強化学習によるマルチロボットシステムの協調行動獲得に関する研究

保田, 俊行

(Degree)
博士（工学）

(Date of Degree)
2006-03-25

(Date of Publication)
2008-04-22

(Resource Type)
doctoral thesis

(Report Number)
甲3705

(URL)
https://hdl.handle.net/20.500.14094/D1003705

博士論文

強化学習によるマルチロボットシステムの協調行動獲得に関する研究

平成18年2月

神戸大学大学院自然科学研究科

保田 俊行

目 次

第1章　緒論

1.1　研究の背景

1.1.1　相互作用に基づくシステム構築

我々を取り巻く環境の変化・複雑化に伴って，従来型のモデルベース手法の限界が指摘され，“相互作用” をキーワードとしたシステムの構築手法が注目されている [1]. 例えば，動物のように様々な環境で適応的に行動する能力を『移動知』と呼び，生物学と工学の融合による構成論的アプローチでその発現メカニズムの解明を図る枠組がある．動くことで生じる身体，脳，環境の動的な相互作用によって移動知は発現されるものとされている．この枠組では，行動主体の能動的な動きを前提とすることで，非構造的で予測不能的に変動する環境に柔軟に適応するための知能の発現・設計原理に関する非経験的なモデル構築を目的としている [2]．この実現のために行動主体に求められることとして，多数の運動自由度を統合して適切に運動することや，多数の運動パターンから目的に適合したものを選択・実行する能力が挙げられている．

同様の着眼点からの研究として，身体性認知科学 (Embodied Cognitive Science)[3] がある．特に行動主体を物理的実体 (身体) を持つもの，すなわち，ロボットに限定すると認知ロボティクス [4]，または認知発達ロボティクス [5] などと呼ばれる研究の枠組がある．

1.1.2　自律型マルチロボットシステム

これらの研究領域が対象とする典型例として，マルチロボットシステム (Multi-Robot System: MRS) がある．MRS では，ロボットと環境，さらにはロボット同士の相互作用を通して，協調によってタスク処理能力が向上したり，分業によって作業効率が向上することが期待できる．すなわち，重い荷物を運ぶといったような単体では取り扱えないタスクが台数が増えることで実行可能になったり，多くのものを収集するような単体でも取り扱えるタスクを並列的に処理することで実行時間を短縮することが可能になる．さらに，個々のロボットを自律分散 [6][7] 的に配置することによって，システム全体としての拡張性・頑健性といった利点がある [8][9].

このような MRS の研究のひとつの流れとして，多数の構成要素 (モジュール) からなるモジュラーロボットによる形態制御がある [10]-[14]．このような研究分野は Swarm

Intelligence[15] と呼ばれる場合もある．単純な制御器による単純な振る舞いの組合せによって全体としていかに複雑な振る舞いを発現するかに着目しているものが多く，各ロボットは比較的簡素なものである．

別の立場として，数台のロボットによるタスクの実行を対象とした研究も行われている．ここでも単純な行動ルールを用意することで与えられたタスクを達成する研究もあるが [16]，多くの場合，個々のロボットが達成すべきタスクはモジュラーロボットと比して複雑であるためにロボットはより高度なものが求められる．現状のロボティクス分野では，ヒューマノイドロボットに代表されるようにハードウェア技術は目覚しい発展を遂げている反面，適応的な行動と協調の発現原理はほとんど解明にいたってはいない [17]．解決すべき問題点として，機能・形態をいかに設計するか，個々のロボットの自律適応性をいかに実現するかということなどが挙げられる．

1.1.3 MRS における機能分化

適応的に振る舞う MRS を構築するにあたり，日常的に協調的に振る舞う生物の社会に目を向けると，Anderson *et al.* は Team Behavior の特徴として

- タスク達成のために各個体が異なる貢献 (異なるサブタスクや役割の実行) をすること
- 各サブタスクや役割に相互依存性があること
- 組織構造が長期にわたって維持されること

を挙げている [18]．このことから，自律型 MRS におけるにおいても，様々なサブタスクや役割を達成する能力を持ち, それらを適切に設計・割り当てること，すなわち，“機能分化” することが重要であるといえる．以下に，その機能分化に必要なサブタスクや役割を設計者が事前に与えるか否かにより，MRS を非均質・均質なものに分類 (Fig. 1.1) し，それぞれの特徴を述べる．

非均質な MRS

一般的な MRS 研究では，問題を特定することでこの機能分化をあらかじめ与え，効率的な協調行動を実現している [19][20]．例えば，サッカーロボットにおけるフォワード，キーパーといった役割や，シュートやパスといった行動，さらにはそれに適した身体的構造といったものをそれぞれのロボットが独自に持っている．つまり，適切であろう機能や形態を人間の設計者が事前に設計しているため，このアプローチにおける MRS は機能・形態的に非均質なものである．ところで，MRS では，ダイナミクスが複雑であるとともにロボットの故障などの様々なシステム内部の変動の発生頻度が

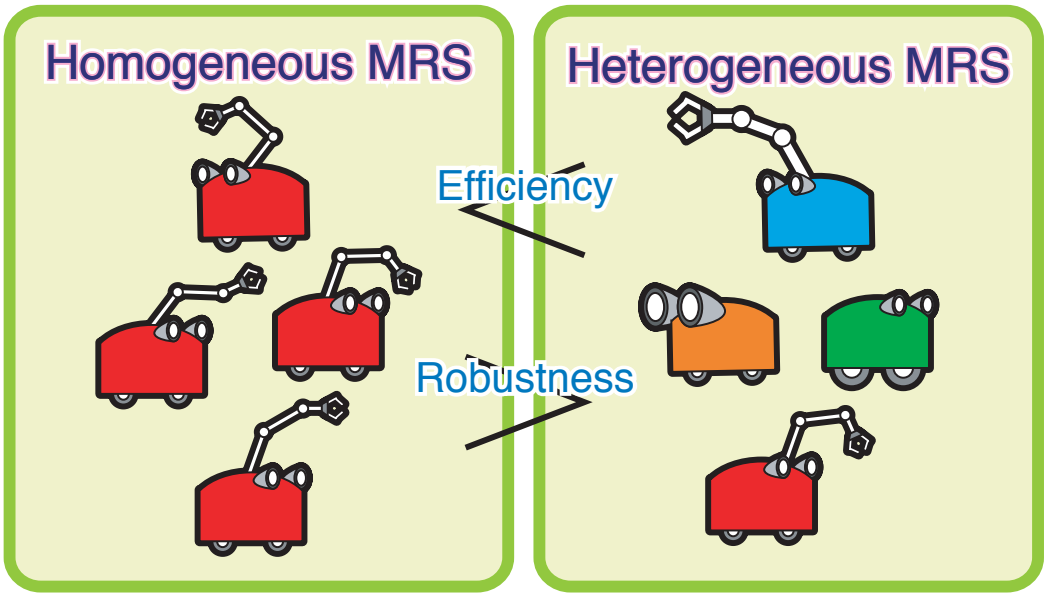


Fig. 1.1: Homogeneous and heterogeneous multi-robot system

単体の場合に比べて大きくなるという問題がある．このように問題空間が広大で相互作用が非線型で複雑である上，不確定要素が存在するために，事前に設計した機能や形態は必ずしもロボットがおかれるあらゆる状況で有効である保証はない．すなわち，非均質な MRS の有効性は設計者依存，状況依存であるといえる．

均質な MRS

非均質な MRS の対極的なアプローチとして，機能・構造が同一のロボットで構成される均質な MRS がある．均質な MRS を考えると，各ロボットはそれほど高度な役割を果たすことはできないものの，システムの自由度が高いために状況に応じた頑健な協調的振る舞いが期待される．しかし，タスク達成には機能分化の発現することが必要であるために，機能分化があらかじめ方向付けられているといえる非均質な MRS よりも個々のロボットは高い自律性を持たなければならない．そのため，このような均質な MRS に関する研究はあまり行われていない．均質な MRS を取り扱った研究においても，あらかじめ行動ルールを用意することで，それらをいかなる状況・タイミングで用いるかによって所望の機能分化を実現する研究がある [21]-[23]．この場合，非均質な MRS と同様に人間が設計したルールによってパフォーマンスが左右され，機能分化は限定的なものとなる．そのような限定的な設計者・状況依存な要素を排除するほど，ロボットの自律適応能力の実現が重要な課題となる．均質な MRS における役割とそれ必要な行動の生成をも相互作用を通して自律的に発現するという**自律的機能分化**には，最も高度な自律適応能力が求められるといえる．

1.1.4 計算知能

ロボットの相互作用に基づく自律適応能力を実現する手法として，計算知能 (Computational Intelligence)[24] に注目する．計算知能は適応・進化・学習といった生物システムが持つ能力を人工的に実現しようとするアプローチであり，ファジー理論 (Fuzzy Theorem)，人工ニューラルネットワーク (Artificial Neural Network)，進化型計算 (Evolutionary Computation)，および強化学習 (Reinforcement Learning) などのフレームワークが中心となっている．

ファジィ理論 [25] は，曖昧さを含む人間の知識を対象とするシステムに組み込むことで，不確実性を柔軟に取り扱う手法である．エキスパートの経験に基づく知識を容易に埋め込むことができ，制御対象に対する厳密な知識が不十分であっても効果のある推論・制御結果を得ることができる．ファジィ理論の工学的応用としてファジィ制御 [26] があり，制御系はセンサ系を通して得られるフィードバック信号と目標状態との差から制御量を決定する．このとき，if-then ルールを基本とするファジィルールがいくつか記述されることで成り立っており，パフォーマンスはエキスパートがいかにファジィ集合を用意し，いかにそれらの関係を記述するかに依存する．

人工ニューラルネットワーク [27] は脳の神経回路網の自己組織化の働きをモデル化したものであり，機能が比較的単純なニューロンが結合したネットワークで構成されている．結合されたニューロン間での信号伝播による相互作用によって情報処理機能を実現している．結合形態によっては時系列情報の利用や非線型写像関係の構築が可能である．しかし，学習に要する時間・収束性・ネットワーク構造によって同定可能な領域があらかじめ制限されるため，適用する問題領域毎に学習則や各種パラメータを設定しなおす必要がある．その他，ニューロン間の結合荷重の修正により学習が進行することから，学習過程・結果の解析を明示的に行うことが困難であることなどが問題である．

進化型計算は，遺伝的アルゴリズム (Genetic Algorithms: GA)[28]，進化戦略 (Evolution Strategies: ES)[29]，進化的プログラミング (Evolutionary Programming: EP)[30] の 3 つの主な領域があり，生物の進化をモデル化した最適化手法である．自律ロボットを対象とした場合，ニューラルネットワークの学習の問題点を解決するために，ニューラルネットワークの構造やニューロン間の重み付けを進化的に獲得しようとする進化ロボティクス (Evolutionary Robotics: ER) の研究が行なわれている [31]-[35]．ER は，ロボットの身体性 [36]-[38](センサ・アクチュエータの特性，位置，数，ボディのサイズなど) や，ロボットと環境の相互作用を陽に意識しなくとも，これらをコントローラの設計に反映させることができるという利点を持つ．しかし，ER ではロボットの行動制御機構をコード化した遺伝子を一個体とし，その集団を進化させて適切な制御機構を発現しようとしているため，実時間でオンラインの行動獲得は困難であり，特に計算機シミュレーションにより適切な行動獲得までに要する時間を短縮する手法が用いられている．しかし，問題毎に適切なシミュレーション・モデル設計が必要となる

ため，多くの問題に対して即座に適用可能であるとはいいがたい．しかし，その探索能力の高さから，**進化型人工ニューラルネットワーク**を制御器として持つ均質な MRS の，役割の発現とその割り当てを同時に行う自律的機能分化が，四台の光源到達問題の計算機実験 [39] や三台のフォーメーション形成問題の実機実験[1] [40] によって実現されている．

強化学習 [41]-[48] は，Q-Learning[49]，Sarsa[50]，TD-Learning[51]，Actor-Critic アルゴリズム [52]，実例に基づく強化学習法 [53] 等の手法がある．強化学習は学習主体が環境との相互作用を通して状態・行動空間の写像関係を構築・更新可能なアルゴリズムであり，オンラインでの教師なし学習により行動を獲得することが可能である．ロボットは環境や自らについての先験的知識を必要とせず，報酬と罰 (強化信号) という行動の評価に基づいて，その報酬の重み和を最大化する行動を獲得する．本来は静的な環境を対象としていることや，状態・行動空間の離散化が学習の成否に大きく影響することなどが課題である．

これらの手法は，MRS のみならずロボット単体に適用する場合においても，それぞれに課題が残されている．しかし，適切な拡張を行うことができれば，それぞれの利点を活かすことでタスク達成に必要な相互作用を生じ得ることから有効な制御器となることが期待できる．ここで，ロボット間，およびロボットと環境との相互作用に基づく手法であることは先に述べた通りである．この相互作用の過程において，適切な入出力関係を構築するまでに，ロボットは壁との衝突などといった不適切な状況に陥ることがある．これは実質的に不可避なことであるために，できる限り試行錯誤の回数は抑えることが望ましい．さらに，MRS はシステム内部の自由度が大きくダイナミクスが複雑であることや，実環境での運用におけるノイズや環境変化などの不確定要素による影響が大きいことに起因してモデルベース手法で対処することができないことも考慮すると，タスク達成に至るまでの行動の良否を詳細に評価することも困難であるといえる．以上の観点から，本論文では解の探索能力は進化型計算よりも劣るものの，教師データを必要とせずにオンラインで適切な入出力関係を構築可能である**強化学習**に着目する．

1.1.5 強化学習法によるアプローチ

概要

強化学習の概要を Fig. 1.2 に示す．強化学習は，報酬と罰という特別な入力を手がかりとして環境に適応する機械学習の一種である．強化学習は，学習に際してロボット自身や環境に関する先験的な知識を必要とせず，行動に対する評価を与えるだけで，ロボットと環境との相互作用を通して目標状態へ至る行動を獲得できる．一般に，環

[1] 計算機実験で得られた結果を実機で利用

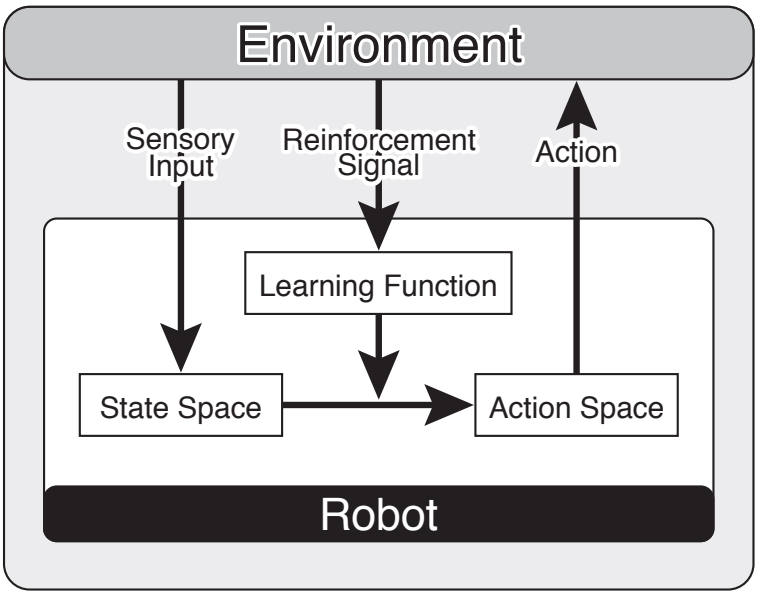


Fig. 1.2: Basic structure of reinforcement learning

境の状態変数の全てが知覚できるわけではないので，不確実性の処理が要求される．報酬は個々の行動の良否を教示するわけではなく，一連の行動の結果に対して与えられるので，遅れの処理が要求される．このような不確実性と報酬遅れを持つ弱い情報しか利用しないところに強化学習の特徴がある．

強化学習において，学習を行う主体であるロボットは各時間ステップにおいて観測される状態 (環境からの感覚入力や学習者の内部状態，あるいはそれらの組み合わせ) から行動 (出力) を決定し，実際に取った行動に対して環境から強化信号 (報酬は正の強化信号，罰は負の強化信号) が与えられる．すなわち，ロボットは，

(1) 時刻 t において環境の状態観測 x_t に応じて意思決定を行い，行動 a_t を出力する．

(2) 環境は x_{t+1} へ状態遷移し，その遷移に応じた報酬 r_t を受ける．

(3) 時刻を $t+1$ に進めて手順 (1) へ戻る．

ということを繰り返して学習していく．報酬は過去数ステップの行動系列に対して与えられる場合 (遅れ報酬型) と各時間ステップごとに与えられる場合 (逐次報酬型) がある．

学習の目的は，ある時間長さにわたる報酬の重み和 (利得) を最大化することであり，そのために状態観測から行動出力への写像関係 (政策 (policy) と呼ばれる) を獲得する．ロボットの目的を形式的に記述すると次のようになる．

$$u_t = \sum_{i=t}^{\infty} \gamma^{i-t} \cdot r_i \tag{1.1}$$

ここで，r_t は時刻 t における強化信号の大きさ，γ は $0 \leq \gamma < 1$ なる定数である報酬の割引率である．ロボットの目的は現在から未来にわたる強化信号の重み和 u_t を最大化することである．$r_t > 0$ の場合には報酬が，$r_t < 0$ の場合には罰が与えられたものと考える．$\gamma = 0$ の場合は，現在の強化信号のみに着目し未来を無視することになる．逆に $\gamma = 1$ では，どんな遠い未来でもよいから大きな報酬が得られる方が良いということになり，行動の評価は極めて長期的なものになる．すなわち，γ の値の大小によってどのくらい先の未来までを考慮するかが決まる．しかし，未来の報酬は観測できないので，一般には過去から現在までの強化信号の重み和：

$$\hat{u}_t = \sum_{i=0}^{t} \gamma^{i-t} \cdot r_i \tag{1.2}$$

を u_t の近似として利用する．

また，強化学習の学習問題は次の二点で特徴付けることができる．

- ロボットが選択すべき行動が教師から直接与えられることはなく，ロボットが実際に行った行動に対する評価という形で与えられる．
- ロボットの行動に対する評価が即座に与えられず，行動の系列に対する評価が遅れて与えられる．$\gamma = 0$ の場合は，次以降のステップにおける報酬を無視することになるため，結果的に即座に与えられる報酬 (逐次報酬) のみを考慮することになる．

強化学習システムは一般に状態に関する評価を決める部分である「学習要素」と，状態から次の行動を決定する「実行要素」に分けることができる．実行要素では，学習によって得られた状態の評価の見積もりを基に行動を決定するが，その時点の評価見積もりを最大にするような行動選択が必ずしも最適な決定となるとは限らない．なぜなら，強化学習ではロボットの経験はロボット自身の行動に強く依存するからである．学習一般に関して，経験の内容によって学習結果が大きく異なることは当然であるが，強化学習ではロボットの行動選択が経験の内容を左右する．u_t を真に最大化するには，環境に対して充分な探索を行う必要がある．

強化学習は，最終的になるべく多くの報酬を獲得するという最適性を重視した環境同定型 (exploration oriented) と，学習途中でもなるべく報酬を獲得し続けるという効率性を重視した経験強化型 (exploitation oriented) のふたつに分類できる [46]．環境同定型は，環境との相互作用がマルコフ的であれば，動的計画法の枠組みを理論的基盤として，最適な政策を決定することが可能である．しかし，最適な政策を獲得するために，多くの探索を行って環境についての知識を蓄積する必要があり学習に時間がかかる．一方，経験強化型では，学習中に報酬を受け取った行為を多く繰り返すことを目的とするため，学習が速く効率性がよい．しかし，最適な政策を得られるという保証はない．このような強化学習を実環境での制御に応用する場合，環境同定型よりも

継続的に報酬を得る行動パターンを獲得する経験強化型のほうが有効であると考えられている．

代表的な手法

強化学習の代表手法として環境同定型強化学習である Q-learning，Sarsa と Actor-Critic アルゴリズムについて述べる．次に，経験強化型強化学習の一例として Profit Sharing Plan[54]，Bucket Brigade Algorithm[55]，および実例に基づく強化学習法を取り上げて説明する．

■ **Q-learning**　Q-learning は，強化学習の中で現在最も一般的なアルゴリズムであり，C.J.C.H. Watkins によって 1989 年に提案された [49]．この手法では,「Q 関数」と呼ばれる関数によって，状態 $x \in X$ と行動 $a \in A$ の組に対する評価値「Q 値」の見積もりを導く (ここで，X，A はロボットの状態空間，行動空間とする)．ロボットは各状態において，ある探索戦略に基づいて行動を選択する．ある時刻 t における状態を x_t，t において選択された行動を a_t，遷移した状態を x_{t+1}，行動選択に伴って得られた強化信号を r_t とすると，更新するべき Q 値の更新幅は次式で更新される．

$$\Delta Q(x_t, a_t) = \alpha(r_t + \gamma \max_b Q(x_{t+1}, b) - Q(x_t, A_t)) \tag{1.3}$$

ここで α は学習係数で $0 < \alpha \leq 1$ の定数，γ は割引率である．つまり，次のステップ $t+1$ で最適と思われる行動を選択したときに得られると見込まれる評価値の見積もり $\max_b Q(x_{t+1}, b)$ を一段階だけ割り引いた値と，時刻 t で直接得られた強化信号 r_t の和に $Q(x_t, a_t)$ を近づけていく．

強化学習の動作選択においては一般に，目的を達成するのにより良い動作を見い出すため，現時点で最良の評価を持つ動作を選ぶのか (exploitation)，それとも別の動作を試すのか (exploration)，という相反する二つの戦略のトレードオフがある．行為を選択するための探索戦略としては，通常は最大の Q 値を与える行為を選択 (貪欲 (greedy) な決定戦略) するが，確率 ϵ でランダム選択を行なう戦略 (ϵ-greedy)，および確率的に選択するソフトマックス法 (ルーレット選択，Boltzmann 選択など) が用いられる．Boltzmann 分布による行為 $a \in A(i)$ の選択確率 $\Pr(a|x)(x \in S)$ は次式で表される．

$$\Pr(a|x) = \frac{\exp(Q(x,a)/T)}{\sum_{b \in A(x)} \exp(Q(x,b)/T)} \tag{1.4}$$

ここで，T は温度パラメータと呼ばれる確率性の度合を決める定数である．$T \to 0$ の極限では，Q 値を最大にする行為が選択される．

$$a = \arg\max_{b \in A(x)} Q_t(x, b) \tag{1.5}$$

これは，貪欲 (greedy) な決定戦略に相当する．Q-learning ではロボットが無限回の試行を行うことで，各状態において Q 値を最大にする行為が最適政策を形成することが証明されている．

■ **Sarsa**　Sarsa は Q-learning と似た式で Q 値を更新するため，modified Q-learning と呼ばれていたこともあった [50]．具体的には，次式によって Q 値を更新する．

$$Q(x_t, A_t) \leftarrow Q(x_t, A_t) + \alpha(r_t + \gamma Q(x_{t+1}, A_{t+1}) - Q(x_t, A_t)) \tag{1.6}$$

選択可能なルールの中で最大の Q 値に近付くように学習する Q-learning とは異なり，次状態 x_{t+1} で実際に実行されたルールの Q 値に近付くように学習を行う．このことから，Q-learning は政策オフ型 (更新に利用する Q 値は選択したものとは無関係)，Sarsa は政策オン型 (更新に利用する Q 値は実際に選択した行動に基づく) と呼ばれる．このように，Sarsa では次時刻で行う行動が必要となるため，行動選択手法が学習過程に影響を及ぼす．

Sarsa は TD-learning[51] における状態価値関数 $V(s)$ を Q 関数と置き換えたものともいえる．つまり，TD-learning に行動という概念を加えて拡張したものである．状態の価値のみを対象とする TD-learning と比べて，細やかな推定ができる．

■ **Actor-Critic アルゴリズム**　Actor-Critic アルゴリズムは，Fig. 1.3 に示すように Actor と呼ばれる制御出力器と Critic と呼ばれる評価予測器から構成されており，Policy-iteration における評価値の計算を Critic による評価値の推定に置き換え，政策改善の判定を TD-error という確率変数を用いた判定に置き換えたものと考えられる [56]．Critic は評価値 $\hat{V}(x)$ を正しく推定するように学習を行ない，Actor は $\hat{V}(x)$ を最大にするように確率的政策を学習する．学習手順は次のように記述できる．

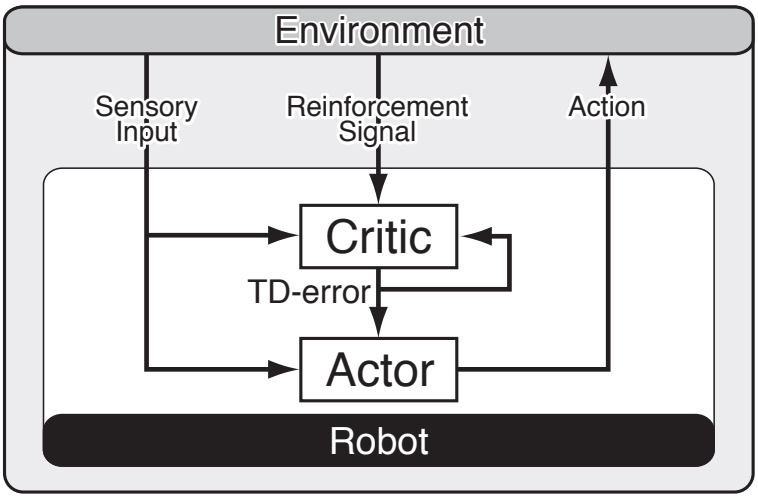


Fig. 1.3: Actor-Critic architecture

(1) ロボットが状態 x_t を観測すると，Actor は政策 $\pi(t)$ に従って行動 a_t を実行する．

(2) Critic は報酬 r_t を受け取り，次の状態 x_{t+1} を観測し，Actor への強化信号として以下の TD-error を計算する．ただし，$\gamma(0 \leq \gamma \leq 1)$ は割引率である．

$$(\text{TD-error}) = \left[r_t + \gamma \hat{V}(x_{t+1})\right] - \hat{V}(x_t) \tag{1.7}$$

(3) TD-error を用いて Actor の行動選択確率を更新する．

(a) (TD-error)>0 の場合，実行した行動 a_t の選択確率を増やす．

(b) (TD-error)<0 の場合，実行した行動 a_t の選択確率を減らす．

(4) TD 法 [41] などを用いて Critic の value の推定値を更新する．例えば TD(0) ならば，以下のように計算する．ただし，α は学習率である．

$$\hat{V}(x_t) \leftarrow \hat{V}(x_t) + \alpha(\text{TD-error}) \tag{1.8}$$

(5) 手順 (1) から繰り返す．

■ **Profit Sharing Plan (PSP)**　Profit Sharing Plan はクラシファイア・システムの枠組の中で研究されてきた手法である．政策は，if-then 形式のルール表現によって記述される．

$$rl :=< x, a, S > \tag{1.9}$$

ここで，x，a，S はそれぞれ入力，出力，および強度である．S はそのルールの評価値を表す．ロボットが状態 x_t を観測すると学習器内のルール集合から状態 x_t に最も適合したルール rl_w を選択し，rl_w に記述されている行動 a を実行する．

PSP は報酬が与えられたとき，それまでの行動系列を次式により一括して強化する．ルール rl_i の強度 S_i は次式により更新される．

$$S_i(t+1) = S_i(t) + f(n) \cdot r_t \tag{1.10}$$

ここで，$f(n)$ はルール rl_i が活性化してから時刻 t に報酬 r_t が得られるまでのステップ数 n を変数とする関数である．この関数には，$\alpha(\alpha \leq 0)$ を定数係数として一定量を与える関数 $f(n) = \alpha$，時間を遡って等比的に減少させる関数 $f(n) = \alpha^t$，等差的に減少させる関数 $f(n) = 1 - n \cdot \alpha^t$ などがある．合理的に報酬を分配するため，$f(n)$ の形についての理論的考察も行なわれている [57][58]．

PSP は一度の報酬で多くのルールを強化することができるため学習効率がよく，獲得報酬に至るまでに活性化するルール系列が長くなるような問題に適している．しかし，報酬が頻繁に与えられる場合には計算コストが高くなる．

■ **Bucket Brigade Algorithm (BBA)**　Profit Sharing Plan と同じく，クラシファイア・システムの枠組の中で研究されてきた．時刻 t における競合に加わるルールは，それぞれ強度 S から賭け値 (bit) $b_i(t) = \alpha S_i(t)$ を出す $(0 < \alpha < 1)$．競合に勝ち活性化した n 個のルールからなる集合を C とすると，C が出した賭け値 $\sum_{rl_i \in C} b_i(t)$ は，C を活性化させる出力を実行した，時刻 $t-1$ での競合の勝者に全て渡される．同様に，C も時刻 $t+1$ において，競合に勝ったルール集合 C_w から賭け値を受け取る．したがって，ルール $rl_i \in C$ の強度は，時刻 t から $t+1$ への遷移にあたり，次のように変化する．

$$S_i(t+1) = S_i(t) - b_i(t) + r_t + \sum_{rl_j \in C_w} \frac{b_j(t+1)}{n} \tag{1.11}$$

BBA は獲得された報酬が直ちに過去のルールに伝播されず，次の実行の際に一時間ステップ前のルールに伝播する．そのため，各時間ステップでかかる計算コストは小さいが，学習は遅くなる [58].

■ **実例に基づく強化学習法**　機械学習の一手法に記憶に基づく学習，あるいは実例に基づく学習と呼ばれる手法がある [59]．これらは多くのインスタンス (実例) を加工せずに記憶し，新しい入力データを与えられた時に何らかの尺度にしたがって類似度を計算し，パターン分類や関数近似を行うことを意図した手法である．

この考え方を強化学習に応用したものが，実例に基づく強化学習法 (Instance-Based Reinforcement Learning: IBRL) である [60]-[63]．IBRL は，実際に経験した入出力データをインスタンスとして記憶していく．このとき，入出力データは連続値でも離散値でも良い．そして，ある入力に対して最も類似したインスタンスを選択し，インスタンスに記憶されている動作を実行する．また，観測した入出力データに基づいて記憶しているインスタンスを変更する，あるいは各インスタンスの評価を更新し，評価の低いインスタンスを置換するといった操作を加える．この操作によって，例えば，ナビゲーション問題を考えた場合，ロボットを目標状態に導く動作系列の学習が可能になる．評価の更新には動的計画法に基づく強化学習や経験強化型の強化学習の適用が考えれる．このように実際に経験したインスタンスを記憶することで，動作ルールのオンライン生成が可能になり，また，経験した状態を中心に探索することで，学習効率の向上が期待できる．

次に，学習手順の一例を示す．インスタンスをルールとして次のように記述する．

$$rl :=< x, a, S > \tag{1.12}$$

ただし，x は入力，a は動作，S はルールの強度を表す．

(1) ロボットが状態 x_t を観測する．

(2) 各ルール rl_i の x_i と入力 x_t の類似度を任意の方法で計算する．

(a) x_t に類似し，評価の高いルール rl_i がある場合，rl_i に記述されている動作 a_i を実行する．

(b) x_t に類似し，評価の高いルールがない場合，ランダムに動作を実行する．

(3) 強化信号 r_t を受け取り，次の状態 x_{t+1} を観測する．

(4) 任意の評価関数により各ルールの強度を更新する．

(5) ルールの生成と消去によりルール集合を更新する．

(6) 手順 (1) から繰り返す．

MRS への適用における問題点

強化学習を MRS に適用するには，理論上，次に挙げる二つが原因となり学習が困難になる．一点目は，通常の強化学習はあらかじめ離散化された状態・行動空間を前提としており，学習の成否はこの離散化に大きく依存するが，この離散化に関する一般的な設計方針は現在のところ存在しないことである．二点目は，ある程度の環境中のダイナミクスを許容するために状態空間の自律的分割を行うこと [64][65]，付加情報を学習器に与えること [66] による MRS への適用に成功した研究例や，Q-learning と比べて Profit Sharing が有効性を持つことを示した研究 [67] はあるものの，本来，強化学習はマルコフ環境を前提とした手法であるので，同時に学習する他ロボットの存在に起因する動的な環境である MRS では有効性が明示的に示せないという問題である．時系列情報を扱うことによって非マルコフ環境における学習を行っている研究例 [68]-[70] があるが，そこでは状態空間の増大などが問題となる．

また，前述のように強化学習はあらかじめ離散化された状態・行動空間を前提としていることから，グリッドワールドのような環境で一台のロボット (エージェント) の計算機実験による研究が多い．しかし，実環境で作動する自律ロボットというそれぞれに異なる物理的実体を持つものを対象とした場合，計算機実験のみでは制御器の有効性を示すことはできない．それは，身体と環境との相互作用によってのみ，そのロボット自身の身体性に適した行動が獲得されるからである．その身体性に起因して，実環境では計算機実験と比べて行動獲得が大幅に困難になる上，MRS を対象とすると単体と比較しても行動獲得の困難性は飛躍的に増す．

このように，MRS の協調行動の実現には，強化学習そのものの問題だけでなく，身体性が問題となる．これらに関する一般的な設計指針は存在していない．

1.2 研究の目的

MRS では身体を通した複雑な相互作用が生じるため，あらかじめ適切な機能・形態や制御則などを与えることは困難である．しかし，システムとしては想定外の状況に陥っても停止することなく，なんらかの対処しうる必要がある．そこで，教師なしデータを必要とせずにオンラインで行動を獲得できるという特徴を持つ強化学習による均質な MRS の協調行動獲得に着目する．

通常の強化学習は静的な環境での，離散的な状態・行動空間における有効性が示されているのみである．そのため，連続で動的な環境における学習と，均質な MRS の自律的機能分化が必要とする高い自律適応能力を実現するには，(a) **連続な状態・行動空間の自律的分割**，(b) **ダイナミクス許容量 (頑健性) の増大**，および (c) **ダイナミクスの軽減**という，適切な機能拡張の必要があるといえる．さらに，ロボットのハードウェア的負担などを考慮した場合，(d) **試行錯誤の回数を削減**するべきである．

以上のような観点から，強化学習による均質な MRS における自律的機能分化に基づく頑健な協調行動の獲得を行うことを目的とする．強化学習の機能拡張を行うことで環境変動やシステム内の変動が生じても状況に応じて自律的機能分化を発現することで適応的 MRS を実現する．そして，それを通して，相互作用に基づくシステム構築の設計指針を構築することを目指す．

1.3 論文の構成

本論文は “強化学習によるマルチロボットシステムの協調行動獲得に関する研究” と題し，7 章から構成されている．第 1 章では，研究の背景と目的を説明した．

第 2 章では観点 (b) から，MRS を対象とするに先だって，ロボット単体を取り上げ，強化学習の一利用法を提案する．まず，行動獲得後に実験を続けることで振る舞いの頑健性が損なわれるという**過学習問題**を指摘する．これは，強化学習は経験に基づく入出力関係の構築手法であるために振る舞いが学習した環境に特化し，環境変動が生じて未経験の状況に陥ると振る舞いが不安定になることである．その後，学習収束後の行動をより柔軟なものにするため，強化学習で獲得した知識を確率ネットワークを用いて保存・利用するという手法を提案する．これにより，強化学習で獲得した知識を明示的表現で示すことができるだけでなく，頑健で柔軟な意思決定を行うことができる．つまり，この章における強化学習器はタスクを達成するための意思決定機構ではなく，頑健な制御器を実現するためのサンプルデータ収集のための機構として用いることになる．その有効性は，移動ロボットの光源到達問題を通して検証する．

第 3 章以降，MRS を問題対象として，自律的機能分化の発現による頑健な協調行動の獲得の実現を目指す．第 3 章では観点 (a) と (b) からの拡張法を提案する．まず，MRS が動作する環境は連続であるという観点から，通常は設計者があらかじめ離散

化した状態・行動空間の写像関係を構築する手法である強化学習を連続空間に応用するための手法を述べる．次に，この章以降で用いるオンラインで状態分割を行うベイズ判別法を用いた強化学習・BRL を紹介する．その後，その頑健性向上のための一拡張法として，ルール集合の多様性維持によるアプローチを提案する．アーム型ロボットの協調荷上げ問題に適用し，その手法の有効性を検証する．

第 4 章では観点 **(a)** と **(c)** から，MRS におけるダイナミクスを軽減することで，BRL の学習環境を安定化し，行動獲得をより効率的に行う手法について考察する．まず，MRS に強化学習を適用する際の問題点とそれに対する関連研究について言及する．その後，時系列情報に基づいて他ロボットの次時刻の状態を予測し，それを強化学習器の入力の一部として付加する手法を提案する．移動ロボットによる協調搬送問題において，その有効性を検証する．

第 5 章では観点 **(a)**，**(c)** と **(d)** から，BRL の特徴の一つである新ルールの生成における課題として，常にランダムに行動空間を探索することを指摘し，探索効率を向上させるための拡張を行う．ロボット単体の光源到達問題と MRS の協調搬送問題を通して，拡張型 BRL における学習効率を検証する．

第 6 章では観点 **(a)**，**(b)**，**(c)** と **(d)** から，環境変動後の再学習をより適応的に行うことを目指し，BRL を拡張する．具体的には，ルール発火の情報エントロピを自身の行動の安定性と捉え，状態入力に付加する．エントロピの変化量の大小により，そのままのルールを継続して用いるか新しいルールを生成するかを状況に応じて選択することができる．有効性を検証するために，協調搬送問題に適用して学習過程を解析する．

第 7 章では，本論文の結論と今後の展望を述べる．

第2章　強化学習ロボットの獲得知識の保存と利用

2.1　緒言

本章では，MRS を制御の対象とするに先だって，一台の自律ロボットの頑健な行動を実現するための一手法として，ロボットが強化学習を用いて獲得した知識を確率論の枠組を導入して保存する手法を提案する．まず，強化学習の問題点のひとつである過学習について述べる．次に，過学習による頑健性の低下に対するアプローチとして，学習知識を確率ネットワークを用いて保存・利用する手法を提案する．その後，光源到達問題を通して有効性の検証を行う．

2.2　強化学習における過学習問題

強化学習を用いることで自律ロボットの障害物回避・壁伝い・ゴール到達行動 [71]，四足ロボットの歩行動作 [72][73] などの獲得が実現されている．その一方，学習成功した後に環境が変化した場合に，通常の強化学習ではロボットの行動は不安定になるという問題点がある．例として，Fig. 2.1 のように変化する環境におけるナビゲーション問題を取り上げる．この環境において自律移動ロボットの制御器として実例に基づく強化学習法のひとつである Instance-Based Classifier Generator (IBCG)[71] を適用した場合の学習履歴の一例を示す．Fig. 2.2 は各エピソードで要したステップ数と壁に衝突した回数である．エピソードとは初期状態から終了状態までに至る系列のことであり，ステップは入出力の一サイクルを表す．ロボットはそれぞれの環境では試行錯誤を通して行動を獲得することができる．しかし，行動獲得後に環境が変化した場合，その度に再学習することで行動が不安定になっている．

このような環境変化時の行動の不安定化は，ロボットの学習は自身がおかれた環境内で進行するものであるために，獲得した知識はその環境においてのみ有効であるからである．つまり，環境変化によりロボットは未経験の状況に遭遇することになり，それまでに獲得した入出力関係を修正・再構築しなければならない．この問題は，以下の二つのいずれかが原因となり学習成功後に実験を続けるほど顕著になってシステムの頑健性は低下する．まず一点目として，状態に対する汎化能力が低下することでセンサ入力の誤差を許容できなくなることである．すなわち，少しのズレであっても異

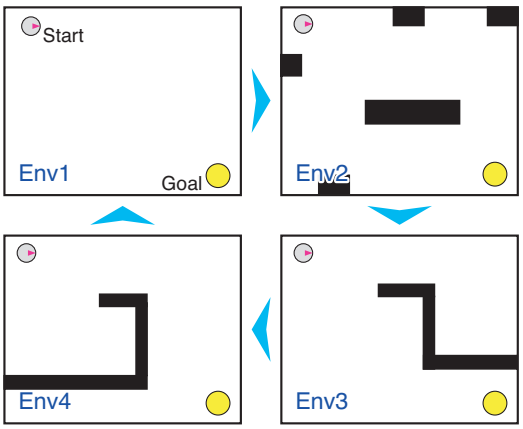


Fig. 2.1: A changeable environment

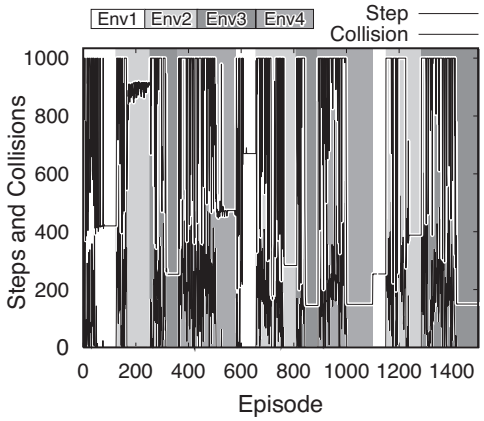


Fig. 2.2: Numbers of steps and collisions

なる状態・未経験の状態と認識するために，探索の頻度が必要以上に高くなることが問題となる．二点目は，特定のルールの発火確率が非常に高くなることで，そのルールが異なる状態であっても発火することである．これは必要な探索を阻害しているといえる．いずれが原因となるかは，適用する強化学習器の特性による．このような学習を繰り返すことによる頑健性の低下は**過学習**と呼ばれる．

本来，自律ロボットは，トップダウンな制御手法では対処できないような複雑で動的な環境におけるタスクの遂行が期待される．この点を考えると，ロボットの知識が特化したものになって振る舞いの頑健性が低下する過学習が生じることは致命的な欠点といえる．

2.3 確率ネットワークを用いた意思決定機構の構築

2.3.1 環境変化を考慮した強化学習

自律ロボットは環境が変化したときに，

- 既存のルールを利用して環境変化に対して頑健に適応する
- それができない場合は，再び学習を行って新しいルールを獲得する

ということが求められる．この実現のため，強化学習の環境変化に対する拡張に関する研究が行われている．

あらかじめ環境の変化を想定した学習に着目している研究がある．Sutton[74] の Dyna-Q はモデルベースの強化学習法であり，学習と同時に環境のモデルを用いたプランニングを行うことで環境に適応している．石井 [75] は逆温度パラメータを制御することにより exploration と exploitation のバランスを制御することで学習が収束しても常に新たな探索を行う手法を提案している．Kato ら [76] は Profit Sharing に忘却

の概念を導入したアプローチを提案している．Szita *et al.*[77] は，ϵ-MDP というモデルを提案し，環境の変化が十分に小さいならば，event-learning[78] を適用することで (準) 最適解を発見できると述べている．また，学習率が異なる学習器を二つ用意し，ひとつは実際の意思決定に用いて，もう一方はオフラインで強化値の更新のみを行うことで，それらの強化値の相関に基づいて環境変化の認識を行う手法も提案されている [79].

一方，港ら [80] はナビゲーション問題においてタスク成功率の低下によりロボットに環境の変動を認識させ，問題が起こった状態付近のみ行動政策を修正する手法を提案している．松井ら [81] はロボットが過去の環境で獲得した行動政策を新たな環境で適用するための政策事前条件を概念学習を用いて獲得し，政策を部分的に学習し直す手法を提案している．これらは，いかにして再学習を効率的に行うかに着目している．

2.3.2 設計指針

ロボットが実環境において行動獲得を実現しようした場合，ノイズや環境変化などの不確実要素を含む大規模で複雑な情報を常に取り扱う必要がある．そこで，本研究では，Q-learning のような最適性を目指した環境同定型の手法ではなく，効率性を重視した経験強化型の手法を対象とする．しかし，経験強化型強化学習では未経験な領域が多いために，部分的な再学習を行ったとしてもその前後の行動系列を学習済みの領域に適切に繋ぐような行動を獲得することが困難であるといえる．さらに，不確実性を含む環境において，学習が収束して振る舞いが安定なときに新しい探索を行うことは，システムの不安定化につながりかねない．

本研究では 2.3.1 節に記述したうち，まずは再学習ではなく，そこに至らないようにできる限り頑健な意思決定手法を実現することを目指す．しかし，環境変動が生じ得る環境を想定すると，ロボットが学習後に未経験の状態に遭遇することを避けることはできず再学習が必要となる．その場合は，部分的に行動を修正するのではなく，環境毎にタスク達成可能な意思決定機構を構築するものとする．そのような環境変化に対して頑健な意思決定機構には，センサ入力に対する汎化能力と共に，個々の環境に対してコンパクトで明示的な知識表現を実現することが重要であると考える．そこで，

- 曖昧な情報を確率的モデルで表現することで，より柔軟に推論を行う．
- 不要な部分を確率的な変動として近似することで，情報を簡略化して計算量を削減する．

という特徴を持つ確率論の枠組に着目し，意思決定機構を確率的なモデルとして構築する．ここでは，強化学習はロボットがある環境においてタスク達成可能な入出力関係を獲得するまでのサンプルデータ収集のために用いるものとし，その獲得した知識を確率的なモデルに置き換えてその後の意思決定を行うことになる．そして，環境が

大きく変化してその確率モデルでは対処でなくなったときに，そのモデルを保持したまま，新たな入出力関係の探索を強化学習により行って別の確率モデルを構築するものとする．

2.3.3 確率ネットワークの適用

確率論の手法は，近年の計算速度やデータ量の飛躍的な向上を背景に，不確実性を扱う計算モデルとして注目が集まっている．なかでも，確率変数をノードで表し，因果関係や相関関係といった依存する関係を持つ変数の間にリンクを張ったグラフ構造によるモデルである確率ネットワーク (あるいはグラフィカルモデル [82]) に着目する．確率ネットワークでは，ネットワークの構造が定まると，直接依存関係のあるノードの条件付き確率を考えることで，結合確率分布を条件付き確率を用いて分解できる．このことから，確率ネットワークは完全な結合確率よりもしばしばコンパクトは表現となる．以上のことから，確率ネットワークを用いて強化学習ロボットの獲得した知識を保存・適用することによって，確率的な意思決定により頑健性の低下を軽減するということに加えて，獲得した知識を陽に記述できるという利点がある．

確率ネットワークの中で，リンクが因果関係の方向に向きを持ち，このリンクをたどったパスが循環しないような非循環有効グラフで表されるモデルがベイジアンネットワークである [83]．ベイジアンネットワークは確率変数間の定性的な依存関係をグラフ構造によって表し，変数間の定量的な依存関係はその変数の間に定義される条件付き確率によって表すことで問題をモデル化する．このベイジアンネットワークを強化学習に適用した研究がある．北越ら [84] はロボットの観測したデータ系列と報酬から，ベイジアンネットワークを構築し，それを用いて Profit Sharing の方策の改善を行っている．山村 [85] は政策を近似するためのベイジアンネットワーク上において，確率的傾斜法を実現する適正度 [86] の伝播法を提案している．なお，これらの研究の目的は学習の効率化であるという点で，システムの頑健性の向上を目指した本研究とは異なる．

2.4 システム構成

本研究では，確率ネットワーク構築のためのサンプルデータ収集を行う行動学習機構 (強化学習器) として，2.2 節の例で用いていた IBCG[71] を適用する．また，確率ネットワークの手法としては，最も単純なベイジアンネットワークである Naive Bayes Model[87] を用いて強化学習による獲得知識の保存を行う．

提案するシステムは以下の要素から構成される (Fig. 2.3)．

- IBCG による行動獲得

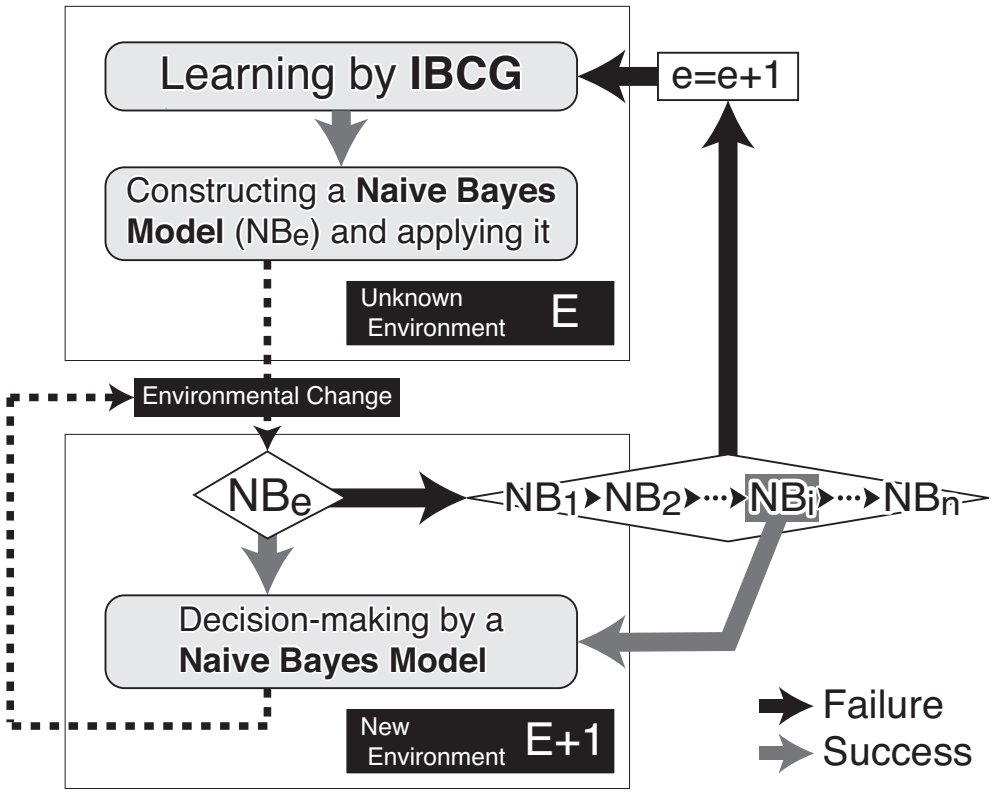


Fig. 2.3: Flowchart of the proposed system

- Naive Bayes Model を用いた学習結果の保存と利用
- 意思決定の切り替え

以下，上記の三項目についての詳細を説明する．

2.4.1 IBCG による行動獲得

IBCG の概要

まず，ロボットが扱う状態空間を，次の関係式で表される三種類の領域に分けて考える．

$$S_{\mathrm{A}}(t) \subset S_{\mathrm{V}} \subset S_{\mathrm{S}}, \quad S_{\mathrm{A}}(0) = S_{\mathrm{V}} \tag{2.1}$$

1. S_{S}：センサの数や分解能によって定まるロボットが構造上観測可能な領域.
2. S_{V}：ロボットが学習環境内で観測できる領域.
3. S_{A}：ある時間内でロボットが実際に観測した状態の集合により定義される領域.

IBCG は S_{S} の全域を対象に学習を進めるのではなく，S_{A} のみを学習の対象とする．つまり，学習システムは観測した状態に対してのみ即応的に動作を決定しながら状態空間 S_{A} を探索し，実際に経験した状態・行動の組を動作ルールとして記憶していく．動作選択の競合とルールの生成・記憶過程が統合されているため，状況と整合性が高いルールだけが記憶される．そして，環境との相互作用と信頼度割り当てによって各ルールの有効度が見積もられ，また不要なルールが削除されることにより有効度の高いルールだけが存続し S_{A} の各部を覆うルールが定まっていく (Fig. 2.4)．

IBCG の定式化

IBCG における動作ルールの集合 R は，ルール $rl \in R$ により構成され，各ルールは次式で記述される．

$$rl = \langle Cls, u, f \rangle \tag{2.2}$$

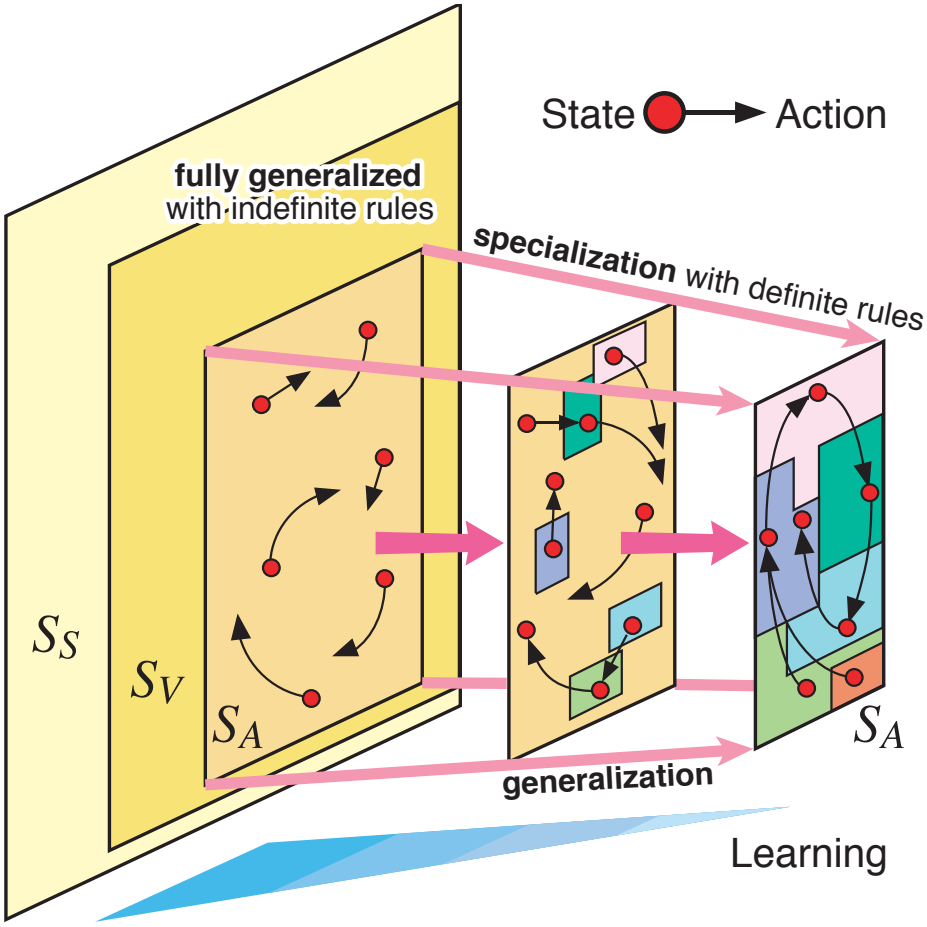


Fig. 2.4: State space structure of IBCG

Cls はクラシファイア，u は各ルールの有効度，f は信頼度割り当て関数をそれぞれ示す．クラシファイアは if-then ルールに相当し，条件部の状態ストリング Str と，行為部の動作コード a からなる．

$$\begin{aligned} Cls &= \{Str, a\} \\ & \begin{cases} Str = \{0, 1, \sharp\}^{N_S} & (\sharp : don't\ care\ symbol) \\ a \in \{a_1, a_2, \ldots, a_{N_a}\} \end{cases} \end{aligned} \tag{2.3}$$

N_s と N_a は，それぞれセンサ数と動作数を表す．集合 R は，状態ストリングの全ビットが $\sharp$ の無限定ルールの集合 R_{IND} と，特定の状態におけるセンサ入力をビット列で状態ストリングに保持している限定ルールの集合 R_{DEF} に分けられる．

$$R = \{R_{\mathrm{IND}}, R_{\mathrm{DEF}}\} \tag{2.4}$$

さらに，同じ動作コード $a_i (i = 1, \ldots, N_a)$ を保持している動作ルール rl_j をまとめ，$M(a_i)$ を設ける．

$$\begin{aligned} M(a_i) &= \{rl_1, \ldots, rl_j, \ldots, rl_{N_i(s_n)} \mid rl_j \in R_{DEF}, \\ & \qquad rl_{N_i(s_n)} \in R_{IND};\ j = 1, \ldots, N_i(s_n) - 1\} \end{aligned} \tag{2.5}$$

各モジュールには，無限定ルールが常時ひとつだけ存在する．特に初期状態では，各モジュールにはひとつの無限定ルールのみである．また，状態 s_n におけるモジュール $M(a_i)$ のルール数 $N_i(s_n)$ は，動作ルールの再生と削除により，状態ごとに変わる．

動作選択

Fig. 2.5 に，IBCG の動作選択とルール生成過程を模式的に示す．動作選択では，はじめに選択対象となる競合ルールを決める．状態 s_n におけるセンサ情報は，N_s ビットのセンサ入力ストリング $Sen = \{0, 1\}^{N_s}$ に変換される．各動作モジュール中の動作ルール $rl_j \in N(a_i)$ は，状態ストリング $Str_j = \{str_j^1, \ldots, str_j^{N_s}\}$ とセンサ入力ストリング $Sen = \{sen_1, \ldots, sen_{N_s}\}$ とのマッチ率 m_j を求める．

$$m_j = \sum_{k=1}^{N_s} str_j^k \oplus \bar{sen}_k \ /\ N_s \tag{2.6}$$

ここで，$\oplus$ は排他的論理和を表す．このマッチ率 m_j とルール rl_j の詳細度 λ_j に基づき，競合ルール R^{com} を定める．詳細度は，状態ストリング Str_j 中の $\sharp$ 記号ではないビットの割合を表す．

$$\begin{aligned} R^{com} &= \bigcup_i^{Na} R_i^{com} \\ where \qquad & R_i^{com} = \{rl^1, rl^2, \ldots, rl^n \mid n - \max_{\forall rl_j \in R_C} \lambda_j m_j u_j\} \\ & R_C = \{rl_j \in M(a_i) \mid \theta_m < m_j\} \end{aligned} \tag{2.7}$$

