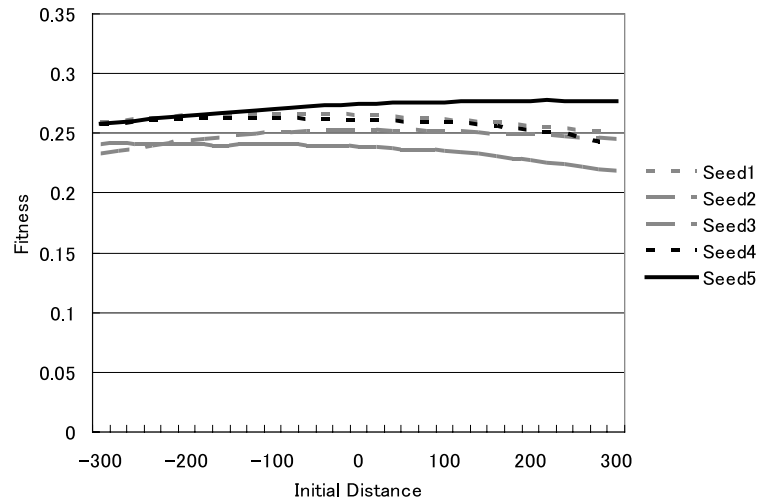


図 4.11: 改良前実時間神経修飾機構を有するニューラルコントローラを用いた場合の初期距離変化に対する評価値の推移

4.5 実時間神経修飾機構を有するニューラルネットワークのノイズに対する頑健性

この節では、改良された実時間神経修飾機構を用いて自律的に構築したバックトラッカ問題に適応したコントローラに対し、入力情報にノイズを意図的に加えた環境下での実験を行なう。そして、出力結果を解析することで、静的なシナプス荷重を進化の対象とした従来手法に比べ、先に示した未経験状況だけでなく、センサノイズに対しても頑健性があることを示す。

まず初めに、進化で得られた改良版の実時間神経修飾機構を有するニューラルコントローラと静的なシナプス荷重を進化の対象とした従来手法のコントローラ、それぞれに対し、タスク達成した際の軌跡とその時のセンサ入力とステアリング出力をすべての時間について記録した。次に、そのタスク達成したセンサ入力履歴のうち、車庫に対する角度と距離を返すセンサ値に対し、センサノイズとして1.2倍した値を入力値系列としてコントローラへ入

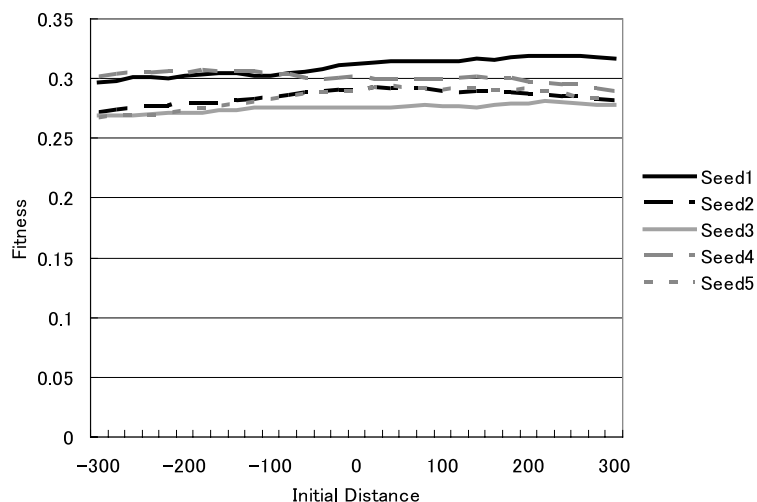


図 4.12: 改良版実時間神経修飾機構を有するニューラルコントローラを用いた場合の初期距離変化に対する評価値の推移

力した際のステアリング出力値履歴を図 4.13 に示す。

図 4.13 より、提案手法である実時間神経修飾機構を有するニューラルコントローラを用いた場合、入力情報にノイズが入った場合でもステアリング出力結果が大きく変化することが無い。それに対し、従来手法を用いた場合、ステアリング出力が常に大きく異なり、ノイズに対する頑健性が無い。これは、提案手法が状況に応じてニューラルコントローラを変更するため、カテゴリ化した状況から逸脱しない限り、同じ系統のコントローラ構造を用い、大きく出力を変更することなく対応するためである。つまり提案手法は、静的なシナプス荷重を進化の対象とした従来手法に比べ、ノイズに対する頑健性を有することが解った。

