

複数ロボットの協調作業における
役割分担の生成手法に関する研究

1999年3月

大 川 一 也

①

目次

第1章 序論

複数ロボットの協調作業における
役割分担の生成手法に関する研究

- 1.1 本論文の目的
- 1.2 本論文の構成

第2章 作業分担の生成アルゴリズムの構築

- 2.1 導入
- 2.2 役割分担の生成アルゴリズムの構築
- 2.3 アルゴリズムの検証
- 2.4 アルゴリズムの実装
- 2.5 アルゴリズムの評価
- 2.6 まとめ

1999年3月

大川 一也

目次

目次

第 1 章 緒論

- 1. 1 本研究の背景 1
- 1. 2 従来の研究 2
 - 1. 2. 1 一对多の制御手法 3
 - 1. 2. 2 作業分担の生成手法 6
- 1. 3 本研究の目的 8
- 1. 4 本論文の構成 9

第 2 章 作業分担の生成アルゴリズムの構築

- 2. 1 はじめに 12
- 2. 2 役割分担の自律生成に必要な機能 14
- 2. 3 アルゴリズムの構築 15
 - 2. 3. 1 アルゴリズムの基本構成 15
 - 2. 3. 2 行動群モジュール 17
 - 2. 3. 3 センサモジュール 18
 - 2. 3. 4 作業選択モジュール 20
 - 2. 3. 5 評価・学習モジュール 22
 - 2. 3. 6 フラストレーションモジュール 26
- 2. 4 アルゴリズムの処理の流れ 27
- 2. 5 アルゴリズムの検証 29
 - 2. 5. 1 ロボット数の影響 29
 - 2. 5. 2 作業数の影響 33
 - 2. 5. 3 考察 34
- 2. 6 まとめ 36

目次

第 3 章 指令に基づく群れの制御手法の提案

| | | |
|---------|------------------|----|
| 3. 1 | はじめに | 37 |
| 3. 2 | 一対多を扱う制御手法 | 38 |
| 3. 3 | 群れの制御手法の提案 | 40 |
| 3. 3. 1 | 制御手法の概略 | 40 |
| 3. 3. 2 | 指令の種類と伝達手法 | 42 |
| 3. 3. 3 | 行動評価の修正手法 | 43 |
| 3. 3. 4 | 行動評価修正法の改良 | 45 |
| 3. 4 | 複数ロボットの制御手法の検証 | 48 |
| 3. 4. 1 | 指令に基づく協調行動の生成 | 48 |
| 3. 4. 2 | 指令に基づく群れの制御手法の検証 | 52 |
| 3. 4. 3 | 考察 | 53 |
| 3. 5 | まとめ | 55 |

第 4 章 群れの効果的制御手法の提案

| | | |
|------------|-------------|----|
| 4. 1 | はじめに | 56 |
| 4. 2 | 群れの誘導手法 | 57 |
| 4. 2. 1 | 誘導手法の提案 | 57 |
| 4. 2. 2 | 群れの誘導手法の検証 | 61 |
| 4. 3 | 小集団毎の通信 | 65 |
| 4. 3. 1 | 通信量と学習領域の関係 | 65 |
| 4. 3. 2 | 小集団の形成 | 65 |
| 4. 3. 3 | 小集団毎の通信の検証 | 69 |
| 4. 3. 3. 1 | 小集団の大きさの決定 | 69 |
| 4. 3. 3. 2 | ロボット数の影響 | 72 |
| 4. 3. 3. 3 | 作業数の影響 | 75 |
| 4. 4 | まとめ | 77 |

目次

| | |
|-----------------------|-----|
| 第 5 章 実作業を想定した手法の検討 | |
| 5. 1 はじめに | 78 |
| 5. 2 ロボットと操作者の装置 | 79 |
| 5. 2. 1 ロボットに必要な装置 | 79 |
| 5. 2. 2 操作者に必要な装置 | 81 |
| 5. 3 適用する作業例 | 82 |
| 5. 4 実作業を想定したシミュレーション | 83 |
| 5. 4. 1 シミュレーション条件 | 83 |
| 5. 4. 2 シミュレーション結果 | 85 |
| 5. 4. 3 考察 | 86 |
| 5. 5 まとめ | 91 |
| 第 6 章 結論 | 92 |
| 謝辞 | 95 |
| 参考文献 | 96 |
| 関連業績一覧 | 105 |

第1章 緒論

1. 1 本研究の背景

現在，実用化されているロボットの一つとして産業用ロボットが挙げられる．この種のロボットの多くは，センサが獲得した環境情報に基づいて，ある程度，自己の動作を修正できるものの，遂行する作業計画から細かい動作に至るまで，人間によって与えられたプログラムに記述されている動作を忠実に実行しているのが現状である^{1),2)}．このため，一般に，作業の変更に対する融通性がないといわれているが，そこで適用されているロボットは汎用型ロボットであり，状況に応じて適切なプログラムを与えることで，多彩な作業を遂行できる可能性を持っている．もちろん，一般的な環境で作業を遂行することは困難であるが，ある程度の定められた環境に限定すれば，近年の数々のロボット技術を統合することで，近い将来，多彩な作業を遂行できるロボットが実現されると予想される³⁾．

その一方で，単体のロボットに高度な機能や能力を搭載しても遂行できる作業には限界があることが指摘され⁴⁾，複数ロボットの協調による多機能化の必要性⁵⁾⁻⁷⁾が認識されてきた．複数のロボットに整合性のある協調行動をとらせることができれば，環境の変化に対する柔軟性や頑健性，効率的処理などの利点が期待できるため，1990年代前半から複数のロボットによる協調行動に関する研究が盛んに行われている^{8),9)}．

このようなロボットの研究の流れから，将来，多彩な作業を自律的に遂行できるロボットを複数台用意し，状況に応じ

て作業を分担させるシステムの実現が期待される¹⁰⁾。この場合、ロボットを管理する操作者は、自己の意図に応じて複数ロボットの作業分担を制御する必要がある。

操作者の意図した作業分担を複数のロボットに行わせるための最も単純な手法は、ロボットを管理する操作者が全てのロボットにあらかじめ作業を割り振り、各役割に応じた専用のプログラムを個別に与えることである。しかし、この手法の場合、ロボットに異なる作業分担を行わせるためには、操作者が新しい役割を再計画し適切なプログラムを与え直さなければならないと、作業の変化に対する融通性がない。また、全てのロボットに対し個別にプログラムを与えなければならないことから、ロボット数が増加すると操作者の負担が増大するという問題も生じる。つまり、一人の人間が個々のロボットを個別に操作する一対一の制御手法では、複数のロボットに操作者の意図した作業分担を行わせることが困難であることから、複数ロボットを制御するための手法が必要と考えられる。

1. 2 従来の研究

複数のロボットに操作者の意図した作業分担を行わせる手法は、大きく二つに分類できる。一つは、一人の操作者が複数のロボットを操作するための、一対多の制御手法の研究であり、もう一つは、複数ロボットによる作業分担の生成手法の研究である。

本節では、これらの従来の研究について、それぞれ述べる。

1. 2. 1 一対多の制御手法

一人の操作者が複数のロボットの行動を制御する研究は、ロボットに遂行させる作業内容や制御するロボット数などによって以下の5種類に大別できる。これらは、作業遂行中に操作者が関与する割合の少ない順に並べてある。

- (1) 自律分散的手法
- (2) 掲示板を用いる手法
- (3) 作業支援による手法
- (4) 共同作業による手法
- (5) 直観的な動作を用いる手法

(1)の自律分散的手法は、システム全体を統合する管理機構をもたず、システムを構成する各ロボットが分散して存在し、それぞれ自律的に行動することにより要素間の協調をとり、全体として任務を達成しようとする手法^{3),11)~14)}である。これらの手法として、例えば、村田らの研究^{15),16)}や Kubeらの研究^{17),18)}などがある。ここでは、個々のロボットが自律的に行動することができるため、非常に多くのロボットが存在する場合にも適用できる点で優れている。しかし、ロボット間で協調するためのルールは、操作者が事前に各ロボットに与えておく必要があり、状況に応じて協調の仕方を教える事はできない。

(2)の掲示板を用いる手法では、環境内に幾つかの掲示板を用意しておき、操作者がそれらの掲示板に指令を書きこむ手法である。操作者の意図は、ロボットが掲示板を直接読むことで伝達される。また、その情報はロボット間のクチコ

ミによって周囲のロボットに広められる。この種の研究は、新井ら¹⁹⁾をはじめとする多くの研究者^{20),21)}が行っている。ここでは、操作者からの指令が掲示板を中心に局所的にしか広がらないため、掲示板ごとに異なる行動を取らせることもできる。しかし、ここでは掲示板の数や配置が重要な意味を持つため、この手法を適用できる問題は限られている。すなわち、掲示板が増えると、それ毎に個別の指示を与える必要があり、ロボット数が増加すると指示が複雑になる。

(3)の作業支援による手法では、作業内容などは操作者が直接指示するものの、局所的な処理は個々のロボットに自律的に行わせる手法である。例えば、森島らの研究²²⁾では、操作者がコンピュータ上のパネルなどで作業対象や作業内容を選択するものの、その後の具体的な処理はロボットによって自律的に行わせている。また、横田らの研究²³⁾では、状況に応じて操作者が複数のロボットと対話することでロボット間の協調行動を支援する手法を検討している。さらに、中村らの研究^{24),25)}では、対象物の協調搬送を取り扱い、搬送する方向や速度は操作者が与えるものの、ロボット間の同期などの処理は個々のロボットに行わせている。これらの手法の場合、複雑な作業も適用できるが、いずれも基本的に操作者がロボットの行動を見て、その様子から個々のロボットに行動を個別に指示することを含み、ロボットの数が増えると問題が生じる。

(4)の共同作業による手法では、操作者が一台のロボットを直接操作することで、操作者が複数ロボットのメンバーの一員となる手法である。この種の研究として、例えば、木村らの研究²⁶⁾がある。ここでは、人間によるロボットの操縦を通して操作者の意図を各ロボットに理解させ、適切な補助を

させようとする研究である。しかし、操作者が操作するロボットの動作から操作者の意図を理解させることは現在の技術では困難である。一方、Chen らの研究²⁷⁾のように、操作者が操作するロボットに忠実に追従させるといった単純な機能を他のロボットに持たせることで、ロボットの群れ全体の行動を操作することを可能としている。しかし、これらの手法も、取り扱える作業は特定の作業に限定されている。

(5) の直観的な動作を用いる手法は、長谷川ら²⁸⁾によって研究されている。ここでは、複数のロボットをディスプレイ上に表示し、机上にいる複数の微小対象物を扱うかのような直観的な動作で、複数ロボットの動きを操作する手法である。ここでは、個々のロボットを個別に観察するのではなく、全体として観察し、直観的な操作で群れを操作する点において、他の制御手法と違った特徴を持っている。しかし、この手法はマイクロロボットの操作として提案された手法であり、個々のロボットの移動方向を指示するような個々のロボットの単純な役割を指示する際には効果を持つが、複数の作業分担の指示に利用するには問題がある。

このように、一人の操作者が複数のロボットを制御する手法に注目した場合、大きく5種類の方法が考えられるが、いずれも特定の作業を想定した手法であったり、ロボットの配置がその効果に重要な意味を持っていたりするため、これらの手法を多数かつ複雑な役割分担の生成を要する群ロボットシステムに適用することはできないと考える。

本研究において、操作するロボット数が非常に多い場合の問題について検討したところ、二つの問題が想定された。一つ目の問題は、ロボットが多数存在することで、操作者は個々のロボットを個別に微細に観察できないという点にある。こ

のような状況では，人間はロボット全体に対する大まかな判断しかできない．このため，群れ全体に対する大まかな判断からロボットを制御する手法を検討しなければならないと考える．二つ目の問題は，ロボットが多数台存在することで，操作者は個々のロボットに個別の指示を与えられないという点にある．つまり，人間がロボットを操作する場合には，複数ロボットの群れ全体に対する指令しか与えざるを得ないことを意味する．このため，個々のロボットは，群れ全体に対する共通の指令から作業分担を生成する手法を検討しなければならないと考える．

本研究では，これらの問題を考慮した複数ロボットの制御手法について検討を行う必要があると考える．

1. 2. 2 作業分担の生成手法

複数のロボット間の作業分担を自律生成する研究は，マルチエージェントロボットシステム（Multi-Agent Robotic Systems, 以降 MARS と呼ぶ）の分散プランニングの分野において行われている．例えば，複数ロボットが対象物の重さに応じて適切に力加減して協調運搬する研究²⁹⁾⁻³⁶⁾，交差点などの共有空間で衝突を回避させる研究³⁶⁾⁻⁴¹⁾，サッカーロボットなどにおいてボールを連携パスする研究⁴²⁾⁻⁴⁵⁾などがある．ここで，MARSとは，複数のエージェント（動作主体）が集まり，共通の目標を持って，作業や処理などを行うロボットシステムとして定義されており⁴⁶⁾，エージェントの個々の制御から整合性のある全体の挙動を得ようとする研究⁴⁷⁾⁻⁵⁰⁾である．

複数のロボットの作業分担を生成するためには，まず，他

のロボットがどのように行動しているかの情報を個々のロボットが獲得する必要があるため、ロボット間の通信手法⁵¹⁾⁻⁵⁴⁾が重要課題として研究されている。ここでの通信手法は、集中的に管理するエージェントが存在する場合と存在しない場合の二つに大別できる。

集中的に管理をおこなうエージェントが存在する通信手法は、非常に多く研究⁵⁵⁾がなされている。この種の研究として、Smithが提案したContact Net Protocolと呼ばれる契約ネット⁵⁶⁾がある。ここでは、まず、全体の作業を管理する役のロボットが他のロボットに作業を依頼し、最も適切なロボットと契約を結ぶという手法である。また、Papeら⁵⁷⁾は、屋内環境で様々な作業を行うための研究を行っている。ここでは、個々のロボットに対する作業分担を集中的に計画するものの、各ロボットの具体的な動作は各々が分散して行うといった集中・分散混合型の手法が採用されている。いずれの場合においても、集中的に作業を管理するエージェントが存在するため、全体として適切な作業分担が行えるが、集中的ゆえにロボットの台数が増加すると全体を管理することができなくなるという問題がある。

これに対し、集中的に管理するエージェントが存在しない手法⁵⁸⁾⁻⁶⁰⁾も多く提案されている。例えば、浅間ら⁶¹⁾は、単体のロボットでは遂行できない作業に遭遇すると、通信を使って他のロボットに支援を依頼する研究を行っている。また、油田ら⁶²⁾は、複数のロボットが同一の空間を共有する場合において、互いに衝突しないように共有空間を譲り合う研究を行っている。さらに、國吉ら⁶³⁾⁻⁶⁵⁾は、無線などの通信は用いず、観察に基づく協調の研究を行っている。これらの研究では、いずれの手法も、不都合な状態に遭遇した時点での局

所的な役割分担が対象となっており，複数ロボット全体としての作業分担生成を論じたものではない。

このように，非常に多数のロボットが存在する場合の作業分担を取り扱っている研究は非常に少ないといえる。

1. 3 本研究の目的

本研究では，幾つかのサブタスクの組み合わせで達成される作業を複数のロボットに協調して行わせるための，各ロボットにおける作業分担の自律的生成法と，こうした複数ロボットに一人の人間が効率的に指示を与えるための，一对多のヒューマン・ロボットインタフェースについて検討することを目的とする。

複数ロボットにおける作業分担の最も単純な手法としては，これまでの節でも触れたが，人間がロボットの作業分担を計画し，個々のロボットに作業を実行するプログラムを直接与える手法が考えられる。この手法では，作業分担の計画からプログラムの埋め込みまでを人間が行わなければならない，ロボット数が増加すると人間の負担が増大し，実用的ではない。また，作業の変更に柔軟に適應することも困難である。

そこで，本研究では，人間が要求する作業の変化にも柔軟に対応しつつ効率的に個々のロボットが適切な作業を実行できるシステムの構築を目指し，それぞれのロボットが均一のプログラム構造を持ち，人間から与えられる作業目標とロボット間の情報交換に基づき，各ロボットが自律的に自己が実行すべき作業を発見し，遂行するアルゴリズムを提案する。

具体的には，このアルゴリズムは，各作業をそれぞれ実行するための複数のプログラムを内蔵する行動モジュール群，

それらを選択する行動選択モジュール，他のロボットの行動を観察するセンサモジュール，作業を指示する人間が与える評価と周囲のロボットの行動観察結果から行動選択モジュールに選択信号を与える，学習機能を内蔵した評価学習モジュールから構成される．全てのロボットは共通にこれらのアルゴリズムを内蔵し，それに基づいて作業を決定し行動する．

また，こうしたアルゴリズムを複数ロボットシステムに適用し，ロボット数や作業数などの種々のシステムパラメータに対するアルゴリズムの性能をシミュレーションで確認するとともに，ロボット数や作業数の増加に伴って発生する課題を指摘する．次に，それらを解決する手法として，ロボット間の通信法や人間のシステムへの効果的な介入法を提案し，最終的には具体的な作業を想定したシミュレーションによって，提案した手法の効果を確かめる．

1. 4 本論文の構成

第1章「緒論」では，複数ロボットによる作業分担の生成手法を論じることとなった，研究の背景及び目的について述べた．

第2章「作業分担の生成アルゴリズムの構築」では，複数ロボット間で作業分担を生成できる，全てのロボットに共通な基本的アルゴリズムを構築を行う．そのために，まず，ロボットに必要な機能および能力について検討する．次に，それらの検討結果を踏まえながら，ロボットの各機能のモジュール化を行う．この際，各モジュールに必要な前提条件や処理する情報の内容などを明確にし，今後，アルゴリズムを構成する時に無理が生じないような配慮を行う．こうして作ら

れたモジュールを組み合わせることで、作業分担を行うためのロボットの基本アルゴリズムを構築し、その後、全体的な処理の流れを通してアルゴリズムの働きを述べる。さらに、単純なシミュレーションを用いて、構築したアルゴリズムの有効性を検証すると共に、残された問題点を明確にする。

第3章「指令に基づく群れの制御手法の提案」では、ロボット全体を管理する操作者によって、ロボットの作業分担を制御するための手法について検討する。ここでは、まず、一人の人間が複数のロボットを操作するという一対多の制御手法における問題点を明確にし、そのような状況においても獲得できる情報の種類について検討する。次に、それらの検討結果および第2章で提案するアルゴリズムの特徴を考慮した上で、操作者が複数ロボットを操作するための手法を提案する。そして、この手法を用いることで、単純な指令から、複数ロボットの作業分担が制御できることをシミュレーションで検証する。

第4章「群れの効果的制御手法の提案」では、ロボット数や作業数が多くなると、目標の作業分担に到達するのに多くの試行数が必要となるという問題に焦点を絞り、それらを解決する手法について検討する。まず、これまでのシミュレーションでは報酬信号を得るまでに非常に多くの試行数が必要であったことに着目し、ロボットの群れを誘導する手法を提案する。次に、ロボット数や作業数の増加に伴い、ロボットが獲得する状態量が増大する点に着目し、その解決策として小集団の形成を行う。

第5章「実作業を想定した手法の検討」では、第4章までに構築してきたアルゴリズムや制御手法を実作業に適用する場合を想定した場合の検討を行う。ここでは、まず、本論文

の手法を実現させるために必要な装置について述べ、現存の機器でも実現可能であることを示す。次に、本論文での手法が有効的に利用できる具体的な作業例について述べる。さらに、実作業を想定した仮想環境をコンピュータ上に構築し、その中で動作する複数のロボットの作業分担を、実際の間が操作するシミュレーション実験を行った。そこでは、実際に人間がロボットの群れ全体に対する単純な指令を与えるだけで、ロボットの作業分担を操作できることをシミュレーションによって確認する。

最後に、第6章「結論」において、本研究の成果を総括して述べる。

第2章 作業分担の生成アルゴリズムの構築

2.1 はじめに

掃除という作業は、例えば、Fig.2-1のように、「机を運ぶ」「ごみを集める」「窓を拭く」といった幾つかのサブタスクから構成される。これらのサブタスクを適切に分担し並列的に処理すれば、作業を効率良く達成できる。複数ロボットにこうした協調作業を行わせるためには、個々のロボットが、作業の進捗状況や他のロボットが遂行するサブタスクに応じて、遂行するサブタスクを自律的に切り替える能力が必要である。そのような能力をロボットに持たせるためには、各ロボットに一つのサブタスクを実行する固定のプログラムを搭載するだけでは不十分である。

本研究では、作業を遂行するために必要な全ての行動プログラムをあらかじめ与えておき、状況に応じてそれらの行動プログラムを切り替える手法を提案する。この場合、個々のロボットが無作為にそれらの行動プログラムを選択したのでは、他のロボットとの適切な協調行動が生成されない。他のロボットの行動や人間からの指令を考慮して行動プログラムを選ぶ機能や、選んだ結果を評価しつつそれを今後の行動プログラム選択に反映させる機能などを検討する必要がある。本章では、まず、これらの検討を行い、その結果に基づいて、複数ロボットによる作業分担を自律生成するための基本アルゴリズムの構築を行う。

2.2 役割分担の生成アルゴリズムに必要な機能

状況に応じて自動的に作業を振り替え、他のロボットとの協働で適切な作業分担を生成する能力をロボットに持たせるための基本機能について述べる。

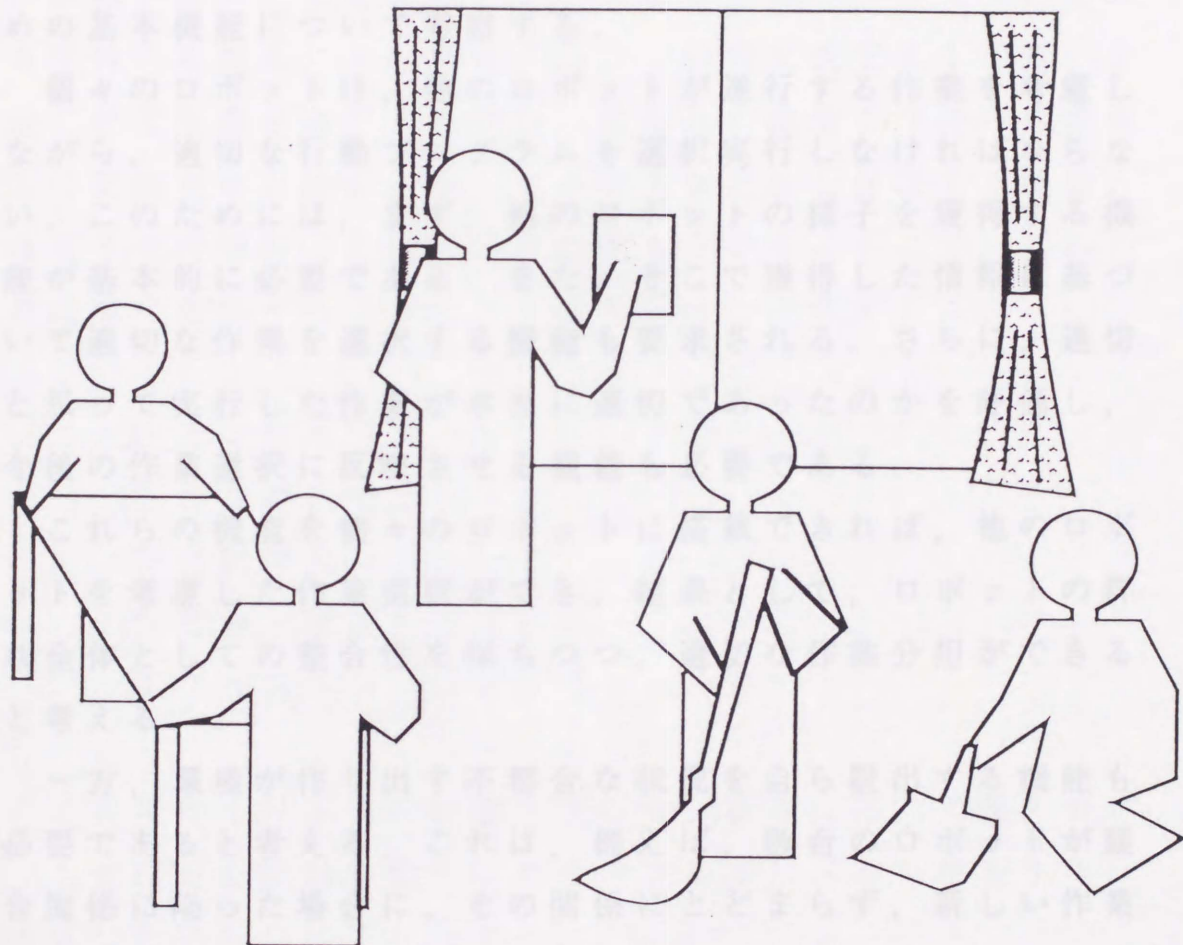


Fig.2-1 作業分担が必要な作業例

2.2 役割分担の自律生成に必要な機能

状況に応じて自律的に作業を切り替え、他のロボットとの間で適切な作業分担を生成する能力をロボットに持たせるための基本機能について考察する。

個々のロボットは、他のロボットが遂行する作業を考慮しながら、適切な行動プログラムを選択実行しなければならない。このためには、まず、他のロボットの様子を獲得する機能が基本的に必要である。また、そこで獲得した情報に基づいて適切な作業を選択する機能も要求される。さらに、適切と思って実行した作業が本当に適切であったのかを評価し、今後の作業選択に反映させる機能も必要である。

これらの機能を個々のロボットに搭載できれば、他のロボットを考慮した作業選択ができ、結果として、ロボットの群れ全体としての整合性を保ちつつ、適切な作業分担ができると考える。

一方、環境が作り出す不都合な状況を自ら脱出する機能も必要であると考えられる。これは、例えば、数台のロボットが競合関係に陥った場合に、その関係にとどまらず、新しい作業分担に切り替えることで、その状況を脱出する機能である。

以上が、複数のロボットによって作業分担を生成するために必要な基本機能であると考えられる。

2.3 アルゴリズムの構築

2.3.1 アルゴリズムの基本構成

前節で述べた基本機能をまとめると、他のロボットの情報を獲得する機能、その情報に応じて作業を決定する機能、決定された作業を遂行する機能、遂行した結果を今後の作業決定に反映させる機能、そして不都合な状況を脱出する機能、以上5つの機能が必要であると考えられる。そこで、これらの機能をそれぞれ以下のようにモジュール化する。

- ① センサモジュール
- ② 作業選択モジュール
- ③ 行動群モジュール
- ④ 評価・学習モジュール
- ⑤ フラストレーションモジュール

これらのモジュールを組み合わせ、状況に応じて他のロボットと作業を役割分担するためのアルゴリズムを構築する。構築したアルゴリズムを Fig.2-2 に示す。

このアルゴリズムを搭載したロボットは、個々のロボットの相互関係から自己が遂行すべき作業を決定し、自律的に遂行できる能力を持つ。このため、ロボットの管理者は、個々のロボットの役割を事前に決定したり、個々のロボットに専用のプログラムを与えたりする必要はなく、このアルゴリズムを全てのロボットに共通に与えることができる。次節では、このアルゴリズムを構成する各モジュールについての具体的な処理手順などを説明する。

