博 士 論 文

自律移動型ロボットのナビゲーションに関する研究

指導教官　藤波 努 助教授

北陸先端科学技術大学院大学
知識科学研究科 知識社会システム学専攻

石川 浩一郎

2005 年 9 月 22 日

要 旨

強化学習は，ある環境に置かれたエージェントが，環境との相互作用を繰り返しながら，行動の結果環境から与えられる報酬をもとに自らの行動を改善する，試行錯誤的学習手法である．強化学習を用いることで，教師情報や事前知識なしの学習が実現可能になる．本研究では，ロボットに望ましい行動を獲得させる課題において，強化学習を効率的に進めるための手法について検討した．

通常の強化学習手法では，行動の結果 (報酬) をもとに，行動価値を推定した Q 値表を更新することで，行動方策の改善を図る．本論文では，複数の Q 値表を同時並行的に利用する，すなわち，強化学習エージェントを複数用いて学習させ，より望ましい行動を獲得した学習エージェントを，優先的に行動決定に利用するという，新しい手法を提案する．この手法により，(1) 各学習エージェントの学習内容の比較が可能となり，学習内容の優れた学習エージェントの特定ができる，(2) 学習内容の優れた学習エージェントを優先的に用いることで，学習を迅速に進める効果が得られる，と予想される．さらに，学習エージェント毎に用いるセンサを変えることで，(3) 冗長なセンサを特定できるという効果も得られると考えられる．

提案手法の評価のため，ロボットのシミュレータ上で実験を実施した．実験に当たっては，手法を実ロボットの学習に応用することを念頭におき，条件設定等に配慮した．また，評価する有効性を，(1) 重要度の高いセンサを特定し，学習を促進させる，(2) 置かれた環境下で，継続的に行動しながら，学習を促進させる，という 2 点とし，各々で適切と思われる強化学習手法と学習エージェントの選択処理を採用した．実験の結果，予想通り，学習を促進する効果が確認された．

提案手法は，ロボットの行動獲得以外にも，強化学習が適用可能な課題に広く用いることができる，汎用的手法である．また，複数の強化学習エージェントを同時に用いるというアイデアに基づく新しい手法であるため，従来提案されていた強化学習の拡張手法の多くとの併用も可能で，相乗効果が得られると予想される．さらに，提案手法の用途は，上記 2 つに限定される訳ではない．新たな用途の考案，手法の理論的側面の研究，及びより効果の高い強化学習エージェントの選択処理の探究を進めることで，有効性が一段と向上することが期待される．

目 次

図 目 次

表 目 次

第 1 章

緒　論

1.1　背景

近年，娯楽目的を中心に，家庭向ロボットの販売が開始され，ロボットが身近な存在となりつつある [17, 14]．しかし，実用に足るロボットの構築を考えた場合，不完全な情報や知識に基づいて行動を決定しなければならないことが，本質的な問題となる．すなわち，

(1) 置かれた世界に関する情報は膨大であり，その全てを把握ないし記述することは極めて困難である

(2) 一方，行動決定という観点からは，膨大な情報の一部のみが重要である

(3) ただし，何が重要な情報であるかを予め決定することは非常に難しい

(4) 日常生活環境は，一般に動的であり，変化に追随することが不可欠である

(5) 様々なノイズの影響を無視できない，

といった点に対処することが不可欠である．また，行動の決定に際しては，正確さは勿論，迅速さが要求されるという点も，こうしたロボットの実現を困難にする要因となっている．

こうした点を踏まえ，日常生活環境で人間と共存し，与えられたタスクを遂行する知的なロボットの構築を，本研究の究極の目標の 1 つとすることにした．過去の研究において，知的なロボットの構築に当たって，まず採用されたのが，人間の

行動決定を参考に，論理的な予測に基づいて行動を決定するというアプローチであった (このようなアプローチは，deliberative なアプローチと呼ばれている)．これに対して，deliberative なアプローチでは，実世界における動作で要求されるレベルの判断の迅速さが実現できないという批判がなされた [6]．こうした，迅速な判断を重視する立場からは，ある時点における環境の状況に対して，即応的 (リアクティブ) に行動を決定することを，時間軸方向に繰り返していくことで，非常に迅速に行動を決定可能であると共に，決定された行動を一連の流れとして見たときに，ある程度妥当性があることが実験により示された (こうしたアプローチは，reactive な，または behavior-based のアプローチと呼ばれている)．

ただし，ロボットが動作する環境や遂行すべきタスクが複雑になった際でも，reactive な手法のみで，十分な機能が実現可能であるとは考え難い．このため，実環境で動作するロボットの構築を目指す研究の多くは，現段階において，reactive な行動決定を基盤としながらも，deliberative な手法を併用して，より高度なタスク遂行能力を目指すというアプローチが一般的である．

とくに，置かれた状況の変化への対応，さらにロボットの構築段階における負荷軽減という観点からは，ロボット自らが学習し，パフォーマンスを向上するという機能をもつことが望ましいと考えられる．実際，過去の研究 [iv],[v] では，実験に先立って，望ましい行動の内容を実装者が記述するという手法を採用したが，この結果，実装作業中，テスト走行時の微調整に大きな作業が発生した上に，実装完了後，環境やタスクに変化が生じた場合に対応が難しいという問題が残った．ロボット自身に学習させるという設計方針は，こうした問題の解決法の 1 つとして，有望であると思われる．また，人間を含めた多くの生物が，学習によって行動を獲得しているという点を考慮すると，きわめて自然なアイデアである．

ロボットの行動内容を，適応的に改善しようとするアプローチは，過去に，2 つの大きな流れがあった．1 つの流れが，制御の領域における適応システムの手法 (例えば [34]) であり，もう 1 つがコンピュータにおける機械学習すなわち人工知能研究の流れ (例えば [43]) である．どちらのアプローチも，一定の成果をあげてきたものの，十分に知的なロボットの構築には，未だ至っていない．その理由の 1 つとして，次のような点が考えられる．

例えば，人工知能研究分野で研究されているニューラルネットワークは，入力データのノイズの影響を受け難く，十分な数の中間素子が与えられれば任意の連続関数を近似できるため，柔軟な行動の決定に利用可能である．しかし、こうしたアプローチは，学習のための正解情報が与えられる，いわゆる教師あり学習 (supervised learning) の手法であって，何が正解であるか明確でないまま自発的な試行錯誤によって学習を進めるという，生物における学習とは異なっている．教師あり学習では，学習すべき正解情報が与えられた際，現状との誤差を拠りどころとして学習を進める．一方，日常生活環境は，どういった行動が望ましいかを即時に判断するにはあまりに複雑過ぎ，さらに行動の評価自体も不完全にしか知覚されないのが一般的であるため，教師情報を定義できない，もしくは定義することが困難となることから，教師あり学習を適用し難い．

人工知能分野では，こうした学習課題は，教師なし (unsupervised learning) ないし，半教師つき学習として取扱われており、現在の中心的手法が，進化的手法と強化学習 (reinforcement learning) である．教師なし学習課題では，到達目標が明確に示されることはなく，環境とインタラクションすることにより，現状の評価値のみが示される．環境とのインタラクションを繰返す中で，現状の評価値を向上させるべく、自ら進んで試行錯誤を行うという能動性は，従来の教師あり学習手法には欠けていた性格であり，環境とインタラクションを行うエージェントという問題定義自体が，ロボットの行動学習と適合することから，ロボットに学習させるという研究において用いられる例が増えている．

進化的手法は，生物における進化を基本的なアイデアとした，非常に強力な手法で，ロボティクス分野においても，複雑な課題への適用例が見られる (例えば [26] 参照) ．その反面，行動を最適化していく過程を解析することは，一般には困難である．また，手法の生物学的な妥当性に関しては，比較的高等な生物においては，行動の獲得は発達段階における (すなわち，認知科学の領域でいう) 学習に拠っており，学習された内容は遺伝的には継承されない (獲得形質は遺伝するというラマルク説は否定されているという) 点で，必ずしも適切でないという主張もある．

一方，強化学習は，

(1) 数学的な解析が (少なくとも部分的には) 成功している (第 2.3 節参照)

(2) 最適化の過程の情報を利用して，効率的な学習が実現できる可能性がある

(3) 生物学的にも，裏付けが主張されている (例えば [18])

点が，進化的手法と異なっている

(強化学習の具体的手法に関しては，第 2 章にて詳述する)．とくに，ロボットに複雑な課題を遂行させることを考えた場合，事前知識を与えることが，学習時間の短縮や，達成可能な課題のレベル向上という面で有効であると考えられるが，進化的手法では，事前知識を遺伝的な形でコーディングし，初期値として採用することは難しいと考えられる．

強化学習は，最適化の手続きと併せて，最適化のための探索過程を提供する手法である．とくに，実際的な問題において，その複雑性のため，解析的な最適化手法が適用できない (ないし適用が現実的でない) 場合に，有望視されている手法の一つである．このため，ロボットの行動獲得という目的への応用が期待されている．例えば，Russell らは，人工知能分野における著名な教科書 [29] の中 (p.626) で，

> 制御戦略を人手で記述する手間を省く可能性があるという意味において，強化学習は，機械学習研究の中でもっとも活発に研究が進められている分野の一つである．ロボット分野の応用は特に価値があるものとなろう．

としている．

1.2 研究目的

以上の考察から，本研究では，強化学習手法を用いて望ましい行動を自ら学習する知的なロボットを構築することを対象とし，日常生活環境で利用可能なロボットの構築を最終目標として，より効率の良い学習手法を考察するものとする．このため，強化学手法習の適用に当たっては，従来の手法に拡張を加え，その効果を評価する．

具体的には，一般的に強化学習で用いられる，Q 値表という表を，複数用いて学習する手法を提案し，その効果を検証する．強化学習は，環境との相互作用を継続しながら，環境から与えられる報酬を最大化するための行動を，教師なしで自発

的に学習する点に特長がある．理論的には，マルコフ決定過程 (MDP) の逐時近似解法として捉えることが可能である (強化学習に関しては，第 2 章にて詳述する)．

現在，適用例の多い強化学習手法は，時間的差分 (TD) 学習と呼ばれる学習手法であり，とくにテーブル型 TD 学習が良く用いられている．テーブル型 TD 学習では，Q 値表と呼ばれるテーブルを基に行動を決定すると共に，行動の結果に基づいて Q 値表の修正を行う．この Q 値表を複数用いることは，複数の強化学習エージェントを，同時並行的に学習させ，それら複数の強化学習エージェントの学習内容を評価しながら行動を決定することに相当する．なお，複数の Q 値表を用いる強化学習の具体的な用途及び利点に関しては，第 3 章にて詳述する．

第 1.1 節で記述した，不完全な情報や知識に基づく行動決定という問題は，強化学習の領域では，部分観測状態における MDP (pertially observable MDP; POMDP) 課題として取り上げられ，とくに理論的な面を中心として，近年積極的に研究が進められている (例えば [19] 参照)．本論文は，部分観測の問題を正面から取扱うものではないが，ロボットの行動決定において本質的な問題であるという認識に基づいて，POMDP に関する先行研究の結果も踏まえて研究を進めるものとした．

また，提案手法の評価実験では，

(1) センサ能力 (探知可能領域) に限界がある

(2) センサ値に観測誤差が含まれる

(3) センサの死角部分が存在する

等の点で，部分観測性が含まれた実験設定となっており，POMDP 課題での応用にも役立つと思われる (実験設定に関しては第 4 及び 5 章にて詳述する)．

ここで，上記部分観測性は，とくに意識して実験設定に追加されたものではないことを指摘しておく．例えば，第 5 及び 6 章の実験では，ロボット研究で広く用いられている Khepera ロボット (図 1.1 参照) の物理的特性に基づいたシミュレータを利用した．また，ロボットに与えた課題も，障害物回避行動という，最も基本的な行動の獲得を意図したものである．行動に当たって，障害物を避けることは，自律移動型ロボットを構築するという点で第 1 条件であり，とくに人工生命系の研究では，学習によってロボットに望ましい行動を獲得させる際に，課題として採用されている例がある (例えば [27, 26] 参照)．また先行研究でも，新しい学習手

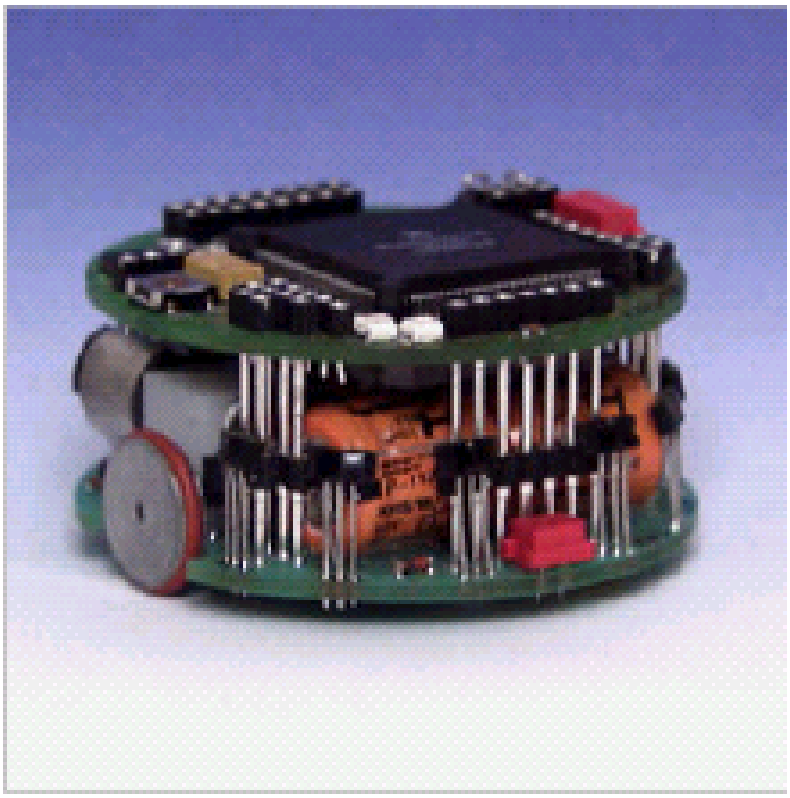

図 1.1: Khepera ロボットの概観．第 5 章で，提案手法の評価実験に用いた Webots は，Khepera ロボットのシミュレータである．なお，ロボットの仕様に関しては，第 5 章参照.

法提案の際，評価に用いられている (例えば，[30, 32] 参照)．しかし，実ロボットを意識した場合，このレベルの実験から既に，部分観測性の問題を考慮する必要が生じる．

1.3 本論文の構成

以下に，本論文の構成を述べる．

第 2 章では，本論文で用いる強化学習に関して，概念や手法の特徴を述べると共に，各強化学習手法に関して説明する．

第 3 章では，本論文で提案及び評価を行う，複数 Q 値表を用いる強化学習に関して，その仕組みと適用例を紹介する．

第 4 章では，第 3 章で述べた手法の評価のために行った，比較的単純な設定における確認実験に関して，その内容と結果を述べる．

第 5 章では，手法の有効性評価のために行った，より現実的な実験の設定に関して説明する．

第 6 章では，第 5 章の実験に関して，結果及び考察を記述する．

第 7 章では，本研究で提案した手法と，関連研究における手法との比較を行う．

第 8 章では，本研究全体に関する考察を記述すると共に，第 3 章で提案した手法の有効性に関して分析する．併せて，本論文では扱えなかった，将来の研究課題に関しても述べる．最後に本研究全体を概観すると共に，まとめを行う．

第 2 章

強化学習

2.1 概要

第 1 章で述べたように，強化学習 (reinforcement learning) 手法が，他の機械学習手法と大きく異なる点は，

(1) 学習に際して，正解が与えられない (教師なし学習)

(2) 学習する内容が，学習者 (の行動) に依存する (能動性)

にある．したがって，

(1) どういう行動が望ましいかを予め明確化する必要がなく，

(2) ロボット自身が，学習すべき内容を能動的に決定し，

学習を進めることが可能となることが，最大の特長である．

強化学習に関する研究の多くでは，環境内で観測・判断・行動するエージェントが，その行動の結果として受取る報酬の累積値を最大化するような行動方策を獲得する過程であるとして定式化する．そして，これをマルコフ決定過程 (Markov decision process; MDP)[1]の枠組みで定式化することが多い (例えば [20])．

強化学習のアルゴリズムは，この問題の解法，すなわち，周囲環境の計測値に基づいて次にとるべき行動を提案する行動決定の方法 (行動方策) と，その行動の

[1] 例えば [28] 参照.

結果得られる報酬の累積値が最大となるように行動方策の推定値を次第に変化させる方法 (学習) の 2 つの方法を同時に提供する．このためには，エージェントが計測する環境の観測値またはその履歴を状態とし，状態から行動への関数と，その関数を適用しつづけた場合に得られるであろう報酬の累積値を最大化するようにその関数を漸次変更する方法とを定めればよい (第 D 章も参照)．多くの強化学習アルゴリズムでは，環境の観測値を状態表現とし，ある状態から開始してある方策に従って行動したときに得られる報酬の累積値をその状態の価値 (状態価値関数) とし，またはある状態と行動の組から開始してある方策に従って行動したときに得られる報酬の累積値をその状態・行動対の価値 (行動価値関数) として，その最適値を求めることにより，最適な行動方策を得ている．環境の観測値と行動が有限種類の時には，状態価値関数や行動価値関数を表で表すことが多く，連続値である場合には離散化するか関数近似を用いる (関数近似に関しては，第 2.2.4 節にて詳述する)．

本論文では，強化学習という用語で，MDP の枠組み内での強化学習だけでなく，完全観測可能でない場合，すなわち部分観測可能な状態下での強化学習 (POMDP) をも含めるものとする．

以下，本章の構成を記述する．

第 2.2 節では，強化学習における基本的な概念や用語の説明を行う．本研究にも深く関連する内容として，行動選択手法 (第 2.2.1 節)，オプティミスティック初期値 (第 2.2.2 節) 及びエピソード (第 2.2.3 節) について紹介したのち，第 2.2.4 節では，強化学習における汎化の問題に関して検討する．

また，本論文では，従来の強化学習手法に対する拡張を提案するが，その際とくに，

(1) 次元の呪い

(2) 割引

の 2 つの特徴に着目した．第 2.2.5 及び 2.2.6 節で，これらについて述べる．

次に，第 2.3 節にて，本研究で用いる手法の説明を行う．まず，時間的差分 (TD) 学習とテーブル型学習について説明 (第 2.3.1 節) した後，第 4 及び 5 章の実験に用いる強化学習手法として，Q 学習 (第 2.3.2 節)，Sarsa 学習 (第 2.3.3 節)，R 学習

(第 2.3.4 節)，及び強化比較手法 (第 2.3.5 節) について，それぞれの手法の具体的な内容を紹介する．最後に，本研究で直接の対象とするものではないが，非定常環境における強化学習手法について，学習率 (ステップサイズパラメータ) との関連で簡単に触れる (第 2.3.6 節)．なお，本章の内容は，主に [35] に拠っている．

2.2 概念及び用語説明

2.2.1 行動選択手法

強化学習における行動選択の際に重要となるのは，単に現在の推定価値 (状態価値または状態行動価値) が最大となる行動を選択するのみではなく，より価値の高い行動を求める探索を行うことである (両者間のトレードオフを，exploration-exploitation 問題という)．探索を継続することは，局所最適解に陥らずに方策の正しい価値推定を行うため，また，とくに非定常問題において環境の変化に追随するために有効である．なお，R 学習 (第 2.3.4 節で詳述する) においては，行動選択手法により学習性能が異なるという報告 [22] もあり，とくに配慮が必要である．

探索と知識 (すなわち，現在までに学習した内容) 利用の両立という観点から，比較的良く用いられている行動選択手法として，ϵ-greedy と softmax がある [35]．以下に，代表的な行動選択手法を説明する．

ϵ-greedy 手法においては，推定される行動価値が最も高い (グリーディ) 行動を $1-\epsilon$ の確率で選択する (これが exploitation に相当する) か，小さい確率 ϵ で一様に任意の行動を選択する (これが exploration に相当する)．本手法は，semi-uniform 手法 [22] とも呼ばれる．

一方，softmax 手法においては，Gibbs 分布に基づいて行動が選択される．例えば，行動 a の優先度 $pref(a)$ が与えられた場合，行動 a を選択する確率 $\pi(a)$ は次式で与えられる．

$$\pi(a) = \frac{e^{pref(a)}}{\sum_{b=1}^{n} e^{pref(b)}} \tag{2.1}$$

本手法は，Bolzman Explorations 手法 [23] とも呼ばれる．なお，通常の softmax 手法の式には，温度 (T) と呼ばれるパラメータが含まれるが，本研究では，行動回数よって変化する温度のパラメータは用いないため，省略している．

以上の行動選択手法は，ある確率でランダムな行動を選択するのみで，学習の結果を探索に反映させることがないため，undirected な探索手法と呼ばれることがある [23] ．これに対して，学習結果をもとに，どこを集中的に探索すべきかを決定する手法を，directed な探索手法と呼ぶ．UE (uncertainty estimation) は，directied な探索手法の 1 つで，例えば行動 a の優先度 $pref(a)$ が与えられた場合，ある決まった確率 p で，以下の式を最大化する行動 a を選択する．

$$pref(a) + \frac{c}{N_f(s,a)} \tag{2.2}$$

一方，確率 $1-p$ で，ランダムな行動を選択する．ここで，c は定数であり，$N_f(s,a)$ は，状態 s で行動 a を選択した回数を示している．

2.2.2 オプティミスティック初期値

オプティミスティック初期値 (optimistic initial values) とは，探索を促進させる目的で，事前知識に基づいて，統計的に妥当と考えられる値より著しく大きい (オプティミスティックな) 初期値を設定する手法である．

例えば，行動価値の初期値をオプティミスティックに設定した場合，どの行動を選択したとしても，実際の行動結果 (報酬) が初期値に達しないため，次に同じ状況を経験した際，他の行動を選択する．この結果，行動価値推定が収束する前に，全ての行動が十分な回数試みられることになる．なお，初期値として与えられたオプティミスティックな値は，より正しい推定値によって置き換えられていくため，収束時までその影響が残ることは少ないと考えられる．

この手法は，とくに定常問題では効果があるとされている．一方，非定常問題では，行動の真の価値が，(例えば環境変化によって) 時間と共に変化する．このため，特別な初期状態用いる手法は，探索が一時的にしか促進されないことから，あまり効果がない．しかし，オプティミスティック初期値は非常に簡潔で，計算量に与える影響もないため，非定常問題においても，他の手法と組合せて使用されることもあり，実用上適切であることも多い．なお，オプティミスティック初期値は，V 値や Q 値の初期値として用いられることが多いが，これに限定されるものではない (第 2.3.5 節参照).

2.2.3 エピソード

強化学習課題においては，一連の行動の後，終端状態と呼ばれる特殊な状態で終わることが自然なものも多い．終端状態に達した場合，標準的な開始状態，若しくは標準的な分布に従って選ばれる開始状態に再設定された後，学習が再開される．例えば，本研究で扱うような，ロボットのナビゲーション課題では，壁に衝突した場合，スタート地点に戻して，新たに学習を開始するという条件に相当する (例えば [30])．こうした課題は，エピソード的課題と呼ばれる．

一方，終端状態をもたず，エージェントと環境との相互作用が限界なく (若しくは，十分長い時間) 続くことが自然な課題もある．こうした課題は，エピソードに分割されないことから，[35] では，連続タスク (continuing tasks) と呼ばれている．

なお，エピソードの終了を，報酬 0 で常に同じ状態に遷移する特殊な状態 (こうした状態は，マルコフ連鎖の吸収状態に相当する) ととらえることで，エピソード的タスクと連続タスクとを，数学的に同一の形で扱うことが可能である．

2.2.4 強化学習と汎化

強化学習が，通常の機械学習と大きく異なる点の 1 つとして，強化学習の仕組み自体には，汎化 (generalization) という機能は含まれていない点が挙げられる．この観点からは，強化学習は，学習ではなく，むしろ学習すべき内容の探索にその中心が置かれていると考えられる．

強化学習課題において，汎化能力が要求される場合には，通常の機械学習手法との組合せが行われる．とくに，状態空間を適切に構築し，Q 関数を効率的に表現する目的で，利用されることも多い (第 2.2.5 節参照)．こうした手法は，関数近似 (function approximation) 手法と呼ばれている．

これまでに，例えば，タイリングを用いる粗いコード化 (coarse coding) や，フィードフォワード型・RBF(radial basis function; 動径基底関数)・自己組織化マップ (self-organizing map; SOM) といったニューラルネットワークによる学習手法，データマイニング手法でも用いられる統計的な性質を用いた手法等に関する研究が報告されている (その一部は，第 7.2 節で議論する．また，第 6.2.2.6 節も参照)．こうした手法は，強化学習に汎化能力をもたせる試みととらえることも可能である．な

お，強化学習と組合せて用いられる関数近似手法は，一般的に，近似すべき対象を教師信号とした教師あり学習を行う．

理論的には，強化学習と関数近似手法を組み合わせた際の，強化学習の収束性証明が，近年積極的に研究されている．様々な強化学習手法と上述の関数近似手法とを組み合わせた場合の収束性証明のほとんどが，今後の研究成果を待つ状況である．

2.2.5 次元の呪い

実世界で動作するロボットに強化学習を適用する際の課題の一つに，環境を観測するためのセンサ数を増加させたいが，センサ数を増加させると状態数が増加し，学習時間が非常に長くなるという問題がある．いわゆる Bellman の次元の呪い (the curse of dimensionality) である [5, 8].