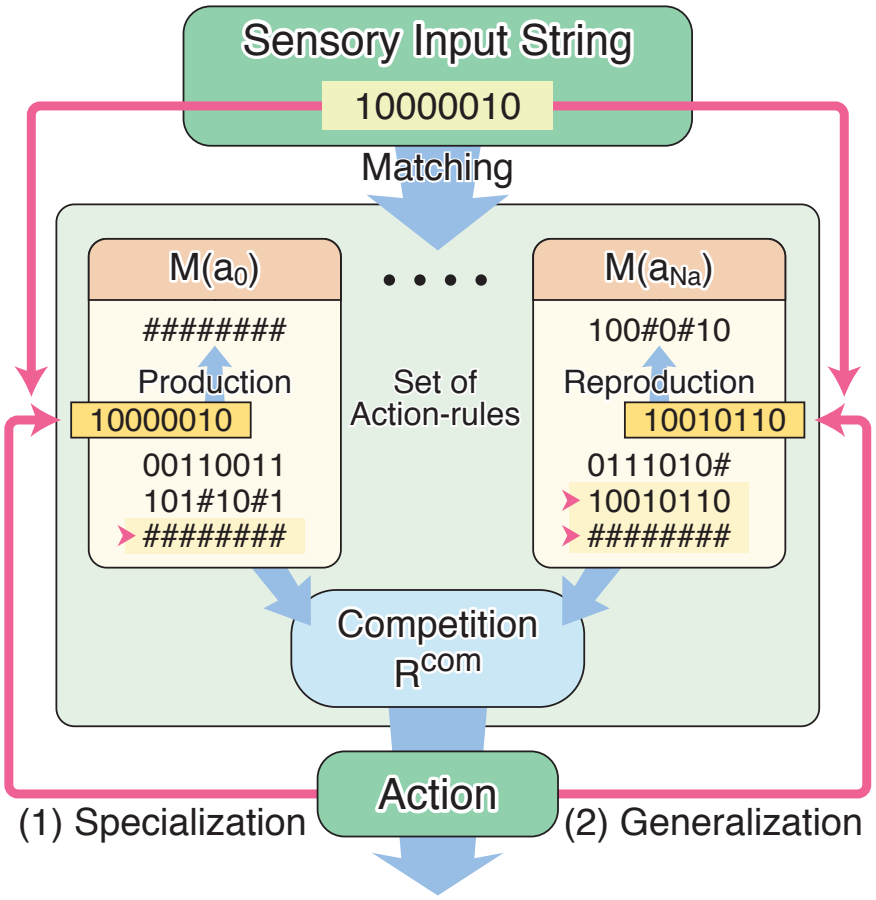


Fig. 2.5: Action selection and generation of new rules

すなわち，しきい値 θ_m を上回るマッチ率をとったルールのうち，各動作モジュールから $\lambda_j m_j u_j$ が大きい上位 n 個が競合に参加する．ただし，強化学習として罰が与えられた直後の状態では，$m_j = 1.0$ のルールだけが競合可能とする．競合に加わる限定ルールがない状態に対しても，無限定ルールは状態ストリングの全ビットが ♯ で全状態にマッチするため，競合に参加できる．競合は確率選択の形式をとる．各競合ルール $rl_j \in R^{com}$ の選択確率は，次式のボルツマン分布で与えられる．

$$\Pr(rl_j) = \frac{\exp(\lambda_j m_j u_j / T)}{\sum_{\forall rl^k \in R^{com}} \exp(\lambda_k m_k u_k / T)} \tag{2.8}$$

選択された動作ルールは，クラシファイアに記述された動作コードに対応した動作を発火する．

動作ルールの再生

IBCG ではひとつの動作ルールから新しいルールが再生される．無限定ルールが発火した場合，そのルールは発火した状態でのセンサ入力ストリングを状態ストリング

Str に記憶し，限定ルールとなる．この際，ルールの発生能力を維持するため，同じ動作モジュール内に同じ動作コードを持つ無限定ルールを再生する．

一方，状態ストリングがセンサ入力に完全には一致していなくても，限定ルールは有効度が大きければ競合に勝ち，動作を発火することができる．その場合，発火したルール rl_P は親となり，新たに汎化ルール rl_G を再生する．そして，rl_P は自己の状態ストリング $Str_P = \{str_P^1, \ldots, str_P^{N_s}\}$ のうちセンサ入力ストリングにマッチしていないビットを $\sharp$ に置き換え，別のストリング $Str_{Rep} = \{str_R^1, \ldots, str_R^{N_s}\}$ を作る．

$$\begin{aligned} for \quad k \quad &\in \quad \{1, \ldots, N_S\} \\ str_R^k \quad &= \quad \begin{cases} \sharp : & str_P^k \neq sen_k \\ str_P^k : & str_P^k = sen_k \end{cases} \end{aligned} \tag{2.9}$$

汎化ルール rl_G は，この Str_{Rep} を状態ストリングとする (Fig. 2.5(2))．したがって，rl_G は再生された状態と親ルールが記憶している状態を含む複数の状態でマッチ率 1.0 をとる．rl_G の動作は親 rl_P と同じく a_P とし，有効度は親の有効度 u_P とマッチ率 m_P から，$u_G = m_P u_P$ とする．モジュール内にすでに同じルールがある場合と，動作に対して罰が与えられた場合を除き，再生されたルールは親と同じ動作モジュールに記憶される．

信頼度割り当て機構

IBCG の状態空間は N_s 次元超立方体となる．このうち実観測領域を覆う適切なルール集合を見出すため，動作を発火した限定ルール rl_A の有効度 u_A を，信頼度割り当てとして次式で更新する．

$$u_A \leftarrow (1 - c_f)u_A + \gamma^t \cdot P \tag{2.10}$$

発火した動作列に対する評価であるペイオフ P には，有効度を増す報酬と有効度を減らす罰がある．ペイオフは分配率 γ で時間的に割り引かれながら，過去複数の状態での競合の勝者に分配される．また，デッドロックやループを避けるため，動作発火時には発火コスト $c_j u_A$ を支払う．全限定ルールは目標状態到達時にも $\eta \cdot u$ 相当の有効度を一律に消散させるため，報酬獲得に寄与しないルールは徐々にその有効度を減らす．しきい値より小さくなった場合 $(u < u^{min})$，そのルールはルール集合から削除される．

一方，連続して発火した二個のルール間で有効度が授受される．状態 s_{n+1} での競合の勝者ルールを rl'_A，その有効度を u'_A と表す．rl'_A から状態 s_n での勝者 rl_A に対して，$u_A < u'_A$ の場合に有効度の一部が受け渡される．その際の伝播量 Δu は，伝播率 α と rl_A の詳細度 l_A によって，次式で定められる．

$$u_A \quad \leftarrow \quad u_A + \Delta u \tag{2.11}$$

$$\begin{aligned} \Delta u &= \begin{cases} \delta u : & 0 < \delta u \\ 0 : & otherwise \end{cases} \\ \delta u &= \alpha \lambda_A (u'_A - u_A) \qquad (0.0 < \alpha < 1.0) \end{aligned}$$

一定期間限られたルール集合だけが発火し，それらのルール間で有効度伝播がなされる場合，各ルールの有効度は発火コスト分だけ減る一方，式 (2.11) により，そのルール集合内でもっとも高い有効度の値に近づいていく．この結果，連続した状態遷移列上で協調しあうルール集合が存続しやすくなる．

2.4.2 Naive Bayes Model を用いた獲得知識の保存と利用

Naive Bayes Model は，一つのクラス変数 (親ノード，H) と複数の属性変数 (子ノード，O) と呼ばれるノード変数からなる (Fig. 2.6)．クラス変数が与えられた場合，各属性変数は互いに条件付き独立となり，条件付き確率は以下の式で表される．

$$\Pr(O_j | H, O_k, \cdots) = \Pr(O_j | H) \tag{2.12}$$

n 個の属性変数について，$O_1 = v_1, \cdots, O_n = v_n$ という観測データが得られたときの，クラス変数についての確率分布 $\Pr(H|v_1, \cdots, v_n)$ は，

$$\begin{aligned} \Pr(H = H_i \quad | \quad v_1, \cdots, v_n) = & \\ \alpha \quad \Pr(H = H_i) \prod_j \Pr(O_j = v_j | H = H_i) & \end{aligned} \tag{2.13}$$

となり，最大の確率値をとるクラス変数を選択することで予測値を得る．この予測値に基づいてロボットの行動を決定する．

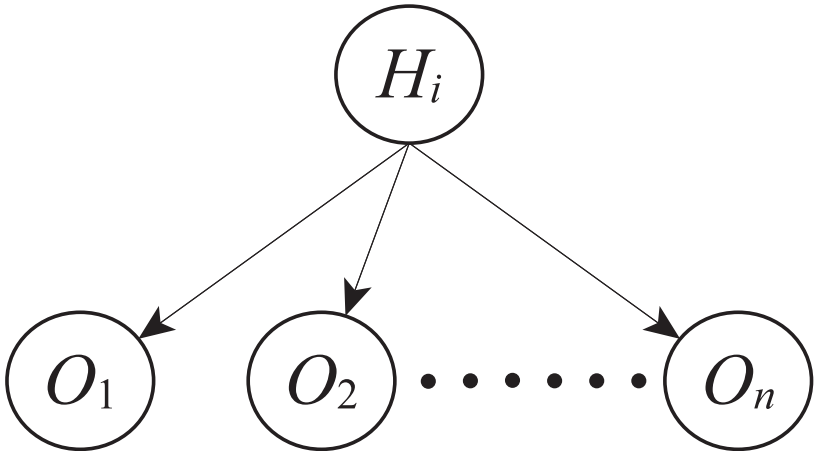


Fig. 2.6: An example of a naive Bayes model

2.4.3　意思決定の切り替え

ロボットは未知環境に直面した場合は，IBCG を用いて行動獲得を試みる．行動が収束すると，それまでに得られたセンサ入力-行動出力のデータをサンプルデータとして Naive Bayes Model を構築する．その後は Naive Bayes Model を用いてロボットの行動決定を行う．環境が変化した時は，まずそれまでの環境で利用していた Naive Bayes Model を利用する．タスクがそのまま達成できた場合は，新しい環境でも同じ Naive Bayes Model を利用して意思決定を行う．タスクに失敗した場合に複数の Naive Bayes Model を保持していれば，他の Naive Bayes Model を利用して新しい環境下で意思決定を行う．タスク達成可能な Naive Bayes Model を保持していれば，それを利用するものとする．全ての Naive Bayes Model について，タスクを達成できなければ，全くの未知環境に直面したとして，再び IBCG を用いた学習を行い，行動収束後にその環境に適した新しい Naive Bayes Model を構築する．

2.5　光源到達問題による検証

2.5.1　問題設定

Fig. 2.7 に示す小型自律移動ロボット Khepera による光源到達問題を通して，提案手法の有効性を検証する．Khepera の車体は直径 55 mm であり，八個の赤外線センサにより環境を識別し，左右の車輪により動作を実行する．赤外線センサは距離センサと光センサの機能を持つために入力は 16 次元であり，それぞれのセンサの分解能は

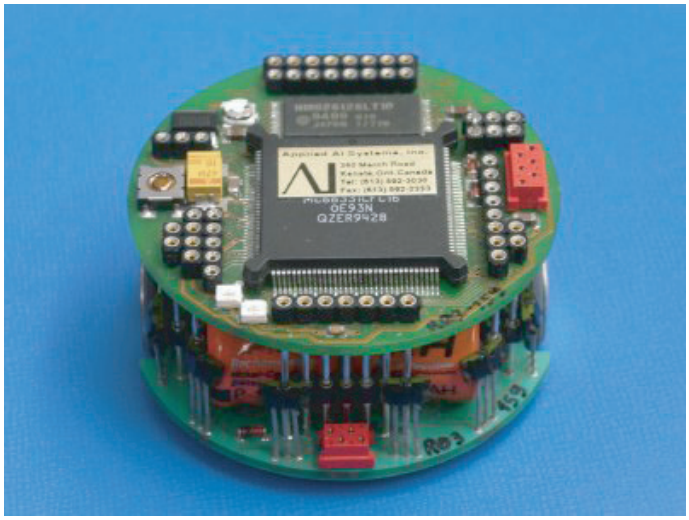


Fig.　2.7: Khepera robot

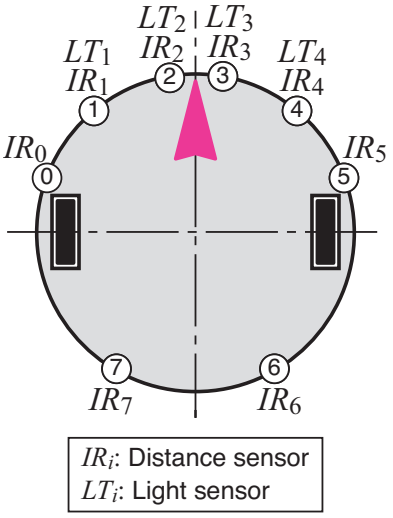
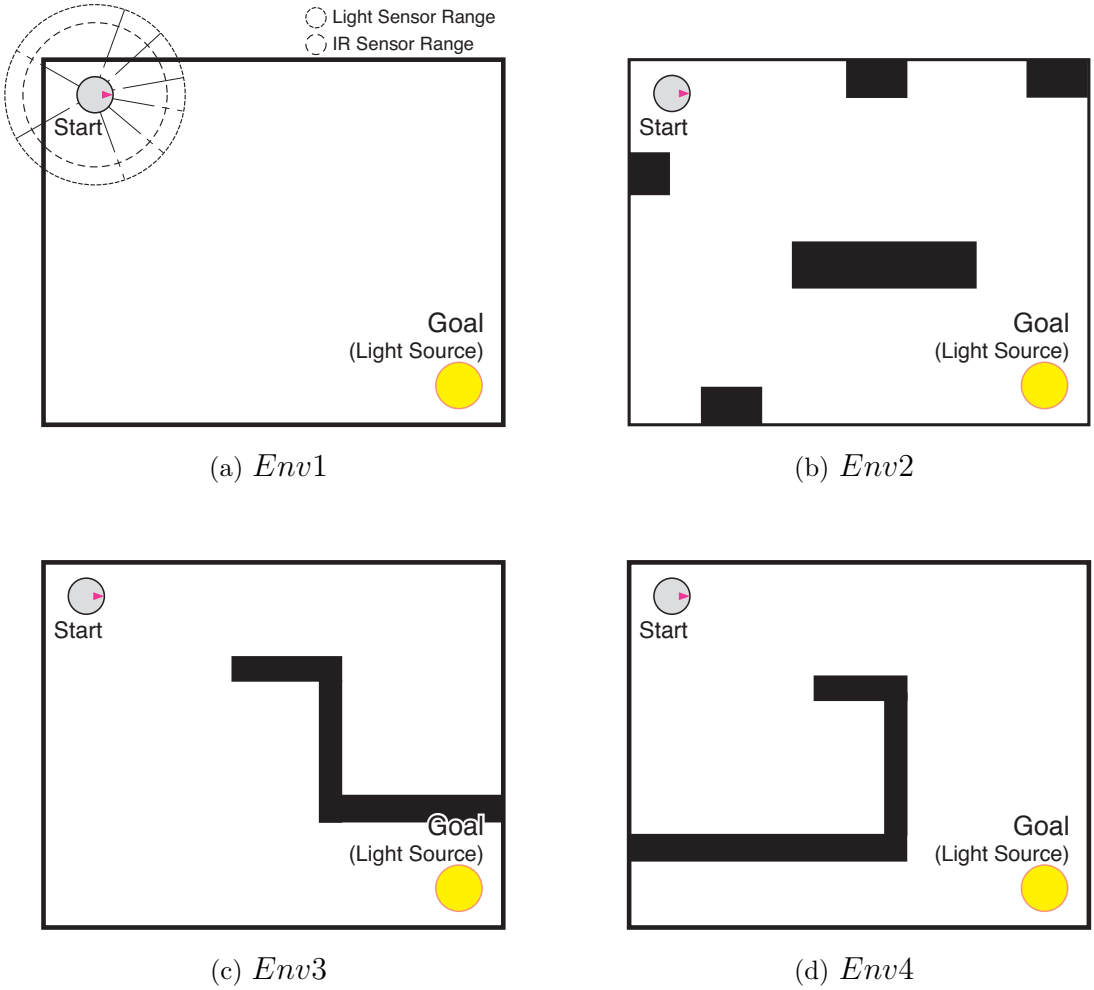


Fig.　2.8: Sensor and motor layout

1024 と 512 である．本実験では，距離センサを八個 (IR_0, ..., IR_7)，光センサを四個 (LT_1, ..., LT_4) を用い，センサ値はしきい値を設定することで二値化する．センサの配置は Fig. 2.8 の通りである．出力は独立して駆動する各車輪の回転速度であり，速度の最大値と最小値はそれぞれ 15°/sec と-7°/sec である．

まず，計算機実験により提案手法の有効性を検証する．ここでは，ロボットは Fig. 2.9 に示すような壁の配置が異なる四通りの環境 ($Env1$，$Env2$，$Env3$，$Env4$) におかれるものとする．なお，Fig. 2.9(a) におけるロボット周囲の円はセンサの有効範囲を示している．ロボットが学習収束の判定条件を満たすようにタスクを達成するたびに，環境は $Env1 \rightarrow Env2 \rightarrow Env3 \rightarrow Env4 \rightarrow Env1 \rightarrow \cdots$ の順番に変化する．実機実験では，Fig. 2.10(a) に示す環境 $EnvA$ から Fig. 2.10(b) の環境 $EnvB$ に変化した場合のシステムの挙動を観察する．環境の大きさは 400mm × 280mm である．



(a) $Env1$　(b) $Env2$

(c) $Env3$　(d) $Env4$

Fig. 2.9: Experimental environments for simulations

(a) $EnvA$ (b) $EnvB$

Fig. 2.10: Experimental environments for Khepera

2.5.2 IBCG の設定

IBCG の状態入力ストリング Str は Fig. 2.8 に示した八個の距離センサ・IR と四個の光センサ・LT を用いて，これらの入力を $\{0, 1\}$ の二進数値に変換した計 12 個のビット列から構成される．出力 a にはあらかじめ離散化された五通りの行動 $a_0, ..., a_4$(a_0：前進，a_1：右旋回，a_2：左旋回，a_3：右回転，a_4：左回転) を用いる．以上より，IBCG におけるクラシファイア CLS は次式で表される．

$$CLS = \{Str, a\} \tag{2.14}$$
$$\begin{cases} Str = \{IR_0, ..., IR_7, LT_1, ..., LT_4\} \\ a \in \{a_0, ..., a_4\} \end{cases}$$

報酬はロボットがゴールした時に与えられ，罰は壁に衝突した時に与えられる．センサからの情報を得て，行動を選択し，評価を得るまでの試行を 1 ステップとして，ロボットがゴールに到達するか，1000 ステップの試行を経過した時にエピソードを更新する．学習収束を判定する条件は，連続して 20 エピソードでゴールしたときのゴール到達までのステップ数の平均，分散がそれぞれ 400，20 以下のときとする．IBCG の諸パラメータを Table 2.1 に示す．

2.5.3 Naive Bayes Model の設定

確率分布の設定

本実験で用いる Naive Bayes Model の基本構造として，クラス変数に行動出力 a，属性変数にそれぞれのセンサ入力 $Str = \{IR_0, \cdots, IR_7, LT_1, \cdots, LT_4\}$ を持つ Fig. 2.11 に示す構造を用いる．このようにモデルを設定することで，各センサを条件付き独立と仮定して計算量を削減できる．

Table. 2.1: IBCG parameters

Parameter		Value
n_{max}	maximum size of the rules	200
P	payoff(reward)	20.0
P	payoff(penalty)	-0.05 u
u_0	initial utility	10.0
c_f	cost for an action	0.01
γ	distribution rate of utility	0.8
κ	utility spread rate	0.25
η	evaporation rate	0.99
λ_{IND}	indefinite rule's spec	0.01
T	temprature of boltzman distribution	3.0

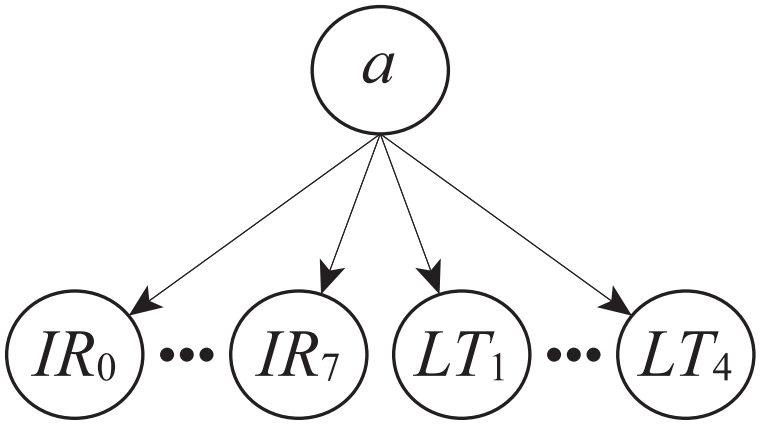


Fig. 2.11: Naive Bayes model of the robot

各ノードについてそれぞれ確率分布を決定する．ここで，決定しなければならないのは，クラス変数の確率分布 $\Pr(a)$ と a ノードを親とする各センサについての条件付き確率 $\Pr(IR_0|a)$,...,$\Pr(LT_4|a)$ である．ここで，事前確率分布について，(i) 強化学習によりどのような行動が生成されるか予測することが困難，(ii) 学習により得られたデータによる情報量の最大限活用，という観点から，無情報事前分布 (一様分布) とする．実際には，学習収束後の各ステップでのセンサ入力と行動出力について，生起回数により，確率分布を決定する．Naive Bayes Model が規定されると，ロボットはセンサ入力に対して次式により得られる $\Pr(a|Str)$ を最大化する行動を実行する．

$$\Pr(a|Str) = \alpha \Pr(a) \prod_j (Str_j|a) \tag{2.15}$$

ここで，Str_j は j 番目のセンサ入力を表すビットである．

各センサと a ノード間の関係

各センサと行動のノード間の相関関係について，χ^2 検定を用いて検証する．センサは計 12 個で各センサの入力は $\{0,1\}$ の二通り，それに対して行動は五通りであるので，Table 2.2 に示すような 2×5 分割表が 12 個作成される．ここで，n_{ij} はデータから得られたそのカテゴリに属するサンプル数であり，n_{+j}，n_{i+} は各行列ごとの合計，n は全サンプル数であり，

$$n_{i+}=\sum_{j=0}^{4}n_{ij}, \tag{2.16}$$

$$n_{+j}=\sum_{i=0}^{1}n_{ij}, \tag{2.17}$$

$$n=\sum_{i=0}^{1}\sum_{j=0}^{4}n_{ij}, \tag{2.18}$$

で表される．次に，全てのカテゴリについて，下式から期待度数 m_{ij} を計算する．

$$m_{ij}=\frac{n_{i+}\times n_{+j}}{n}. \tag{2.19}$$

そして，以上の n_{ij}，m_{ij} を用いて下式から χ_0^2 を計算する．

$$\chi_0^2=\sum_{i=0}^{1}\sum_{j=0}^{4}\frac{(n_{ij}-m_{ij})^2}{m_{ij}} \tag{2.20}$$

有意水準 $\alpha=0.05$，自由度 $f=4$ として，χ^2 分布表から得られる値 χ^2 と比較して，$\chi_0^2\geq\chi^2$ ならば，そのセンサと行動の間にはなんらかの関係が存在するといえる．一方，この不等式を満たさない場合は，そのセンサは行動にあまり影響を与えないとして，Fig. 2.11 から，該当するノードを削除する．これによって，より簡潔な Naive Bayes Model を構築する．

Table. 2.2: 2×5 contingency table (conditional probability table)

		Action					Sum
		a_0	a_1	a_2	a_3	a_4	
Sensor Input	0	n_{00}	n_{01}	n_{02}	n_{03}	n_{04}	n_{0+}
	1	n_{10}	n_{11}	n_{12}	n_{13}	n_{14}	n_{1+}
Sum		n_{+0}	n_{+1}	n_{+2}	n_{+3}	n_{+4}	n

2.5.4 計算機実験

学習履歴

IBCG は実例に基づく強化学習の一手法であり，最適解ではなく実行可能解を効率的に発見することを目指しているため，実験毎に獲得する行動は異なる．ここでは，実験結果の一例として，代表的なものについて述べる．

各エピソードでのゴール到達までに要したステップ数と壁や障害物との衝突回数を表したグラフを Fig. 2.12 に示す．各環境下でのロボットの取った行動は以下のようになっている．

■ 一巡目

*Env*1：最初ロボットは，センサ入力-行動出力に関して何も知識を持っていないので IBCG を通して，行動を獲得していく．そして，行動が収束すると，各ステップでのセンサ入力-行動出力のデータを用いて，Naive Bayes Model (NB1) を構築し，以降は NB1 を用いて意思決定を行う．

*Env*2：NB1 でそのままゴールに到達できたため，引続き NB1 を用いて意思決定を行う．

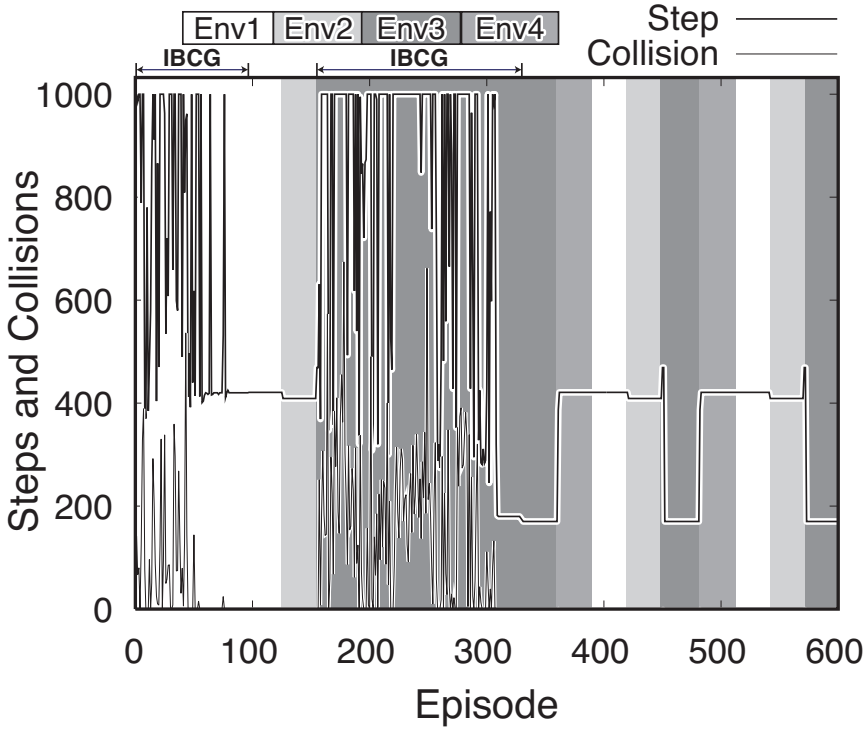


Fig. 2.12: Number of steps and collisions

*Env*3：まずは NB1 を適用したが，ゴールには到達できなかった．このとき，ロボットは他に Naive Bayes Model を持っていないため，再び IBCG を用いて行動を獲得する．その後，その行動を基に新たな Naive Bayes Model (NB2) を構築し，意思決定を行う．

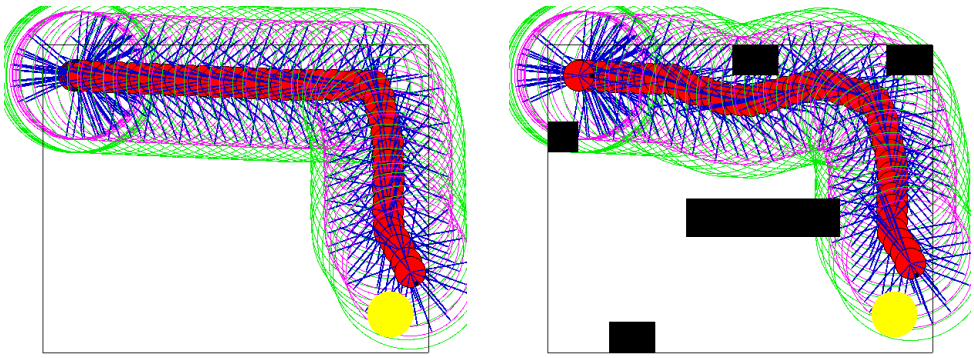
*Env*4：まずは NB2 を適用したが，ゴールに到達できなかった．NB1 に変更するとゴールに到達できたので，NB1 を用いて意思決定を行う．

■ 二巡目以降

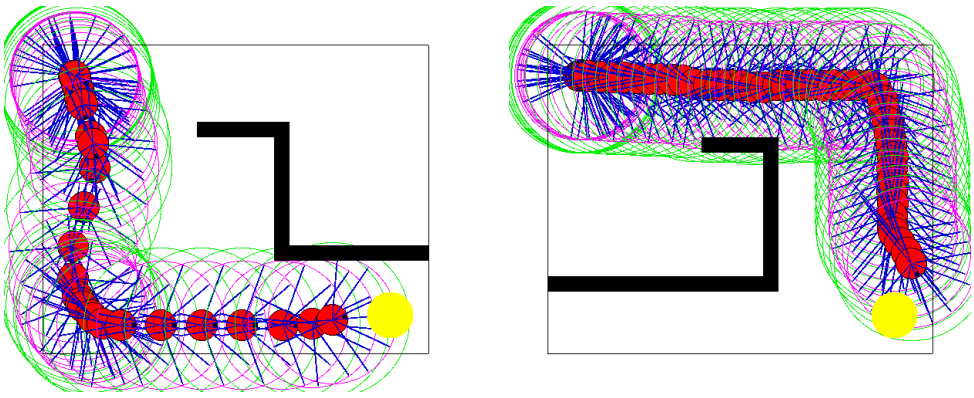
*Env*1, *Env*2, *Env*4：NB1 を用いる．

*Env*3：NB2 を用いる．

各環境下で，Naive Bayes Model による意思決定を行っているロボットのゴール到達までの軌跡を Fig. 2.13 に示す．NB1 の場合 (*Env*1，*Env*2，*Env*4) は右側の IR センサを用いて壁伝い行動を行い，NB2 の場合 (*Env*3) は右側を用いて壁伝い行動を行っている．



(a) NB1 for *Env*1　　(b) NB1 for *Env*2



(c) NB2 for *Env*3　　(d) NB1 for *Env*4

Fig. 2.13: Locus chart of the robot with naive Bayes models

IBCG による行動獲得

本実験では，環境の一巡目の $Env1$ と $Env3$ において，IBCG による行動獲得が行われた．ここでは，それぞれの学習過程について述べる．

Fig. 2.14 に IBCG による学習過程での各エピソードにおいて，生成したルール数とエピソード終了時点でロボットが持っていたルール数の推移を示す．途中で線が途切れているのは，そのエピソードは IBCG ではなく，Naive Bayes Model を用いて意思決定を行っているためである．$Env1$ と $Env3$ でのルール数の変化はそれぞれ図中の 1-96 エピソード，および 154-329 エピソードのものである．学習初期段階では，ルールの増減が激しい．収束段階で連続してゴールしている時は，新たなルールは作成されおらず，ルール数は減少していることがわかる．次に，各環境におけるロボットの振る舞いと Fig. 2.15 と Fig. 2.16 に獲得した行動を示す．Fig. 2.15(a) と Fig. 2.16(a) は学習収束後のロボットの軌跡，Fig. 2.15(b) と Fig. 2.16(b) はルールの発火系列である．ロボットは $Env1$ では右側，$Env3$ では左側の IR センサで壁を感知するようにして壁沿いを移動する行動を獲得した．このとき，$Env1$ と $Env3$ においてロボットが保持しているルール集合は Table 2.3，2.4 である．表のルール番号に下線が付いているものは，行動獲得時に発火しているルールである．

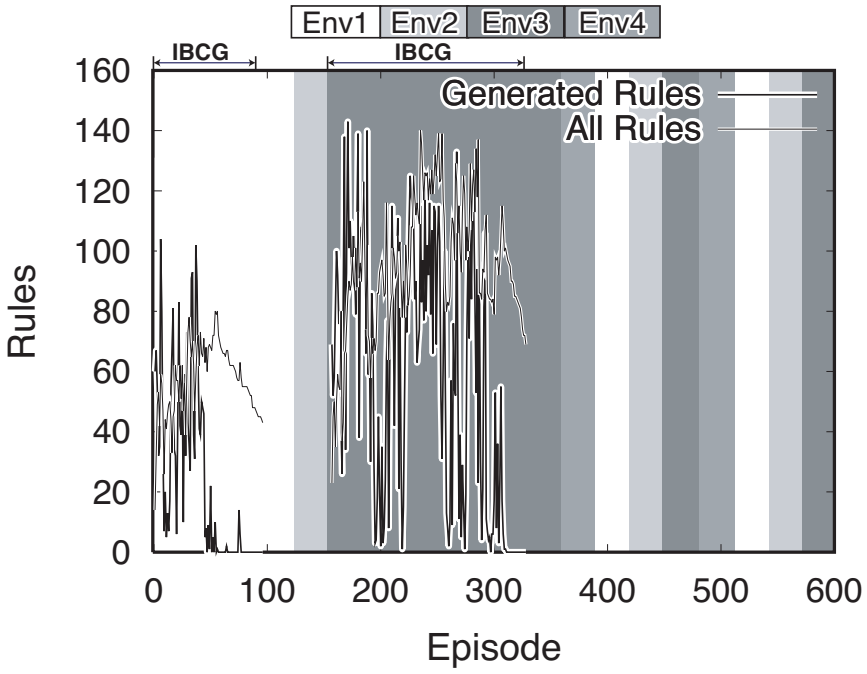
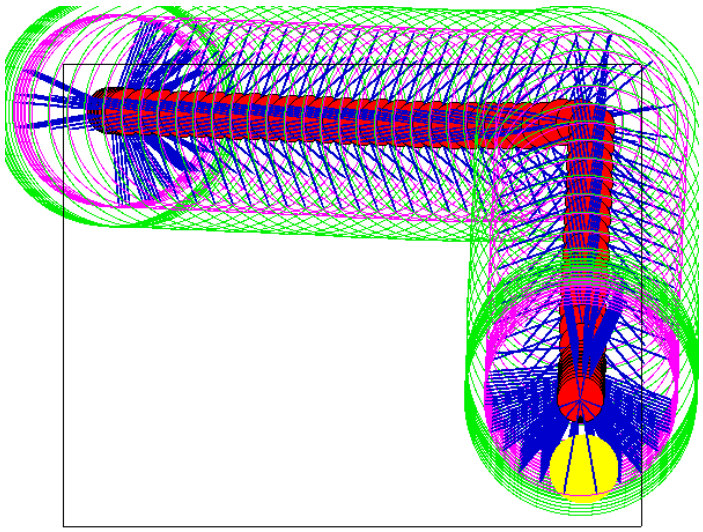
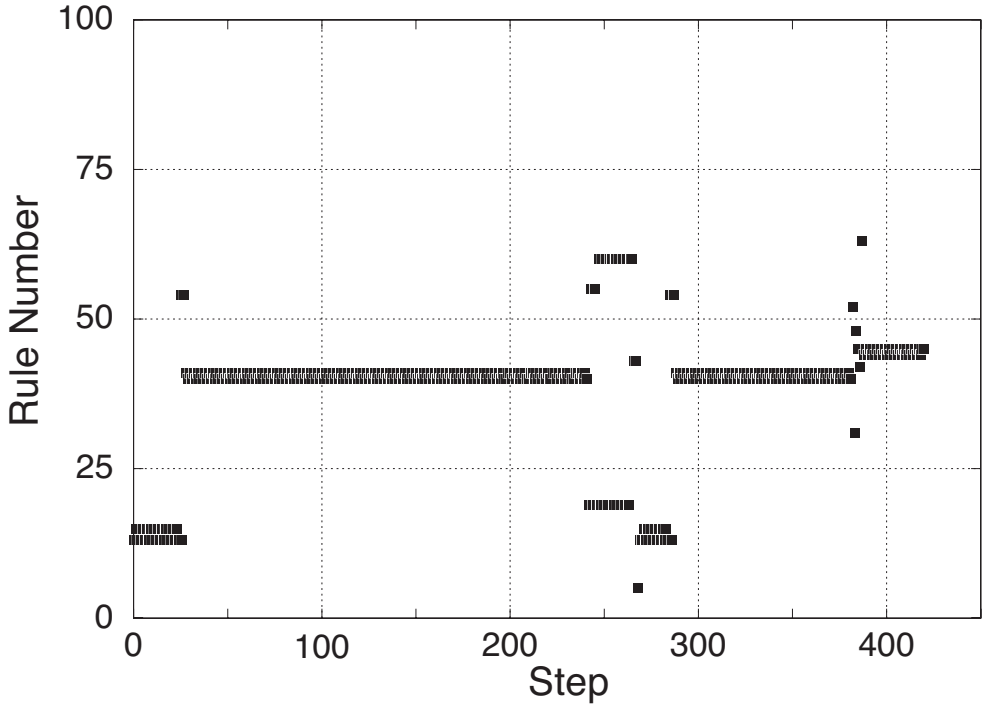


Fig. 2.14: Number of rules for IBCG

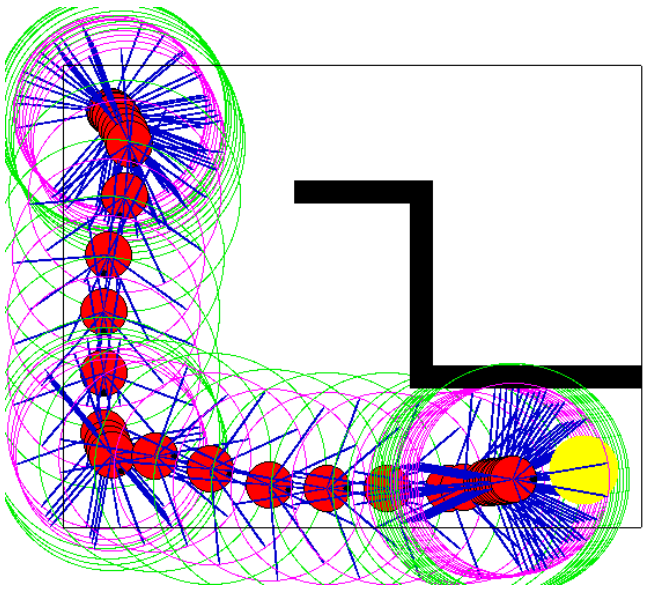


(a) Traces of behavior

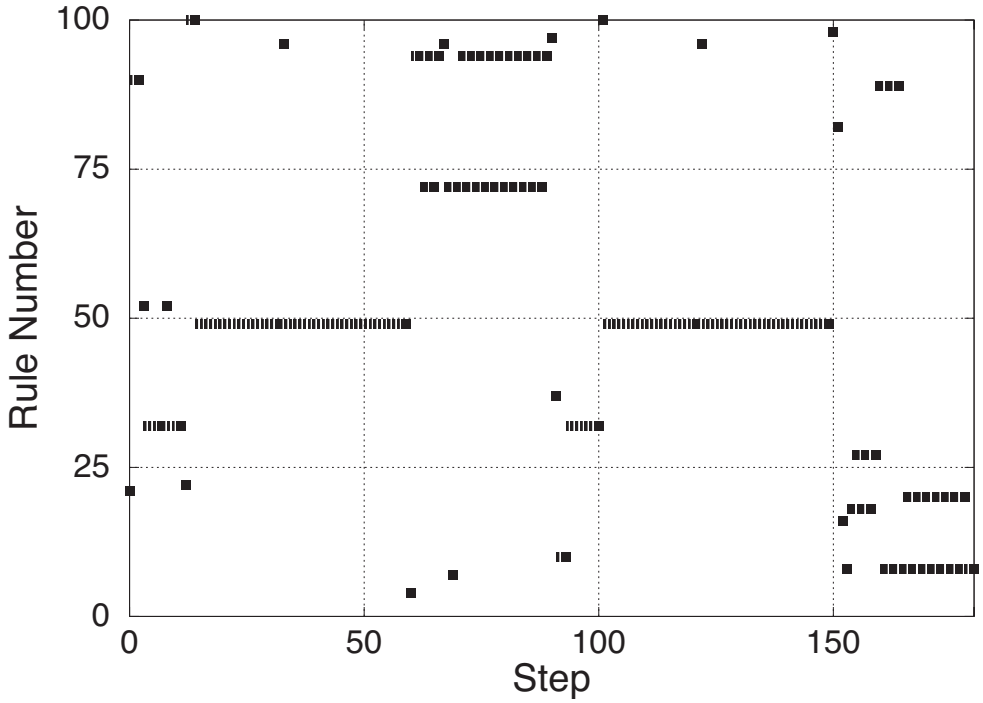


(b) Sequence of firing rules

Fig. 2.15: Acquired behavior of the robot with IBCG in $Env1$



(a) Traces of behavior



(b) Sequence of firing rules

Fig. 2.16: Acquired behavior of the robot with IBCG in $Env3$

Table. 2.3: Rule set of IBCG for $Env1$

Rule No.	Str	a	Rule No.	Str	a
5	111000110000	a_3	8	110000010000	a_1
9	111000001111	a_3	10	000011101111	a_0
13	110000110000	a_3	15	100000110000	a_2
16	100000011000	a_2	17	110000001110	a_4
19	111100000000	a_2	21	000000111000	a_2
22	110000011111	a_3	27	000001001111	a_2
29	100000111111	a_4	30	101000000000	a_4
31	110000000010	a_0	34	000000000111	a_3
35	000000001110	a_2	37	000000000000	a_2
39	100000011111	a_4	40	100000000000	a_2
41	110000000000	a_3	42	100000001110	a_3
43	111100010000	a_1	44	100000001111	a_2
45	110000001111	a_3	46	101100000000	a_1
47	011000000000	a_0	48	110000000111	a_1
50	111000000000	a_1	51	100000010000	a_2
52	110000000110	a_4	54	100000100000	a_2
55	110100000000	a_3	60	111110000000	a_3
63	100000011110	a_4	64	111000010000	a_0
66	111000000011	a_1	71	000000100111	a_1
80	100000000100	a_3	81	000000001000	a_2
82	000000001100	a_1	83	000000011100	a_0

Table. 2.4: Rule set of IBCG for $Env3$

Rule No.	Str	a	Rule No.	Str	a
1	111001000000	a_0	3	000001100111	a_0
4	001011000000	a_3	5	010001000000	a_2
6	001000100000	a_1	7	000101000000	a_2
8	000011001111	a_2	10	000011110000	a_2
14	100011000000	a_4	15	100001000000	a_2
16	000111001110	a_4	18	000011101111	a_4
19	110001100011	a_3	20	100001001111	a_1
21	110000110000	a_3	22	000001010000	a_0
24	110000000000	a_3	25	111000000000	a_1
26	000001111111	a_0	27	000001101111	a_3
31	100000011110	a_1	32	000001110000	a_0
35	111000000001	a_1	36	110011000000	a_1
37	000111110000	a_4	39	100001000110	a_2
40	110001000110	a_3	41	100011000100	a_0
42	110001000010	a_4	44	110001001111	a_3
45	000001100000	a_1	46	110000001111	a_3
47	001001100000	a_3	49	000011000000	a_0
50	100011000110	a_0	52	000000110000	a_3
53	100000011111	a_0	55	100011011111	a_2
58	100001011111	a_2	63	001000000000	a_3
64	010000000000	a_1	70	000011000100	a_0
72	000111100000	a_3	73	100001000111	a_3
74	100011001111	a_2	79	000111001100	a_0
81	100011001110	a_4	82	000011001110	a_1
84	100010000000	a_0	89	000001001111	a_3
90	100000110000	a_1	91	001001000000	a_1
94	001111000000	a_4	95	110000000111	a_2
96	000111000000	a_2	97	001111100000	a_4
98	000011000110	a_0	99	110001000000	a_0
100	000001000000	a_1	103	001110000000	a_2
105	011000000000	a_3	106	110000000011	a_1
108	100001001110	a_2	109	000011100000	a_3
111	001111001100	a_2	123	110001000111	a_3
138	000111001000	a_2	154	000011011110	a_2

構築したNaive Bayes Model

本実験ではロボットが$Env1$と$Env3$で獲得したIBCGの学習結果を基にNB1，NB2という二個のNaive Bayes Modelを構築した．以下，それぞれについて述べる．

■ **NB1**　NB1は$Env1$下でIBCGによる学習が収束した後の各ステップでのセンサ入力と行動出力をサンプルデータとして構築したモデルである．親ノードの確率分布(Table 2.5)と各センサについての条件付き確率を表す分割表(Table 2.6)をサンプルデータから作成し，χ^2検定を行った．その結果，Fig. 2.11からIR_0，IR_5，およびIR_6ノードを取り除いた構造のモデルが得られた．NB1は$Env2$と$Env4$においても，ロボットが主に右側に壁を感知するように壁沿いを移動する行動を取る際にも有効に用いられている．

Table. 2.5: Probability of actions for $Env1$

$\Pr(a_0)$	$\Pr(a_1)$	$\Pr(a_2)$	$\Pr(a_3)$	$\Pr(a_4)$
0.003	0.008	0.483	0.502	0.005

Table. 2.6: Conditional probability table for $Env1$

		$\Pr(\cdot\|a_0)$	$\Pr(\cdot\|a_1)$	$\Pr(\cdot\|a_2)$	$\Pr(\cdot\|a_3)$	$\Pr(\cdot\|a_4)$
IR_1	0	0.143	0.059	0.945	0.015	0.500
	1	0.857	0.941	0.055	0.985	0.500
IR_2	0	0.857	0.353	0.945	0.943	0.917
	1	0.143	0.647	0.055	0.057	0.083
IR_3	0	0.857	0.353	0.945	0.938	0.917
	1	0.143	0.647	0.055	0.062	0.083
IR_4	0	0.857	0.941	0.999	0.947	0.917
	1	0.143	0.059	0.001	0.053	0.083
IR_7	0	0.857	0.353	0.906	0.881	0.500
	1	0.143	0.647	0.094	0.119	0.500
LT_1	0	0.857	0.941	0.921	0.905	0.500
	1	0.143	0.059	0.079	0.095	0.500
LT_2	0	0.857	0.647	0.921	0.905	0.083
	1	0.143	0.353	0.079	0.095	0.917
LT_3	0	0.143	0.647	0.921	0.905	0.083
	1	0.857	0.353	0.079	0.095	0.917
LT_4	0	0.857	0.647	0.921	0.910	0.917
	1	0.143	0.353	0.079	0.090	0.083

■ **NB2**　NB2 は $Env3$ において構築したモデルである．親ノードの確率分布 (Table 2.7) と各センサについての条件付き確率を表す分割表 (Table 2.8) を作成し，χ^2 検定を行った結果，Fig. 2.11 から IR_7 ノードを取り除いた構造の Naive Bayes Model が作成された．

以上から，環境やそのときの学習結果から異なる構造の Naive Bayes Model が構築されたことが確認された．また，NB1 を用いて複数の環境に適応できていることから，構築されたモデルはその環境特有ではなく，その他の環境においても適用し得る，頑健で汎用性を持ったモデルである可能性が示された．

Table. 2.7: Probability of actions for $Env3$

$\Pr(a_0)$	$\Pr(a_1)$	$\Pr(a_2)$	$\Pr(a_3)$	$\Pr(a_4)$
0.586	0.072	0.105	0.127	0.110

Table. 2.8: Conditional probability table for $Env3$

		$\Pr(\cdot\|a_0)$	$\Pr(\cdot\|a_1)$	$\Pr(\cdot\|a_2)$	$\Pr(\cdot\|a_3)$	$\Pr(\cdot\|a_4)$
IR_0	0	0.998	0.313	0.887	0.949	0.990
	1	0.002	0.687	0.113	0.051	0.010
IR_1	0	0.998	0.985	0.990	0.949	0.990
	1	0.002	0.015	0.010	0.051	0.010
IR_2	0	0.998	0.985	0.990	0.949	0.255
	1	0.002	0.015	0.010	0.051	0.745
IR_3	0	0.998	0.985	0.784	0.436	0.157
	1	0.002	0.015	0.216	0.564	0.843
IR_4	0	0.142	0.910	0.165	0.393	0.010
	1	0.858	0.090	0.835	0.607	0.990
IR_5	0	0.002	0.164	0.010	0.137	0.010
	1	0.998	0.836	0.990	0.863	0.990
IR_6	0	0.868	0.836	0.887	0.179	0.745
	1	0.132	0.164	0.113	0.821	0.255
LT_1	0	0.998	0.388	0.320	0.735	0.794
	1	0.002	0.612	0.680	0.265	0.206
LT_2	0	0.989	0.388	0.320	0.735	0.794
	1	0.011	0.612	0.680	0.265	0.206
LT_3	0	0.989	0.388	0.320	0.735	0.794
	1	0.011	0.612	0.680	0.265	0.206
LT_4	0	0.998	0.463	0.320	0.735	0.843
	1	0.002	0.537	0.680	0.265	0.157

2.5.5　実機実験

学習履歴

各エピソードでのゴール到達までに要したステップ数と壁や障害物との衝突回数を Fig. 2.17 に示す．*EnvA* においてロボットは試行錯誤を通して入出力データの収集を行う．実験初期は壁に頻繁に衝突し，ゴールに到達できていない．その後，IBCG による学習が進行して壁に衝突せずにゴールに到達する行動を獲得すると，獲得した入出力のデータを用いて Naive Bayes Model を構築する．Naive Bayes Model による意思決定をした場合であっても，*EnvA* においてタスクを達成し続けている．その後，環境が *EnvB* に変化しても，引き続いて Naive Bayes Model を用いて意思決定を行うが，行動が不安定になることなくゴールに到達していることがわかる．

Fig. 2.18 は *EnvA* において IBCG によって獲得したのロボットの振る舞いである．また，*EnvA* と *EnvB* において Naive Bayes Model によって意思決定を行っているロボットのゴール到達までの軌跡を Fig. 2.19 と Fig. 2.20 にそれぞれ示す．Fig. 2.18 と Fig. 2.19 より，Naive Bayes Model で意思決定を行った場合も，IBCG で意思決定を行った場合と同じような軌道で，ゴールに到達していることがわかる．さらに，環境が *EnvB* に変化しても障害物を回避してゴールに到達していることがわかる．以上から，計算機実験と同様に構築されたモデルはその環境に特化したものではなく，環境が変化した場合であっても適用し得る汎用性を持った意思決定機構であるといえる．

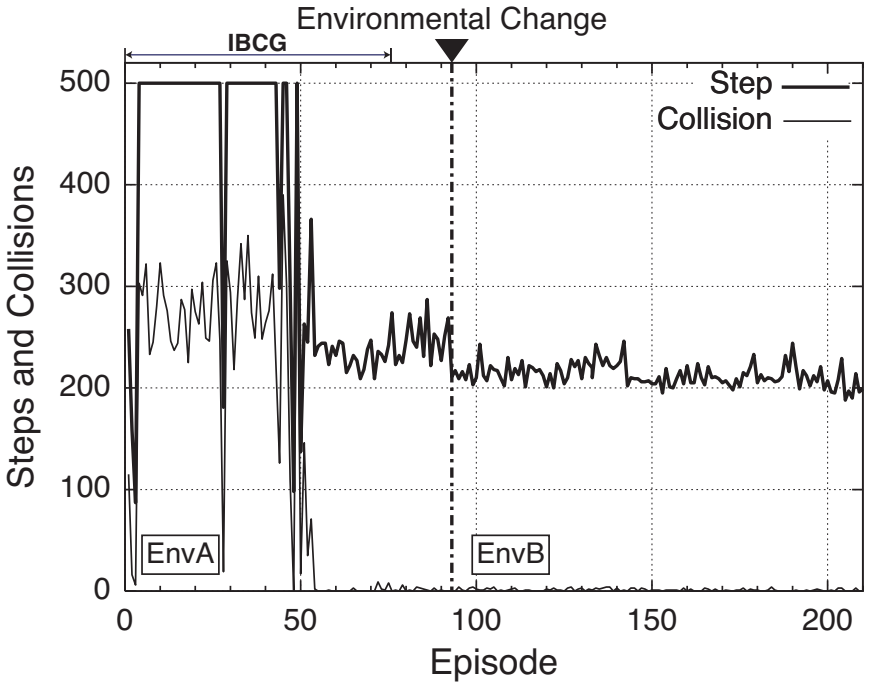


Fig. 2.17: Number of steps and collisions for Khepera

Fig. 2.18: Behavior in *EnvA* (IBCG)

Fig. 2.19: Behavior in *EnvA* (naive Bayes model)

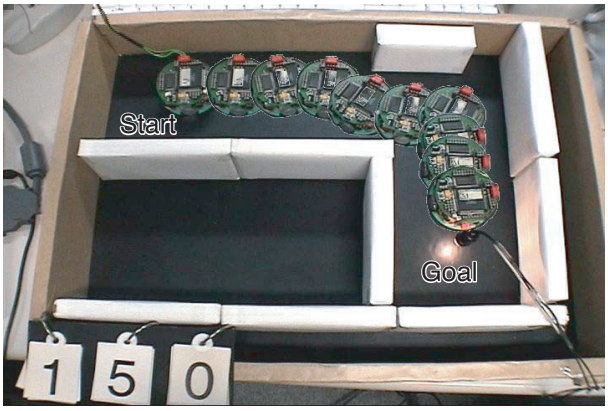


Fig. 2.20: Behavior in *EnvB* (naive Bayes model)

IBCG による行動獲得

IBCG による学習収束判断時に，ロボットが保有していたルールセットを Table 2.9 に示す．また，この時に各ステップで発火しているルールを Fig. 2.21 に示す．Table 2.9 における，ルール番号に下線が付いているものが学習収束時に発火しているルールである．

構築した Naive Bayes Model

本実験で得られた Naive Bayes Model は，Fig. 2.11 から，IR_5，IR_7，IR_8，および LT_4 ノードを取り除いた構造である．親ノードの確率分布と各センサについての条件付き確率を表す分割表を，それぞれ Table 2.10 と Table 2.11 に示す．

Table. 2.9: Rule set of IBCG for $EnvA$ for Khepera

Rule No.	Str	a	Rule No.	Str	a
1	010000000010	a_3	2	100100000110	a_3
3	110100000110	a_2	4	001000001100	a_4
5	000100000110	a_2	6	010000000000	a_3
7	110000000110	a_3	8	000001000000	a_4
9	100000001110	a_0	10	110000000000	a_3
11	000011001100	a_4	12	000000000100	a_2
13	100000000110	a_3	14	000010001110	a_2
15	110010000010	a_4	16	010000000110	a_4
17	010000001110	a_3	18	000000001110	a_4
19	010000001100	a_4	20	100000000010	a_4
21	100001000000	a_4	22	110000000010	a_1
23	000001000110	a_2	24	100100000010	a_4
25	000011001110	a_2	26	000001001110	a_2
27	001000001110	a_3	28	100000000000	a_1
29	101000000110	a_2	30	111000000110	a_1
31	110000001110	a_2	32	000000000000	a_4
34	011000000110	a_0	35	000100001110	a_4
37	000000000010	a_4	38	000000000110	a_3
39	000000001100	a_4	40	001000000110	a_4

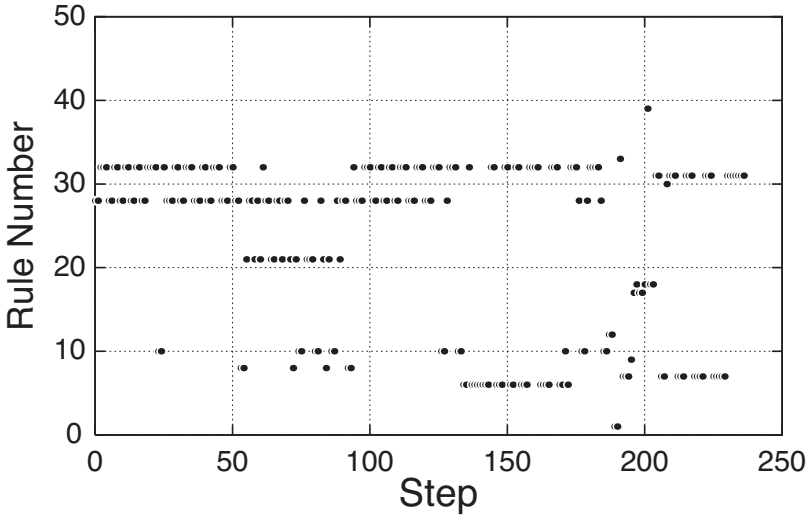


Fig. 2.21: Sequence of firing rules in $EnvA$ for Khepera

Table. 2.10: Probability of actions for $EnvA$

$\Pr(a_0)$	$\Pr(a_1)$	$\Pr(a_2)$	$\Pr(a_3)$	$\Pr(a_4)$
0.004	0.330	0.020	0.189	0.455