2.3.1 行動群モジュール

状況に応じて他のモジュールと作業分担するためには、本システムでは、作業を実行するために必要な全ての行動プログラムをあらかじめ学習しておくことを必要としている。これらの行動プログラムは、Fig.2-1の構成を仮にすれば「机を運ぶ」「ゴミを拾める」「窓を拭く」といったタスクプログラムを実行する手順が記述されたプログラムに相当する。これらの行動プログラ

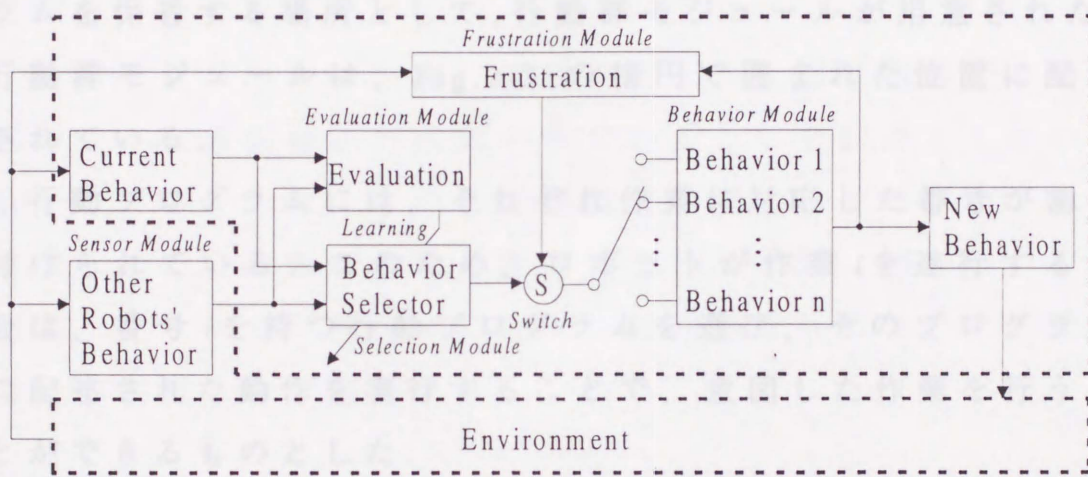


Fig.2-2 作業分担のための基本アルゴリズム

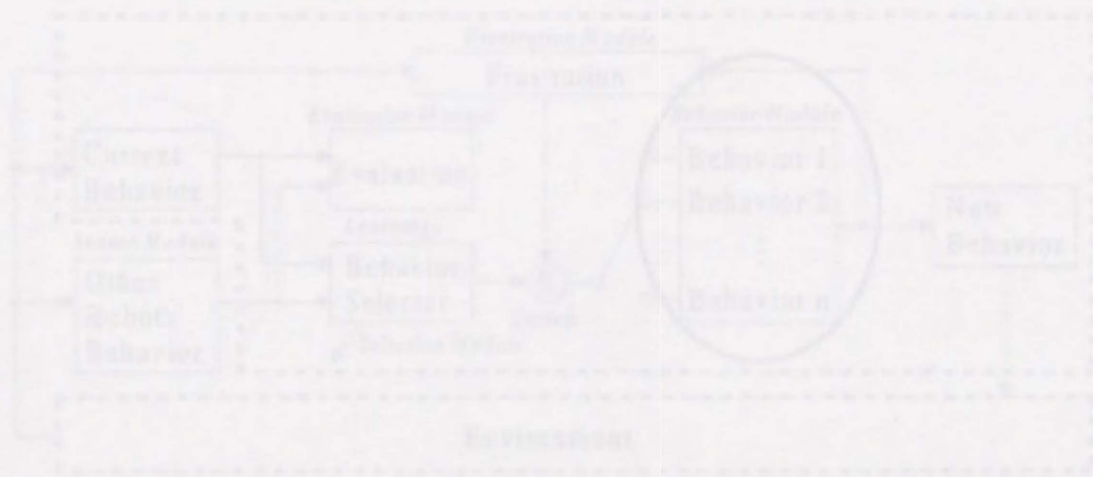


Fig.2-3 行動群モジュールの配置

2.3.2 行動群モジュール

状況に応じて他のロボットと作業分担するために、本研究では、作業を遂行するために必要な全ての行動プログラムをあらかじめ与えておくことを基本としている。これらの行動プログラムは、Fig.2-1の掃除を例にすれば「机を運ぶ」「ごみを集める」「窓を拭く」といったサブタスクを遂行する手順が記述されたプログラムに相当する。これらの行動プログラムを保管する場所として、行動群モジュールが用意された。行動群モジュールは、Fig.2-3の楕円で囲まれた位置に配置されている。

行動プログラムには、それぞれ作業に対応した番号が割り付けられている。このため、ロボットが作業*i*を遂行する場合は、番号*i*を持つ行動プログラムを選び、そのプログラムに記述された動作を実行することで、意図した作業を行うことができるものとした。

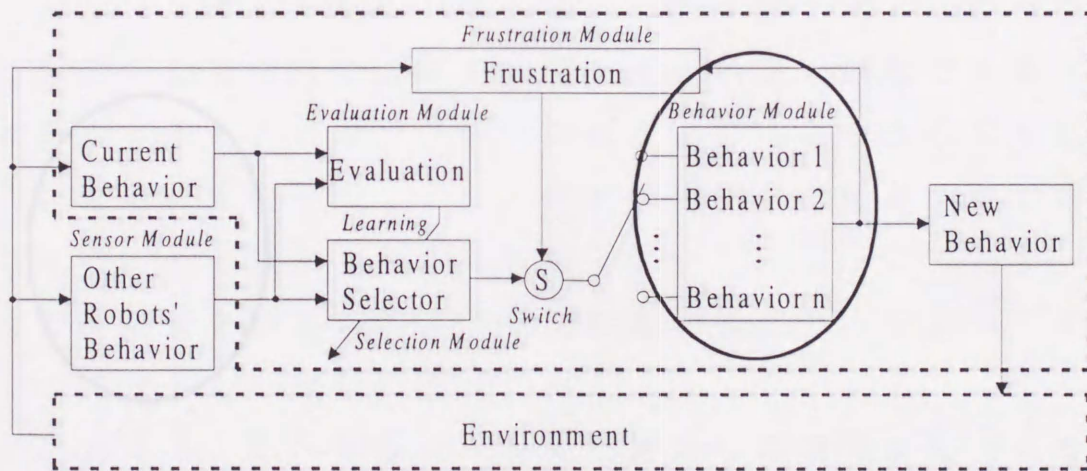


Fig.2-3 行動群モジュールの配置

2. 3. 3 センサモジュール

複数のロボットが全体として適切に作業を分担するためには、個々のロボットは他のロボットが遂行している作業を考慮しながら自己の作業を決定しなければならない。そのためには、まず、他のロボットが遂行している作業の情報を獲得する機能が必要であることから、センサモジュールが用意された。センサモジュールは、Fig.2-4 の楕円で囲まれた場所に配置されている。

センサモジュールには、他のロボットと相互通信できる無線のような通信機能が内蔵されていることを前提としている。個々のロボットは、その通信機能を用いて、他のロボットが遂行している作業の番号を獲得できるとした。個々のロボットは、センサモジュールで獲得した作業番号の集合を、その時点の状態として認識する。

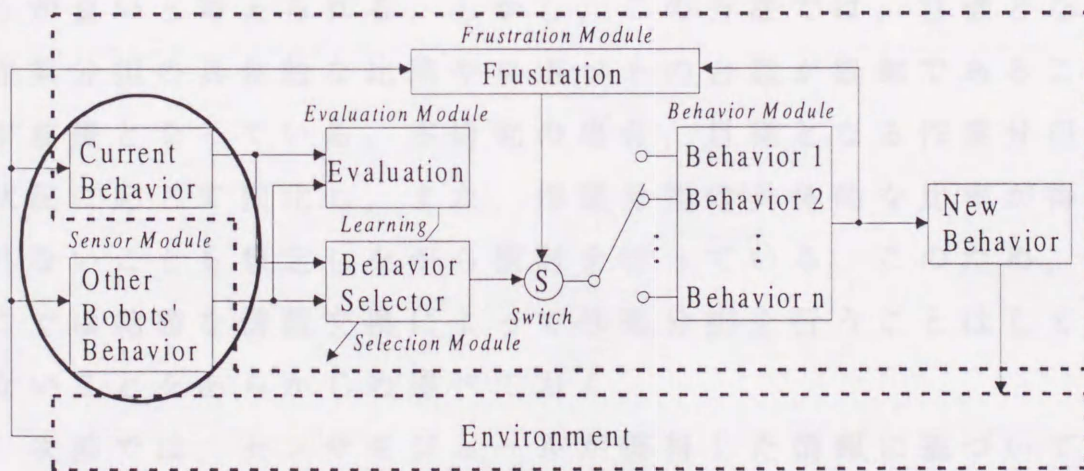


Fig.2-4 センサモジュールの配置

例えば、ロボット i が遂行している作業番号を p_i と記述するならば、時刻 t における自己の作業 a_t 及び状態集合 \mathbf{x}_t は、以下の式で表現できる。

$$\begin{aligned} a_t &= p_i \\ \mathbf{x}_t &= (p_1, p_2, \dots, p_{i-1}, p_{i+1}, \dots, p_m) \end{aligned} \quad (2-1)$$

ただし、 m は環境に存在するロボットの総数である。

従って、時刻 t において Fig.2-4 のセンサモジュール内の Current Behavior および Other Robots' Behavior には、それぞれ a_t と \mathbf{x}_t が獲得されている。これらの情報を、次節で説明する作業選択モジュールに受け渡すことで、他のロボットが遂行する作業を考慮した作業選択が行えるようにしている。

ここで、個々のロボットが他の全てのロボットと相互通信可能であるならば、単に、その時点の状態を獲得するだけでなく、綿密な情報交換のもとで全ての作業分担を決定したほうが良いと考えられる。しかし、この方法では、目標となる作業分担の具体的な比率やロボットの台数が既知であることが前提となっている。本研究の場合、目標となる作業分担は状況に応じて変化し、また、作業分担の具体的な比率が得られないことも想定しながら検討を行っている。このため、ここでは綿密な情報交換によって作業分担を行うことはしていないことをあらかじめ述べておく。

次節では、センサモジュールが獲得した情報に基づいて作業を決定するための、作業選択モジュールについて説明する。

2. 3. 4 作業選択モジュール

複数のロボットが全体として適切に作業を分担するためには，個々のロボットが状況に応じて適切に作業を決定し，対応した行動プログラムを選択実行しなければならない．この機能をロボットに持たせるために，作業選択モジュールを用意した．作業選択モジュールは，Fig.2-5 の楕円で囲まれた場所に配置されている．

状況に応じて適切に作業分担するためには，各状況においてどの作業が適切なかを判断しなければならない．この問題の解決法の一つとして，各状況における最適な作業選択をあらかじめルールとして与えておくことが考えられる．しかし，この場合，目標となる作業分担が事前に分かっている必要はなく，また，目標を変更する際にはルールを与え直さなければならない．そこで，本研究では，個々のロボットに学習機能を付加し，他のロボットが遂行する作業を考慮した適切な作業を自ら学習する手法を検討する．

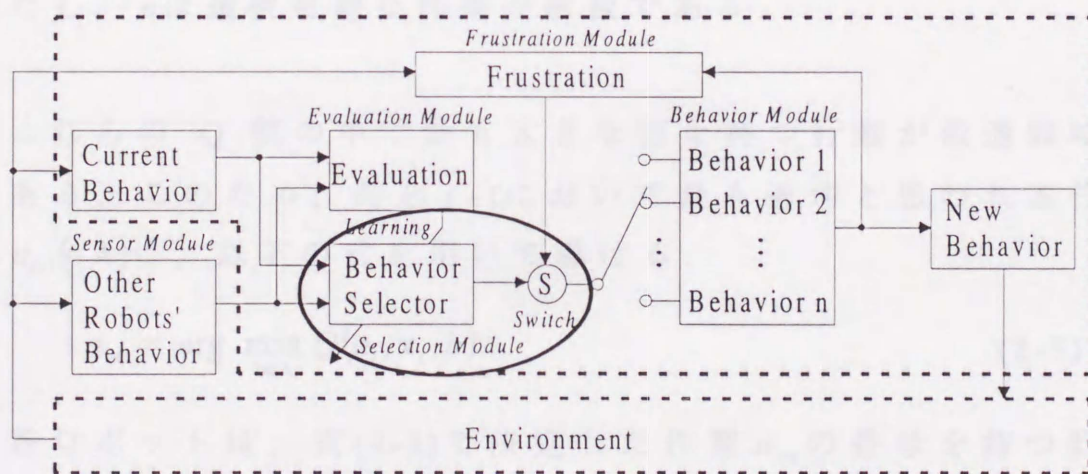


Fig.2-5 作業選択モジュールの配置

本研究では、自己と他のロボットの作業関係を学習し、その後の作業選択に反映させる能力を持たせるために、強化学習⁶⁶⁾⁻⁶⁸⁾の一つであるQ学習を適用した。適用理由としては、各状況において選択すべき作業は唯一ではないため、ニューラルネットワークのバックプロパゲーション法などが適用できないこと、また、作業分担の結果として得られるのが教師信号ではなく、人間のたまかな評価であることなどが挙げられる。

Q学習では、有限マルコフ的環境において十分な学習試行を繰り返した後Q値が収束しているならば、最大のQ値を持つ行動を選ぶことが最適戦略となる。この性質を利用し、各ロボットは最適と思われる作業を選択する。

\mathbf{A} を選択可能な作業の離散集合、 \mathbf{X} を有限な状態の離散集合とする。時刻 t において、自己が遂行している作業を $a_t(\in \mathbf{A})$ 、その状態を $\mathbf{x}_t(\in \mathbf{X})$ とすると、その時点で対象となるQ値は以下の n 個である。

$$Q(a_t, \mathbf{x}_t, 1), Q(a_t, \mathbf{x}_t, 2), \dots, Q(a_t, \mathbf{x}_t, n) \quad (2-2)$$

ただし、 n は選択可能な作業の総数である。

これらのQ値の中で最も大きな値を持つ行動が最適戦略である。このため、時刻 $t+1$ において最も適切と思われる作業 $a_{t+1}(\in \mathbf{A})$ は、以下の式を用いて導ける。

$$a_{t+1} = \arg \max_{b \in \mathbf{A}} Q(a_t, \mathbf{x}_t, b) \quad (2-3)$$

各ロボットは、式(2-3)で決定した作業 a_{t+1} の番号を持つ行動プログラムを行動群モジュールの中から選び出し、そのプログラムに従って作業を実行する。

2. 3. 5 評価・学習モジュール

センサモジュールが獲得した情報に基づいて、適切と思われる作業を作業選択モジュールで決定する。そして、そこで決定された作業番号に対応した行動プログラムを行動群モジュール内から選び、実行する。ここで、個々のロボットは、選択した作業が本当に適切であったのかを評価し、今後の作業選択に反映させる必要がある。このため、本研究では、評価・学習モジュールを用意した。このモジュールは、Fig.2-6の楕円で囲まれた場所に配置されている。

ロボットは、時刻 $t+1$ において、式(2-3)で決定した作業 a_{t+1} を実行する。他のロボットも同様に作業を切り替えるため状態は、 x_{t+1} となる。この状況において自己が遂行している作業 a_{t+1} が適切であったのかを評価する。しかし、何が適切なのかをロボット自身が理解することは困難であることから、本章では、ロボットを管理する操作者が、この評価値をあらかじめ与えておくこととした。

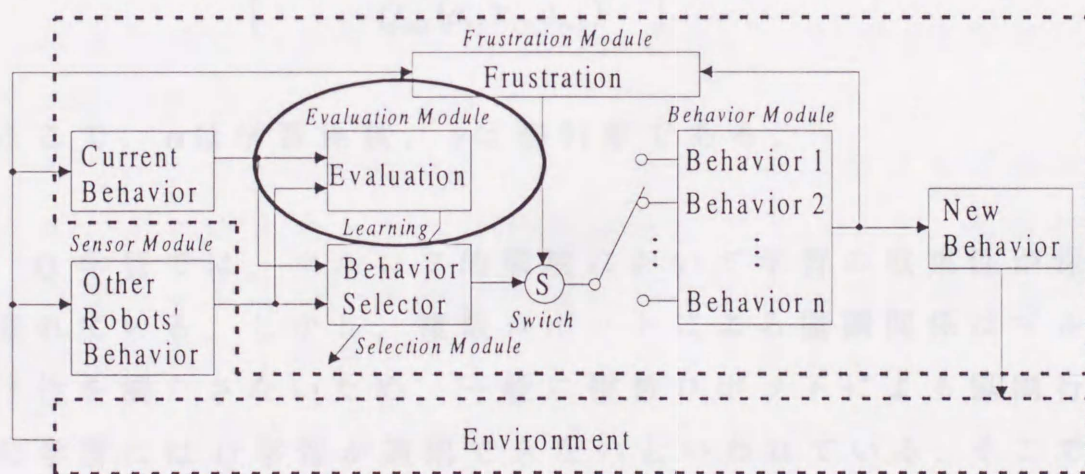


Fig.2-6 評価・学習モジュールの配置

本研究では、時刻 t において選んだ作業 a_{t+1} を時刻 $t+1$ に実行した結果、状態が \mathbf{x}_{t+1} となった場合、その時点における評価値は $E(a_{t+1}, \mathbf{x}_{t+1})$ であり、具体的には Table 2-1 で表現するものとした。このように評価を行列として記述する方法は、二つの関係を極度に圧縮し抽象化した表現と考えられているゲーム理論⁶⁹⁾⁻⁷²⁾の利得行列を参考にした。本研究での評価の値は、目標の作業分担であれば1、そうでなければ-1の値が代入してある。

ロボットは、上述の評価・学習モジュールを用いて、人間の意図した作業分担であるかを判断し、それに基づいて適切な作業選択を学習する。具体的には、一般的なQ学習の更新式に従って学習が行われる。

$$Q_{new}(a_t, \mathbf{x}_t, a_{t+1}) \leftarrow Q_{old}(a_t, \mathbf{x}_t, a_{t+1}) + \Delta Q \quad (2-4)$$

ただし、 ΔQ は次式で表現される。

$$\Delta Q = \alpha \times \left\{ \begin{array}{l} E(a_{t+1}, \mathbf{x}_{t+1}) \\ + \gamma \max_{b \in \mathbf{A}} Q_{old}(a_{t+1}, \mathbf{x}_{t+1}, b) \\ - Q_{old}(a_t, \mathbf{x}_t, a_{t+1}) \end{array} \right\} \quad (2-5)$$

ここで、 α は学習係数、 γ は割引率である。

Q学習では、マルコフ的環境において学習の収束性が保証されている。しかし、複数ロボットによる協調関係はマルコフ性を満たさないため、一般に複数ロボットによる協調行動の学習にはQ学習が適用できないといわれている。そこで、学習が収束しない理由を検討したところ、以下の3つの理由が挙げられた。

一つは、一般的な Q 学習で行われているように、行動決定の際に Q 値に応じて確率的に行動を選択すると、学習していた協調関係を崩してしまう点にある。これは、人間関係における一種の裏切り行為に相当するものであり、このような現象は、協調関係を成立させる上で悪影響を及ぼす。

二つ目は、2 台以上のロボットの学習係数 α が同じ値に設定されていると、全く同じ行動を選択するため、互いの関係を変更できず、局所解に陥る可能性がある。人間の場合、人それぞれに多少の個性があるため、このようなことは起こりにくいですが、ロボットではこのような点も考慮しなければならない。

三つ目は、割引率 γ を大きくすると、将来得るであろう報酬の見積もりが Q 値に大きく影響し、適切な協調関係でなくても Q 値が大きくなってしまいう点にある。ロボットは、最大の Q 値を持つ行動を遂行するから、結果として適切でない作業選択になってしまう点にある。

これらの問題点を考慮し、作業選択の際は常に最大の Q 値を持つ作業を選択すること、学習係数 α は乱数を用いてある程度の範囲内で値を変動させること、割引率 γ は非常に小さい値に設定することとした。

このような学習を行うことで、ロボットを管理する操作者があらかじめ目標となる作業分担を示す Table 2-1 の評価値を用意するものの、その評価値に応じて個々のロボットが適切な作業選択を学習することができると考える。

2.3.3 フラストラレーションモジュール

前節までに述べたアルゴリズムを用いることで、操作者が与えた評価値に応じて作業分担を生成できる。しかし、実際の作業に適用すると、予想もしなかった不都合な状況に陥る可能性もある。その場合には、個々のロボットが自ら実行している作業を変更する能力が必要である。このような状況を脱出するために、フラストラレーションモジュールを用いた。このモジュールは、評価値が与えられた部分に配置されている。

Table 2-1 目標の作業分担を示す評価値

| $E(i, j)$ | | x_{t+1} | |
|-----------|-------|------------|------------|
| | | T1, T1 ... | T2, T1 ... |
| a_{t+1} | Task1 | -1 | +1 |
| | Task2 | +1 | -1 |

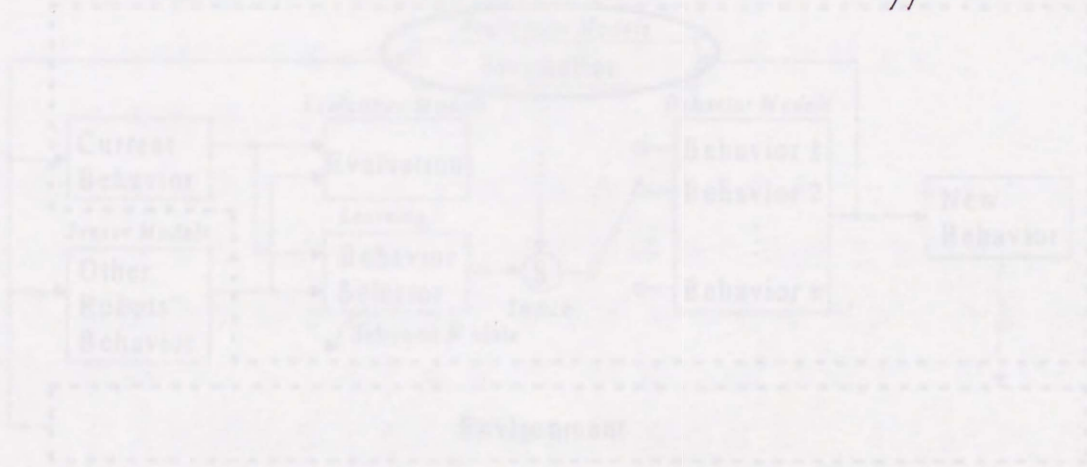


Fig 2-7 フラストラレーションモジュールの配置

2. 3. 6 フラストレーションモジュール

前節までに述べてきたアルゴリズムを用いることで、操作者が与えた評価値に応じて作業分担を生成できる。しかし、実際の作業に適用すると、予想もしなかった不都合な状況に陥る可能性もある。その場合には、個々のロボットが自ら遂行している作業を変更する能力が必要である。このような状況を脱出するために、フラストレーションモジュールを用意した。このモジュールは、Fig.2-7の楕円で囲まれた部分に配置されている。

学習心理学^{73),74)}によると、人間などの動物では、不都合な状況に遭遇すると、フラストレーション(欲求不満)の働きによって、別な行動に切り替える様子が観察されるという。このような動物の性質を模倣し、フラストレーションモジュールでは、ある一定時間以上の間、作業が遂行できないと判断された場合には、他の作業に切り替える働きをする。

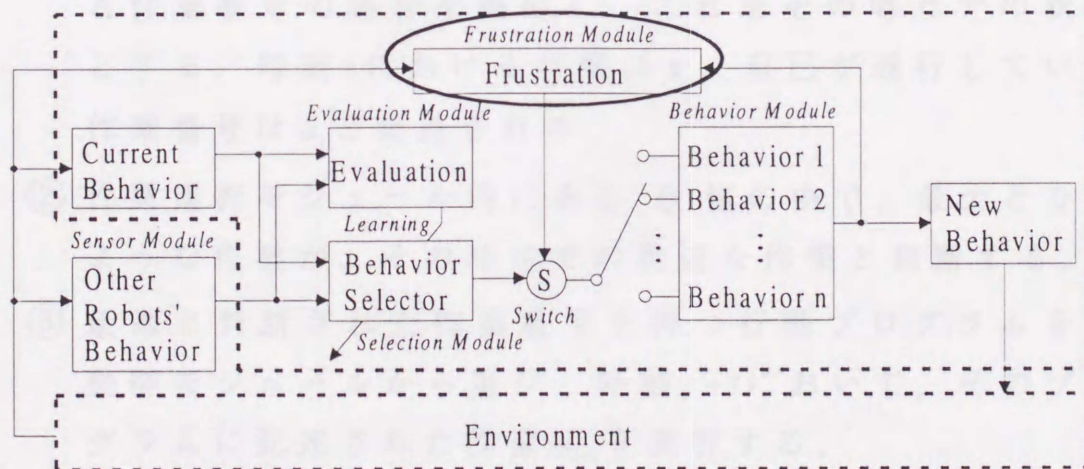


Fig.2-7 フラストレーションモジュールの配置

2.4 アルゴリズムの処理の流れ

本節では、これまでに述べてきたモジュールから構成されるアルゴリズムをロボットに搭載した際、個々のロボットがどのように情報を処理し作業分担するのかをまとめる。

作業環境には m 台のロボットを用意し、それぞれのロボットに Fig.2-2 のアルゴリズムを搭載した。このアルゴリズムには、 n 種類の作業を遂行するための行動プログラム、及び、目標となる作業分担を示す評価値を与えておくことを前提としている。また、作業選択の基準となる Q 学習の初期値は乱数を与えることとした。したがって、初期状態において、各ロボットは適切な作業分担ができない状況にある。

個々のロボットは、以下の ①から⑤までの処理を繰り返し行い、適切に作業選択できるように学習していく。その様子を Fig.2-8 に示す。

- ① センサモジュールを用いて他のロボットが遂行している作業番号の集合を獲得し、これをその時点での状態とする。時刻 t における状態は x_t 、自己が遂行している作業番号は a_t と記述される。
- ② 作業選択モジュール内にある Q 値の中で、最大となるような作業が、その時点での最適な作業と判断する。
- ③ 最適と判断された作業番号を持つ行動プログラムを行動群モジュールから選び、時刻 $t+1$ において、そのプログラムに記述された作業 a_{t+1} を実行する。
- ④ 時刻 $t+1$ では、他のロボットも作業を切り替えるため状態は x_{t+1} に移行する。ここで、適切な作業決定であったのかを、評価 E を用いて判断する。

- ⑤ この評価 E を用いて, Q 学習における ΔQ を求め, Q 値を更新する. これは, 時刻 t の状態 x_t において作業 a_{t+1} を選択したことに対する学習になる.