次元の呪いとは，状態変数の個数が増えると，状態数が指数関数的に増加し，この結果，必要となる計算量も指数関数的に増大する問題を意味している．実世界のロボットに搭載するセンサには精度・信頼性の問題があり，また，センサの測定範囲は狭いという問題点がある．これらを解決するために，できるだけ多数のセンサを利用するので，上記の問題が顕在化する．なお，表を用いる代わりに関数近似を用いる場合にも，基底関数の個数やパラメータ数に依存して近似精度が決まるため，それらの増加がさけられず，表の場合と同様に，上記の問題が発生する．

次元の呪いの問題を解消するためには，行動を決定するという観点から，必要最小限の状態空間に絞り込むことが有効である．なお，第 2.2.4 節で紹介した関数近似手法は，パラメータ数を少なく抑えることにより，こうした絞り込みの効果を実現することができる．本研究では，関数近似手法を用いた従来の手法とは異なるアイデアで，この問題に対処することを提案する (第 3.2 節参照)．なお，両者の比較に関しては，第 7.2 節にて詳述する．

2.2.6 割引

強化学習における最終目的は，累積報酬の最大化, すなわち時間ステップ t の後に受け取った報酬の系列を，$r_{t+1}, r_{t+2}, \ldots$ とした場合，

$$R_t = r_{t+1} + r_{t+2} + r_{t+3} + \cdots \tag{2.3}$$

で表される R_t の最大化である．とくに，上述のエピソード分割される課題においては，T を最終時間ステップとした場合，

$$R_t = r_{t+1} + r_{t+2} + r_{t+3} + \cdots + r_T \tag{2.4}$$

と表すことが可能である．

一方，無限に動作を継続する場合 (infinite horizon)，累積報酬も無限に大きくなるため，通常，将来の獲得報酬を割引して考える．例えば，現時点で適用例が多い強化学習手法である Q(ないし Sarsa) 学習 (第 2.3.2 及び 2.3.3 節を参照) では，割引を考慮した期待報酬を最大化する方策を，学習によって獲得させることを目的とする．すなわち，上と同じ条件で，将来にわたり受け取る減衰収益の合計，

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{2.5}$$

で表される R_t の最大化である．ここで，γ は，割引率 (discount rate) と呼ばれる定数で，$0 \leq \gamma < 1$ である．$\gamma < 1$ が成り立つことで，式 2.5 は，報酬の系列 r_k が上限をもつ限り，無限に加算を繰返しても有限の値をとる．

割引率は，将来の報酬が，現時点においてどれだけの価値があるかを決定するパラメータである．割引率 $\gamma = 0$ とした場合には，エージェントは即時報酬の最大化のみに注目する．一般には，γ を 1 に十分近い値を設定することで，単に即時報酬の最大化を行うのみではなく，将来にわたって獲得する報酬の最大化を図る．

しかし，式 2.5 における R_t の最大化を目指した場合，より良い方策であるが，時間的に後にしか大きな報酬が得られない方策より，時間的に近くに比較的大きな報酬が得られる方策が選好され，真に大きな報酬の得られる方策の学習が遅くなる可能性がある (本論文では，より少ない回数の環境との相互作用によって，望ましい行動を獲得することを，学習が速いと理解するものとする) だけでなく，(割

引なしの) 累積報酬という観点からは準最適な方策が最適解となってしまう場合がある．

なお，この事情は，有限時間でゴールに達する (finite horizon) 課題でも，ゴールが存在しない (すなわち infinite horizon に相当する) 課題でも，同様である [23]．また，有限 MDP では，割引率を 1 に十分に近づければ，こうした課題は解消するが，その反面学習速度は急速に低下する [12]．従って, 最適な割引率を予め決めることは困難である [31]．

以上のように，割引は，タスクがエピソード分割されるか否かに深く関連する．一般に，割引を行わない定式化はエピソード的タスクに向いており，割引を行う定式化は連続タスクに向いているとされる．しかし，同じタスクを，エピソード的にも連続タスク的にも定式化可能な場合も存在する．こうした場合の多くは，定式化の違いにより，最適化の目標となる期待収益の定義が異なる．このように，タスクのエピソード分割するか否かには，多くの考慮すべき要素がある．

2.3 一般的な強化学習手法

2.3.1 時間的差分学習及びテーブル型学習

第 2.2 節での議論からも明らかなように，多くの強化学習手法は，離散化された状態空間と時間の上に組み立てられている．

本論文では，第 4 及び 5 章で述べる実験で，いくつかの強化学習手法を用いるが，そのうち Q 学習 (第 2.3.2 節), Sarsa 学習 (第 2.3.3 節), 及び R 学習 (第 2.3.4 節) に関しては，継続する状態間の効用の差分を利用することから，時間的差分 (TD) 学習と呼ばれる [29] 強化学習手法に分類される．

TD 学習のうち，ある時点と次の時点との効用の差のみ (すなわち効用の差を 1 つだけ) に注目する学習手法は，1 ステップ TD 法 (one-step TD method) と呼ばれる．これに対して，各時間の間の効用の差を複数同時に取扱う手法も考えられる．こうした手法は，n ステップ TD 法と呼ばれる (例えば，第 C 章にて詳述する適格度トレースは，n ステップ TD 法実現のための，具体的実装法である)．本研究では，主に 1 ステップ Q/Sarsa/R 学習による実験を行ったため，以下の説明は，1 ス

テップ法の場合に関して記述する．

これらの手法では，効用として行動価値 (ある状態である行動をとる価値で，一般に Q 値と呼ばれる) を利用する [35]．状態及び行動が離散化されている場合，行動価値の関数 (Q 関数) は表の形で表すことができ [41]，この表 (Q 値表と呼ばれる) を用いるテーブル型 TD 学習による研究例が多く報告されている．なお，第 2.3.4 節にて説明する R 学習では，R 値という表現を用いることがあるが，本論文では，状態行動価値を，R 学習であっても，Q 値と呼ぶことにする．テーブル型 TD 学習では，Q 値表と呼ばれるテーブルを基に行動を決定すると共に，行動の結果に基づいて Q 値表の修正を行う点に特徴がある．

なお，観測された状態及び，実際にとった行動に関する推定価値のみを更新する手法は，asynchronous な手法 [12] と呼ばれることがある．一方，synchronous な手法では，各時点の状態 – 行動対以外の Q 値に関しても，更新を行う．

次節以下では，本論文で用いる強化学習手法の詳細説明を行う．

2.3.2 Q 学習

Q 学習は，方策オフ (off-policy) 型の TD 学習手法であり，ある方策 (挙動方策と呼ばれる) に基づいて行動しながら，最適方策を学習する点に特徴がある [35]．例えば，行動選択手法として，ϵ-greedy 手法を用いた場合，ϵ-greedy 手法に基づく行動決定を行いながら，実際には最適方策を学習する．

1 ステップテーブル型 Q 学習における，Q 値の推定の改善は，次式によって行われる．

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)] \tag{2.6}$$

ここで，s は現在の状態，a は採用した行動，r は行動によって得られた報酬を示し，s' は行動後の新しい状態，a' は新しい状態において選択される行動である．また，$Q(s,a)$ は，状態 s における行動 a の行動価値推定を示し，α $(0 < \alpha < 1)$ は学習率，γ $(0 \leq \gamma < 1)$ は割引率 (第 2.2.6 節参照) と呼ばれ，一般的には定数を用いる (ただし，数学的には，収束させるため，ある速度で 0 に漸減させる必要がある．第 2.3.6 節も参照)．式 2.6 において，[ ] 内は，1 回の Q 値更新における，更新

量を決めるもので，δ 項 [31] と呼ばれる．なお，以上の記法は，次節以降も同様に用いる．

Q 学習は，挙動方策と推定方策とを分離したことで，比較的早い時期に学習の収束性の証明が行われた [42]．ただし本研究における実験では，①センサの到達範囲に制限がある，②センサ値にノイズが含まれる，等の点で部分観測可能状態となっている (第 1.2 節参照) こと，及び α に定数を用いていることもあり，収束することはない．

2.3.3 Sarsa 学習

Sarsa 学習は，方策オン (on-policy) 型の TD 学習手法であり，挙動方策に従って行動しながら，その挙動方策自体に基づいた Q 値の改善を行う点が Q 学習と異なる．例えば，行動選択手法として，ϵ-greedy 手法を用いた場合，ϵ-greedy 手法に基づく行動決定を行いながら，その行動決定手法に基づく方策の改善を行う．

1 ステップテーブル型 Sarsa 学習における，Q 値の改善は，次式によって行われる．

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma\, Q(s',a') - Q(s,a)] \tag{2.7}$$

2.3.4 R 学習

R 学習は，Schwartz が提案した [31] ，割引が行われない強化学習課題を扱うための手法で，単位時間ステップ当たりの平均報酬の最大化を目標とする点が，Q 及び Sarsa 学習と異なる特徴となっている [35].

方策オフ型の 1 ステップテーブル型 R 学習における，状態行動対の価値の推定の改善は，次式によって行われる．

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r - \rho + \max_{a'} Q(s',a') - Q(s,a)] \tag{2.8}$$

また，実際とった行動が，探索行動でなかった際のみ ρ の更新が行われ，更新式は次式の通りである．

$$\rho \leftarrow \rho + \beta[r - \rho + \max_{a'} Q(s',a') - Q(s,a)] \tag{2.9}$$

ここで，α 及び β $(0 < \alpha\,, \beta < 1)$ は学習率と呼ばれ，通常定数を用いる．ρ は学習が収束した際，平均報酬に収束する．式 2.8 において，経験した s 及び a に関する Q 値のみが更新の対象になっているため，本手法は asynchronous な手法 (第 2.3.1 節参照) である．また，式 2.8 の [ ] 内は，1 回の Q 値更新における更新量を決めるもので，Q 学習における δ 項に対して，σ 項と呼ばれることがある ([31] 参照)．

なお，R 学習は，現時点で，学習の収束性が理論的に証明されていない．また，R 学習の利点及び問題点に関しては，第 3.3 節で詳細な検討を行う．

2.3.5 強化比較手法

強化比較手法 (Reinforcement Comparison) は，状態遷移のない，比較的単純な強化学習課題に用いられる手法である．強化比較手法では，与えられた報酬の大小を評価するための基準レベルをリファレンス報酬 (reference reward) と呼び，即時報酬の指数減衰加重平均値を用いる．

この基準レベルより大きい報酬が得られた行動は，良い行動として，以後この行動をとる確率が上がる．一方，基準レベルを下回った報酬につながった行動に関しては，以後この行動をとる確率を下げることにより，次第に報酬の大きな行動が選択される傾向が強まる．

実際の行動の選択に当たっては，通常 softmax 手法 (第 2.2.1 節参照) が用いられる．行動 a を選択する優先度 $pref(a)$ 及びリファレンス報酬 $\bar{r}$ は，具体的には次式によって更新される．

$$\begin{aligned} pref(a) &\leftarrow pref(a) + \alpha\,(r - \bar{r}) \\ \bar{r} &\leftarrow \bar{r} + \kappa\,(r - \bar{r}) \end{aligned} \tag{2.10}$$

ここで，κ $(0 < \kappa < 1)$ はリファレンス報酬の学習率を示している．なお，強化比較手法を適用する際，リファレンス報酬の初期値として，オプティミスティック初期値 (第 2.2.2 節参照) を採用することも多い．

2.3.6 非定常問題への追随

前節までの強化学習の処理の説明において，通常，定数値の学習率 (ステップサイズパラメータ) が用いられることを述べた．定数値のステップサイズパラメータを用いた場合，数学的な面からは，処理の収束性が保証されない．一方，非定常問題において，変化に追随し，最適な方策の探索を続けるという効果が得られる．また，第 2.2.1 節で述べた行動選択に関しても，探索を継続することで，課題の非定常性への追随を狙うことも多い．

第 3 章

提案手法

3.1　手法の概要

3.1.1　複数の状態行動価値表

テーブル型 TD 強化学習では，1 つの Q 値表をもち，これを基に行動を決定すると共に，行動後に得られる報酬によってこの表を更新する．本論文で提案する手法は，複数の Q 値表，すなわち複数の強化学習エージェントを用いて行動決定と学習を行う点に特徴がある．

機械学習の領域では，一般に，学習の速度や学習の結果獲得される内容が，初期値等の学習条件や学習過程の影響を受けることが知られている．このため，条件が異なる強化学習エージェントが同時に複数存在した場合，学習速度や学習内容が各々異なると予想される．とくに，本研究で用いた強化学習手法は，前章で述べたように，能動的な学習手法である．ロボットは，自らの決定に基づいて行動し，そこで経験される内容に沿って学習する．すなわち，学習すべき内容を，自らの行動によって選択する．

したがって，複数の強化学習エージェントを同時並行的に用いて，それらの学習速度や学習内容を比較しながら，学習速度が早く，かつ学習内容が優れたものを優先的に利用して行動決定を行うことで，学習に要する試行数を削減し，優れた内容の学習を実現できる可能性がある．本研究では，以上のようなアイデアに基づき，複数の Q 値表を同時並行的に用いることで，強化学習のパフォーマンス

の改善を図るものとした.

3.1.2 行動決定と学習

ここで問題となるのは,

(1) 条件の異なる強化学習エージェントをどのように準備するか

(2) 各強化学習エージェントをどのように比較し，評価するか

(3) 各強化学習エージェントの評価をどのように行動に反映するか

という点である．以下，これらの点に関して，本研究での対処法を記述する．

第 1 の問題である，異なる複数の強化学習エージェントをどのように準備するかに関しては，本論文では，各強化学習エージェント毎に，学習及び行動決定に用いるセンサ (の組合せ) をそれぞれ異なるものとした (以下，センサの組合せの各々をセンサ集合と呼ぶものとする) ．すなわち，複数のセンサ集合を対象とし，利用するセンサ集合毎に 1 つの Q 値表をもつ．例えば，第 5 及び 6 章の実験で用いたシミュレーションロボットは，8 つのセンサを持つ (第 5.2 節参照)．8 つのセンサのどれを利用するかの組合せは，$2^8 - 1 = 255$ 通り存在する (センサを全く利用しないという組合せは考慮しない) ため，255 の Q 値表を基に行動の決定と報酬による更新を行う．利用するセンサ集合が異なれば，同一の環境に置かれても，各強化学習エージェントは，それぞれ内容の異なった状態の同定を行うことになる．

次に，第 2 及び第 3 の問題の対処法について述べる．実際の行動の決定に当たって，これら複数の強化学習エージェントのいずれを用いるかに関しては，各々の Q 値表に対応して，その Q 値表が選択される優先度を司る変数をもつものとする．ここで，ロボットは，どの Q 値表を用いれば望ましい行動が実現できるかを判断し，自らこの優先度を更新していく．この優先度を基に，softmax 計算 (第 2.2.1 節参照) を行って，各々の Q 値表の選択確率 (π) を求め，その選択確率に基づいて，Q 値表のうちの 1 つを選択する (以上の処理が評価に相当する).

ロボットが実際にとる行動は，以上のようにして選択された Q 値表に基づき，ϵ-greedy 手法 (第 2.2.1 節参照) で決定する．Q 値の更新に関しては，観測された状態と実際にとった行動を基に，各々の Q 値表を更新する．この更新処理自体は通

常の強化学習における更新と同一とする (第 2.3.2，2.3.3，及び 2.3.4 節参照)．なお，R 学習を用いる際は，Q 値の更新と併せて，ρ 値の更新に関しても，各々の Q 値表において，通常の R 学習の更新式 (式 2.9) にしたがって実施される．

一方，各 Q 値表の優先度の更新処理に関しては，いくつかの方法が考えられる．本研究では，2 つの異なる処理を考案した．具体的には，第 3.2 及び 3.3 節にて詳述する．

入力情報 (すなわち，置かれた環境の状態同定の内容) がそれぞれ多少異なる，複数の強化学習エージェントを，同時に並列的に準備することの用途及び期待されるメリットとして，本研究では，次の 2 点を考えた．

(1) 各強化学習エージェントの学習内容の比較により，最適なセンサ集合が特定できる

(2) R 学習において，学習速度低下の原因となる局所解に陥った際，迅速な脱出を実現する

各々の内容に関して，以下の節で詳述する．

3.2 最適センサ集合の特定

3.2.1 期待効果

強化学習課題において，エージェントの行動は，置かれた状態に基づいて決定される．ここで，状態はセンサ値の組合せで定まるものと考えて良い．この際，状態空間は，行動決定に必要最小限な範囲で構成されることが望ましい [18]．一般に，センサ数を増加させれば，状態記述が正確になり，より適切な行動が選択できると考えられる．しかし，現実のロボットでは，必要のないセンサは，状態空間を無用に大きくすることで強化学習の進行を遅らせるばかりでなく，行動決定に本来不要な情報が雑音となって，学習を阻害する可能性がある．一方，学習開始以前に，行動決定に必要なセンサを特定することは，通常困難である．

そこで，本節では，センサの組合せ (以下，センサ集合と呼ぶ) に対応する状態空間を複数有し，これらの強化学習を並行して行うとともに，最適なセンサ集合 (本論文では，与えられた課題を達成する上で，最低限必要なセンサの集合を，最

適なセンサ集合と呼ぶものとする) を，オンラインで (すなわち，学習させながら) 選択する方法を提案する．すなわち，利用するセンサ集合が異なる，複数の Q 値表を準備する．これら複数の Q 値表の 1 つを選んで (その Q 値表の決定に基づいて) 行動し，その結果を蓄積していくことで，各 Q 値表が学習 (及び行動決定) に用いているセンサ集合の優劣を判断することが可能になる．この結果，行動決定に重要なセンサが特定される．また，この判断を学習にフィードバックさせることで，適切なセンサ集合を用いた学習と行動決定が可能である．この手法を適用することで得られるメリットは，以下のようにまとめられる．

(1) 適切な行動の実現及び学習の高速化

- 状態空間の合理的な構築
- ノイズの影響の軽減

(2) センシングコストが発生する課題でのコスト軽減

(3) 適切なセンサのみ搭載したロボットの構築 (経済的有利性)

複数のセンサ集合を比較する最も単純な方法は，それぞれのセンサ集合を用いる強化学習を，それぞれ別個に実施することである．しかし，そうした方法では，実験時間が長くなる (実験に要する行動の合計回数が増加する) という欠点がある例えば，n 個のセンサ集合の比較のためには，用いるセンサ集合を変えて，n 回の実験を繰り返すことが必要となり，単純計算では n 倍の行動回数を要する．また，この方法を採用するためには，前提条件として，センサ集合の適切さを判断するためにどの程度の行動回数の強化学習が必要であるかが，予め把握されている必要がある．さらにこの場合，センサ集合の優劣の判断は，複数の実験の結果が全て得られるまで待つ必要がある．なお，この方法は，オンライン手法ではない，すなわち，行動しながら学習を進め，自己の機能を高めるという，ロボットにおける学習の本質に則していないという欠陥をも含んでいる．

本節で提案する方法は，1 つのロボットにおいて，複数の強化学習 (各強化学習は，例えば，1 組のセンサに対応する Q 値表・割引率・学習係数からなる) を同時に動作させる (すなわち，複数の強化学習エージェントを，同時に学習に参加させる) と共に，どの強化学習エージェントを行動決定に用いるかに関して，(別の) 強

化学習によって決定するという方法である．この結果，ロボットの 1 行動当たりの計算時間は増大するものの，実験に要する行動の総回数という点では，少ない行動回数で望ましい行動が獲得可能なセンサ集合において，強化学習に要する行動回数程度で終了することが期待できる．

こうした手法を採用することで，センサ集合の選択を自動的に行うことが可能であることを第 4 及び 6 章で示した．本手法の適用により，センサ集合を絞込みができれば，状態空間をより適切に構築し，次元の呪い (第 2.2.5 節参照) を回避する効果が得られる．

なお，提案手法は，最適なセンサ集合の選択という用途に限定される訳ではなく，複数の強化学習を比較しながら学習する一般的な枠組みであり，それ以外の用途に用いることも可能である．

以上，本節の内容をまとめると，提案手法を適用することで，

(1) センサ数やセンサの組合せを変化させながら実験を繰り返す必要がなく，オンラインで (すなわち，学習を進めながら)，適切なセンサの選択が可能になる

(2) したがって変化する環境にも (おそらく) 適応可能である

(3) 適切なセンサを利用することで，より望ましい行動が，より迅速に学習される

という点が，本節で提案した手法の最大のメリットである．

3.2.2 処理

各々が Q 学習を行う複数の Q 値表を用いて，最適センサ集合のオンライン特定を行う際の具体的処理を，図 3.1 に示す．利用するセンサ集合の異なる複数の Q 値表を用意する．各 Q 値表で利用するセンサ集合 m を要素とする集合 M を考える．M は，例えばセンサが k 個で事前知識を用いない場合，センサを 1 つ以上利用するセンサ集合の全て ($2^k - 1$ 通り) となる．なお，予め適切なセンサ集合が推測可能な場合には、それらのみを用いれば良い．$pref(m)$ は Q 値表 Q_m の優先度を表す (行 2)．この優先度に，softmax 手法 [35] を適用し行動を決定する Q 値表を選択し (行 8–11)，ϵ-greedy 手法 [35] でロボットが実際にとる行動を決定する (行 13–20)．

行動後，通常の Q 学習と同一の更新式を用いて，各 Q 値表を更新する (行 23–25)．さらに，実際に行動決定に用いられた Q 値表が，グリーディに行動を決定した場合のみ，この Q 値表の優先度を更新する (行 26–28)．

複数の Q 値表から，ロボットの実際の行動を決定するものを選択する処理 (行 8–10, 26–31) に関しては，この問題を n 本腕バンディット問題 (n-armed bandit problem)[35] と見做し，強化学習で学習させている．なお，この強化学習には，強化比較手法 (第 2.3.5 節参照) を用いた．

n 本腕バンディット問題は，異なる確率分布に従って報酬を返す複数の腕のうち，期待報酬最大のものの特定を課題とする．通常，各腕の統計的性質 (報酬の多寡，報酬が得られる頻度) は定常であると仮定されている．そして，得られた報酬の大小を評価するための基準レベルをリファレンス報酬と呼び，獲得報酬の指数減衰加重平均値を用いる．

しかし，本論文では，学習中の Q 値表を腕とみなすことから，その性質は定常ではない．そのため，通常のリファレンス報酬を用いることの妥当性に疑問がある．実際，第 6.1 節の実験の予備実験において，優先度の比較的高い Q 値表が選択され壁に衝突した場合，壁にトラップされる現象が観測された (なお，本論文では，壁に接触した状況が長時間継続することを，トラップされたと表現するものとする)．そこで，図 3.1 の処理では，指数減衰加重平均 ($\bar{r}'$) と，実験開始時からの獲得報酬の平均 ($\bar{r}''$) のうち，値の大きいものをリファレンス報酬 ($\bar{r}$) とした (行 29–31)．

通常のレファレンス報酬を用いた場合，不適切な Q 値表の優先度がたまたま高くなった際に，この Q 値表によって例えば最低報酬の行動が継続して選択されると，リファレンス報酬が最低報酬値に急速に漸近するため，優先度の更新量も 0 に近づく結果，本来低下すべき当該 Q 値表の優先度が十分に低下しないことがある．すなわち，この不適切な Q 値表が選択され続けることになる．

ここで，開始時からの平均獲得報酬 ($\bar{r}''$) は，報酬の変化に穏やかに追従するため，これと指数減衰加重平均値 ($\bar{r}'$) との最大値をリファレンス報酬 ($\bar{r}$) とすれば，期待報酬が急速に高くなるときにはその Q 値表を用い，期待報酬が急速に低下するときには当該 Q 値表の優先度を低下させ続けることができることになる．この結果，非定常性が原因で通常の強化比較手法では学習が進まない状況が生じた場

合でも，探索と学習を継続できると考える．

3.3 R 学習における局所解の回避

3.3.1 期待効果

第 2.2.6 節では，割引を用いる手法に潜在する問題に関して記述した．これらの問題を解決するため，割引しない累積報酬を最大化する手法の研究も進められている．

R 学習は，Schwartz が提案した [31]，平均報酬の最大化を目指す学習手法，すなわち，R_t を

$$R_t = \frac{1}{k+1}\left(r_{t+1} + r_{t+2} + r_{t+3} + \cdots + r_{t+k+1}\right) \tag{3.1}$$

として，行動回数 k を無限にした場合の R_t の極限

$$\lim_{k\to\infty} \frac{1}{k+1} \sum_{k=0}^{\infty} r_{t+k+1} \tag{3.2}$$

の最大化を目指す学習手法[1]であり，Q 学習のように model-free かつ asynchronous (第 2.3.4 節参照) な更新を行うことを特徴とし，一般にエピソード分割されない (すなわち，無限に動作を継続する) 課題に適用される [35].

Schwartz は，強化学習で割引を用いることの理由をいくつか想定して，批判を加えている．例えば，

(1) 金利との類推については，強化学習の研究者が，報酬の現在価値や利子の累積に興味があるとは考えられない

(2) エージェントの寿命や環境の変化に対応するためだとする理由については，実際の強化学習の研究分野で，実際に寿命や環境変化を対象とすることはない

などとしている．この点，平均報酬は，割引された期待報酬と比較して，より自然なパフォーマンスの評価基準であると指摘する．さらに，R 学習は Q 学習を包

[1] 同様にして，平均獲得報酬最大化を図る DP アルゴリズム等も考え得る．詳細は [23, 28] 参照.

含するもので，Q 学習における割引率に対する敏感性 (第 2.2.6 節) や，状態間の報酬伝播の遅さを解消可能であると主張している [31]．したがって，Q 学習の代わりに，R 学習を適用することは，迅速な学習や結果のロバスト性の面で有利であると考えられる．

一方，R 学習の適用に際しては，探索方法を適宜選ばないと，後述する局所解状況に容易に陥り，学習が十分進まなくなることがあるという欠点も指摘されている．Mahadevan は，ロボットの箱押し課題を取り上げ，R 学習の結果が Q 学習に劣り，とくに行動選択手法に softmax 手法 (第 2.2.1 節参照) を用いた場合に性能の劣化が著しいことを報告している [22]．

しかし，継続的に行動しながら，望ましい行動を強化学習で獲得していくロボットを考えた場合，infinite horizon 課題を対象とする平均報酬学習を適用することは，ごく自然である (実ロボットを実験に用いた場合，エピソード分割された課題が現実的でない点に関しては，第 5.3 節で述べる) ．このため，R 学習の欠点を解消し，Q 学習以上の学習速度を常時実現可能な探索方法を確立することは，大きな意義をもつ．

本節では，それぞれ別個の Q 値表をもつ複数個の学習エージェントを用いる，新たな探索方法を提案する．本探索方法は，複数のセンサをもつ現実のロボットを想定し，使用するセンサを限定した仮想の強化学習エージェントを複数同時に用いて，学習と行動決定を行う方法である．具体的には，一部のセンサのみを用いる R 学習 (各学習では，ϵ-greedy 探索を行う) を複数個用意し，すなわち複数の異なるセンサの組合せ 1 つに対して 1 つの R 学習器を割り当てて，同時並行的に学習させる．複数の学習エージェントを用いる目的は，例えば，壁にトラップされた状態に入っても (すなわち局所解状況に陥っても)，多数の強化学習エージェントの中には，トラップから脱出可能な行動を選択するものがあると予想され，そうした行動を実際に実行すると共に，他の強化学習エージェントにもこの行動を学習させることである．

[22] では，学習の学習速度の低下をもたらす原因の 1 つと考えられている limit cycle 状況が，交互に訪問される 2 つの状態の状態価値が変化しなくなることにより発生すると説明されている．実際，我々の実験で発生した，ロボットが壁に長時間トラップされた状態も，以下で説明するように局所解状況であると考えられ

る．したがって，局所解状況が回避できれば，R 学習の良い性質が実現し，良好な学習速度が得られることが期待される

limit cycle 状況を回避するには，探索行動を採用する確率を高くすればよいことが確かめられ，その結果 Q 学習より良い成績 (累積報酬) が得られることが知られている [23]．しかし，Mahadevan が実験に使用した探索方法は，ϵ-greedy または UE (第 2.2.1 節参照) である． [23] では，ϵ- greedy 探索で成功したと報告されているが，我々の実験では，第 6.2.2.2 節に述べるように，これでは探索が弱すぎ，壁にトラップされた状態から脱出できなかった．一方，UE は，利用頻度の少ない行動を選んで探索する人為的な探索手法であり，式 2.2 のパラメータ c の値によって敏感に動作を変えると考えられる．そこで我々は，より自然かつ有効な探索方法として，上述の手法を考案した．

次に，局所解状況について詳述する．Mahadevan が例示した limit cycle 状況 [22] は，①即時報酬が 0 (したがって平均報酬も 0) である行動によって構成されている，②状態数が 2 と仮定されている．しかし，即時報酬が 0 でない場合も，同じ現象が起きると指摘している [23]．また，以下のように，複数状態にわたる局所解状況も考え得る．

R 学習における推定行動価値の更新式 (式 2.8 及び 2.9 参照) において，仮に，あるループに入り，かつその間 $r-\rho$ がほとんど 0 であるとする．このとき，$Q(s,a)$ はある一定の値に収束する (ループに入っているという仮定から，$s \rightarrow a$ はこのループ内で一意に決まっている)．その値は，ループ内の複数の $Q(s,a)$ の初期値によって決まり (それらの平均値と予想される)，本来 R 学習が想定している σ ではない (R 学習が想定する σ は，$s \rightarrow a$ はこのループ内で一意であるため，$\sigma = r - \rho$, したがって上の仮定より $\sigma \approx 0$ である)．さらに，上記の条件よりもっと緩い条件でも，同様のことは起こり得る．例えば，ある状態集合のなかを遷移しているが，各状態について，$r-\rho$ の時間平均値が 0 であるといった場合である．

このような事態に陥る場合の一例は，壁にトラップされ，そこから脱出するには数行動を要し，トラップ状態が継続する間，報酬は行動にかかわらず同一である場合である．この場合，r は状態・行動にかかわらず同一であるため，暫く後には，ρ がほぼ r と等しくなる．より正確には，行動前後の状態の状態価値の差が，ρ に影響を与える (これは $r-\rho$ と相互依存しているため，厳密には評価が必要で