Table. 2.11: Conditional probability table for $EnvA$

		$\Pr(\cdot\|a_0)$	$\Pr(\cdot\|a_1)$	$\Pr(\cdot\|a_2)$	$\Pr(\cdot\|a_3)$	$\Pr(\cdot\|a_4)$
IR_1	0	0.333	0.003	0.042	0.600	0.866
	1	0.667	0.997	0.958	0.400	0.134
IR_2	0	0.667	0.920	0.208	0.107	0.986
	1	0.333	0.080	0.792	0.893	0.014
IR_3	0	0.667	0.989	0.875	0.981	0.994
	1	0.333	0.011	0.125	0.019	0.006
IR_4	0	0.833	0.997	0.917	0.995	0.994
	1	0.167	0.003	0.083	0.005	0.006
IR_6	0	0.833	0.997	0.958	0.995	0.884
	1	0.167	0.003	0.042	0.005	0.116
LT_1	0	0.167	0.997	0.250	0.856	0.924
	1	0.833	0.003	0.750	0.144	0.076
LT_2	0	0.167	0.989	0.042	0.651	0.913
	1	0.833	0.011	0.958	0.349	0.087
LT_3	0	0.167	0.920	0.125	0.651	0.818
	1	0.833	0.080	0.875	0.349	0.182

比較実験：IBCG のみの場合

Naive Bayes Model による意思決定機構の有効性を検証するため，IBCG のみで意思決定を行った場合の結果を示す．Fig. 2.22 は各エピソードにおけるゴール到達までのステップ数と壁や障害物との衝突回数である．図より，$EnvA$ から $EnvB$ に環境が変化することでロボットは壁に衝突をするようになり，ゴールに到達できなくなっていることがわかる．これは，環境が変化したことによって未経験のセンサ入力を得たときに，ランダムな行動を持つ新しいルールを生成するためである．その後，行動を再学習するまでに多くのエピソードを要している．環境変化後に不安定になったロボットの振舞いと，再学習したときの振る舞いを Fig. 2.23 と Fig. 2.24 に示す．

2.5.6 まとめ

提案手法は，強化学習のみを用いて意思決定する場合よりも環境変化に対する頑健性が高いことが計算機実験，および実機実験により示された．それにより，確率ネットワークを用いて意思決定することで，センサ入力に対する汎化能力が向上しているといえる．また，未経験の状態に陥った場合でも確率的に行動を選択するためにランダムな行動探索を抑制することも，システムの頑健性に寄与している．

計算機実験で対象とした問題では，環境が周期的に変化するために構築した意思決定機構を保持しておく必要があった．強化学習で獲得したルール集合としてではなく，

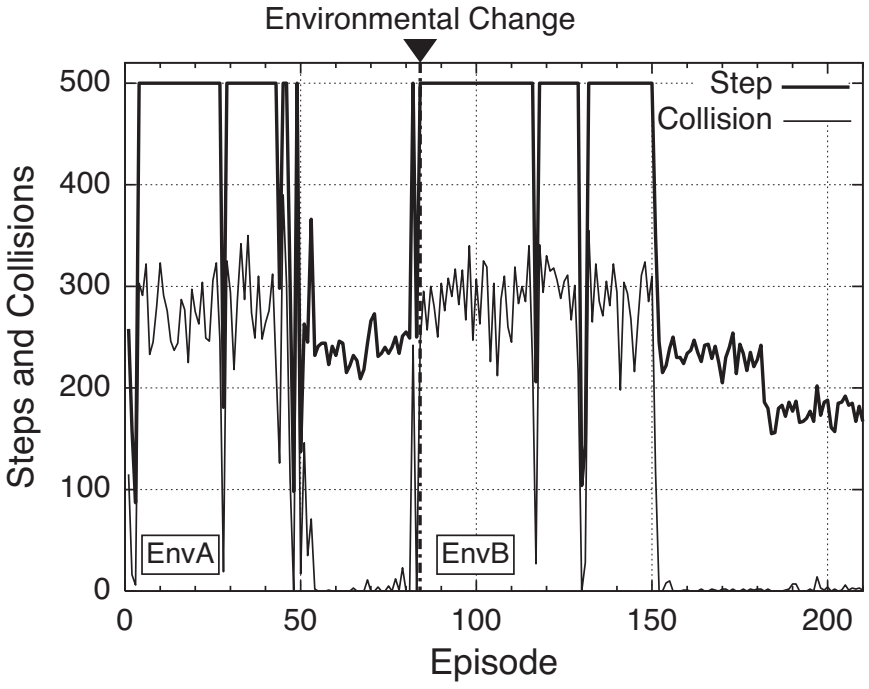


Fig. 2.22: Number of Steps and Collisions

Fig. 2.23: Behavior in $EnvB$: before successful learning (IBCG)

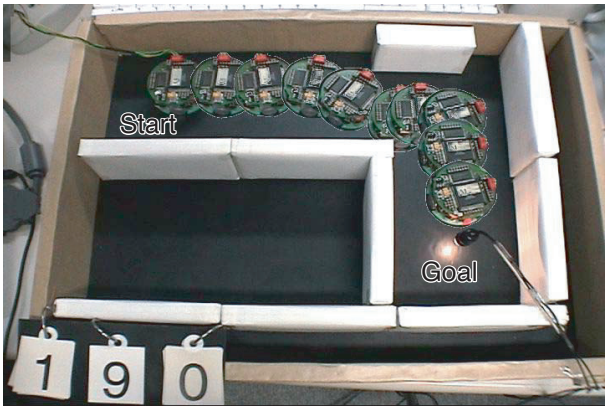


Fig. 2.24: Behavior in $EnvB$: after successful learning (IBCG)

確率モデルとして保持することで，獲得した知識をコンパクトかつ明示的な形式で記述できた．

2.6 結言

本章では，強化学習ロボットの環境変動に対する頑健性向上のための手法として，確率ネットワークを用いたロボットの獲得戦略の保存・適用手法を提案した．行動獲得後に実験を繰り返すにつれ，過学習が生じて振る舞いは環境に特化したものとなりシステムの頑健性が低下する．その問題に対処するため，強化学習は環境内を探索して適切な入出力関係を収集するために用い，そのデータを基に確率ネットワークを用いて意思決定機構を構築する手法を提案した．計算機実験，および実機実験の結果，障害物の形状に変化が生じる環境における光源到達問題において，強化学習器のみを用いたロボットよりも頑健に振る舞うことを確認した．

第3章　連続空間における頑健な強化学習法

3.1　緒言

前章では，強化学習によって獲得した知識を基に構築した確率ネットワークを用いてロボットが意思決定することで，強化学習における過学習問題を回避して頑健に振る舞う手法を提案した．本章以降，MRS を対象として，強化学習器そのものの頑健性の向上を目指す．

以下，まず，強化学習を適用するときに問題となる状態・行動空間の離散化について述べ，状態空間の抽象化手法に関する研究について説明する．次に，MRS における自律的機能分化による協調行動の実現のため，ベイズ判別法を用いた強化学習法・BRL を取り上げる．その後，BRL の頑健性向上のための拡張として，ルール集合の多様性維持のアプローチを提案する．アーム型ロボットの荷上げ問題を通して，その有効性を検証する．

3.2　強化学習のマルチロボットシステムへの適用

3.2.1　状態・行動空間の離散化の困難性

強化学習において，政策 π すなわち自律ロボットの内部モデルは，一般に Fig. 3.1 に示すような離散的な状態・行動空間を用いる．代表的な表現形式として，Look-up table[41] がある．強化学習のパフォーマンスは，状態・行動空間の離散化具合に依存する．例えば，分割の粒度が大きい場合，実際には異なる状態を同一の感覚入力だと知覚することで生じる隠れ状態問題 [88] や不完全知覚問題 [89] に陥り，マルコフ性を失って所望の振る舞いが獲得できなくなる．逆に粒度が小さければ，報酬伝搬に時間がかかって学習速度が低下し，最悪の場合は学習できなくなる．その他，人が設計することによって次のような問題が生じる．

- ロボットが必要とする情報を排除する可能性がある．
- 環境やロボット自身のシステム特性の変化によって事前に設計された状態・行動空間が無効になる可能性がある．

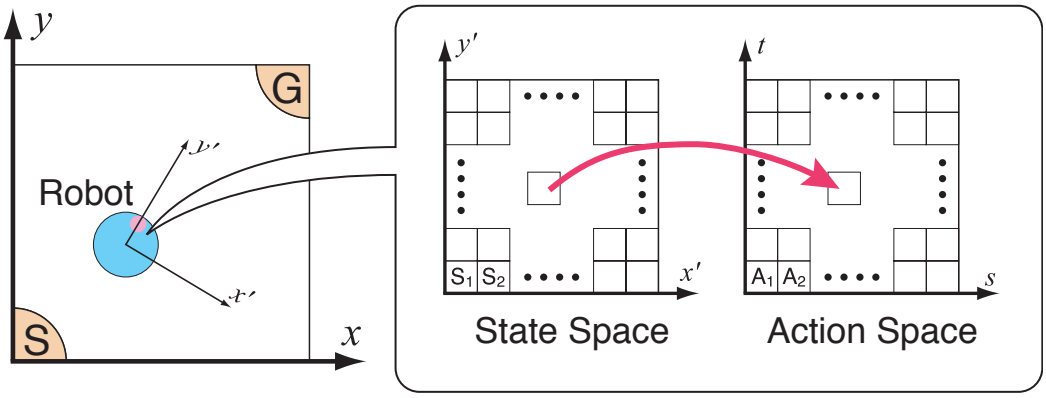


Fig. 3.1: Discrete state and action spaces

- 事前に設計したものがロボットのおかれる全ての状況で適切であるとは限らない.
- ロボットの自律性を制限する.

一般にロボットがタスクを遂行するために必要十分な情報を含む状態空間の構成はロボットの行動能力に依存し，行動空間もまたロボット自身の知覚能力に依存する [90]．そのため，ロボットにとって適切な状態・行動空間を設計者が事前に定義することは困難であり，現在のところ一般的な設計指針は存在しない.

3.2.2 状態空間の抽象化手法

本研究で対象とする MRS は，連続な空間で動作するものである．強化学習を用いて MRS の協調行動の獲得を実現するには，与えられたタスクとその動作環境，さらに，第 1 章で述べたようにロボット毎に独自に持つ入出力機構による様々な要因，すなわち身体性を考慮して，連続な状態・行動空間の離散化 (Fig. 3.2) を適切に行わなければならない.

以上のような問題に対して，ロボット自身の経験に基づいて適切な抽象化を学習によって獲得しようとする研究が行なわれている．状態空間を抽象化することで，次のような利点がある.

- 不要な情報を排除し，記憶量や計算量を軽減することができる.
- 類似状態において適応的な振る舞いが期待できる.
- 隠れ状態問題や不完全知覚問題の軽減が期待できる.

強化学習法において連続な状態空間を抽象化する方法として，次に挙げる接近法が研究されている.

- 価値関数の近似によるアプローチ

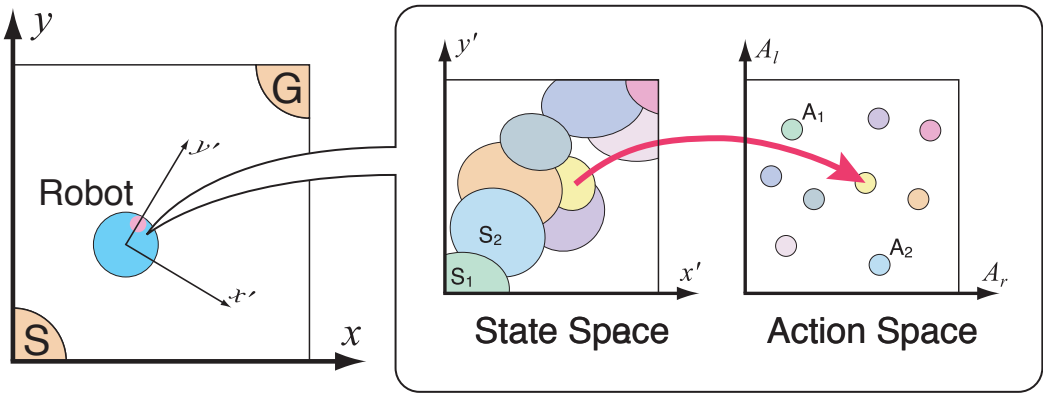


Fig. 3.2: Continuous state and action spaces

- Actor-Critic アルゴリズムによるアプローチ
- 実例に基づく強化学習によるアプローチ

以下，それぞれについて関連研究を取り上げる．

価値関数の近似によるアプローチ

ニューラルネットワークの一手法である CMAC (Cerebellar Model Arithmetic Controller) [91] を用いた価値関数の近似方法が提案されている [92]．しかし，CMAC はタイル状に分割した状態空間を階層状に組み合わせることによって価値関数の近似を行うため，Look-up table と同様に量子化の問題が生じる．この問題を解決するために，進化型計算を用いて各層の状態空間分割を設計する方法がある [93]．

一方，Lin[94] は Q-net と呼ばれる階層型ニューラルネットワークによって Q 関数を表現する手法を提案している．また，この Q-net をリカレント型に組むことで不完全知覚問題への対応をはかっている．しかし，ネットワークの構成は設計者の先験的知識に基づいて適切に設計する必要があった．深尾ら [95] は，動径基底関数 (Radial Basis Function: RBF) ネットワークにより Q 関数を表現し，Q 値を滑らかに近似するために正則化理論を用いてネットワークの結合係数を更新する方法を提案している．

Takahashi *et al.*[96] は連続な状態・行動空間の学習が可能な Continuous Valued Q-learning を提案し，サッカーロボットを用いて有効性の検証を行っている．しかし，この手法ではあらかじめ状態・行動空間を適当な粗さで離散化する必要があるために CMAC と同じような問題が生じる．この問題に対処するため，Takeda *et al.*[97] は代表点を逐次追加する手法を提案している．

堀内ら [98] は Q 値の導出にファジィ推論を導入したファジィ内挿型 Q-learning を提案している．これはファジィルールを用いて行動価値関数を近似するものであり，Q 値をなめらかに近似するとともに汎化能力を実現している．しかし，メンバシップ関

数の構造と初期値は設計者の先見的知識に基づいてあらかじめ与えなければならない．この問題に対し，梅迫ら [99] はファジィ集合やファジィルールを適応的に追加可能である自己組織型ファジィ強化学習システムを提案している．

Actor-Critic アルゴリズムによるアプローチ

Actor-Critic アルゴリズムにおける，制御出力器 (Actor) と評価予測器 (Critic) をニューラルネットワークで構成することで連続な状態・行動空間の学習を実現できる．しかし，シグモイド関数を中間層に用いたネットワークでは，(i) 各パラメータの変化がオンラインで状態空間を構成し，運動学習を行う場合には適していない [100]，(ii) 強化学習に用いた場合は学習が不安定になる [101]，という問題が指摘されており，一般に局所的な基底関数が用いられる．

Doya[102] は Actor と Critic を RBF ネットワークにより構成し，台車-振子系における振り子の振り上げの安定化に実現している．ここでは，状態空間を表現するための基底関数の数，配置，大きさなどは設計者の先験的知識によりあらかじめ決定され，また，状態空間は 2–4 次元と低次元なため，格子状に配置しても基底関数の数は十分少なく抑えることができている．しかし，未知環境下での探索や高次元状態空間中の探索を行う場合，状態空間の可能な全ての範囲に充分な精度の基底関数をあらかじめ用意すると，近似精度が高まる反面，その数が膨大になるという問題がある．

これに対し Morimoto *et al.*[103]-[105] は，正規化ガウス関数ネットワーク (Normalized Gaussian network : NGnet) によって Actor と Critic を構成することで，RBF と同様に局所的な近似を行うと同時に，基底関数が覆っている範囲外の状態空間の領域も緩やかな外挿によって汎化している．また，状態空間上に基底関数をあらかじめ配置せず，学習中に基底関数を逐次追加し，NGnet を構成するパラメータを勾配法を用いて更新する方法を提案している．近藤ら [106] は NGnet を用いて Actor と Critic を関数近似し，RBF の形状・位置の更新と行動学習を同時に行う手法として，クラシファイヤシステムの発展型である進化型 Recruitment 戦略を提案している．

一方，鮫島ら [107] は，正規化ガウス基底関数 (Normalized Gaussian Radial Basis Function : NRBF) を用いてネットワークを構成し，基底関数を追加するのではなく，TD-error に基づいて基底関数を分割することで NRBF ネットワークの近似精度を上げる方法を提案している．そして，8 次元のシミュレーション環境において自律エージェントによる壁伝い行動を実現している．この手法では，エージェントの強化は逐次報酬によって行われている．また，分割方向を決定するために基底関数の分散共分散行列の固有値，固有ベクトルを求める必要がある．また，石井ら [108][109] は，NGnet により Actor と Critic を表現し，入出力両方の統計的性質から基底関数の分割，追加，削除を行なう手法を提案している．基底関数およびネットワークの結合係数のパラメータの更新には On-line EM アルゴリズム [110] を用いる．

以上の方法では，Actor と Critic のそれぞれに関してネットワークを構成するのに対し，柴田ら [111] は，学習が不安定になることが指摘されていたシグモイド型ニューラルネットワークをひとつで構成している．また，Critic の学習に時間スムージング学習 [112] を用い，Actor の学習には Back Propagation(BP) 法を用いた学習アルゴリズムを提案し，視覚センサによる障害物回避行動獲得を通して有効性を検証している．さらに，柴田 [113] は連続な出力と離散的な出力を同時に学習することができる Actor-Q アーキテクチャを提案している．この Actor-Q アーキテクチャでは，Critic の学習を TD(λ) ではなく，Q-learning により行っている．

木村ら [56][72] は Actor の政策改善に REINFORCE アルゴリズム [86] を用いて行動空間の汎化を行なっている．そして，Actor の学習に適正度の履歴 (eligibility trace) を用いることで非マルコフな環境において頑健な学習法を提案している．

以上のように Actor-Critic アルゴリズムは連続な入出力間の写像関係を学習することができる．しかし，目的関数を適切に近似するためには，RBF などの基底関数を適切に配置する必要がある．このとき，基底関数を多く配置するとネットワークの汎化性が低下し，ノイズへの耐性が低下する，逆に少ない場合は目的関数を近似できないなどの問題がある．また，強化学習の特徴である，知識表現の透明性を損ないかねない．

実例に基づく強化学習によるアプローチ

実例に基づく強化学習では，多くのインスタンス (実例) を加工せずに記憶し，新しい入力データを与えられた時に任意の尺度にしたがって類似度を計算する．そして，入力に対して最も類似性が高いインスタンスを選択し，インスタンスに記憶されている動作を実行する．実例に基づく強化学習の枠組では，各学習法はインスタンスの記憶方法と類似度の計算方法によって特徴付けられる．

畝見 [60][61] は入力データと記憶しているインスタンスの類似度をユークリッド距離によって計算し，連続な入力を学習する方法を提案している．動作選択時，類似したインスタンスがある場合は，それに記述されている動作を実行し，類似したインスタンスが無い場合は，ランダムに動作を実行し，新たにインスタンスを記憶する．そして，Bucket Brigade 的な報酬伝播を用いて各インスタンスの信頼度を更新し，信頼度の低いインスタンスを消去することで記憶量の増大を抑えている．

深尾ら [114] は Q-learning において，経験したデータをそのままデータとして蓄積，更新，または削除することで，状態空間を適応的に分割する手法を提案している．データの追加と削除は，選ばれたデータとデータベース内の他のデータとの距離を基に決定する．

小林ら [115] は，二値の即時評価のわかる問題において，同じ行動をとっても評価が異なる状態同士を識別するという考えを基に，状態を表すノードを逐次生成するた

めにベクトル量子化を行う．そして，量子化された状態の位相関係を利用して位相近傍の状態同士を結合するように行動を修正するアルゴリズムを提案している．

Asada[116] は，実際にロボットが観測した入出力データをオフラインで処理し，同一の動作で到達できる状態を一つのクラスとして切り出し，マハラノビス距離によって状態空間を超楕円体で分割する方法を提案している．Ishiguro[117] はセンサ入力と獲得した報酬を入力データとし，オフラインで獲得報酬が類似している状態を一つのクラスとして切り出し，線形近似関数によって状態空間を超平面で分割する方法を提案している．これらの状態空間分割の特徴は，記憶した入出力データをオフラインで処理し，マハラノビス距離や線形近似関数などのパラメトリックな手法でクラスタリングされた状態を表現している点である．

Takahashi *et al.*[118][119] はセンサ入力の変化の類似性に基づいて状態を一つのクラスとして切り出し，Nearest Neighbor 法により状態空間を凹型に分割・統合する方法を提案している．また，Ueno *et al.*[120][121] は Nearest Neighbor 法とマハラノビス距離を用いた「齧られた超楕円体表現 (bitten hyper-ellipsoid representation)」を用い，より詳緻な状態空間の表現方法を提案している．これらの状態空間分割の特徴は，記憶した入出力データをオンラインで処理し，クラスタリングされた状態をノンパラメトリックな手法で表現している点である．

Yairi *et al.*[122] は異なるセンサを統一的に扱う方法としてベイズ分類器を用いた状態空間の分割法を提案している．観測した入出力データをオフラインで処理し，状態空間をノンパラメトリックな手法を用いて表現している．この手法では，連続値と離散値を同時に扱うことが可能である．また，観測した入出力データをオフラインで処理し，行動結果の類似性を基に状態をクラスタリングし，決定木によって分割された状態空間を表現する方法を提案している [123].

上述のように実例に基づく強化学習における状態空間の表現方法として，超楕円体や超平面などのパラメトリックなモデルを用いる方法と，k-Nearest Neighbor 法のようなノンパラメトリックなモデルを用いる方法がある．パラメトリックな表現は，小数のパラメータで関数が表現できるため，新しいデータに対する確率密度の計算が比較的簡単である．しかし，真の分布と仮定したモデルが異なる場合には必ずしも良い推定結果が得られるとは限らない．ノンパラメトリックな表現は，真の分布がどんな関数系であっても推定できるという点で有効である．しかし，新しいデータに対して確率的な評価をするために大量のデータを記憶する必要があり，記憶量軽減の方法を考えなければならない．

記憶したインスタンスの更新方法には，オフライン学習 (バッチ処理) とオンライン学習 (リアルタイム処理) がある．オフライン学習は，学習フェーズと実行フェーズから構成されており，学習フェーズでは環境中をランダムに探索し，大量に収集したデータから固定的な状態を構成する．そして，実行フェーズでは固定された状態においてタスクを遂行する．また，状態表現が不十分であればオフライン学習によって新

たな状態を追加することができる．しかし，既存の状態を修正，消去することはできない．学習フェーズでは，効率良くデータを集めることが目的なので，そのための制御則が必要となる．オンライン学習は，ロボットが観測したデータから逐次状態空間を修正する．学習の目的は環境やシステムが変化しても破綻せずにタスクを遂行することである．環境の小さな変化には，状態形状を修正することで性能を維持したまま適応でき，状態形状を逐次修正するため，オフライン学習のように事前に大量のデータを収集する必要が無い．そのため，報酬に到達する状態・行動の流れが生成され，状態を表現するのに有効なデータが効率良く収集できる．

3.3 連続空間における強化学習法における頑健性の向上

3.3.1 ベイズ判別法を用いた強化学習法：BRL

設計指針

連続空間における強化学習の関連研究を踏まえて，本研究で用いる強化学習手法 Bayesian-discrimination-function-based Reinforcement learning (BRL)[65] の位置づけを明確にする．

■ 実例に基づく強化学習の枠組み Actor-Critic アルゴリズムは，基底関数ネットワークにより入出力関係を記述するため，状態・行動空間の分割や隠れ状態の発生に煩わされない．しかし，前述のように基底関数の数や配置位置，大きさなどの設定の必要があり，推論やプランニングといった問題への拡張のが難しい．

一方，実例に基づく強化学習は，状態空間をクラスタリングし，獲得された知識を if-then 形式のルールとして記述するため，推論やプランニングへの拡張が容易である．自律ロボットの行動獲得には有効であると考えられる．

BRL は実例に基づく強化学習の枠組を用い，if-then 形式のルール表現している．そして，状態空間をクラスタリング手法により分割している．

■ パラメトリック・モデルによる状態空間表現 ノンパラメトリック・モデルは，真の分布がどんな関数系であっても推定できるという利点がある．しかし，新しいデータに対して確率的な評価をするために大量のデータを記憶する必要があり，状態空間の分割数の増加に合わせて計算量が増える．

一方，パラメトリック・モデルは，仮定したモデルが新しいデータに対して異なる場合には必ずしも良い推定結果が得られるとは限らない．しかし，ひとつの代表点でひとつの状態を表現でき，記憶量を低く抑えることができ，計算量も比較的少ない．自律ロボットの内部モデルを逐次改善することを考え，多数の記憶データを取り扱うノンパラメトリック・モデルより，パラメトリック・モデルを採用する．

■ オンラインでの内部モデル更新　実環境で作動する MRS では，時々刻々と変化する環境やシステムに対し，柔軟に対応する必要がある．そのためには，内部モデルをオンラインで更新しなければならないと考える．また，実例に基づく強化学習では，内部モデルは状態空間の分割の粒度で定義され，オンラインで状態空間の分割を更新する方が一般に良いとされている [121].

以下，これらの観点から設計した BRL の詳細を述べる．

アルゴリズムの概要と特徴

自律的に状態空間を分割する実例に基づく強化学習において，分割された状態空間では入力がどの状態に分類されるべきか識別する必要があり，このとき，センサに含まれるノイズなどを考慮して確率的に識別することが望ましい．そこで本手法では，統計的にパターン分類を行うベイズ判別法 [124][125] を用いて入力がどの状態に属するのか識別する．

ベイズ判別法は，識別したい K 個のクラス $C = \{C_k\}_{k=1}^{K}$，識別対象を計測して得られる特徴ベクトル $\boldsymbol{x} = \{x \in R^M\}$，事前確率，すなわち識別対象がクラス C_k に属している確率 $\Pr(C_k)$，およびクラス C_k に属する対象を計測したときに入力 x が観測される確率密度関数 $\Pr(x|C_k)$ が既知の場合，入力 x をクラス C_k に分類したときの事後確率 $\Pr(C_k|x)$ を以下のベイズの公式 [126] より求めることができる．

$$\Pr(C_k|x) = \frac{\Pr(C_k)\Pr(x|C_k)}{\sum_{k=1}^{K}\Pr(C_k)\Pr(x|C_k)}, \tag{3.1}$$

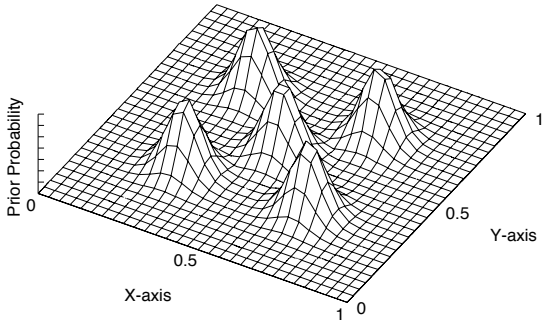
ただし，　$\sum_{k=1}^{K}\Pr(C_k|x) = 1$，　$\sum_{k=1}^{K}\Pr(C_k) = 1$ である．したがって，$\Pr(C_k|x)$ 最大となるクラスに入力を分類することが最良である．

しかし，確率モデルが事前に分かっていることは稀であり，自律ロボットの行動学習では入出力の完全なデータセットをあらかじめ用意することは困難である．そのため，観測データから確率モデルを推定する必要がある．BRL では，(1) クラスの追加と削除，(2) 確率分布モデルのパラメータ更新によって観測データから環境の確率モデルをリアルタイムに更新し，状態空間の分割を行う．

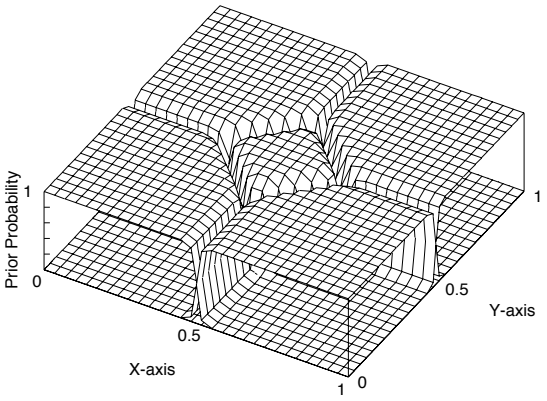
BRL は，各クラスの確率分布を Gauss 分布によって近似し，**各クラスの確率分布を表すパラメータ**とそのときの**出力**を if-then 形式で記述したルールとして学習器に記憶する．これ以降，クラスとルールを同義として扱う．

学習初期，状態空間にはクラスは存在せず，ロボットが観測した入出力を基に状態空間にクラスを追加し，Fig.3.3(a) に示すように状態空間をガウス分布で覆っていく．各ルールの事後確率を求めると，Fig.3.3(b) のように大域的な領域を覆うが，未経験な領域まで覆うのは好ましくない．そこで，Fig.3.3(d) に示すようにしきい値 P_{th} を設け，事後確率を計算し，Fig.3.3(c) に示すような状態分割を得る．

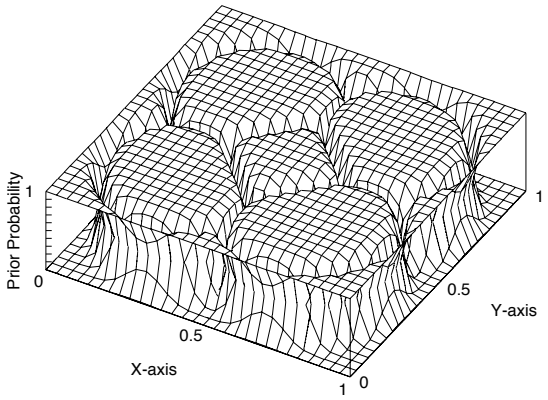
動作選択は入力に対する各ルールの事後確率をベイズの公式から求め，事後確率最



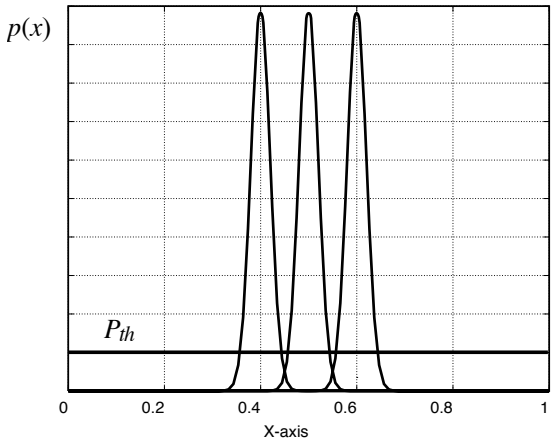
(a) Probability Distribution



(b) Posterior Probability without Threshold



(c) Posterior Distribution with Threshold



(d) Threshold

Fig. 3.3: Segmented state space by Bayes boundaries

大のルールを選択し，そのルールの出力を実行する．入力がどのルールにも分類されない場合，ランダムに動作を実行し，このときの入出力を記憶した新しいルールを状態空間に追加する．

また，BRL では独自に設計された評価の更新法によって報酬獲得に協調的に寄与するルール群を強化し，罰を受けたり報酬獲得に寄与しないルールを削除する．この操作により，あらかじめ動作集合を用意しなくても，ランダムに動作を生成し，評価に基づいて不要なルールを削除することで環境に適した動作集合が獲得できる．

各ルールは強化信号と入力をもとに確率分布のパラメータをオンラインで更新し，各ルールが覆っている領域，すなわちルール条件部を更新する．

このようにリアルタイムにルールの追加や削除，ルール条件部の更新を行なうことで環境やシステムの変化に対し，迅速に対応することが期待できる．しかし，オンライン学習の場合，ノイズや一時的な入力の偏りに対処する必要がある [121] ．そのため，BRL では，区間推定法 [127] を用いたパラメータ更新によりこの問題を解決する．

区間推定法は，確率分布のパラメータがある区間に入る確率を設定した確率以上になるように保証する手法であり，サンプル数が増大するにつれて推定精度が上がる．そのため，観測データの増加に伴ってより信頼性の高いパラメータ推定が期待できる．以上のことより，本手法の特徴をまとめる．

- ルールは確率分布のパラメータとそのときの出力で構成され，ルールの生成と削除により，適切な状態・行動空間の分割を学習する．
- 各クラスの持つパラメータをリアルタイムで更新することで，環境やシステムの変動に対する適応能力を高めている．
- 各クラスの持つパラメータの更新を区間推定法に基づき行う事で，ノイズや一時的な入力の偏りに対しての頑健性を高めている．

ルール構成

ルール集合 R はルール $rl \in R$ により構成され，各ルールは次式で記述される．

$$rl :=< \boldsymbol{v}, \boldsymbol{\Sigma}, f, u, \Phi, \boldsymbol{a} > \tag{3.2}$$

各ルール rl は特徴ベクトル $\boldsymbol{v} = \{v_1, \cdots, v_{n_d}\}^T$ ，分散共分散行列 $\boldsymbol{\Sigma}$，クラスの生起確率 f，クラスの信頼性を表す有効度 u，各クラスで観測された入力の記憶集合 $\boldsymbol{\Sigma} = \{\phi, \ldots, \phi_{n_s}\}^T$，そして，動作 $\boldsymbol{a} = \{a_1, \ldots, a_{n_a}\}^T$ より構成されている．ただし，記憶集合 Φ は $\phi_i = \{x_1, \ldots, x_{n_d}\}$ より構成され，n_d は入力空間の次元数，n_a は出力空間の次元数，n_s は各クラスが記憶しているサンプル数を表す．

学習手順

(1) ベイズ判別法により入力の分類先を識別する．

  (a) 分類先のルールがある場合
  $\Rightarrow$ 分類先のルールの動作 $\boldsymbol{a}$ を実行．

  (b) 分類先のルールがない場合
  $\Rightarrow$ ランダムに動作を実行．

(2) 実行した動作の評価として強化信号 r を受ける．

(3) 手順 (1) において分類先のルールがなく，動作実行時に負の強化を受けない場合，新しいルールを生成．

(4) 強化信号をもとに全ルールの有効度を更新．

(5) 入力データを基に全ルールのパラメータを更新.

(6) 終了条件を満たさない場合 (1) に戻る.

動作選択

動作選択はベイズ判別法によって行う．ベイズ判別法では，事後確率を最大とするクラスに分類するのが最適であるが，ここでは，まず事後確率の負の対数を取り，誤って識別する確率 g_i が最小となるルールを勝者ルール rl_w とする．そして，g_i としきい値 $g_{th} = -\log\{f_0 \cdot P_{th}\}$ を比較して，

- $g_w < g_{th}$ の場合，rl_w の動作 A_w を実行.
- $g_w \geq g_{th}$ の場合，ランダムに動作を実行.

とする．ただし，n_{rl} は学習器内の全ルール数を表し，f_0 と P_{th} は定数である．

$$\begin{aligned} g_w &= \min_i\{g_i\} \qquad i \in [0, n_{rl}] \\ g_i &= -\log\{f_i \cdot p(\boldsymbol{x}|C_i)\} \\ &= \frac{1}{2}(\boldsymbol{x}-\boldsymbol{v}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{v}_i) + \log\left\{(2\pi)^{\frac{n_s}{2}}|\boldsymbol{\Sigma}_i|^{\frac{1}{2}}\right\} - \log f_i \end{aligned} \tag{3.3}$$

ここで，ルール rl_i の領域において入力分布の次元が入力空間の次元から縮退したとき，$\boldsymbol{\Sigma}^{-1}$ と $\log|\boldsymbol{\Sigma}|$ が発散するのを防ぐため，

$$\boldsymbol{\Sigma}_B = \boldsymbol{\Sigma} + \delta I \tag{3.4}$$

とし，式 (3.3) の計算には $\boldsymbol{\Sigma}_B$ を用いる．ただし，ここでは計算量軽減のために $\boldsymbol{\Sigma}$ として対角行列を用いる．なお，δ は非常に小さい定数であり，I は $\{n_d \times n_d\}$ の単位行列である．

ルール生成

新しいルール rl_n は，(i) 動作選択時に分類先のルールがなく，(ii) 動作実行時に負の強化を受けない場合，生成される．rl_n のパラメータは (3.5) 式によって与えられ，状態ベクトル $\boldsymbol{v}_n$ と動作 $\boldsymbol{a}_n$ は，分類先がなかったときの入力 $\boldsymbol{x}$ とランダムに実行した動作 $\boldsymbol{a}$ を割当てる．分散共分散行列 $\boldsymbol{\Sigma}_n$ の対角成分は，近傍の状態空間の分割粒度に合わせるため，最も近いルールの分散 σ^2 から (3.6) 式によって求める．ただし，ルール集合にルールがない場合，$\sigma_n = \sigma_0$ とする．

$$\boldsymbol{v}_n = \boldsymbol{x},\ \boldsymbol{a}_n = A,\ \boldsymbol{\Sigma}_n = \sigma_n^2 I,\ u_n = u_0,\ f_n = f_0 \tag{3.5}$$

$$\sigma_n = \frac{1}{n_d}\sum_{i=1}^{n_d}\sigma_i \tag{3.6}$$

ただし，u_0，f_0，σ_0 は定数であり，I は単位行列である．

有効度の更新

各ルールの有効度は次の四つの方法で更新される．

■ **Profit Sharing (PS)**　[54] ペイオフには有効度を増す報酬 ($P > 0$) と有効度を減らす罰 ($P < 0$) がある．これらのペイオフは割引率 ($0 < \gamma < 1$) によって減衰しながら，報酬を獲得した時点から過去に遡って勝者ルールに与えられる．

$$u_w^{(n_t)} \quad \leftarrow \quad u_w^{(n_t)} + \gamma^{n_t} P \tag{3.7}$$

ただし，$u_w^{(n_t)}$ はペイオフを与えられた時点から n_t ステップ前の勝者ルール $rl_w^{(n_t)}$ の有効度を表す．

■ **Bucket Brigade 的戦略 (BB)**　勝者ルール $rl_w^{(t)}$ は，その有効度の一部 Δu を 1 ステップ前の勝者 $rl_w^{(t-1)}$ に伝播させる．

$$u_w^{(t-1)} \leftarrow u_w^{(t-1)} + \Delta u \tag{3.8}$$

$$\Delta u = \begin{cases} \kappa\, \lambda_w\, (u_w^{(t)} - u_w^{(t-1)}), & \text{if } \ u_w^{(t)} > u_w^{(t-1)} \\ 0, & \text{otherwise} \end{cases} \tag{3.9}$$

ここで，$\kappa \in [0, 1]$ である．一般的な Bucket brigade 法 [55] との違いは勝者ルール rl_w の有効度が減少しない点である．

■ **コスト**　動作選択において選択され，動作を実行した rl_w は，コスト $c_f \cdot u_w$ を払う．これにより，デッドロックやループ行動の生成を抑制する．

$$u_w^{(t)} \leftarrow (1 - c_f)\, u_w^{(t)} \tag{3.10}$$

■ **消散**　全ルールはタスク到達時に消散率 η に応じて有効度を割り引かれる．

$$u_w \leftarrow \eta\, u_w \tag{3.11}$$

以上の操作によって有効度がしきい値 u_{min} 以下になったルールは削除される．

パラメータの更新

選択されたルール rl_w は，そのときの入力 $\boldsymbol{x}$ を記憶集合 Φ に加える．ただし，入力が rl_w の $3\sigma(\sigma$:標準偏差) の範囲外にある場合はノイズとして記憶せず，また，最大記憶容量 $n_{s_{max}}$ を越えると最も古いデータと入れ換える．そして，各ルールのパラメータは次の手順で更新する．

■ 勝者ルール rl_w のパラメータ更新

(1) rl_w のサンプル集合 Φ から各次元ごとに標本平均 $\bar{x}_j$ と標本分散 s_j^2 を求める．[(3.12) 式]

(2) $\bar{x}_j$ と s_j^2 から平均と分散の信頼区間を求め，rl_w の平均 v_j と分散 σ_j^2 を比較する．[(3.13) 式]

(3) 平均 v_j と分散 σ_j^2 が信頼区間外に存在する次元のパラメータを更新する．[(3.14) 式]

(4) f_w を更新する．

(a) $P \geq 0$ の場合，(3.15-a) 式より f_w を更新.

(b) $P < 0$ の場合，(3.15-b) 式より f_w を更新.

■ その他のルール rl のパラメータ更新

(a) $P \geq 0$ の場合，(3.15-b) 式により f を更新.

(b) $P < 0$ の場合，更新なし．

以下にパラメータ更新で使用した式を示す.

$$\bar{x}_j = \sum_{k=1}^{n_s} \phi_k^j / n_s \tag{3.12-a}$$

$$s_j^2 = \sum_{k=1}^{n_s} (\phi_k^j - \bar{x}_j)^2 / (n_s - 1) \tag{3.12-b}$$

ここで，$j \in [1, n_d]$ は j 番目の次元を表し，$k \in [1, n_s]$ は k 番目のサンプルデータを表す．したがって，ϕ_k^j は k 番目のサンプルデータの j 次元成分を表す.

$$\left[\bar{x}_j - \frac{t_{\alpha/2}(n_s - 1) \cdot s_j}{\sqrt{n_s}}, \bar{x}_j + \frac{t_{\alpha/2}(n_s - 1) \cdot s_j}{\sqrt{n_s}} \right] \tag{3.13-a}$$

(平均の信頼区間)

$$\left[\frac{(n_s - 1) \cdot s_j^2}{\chi^2_{\alpha/2}(n_s - 1)}, \frac{(n_s - 1) \cdot s_j^2}{\chi^2_{1-\alpha/2}(n_s - 1)} \right] \tag{3.13-b}$$

(分散の信頼区間)

ここで，$t_{\alpha/2}(n_s - 1)$ と $\chi^2_{\alpha/2}(n_s - 1)$ はそれぞれ t 分布と chi^2 分布のパーセント点を表す.

$$v_j \leftarrow v_j + \alpha(\bar{x}_j - v_j) \tag{3.14-a}$$

$$\sigma_j^2 \leftarrow \sigma_j^2 + \alpha^2[s_j^2 - \sigma_j^2] \tag{3.14-b}$$

$$f \leftarrow f + \beta(1 - f) \tag{3.15-a}$$

$$f \leftarrow (1 - \beta) f \tag{3.15-b}$$

ここで，α と β は学習係数である.

3.3.2 BRL における過学習問題

BRL はパラメータの更新やルールの生成によって，環境変化に適応することができる．しかし，行動獲得後に実験を繰り返した場合，有効度の更新法のひとつである消散の作用によってタスク達成に寄与しないルールは削除され，寄与するルールのみが強化されてルール集合に残る．それにより，環境変化後であってもそれまでのルールが有効であるか否かに関わらずに頻繁に発火し，新しい行動の獲得を妨げる．換言すれば，BRL ではこのような環境に特化したルール集合となることが原因で過学習が生じ，システムが不安定になる．このように従来型 BRL で生じる過学習による環境変化への頑健性の低下に対する対処法が必要になる．BRL は実例に基づく強化学習法であることやベイズ判別法を用いるという特徴を持つ．これらの特徴に起因して，部分的な再学習ではその前後を適切に繋ぐ行動を獲得することは難しいなどの問題点が挙げられる．また，MRS 環境において，学習収束後に新しい探索を行うことはシステムの不安定化につながりかねない．そのため，上述の手法をそのまま適用することができず，異なるアプローチが求められる．

3.3.3 過学習の抑制のための BRL の拡張

ルール集合の多様性維持によるアプローチ

実環境ではノイズなどの外乱やロボットの故障などの不確定要素により，なんらかの環境変化が常に生じているといえる．本研究では，環境の変化を陽に意識することなく，さまざまな環境に対応し得るルール集合を構成することで頑健性の向上をはかる．すなわち，過学習に対処するために，報酬獲得に直接寄与するルールだけではなく，不確定要素に対応し得るように多様性のあるルール群により状態空間を幅広く覆うことで頑健な大域的秩序形成を試みる．これにより，(i) はじめに獲得した行動とは異なる安定な行動に寄与する，(ii) 特定ルールが発火し続ける確率の上昇を抑制する，という効果が期待できる．

しかし，単に学習過程において生成したルールをすべて保持すれば，探索するほどルール数が増加するのみである．この場合，不要になったルールまで記憶することになり，学習効率を落とす可能性が生じる．そこで，学習過程において削除から保護すべき信頼度の高い有効ルールを選定する基準が必要となる．

有効ルール保護のための指標

強化学習は試行錯誤を通して徐々に適切な入出力関係を構築する手法であるため，学習進度を基にルールの有効度を判断できると考える．これまでに，学習進度は学習効率の向上を主な目的とした研究で用いられている．石井ら [75] は状態価値関数の逆

分散を行動の確信度として各状態における学習の進行度を定義している．また，尾川ら [128] は予測誤差に基づき更新される内部モデルの信頼度 [129] を学習進行度の指標として用いている．本論文ではこれらの研究と同様に，学習を通して更新されるパラメータを基に，学習進度を計算する．つまり，ルール (パラメータ) 構成の進行度を，信頼度の高い保持すべきルールの選択指標として捉える．

BRL のルールパラメータのうち，有効度と正規分布の特徴量である分散共分散行列が学習の進行に伴って変化する．それらの更新方法は以下の通りである．

■ **有効度：** Profit Sharing と Bucket Brigade 的戦略により報酬を過去に遡って伝播させる．その他，ループ行動を防ぐために選択されたルールに課すコスト，報酬獲得に寄与しないルールを削除してメモリ消費量を抑えるためにタスク達成時に全ルールに作用させる消散がある．この四つの更新方法のうち，Profit Sharing と Bucket Brigade 的戦略によって有効度は向上するが，このときの更新量はゴール付近のルールほど大きい．また，ゴール判定にノイズによる不確定な要素が含まれる．そのため，有効度の大小のみで，エピソードを通してのルールの構成度合を判断することはできない．

■ **分散共分散行列：** 各成分は，そのルールが発火した場合に (3.14-b) 式により更新される．分散共分散行列は用いられる回数が増えるほど，更新量が大きくなる．つまり，学習が進行するにしたがって，行動は安定して特定のルールの発火する回数が増えることから，ルールの構成が進むほど分散共分散行列の行列式の値が小さくなる特性を持つ．

BRL の拡張

以上より，本論文では分散共分散行列に着目し，その行列式を用いて構成状況の指標 $\mathcal{R}$ を次式のように定義する．

$$\mathcal{R}_i = \frac{|\mathbf{\Sigma}_i|}{|\mathbf{\Sigma}_0|} \quad i = 1, 2, \cdots, n. \tag{3.16}$$

ただし，$\mathbf{\Sigma}_0$ は分散共分散行列の初期値，n はエピソード終了時におけるルール数である．

従来の BRL ではゴール到達した場合，全てのルールの有効度を η 倍 $(0.0 < \eta < 1.0)$ して減じさせる (消散)．これを変更し，各ルールの有効度の更新法のうち，消散は $\mathcal{R}_i$ が大きいルールについてのみ適用することとする．すなわち，定数のしきい値 $\mathcal{R}_{th}$ よりも $\mathcal{R}_i$ が小さいものを有効ルールとして消散を適用せずに削除から保護することでルール集合の多様性を保持させ，過学習に陥るのを防ぐ．

$$u_i \leftarrow \eta\, u_i \quad \text{if} \quad \mathcal{R}_i > \mathcal{R}_{th} \tag{3.17}$$

3.4　アーム型ロボットの協調荷上げ問題による検証

3.4.1　問題設定

協調タスクとして三台のアーム型ロボットによる協調荷上げ問題を扱う (Fig. 3.4). 各ロボットのアーム先端は正三角形の荷の各頂点に回転自由に連結されている．各ロボットは自身の各関節の角度と荷の傾きを知覚するのみであり，直接的に他ロボットの状態を知覚することはできない．また，他ロボットとの通信や報酬伝播などは行なわない.

問題の特徴は次のように記述できる.

- ロボット毎に独立した学習器を持つため，互いに非同期に制御される.
- 実空間での行動学習であるため，センサ・ノイズや決定行動と実行された行動のズレが生じる.
- 荷が所定の高さに到達したかどうかは，荷の三つの頂点付近に取り付けられた IR センサにより判定される．そのため，ノイズによってタスク達成の判定が正しく行えない場合があり，評価系にも不確定な要素が含まれる.
- ロボットは受動車輪を持ち，他ロボットに押されるまたは引っ張られることで前後に移動する.

三台のロボットは機能や構造的な初期状態は均質であるが，非同期で動作するためタスクの達成にはそれぞれが異なる働きをしながらシステム全体として安定した協調

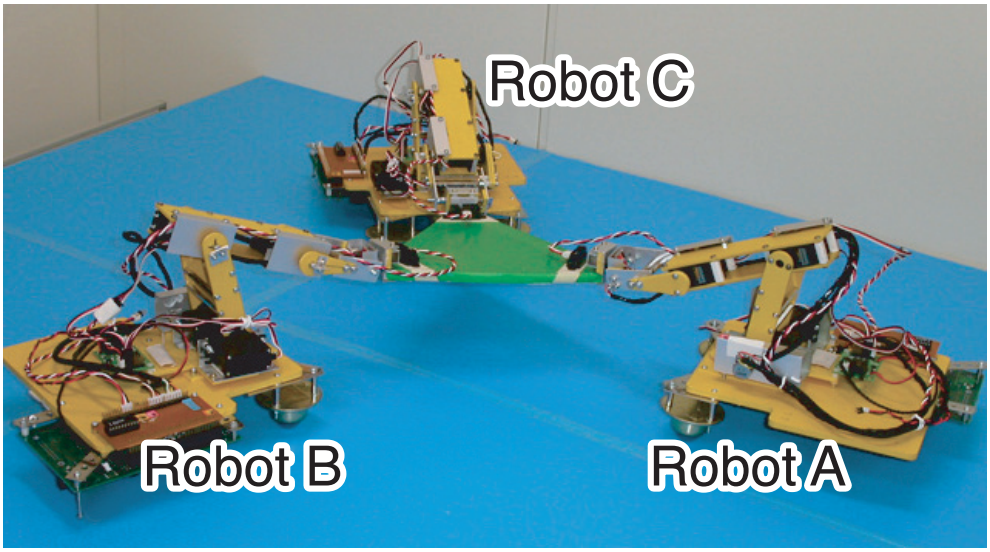


Fig.　3.4: Three arm-type autonomous robots

的振舞いをしなければならない．そのため，各ロボットは自身の振舞いだけでなく，集団の一員としての役割を獲得し，それらを適切に割り当てる必要がある．

3.4.2 実験設定

Fig. 3.5 にアーム型ロボットの制御システムを示す．アームは三関節を持ち，それぞれに取り付けられたサーボモータの動作範囲は 0〜90 度で 0.35 度の精度で制御できる．各関節角度 (p_1, p_2, p_3)，およびアーム先端における荷のピッチ角 (Pit) とロール角 (Rol) を測定するためにポテンショメータ (角度センサ) がそれぞれ取り付けられている．各ロボットの制御システムはロボットのサーボモータとセンサを管理し，各ロボットの意思決定はそれぞれの PC に実装された学習システムによって行われる．

ロボットはいずれも床から約 130mm の高さから荷を持ち上げ始める．荷の各頂点付近に取り付けた IR センサの全ての値がしきい値 θ_{IR} を越えた時点でゴールとする．その時の床から荷までの高さは約 270mm である．入力 $x = \{Pit, Rol, p_1, p_2, p_3\}$ は正規化した連続値を用いる．出力 $y = \{\theta_1, \theta_2, \theta_3\}$ は関節の角度変化量であり，θ_i は $|\theta_i| < \theta_{max}$ の範囲で出力される $(\theta_{max} = 9°)$．入出力の 1 サイクルを 1 ステップとし，ゴールに到達するか，あるいは 120 ステップまでにゴールできない場合，エピソードを終了する．エピソードの開始は手動で行い，終了後も手動で初期状態に戻し，エピソードを更新する．強化信号 (正：報酬，負：罰) は，それぞれ以下の場合に与える．

- 報酬： ゴール到達時
- 罰：荷の傾きがしきい値 $(T_l \approx 27°)$ 以上になった時

実験で用いた学習パラメータを Table 3.1 に示す．なお，これらの BRL のパラメータは推奨値である．

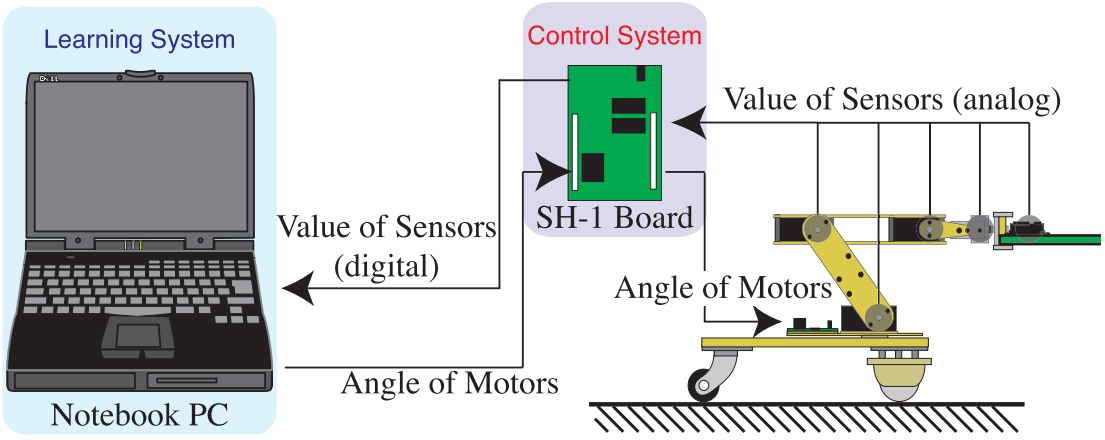


Fig. 3.5: Control system

Table. 3.1: Parameters of learning system

n_{max}	maximum size of the rules	100
P	payoff(reward)	30.0
P	payoff(penalty)	$-0.01u$
u_0	initial utility	10.0
u_{min}	threshold for extinction	$0.92u_0$
c_f	cost for an action	0.007
γ	distribution rate of utility	0.5
κ	utility spread rate	0.15
η	evaporation rate	0.98
f_0	initial prior probability	0.001
σ_0	initial variance	0.04
$1-\phi$	confidence limit	0.99
α	constant in eq.(3.14)	0.1
$\mathcal{R}_{th}$	threshold for evaporation	0.90

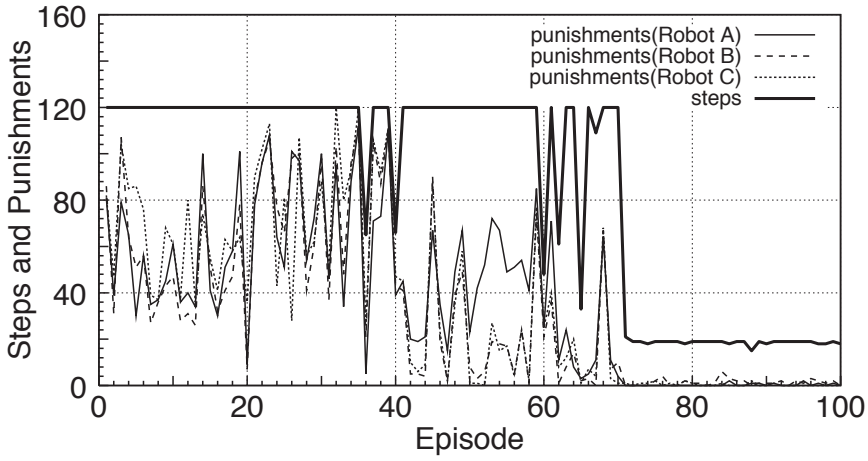
3.4.3 大域的秩序獲得実験：実験 1

本論文で目的とする環境変化に対する頑健性の検証に先立つ予備実験として，拡張型 BRL を用いた協調荷上げ行動獲得実験を行う．なお，BRL の学習は経験に基づく実行可能解 (局所解) に収束するために試行によって獲得する行動は異なる．ここで示す結果は複数行った実験のなかの一例である．

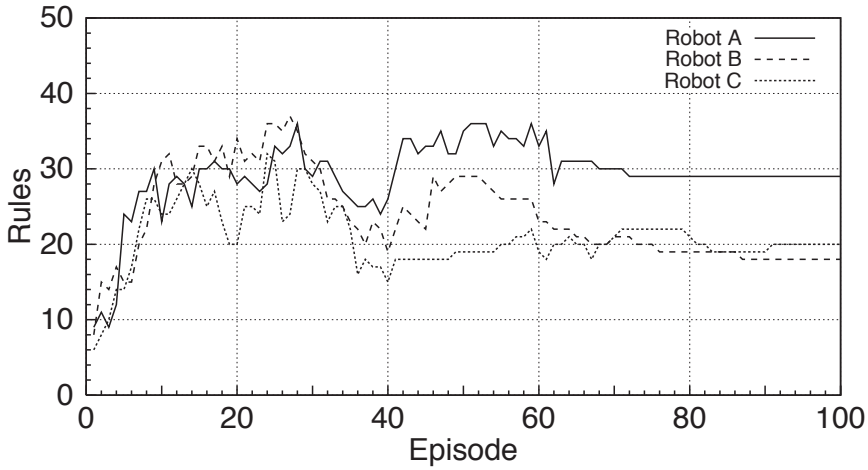
学習履歴

Fig. 3.6 に学習履歴を示す．Fig. 3.6(a) は各エピソードにおいて終了までに要したステップ数[1]と各ロボットが罰を受けた回数である．実験開始後，タスクを達成できず，多くの罰を受けるエピソードが続く．その後，荷を傾けずにゴールする行動を獲得している．Fig. 3.6(b) は各エピソード終了時にロボットが保持しているルール数である．実験初期は探索が多く行われているために多くのルールを生成し，保持するルールが徐々に増えている．学習収束後，従来型 BRL のように発火しているルール以外は徐々に削除されることなく，ルールが保護されて削除されるルールは少ない．すなわち，ルール構成の進行度に基づく学習過程における有効ルールの保護により，ルール集合に多様性が維持できている．