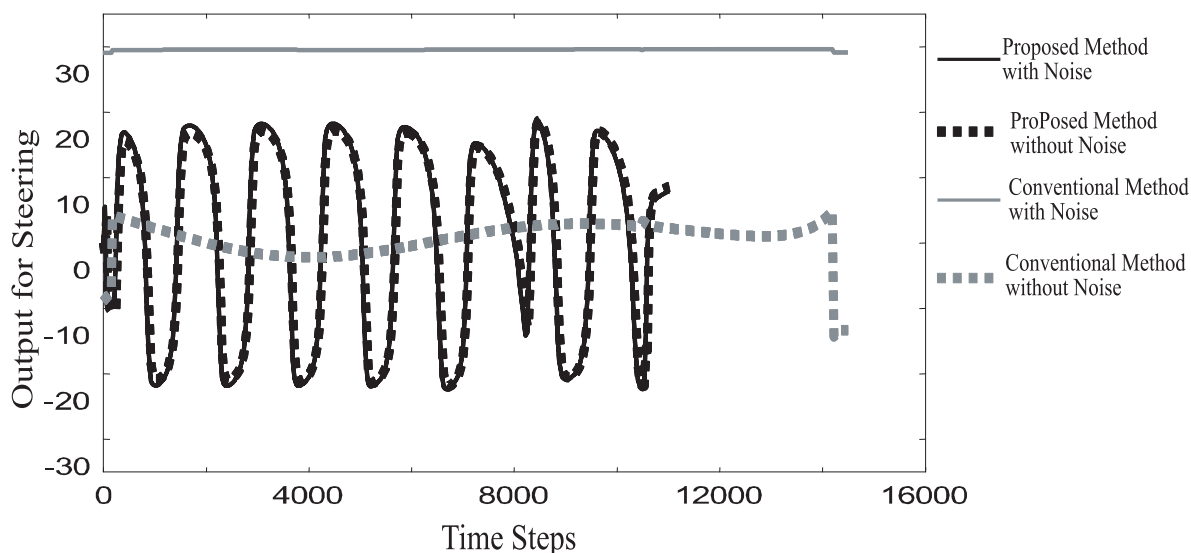


図 4.13: 従来手法のコントローラと改良版の実時間神経修飾機構を有するニューラルコントローラ、それぞれをノイズが含まれていない環境下で実行した時のステアリング出力 (without Noise) と入力情報にノイズを入れた時のコントローラのステアリング出力 (with Noise) との比較

4.6 まとめ

本章では、提案してきた実時間神経修飾機構を有するニューラルコントローラの解析と改良を行った。解析により、状況のカテゴリ化と入力履歴を利用したコントローラの再編成が行われていることを確認した。加えて、任意のコントローラ入力に対して、ほぼ同様の状態遷移を行える専門のコントローラを、各状況下において構築していることを示した。そして、獲得した状況のカテゴリ化に応じて、この専門のコントローラを再編成することにより、未経験状況やセンサノイズが存在する環境下においても、頑健に行動し、タスク達成できることを示した。加えて、より複雑なタスクや頑健性の向上を目指し、ニューロモジュレータの放出条件やシナプス荷重の変更式を改良することで、実時間神経修飾機構の入出力写像の向上を行い、機能向上を行った。

第5章 結論

5.1 本研究の成果

本論文では、ロボットに影響を与える外界である環境が、コントローラを構築する際に想定した環境だけでなく、シミュレーション環境と実環境など、若干異なった場合においてもタスクを達成するコントローラを自律的に構築する手法の提案についてまとめた。同時に、提案手法の実験とコントローラの解析を通して、頑健性実現について研究した結果をまとめた。それらの結果、以下の成果を得た。

第1章では、モデル化することが困難である環境に対するロボットコントローラの自律的構築手法として、強化学習手法や進化ロボティクス手法を示し、各手法の問題点を明らかにした。さらに、頑健なコントローラを構築する際に、タスクを達成する一連の行動すべてを記述するのではなく、状況に応じてカテゴライズ化した行動をそれぞれ安定化させることで頑健な行動が可能であることを述べた研究例をあげた。そして、状況のカテゴライズ化と行動の同時獲得が問題となっていることをも述べた。加えて、自律ロボットを設計する際の考慮すべき点を明らかにし、コントローラのハードウェア化をプログラマブルハードウェアで行うことによる利点を示した。

第2章では、生物が具備している状況に応じて動的に神経回路の機能を変更する機構をモデル化し、状況に応じてコントローラの入出力関係を変更できるニューラルネットワークを提案した。加えて、得られたコントローラを用いて、静的なシナプス荷重を持つ従来手法のニューラルコントローラでは行うことが困難であった、シミュレーション環境から実環境へのコントローラの移行がパラメータの調整を必要とせず行えた実験結果を示し、提案手法の有効性を示した。同時に、未経験状況からのタスク成功率を例に示し、頑

健性を示した。さらに、実行中のニューラルコントローラのシナプス荷重の時間推移の解析を通して、未経験環境における、状況に応じた行動の切り替え発現を示し、本手法の有効性を示唆した。

第3章では、実用化に向けて、消費電力の低減、構成要素の小型化、部品点数の削減、コントローラ改変の容易性などの利点を持つ、FPGAによるハードウェアコントローラ化を、実時間神経修飾機構を有するニューラルネットワークに対して行った。その際に、川島らが提唱している確率的エンコーディング手法に対して、親和性の高い実時間神経修飾機構のハードウェア化手法を用いることで、回路規模の増大を防ぐことが可能であり、有用であることを示した。そして、CPUを用いないロボットにおいても、状況に応じたコントローラの入出力変更を利用して、頑健に動くことをも示した。

第4章では、実時間神経修飾機構を有するニューラルコントローラの解析を行った。解析を通じて、状況のカテゴリ化と状況に応じたコントローラの再編成が行えていることが解った。さらに、各状況下において、任意のコントローラ入力に対して、ほぼ同様の状態遷移を行える専門のコントローラを構築していることが解った。これにより、獲得した状況のカテゴリ化に応じて、この専門のコントローラを状況に応じて発現させ、未経験状況やセンサノイズが存在する環境下においても、頑健に行動し、タスクを達成していることを示した。加えて、より複雑なタスクや頑健性の向上を目指し、ニューロモジュレータの放出条件やシナプス荷重の変更式を改良することで、実時間神経修飾機構の入出力写像の向上を行い、機能向上を行った。