はある) ものの，前々段落の説明の通り 0 に近づくためである．こうした状況下では，仮に r が大きな負の報酬であったとしても，それを回避する傾向は，行動数が増すにつれて減少してしまう．このため，ロボットは limit cycle 状況から離脱できなくなってしまうと考えられる．

limit cycle 状況を回避する方法の 1 つは，平均報酬値の継続的な低下を認識し，探索行動を促進することである．ただし，平均報酬値が，局所解状況が原因で低下しているのか，あるいは環境条件を正当に反映して低下しているのかを判断することは難しい．このため，条件が多少異なる複数の Q 値表を並置することで，両者の区別を図る．すなわち，並置された全ての Q 値表において，同様に平均報酬値が低下した場合は，環境条件を反映したものと判断する．一方，特定の Q 値表において，平均報酬の低下が少ないことは，その Q 値表において別の行動がグリーディであること，すなわち，これまでとは別の行動 (列) をとることで，limit cycle 状況のループを脱し，平均報酬が上昇する可能性があることを示唆している．利用するセンサが異なれば，同一の「状態 – 行動 – 報酬」に基づく学習を行っていても，選択する行動に差異が発生すると予想され，このばらつきを積極的に利用することを考えた．

例えば壁にトラップされた際は，トラップ状態の継続につながる行動がグリーディである Q 値表の ρ 値は，即時報酬が負であれば，行動と共に低下していく．したがって，こうした Q 値表に対応する優先度も，徐々に下がるため，別の行動がグリーディになっている Q 値表の選択確率が，相対的に増加する．この結果，探索的な行動が選択され，トラップから離脱する可能性が高くなる．

以上，本節の内容をまとめると，提案手法を適用することで，継続的に行動しながら強化学習で望ましい行動を獲得していくロボットに適した学習手法の問題点を解消し，より望ましい行動が，より迅速に学習されるという点が，本節の手法の最大のメリットである．

3.3.2 処理

off-policy 型の R 学習を進める複数の Q 値表を用いて，R 学習の高速化を図る際の具体的処理を，図 3.2 に示す．利用するセンサ集合が異なる複数の Q 値表を用

意する．各 Q 値表で利用するセンサ集合を m と考える．Q 値表 Q_m の優先度を $pref(m)$ とする (行 2)．この優先度に，softmax 手法 [35] を適用して行動を決定する Q 値表を選択し (行 8–11)，ϵ–greedy 手法 [35] でロボットが実際にとる行動を決定する (行 13–20)．行動後，各 Q 値表を更新するが，更新式は通常の off-policy 型 R 学習と同一である (行 23–26,28)．さらに，ロボットが実際にとった行動が，各 Q 値表で Q 値最大であった場合のみ，この Q 値表の優先度を更新する (行 23,25,27,28)．

行 27 の式では，ロボットの実際の行動と同じ行動がグリーディである Q 値表の優先度について，その Q 値表の平均報酬の近似である ρ が累積されていく．この処理によって，ρ の値の大きい，すなわち期待報酬の大きい Q 値表はより選択されやすくなる．一方，グリーディに選択した行動の報酬が良くなかった Q 値表や，これまでの行動の結果が良くない Q 値表の優先度には，負ないし小さい正の値が加算されるため，徐々に選択される確率が減っていく．結果として，適切な行動を決定可能な (すなわち適切な行動が，その Q 値表上でグリーディとなっていた) 回数の多い Q 値表が，実際の行動を決定することになると考えられる．

3.4 第 3 章のまとめ

以上のように，本研究で提案する新しい強化学習手法の特長は，複数の Q 値表が同時並行的に強化学習に用いられる，すなわち，強化学習エージェントを複数用いてそれらの学習内容の妥当性を考慮しながら行動の決定を行うという点にある．本手法では，複数の Q 値表間で競合が起こる．この競合の中で，より望ましい行動決定を行う Q 値表を優先的に利用することで，行動の適切さと，学習の迅速性を向上させるという点が，提案手法の最大の目的である．また，行動決定に，複数の Q 値表を同時並行的に用いる際の，具体的な用途として，

(1) 冗長なセンサの特定

(2) R 学習における局所解の回避

の 2 例をあげた．なお，これらの例は，あくまで活用例であって，複数 Q 値表を用いる手法が，これらの用途に限られる訳ではないことを再度指摘しておく．

1: $N \leftarrow$ number of actions avairable to the robot
2: $pref(m) \leftarrow 0,\ \forall m \in M$
3: $Q_m(s,a) \leftarrow$ initial value, $\forall\, m \in M,\ s,\ a$
4: $t,\ total,\ \bar{r},\ \bar{r}' \leftarrow 0$
5: $s \leftarrow$ initial state
6: **while** $t <$ planned transision times **do**
7: $t \leftarrow t+1$
8: **for** $m \in M$
9: $\pi(m) \leftarrow \dfrac{e^{pref(m)}}{\sum_{m' \in M} e^{pref(m')}}$
10: **end for**
11: according to $\pi(\cdot)$, select Q-table $Q_{\hat{m}}$
which will decide the action
12: $n \leftarrow$ number of $a,\ a = \mathrm{argmax}_a\, Q_{\hat{m}}(s,a)$
13: **for** $a \in A$
14: **if** $a = \mathrm{argmax}_a\, Q_{\hat{m}}(s,a)$ **then**
15: $\pi_{\hat{m}}(a) \leftarrow \frac{1-\epsilon}{n} + \frac{\epsilon}{N}$
16: **else**
17: $\pi_{\hat{m}}(a) \leftarrow \frac{\epsilon}{N}$
18: **end if**
19: **end for**
20: choose action $\hat{a}$ according to $\pi_{\hat{m}}(\cdot)$, and execute $\hat{a}$
21: get reward r, and observe state s'
22: $total \leftarrow total + r$
23: **for** $m \in M$
24: $Q_m(s,\hat{a}) \leftarrow Q_m(s,\hat{a}) + \alpha\,[r + \gamma\ \max_{a'} Q_m(s',a') - Q_m(s,\hat{a})]$
25: **end for**
26: **if** $Q_{\hat{m}}(s,\hat{a}) = \max_a Q_{\hat{m}}(s,a)$ **then**
27: $pref(\hat{m}) \leftarrow pref(\hat{m}) + \psi\,[r - \bar{r}]$
28: **end if**
29: $\bar{r}' \leftarrow \bar{r}' + \kappa\,[r - \bar{r}']$
30: $\bar{r}'' \leftarrow total\ /\ t$
31: $\bar{r} \leftarrow \max(\bar{r}',\ \bar{r}'')$
32: $s \leftarrow s'$
33: **end while**
34: **return** $\mathrm{argmax}_m\, pref(m)$

図 3.1: 最適センサ集合選択のための提案手法の処理．詳細は第 3.2.2 節本文参照.

1: $N \leftarrow$ number of actions avairable to the robot

2: $pref(m),\ \rho(m) \leftarrow 0,\ \ \forall m \in M$

3: $Q_m(s,a) \leftarrow$ initial value, $\ \forall m \in M,\ s,\ a$

4: $t, total \leftarrow 0$

5: $s \leftarrow$ initial state

6: **while** $t <$ planned transision times **do**

7: $\quad t \leftarrow t+1$

8: $\quad$ **for** $m \in M$

9: $\quad\quad \pi(m) \leftarrow \dfrac{e^{\,pref(m)}}{\sum_M e^{pref(M)}}$

10: $\quad$ **end for**

11: $\quad$ according to $\pi(\cdot)$, select R-table $R_{\hat{m}}$
$\quad\quad$ which will decide the action

12: $\quad n \leftarrow$ number of $a,\ a = \mathrm{argmax}_a\, R_{\hat{m}}(s,a)$

13: $\quad$ **for** $a \in A$

14: $\quad\quad$ **if** $\ a = \mathrm{argmax}_a\, Q_{\hat{m}}(s,a)\ $ **then**

15: $\quad\quad\quad \pi_{\hat{m}}(a) \leftarrow \frac{1-\epsilon}{n} + \frac{\epsilon}{N}$

16: $\quad\quad$ **else**

17: $\quad\quad\quad \pi_{\hat{m}}(a) \leftarrow \frac{\epsilon}{N}$

18: $\quad\quad$ **end if**

19: $\quad$ **end for**

20: $\quad$ choose action $\hat{a}$ according to $\pi_{\hat{m}}(\cdot)$, and execute $\hat{a}$

21: $\quad$ get reward r, and observe state s'

22: $\quad total \leftarrow total + r$

23: $\quad$ **for** $m \in M$

24: $\quad\quad Q_m(s,\hat{a}) \leftarrow\ Q_m(s,\hat{a}) + \alpha\,[r - \rho(m) + \max_{a'} Q_m(s',a') - Q_m(s,\hat{a})]$

25: $\quad\quad$ **if** $\ Q_m(s,\hat{a}) = \max_a Q_m(s,a)\ $ **then**

26: $\quad\quad\quad \rho(m) \leftarrow\ \rho(m) + \beta\,[r - \rho(m) + \max_{a'} Q_m(s',a') - Q_m(s,\hat{a})]$

27: $\quad\quad\quad pref(m) \leftarrow pref(m) + \xi\,\rho(m)$

28: $\quad\quad$ **end if**

29: $\quad$ **end for**

31: $\quad s \leftarrow s'$

33: **end while**

34: **return** $\ \mathrm{argmax}_m\, pref(m)$

図 3.2: R 学習高速化のための提案手法の処理．詳細は，第 3.3.2 節本文参照.

第 4 章

グリッドワールド実験

提案手法の適用によって，前章で期待した効果が得られることを確認するため，グリッドワールドを用いた，比較的簡単な設定の実験を実施した．ロボットに与えた課題は，障害物回避行動の獲得である．この課題は，自律移動型ロボットにとって最も基本的なものの 1 つである．とくに人工生命系の研究では，学習によってロボットに望ましい行動を獲得させる際に，第 1 に採用されている例がある (例えば，[27, 26] 参照)．

4.1 実験設定

4.1.1 行動環境，行動目標及び報酬

設定したロボットの行動環境を図 4.1 に示す．

ロボットは，図 4.1 左の S から出発し，図 4.1 右の 4 行動のうち 1 つを選択し，実行する．したがって，選択し得る行動の集合 A は，

$$A = \{\, \text{前}, \text{右}, \text{左}, \text{後} \,\}$$

となる．

実線は壁を示し，この壁に向かって行動した場合，元の場所に止まるものとした．ゴールは図 4.1 左の G であり，到達した場合，報酬 1 が与えられる．それ以外の行動の報酬は 0 とし，すなわち遅延報酬が与えられる環境となる．したがって，

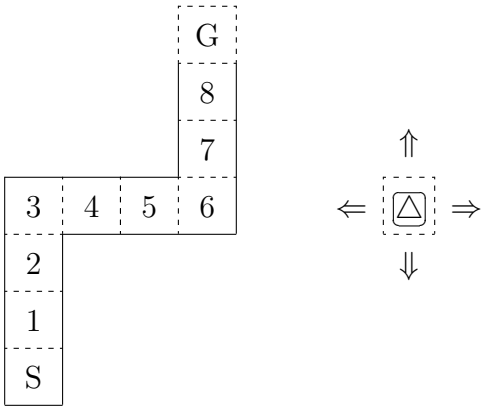

図 4.1: グリッドワールド実験における実験環境 (左) 及びロボットの行動 (右).

得られる報酬の集合 R は,

$$R = \{0, 1\}$$

である．ゴール到達後は S に戻って，規定行動回数まで行動を続けるものとした．

また，ロボットには前後左右に 4 つの近接センサが搭載され，それぞれの方向に衝突なしに移動可能であれば 0，移動すれば壁に衝突する場合 1 と観測するものとした．したがって，全センサを用いた場合の観測の集合 O は,

$$O = \{O^f, O^r, O^l, O^b\}, \quad O^{f,r,l,b} = \{0, 1\}$$

となる．

この実験は，簡単な内容ではあるが，ロボットはいずれのセンサ (の組合せ) を用いても，複数の状態において同一の観測しか得ることができない (例えば状態 1 と 2) ため，部分観測課題となっている．

仮に状態 3 にいて，後進によって状態 2 に移動した場合，状態 8 と同一の観測を得ることになり，ロボットはゴールに近付いたのか否かを Q 値から判断することが難しい．

4.2 実験とその結果

4.2.1 Q 学習 (最適センサ集合の特定)

提案手法の有効性評価のため，単純なグリッドワールド環境における実験を実施した．この課題は，センサの取付位置や数，ロボットに許された行動の種類により，冗長なセンサが搭載された例となり得る．

強化学習に用いた各パラメータは以下の通りである．

(1) Q 値表の学習率 (α)	0.05
(2) 割引率 (γ)	0.9
(3) 各 Q 値表の優先度の学習率 (ψ)	0.6
(4) リファレンス報酬の学習率 (κ)	0.001
(5) 探索行動選択確率 (ϵ)	0.1

(1), (2), (5) に関しては，一般的と思われる値を採用し，とくに最適化は行っていない．また，図 3.1 における Q 値表の優先度 ($pref$) の初期値及び各 Q 値表の Q 値の初期値は 0 に設定した. 上述の条件で，提案手法を用いて 10,000,000 回行動させた．以下では，規定回数行動させた実験の 1 回を試行と呼ぶものとする．各試行では，1 つ以上のセンサを利用する全センサ集合 15 ($= 2^4 - 1$) 通りを M の要素とした．20 試行繰返した結果，各試行の終了時に選択確率が最高のセンサ集合は，

(1) 前方のセンサのみ用いるもの	10 例
(2) 右側のセンサのみ用いるもの	3 例
(3) 前方と右側のセンサを用いるもの	7 例

で，いずれもその確率は 1.0 に達していた．

最適行動は，前方に壁がない場合は前進し，壁がある場合は右方向に動くという行動である．この行動の決定には，前方と右側のセンサしか関与しない．実際，これらのセンサのいずれか 1 つがあれば最適方策を表現可能で，すなわち，

- 前方に壁を観測した際は右方向に動く，それ以外は前進

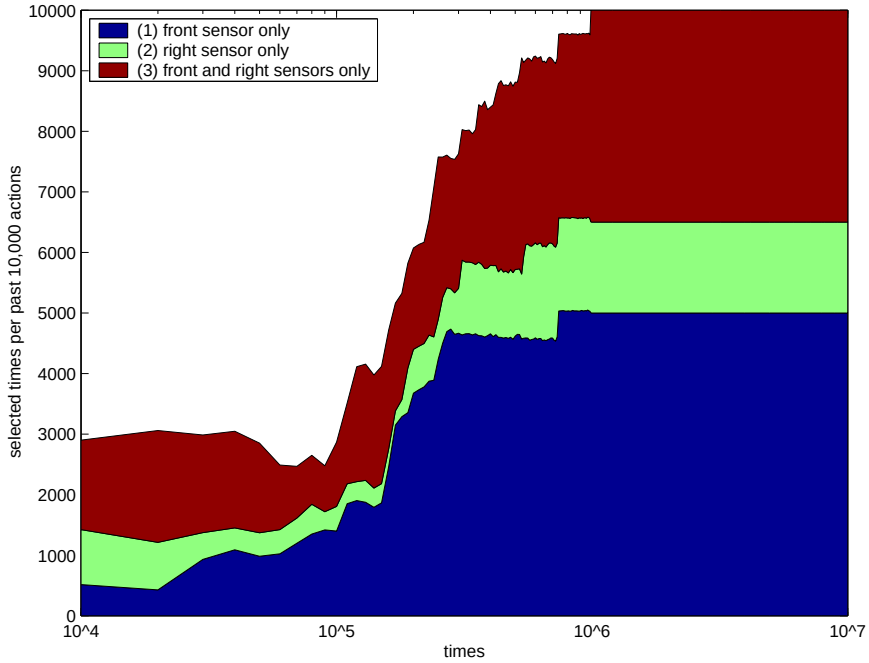

図 4.2: センサ集合の選択頻度の推移．横軸は行動回数 (log スケール)．縦軸は，過去 10,000 行動のうち各センサ集合が選択された回数を 10,000 行動で割った値であり，10,000 行動毎にプロットした．なお，選択回数は 20 試行の平均を用いた．

ないし

- 右方に壁を観測した際は前進，それ以外は右方向に動く

となる．以上のことから，提案手法の適用によって得られたセンサ集合は妥当であると考えられる．

なお，(3) のセンサ集合を利用した場合，観測されるパタンの数は，ゴール状態を除けば 2 つであり，(1) 及び (2) で観測されるパタン数と等しい．すなわち，上記 3 センサ集合は，学習速度の点ではいずれもほぼ同一であると考えてよい．(3) のセンサ集合を用いるという学習結果が得られた理由は，このためであると推察される．

また，上記 (1)–(3) のセンサ集合のそれぞれが，試行の各段階で，行動決定に用いられた頻度を確認した．結果を図 4.2 に示す．横軸は行動回数を log スケールで表示した．縦軸が，各行動回数において，過去 10,000 行動中 (1)–(3) のセンサ集合が選択され実際に行動を決定した頻度，すなわち 20 試行における選択回数の平均

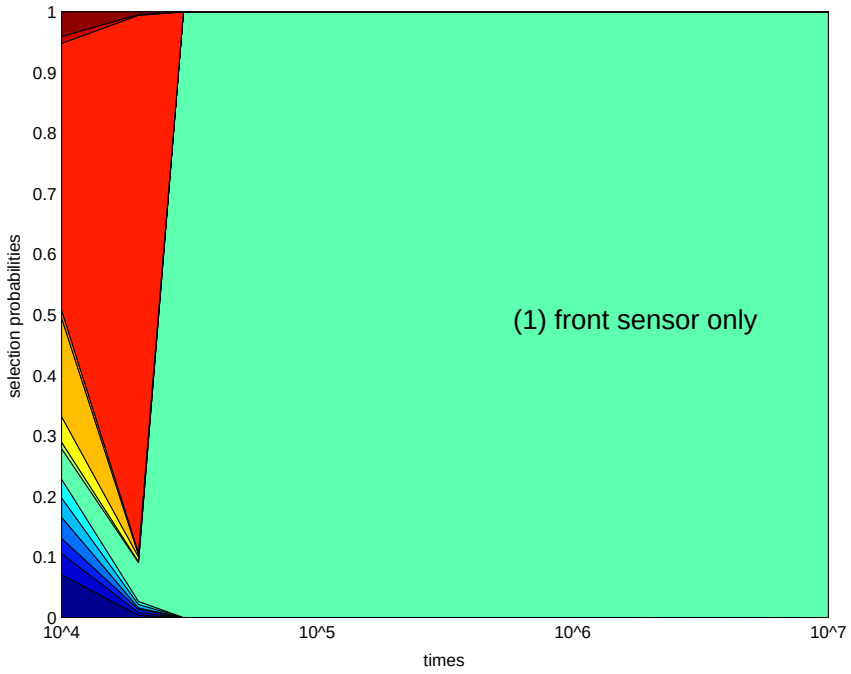

図 4.3: センサ集合の選択確率の推移．横軸は行動回数 (log スケール)，縦軸が各センサ集合の選択確率であり，10,000 行動毎にプロットした．

を 10,000 で割った値で，10,000 行動毎にプロットした．約 1,000,000 行動付近で，上記 3 センサ集合の選択確率の合計が，ほぼ 1 に達している．この時点で，3 センサ集合のうちのいずれかに収束したものと考えられる．

さらに，20 試行のうち 1 例に関して，各行動回数におけるセンサ集合の選択確率の推移を図 4.3 に示す．横軸は行動回数を log スケールで表示した．縦軸が各行動回数におけるセンサ集合の選択確率であり 10,000 行動毎にプロットした．注を付した領域が，前方のセンサのみを用いるセンサ集合の選択確率を示し，約 30,000 行動以降，継続的に 1 になっている．

センサの要・不要に関する事前知識がない場合，全センサを用いた強化学習を試みることは自然である．学習に必要な行動回数が，全センサを用いた通常の Q 学習より増大する場合，提案手法の適用に疑問が残る．そこで，全センサを用いる通常の Q 学習と平均獲得報酬の比較をおこなった．なお，対象実験の処理の内容は，第 A.1 節参照．

各 20 試行の結果は，図 4.4 の通りである．なお，20 個の乱数シードを用意し，

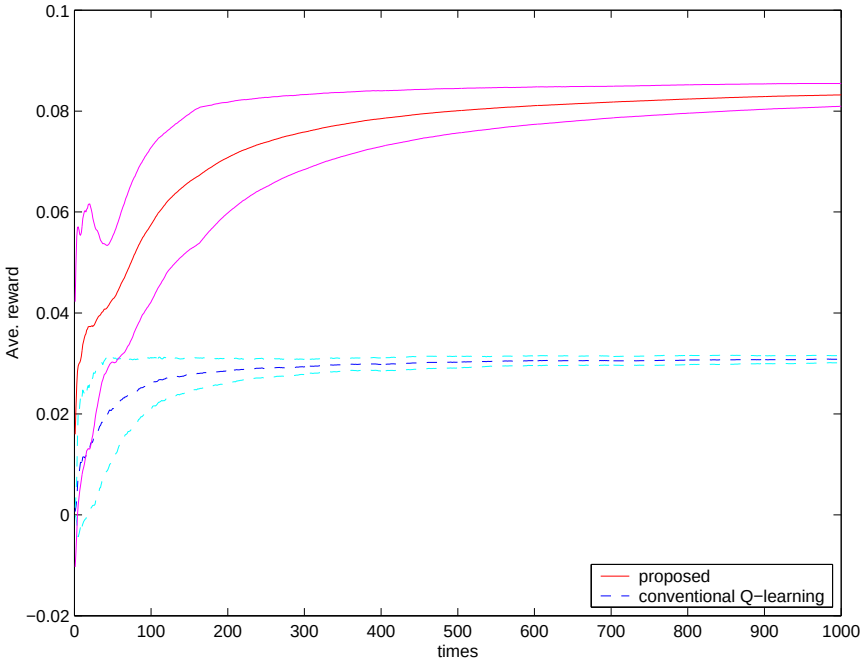

図 4.4: 平均獲得報酬の推移．実験開始時からの平均獲得報酬 (縦軸) を 10,000 行動毎に出力 (各プロット間を直線補間している)．横軸は，行動回数 (単位:10,000 行動)．実線が提案手法破線が (全センサを用いた) 従来手法の結果．それぞれ 20 試行の平均と，平均から標準偏差分離れた値を示す．

提案手法及び従来手法の各試行にそれぞれ適用して得られた実験結果である．また，強化学習パラメータに関しては，(1), (2), (5) と同一とした．

図の横軸が行動回数 (単位:10,000 行動)，縦軸が試行開始から各行動回数までに獲得した報酬の合計を行動回数で割った値を示す．実線が提案手法，点線が従来手法の結果であり，各々20 回の試行の平均値及び平均からの標準偏差を 10,000 行動毎にプロットした．

提案手法を適用した場合，従来手法と比較して高い平均獲得報酬を得ている．その値は，理論上の最高平均獲得報酬値である 0.1111(=1/9) に近い．ここで，実験の設定では，終了時まで $\epsilon = 0.1$ の探索行動を行っており，実際はこの値には到達しない．

提案手法によって選択されたセンサ集合を用いて，望ましい行動が学習されたことを確認するため，各試行終了時のセンサ集合と Q 値表を用いて，それ以上の

Q 値の学習は行わせず，第 4.2.1 節の (5) における $\epsilon = 0$ として，100,000 回行動させた．その結果，提案手法では，追加試行 19 回の平均獲得報酬が 0.1015 で，1 試行のみゴール到達なし，という結果であった．したがって，提案手法適用時，選択されたセンサ集合を用いて，最適行動が獲得されたと考えられる．

なお，ゴール到達回数が 0 であった追加試行に関しては，適切なセンサ集合は獲得され，Q 値の学習も進んだものの，探索による Q 値の再学習が始まったところで規定行動回数に達したと推測される．実際，実験条件中 ϵ のみを変更し，$\epsilon = 0.1 \times (1.0 - \frac{t}{10{,}000{,}000})$ として，行動回数 t が増えるごとに値を減少させた場合，試行終了時の平均獲得報酬は 0.1023 となった．また，この ϵ を用いて，図 4.4 と同一の乱数シードで行った 20 試行における，試行終了時の平均獲得報酬の平均は 0.1014 であり，さらに終了直前の 10,000 行動における平均獲得報酬は 20 試行の平均で 0.1110 と，上述の最高平均獲得報酬値にほぼ一致した．

一方，従来手法の Q 学習に関して，図 4.4 の実験における各試行終了時の Q 値表を用い，それ以上の Q 値の学習は行わせず，$\epsilon = 0$ とした実験の結果は，7 試行の平均獲得報酬が 0.1111，残りの 13 試行に関してはゴール到達回数が 0 であった．各試行ごとにその学習過程を観察すると，どの試行でも Q 値の学習に伴って最適方策の獲得と再探索を繰返している．すなわち，特定の試行において最適方策を獲得し，残りは最適方策の獲得に失敗した，という訳ではない．この点は前段落の提案手法適用時の結果と同様である．

以上の結果から，一時的に最適方策を獲得するものの，それ以外の方策で行動している期間が長いことが，図 4.4 において，従来手法の平均獲得報酬が提案手法に劣っている理由であると考えられる．なお，$\epsilon = 0.1 \times (1.0 - \frac{t}{10{,}000{,}000})$ とした場合も，試行終了時の平均獲得報酬は 20 試行の平均で 0.0639 にとどまった．この際，試行終了直前の 10,000 行動における平均獲得報酬は，20 試行の平均で 0.1000 で，最高平均獲得報酬値との差が残っている．

図 4.1 の実験環境は，4 センサ全てを利用した場合でも，部分観測課題となることは上述の通りである．従来手法の Q 学習が最適行動獲得の点で提案手法に劣る理由は，冗長なセンサの存在が最適方策獲得を困難にしたためであると予想される．一方，提案手法においては，行動決定上重要なセンサを選択することで，観測と最適行動との関係が明確になり，最適方策を獲得したと考えられる．

4.2.2 R 学習 (学習効率化)

提案手法の適用によって，高確率で使用される Q 値表が絞り込まれ，従来の R 学習法より高速に学習が進むことを検証するために，グリッドワールドにおける実験を行った．

この実験では，上記 4 つの近接センサに加え，グローバルなセンサを 1 つ持ち，このセンサで，コースの S から 5 にいるときには 0，6 から 8 にいるときには 1 を観測するものとした．したがって，全センサを用いた場合の観測の集合 O' は，前述の O を用いて，

$$O' = \{O, O^g\}, \quad O^g = \{0, 1\}$$

となる．グローバルセンサを用いることで，近接センサのみの場合に比べエイリアス状態は減少しており，近接センサのみを用いた場合に発生する，状態 1, 2, 7, 8 の全てが同一観測となる深刻なエイリアスは回避される．

実験では，上述のグローバルなセンサは必ず用いるものとし，それ以外の 4 センサについて，1 つ以上のセンサの全ての組合せを各々利用する，15 の R 学習器を並列させた．

強化学習に用いた各パラメータは以下の通りである．

(1) Q 値表の学習率 (α)	0.05
(2) ρ の学習率 (β)	0.001
(3) 各 Q 値表の優先度の変化速度パラメータ (ξ)	0.1
(4) 探索行動選択確率 (ϵ)	0.1

各パラメータの値に関しては，一般的と思われる値を採用し，とくに最適化は行っていない．

提案手法を用いて 1,000,000 回行動させた結果を図 4.5 に示す．提案手法適用時の実験開始時からの平均獲得報酬を，10 実験で平均し，10 行動毎に実線でプロットした．併せて，平均から標準偏差分離れた値を表示した．なお，横軸は行動回数で，log スケールでプロットした．