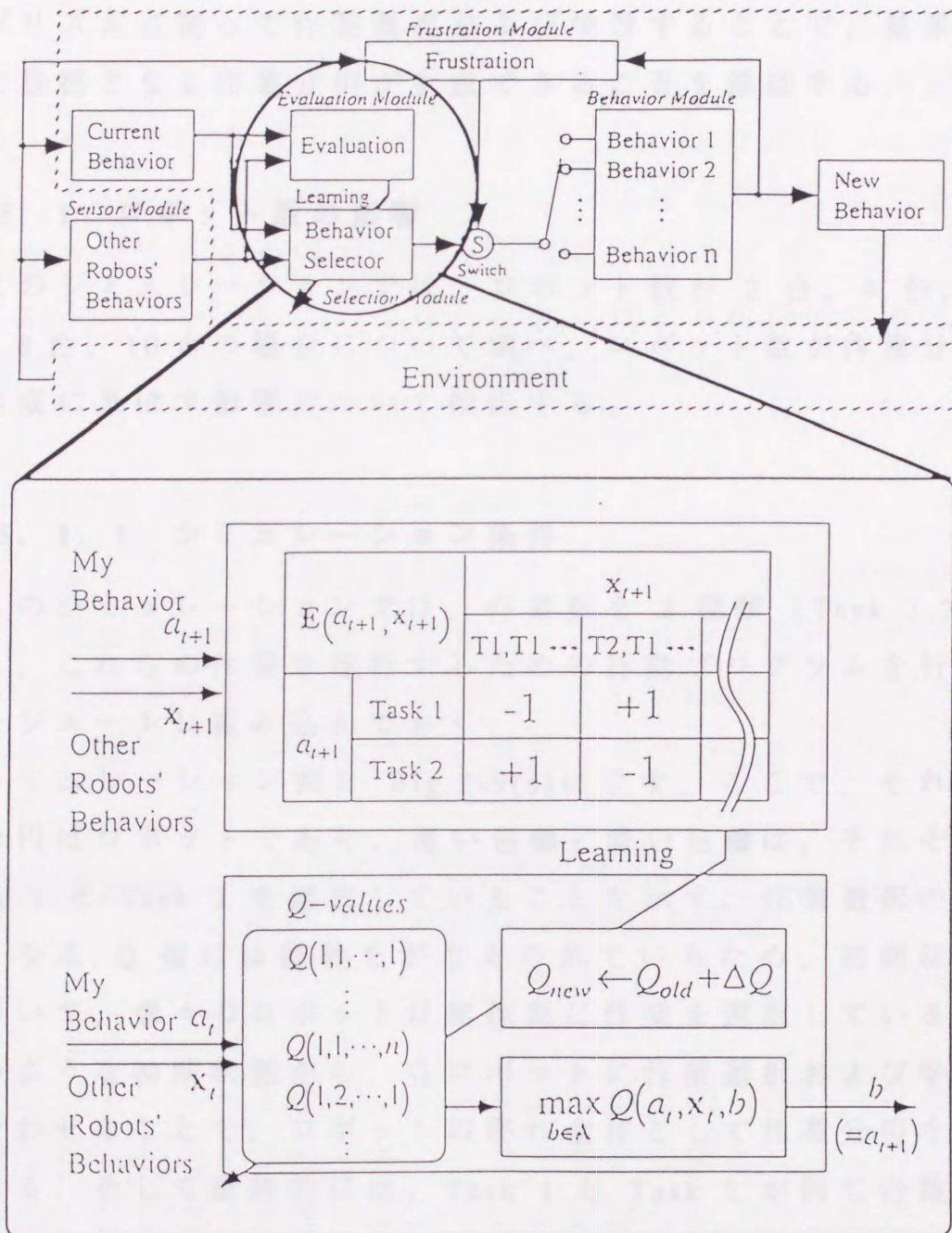


Fig.2-8 アルゴリズムの処理の流れ

2.5 アルゴリズムの検証

前節までに述べてきたアルゴリズムの有効性を検証するためにシミュレーションを行った。ここでは、各ロボットがアルゴリズムに従って作業選択および学習することで、結果として目標となる作業分担が生成できることを確認する。

2.5.1 ロボット数の影響

このシミュレーションでは、ロボット数が2台、4台、6台、8台、10台の場合について調べ、ロボット数が作業分担の生成に及ぼす影響について検証する。

2.5.1.1 シミュレーション条件

このシミュレーションでは、作業数を2種類 (Task 1, 2) とし、これらの作業を遂行するための行動プログラムを行動群モジュールに組み込んでおく。

シミュレーション例を Fig.2-9(a)に示す。ここで、それぞれの円はロボットであり、薄い色●と濃い色●は、それぞれ Task 1 と Task 2 を選択していることを示す。作業選択の基準となる Q 値には乱数が与えられているため、初期状態において、個々のロボットは無作為に作業を選択している。このような初期状態から、各ロボットに作業選択および学習を行わせることで、ロボットの群れ全体として作業分担を行わせる。そして最終的には、Task 1 と Task 2 が同じ台数のロボットによって作業分担に収束することが目標である。したがって、例えば、ロボット数が10台の場合、Task 1 と Task 2 を5台ずつで作業分担することが目標となる。この目標の

作業分担を示す評価値は，ロボットを管理する人間によってあらかじめ与えられていることを前提としている。

ロボットによる学習手順を Fig.2-9(b)に示す。個々のロボットは，まず，センサモジュール内の通信機能を用いて他のロボットが遂行している作業番号を獲得し，その時点の状態を知る。そして，自己が遂行している作業とその状態の関係から，前回の作業選択について評価し Q 学習によって学習する。さらに，学習結果に基づいて，適切と思われる作業選択を行う。これを個々のロボットは繰り返し行う。

このシミュレーションでは，学習に対する初期値依存性の問題を考慮し，100000通りの初期値から検証を行った。

2.5.1.2 シミュレーション結果

シミュレーション結果を Fig.2-10 に示す。横軸は学習試行数，縦軸は協調成功率を示す。この協調成功率は，次式によって導かれる値であり，100000通りの初期値から何パーセントの組が目標の作業分担に収束できたかを意味する。

$$\text{協調成功率} = \frac{\text{目標の作業分担に達成した数}}{\text{初期状態の総数}(=100000\text{種類})} \times 100 [\%]$$

(2-6)

シミュレーションでは，ロボット数の増加に伴い，目標の作業分担に到達するのに多くの試行数が必要となるものの，最終的には目標に到達できることを確認した。

- ☉ : Task 1 を選択しているロボット
- : Task 2 を選択しているロボット

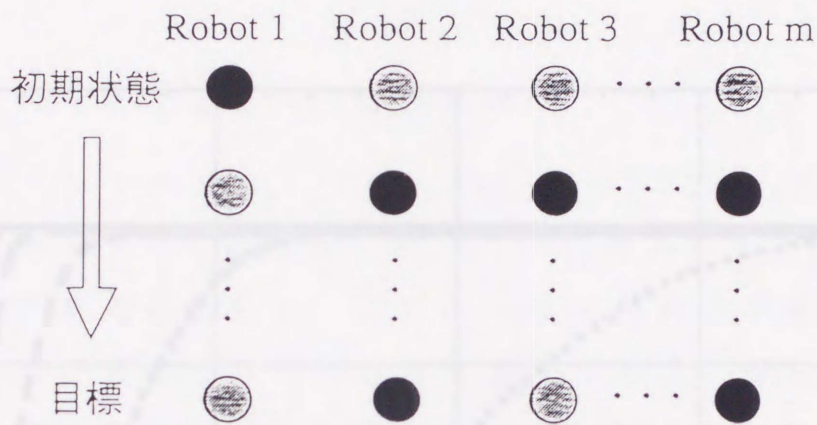


Fig.2-9(a) シミュレーション例

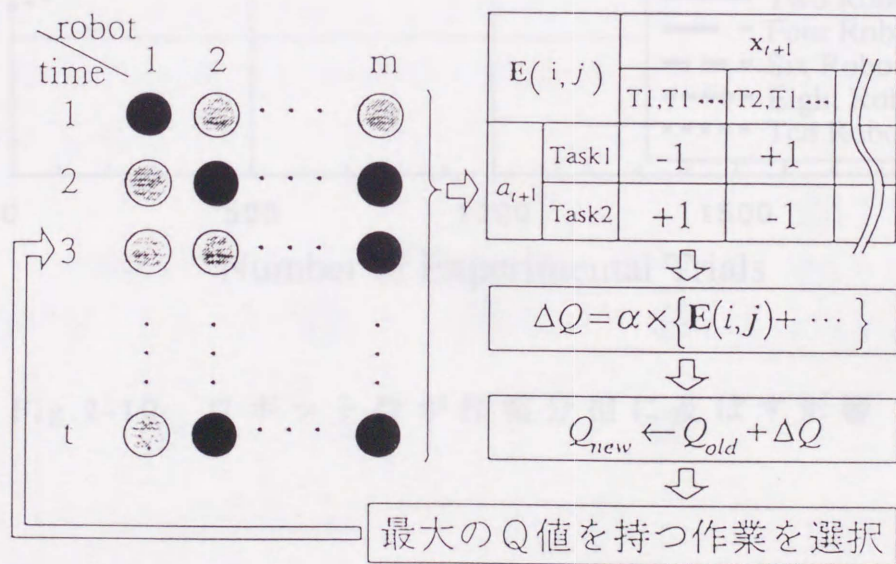


Fig.2-9(b) 作業分担を生成するまでの手順

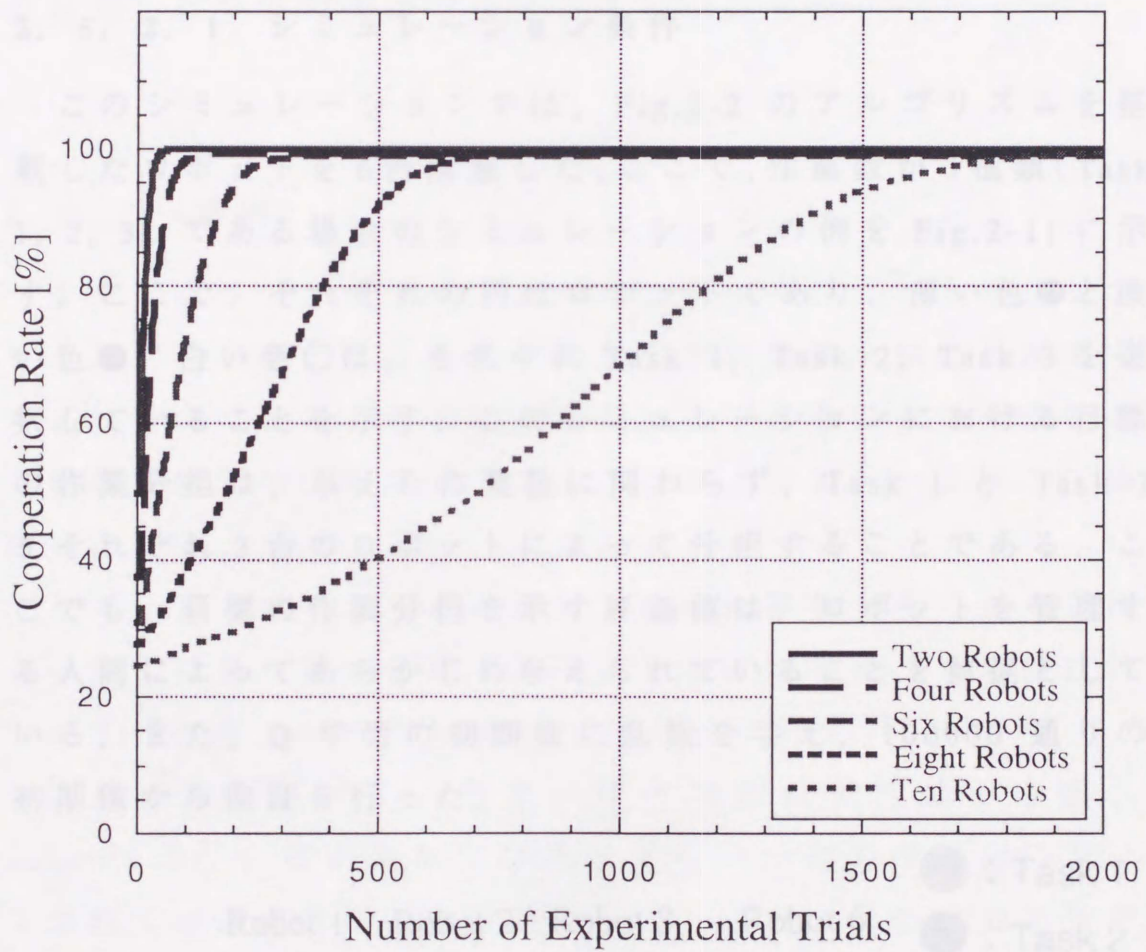


Fig.2-10 ロボット数が作業分担に及ぼす影響

2.5.2 作業数の影響

このシミュレーションでは、作業数が2種類から5種類までのそれぞれの場合について調べ、作業数が作業分担の生成に及ぼす影響について検証する。

2.5.2.1 シミュレーション条件

このシミュレーションでは、Fig.2-2のアルゴリズムを搭載したロボットを6台用意した。ここで、作業数が3種類(Task 1, 2, 3)である場合のシミュレーションの例をFig.2-11に示す。ここで、それぞれの円はロボットであり、薄い色●と濃い色●、白い色○は、それぞれTask 1, Task 2, Task 3を選択していることを示す。このシミュレーションにおける目標の作業分担は、与えた作業数に関わらず、Task 1とTask 2をそれぞれ3台のロボットによって分担することである。ここでも、目標の作業分担を示す評価値は、ロボットを管理する人間によってあらかじめ与えられていることを前提としている。また、Q学習の初期値に乱数を与え、100000通りの初期値から検証を行った。

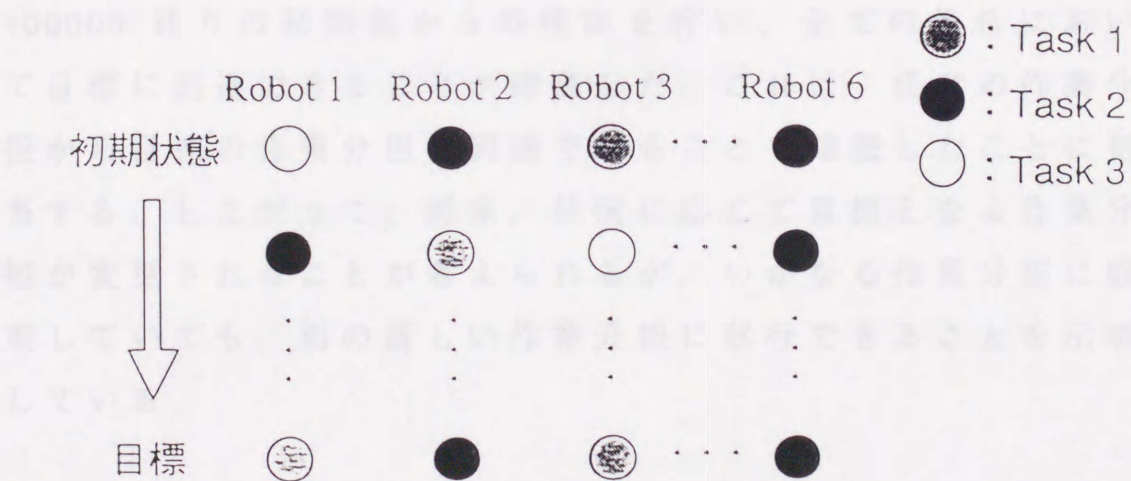


Fig.2-11 作業数が3種類の場合のシミュレーション例

2. 5. 2. 2 シミュレーション結果

シミュレーション結果を Fig.2-12 に示す。横軸は学習試行数、縦軸は協調成功率である。選択可能な作業が増加すると、目標の協調関係に収束するのに非常に多くの試行数が必要であることがわかった。しかし、これらの場合においても、最終的には目標に到達できることを確認した。

2. 5. 3 考察

本章で述べてきたアルゴリズムを個々のロボットに搭載することで、目標となる作業分担を示す評価値を操作者があらかじめ与えておくものの、各ロボットの自律的な作業決定から目標の作業分担を生成できることをシミュレーションで確認した。

シミュレーションではロボット数や作業数が多くなると非常に多くの試行数が必要であることも分かった。この問題については、後の章にて検討を行うこととする。

また、本研究では、学習に伴う初期値依存性を考慮し100000通りの初期値からの検証を行い、全ての場合において目標に到達できることが確認した。これは、任意の作業分担から目標の作業分担に到達できることを確認したことに相当する。したがって、将来、状況に応じて目標となる作業分担が変更されることが考えられるが、いかなる作業分担に収束していても、別の新しい作業分担に移行できることを示唆している。

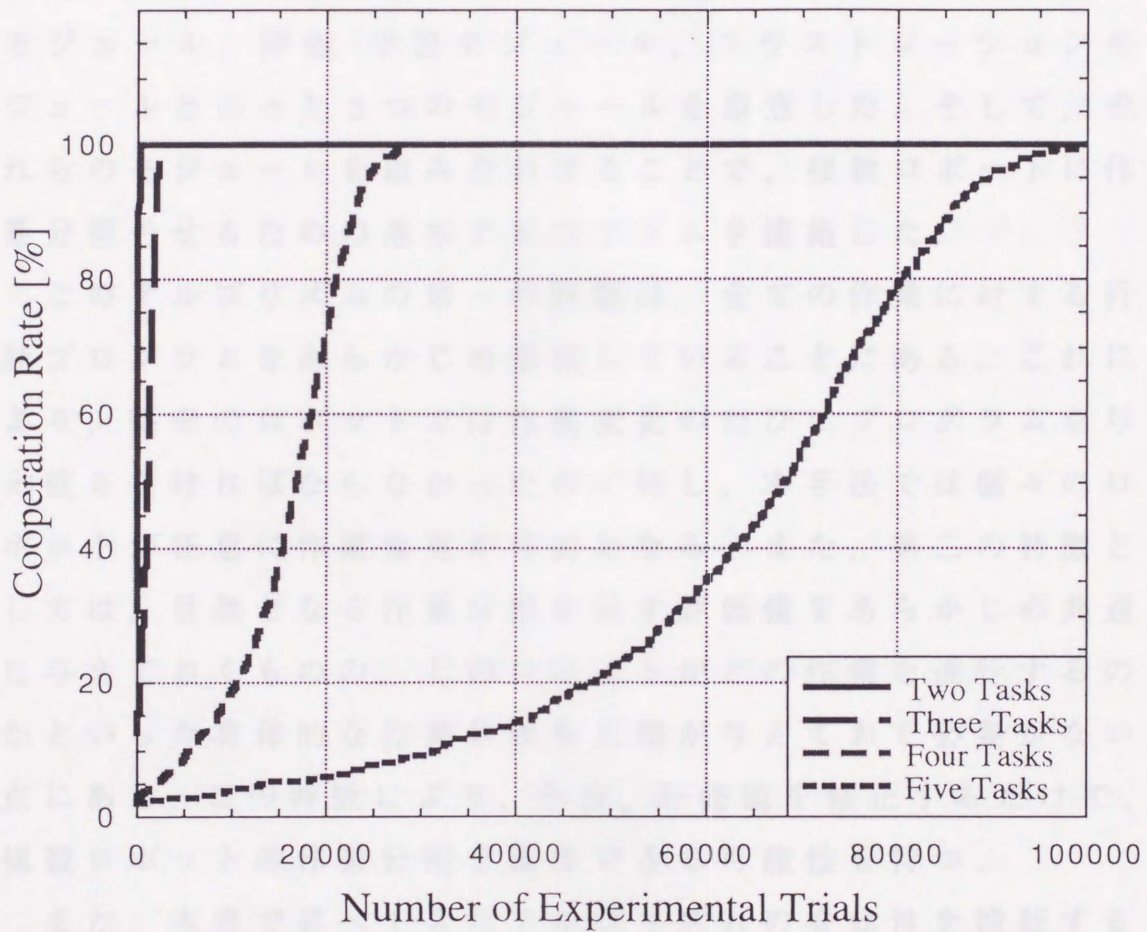


Fig.2-12 作業数が作業分担に及ぼす影響

2.6 まとめ

本章では、複数ロボットに作業分担させるための基本アルゴリズムを構築するため、まず、複数のロボットに作業分担に必要な機能を列挙し、それらのモジュール化を行った。具体的には、センサモジュール、行動選択モジュール、行動群モジュール、評価・学習モジュール、フラストレーションモジュールといった5つのモジュールを用意した。そして、それらのモジュールを組み合わせることで、複数ロボットに作業分担させるための基本アルゴリズムを構築した。

このアルゴリズムの第一の特徴は、全ての作業に対する行動プログラムをあらかじめ搭載していることにある。これにより、従来のロボットでは作業変更のたびにプログラムを与え直さなければならなかったのに対し、本手法では個々のロボットが任意に作業変更が可能となる。また、第二の特徴としては、目標となる作業分担を示す評価値をあらかじめ共通に与えておくものの、どのロボットがどの作業を遂行するかといった具体的な作業分担を人間が与えておく必要がない点にある。この特徴により、今後、評価値を修正するだけで、複数ロボットの作業分担を操作できる可能性を持つ。

また、本章で述べてきたアルゴリズムの有効性を検証するためにシミュレーションを行った。シミュレーションでは、ロボット数や作業数が多くなるにつれ、目標の作業分担に到達するのに多くの試行数が必要であることがわかった。しかし、いずれの場合においても、最終的には目標の作業分担を生成できることを確認した。これらのシミュレーションから、複数ロボットに作業分担を行わせるための基本アルゴリズムとして有効であると考えられる。

第3章 指令に基づく群れの制御手法の提案

3.1 はじめに

前章で構築したアルゴリズムをロボットに適用することにより、個々のロボットの相互関係から作業分担を生成できることをシミュレーションで確認した。しかし、そこでは、ロボットを管理する操作者が目標となる作業分担を示す評価値をあらかじめ与えることを前提としていた。このため、異なる作業分担を行わせるためには、目標となる作業分担に基づき適切な評価値を決定し、その評価値を全てのロボットに与え直さなければならなかった。そこで、本章では、状況に応じて人間の意図した作業分担に容易に操作するための手法について検討する。

状況に応じて意図した作業分担を複数のロボットに行わせるためには、何らかの形で操作者の意図をロボットに伝達しなければならない。特に、ロボットが複数存在する場合には、個々のロボットに対し個別に指示を与えることができないなどの制約があるため、これらを考慮しながら操作者の意図をロボットに伝達するための手法を検討する必要がある。

本章では、まず、一人の人間が複数のロボットを制御する場合の問題を明確にし、次にその解決法について検討する。そして、その検討結果に基づき、具体的な複数ロボットの制御手法を提案する。

3.2 一対多を扱う制御手法

一人の人間が一台のロボットを制御する場合は、人間はロボットの様子を常に観察することができ、また、状況に応じて細やかな指示を与えることができる。この一対一の制御手法を用いて複数のロボットを操作しようとするすると、ロボットを観察するのに時間がかかるなどの問題が生じる。このため、一人の人間が複数のロボットを制御する一対多を扱う制御手法の研究が行われてきた。これらを手法の性質から分類すると、大きく2つに分けることができる。一つは、ロボット全体としての目標を個々のロボットにあらかじめ与えておく手法である。もう一つは、人間が複数ロボットを観察し、状況に応じて何らかの指令を与えることで、ロボットの群れを操作する手法である。

前者の場合、操作者が事前に目標を与えておかなければならないため、目標を変更するためには、全てのロボットに新しい目標をプログラムし直さなければならないという問題が生じる。一方、後者の場合には、状況に応じて指令が与えられるため、柔軟にロボットの群れを制御できる。しかし、一部の処理は個々のロボットが自律的に行っているものの、作業決定から具体的な動作まで、ロボットを管理する操作者が直接操作しており、一度に操作できる作業数には限りがある。このように、一人の人間が複数のロボットを制御する、従来の一対多の制御手法では、状況に応じた作業の役割分担を生成させる事は困難であることから、本研究では複数のロボットに複数の作業を与えた場合の作業分担の生成手法について検討する。

操作するロボットの台数が一台から多数台になることに伴

操作するロボットの台数が一台から多数台になることに伴う問題は、二つの挙げられる。一つは、ロボットが多数存在することで、操作者は個々のロボットを個別に観察できないという点にある。このような状況では、人間はロボット全体に対する大まかな判断しかできないと考えられる。ここでの大まか判断とは、例えば、群れ全体に対する良い・悪いといった単純な評価に相当する。したがって、ロボット数が多い場合には、このような単純な評価からロボットを制御する手法を検討しなければならない。二つ目の問題は、ロボットが多数存在することで、操作者は個々のロボットに個別の指示を与えられないという点にある。つまり、人間がロボットを操作する場合には、複数のロボットの群れ全体に対する指令しか与えざるを得ないことを意味する。このため、個々のロボットは、群れ全体に対する共通の指令から作業分担を生成する手法を検討しなければならない。

つまり、ロボットの台数が多い場合を考慮するならば、複数のロボットの群れを観察し、群れに対して指令を与えることを前提としなければならず、このような状況から作業分担を生成する手法が必要であると考えられる。

3. 3 群れの制御手法の提案

3. 3. 1 制御手法の概略

ロボットの台数が多い場合には、複数のロボットの群れを観察し、群れに対して指令を与えることを前提としなければならない。このような状況から作業分担を生成する手法が必要であることを前節で述べた。そこで、本節では、そのような前提条件からの作業分担する手法について検討する。

第2章で提案してきたアルゴリズムでは、目標となる作業分担を示す評価値 E をあらかじめ与えておくことで、人間の意図した作業分担が生成できた。したがって、各ロボットが、その評価値を、状況に応じて任意にかつ的確に修正できれば、間接的にロボットの群れを操作できる可能性を持つ。このため、本研究では、Fig.3-1のように、複数のロボットに対し、何らかの指令を与えるものの、その指令に基づいて個々のロボットが自己の評価値を修正することで、ロボットの群れを制御する手法を検討した。その手法の概略を以下に示す。

まず、第2章で提案したアルゴリズムを搭載したロボットを複数台用意し、これまで通りの手法で作業を分担させる。ロボット全体を管理する操作者は、このロボットの群れを観察し、全体として意図した作業分担ができているかを判断する。そして、その判断結果を何らかの指令に置き換え、全てのロボットに共通に与える。個々のロボットは、得られた指令に基づいて、操作者がどのような作業分担を望んでいるかを理解し、自己の評価を修正する。また、その評価に基づいて学習することで、結果として操作者の意図した作業分担を実現する手法である。