5.2 今後の課題

本論文ではコントローラを進化的に構築し、ロボットの身体を通じた状況のカテゴリ化と、対応した行動を促すコントローラを自律的に同時に獲得した。しかし、本手法はコントローラ構築時の環境において、タスクを達成できる状況のカテゴリ化を終了させており、実際の実行する環境、ロボット部品の経年劣化による制御対象の変化などに対して、最適なカテ

イズ化がなされているとは断定できない。そこで、更なる頑健性の向上を目指して、実行環境下において状況のカテゴリイズ化と行動の獲得を引き続き行い続け、頑健性を高める手法の研究開発が望まれる。加えて、より複雑なタスクに対処するコントローラ構築手法の研究開発が望まれる。

付 録 A

A.1 ニューラルネットワークの確率的エンコーディング手法によるハードウェア化

ここでは、川島らが提案しているニューラルネットワークのパルスニューロンを応用した確率的エンコーディング手法について説明する [54] [55]。

A.1.1 ニューラルネットワークのハードウェア化

多くの研究で用いられているニューラルネットワークは、ニューロンとシナプス結合により構成される。そして、注目するニューロン j の活性度 a_j は、ニューロン j に向けてシナプス結合を結線しているニューロン i の活性度を a_i とし、ニューロン j からニューロン i に結線されているシナプス荷重を w_{ji} とすると、非線形関数であるシグモイド関数を含んだ下記に示す計算式にて計算される：

$$a_j = f\left(\sum_i w_{ji} \cdot a_i\right) \quad (\text{A.1})$$

$$f(x) = \frac{1}{1 + e^{-\alpha x}} \quad (\text{A.2})$$

ここで、シグモイド関数を用いたニューラルネットワークをハードウェア化するには、いかにして精度良くシグモイド関数の非線形写像を再現しつつ、乗算回路などの回路規模を抑えて実装するかが重要となる。そこで、川島らは従来のルックアップテーブル手法に比べ回路規模が小規模で、かつ、近似手法 [53, 52] に比べ高い精度が実現できる手法として、確率的エンコーディ

ング手法を提案した。以下、各項では確率的エンコーディング手法による乗算とシグモイド関数のハードウェア化について説明する。

A.1.2 乗算のハードウェア化

図 A.1 に乗算回路のハードウェア化に際し、用いたパルス遅延とラプラスの定理を用いた手法のイメージ図を示す。ラプラスの定理は以下の定理で表

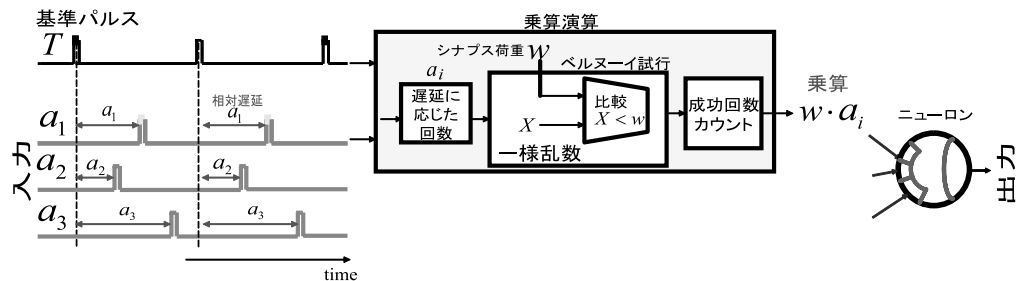


図 A.1: パルス遅延を用いた乗算手法

される。

(ラプラスの定理) 試行回数 n 、成功の確率 p の二項分布 $B(n, p)$ は、 n が十分大きいとき、正規分布 $N(np, np(1-p))$ によって近似される。

この定理より、確率 p で成功するベルヌーイ試行を n 回行った時の成功する回数の期待値は np となる。つまり、 n 個の 0.0 から $|w|$ の間の値を取る乱数 X に対し $X < w$ となる回数をカウントすることで、 n と w の乗算を近似することができる。これは、一様乱数を発生する回路、乱数とシナプス荷重 w_{ji} の大小を比較する回路、比較結果を計数するアップダウンカウンタで実現できる。図 A.2 にハードウェア構成を示す。なお、一様乱数の発生には、線形フィードバックシフトレジスタ (linear feedback shift register: 以下 LFSR) を用いた。

以上のように確率的エンコーディング手法では、乗算回路を LFSR、比較

A.1. ニューラルネットワークの確率的エンコーディング手法によるハードウェア化89

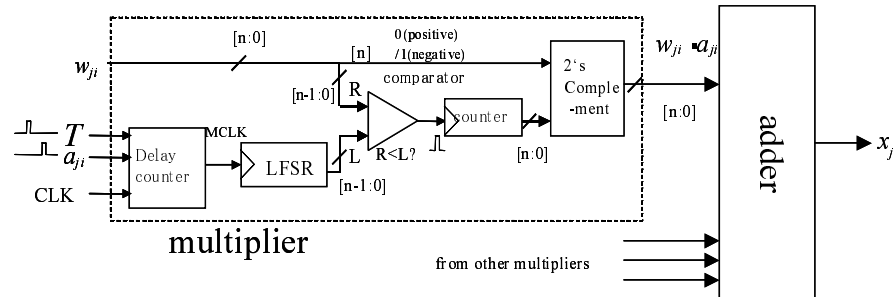


図 A.2: 乗算のハードウェア化

器およびカウンタで構成する。これにより、従来の乗算回路を構成していた並列乗算器に比べ、小規模回路での実装を可能にしている。

A.1.3 シグモイド関数のハードウェア化

シグモイド関数回路は、シグモイド関数を正規分布関数の原始関数に近似することで実現する。なお、正規分布は以下に示す中心極限定理により、一様乱数に従う変数を複数加算することで近似することができる。