[1]非同期の制御のために各ロボットで若干異なる．この図では，ロボット A のもののみを示す．



(a) Transitions of time steps and punishments



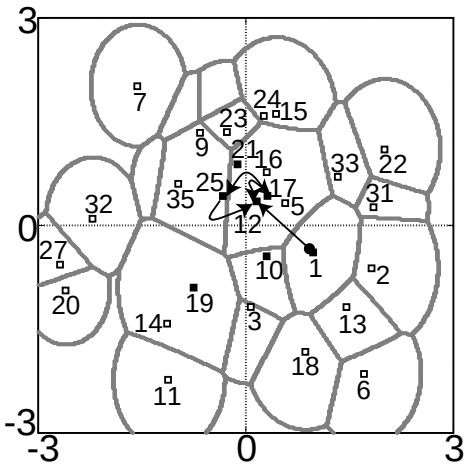
(b) Transitions of numbers of rules

Fig. 3.6: Learning history for Experiment-1

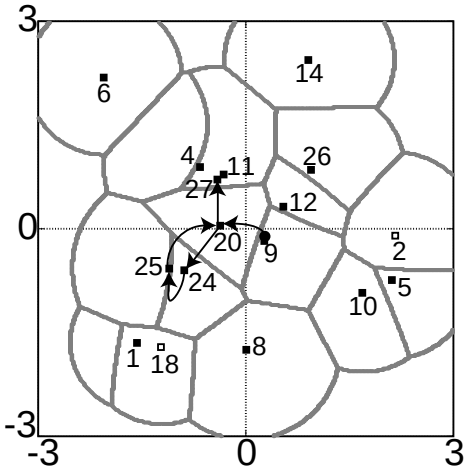
状態空間の構成

各ロボットは自身の経験に基づいて状態・行動空間を自律的に分割し，その違いにより異なる役割を果たすようになる．ここでは，状態空間を可視化するために，各学習器が記憶しているルールの特徴ベクトルに関して相関行列による主成分分析を行い，第一・第二主成分からなる二次元状態空間を生成する．各ルールの特徴ベクトルと分散共分散行列を新たに生成した状態空間に射影し，状態空間を観測する．

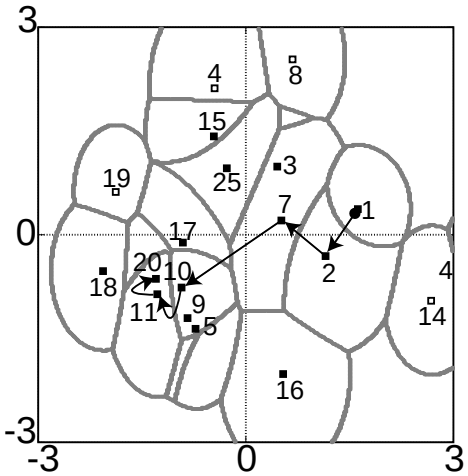
Fig. 3.7 に 100 エピソードにおける各ロボットの状態空間を示す．図中の四角は獲得されたルール，矢印はルールの発火系列を表す．白四角は提案手法によって保護されたルールである (従来の BRL では消散の作用により削除される)．ロボット毎に異なる数・粒度のルールで状態空間を分割していることがわかる．さらに，発火ルール以外にもルールを削除せずに保護することで，状態空間を広く覆っていることがわかる．



(a) Robot A



(b) Robot B

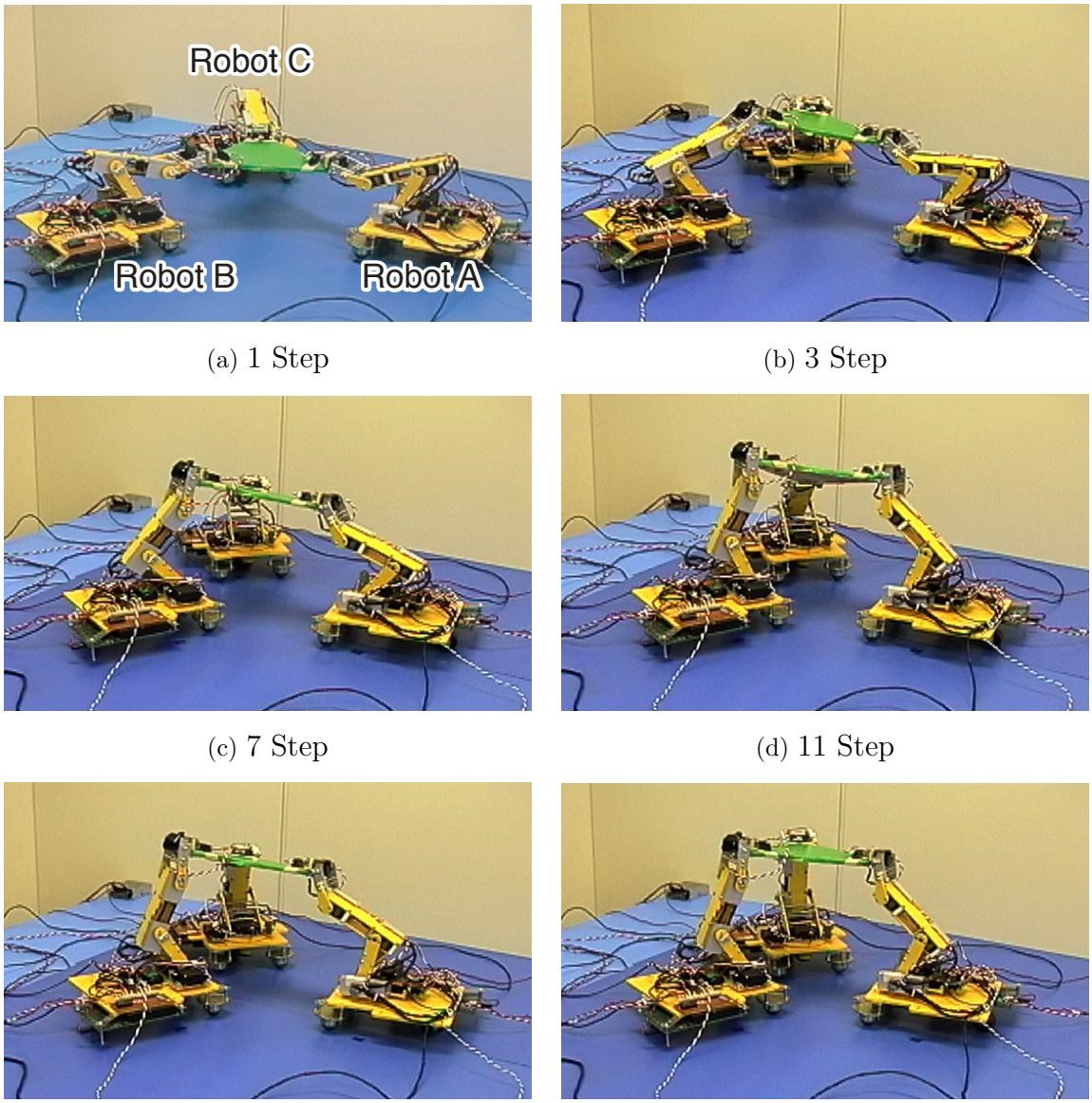


(c) Robot C

Fig. 3.7: Projected segmented state spaces of the three robots for Experiment-1 in the 100th eposode

獲得した振る舞いの観測

前節に記述した状態空間の構成に基づいて，各ロボットは状況に応じて動作を切り替え (機能分化)，荷をしきい値以上に傾けずに持ち上げる協調行動を実現する．Fig. 3.8 に 100 エピソードにおけるシステムの振る舞いを示す．Fig. 3.9 は，100 エピソードにおける各ステップで観測したセンサ値を用いて，ロボットの姿勢を再現したもの，および発火ルールの番号である．(ただし，センサ値はノイズを含むために，厳密には実際の姿勢と一致しない．また，非同期であるために，横軸はロボットによって異なる．) 各ロボットの振る舞いは，以下の通りである．



(a) 1 Step　(b) 3 Step

(c) 7 Step　(d) 11 Step

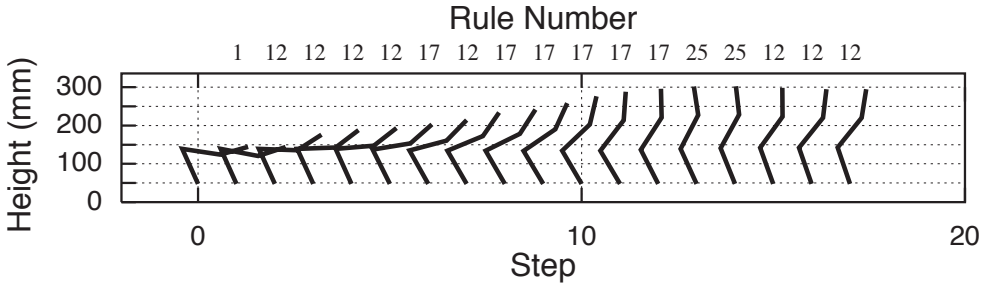
(e) 15 Step　(f) 18 Step

Fig. 3.8: Acquired behavior for Experiment-1 in the 100th episode

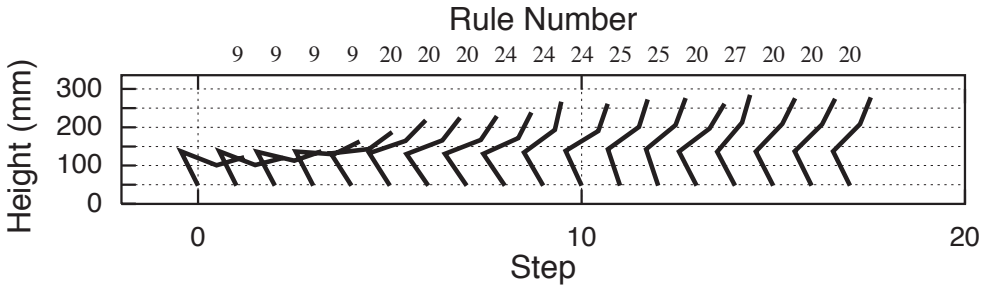
ロボット A：まず，ロボット B に次ぐ早さで荷を持ち上げる．その後，いったん荷を下ろす．最終的に，ロボット C が高く持ち上げた後に持ち上げを再開してタスク達成に至る．どちらかのロボットの傾きに応じて行動を変更する**調整**役を果たしているといえる．

ロボット B：前半はアームを大きく動かしながら持ち上げた後，持ち上げる速度を落とす．最終的には高さを維持する．常に持ち上げを先導し，**リーダ**役を果たしているといえる．

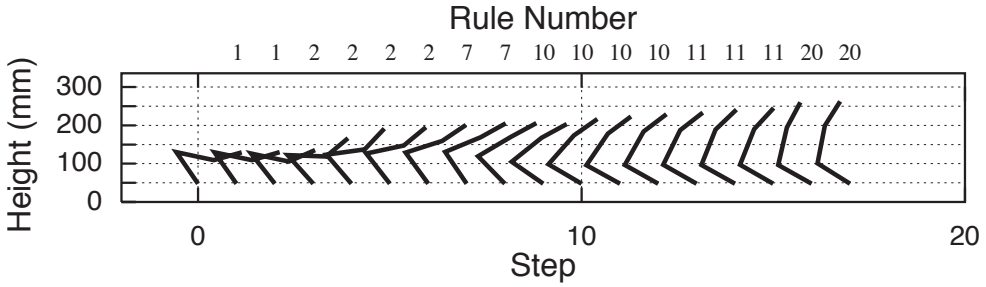
ロボット C：後半まで少しずつ持ち上げ，ロボット A，ロボット B がある高さまで持ち上げた後に持ち上げる速度を上げており，**フォロワ**役を果たしているといえる．



(a) Robot A



(b) Robot B



(c) Robot C

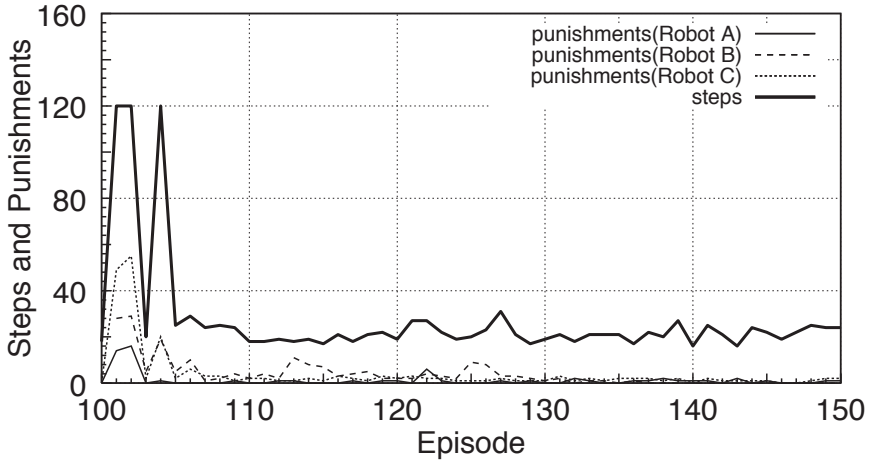
Fig. 3.9: Postures of the robots for Experiment-1 in the 100th episode

3.4.4 システムの頑健性の検証実験：実験 2

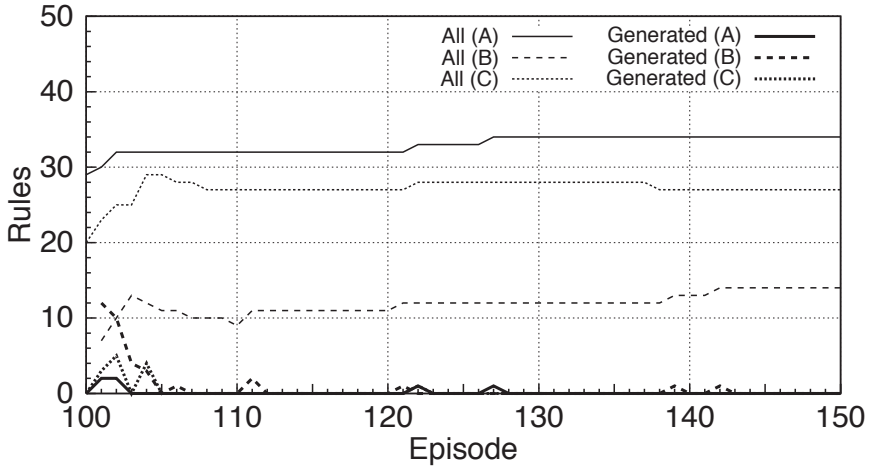
提案手法による環境変動に対する頑健性の向上を検証するため，安定な行動を獲得後に一台のロボットを初期化する．なお，予備実験から調整役 (ロボット A) とフォロワ役 (ロボット C) を初期化した場合は，従来型 BRL でも安定に行動を獲得できることを確認している．ここでは従来型 BRL では過学習の影響が最も大きかったリーダ役のロボット B を初期化した実験について詳細を述べる．

学習履歴

Fig. 3.10 に拡張型 BRL を用いた場合の学習履歴を示す．Fig. 3.10(a) は各エピソードでエピソード終了までに要したステップ数と各ロボットが罰を受けた回数である．



(a) Transitions of time steps and punishments



(b) Transitions of numbers of rules

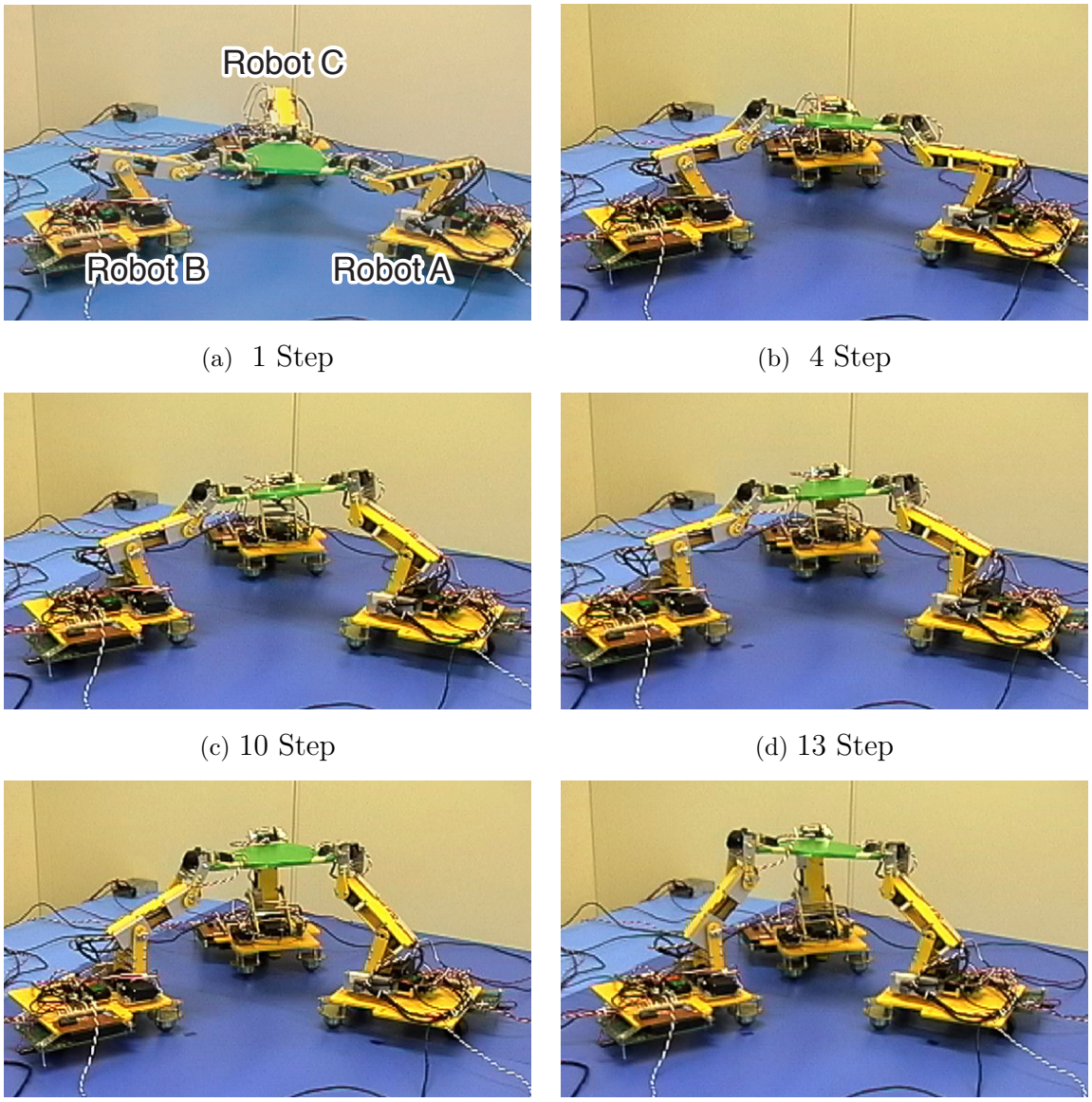
Fig. 3.10: Learning history for Experiment-2

図が示すように，環境変化後にシステムの振舞いは一時的に不安定になるものの，その後は安定してタスクを達成している．また，Fig. 3.10(b) は各ロボットの BRL におけるエピソード終了時の保持ルール数とエピソード毎の生成ルール数の推移である．ロボット A と C はロボット B のランダムな行動に合わせて新しい協調行動を獲得するためにルールを生成するが，ロボット B はそれまでの知識がないためにより多くのルールを生成する．再び安定な行動を獲得した後もノイズなどの影響によって若干のルール生成があるものの，ここでもルールの保護の影響により，ルール数の増減は小さい．

獲得した振る舞いの観測

Fig. 3.11 に 150 エピソードにおけるシステムの振る舞いを示す．まずロボット A，ロボット B が持ち上げ (Fig. 3.11(b))，ロボット A はさらに持ち上げ続ける (Fig. 3.11(c))．ロボット A が持ち上げの速度を落としている間に，ロボット B，ロボット C が持ち上げる (Fig. 3.11(d))．その後，ロボット A とロボット C がゴール位置まで持ち上げ (Fig. 3.11(e))，ロボット B が最後に持ち上げを完了する (Fig. 3.11(f))．ロボット A がリーダ役を，ロボット B と C がフォロワ役を果たしている．これは，実験 1 とは異なる機能分化の形態である．

次に，Fig. 3.12 に，このときの各ロボットの姿勢と発火ルールの番号を示す．図中の括弧なし，中括弧付き，および大括弧付きのルールはそれぞれ 100 エピソードで発火していたルール，提案手法によって保護されていたルール，新たに生成したルールを表す．ロボット A とロボット C は，保護ルールの利用と新ルールの生成によって 100 エピソードとは明らかに異なる行動を取っている．このことから，システム変動に合わせて適応的に行動を調整していることがわかる．



(a) 1 Step (b) 4 Step

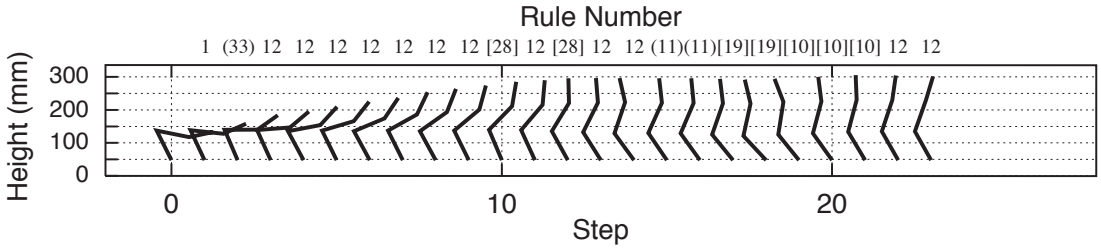
(c) 10 Step (d) 13 Step

(e) 19 Step (f) 24 Step

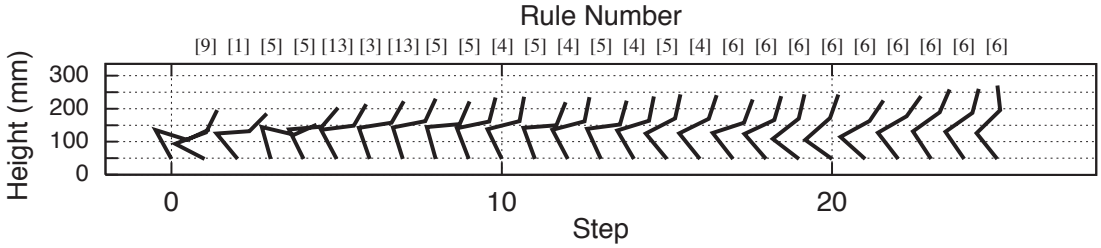
Fig. 3.11: Acquired behavior for Experiment-2 in the 150th episode

3.4.5 比較実験

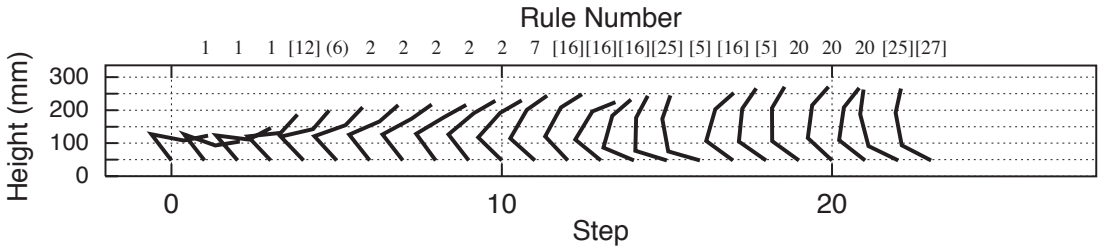
従来型 BRL を用いて比較実験を行う．前節の実験 2 と同様に，101 エピソードにおいてロボット B の学習結果が初期化されるという環境変化に対して，システムの振舞いを解析する．ここで，比較のために獲得した協調行動を同じものにする必要があるため，従来型 BRL で獲得したルールは拡張型 BRL の実験 1 において獲得されたルール群から提案手法により保護されていたルール (Fig. 3.7 における白四角) を削除したものとして仮想的に再現している．従来型 BRL と提案型 BRL では，学習途中に関してはほぼ差異はなく，安定してタスク達成が可能になった後 (実験 1 の 71 エピソード



(a) Robot A



(b) Robot B



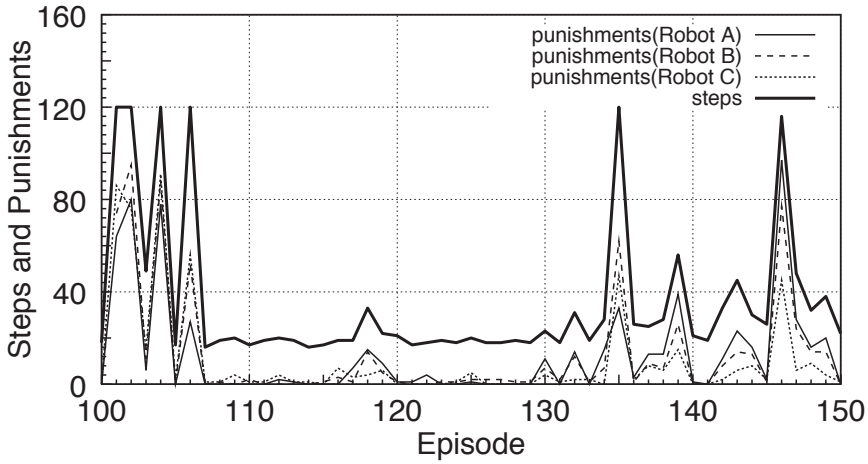
(c) Robot C

Fig. 3.12: Postures of the robots for Experiment-2 in the 150th episode

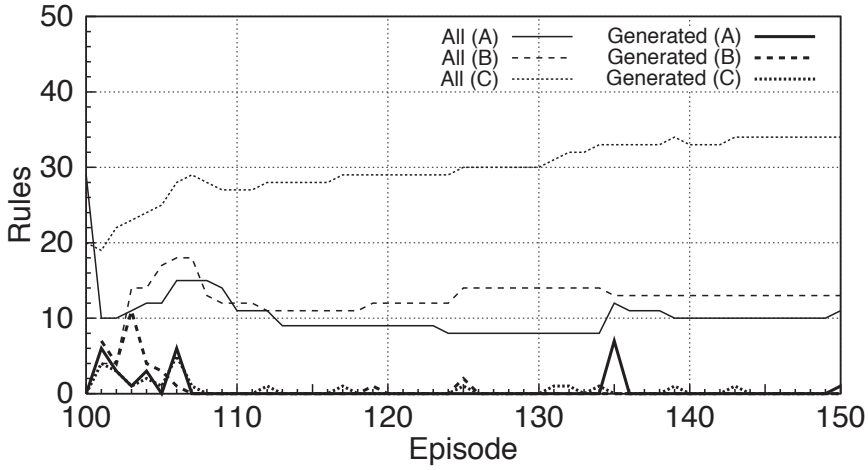
以降) のルール集合の構成に関して影響を与えるのみであるため，この操作による不利益はない.

学習履歴

Fig. 3.13 に学習履歴を示す．Fig. 3.13(a) は各エピソードでエピソード終了までに要したステップ数と各ロボットが罰を受けた回数である．環境変化後，システムは不安定になり，再び安定化するまでに拡張型 BRL よりも多くのエピソードを要していることがわかる．また，罰を受けながらタスクを達成しているエピソードが存在するため，ルール集合の構成も徐々に変化することでしばらくして振舞いが不安定になり，タスクを達成できない場合もある．Fig. 3.13(b) は各ロボットの BRL におけるエピソード終了時の保持ルール数とエピソード毎の生成ルール数の推移である．どのロボットも環境変化後に，罰状態に陥ることで新しいルールを生成して適応しようとす



(a) Transitions of time steps and punishments



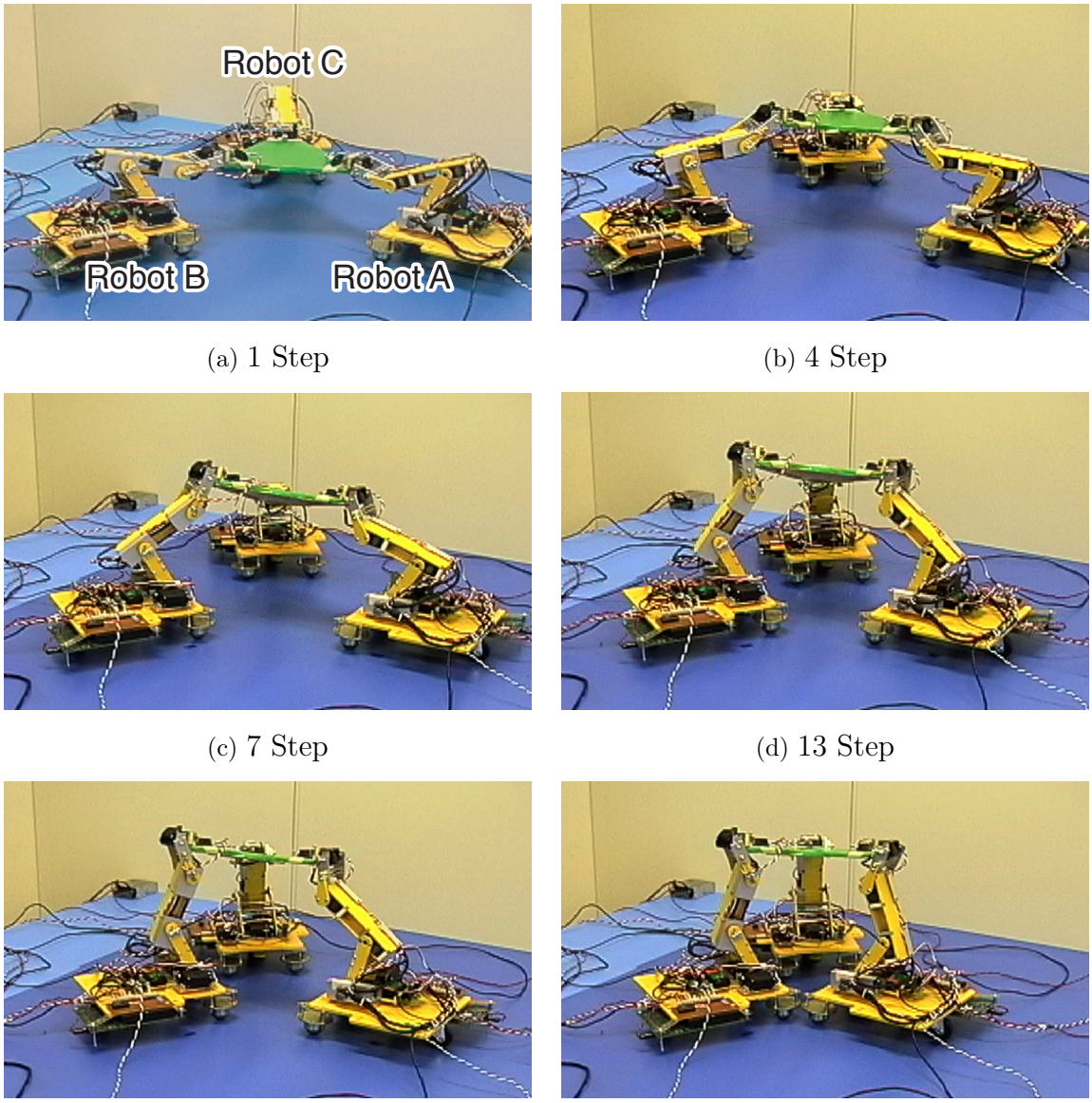
(b) Transitions of numbers of rules

Fig. 3.13: Learning history: the standard BRL

る．一旦，安定してタスクを達成するようになるとルール生成が行われず，消散の作用によって徐々に保持ルール数が減少する．しかし，その後のエピソードで罰を受けることで徐々にルールが増加していく傾向がみてとれるように，拡張型 BRL と比較してルール数の増減が大きい．

獲得した振る舞いの観測

Fig. 3.14 に 150 エピソードにおけるシステムの振る舞いを示す．Fig. 3.15 は，150 エピソードにおけるロボットの姿勢と発火しているルールの番号である．新しいルールを生成するなどの相違がわずかにあるものの，ロボット A とロボット C は 100 エピソードとほぼ同様のルールを用いている．すなわち，過学習状態に陥ったロボット A



(a) 1 Step (b) 4 Step

(c) 7 Step (d) 13 Step

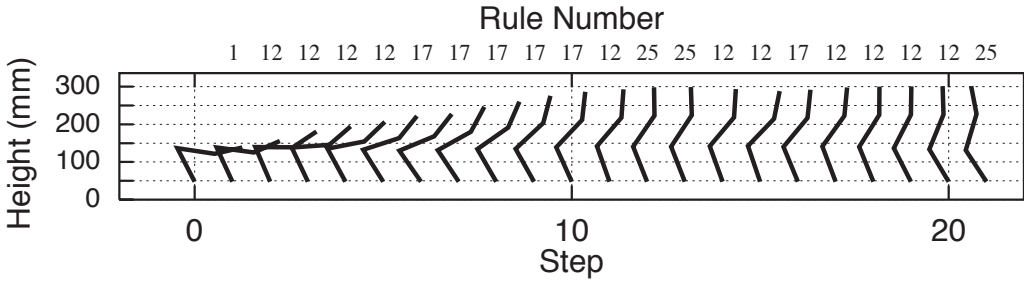
(e) 17 Step (f) 22 Step

Fig. 3.14: Acquired behavior for Experiment-1 in the 100th episode

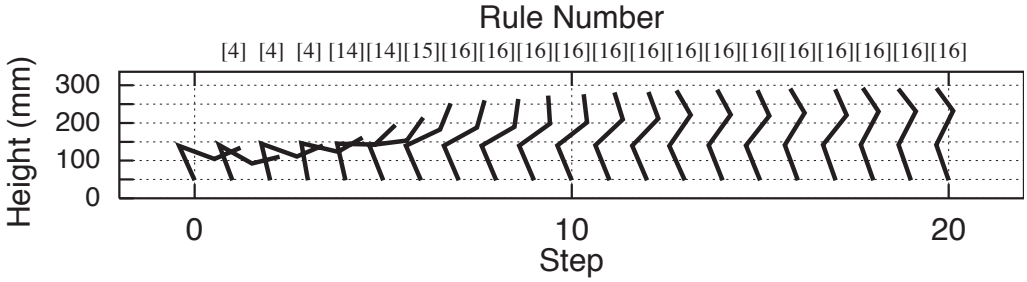
とロボット C に合わせるような行動をロボット B が学習したといえる．

3.4.6 まとめ

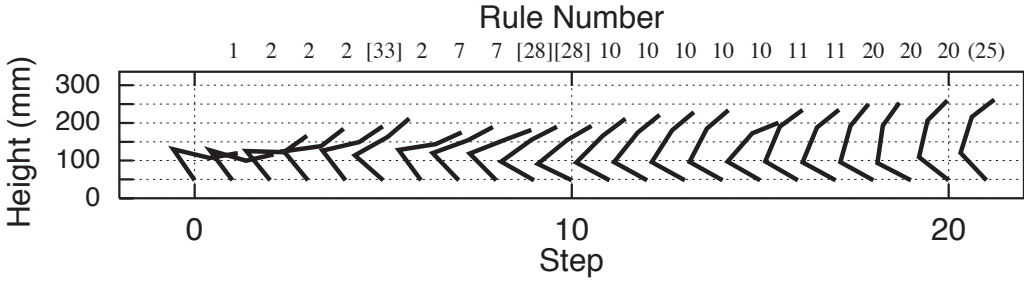
BRL は適応的に状態空間の分割を更新する機能を持つため，ある程度の環境変化に対してはシステムの安定性を維持できる．しかし，実験を繰り返すことで過学習が生じた場合，従来型 BRL では頑健性が低下する．この問題に対処するため，分散共分散行列に基づいて学習過程の有効ルールを保護するという手法を提案した．提案型 BRL は従来型 BRL と比較してより安定な行動を迅速に獲得することを実験的に確認



(a) Robot A



(b) Robot B



(c) Robot C

Fig. 3.15: Postures of the robots for the standard BRL in the 150th episode

した．このことから，提案手法は過学習を抑制することで MRS の可塑的な自律的機能分化を可能にし，システムの頑健性を向上させているといえる．

3.5 結言

本章では，MRS における頑健な自律的機能分化を実現するための強化学習の拡張法を提案した．まず，状態・行動空間の離散化の困難性，および連続空間を取り扱う強化学習の研究例について述べた．次に，連続な状態行動空間を自律的に分割する機能を持つ強化学習法・BRL の位置付けを明確にすると共に，アルゴリズムの詳細を示した．その後，過学習を抑制してシステムの頑健性を向上させるために学習過程で有効であったルールを保護することでルール集合の多様性を維持する機構を付加した．そ

の有効性を検証するために，三台のアーム型自律ロボットによる協調荷上げ問題に適用して実機実験を行った．その結果，頑健性が向上し，環境変化に対して従来型 BRL よりも安定した大域的秩序を迅速に再形成していることを確認した．

第4章　ダイナミクスの軽減による学習の安定化

4.1　緒言

前章では，連続空間を自律的に分割する強化学習である BRL の頑健性を向上させるため，ルール集合の多様性を維持するという拡張法を提案し，その有効性を示した．そこでは，内部状態のオンライン更新による汎化能力によって，MRS が持つダイナミクスに対処することで，協調行動の獲得を実現していた．本章では，BRL を用いた MRS の協調行動獲得において，学習をより安定化させるための手法として，次時刻における他ロボットの状態を予測する機構を付加する．提案手法を自律移動ロボットによる協調搬送問題に適用し，その有効性を検証する．

4.2　マルチロボット強化学習の困難性

本来，強化学習は静的 (マルコフ的) な環境で有効性が保証されているのみであり，MRS に適用する場合のその困難性が報告されている [67][130]．MRS において強化学習を運用する際に直面する問題は，問題空間が大規模化することのみならず，システムが動的な環境であること，すなわちマルコフ性が成立しないことが一因である．例えば，Fig. 4.1 のような二台のロボットの衝突回避を考えた場合，左側のロボットが時刻 t の状態入力を基に行動しても，衝突回避できるか否かはもう一方 (右側) のロボットの行動に左右される．本来，強化学習は静的な環境の下で，すなわちマルコフ決定過程の下における無限試行により収束が保証されている．それゆえに，“他のロボット” という動的要素が存在する MRS においては，基本的に学習の収束は保証されないことになる．しかし，第 3 章では，連続な状態・行動空間を自律的に分割する BRL を拡張し，アーム型ロボットの頑健な協調行動の獲得を実現している．このように強化学習はある程度の環境中のダイナミクスを許容するために，MRS への適用に成功した例が他にも存在することから [64][66]，環境ダイナミクスを軽減できれば，より安定的に行動獲得が可能になるといえる．

そのような観点から，学習を安定させるためにさまざまなアプローチが提案されている．Tan[131] は，情報を共有することの効果を検証している．そして，共有情報の有効性は状況依存であり，適切に行われれば有効であると述べている．Asada *et al.*[132]

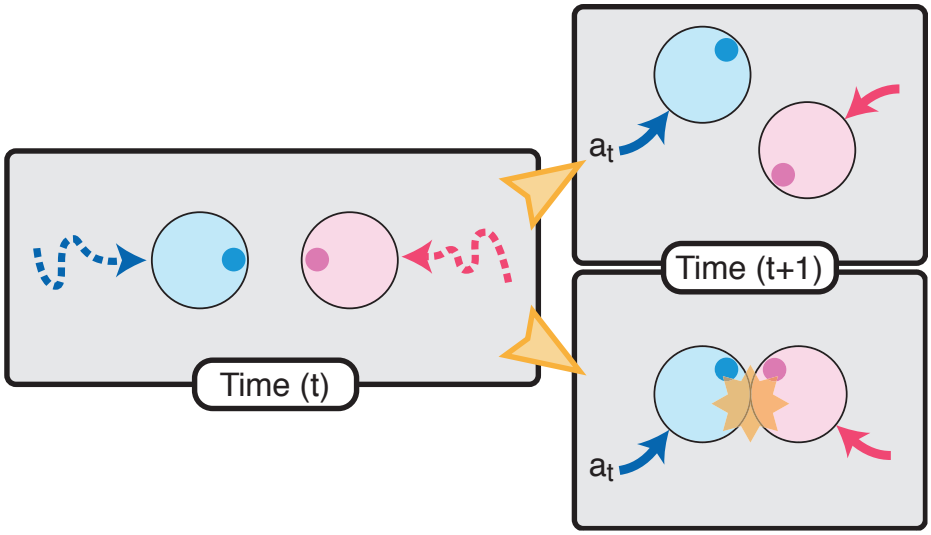


Fig. 4.1: An Example of Dynamic Environment: Collision Avoidance

と Ikenoue *et al.*[133] は，二台のロボット (シューター，パサー) によるサッカー環境における視覚に基づく強化学習法を提案している．Asada *et al.* は学習過程を安定化させるために，学習するロボットと過去に獲得した固定政策を用いるロボットに分けるスケジューリングをおこなっている．これを拡張したものとして，Ikenoue *et al.* は非同期で政策の更新を行う手法を提案している．Buffet *et al.*[134] は箱押しタスクにおいて，箱の数や初期状態での距離を変化させることで，簡単な問題から徐々に難しくする段階的な学習法を提案している．Elfwing *et al.*[135] は，マクロアクションを付加する手法を提案している．マクロアクションとは一定期間継続して実行される行動であり，それにより他ロボットに取っては動きの予測が簡単になる．Matarić[136] は複数ゴールに関する強化を扱うための非均質な強化関数を各ロボットに持たせるとともに，特定ゴールに対する進展度を推定するアプローチを提案している．また，他ロボットに関する情報を推定することで，安定した意思決定を行う手法がある．Littman[137] は行動，Hu and Wellman[138] は Q 値，Nagayuki *et al.*[139] は政策の推定を行う手法を提案している．さらに，他ロボットに関する情報を確率学習オートマトンで予測する手法 [140] などが提案されている．

4.3 予測を用いた環境ダイナミクスの軽減

本研究では，均質な MRS が自律的機能分化するための手法の構築を対象としている．そのため，汎用的なアプローチを取る必要があるという立場からあらかじめ通信様式，役割，行動などを与えることなく協調行動の獲得を目指す．また，実環境という連続な空間における運用を考えると，相互作用の複雑性から，他ロボットの内部状

態に関する推定を行うことは困難といえる．そこで，実際に観測するセンサ入力に関する時系列情報を利用したアプローチである，川上ら [140] の手法に着目する．この手法では，状態遷移をある程度の長さをもった時系列単位で認識するメモリーベース法 [68] や決定木 [69] によるものと異なり，別の学習器で予測機構を構築することで強化学習器の扱う問題空間を拡大することなく，他ロボットに関する時系列情報を取り扱うことができる利点もある．予測機構が出力する次時刻の他ロボットの姿勢のみを強化学習器の入力として用いることで，行動獲得を安定的に行えることが実験的に示されている．

しかし，この手法で用いられている強化学習器と予測機構は，いずれも離散的な状態・行動空間を取り扱ったものであるために，連続空間に拡張する必要がある．ここで，

- 予測の良否は次ステップには判明することから，毎時間ステップにおける評価が可能である．学習問題としては逐次評価型となる．
- 予測対象となる時系列に対してある程度速い適応が必要となる．

という問題環境の特徴から，予測機構をニューラルネットワークで構成する．構造はフィードフォワード型とし，次時刻のセンサ入力に基づいて予測誤差を常に知ることができるということから，誤差逆伝播法を用いてニューロン間の結合荷重の更新を行う．ここで，出力関数にはシグモイド関数を用いるものとする．

提案する制御器は次のようにまとめられる．他ロボットの影響を受けるセンサの時系列データに基づいて，ニューラルネットワークによるパターン予測を行う．そして，その予測値を BRL の入力の一部として付加する (Fig. 4.2)．これによって，BRL 自身が時系列データを取り扱うことによる状態空間の拡大を防ぐことができる．

4.4 協調搬送問題による検証

4.4.1 問題設定

ロボットが荷に回転自由に連結された状態でゴールまで到達する問題を取り上げる．荷の形状として，二台の場合は直線の棒，三台の場合は正三角形の板を想定する (Fig. 4.3)．連結部分を通して, 他のロボットのモータ出力に伴う力が伝わるため，タスク達成のためにはロボットは協調的に振舞わなければならない．

この問題は，通常のフィードバック制御手法が適用できない非ホロノミック問題のひとつである．太田ら [141] は仮想インピーダンス法，小菅ら [142] はコンプライアンス制御を用いて，同様のシステムにおける効率的な目標地点までの移動を実現している．これらの研究では，ロボットの役割 (リーダ・フォロワ) はあらかじめ割り当てられており，ゴールへ先導するリーダに対し，荷との連結部分の負荷を基にフォロワがいかに追従するかに焦点を当てている．計算知能のアプローチでは，あらかじめ

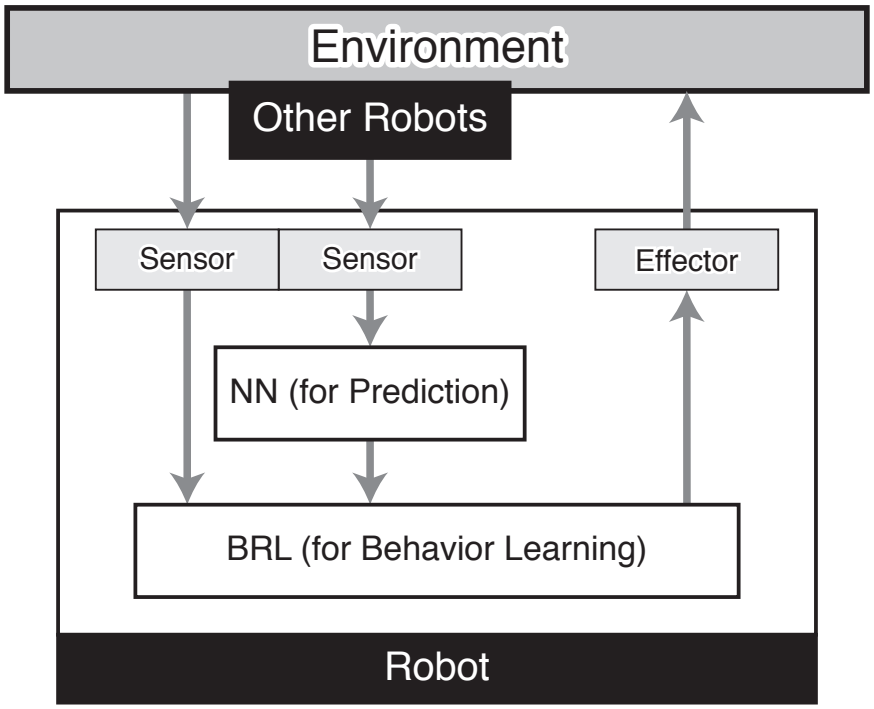
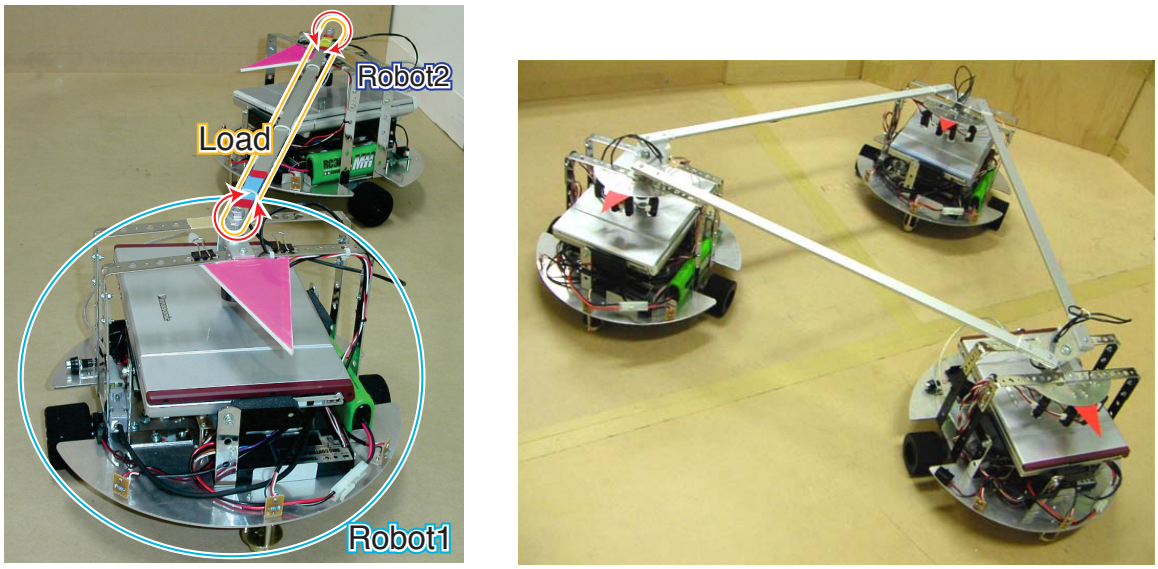


Fig. 4.2: Our proposed controller



(a) Two robots (b) Three robots

Fig. 4.3: Cooperative carrying problem

モジュール化した行動パターンの運用をファジー制御によって行っている Ghanea *et al.*[143] の研究がある．Masek *et al.*[144] らは GA を用いてパスプランニングを行う手法を提案している．

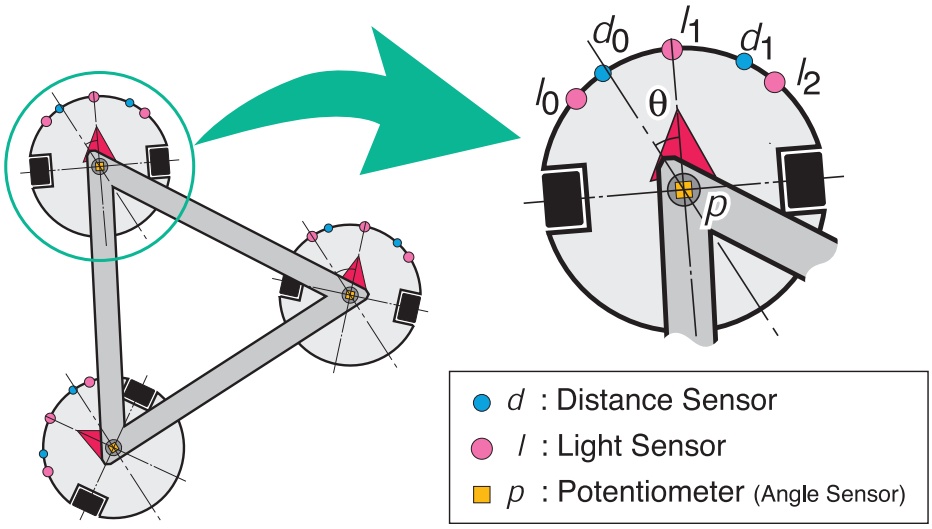


Fig. 4.4: Sensor Layout

ロボットは独立の制御器で制御される．Fig. 4.4 に示すように，ロボットは二個の車輪を持ち，壁との距離を測定する測距センサを二個，光の強さを測定する光センサを三個，連結部分に各ロボットの角度を測定する角度センサを持つ (ロボットは全ロボットの連結部分に対する角度を知覚できる)．ロボットの中心間の距離はロボットの直径の二倍である．ロボット間の明示的な通信は用いないため，各ロボットは角度センサの値からのみ他ロボットの状態 (連結部分に対する角度) を知覚する．移動においては，ロボットの向きと移動速度が大きな影響を持つが，このうちのロボットの向きしか知覚できないことになる．また，他のロボットが衝突状態にあることも直接知覚することができないため，いずれかのロボットが壁に衝突した場合には，デッドロック状態に陥りやすく，そこから抜け出すことは困難である．

4.4.2 大域的秩序獲得実験：計算機実験 1

実験設定

三台による協調搬送問題の計算機実験の環境を Fig. 4.5 に示す．タスクは左上のスタート地点から右下のゴールエリア (光源) までの移動である．ここで，各ロボットの初期位置は固定であるが，初期角度は図右を中心にしてランダムに $\pm 10^\circ$ の範囲で変動させる．ゴールに到達すると全ロボットに報酬が与えられ，ロボットが壁に衝突した (測距センサの値が閾値を越えたとき) ときに衝突したロボットのみに罰が与えられる[1]．センサ入力を得て行動し，評価を得るまでを単位ステップとする．ゴールに到達

[1]衝突状態が続いているときは罰を与えない．これは，衝突状態から抜け出すための行動が罰を受けることで学習が進まないことに対する処置である．

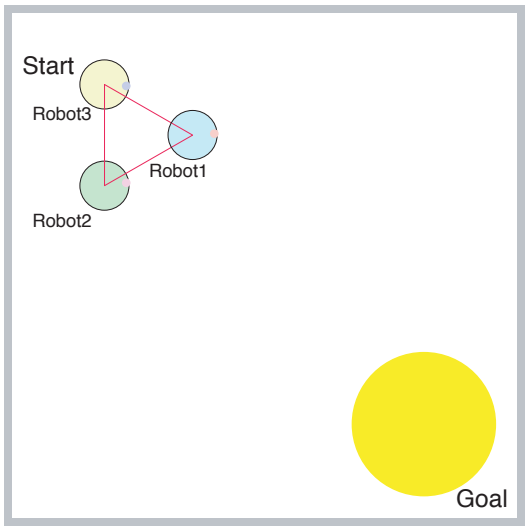


Fig. 4.5: Experimental Environment

するか，ゴールに到達せずに 300 ステップが経過するまでを単位エピソードとする．

測距センサの有効範囲はロボットの直径と等しく，光センサの有効範囲はロボットの直径の 1.2 倍である．本実験では，他ロボットの結合部分に配置された角度センサの値を予測し，その予測値を BRL の入力に付加する．ここで，ロボットの数が増えるにつれて BRL が取り扱う状態空間が増加することへの処置として, 予測する他ロボットの角度センサ値は，それぞれを単独で取り扱うのではなく二つの平均値を用いるものとする．つまり，各ロボットは他のロボット全体として向いている角度を知覚することになる．角度センサ値は BRL は [0.0, 1.0] に正規化した状態入力を用いることと，角度情報を連続的に表現する必要があるために三角関数 (正弦，余弦) で変換したものを用いる．以下に，予測機構 (ニューラルネットワーク) と BRL の設定を示す．

■ 予測機構の設定　第 4 章で行った実験と同一の設定である．三層構造のフィードフォワードニューラルネットワークで構築する．入力は現在から過去 2 ステップまでの計 3 ステップ間の自他の角度センサ値であり，出力は次時刻の他ロボットの角度センサ値である．上記の通り，角度センサ値はそれぞれ三角関数で変換したものを用いる．つまり，入力は $I = \{ \cos\theta^i_{t-2}, \sin\theta^i_{t-2}, \cos\psi^i_{t-2}, \sin\psi^i_{t-2}, \cos\theta^i_{t-1}, \sin\theta^i_{t-1}, \cos\psi^i_{t-1}, \sin\psi^i_{t-1}, \cos\theta^i_t, \sin\theta^i_t, \cos\psi^i_t, \sin\psi^i_t\}$ であり，出力は $O = \{\cos\psi^i_{t+1}, \sin\psi^i_{t+1}\}$ である．ここで，$\psi^i_t = (\theta^j_t + \theta^k_t)/2$ とする $(i \neq j \neq k)$．中間層のニューロンは 8 個である．学習則には誤差逆伝播法を用い，学習率は 0.8，結合荷重の変化項に付加した慣性項のモーメント係数は 0.9 とした．

■ BRL の設定　入力は $\boldsymbol{x} = \{ \cos\theta^i_t, \sin\theta^i_t, \cos\psi^i_{t+1}, \sin\psi^i_{t+1}, d^i_0, d^i_1, l^i_0, l^i_1, l^i_2 \}$ であり，出力は $\boldsymbol{a} = \{m^i_{rud}, m^i_{th}\}$ である．ここで，m^i_{rud} と m^i_{th} はそれぞれ，モータのステ

アリング量・スロットル量である．ゴールに到達すると全ロボットに報酬が与えられ，ロボットが壁に衝突した (測距センサの値がしきい値を越えたとき) ときに衝突したロボットのみに罰が与えられる．BRL の各パラメータは第 3 章で示した推奨値に基づく．ただし，協調荷上げ問題と比べて自由度が高いために，状態空間のパラメータの更新量が大きいと，振る舞いの不安定化につながるため，(3.14) 式の $\alpha = 0.001$，(3.15) 式の $\beta = 0.0001$ とした．