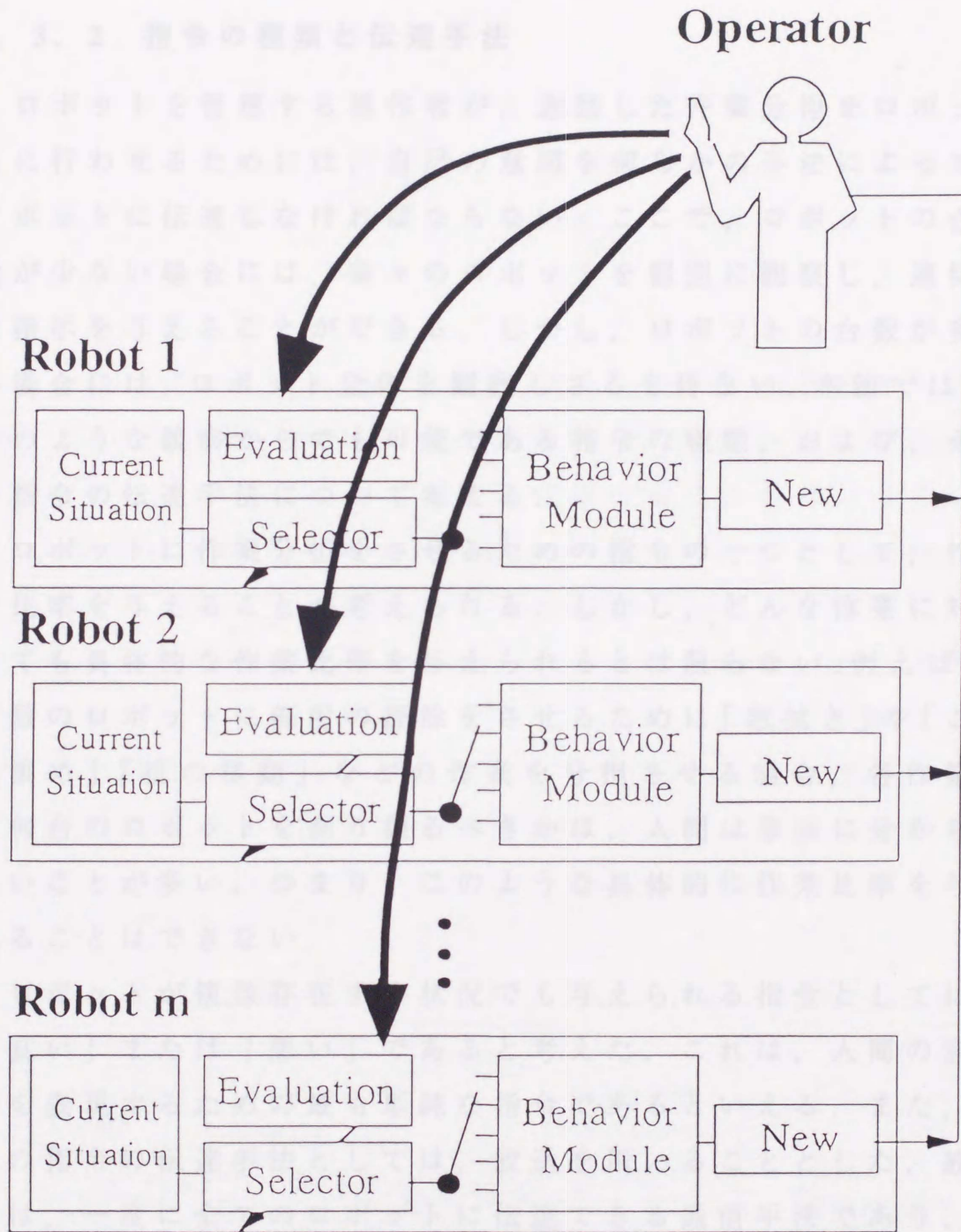


Fig.3-1 評価を修正することによる
複数ロボットの制御手法

3. 3. 2 指令の種類と伝達手法

ロボットを管理する操作者が、意図した作業分担をロボットに行わせるためには、自己の意図を何らかの手法によってロボットに伝達しなければならない。ここで、ロボットの台数が少ない場合には、個々のロボットを個別に観察し、適切な指示を与えることができる。しかし、ロボットの台数が多い場合には、ロボット全体を観察しざるを得ない。本節では、このような観察からでも可能である指令の種類、および、その指令の伝達手法について考える。

ロボットに作業分担をさせるための指令の一つとして、作業比率を与えることが考えられる。しかし、どんな作業に対しても具体的な作業比率を与えられるとは限らない。例えば、複数のロボットに部屋の掃除をさせるために「窓拭き」や「ごみ集め」「机の移動」などの作業を分担させる場合、各作業に何台のロボットを割り振るべきかは、人間は事前に分からないことが多い。つまり、このような具体的に作業比率を与えることはできない。

ロボットが複数存在する状況でも与えられる指令としては、「良い」または「悪い」であると考えた。これは、人間の意図を表現するための最も単純な指令であるといえる。また、この指令の伝達手法としては、放送を用いることとした。放送は、一度に全てのロボットに伝達できる通信手法であり、ロボットの台数に依存しないという特徴がある。

したがって、ロボットを管理する操作者は、複数ロボットの群れを観察し、全体として「良い」または「悪い」を判断し、それを操作者の意図として、ロボットの群れに放送することとした。

3.3.3 行動評価の修正手法

操作者からの指令に基づき行動評価を修正するためには、個々のロボットは、操作者からの指令を受信するための機能を追加しなければならない。このため、これまで適用してきた Fig.2-2 のアルゴリズムを Fig.3-2 に変更する。ここで、楕円で示された部分が新たに追加された機能である。

個々のロボットは、追加された機能を用いて操作者からの「良い」または「悪い」という指令を獲得し、自己の行動評価を修正する。この場合、修正した評価値は強化学習の一つである Q 学習に使用されるという理由から、修正された評価値の全てが 0 より大きい、或いは 0 未満であることを避けなければならない。その理由として、それらの評価は、それぞれ報酬のみ、ペナルティのみを意味し、これらの条件を満たすと適切な学習が行えないからである。もちろん、評価値の全てが 0 であることも避ける必要がある。このような強化学習への適用に伴う制約条件を利用し、評価値を修正する。

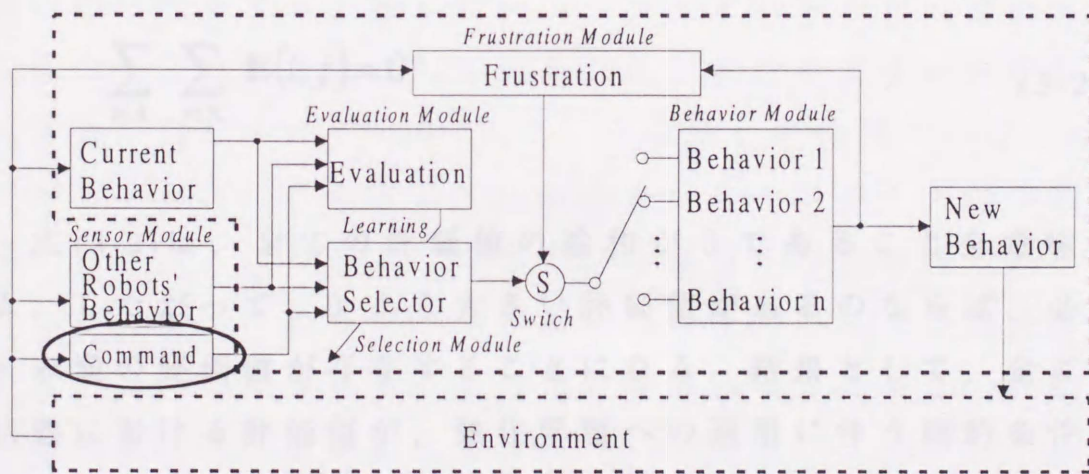


Fig.3-2 操作者からの指令を受信する装置の追加

時刻 $t+1$ において、自己が遂行している作業を a_{t+1} 、その状態を \mathbf{x}_{t+1} とすると、その時点での評価値 $E(a_{t+1}, \mathbf{x}_{t+1})$ は、以下の式を用いて修正する。

$$\begin{aligned} \text{If the signal was a penalty ...} & \quad \mathbf{R}(a_{t+1}, \mathbf{x}_{t+1}) = -1 \\ \text{If the signal was a reward ...} & \quad \mathbf{R}(a_{t+1}, \mathbf{x}_{t+1}) = +1 \end{aligned}$$

$$E(a_{t+1}, \mathbf{x}_{t+1}) = \mathbf{R}(a_{t+1}, \mathbf{x}_{t+1}) - \frac{\sum_{i \in \mathbf{A}} \sum_{j \in \mathbf{X}} \mathbf{R}(i, j)}{\sum_{i \in \mathbf{A}} \sum_{j \in \mathbf{X}} \|\mathbf{R}(i, j)\|} \quad (3-1)$$

ただし、 \mathbf{A} は選択可能な作業の離散集合、 \mathbf{X} は有限な状態の離散集合、 $\mathbf{R}(i, j)$ は操作者からの指令であり初期値は 0 が代入されている。

個々のロボットは、式(3-1)を用いて、操作者からの指令に基づき、自己の行動評価を修正する。式(3-1)では、過去に「良い」と「悪い」を1回以上得たことがあれば、修正された行動評価の値は以下の式を満たす。

$$\sum_{i \in \mathbf{A}} \sum_{j \in \mathbf{X}} \mathbf{E}(i, j) = 0 \quad (3-2)$$

式(3-2)は、全ての評価値の総和が0であることを意味する。したがって、0より大きい評価値があるのならば、必ず0未満の評価値が存在することになる。結果として、全ての状態における評価値が、強化学習への適用に伴う制約条件を満足することを示す。この手法により、操作者からの単純な指令から複数ロボットの作業分担を制御できると考える。

3. 3. 4 行動評価修正法の改良

前節での行動評価の修正手法を用いることで、操作者は複数ロボットの作業分担を制御できる可能性を持つ。しかし、その手法では、個々のロボットの機能や性能などについては考慮されていない。作業分担を行う場合には、本来、個々のロボットの性能なども考慮した上で決定されるべきであり、この点においては、適切な手法とは言えない。つまり、Fig.3-3(a)のような人間からのトップダウン的な指令のみに基づいた行動評価の修正法ではなく、Fig3-3(b)のように個々のロボットの機能の違いから生じるボトムアップ的な評価も考慮するべきであると考えた。このため、本節では、行動評価の修正手法に改良を加えることを検討した。

人間を含む多くの動物社会では、具体的な作業分担が指示されなくても、しばしば適切な行動を選択する。これは、各メンバがそれぞれ持っている基準となる評価（以降、自己評価基準と呼ぶ）に基づいて行動を決定している⁷⁵⁾からである。人間における自己評価基準とは、疲労や作業を遂行する満足感に相当する。個々のメンバは、それらの自己評価基準に基づいて行動を決定しているため、疲労の少ない効率の良い行動が選択されている。このような自己評価基準の考えは、ロボットにおいても有効であることから、ロボットへの適用を検討する。ただし、ロボットには疲労や作業の満足感などはないため、代わりに作業の遂行に伴うエネルギー消費量を用いる。このようなロボットの自己評価基準を利用し、行動評価の修正法を改良する。

時刻 $t+1$ において，自己が遂行している作業を a_{t+1} ，その状態を \mathbf{x}_{t+1} とすると，その時点での評価値 $\mathbf{E}(a_{t+1}, \mathbf{x}_{t+1})$ は，以下の式を用いて修正する．

If the signal was a penalty ...

$$\begin{aligned} \mathbf{R}(a_{t+1}, \mathbf{x}_{t+1}) &= 0 \\ \mathbf{E}(a_{t+1}, \mathbf{x}_{t+1}) &= \mathbf{C}(a_{t+1}) \end{aligned}$$

If the signal was a reward ...

$$\begin{aligned} \mathbf{R}(a_{t+1}, \mathbf{x}_{t+1}) &= 1 \\ \mathbf{E}(a_{t+1}, \mathbf{x}_{t+1}) &= \mathbf{C}(a_{t+1}) - \frac{\sum_{i \in \mathbf{A}} \mathbf{C}(i)}{\sum_{i \in \mathbf{A}} \sum_{j \in \mathbf{X}} \mathbf{R}(i, j)} \end{aligned} \quad (3-3)$$

ただし， $\mathbf{C}(i)$ は，自己が作業 i を実行した際の自己評価基準を示し，常に 0 未満の値を持つ． $\mathbf{R}(i, j)$ は操作者からの指令であり初期値は 0 が代入されている． \mathbf{A} は選択可能な作業の離散集合， \mathbf{X} は有限な状態の離散集合である．

式 (3-3) においても式 (3-2) が成立する．つまり，修正された評価値が，強化学習への適用に伴う制約条件を満足するといえる．

この手法により，操作者がロボットの群れに対する「良い」または「悪い」といったトップダウン的な指令と，個々のロボットの機能の違いなどからくるボトムアップ的な自己評価基準に基づいて行動評価の修正が行われる．このため，操作者が個々のロボットの機能や性能などを考慮しなくても，ロボットの群れ全体として適切な作業分担を生成できる可能性を持つ．

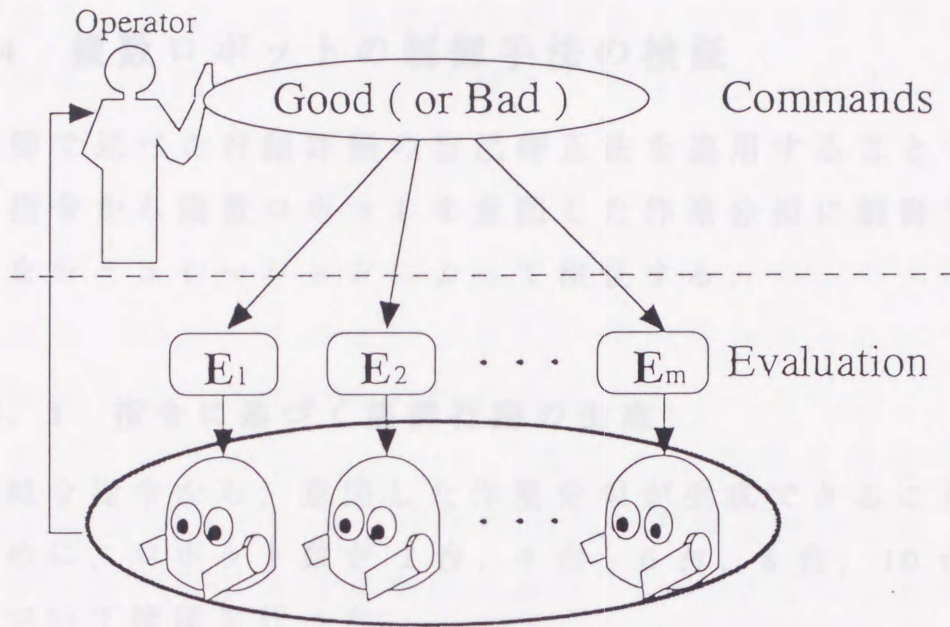


Fig.3-3(a) 人間の指令のみに基づく評価の修正

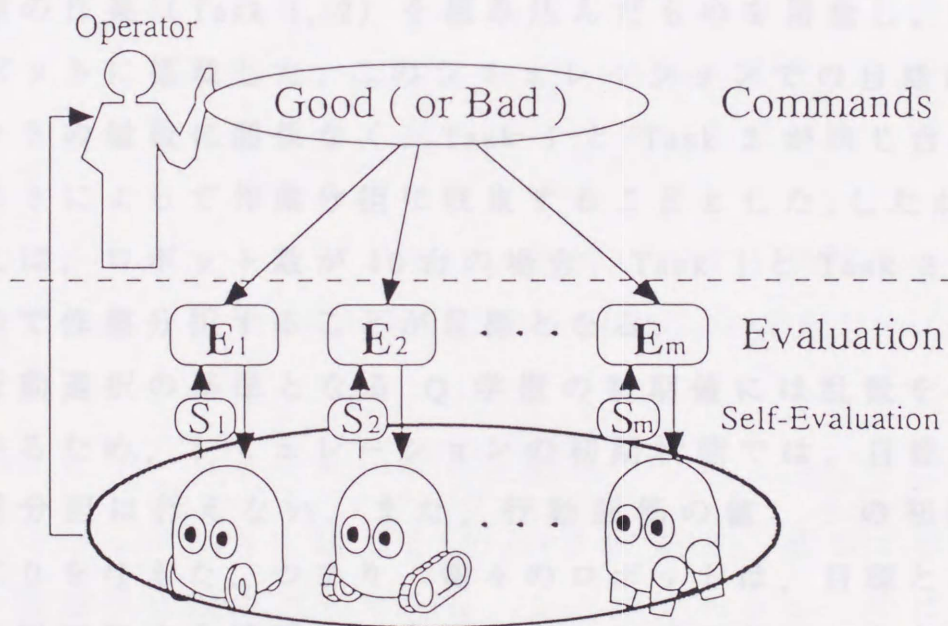


Fig.3-3(b) 機能の違いも考慮した評価の修正

3. 4 複数ロボットの制御手法の検証

前節で述べた行動評価の自己修正法を適用することで、単純な指令から複数ロボットを意図した作業分担に制御できることをシミュレーションによって検証する。

3. 4. 1 指令に基づく協調行動の生成

単純な指令から、意図した作業分担が生成できることを示すために、ロボット数が2台、4台、6台、8台、10台の場合について検証を行った。

3. 4. 1. 1 シミュレーション条件

このシミュレーションでは、Fig.3-2 のアルゴリズムに2種類の作業 (Task 1, 2) を組み込んだものを用意し、個々のロボットに搭載した。このシミュレーションでの目標は、ロボットの総数に関係なく、Task 1 と Task 2 が同じ台数のロボットによって作業分担に収束することとした。したがって、例えば、ロボット数が10台の場合、Task 1 と Task 2 を5台ずつで作業分担することが目標となる。

行動選択の基準となる Q 学習の初期値には乱数を代入しているため、シミュレーションの初期状態では、目標となる作業分担は行えない。また、行動評価の値の初期値は全て0を与えた。つまり、個々のロボットは、目標となる作業分担に関する情報は、全く与えられていないこととなる。

このようなロボットによって作業分担を行わせる。本研究では、ロボットを管理する操作者がロボットの群れを観察し、目標の作業分担になっていれば「良い」という指令 (以降、

報酬信号と呼ぶ)を, 目標の作業分担でなければ「悪い」という指令(以降, ペナルティ信号と呼ぶ)を, ロボットの群れに対して放送することとしている. しかし, 実際の人間が指令を与える場合, 指令の与え間違いや与え忘れが生じる. このため, このシミュレーションでは, とりあえず人間の代わりにコンピュータが確実に正しい指令を与えるという理想的な状況下で, 手法の有効性を検証することとした.

ロボットは, コンピュータが放送した指令を受信し, 式(3-3)に基づいて自己の行動評価の値を修正する. ここで, 各ロボットの自己評価基準は, 簡単のため全て-1とした.

ロボットによる学習手順を Fig.3-4 に示す. 個々のロボットは, まず, センサモジュール内の通信機能を用いて他のロボットが遂行している作業番号を獲得し, その時点の状態を知る. ここで, 操作者の代わりとなったコンピュータが作業分担の良し悪しを判断し, 自動的に指令を与える. 個々のロボットは, この指令に基づき自己の評価を修正し, 学習を行う. さらに, 学習結果に基づいて, 適切と思われる作業選択を行う. これを個々のロボットが繰り返し行う.

このシミュレーションでは, 学習に対する初期値依存性の問題を考慮し, 100000通りの初期値から検証を行った.

3. 4. 1. 2 シミュレーション結果

シミュレーション結果を Fig.3-5 に示す. 横軸は学習試行数, 縦軸は協調成功率である. シミュレーションでは, ロボット数の増加に伴い, 目標の作業分担に到達するのに多くの試行数が必要となるものの, 最終的には目標に到達できることを確認した.

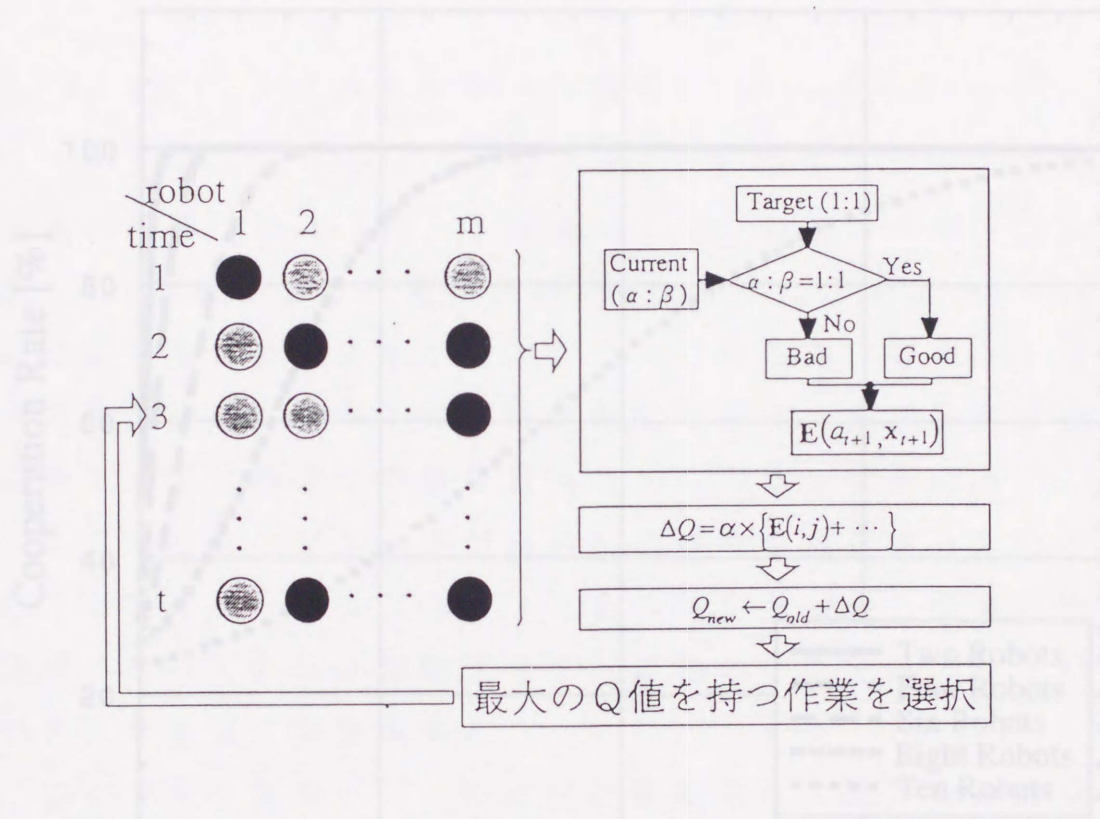


Fig.3-4 コンピュータの指令に基づく評価の生成

3.4.3 指令に基づく群れの制御の実験

実験では、コンピュータからの正確な指令から、目標の作業分担が達成できることを示した。ここでは、シミュレーション中であっても、指令を実行することで作業分担が達成できることを確認する。

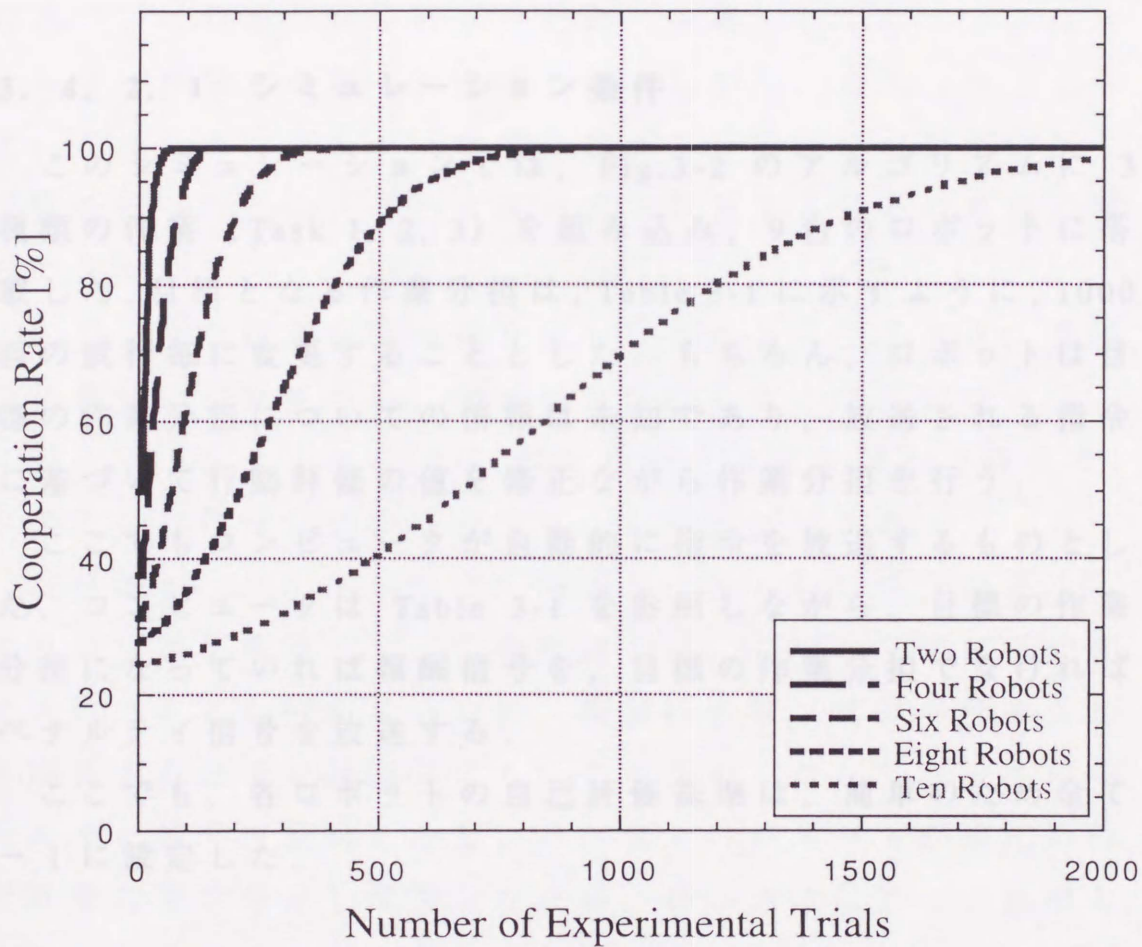


Fig.3-5 ロボット数が作業分担に及ぼす影響

| Number of Experimental Trials | Two Robots | Four Robots | Six Robots | Eight Robots | Ten Robots |
|-------------------------------|------------|-------------|------------|--------------|------------|
| 1-1000 | 3台 | 3台 | 3台 | 3台 | 3台 |
| 1001-2000 | 1台 | 1台 | 1台 | 7台 | 7台 |
| 2001-3000 | 5台 | 4台 | 4台 | 0台 | 0台 |
| 3001-4000 | 2台 | 3台 | 3台 | 4台 | 4台 |