(定理 2) $\{X_1, \dots, X_n\}$ を互いに独立に期待値 μ 、分散 σ^2 を持つある確率分布に従う確率変数としたとき、それらの和 $Y = X_1 + \dots + X_n$ は、 n が十分大きいとき近似的に正規分布 $N(n\mu, n\sigma^2)$ に従う。

図 A.4 にハードウェア構成を示す。正規分布に従う数は、LFSR からの出力を 4 つ加算することで発生する。この数と x_j を加算して定数との比較を行い、その大きさをカウントすることで、正規分布関数の原始関数を実現する回路を構築する。これにより、シグモイド関数の実現に必要なメモリを不要にすることができる。

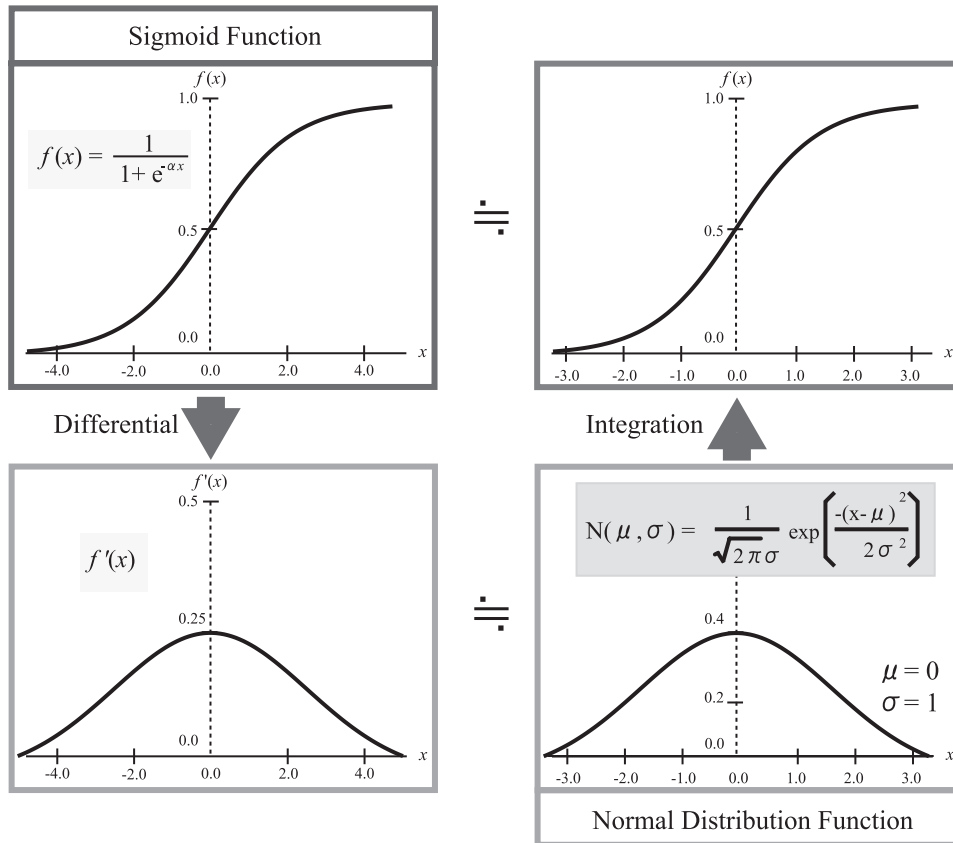


図 A.3: シグモイド関数と正規分布の関係

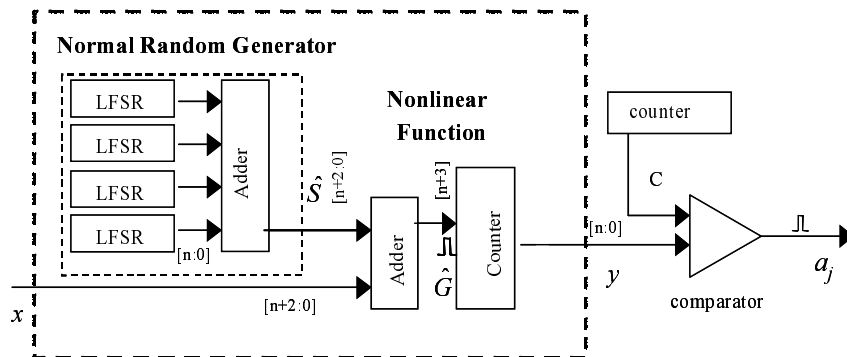


図 A.4: シグモイド関数のハードウェア化

A.1.4 ニューラルネットワークのハードウェア化

先の2項で述べた乗算回路とシグモイド関数を近似再現する回路を用いたニューロン一個分のニューラル回路を図 A.5 に示す。なお、図中の LFSR は、各ニューロンがそれぞれ独立に有する必要はなく、外部に共通の LFSR を 1 個用意して各ニューロンで利用することで、さらに回路規模を抑えることができる。