また，比較のため，従来手法の R 学習の結果を，破線で示した．従来手法の R 学習では，全てのセンサを利用するものとした．なお，パラメータについては，上

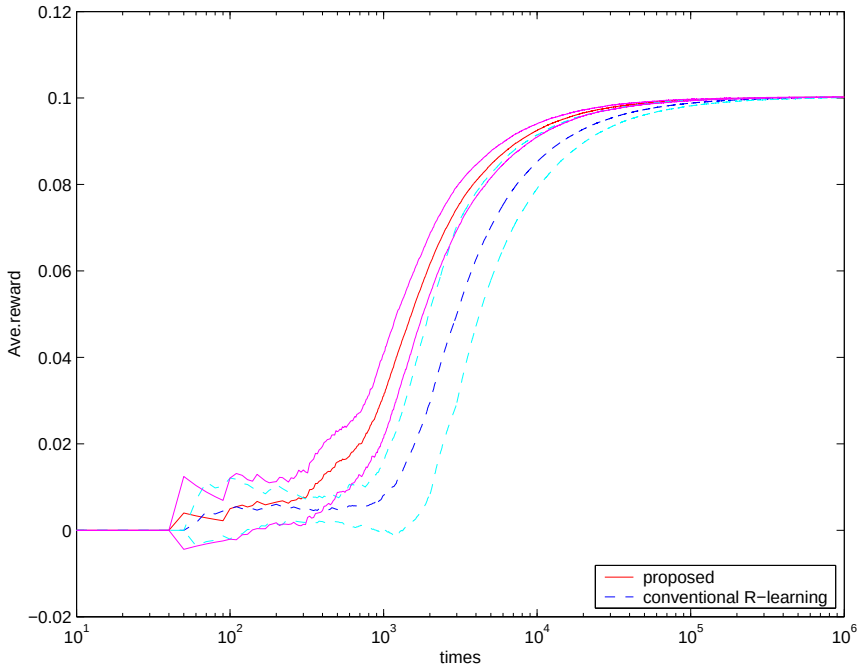

図 4.5: グリッドワールド実験における平均獲得報酬の推移．提案手法 (実線) 及び従来手法の R 学習 (破線) の結果．実験開始時からの平均獲得報酬の 10 実験平均及び平均値から標準偏差分離れた値をプロット．横軸は行動回数で，log 表示している．

記 (1), (2), (4) と同一とし，ϵ-greedy 探索手法を適用した．対象実験の処理内容は，第 A.2 節参照．

さらに，提案手法を適用した実験の 1 つにおいて，各強化学習器の選択確率の推移を 10 行動毎にプロットした結果を，図 4.6 に示す．手法の適用により，学習器の選択が進み，およそ 10,000 行動程度で，前及び後ろのセンサを用いる学習器に収束している．

この課題は，比較的単純であり，従来手法でも学習が可能である．実験終了時，従来手法と同等の平均獲得報酬に達していることから，提案手法でも学習が行われたと判断される．一方，平均獲得報酬の立ち上がりは，従来手法と比較して，提案手法の方がやや早い．この例のような単純な課題でも，学習速度の面で優位性があることが判る．

なお，前後左右の 4 つの近接センサのみを利用した場合，従来手法の R 学習，提

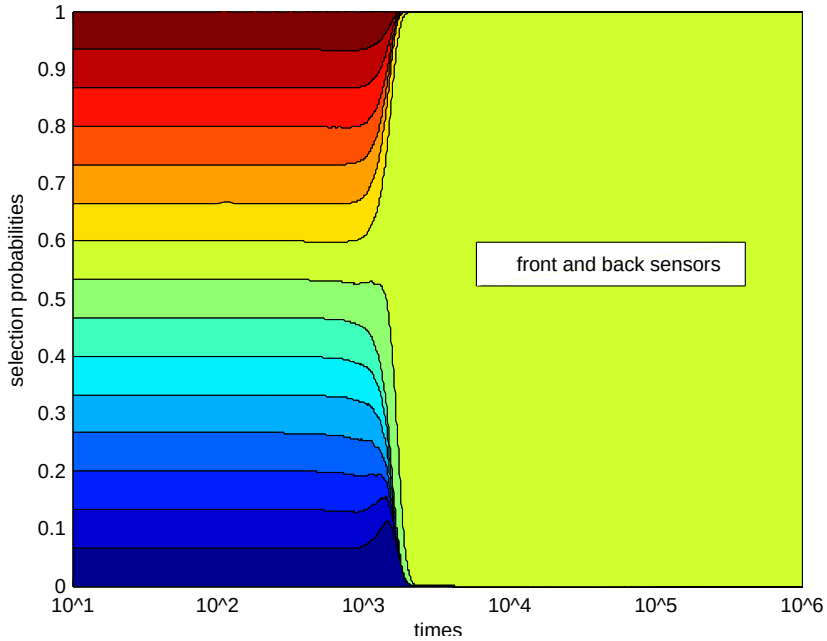

図 4.6: グリッドワールド実験における各強化学習器の選択確率の推移．横軸は行動回数で，log 表示している．

案手法共に学習が進まなかったため，上述の通りグローバルセンサを追加した．この点，前節の通り，Q 学習ではこのようなセンサの追加なしでも学習が可能であった．これはグローバルセンサを用いない場合，状態 1, 2, 7, 8 がエイリアスとなるとなるが，学習速度の速い R 学習にとって，とくに性質の悪いエイリアスとなったものと推測される．

部分観測状態が存在する場合，一般には Q 学習であっても収束は保証されない．R 学習の場合は，完全観測の場合にもその収束性は証明されていないため，提案手法の妥当性を数学的に示すことは極めて困難である．しかしながら，完全観測状態である場合や，不完全観測状態であっても，個々の R 学習器におけるグリーディな方策が，ある方策に収束する (Q 値は収束しない) 場合には，提案手法が収束することは，我々が行った実験の範囲では確認している．この場合，提案手法が収束するというのは，ある学習器または学習器群が確率 1 で選択され，それらが，同じ方策を持つようになることである．なお，探索を許しているので，それによる擾乱があっても，もとの方策を再学習するということである．

提案手法は，行動決定に用いる Q 値表の選択基準として ρ の累積値を用いている．このため，不完全観測状態であって，ρ 値の立ち上がりが非常に速い R 学習器が，実際にはどの方策にも収束しない学習器であった場合，学習器選択基準の更新 (図 3.2 行 27) が間に合わず，この学習器が選択され続ける可能性があると考えられる．

第 5 章

実ロボットシミュレータ実験

提案した手法の有効性を，より現実的な課題で評価するため，実ロボットのシミュレータを用いた実験を実施した．実験環境は，Cyberbotics 社製 Khepera ロボット[1] 用シミュレータ Webots2.0.8 [9, 10] 上に構築した．

第 4 章同様，障害物回避行動の獲得を，課題としてロボットに与えた．同様の課題は，先行研究でも，新しい学習手法提案の際評価に用いられている (例えば，[30, 32] 参照)．前章で述べた通り，ロボットに搭載されたセンサの位置や数及びロボットに許された行動の種類によっては，行動を決定する上で不必要な情報が与えられる例となり得る．このため，センサに冗長性のある例として本課題を採用した．

5.1 実験環境

実験に用いた環境及び環境上のロボットを図 5.1 に示す．これは，[iii] において，実ロボットを用いた実験を行った環境と近いものとなっている．鮫島ら [30] の実験では，環境は右折コーナのみであり，これに比較してやや難しい設定となっている．また，塩瀬ら [32] が用いた実験環境との比較考察は第 6.1.4 節にて行う．屈曲した領域を囲む，外側の正方形の 1 辺は，実世界で 1m に相当する．

なお，実験時，コーナ部分にトラップされる例が散見されたため，コーナ部分を滑らかにするため円筒形の物体 (図 5.1 左で壁の曲がり角部分の円) を配置した．

[1] ロボットの概観は，図 1.1 参照．

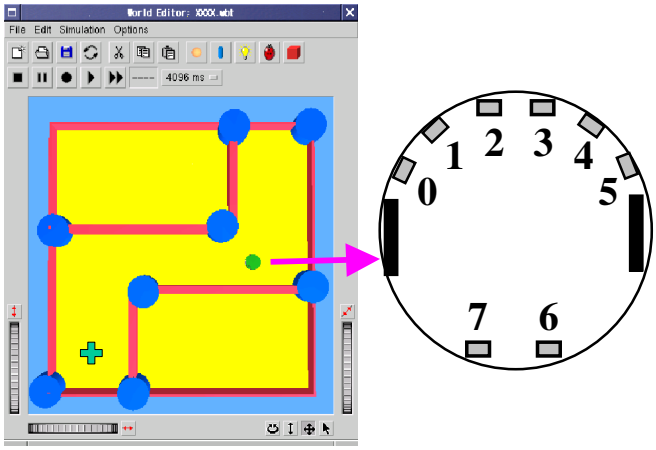

図 5.1: 実験環境 (左図) 及びロボット (右図)．右図中，黒い部分が車輪，網掛け部分が接近センサの取付位置及び向きを示している．なお，図中の+ 印は，実験の開始位置を示している．

トラップが発生する原因に関しては，実験環境の形状の他，壁との衝突 (異常接近) を判定する基準との関連が深いと考えられ，第 5.3 節にて詳述する．

5.2 ロボット

Webots は，Khepera ロボット (図 1.1 参照) 用のシミュレータであり，シミュレータ上のロボットの物理的特徴は実機に基づいたものとなっている．ロボットは，直径約 5cm の円筒型である．ロボットの周囲に沿って近接 (赤外線) センサが 8 つ搭載され，各々の取付位置は図 5.1(右) に示されている．近接センサの到達距離は，約 5cm となっている．

センサは，周囲の物体までの距離に応じて，1(遠い) から 1023(近い) の間の整数値を報告する．本研究では，これらの値を 5 つに離散化して用いることとした．本シミュレータには，センサ値を色別に報告する機能がある．離散化は，この色による区分を参考に，1–229, 230–329, 330–699, 700–929, 930–1023 の 5 分類とした．すなわち，本実験における観測の集合 O は，

$$O = \{O^0, O^1, O^2, O^3, O^4, O^5, O^6, O^7\}, \quad O^i = \{1, 2, 3, 4, 5\}$$

表 5.1: ロボットのとり得る 5 行動．各行動に対応する，ロボットの右輪及び左輪の速度コマンド (数字) を示す．絶対値が大きい程，車輪の回転速度が速く，正負は，正転 (前進) 及び逆転 (後退) に対応する．併せて，1 秒間に移動する，およその距離 (ロボットの中心で計測) 及び角度を示す．距離の単位は mm，角度の単位は °である．

	右輪速度	左輪速度	移動距離	角度変化
直進	+1.0	+1.0	8.0	0.0
右方向への前進	0.0	+1.0	3.6	8.7
左方向への前進	+1.0	0.0	3.6	8.7
その場での右転	−1.0	+1.0	0.0	17.6
その場での左転	+1.0	−1.0	0.0	17.6

である．なお，シミュレータは標準で，各センサ値に 10%のホワイトノイズを乗せる仕様となっている．これにより実機に近い実験条件が実現される．

5.3 実験条件

ロボットは，実験環境の一端の特定の位置 (図 5.1 左の + 印) に特定の方向に向けて置かれ，実験を開始する．1 実験当たり 6,240,000 回の行動を選択・実行させるものとした．ロボットは，シミュレーション環境上の時間で 64ms 毎に，行動選択を繰り返すため，1 実験は約 111 時間の行動に相当する．

ロボットに障害物を回避する行動を獲得させるため，鮫島ら [30] の実験を参考に，ロボットが選択した行動及び行動の結果に基づく即時報酬を与えた．Khepera ロボット (及びシミュレータ上のロボット) では，独立した左右輪の各々の回転速度を指定することで走行する．ロボットが選択可能な行動は 5 つ，すなわち，表 5.1 の通りとした．したがって，本実験の行動集合 A は，

$$A = \{\text{直進}, \text{右方向への前進}, \text{左方向への前進}, \text{その場での右転}, \text{その場での左転}\}$$

である．

また，行動の結果に伴う報酬は，

(1) 壁への異常接近 -0.5

(2) 直進 $+0.01$

(3) 右/左方向への前進 -0.01

(4) (その場での) 右/左転 -0.03

の通りである．したがって，本実験における報酬集合 R は，

$$R = \{+0.01, -0.01, -0.03, -0.5\}$$

と表される．

壁への異常接近の判定には，ロボットの近接センサデータを用いた．すなわち，1 つ以上のセンサ値が 930 以上となった場合，異常接近が生じたものとし，上述 ((1) 参照) の負の報酬を与えた．この際，ロボットの外周上で，センサの死角部分 (例えば，図中，タイヤ部分) が実験環境の凸部に接触しても，負の報酬が与えられず，ロボットがそこに止まるという現象が確認されたため，第 5.1 節に述べたようにコーナ部分を滑らかにした．

以上の設定は，エピソード分割されていない強化学習タスクと見做すことができ，一般に強化学習が適用されるエピソード分割されたタスクとは異なっている．ここでは，ロボットは，壁に衝突した場合，壁から離れる行動を自ら獲得する必要がある．通常の実験 (例えば [30]) では，壁に衝突した際は負の報酬を得て再度スタート状況に戻るという，完全にエピソード分割された実験設定が採用されている．しかし，シミュレーション上ではない実験環境を考えた場合，エピソード終了後，ロボットを再度スタート状況に戻すためには，時間的・物理的な負担が発生すると思われる．こうした環境では，本実験の設定の方が適すると考えられ，実際のロボットに応用する際有益であると考えた．

第 6 章

実ロボットシミュレータ実験の結果

6.1 実験 1: オンラインセンサ選択

6.1.1 実験 1 の設定

第 3.2 節では，複数 Q 値表を用いて，最適センサ集合のオンライン選択を実現する処理を提案した．実際の行動の決定に，複数の Q 値表のいずれを用いるかに関しては，この問題を n 本腕バンディット問題 (n-armed bandit problem) [35, 28] と見做し，強化比較手法 (第 2.3.5 節参照) を用いた強化学習で学習させることとした．すなわち，本論文で提案する手法は，通常の強化学習と同様に，適切な行動を (Q/Sarsa 学習で) 学習すると同時に，これとは独立に，複数の Q 値表のいずれを行動決定に採用すべきか (すなわち，行動決定に当たって，どのセンサ集合を用いることが適切か) を，n 本腕バンディット強化学習課題として学習し，その結果として，(報酬の累積値が最大となる) センサ選択を実現する枠組みといえる．

より現実的な，第 5 章の環境において，提案手法の効果を確認するため，以下の設定で実験を行った．強化学習に用いた各パラメータは，以下の通りである．

(1) Q 値表の学習率 (α) 0.05

(2) 割引率 (γ) 0.9

(3) 各 Q 値表の優先度の学習率 (ψ) 0.6

(4) リファレンス報酬の学習率 (κ) 0.01

(5) 探索行動選択確率 (ϵ) 0.1

なお，(1), (2), (5) に関しては，一般的と思われる値を採用し，とくに最適化は行っていない．また，Q/Sarsa 学習によって学習し，行動の決定を行う各 Q 値表の，Q 値の初期値に関しては，0 とした．

6.1.2 実験 1 の結果

提案手法を適用した場合と，全センサを用いた通常の強化学習を適用した場合とを，それぞれ 10 実験ずつ実施した結果を示す．図 6.1 上段が Q 学習，下段が Sarsa 学習による実験結果である．横軸は，行動回数 (単位:10,000 行動) であり，縦軸は，学習開始時からの平均獲得報酬を示す．両図共，提案手法適用時の 10 実験の平均及び平均からの標準偏差を実線で示した．また比較のため，全センサを用いた通常の強化学習適用時の，10 実験の平均及び平均からの標準偏差を破線で表した．なお，比較実験での行動選択にも，ϵ-greedy 手法 ($\epsilon = 0.1$) を用い，強化学習パラメータとしては，第 6.1.1 節の (1), (2), (5) と同一とした．処理の詳細は，第 A.1 節参照.

提案手法適用時，Q 学習及び Sarsa 学習共，比較的早い段階から高い平均獲得報酬値を示し，従来手法の Q/Sarsa 学習にまさる結果が実験終了時まで継続している (提案手法では，約 $\frac{1}{6}$ の行動回数で，全センサを用いる従来手法における，実験終了時の平均獲得報酬と同等の値に達している)．一方，通常の学習手法と比較して，提案手法の方が結果のばらつきが大きい．この点は，実際にどのような利用センサ集合が選択されたかによって，成績に違いがでた結果と予想される．

次に，過去 10,000 行動の間に壁に衝突した率 (単位:%) を 10,000 行動毎にプロットした図を示す．図 6.2 上段が Q 学習，下段が Sarsa 学習の結果であり，横軸が行動数 (単位:10,000 行動)，縦軸が衝突率を表している．各々実線が提案手法の適用時，点線が (全センサを用いる) 従来手法の強化学習適用時の推移である．これらの図でも，提案手法適用時，比較的早い段階から障害物を回避する行動を獲得していることが明らかである．ただし，Q 学習においては，実験の後半，通常の学習手法に追い付かれてしまっている (この点に関しては，第 6.1.4 節で詳述する).

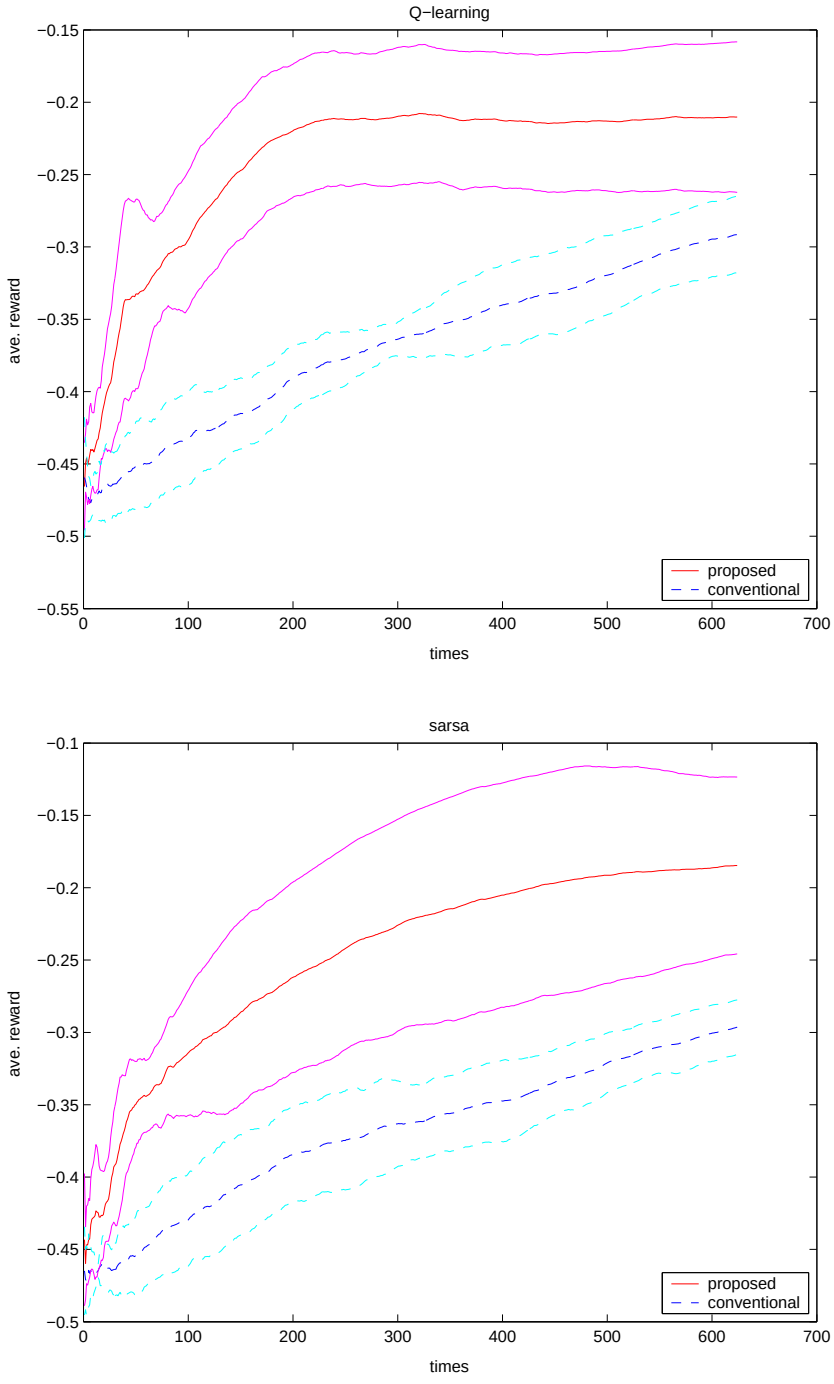

図 6.1: 平均獲得報酬の推移．Q 学習 (上段) 及び Sarsa 学習 (下段)．実験開始時からの平均獲得報酬 (縦軸) を 10,000 行動毎に出力 (各プロット間を直線補間している)．横軸は，行動回数 (単位:10,000 行動)．実線が，提案手法 10 実験の平均と，平均から標準偏差分離れた値を示す．比較のため，従来手法 (全センサ利用時) の結果を破線で示す．

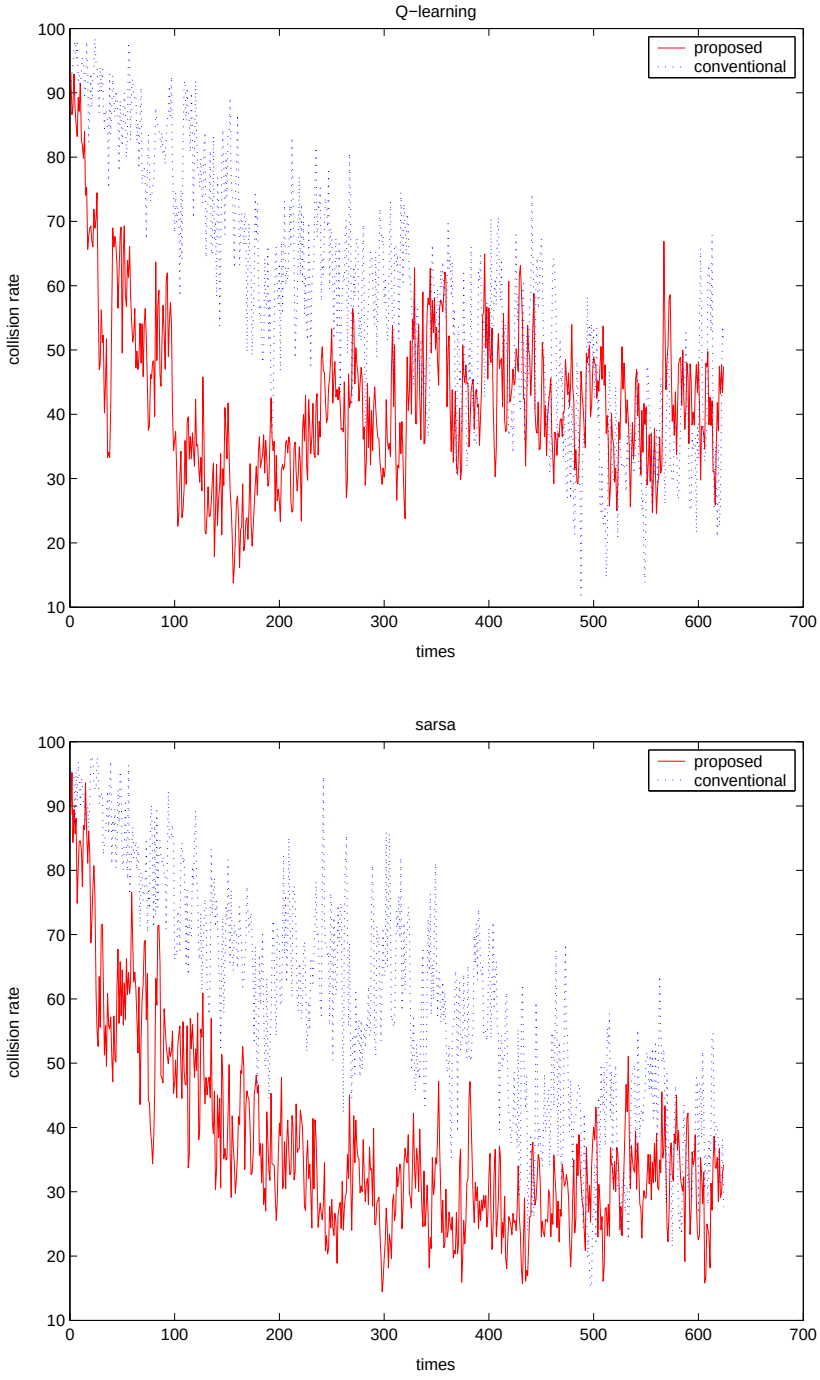

図 6.2: 衝突率の推移． Q 学習 (上段) 及び Sarsa 学習 (下段)．過去 10,000 行動の衝突率 (単位:%) を 10,000 行動毎に出力 (各プロット間を直線補間している)．横軸は，行動回数 (単位:10,000 行動)．実線が，提案手法 10 実験の平均. 比較のため，従来手法 (全センサ利用時) の結果を点線で示す.