実験結果

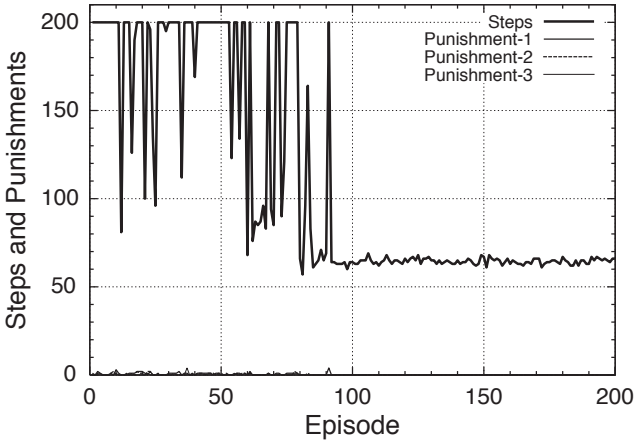
Fig. 4.6 に計算機実験で得られた結果を示す．Fig. 4.6(a) は各エピソードでのゴール到達までのステップ数と与えられた罰の数である．92 エピソードまでは試行錯誤の過程で壁に衝突して動けなくなるなど罰を受けながら学習を続ける．その後は初期姿勢のランダム性によってステップ数に若干の変動があるものの連続してゴールしており，安定した協調行動を獲得しているといえる．Fig. 4.6(b) は各ロボットの BRL の全ルール数と生成したルール数である．学習初期では多くのルールが生成されているものの，行動学習後は新ルールの生成が抑制されると共に不要なルールが削除され，保持するルール数が減っていることがわかる．Fig. 4.6(c) はニューラルネットワークによる予測の平均二乗誤差である．学習初期では予測誤差の変動が大きいが，行動が安定してからは予測の誤差も徐々に小さくなり約 15°–30° に収束している．このことから，予測機構の学習が収束していることがわかる．

次に，Fig. 4.7 に実験中のロボットの振舞いを示す．学習初期はランダムに探索をするため，壁に衝突してデッドロック状態に陥ることが多い (Fig. 4.7(a))．その後，試行錯誤を繰り返すうちにゴールに到達する行動を獲得した (Fig. 4.7(b))．200 エピソードでは，Table 4.1 に示す状態ベクトルと出力を持つルールを，Fig. 4.8 のような系列・タイミングで発火している．Table 4.1 において，全ての入出力は [0.0 ,1.0] に正規化されたものである．$m^i_{rud}(< 0.5)$ が小さくなるほどロボットの左車輪が右車輪と比べて速く回転 (ロボットは大きく左回転) し，$m^i_{rud}(> 0.5)$ が大きくなるほど右車輪が左車輪と比べて速く回転 (ロボットは大きく右回転) する．m^i_{th} が大きいほど速く移動し，小さいほど移動は遅くなる．

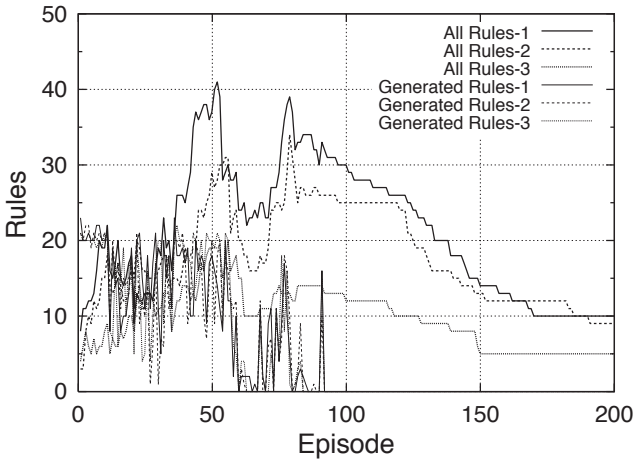
振る舞いを観測すると，三つの領域に分割することができた．以下に，それぞれの領域における各ロボットの振る舞いを示す．

領域 1：スタート付近 (1–30 エピソード) 全てのロボットは異なるタイミングでルールを切替えて右に曲がる．特に，Robot1 が最も頻繁にルールを切替えて進行方向の調整を行う．Robot3 はもっぱらひとつのルールのみを用いる．

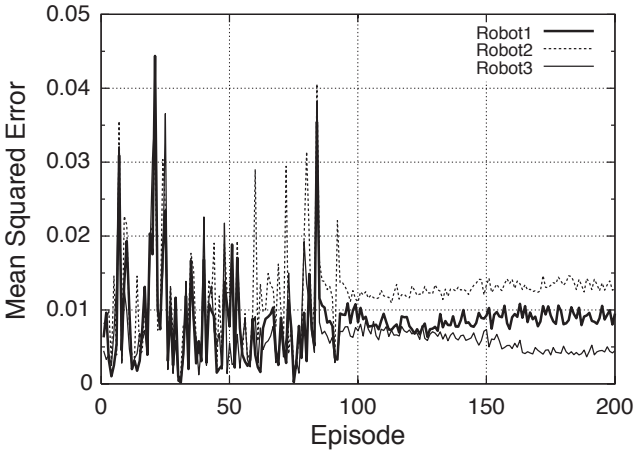
領域 2：環境中央付近 (30–55 エピソード) 全てのロボットが頻繁にルールの切替えを行う．



(a) Learning history

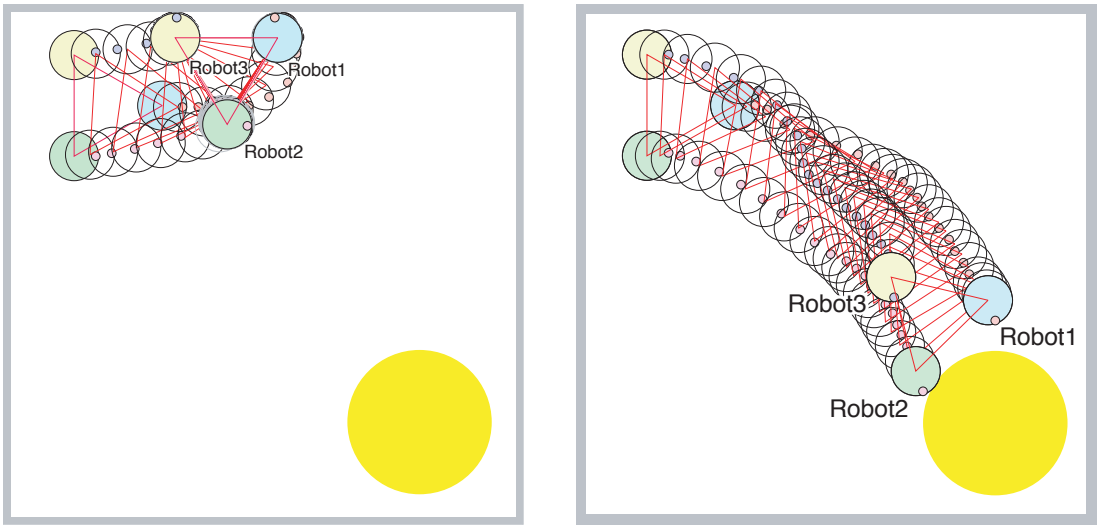


(b) Number of rules



(c) Prediction error

Fig. 4.6: Learning result



(a) 1st episode (b) 200th episode

Fig. 4.7: Behavior

Table. 4.1: Firing rules after successful learning

Robot-Rule number	$\boldsymbol{v}$: State vector									$\boldsymbol{a}$: Action	
	$\cos\theta$	$\sin\theta$	$\cos\psi$	$\sin\psi$	d_0	d_1	l_0	l_1	l_2	m_{rud}	m_{th}
1-1	0.991	0.590	0.877	0.164	0.001	0.001	0.200	0.200	0.200	0.008	0.891
1-4	0.397	0.012	0.729	0.093	0.001	0.001	0.200	0.200	0.200	0.710	0.961
1-7	0.662	0.027	0.682	0.067	0.001	0.001	0.200	0.002	0.029	0.568	0.224
1-23	0.967	0.325	0.952	0.375	0.001	0.001	0.255	0.200	0.200	0.653	0.925
1-38	0.874	0.170	0.732	0.038	0.001	0.001	0.200	0.200	0.200	0.065	0.807
2-9	0.841	0.134	0.686	0.045	0.001	0.001	0.200	0.008	0.051	0.818	0.966
2-11	0.867	0.161	0.550	0.086	0.001	0.001	0.346	0.377	0.405	0.183	0.841
2-12	0.905	0.207	0.926	0.249	0.001	0.001	0.001	0.128	0.253	0.238	0.584
2-13	0.993	0.574	0.922	0.234	0.001	0.001	0.200	0.200	0.200	0.097	0.605
2-17	0.783	0.088	0.678	0.122	0.001	0.001	0.692	0.713	0.541	0.016	0.720
2-24	0.997	0.449	0.548	0.097	0.001	0.001	0.200	0.200	0.200	0.055	0.865
3-1	0.658	0.028	0.740	0.090	0.001	0.001	0.200	0.200	0.200	0.090	0.673
3-4	0.982	0.371	0.945	0.254	0.001	0.001	0.200	0.200	0.200	0.191	0.920
3-11	0.312	0.038	0.781	0.035	0.001	0.001	0.200	0.200	0.200	0.857	0.756

領域 3：ゴール付近 (55–65 エピソード) Robot1 と Robot2 がゴールに向かって進み，Robot3 は進行方向を調整するように頻繁にルールを切替える．

以上のように，この実験で獲得した振る舞いを観測することで，**ゴールへ先導する**，**進行方向を調整する**，**同じ行動を繰り返して先導する**といった機能分化が状況応じてなされていることがわかった．このような機能分化は，第 3 章のアーム型ロボットの協調荷上げ問題と同様に状態空間の自律的分割に基づいて発現する．状態空間の粒度を調べるため，連続して発火するルール間のユークリッド距離を求めたものを Table

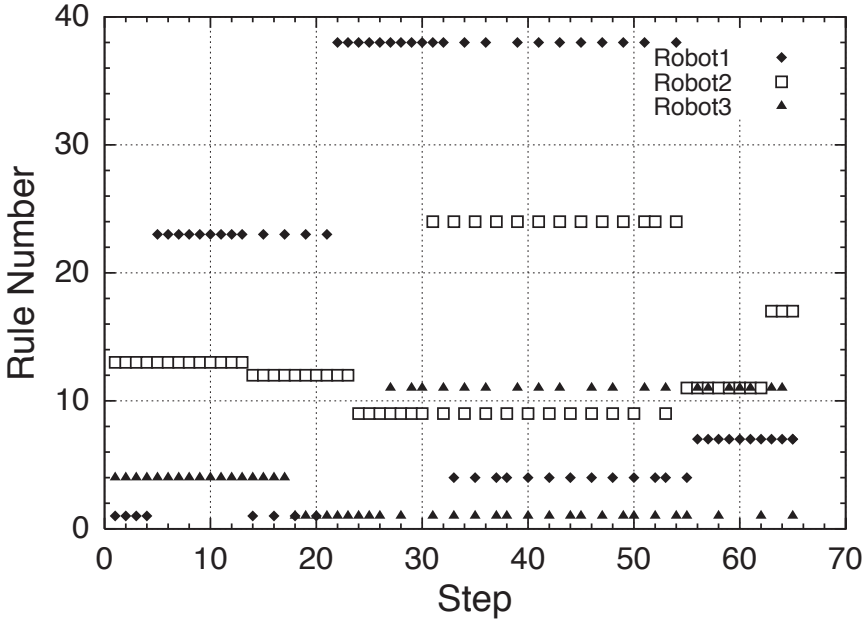


Fig. 4.8: Sequence of firing rules

Table. 4.2: Euclidean distance between rules in the 200th episode (simulated robots)

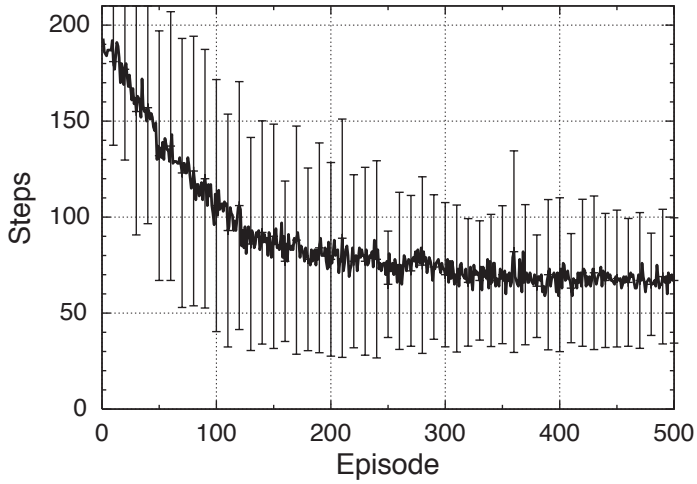
Robot number	Rule number	Euclidean distance			
		$\boldsymbol{v}$	θ	ψ	l
1	$1 \leftrightarrow 23$	0.352	0.266	0.224	0.055
	$23 \rightarrow 38$	0.446	0.181	0.401	0.055
	$38 \leftrightarrow 4$	0.505	0.502	0.055	0.000
	$4 \rightarrow 7$	0.377	0.265	0.054	0.262
2	$13 \rightarrow 11$	0.436	0.377	0.016	0.218
	$12 \rightarrow 9$	0.451	0.097	0.315	0.308
	$9 \leftrightarrow 24$	0.452	0.352	0.147	0.243
	$24 \rightarrow 11$	0.441	0.316	0.011	0.308
	$11 \rightarrow 17$	0.530	0.111	0.133	0.501
3	$4 \rightarrow 1$	0.540	0.471	0.263	0.000
	$11 \leftrightarrow 1$	0.353	0.346	0.069	0.000

4.2 に示す．ゴールに先導するとき (Robot1：4→7．Robot2：24→11，11→17)，θ と ψ に関して細かい分割であり，l は粗く分割している．これにより光に向かう行動が実現できている．進行方向の調整を行うとき，ψ は θ と比べて細かい分割である．つまり，他ロボットに関して敏感であるといえる (Robot1：1↔23，38↔4．Robot2：9↔24．Robot3：11↔1)．同じ行動を繰り返すとき (Robot3：4→1)，角度調整を行うときよりも全体的に粗い分割である．これにより，ルールの切替えを行わないともいえる．

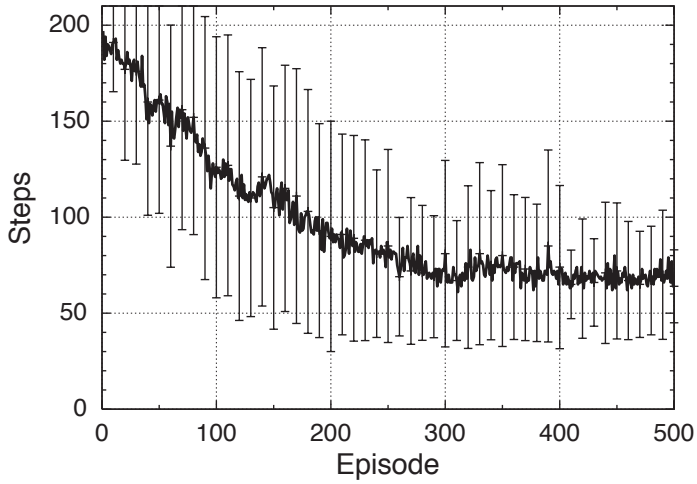
4.4.3 比較実験：計算機実験 2

ここでは，予測が行動獲得にいかに貢献しているかを検証する．比較対象は，予測機構を持たない BRL のみの制御器とする．この制御器は，他ロボットの角度情報に関しては現時刻のセンサ入力をそのまま用いることになる．前述の通り，予測を用いないこの制御器であっても，ダイナミクスをある程度許容するために行動獲得が可能である．そのため，予測機構を付加することで，学習がいかに効率的になるかという観点からの比較を行う．なお，ここではそれぞれ制御器に関して 50 試行の実験を行う．100 エピソード連続でゴールしたときを行動獲得に成功したとみなす．

Fig. 4.9 に各エピソードで要したステップ数の平均と分散を示す．実験初期におけるステップ数の傾きが予測を用いた場合の方が大きい．このことから，予測を用いる



(a) BRL with NN



(b) BRL without NN

Fig. 4.9: Learning history: number of steps (average for 50 runs)

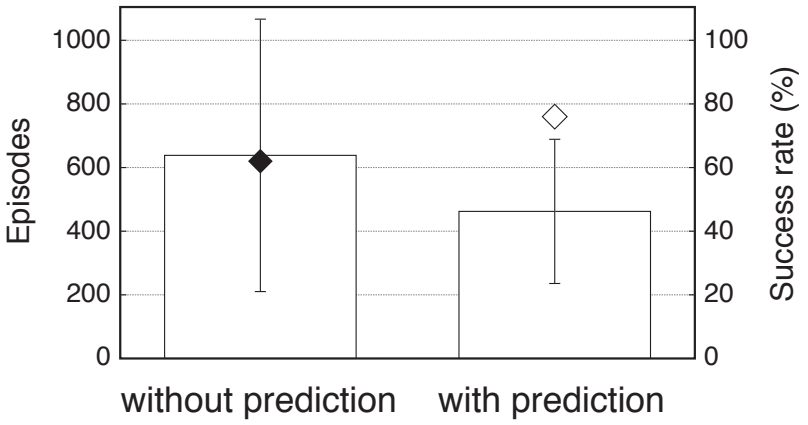


Fig. 4.10: Episodes required for successful learning and success rates

ことで行動獲得が迅速に行えているといえる．Fig. 4.10 は行動獲得までに要したエピソード数の平均・分散と，学習の成功確率である．この図においては，提案手法の行動獲得に必要なエピソード数を基準として実験を打ち切り，その時点で行動が獲得できていれば成功とみなしている．タスク達成可能な行動を学習するまでに要するエピソード数は，予測を用いた場合と用いない場合のそれぞれに関して 462.26，634.38 エピソードであった．さらに，実験を 460 エピソードで打ち切った場合，予測を用いた場合は 38 試行において学習に成功し，用いない場合の成功数は 31 であった．これらの結果から，ニューラルネットワークを用いた他ロボットに関する予測情報を BRL に付加する手法の有効性が示されたといえる．

4.4.4 頑健性の検証実験：計算機実験 3

第 3 章のアーム型ロボットを用いた実験では，環境変動として行動収束後に一台のロボットの初期化を行った．この操作は，ロボットが故障したときに代わりとなる新しいロボットを投入するのに相当する．本節では，システムの頑健性の検証をするための環境変化として，ロボットの追加と削除を行う．これにより，MRS が持つ利点の一つである，拡張性に関する検証が行えるという側面もある．以下，追加と削除の実験の典型例について述べる．

環境変化前の振る舞い

この実験では，BRL のパラメータは前節のものとほぼ同一であるが，これまでの状態空間を覆うガウス分布の幅を表す $\sigma = 0.1$ と変更した．これは各ロボットが多くのルールを生成しないための処理であり，各ロボットはエピソード内での役割の切替えを抑制するためである．それにより，ロボットが削除された場合のシステムの挙動を

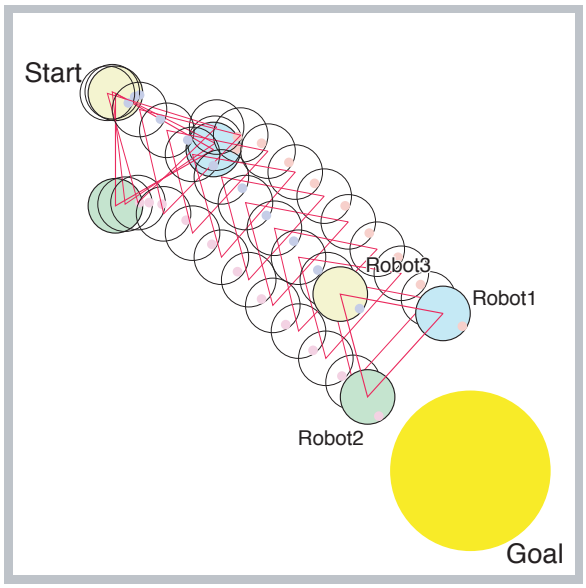


Fig. 4.11: Behavior in the 200th Episode

Table. 4.3: Firing Rules in the 200th Episode

Robot-Rule number	$\boldsymbol{v}$: State vector									$\boldsymbol{a}$: Action	
	$\cos\theta$	$\sin\theta$	$\cos\psi$	$\sin\psi$	d_0	d_1	l_0	l_1	l_2	m_l	m_r
1-2	0.971	0.327	0.801	0.111	0.001	0.001	0.202	0.083	0.094	0.090	0.429
1-30	0.709	0.039	0.751	0.262	0.001	0.001	0.204	0.217	0.190	0.884	0.162
2-3	0.999	0.546	0.872	0.175	0.001	0.001	0.209	0.205	0.215	0.453	0.746
2-4	0.846	0.159	0.830	0.088	0.001	0.001	0.217	0.183	0.181	0.251	0.830
2-7	0.699	0.044	0.863	0.109	0.001	0.001	0.495	0.502	0.207	0.136	0.962
2-8	0.600	0.009	0.881	0.125	0.001	0.001	0.670	0.572	0.570	0.340	0.290
2-42	0.452	0.004	0.838	0.182	0.001	0.001	0.185	0.194	0.212	0.764	0.508
3-1	0.996	0.444	0.629	0.093	0.001	0.001	0.182	0.212	0.210	0.047	0.202
3-3	0.681	0.041	0.773	0.184	0.001	0.001	0.213	0.183	0.182	0.921	0.987

調べることで，削除されたロボットがタスク達成に対していかに貢献していたかを知ることができる．

Fig. 4.11 は 200 エピソードにおける振舞いである．このときに発火しているルールのパラメータを Table 4.3 に示す．この実験では，BRL の出力は左右のモータの回転速度 m_l と m_r である．

ロボットは，それぞれ

- **Robot1**:
  (2↔30)
- **Robot2**:
  3→4→(42↔4)→7→8

- **Robot3**:
  1→3

というようにルールの切替えを行う．Robot1 はエピソードを通してルールを頻繁に切替え，Robot2 はゆっくりと切替える．Robot3 ほとんどルールの切替えを行わない．以上のように，三台の場合は Robot1 が進行方向を調整 (リーダ)，Robot2 は始めは補助的に角度調整・ゴール付近ではゴールへ先導 (サブリーダ・リーダ)，Robot3 は他の二台が導く方向に押していく (フォロワ) といった役割分担を行っているといえる．このような協調行動を獲得した後，201 エピソードにおいてロボット数を変化させる．

ロボットの追加

この実験では，荷の形状は正方形とし，他ロボットに関する角度情報は隣り合う二台の平均値を知覚できるものとした．ロボットを追加した直後の 3 エピソードはゴールに到達できないが，その後は追加前とほぼ同様のステップ数でゴールに到達している (Fig. 4.12)．このときに発火しているルールのパラメータを Table 4.4 に示す．各ロボットの振る舞いは次の通りである．

- **Robot1**:
  (2↔30)

- **Robot2**:
  3→(4↔42)→7→9”→8

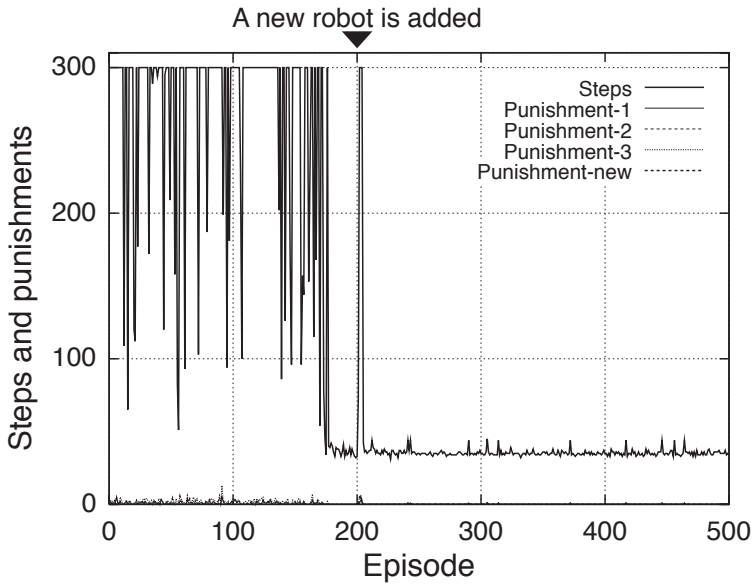
Table. 4.4: Firing Rules in the 500th Episode (Addition)

Robot-Rule number	$\boldsymbol{v}$: State vector									$\boldsymbol{a}$: Action	
	$\cos\theta$	$\sin\theta$	$\cos\psi$	$\sin\psi$	d_0	d_1	l_0	l_1	l_2	m_l	m_r
1-2	0.973	0.341	0.831	0.141	0.001	0.001	0.202	0.083	0.094	0.090	0.429
1-30	0.735	0.060	0.768	0.237	0.001	0.001	0.203	0.213	0.194	0.884	0.162
2-3	0.998	0.535	0.871	0.177	0.001	0.001	0.207	0.204	0.213	0.453	0.746
2-4	0.840	0.150	0.838	0.112	0.001	0.001	0.210	0.189	0.188	0.251	0.830
2-7	0.696	0.043	0.863	0.112	0.001	0.001	0.496	0.504	0.207	0.136	0.962
2-8	0.599	0.009	0.879	0.126	0.001	0.001	0.667	0.575	0.569	0.340	0.290
2-9”	0.689	0.032	0.811	0.119	0.001	0.001	0.261	0.505	0.753	0.411	0.672
2-42	0.492	0.008	0.833	0.168	0.001	0.001	0.189	0.195	0.209	0.764	0.508
3-1	0.996	0.455	0.671	0.103	0.001	0.001	0.187	0.208	0.207	0.047	0.202
3-3”	0.785	0.079	0.882	0.286	0.001	0.001	0.213	0.207	0.202	0.918	0.828
4-3	0.998	0.493	0.855	0.116	0.001	0.001	0.191	0.200	0.201	0.121	0.839
4-4	0.777	0.100	0.849	0.151	0.001	0.001	0.207	0.205	0.207	0.377	0.975
4-5	0.831	0.111	0.837	0.120	0.001	0.001	0.357	0.627	0.770	0.064	0.592

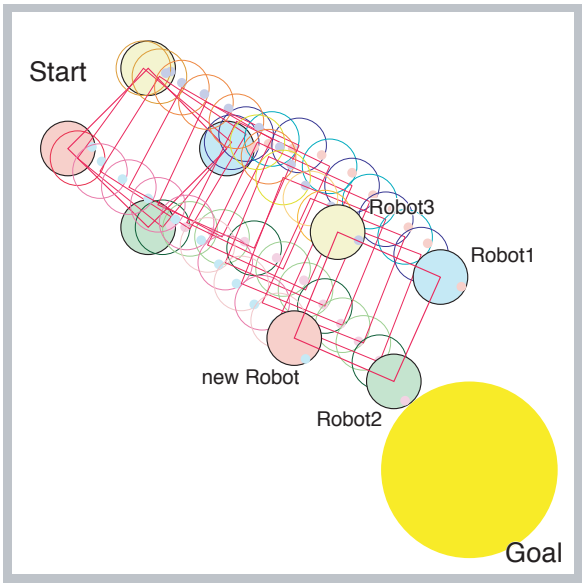
- **Robot3**:
  1→3”

- **new Robot**:
  3→(4↔5)

最終的に，Robot1 は行動の変化はなく，Robot2 はゴール付近で左旋回が追加され，Robot3 は右寄りの前進に加えてと新ルールの左旋回を補助的に行うようになる．新



(a) Learning History



(b) Acquired Behavior

Fig. 4.12: Result (Addition)

しいロボットはスタート直後に左旋回した後，左旋回と右回転を交互に行う．追加前と比べて，Robot1 と Robot2 はほぼ同様の役割を果たし，Robot3 は押す役割を主に果たしながらも角度変化が大きくなったときには補助的に角度調整を行う．新しいロボットは三台のロボットの動きを大きく妨げないように追従する役割を新たに獲得したといえる．

ロボットの削除

■ リーダの削除　削除前にシステムの進行方向を調整するリーダ的役割であった Robot1 を削除した場合，システムの挙動は長期にわたり不安定になる (Fig. 4.13)．このときに発火しているルールのパラメータを Table 4.5 に示す．各ロボットの振る舞いは次の通りである．

- **Robot2**:
  29”→4”→(29”↔19”)→ (23”↔9”)→43”→45”

- **Robot3**:
  2”→7”→1”→(12”↔35”)

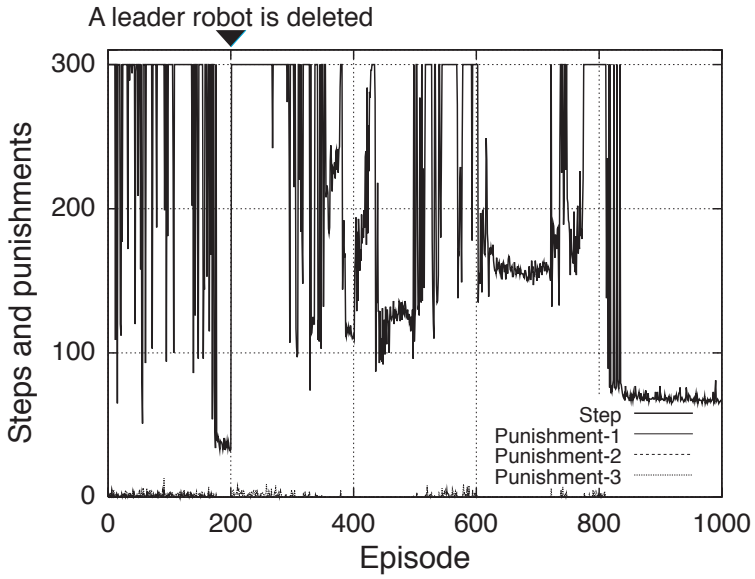
ロボットはスタート付近ではゴールを知覚していないため，ゴール付近までは角度センサ値のみから意思決定を行う．削除前にゴール付近まで進んでいた振舞いは三台のバランスによるものであり，角度調整を主導的に行っていた Robot1 が削除されることでスタート付近で進行方向が定まらなくなる．最終的に，削除前に保持していたルールが全てなくなり，全く新しいルールを獲得することで安定してタスクを達成するようになった．Robot2 はゴールを知覚するまでは左右旋回を交互に行って角度を

Table. 4.5: Firing Rules in the 1000th Episode (Deletion: Leader)

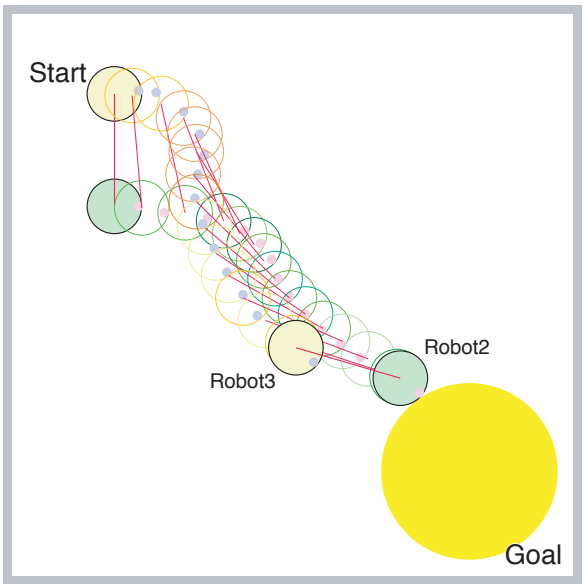
Robot-Rule number	$\boldsymbol{v}$: State vector									$\boldsymbol{a}$: Action	
	$\cos\theta$	$\sin\theta$	$\cos\psi$	$\sin\psi$	d_0	d_1	l_0	l_1	l_2	m_l	m_r
2-4”	0.585	0.990	0.856	0.923	0.001	0.001	0.211	0.202	0.197	0.998	0.980
2-9”	0.915	0.213	0.710	0.125	0.001	0.202	0.200	0.196	0.211	0.904	0.019
2-19”	0.984	0.617	0.899	0.508	0.001	0.001	0.189	0.185	0.191	0.646	0.313
2-23”	0.958	0.692	0.754	0.160	0.001	0.006	0.207	0.206	0.196	0.179	0.880
2-29”	0.747	0.938	0.919	0.415	0.001	0.001	0.205	0.214	0.178	0.013	1.000
2-43”	0.925	0.250	0.856	0.253	0.001	0.001	0.338	0.412	0.384	0.894	0.812
2-45”	0.949	0.283	0.926	0.197	0.001	0.001	0.381	0.581	0.684	0.296	0.654
3-1”	0.913	0.776	0.872	0.845	0.001	0.001	0.195	0.206	0.214	0.379	0.799
3-2”	0.500	1.000	0.957	0.262	0.001	0.001	0.208	0.213	0.207	0.719	0.154
3-7”	0.480	0.998	0.903	0.699	0.001	0.001	0.204	0.194	0.188	0.422	0.921
3-12”	0.991	0.452	0.954	0.425	0.001	0.002	0.190	0.201	0.214	0.493	0.881
3-35”	0.878	0.192	0.935	0.309	0.001	0.001	0.195	0.198	0.202	0.869	0.602

調整し，ゴール付近ではゴールへ先導するように左旋回を行う．Robot3 はゴール付近までは Robot2 を押すように進むが，ゴール付近では左右旋回を交互に行う．

なお，Robot2(サブリーダ) を削除した場合も削除後にシステムの挙動は不安定になるものの，400 エピソード付近で学習が収束する．このことからも，削除前は Robot1 がタスク達成に主導的な役割を果たしていたといえる．



(a) Learning History



(b) Acquired Behavior

Fig. 4.13: Result (Deletion: Leader)

■ フォロワの削除　削除後の 3 エピソードはタスクを達成できないが，すぐに安定した行動を獲得する (Fig. 4.14)．このときに発火しているルールのパラメータを Table 4.6 に示す．各ロボットの振る舞いは次の通りである．

- **Robot1**:
  (36'↔54')→(2↔30)→(36'↔30)→(54'↔34')

- **Robot2**:
  (4↔42)→30'→6'→9"→18"

ゴールへ先導するように角度を調整する二台のロボットを押すように行動していた Robot3 が削除されたため，タスク達成に必要なステップ数は多くなっている．Robot1 は削除前に交互に発火していたルールだけでなく，保持しているものの発火していなかったルールを用いて角度を調整する．Robot2 も同様に，削除前に交互に発火していたルールと保持はしているが発火していなかったルールを用いて角度を調整してゴール付近まで進み，ゴールを知覚した後は新たに獲得したルール (左旋回) をしてゴールに到達する．Robot1 が用いる新ルールは左右に大きく角度変化するものであり，Robot2 が用いる新ルールはほぼ前進に近い行動が多い．Robot3 が削除されたことで，ゴールへ向かう力が小さくなったので，Robot2 がそれを補うように振舞い，Robot1 は削除前より細かく角度調整をするようになる．

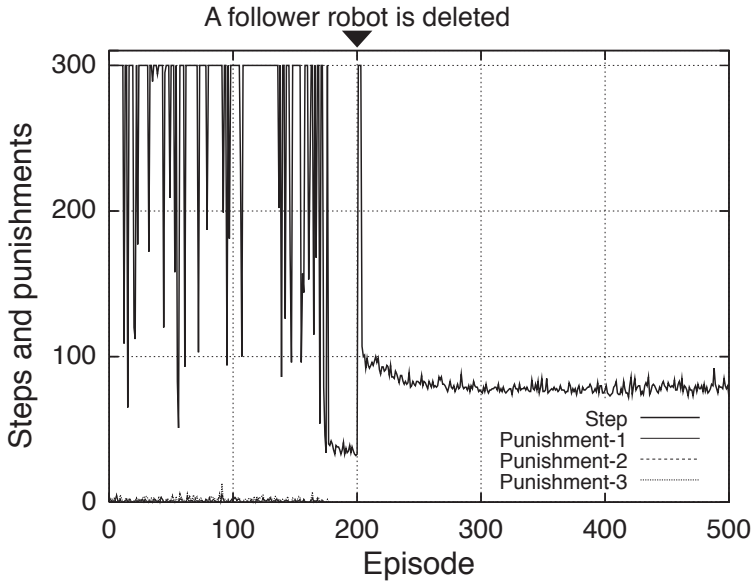
まとめ

ロボットの追加やフォロワロボットの削除を行った場合は，システムに大きな影響がなかった．つまり，ルールの発火系列をそのまま用いたり一部のみを修正することで不安定になることなくタスクを達成し続けることが可能であり，小さなレベルの環

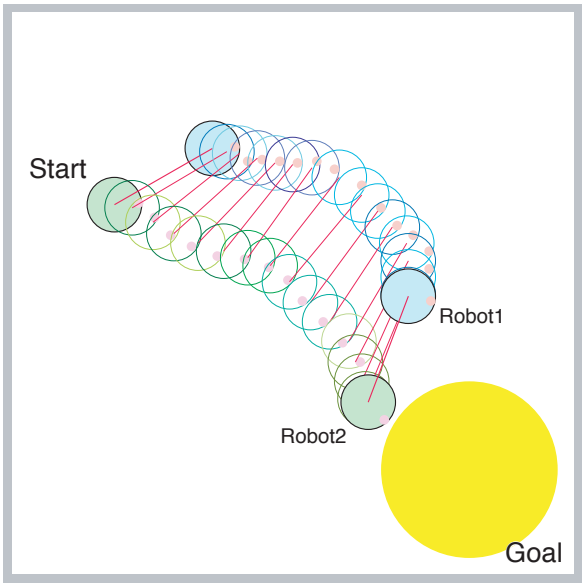
Table. 4.6: Firing Rules in the 500th Episode (Deletion: Follower)

Robot-Rule number	$\boldsymbol{v}$: State vector									$\boldsymbol{a}$: Action	
	$\cos\theta$	$\sin\theta$	$\cos\psi$	$\sin\psi$	d_0	d_1	l_0	l_1	l_2	m_l	m_r
1-2	0.971	0.327	0.805	0.112	0.001	0.001	0.202	0.095	0.105	0.090	0.429
1-30	0.709	0.039	0.751	0.262	0.001	0.001	0.204	0.217	0.190	0.884	0.162
1-34'	0.473	0.004	0.438	0.078	0.001	0.001	0.208	0.192	0.214	0.747	0.002
1-36'	0.938	0.272	0.577	0.088	0.001	0.001	0.328	0.337	0.387	0.206	0.898
1-54'	0.649	0.034	0.327	0.039	0.001	0.001	0.202	0.196	0.206	0.569	0.840
2-4	0.846	0.156	0.807	0.076	0.001	0.001	0.212	0.187	0.186	0.251	0.830
2-6'	0.225	0.088	0.313	0.131	0.026	0.001	0.198	0.147	0.235	0.754	0.784
2-9"	0.157	0.122	0.490	0.049	0.001	0.001	0.415	0.457	0.205	0.904	0.735
2-18"	0.347	0.025	0.471	0.041	0.001	0.001	0.461	0.665	0.697	0.851	0.853
2-30'	0.624	0.022	0.482	0.166	0.001	0.001	0.205	0.203	0.202	0.268	0.734
2-42	0.500	0.008	0.838	0.173	0.001	0.001	0.192	0.194	0.207	0.764	0.508

境変動に対しては頑健性を発揮することが示された．一方，リーダロボットというタスク達成に大きな役割を果たすロボットが削除された場合，システムの振る舞いは長期に渡って不安定になった．これは既存のルールが不必要になったとしても発火し続ける傾向にあり，それにより壁への衝突やデッドロック状態が発生することが原因であった．



(a) Learning History



(b) Acquired Behavior

Fig. 4.14: Result (Deletion: Follower)

4.4.5　実機実験

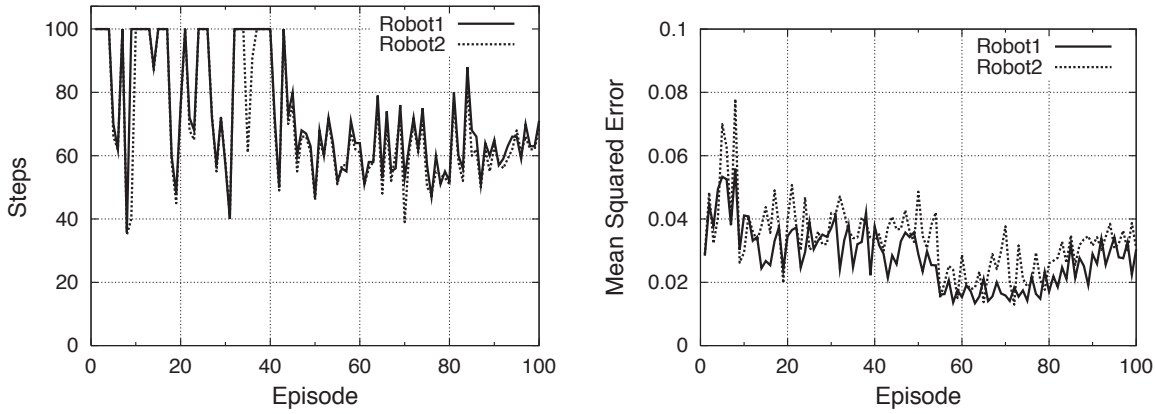
二台の場合

■ 実験設定　実験環境はスタートとゴールの間に障害物がある 2.7m×2.7m の長方形の環境である (Fig. 4.15)．右のロボットを Robot1，左のロボットを Robot2 とする．この実験では，スタート地点で各ロボットはゴールの方向を向ように手動で設置する．エピソード終了までの最大ステップ数は 100 とする．その他の設定は，計算機実験と同一である．

■ 実験結果　Fig. 4.16 に実験結果の一例を示す．Fig. 4.16(a) は，計算機実験の場合と同じく，学習開始直後は不安定でゴールに到達したりしなかったりを繰り返す．42 エピソード以降は安定してタスクを達成していることがわかる．Fig. 4.16(b) は，予測機構による予測の平均二乗誤差である．行動が安定するにしたがって変動の幅が小さくなっている．

Fig. 4.17 に獲得した協調行動を示す．ルールの発火するタイミングなどから，振る舞いは大きく三つに分けることができた．スタート付近では，Robot1 はほとんどルールを変更せずに進み，Robot2 はルールの切替えを頻繁に行う．その後，Robot2 がルールを頻繁に切替えながら進み，Robot1 は角度を調整するようにルールを切替える．それにより，障害物を避けるような進路を取っている．最終的に Robot2 が光源の方向へ進路を変更し，Robot1 はそれに追従する．このような自律的機能分化の発現によりタスク達成を実現している．

Fig. 4.15: Experimental Environment: two robots



(a) Steps (b) Mean squared error of prediction

Fig. 4.16: Learning history in a typical run: two robots

Fig. 4.17: Traces of behavior: two robots

三台の場合

■ 実験設定 実験環境は 2.7m×2.7m の長方形の環境である (Fig. 4.18)．この実験では，スタート地点で各ロボットはゴールの方向を向ように手動で設置する．エピソード終了までの最大ステップ数は 100 とする．その他の設定は，計算機実験と同一である．

■ 実験結果 Fig. 4.19 に実験結果の一例を示す．Fig. 4.19(a) は，計算機実験や二台の実機実験の場合と同じく，学習開始直後 は不安定でゴールに到達したりしなかっ

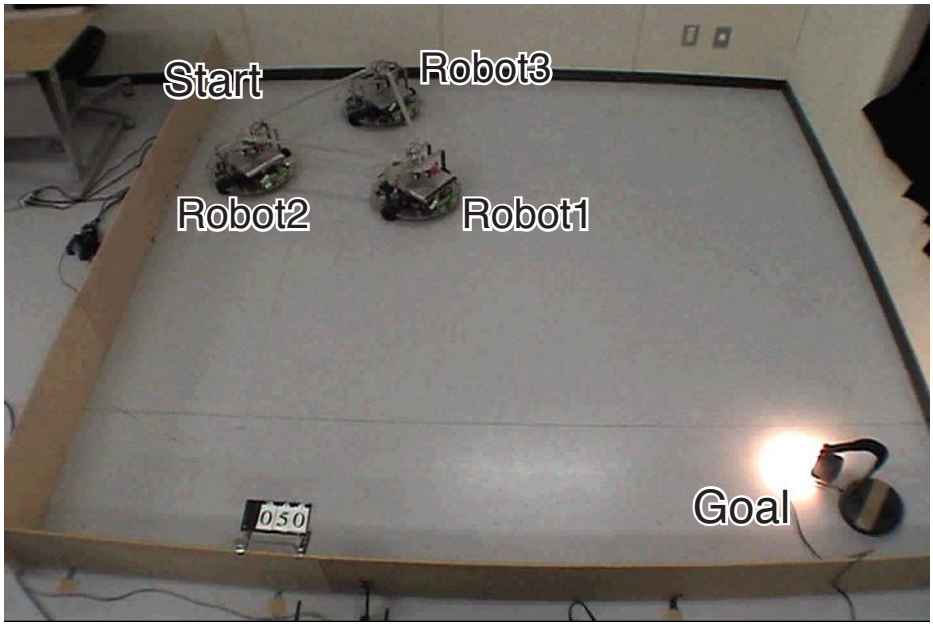
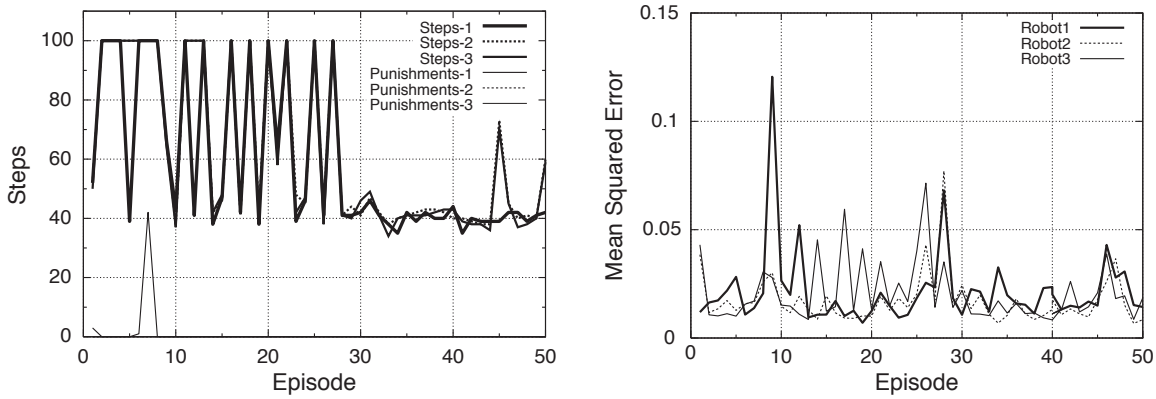


Fig. 4.18: Experimental Environment: three robots



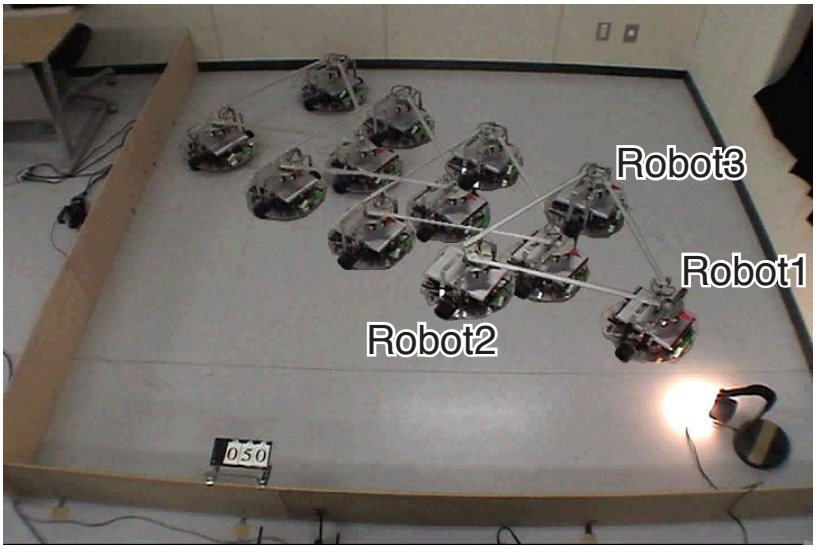
(a) Steps and Punishments

(b) Mean squared error of prediction

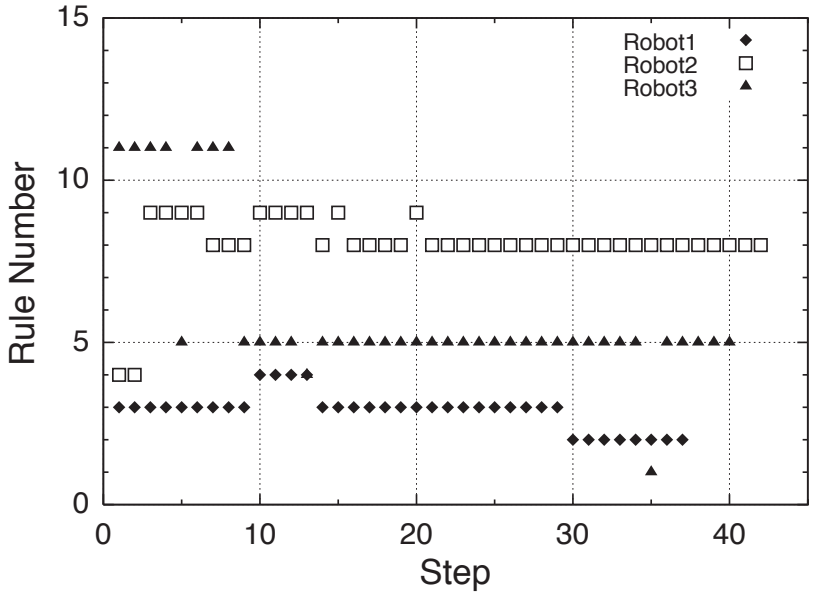
Fig. 4.19: Learning history in a typical run: three robots

たりを繰り返す．28 エピソード以降は安定してタスクを達成していることがわかる．Fig. 4.19(b) は，予測機構による予測の平均二乗誤差である．28 エピソード以前は誤差が大きい場合があるが，行動が安定するにしたがって変動の幅が小さくなっている．この問題では，ロボット三台が共にほぼ同じ方向を向き，同じ速度で移動しないと，デッドロック状態に陥ってその場に留まる傾向が見られた．学習収束前に予測誤差が小さいエピソードがみられるのは，そのようなデッドロック状態で角度センサの値がほとんど変化していなかったことが原因である．

次に，この実験における獲得した行動について述べる．スナップショットを Fig. 4.20(a),



(a) Traces of Behavior



(b) Sequence of Firing Rules

Fig. 4.20: Acquired behavior: three robots)

各ロボットの発火ルールの履歴を Fig. 4.20，また，各ロボットの保持しているルールパラメータの詳細を Table 4.7 に示す．このとき，振る舞いを観測することで，その違いからスタートからゴールに至るまでに四つの領域に分けられる．まず，第一領域 (1–約 5 ステップ：スタートから中央に至るまで) では，Robot1 は左回転，Robot2 と Robot3 は右回転を行う．第二領域 (約 5–約 20 ステップ：中央付近) では，全てのロボットは異なるタイミングでルールのスイッチングを行っている．特に Robot2 が最も頻繁に行っていることから，主導的に進行方向の調整を行っているといえる．続い

Table. 4.7: Firing rules in the 50th episode: three robots

Robot-Rule number	$\boldsymbol{v}$: State vector									$\boldsymbol{a}$: Action	
	$\cos\theta$	$\sin\theta$	$\cos\psi$	$\sin\psi$	d_0	d_1	l_0	l_1	l_2	m_{rud}	m_{th}
1-2	1.000	0.500	0.942	0.298	0.003	0.003	0.476	0.567	0.331	0.427	0.047
1-3	0.996	0.566	0.932	0.273	0.003	0.003	0.177	0.314	0.334	0.434	0.991
1-4	0.909	0.788	0.943	0.609	0.003	0.003	0.139	0.175	0.216	0.169	0.361
2-4	0.996	0.560	0.852	0.161	0.003	0.003	0.111	0.203	0.332	0.760	0.547
2-8	0.803	0.102	0.950	0.511	0.003	0.003	0.002	0.002	0.296	0.694	0.239
2-9	0.977	0.349	0.957	0.702	0.003	0.003	0.001	0.124	0.294	0.238	0.584
3-1	0.579	0.006	0.941	0.270	0.003	0.003	0.002	0.002	0.172	0.482	0.622
3-4	0.926	0.238	0.827	0.167	0.003	0.003	0.002	0.003	0.003	0.544	0.309
3-5	0.870	0.165	0.953	0.515	0.124	0.003	0.002	0.002	0.209	0.264	0.805
3-11	0.955	0.292	0.879	0.857	0.003	0.003	0.002	0.002	0.002	0.802	0.996

て，第三領域 (約 20–約 30 ステップ：スタート付近) では，全てのロボットは一つのルールを用い，安定して前進している．最後に，第四領域 (約 30–約 40 ステップ：ゴール付近) では，Robot1 はゴールに向かって進む．Robot2 は第三領域から引き続いて前進を行う．そして，Robot3 は進行方向を修正するためにルールのスイッチングを若干ではあるが，ルールのスイッチングを行う．

まとめ

以上のように，ノイズや摩擦などの不確定要素を含む実環境においても，行動獲得が可能であることが実験的に示された．また，状況に応じて自律的に機能分化し，タスクを達成していることがわかった．システムを構成するロボットが二台，三台とシステム形態が異なっても，BRL の設定を変更することなく，

4.5 結言

本章では，BRL を用いた MRS の協調行動獲得において，環境のダイナミクスを軽減することで学習をより安定化することを目指した．次時刻における他ロボットの状態を予測する機構を構築し，その予測機構の出力を BRL の入力の一部として付加する手法を提案した．提案手法を自律移動ロボットによる協調搬送問題に適用した．計算機実験により予測情報を用いない場合よりも提案手法が効率的に学習していることを確認した．また，実環境においても，行動獲得に成功したことからノイズなどの不確定要素にも耐え得る頑健な学習が行えていることを確認した．

第5章　適応的な行動空間の探索による学習速度の向上

5.1　緒言

強化学習では試行錯誤を通して強化信号を受け取り，適切な入出力関係を構築する手法である．そのため，特に実環境での運用を考えた場合はできるだけ罰を受けないようにするとともに試行錯誤の期間を短縮したい基本的要求がある．この観点からBRL を考えると，行動の探索はランダムであるために行動空間の分割法を改善することが望ましい．

本章では，BRL の行動空間探索の効率化を目指し，既存ルールのパラメータに基づくルール生成法を付加した拡張法を提案する．移動ロボット一台のゴール到達問題とMRS の協調搬送問題の計算機・実機実験を行い，提案手法の有効性を検証する．

5.2　獲得ルールのパラメータを利用した行動空間の適応的探索

5.2.1　BRL における行動空間探索法の問題点

BRL では $g_w \geq g_{th}$ の場合はランダムに動作を実行して新しいルールを生成する．このとき，学習が進んだ状況および安定した行動を獲得した後に環境変動が起こった場合であってもこのようなランダム探索をすると，大きく動作が変化してシステムの不安定化につながる場合がある (Fig. 5.1)．すなわち，常に行動空間をランダムに探索するのは非効率であり，知識獲得がある程度行われた状況では幅広く行動空間を探索するよりも既存のルールの近傍を探索して行動の調整を行うことが有効であると考える (Fig. 5.2)．

5.2.2　行動空間を適応的に分割する拡張型 BRL

BRL はゴール到達時にのみ与えられる報酬を手がかりに学習を行う．そのため，スタートからゴールに至るまでの系列全体として評価されることになり，各時刻における行動のみを評価することは難しい．そのため，Actor-Critic や REINFORCE アルゴ