3.4.2 指令に基づく群れの制御の検証

前節では、コンピュータからの単純な指令から、目標の作業分担を生成できることを示した。ここでは、シミュレーション中であっても、指令を変更することで作業分担を制御できることを確認する。

3.4.2.1 シミュレーション条件

このシミュレーションでは、Fig.3-2 のアルゴリズムに3種類の作業 (Task 1, 2, 3) を組み込み、9台のロボットに搭載した。目標となる作業分担は、Table 3-1 に示すように、1000回の試行毎に変更することとした。もちろん、ロボットは目標の作業分担についての情報は未知であり、放送される指令に基づいて行動評価の値を修正ながら作業分担を行う。

ここでもコンピュータが自動的に指令を放送するものとした。コンピュータは Table 3-1 を参照しながら、目標の作業分担になっていれば報酬信号を、目標の作業分担でなければペナルティ信号を放送する。

ここでも、各ロボットの自己評価基準は、簡単のため全て-1に設定した。

Table 3-1 目標となる作業分担

| 試行数 | Task 1 | Task 2 | Task 3 |
|-----------|--------|--------|--------|
| 1-1000 | 3台 | 3台 | 3台 |
| 1001-2000 | 1台 | 1台 | 7台 |
| 2001-3000 | 5台 | 4台 | 0台 |
| 3001-4000 | 2台 | 3台 | 4台 |

3. 4. 2. 2 シミュレーション結果

シミュレーション結果を Fig.3-6 に示す。横軸は学習試行数、縦軸は各作業を選択したロボットの台数を示す。

はじめのうちは、個々のロボットは、どんな作業分担をすれば良いのかがわからず、乱数で初期化された Q 学習の値に基づいて作業分担を行ったため、結果としてロボットはペナルティ信号しか得られなかった。しかし、個々のロボットは、それらのペナルティ信号に基づき行動評価の値を修正し学習を行ったため、遂行する作業を変更する様子が頻繁に観察できた。

このような作業変更が繰り返し行われることにより、目標の作業分担が発見され、報酬信号が得られるようになった。このシミュレーションでは、試行数が 260 回を越えた時点で、目標となる作業分担に収束することが確認できた。

試行数が 1000 回になった時点で、新しい作業分担となるように目標を変更した。この時点では、ロボットは目標が変更されたことを知らないため、数試行の間、前回の作業分担を維持する様子が確認できた。このため、コンピュータはペナルティ信号を放送したところ、個々のロボットが自己の行動評価の値を修正し学習したため、最終的には新しい目標となる作業分担に到達できることが確認できた。

3. 4. 3 考察

このシミュレーションでは、コンピュータが確実に指令を与えていた。これを実際の人間に適用すると、与え間違いなどが生じることが考えられるが、指令は複雑でないため、実際の人間でも十分適用できると考える。

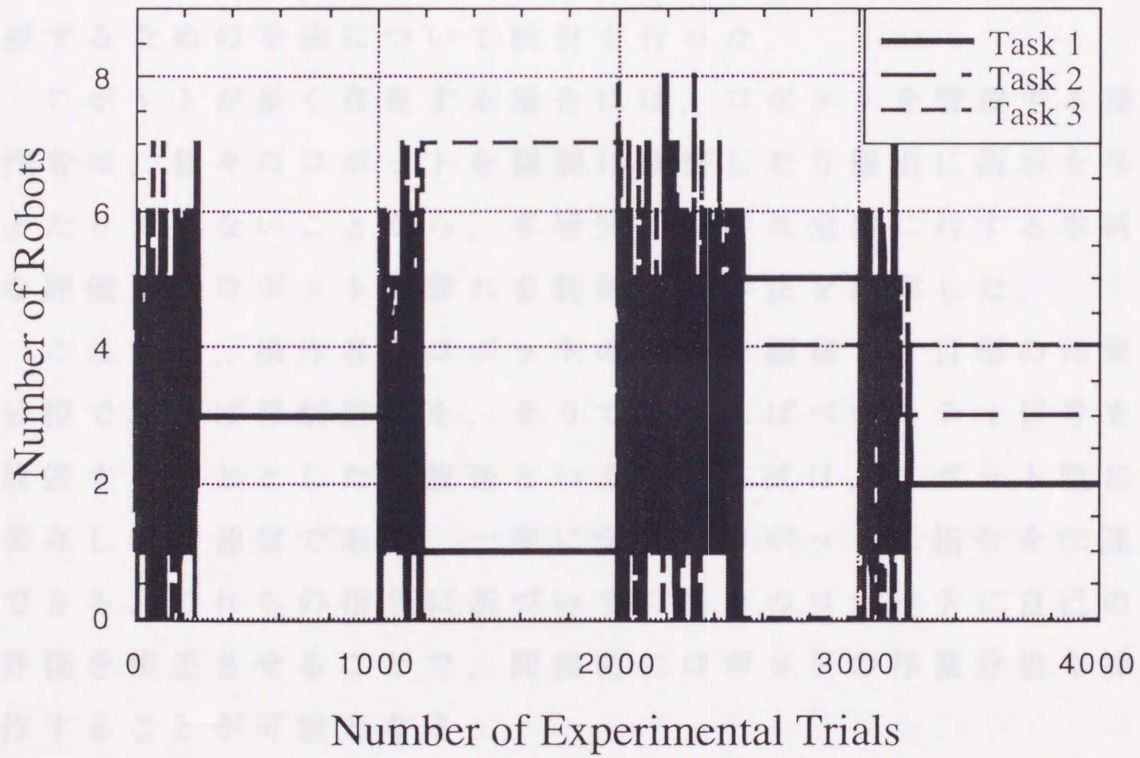


Fig.3-6 単純な指令に基づく作業分担

3.5 まとめ

第2章では、目標の作業分担を示す行動評価を、操作者があらかじめ与えておくことを前提としてアルゴリズムを構築し、その有効性をシミュレーションで検証してきた。そこで、本章では、ロボットを管理する操作者の意図に応じて作業分担するための手法について検討を行った。

ロボットが多く存在する場合には、ロボットを管理する操作者は、個々のロボットを個別に観察したり個別に指示を与えたりできないことから、本研究では群れ全体に対する単純な評価からロボットの群れを制御する手法を提案した。

ここでは、操作者がロボットの群れを観察し、目標の作業分担であれば報酬信号を、そうでなければペナルティ信号を放送することとした。放送という通信方式は、ロボット数に依存しない通信であり、一度に全てのロボットに指令を伝達できる。これらの指令に基づいて、個々のロボットに自己の評価を修正させることで、間接的にロボットの作業分担を操作することが可能となる。

実際に人間が指令を与える場合、指令の与え間違いなどが生じるため、このシミュレーションでは、人間の代わりにコンピュータが確実にロボットの群れを観察し指令を与える事とした。このような理想的な条件下ではあるものの、外部からの単純な指令から作業分担を制御できることを確認した。また、作業中に目標となる作業分担が変更しても、ロボットのプログラムを変更したり、新しく評価を与え直すことなく、外部からの単純な指令だけでロボットの群れを操作できることを確認した。

第4章 群れの効果的制御手法の提案

4.1 はじめに

第2章のアルゴリズムと第3章での群れの制御手法を組み合わせることにより、単純な指令からロボットの作業分担が制御できることをシミュレーションによって確認した。しかし、ロボット数や作業数が多くなると、目標の作業分担に到達するのに非常に多くの学習試行数が必要であることが分かった。

適用する作業によって一概には言えないが、本手法を実際の作業に適用するならば、人間が数千回の指令を与えることになり、事実上困難であると考えられる。特に、Fig.3-5の結果から推測すると、ロボット数が50台や100台などと増加した場合には、数十万回の学習試行数が必要だと予想される。しかし、将来的には、100台以上のロボットを操作することも十分に考えられ、これまでの手法では適用が困難になる。

そこで、本章では、まず、ロボット数や作業数が多くなると、目標の作業分担に到達するのに多くの学習試行数が必要となる理由について検討し、そこでの問題を明確にする。次に、それらの問題を解決するための手法を提案する。そして、その手法を適用することで、ロボット数が増加しても学習試行数が増加しなくなることをシミュレーションによって検証する。

4. 2 群れの誘導手法

4. 2. 1 誘導手法の提案

本節では，ロボット数や作業数が増加すると，目標の作業分担に到達するのに非常に多くの学習試行が必要であるという問題について検討し，その問題を明らかにする．また，目標への到達を促進するための手法について検討する．

これまでのシミュレーションを振りかえると，まず，報酬信号を得るまでに非常に多くの試行数が必要であることがわかった．そこで，ロボット数がどの程度の影響を与えるのかについて，3種類の作業を1:2:3で分担することを例に検討を行った．まず，6台のロボットにこの作業を遂行させる場合，個々のロボットが無作為に作業を選択しても，およそ1/12の確率で報酬信号を獲得できる．一方，50台のロボットにこの作業を遂行させると，およそ1/5251の確率となる．これは，ロボット数が増加した場合，目標の作業分担への指針となる報酬信号が得られにくくなることを意味し，それ故，適切な役割分担へ到達するのに時間がかかっていたといえる．そこで，本研究では，目標となる作業分担になっていなくても報酬信号を与え，徐々に目標へ近づけることを試みた．具体的な指示の与え方を以下に述べる．

まず，作業分担を始めた最初($t=0$)は，いかなる作業分担であったとしても報酬信号を与えることとする．また，時刻 t における作業分担が時刻 $t-1$ の作業分担より目標に近ければ報酬信号を，遠ければペナルティ信号を与えることとした．このような指令を与えることで，少しずつではあるが目標の作業分担に誘導できると期待される．ここで，時刻 t の作業

分担が時刻 $t-1$ の作業分担と同じである場合には、操作者は指令を与えないものとした。この場合には、ロボットは、時刻 $t-1$ で得た指令（報酬信号またはペナルティ信号）を使用することとした。

しかし、この指令の与え方では、目標の作業分担には到達していないものの前回の作業分担より良かったとして報酬信号を得て、そのままの作業分担に収束し、いつまでも目標の作業に到達できなくなるという局所解に陥る可能性がある。このように目標の作業分担でないのに作業分担が収束している場合には、ペナルティ信号を与えることで局所解を脱出させる必要がある。そこで、そのような局所的な状況に陥るまでの学習試行数を導出する。

Fig.3-2 のアルゴリズムでは、一般的な Q 学習が適用されており、各状態に対する全ての作業にそれぞれ Q 値が割り当てられている。今、あるロボットが Q_1 の値を持つ作業を実行し、報酬信号が得られたとする。そして、この作業を選択することで n 回および $n-1$ 回連続して報酬信号が得られたとき、それぞれの Q 値は以下の値を持つ。

$$\begin{aligned} Q_n &= \sum_{k=1}^{n-1} \alpha \times E \times (1 + \alpha \times \gamma - \alpha)^{k-1} + Q_1 \times (1 + \alpha \times \gamma - \alpha)^{n-1} \\ Q_{n-1} &= \sum_{k=1}^{n-2} \alpha \times E \times (1 + \alpha \times \gamma - \alpha)^{k-1} + Q_1 \times (1 + \alpha \times \gamma - \alpha)^{n-2} \end{aligned} \quad (4-1)$$

ここで、 α と γ はそれぞれ Q 学習の学習係数と割引率である。また、 E は報酬の値である。

したがって、Q 学習で Q 値が更新される値 ΔQ_{n-1} は以下の式で表現できる。

$$\begin{aligned}\Delta Q_{n-1} &= Q_n - Q_{n-1} \\ &= \alpha \times (1 + \alpha \times \gamma - \alpha)^{n-2} \times \{E + Q_1 \times (\gamma - 1)\}\end{aligned}\quad (4-2)$$

Fig.4-1 は、学習が収束していく様子を示したものである。この図からもわかるように ΔQ_{n-1} は Q 値の更新される値であるから、もし ΔQ_{n-1} が非常に小さいならば、ロボットの学習が収束しつつあることに相当する。ここで、 ΔQ_{n-1} が収束判定用の値 T より小さい場合、学習が収束したと仮定するならば、学習のために必要な試行数 n は、以下の式を用いて導ける。

$$\begin{aligned}\text{if } \Delta Q_{n-1} \leq T \\ n \geq 2 + \frac{\log \left\{ \frac{T}{\alpha \times \{E + Q_1 \times (\gamma - 1)\}} \right\}}{\log(1 + \alpha \times \gamma - \alpha)}\end{aligned}\quad (4-3)$$

ただし、 T は学習の収束判定用の値である。

式(4-3)を用いることで、ロボットが学習に必要とする試行数を求めることができる。ここで、 Q_1 は一般に 0 以上であることから、式(4-3)を適用する際には Q_1 が 0.0 であると仮定している。また、本研究では Q 学習の学習係数に乱数を与え、ある程度の範囲で変化させている。そこで、ここでの学習係数 α は、変化する値の中で最小の値を用いる。

したがって、式(4-3)を満たす試行回数以上、同じ作業分担が継続された場合、個々のロボットの学習が収束し局所解に陥ったものと判断し、操作者はペナルティ信号を与える。これにより、再び作業分担が行われるようになり、目標の作業分担にまで導くことができる。

4.2.2 群れの調整手法の検証

目標の作業分派でなくても、前回の作業分派より改善されれば報酬信号を与え、最終的には目標の作業分派まで誘導する手法を、シミュレーションを用いて検証する。

4.2.2.1 シミュレーション条件

Fig.3-2のアルゴリズムを持つロボットを10台用意し、それらに1種類の作業 (Task 1, 2, 3) を与えた。ここでの目標は、

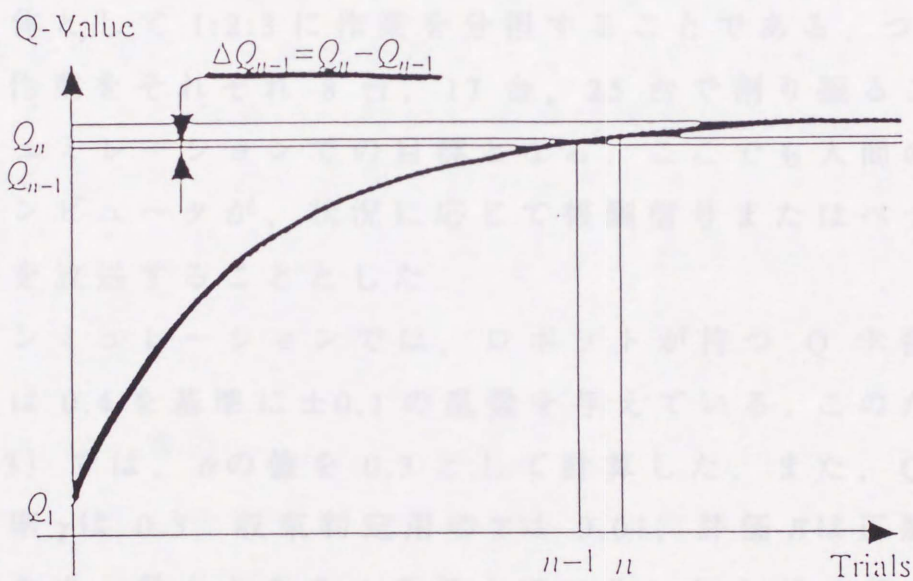


Fig.4-1 Q学習が収束していく様子

4.2.2.2 これまでの手法

ここでは、目標の作業分派が正確に行われた場合のみ報酬信号を与え、手法を用いてシミュレーションを行った。その結果を Fig.4-2 に示す。横軸は学習試行数、縦軸は各作業を実行したロボットの台数である。このシミュレーションでは、

4. 2. 2 群れの誘導手法の検証

目標の作業分担でなくても，前回の作業分担より改善されれば報酬信号を与え，最終的には目標の作業分担まで誘導する手法を，シミュレーションを用いて検証する．

4. 2. 2. 1 シミュレーション条件

Fig.3-2 のアルゴリズムを持つロボットを 50 台用意し，それらに 3 種類の作業 (Task 1, 2, 3) を与えた．ここでの目標は，全体として 1:2:3 に作業を分担することである．つまり，3 つの作業をそれぞれ 8 台，17 台，25 台で割り振ることがこのシミュレーションでの目標となる．ここでも人間の変わりにコンピュータが，状況に応じて報酬信号またはペナルティ信号を放送することとした．

このシミュレーションでは，ロボットが持つ Q 学習の学習係数は 0.4 を基準に ± 0.1 の乱数を与えている．このため，式 (4-3) では， α の値を 0.3 として計算した．また，Q 学習の割引率 γ は 0.3，収束判定用の T は 0.01，評価 E は正規化しているため，最大となる 1 を与えている．したがって，これらの値を式 (4-3) に代入することで，学習が収束するまでに 17 回以上の試行が必要であることが分かる．

4. 2. 2. 2 これまでの手法

ここでは，目標の作業分担が正確に行われた場合のみ報酬信号を与える手法を用いてシミュレーションを行った．その結果を Fig.4-2 に示す．横軸は学習試行数，縦軸は各作業を実行したロボットの台数である．このシミュレーションでは，

800回の試行中に偶然1回の報酬を得ることができたが、結果として目標となる役割分担は生成できなかった。

4. 2. 2. 3 群れの誘導手法

ここでは、本節で述べた群れの誘導手法を導入した。その結果を Fig.4-3 に示す。このシミュレーションでは、学習試行数が50回を超えたあたりから徐々に報酬信号が得られ始め、学習試行数が100回の時点で連続して報酬信号が得られるようになった。連続して17回報酬信号が得られた場合、その作業分担は十分学習が収束したとして、ペナルティ信号を与えた。その結果、一時的に協調関係が崩れたが、すぐに目標の作業分担に近い作業分担が生成できることを確認した。最終的に、558回の試行で目標となる作業分担まで群れを誘導することができた。

4. 2. 2. 4 考察

ロボット数が増加すると目標の作業分担に到達するのに時間がかかるという問題に対処するために、前回の作業分担より改善されれば報酬信号を、悪くなればペナルティ信号を与える。また、前回の作業分担と同じである状態が式(4-3)を満たす回数以上続いた場合には、局所解に陥ったとしてペナルティ信号を与えた。これにより、偶然的に報酬が得られるのを待つ従来の手法に比べ、より早く目標の作業分担に導けることを確認した。

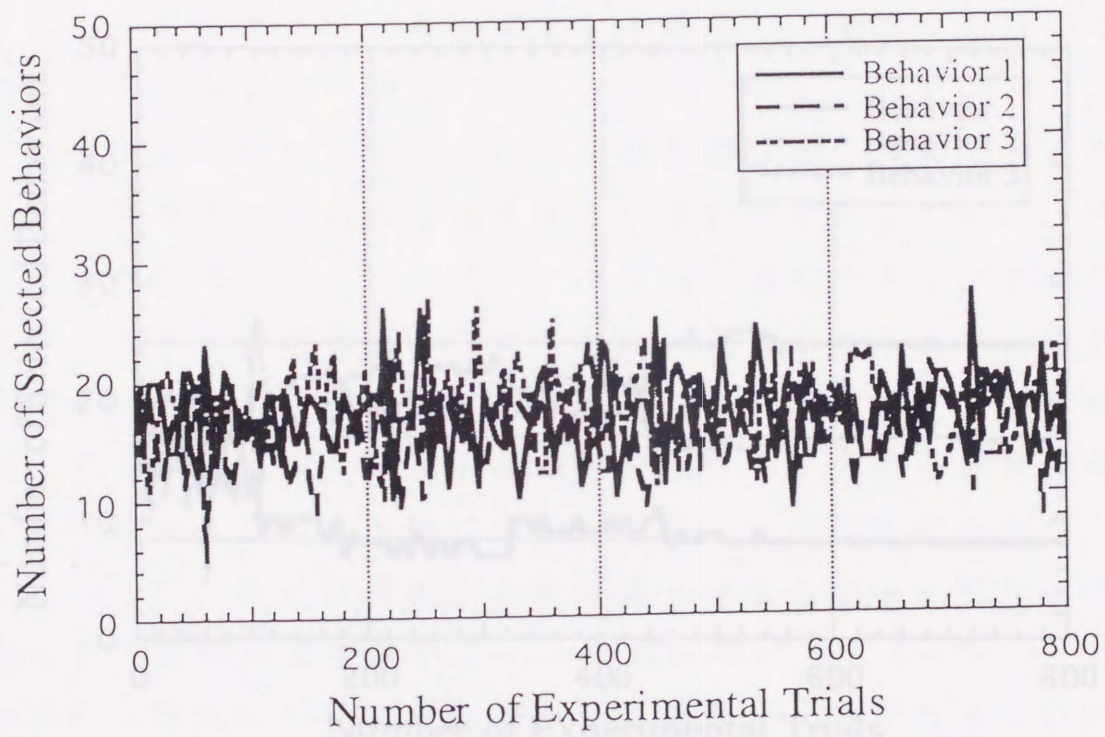


Fig.4-2 これまでの手法（群れの誘導なし）

4.3 小集団毎の通信

4.3.1 通信量と学習領域の調査

ロボット数や作業数が増加すると、各員の作業分析に必要十分な通信に多くの学習試行が必要であるという問題について、通信量や学習領域の調査を行った。

本研究でのロボットは、他のロボットと相互通信をすることによって学習している作業動作を共有し、それらの動作の

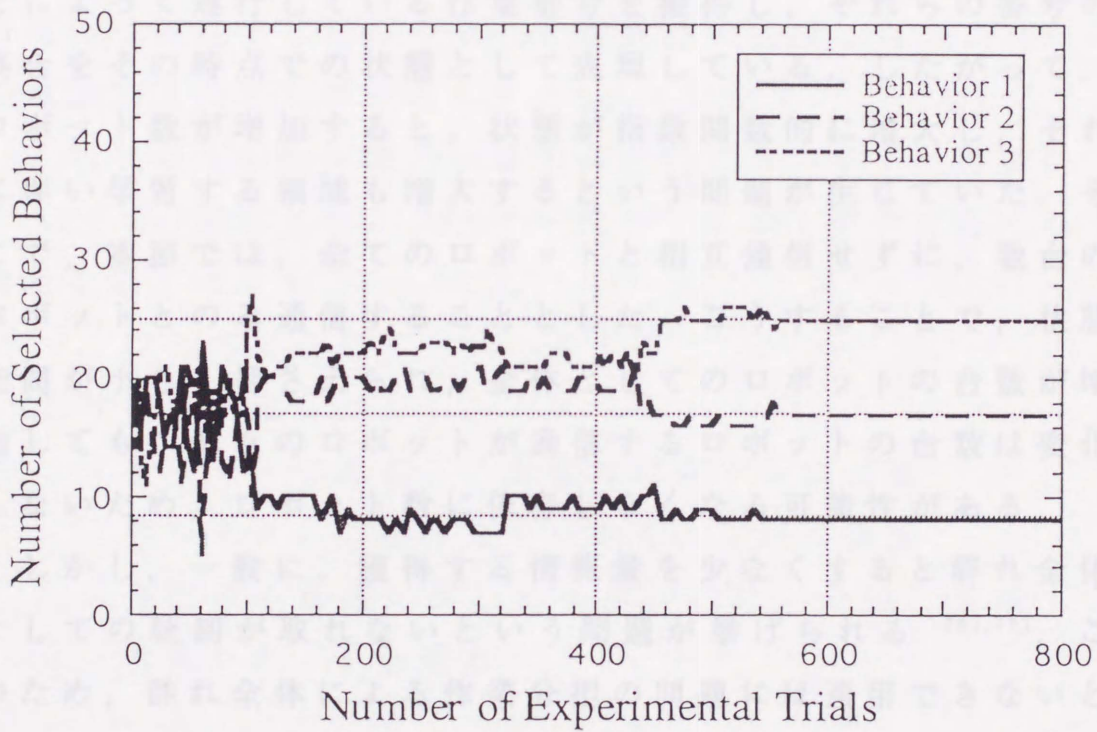


Fig.4-3 群れの誘導手法の導入

4.3.2 小集団の形成

この手法を適用するにあたり、個々のロボットはどのロボットと通信するのかが決定する必要がある。このため、本研

4. 3 小集団毎の通信

4. 3. 1 通信量と学習領域の関係

ロボット数や作業数が増加すると、目標の作業分担に到達するのに非常に多くの学習試行が必要であるという問題について、別の視点からの検討を行った。

本研究でのロボットは、他のロボットと相互通信をすることによって遂行している作業番号を獲得し、それらの番号の集合をその時点での状態として表現している。したがって、ロボット数が増加すると、状態が指数関数的に増大し、それに伴い学習する領域も増大するという問題が生じていた。そこで、本節では、全てのロボットと相互通信せずに、数台のロボットとのみ通信することとした。こうすることで、状態空間が小さく押さえられ、全体としてのロボットの台数が増加しても、個々のロボットが通信するロボットの台数は変化しないため、ロボット数に依存しなくなる可能性がある。

しかし、一般に、獲得する情報量を少なくすると群れ全体としての統制が取れないという問題が挙げられる^{76),77)}。このため、群れ全体による作業分担の問題には適用できないとされているが、本研究での手法では、群れ全体の様子が操作者による単純な指令によって個々のロボットにフィードバックしており、群れ全体としての統制は維持できると考える。そこで、数台のロボットとのみ通信する手法を検討する。

4. 3. 2 小集団の形成

この手法を適用するにあたり、個々のロボットはどのロボットと通信するのかを決定する必要がある。このため、本研

究では、まず、小集団を形成し、その集団に属するロボットの間による通信に基づいて作業分担を生成することとした。

このような小集団の形成を行うために、これまで本研究で適用してきた Fig.3-2 のアルゴリズムに機能の追加を行う。まず、行動群モジュールにリーダー役とアシスタント役を追加した。また、小集団を形成するモードから作業分担を行うモードに切り替えるための操作者からの信号を獲得するモジュールを追加した。さらに、異なる小集団に属するロボットの情報は排除するためのデータ選択器を追加した。これらの機能追加後のアルゴリズムを Fig.4-4 に示す。

これらの具体的な働きを以下に説明する。まず、行動群モジュールに追加されたリーダー役とアシスタント役のそれぞれの役目を以下に示す。

○ リーダー役

- ・リーダーであることを他のロボットに伝達
- ・アシスタント役のロボットをメンバに追加
- ・自己の小集団に属した各メンバに、今後、通信すべきロボットを通知

○ アシスタント役

- ・任意のリーダーに、メンバになれるかを依頼

ロボットを管理する操作者は、まず、小集団を形成するモードであることをロボットの群れに放送する。個々のロボットは、この指令を受け取ると行動群モジュール内のリーダー役かアシスタント役のいずれかを選ぶことになる。ここで、どちらの役になるかは、Fig.4-5(a)のように操作者が選ぶも