表 6.1: 実験終了時の利用センサ集合．上段が Q 学習，下段が Sarsa 学習の結果を示す．比較のため，Q 学習，Sarsa 学習共，全センサを用いた従来手法による実験 10 回の平均獲得報酬の平均を最下行に示した．

Q 学習

	利用センサ集合の内訳 (センサ番号)	選択確率	平均獲得報酬 (終了時)
実験 1	0, 1, 2, 7	0.020	–0.1842
実験 2	1, 4	1.000	–0.0713
実験 3	7	0.042	–0.2123
実験 4	0, 4, 7	0.018	–0.2413
実験 5	0, 1, 3, 4, 5, 6	0.008	–0.2220
実験 6	1, 3, 6, 7	0.007	–0.2368
実験 7	1, 2, 4, 5, 6	0.020	–0.2378
実験 8	1, 3, 6, 7	0.009	–0.2182
実験 9	1, 2, 6	0.970	–0.2376
実験 10	0, 1, 3, 5, 6	0.013	–0.2414
提案手法 10 実験の平均			–0.2103
従来手法 10 実験の平均			–0.2915

Sarsa 学習

	利用センサ集合の内訳 (センサ番号)	選択確率	平均獲得報酬 (終了時)
実験 11	1, 2, 3, 6	1.000	–0.1504
実験 12	1, 3, 5, 6	0.827	–0.2324
実験 13	2, 4, 6, 7	0.010	–0.1720
実験 14	1, 2, 3, 4, 5, 6, 7	0.018	–0.1279
実験 15	2, 3, 4, 5, 6, 7	0.008	–0.2490
実験 16	0	0.016	–0.0744
実験 17	3, 6	0.016	–0.2441
実験 18	0, 1, 2, 3, 6	0.008	–0.2368
実験 19	1, 4, 6, 7	1.000	–0.1336
実験 20	0, 1, 3	1.000	–0.2257
提案手法 10 実験の平均			–0.1846
従来手法 10 実験の平均			–0.2964

さらに，提案手法を適用した 10 実験に関して，実験終了の際，選択確率 (π) が首位であった利用センサ集合の内訳 (センサ番号に関しては，図 5.1 右を参照) とその選択確率，及び実験開始時からの平均獲得報酬を表 6.1 に示す．

Q 学習では 10 例中 2 例，Sarsa 学習では 4 例において，選択確率が 1.0 に近い値となっており，利用するセンサ集合が特定されている．これらの例に関して，特定されたセンサ集合のみを用い，さらに $\epsilon = 0$ として走行させる実験を行った．この結果，6 例中 5 例に関しては，10,000 回行動しても壁との衝突が見られず，平均獲得報酬 (最下位は四捨五入値) は，

実験 2	+0.0044
実験 11	+0.0067
実験 12	−0.0100
実験 19	+0.0073
実験 20	−0.0100

の通りであった (実験 12 及び 20 に関しては，値は良くないもののきちんと前進している)．このため，センサ集合の選択，及び選択されたセンサ集合に基づく行動の学習が適切に行われたと判断できる．

残る 1 例 (実験 9) に関しては，左側の障害物に関しては適切に回避できるが，右側障害物に衝突してしまう．これは，選択されたセンサ集合に，右前方を感知するセンサ (センサ番号 3–5) が含まれていない (表 6.1 上段参照) 点に問題があると考えられる．このセンサ集合が選択確率最大になったのは，6,220,000 行動より後であり，望ましいセンサ集合が選択されて学習が終了したわけではない，すなわち，このセンサ集合を用いて行動を続けることで，より望ましいセンサ集合 (もしくは，より適切な Q 値表の内容) が獲得されるものと思われる．

一方，実験終了時，センサ集合の選択確率が低い 14 例に関しては，利用センサ集合の絞り込みが十分であるとはいえない．しかし，これらの実験例においても，従来手法による学習結果と比較して，実験終了時の平均獲得報酬が向上している (表 6.1 参照)．これは，センサ集合を絞り込みを進める過程で，複数のセンサ集合を行動決定に用いている段階でも，より適切な行動を選択する傾向があるものと理解される．

さらに，これら 14 の実験例に関して，実験開始時からの平均獲得報酬が最大であった時点の，行動数，利用センサ集合の内訳とその選択確率，その時点の平均獲得報酬を，表 6.2 に示す．上段が Q 学習，下段が Sarsa 学習の結果である．

これらの結果のうち，選択確率が 1.0 に達している 8 例に関して，特定されたセンサ集合のみを用い，$\epsilon = 0$ として走行させる実験を行った．この結果，Q 学習の 6 例に関しては，10,000 行動後も衝突が見られず，平均獲得報酬 (最下位は四捨五入値) は，

実験 1	+0.0065
実験 3	+0.0058
実験 4	+0.0065
実験 5	+0.0064
実験 8	+0.0065
実験 10	+0.0064

の通りであった．

全ての結果で，平均獲得報酬が 0.005 以上になっていることから，主として直進行動をとっていることは明らか (直進と斜め方向前進の組合せのみと仮定すれば，$\frac{3}{4}$ 以上の行動が直進，斜め方向前進は $\frac{1}{4}$ 以下と考えられる) であり，適切な Q 値表が得られたと推定される．さらに，これらの結果で共通して用いられている，左前方の障害物を感知するセンサ (センサ番号 4) に関しては，予備実験の際，このセンサのみを利用する設定で，壁沿い行動の獲得が可能であることが確認された．これらのことから、適切なセンサ集合も学習できたと考えられる．

したがって，この時点で既に適切な利用センサ集合と挙動方策を得ていたが，その後学習を継続させたため，再度センサ集合の探索に入ったと考えられる．図 6.2 上段に示された，Q 学習における衝突率の低下が，とくに実験後半で十分でない点 (先述) も，この再度のセンサ集合探索で説明可能である．一方，Sarsa 学習の 2 例に関しては，部分的に正しい行動が見られるものの，完全ではなかった．

なお，典型的と思われる実験結果に関して，学習の推移を含めた詳細を第 B 章に添付した．これらの結果も，上記の理解を裏付けると考える．

表 6.2: 最大平均獲得報酬時の利用センサ集合．表 6.1 中，終了時にセンサ集合の選択確率が 1.0 から離れている実験のみを表示．上段が Q 学習，下段が Sarsa 学習．

Q 学習

	行 動 数	利用センサ集合の内訳 (センサ番号)	選 択 確 率	平均獲得 報 酬
実験 1	3,200,000	1, 4, 6	1.000	–0.1344
実験 3	2,210,000	0, 4	1.000	–0.1391
実験 4	530,000	1, 4, 6, 7	1.000	–0.2119
実験 5	2,390,000	0, 4	1.000	–0.1732
実験 6	6,110,000	1	0.086	–0.2342
実験 7	400,000	0, 4, 7	0.091	–0.1834
実験 8	1,700,000	1, 4, 6	1.000	–0.1316
実験 10	1,120,000	4	1.000	–0.1876

Sarsa 学習

	行 動 数	利用センサ集合の内訳 (センサ番号)	選 択 確 率	平均獲得 報 酬
実験 13	3,710,000	0, 1, 2, 6, 7	0.024	–0.1366
実験 14	4,700,000	0, 1, 2, 6, 7	1.000	–0.0940
実験 15	6,120,000	6, 7	0.016	–0.2458
実験 16	5,290,000	1, 4, 6	1.000	–0.0367
実験 17	6,240,000	3, 6	0.016	–0.2441
実験 18	6,220,000	6	0.030	–0.2359

6.1.3 実験 1 の補足実験

前節で報告した提案手法の効果をさらに確認するため，以下の項目に関する追加実験を実施した．

(1) 複数 Q 値表優先度の学習の有効性の確認

(2) 強化学習パラメータ値の変更に対するロバスト性の確認

(3) 実験環境の変更に対するロバスト性の確認

次章以降で，その結果を報告する．

6.1.3.1 Q 値表選択処理の効果の確認

本研究では，Q 値表の優先度の学習に基づいて，複数の Q 値表から，実際に行動を決定する Q 値表を選択する手法を提案した．前節の実験では，Q 値表の選択問題を n 本腕バンディット問題と定式化し，強化比較手法を適用する処理を採用した．

この処理が有効に機能していることを確認するため，優先度の学習を行わない，すなわち，実験開始から修了まで，常に等確率 $(\frac{1}{255})$ で各 Q 値表が選択されるという条件で，Q/Sarsa 学習各々3 回の実験を実施した．その結果を，表 6.3 に示す．

表 6.3 の結果から，等確率で Q 値表の選択を行った場合，提案手法による優先度学習を実施した結果と Q 値表を 1 つしか用いない従来手法の結果の，中間的な平均獲得報酬が得られたと考えられる．

6.1.3.2 強化学習パラメータに関するロバスト性の確認

本研究では，提案手法の有効性の評価のための実験を，第 6.1.1 節に記述した強化学習パラメータ値にて行った．本節では，提案手法における強化学習パラメータに関するロバスト性の確認のため，上記以外のパラメータを用いて実施した実験の結果に関して述べる．

本節の実験で採用したパラメータ値は，

表 6.3: Q 値表を等確率で選択した際の平均獲得報酬．実験終了時の平均獲得報酬 (3 実験の平均値) を，優先度学習を行った場合及び全センサを利用する従来手法 (各々10 実験の平均) との比較で示す．

			平均獲得報酬
Q 学習	提案手法 (優先度学習)	10 実験平均	−0.2103
	等確率選択時	3 実験平均	**−0.2757**
	従来手法 (全センサ利用)	10 実験平均	−0.2915
Sarsa 学習	提案手法 (優先度学習)	10 実験平均	−0.1846
	等確率選択時	3 実験平均	**−0.2310**
	従来手法 (全センサ利用)	10 実験平均	−0.2964

各 Q 値表の優先度の学習率 (ψ)	0.6, 0.1
リファレンス報酬の学習率 (κ)	0.01, 0.001

の通りである．これらのパラメータ値を用いて，オンラインセンサ選択手法を Q/Sarsa 学習に適用した実験の，実験終了時の平均獲得報酬 (10 実験の平均) を表 6.4 に示す．また，比較のため，全センサを用いた通常の Q/Sarsa 学習の結果を再掲する．表から明らかなように，提案手法を適用した場合，ψ 及び κ の値の設定によらず，通常の全センサを用いる学習手法と比較して良い平均獲得報酬が得られた．

6.1.3.3 実験環境に関するロバスト性の確認

本研究では，提案手法の有効性の評価のための実験を，第 5.1 節に記述した実験環境にて行った．本節では，上記以外の実験環境における，提案手法の有効性確認のための実験の結果に関して述べる．本節の実験で採用した実験環境は，図 6.3 の通りである．この環境は，図 5.1 における実験環境の外周を斜めに切り取った，三角形の形状をしている．また，サイズ等は，図 5.1 の実験環境と同一である．

この環境において，第 6.1.1 節と同一の条件で，オンラインセンサ選択手法を適用した Q 学習と，全センサを用いる通常の Q 学習による実験を各々3 回実施した．

表 6.4: 提案手法適用時，異なる強化学習パラメータを用いた実験の結果．実験終了時の平均獲得報酬 (10 実験の平均) を示す．ψ は，各 Q 値表の優先度の学習率，κ は，強化比較手法のリファレンス報酬の学習率である．比較のため，Q/Sarsa 学習共，全センサを用いた従来手法の結果 (10 実験の平均) を再掲する．

	κ	ψ	平均獲得報酬 (終了時)
Q 学習+センサ選択	0.01	0.6	−0.2103
	0.01	0.1	−0.2600
	0.001	0.6	−0.0864
	0.001	0.1	−0.2573
Q 学習 従来手法 (全センサ利用)			−0.2915
Sarsa 学習+センサ選択	0.01	0.6	−0.1846
	0.01	0.1	−0.2409
	0.001	0.6	−0.0546
	0.001	0.1	−0.2410
Sarsa 学習 従来手法 (全センサ利用)			−0.2964

結果を，表 6.5 に示す．この実験環境でも，提案手法を適用した場合，全センサを利用した従来手法と比較して，高い平均獲得報酬を達成している．

さらに，この実験における平均獲得報酬の推移を図 6.4 に示す．横軸は，行動回数 (単位:10,000 行動) であり，縦軸は，学習開始時からの平均獲得報酬を表す．提案手法を適用した Q 学習 10 実験の平均及び平均からの標準偏差を実線で，全センサを用いた通常の Q 学習の 10 実験の平均及び平均からの標準偏差を破線で示した．

図から明らかな通り，提案手法適用時，全センサを用いる通常の Q 学習と比較して，早い時期から平均獲得報酬の向上が見られ，実験終了まで優位性が継続している．しかし，平均獲得報酬値自体は，提案手法適用時及び従来手法共，図 5.1 の実験環境での結果に及ばない (第 6.1.2 節の実験結果参照)．

本実験の環境は，図 5.1 の実験環境と比較して，よりフリースペースが多い．こ

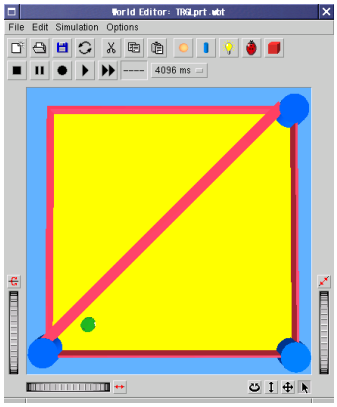

図 6.3: 実験環境に対するロバスト性確認のために用いた環境．サイズ等は，図 5.1 の実験環境と同一である．

のため，より自由に行動を選択できると考えられる．しかし，このことが逆に，学習に必要な制約の不足を招き，適切な行動の獲得に至らなかったのではないかと予想される．

また，図 5.1 の実験環境を用いた実験では，壁沿い行動及び領域の一部を周回する等，特徴的な行動を獲得する例があった．一方，本実験で用いた環境においては，学習の結果獲得された行動に，とくに特徴的な規則性は感じられなかった．この点に関しても，本実験で用いた環境が制約に乏しく，規則性の学習が十分に進まなかったことを裏付けると考える．

表 6.5: 三角形環境における実験の平均獲得報酬．実験終了時の平均獲得報酬 (3 実験の平均値) を，オンラインセンサ選択手法を適用した Q 学習及び全センサを利用する従来手法の Q 学習に関して示す．

		平均獲得報酬
Q 学習	提案手法 (オンラインセンサ選択)　3 実験平均	−0.2460
	従来手法 (全センサ利用)　3 実験平均	−0.3343

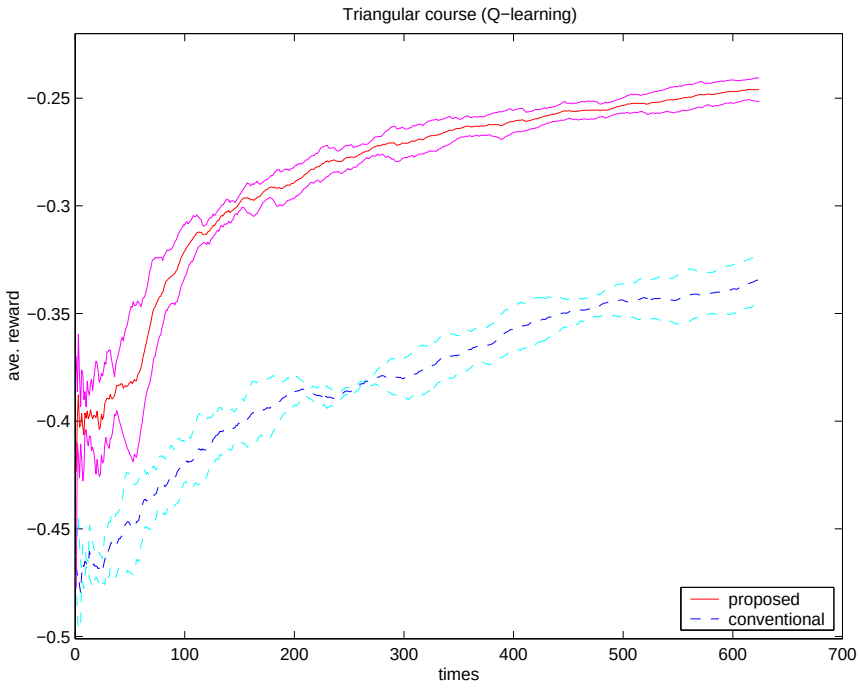

図 6.4: 三角形環境の実験における平均獲得報酬の推移．横軸は，行動回数 (単位:10,000 行動)，縦軸は，学習開始時からの平均獲得報酬．提案手法を適用した Q 学習 10 実験の平均及び平均からの標準偏差を実線で，全センサを用いた通常の Q 学習の，10 実験の平均及び平均からの標準偏差を破線で表した．

6.1.4 実験1の考察

強化学習においては，状態数爆発が，学習の阻害になることが指摘されてきた．この問題に対する従来のアプローチは，状態空間の効率的分割方法であった [7, 39, 2, 15, 30]．しかし，分割前後の状態空間を，実際の行動によって評価することは行われていない．すなわち，分割の基準に関して，事後の評価がないため，妥当性に疑問が残る結果となっている (詳細は，第 7.2 節参照)．

本研究で提案した手法は，センサ集合に対応して，複数の行動価値推定を同時に有し，その有効性を比較しながら，利用するセンサの選択を進めていく点に特長がある．本手法を採用することで，必要十分なセンサのみを用いた状態空間が選択されてくる．さらに，上記の従来手法と併用し，分割後の状態空間の適切さを実際の行動結果に基づいて評価することができるので，より適切な状態空間分割が可能となると考える．すなわち，よりコンパクトな状態空間を構成できる可能性がある．

実験の結果，提案手法の適用によって，通常の強化学習に比較して，平均獲得報酬及び衝突率の面で成績が向上した．したがって，同じ行動回数で，より適切な学習結果を獲得したといえる．一方，オンラインで最適な利用センサ集合を選択するという点では，一部の実験で目的が達成されたものの，必ずしも十分な結果ではない．これは，ϵ-greedy による探索が継続しているためであると考えられ，利用センサの絞り込み作業の完了を判断する基準 (第 6.1.2 節参照) を設け，学習途中で Q 値表の学習のみに切り替えるなどの手法を考えることで，ある程度解決できると思われる．実際，このようにすれば，Q 学習で 10 例中 6 例，Sarsa 学習で 10 例中 5 例まで，最適なセンサ集合が選択されたことになる．なお，この基準に関しては，今後の研究が必要である．

第 3.2 節で指摘した通り，通常の強化学習手法を用いてセンサ選択を行う場合，利用センサの組合せの数だけ実験を繰り返す必要がある．今回実験を行った計算機環境では，通常の強化学習手法を適用した実験 1 回につき，約 1 時間を要した．このため，センサ選択を行うために必要となる実験時間は，単純計算で約 255 時間であると考えられる．一方，提案手法を適用した場合，実験 1 回に要する時間は，約 2 時間であり，上述の通りセンサ選択効果が得られた．さらに，提案手法を適用した場合，迅速に学習が進む．第 6.1.2 節で示したように，全センサを用いた

通常の強化学習手法において，約 6,000,000 行動の学習後に得られた平均獲得報酬に達するために，提案手法では約 1,000,000 行動しか要しない．以上のことから，より少ない行動回数でセンサ選択を実現するという当初の目的は，十分達成されたといえる．

本論文で提案した手法は，可能なセンサ集合全てと，それに基づく Q 値表をもち，行動毎に更新していくため，1 行動あたりの計算負荷が増大し，適切な時間内に次の行動が決定できなくなるおそれがある．この点に関しては，本研究では実験終了時まで全てのセンサ集合を保持して実験を行ったが，重要度の低いセンサ集合を随時除去することで，処理の負荷軽減を図ることも可能である．除去にあたっては，その基準に関する検討が必要であろう．

行動決定に必要なセンサを特定することは，とくにセンシングにコストが発生する条件下で，非常に有益である．提案手法の適用でセンサの絞り込みが実現できれば，より少ないコストで望ましい行動が決定できる．Tan らの手法 ([39]．なお第 7.2 節も参照) は，逐次的な状態空間分割の際，センシングコストに配慮することで，学習の段階からセンシングコストを抑えることが可能であり，この点では提案手法より優れているものの，事前にセンシングコストが判明していることが前提とされている．提案手法は，コストに関する事前知識なしに適用可能で，センサ絞り込みによりコストが低減する可能性がある．さらに，コストに関する事前知識が得られる場合には，上述のセンサ集合除去の基準にコストを反映させることで，学習段階からある程度のコスト低減を実現させることも可能であると考える．

次に，第 1.1 節で紹介した，進化的手法との比較という観点から考察する．塩瀬ら [32] は，自律移動型ロボットの通路通過課題に進化的手法を適用する実験を行った．この結果，進化した世代では，全センサのうち一部のセンサのみで行動を決定していることが確認された．この点では，本実験と同等の結果を得たものと考えられる．しかし，実験条件の軽微な変更 (センサの到達可能距離をせばめる) で，コースを通過する行動の学習に失敗したと報告されており，ロバスト性の点で問題があると思われる．こうした結果は，コース設定にも一因があるのではないかと考える．塩瀬らが採用したコースは，左折が 2 回続いた後，右折を迎える．このため，左折したコーナに過度に適応した結果，続く右コーナでも左折行動を選

択してしまうのではないかと予想される．

進化的手法を用いた場合，左折したコーナに適応できない個体は，右折したコーナでの適応可否を試される前に，集団から排除されてしまう可能性がある．その理由は，進化的手法においては，方策の更新は世代の交代時のみ行われ，更新に反映されるのは，その時点の最終結果に限定される．すなわち，最終結果に至るまでの過程は無視され，各方策は，個々の行動が最終結果に及ぼした影響 (重要度) とは無関係に，単に最終結果の良し悪しに基づいて評価される．したがって，1 度も実行されなかった行動ですら，評価の対象になる [35]．一方，強化学習では，各状況毎に評価が行われる．本研究では，学習環境に右左折をバランス良く配置すると共に，学習手法として強化学習を採用した．強化学習を用いた場合，実験環境 (コーナの出現順序) に影響を受けることが少ないと考えられ，よりロバストな結果が期待できる．

本実験では，予備実験の結果，実際に行動決定に用いたセンサ集合のみを学習の対象にする手法を採用した (第 3.2.2 節参照)．しかし，適切なセンサ集合を選択させる強化学習課題では，センサ集合とロボットの行動とが，1 対 1 に対応付けられている訳ではない．例えば，本実験の例では，行動の選択肢の数が 5 に対して，センサ集合の数は 255 となっている．したがって，実際に行動の決定に関与した以外のセンサ集合が選択された場合にも，結果としてロボットが同じ行動をとる可能性がある．このことは逆に，実際には選択されなかったセンサ集合に関しても，そのセンサ集合が行動を決定した際の結果 (報酬) を，実質的に知る可能性があることを示しており，通常の強化学習課題と大きく異なる点の 1 つとなっている．こうした場合，同じ行動につながるセンサ集合すべてを，得られた報酬による学習の対象にするという考え方も自然であり，より少ない行動回数で最適なセンサ集合を選択できる可能性がある．こうした考えに基づく拡張の，1 つの例として，第 3.3 節の処理を考えることも可能である．次節にて，この処理を適用した実験の説明を行う．

6.2 実験 2: R 学習の効率化

6.2.1 実験 2 の設定

第 3.3 節では，R 学習において，学習速度低下の原因となる局所解に陥った際，迅速な脱出を実現するための手法として，複数 Q 値表を用いる手法を提案した．実際の行動の決定に当たって，これら複数の Q 値表のいずれを用いるかに関しては，各々の Q 値表 (すなわち利用センサの組合せ) に対して，その Q 値表の優先度を与えておく．この優先度を基に，softmax 計算 (第 2.2.1 節参照) を行って，各々の Q 値表の選択確率 (π) を求め，その選択確率に基づいて，Q 値表のうちの 1 つを選択する．

Q 値表の優先度については，その Q 値表の平均報酬の近似である ρ の累積値を用いる．この処理によって，ρ の値の大きい (すなわち，期待報酬の大きい) Q 値表はより選択されやすくなる．一方，グリーディに選択した行動の報酬が良くなかった Q 値表や，これまでの行動の結果が良くない Q 値表の優先度に対しては，負ないし小さい正の値が加算されることにより，徐々に選択される確率が減っていく．結果として，報酬の大きい行動が，グリーディとなっていた回数の多い Q 値表が，実際の行動を決定することになると考えられる．

以上の提案手法を，より現実的な第 5 章の環境で評価した．なお，本実験で強化学習に用いた各パラメータは，第 4.2.2 節と同一である．以下に再掲する．

(1) Q 値表の学習率 (α) 0.05

(2) ρ の学習率 (β) 0.001

(3) 各 Q 値表の優先度の変化速度パラメータ (ξ) 0.1

(4) 探索行動選択確率 (ϵ) 0.1

各パラメータは，実験 1 の結果との比較のため，第 6.1.1 節と極力同じ値を採用した (詳細は次節参照)．

また，複数の Q 値表に関しては，次のようにして準備した，利用センサが 3 つ以下のセンサ集合 (92 通り) から，一様ランダムに 20 のセンサ集合を選ぶ．これに，全センサを用いるセンサ集合を加え，計 21 のセンサ集合を要素とする集合を

表 6.6: 提案手法適用時の実験結果．終了時の利用センサ集合とその選択確率，及び平均獲得報酬を示す．比較のため，Q 学習，Sarsa 学習共，オンラインセンサ選択手法を適用した場合と，従来手法 (全センサを用いる) による実験 10 回の平均獲得報酬の平均を下行に示した (これらの結果の詳細は，第 6.1.2 節参照)．

		平均獲得報酬 (終了時)
R 学習+提案手法	10 実験の平均	−0.0492
Sarsa 学習+センサ選択	10 実験の平均	−0.1846
Q 学習+センサ選択	10 実験の平均	−0.2103
Sarsa 学習 (全センサ利用)	10 実験の平均	−0.2964
Q 学習 (全センサ利用)	10 実験の平均	−0.2915

図 3.2 における M とする．全センサ利用のセンサ集合を含めた理由は，ランダム選択の 20 のセンサ集合のみでは，必要なセンサがないため学習が原理的に不可能または非常に難しくなる場合があり得ると考えられるためである．

6.2.2 実験 2 の結果

6.2.2.1 提案手法

R 学習を行う複数の Q 値表 (第 3.3 節参照) を用いた実験の結果は，表 6.6 の通りである．この表は，提案手法を適用した 10 実験に関して，実験開始時からの平均獲得報酬を表している．比較のため，前節の，Q 及び Sarsa 学習にオンラインセンサ選択手法を適用した場合と，全センサを用いた通常の手法の結果を示す．R 学習を採用した前の実験，さらには Q 学習及び Sarsa 学習の成績と比較しても，格段の成績向上が明らかである．

次に，学習の経緯を示す．図 6.5 は，平均獲得報酬の推移である．横軸は，行動回数 (単位:10,000 行動) であり，縦軸は，学習開始時からの平均獲得報酬を 10,000

表 6.7: 第 6.2.2.2 及び 6.2.2.5 節の実験結果．従来手法 (全センサを用いる) による R/Q/Sarsa 学習の実験 10 回の平均獲得報酬の平均を下行に示した．Q/Sarsa 学習の結果に関しては第 6.2.2.5 節本文参照.

	average reward (end of experiments)
R-learning (with all equiped sensors)	−0.4945
Q-Learning (with all equiped sensors)	−0.2915
Sarsa (with all equiped sensors)	−0.2964

行動毎にプロットした．10 実験の平均及び平均からの標準偏差を実線で示した．また，比較のため，オンラインセンサ選択手法を適用した Q 学習 (破線) 及び Sarsa 学習 (点線) の結果 (10 実験の平均のみ) を再掲した.

本手法適用時，きわめて早い段階から，高い平均獲得報酬値が得られる．この状態は，実験終了時まで継続し，Q 及び Sarsa 学習にオンラインセンサ選択手法を適用した場合にまさる結果につながっている．さらに，過去 10,000 行動の間に壁に衝突した率 (単位:%) を 10,000 行動毎にプロットした結果を示す (図 6.6)．横軸が行動数 (単位:10,000 行動)，縦軸が 10 実験で平均した衝突率を表している．実線が本手法の適用時であり，比較のため，オンラインセンサ選択手法を適用した Q 学習 (破線) 及び Sarsa 学習 (点線) の結果 を付した．この図からも，障害物を回避する行動を早い段階で獲得し，維持するという点で，本手法適用時の結果が，Q 学習及び Sarsa 学習より優れていることが明らかである.

6.2.2.2 R 学習 (ϵ–greedy)

次に，従来手法の R 学習を適用した結果を，表 6.7 最上段に示す．探索手法としては，ϵ–greedy を用い，第 6.2.1 節の (1), (2), (4) と同一の強化学習パラメータを採用し，実験終了時の平均獲得報酬を 10 実験で平均した．なお，処理の詳細は，付図 A.2 参照.