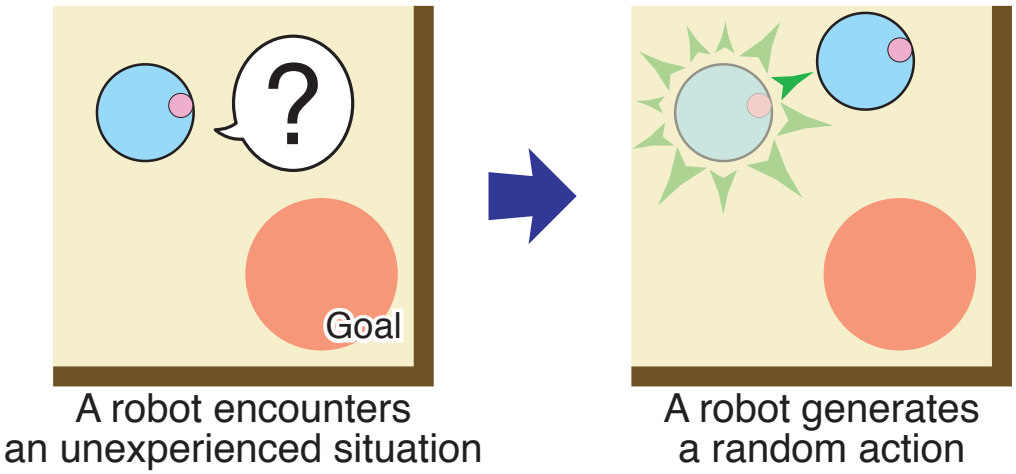


Fig. 5.1: Rule parameters for standard BRL

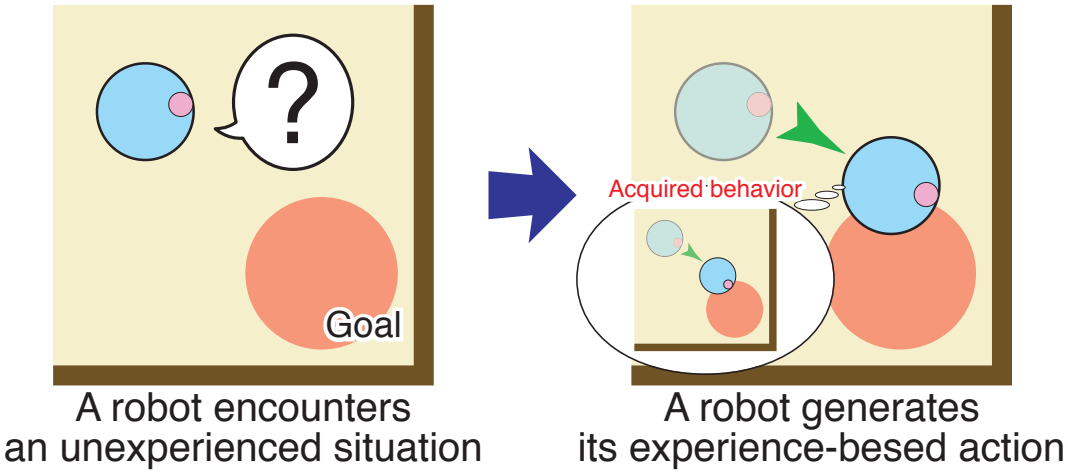


Fig. 5.2: An example of rule generation for the extended BRL

リズムのように時々刻々の行動の調整を行うことはできない．また，MRS 環境では一台の行動の変化が全体に影響を及ぼし得るので，行動を獲得した後に行動の調整や探索を頻繁に行うことで，システムが不安定になるのは避けるべきであると考える．

そこで，未経験な状態であるものの，過去に経験したものと比較的近い場合においてのみ，行動を微調整するようなルールを生成するものとする．その近傍のルールの選定のために，ルール選択時に計算する事後確率を用いる．Fig. 5.3 のようにランダム探索をするためのしきい値 P_{th} の他に新たにしきい値 P'_{th} を設定 $(P'_{th} < P_{th})$ し，$g_{th} \leq g_w < g'_{th}$ の場合はその間にあるルールのパラメータを参照して新しいルールパラメータを決定する．つまり，行動選択を以下のように変更する．

- $g_w < g_{th}$ の場合，rl_w の動作 A_w を実行する．

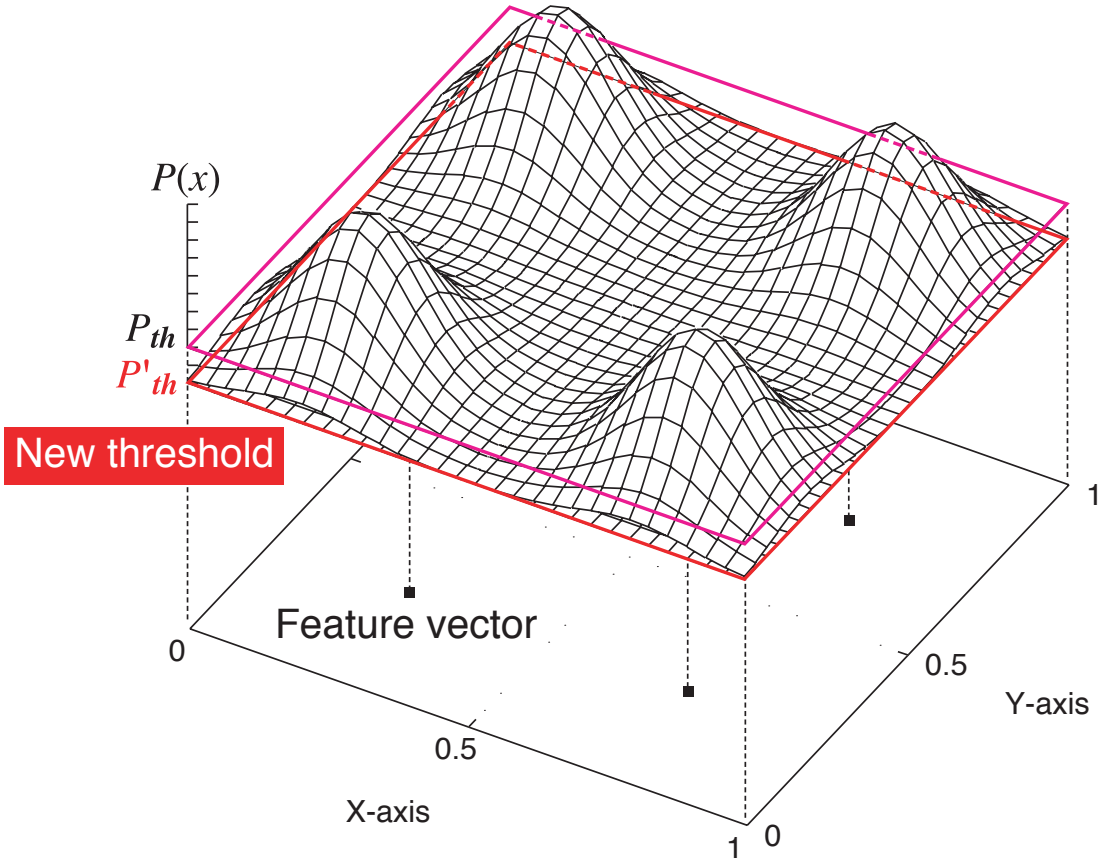


Fig. 5.3: A new threshold for extended rule generation

- $g_{th} \leq g_w < g'_{th}$ の場合，この間にあるルールを基に動作を生成する．
- $g_w \geq g'_{th}$ の場合，ランダム動作を実行する．

■ $g_{th} \leq g_w < g'_{th}$ **の行動**　この範囲には，rl_w 以外にも複数のルールが含まれる場合がある．これらのルールはその状況下での選択確率としては大きな差はないものの，それまでの学習過程におけるタスク達成への貢献度にしたがって有効度が異なる．そのため，新しいルールの動作 A' は，この範囲に含まれるルールの有効度に基づく動作の加重平均により求める．

$$A' = \sum_{l=1}^{n_r} \left(\frac{u_l}{\sum_{k=1}^{n_r} u_k} \cdot A_l \right) + N(0, \sigma_{act}) \tag{5.1}$$

ここで，n_r はこの範囲に含まれるルール数であり，$N(0, \sigma_{act})$ は平均 0・標準偏差 σ_{act} の正規分布を用いたノイズである．ノイズを付加することで，rl_w 以外のルールがない場合であっても，A_w の近傍を探索することができる．

5.3　光源到達問題による検証

5.3.1　実験設定

障害物が一つ存在する環境における移動ロボットのゴール (光源) 到達問題を取り扱う．ロボットは距離センサ (d_i)，光センサ (l_j) をそれぞれ四，三個持つ (Fig. 5.4)．本実験では，センサ入力を得て行動し，評価を得るまでを単位ステップとし，ゴールに到達するまでを単位エピソードとする．Fig. 5.5 と Fig. 5.6 に計算機実験，および実機実験の環境を示す．それぞれの実験において，スタート地点とゴール地点の間に大きさの異なる障害物がある二種類の環境を用意した．計算機実験では $Env1$ で実験を開始して 201 エピソードにてから $Env2$ に変化する実験と，$Env1$ で実験を開始して 201 エピソードにてから $Env2$ に変化する実験を行う．実機実験では 101 エピソードにおいて $EnvA$ から $EnvB$ に環境が変化するものとする．

本実験では拡張型 BRL の有効性の検証のため，従来型 BRL と最も一般的な強化学習法である Q-learning を用いた比較実験を行う．なお，ここで用いる Q-learning は，連続な空間を取り扱うために状態空間を正規化 RBF ネットワークを用いて関数近似したものである．以下に，BRL と Q-learning の設定を示す．

■ BRL の設定　入力は $\boldsymbol{x} = \{\, d_0, d_1, d_2, d_3, l_0, l_1, l_2 \,\}$ であり，出力は $\boldsymbol{a} = \{m_{rud}, m_{th}\}$ である．ここで，m_{rud} と m_{th} はそれぞれ，モータのステアリング・スロットル量である．ゴールに到達 (いずれかの光センサ値がしきい値を越えたとき) すると報酬が与えられ，ロボットが壁に衝突した (測距センサの値がしきい値を越えたとき) ときに罰が与えられる．ルール生成におけるしきい値は $P_{th} = 0.12, P'_{th} = 0.10$ とし，式 (5.1) の $\sigma_{act} = 0.1$ とした．その他のパラメータは前章のものと同一である．

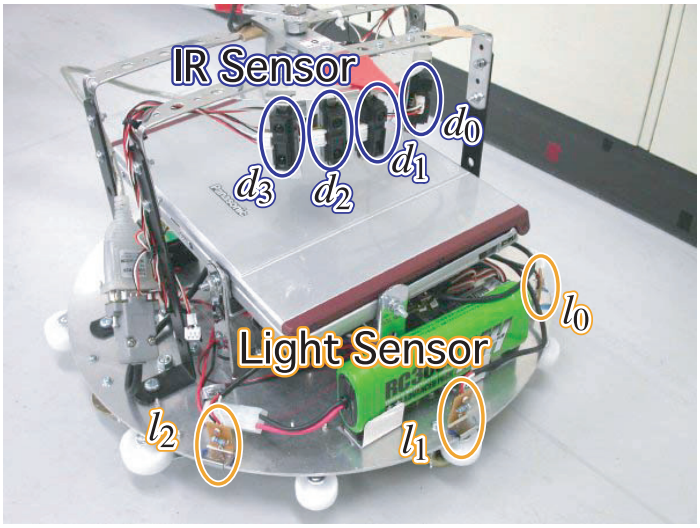
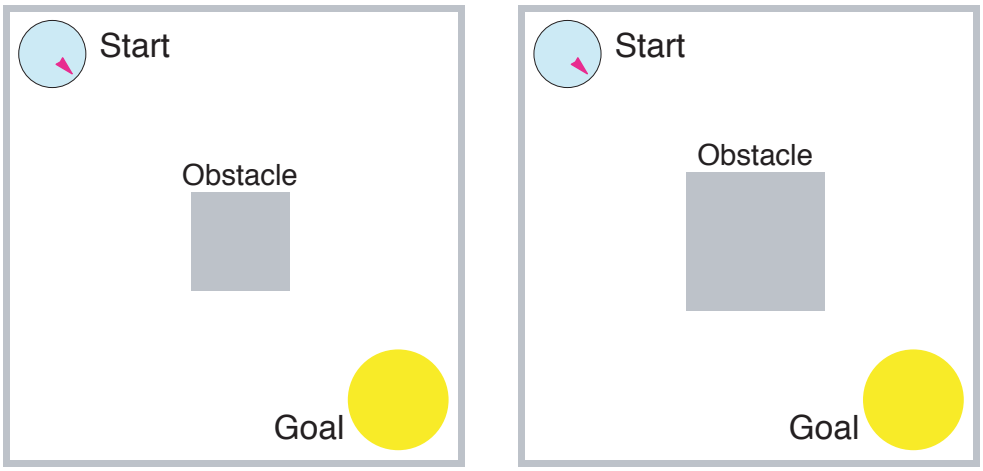
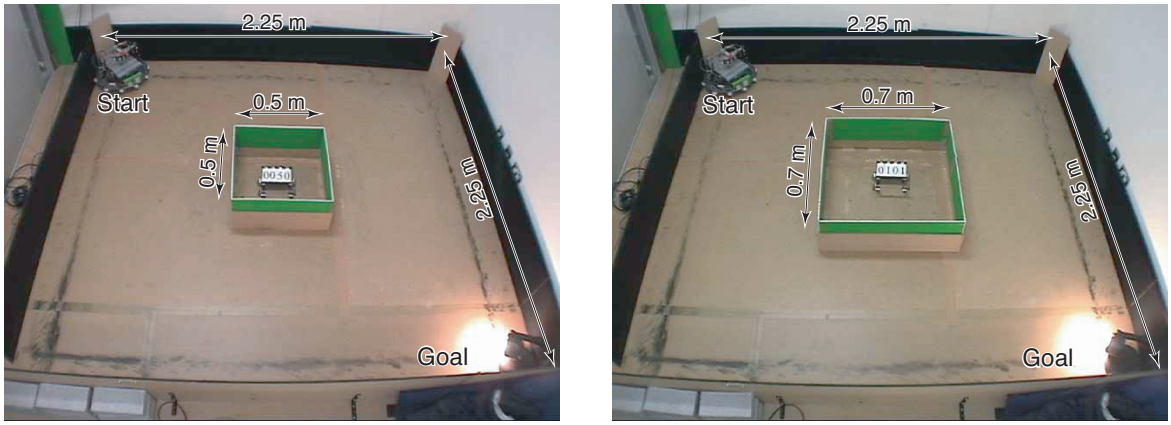


Fig.　5.4: Autonomous mobile robot



(a) $Env1$ (b) $Env2$

Fig. 5.5: Experimental environments: simulations



(a) $EnvA$ (b) $EnvB$

Fig. 5.6: Experimental environments: physical experimets

■ Q-learning の設定　まず，正規化 RBF ネットワークを用いて Q 関数を近似した Q-learning の概要を述べる．Q 値は次式により得られる．

$$Q(x,a) = \sum_{i=1}^{n} \phi(i,x) w(i,a) \tag{5.2}$$

ここで，n は基底関数の個数，$\phi(i,s)$ は状態 x に対する i 番目の基底関数の出力，$w(i,a)$ は i 番目の基底関数と行動 a の結合荷重である．正規化 RBF の出力は，

$$\phi(i,x) = \frac{\exp(-\frac{\|x-c_i\|^2}{2\sigma_i^2})}{\sum_{j=1}^{n} \exp(-\frac{\|x-c_j\|^2}{2\sigma_j^2})} \tag{5.3}$$

で定義される．c_i，σ_i^2 はそれぞれ，i 番目の基底関数の中心，分散 (幅) である．学習を通して，結合荷重は次のように更新される．

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha \text{TD-error} \frac{\partial Q(x,a)}{\partial \boldsymbol{w}}, \tag{5.4}$$

$$\text{TD-error} = r_t + \gamma \max_b Q(x_{t+1}, b) - Q(x_t, a_t) \tag{5.5}$$

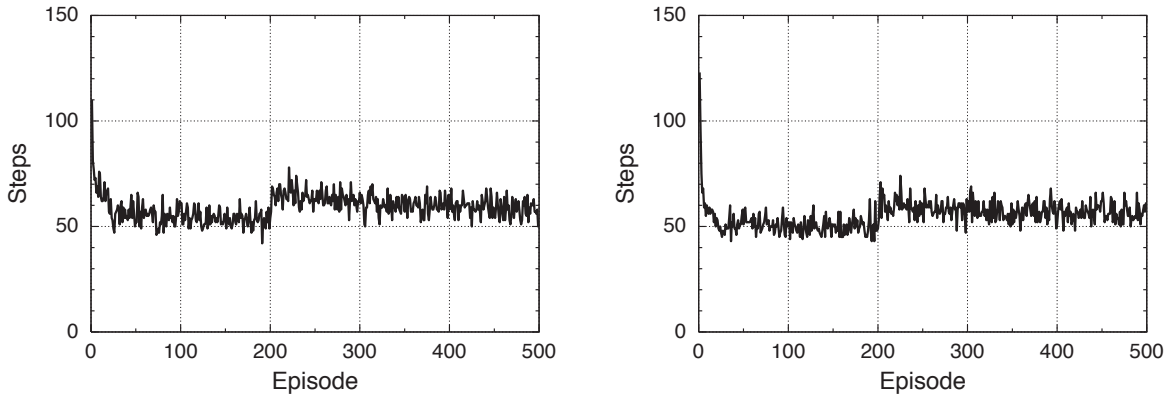
α は学習率，γ は割引率である．

本実験において，入力は $\boldsymbol{x} = \{\ d_0,\ d_1,\ d_2,\ d_3,\ l_0,\ l_1,\ l_2\ \}$ であり，それぞれのセンサに関して二個の基底関数を格子状に配置する．基底関数は計 128 個である．出力はあらかじめ離散化された五通りの行動 $\boldsymbol{a} \in \{a_0, ..., a_4\}$ である (a_0：前進，a_1：左旋回，a_2：右旋回，a_3：左回転，a_4：右回転)．行動は Boltzmann 分布を用いて確率的に選択する．学習パラメータは，$\alpha = 0.6$，$\gamma = 0.9$，$\sigma = 0.3$，$T = 0.8$ とする．ゴールに到達すると報酬 $r_t = 1.0$ が与えられ，ロボットが壁に衝突した (測距センサの値がしきい値を越えたとき) ときに罰 $r_t = -0.1 \times Q(s_t, a_t)$ が与えられる．

5.3.2 計算機実験

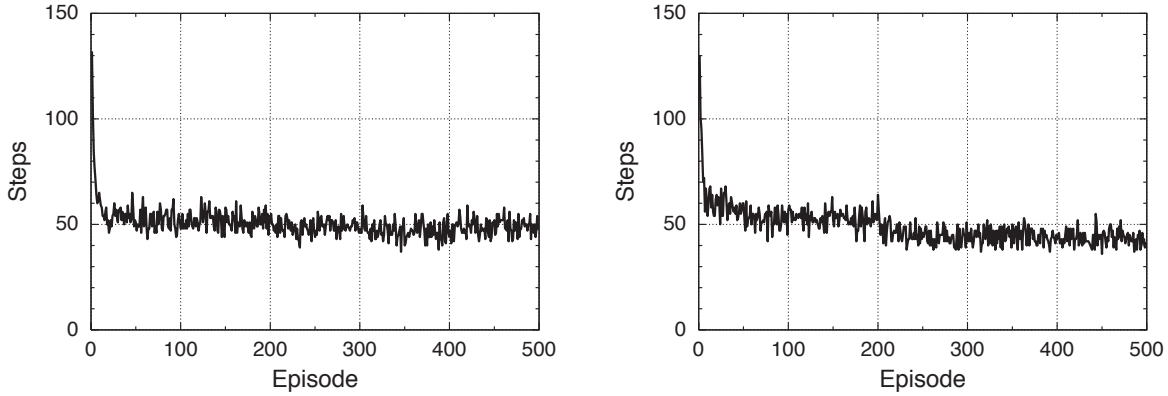
まず，提案手法における行動空間の探索効率向上を検証するため，計算機実験を行った．従来型 BRL と拡張型 BRL を用いた場合 ($Env1 \rightarrow Env2$) の，各エピソードで要したステップ数の 50 試行平均を Fig. 5.7 に示す．拡張型 BRL を用いることで学習開始後の傾きが大きいこと，および 201 エピソード以降のステップ数の増加量が小さいことから，効率的に初期学習および再学習を行っているといえる．スクを達成していることがわかる．また，環境が $Env1$ から $Env2$ に変化する場合の結果は Fig. 5.8 である．ここでも，201 エピソード以降に要しているステップ数を比較すると拡張型 BRL の方が少ないことから，効率的な学習が行われていることが確認できる．なお，201 エピソード以降のステップ数の増加が見られないのは，障害物が小さくなったために小回りする行動を獲得したためである．以上の二つの実験により，提案手法の有効性が実験的に示されたといえる．

次に，$Env1$ における Q-learning の学習履歴の一例を Fig. 5.9 に示す．200 エピソードまで実験を行っても行動が収束していない．これは，最適性を重視した学習法であるために，BRL と比べて探索を頻繁に行うことや報酬の伝播に時間がかかることに起因する．自律ロボットに強化学習を適用する場合，実際問題として最適解を発見することは不可能であるため，効率的に学習できる BRL のような経験強化型の学習法が有効であると考える．



(a) Standard BRL (b) Extended BRL

Fig. 5.7: Learning history ($Env1 \rightarrow Env2$): simulations



(a) Standard BRL (b) Extended BRL

Fig. 5.8: Learning history ($Env2 \rightarrow Env1$) : simulations

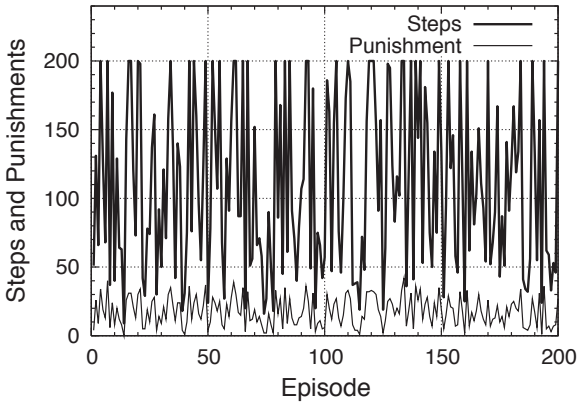


Fig. 5.9: Learning history for Q-learning ($Env1$): simulations

5.3.3 実機実験

学習履歴

前節で，拡張型 BRL が従来型 BRL と比べて学習効率が高いことが計算機実験を通して実験的に示された．本節では，実環境における頑健性を検証する．なお，ここで示す実験結果は，それぞれ拡張型 BRL，従来型 BRL，および Q-learning に関して複数回行ったうちの典型例である．Fig. 5.10，および Fig. 5.11 に，$EnvA$ と $EnvB$ での拡張型，従来型 BRL に関する各エピソードにおける，ゴール到達までのステップ数，および壁に衝突して罰を受けた回数を示す．Fig. 5.12 は，Q-learning に関する $EnvA$ で各エピソードでのステップ数と罰を受けた回数である．

$EnvA$ における学習結果に着目すると，拡張型 BRL と従来型 BRL ともに学習の収束が見られたが，従来型 BRL に比べて拡張型 BRL は学習の収束が早い．これは，試行錯誤の段階であってもタスク達成に寄与するルールを保持することで，それらのルール近傍を探索してすることが原因であると考えられる．Q-Learning では計算機実験と同様に学習の収束は見られなかった．

次に，従来型 BRL と拡張型 BRL に関して，101 エピソードに障害物の大きさが変化した後 ($EnvB$) の学習履歴を検証する．従来型 BRL では，環境変動が生じて未経験のセンサ入力を得たときに，ランダムな行動を持つルールを生成する．これに起因して，振る舞いが大きく乱れているといえる．それに対して，拡張型 BRL は環境変動直後は挙動が乱れたが，従来型 BRL に比べて，ゴール到達にかかったステップ数

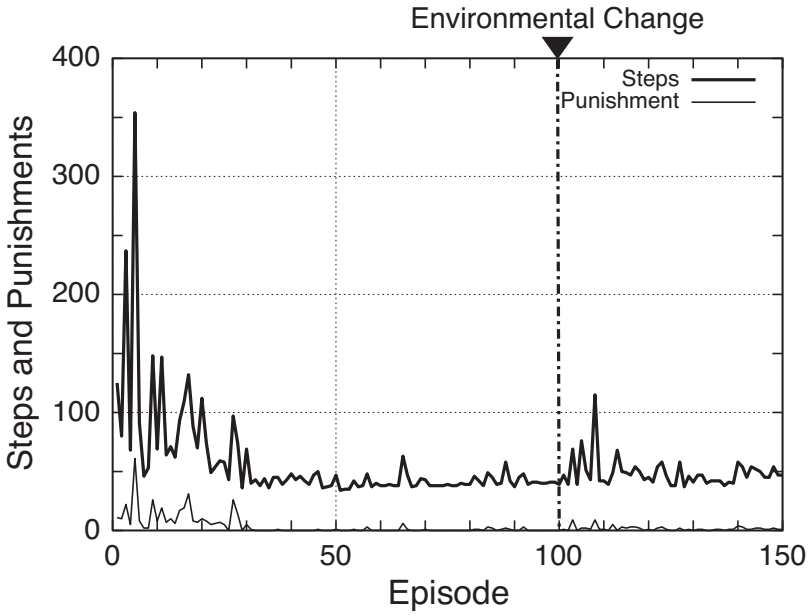


Fig. 5.10: Learning history for extended BRL

が明らかに少ないだけでなく，学習の収束も早い．これは環境変動が起こり探索が必要な状況になった場合，従来型 BRL では完全にランダムな探索ではなく既存ルールの近傍を探索することにより大きな行動の変化を生じることなく学習できたためである．以上より，拡張型 BRL の実環境における行動獲得に関する有効性とともに，学習・再学習が効率化が示された．

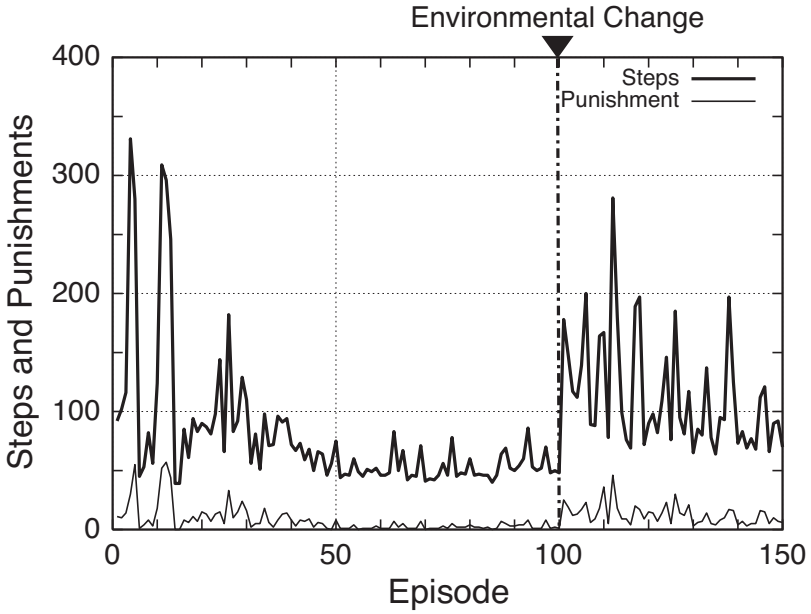


Fig. 5.11: Learning history for standard BRL

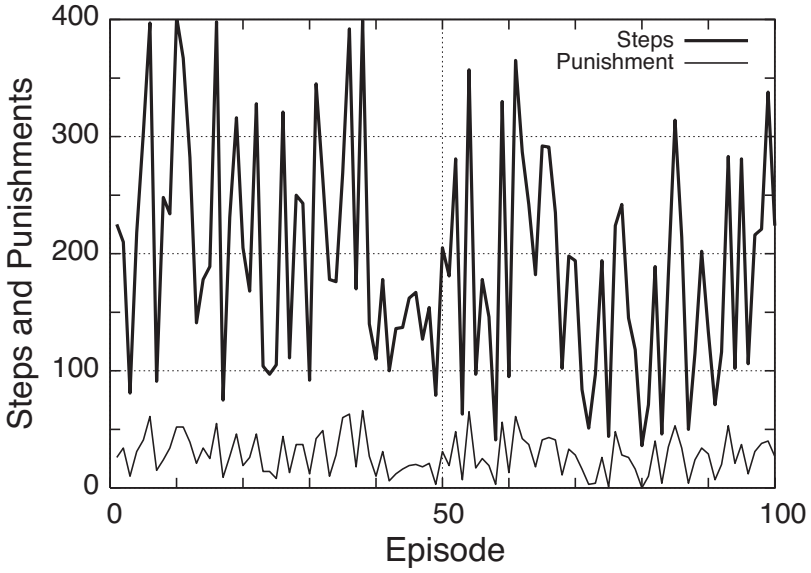


Fig. 5.12: Learning history for Q-learning

獲得した行動

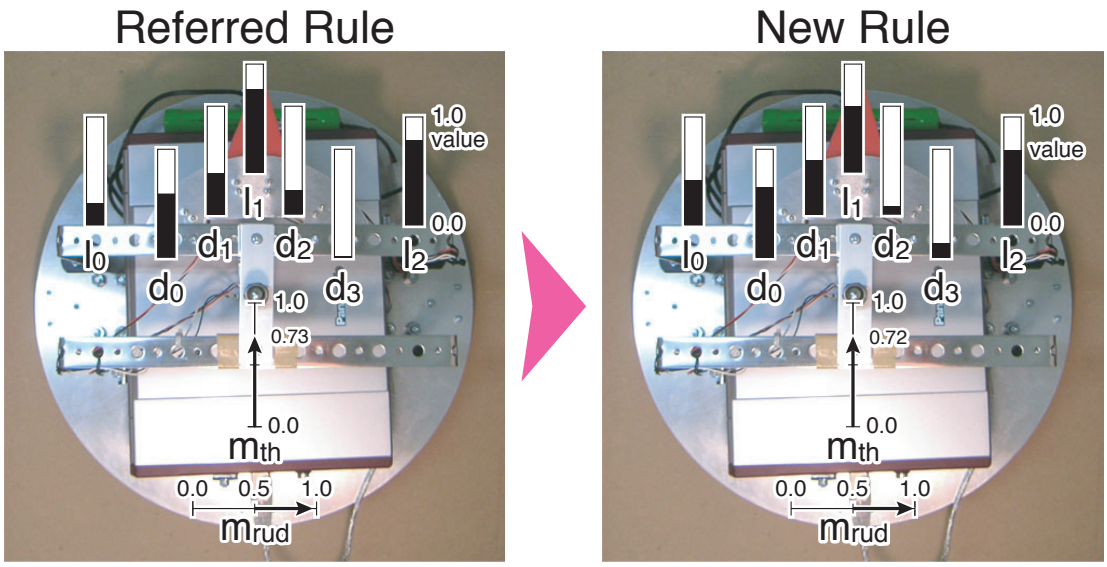
拡張型 BRL の獲得した振る舞いの例として $EnvA$ と $EnvB$ のものを Fig. 5.13 に示す．ふたつの環境での振る舞いはほぼ同一であり，$EnvB$ においても $EnvA$ で獲得したルールを適切に利用しているといえる．

次に，拡張型 BRL により得られたルールの例を Fig. 5.14 示す (黒棒：センサ値，矢印：モータ信号)．左図は動作決定において参照されたルール，右図は生成されたルールである．学習初期 (Fig. 5.14(a)) では，「ゴール付近でロボット右寄りに光を感知したときに，速度を下げて大きく右回転」するルールを生成している．また，連続してゴールをした後 (Fig. 5.14(b)) では，「前方やや左寄りに壁を感知したときに，速度を下げて大きく右回転」することで障害物回避行動の修正を行っている．このように新しいルールが必要になったときにランダムに行動を生成するのではなく既存ルールの近傍を探索することで行動の調整が行えるために，効率的なタスク達成が実現できた．

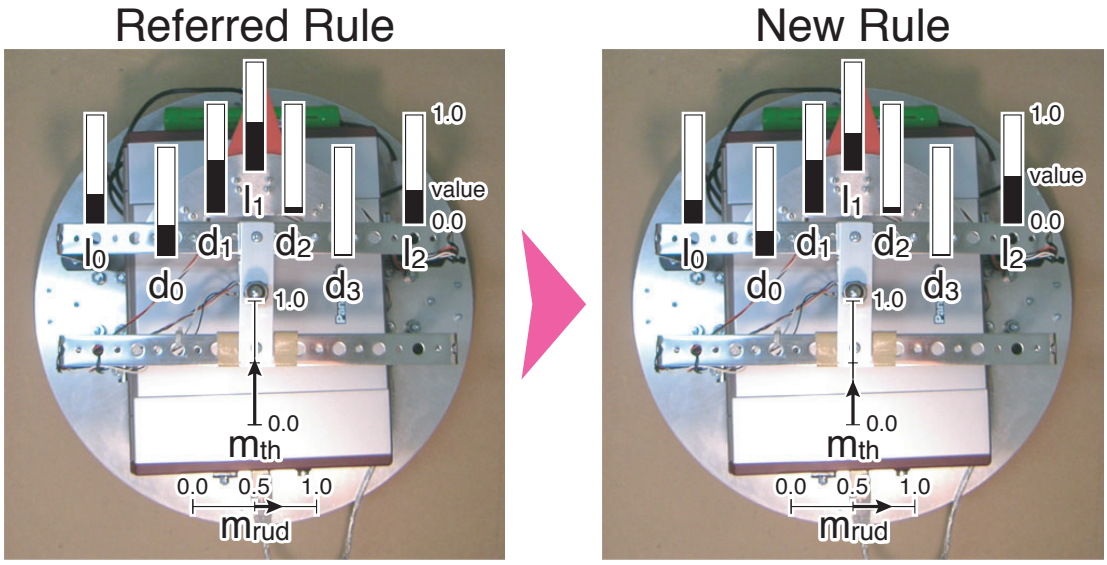
(a) $Env1$

(b) $Env2$

Fig. 5.13: Acquired behavior for extended BRL: physical experiment



(a) *EnvA*



(b) *EnvB*

Fig. 5.14: examples of generated rules: physical experiment

5.4　協調搬送問題による検証

前節の光源到達問題では，一台のロボットを対象として提案手法の有効性を検証する基礎実験を行った．本節では，提案手法の MRS の協調行動獲得に対する有効性を検証する．協調搬送問題を取り上げ，計算機・実機実験を行う．まず，計算機実験で行動獲得と環境変化後の再学習に対する有効性を検証する．その後，実機実験では行動獲得実験を行い，実環境で行動を獲得し得るか検証する．

5.4.1　計算機実験

実験設定

第 4 章と同様に，三台の協調搬送問題を取り上げる．タスクは Fig. 5.15 における左上のスタート地点から右下のゴールエリア (光源) までの移動である．ここで，各ロボットの初期位置は固定であるが，初期角度は図右を中心にしてランダムに $\pm 10^\circ$ の範囲で変動させる．センサ入力を得て行動し，評価を得るまでを単位ステップとする．ゴールに到達するか，ゴールに到達せずに 150 ステップが経過するまでを単位エピソードとする．制御器は第 4 章で提案した予測機構を付加した BRL を用いる．

■ 予測機構の設定　4.4.2 節で行った実験と同一の設定である．三層構造のフィードフォワードニューラルネットワークで構築する．入力は $I = \{ \cos\theta^i_{t-2}, \sin\theta^i_{t-2}, \cos\psi^i_{t-2}, \sin\psi^i_{t-2}, \cos\theta^i_{t-1}, \sin\theta^i_{t-1}, \cos\psi^i_{t-1}, \sin\psi^i_{t-1}, \cos\theta^i_t, \sin\theta^i_t, \cos\psi^i_t, \sin\psi^i_t\}$ であり，出力は $O = \{\cos\psi^i_{t+1}, \sin\psi^i_{t+1}\}$ である．ここで，$\psi^i_t = (\theta^j_t + \theta^k_t)/2$ とする $(i \neq j \neq k)$．中

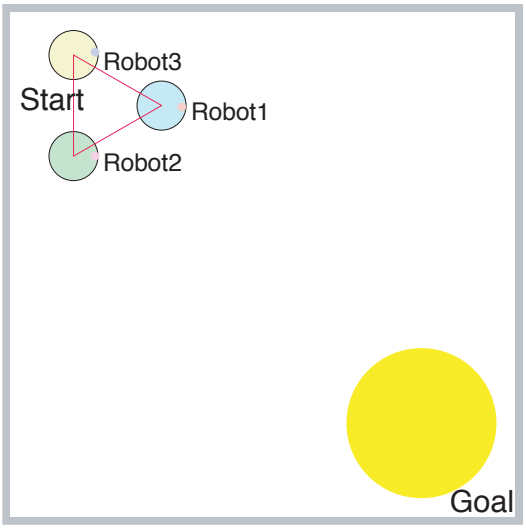


Fig.　5.15: Experimental environment

間層のニューロンは 8 個である．学習則には誤差逆伝播法を用い，学習率は 0.8，結合荷重の変化項に付加した慣性項のモーメント係数は 0.9 とした．

■ **BRL の設定**　4.4.2 節と同一の設定である．入力は $\boldsymbol{x} = \{\ \cos\theta_t^i,\ \sin\theta_t^i,\ \cos\psi_{t+1}^i,\ \sin\psi_{t+1}^i,\ d_0^i,\ d_1^i,\ l_0^i,\ l_1^i,\ l_2^i\ \}$ であり，出力は $\boldsymbol{a} = \{m_{rud}^i, m_{th}^i\}$ である．ここで，m_{rud}^i と m_{th}^i はそれぞれ，モータのステアリング量・スロットル量である．ゴールに到達すると全ロボットに報酬が与えられ，ロボットが壁に衝突した (測距センサの値がしきい値を越えたとき) ときに衝突したロボットのみに罰が与えられる．

協調行動獲得に関する実験：実験 1

まず，行動獲得に関する行動空間の探索効率を調べるため，拡張型 BRL と従来型 BRL について協調行動獲得実験を行った．Fig. 5.16 に，行動獲得までに要したエピソード数の 10 試行平均 (および標準偏差) を示す．なお，ここで用いたデータは，第 4 章で提案した BRL が行動を獲得するのに要する平均エピソード (460 エピソード) 以内に行動獲得に成功していたものから，それぞれの制御器に対して 10 例ずつを取り出したものである．

拡張型 BRL では少ないエピソード数で協調行動を獲得していることがわかる．これは，学習初期で試行錯誤を行っている段階であっても，ランダム探索をするなかでタスク達成に貢献したルールが少数ながらも存在するので，以降のエピソードではそれらのルールの近傍を探索することで行動を調整しているためであると考えられる．しかし，提案型の動作選択において参照するのは学習途中のルールであることもあり，拡張型 BRL の優位性はそれほどみられない．

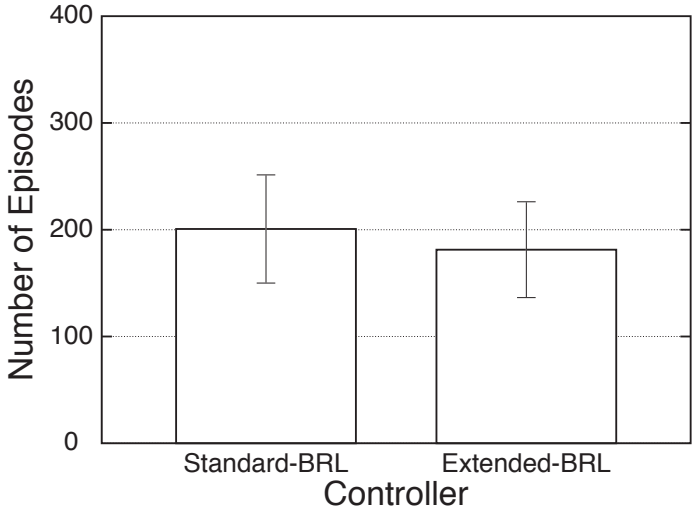
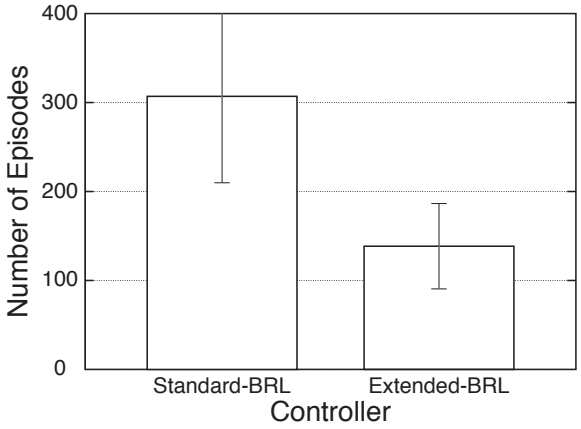


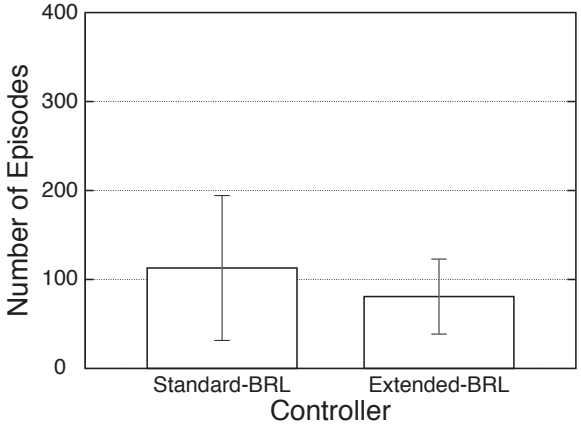
Fig. 5.16: Numbers of episodes to acquire cooperative behavior

環境変動後の再学習に関する実験：実験 2

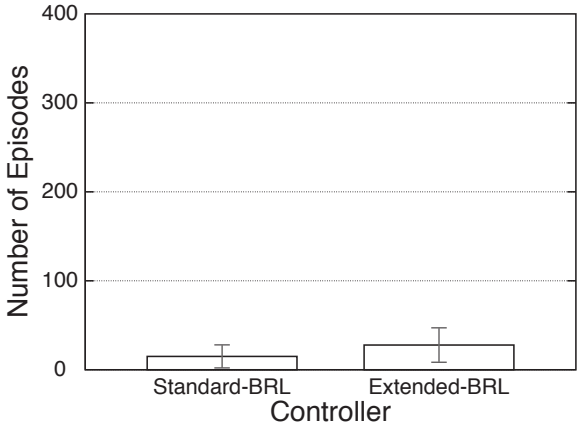
次に，獲得した知識に基づく行動探索の有効性を検証するために，行動獲得後に環境変動が生じた場合における再学習に関する実験を行った．ここでは，実験 1 で行動



(a) For the leader cases



(b) For the sub-leader cases



(c) For the follower cases

Fig. 5.17: Numbers of episodes to relearn behaviors

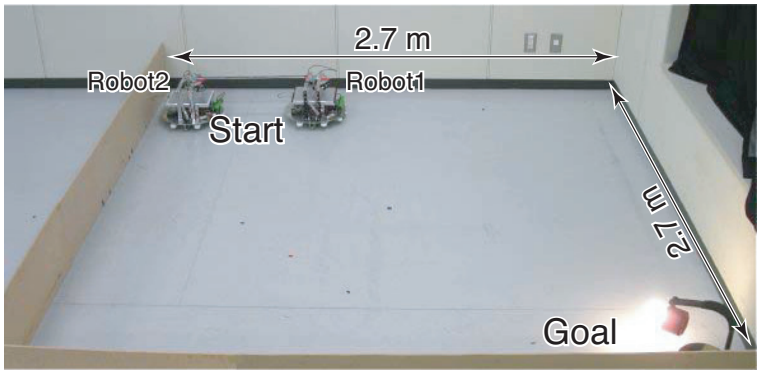
を獲得したエピソードから 100 回連続ゴールをした後，三台のロボットのうち一台の学習結果が初期化されるという環境変化が生じたものとする．なお，実験 1 における 10 試行の全てにおいて，三台のロボットは自律的機能分化をして異なる役割を果たすことでタスク達成をしていることが観測された．ここでは，行動を頻繁に切替えて進行方向の調整をする，補助的に行動を切替えて先導を補佐する，ほとんど行動を切替えずに振舞うといった行動を確認した．それぞれがこのような行動を獲得した後に一台の学習結果が初期化された場合，各ロボットのタスク達成に対する貢献度は再学習に必要なエピソード数によって評価できる．すなわち，タスク達成に大きく貢献していたリーダ的ロボットが初期化された場合はシステムに大きな影響を与えて再学習が困難になり，それほど貢献していないロボットであるとほとんど不安定になることなくタスクを達成し続けることができるといえる．以下，ロボットの初期化後の再学習に要したエピソード数について述べる．

Fig. 5.17 は，それぞれ，環境変化前に最も貢献していたリーダ的ロボット，それに次いで貢献していたサブリーダ的ロボット，貢献度が最も低いフォロワ的ロボットを初期化した場合に関するものである．大きな環境変化であり最も再学習が必要なリーダロボットの初期化の場合において，拡張型 BRL の有効性はより顕著に示されており，サブリーダロボットにおいても同様に拡張型 BRL が効率的に再学習が行えている (Fig. 5.17(a)，5.17(b))．しかし，フォロワロボット初期化の場合には従来型 BRL がよい結果を示している (図 5.17(c))．これは拡張型 BRL の一試行において，環境変化前に獲得していた行動の軌跡が安定しておらずタスク達成に必要なステップ数にばらつきがあったものがあり，環境変化後に比較的大きな行動の修正が必要であったにも関わらず提案手法による近傍探索を行ったので，再学習に多大なエピソードを要したためである．しかし，ほとんどの実験において，拡張型 BRL に付加した既存の知識に基づく行動生成法によって再学習の効率が改善されたことから，提案手法の有効性が示されたといえる．

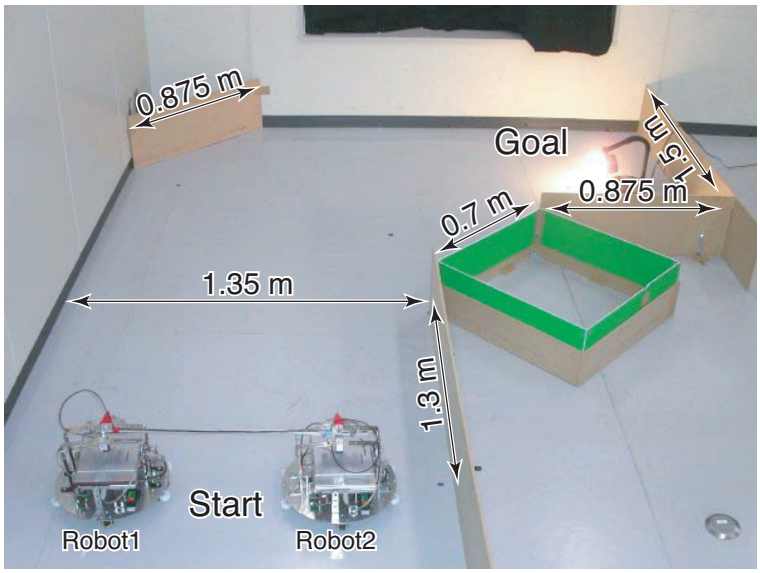
5.4.2 実機実験

実験設定

二通りの実験環境を用意し，それぞれの環境で行動獲得実験を行った (Fig. 5.20)．ここで用いたロボットのタイヤは，オムニホイールを採用した (A.2.3 節)．オムニホイールは円周方向に駆動し，軸方向には受動的に回転するものである．これを用いた理由としては，第 4 章で用いていた通常のスポンジタイヤであるとロボットがデッドロック状態に陥ること多くハードウェア的な負担が大きかったからである．オムニホイールを用いることで，その場に留まることはなくなった．しかし，受動的な移動ができるということから，他ロボットの意思決定による影響が大きくなる．すなわち，状態遷移の不確定性の影響が大きくなるという新たな問題が生じる．そのため，状態・



(a) $EnvS$



(b) $EnvL$

Fig. 5.18: Numbers of episodes to relearn behaviors

行動空間の分割がより重要になるといえる．

Fig. 5.18(a) の環境・$EnvS$ は，一辺が 2.7m の正方形の環境であり図の左上をスタート地点，右下の光源をゴール地点とする．Fig. 5.18(b) に示す L 字型の環境・$EnvL$ では，図の左下をスタート地点，右上の光源をゴール地点とする．$EnvS$ では壁伝いに移動したり環境を斜めに横切るなど，軌跡に制限がそれほどなくゴールに到達できる．一方，$EnvL$ ではゴールに到達するには壁に沿って移動した後，右に曲がる必要がある．

どちらの環境における実験においても，ロボットの初期姿勢は図のように固定する．学習はエピソード単位で行われ，単位エピソードは，ゴールに到達するか，または 100 ステップ経過するまでとする．

■ 予測機構の設定　三層構造のフィードフォワードニューラルネットワークで構築する．入力は $I = \{ \cos\theta^i_{t-2}, \sin\theta^i_{t-2}, \cos\theta^j_{t-2}, \sin\theta^j_{t-2}, \cos\theta^i_{t-1}, \sin\theta^i_{t-1}, \cos\theta^j_{t-1}, \sin\theta^j_{t-1}, \cos\theta^i_t, \sin\theta^i_t, \cos\theta^j_t, \sin\theta^j_t \}$ であり，出力は $O = \{\cos\theta^j_{t+1}, \sin\theta^j_{t+1}\}$ である $(i \neq j)$．中間層のニューロンは 8 個である．学習則には誤差逆伝播法を用い，学習率は 0.8，結合荷重の変化項に付加した慣性項のモーメント係数は 0.9 とした．

■ BRL の設定　入力は $\boldsymbol{x} = \{ \cos\theta^i_t, \sin\theta^i_t, \cos\theta^j_{t+1}, \sin\theta^j_{t+1}, d^i_0, d^i_1, l^i_0, l^i_1, l^i_2 \}$ であり，出力は $\boldsymbol{a} = \{m^i_{rud}, m^i_{th}\}$ である．ここで，m^i_{rud} と m^i_{th} はそれぞれ，モータのステアリング量・スロットル量である．ゴールに到達すると全ロボットに報酬が与えられ，ロボットが壁に衝突した (測距センサの値がしきい値を越えたとき) ときに衝突したロボットのみに罰が与えられる．

正方形環境

各エピソードにおけるゴール到達時のステップ数の推移を Fig. 5.19 に示す．実機実験の結果，4 エピソードではじめてゴールし，19 エピソード以降はゴールし続けた．またルール数もエピソードを重ねるとほぼ一定になっている．以上のことより，学習が収束していることがわかる．

次に，学習過程のシステムの振る舞いについて検証する．学習初期では，Fig. 5.20(a) のように，スタート後左旋回して壁に衝突しそのまま右上角まで進むがゴールできない行動が多い．学習が進むと Fig. 5.20(b) のような，ゴールに到達する行動を獲得した．Table 5.1 に行動収束時に各ロボットが保持しているルールのパラメータであり，

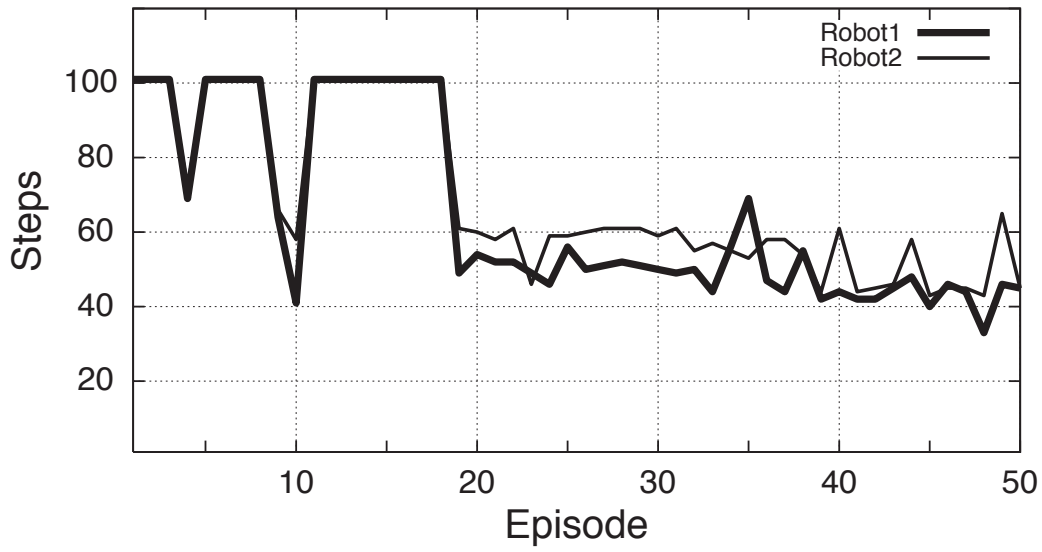
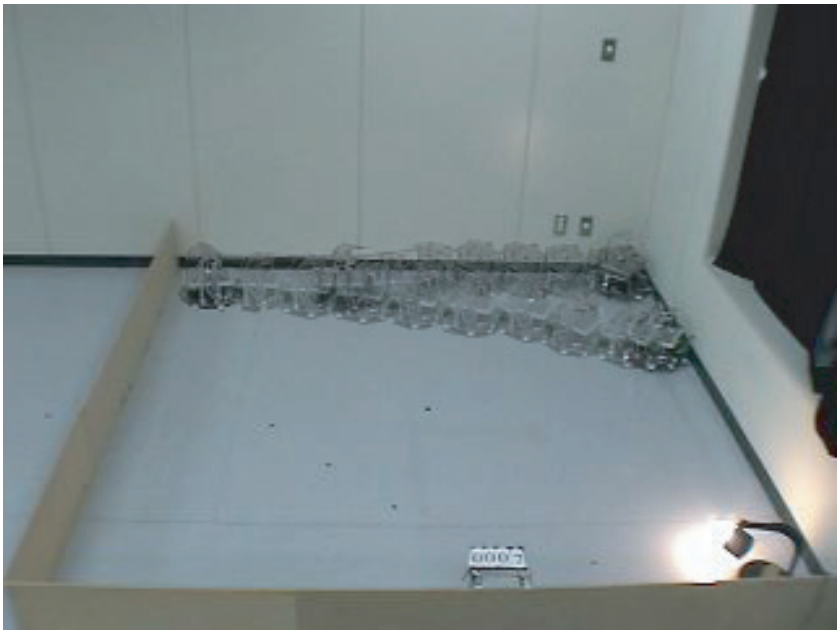
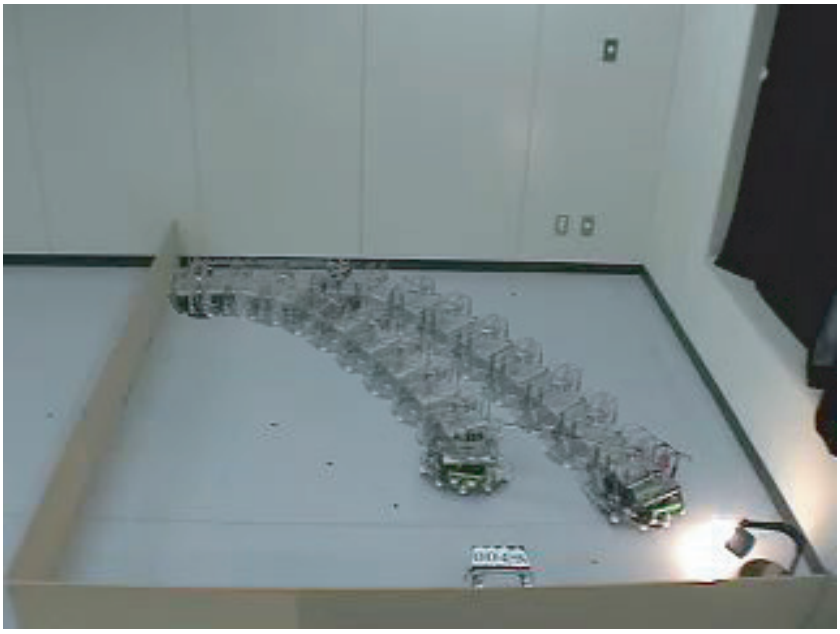


Fig. 5.19: Learning history: number of steps for *EnvS*



(a) Before successful learning



(b) After successful learning

Fig. 5.20: Behavior: *EnvS*

Fig. 5.21 はこの時の各ロボットに関するルールの発火系列である．Robot1 はスタート後に右旋回し，その後は前進を続ける．Robot2 は，スタート後に右旋回と前進した後，最終的には右旋回した．二台のロボットのそのような行動の組合せにより，オムニホイールによる受動移動を適切に利用してタスクを達成している．

Table. 5.1: Firing Rules in the 50th Episode for $EnvS$

Robot-Rule number	$\boldsymbol{v}$: State vector									$\boldsymbol{a}$: Action	
	$\cos\theta$	$\sin\theta$	$\cos\psi$	$\sin\psi$	d_0	d_1	l_0	l_1	l_2	m_l	m_r
1-2	0.891	0.809	0.993	0.675	0.003	0.003	0.396	0.492	0.496	0.646	0.789
1-5	0.996	0.440	0.887	0.442	0.003	0.003	0.317	0.347	0.423	0.042	0.503
2-2	0.803	0.898	0.993	0.995	0.003	0.003	0.423	0.549	0.428	0.620	0.008
2-7	0.977	0.651	0.516	0.716	0.003	0.003	0.243	0.381	0.347	0.666	0.583
2-8	0.976	0.652	0.959	0.556	0.003	0.003	0.298	0.440	0.390	0.174	0.867