のとした。リーダー役となったロボットは、自分がリーダーであることを他のロボットに伝達し、メンバを募集する。アシスタント役のロボットは、メンバになれるかをリーダーロボットに依頼することで Fig.4-5(b)のような小集団を形成する。ただし、各小集団の大きさを均等にするために、リーダーロボットは操作者によって決められたロボットの台数に到達すると、仲間を募集することを中止することとした。このため、小集団に加われなかったロボットは別の集団のところへ移動することにより均等な大きさの小集団が形成される。小集団の形成後は、Fig.4-5(c)のように、同じ集団に属しているロボットとのみ通信を行うこととしている。

操作者が作業分担を開始する指令をロボットに放送すると、各ロボットは、これまで選択していたリーダー役またはアシスタント役を破棄し、作業分担を行うための行動プログラムを選択するようになる。

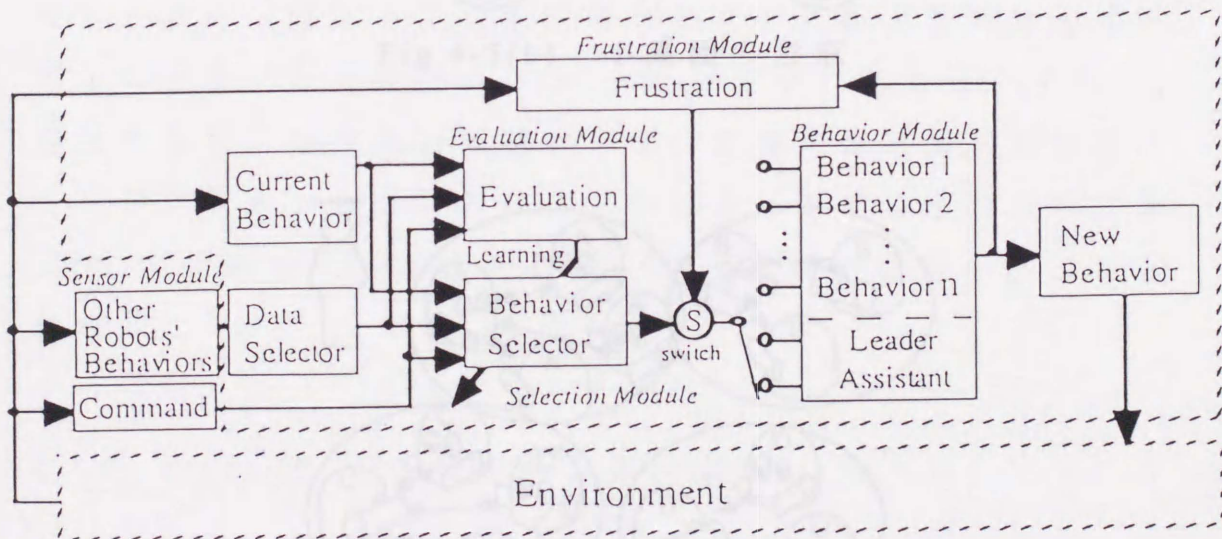


Fig.4-4 小集団を形成するための機能の追加

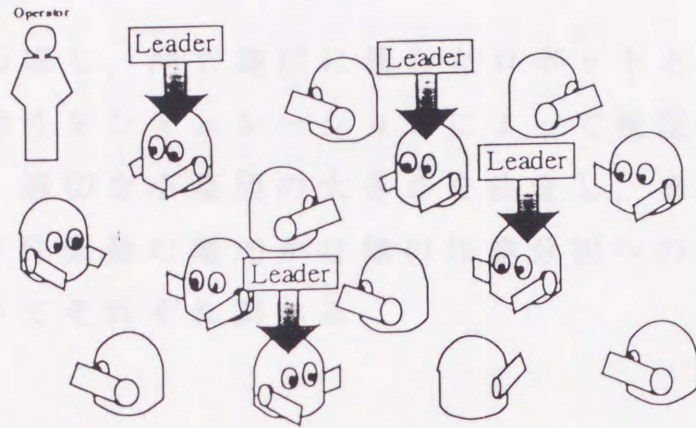


Fig.4-5(a) 操作者によるリーダーの選出

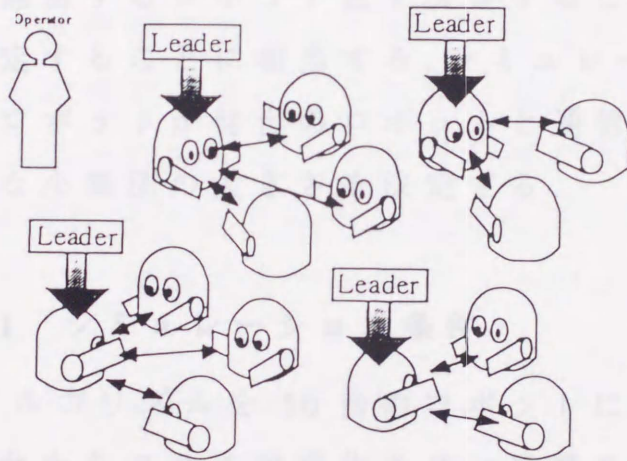


Fig.4-5(b) 小集団の形成

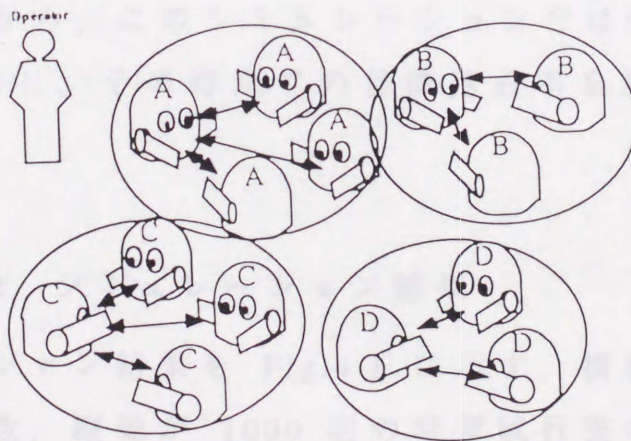


Fig.4-5(c) 通信するロボットの決定

4. 3. 3 小集団毎の通信の検証

小集団を形成し、同じ集団に属したロボットとのみ通信する手法の有効性をシミュレーションによって検証する。ここでは、まず、適切な小集団の大きさを決定し、その後、ロボット数および作業数の増加が目標の作業分担への到達に及ぼす影響についてそれぞれ調べる。

4. 3. 3. 1 小集団の大きさの決定

本節では、通信するロボット数を決定することは、小集団の大きさを決定することに相当する。シミュレーションでは、まず、個々のロボットが何台のロボットと通信するべきかを検証し、適切な小集団の大きさを決定する。

4. 3. 3. 1. 1 シミュレーション条件

Fig.4-4 のアルゴリズムを 50 台のロボットに適用し、通信する台数を 1 台から 7 台まで変化させ、目標の作業分担まで到達する成功率を調べた。ここで、学習試行数を多くすると、通信する台数や作業数に関わらず、目標の作業分担に到達してしまうことから、このシミュレーションでは学習試行数を 1000 回に制限し、その時点での目標達成率を調べることにした。

4. 3. 3. 1. 2 シミュレーション結果

シミュレーション結果を Fig.4-6 に示す。横軸が通信したロボットの台数、縦軸が 1000 回の学習試行後の目標到達率である。このシミュレーションから、通信するロボット数が

少ないと目標到達率が低いことが確認された。そこで、通信する台数を多くすると、目標到達率は高くなることが確認できた。しかし、通信するロボット数が5台以上になると目標到達率は再び減少することが確認できた。

4. 3. 3. 1. 3 考察

このシミュレーションを通して、通信するロボット数が少ないと目標到達率が低いことが確認できた。これは、個々のロボットが得る情報量が少なすぎるためと考えられる。一方、通信するロボット数が5台以上になった場合にも目標到達率が低くなることが確認できた。これは、通信するロボット数の増加に伴い、個々のロボットの学習領域が大きくなったため、1000回の学習試行数では不十分であったためと考えられる。このように、通信するべきロボットの台数には制限があることが分かった。

また、ロボットに遂行させる作業数について検討を行った結果、作業数が少ない場合にはあまり顕著には現れないものの、作業数が4種類や5種類になるとそれぞれ3台、4台のロボットと通信するべきという傾向が見られた。つまり、作業数が増加すると通信するべきロボット数も増加し、また小集団の大きさは作業数と等しくすると良い傾向があることが分かった。

したがって、小集団の大きさを決定する際には、これらのシミュレーション結果に基づいて決定する必要があるといえる。

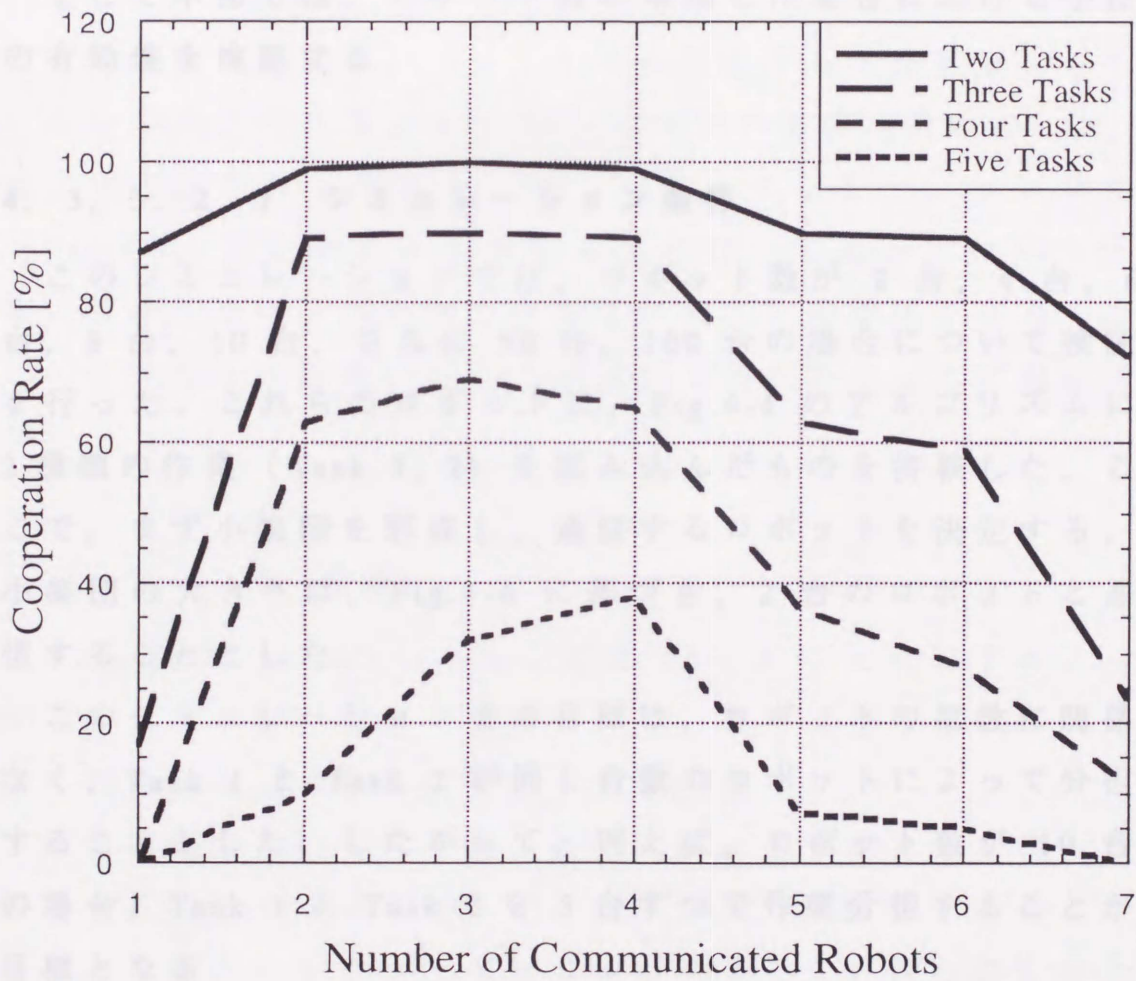


Fig.4-6 通信すべきロボット数と目標到達率の関係

4. 3. 3. 2 ロボット数の影響

同じ小集団に属するロボットとのみ通信する手法を適用することにより、全体としてのロボットの総数が増加しても、個々のロボットが通信するロボットの台数は変化しないため、ロボット数に依存しなくなる可能性がある。

そこで本節では、ロボット数が増加した場合における手法の有効性を検証する。

4. 3. 3. 2. 1 シミュレーション条件

このシミュレーションでは、ロボット数が2台、4台、6台、8台、10台、さらに50台、100台の場合について検証を行った。これらのロボットに、Fig.4-4のアルゴリズムに2種類の作業(Task 1, 2)を組み込んだものを搭載した。ここで、まず小集団を形成し、通信するロボットを決定する。小集団の大きさは、Fig.4-6に基づき、2台のロボットと通信することとした。

このシミュレーションでの目標は、ロボットの総数に関係なく、Task 1とTask 2が同じ台数のロボットによって分担することとした。したがって、例えば、ロボット数が10台の場合、Task 1とTask 2を5台ずつで作業分担することが目標となる。

行動選択の基準となるQ学習の初期値には乱数を代入した。また、各ロボットの自己評価基準は-1を、行動評価の値 $E(i, j)$ の初期値は全て0を与えた。そして、第2章の手法を用いて、単純な指令から目標の作業分担を生成した。ここでも、人間の代わりにコンピュータが自動的に指令を放送するものとした。

このシミュレーションでは，学習に対する初期値依存性の問題を考慮し，100000通りの初期値から検証を行っている．

4. 3. 3. 2. 2 シミュレーション結果

シミュレーションの結果を Fig.4-7 に示す．横軸が学習試行数，縦軸が協調成功率である．このシミュレーションでは，同じ小集団に属したロボットとのみ通信する手法を導入しているものの，その他のシミュレーション条件は前章の 3.4.1 節と同じ設定にしてある．そこでのシミュレーション結果 (Fig.3-5) と比較すると分かるように，ほとんどロボット数に依存しないシステムとなっていることが確認できた．また，目標となる作業分担への到達が促進されたことが確認できた．

4. 3. 3. 2. 3 考察

3.4.1 節のシミュレーション結果 (Fig.3-5) と比較すると，ロボット数に依存しなくなっていること，目標となる作業分担への到達が促進されたことが確認できた．

前者の理由としては，同じ集団に属するロボットとのみ通信させることで，ロボットの総数が例え 50 台や 100 台となっても，個々のロボットが通信するロボット数には影響を与えないためと考えられる．

また，後者の理由としては，得られる情報量が制限され，結果として学習する状態領域が小さくなり，速やかに学習が行われたためと考えられる．

4.3.3.3 作業数の影響

本節では、作業数が2個から10個までについて調べ、作業数が増加した場合における手法の学習性を検証する。

4.3.3.3.1 シミュレーション結果

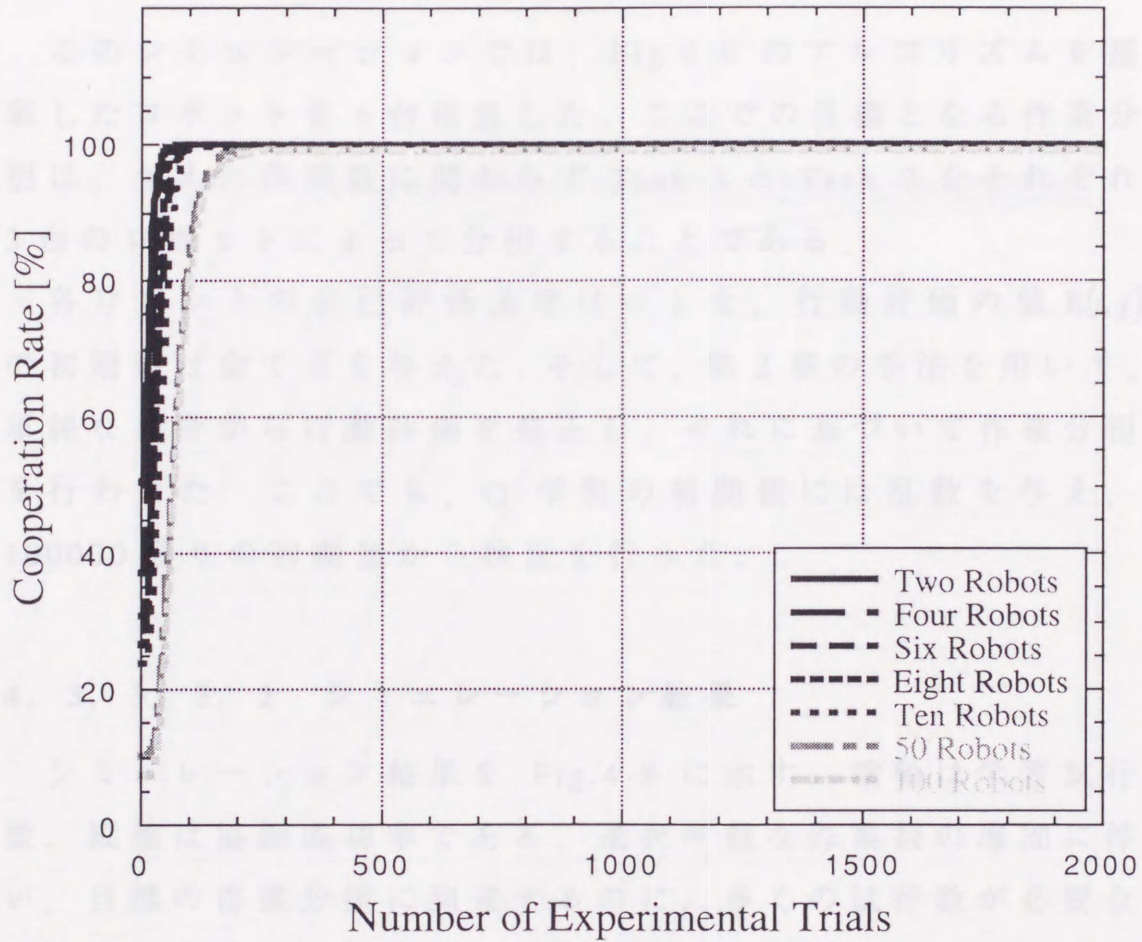


Fig.4-7 ロボット数が作業分担に及ぼす影響

2.3.2節のシミュレーション結果 (Fig.4-7) より、学習に少ない学習試行数で目標の作業分担に到達できることを確認した。これは、両側通信の導入によって学習する状態領域が小さくなり、速やかに学習が行われたためと考えられる。

4. 3. 3. 3 作業数の影響

本節では，作業数が2種類から5種類までについて調べ，作業数が増加した場合における手法の有効性を検証する。

4. 3. 3. 3. 1 シミュレーション条件

このシミュレーションでは，Fig.4-4 のアルゴリズムを搭載したロボットを6台用意した。ここでの目標となる作業分担は，与えた作業数に関わらず Task 1 と Task 2 をそれぞれ3台のロボットによって分担することである。

各ロボットの自己評価基準は-1を，行動評価の値 $E(i,j)$ の初期値は全て0を与えた。そして，第2章の手法を用いて，単純な指令から行動評価を修正し，それに基づいて作業分担を行わせた。ここでも，Q学習の初期値には乱数を与え，100000通りの初期値から検証を行った。

4. 3. 3. 3. 2 シミュレーション結果

シミュレーション結果を Fig.4-8 に示す。横軸は学習試行数，縦軸は協調成功率である。選択可能な作業数の増加に伴い，目標の作業分担に到達するのに，多くの試行数が必要なものの，最終的には目標に到達できることを確認した。

4. 3. 3. 3. 3 考察

2.5.2節のシミュレーション結果 (Fig.2-12) より，非常に少ない学習試行数で目標の作業分担に到達できることを確認した。これは，局所通信の導入によって学習する状態領域が小さくなり，速やかに学習が行われたためと考えられる。

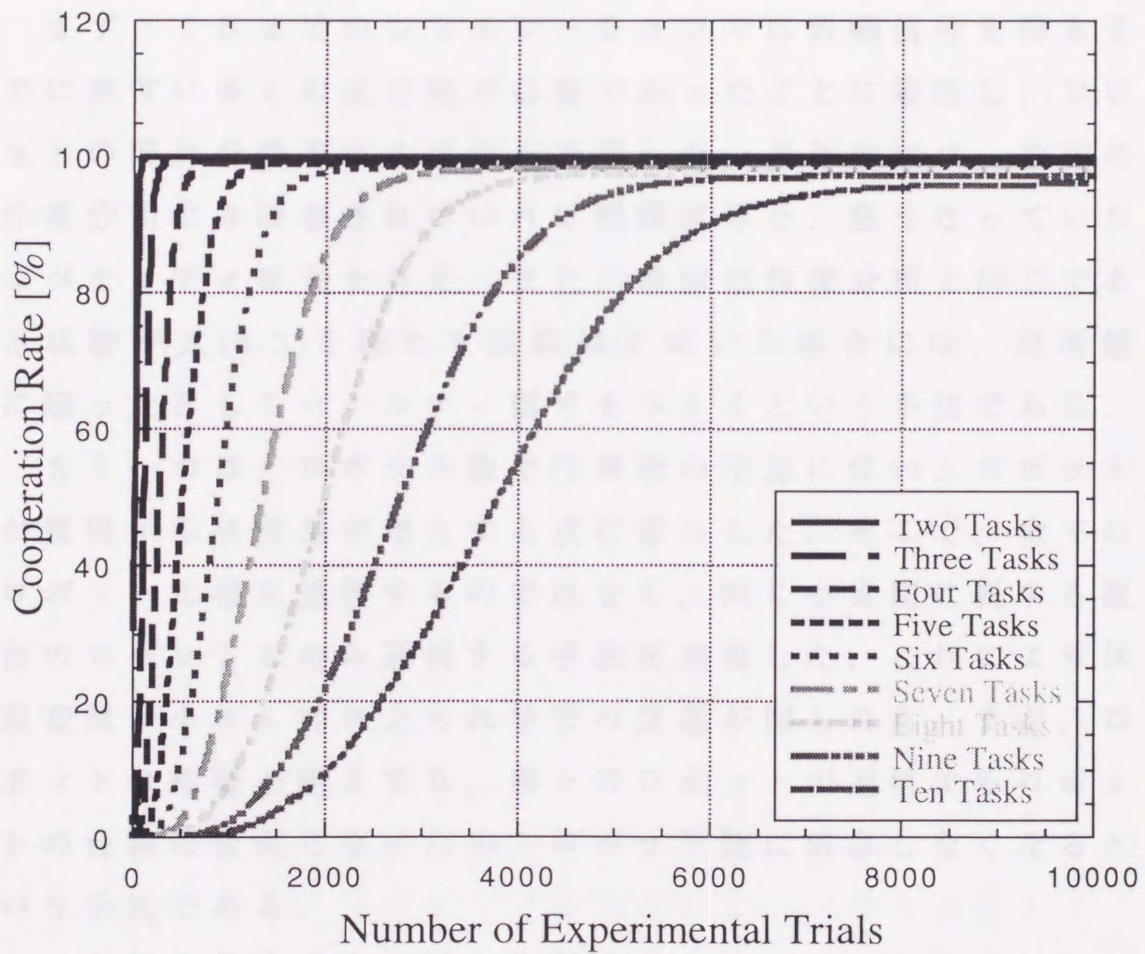


Fig.4-8 作業数が作業分担に及ぼす影響

4.4 まとめ

本章では、ロボット数や作業数が多くなると、目標の作業分担に到達するのに多くの学習試行数が必要となるという問題に焦点を絞り、それらを解決する手法について検討を行った。

まず、これまでのシミュレーションでは報酬信号を得るまでに非常に多くの試行数が必要であったことに着目し、ロボットの群れを誘導する手法を提案した。具体的には、前回の作業分担より改善されていれば報酬信号を、悪くなっていればペナルティ信号を与え、また、前回の作業分担と同じである状態が式(4-3)を満たす回数以上続いた場合には、局所解に陥ったとしてペナルティ信号を与えるという手法である。

もう一つは、ロボット数や作業数の増加に伴い、ロボットが獲得する状態量が増大する点に着目した。そこで、全てのロボットと相互通信するのではなく、同じ小集団に属する数台のロボットとのみ通信する手法を適用した。これにより状態空間が小さく押さえられ学習の促進が図られる。また、ロボットの総数が増えても、個々のロボットが通信するロボットの台数は変化しないため、ロボット数に依存しなくなるという手法である。

これらの手法の有効性を検証するためにシミュレーションを行い、これまでの手法に比べ、目標となる作業分担に早く到達できることを確認した。

第5章 実作業を想定した手法の検討

5.1 はじめに

前章までは、コンピュータによる単純な指令に基づいて複数のロボットに作業分担を行わせるための手法を提案した。ここでは、常に確実な指令が与えられるといった理想的な環境でシミュレーションを行った。本章では、本研究の手法を実作業に適用することを想定した上での検討を行う。

まず、本手法を実ロボットに実現することを想定し、ロボットに必要な装置と操作者に必要な装置を列挙する。そして、本手法が、現存する機器でも十分に実現可能であることを示す。

次に、本手法が有効的に活用されると考えられる具体的な作業例について述べる。また、その作業を想定したシミュレーションによって手法の有効性を再検証する。ここでは、これまでのシミュレーションでは検証できなかったことを確認する。例えば、これまでのシミュレーションでは、目標の作業分担を示す指令を確実に与えていた。しかし、実際には人間が指令を与えるため、指令の与え間違いや与え忘れなどが生じる。このような状況下での群れの作業分担を確認する。また、アルゴリズムに組み込んだものの、これまでのシミュレーションでは使われていなかったフラストレーションモジュールの有効性を確認する。

最後に、本手法が、実作業にも十分適用できる可能性を持つことをまとめて述べる。

5.2 ロボットと操作者の装置

本節では、本研究で述べてきた手法を実際のロボットに適用する場合を想定して、ロボット側と操作者側に必要な装置について検討する。

5.2.1 ロボットに必要な装置

本研究ではロボットに作業分担させるため、個々のロボットは、状況に応じて多彩な作業を遂行できる能力が要求される。そのようなロボットとしては、達成させる作業にもよるが、工場などで産業用として実用化されている汎用型ロボットなどが適用できると考えている。