上の結果が示すように最低報酬値 (−0.5) に近い平均獲得報酬しか得られていな

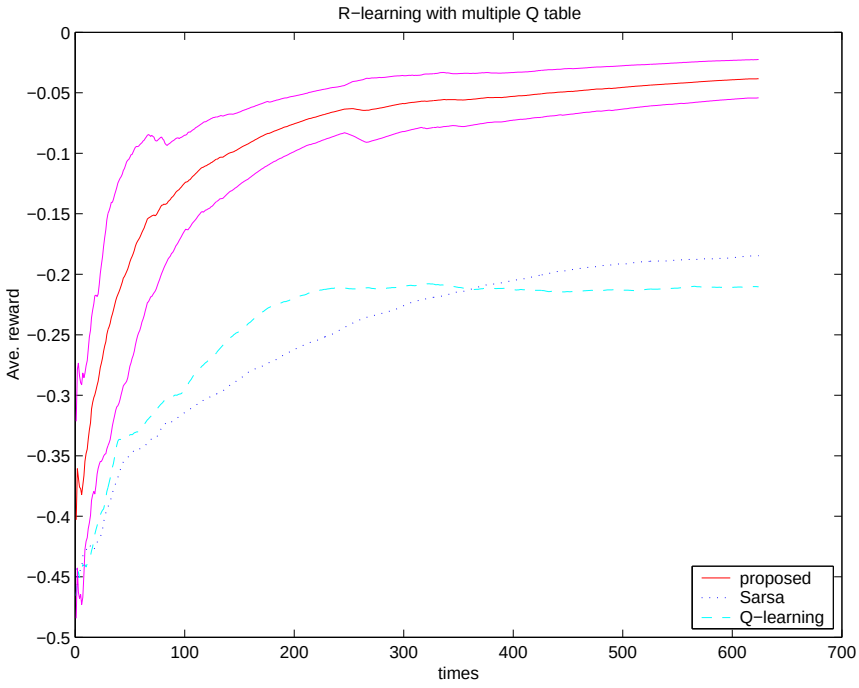

図 6.5: 平均獲得報酬の推移．提案手法を適用した R 学習の結果 (実線)．実験開始時からの平均獲得報酬 (縦軸) を 10,000 行動毎に出力．横軸は，行動回数 (単位:10,000 回)．10 実験の平均と，平均から標準偏差分離れた値を示す．比較のため，オンラインセンサ選択手法を適用した Q 学習 (破線) 及び Sarsa 学習 (点線) の結果 (10 実験の平均) を点線で示す．

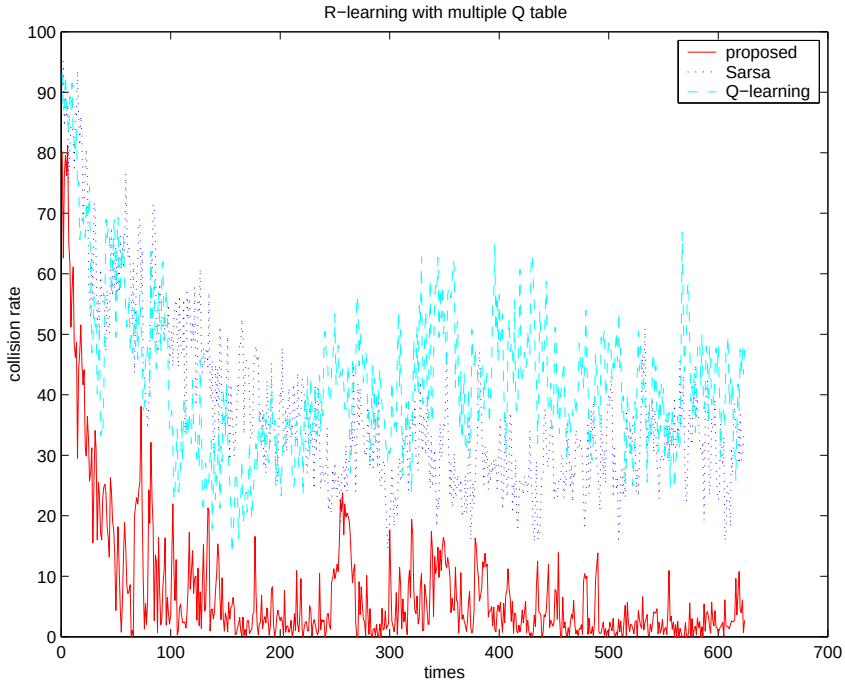

図 6.6: 衝突率の推移．過去 10,000 行動の衝突率 (単位:%) を 10,000 行動毎に出力．横軸は，行動回数 (単位:10,000 回)．実線が，提案手法 10 実験の平均．比較のため，オンラインセンサ選択手法を適用した Q 学習 (破線) 及び Sarsa 学習 (点線) の結果 (10 実験の平均) を点線で示す．

表 6.8: 実験 6.2.2.3 UE を用いた実験結果．実験終了時の平均獲得報酬を示す．

	average reward (end of experiments)
experiment 6.2.2.3–1	−0.3401
experiment 6.2.2.3–2	−0.3446
experiment 6.2.2.3–3	−0.3338
experiment 6.2.2.3–4	−0.3371
experiment 6.2.2.3–5	−0.3409
experiment 6.2.2.3 (R-learning by UE)	mean −0.3393

い．この成績は，第 6.2.2.5 節における従来手法の Q 及び Sarsa 学習の結果にも遠く及ばず，きわめて悪い結果といえる．これは，大半の実験において，実験の初期段階に壁に衝突しそのまま脱出できなかったためである．

なお R 学習では，Q 学習より探索行動を増やすことで好成績が得られる場合があるとの指摘 [23] に基づき，ϵ 値ないし Q 値の学習率 (α) の値の割増し，オプティミスティック初期値の採用等を試したが，効果は確認できなかった．実験当初は，細かく動きながら壁からの脱出を試みているように見えたが，次第にそうした試みも弱まる．これは文献 [22] で報告されている状況と似ており，局所解に陥っていると推定される．

6.2.2.3 R 学習 (UE)

従来手法の R 学習において，探索手法として UE [23] を用いた実験の結果を表 6.8 に示す．5 回の実験の終了時の平均獲得報酬及びこれらの平均を記した．グリーディな行動を選択する確率 $p = 0.9$ とし，パラメータ c に関しては，文献 [23] と同一の $c = 60$ を採用した．また，文献 [23] における $N_f(s, a)$ については，初期値 $= 1$ とした．なお，それ以外のパラメータは，第 6.2.1 節の (1), (2) と同一である．処理の詳細は，付図 A.2 参照．

この結果も，提案手法は勿論，第 6.2.2.5 節の Q 及び Sarsa 学習の成績に及ばな

表 6.9: 実験 6.2.2.4 softmax を用いた実験結果．実験終了時の平均獲得報酬を示す．

	average reward (end of experiments)
experiment 6.2.2.4–1	−0.4647
experiment 6.2.2.4–2	−0.2210
experiment 6.2.2.4–3	−0.4412
experiment 6.2.2.4–4	−0.0564
experiment 6.2.2.4–5	−0.4343
experiment 6.2.2.4 (R-learning by softmax)	mean −0.3235

い．ただし，第 6.2.2.2 節の実験のように長期間壁にトラップされる現象はあまり見られず，探索を増した効果はあると思われる．

なお UE は，利用頻度の少ない行動を選んで探索する探索手法である．文献 [23] では，パラメータ c の値が R 学習の成績のばらつきに大きな影響を与えると報告されている．しかし，適切な値の設定には実験の繰返しが不可欠であり，現実的には難しい．

6.2.2.4 R 学習 (softmax)

従来手法の R 学習において，softmax 探索手法を用いた実験の結果を表 6.9 に示す．温度パラメータは定数 1 とし，第 6.2.1 節の (1) 及び (2) と同一の強化学習パラメータを採用した．5 回の実験における終了時の平均獲得報酬と，それらの平均を記した．なお，処理の詳細は，付図 A.2 参照．

softmax 手法は，文献 [22] では良い評価が与えられていないが，我々の実験では R 学習を用いた従来手法中最も成績が良かった．とくに 5 実験中 2 回の高い平均獲得報酬が得られた実験では，比較的早い段階で壁沿い行動を獲得していた．一方，他の 3 実験ではこの行動の獲得が十分には進まず，壁にトラップされることが多いため，低い平均獲得報酬にとどまった．

6.2.2.5 Q/Sarsa 学習

第 6.1.2 節における，全センサを用いた従来手法の Q 及び Sarsa 学習を適用した 10 実験に関して，実験開始時からの平均獲得報酬を表 6.7 の下 2 段に再掲する．実験では，割引 ($\gamma = 0.9$) を用い，強化学習パラメータは第 6.1.1 節の (1), (4) と同一，探索手法は ϵ–greedy である．なお，処理の詳細は，付図 A.1 参照．

従来手法の Q 及び Sarsa 学習は，本研究の実験課題では，提案手法を除く R 学習手法を適用した実験より良い成績であった．しかし，図 6.5 に示した通り，従来手法の Q 学習が約 6,000,000 行動後に達した平均獲得報酬値は，提案手法では約 100,000 行動も要せずに得ている．この違いにより，従来手法の Q 学習は提案手法に劣る結果となっている．さらに図 6.6 に示したように，壁に異常接近する率についても，従来手法の Q 学習は提案手法より高く，またそれが低下する速度が遅いため，提案手法に劣っているといえる．

6.2.2.6 CMAC+R 学習

強化学習において，推定価値関数の近似を行い，学習の高速化を図る手法として，タイルコーディングを用いる手法が提案されている [35]．この手法は，歴史的経緯から CMAC(cerebellar model articulation controller)[1] とも呼ばれる．

CMAC では，複数の相異なるタイリングを用いる．タイリングは，重なりあわない複数のタイルの集合であり，それぞれのタイルが各タイリングにおける受容野に相当する．観測状態は，その状態が属するタイルの集合で表現される．すなわち，状態 s がタイリング m のタイル n に属する場合 $B_{m,n}(s) = 1$，それ以外では $B_{m,n}(s) = 0$ とする．行動 a をとる Q 値を近似表現するため，各タイル毎にウエイト $w_{m,n}(a)$ を定める．すなわち，

$$Q(s,a) \approx \sum_{m,n} w_{m,n}(a)\ B_{m,n}(s)$$

とする．各ウエイトは，行動後，TD 誤差を小さくするよう更新される．

本実験では，可能な全てのセンサ集合に対応させてタイリングを作成した．したがって，255 のタイリングを用いた．CMAC を Acrobot の学習に適用した例 [35] では，タイリングとして，可能なセンサ集合の全てを用いている．また，センサ

表 6.10: 実験 6.2.2.6 CMAC を用いた R 学習の結果．実験終了時の平均獲得報酬を示す．

	average reward (end of experiments)
experiment 6.2.2.6–1	−0.4909
experiment 6.2.2.6–2	−0.4996
experiment 6.2.2.6–3	−0.4996
experiment 6.2.2.6–4	−0.0095
experiment 6.2.2.6–5	−0.4903
experiment 6.2.2.6 (R-learning with CMAC)	mean −0.3980

リーディングに関する汎化を得るため，各センサ集合に対して，ランダムにオフセットさせた複数のタイリングを準備している．この例を参考にした．なおセンサ値に関しては，各タイリングでのオフセットはさせなかった．すなわち第 5.2 節で説明した離散化が，そのまま各タイリングにおけるタイル分割となっている．このため，各タイリングの観測内容は，提案手法で想定する各学習器の観測と完全に同一である．なお，提案手法では，これらの学習器の部分集合を用いた点と，学習や行動決定の方法が異なる．

CMAC を適用した R 学習の実験結果を表 6.10 に示す．実験では第 6.2.1 節の (1), (2), (4) と同一の強化学習パラメータを用い，ϵ-greedy による探索を行った．なお，処理の詳細は，付図 A.3 参照．

R 学習と CMAC の併用では，表 6.10 から明らかな通り，学習が迅速に進む場合もあるものの，その数は 5 例中 1 例に止まっている．しかもこの例では，実験末期に障害物のない場所で，その場での旋回を繰返す行動を発現することがあり，適切な方策を学習したとはいえない．

表 6.11: 実験 6.2.2.7 CMAC を用いた Q 学習の結果．実験終了時の平均獲得報酬を示す．

	average reward (end of experiments)
experiment 6.2.2.7–1	−0.0659
experiment 6.2.2.7–2	−0.0199
experiment 6.2.2.7–3	−0.3317
experiment 6.2.2.7–4	−0.0722
experiment 6.2.2.7–5	−0.2246
experiment 6.2.2.7 (Q-learning with CMAC)	mean −0.1429

6.2.2.7 CMAC+Q 学習

CMAC と R 学習の併用では，R 学習が局所解に入って学習が進まない可能性がある．このため，CMAC と Q 学習を併用した実験を行った．結果を表 6.11 に示す．なお，この実験でも，実験 6 と同様のタイリングを用いた．また，割引 ($\gamma = 0.9$) を用い，強化学習パラメータは，第 6.2.1 節の (1), (4) と同一とし，ϵ-greedy による探索をさせた．なお，処理の詳細は，付図 A.3 参照．

Q 学習と CMAC の併用では，表 6.11 の通り，従来手法の Q/Sarsa 学習 (第 6.2.2.5 節参照) に対する優位性が認められるものの，第 6.2.2.6 節の実験同様，学習速度のばらつきが大きい結果となっている．

6.2.3 実験 2 の補足実験

前節で報告した提案手法の効果をさらに確認するため，以下の項目に関する追加実験を実施した．

(1) 複数 Q 値表優先度の学習の有効性の確認

(2) 強化学習パラメータ値の変更に対するロバスト性の確認

次章以降で，その結果を報告する．

表 6.12: Q 値表を等確率で選択した際の平均獲得報酬．実験終了時の平均獲得報酬 (10 実験の平均値) を，優先度学習を行った場合及び全センサを利用する従来手法の R 学習 (10 実験の平均) との比較で示す．

	平均獲得報酬
提案手法による優先度更新を行う場合 (第 6.2.2.1 節)	−0.0492
等確率で各 Q 値表が選択される場合	**−0.3930**
全センサを利用した従来手法の R 学習 (第 6.2.2.2 節)	−0.4945

6.2.3.1 Q 値表選択処理の効果の確認

第 6.2.2.1 節の実験では，Q 値表の選択のために，ρ の累積を用いる処理 (第 6.2.1 節参照) を採用した．この処理が有効に機能していることを確認するため，優先度の学習を行わない，すなわち実験開始から終了まで，常に等確率 ($\frac{1}{255}$) で各 Q 値表が選択されるという条件で，R 学習の実験を 10 回実施した．

実験の結果は，表 6.12 の通りであった．実験 1 の場合 (第 6.1.3.1 節参照) と同様，等確率で Q 値表の選択を行った場合，提案手法による優先度学習を実施した結果と Q 値表を 1 つしか用いない従来手法の結果の，中間的な平均獲得報酬が得られたと考えられる．

6.2.3.2 強化学習パラメータに関するロバスト性の確認

本研究では，提案手法の有効性の評価のための実験を，第 6.1.1 節に記述した強化学習パラメータ値にて行った．本節では，提案手法における強化学習パラメータに関するロバスト性の確認のため，上記以外のパラメータを用いて実施した実験の結果に関して述べる．

本節の実験で採用したパラメータ値は，

ρ の学習率 (β)	0.6, 0.1
各 Q 値表の優先度の変化速度パラメータ (ξ)	0.01, 0.001

の通りである．これらのパラメータ値を，複数の Q 値表を用いる R 学習に適用した実験の，実験終了時の平均獲得報酬 (5 実験の平均) を表 6.13 に示す．

表 6.13: 提案手法適用時，異なる強化学習パラメータを用いた実験の結果．実験終了時の平均獲得報酬 (5 実験の平均) を示す．比較のため，センサ選択手法を適用した Q 学習，Sarsa 学習 10 回の平均獲得報酬の平均を下行に再掲した (これらの結果の詳細は，第 6.1.3.2 節参照)．なお，ψ は，各 Q 値表の優先度の学習率，κ は，強化比較手法のリファレンス報酬の学習率である．

	β	ξ	平均獲得報酬 (終了時)	
R 学習+	0.001	0.6	−0.0504	
提案手法	0.001	0.1	−0.0564	
			−0.0492	*10 実験の平均
	0.01	0.6	−0.0727	
	0.01	0.1	−0.1160	
	κ	ψ	平均獲得報酬 (終了時)	
Sarsa 学習+	0.001	0.6	−0.0546	*10 実験の平均
センサ選択	0.001	0.1	−0.2410	*10 実験の平均
	0.01	0.6	−0.1846	*10 実験の平均
	0.01	0.1	−0.2409	*10 実験の平均
Q 学習+	0.001	0.6	−0.0864	*10 実験の平均
センサ選択	0.001	0.1	−0.2573	*10 実験の平均
	0.01	0.6	−0.2103	*10 実験の平均
	0.01	0.1	−0.2600	*10 実験の平均

表から明らかなように，提案手法を適用した場合，β および ξ の値の設定によらず，センサ選択手法を適用した Q/Sarsa 学習と比較して良い平均獲得報酬が得られた．提案手法は，平均獲得報酬ばかりではなく，ロバスト性の面でも優れていると考えられる．

6.2.4 実験 2 の考察

R 学習の学習速度低下の一因とされる，limit cycle [22] を回避するためには，行動のループに陥った際，そこから脱出するための探索的行動が有効である．これを実現するため，本論文では，複数の行動価値推定 (すなわち，Q 値表) を並置し，強化学習を用いて全ての Q 値表を同時に望ましい行動の獲得に向けて学習させると共に，現在までの平均報酬 (ρ) の累積が最も大きい Q 値表を，行動決定に用いるという手法を提案した．また， Q 値表に関しては，利用するセンサが少しずつ異なるものを複数用いた (第 3.3 節参照)．実験の結果，こうした設定が有効であることが明らかとなった．

ただし，複数 Q 値表の選択にあたっては，ρ の累積以外の基準を用いることも考えられる．また，複数の Q 値表を構成する方法に関しても，利用センサ集合を変える他に，例えば状態空間の離散化を変える方法等も可能である．複数の Q 値表を並置して，行動を決定するというアイデアは，[i] で初めて提案されたもので，さらに条件を変えた実験を重ねる必要がある．

この実験課題では，ロボットが壁にトラップされ学習が進まなくなる現象がしばしば発生する (本論文では，壁に接触もしくは異常接近した状況が継続することを，トラップ状況にあると表現するものとする)．実験の設定では，壁に一定以上近付き過ぎた場合，壁から十分な距離まで離れるまで，選択した行動によらず一定の負の報酬が与えられ続けるため，$r - \rho$ は Q 値が定常化するより速く 0 に近付く．このため，得られた報酬が学習に正当に反映されない．したがって，ある時点でグリーディであった状態 – 行動対が保存されやすく，学習結果がその状態 – 行動対の影響を受けることになる．これにより局所解が発生しやすくなる，すなわち，グリーディな行動選択によって得られる行動列は最適でない行動列に陥ることになると予想される．なお，Q 学習では，こうした局所解は基本的には発生

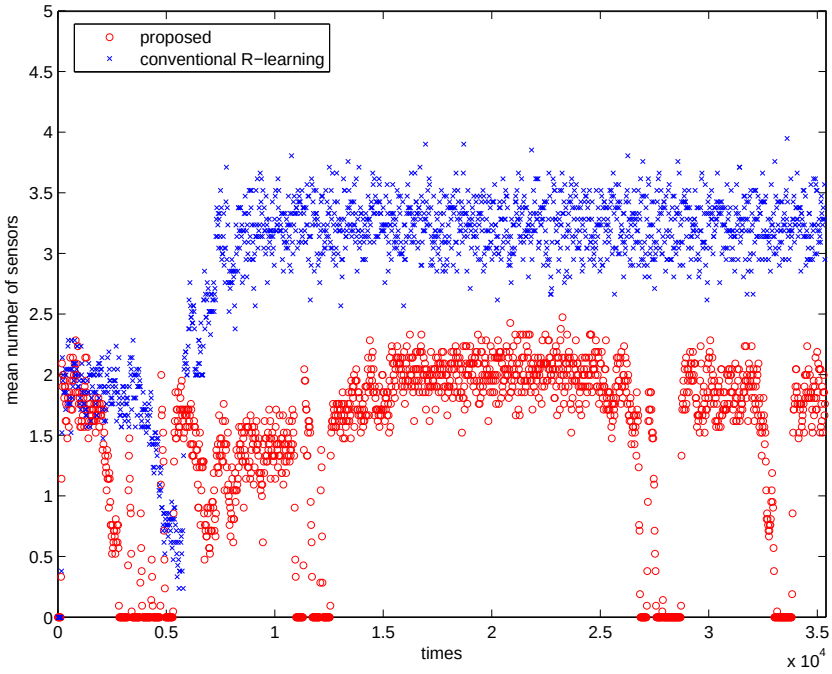

図 6.7: センサ値 > 910 のセンサ数の推移．20 行動の間の平均値を 20 行動毎にプロット．

しない．

R 学習を用いた実験 (第 6.2.2.2—6.2.2.4, 6.2.2.6 節) に対する優位性に関しては，提案手法の局所解を回避する効果によるものと推測される．

壁にトラップされている際の，提案手法と従来手法の R 学習との比較を行った．トラップの深刻さの指標として，センサ値が異常接近値である 910 に達したセンサの個数を用いた．図 6.7 は，第 6.2.2.1 及び 6.2.2.2 節と同一の条件で行った実験において，実験開始から約 35,000 回行動させた際，値が異常接近値を超えたセンサ数を 20 行動毎に平均しプロットしたものである．縦軸がセンサ数，横軸は行動回数を示す．丸印が提案手法，X 印が従来手法の結果である．

提案手法採用時の値が一貫して低いことは，実際に状態表現の分析により確認した結果，異常接近を示すセンサ数が少ない状態への探索が促進されたことを反映していると推測され，この探索促進の結果トラップの脱出ないし回避が可能となったと考えられる．

一方，第 6.2.2.5 節の Q 及び Sarsa 学習と比較して，提案手法の成績が良い点

は，トラップされていない状態において，R 学習が Q 及び Sarsa 学習より早く直進行動を学習するためと推察される．これは，R 学習が Q 学習より高速であるとの Schwartz の指摘 [31] を裏付ける結果と考えられる．

本研究の手法は，行動決定に用いる Q 値表の選択処理に関して，オンラインセンサ選択のために用いたもの (第 6.1.1 節参照) と異なる．実験 1 の際用いた手法では，どの Q 値表を用いるかという問題を，n 本腕バンデット問題 (n-armed bandit problem) として定式化し，強化比較を用いて選択を行い，1 回の行動で優先度更新が行われる Q 値表は，たかだか 1 つであった．一方，本実験の手法では，行動の結果に基づき，複数の Q 値表の優先度の同時更新を許す．R 学習では，行動価値推定の更新に，平均報酬 (ρ) 値を用いるが，この値は，強化学習エージェントの平均成績の見積もりを表現するものであるため，各 Q 値表の比較に適している．このため，上記一括更新に，ρ を用いた．

複数の Q 値表を並置する手法は，計算負荷の増大が 1 つの課題ではあるが，本研究の実験結果では，その欠点を超える効果が確認された．実験結果に関して，とくに，オンラインセンサ選択手法を適用した Q 及び Sarsa 学習に比べ，良い成績が得られたことに関しては，学習の対象となる Q 値表が，単数から複数に増え (上述)，効率良く学習ができた点が大きいと思われる．また，Schwartz が主張する，R 学習の迅速な学習を生かし，相乗効果が得られたという理由も考えられる．好成績の要因に関しても，必ずしも十分な分析を行えたとはいえず，今後，他の実験課題における結果や，他手法との比較も含めた分析が必要である．

R 学習の高速化に関しては，モデル (各状態間の遷移確率表) を用いる手法 (H-learning) が提案されている [36]．提案手法との成績の比較検討を含めた考察が，さらに必要であろう．H-learning を用いて，実ロボットにおける障害物回避行動の獲得課題を試みた研究例 [3] もある．ロボットの行動決定処理への応用という観点からは，シミュレーションと実ロボットでの実験との間に大きな懸隔があることが指摘されている [6]．このため，本研究の成果を実ロボット上に移植し，実験を行うことは，提案手法のロボティクス分野における応用という観点から興味深いと考える．

第 7 章

関連研究との比較

7.1 複数の Q 値表が存在する手法との比較

従来提案されていた強化学習手法の中には，複数の Q 値表を用いて行動決定を行うことを目的としている訳ではないが，結果的に，複数の Q 値表を用いる形が提案されているものも存在する．例えば，

(1) Actor-critic 手法

(2) 階層型強化学習手法 (hierarchical reinforcement learning)

があげられる．なお，複数のエージェント (例えばロボット) が，各々Q 値表をもち，同じ環境で行動すること (一般的にマルチエージェントといわれる実験状況) も，実験全体を考えた場合，複数の Q 値表を用いて学習をさせていることになるが，提案手法との差異は明らかなため，とくには取り上げない．以下，上記 2 手法との差異について考察する．

7.1.1 Actor-critic 手法との比較

Actor-critic 手法 [35] は，強化比較手法 (第 2.3.5 節参照) を，状態遷移のある強化学習課題に拡張した手法で，actor と呼ばれる，実際の行動を決定する部分と，critic と呼ばれる，行動結果を基に，actor を評価する部分とが共同し，全体とし

て強化学習の効果を生み出す．こうした仕組みは，脳における情報処理と近い [18] として，積極的に研究が進められ，適用例の多さでは，Q 学習，Sarsa 学習に迫っている．

actor, critic 共に強化学習機械と考えた場合，複数の強化学習が同時並行的に用いられているという見方も可能ではあるが，2 つが揃うことが，学習を進める上で不可欠である．一方，提案手法では，並列された Q 値表の 1 つを取り出して，学習と行動決定をさせることが可能である (ただし，学習の結果獲得された行動の妥当性は，取り出した Q 値表に依存する)．また，critic の用途は，actor の評価であり，行動決定を行う必要がない (行動決定は，actor の役割である) ため，状態価値のみを扱う，すなわち (テーブルではなく) 状態価値ベクトルの形をとる点も，相違点といえる．

7.1.2 階層型強化学習手法との比較

次に，階層型強化学習との比較検討を行う．本手法に基づいた研究例に関しては，2003 年に公表された解説論文 [4] で網羅的に紹介されているが，その後も成果の公表が続いており，強化学習領域において，現段階で積極的に研究が進められているテーマの 1 つといえる．

階層型強化学習では，上位の階層と下位の階層で，別々の Q 値表を用いる手法が用いられることがある．ただし，階層化は，本来行動のマクロ化を目的とした手法であって，上位の階層と下位の階層では，一般的に，扱う強化学習のタスクが異なる．また，下位の階層が，複数の Q 値表で構成される場合でも，これらの Q 値表が扱うべき強化学習のタスクは，各々異なる．例えば，高橋ら [37] は，課題をサブタスク分割し，各々のサブタスクの遂行を目的とする複数の Q 値表を用いると共に，現状況下でどのサブタスク遂行をすべきか (すなわち，下位階層の複数の Q 値表の，いずれに基づいて行動するかを，上位階層の Q 値表で決定するという手法を提案する．

この手法が，本研究で提案した手法と異なる点は，以下の通りである．本研究の手法は，課題をサブタスク化するものではない．各々の Q 値表は同一のタスクを遂行する目的のために用いられており，すなわち，(ロボットが遭遇する状態の

一部における適切な行動を学習しているのではなく) 全ての状態に対する望ましい行動を学習している．一方，高橋らの手法では，下位階層の Q 値表は，上位階層の Q 値表によって割り振られたサブタスクのみを学習することになる．すなわち，この手法においても，上述の Actor-critic 手法の場合と同様に，Q 値表のどれか一つを取り出して，タスクを遂行させることはできないと考えられる．

提案手法では，複数の Q 値表は，それぞれ独立して並列的に学習を進め，各時点で望ましいと考えられる行動を判断する．これらの行動判断は，ロボットが実際に選択する行動を決定する際の選択肢に相当し，ある意味で各 Q 値表が，それぞれ行動を提案し，ロボットは，これら競合する提案のうち，最も適切であると考えられる行動を決定し，実行するととらえることも可能である．

複数の Q 値表，それぞれをエージェントとみなすことが可能であることから，本論文では，両者を同一の概念として取り扱ってきた．これは，すなわち，1 つのロボットが，複数の内部エージェントから構成されるという理解に相当する．こうした理解は，例えば，[25] における，

> “管理” を行なう新しい層を，いろいろな時点で挿入することによって，精神発達を説明する理論を紹介した．(中略) 多層構造を構築しなければならないことについて議論した．そのネットワークの低いレベルには，さまざまな空間的，時間的な観察に専門化したエージェントがある．そして，高いレベルのエージェントは，それより低いレベルのエージェントの活動を分類し，そして制御することを学習する．(p. 315)

といった，内部に構造をもつエージェントにつながるアイデアであると考えられる．

ここで，従来の階層型強化学習研究は,「空間的な観察に専門化したエージェント」の併用を目指したものと考え得る．一方，例えば，[40] のように,「時間的な観察に専門化したエージェント」を主に意識した研究例もある．これらの研究では，最終的に選択されるエージェントの妥当性の判断が明確となるよう，ヒューリスティックに基づいてエージェント群が準備されている例が多い．一方，本研究では，創発性に重点を置き，より自然な形で複数のエージェントが準備されていると考える．以上の分析から，本研究及びこれらの選考研究は，複数の内部エージェントから構成される，すなわち内部に構造をもつエージェントの研究という

同一の目的に向かって，別の方向からアプローチをしているとみなすことも可能であると思われる．

7.2 関数近似手法との比較

第 3.2 節では，提案手法を用いて，最適なセンサ集合を特定する手法を提案した．最適センサ集合の特定は，適切な状態空間の獲得とも密接に関連している．この問題は，従来，強化学習における次元の呪い (第 2.2.5 節参照) 回避の一環として，研究が進められてきた．以下に，概要の説明と提案手法との比較を記す．

次元の呪いを解決するために，少ない状態数 (センサ数・基底関数個数) から開始し，状態分割 (新たなセンサ・基底関数の追加) を行って最適な状態数を求める方法が試みられている [7, 39, 2, 15, 30, 13]．