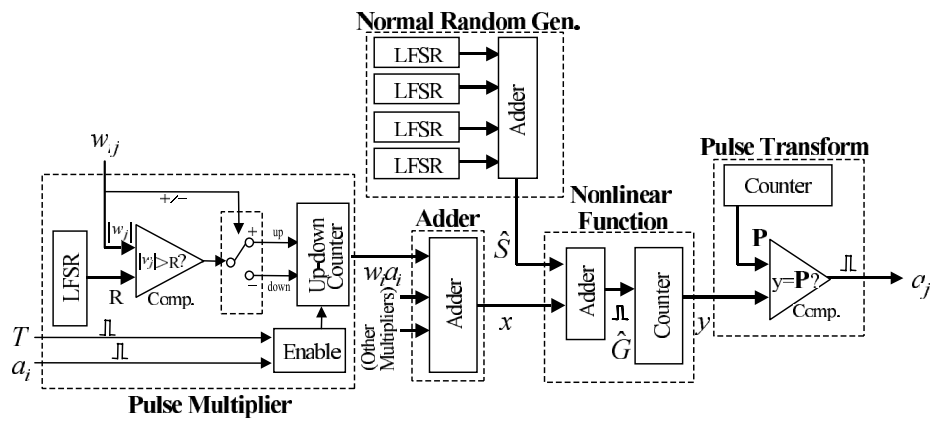


図 A.5: ニューロン一個分の回路

参考文献

- [1] K. Wada, T. Shibata, T. Saito, K. Sakamoto, K. Tanie, “Robot assisted activity at a health service facility for the aged for 17 months: an interim report of long-term experiment”, 2005 IEEE Advanced Robotics and its Social Impacts, pp.127-132, 2005
- [2] M. Fujita, K. Kageyama, “An Open Architecture for Robot Entertainment”, Proceeding of the 1st Interactional Conference on Autonomous Agents, pp.435-440, 1997
- [3] M. Fujita, “AIBO: Toward the Era of Digital Creatures”, The International Robotics Research, Vol.20, pp.787-794, 2001
- [4] T. Tomonaka, R. Hiura, Y. Koketsu, K. Ohnishi, K. Sugimoto, “Computer Vision Technologies For Home Use Robot Wakamaru”, Mitsubishi Heavy Industries, Ltd. Technical Review Vol. 42 No. 1, 2005
- [5] 藤田善弘, “チャイルドケアロボット PaPeRo”, 日本ロボット学会誌, Vol.24, No.2, pp162-163, 2006
- [6] T. Hayashi, H. Kawamoto, Y. Sankai, “Control method of robot suit HAL working as operator’s muscle using biolog information”, Intelligent Robots and Systems 2005, pp.3063- 3068, 2005
- [7] 山本大介, 吉見卓, 鈴木薫, 廣川潤子, 中本秀一, 小川秀樹, 松日楽信人, “ロボット情報家電コンセプトモデル ApriAlpha の開発-機能概要および移動制御について-”, 第 21 回日本ロボット学会学術講演会予稿集, 1E27, 2003

- [8] 三浦純, 河村貞夫, “ロボティクスにおけるモデリング研究の重要性”, 日本ロボット学会誌, Vol.18, No.3, pp.325-330, 2000
- [9] 細江繁幸 編著, “システムと制御”, オーム社, 1997
- [10] 陶山貢市, 瀬部昇, 小原敦美, 藤崎泰正, 浅井徹, 劉康志, 藤田政之, 坂本登, 藤本健治, 石川将人, 増田士朗, 大森浩充, “代表的な制御方法”, 計測と制御, 42 巻, 4 号, pp.268-303, 2003
- [11] J.C. Zufferey, D. Floreano, M.van Leeuwen, T. Merenda, “Evolving Vision-based Flying Robots”, *Proc. of the 2nd International Workshop on Biologically Motivated Computer Vision LNCS*, 2002
- [12] R.A. Brooks, “A Robust Layered Control system for a Mobile robot”, *IEEE Transaction on Robotics and Automation*, Vol. 2, No. 1, pp.14-23, 1986
- [13] R. Pfeifer, C. Scheier, “UNDERSTANDING INTELLIGENCE”, MIT, 1999
- [14] 岡田美智男, 三嶋博之, 佐々木正人, “身体性とコンピュータ”, 共立出版, 2000
- [15] J. Tani, “Learning to generate articulated behavior through the bottom-up and top-down interaction process”, *Neural Networks*, Vol.16, No.1, pp.11-23, 2003
- [16] J. Tani, M. Ito, “Self-organization of behavioral primitives as multiple attractor dynamics: a robot experiment”, *IEEE Trans. on Systems, Man, and Cybernetics Part A: Systems and Humans*, Vol.33, No.4, pp.481-488, 2003
- [17] D. Wolpert, M. Kawato, “Multiple paired forward and inverse models for motor control”, *Neural Networks*, 11, pp.1317-1329, 1998