このような多彩な作業を遂行できる能力を持つロボットにコンピュータを搭載し、ロボットを管理する操作者によってダウンロードされたプログラムに従って忠実に作業を遂行させる。ここで、ダウンロードされるプログラムは2種類ある。一つは本研究で提案したアルゴリズムが記述されたプログラムであり、全てのロボットに共通に与えられている。もう一つは、作業を遂行するための具体的な動作手順が記述された行動プログラムであり、個々のロボットの機構に応じて用意されている。後者のプログラムには、各作業毎に番号が割り当てられており、前者のプログラムによって作業番号が決定されると、その番号に対応した行動プログラムに切り替えて実行することで作業の変更を行うものとした。

このような機構を持つロボットに、複数の行動プログラムとそれを処理するコンピュータを搭載することで、個々のロボットに多彩な作業を遂行させることができる。

このようなロボットに、他のロボットと作業分担をさせるための装置を付加させる。本研究の手法では、まず、局所通信する相手を決定するために小集団を形成し、その後、自己の小集団に属するロボットとの間で通信することとしている。このような通信機能を満たす装置としては、全てのロボットと相互通信できる能力と、個々のロボットを識別する能力を持つ必要があり、例えばイーサネット無線が適用できる。作業分担するために通信される内容は、その時点で遂行している作業番号だけなので、ロボットが多少多くても現在の通信技術で十分可能であると考えられる。

これらの装置の他に、人間からの指令を受け取る装置が必要である。この場合、ロボット数がさほど多くなければ、上記のイーサネット無線を使って個々のロボットに指令を与えることも可能である。しかし、ロボット数が非常に多い場合には、一度に全てのロボットに指令を送ることが困難であることから、放送という通信方法によって指令を送ることとしている。このため、個々のロボットには、人間から放送された指令を受信するラジオのような受信機が必要である。

本研究での手法を実現するためには、これらの装置をロボットに搭載する必要があるが、いずれも現存の装置で実現できると考える。

5. 2. 2 操作者に必要な装置

ロボットを管理する操作者は、まず、小集団を形成する際においてリーダーとなるロボットを選出しなければならない。このロボットの選出には、イーサネット無線を搭載したコンピュータを使って、個々のロボットに直接指名することとしている。

一方、意図した作業分担になるよう、操作者はロボットの群れに対し指令を与えなければならない。この場合、ロボット数が非常に多い場合においても、ロボットの群れ全体に対して一度に指令を伝達しなければならないという理由から、指令を放送することとした。指令を放送する装置には、作業分担の開始、終了、報酬、ペナルティの4つのスイッチを用意しておく。これらのスイッチは、押している間だけ、それぞれの信号を広範囲に放送する。個々のロボットは、これらの放送を受信し、作業分担を行う。

具体的には、操作者が作業分担の開始のスイッチを押すと、個々のロボットは行動選択モジュール内の行動プログラムを選ぶようになり、作業分担が開始される。ここで、操作者がロボットの群れ全体を観察し、報酬またはペナルティのスイッチを押すことで、群れ全体としての指令が放送される。どちらのスイッチも押されていない場合には、個々のロボットの持つ行動評価の値が適切であり、修正の必要がないものと判断し、それらを用いて学習が行われる。目標としていた作業が達成されたら、操作者は作業分担終了のスイッチを押すことによって、作業を中止させる。

これらは操作者側に必要な装置であるが、いずれも現存の装置で実現可能であると考えられる。

5.3 適用する作業例

本節では、本研究で提案した手法を有効的に利用できる作業について検討する。

本研究での手法は、幾つかのサブタスクを複数のロボットによって分担させる作業を対象としている。このため、幾つかの離散的なサブタスクから構成され、それらを並列的に処理できる作業である必要がある。また、操作者が作業の様子を容易に観察できるように、作業結果を一箇所に集約できる作業であることが望ましい。

このような作業例として、組立作業が挙げられる。組立作業では、一般に、幾度かの工程を経て完成されるため、各工程をそれぞれ離散的なサブタスクとしてプログラムすることができる。また、それらを並列的に処理することも可能である。さらに、組立作業では、ある一定時間内に完成される部品数が多ければ、適切な配分であると容易に判断できることから、作業結果が集約されているといえる。

組立作業では、各工程によって遂行に必要な時間が異なるため、多くの時間が必要とする工程にはそれだけ多くのロボットを配分する必要がある。この場合、従来のように人間が全てのロボットに作業を割り当てる手法では、何台のロボットを各工程に割り当てるべきか事前に決定する必要があり、また分担を変更するためには、もう一度プログラムを与え直さなければならなかった。しかし、本手法の場合、各工程に何台のロボットを割り当てるのか分からなくても、実際にロボットに工程を分担させた結果から適切な作業分担に導ける点において有効であると考えられる。

5.4 実作業を想定したシミュレーション

コンピュータ上に組立作業を行わせる環境を用意し、実際の人間の指令によってロボットの作業分担を制御できることを示す。また、そのシミュレーションを通して、手法の有効性を検証する。

5.4.1 シミュレーション条件

ここでは、本研究での手法を組立作業に適用した場合を想定し、そこでの作業分担について検証を行う。このため、組立作業を単純化したモデルを用いてシミュレーションを行った。ここでの組立作業には、Task 1, Task 2, Task 3といった3つの工程があり、それらを順に全て行うことで組み立てが完成することとした。

シミュレーションの初期状態を Fig.5-1 に示す。大きな円は作業環境であり、右上と左と右下の3つの領域に区分され、それぞれ Task 1, Task 2, Task 3 を遂行するための作業領域である。それぞれの作業領域にある合計6個の小さな円がロボットである。また、作業環境の中央部には組み立てに必要な部品が用意されており、それらの部品を用いて各工程を遂行する。ロボットは、各領域を任意に行き来することができ、これにより作業変更が行えるものとした。

具体的な作業手順を、Fig5-2 を使って示す。はじめ、Fig.5-1 のように無作為にロボットが配置されているため、まず、ロボットは選択した作業に対応した作業領域に移動する。その後、Task 1 を選択したロボットは、Fig.5-2 における①の地点で一次加工用の部品を入手し②の地点に移動する。そして、

ロボットは②の地点から④の地点まで壁伝い（③の状態）に移動する。このシミュレーションでは、この壁伝いをしている時間が、実作業での一次加工を行っている時間に相当するものとしている。したがって、実作業において多くの時間が必要な工程は、シミュレーション上で遅く移動することで表現されている。このロボットが④の地点に到達する頃には、Task 1の加工が終了されるため、これを、Task 2を選択しているロボットに受け渡す。その後、⑤の地点に移動し、作業の継続または変更が行われる。作業を変更する場合には、部品の置いてある所を時計回りで移動する。

Task 2を遂行するロボットは、⑥の地点で二次加工用の部品を入手し、さらに、④の地点で一次加工が終了した部品を受け取り、二次加工をし、⑦の地点に移動する。同様に、Task 3にいるロボットは、⑧の地点で三次加工用の部品を入手し、⑦の地点で二次加工が終了した部品を受け取り、三次加工に入る。そして、⑨の地点で三次加工が終了した完成品を搬出する。

シミュレーションでは、各加工に要する時間を設定しなければならない。そこで、はじめ、Task 1, Task 2, Task 3にそれぞれ要する時間を1:2:3の比率で与えることとした。そして、しばらくしてから、加工内容が変更されたとして、各加工に要する時間を3:2:1に変更した。これらの加工に要する時間比率は、ロボットを管理する人間もロボットも未知とする。

このシミュレーションでは、3つの工程を6台のロボットに分担することとなる。ここでの目標は、単位時間内においてできるだけ多くの部品を完成させることである。したがって、加工に多くの時間を要するTaskには、多くのロボッ

トを配置しないと作業が円滑に行えないといえる。操作者はコンピュータ上のロボットを観察し、報酬信号またはペナルティ信号をコンピュータのキーボードを介して指示することとした。

このシミュレーションは作業数が3種類であることから、3台からなる小集団を形成させている。また、フラストレーションが起こるまでの時間は、最も加工に時間がかかるであろうと予想される作業時間の2倍に設定してある。

5.4.2 シミュレーション結果

シミュレーション結果を Fig.5-3(a)~Fig.5-3(f)に示す。ロボットに作業を開始させると、個々のロボットはそれぞれ最も適切と思われる Task を遂行する。その結果、Fig.5-3(a)の作業分担が確認できた。ここで、Task 2 を遂行しているロボットは1台しかいないため常に加工を行っているのに対し、Task 1 のロボットは一次加工の済んだ部品を Task 2 のロボットに渡そうと待機し、また、Task 3 のロボットは二次加工の済んだ部品を受け取ろうと待機しているのが観察された。結果として、三次加工の済んだ完成品の数は、最も負担の多かった Task 2 の加工した数に相当していた。このように、遂行すべき作業が存在するのにも関わらず、待機するロボットが多く存在するという状況は、適切な作業分担ではないことが明らかであることから、操作者はロボットの群れにペナルティ信号を放送した。

個々のロボットは、操作者からのペナルティ信号を受信し、これに基づいて自己の行動評価を修正した。さらに、これに基づいて作業を切り替えたところ、Task 2 を選ぶロボットが

いなくなってしまう、結果として Fig.5-3(b)の状態に陥ることが確認できた。ここで、しばらくすると、そのロボットは Fig.5-3(c)のように別の作業に切り替える様子が確認できた。この作業の切り替えは、アルゴリズムに組み込んであるフラストレーションモジュールの働きによるものであり、不都合な状況を脱出するのに有効であることを確認した。

その後も、操作者が状況に応じて報酬信号やペナルティ信号を放送したところ、結果として、Fig.5-3(d)に示すように、単位時間あたり最も多くの部品を完成できる作業分担に到達し、その関係を継続する様子が確認できた。

しばらく、その作業分担を継続していたところ、加工内容が変更されたとして、各加工に要する時間が大幅に変化した。このため、Fig.5-3(d)の作業分担を継続していたところ、Fig.5-3(e)のように待機するロボットが多くなり、単位時間あたりの完成品の数も減少した。このため、操作者はペナルティ信号を放送し、新しい作業分担となるよう群れを制御した。結果として、Fig.5-3(f)に示すように、単位時間あたり最も多くの部品を完成できる作業分担に制御できることを確認した。

5. 4. 3 考察

前章までのシミュレーションでは、ロボットの群れに対する指令をコンピュータが確実に与えていたが、ここでは実際に人間が指令を与えることとした。この場合においても、容易に適切な作業分担が生成できることを確認した。

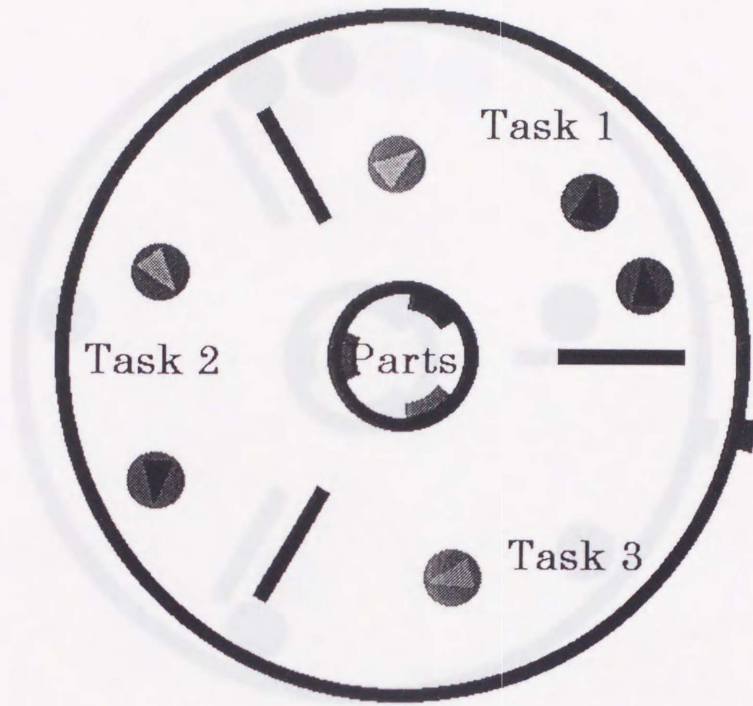


Fig.5-1 シミュレーションの初期状態

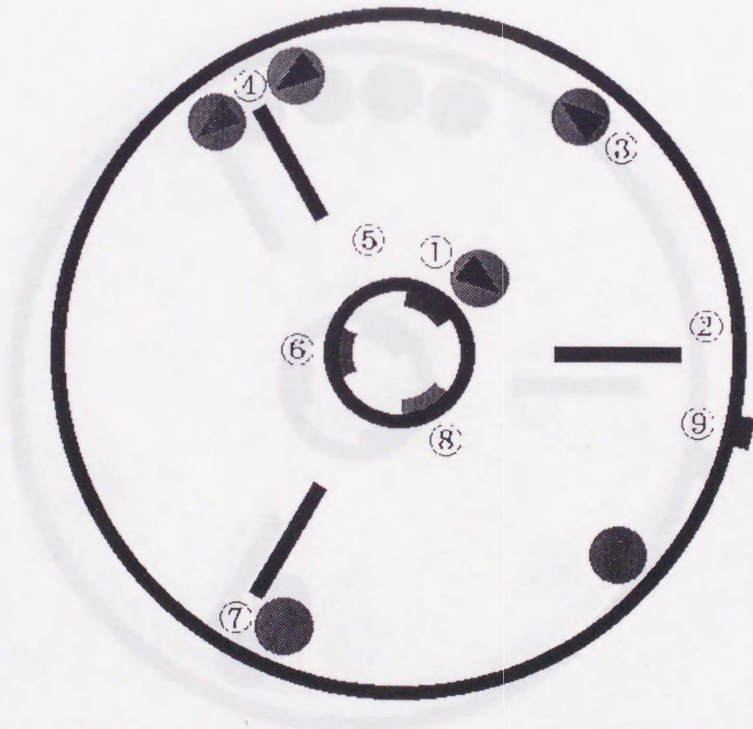


Fig.5-2 各配置におけるロボットの行動

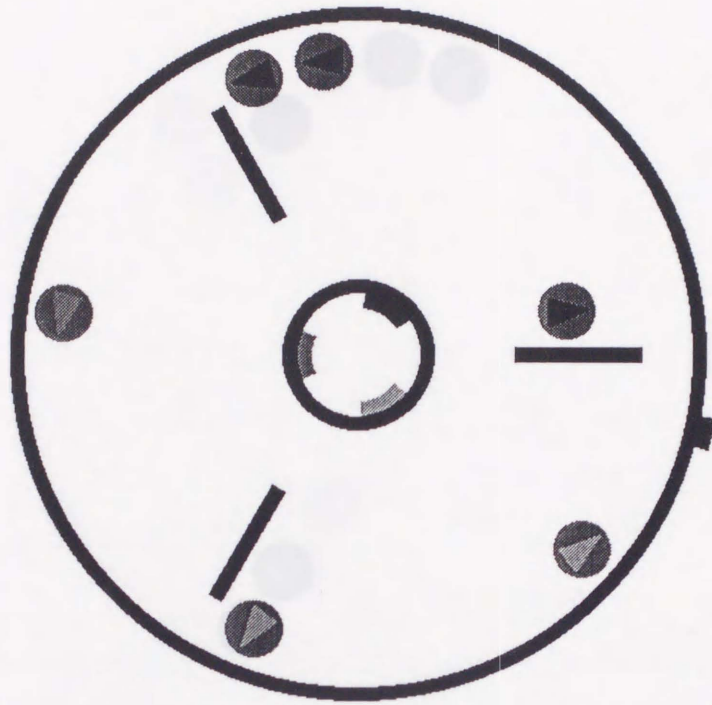


Fig.5-3(a) 無駄のある作業分担

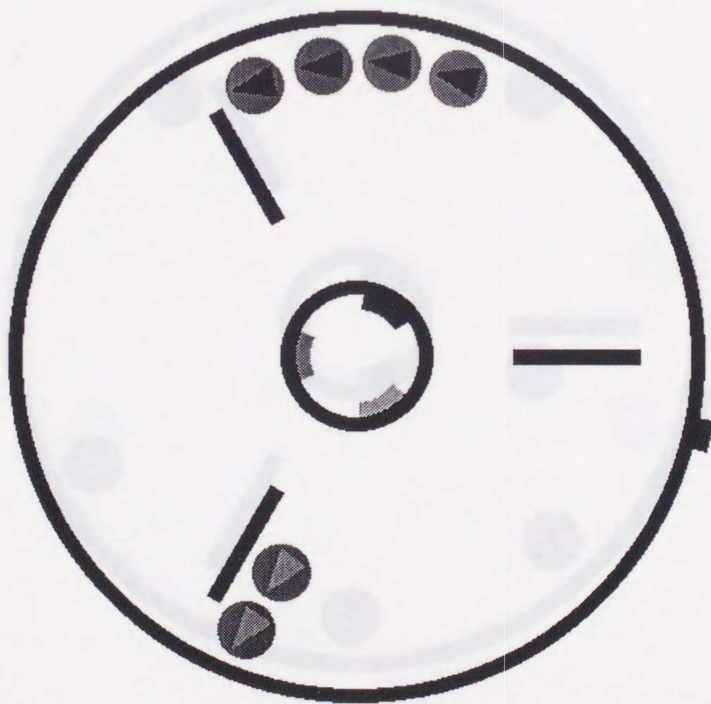


Fig.5-3(b) 作業が遂行できない不都合な状態

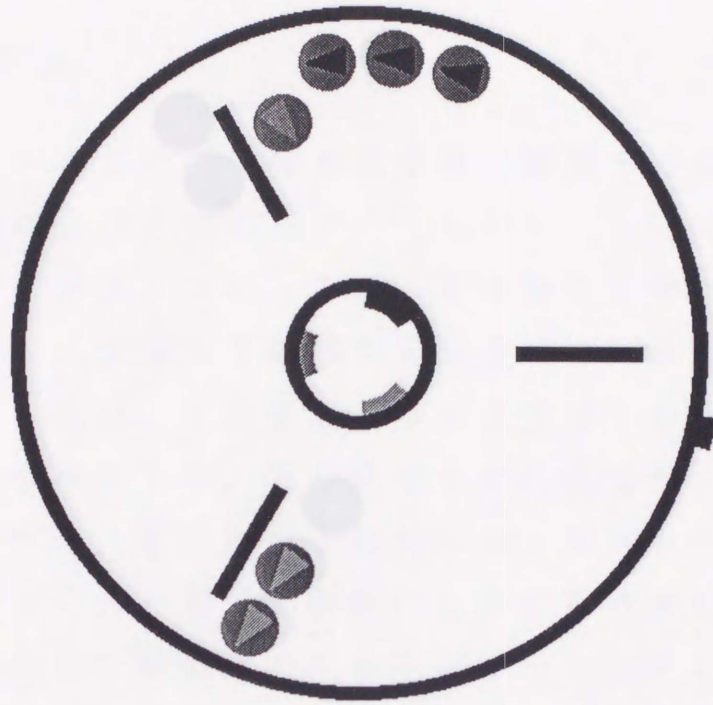


Fig.5-3(c) フラストレーションによる不都合な状態の脱出

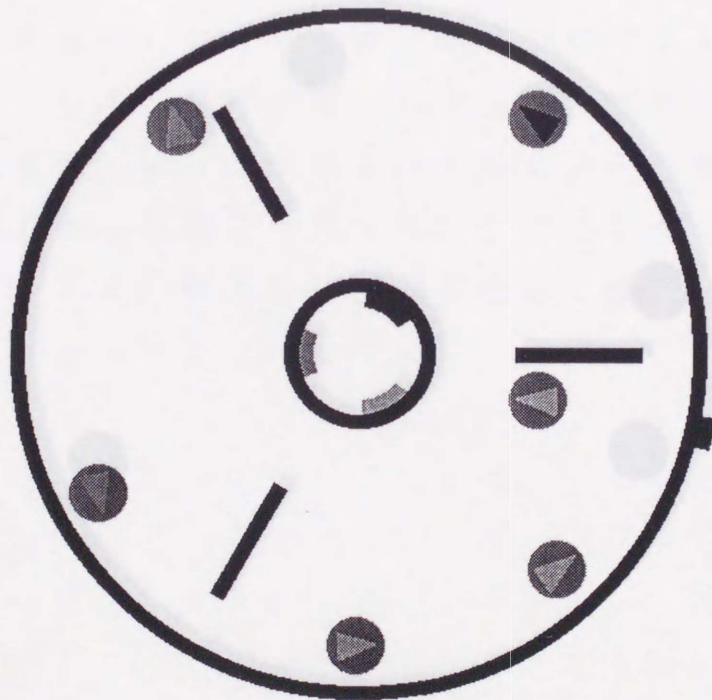


Fig.5-3(d) 適切な作業分担の生成

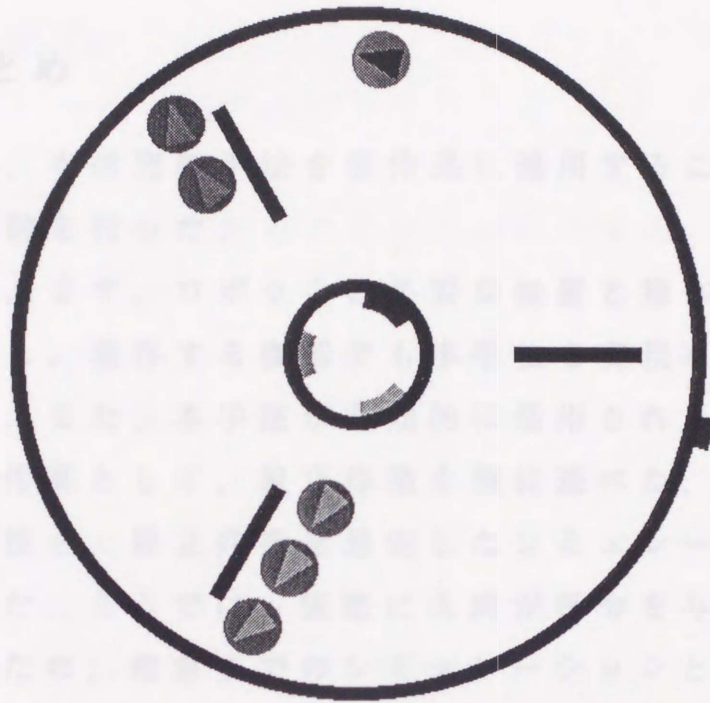


Fig.5-3(e) 組立内容が変更された時の状態

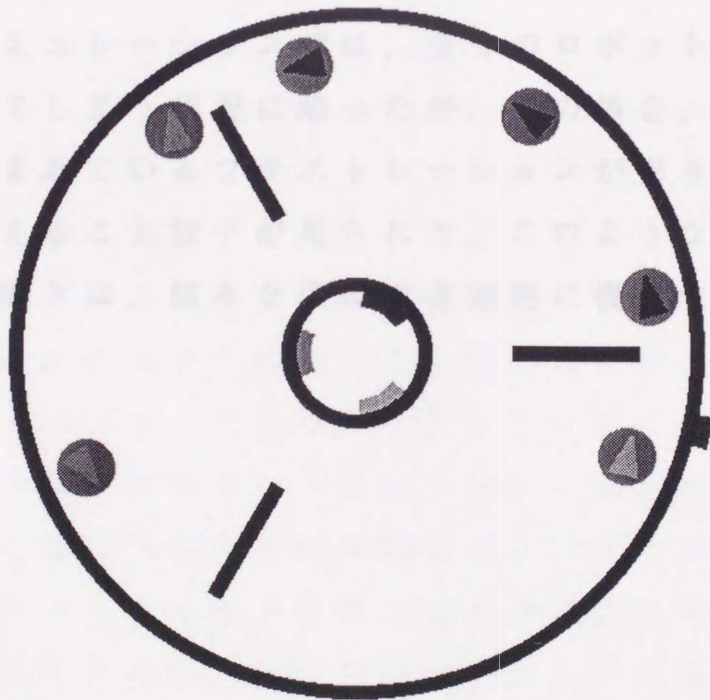


Fig.5-3(f) 組立内容の変更後の適切な作業分担の生成

5.5 まとめ

本章では、本研究の手法を実作業に適用することを想定した上での検討を行った。

ここでは、まず、ロボットに必要な装置と操作者に必要な装置を列挙し、現存する機器でも本手法を実現可能であることを示した。また、本手法が有効的に活用されることが考えられる具体的な作業として、組立作業を例に述べた。さらに、本手法の有効性を、組立作業を想定したシミュレーションによって検証した。ここでは、実際に人間が指令を与えることとした。このため、前章までのシミュレーションと異なり、指令の与え間違いや与え忘れなどが生じていたが、このような状況下においても、操作者の意図した作業分担が生成できることを確認した。

また、シミュレーションでは、全てのロボットが特定の作業を選択してしまう状況に陥ったが、この場合、アルゴリズムに組み込まれているフラストレーションが働き、異なる作業に切り替えること様子が見られた。このようなフラストレーションの働きは、様々な環境で有効的に機能するであろうと考えられる。

第6章 結論

数々のロボット技術を統合することで、近い将来、多彩な作業を自律的に遂行できるロボットが存在すると考えられる。一方、単体のロボットに搭載できる機能や能力には限界があることから、将来、同一の目標を達成させるために複数のロボットを用意し、状況に応じて作業分担させることが予想される。この場合、ロボットを管理する操作者は、自己の意図した作業分担になるように制御できなければならない。

このような背景から、本論文では、複数ロボットに操作者の意図した作業分担を行わせる手法を研究した。本研究の結論を要約すると以下の通りである。

第2章「作業分担の生成アルゴリズムの構築」では、まず、複数のロボットに作業分担に必要な機能を列挙し、それらのモジュール化を行った。具体的には、センサモジュール、行動選択モジュール、行動群モジュール、評価・学習モジュール、フラストレーションモジュールといった5つのモジュールから構成される基本アルゴリズムを構築した。

このアルゴリズムの第一の特徴は、全ての作業に対する行動プログラムをあらかじめ搭載していることにある。これにより、従来のロボットでは作業を変更する度にプログラムを与え直さなければならなかったのに対し、本手法では個々のロボットが任意に作業変更が可能となる。また、第二の特徴としては、目標となる作業分担を示す評価値をあらかじめ共通に与えておくものの、どのロボットがどの作業を遂行するのかといった具体的な作業分担を人間が与えておく必要がない点にある。

この章で構築したアルゴリズムを用いることで、目標の作

業分担を示す評価を与えておくものの、ロボットを管理する操作者がプログラムを与え直すことなく意図した作業分担が生成できることを確認した。

第3章「指令に基づく群れの制御手法の提案」では、ロボットを管理する操作者の意図に応じて作業分担するための手法について検討を行った。ロボットが多く存在する場合には、操作者は、個々のロボットを個別に観察したり個別に指示を与えたりできないことから、本研究では群れ全体に対する単純な評価からロボットの群れを制御する手法を提案した。