Chapman らの研究 [7] や，Tan の研究 [39] ではセンサを増加させている．増加の基準は，[7] では，センサを使用するか否かで即時報酬・累積報酬に統計的に有意な差が発生するか否かであり，[39] では，ある状態の状態価値が減少するという不整合な動きが発生するか否かである．[7] では，各センサは 2 値画像の各ピクセルに対応している．各状態は「センサ値=1, センサ値=0, 参照せず $= *$」からなるベクトルに相当する．各状態では，行動をとる毎に，参照していないセンサについて，そのセンサ値が 1 の場合と 0 の場合に分けて，得られる報酬を記録しておき (参照しないセンサが k 個であれば k 組)，両者間，すなわち，あるセンサを用いるか否かで報酬に統計的有意差があるとき，当該センサを使うように状態を分割している．その結果，一部のピクセルに反転ノイズがあっても，必要な状態分割のみ行われたと報告されている．この手法の短所は，第 1 に，あるセンサ 1 つを用いるか否かで有意な差が発生する状況が対象である，という点である．すなわち，センサが 2 個以上集まって意味が出てくるような場合には，本方法は適用できない．第 2 に，状態分割が行われる毎に，Q 値表を初期化して再学習しなければならない．このため，学習の効率性の面で問題がある．また第 3 として，ノイズが正規分布から大きくはずれるとき，統計的有意性の正当性が保証できず，分割 (すなわち複雑度の上昇) 後，その効果に関する事後検証が必要になる．

一方，[39] の手法は，センサ数が不足することによる，状態分割の必要性を，

information gain に基づいて判定する．分割が必要と判断された場合は，最小コストのセンサを追加することによりセンサ数を増加させる．この手法においても，適用の際，センサ値にノイズがないことが要求されるという短所がある．さらに，結果として，センサ数のかなりの増大 (この場合，観測すべき升目の個数であり，世界全体で 100 個あるうちの 50 個となった) がみられている点でも，有効性に疑問が生じる．

これに対して，センサ数を固定した上で，状態分割を適切に行うことを目的とする手法も提案されている．浅田らの研究 [2] では，ゴール状態を最初の既知状態 (状態はセンサ値ベクトルで表現する) とし，同一の行動 (行動は同一要素行動の繰返し) をとったとき同一の既知状態に到達する状態をグループ化 (クラスタリング) して，新たな状態とする方法を提案している．しかし，手法の適用に当たっては，ゴール状態が他状態から明確に区別され，どの状態もゴール状態からあまり遠くないことが必要である．すなわち，センサ値にノイズがなく (またはノイズが小さく)，状態数が十分に大きくなる前に，状態記述が完全になる (最適行動を決定するに必要かつ十分な状態記述になる，POMDP ではなく MDP になる) という暗黙の前提がある．一方，実世界のロボットのセンサ値にはノイズが重畳するのが通常であり，また POMDP であることも普通である．従って，実世界ロボットでは，この仮定は一般には成立たないと考えるべきである (勿論課題による．[2] の課題では MDP であった)．このため，例えば，障害物を避けながら長時間探索を行うことを学習するロボットに適用するのは適当でないと予想される．

石黒らの研究 [15] は，長時間探索を行うロボットの学習を対象とし，線形判別関数を用いて，即時報酬 (または割引累積報酬) が異なるとき状態を分割するという方法を提案している．しかし，ゴール状態から遠い状態の分割が遅れるという短所は残っている．

鮫島らは，状態空間を正規化ガウス関数を用いて分割・表現し，基底関数となっているガウス関数を必要に応じて追加していく方法を提案している [30]．ロボットの行動学習には Actor-critic 法 (第節参照) を用いている．各行動毎の選択度関数や状態価値関数は，基底関数の線型和である．基底関数の追加は，1 個の基底関数の代わりに 2 個の基底関数を用いることであるため，分割と呼ばれる．分割は，その基底関数が支配する局所領域上で，TD 誤差の平均が 0 に近いにも関わらず，分

散が大きいときに行われる．

以上の 5 研究に共通する短所として，状態の融合は行われていない，という点がある．すなわち，状態数を増加させる方法のみ提案しており，減少させる方法は提案されていない．一方，Mahadevan らの研究 [24] では，センサ値と Q 値が近ければ同じ状態とし，新状態の導入及び状態の融合を行っている．しかし，この手法は，センサ選択という用途に用いることはできない．さらに,「近さ」の判定がアドホックになる，すなわち，センサ間に重要度や reading のスケールの点で差があるときには，前提知識なしには使えないという欠点がある．

また，Kröse らの研究 [21] では，状態空間を Kohonen の SOM (Self-organizing map．例えば [11] 参照) で表現し，センサ値が近く，行動への関数が類似している状態 (SOM のニューロンで表現される) を融合する方法を提案している．SOM ニューロンを用いる手法は，強力な反面，一般的に収束が遅いため，学習に時間を要することが懸念される．

石井らは，NGnet (Normalized Gaussian Network) を対象に，データの入出力分布に応じた基底の配置が提案している [16]．すなわち，基底関数の追加を，事後確率が小さい入出力値（観測値と行動に相当する）がある場合に行ない，基底関数の削除を，その関数の使用頻度が減少したときに行う方法である．この手法では，出現頻度は少ないが，課題達成という観点から不可欠な状態行動対を適切に扱い得るか疑問が残る．この点に関しては，基底関数の追加・削除を判断する閾値の設定に大きく左右されると思われる．

深尾らは，[13] において，確率的な状態遷移が少ない課題を想定し，経験した状態をそのままデータとして蓄積しておき，必要に応じて，更新または削除を行う方法を提案している．ここで，蓄積されたデータは，状態を離散化するためのカーネルと同様に機能する．すなわち，ある状態に遭遇した際，その状態との距離が最も近いカーネルにおいて推定された Q 値を，遭遇した状態の Q 値として代用する．したがって，この手法も，行動の各時間ステップにおいて，入力情報と Q 値の間の関係が近いものを同一視するという観点に立つものと理解できる．

行動の結果得られた報酬と，蓄積済のデータとの整合性を評価することにより，データの追加や削除を行うが，これらは各々状態空間の分割と融合に相当する．しかし，POMDP 環境であるか否かを判断するために用いるパラメータ (b) は，ある

程度人為的に設定せざるを得ない．さらに，POMDP 環境と判断した場合，データが追加されるのみで，削除される頻度が低く，結果的にカーネル数が期待程少なくはならない可能性がある．

なお，高橋ら [38] は，状態分割・融合を行うのではなく，センサ値から行動によるセンサ値変化・報酬への関数を線型関数で近似表現し，この近似線型関数が同一である範囲を 1 つの状態とする方法をとっている．すなわち，新しく観測されたデータが，現在利用しているモデルに整合しないとき，今持っている全ての data sets (d_i) を用いてモデルを作りなおす．整合性の判定に当たっては，①クラスタリング及び線型回帰の際の残余誤差 (すなわち線型回帰しても残差が大き過ぎるとき)，②行動結果 (同一の行動で異なる報酬が得られた，または異なる次状態に遷移したとき)，が採用されている．さらに，ゴールに近い場所では，[2] の方法と組合せている．

以上の 9 手法は，いずれも，組合せ最適化問題の近似解法に用いられる近傍探索に相当する手法である．すなわち，現在の解候補に最も近い別解の中から，ヒューリスティックスを用いて，新たな解候補を作るという操作を繰返す手法である．これは，計算負荷の大きな最適化問題を，近似的に，しかし効率的に解く良い方法ではあるが，中途でまたは最終的に得られた状態空間が最適なものであることが確認されていない，少なくも他の状態空間に比べより適当なものであるということも確認されていない短所がある．状態分割の手法に関していえば，問題に依存したパラメータが多く，その設定が容易ではないという別の短所もある．

これに対して，提案手法は，従来提案されていた状態分割手法とは異なった，複数の Q 値表を同時並列的に用いるというアイデアに基づいている．そのため，複数の Q 値表を直接比較することが可能である．また，例えば従来の状態分割手法に提案手法を併用することができ，その結果，従来の手法の問題点であった，状態分割後の事後検証も可能となる．さらに，部分観測環境にあって，たとえ全てのセンサを使っても強化学習が収束しないような場合でも，その時々でより適切な状態空間を用いて (学習しながら) 行動するエージェントが構成できる可能性もある．

第 8 章

結　論

8.1　考察及び将来の研究

第 6.1.4 及び 6.2.4 節では，提案手法の具体的な用途 (オンラインセンサ選択及び R 学習の性能改善) と，その実現方法に関連した内容について，考察及び将来研究の検討を行った．本章では，個々の用途及び実現方法を超えた，提案手法の一般的な部分に限定して，考察及び将来研究の検討を行う．

本研究では，強化学習手法を用いることで，ロボットに自発的に望ましい行動を獲得させることを目的とし，この目的のために従来の強化学習の性能を向上する新しい手法を提案した．提案手法では，複数の Q 値表を同時並行的に学習に参加させる，すなわち強化学習エージェントを同時に複数用いて学習する．

一般的に，学習速度及び学習によって獲得される内容は，初期値等の学習条件や，学習の過程により異なると考えられている．このため，条件の異なる複数の強化学習エージェントが同時に存在する場合，各々の学習エージェントで，学習速度や獲得内容に差が生じると予想される．さらに，強化学習は能動的な学習手法であり，学習すべき内容は，学習エージェント自身の行動によって変化する．

本研究では，複数の強化学習エージェントを同時並行的に動作させ，それらの学習速度や獲得内容を比較を通して，より学習の進んだエージェントの特定を可能にする．学習の進んだエージェントを，優先的に行動決定に利用することで，さらに迅速かつ内容の優れた学習が期待できる．

検証実験においては，条件の異なる強化学習エージェントは，各々利用センサ

集合が異なるものとした．実験の結果，提案手法が，通常の強化学習手法と比較して，より少ない回数の環境とのインタラクションで優れたパフォーマンス (平均獲得報酬及び壁との衝突率) を実現可能であることが確認された．

強化学習において，複数の Q 値表を同時並行的に学習に用いるという手法は，[i],[ii] 及び，本論文で初めて提案されたもので，今後の研究により，手法の有効性がさらに向上する可能性がある．提案した手法は，本論文で評価に用いた課題では，大きな効果があることが認められたが，有効性の範囲や程度に関しては，他の課題に適用することで，今後確認が必要である．また，Q 値表を並置して行動を決定するというアイデアに関しても，理論面を含めた今後の研究により，手法の有効性がさらに向上する可能性がある．なお，提案手法は，強化学習の一般的な拡張手法であり，その用途に関しても，本論文であげた具体例に限定されるものではなく，様々な局面で，強化学習の効率化に貢献可能であると考える．

人工知能分野においては，近年アンサンブル学習 (Bagging 手法，Boosting 手法等の総称．各手法に関しては，例えば [11] 参照) の研究成果が多数報告されている．アンサンブル学習では，問題に対する正解率が 50%を多少超える，多数の弱学習器 (weak learner) を並列的に用い，これらの学習器の回答を統合することで，個々の学習器の能力を超えた，高い正解率を達成することを目的とする．本論文で提案した手法とは，アイデアが似ている反面，

(1) 弱学習器の学習は，教師あり学習である

(2) 強化学習課題に適用する場合，行動の決定に向けた (各学習器の) 統合に関して，現段階で明確な指針がない

という点に違いがある．ただし，こうした面に関する考察は，強化学習側からも，アンサンブル学習の側からも行われておらず，今後の研究によって，両者の統合や相補も可能ではないかと考える．とくに，強化学習の側からは，各学習機の性能が低い場合であっても，それらを超えた性能が達成可能であるという点は，非常に魅力がある．

第 6.1.2 及び 6.2.2 節で示した通り，実験 1,2 の結果に共通して，提案手法が，学習を迅速に進める効果をもつことが確認された．この点に関して，複数の Q 値表から，行動決定に用いる Q 値表を選択するための処理 (第 6.1.1 及び 6.2.1 節参照)

が，どの程度貢献しているかを確認するため，全ての Q 値表が，常に同一の確率 ($\frac{1}{255}$) で選択されるという設定の実験も実施した．その結果，通常の Q/Sarsa/R 学習と，提案手法を適用した場合の，中間的な平均獲得報酬が得られた (第 6.1.3.1 及び 6.2.3.1 節参照)．このため，提案手法の適用によって得られる迅速な学習は，

(1) Q 値表を同時に複数用いること

(2) それらを学習内容の適切さに応じて利用すること

両者の，相乗効果によって実現されていると考えられる．

さらに，全ての Q 値表の選択確率が常に同一という設定の実験結果の 1 つを抽出し，各 Q 値表の学習内容を分析した．学習済の各々の Q 値表を，単独で用いて走行させる実験の結果，各 Q 値表の単独走行時の平均獲得報酬と各 Q 値表が獲得した行動の良し悪しが必ずしも一致しないことが判明した．すなわち，いくつかの Q 値表は，その場で旋回を繰返すことで，比較的良好な平均獲得報酬を得ている (この場合，平均獲得報酬は –0.03 に近く，壁に衝突する場合と比較して良い値となる)．このため，平均獲得報酬のみで，望ましいセンサ集合を特定することは難しいと思われる．したがって，センサ選択効果は，学習内容の適切さに応じて各 Q 値表を利用する処理によって得られると判断される．

以上の結果は，単に Q 値表を複数用いるだけで，学習が促進される可能性を示唆しており，非常に興味深い．この点に関しては，上記アンサンブル学習と似た効果が得られたためではないかと推測されるが，詳細は不明であり，そのメカニズム等に関しては，今後実験を重ねて解明する必要がある．また，第 3.3 節で指摘した,「ある強化学習エージェントによる望ましい行動決定を，他の強化学習エージェントが学習していく」という仮説を裏付ける結果であり，分析を進めることで，全く新しい強化学習につながる可能性を秘めている．

さらに，提案手法の適用による，学習促進効果のロバスト性を評価するため，強化学習に用いるパラメータ (実験 1 における ψ, κ 及び実験 2 における ξ, β) の値を変化した場合の実験も実施した．この結果，これらのパラメータの設定に関わらず，学習を迅速に進める効果が広く得られることが確認された (第 6.1.3.2 及び 6.2.3.2 節参照)．また，上述の全ての Q 値表が常に同一の確率で選択されるという設定の実験は，Q 値表の優先度の更新パラメータを 0 に，リファレンス報酬ない

し平均報酬の学習率を任意に設定したことに相当する．これら全てのパラメータ設定で，従来手法より優れた結果が得られたことから，提案手法はロバストに性能を発揮すると考えられる．

学習促進効果という点では，とくにPOMDP課題において，適格度トレースと呼ばれる手法を用いることが有効であるとの指摘がある (例えば [35] 参照．なお，適格度トレース手法の詳細は，第C章にて詳述する)．提案手法と，適格度トレースを用いた手法の学習速度の比較のため，$\lambda = 0.9$ の入替え更新トレース手法 (第C.3節参照) を用いた Q(λ)/Sarsa(λ) 学習の実験を行った．この結果，6,240,000行動後の平均獲得報酬は，実験1と Q(λ)/Sarsa(λ) で，ほぼ同等であった．一方，Q(λ)/Sarsa(λ) の実験では，実験終了までに，提案手法の約10倍の時間を要することが判明した (今回実験を行った計算機環境では，Q(λ)/Sarsa(λ) の1実験に要する時間は，24時間前後であった)．なお，提案手法と適格度トレース手法は，択一的なものではなく，併用も可能である．両者の併用によって，さらに学習速度の向上が実現される可能性があるが，このためには，Q(λ)/Sarsa(λ) の処理時間を短縮する必要があると思われる．

テーブル型強化学習手法では，タイルコーディング法 [35] (いわゆるCMAC) が用いられることもある．タイルコーディング法は，状態空間を複数の (必ずしも全ての次元を使用する訳ではない) タイリングを用いて表現する (ちなみに，通常のQ/Sarsa/R学習は，全次元を用いる1個のタイリングを使用することに相当する)．提案手法の特徴の1つは，タイルコーディング手法とも併用し，複数のタイルコーディングが考えられる時，これらをオンラインで比較しながら，累積報酬が最大となるものの選択を可能とすることである．特に，タイルコーディング時に無視する次元 (状態空間の次元で，センサに対応する) があり得るとき，その選択をオンラインで行うことを可能にする．そのため，タイルコーディング手法と提案手法との組合せの有効性を検証することが必要である．しかし，タイルコーディング手法を適用する場合，タイリングの調整作業が不可欠になると考えられる．また，とくにPOMDP課題において，タイルコーディング手法と，適格度トレースを併用する研究例もみられる．したがって，適格度トレース，タイルコーディング及び提案手法を併用した場合の，有効性確認も重要である．以上の2点に関しては，今回採用した課題，及び適格度トレースと提案手法を併用した実験の状況 (上記参

照) を考慮すると，著しく長い実験時間が必要と予想され，確認は行えなかった．

POMDP 課題に対応するために，従来提案されていたアプローチには，観測した状態の同定に当たって，精度の向上を目指すものもあった．一方，本論文で提案した手法は，ある意味で，これらとは全く逆のアプローチをとっている．すなわち，本研究では，適切な形で部分観測性を強めることは，より望ましい行動を，より迅速に獲得する可能性があることを示した．この点で，本研究で得られた結果は，非常に斬新であると考える．実験に当たっては，センサの冗長性を含むやや特殊な例が対象としていたものの，ロボットの設計や構築に当たって，冗長性のないセンサを搭載することは，実質的には極めて困難であることは明らかであり，本論文のアプローチの一般性を，必ずしも損なうものではないと思われる．

強化学習における重要な問題の一つとして，遅延報酬がある (例えば [41] 参照)．本手法では，行動決定に用いる各 Q 値表は，通常の Q/Sarsa/R 学習と同一の更新式を用いて学習している．また，複数の Q 値表から，実際に行動を決定する Q 値表を選択する処理が，各 Q 値表の報酬の遅れに対する効果を打ち消すことはない (選択されるどの Q 値表も通常の Q/Sarsa/R 学習を行っている)．このため，処理 (すなわちエージェント) 全体としては，通常の Q/Sarsa/R 学習手法同様，報酬の遅れに対応可能であると考えられる．

提案手法の評価において，実ロボットのシミュレータを用いた実験では，即時報酬を与えた (第 5.3 節参照)．一方，著しい報酬遅れが発生する課題における手法の効果を調べるため，実験 1 に関して，ゴール到達時のみ報酬 (+50.0) を与える設定での実験も行った．しかし，実験に用いた環境では，ゴール到達に要する行動数が多過ぎるため，各 Q 値表における，Q/Sarsa 手法を用いた学習自体が十分に進まなかった．この点は，学習が，ゴールへの到達に基づいて主に行われ，ゴール到達が偶然性に強く支配される稀な現象であるため，ゴール到達数がある程度あれば学習が進むが，そうでなければ学習は全く進まないという現象が発生していることが理由であると推察される．提案手法はセンサ選択を行う結果，学習内容の良い強化学習エージェントを選択して，集中的に学習させる手法であるため，逆にいえば，どの強化学習エージェントも学習が進まない場合には，仮に最良の強化学習エージェントを選択可能であっても，学習の進行は遅いと考えられる．したがって，このような状況では，より学習内容の良い強化学習エージェントを優先

的に用いるという，提案手法の有効性の確認も不可能であった．ただし，第 4.2.1 節の実験結果が示すように，より単純な環境であって，ゴール到達時の報酬のみで Q 学習が収束する課題においては，提案手法の有効性は示されており，したがって，原則的には遅延報酬に対応可能と考えてよいと思われる．

以上のように，提案手法の有効性に関しては，他手法との比較や，条件を変えた実験を引き続き行う必要がある．例えば，提案手法と，タイルコーディング手法を併用した Q(λ) / Sarsa(λ) との学習性能の比較や，これらと併用した場合の有効性についての確認等が考えられる．

第 3 章の議論で明らかなように，提案手法は，複数の強化学習エージェントのうち，学習な内容が最も良好な，すなわち，最も望ましい形で課題を達成可能なものを選択する手法である．このため，優位な強化学習エージェントを，ロボットの状態に応じて随時選択するのではなく，観測される全状態を通じた総合点で選択しようという目的で設計されている．したがって，例えば，最適なセンサ集合の選択を目的とした場合 (第 3.2 節参照)，置かれた状態に拠らず適切な行動を決定可能なセンサ集合が特定されるよう，状態遷移のない，n 本腕バンディット問題としての定式化を行った (第 6.1.1 節参照)．また，R 学習の性能改善課題 (第 3.3 節参照) においても，学習の進んだ強化学習エージェントの選択確率が増加する (第 6.2.1 節参照) ことで，最終的には，特定の強化学習エージェントが行動を決定することになると推察される．

しかし，より複雑な課題においては，置かれた状態に応じて，複数の強化学習エージェントを使い分ける手法が有効である可能性もあり，提案手法の拡張として興味深いと考える．この場合，強化学習エージェントを 1 つに絞ってしまうのではなく，いくつかの強化学習エージェントを保持しながら，状況に応じて，より良い行動が決定できる強化学習エージェントを得る目的で，状態を反映させた優先度更新を行うことも考えられる．この場合，優先度は状態の関数となる．こうした拡張は，第 7.1.2 節で述べた (とくに Q 値表を複数用いる) 階層型強化学習手法との関連も深いと思われる．また，状況に応じて学習するという点では，アンサンブル学習との共通性も感じられる．したがって，こうした手法も視野に入れつつ，検討を行うことが望ましいと予想される．

8.2 まとめ

本研究では，日常生活環境で動作可能な，知的なロボットの構築を究極の目的の一つとし，その目的の達成には学習によって，ロボット自らに望ましい行動を獲得させることが有望であると考えた．どういう行動が望ましいかを，ロボットに予備知識として全て与えることは困難であり，また動的な環境においては与えた知識の陳腐化という問題も発生することから，ロボットは置かれた環境とのインタラクションの中で，行動の結果に基づき，何が望ましい行動であるかを探り出していくことが必要となる．このような問題設定において学習を行う手法として，強化学習がある．

強化学習は，学習のための教師情報や事前知識を必要とせず，望ましい行動をロボットに自律的に獲得させることを可能にする手法である．強化学習手法では，最適化の手続きと併せて，最適化のための探索過程が提供される．とくに，実際的な問題において，その複雑性のため，解析的な最適化手法が適用できない (ないし適用が現実的でない) 場合に，有望視されている手法の一つである．

強化学習手法を適用することで，

(1) どういう行動が望ましいかを予め明確化する必要がなく，

(2) ロボット自身が，学習すべき内容を能動的に決定し，

学習を進めることが可能となる．手法の理論的背景には，マルコフ決定過程 (MDP) において，Bellman 方程式の解の近似を逐時的に高めていくことで，方策の改善を行う手続きがある．

本研究では，従来の強化学習手法の拡張として，複数の Q 値表を用いて学習を行う，すなわち複数の強化学習エージェントを同時並行的に用いる新しい手法を提案し，効率的に学習を進める効果があることを示した．

複数 Q 値表の具体的な適用例として，実験により有効性を示したのは，次の 2 点である．

(1) 学習に用いるセンサの，オンラインで (すなわち学習を進めながら) の選択

(2) R 学習における局所解問題の解消

第 1 のオンラインセンサ選択効果に関しては，ロボットの行動決定において，より多くのセンサを用いることは，状況特定の精密化，ノイズの影響の軽減，故障時の予防等の意味で有益であるが，製造コストの上昇を招くおそれがある．また，望ましい行動を学習によって獲得させる際には，過多なセンサの搭載によってノイズの影響が重畳し，かえって行動の学習が遅れることが危惧される．さらに，強化学習においては，多くのセンサを用いて学習させる際，状態数の爆発 (いわゆる Bellman の次元の呪い) が効率的な学習を阻害することが知られている．この点に関して，従来，状態分割を逐時的に行うことにより，適切な状態空間を形成する手法が提案されていた [7, 39, 2, 15, 30].

本研究では，強化学習課題において，冗長性のあるセンサ群が与えられた際に，どのセンサを学習に利用するかを，望ましい行動と同時に強化学習させることにより，適切な状態空間を決定する手法を提案した．冗長なセンサ群が提供される強化学習課題の例として，ロボットの障害物回避行動の獲得を取り上げ，手法の有効性をシミュレーション実験により検証した．実験に当たっては，手法及び実験結果を，実ロボットに応用することに十分配慮し，実験環境設定等を決定した．実験の結果，衝突行動の回避と平均獲得報酬の面で，本手法が有効であることが確認された．

本手法の最大の特長は，行動決定のための基準 (Q 値表，すなわち強化学習エージェント) を複数もち，学習によって行動の適切性を高めながら，最も適切な行動を決定可能な (すなわち学習内容が最も妥当な) 学習エージェントを，行動結果に基づいて選択することを可能にすることにある．こうしたアプローチは，センサ群の選択以外の目的にも利用可能な汎用的手法であると考えられ，従来以上に効率的な強化学習の実現が期待される．

第 2 の R 学習の効率化に関しては，とくに実際的で複雑な課題に強化学習を適用する際，学習速度 (本論文では，より少ない環境との相互作用で，パフォーマンスが向上することととらえた) の点が重要な問題となる．学習高速化に関しては，様々な試みがなされているが，本研究では，R 学習を用いることで，迅速な学習の実現を目指すと共に，強化学習エージェントを並列的に複数用いる (すなわち複数の Q 値表を用いる) ことで，R 学習の欠点を補う新しい手法を提案した．

R 学習は，強化学習で通常用いられる，割引を用いた累積報酬の代わりに，平

均報酬を用いる手法で，エピソード分割のない (行動を無限に継続する) 課題に適用される [35]．このため，例えばロボットを継続的に行動させながら，望ましい行動を獲得させるといった課題に適している．R 学習は，Q 学習等と比較して，学習が迅速な可能性がある [31] 反面，行動決定手法や学習パラメータに敏感であるとの報告もある [22, 23]．このため，提案手法の適用により，R 学習の敏感性を解消し，ロバストな結果を得ることが可能になれば，手法の適用の幅が広がる．

エピソードが明確に分割されていない強化学習課題の例として，ロボットの障害物回避行動の獲得を実験課題として取り上げ，手法の有効性をシミュレーション実験により検証した．その結果，従来の Q 及び Sarsa 学習，及び (第 1 実験で採用した) オンラインセンサ選択手法を適用した Q 及び Sarsa 学習と比較して，きわめて良い成績が確認された．提案手法の適用により，少ない行動回数で衝突の回避を学習し，高い平均獲得報酬が得られる．このため，R 学習のロバスト性を向上させ，学習の高速化を図る手法として，有望であると考えられる．

以上のように，本論文では，具体的な用途を 2 つ示し，実験によって有効性を確認することで，複数の強化学習エージェント (Q 値表) を同時に並列的に学習に用いるという，従来十分検討されていなかったアイデアに基づく，新しい手法の利点を明らかにした．しかし，複数の学習エージェントを学習及び行動決定に用いる手法は，これらの用途に限定されるものではない．理論面での裏付けを進め，実験面での確認を行うことで，手法の用途はさらに広がり，有効性は向上すると思われる．

第 A 章

対照実験の処理詳細

第 4，5 及び 6 章において，対照実験で用いた処理のアルゴリズムを，以下に示す。

A.1　Q/Sarsa 学習 (従来手法)

付図 A.1 参照.

A.2　R 学習 (従来手法)

付図 A.2 参照.