- [18] 土井利忠, 藤田雅博, 下村秀樹 編, “脳・身体性・ロボット 知能の創発をめざして インテリジェンス・ダイナミクス 1”, Springer, 2005
- [19] 土井利忠, 藤田雅博, 下村秀樹 編, “身体を持つ知能 脳科学とロボティクスの共進化 インテリジェンス・ダイナミクス 2”, Springer, 2005
- [20] 佐藤知正, “学術創成 知能ロボットによる模倣の構成論的研究 最終報告会概要集”, 2006
- [21] 岡田昌史, 中村仁彦, “力学的情報処理による連続的記号空間の設計と全身運動の生成”, 日本ロボット学会誌, vol.23, no.7, pp.583-593, 2005
- [22] T. Inamura, I. Toshima, H. Tanie, Y. Nakamura, “Embodied Symbol Emergence based on Mimesis Theory”, International Journal of Robotics Research, Vol.23, No.4, pp.363-377, 2004
- [23] 稲邑哲也, 中村仁彦, 戸嶋巖樹, 江崎英明, “ミメシス理論に基づく見まね学習とシンボル創発の統合モデル”, 日本ロボット学会誌, Vol.22, No.2, pp.256-263, 2004
- [24] 石井信, “制御理論・強化学習への展開”, 数理科学, 2004年3月号, pp.38-44, 2004
- [25] 浅田稔, “ロボットの行動学習”, 日本機械学会論文集 C 編, Vol.62, No.602, pp.3746-3751, 1996
- [26] 高橋泰岳, 浅田稔, “実ロボットによる行動学習のための状態空間の漸次的構成”, 日本ロボット学会誌, Vol.17, No.1, pp.118-124, 1999
- [27] 伊庭齊志, “遺伝的アルゴリズムの基礎 GA の謎を解く”, オーム社, 1994
- [28] 北野宏明, “遺伝的アルゴリズム (1)”, 産業図書, 1993
- [29] 北野宏明, “遺伝的アルゴリズム (2)”, 産業図書, 1995
- [30] 北野宏明, “遺伝的アルゴリズム (3)”, 産業図書, 1997

- [31] 北野宏明, “遺伝的アルゴリズム (4)”, 産業図書, 2000
- [32] F. Mondada, D. Floreano, “Evolution of neural control structures: Some experiments on mobile robots”, *Robotics and Autonomous Systems*, Vol.15, pp.183-195, 1994
- [33] D. Floreano, F. Mondada, “Automatic Creation of an autonomous agent: Genetic evolution of a neural network driven robot”, *Proc. of the third International Conference on Simulation of Adaptive Behavior*, pp.421-430, 1994
- [34] R. Beer, J. Chiel, L. Sterling, “An artificial insect”, *American Scientist*, 79, pp.444-452, 1989
- [35] D.T.Cliff, I. Harvey, P.Husbands, “Explorations in Evolutionary Robotics”, *Adaptive Behavior* 2, pp.73-110, 1993
- [36] C.W. Reynolds, “An Evolved, Vision-Based Model of Obstacle Avoidance Behavior”, *ARTIFICIAL LIFE III*, pp.327-346, 1994
- [37] D. Whitley, K. Mathias, P.Fitzhorn, “Delta coding: An iterative search strategy for genetic algorithms”, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pp.77-84, 1991
- [38] O. Miglino, H.H. Lund, S. Nolfi, “Evolving Mobile Robots in Simulated and Real Environments”, *Artificial Life* 2, pp.417-434, 1995
- [39] N. Jacobi, P. Husbands, I.Hervey, “Noise and the Reality Gap: The Use of Simulation in Evolutionary Robotics”, *Third European Conf. on Artificial Life (ECAL95)*, *Advances in Artificial Life*, pp.704-720, Springer, 1995
- [40] J. Torren, “Reconfigurable Logic Applied for Designing Adaptive Hardware Systems”, *International Conference on Advances in Infrastructure*

for Electronic Business, Education, Science, and Medicine on the Internet (SSGRR 2002W), 2002

- [41] K.A. Vinger, J. Torresen, “Implementing Evolution of FIR-Filters Efficiently in an FPGA”, In proc. of 2003 NASA/DoD Conference on Evolvable Hardware (EH-2003), 2003
- [42] 村川正宏, 河西勇二, 高橋栄一, 坂無英徳, ニール マーストン, 桐生昭吾, 樋口哲也, “アナログ進化型ハードウェアの研究開発”, 電子情報通信学会技術研究報告 AI2000-10, pp.1-8, 2000
- [43] D. Keymaulen, M. Iwata, K. Konaka, R. Suzuki, Y. Kuniyoshi, T. Higuchi, “Off-line Model-free and On-line Model-based Evolution for Tracking Navigation using Evolvable Hardware”, Proceedings of the First European Workshop on Evolutionary Robotics, pp. 208-223, Springer Verlag, 1998
- [44] M. Murakawa, S. Yoshizawa, I. Kajitani, T. Higuchi, “On-line Adaptation of Neural Networks with Evolvable Hardware”, Koza et al. eds., Genetic Programming 1998: Proc. of the Third Annual Conference, Morgan Kaufmann, pp. 816-823, 1998.
- [45] 樋口哲也, 丹羽竜哉, 伊庭斉志, “進化するハードウェア”, 北野編, “グラウンドチャレンジ”, pp.38-52, 共立出版, 1993
- [46] M. Murakawa, S. Yoshizawa, I. Kajitani, X. Yao, N. Kajihara, M. Iwata, T. Higuchi, “The GRD Chip: Genetic Reconfiguration of DSPs for Neural Network Processing”, IEEE Transactions on Computers, vol. 48, no. 6, pp.628-639, 1999
- [47] E. Kandel, J. Schwartz, T. Jessell, “Essentials of Neural Science and Behavior”, Prentice Hall International, 1995