ここでは、操作者がロボットの群れを観察し、目標の作業分担であれば「良い」という報酬信号を、そうでなければ「悪い」というペナルティ信号を放送することとした。個々のロボットは、これらの指令に基づいて自己の評価を修正し、その評価に基づいて作業選択の学習を行うことで、結果として、指令によって間接的に作業分担を操作できる手法である。

また、シミュレーションによって、単純な指令から意図した作業分担へ操作できることを確認した。

第4章「群れの効果的制御手法の提案」では、ロボット数や作業数が多くなると、目標の作業分担に到達するのに多くの学習試行数が必要となるという問題に焦点を絞り、それらを解決する手法について検討した。

まず、これまでのシミュレーションでは報酬信号を得るまでに非常に多くの試行数が必要であったことに着目し、ロボットの群れを誘導する手法を提案した。具体的には、前回の作業分担より改善されていれば報酬信号を、悪くなっていればペナルティ信号を与える。また、同じ作業分担が式(4-3)の回数以上繰り返された場合には、局所解に陥ったとしてペナルティ信号を与える手法である。これにより、偶然的に報

酬が得られるのを待つ従来の手法に比べ、早く目標の作業分担に誘導することができる。

もう一つは、ロボット数や作業数の増加に伴い、ロボットが獲得する状態量が増大する点に着目した。そこで、全てのロボットと相互通信するのではなく、同じ小集団に属する数台のロボットとのみ通信する局所的通信を適用した。これにより状態空間が小さく押さえられ学習の促進が図られる。また、ロボットの総数が増えても、個々のロボットが通信するロボットの台数は変化しないため、ロボット数に依存しなくなるという手法である。

これらの手法を適用することにより、目標となる作業分担に早く到達できることをシミュレーションで確認した。

第5章「実作業を想定した手法の検討」では、本研究の手法を実作業に適用することを想定した上での検討を行った。

ここでは、まず、ロボットに必要な装置と操作者に必要な装置を列挙し、現存する機器でも本手法を実現可能であることを示した。また、本手法が有効的に活用されると考えられる具体的な作業として、組立作業を例に述べた。さらに、本手法の有効性を、組立作業を想定したシミュレーションによって検証した。ここでは、実際の間人が指令から、意図した作業分担が生成できることを確認した。

以上、本研究では、複数のロボットに対し操作者が単純な指令を放送するだけで、意図した作業分担に制御できる手法を提案し、その有効性をシミュレーションで確認した。これらの結果から、本論文での手法は、複数のロボットが様々な用途に適用されるであろう将来において有効な制御手法であると考えられる。

謝辞

本研究について、終始懇切なる御指導、御鞭撻を賜りました、通商産業省 工業技術院 機械技術研究所 ロボット工学部 谷江和雄部長（筑波大学 構造工学系 併任教授）に感謝し、心から御礼申し上げます。また、本研究をまとめる段階で、広い範囲に渡り貴重なご助言を賜りました同大学 構造工学系 太田道男 教授、同大学 電子・情報工学系 油田信一 教授、通商産業省 工業技術院 電子総合研究所 知能システム部 築根秀男 部長（同大学 電子・情報工学系 併任教授）、通商産業省 工業技術院 機械技術研究所 小鍛治繁 首席研究官（同大学 構造工学系 併任教授）に深甚なる謝意を表します。

また、本研究を遂行するにあたり、絶えず御指導を賜りました 機械技術研究所 ロボット工学部 バイオロボティクス研究室 柴田崇徳 研究官に心より御礼申し上げます。

さらに、研究内容の討論を始め、種々の面で暖かいご支援を戴きました 機械技術研究所 ロボット工学部 バイオロボティクス研究室の方々に謝意を表します。

参考文献

- 1) 中野: ロボット工学入門, オーム社 (1983).
- 2) Newton 別冊, ロボットのすべて, 教育社, 8/11 (1984).
- 3) 佐藤: 群知能ロボットの地平と将来展望, 計測と制御, Vol.31, No.11, 1115/1118 (1992).
- 4) K.Singh and K.Fujimura: Map Making by Cooperating Mobile Robots, Proc. of IEEE Int. Conf. on Robotics and Automation, 254/259 (1993).
- 5) 沼岡: 自律ロボット集団のチームワーク, ロボティクス・メカトロニクス講演会'92, 69/74 (1992).
- 6) D.McFarland: Towards Robot Cooperation, Proc. of the Third Int. Conf. on Simulation of Adaptive Behavior, 440/444 (1994).
- 7) 福田, 植山: マルチモジュールロボットー自己組織化ロボットを中心にー, 計測と制御, Vol.31, No.11, 1137/1142 (1992).
- 8) 日本ロボット学会誌: 特集 マルチエージェントロボットシステム, 10-4, 1/47 (1992).
- 9) 計測と制御: 特集 群知能ロボット, 31-11, 1115/1193 (1992).
- 10) 宮田 他: 機能分化を伴う協調搬送, ロボティクス・メカトロニクス講演会'98, 2A12-6(1/2) (1998).
- 11) 福田 他: 動的再構成可能ロボットシステムに関する研究 (第 22 報, 意思協調による自律分散ロボット群の行動の発生と組織化), 日本機械学会論文集(C編), 62-589, 3635/ 3642 (1995).

- 12) 福田 他: 動的再構成可能ロボットシステムに関する研究 (第 22 報, 意思協調による自律分散ロボット群の行動の発生と組織化), 日本機械学会論文集(C編), 62-589, 3635/ 3642 (1995).
- 13) 加賀, 福田: 動的再構成可能ロボットに関する研究 (第 108 報, 自律分散型ロボットシステムにおける集団現象の解析), ロボティクス・メカトロニクス講演会'98, 1CII3-6(1/2) (1998).
- 14) T.Kaga and T.Fukuda: An Oscillation Analysis on Distributed Autonomous Robotic System, Proc. of IEEE Int. Conf. on Robotics and Automation, 2846/2851 (1998).
- 15) 村田 他: 自己修復する機械 - 自律機械要素フラクタム -, 第 3 回インテリジェントシステム・シンポジウム, 387/394 (1993).
- 16) S.Murata, H.Kurokawa and S.Kokaji: Self-Assembling Machine, Proc. of IEEE Int. Conf. on Robotics and Automation, 441/448 (1994).
- 17) C.R.Kube and H.Zhang: Controlling Collective Tasks with an ALN, Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 289/293 (1993).
- 18) C.R.Kube and H.Zhang: Collective Robotics; From Social Insects to Robots, Adaptive Behavior, Vol.2, No.2, 189/218 (1994).
- 19) 新井 他: 複数移動ロボット系の局所的な通信に関する研究, 日本ロボット学会誌, Vol.12, No.6, 886/892 (1994).
- 20) J.Wang: On Sign-board Based Inter-Robot Communication in Distributed Robotic Systems, Proc. of IEEE

- Int. Conf. on Robotics and Automation, 1045/1050 (1994).
- 21) M.Kaiser et al.: Learning Coordination Skills in Multi-Agent Systems, Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 1488/1495 (1996).
- 22) 森島, 平井: マルチロボットによる作業計画の研究 (第2報, 1対多のインタフェース), 第12回日本ロボット学会学術講演会, 999/1000 (1994).
- 23) 横田 他: マルチロボット環境におけるヒューマン・インタフェース, 日本機械学会ロボティクス・メカトロニクス講演会, 513/516 (1994).
- 24) A.Nakamura, et al.: Multiple Mobile Robot Operation by Human, Proc. of IEEE Int. Conf. on Robotics and Automation, 2852/2857 (1998).
- 25) 中村 他: 人間による移動ロボット群操作システム, ロボティクス・メカトロニクス講演会'98, 2AI2-5(1/2) (1998).
- 26) 木村, 梶浦, 藤田: 人間が操縦するロボットとの協調-動作理解による補助-, 第12回日本ロボット学会学術講演会, 851/852 (1994).
- 27) Q.Chen and J.Y.S.Luh: Distributed Motion Coordination of Multiple Robots, Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 1493/1500 (1994).
- 28) 長谷川 他: 人の直観的動作を利用するマイクロロボット群の遠隔操作システム, 日本ロボット学会誌, 14-4, 567/573 (1996).
- 29) H.Asama, et al.: Development of Task Assignment System Using Communication for Multiple Autonomous Robots,

- Journal of Robotics and Mechatronics, 4-2, 122/127 (1992).
- 30) 小菅: 複数のマニピュレータによる協調制御, 日本ロボット学会誌, 10-4, 439/443 (1992).
- 31) J.Ota et al.: Transferring and Regrasping a Large Object by Cooperation of Multiple Mobile Robots, Proc. of IEEE Int. Conf. on Robotics and Automation, 543/548 (1995).
- 32) H.Osumi et al.: Cooperative Control of Two Industrial Robots with Force Control Devices, Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 550/555 (1995).
- 33) 沢崎, 井上: 複数の知能ロボットによる協調作業, 第 1 回 日本ロボット学会ロボットシンポジウム, 271/276 (1992).
- 34) L.E.Parker: ALLIANCE -An Architecture for Fault Tolerant Cooperative Control of Heterogeneous Mobile Robots-, Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 776/783 (1994).
- 35) M.J.Mataric, et al.: Cooperative Multi-Robot Box-Pushing, Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 556/561 (1995).
- 36) R.G.Brown and J.S.Jennings: A Pusher/Steerer Model for Strongly Cooperative Mobile Robot Manipulation, Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 562/568 (1995).
- 37) T.Fraichard and C.Laugier: Planning Movements for Several Coordinated Vehicles, Proc. of IEEE/RSJ Int. Symp. on Intelligent Robots and Systems, 466/472 (1989).

- 38) 柴田 他: Genetic Algorithm を用いた移動ロボットの最適行動計画, 第 9 回日本ロボット学会学術講演会, 589/590 (1991).
- 39) プレムウッティ, 油田: 複数台の移動ロボットの遠慮的協調の一例, 第 8 回日本ロボット学会学術講演会, 891/894 (1990).
- 40) プレムウッティ, 油田: 簡単な自律型移動ロボットの協調行動モデルのインプリメンテーション, 第 9 回日本ロボット学会学術講演会, 591/592 (1991).
- 41) 浅間 他: 通信及びルールに基づく複数の移動ロボットの衝突回避, 第 1 回日本ロボット学会ロボットシンポジウム, 73/78 (1991).
- 42) F.Montesello et al.: Implicit Coordination in a Multi-Agent System using a Behavior-based Approach, Distributed Autonomous Robotic Systems 3 (T.Lueth, R.Dillmann, P.Dario and H.Worn Eds.), Springer-Verlag Berlin Heidelberg NewYork, 351/360 (1998)
- 43) 川端 他: 全方向移動ロボットによる協調サッカープレイ, 日本機械学会ロボティクス・メカトロニクス講演会'98 講演論文集, 1AIII4-4 (1998).
- 44) B.B.Werger: Principles of Minimal Control for Comprehensive Team Behavior, Proc. of IEEE Int. Conf. on Robotics and Automation, 3504/3509 (1998).
- 45) J.H.Kim et al.: Path Planning and Role Selection Mechanism for Soccer Robots, Proc. of IEEE Int. Conf. on Robotics and Automation, 3216/3221 (1998).

- 46) 浅間: マルチエージェントロボットシステムの研究の
動向と展望, 日本ロボット学会誌, **10-4**, 429/432 (1992).
- 47) M.J.Mataric: Issues and Approaches in the Design of
Collective Autonomous Agents, Robotics and Autonomous
Systems, Vol.16, Nos.2-4, 321-331 (1995).
- 48) M.J.Mataric: Learning to Behave Socially, Proc. the Third
Int. Conf. on Simulation of Adaptive Behavior, 453-462
(1994).
- 49) R.A.Brooks, et al.: Lunar Base Construction Robots, Proc.
IEEE/RSJ Int. Workshop on Intelligent Robots and Systems,
389/392 (1990).
- 50) F.Michaud, et al.: Learning from History for Adaptive
Mobile Robot Control, Proc. IEEE/RSJ Int. Conf. on
Intelligent Robots and Systems, 1865/1870 (1998).
- 51) 金森, 梶谷: 複数台の移動ロボットのための通信方式,
第10回 日本ロボット学会学術講演会, 287/290 (1992).
- 52) T.Suzuki et al.: Cooperation between a Human Operator and
Multiple Robots for Maintenance Tasks at a Distance,
Distributed Autonomous Robotic Systems 2, Springer-
Verlag Tokyo, 50/59 (1996).
- 53) 尾崎 他: 自律分散型ロボットシステムにおける通信を
用いた協調チーム編成手法の開発と評価
- 54) 倉林 他: 知的データキャリアを用いた, 自律移動ロボ
ットの協調作業, ロボティクス・メカトロニクス講演
会'98, 1CII2-3(1/2) (1998).
- 55) K.Hosokawa et al.: Self-Organizing Collective Robots with
Morphogenesis in a Vertical Plane, Proc. of IEEE Int. Conf.

- on Robotics and Automation, 2858/2863 (1998).
- 56) R.G.Smith: The Contract Net Protocol -High-Level Communication and Control in a Distributed Problem Solver-, IEEE Trans. on Computer, C-29-12, 1104/1113 (1980).
- 57) C.L.Pape: A Combination of Centralized and Distributed Methods for Multi-Agent Planning and Scheduling, Proc. of IEEE Int. Conf. on Robotics and Automation, 488/493 (1990).
- 58) 長田: 自律分散ロボットシステムの制御, コンピュータロール, No.34, コロナ社 (1991).
- 59) F.R.Noreils : Architecture for Cooperative and Autonomous Mobile Robots, Proc. of IEEE Int. Conf. on Robotics and Automation, 2703/2710 (1992).
- 60) A.Cai, et al.: Integration of Distributed Sensing Information in DARS Based on Evidential Reasoning, Distributed Autonomous Robotic Systems 2, 268/279 (1996).
- 61) 浅間 他: 通信を用いた分散的管理に基づく複数の自律型ロボットの協調的作業分担決定手法, 日本ロボット学会誌, 10-7, 955/963 (1992).
- 62) 油田, プレムウッティ: 複数の自律移動ロボットの協調に関する検討 -遠慮的協調の提案-, 第8回日本ロボット学会学術講演会, 887/890 (1990).
- 63) 國吉: 観察に基づく協調 -第1報 協調行動パターンの分類に関する検討-, 第9回日本ロボット学会学術講演会, 583/584 (1991)

- 64) 國吉 他: 観察に基づく協調 - 第 5 報 基本協調行動アーキテクチャー, 第 12 回日本ロボット学会学術講演会, 855/856 (1994)
- 65) Y.Kuniyoshi, et al.: Cooperation by Observation - The Framework and Basic Task Patterns -, Proc. of IEEE Int. Conf. on Robotics and Automation, 767/774 (1994).
- 66) 堀内, 谷: 分散実行可能な遺伝的アルゴリズムを用いた移動ロボット群の行動学習, 日本ロボット学会誌, Vol.11, No.8, 1212/1219 (1993).
- 67) 三上 他: 群協調行動の獲得を目的としたカオティック強化学習の提案, ロボティクス・メカトロニクス講演会'96, 1353/1354 (1996).
- 68) C.G.Atkeson and J.C.Santamaria: A Comparison of Direct and Model-Based Reinforcement Learning, Proc. of IEEE Int. Conf. on Robotics and Automation, 3557/3564 (1997).
- 69) 北川, 喜安: ゲーム理論と行動理論, 共立出版社 (1968).
- 70) A.ラハポート: 囚人ジレンマ紛争と協力に関する心理学的研究 -, 啓明社 (1983).
- 71) 西山: 勝つためのゲーム理論 - 適応戦略とは何か -, 講談社 (1986).
- 72) M.A.Nowak et al.: 囚人ジレンマと生物の進化, 日経サイエンス, 1995年8月号, 50/57 (1995).
- 73) 山内, 春木: 学習心理学 - 行動と認知 -, サイエンス社 (1985).
- 74) Maier, et al.: Studies of abnormal behavior in the rat:

- III. The development of behavior fixations through frustration, *Journal of Experimental Psychology*, Vol.26, 521/546 (1940).
- 75) 日高, 群れとなわばりの経済学, 岩波書店, 26/39 (1983).
- 76) J.Wang: Towards a Generic Computer Simulation Platform for Distributed Robotic System Development and Experiments, *Distributed Autonomous Robotic Systems 2*, Springer-Verlag Tokyo, 38/49 (1996)
- 77) 吉田 他: 群ロボットの局所的通信システム評価の一手法 - 大域的通信との比較 -, 第 14 回 日本ロボット学会 学術講演会, 1053/1054 (1996).
- 4) K. Ohkawa, T. Shibata and K. Tani: Self-Generating Algorithm of Evaluation for Cooperative Behavior among Distributed Autonomous Robots, *Proc. of Int. Symp. on Artificial Life and Robotics*, 135/126 (1997).
- 5) K. Ohkawa, T. Shibata and K. Tani: Self-Generating Method of Behavioral Evaluation for Reinforcement Learning among Multiple Coordinated Robots, *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1451/1436 (1997).
- 6) K. Ohkawa, T. Shibata and K. Tani: A Method for Controlling a Group of Robots by an Operator, *Proc. of 4th Int. Symp. on Distributed Autonomous Robotic Systems*, 309/318 (1995).
- 7) K. Ohkawa, T. Shibata and K. Tani: Self-Generating Algorithm of Evaluation for Cooperative Behavior, *Journal of Artificial Life and Robotics*, Vol.2, No.3, 133/143 (1995).

関連業績一覧

印刷論文

- 1) T.Shibata, K.Ohkawa and K.Tanie: Spontaneous Coordinated Behavior of Robots through Reinforcement Learning, Proc. of IEEE Int. Conf. on Neural Networks, 2908/2911 (1995).
- 2) T.Shibata, K.Ohkawa and K.Tanie: Spontaneous Behavior for Cooperation of Distributed Autonomous Robots, Proc. of Int. Symp. on Artificial Life and Robotics, (1996).
- 3) T.Shibata, K.Ohkawa and K.Tanie: Spontaneous Behavior of Robots for Cooperation -Emotionally Intelligent Robot System-, Proc. of IEEE Int. Conf. on Robotics and Automation, 2426/2431 (1996).
- 4) K.Ohkawa, T.Shibata and K.Tanie: Self-Generating Algorithm of Evaluation for Cooperative Behavior among Distributed Autonomous Robots, Proc. of Int. Sympo. on Artificial Life and Robotics, 123/126 (1997).
- 5) K.Ohkawa, T.Shibata and K.Tanie: Self-Generating Method of Behavioral Evaluation for Reinforcement Learning among Multiple Coordinated Robots, Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 1451/1456 (1997).
- 6) K.Ohkawa, T.Shibata and K.Tanie: A Method for Controlling a Group of Robots by an Operator, Proc. of 4th Int. Symp. on Distributed Autonomous Robotic Systems, 309/318 (1998).
- 7) K.Ohkawa, T.Shibata and K.Tanie: Self-Generating Algorithm of Evaluation for Cooperative behavior, Journal of Artificial Life and Robotics, Vol.2, No.3, 138/143 (1998).

- 8) K. Ohkawa, T. Shibata and K. Tanie: A Generation Method of Evaluation for a Robot Considering Relation with Other Robots, Journal of Robotics and Mechatronics, Vol.10, No.4, 284/288 (1998).
- 9) K. Ohkawa, T. Shibata and K. Tanie: Method for Generating of Global Cooperation Based on Local Communication, Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 108/113 (1998).
- 10) K. Ohkawa, T. Shibata and K. Tanie: Task Sharing in a Multiple Robot System Controlled by Macro-Commands of an Operator, Proc. of Int. Symp. on Artificial Life and Robotics, 24/27 (1999).

その他の論文

口頭発表

- 1) 柴田, 大川, 谷江: ロボットの協調動作の自発的生成アルゴリズムの基礎的研究, 第5回 インテリジェントシステムシンポジウム, 359/362 (1995).
- 2) 柴田, 大川, 谷江: 分散型複数ロボットの自発的協調行動の生成, 第13回 日本ロボット学会学術講演会, 641/642 (1995).
- 3) 大川, 柴田, 谷江: 自発的に協調行動をする複数ロボットのための行動評価自己生成アルゴリズム, ロボティクス・メカトロニクス講演会'96, 495/496 (1996).
- 4) 大川, 柴田, 谷江: 自発的協調行動生成アルゴリズムの研究(協調のための評価生成法の提案), 第6回 インテリジェントシステムシンポジウム, 13/14 (1996).
- 5) 大川, 柴田, 谷江: 局所的な通信に基づく群の協調行動の

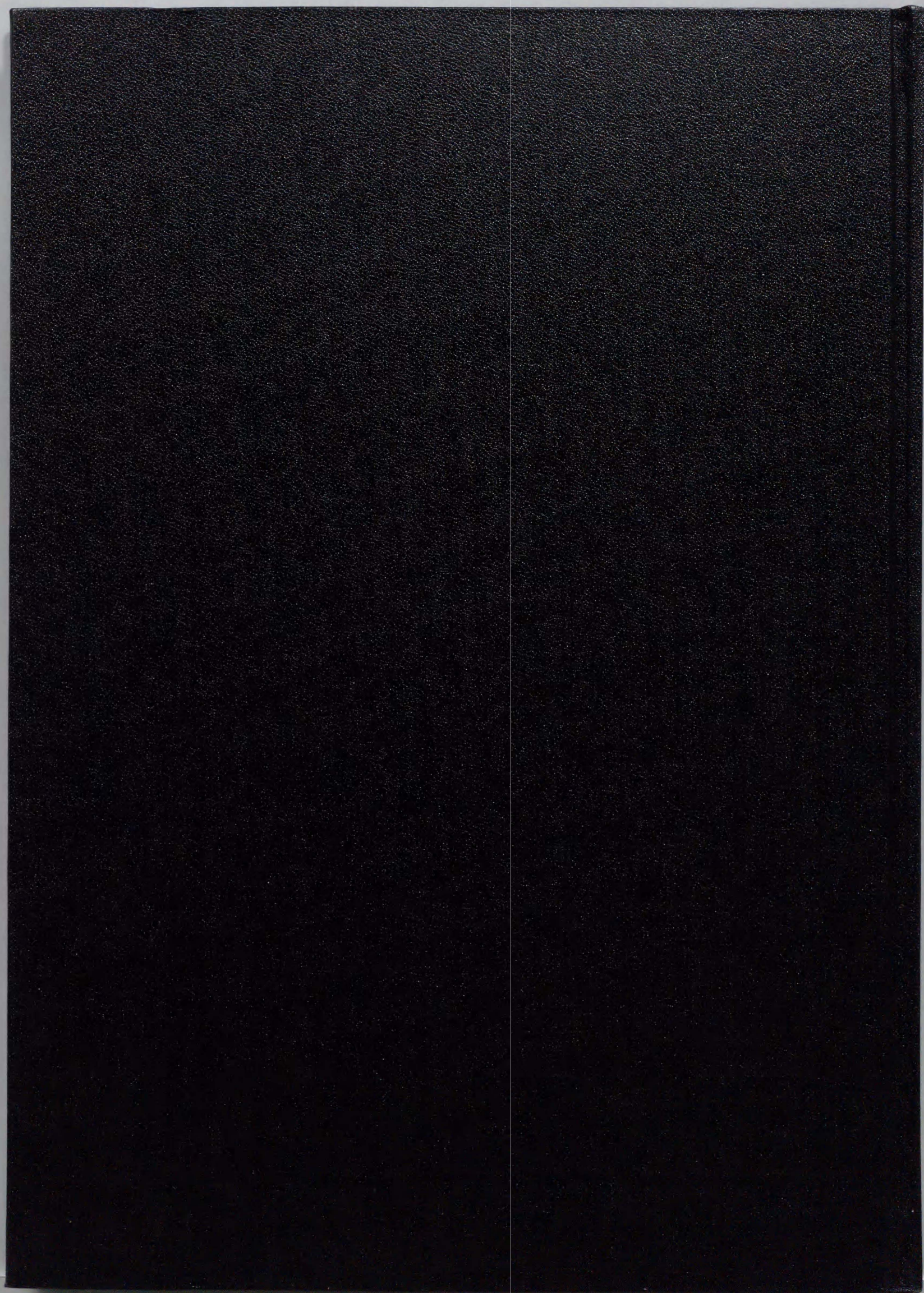
- 生成, 第 15 回 日本ロボット学会学術講演会, 859/860 (1997).
- 6) 大川, 柴田, 谷江: マクロな指令に基づく群ロボットの協調制御, ロボティクス・メカトロニクス講演会'98, 2C11-5(1-2) (1998).
- 7) 大川, 柴田, 谷江: オペレータのマクロな指示によるロボット群の自己組織化, 電気学会, 電子・情報・システム部門大会, 131/134 (1998).
- 8) 大川, 柴田, 谷江: 小集団を形成することによる群ロボットへの効果的指令伝達手法, 第 16 回 日本ロボット学会学術講演会, 581/582 (1998).

その他の論文

- 1) 柴田, 大川, 谷江, 川崎: 親子型移動群ロボットの協調行動に関する研究 (第 1 報、協調センシングアルゴリズムと障害物回避行動への応用), 日本機械学会論文集 (C 編), 1994 年 2 月 第 570 号 第 60 巻 605/611 (1994).
- 2) T.Shibata, K.Ohkawa and K.Tanie: Sensor-Based Behavior using A Neural Network for Incremental Learning in Family Mobile Robot System, Proc. of IEEE Int. Conf. on Neural Networks, Vol. V, 2825/2830 (1994).
- 3) T.Shibata, K.Ohkawa and K.Tanie: Sensor-Based Family Mobile Robot System using A Neureal Network for Incremental Learning, Proc.of Japan-USA Symp. on Factory Automation, Vol. 1, 31/38 (1994).
- 4) T.Shibata, K.Ohkawa and K.Tanie: Cooperative Sensing and Adaptive Behavior of Parent-Children Type Robot System in

Unstructured Environment, Proc. of From Perception To Action, 318/321 (1994).

- 5) T. Shibata, K. Ohkawa and K. Tanie: Cooperative Behavior of Parent-Children Type Mobile Robots, Distributed Autonomous Robotic Systems (H. Asama, T. Fukuda, T. Arai and I. Endo(Eds.)), Springer-Verlag, 322/332 (1994).



inches 1 2 3 4 5 6 7 8
cm 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

Kodak Color Control Patches

© Kodak, 2007 TM: Kodak

| Blue | Cyan | Green | Yellow | Red | Magenta | White | 3/Color | Black |
|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| [Patch] | [Patch] | [Patch] | [Patch] | [Patch] | [Patch] | [Patch] | [Patch] | [Patch] |
| [Patch] | [Patch] | [Patch] | [Patch] | [Patch] | [Patch] | [Patch] | [Patch] | [Patch] |

Kodak Gray Scale



© Kodak, 2007 TM: Kodak

A 1 2 3 4 5 6 **M** 8 9 10 11 12 13 14 15 **B** 17 18 19