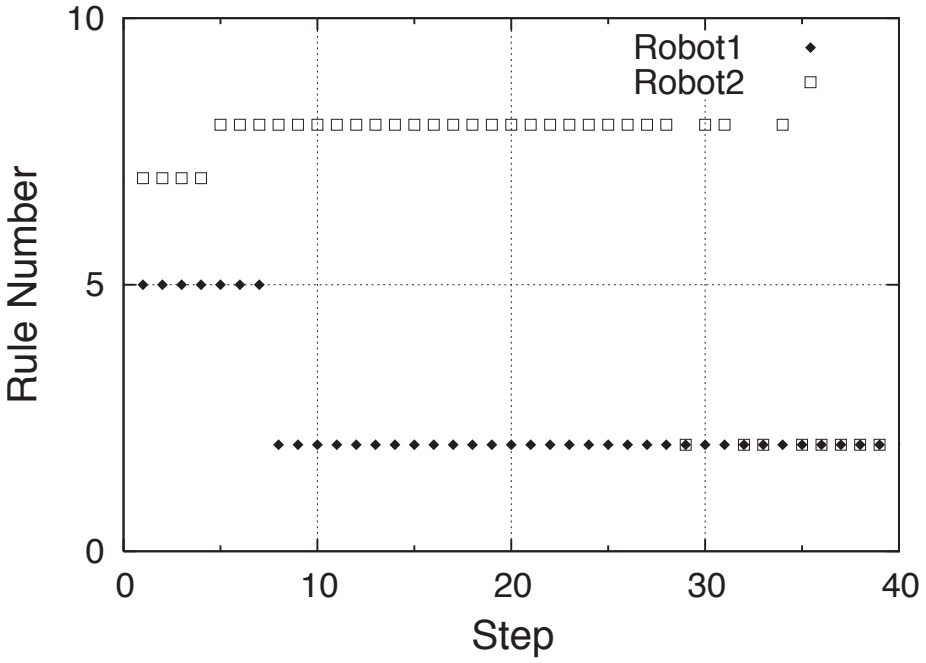


Fig. 5.21: The sequence of firing rules in the 50th episode

L 字環境

各エピソードにおけるゴール到達時のステップ数の推移を Fig. 5.22 に示す．試行錯誤を繰り返した後，39 エピソード以降はゴールし続けた．またルール数もエピソードを重ねるとほぼ一定になっている．以上のことより，学習が収束し行動獲得に成功したといえる．

システムの振る舞いを Fig. 5.23 に示す．学習過程においては，Fig. 5.23(a) のように Robot1 がスタート後左旋回し壁に衝突してそのまま左上角まで進むがゴールできない行動や，Fig. 5.23(b) のように左右の壁に衝突せずに前進した後に正面の壁に衝突してゴールできない行動が観察された．その後，最終的には，Fig. 5.23(c) のような壁に衝突することなくゴールに到達する行動を獲得した．

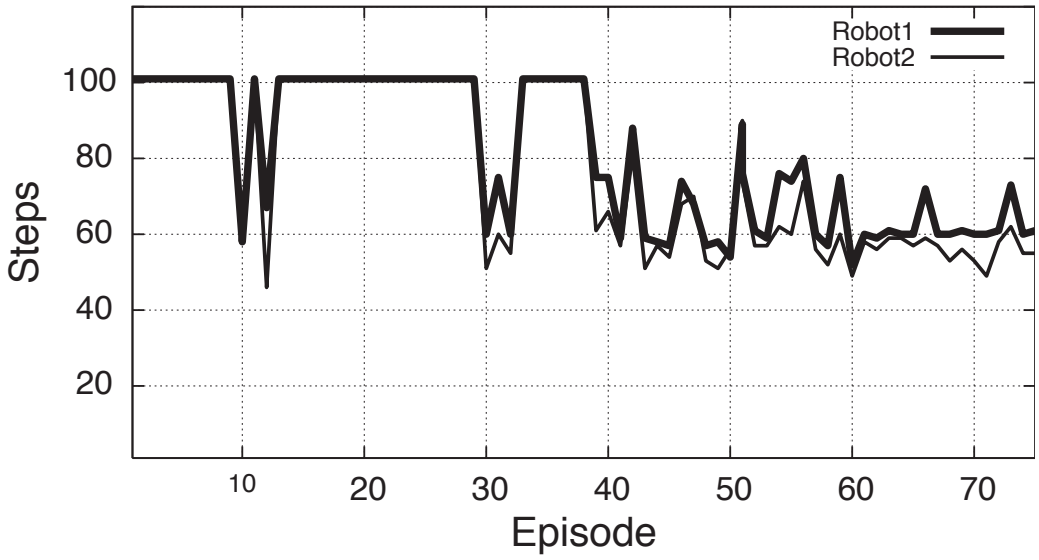
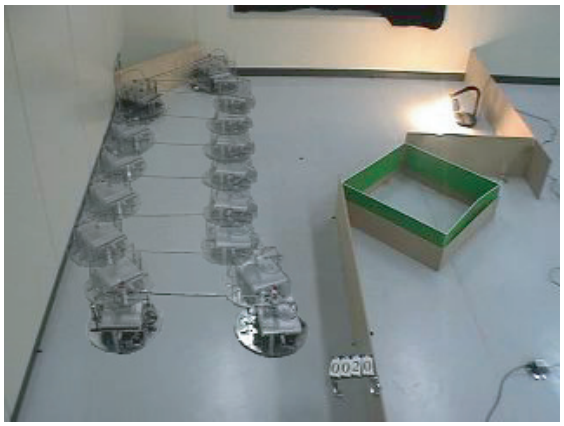


Fig. 5.22: Learning history: number of steps for *EnvS*

ここで，Table 5.2 は行動獲得後に発火しているルールのパラメータであり，Fig. 5.24 はそのときのルールの発火系列を示す．Robot1 は大きく左右旋回するのみである．Robot2 は始めはまっすぐに進み，左側の光センサが光を感知すると右旋回してゴールに向かい，ゴール近くで左右旋回してゴールに到達した．Robot1 の左右旋回によって，Robot2 は壁に衝突することなく壁に沿って動くことができている．この環境においても，ロボットはオムニホイールの特性を活かしてタスクを達成しているといえる．

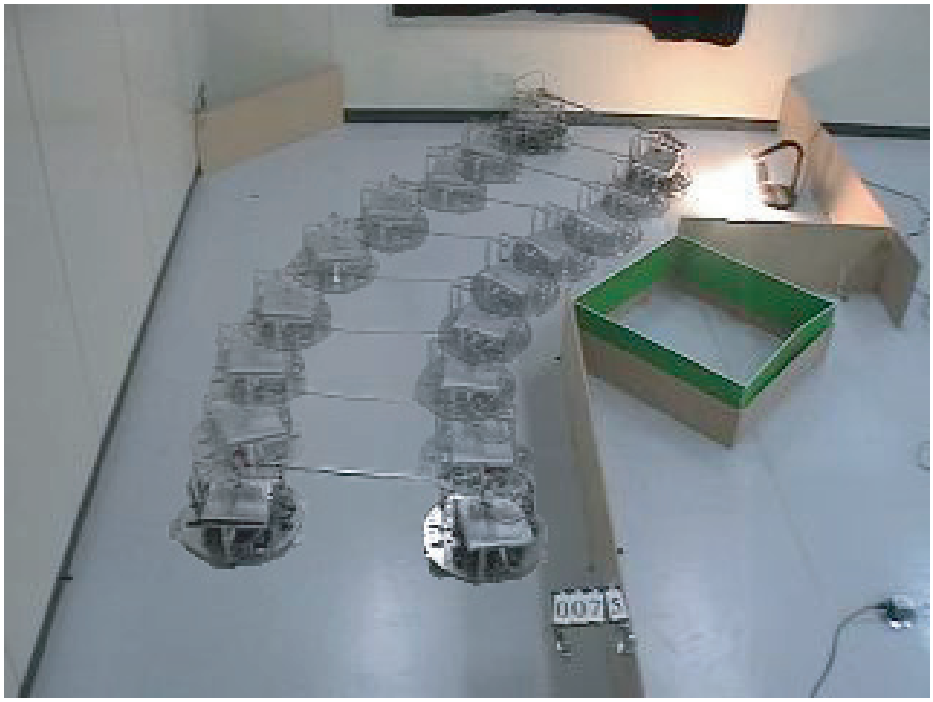
5.5 結言

本章では，行動空間の探索効率の向上を目的として BRL の拡張を行った．行動選択における事後確率の計算において新たなしきい値を設定し，近傍のルールのパラメータを基に新ルールを生成する手法を付加した．提案手法を光源到達問題，および協調搬送問題に適用し，行動獲得および獲得後の環境変動に対する再学習に関する計算機実験を行った．いずれの実験でも従来型 BRL と比較して短いエピソードでの行動獲得を確認し，提案手法の有効性を示した．また，それらの二つの問題に対して実機実験を行い，実環境における行動獲得に対する有効性が実験的に示された．



(a) Before successful learning 1

(b) Before successful learning 2



(c) After successful learning 1

Fig. 5.23: Behaviors for *EnvL*

Table. 5.2: Firing Rules in the 50th Episode for $EnvS$

Robot-Rule number	$\boldsymbol{v}$: State vector									$\boldsymbol{a}$: Action	
	$\cos\theta$	$\sin\theta$	$\cos\psi$	$\sin\psi$	d_0	d_1	l_0	l_1	l_2	m_l	m_r
1-2	0.820	0.881	0.833	0.909	0.007	0.003	0.084	0.110	0.320	0.135	0.080
1-4	0.404	0.989	0.632	0.955	0.003	0.003	0.085	0.135	0.154	0.936	0.085
2-2	0.178	0.882	0.159	0.926	0.003	0.003	0.378	0.537	0.297	0.822	0.631
2-3	0.573	0.994	0.287	0.903	0.003	0.003	0.031	0.103	0.002	0.612	0.305
2-4	0.202	0.902	0.304	0.947	0.003	0.003	0.039	0.300	0.267	0.151	0.910
2-7	0.517	1.000	0.410	0.936	0.003	0.003	0.113	0.549	0.521	0.319	0.239
2-14	0.202	0.902	0.461	0.995	0.003	0.311	0.077	0.210	0.001	0.550	0.800

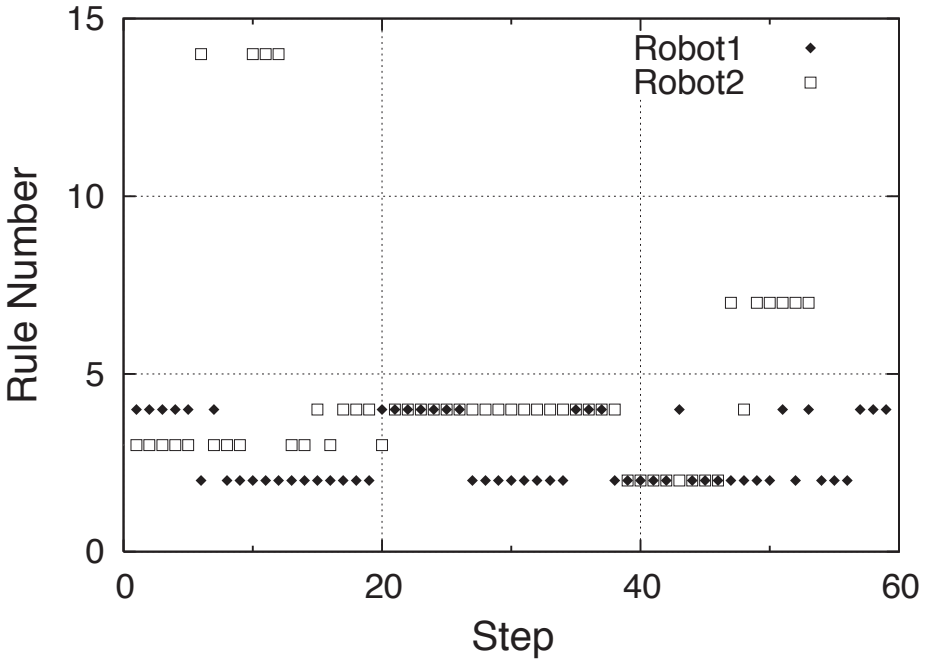


Fig. 5.24: Sequence of firing rules in the 75th episode

第6章　情報エントロピを用いた環境変化の認識と適応

6.1　緒言

本章では，環境変動への頑健性の向上のため，ロボットの行動の安定性を考慮したルールパラメータの構成，およびルールの保存を行う手法を提案する．そのためのパラメータとして各エピソードにおけるルール発火の情報エントロピを用いる．三台のロボットによる協調タスクの計算機実験において，学習収束後にロボット一台を未学習のロボットと交換するという環境変動を加えて，新しい協調関係の効率的な構築を目指すことで提案手法の有効性を検証する．

6.2　ルール発火の情報エントロピを用いた適応性の向上

6.2.1　行動の安定性に基づく指標

学習集束後に環境変動が生じた場合，BRL では変動を必ずしも認識する必要はなく，それまでの知識を利用してルールパラメータを修正することで対処可能な場合がある．しかし，新たな知識の獲得が必要な場合もあり，そのような状況においてのみ迅速な再学習への移行することが有効であると考える．BRL では，安定してタスクを達成することが可能になった後も実験を繰り返すと，特定のルールが多くの報酬を与えられることで他の学習法と同様に，過学習を引き起こす場合がある．そのような状況では，既存のルールが新しいルールの生成を阻害して，システムの頑健性が損なわれる．そのため，環境変動をなんらかの形で認識することが有効であるといえ，その指標として各エピソードにおけるルール発火の情報エントロピ E：

$$E = -\sum Q(i) \log Q(i) \tag{6.1}$$

$$Q(i) = h(i)/T \tag{6.2}$$

を採用する．ここで，$Q(i)$ はそのエピソードにおけるルール i の発火確率である ($h(i)$：ルール i の発火回数，T：エピソード終了までのステップ数)．E は行動が安定している場合はある値にほぼ収束するが，学習初期や環境変動によりシステムが不安定になった場合は値が大きく変動する (Fig. 6.1)．

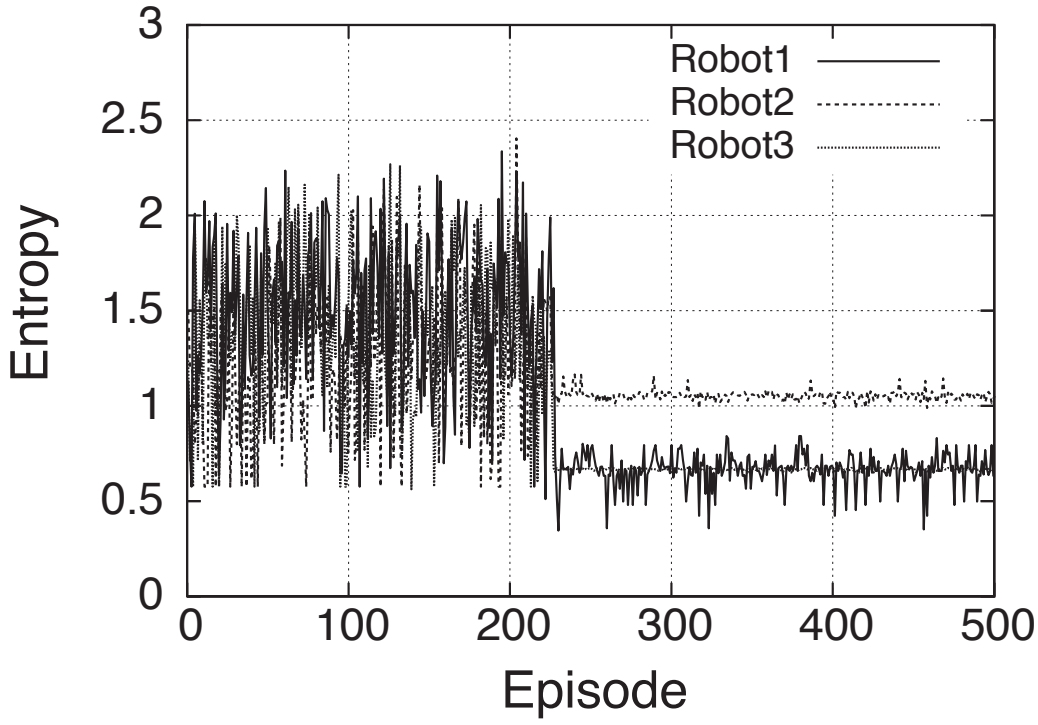


Fig. 6.1: Typical transition of entropy of firing rules

6.2.2 情報エントロピを用いた環境変化の認識と意思決定

ルールパラメータの一部として E を用いることで行動の安定度に基づいたルールを構成する．具体的には前エピソードでの E をルールの状態ベクトルに加える．これにより，BRL の入力はセンサ情報だけでなく，行動の安定度も含まれることになり，環境が変化して行動が不安定になった場合，E も大きく変化するために新しいルールの生成に寄与することになる．

6.2.3 ルールの保護による過学習の抑制

第 3 章では，多くのルールで状態空間を覆うことが過学習を抑制し，環境変動時における知識の再利用に有効であることを示した．ここではルールパラメータのひとつである分散共分散行列を基にルール構成が進んだルールを選別し，ルールの削除から保護するというアプローチをとっている．本研究では，これと同様の観点から，E を削除から保護するルールを選別するための指標としても用いる．つまり，通常の BRL ではタスク達成に寄与しないルールを削除してメモリ量を抑えるためにゴール到達時に全ルールの有効度に消散率 η をかけてあるしきい値以下になった場合はそのルールを削除するが，ルール i のエントロピ $e(i)$ と現在の E の差がしきい値以下の場合は同

程度に安定した (学習が進んだ) ルールであるとして消散を適用しないものとする．

$$u_i \leftarrow \eta\, u_i \quad \text{if} \quad |E - e(i)| > E_{th} \tag{6.3}$$

これにより，第 3 章での拡張と同様の効果が期待できる．第 3 章での手法は BRL が独自に持つパラメータを用いているために一般的な指標であるとはいい難いが，本章での E による選別は Q-learning などの他の学習器にも適用可能である．

6.3 協調搬送問題による検証

6.3.1 実験設定

計算機実験の環境を Fig. 6.2 に示す．第 4，5 章と同じく，タスクは左上のスタート地点から右下のゴールエリア (光源) までの移動である．各ロボットの初期位置は固定であるが，初期角度は図右を中心にしてランダムに $\pm 10^\circ$ の範囲で変動させる．ゴールに到達すると全ロボットに報酬が与えられ，ロボットが壁に衝突した (測距センサの値がしきい値を越えたとき) ときに衝突したロボットのみに罰が与えられる．センサ入力を得て行動し，評価を得るまでを単位ステップとする．ゴールに到達するか，ゴールに到達せずに 150 ステップが経過するまでを単位エピソードとする．

■ 予測機構の設定　第 4，5 章までの実験と三層構造のフィードフォワードニューラルネットワークで構築する．三層構造のフィードフォワードニューラルネットワークで構築する．入力は $I = \{ \cos\theta^i_{t-2}, \sin\theta^i_{t-2}, \cos\psi^i_{t-2}, \sin\psi^i_{t-2}, \cos\theta^i_{t-1}, \sin\theta^i_{t-1}, \cos\psi^i_{t-1},$

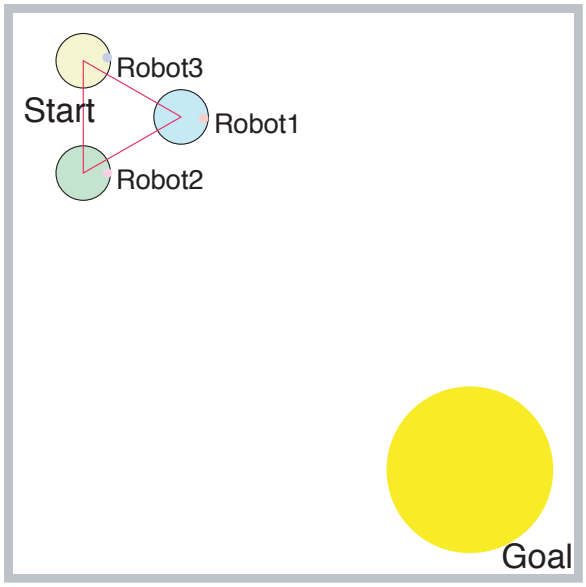


Fig. 6.2: Experimental Environment

$\sin\psi^i_{t-1}$, $\cos\theta^i_t$, $\sin\theta^i_t$, $\cos\psi^i_t$, $\sin\psi^i_t\}$ であり，出力は $O = \{\cos\psi^i_{t+1}, \sin\psi^i_{t+1}\}$ である．ここで，$\psi^i_t = (\theta^j_t + \theta^k_t)/2$ とする $(i \neq j \neq k)$．中間層のニューロンは 8 個である．学習則には誤差逆伝播法を用い，学習率は 0.8，結合荷重の変化項に付加した慣性項のモーメント係数は 0.9 とした．

■ BRL の設定　制御器は第 4 章で提案したものを基に拡張する．入力は $\boldsymbol{x} = \{\cos\theta^i_t, \sin\theta^i_t, \cos\psi^i_{t+1}, \sin\psi^i_{t+1}, d^i_0, d^i_1, l^i_0, l^i_1, l^i_2, E\}$ であり，出力は $\boldsymbol{a} = \{m^i_{rud}, m^i_{th}\}$ である．消散からルールを保護するためのしきい値は $E_{th} = 0.2$ とする．比較のため，エントロピの入力への付加，およびエントロピに基づくルール保護のどちらも採用しない (制御器 1)，入力付加のみ採用 (制御器 2)，ルール保護のみ採用 (制御器 3)，どちらも採用 (制御器 4) の四種の制御器を構築する．

6.3.2　実験結果

大域的秩序獲得実験：性能比較実験 1

まず，基本的な性能を評価するため，初期状態から協調行動を獲得するまでに要するエピソード数を比較する．この実験では，100 エピソード連続でゴールした場合に，行動を獲得したものとみなす．各制御器に関して 10 個の成功データから検証する．実験成功の基準は 5.4.1 節における協調搬送問題の計算機実験と同一である．Fig. 6.3 は各制御器が要したエピソード数の平均と分散である．図中の A はエントロピの入力への付加，B はルール保護を表す．エントロピを入力に付加した制御器 2 と制御器 4 は従来型 BRL である制御器 1 と比べて約 75%のエピソードしか必要としていないことがわかる．このことから，未学習の行動探索においても，エントロピが大きく変化した場合は積極的に探索を行い，タスク達成可能な行動を早く発見しているといえる．なお，制御器 3 に関しては，第 3 章の拡張法と同様に初期学習のパフォーマンスに影響を与えるものではないため，制御器 1 とほぼ同等のパフォーマンスである．

Fig. 6.4 は，各制御器の各エピソードにおける総ルール数，および生成したルール数である．この図から，エントロピに基づく消散からのルールの保護をした場合 (制御器 3 と制御器 4)，ルールが減ることなく最終的に多くのルールを保持していることから，ルール集合の多様性の維持のための指標として用いることが可能といえる．

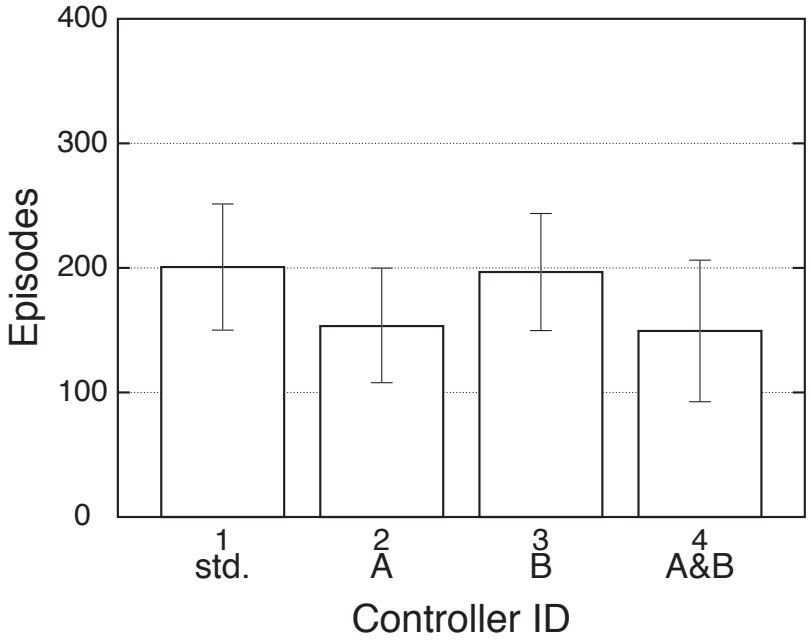
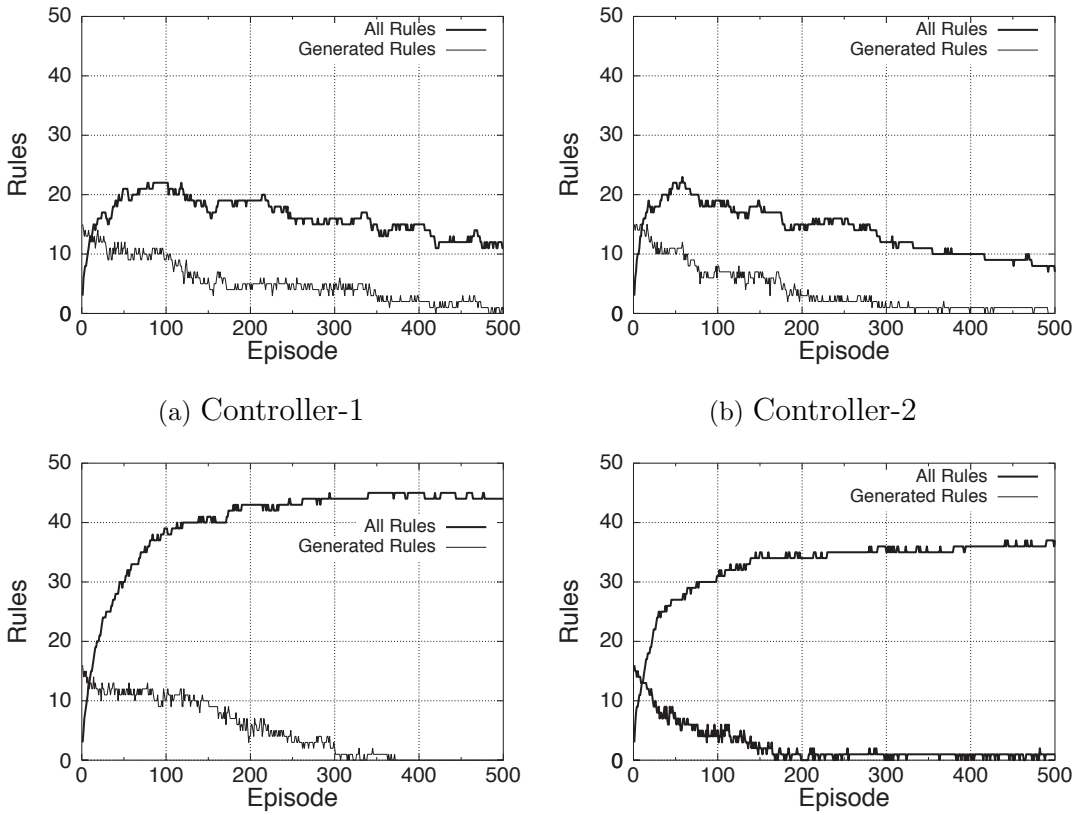


Fig. 6.3: Numbers of episodes until the MRS achieves a GSB



(a) Controller-1

(b) Controller-2

(c) Controller-3

(d) Controller-4

Fig. 6.4: Number of Rules

頑健性の検証実験：性能比較実験 2

環境変動へのシステムの頑健性の検証のため，安定的な協調行動 (100 エピソード連続してゴール到達) をしている状態で一台のロボットを未学習のロボットと交換するという環境変動を加え，再び連続してタスクを達成するようになるまでのシステムの振舞いを調べる．なお，性能比較実験 1 では，前章までと同様に自律的機能分化の発現によってタスクを達成している．そこでは，タスク達成に最も貢献しているリーダロボットが取り除かれた場合において，システムが長期に渡って不安定になっていた．これは，第 5 章の計算機実験で確認した傾向と一致する．すなわち，従来型 BRL であっても，小さな環境変動に対しては頑健に適応できるが，大きな環境変動に対しては行動の不安定化が長期化するためにそのような環境変動後の再学習の効率化が最も重要になる．そこで，本節では，前節の 10 試行それぞれにおけるリーダロボットを初期化した場合，つまり，最も再学習の効率の改善が望まれる場合において，パフォーマンスの検証を行う．

Fig. 6.5 は各制御器が要したエピソード数の平均と分散である．エントロピを入力に付加した制御器 2 と制御器 4 は従来型 BRL である制御器 1 と比べて約 50%のエピソードを必要とするのみであることがわかる．このことから，リーダロボットがシステムから取り除かれることで，エントロピが大きく変化し，頻繁にルールが生成されているといえる．それにより，新たな大域的秩序の形成がより迅速に行われている．ルール保護を行う制御器 3 は関しては，約 60%のエピソード数で再学習をしており，第 3 章の拡張法と環境変化後の頑健性の向上に貢献していることがわかる．入力への

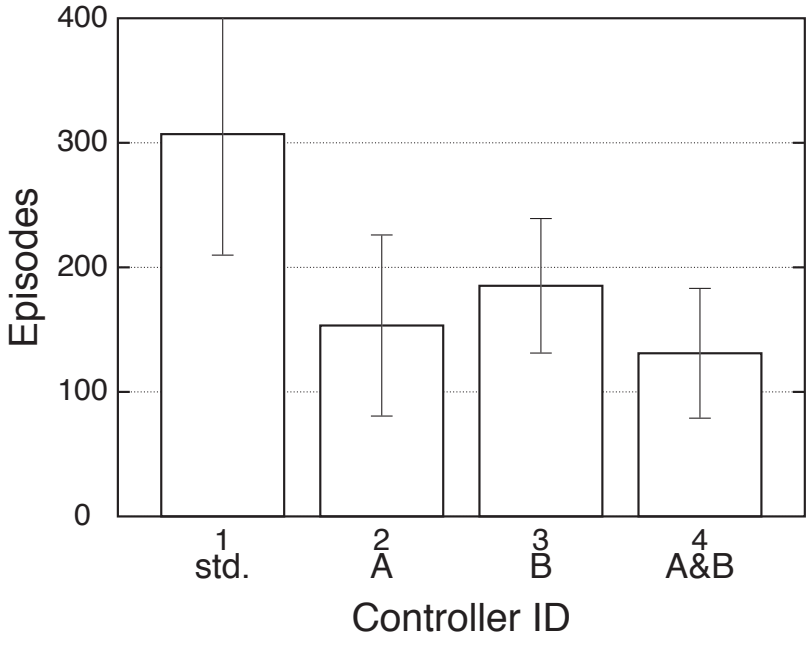


Fig. 6.5: Numbers of episodes for re-learning

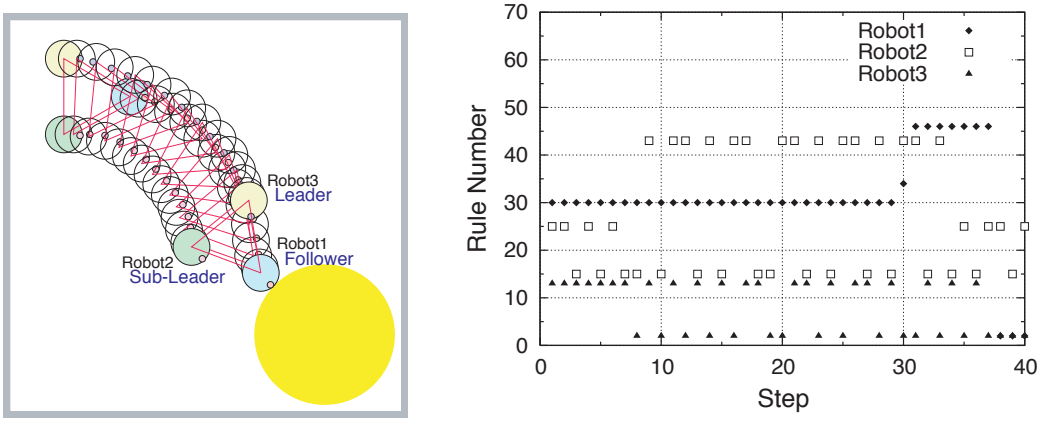
付加，ルール保護のどちらも行っている制御器 4 が最も高いパフォーマンスを示している．

環境変動後のシステムの振舞い (制御器 4)

提案手法を用いた場合の結果として，各ロボットを交換した場合のシステムの挙動の一例を示す．Fig. 6.7–6.9 は，それぞれ Robot1，Robot2，および Robot3 を未学習ロボットと交換した場合の (a) 学習履歴，(b) ルール数の推移，(c) エントロピの推移，(d) 再学習後の振舞い，および (e) ルールの発火系列である．ここで示す実験にでは，未学習ロボットとの交換前の振る舞いは Fig. 6.6 のようであった．Fig. 6.6(a) は獲得した行動である．各ロボットは Fig. 6.6(b) に示すように，

- **Robot1:** $30 \rightarrow 46 \rightarrow 2$
- **Robot2:** $(25 \leftrightarrow 15) \rightarrow (45 \leftrightarrow 15) \rightarrow (25 \leftrightarrow 15)$
- **Robot3:** $13 \rightarrow (2 \leftrightarrow 13)$

という系列でルールを切替えながらタスクを達成する．そのルールパラメータと発火系列から Robot1，Robot2 と Robot3 はそれぞれ，フォロワ，サブリーダ，およびリーダ的役割を果たしていた．以下，変動後の発火系列において，ルール x' は交換前に発火していなかったものの保持されていたルール，ルール y'' は交換後に生成されたルールを表す．なお，それぞれのロボットが独自の制御器を持つため，同じ番号であってもロボット毎に異なるルールを表す．各ロボットを未学習のロボット交換したそれぞれの場合の結果について述べる．



(a) Behavior (b) Sequence of firing rules

Fig. 6.6: Learning result before an environmental change

■ **Robot1**　ほとんどルールを切替えずに他の二台に追従するフォロワ的役割を果たし，タスク達成への貢献度は比較的低いロボットであったため，交換してもシステムは不安定になることなく継続してタスクを達成している．Robot2 はエントロピの変化に伴って保持していたルールの再利用と新ルールの生成を迅速に行っている．Robot3 は交換後も振舞いをほとんど変えていない．このようにシステムの変動が大きくない場合は，再学習をして時間をかけることなく必要でなくパラメータ修正と保持ルールの有効利用を行うことで対処する．新ロボットは交換前のロボット同様にあまりルールを切替えることなく追従する行動を獲得している．各ロボットのルール発火系列は次の通り．

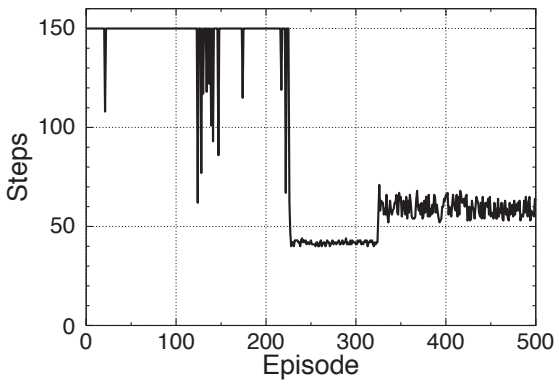
- **Robot1':** 2" → 5" → 2 " → 5"
- **Robot2:** ((25 ↔ 15) ↔ 43') → 27" → 44'
- **Robot3:** 13 → (2 ↔ 13)

■ **Robot2**　交換によって一時的にシステムの振舞いは不安定になるものの，Robot1 と Robot3 は交換前と同様のルールを用い，新ロボットはそれらの振舞いの妨げとならないような行動を獲得する．ここでも，再学習には移行せずに既存の知識の再利用で対処している．各ロボットのルール発火系列は次の通り．

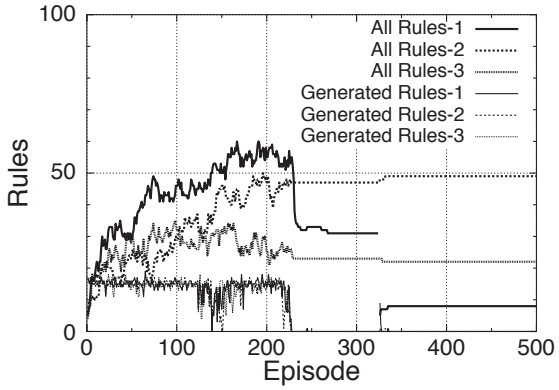
- **Robot1:** 30 → 46 → 2
- **Robot2':** 9" → 3" → (9" ↔ 3") → 1"
- **Robot3:** 13 → (2 ↔ 13)

■ **Robot3**　交換前には最も安定した振舞いをするリーダが取り除かれたことでシステムは大きく不安定になる．残りのロボットはエントロピが大きく変化することで大きな環境変動が起こったことを認識して再学習へと移行して新たなルールを生成するとともに，保持ルールの再利用を行い，新たな協調的振舞いを獲得している．各ロボットのルール発火系列は次の通り．

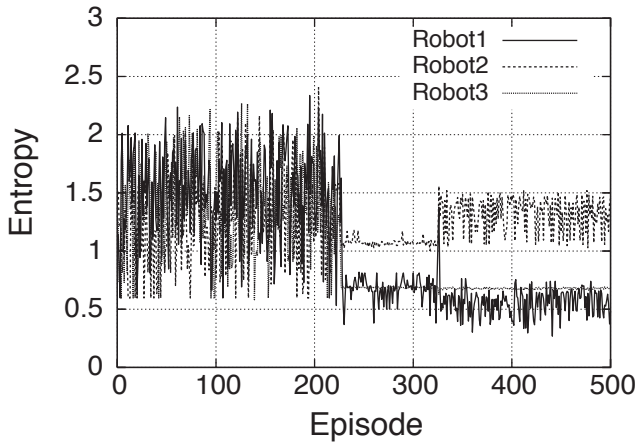
- **Robot1:** 67" → 69" → 46 → 48' → 2
- **Robot2:** 37" → 27" → (49" ↔ 27")
- **Robot3':** 23" → (15" ↔ 16")



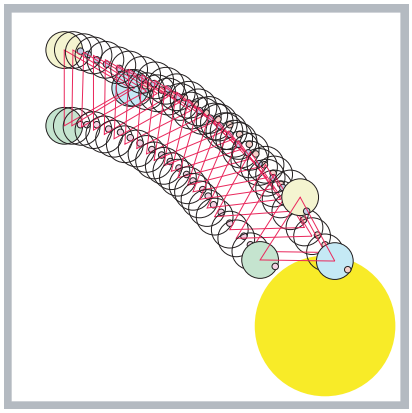
(a) Learning History



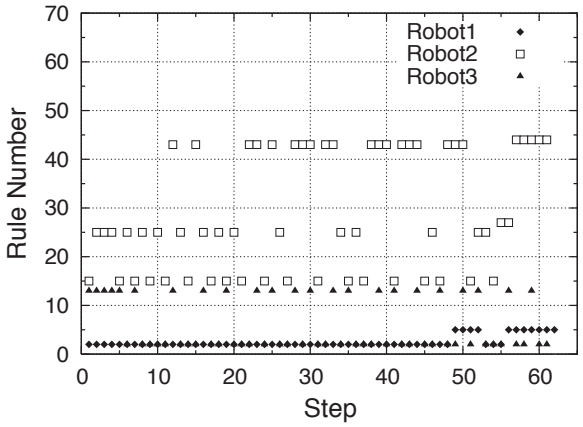
(b) Number of Rules



(c) Entropy

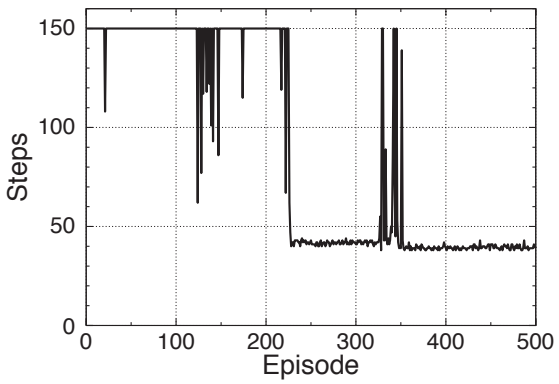


(d) Behavior

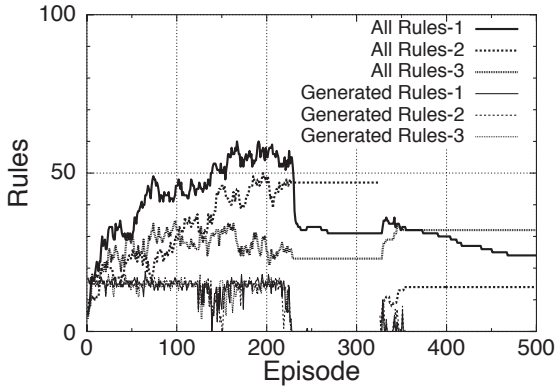


(e) Sequence of firing rules

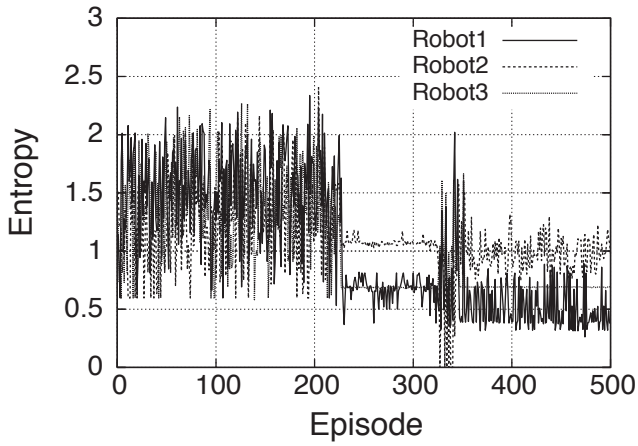
Fig. 6.7: Learning Result (Initialization: Robot1)



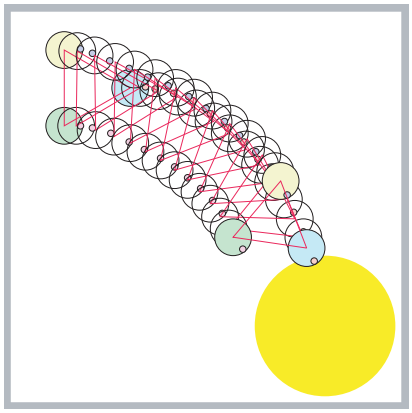
(a) Learning History



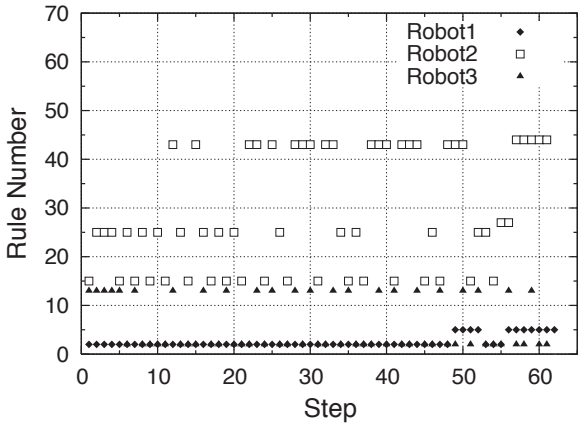
(b) Number of Rules



(c) Entropy

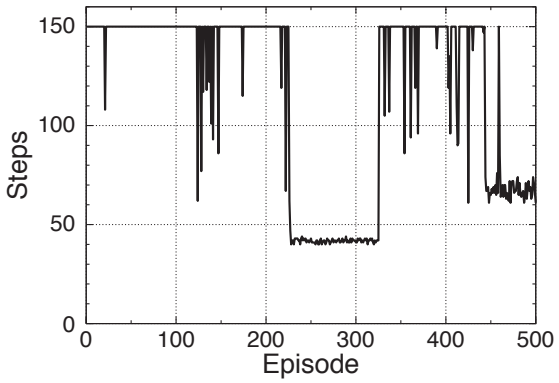


(d) Behavior

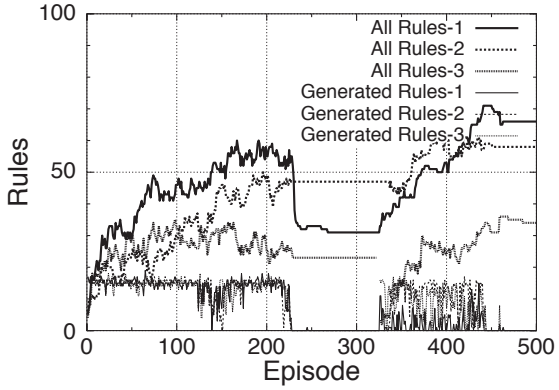


(e) Sequence of firing rules

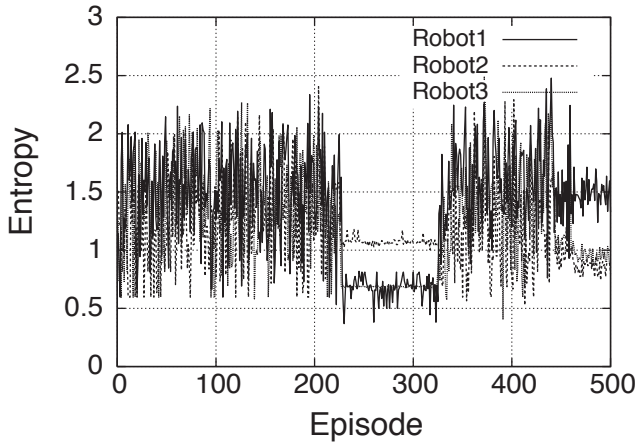
Fig. 6.8: Learning Result (Initialization: Robot2)



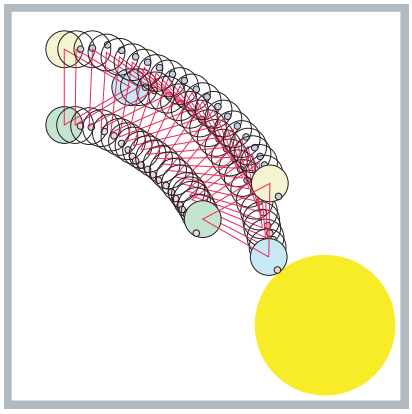
(a) Learning History



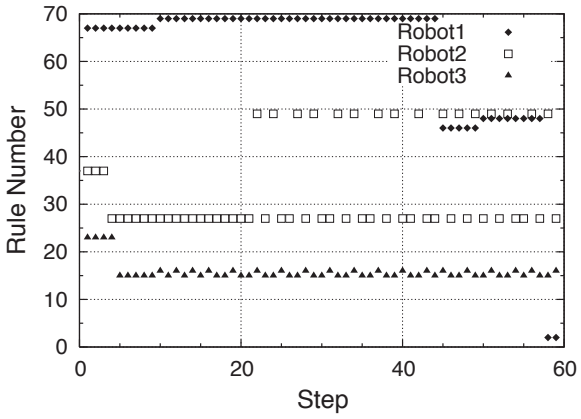
(b) Number of Rules



(c) Entropy



(d) Behavior



(e) Sequence of firing rules

Fig. 6.9: Learning Result (Initialization: Robot3)

考察

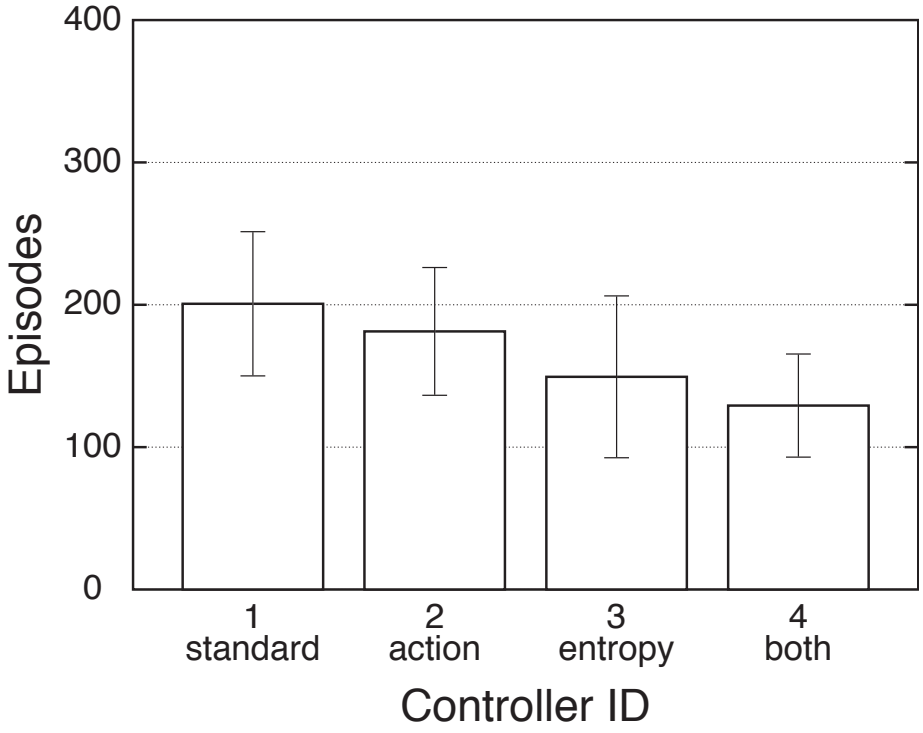
環境変動が生じてもシステムの挙動にあまり影響を与えない場合 (上記実験における Robot1・2 の交換時) は，ルールパラメータの更新したり，保持ルールを再利用するといったことで新しい環境にすぐに適応することができる．ルールを削除から保護することは，過学習の抑制ともにルール再利用の効率化に寄与する．また，リーダロボットの初期化といった，そのような知識の再利用だけでは不十分な大きな変動が生じた場合は，新たに有効なルールを探索する必要がある．このとき，提案手法では入力の一部として用いるエントロピが大きく変化する．そのため，保持しているルールでは未経験の状態となることで，効率的な探索への移行が可能となっている．

次に，第 5 章で提案した行動空間の効率的探索を目指した拡張との融合に関して検証する．Fig. 6.10 に，本章における前述の実験と同様に行動空間探索に関する拡張型 BRL にエントロピを導入 (入力付加とルール保護のいずれもを付加) した場合の実験結果を示す．Fig. 6.10(a) は初期状態からの学習，Fig. 6.10(b) は行動獲得後にリーダを初期化した後の再学習に要したエピソード数を示す．図のデータは左から，1：第 4 章で提案した制御器 (従来型 BRL)，2：第 5 章で提案した制御器 (行動空間探索の拡張を行った BRL)，3：本章で提案した制御器 (エントロピを導入した BRL)，および 4：第 5 章で提案した制御器にエントロピを導入した制御器のものである．初期状態からの学習，および再学習のいずれにおいても，両拡張を行った制御器が最もよいパフォーマンスを示している．

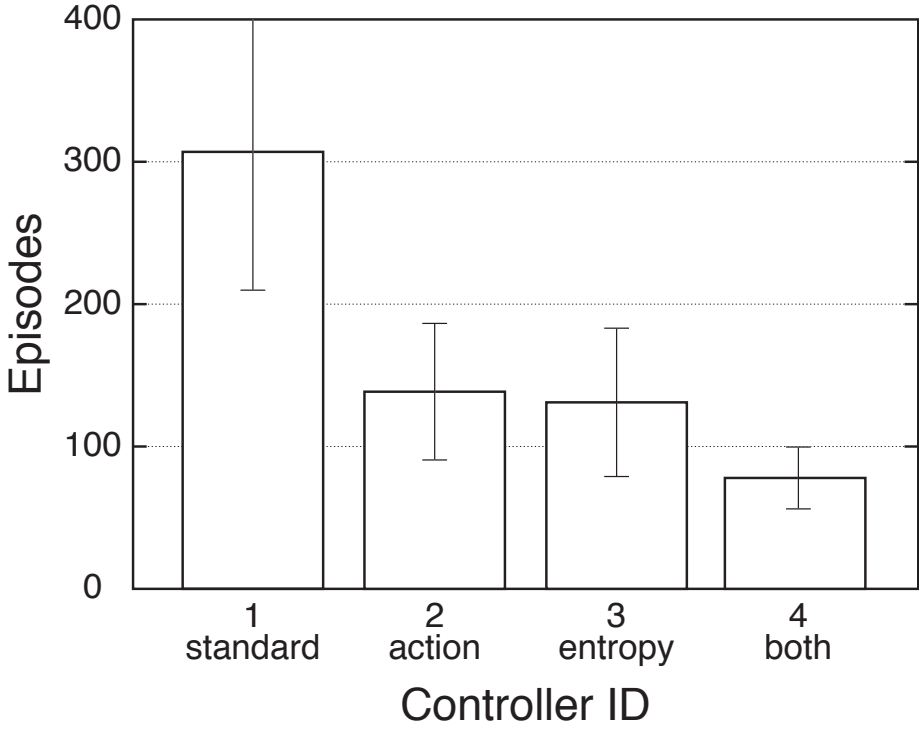
第 5 章における拡張は生成するルールの質を向上させる手法といえ，本章での拡張はルールを生成するタイミングを向上させる手法といえる．これらは，異なる部分を対象とした拡張であるため，干渉することなく独立して作用し，従来型 BRL と比較すると格段に性能が向上していることがわかる．

6.4 結言

本章では強化学習による MRS の協調行動獲得問題において，システムの環境変動への頑健性の向上のために，i) ルール発火の情報エントロピを入力の一部として用いて行動の安定性を考慮したルールを構成する，ii) エントロピに基づいて学習過程で有益であったルールを削除から保護するというアプローチを提案した．提案手法を三台のロボットの協調搬送問題に適用し，学習集束後の環境変動として未学習のロボットとの交換をしてシステムの挙動を観察した．保持しているルールを有効に利用するとともに再学習へ効率的に移行することで新しい協調行動を獲得していることを確認したことから，環境変動に対して頑健な MRS 構築のための一指針を示したといえる．



(a) Episodes required for learning



(b) Episodes required for relearning after an environmental change

Fig. 6.10: Performance comparison for BRLs

第7章　結論

生物のような知的な振る舞いの実現やその原理の解明には，“相互作用” が重要であり，その観点から多くの研究が行われている．この枠組において，ロボット間およびロボットと環境間の相互作用を通して，協調的に振る舞う MRS は，典型例のひとつである．その MRS において，本来期待される特徴である頑健性を実現するための概念として，自律的機能分化を導入した．また，機能分化を実環境で実現するために，計算知能のアプローチのなかでも教師データを必要とせずにオンラインで行動獲得が可能な手法である強化学習に着目した．本論文では，強化学習を MRS に適用するにおいて問題となる点を指摘し，それに対処するために **(a) 連続な状態・行動空間の自律的分割**，(b) **ダイナミクス許容量 (頑健性) の増大**，(c) **ダイナミクスの軽減**，および **(d) 試行錯誤の回数を削減**という観点から強化学習の機能拡張を行った．以下，各章で得られた研究結果を要約する．

第 2 章では観点 **(b)** から，MRS を制御対象とするに先だって，ロボット単体レベルで頑健性な意思決定を行う手法を提案した．まず，強化学習で問題となることとして，行動収束後に実験を繰り返すことによる頑健性の低下について指摘した．その問題に対処するため，強化学習ロボットの環境変動に対する頑健性向上のための手法として，確率ネットワークを用いたロボットの獲得戦略の保存・適用手法を提案した．この手法は，強化学習は環境内を探索して適切な入出力関係を収集するために用い，そのデータを基に確率ネットワークを用いて意思決定機構を構築するというものである．計算機実験と実機実験の結果，環境変化が起こる光源到達問題において，強化学習器のみを用いたロボットよりも頑健に振る舞うことを確認した．タスク達成可能な知識を確率ネットワークで表現することで，環境変化が生じて未経験な状態に陥っても，柔軟に対処することができた．この手法は強化学習で構築した入出力関係がセンサと行動間の結合関係という明示的な形式で表現されるという利点も有している．

第 3 章では観点 **(a)** と **(b)**，MRS における頑健な自律的機能分化を実現するための強化学習の拡張法を提案した．まず，状態・行動空間の離散化の困難性，および連続空間を取り扱う強化学習の研究例について述べた．次に，連続な状態行動空間を自律的に分割する機能を持つ強化学習法・BRL を取り上げ，その過学習問題を指摘した．その後，過学習を抑制してシステムの頑健性を向上させるために学習過程で有効であったルールを保護することでルール集合の多様性を維持する機構を付加した．その有効性を検証するために，三台のアーム型自律ロボットによる協調荷上げ問題に適用して実機実験を行った．その結果，可塑的な自律的機能分化が発現するなど頑健性が向上

し，環境変化に対して従来型 BRL よりも安定した大域的秩序を迅速に再形成していることを確認した．

第 4 章では観点 **(a)** と **(c)**，BRL を用いた MRS の協調行動獲得において，環境のダイナミクスを軽減することで学習をより安定化することで学習効率を向上させる手法を提案した．まず，MRS に強化学習を適用する場合の問題点を指摘し，それに対処するための従来手法を述べた．その後，次時刻における他ロボットの状態を予測する機構を構築し，その予測機構の出力を BRL の入力の一部として付加する手法を提案した．提案手法を自律移動ロボットによる協調搬送問題に適用した．計算機実験により予測情報を用いない場合よりも提案手法が効率的に学習していることを確認した．また，実環境においても，行動獲得に成功したことからノイズなどの不確定要素にも耐え得る頑健な学習が行えていることを確認した．

第 5 章では観点 **(a)**，**(c)** と **(d)** から，強化学習における試行錯誤の回数を極力削減したいという基本的要求から，行動空間の探索効率の向上を目的として BRL の拡張を行った．従来型 BRL におけるルール生成時の行動の決定法における問題を指摘した．機能拡張として，行動選択における事後確率の計算において新たなしきい値を設定し，近傍のルールのパラメータを基に新ルールを生成する手法を付加した．提案手法をロボット一台による光源到達問題に適用し，行動獲得および獲得後の環境変動に対する再学習に関する計算機・実機実験を行って基本性能の検証を行った．その後，協調搬送問題においても同様の実験を行った．いずれの実験でも従来型 BRL と比較して短いエピソードでの行動獲得を確認し，提案手法の有効性を示した．ここでは，実例に基づく強化学習の行動生成法を拡張することでロボットの自律性を向上させたといえる．