- [48] P. Meyrand, J. Simmers, M. Moulines, “Construction of a pattern-generating circuit with neurons of different networks”, NATURE, Vol.351, 2MAY, pp.60-63, 1991
- [49] 柳原大, 伊藤聡, “歩行運動の適応制御と小脳”, 生物物理, Vol.39, No.3, pp.165-171, 1999
- [50] John E. Dowling, “NEURONS AND NETWORKS An Introduction to Behavioral Neuroscience”, BELKNAP HARVARD, pp241-249.
- [51] R.J. Haycock, T.A. York, “Hardware implementation of a pulse-stream neural network”, Circuits, Devices and Systems, IEE Proceedings, Vol. 145, Issue 3, pp. 141-147, 1998
- [52] M. Marchesi, G. Orlandi, F.Piazza, L. Poonara, A. Uncini, “Multi-layer Perceptrons with Discrete Weights”, Proceedings of IJCNN’90, vol. 2, pp.623-630, 1990
- [53] 安藤道則, 太田則一, 渡邊章弘, “高集積化を目指した簡易型ニューロンの設計”, 豊田中央研究所 R&D レビュー, Vol.31, No.4, pp.13-21, 1996
- [54] 川島毅, 石黒章夫, 大熊繁, “小規模回路で可能なニューラルネットワークのハードウェア化手法”, 電子情報通信学会技術研究報告, NC99-90, pp.23-28, 2000
- [55] 川島毅, 石黒章夫, 大熊繁, “確率演算に基づいたパルスニューロンによるニューロハードウェアの小規模化”, 電気学会論文誌 C, pp.586-593, 2001
- [56] 國吉康夫, 大村吉幸, 寺田耕志, 長久保晶彦, “ヒューマノイドのコツに基づく全身運動の摸倣”, 日本機械学会ロボティクス・メカトロニクス講演会 (ROBOMECH04), 2004
- [57] 國吉康夫, 大村吉幸, 寺田耕志, 長久保晶彦, “等身大ヒューマノイドロボットによるダイナミック起き上がり行動の実現”, 日本ロボット学会誌, Vol.23, No.6, pp.706-717, 2004

- [58] 中村仁彦, “非ホロノミックロボットシステム 第2回幾何学的な非ホロノミック拘束下での運動計画”, 日本ロボット学会誌, Vol.11, No.5, pp.655-662, 1993
- [59] 中村仁彦, “非ホロノミックロボットシステム 第3回幾何学的な非ホロノミック拘束下での運動制御”, 日本ロボット学会誌, Vol.11, No.6, pp.837-844, 1993
- [60] 中村仁彦, 江崎秀明, 鄭宇眞, “非ホロノミック・トレーラシステムの操舵機構設計と制御”, 日本ロボット学会誌, Vol.17, No.6, pp.839-847, 1999
- [61] 田中一男, 小寄貴弘, “並列分散的補償の概念を用いたトレーラ・トラックタイプロボットのファジィ後退制御”, 計測自動制御学会論文集, Vol.32, No.3, pp.363-368, 1996
- [62] 田中一男, “アドバンストファジィ制御”, 共立出版, pp.140-139, 1994
- [63] D. Nguyen, B. Widrow, “The Truck Backer-Upper: An Example of Self-Learning in Neural Networks”, International Joint Conference on Neural Networks, pp. II-357-II363, 1989

業績

投稿論文

1. “Toward Seamless Transfer from Simulated to Real Worlds: A Dynamically-Rearranging Neural Network Approach”
Peter Eggenberger, Akio Ishiguro, Seiji Tokura, Toshiyuki Kondo, Yoshiki Uchikawa
Advances in Robot Learning(Eds.J.Wyatt and J.Demiris), Lecture Notes in Artificial Intelligence 1812, Springer, pp.44-60, 2000.
2. “神経修飾機構を有するニューラルネットワークのハードウェア化とその自律移動ロボットへの実装”
十倉征司, 石黒章夫, 大熊繁
電気学会論文誌 C(電子・情報・システム部門誌), Vol. 124-C, pp.2336-2344, 2004.
3. “Hardware Implementation of Neuromodulated Neural Network for a CPU-less Autonomous Mobile Robot”
Seiji Tokura, Akio Ishiguro and Shigeru Okuma
Advanced Robotics, Volume 20, No. 12, pp. 1341-1358

国際学会

1. , “The Effect of Neuromodulations on the Adaptability of Evolved Neurocontrollers”

Seiji Tokura, Akio Ishiguro, Hiroki Kawai, Peter Eggenberger
6th European Conf. on Artificial Life (ECAL2001), Advances in Artificial Life(Eds. J.Kelemen and P.Sosik), (Lecture Notes in Artificial Intelligence 2159), Springer, pp.292-295, 2001.

2. “Analysis of Adaptability of Evolved Neurocontroller with Neuromodulations”

Seiji Tokura, Akio Ishiguro, Hiroki Kawai, Peter Eggenberger
Intelligent Autonomous Systems 7 (IAS-7) (Eds.Maria Gini, Wei-Min Shen and Hideo Yuasa), pp.341-348, 2002.

3. “Concurrent Creation of Behavior and Segmentation by a Neuromodulated Neural Network”

Seiji Tokura, Akio Ishiguro, Peter Eggenberger Hotz
SICE Annual Conference 2003, pp.715-719 , 2003.

4. “A Hardwired Polymorphic Neuralnetwork for a CPU-less Autonomous Mobile Robot”

Seiji Tokura, Akio Ishiguro
2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2003), pp.1216-1221, 2003.

国内学会

1. “動的再編成機能を有する神経回路モデルを用いた自律移動ロボットの行動制御 コントローラの頑健性の実験的検証”

石黒章夫, 十倉征司, 近藤敏之, 内川嘉樹, Peter Eggenberger
ロボティクス・メカトロニクス講演会 99, 1A1-28-039, 1999.

2. “動的再編成機能を有するニューラルコントローラの提案：シミュレーションと実環境間のギャップギャップ吸収”

石黒章夫，十倉征司，近藤敏之，内川嘉樹，Peter Eggenberger
第 38 回計測自動制御学会学術講演会論文集, pp.359-360, 1999.

3. “頑健なコントローラはいかにして構築できるか？動的再編成機能を有するニューラルネットワークの提案”
石黒章夫，十倉征司，近藤敏之，内川嘉樹，Peter Eggenberger
第 5 回創発システムシンポジウム 99 資料,1999.
4. “シミュレーションから実環境へのシームレスな移行の実現 動的再編成機能を有する神経回路モデルの頑健性の実験的検証 ”
十倉征司，石黒章夫，近藤敏之，内川嘉樹，Peter Eggenberger
第 17 回日本ロボット学会学術講演会予稿集，pp.645-646, 1999.
5. “動的再編成機能を有するニューラルネットワークによる頑健生の実現”
十倉征司，石黒章夫，近藤敏之，内川嘉樹，Peter Eggenberger
第 8 回インテリジェントシステム・シンポジウム講演論文集，pp.209-214, 1999.
6. “発生過程を導入した動的再編成機能ニューラルネットワークの進化的構築”
十倉征司，石黒章夫，内川嘉樹，Peter Eggenberger
ロボティクス・メカトロニクス講演会 00, 2P1-31-031，2000.
7. “動的再編成ニューラルコントローラの頑健性の定量的検証”
河合 宏紀，十倉 征司，Peter Eggenberger，石黒 章夫
ロボティクス・メカトロニクス講演会 01, 2P1-B8，2001.

8. “神経修飾メカニズムを導入した完全 CPU-less ロボットの構築”
杉山 真史, 十倉 征司, 石黒 章夫, 川島 毅
ロボティクス・メカトロニクス講演会 02, 2A1-L06, 2002.
9. “動的再編成ニューラルネットワークの適応能力の定量的検証”
河合 宏紀, 十倉 征司, 石黒 章夫, Peter Eggenberger
ロボティクス・メカトロニクス講演会 02, 1P1-E03, 2002.
10. “神経修飾機能を有するニューラルネットワークによる行動生成と分節化”
十倉征司, 石黒章夫, Peter Eggenberger Hotz
ロボティクス・メカトロニクス講演会 03, 2A1-3F-C2, 2003.
11. “自律移動ロボットにおける行動と分節化の同時生成”
十倉征司, 石黒章夫
第 21 回日本ロボット学会学術講演会予稿集, 3B32, 2003.

著書

1. “知の創成 身体性認知科学への招待 ” 14 章翻訳担当
共訳
共立出版, ISBN 4-320-12032-9, 2001 (Translation of ”Understanding Intelligence” by Rolf Pfeifer and Christian Scheier, MIT Press, Cambridge, 1999)

賞

1. 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2003) Best Paper Award Nomination Finalist

謝辞

本研究を行うにあたり、研究の場を与えて下さると共に、様々な御指導、御鞭撻を賜りました名古屋大学大学院工学研究科計算理工学専攻教授 故内川嘉樹先生、ならびに名古屋大学大学院工学研究科電子情報システム専攻教授 大熊 繁先生に深く感謝いたします。加えて学士取得時から、研究活動において並々ならぬ御指導、御鞭撻を賜りました東北大学大学院工学研究科電気・通信工学専攻教授 石黒 章夫先生に深く感謝の意を表します。また、本論文執筆にあたり、多くの貴重なご助言を賜りました名古屋大学大学院工学研究科計算理工学専攻教授 古橋 武先生、名古屋大学大学院情報科学研究科情報システム学専攻教授 高濱 盛雄先生、名古屋大学大学院工学研究科電子情報システム専攻助教授 道木 慎二先生に感謝いたします。

研究ならびに学生生活において多くの助言をいただきました東京農工大学工学府情報工学科講師 近藤 敏之先生に心より感謝いたします。また、研究を進めるにあたり、献身的に協力を頂いた高畑 敦志氏、河合 宏紀氏、杉山 雅史氏、茶木 健至氏に感謝いたします。加えて、有益な御助言、ご協力をいただきました藤井 亮暢氏、石丸 和寿氏、川角 健太氏、斎藤 信宏氏、西井 康人氏、清水 正宏氏、早川 宏治氏、梅舘 拓也氏、鈴木 梓氏、山元 秀洋氏、森 祐樹氏、に厚く御礼申し上げます。さらに、9年間に渡り多くの苦楽を共にした、中平 健治氏、船瀬 新王氏に心より感謝します。

また、多大な迷惑をかけたにもかかわらず、日頃の研究活動を暖かく見守り、時に叱咤激励してくれた職場の方々、大熊研究室と古橋研究室の秘書の方々および友人の皆様我心からの感謝の意を表します。

最後に、末筆ながら、本研究を進めるにあたり陰ながら研究生生活を支えてくれた妻に心から感謝します。