第 6 章では観点 **(a)**，**(b)**，**(c)** と **(d)** から，強化学習による MRS の協調行動獲得問題において，システムの環境変動への頑健性の向上のために，より迅速な再学習を目指して BRL の拡張を行った．i) ルール発火の情報エントロピを入力の一部として用いて行動の安定性を考慮したルールを構成する，ii) エントロピに基づいて学習過程で有益であったルールを削除から保護するというアプローチを提案した．提案手法を三台のロボットの協調搬送問題に適用し，学習集束後の環境変動として未学習のロボットとの交換をしてシステムの挙動を観察した．保持しているルールを有効に利用するとともに，エントロピの変化量を基に再学習へ効率的に移行することで新しい協調行動を獲得していることを確認した．このことから，ルールの生成を適切なタイミングで行うことによりロボットの適応能力が向上したといえる．また，第 5 章で拡張法と融合することで，従来型 BRL と比較してさらなるパフォーマンスの向上が確認された．

本研究では，以上のように五つのアプローチで，強化学習を制御器としたロボットの頑健性・自律適応能力の向上を実現した．対象とした MRS は，一般的に用いられる非均質なものではなく均質なものであった．そのため，機能分化にいたるまでにコストがかかることやタスク達成の効率が劣るといった問題がある．しかし，本来，MRS

環境は動的で複雑なものであり，将来の人間の生活環境などでの運用を考えた場合は，想定しないような状況に陥る可能性が十分に考えられる．そのような状況では，非均質なMRSではその非均質性が新しい協調関係を阻害する場合があるといえる．そのとき，自律的機能分化を発現することが重要となると考える．ところで，本研究ではロボット同士の協調に特化していたが，ロボットが均質性に起因する自由度の高さから，協調する相手はロボットに限定しないアプローチとも捉えることができる．つまり，将来的に，より多入力・多出力のシステムにおける迅速な行動獲得が可能となれば，様々な性格の人間との共同作業においても，柔軟に役割を発現することが期待できる．以上のように，本研究で示した自律的機能分化を発現する均質なMRSの協調行動獲得は，MRS分野のみならず，人間機械協調系への展開など，様々なレベルの相互作用を有するシステム構築における一指針になるといえる．

今後の展望

本研究では，MRSを構成する個々のロボットの制御をひとつの強化学習器のみで構成し，その学習能力を向上させることで環境変動などに対処している．さらに，タスクとしては，太田らの分類[145]のうちでも単純なものである単発・点到達の問題を対象としている．より複雑な問題・環境での適用を考えた場合，様々な状況で複数のタスクを実行するには，本研究が単一の学習器では限界が生じる．この問題に対し，近年，学習器を複数用意してモジュール性を持たせる手法が提案されている[146]-[149]．このように拡張することで，より迅速に環境変化に適応するだけでなくタスクの処理能力の向上，つまり，より高度な自律的機能分化の実現が期待される．

また，本研究では取り扱ってはいないが，MRSの制御において通信や交渉は有効な手段のひとつである[150]-[152]．特に，協調関係だけでなく競合関係が存在するような問題においてより重要性が増す．これは，マルチエージェントシステムの分野でも様々な研究が行われているが，いつ・なにを・どのように通信したらよいのかという点で，今のところ決定的な方法論はなく，現状は有効性は状況依存といえる．これに対処し，状況に応じた適応的な通信を構築することにより，MRSにおける自律的問題解決能力を向上させることができると考える．

参考文献

[1] 高玉ら：“特集・相互作用の本質にせまる：知的システムの理解と設計の新視点”，計測と制御，**44**-12，pp.817-882, (2005)

[2] 土屋ら：“特集・移動知：能動的な移動機能がもたらす創発的知能”，計測と制御，**44**-9，pp.579-645, (2005)

[3] R. Pfeifer and C. Scheier: “*Understanding Intelligence*”, The MIT Press, (1999)

[4] 浅田，石黒，國吉：“認知ロボティクスの目指すもの”，日本ロボット学会誌，**17**-1，pp.2-6, (1999)

[5] けいはんな社会的知能発生学研究会 (著)：“知能の謎：認知発達ロボティクスの挑戦”，講談社ブルーバックス，(2004)

[6] 伊藤：“自律分散宣言：適応・学習・進化システムと計算機知能 明日を拓くシステムパラダイム”，オーム社，(1995)

[7] 井原：“自律分散システム”，計測と制御，**26**-1，pp.476-481，(1987)

[8] 嘉数：“マルチエージェントシステムの研究動向”，システム/制御/情報，**41**-8，pp.291-296，(1997)

[9] P. Stone and M. Veloso: “Multiagent systems: A Survey from a Machine Learning Perspective”, Autonomous Robots, **8**-3, pp.345-383, (2000)

[10] T. Fukuda and Y. Kawauchi: “Cellular Robotic System (CEBOT) as One of the Realization of Self-organizing Intelligent Universal Manipulators”, Proc. of IEEE International Conference on Robotics and Automation, pp.662-667, (1990)

[11] A. Castano, W.M. Shen and P. Will: “CONRO: Towards Miniature Self-Sufficient Metamorphic Robots”, Autonomous Robots, pp.309-324, (2000)

[12] E. Şahin *et al.*: “SWARM-BOT: Pattern Formation in a Swarm of Self-Assembling Mobile Robots”, Proc. of The IEEE International Conference on Systems, Man and Cybernetics, (2002)

[13] 黒河，吉田，神村他：“変形し移動する自立モジュール型ロボット (M-TRAN)”，日本ロボット学会誌，**21**-8，pp.855-859，(2003)

[14] 清水，高橋，川勝，石黒：“創発現象を活用したモジュラーロボット：形態安定性および耐故障性検証に基づく実機デザイン”，第 17 回自律分散システム・シンポジウム資料，pp.31-36，(2005)

[15] E. Bonabeau, M. Dorigo, and G. Theraulaz: “*Swarm Intelligence: From Natural to Artificial Systems*”, Oxford University Press, (1999)

[16] T. Balch and R.C. Arkin: “Behavior-Based Formation Control for Multi-robot Teams”, IEEE Transactions on Robotics and Automation, **14**-6, pp.926-939, (1998)

[17] 太田：“適応的行動と協調の発現原理”，計測と制御，**44**-9，pp.628-633，(2005)

[18] C. Anderson and N. Franks: “Teams in animal societies, Bahavioural Ecology”, **12**-6, pp.534-540, (2001)

[19] L.E. Parker: “ALLIANCE: An Architecture for Fault Tolerant Multi-Robot Cooperation”, IEEE Transactions on Robotics and Automation, **14**-2, pp.220-240, (1998)

[20] P. Stone and R.S. Sutton: “Scaling Reinforcement Learning toward RoboCup Soccer”, Proc. of 18th International Conference on Machine Learning, pp.537-544, (2001)

[21] P. Stone and R.S. Sutton: “Task Decomposition, Dynamic Role Assignment, and Low-Bandwidth Communication for Real-Time Strategic Teamwork”, Artificial Intelligence, **110**-2, pp.241-273, (1999)

[22] L. Chaimowicz, M.F.M. Campos and V. Kumar: “Dynamic Role Assignment for Cooperative Robots”, Proc. of the IEEE International Conference on Robotics and Automation (ICRA2002), **1**, pp.293-298, (2002)

[23] L. Li, A. Martinoli and Y.S. Abu-Mostafa: “Emergent Specialization in Swarm Systems”, Proc. of Intelligent Data Engineering and Automated Learning, pp.261-266, (2002)

[24] 福田：“インテリジェントシステム：適応・学習・進化システムと計算機知能”，昭晃堂，(2000)

[25] L.A. Zadeh: “The Concept of a Linguistic Variable and its Application to Approximate Reasoning”, Information Science, **8**-3, pp.199-249, (1965)

[26] E.A. Mamdani: “Application of Fuzzy Algorithms for Control of Simple Dynamic Plant”, Proc of Institution of Electrical Engineers, **121**-12, pp.1585-1599, (1974)

[27] 北村：“ニューラルネットワーク応用の現状と展望：計測制御の立場から”，システム/制御/情報，**35**-1，pp.2-10，(1991)

[28] D.E. Goldberg: “*Genetic Algorithms in Search, Optimization, and Machine Learning*”, Addison-Wesley, (1989)

[29] T. Bäck: “*Evolutionary Algorithms in Theory and Practice*”, Oxford University Press, (1996)

[30] D. Fogel: “*Evolutionary Computation Toward a New Philosophy of Machine Intelligence*”, IEEE Press, (1995)

[31] D. Floreano, F. Mondada: “Automatic creation of an autonomous agent: Genetic evolution of a neural-network driven robot”, Proc. of the 3rd International Conference on Simulation of Adaptive Behavior(SAB’94): FROM ANIMALS TO ANIMATS, pp.421-430, (1994)

[32] R. Pfeifer: “Cognition :Perspectives from autonomous agents”, Robotics and Autonomous Systems, **15**, pp.47-70, (1995)

[33] I. Hervey: “Evolutionary Robotics and SAGA: The case for hill crawling and tournament selection”, Artificial life, **3**, pp.299-326, (1994)

[34] 近藤，石黒，内川，P. Eggenberger：“ニューラルネットワーク応用の現状と展望–計測制御の立場から”，計測自動制御学会論文集，**35**-11，pp.1407-1414，(1999)

[35] S. Nolfi and D. Floreano: “*Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines (Intelligent Robotics and Autonomous Agents)*”, Bradford Books, (2000)

[36] 浅田：“身体性による知能の発現”，日本人工知能学会誌，**13**-1，pp.14-15，(1998)

[37] 國吉，ベルテゥーズ：“身体性に基づく相互作用の創発に向けて”，日本ロボット学会誌，**17**-1，pp.29-33，(1999)

[38] 藤井，石黒，内川，青木，P.Eggenberger：“動的再編性機能を有する神経回路モデルを用いた歩容制御”，電気学会論文誌 C 分冊，pp.1567-1572，(1999)

[39] G. Baldassarre, S. Nolfi and D. Parisi: "Evolution of Collective Behaviour in a Team of Physically Linked Robots", Applications of Evolutionary Computing, pp.581-592. Springer Verlag, Heidelberg, Germany, (2003)

[40] M. Quinn, L. Smith, G. Mayley and P. Husbands: "Evolving Team Behavior for Real Robots", Proc. of EPSRC/BBSRC International Workshop on Biologically-Inspired Robotics, pp.217-224, (2002)

[41] R.S. Sutton and A. G. Barto: "*Reinforcement Learning: An Introduction*", MIT Press, (1998)

[42] L.P. Kaebling: "Reinforcement Learning: A survey", Journal of Artificial Intelligence Research, **4**, pp.237-285, (1996)

[43] 畝見："強化学習法とロボットへの応用"，日本ロボット学会誌，**13**-1，pp.51-56，(1995)

[44] 浅田："ロボットの行動学習"，日本機械学会論文集 C 編，**62**-602，pp.3746-3751，(1996)

[45] 浅田："ロボットの行動獲得のための能動学習"，情報処理，**38**-7，pp.583-586，(1997)

[46] 山村，宮崎，小林："強化学習法の特徴と発展の方向"，システム/制御/情報，**39**-4，pp.33-38，(1995)

[47] 木村，宮崎，小林："強化学習システムの設計指針"，計測と制御，**38**-10，pp.618-623，(1999)

[48] 浅田："強化学習の実ロボットへの応用とその課題"，日本人工知能学会誌，**12**-6，pp.831-836，(1997)

[49] C.J.C.H. Watkins: "Technical Note: Q-learning", Machine Learning, **8**, pp.55-68, (1992)

[50] G.A. Rummery and M. Niranjan: "On-line Q-learning Using Connectionist Systems", Technical Report CUED/F-INFENG/TR 166, Engineering Department, Cambridge University, (1994)

[51] R. S. Sutton: "Learning to Predict by the Methods of Temporal Difference", Machine Learning, **3**, pp.9-44, (1988)

[52] A.G. Barto, R.S. Sutton and C.W. Anderson: “Neuronlike adaptive elements that can solve difficult learning control problems”, IEEE Trans. on Systems, Man, and Cybernetics, **13**, pp.834-846, (1983)

[53] 畝見：“実例に基づく強化学習法”，日本人工知能学会誌，**7**-4，pp.141-151，(1992)

[54] J.J.Grefenstette: “Credit Assignment in Rule Discovery Systems Based on Genetic Algorithms; Machine Learning, **3**, pp.225-245, (1988)

[55] J.H. Holland: “Properties of the bucket brigade algorithm; Proc. of the 1st International Conference on Genetic Algorithms and Their Applications, pp.1-7, Lawrence Erlbaum Associates, (1985)

[56] 木村，小林：“Actor に適正度の履歴を用いた Actor-Critic アルゴリズム：不完全な Value-Function のもとでの強化学習”，日本人工知能学会誌，**15**-2，pp.267-275，(2000)

[57] 宮崎, 山村, 小林：“強化学習における報酬割当ての理論的考察; 人工知能学会誌, **9**-4, pp.580-587, (1994)

[58] 宮崎, 木村, 小林：“Profit Sharing に基づく強化学習の理論と応用; 人工知能学会誌, **14**-5, pp.800-807, (1999)

[59] T.M. Mitchell: “*Machine Learning*”, McGraw-Hill, (1997)

[60] 畝見：“実例に基づく強化学習法”，人工知能学会誌，**7**-4, pp.697-707, (1992)

[61] 畝見：“実例に基づく強化学習法による失敗しない制御方法の学習”，人工知能学会誌，**7**-6, pp.1001-1008, (1992)

[62] R.A. McCallum: “Instance-based state identification for reinforcement learning”, Advances in Neural Information Processing Systems,pp.377-384, (1995)

[63] R.A. McCallum: “Instance-based utile distinctions for reinforcement learning with hidden state”, Proc. of the Twelvth International Conference Machine Learning, Morgan Kaufmann, pp.387-395, (1995)

[64] M.M. Svinin, F. Kojima, Y. Katada and K. Ueda: “Initial Experiments on Reinforcement Learning Control of Cooperative Manipulators”, Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.416-422, (2000)

[65] 山田，大倉，上田：“強化学習による自律型アームロボットの協調行動獲得”，計測自動制御学会誌，**39**-3，pp.266-275，(2003)

[66] 大倉，川上，上田：均質な自律ロボット群による協調行動獲得問題：機能分化に基づくアプローチ；システム制御情報学会論文誌，**15**-9, pp. 451-458 (2002)

[67] 荒井：“マルチエージェント強化学習：実用化に向けての課題・理論・諸技術との融合”，日本人工知能学会誌，**16**-4，(2001)，pp.476-481

[68] A.W. Moore and C.G. Atkeson: “Memory-Based Reinforcement Learning: Converging with Less Data and Less Real Time”, *Machine Learning*, 13: 103-130, (1993)

[69] S. Suzuki, T. Tamura, and M. Asada: “Learning from conceptual aliasing caused by direct teaching”, Proc. of the IEEE International Conference on Systems, Man, and Cybernetics, pp.698-703, (1999)

[70] S. Mikami: “Prediction based reinforcement learning for dynamic environment”, Proc. of Intelligent Engineering Systems Through Artificial Neural Networks 7, pp.139-144, (1997)

[71] 中村，黒山，大倉他：“Instance-Based Classifier Generator による自律ロボットの行動獲得”， 日本ロボット学会，**17**-3，pp.61-69，(1999)

[72] 木村，山下，小林：“強化学習による 4 足ロボットの歩行動作獲得”，電気学会電子情報システム部門誌，**37**-12, pp.1147-1155 (2002)

[73] 伊藤，松野：“QDSEGA による多足ロボットの歩行運動の獲得”，日本人工知能学会誌，**17**-4，pp.363-372，(2002)

[74] R.S. Sutton: “Integrated Architectures for Learning, Planning, and Reacting Based on Approximating Dynamic Programming”, Proc. of the Seventh International Conference on Machine Learning, pp.216-224, (1990)

[75] 石井：“強化学習におけるランダムさの自己調節”，日本神経回路学会誌，**6**-4，pp.254-262，(2002)

[76] S. Kato and H. Matsuo: “A Theory of Profit Sharing in Dynamic Environment”, Proc. of the Sixth Pacific Rim International Conference on Artificial Intelligence, pp. 136-145, (2000)

[77] I. Szita, B. Takács and A. Lőrincz: “ϵ-MDPs: Learning in Varying Environments”, Journal of Machine Learning Research, **3**, pp.145-174, (2002)

[78] Z. Kalmár, I. Pólik and I. Szita: “Event-Learning and Robust Policy Heuristics”, Cognitive Systems Research, **4**, pp.319-337, (2003)

[79] 福田，船戸，新井：“多重強化学習法に基づく群ロボットシステムにおける環境変化認識”，日本機会学会論文集 (C 編)，**66**-643, pp.864-869, (2000)

[80] 港，浅田：“環境の変化に適応する移動ロボットの行動獲得”，日本ロボット学会誌，**18**-5，pp.706-712，(2000)

[81] 松井，犬塚，世木，伊藤：“強化学習結果の再構築への概念学習の適用”，人工知能学会論文誌，**17**-2，pp.135-144，(2002)

[82] 宮川：“グラフィカルモデリング”，朝倉書店，(1998)

[83] 本村：“ベイジアンネットによる確率的推論技術”，計測と制御，**42**-8，pp.649-654，(2003)

[84] 北越，塩谷，栗原：“ベイジアンネットを利用した強化学習エージェントの方策改善”，情報処理学会論文誌，**44**-11，pp.2884-2894，(2003)

[85] 山村：“Bayesian Network 上の強化学習”，第 24 回知能システムシンポジウム，pp.61-66，(1997)

[86] R.J. Williams: “Simple Statistical Gradient Following Algorithms for Connectionist Reinforcement Learning”, Machine Learning, **8**, pp.229-256, (1992)

[87] Stuart Russell and Peter Noving: “*Artificial Intelligence A Modern Approach Second Edition*”, Pearson Education International, (2003)

[88] R.A.McCallum: “Hidden State and Reinforcement Learning with Instance-Based State Indentification”, IEEE Transactions on Systems, Man and Cybernetics, **26**-3, pp.396-407, (1996)

[89] S.D. Whitehead: “A Complexity Analysis of Cooperative Mechanisms in Reinforcement Learning”, Proc. of the 9th National Conference on Artificial Intelligence, **2**, pp.607-613, (1991)

[90] 浅田，野田，細田：“ロボットの行動獲得のための状態空間の自律的構成”，日本ロボット学会誌，**15**-6，pp.889-892，(1997)

[91] J.S.Albs，小杉幸夫 [ほか] 訳：“ロボティクス：ニューロンから知能ロボットへ”，pp.151-198, (1984)

[92] R.S. Sutton: “Generalization in reinforcement learning: Successful Examples Using Sparse Coarse Coding”, Advances in Neural Information Processing Systems 8, MIT Press, pp.1038-1044, (1996)

[93] Y. Akisato, K. Suzuki and A. Ohuchi: “GA-Based Q-CMAC Applied to Airship Evasion Problem”, Journal of Robotics & Mechatronics, Special Issue of Complex Adaptive Systems, **10**-5, (1998)

[94] L.J. Lin: “Scaling Up Reinforcement Learning for Robot Control”, Proc. of the 10th International Conference on Machine Learning, pp.182-189, (1993)

[95] 深尾，稲山，足立：“正則化理論を用いた連続的状態空間と行動を扱う強化学習”，システム制御情報学会論文誌，**11**-11, pp.593-599，(1998)

[96] Y. Takahashi, M. Takeda, and M. Asadad: “Continuous Valued Q-learning for Vision-Guided Behavior Acquisition”, Proc. of the 1999 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, pp.255-260, (1999)

[97] M. Takeda, T. Nakamura, M. Imai, T. Ogasawara, and M. Asada: “Enhanced Continuous Valued Q-learning for Real Autonomous Robots; Proc. of Supplement”, Proc. of The Sixth International Conference on Simulation of Adaptive Behavior (SAB00): FROM ANIMALS TO ANIMATS, pp.195-202, (2000)

[98] 堀内，藤野，片井，椹木：“連続値入出力を扱うファジィ内挿型 Q-learning の提案”，計測自動制御学会論文集，**35**-2，pp.271-279，(1999)

[99] 梅迫，大林，小林：“自己組織化型ファジィ強化学習システム”，計測自動制御学会論文集，**39**-7，pp.699-701，(2003)

[100] S. Schaal and C.G. Atkeson: “From Isolation to Cooperation; An Alternative View of a System of Experts”, Advances in Neural Information Processing Systems, **8**, pp.605-611, (1996)

[101] J. Boyan and A. Moore: “Generalization in Reinforcement Learning; Safely Approximating the Value Function”, Proc. of Neural Information Processings Systems 7, Morgan Kaufmann, pp.369-376, (1995)

[102] K. Doya: “Reinforcement learning in continuous time and space”, Neural Computation, **12**, pp.219-245, (2000)

[103] J. Morimoto and K. Doya: “Reinforcement Learning of Dynamic Motor Sequence: Learning to Stand Up”, Proc. of International Conference on Intelligent Robots and Systems, pp.1721-1726, (1998)

[104] 森本，銅谷：“強化学習を用いた高次元連続状態における系列運動学習：起き上がり運動の獲得”，電子情報通信学会論文誌 D-II, **J82-D2**-11，pp.2118-2131，(1999)

[105] J. Morimoto and K. Doya: “Acquisition of Stand-up Behavior by a Real Robot using Hierarchical Reinforcement Learning”, Proc. of International Conference on Machine Learning, pp.623-630, (2000)

[106] 近藤，伊藤：“進化的 recruitment 戦略を用いた強化学習による自律移動ロボットの制御器設計”，計測自動制御学会論文集，**39**-9，pp.857-864，(2003)

[107] 鮫島，大森：“強化学習における適応的状態空間構成法”，日本神経回路学会誌，**6**-3，pp.144-154，(1999)

[108] 石井，佐藤：“正規化ガウス関数ネットワーク，Mixture of exports と EM アルゴリズム”，日本神経回路学会誌，**6**-1，pp.30-40，(1999)

[109] J. Yoshimoto, S. Ishii and M. Sato: “On-line EM Reinforcement Learning”, Proc. of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN 2000), **III**, pp.163-168, (2000)

[110] M. Sato, S. Ishii: “On-line EM algorithm for the normalized Gaussian network”, Neural Computation, **12**-2, pp.407-432, (2000)

[111] 柴田，岡部，伊藤：“ニューラルネットワークを用いた Direct-Vision-Based 強化学習”，計測自動制御学会論文集，**37**-2，pp.168-177，(2001)

[112] 柴田，岡部：“時間軸スムージング学習”，電気学会論文誌 C 分冊，**117-C**-9，pp.1291-1299，(1997)

[113] 柴田，西野，岡部：“Actor-Q アーキテクチャに基づく能動認識学習システム”，電子情報通信学会論文誌，**J84-D-II**-9，pp.2121-2130，(2001)

[114] 深尾，大村，足立：“Q-learning における状態空間の適応的分割法”，計測自動制御学会論文集，**37**-3，pp.242-249，(2001)

[115] 小林，太田，井上，新井：“視覚情報を用いた状態・行動空間の自律的生成”，計測自動制御学会論文集，**36**-11，pp.1029-1036，(2000)

[116] M. Asada, S. Noda and K. Hosoda: “Action-Based Sensor Space Categorization for Robot Learning”, Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS96), pp.1502-1509, (1996)

[117] H. Ishiguro, R. Sato, and T. Ishida: “Robot Oriented State Space Construction”, Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS96), pp.1496-1501, (1996)

[118] Y. Takahashi, M. Asada, and K. Hosoda: “Reasonable Performance in Less Learning Time by Real Robot Based on Incremental State Space Segmentation”, Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS96), pp.1518-1524, (1996)

[119] 高橋，浅田：“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，**17**-1，pp.118-124，(1999)

[120] A. Ueno, H. Takeda, and T. Nishida: “Learning of the way of abstraction in real robots”, Proc. of 1999 IEEE International Conference on Systems, Man, and Cybernetics (SMC99), **1**, pp.II746-II751, (1999)

[121] 上野，中須，堀：“自律エージェントのための状況認識と行動規則の同時学習”，人工知能学会誌，**15**-2，pp.297-308，(2000)

[122] T. Yairi, S. Nakasuka, and K. Hori: “Sensor Fusion for State Abstraction Using Bayesian Classifier”, Proc. of IEEE International Conference on Intelligent Engineering Systems (INES’98), pp.99-104, (1998)

[123] 矢入，堀，中須賀：“複数行動結果を考慮した最尤推定に基づく状態一般化法”，人工知能学会誌，**16**-1，pp.130-140，(2001)

[124] 鳥脇：“認識工学：パターン認識とその応用”，コロナ社，(1993)

[125] 石井，上田，前田，村瀬：“わかりやすいパターン認識”，オーム社，(1998)

[126] 繁桝：“ベイズ統計入門”，東京大学出版会，(1985)

[127] 松原，縄田，中井：“統計学入門”，東京大学出版，(1991)

[128] 尾川 ，並木，石川：“学習進度を反映した割引率の調整”，電子情報通信学会ニューロコンピューティング研究会電子情報通信学会技術研究報告，**NC2002**-129，pp.73-78，(2003)

[129] 阪口：“動きの予測を伴う能動的認識のアルゴリズム”，ロボット学会誌，**12**-5，pp.708-714，(1994)

[130] 三上：“強化学習のマルチエージェント系への応用”，日本人工知能学会誌，**13**-4，pp.609-618，(1998)

[131] M. Tan: “Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents”, Proc. of the Tenth International Conference on Machine Learning, pp.330-337, (1993)

[132] M. Asada, E. Uchibe, and K. Hosoda: “Cooperative Behavior Acquisition for Mobile Robots in Dynamically Changing Real Worlds via Vision-Based Reinforcement Learning and Development”, Artificial Intelligence, **110**, pp.275-292, (1999)

[133] S. Ikenoue, M. Asada, and K. Hosoda: “Cooperative Behavior Acquisition by Asynchronous Policy Renewal that Enables Simultaneous Learning in Multiagent Environment”, Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.2728-2734, (2002)

[134] O. Buffet, A. Dutech, and F. Charpillet: “Incremental Reinforcement Learning for designing Multi-Agent Systems”, Proc. of the Fifth International Conference on Autonomous Agents (Agents’01), (2001)

[135] S. Elfwing, E. Uchibe, K. Doya, and H.I. Christensen, “Multi-Agent Reinforcement Learning: Using Macro Actions to Learn a Mating Task”, Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.3164-3169, (2004)

[136] M.J. Matarić: “Reinforcement Learning in the Multi-Robot Domain”, Autonomous Robots, **4**-1, pp.73-83, (1997)

[137] M.L. Littman: “Markov Games as a Framework for Multi-Agent Reinforcement Learning”, Proc. of Eleventh International Conference on Machine Learning, pp.157-163, (1994)

[138] J. Hu and M.P. Wellman: “Multiagent Reinforcement Learning: Theoretical Framework and an Algorithm”, Proc. of Fifteenth International Conference on Machine Learning, pp.242-250, (1998)

[139] Y. Nagayuki, S. Ishii, and K. Doya: “Multi-Agent Reinforcement Learning: An Approach Based on the Other Agent’s Internal Model”, Proc. of Fourth International Conference on Multi-Agent Systems, pp.215-221, (2000)

[140] 川上，大倉，上田：“マルチエージェント環境における強化学習の一適用法”，日本機械学会論文誌 C 編，**69**-677，pp.212-218，(2003)

[141] 太田，武衛，新井，大隅，陶山：“2 台の移動ロボットの協調による搬送制御”，日本ロボット学会誌，**14**-2，pp.263-270，(1996)

[142] 小菅，大住，千葉：“単一物体を操る複数移動ロボットの分散協調制御”，日本ロボット学会誌，**16**-1，pp.957-964，(1998)

[143] R. Ghanea-Hercock, D. P. Barnes: “Evolved Fuzzy Control System for Cooperation, International Journal of Advanced Robotics, Special Issue on Learning and Behaviors in Robotics, pp.599-607, (1996)

[144] V. Masek, M. Kajitani, A. Ming, C. Kanamori: “Local Path Planning for Cooperative Mobile Robots”, Proc. of JSME Annual Conference on Robotics and Mechatronics, **A**, pp.734-737, (1996)

[145] 太田，新井：“群知能ロボットシステム”，日本ロボット学会誌，**20**-5，pp.487-490，(2002)

[146] D.M. Wolpert and M. Kawato: “Multiple Paired Forward and Inverse Models for Motor Control”, Neural Networks, **11**, pp.1317-1329, (1998)

[147] 鮫島，片桐，銅谷，川人：“モジュール競合による運動パターンのシンボル化と見まね学習”，電気情報通信学会論文誌 D-II，**J85-D-I I**-1，pp.90-100，(2002)

[148] 枝澤，高橋，浅田：“複数学習器を用いたマルチエージェント環境における行動獲得”，第 22 回日本ロボット学会学術講演会 CD-ROM，(2004)

[149] 谷口，椹木：“身体と環境の相互作用を通した記号創発”，システム制御情報学会論文誌，**18**-12，pp.440-449，(2005)

[150] M.J. Matarić, G.S. Sukhatme, and E.H. Ostergaard: “Multi-Robot Task Allocation in Uncertain”

[151] N. Kalra, D. Ferguson, and A. Stentz: “Hoplites: A Market-Based Framework for Planned Tight Coordination in Multirobot Teams”, Tech. report CMU-RI-TR-04-41, (2004)

[152] A. Gage, R. Murphy, K. Valavanis, and M. Long: “Affective Task Allocation for Distributed Multi-Robot Teams”, IEEE Transactions on Robotisc, (2004)

謝辞

研究の機会を提供していただき，懇切なる御指導を賜わった神戸大学 田浦俊春教授に深甚なる感謝の意を表します．本論文を作成するにあたり貴重な御教示を賜わりました小島史男教授，大須賀公一教授に謹んで感謝の意を表します．本研究を進めるにあたり，常に細やかで適切な御指導と御教示を賜わりました大倉和博助教授に心より感謝の意を表します．また，学部生時代より博士前期課程まで研究に関して幅広く御指導を賜わった上田完次教授，Mikhail Svinin 先生，鳩野逸生教授，右田正夫助教授，長坂一郎助教授に謹んで感謝の意を表します．さらに，研究生活面で多くの貴重なご助言を賜わりました摂南大学 諏訪晴彦助教授，神戸市立工業高等専門学校 尾崎純一助教授に感謝の意を表します．

苦楽をともにした Robotics & Collective Intelligence Group と旧 Robot Group の皆さんに心より感謝と敬意を表します．山田和明講師，川上賢一郎氏の貴重な御指導と御協力を受け，その偉大な研究成果を引き継ぐことでこの論文は完成に至りました．松村嘉之講師には，常に親身になって温かい助言をいただきました．片田喜章講師には，研究のみならず様々な面で常に厳しくも愛情ある御指導をいただきました．また，多大なる御指導と協力を頂いた先輩・木下慎太郎氏，牛尾将蔵氏，河内達麿氏，矢島英明氏，安田豊氏，伍賀征典氏に心より感謝致します．旧機械棟 112 号室横の実験部屋で共に悪戦苦闘した児島史周氏，ならびに同輩の河田洋平氏，石井洋司氏に感謝の意を表します．研究に協力して頂いた篠原一真氏，鷲崎亮太氏，平松久征氏，大嶋力氏，瀧川孝輔氏，仁宇昭雄氏に心より感謝致します．また，良き後輩の五十嵐隆史氏，太田将治氏，弘津雄三氏，藤本圭介氏，加藤豊氏，植田直樹氏，谷口友紀氏，山崎謙太氏，岩波悟史氏，高崎真也氏，尾上豪啓氏，足立昌彦氏，松浦芳樹氏，宮川竜治氏，横田英二氏，西原志乃氏，福森淳一氏，三宅慧介氏，戸田禎孝氏，赤尾剛志氏，高田新一氏，中村明彦氏に感謝致します．

私が長きにわたり在籍致しました神戸大学工学部機械工学科知能システム創成学研究室 (前 MI-2)，および創造設計工学研究室 (MA-5) の歴代諸氏には，さまざまな面で御協力，激励を頂きました．沖田淳也氏，張明氏，藤井信忠助教授，岩崎敦助手，井寄幸平氏，佐藤修一氏，瀬良香織氏，吉川達也氏，Zlatan Car 助教授，西野成昭氏，Attila Lengyel 氏，中西大介氏をはじめとする諸先輩方，同輩の内田潤青氏，小倉武哲氏，津田和城氏，ペ正樹氏，細井智明氏，良き後輩である三宅史朗氏，吉村悠紀氏をはじめとする諸氏に感謝いたします．また研究室生活のさまざまな面で御協力いただきました歴代秘書の方々に感謝の意を表します．

報徳学園中学校・高等学校の友人諸氏と恩師，神戸大学バレーボールサークル PVC のメンバー各位，その他，これまでの学生生活において出会った全ての方々に深く感謝致します．また，わずかの期間でしたが在籍したミノルタ株式会社 (現 コニカミノルタ) の 2002 年度入社の友人諸氏，および OC 二課 (当時) とその関係部署のみなさまに感謝の意を表します．

最後に，私のわがままを許していただき，あらゆる面で長年に渡り支え励ましてくださった両親と姉に心より深く感謝致します．

平成 18 年 2 月吉日
保田 俊行

付 録A　試作したロボット

A.1　アーム型ロボット

第 3 章の研究では, 実験用ロボットとして Lynxmotion 社から市販されているアームロボット (Fig. A.1) を使用した．このアームロボットはアーム部分の 5 Axis Robot Arm Kit と台座部分の Mobile Arm Robot Kit からなり，それぞれ組み立てキットである．アーム型ロボットを構成する CPU ボード，センサ，駆動部，および電源について示す.

A.1.1　CPU ボード

ロボットそれぞれが CPU ボードを搭載しており，独立に制御される．CPU ボードは，SH7032(SH-1) が搭載された京都マイクロコンピューター株式会社の KZ-SH1-01 を使用している (Fig. A.2)．SH7032 は，RISC(Reduced Instruction Set Computer) 方式の CPU により高性能な演算処理を実現し，システム構成に必要な周辺機能を集積す

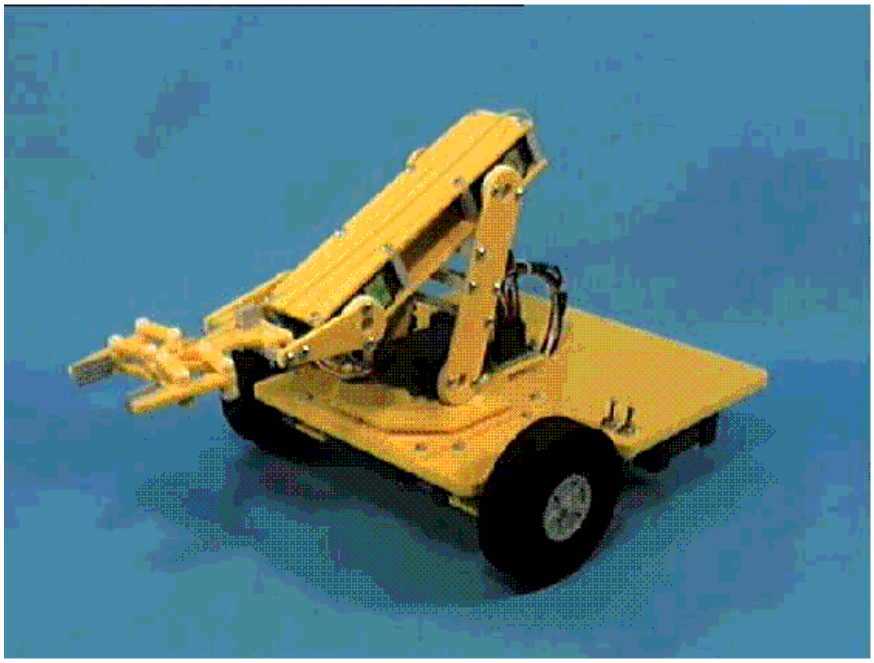


Fig.　A.1: Lynxmotion arm robot

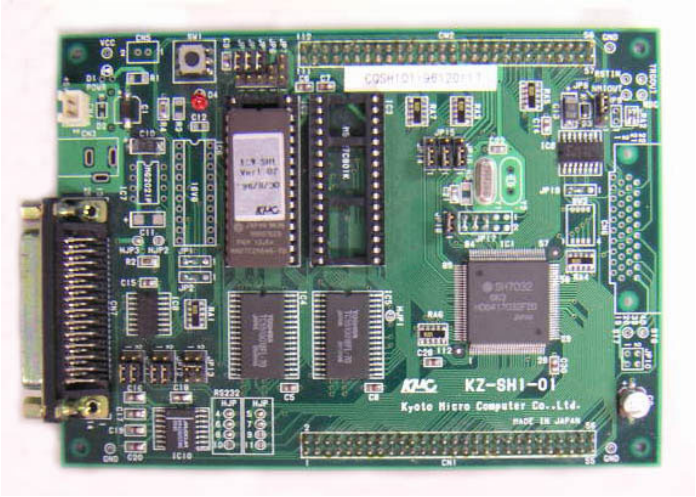


Fig. A.2: CPU board (SH-1)

るとともに，携帯機器応用に不可欠な低消費電力を実現するシングルチップマイコンである．SH7032 は最小部品点数でユーザーシステムを構成できるように周辺機器としてシリアル・コミュニケーション・インターフェース (SCI)，A/D コンバータ (ADC)，I/O ポート，ピンファンクションコントローラ (PFC) 等を内蔵している．以下に，それらの詳細を述べる．

1. **シリアル・コミュニケーション・インターフェース (SCI)**
   SCI は 2 チャネルあり，同期/非同期のインターフェースが可能である．一方のチャネルを Mini SSC-II との通信に，他方を PC との通信に使用している．CPU(SH-1) と Mini SSC-II 間，CPU(SH-1) と PC 間においてシリアル通信を行う際，信号のレベルを TTL(SH-1) から RS232C(Mini SSC-II，PC) へ変換しなければならない．そこで信号レベルの変換を行うため，TTL/RS232C の相互変換が可能な MAX233 チップを使用する．

2. **A-D コンバータ (ADC)**
   ADC は 10 ビットの変換精度でアナログ入力をデジタルに変換する機能であり SH-1 は 8 チャネルの入力を扱うことができる．本研究では，測距センサ 1 個に対して 1 チャネル，傾斜センサ用に 1 チャネルを割り当て，合計 3 チャネル使用している．

3. **ピンファンクションコントローラ (PFC)**
   SH7032 は 16 本の入出力兼用ポートと 8 本の入力専用ポートを持っている．これらの端子はバス制御信号と SCI，ADC などの内蔵周辺機器との兼用端子になっており，PFC で端子の機能を切替える．本研究では，周辺機器として SCI，ADC を使用している．これらの機能を PFC により使い分けている．

A.1.2　センサ

ロボットが自律的に目的の行動を獲得するためには自分や外界の状態を認識する必要がある．本研究では，ロボットのアームの先端位置を測定するための測距センサと関節角度と荷の傾きを測定するためのポテンショメータを使用する．以下に使用したセンサについて述べる．

1. **測距センサ**
   シャープ株式会社の測距センサ GP2D12 を用いる (Fig. A.3)．GP2D12 は Prosition Sensitive Detector，赤外線発光ダイオード，及び信号処理回路を一体化した普及型の測距センサである．特徴として，反射物の色，反射率，周囲の明るさによる影響を受けにくく，精度の良い測距が可能であること，外部制御回路を用いずにマイコンに直接取り付けることが可能であることなどが挙げられる．このセンサは測定距離により異なる電圧を出力する．その電圧を CPU の A/D 変換器に読み込み，10 ビットの分解能 (0〜1023) でデジタル信号に変換している．

2. **ポテンショメータ**
   巻線一回転ポテンショメーターを用いる (Fig. A.4)．このポテンショメータの軸は一回転することができ，抵抗値はこの軸の回転角度に比例して直線的に変化する．CPU の A/D 変換器に電圧を読み込み，デジタル信号に変換している．

A.1.3　駆動部分

1. **Mini SSC II**
   Lynxmotion 社のサーボコントローラである Mini SSC II を用いる (Fig. A.5)．SSC(serial servo controller) は，あらかじめ小さく集積された専用のサーボコン

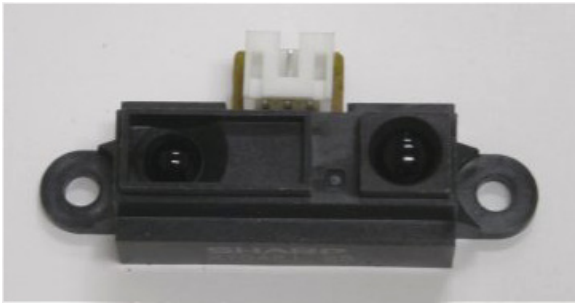


Fig. A.3: IR Sensor

Fig. A.4: Potentiometer for an arm-type robot

トローラである．これ一個でマイクロコントローラや PC シリアルポート等を用いて最大八個までのサーボモータを制御することができる．

2. サーボモータ Futaba 社のコアレスサーボモータ FP-S9206 を用いている (Fig. A.6)．サーボモータはコンパクトで制御しやすく，360 度以上回転しないのでアーム関節に適している．モータの構成要素は，ギア列，軸の回転を抑制するストッパ，位置フィードバック用ポテンショメータ，および位置制御用回路である．

A.1.4　電源

アーム型ロボットを動作させるためには，MiniSSC-II 用の 9V マンガン電池 (Fig. A.7)，および CPU ボードとサーボモータ用に 5V の AC 電源が必要となる．5V の AC 電源は，外部より有線で供給する．

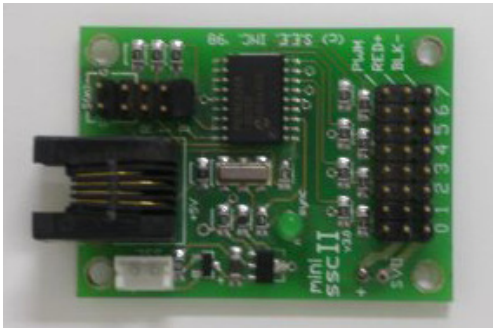


Fig.　A.5: Mini SSC II

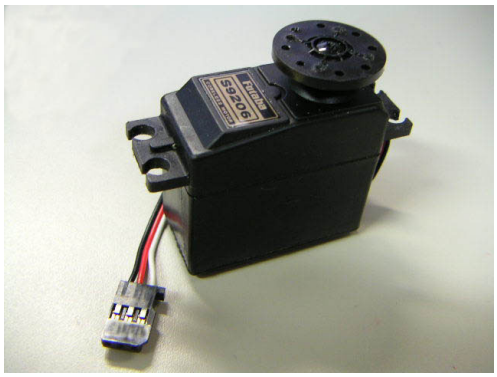


Fig.　A.6: Servo motor

Fig.　A.7: Battery for Mini SSC II

A.2　車輪移動ロボット

第 4，5 章の実機実験で用いた自律車輪移動ロボットを Fig. A.8 に示す．このロボットは第 2 章で用いた小型自律移動ロボット Khepera を模して設計したものである．車輪移動ロボットに搭載した CPU ボード，センサ，駆動部，および電源について示す．

A.2.1　CPU ボード

ロボットを制御する CPU ボードには，SH-1 を機能拡張した SH-2(SH7050) が搭載された YellowSoft 社の SH7050F CPU ボード YS50-1 を使用している (Fig. A.9)．基本命令は 1 命令/1 ステートで動作し，20MHz 動作時には 1 命令 50nsec で実行する．また，乗算器を内蔵しており，32 桁 ×32 桁の乗算を 100〜200nsec で実行する．さらに，128KB のフラッシュROM と 6KB の RAM を内蔵している．周辺機能として，SH-1 と同様に SCI，ADC，PFC などがある．SCI は 3 チャネル，ADC は 16 チャネル備えている．なお，MAXIM 社の MAX233 を用いて SH-2 とコンピュータ間，SH-2 とサーボコントローラ間でシリアル通信を行う場合の信号レベルを TTL(SH-2) から RS232C(コンピュータ，サーボコントローラ) へ変換している．

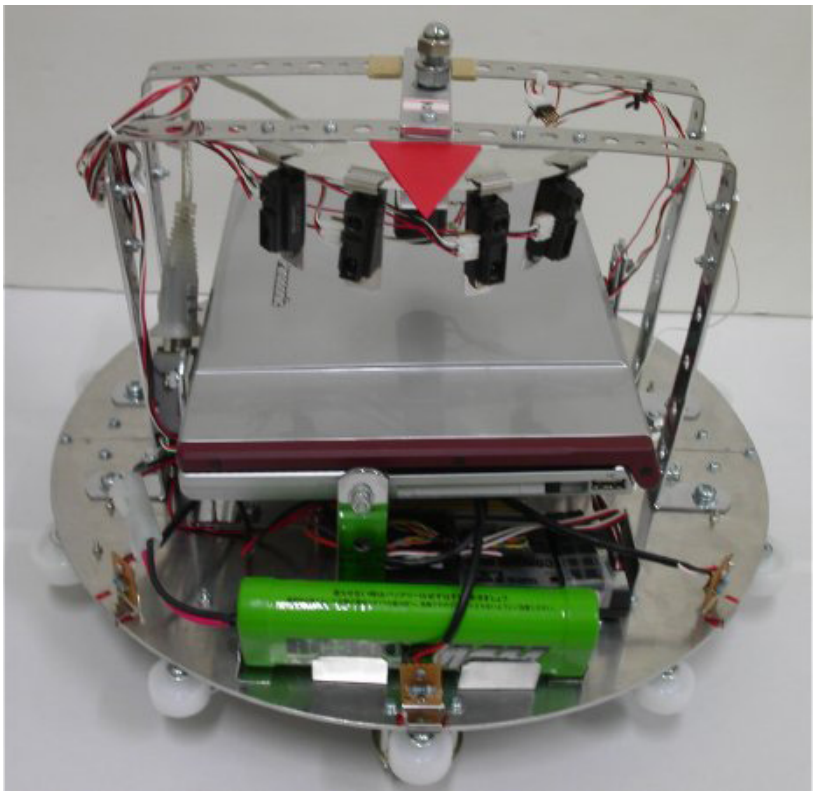


Fig.　A.8: Our hand-made autonomous mobile robot

A.2.2　センサ入力部

1. **測距センサ**
   シャープ株式会社の測距センサ GP2D12 を用いる．

2. **光センサ**
   光センサとして Cds セルを用いる．Cds セルは，光を照射させると抵抗値が減少する光導電効果を利用した光抵抗器である．特徴として，小型で低価格であり，実装回路が簡単に構成できる点が挙げられる．CPU の A/D 変換器に電圧を読み込み，デジタル信号に変換している．

3. **ポテンショメータ**
   巻線多回転ポテンショメーターを用いる (Fig. A.10)．このポテンショメータの軸は 10 回転することができ，抵抗値はこの軸の回転角度に比例して直線的に変化する．CPU の A/D 変換器に電圧を読み込み，デジタル信号に変換している．ロボットの正面とポテンショメータの中心 (センサ値が 512 となる位置) が一致するように取り付けている．

A.2.3　駆動部分

1. **Mini SSC II**
   アーム型ロボットと同じく，Lynxmotion 社のサーボコントローラである Mini SSC II を用いる．車輪ロボットでは，SSC は CPU からの制御信号を受け，その信号をパルスに変換して DMD コントローラに送信する．

2. **DMD コントローラ**

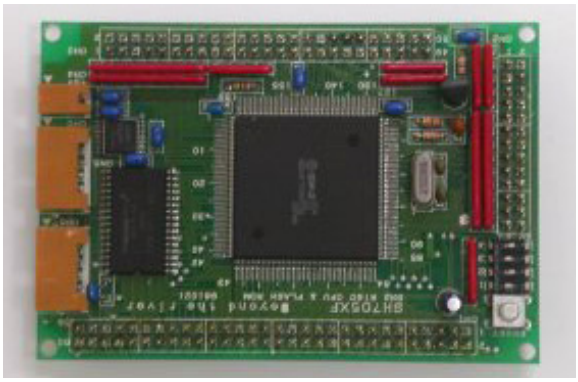


Fig. A.9: CPU board (SH-2)

Fig. A.10: Potentiometer for a mobile robot

株式会社タミヤの DMD(DIGITAL TWIN MOTOR DIFFERENTIAL) コントローラユニット T-01[1](第 4.4.5 節における二台の実機実験) と T-03(以降の実験) を用いる．DMD ユニットは Mini SSC II からの信号を受け，二つのモータを制御する．T-01 では，二つのモータそれぞれの回転速度を独立して制御する．T-03 では，二つのモータの組合せとしてのステアリング量とスロットル量を制御する．DMD コントローラはモータの回転数を調整し，ロボットの動作と進行方向を制御する．

3. **DC モータ**
   株式会社タミヤのタミヤ ギヤードモータ・3633k300(第 4 章)/3633k75(第 5 章) を用いる．3633k300 は，ギヤ比 1/300 で DC12V の最大効率時トルク 30.0kg·cm，無負荷回転数 39rpm の性能を持つ．3633k75 のギア比は 1/75 である．ギア比とトルクは反比例，ギア比と回転数は比例の関係にある．

4. **スポンジタイヤ**
   第 4 章の実験では，タイヤには株式会社タミヤのブチルスポンジタイヤ・ミディアム (F1-1 後輪用 RD3645) を用いた (Fig. A.11(a))．直径が 60mm のスポンジ製である．また RD ディシュホイールセットを用いて，タイヤとモータを接続している (Fig. A.11(b))．

5. **オムニホイール**
   第 5 章の実験では，株式会社土佐電子のオムニホイール (TYPE2570) を用いた (Fig. A.12(a))．オムニホイールの直径は 45mm である．オムニホイールと DC モータを接続・固定する部品を試作し，オムニホイールと DC モータを接続した (Fig. A.12(b))．

(a) Sponge tire

(b) Sponge tire connected with a DC Motor

Fig. A.11: Sponge tire and DC motor

[1]平成 14 年の時点で製造終了．

A.2.4　電源

CPU と Mini SSC II のために 9V 電池を一個，モータ駆動のためにサンヨー製 7.2V アドバンスパック RC3000MH(Fig. A.13) を一個を搭載している．これらにより，この自律移動ロボットは，外部と有線による常時電源供給の必要が無く，制限をうけることなく動作できる．

(a) Omni wheel

(b) Omni wheel connected with a DC motor

Fig. A.12: Omni wheel and DC motor

Fig. A.13: Battery for motors

付 録B　研究業績

学術論文

1. Toshiyuki Yasuda, and Kazuhiro Ohkura: “Autonomous Role Assignment in Homogeneous Multi-Robot System”, Journal of Robotics and Mechatronics, Vol. 17, No.5, pp. 596-604, (2005)

2. Toshiyuki Yasuda, Kazuhiro Ohkura, and Kanji Ueda: “A Homogeneous Mobile Robot Team That Is Fault-Tolerant”, the Advanced Engineering Informatics Journal, Elsevier, (To be published)

3. 保田俊行，大倉和博：“実例に基づく強化学習法の頑健性向上に関する一考察：マルチロボットシステムによる検証”，計測自動制御学会論文集，(投稿中)

4. 保田俊行，大倉和博：“確率ネットワークを用いた強化学習エージェントの獲得戦略の保存”，日本機械学会論文集 (C 編)，(投稿予定)

国際会議 (査読付)

1. Toshiyuki Yasuda, Kazuhiro Ohkura and Toshiharu Taura: “Cooperative Behavior Acquisition Mechanism for a Multi-Robot System Based on Reinforcement Learning in Continuous Space”, Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, pp.1539-1544, (2003)

2. Toshiyuki Yasuda, Kazuhiro Ohkura and Kanji Ueda: “A HOMOGENEOUS MOBILE ROBOT TEAM THAT IS FAULT-TOLERANT”, Proceedings of The 5th International Workshop on Emergent Synthesis, pp.139-146, (2004)

3. Toshiyuki Yasuda, Kazuhiro Ohkura and Toshiharu Taura: “Autonomous Role Assignment in Homogeneous Multi-Robot Sytems”, Proceedings of The 4th International Conference on the Advanced Mechatronics, pp.589-594, (2004)

4. Toshiyuki Yasuda, Kazuhiro Ohkura and Toshiharu Taura: “On the Performance of Reinforcement Learning for a Multi-Robot Team”, Proceedings of the

4th International Symposium on Human and Artificial Intelligence Systems, pp.354-358, (2004)

5. Toshiyuki YASUDA and Kazuhiro OHKURA: “Improving the Robustness of Reinforcement Learning for a Multi-Robot System Environment”, Proceedings of The 4th IEEE International Workshop on Soft Computing as Transdisciplinary Science and Technology (WSTST’05), pp.263-272, Springer Verlag, (2005)

6. Toshiyuki Yasuda and Kazuhiro Ohkura: “Behavior Acqusition Based on Reinforcement Learning with an Adaptive Action Generator”, Proceedings of The 6th International Workshop on Emergent Synthesis, pp.139-146, (To be submitted)

7. Toshiyuki Yasuda and Kazuhiro Ohkura: “Robust Instance-Based Reinforcement Learning with an Adaptive Action Generator”, Proceedings of The 9th International Conference on the SIMULATION OF ADAPTIVE BEHAVIOR (SAB’06), (To be submitted)

口頭発表

1. 保田俊行，大倉和博，上田完次：“強化学習による均質なマルチロボットの協調的行動獲得：学習の解析”，2003 年度精密工学会春季大会学術講演論文集，pp.514，(2003)

2. 保田俊行，大倉和博，上田完次，田浦俊春：“身体性認知に基づくマルチロボットシステムの設計”，日本機械学会生産システム部門講演会 2003 講演論文集，pp.55-56，(2003)

3. 保田俊行，大倉和博，上田完次，田浦俊春：“連続空間における強化学習法を用いたマルチロボットシステムの協調行動獲得”，ロボティクス・メカトロニクス講演会 CD-ROM 予稿集，2P1-3F-A7，(2003)

4. 保田俊行，大倉和博，田浦俊春：“連結された 3 台の移動ロボットの自律的機能分化に基づく協調行動獲”，第 16 回自律分散システム・シンポジウム講演論文集，pp.271-276，(2004)

5. 保田俊行，大倉和博，田浦俊春：“均質な自律移動ロボット群におけるロボットの追加・削除に対する頑健性”，2004 年度精密工学会春季大会学術講演会講演論文集 CD-ROM，pp.1307-1308，(2004)

6. 保田俊行，大倉和博，田浦俊春：“マルチロボットシステムのための頑健な強化学習法：環境変動の認識と適応に関する一考察”，第 17 回自律分散システム・シンポジウム講演論文集，pp.177-182，(2005)

7. 保田俊行，大倉和博，田浦俊春：“マルチロボットシステム環境における強化学習の頑健性向上のための一手法”，2005 年度精密工学会春季大会学術講演会講演論文集，pp.1103-1104，(2005)

8. 保田俊行，大倉和博，田浦俊春：“適応的な行動空間の分割を行う強化学習を用いたマルチロボットシステムの行動獲得”，ロボティクス・メカトロニクス講演会’05，CD-ROM 予稿集，1P1-S-066，(2005)

9. 保田俊行，大倉和博，田浦俊春：“適応的な行動空間の分割を行う強化学習を用いた実ロボットの行動獲得”，日本機械学会 2005 年度年次大会講演論文集，pp.251-252，(2005)

10. 平松久征，保田俊行，大倉和博：“オムニホイールを備えた自律ロボットの強化学習による行動獲得”，第 48 回自動制御連合講演会，pp.181-182，(2005)

11. 横田英二，保田俊行，大倉和博：“強化学習に基づく自律アーム型ロボット群の拡張性に関する一考察”，第 48 回自動制御連合講演会，pp183-186，(2005)

口頭発表 (参考)

1. 保田俊行，Svinin Mikhail，上田完次：“身体性を考慮した 4 足歩行ロボットの行動学習”，日本機械学会関西学生会卒業研究発表講演会前刷集，pp.117，(1999)

2. 保田俊行，Svinin Mikhail，上田完次：“階層型強化学習を用いた自律歩行ロボットの行動学習”，ロボティクス・メカトロニクス講演会 CD-ROM 予稿集，2P1-30-027，(2000)

3. 保田俊行，大倉和博，上田完次：“強化学習による棒押し問題へのアプローチ：実機実験と学習過程の解析”，2001 年度精密工学会秋季大会学術講演論文集，pp.24，(2001)

4. 保田俊行，川上賢一郎，大倉和博，上田 完次：“強化学習による協調搬送問題へのアプローチ：実機による検証”，第 14 回自律分散システム・シンポジウム資料，pp.205-210，(2002)

付 録C　動画資料

付録CDについて

本論文の各章に記載している計算機実験と実機実験の動画，および参考となる動画を収録している．

閲覧の際は，CD内のindexファイルを参照のこと．