A.3　CMAC 手法

付図 A.3 参照.

```
 1 :  N ← number of actions avairable to the robot
 2 :  Q(s, a) ← initial value,  ∀ s, a
 3 :  t, total ← 0
 4 :  s ← initial state
 5 :  while  t < planned transision times do
 6 :    t ← t + 1
 7 :    n ← number of a,  a = argmax_a Q(s, a)
 8 :    for a ∈ A
 9 :      if  a = argmax_a Q(s, a)  then
10 :        π(a) ← (1−ε)/n + ε/N
11 :      else
12 :        π(a) ← ε/N
13 :      end if
14 :    end for
15 :    choose action â according to π(·), and execute â
16 :    get reward r, and observe state s′
17 :    total ← total + r
        ------------------------------------------------
18-1:   Q(s, â) ← Q(s, â) + α [r + γ max_a′ Q(s′, a′) − Q(s, â)]
        ------------------------------------------------
18-2:   Q(s, â) ← Q(s, â) + α [r + γ Q(s′, a′) − Q(s, â)]
        ------------------------------------------------
19 :    s ← s′
20 :  end while
21 :  return  total
```

図 A.1: 対照実験の処理．18-1 は従来手法の Q 学習 (第 4.2.1，6.1.2 及び 6.2.2.5 節)．18-2 は従来手法の Sarsa 学習 (第 6.1.2 及び 6.2.2.5 節)．探索手法として ϵ-greedy を用いる場合．

1 : $N \leftarrow$ number of actions avairable to the robot
2 : $Q(s,a) \leftarrow$ initial value, $N_f(s,a) \leftarrow 1,\ \forall\, s,\, a$
3 : $t, total \leftarrow 0$
4 : $s \leftarrow$ initial state
5 : **while** $t <$ planned transision times **do**
6 : $t \leftarrow t+1$
7 : $n \leftarrow$ number of $a,\ a = \mathrm{argmax}_a\, Q(s,a)$
8 : **for** $a \in A$

9-1-1: **if** $a = \mathrm{argmax}_a\, R(s,a)$ **then**
9-1-2: $\pi(a) \leftarrow \frac{1-\epsilon}{n} + \frac{\epsilon}{N}$
9-1-3: **else**
9-1-4: $\pi(a) \leftarrow \frac{\epsilon}{N}$
9-1-5: **end if**

9-2-1: **if** $a = \mathrm{argmax}_a\, R(s,a) + \frac{c}{N_f(s,a)}$ **then**
9-2-2: $\pi(a) \leftarrow p + \frac{1-p}{N}$
9-2-3: **else**
9-2-4: $\pi(a) \leftarrow \frac{1-p}{N}$
9-2-5: **end if**

9-3 : $\pi(a) \leftarrow \dfrac{e^a}{\sum_A e^A}$

---

10 : **end for**
11 : choose action $\hat{a}$ according to $\pi(\cdot)$, and execute $\hat{a}$
12 : $N_f(s,a) \leftarrow N_f(s,a) + 1$
13 : get reward r, and observe state s'
14 : $total \leftarrow total + r$
15 : $Q(s,\hat{a}) \leftarrow Q(s,\hat{a}) + \alpha\,[r - \rho + \max_{a'} Q(s',a') - Q(s,\hat{a})]$
16 : **if** $Q(s,\hat{a}) = \max_a Q(s,a)$ **then**
17 : $\rho \leftarrow\ \rho + \beta\,[r - \rho + \max_{a'} Q(s',a') - Q(s,\hat{a})]$
18 : **end if**
19 : $s \leftarrow s'$
20 : **end while**
21 : **return** *total*

図 A.2: 対照実験の処理．従来手法の R 学習．探索手法として，9-1 は ϵ-greedy (第 6.2.2.2 節)，9-2 は UE (第 6.2.2.3 節)，9-3 は softmax (第 6.2.2.4 節) を用いた場合．

```
1    :  N ← number of actions avairable to the robot
2    :  C ← number of elements (tilings) in I
3    :  U_i(s,a) ← initial value,  ∀ i ∈ I, s, a
4    :  t, total ← 0
5    :  s ← initial state
6    :  while  t < planned transision times do
7    :    t ← t + 1
8    :    for a ∈ A
9    :      U(s,a) ← Σ_{i∈I} U_i(s,a)
10   :    end for
11   :    n ← number of a,  a = argmax_a U(s,a)
12   :    for a ∈ A
13   :      if  a = argmax_a U(s,a)  then
14   :        π(a) ← (1−ε)/n + ε/N
15   :      else
16   :        π(a) ← ε/N
17   :      end if
19   :    end for
18   :    choose action â according to π(·), and execute â
19   :    get reward r, and observe state s'
20   :    total ← total + r
21   :    for i ∈ I
     ------------------------------------------------------------
22-1-1:      U_i(s,â) ← U_i(s,â) + (1/C) α [r − ρ + max_{a'} U(s',a') − U(s,â)]
22-1-2:      if  U(s,â) = max_a U(s,a)  then
22-1-3:        ρ ←  ρ + β [r − ρ + max_{a'} U(s',a') − U(s,â)]
22-1-4:      end if
     ------------------------------------------------------------
22-2 :       U_i(s,â) ← U_i(s,â) + (1/C) α [r + γ max_{a'} U(s',a') − U(s,â)]
     ------------------------------------------------------------
23   :    end for
24   :    s ← s'
25   :  end while
26   :  return  total
```

図 A.3: 対照実験の処理．22-1 は off-policy 型 R 学習による CMAC (第 6.2.2.6 節)．22-2 は Q 学習による CMAC (第 6.2.2.7 節)．探索手法として ϵ-greedy を用いる場合．なお，U は状態行動価値を示す．

第 B 章

実験1の結果の詳細分析

第 6.1.2 節で結果報告を行った 20 実験 (Q 学習 10 実験，Sarsa 学習 10 実験) のうち，典型的な結果 3 例に関して，実験の詳細な推移を以下に掲げる．なお，以下の実験番号に関しては，表 6.1 参照．また，各図は，上段が実験開始時からの平均獲得報酬 (縦軸:全平均獲得報酬と呼ぶ) を 10,000 行動毎にプロットしたもの，中段が過去 10,000 行動の平均獲得報酬 (縦軸:直前平均獲得報酬と呼ぶ) を 10,000 行動毎にプロットしたもの，下段が過去 10,000 行動中特定 (以下参照) のセンサ集合が選択された回数 (縦軸, log 表示) である．さらに，横軸は全て行動回数 (単位:10,000 行動) で，各プロット間は直線補間している．

B.1 実験 19

実験途中に適切なセンサ集合を獲得し，実験終了時までそれを維持した実験結果．実験終了時に，選択確率 1.0 であったセンサ集合を用い，$\epsilon = 0$ にて走行させた結果も良好であった．

図 B.1 上段は，実験 19 の結果 (実線) の他，比較のため全センサを利用して従来手法の Sarsa 学習を行った 10 実験の平均 (破線) をプロットした．また，下段の図に関しては，実験終了時選択確率 1.0 であったセンサ集合に関して作成した．

3,000,000 行動前後で，このセンサ集合が選択される確率がほとんど 1 になる (下段) とともに，直前平均獲得報酬が高水準 (正値) で安定し (中段)，したがって実験開始時からの全平均獲得報酬も延びている (上段)．

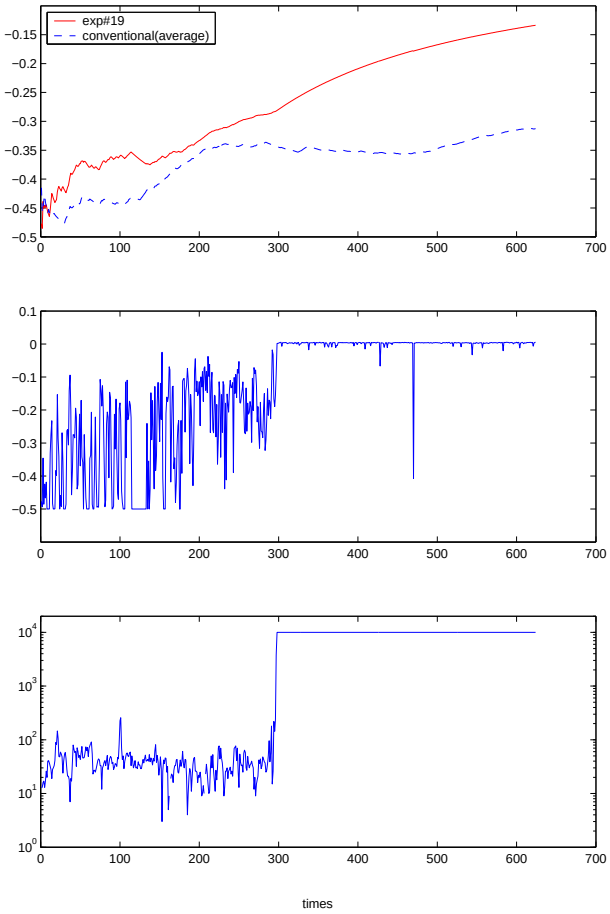

図 B.1: 実験 19 の詳細推移

B.2 実験 3

実験途中に適切なセンサ集合を獲得したが，実験終了時までは維持できなかった実験結果．実験開始時からの平均獲得報酬が最大であった際のセンサ集合を用い，$\epsilon = 0$ にて走行させた結果も良好であった．

図 B.2 上段は，実験 3 の結果 (実線) の他，比較のため全センサを利用して従来手法の Q 学習を行った 10 実験の平均 (破線) をプロットした．また，下段の図に関しては，実験開始時からの平均獲得報酬が最大であった際のセンサ集合に関して作成した．

1,000,000 行動前後で，このセンサ集合が選択される確率がほとんど 1 になる (下段) とともに，直前平均獲得報酬が高水準 (正値) で安定し (中段)，全平均獲得報酬も延びている (上段)．しかし，2,000,000 行動を超えた時点で，このセンサ集合の

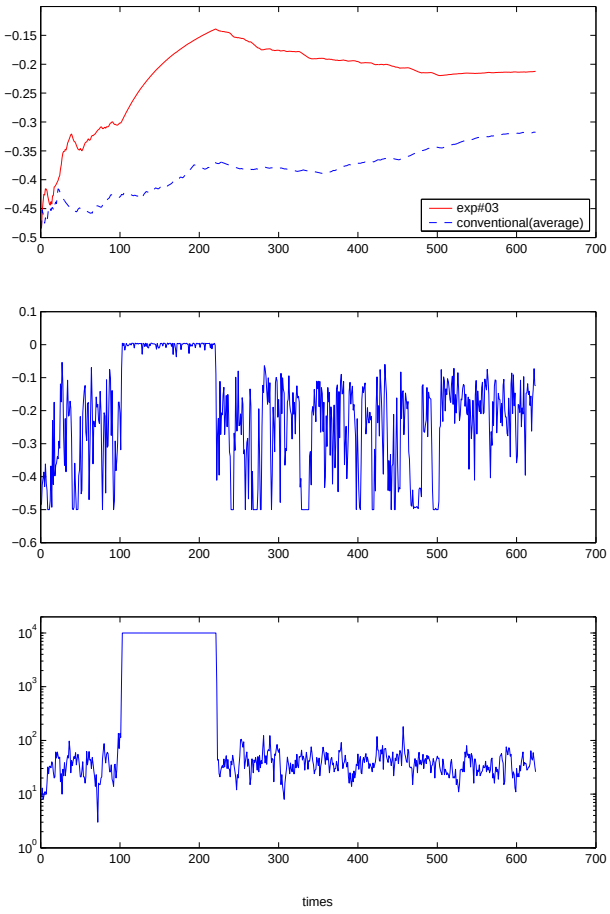

図 B.2: 実験 3 の詳細推移

選択回数が減少し，直前平均獲得報酬も低下して負となり，全平均獲得報酬も低下が見られる．終了前には，直前平均獲得報酬も徐々に改善し，全平均獲得報酬の低下が止まり始めている．

B.3　実験 15

適切なセンサ集合を獲得することなしに，実験終了を迎えた実験結果．図 B.3 上段は，実験 15 の結果 (実線) の他，比較のため全センサを利用して従来手法の Sarsa 学習を行った 10 実験の平均 (破線) をプロットした．また，下段の図に関しては，実験終了時の選択確率が最大であったセンサ集合に関して作成した．

上記 2 実験と異なり，選択回数や直前平均獲得報酬の急激な上昇はない．しか

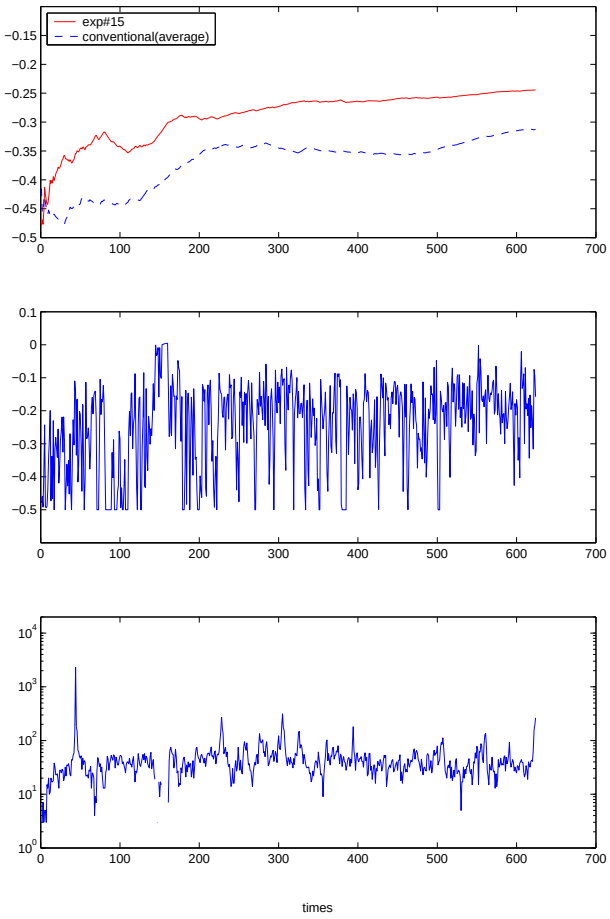

図 B.3: 実験 15 の詳細推移

し，直前平均獲得報酬が下振れすることが減少し，上段の図の全平均獲得報酬の向上に繋がっていると予想される．

第 C 章

適格度トレース

第 2.3.1 節では，1 ステップ分の時間的差分 (TD) を用いる強化学習について述べた．ここでは，同時に扱う時間的差分を n に拡張した (すなわち，1 ステップを n ステップに拡張した) 手法について説明する．この拡張のためには，n ステップにわたる行動の結果を基に Q 値の更新を実現するための，いわば記憶領域が必要となる．こうした仕組みを実現するために，通常，適格度トレースと呼ばれる変数群を用いることから，n ステップ TD 法は，適格度トレース (eligibility trace) を適用した手法と呼ばれる．

なお，第 C.2.0.1 節で説明するように，適格度トレースでは，λ というパラメータによって，何ステップ分の情報を利用するかを規定する (すなわち，目的とする n に合わせて，λ の値を設定する)．このため，適格度トレース手法を適用した TD 学習は TD(λ) と呼ばれることが多い．したがって，適格度トレースを用いた Q 学習，Sarsa 学習は，各々 Q(λ), Sarsa(λ) と呼ばれる．

適格度トレース手法は，訪問された状態と選択した行動の履歴を管理することで，Q 値更新における伝播速度の向上を図る手法であるといえる．とくに，長期遅延報酬 (long-delayed rewards) 課題や，非マルコフ課題においては，適格度トレース手法を用いることが有効である．

適格度トレース手法の実現する方法は，いくつか提案されているが，主に累積トレースと入替え更新トレースに分類される．各手法に共通する，適格度トレース手法を用いた際の Q 値の更新方法に触れた後，各々の説明を行う．

C.1 Q値更新

適格度トレース手法を適用した際，時刻tのQ値の更新は，全てのs, aに対して

$$Q_{t+1}(s,a) \;=\; Q_t(s,a) + \alpha\,\delta_t\,e_t(s,a) \tag{C.1}$$

によって行われる．ここで，$e_t(s,a)$が，(状態行動価値に対する) 適格度トレースである (詳細は，次節以降参照)．また，δ_tは，時刻tにおけるQ(0)/Sarsa(0)のδ値を指し，具体的には，

Q学習の場合: $\quad \delta_t = r + \gamma\,\max_{a_{t+1}} Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)$

Sarsa学習の場合: $\quad \delta_t = r + \gamma\,Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)$

である．

C.2 累積更新トレース

累積更新トレース (accumulating trace) 手法では，時刻tの適格度トレースの更新は，Q学習とSarsa学習で異なった更新式を用いる．具体的には，全てのs, aに対して，次節以下の式を適用することで行われる．

C.2.0.1 Q学習の場合

Q学習は，方策オフ型手法であり，ある方策にしたがって行動しながら，最適方策を学習する (第2.3.2節参照)．このため，適格度トレースの計算に関しては，配慮を要し，いくつかの方法が考えられる．ここでは，Watkinsの$Q(\lambda)$と呼ばれる手法を紹介する [35].

Watkinsの$Q(\lambda)$では，適格度トレースの更新式は，以下の通りである．

$$e_t(s,a) = \mathcal{I}_{ss_t} \cdot \mathcal{I}_{aa_t} + \begin{cases} \gamma\,\lambda\,e_{t-1}(s,a) & (Q_{t-1}(s_t,a_t) = \max_a Q_{t-1}(s_t,a)\ \text{のとき}) \\ 0 & (\text{それ以外のとき}) \end{cases}$$

ここで，$\mathcal{I}_{xy}$は，一致関数 (identity-indicator function) で，$x = y$ならばその値は1で，それ以外であれば0である．また，γは割引率 (第2.2.6節参照)，λ $(0 \le \lambda \le 1)$

は，指数関数的重み付けが減少する速度 (すなわち，どれだけのステップを考慮に入れるか) を決定するパラメータで，トレース減衰パラメータ (trace-decay parameter) と呼ばれる (これらの記法は，次節以下でも用いる)．$\lambda = 0$ の場合，式 C.1 が，通常の Q 学習における Q 値表更新式と等しくなることは明らかである．一方，$\lambda = 1$ の場合，エピソード終端までの報酬が考慮の対象となる．この際，TD 学習手法は，MC 法と等価となる (第 D.2 節も参照のこと)．

すなわち，Watkins の Q(λ) では，現在の状態と行動に対応するトレースは 1 だけ増加する．一方，過去の全ての状態行動対のトレースは，$\gamma\lambda$ の割合で減衰するか，あるいは探索的行動 (グリーディではない行動) がとられた場合，0 に設定される．

C.2.0.2 Sarsa 学習の場合

Sarsa 学習における適格度トレースの更新式は，以下の通りである．

$$e_t(s,a) = \begin{cases} \gamma\lambda e_{t-1}(s,a) + 1 & (s = s_t \text{ かつ } a = a_t \text{ のとき}) \\ \gamma\lambda e_{t-1}(s,a) & (\text{それ以外のとき}) \end{cases} \tag{C.2}$$

すなわち，現在の状態と行動に対応するトレースは 1 だけ増加する．一方，過去の全ての状態行動対のトレースは，$\gamma\lambda$ の割合で減衰する．

C.3　入替え更新トレース

入替え更新トレース (replacing trace) の実現に関しては，いくつかの方法が考えられるが，[33] で推奨されているアプローチでは，適格度トレースは以下の式で更新される．

$$e_t(s,a) = \begin{cases} 1 & (s = s_t \text{ かつ } a = a_t \text{ のとき}) \\ 0 & (s = s_t \text{ かつ } a \neq a_t \text{ のとき}) \\ \gamma\,\lambda\,e_{t-1}(s,a) & (s \neq s_t \text{ のとき}) \end{cases} \tag{C.3}$$

すなわち，ある状態が再訪問され，行動が選択された際，その行動に対するトレースは 1 に再設定されるが，それ以外の全ての行動に関しては，再訪問状態から 0 に設定される点が，累積更新トレースと異なっている．用いる適格度トレース手法を，累積トレースから入替え更新トレースに変更することで，学習速度が大きく改善される課題があることが確認されている [35].

第 D 章

MDP 問題に対する解法の比較検討

本章では，MDP 問題に対する代表的な解法について，それぞれの比較を行う．なお，本章の内容は，主に [35] によっている．

D.1 動的計画法

本節では，有限 MDP 問題の解法の 1 つである，動的計画法 (dynamic programming; DP) について概要を記述する．以下 DP 手法で，強化学習手法のうち，動的計画法の直接の適用である方策反復 (policy iteration) 手法と価値反復 (value iteration) 手法を指す．

DP 手法は，数学的に十分確立された方法であるが，環境のモデルは完全かつ正確でなければならない．方策反復及び価値反復では，与えられた方策に関する価値関数を求める方策評価 (policy evaluation) と，方策に対する価値関数を基に改善された新しい方策を求める方策改善 (policy improvement) と呼ばれる手続きから構成される [35, 28].

D.1.1 方策評価

方策 π のもとでの状態 s の価値 $V^\pi(s)$ は,

$$V^\pi(s) = E_\pi\{R_t|s_t = s\} = E_\pi\left\{\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t = s\right\}$$

と表され，関数 V^π を方策 π に対する状態価値関数 (state-value function) と呼ぶ．ここで，$E_\pi\{\,\}$ は，方策 π に従った場合の期待値をである．

DP 手法 (ないし強化学習) で扱う価値関数は，特定の再帰的関係を満たすという基本的性質をもっており，以下の整合性条件 (consistency condition) が成り立つ..

$$\begin{aligned} V^\pi(s) &= E_\pi\left\{ r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} | s_t = s \right\} \\ &= \sum_{a\in\mathcal{A}(s)} \pi(s,a) \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} | s_{t+1} = s' \right\} \right] \\ &= \sum_{a\in\mathcal{A}(s)} \pi(s,a) \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma V^\pi(s') \right] \end{aligned} \tag{D.1}$$

ここで，式 D.1 は，V^π に対する Bellman 方程式 (Bellman equation) である．なお，$\mathcal{P}$ は遷移確率 (transition probability) と呼ばれ，任意の s, a が与えられた場合，次に可能な各状態 s' の確率は，$\mathcal{P}^a_{ss'} = \Pr\{s_{t+1} = s' | s_t = s, a_t = a\}$ である．また，$\mathcal{R}$ は次の報酬の期待値を示し，$\mathcal{R}^a_{ss'} = E\{r_{t+1} | s_t = s, a_t = a, s_{t+1} = s'\}$ である．

次に，任意の方策 π に対する状態価値関数 V^π を反復的に求めることを考える．近似価値関数列 $V_0, V_1, V_2, \cdots$ について，式 D.1 を更新規則として適用し，連続した近似

$$\begin{aligned} V_{k+1}(s) &= E_\pi \left\{ r_{t+1} + \gamma V_k(s_{t+1}) | s_t = s \right\} \\ &= \sum_a \pi(s,a) \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma V_k(s') \right] \end{aligned}$$

を得る．V^π に対する Bellman 方程式から，この更新式の固定点は $V_k = V^\pi$ となる．このアルゴリズムは反復方策評価 (iterative policy evaluation) と呼ばれる．

反復方策評価では，この操作を各状態 s に適用して，V_k から V_{k+1} の近似を得る．すなわち，s の推定価値と即時報酬の期待値をもとに，s の価値を新たに近似する操作を，現在評価している方策下で可能な 1 ステップ遷移の全てに対して行う．1 回の更新処理で複数の価値が更新されることから，DP 手法は一般に synchronous(第 2.3.1 節参照) なアルゴリズムである (この点に関しては，第 D.3 節でも検討する)．

Sutton は，ある推定価値を (部分的に) 基にして別の価値を推定する，すなわち推測から推測を学習するような処理を，ブートストラップと呼んでいる．また，価値の更新処理自体をバックアップと呼び，DP 手法が可能な全ての遷移を考慮することから，DP 手法を完全バックアップの手法と位置付けている [35]．なお，第 D.3 節も参照のこと．

D.1.2 方策改善

方策改善とは，状態 s で現在の方策に従った場合の価値 $V^\pi(s)$ が与えられた際，現在の方策を維持すべきか，あるいは新しい方策への切替えを行うべきかの判断を意味する．判断方法の 1 つとして，状態 s で行動 $a \neq \pi(s)$ を選択し，その後は既存の方策に従うことで，推定価値がどのように変わるかを確認することが考えられる．すなわち，

$$\begin{aligned} Q^\pi(s,a) &= E_\pi\{r_{t+1} + \gamma V^\pi(s_{t+1}) | s_t = s, a_t = a\} \\ &= \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma V^\pi(s')\right] \end{aligned}$$

と $V^\pi(s)$ との値の比較に基づく判断であり，s で a の行動を選択することが，現在の方策に従った場合より価値が高い場合，方策は変更によって改善することが期待される．

D.1.3 方策反復

最適方策の発見のため，方策評価と方策改善を交互に繰り返す手法は，方策反復と呼ばれる．方策反復の過程は，以下のような系列としてとらえられる．

$$\pi_0 \xrightarrow{E} V^{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} V^{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \cdots \xrightarrow{I} \pi^* \xrightarrow{E} V^*$$

ここで，$\xrightarrow{E}$ は方策評価，$\xrightarrow{I}$ は方策改善を示す．

この手法では，V^π を用いて π を改善し，より優れた π' を得ることができれば，その π' に基づいた $V^{\pi'}$ を用いて，さらに優れた π'' を得ることを図る．有限 MDP の方策は有限個であるため，この過程は有限回の繰り返しで最適価値関数に収束する．

D.1.4 価値反復

前節で述べた方策反復は，方策の更新毎に方策評価を行う点で，必ずしも効率が良くない．価値反復では，各状態で 1 回の方策評価を行った後，評価を打ち切る．

価値反復は以下のように記述され,

$$\begin{aligned} V_{k+1}(s) &= \max_a E\{r_{t+1} + \gamma V_k(s_{t+1}) | s_t = s, a_t = a\} \\ &= \max_a \sum_{s'} \mathcal{P}^a_{ss'} \left[\mathcal{R}^a_{ss'} + \gamma V_k(s')\right] \end{aligned}$$

全ての行動に対する最大値を用いる点に特徴がある.

D.1.5 DP 手法の有効性

DP 手法を適用した場合，最悪でも状態数と行動数の多項式時間で最適方策が得られるとされ，方策空間を直接探索するどのような手法よりも指数関数的に早い. MDP を解く方法として，線形計画法 (liniar programming; LP) を用いることも可能であり (例えば [28] 参照.)，最悪の場合の収束の保証が DP 手法より優れる場合もある．しかし，LP 手法は DP 手法よりはるかに小さな状態数 (DP 手法の $\frac{1}{100}$ 程度) で非実用的になってしまうため，大規模問題群には適用できないとされる [35]. 一方，Puterman は，近年の LP アルゴリズムの革新が，こうした事態を変える可能性もあるのではないかとの見方も示している [28].

D.2 モンテカルロ法

モンテカルロ (Monte Carlo; MC) 法は，ゴールをもつ (すなわちエピソード的)MDP 問題に適用される．DP 手法のように，全ての遷移の完全な確率分布は必要ではなく，1 つ (ないし複数) の経験，すなわち (ゴールに達するまでの) 遷移のサンプルに基づいて学習を行う．一般には，ある確率分布に従って経験を生成することは容易であるが，分布自体を明示的に得ることはそれと比較して難しい.

MC 手法では，価値の推定と方策の変更は，エピソード完了後に限って実施される点が特徴である．また，学習はサンプル収益の平均化に基づいて実現される.

MC 手法で主に用いられるアルゴリズムとして，初回訪問 (first visit)MC 手法と逐一訪問 (every-visit)MC 手法がある．エピソード中で状態 s が発生することは，訪問と呼ばれる．初回訪問 MC 手法は，s への初回訪問の結果発生した収益を平均化する．一方，逐一訪問 MC 手法では，経験したエピソード群において，s への訪

問全ての結果発生した収益の平均値として $V^{\pi}(s)$ を推定する．また，MC 手法でも，TD 手法同様，方策オン型とオフ型の学習手法が提案されている．

D.3　統一された見方と手法比較

本節では，TD 手法・DP 手法・MC 手法を，統一された見方の中に位置付け，それらの比較を行う．

第 2.3.1 節で紹介したように，1 ステップ TD 手法では，ある時点と次の時点との効用の差のみ (すなわち，効用の差を 1 つだけ) に注目する．また，Q/Sarsa/R 学習では，通常，行動の結果観測された状態に関する価値推定を更新する，すなわち asynchronus な推定更新を行う点に特徴がある．

Sutton は，TD 手法は MC 手法と DP 手法の考え方を組み合わせたものであると位置付ける．TD 手法は，MC 手法と同様，環境のダイナミクスのモデルを用いずに，経験から直接学習することができ (すなわち経験した遷移をもとにバックアップする)，DP 手法と同様，最終結果を待たずに，他の推定値の学習結果を一部利用し推定値を更新する (すなわちブートストラップを行う)．

これら 3 手法は，バックアップの深さ (1 行動後推定値の更新を行うか，エピソードの終了まで更新を行わないか)，及びバックアップの広さ (ある状態から遷移可能な全状態のバックアップを行うか，特定の状態に対してバックアップするか) という特徴軸で，図 D.1 のように分類される．そして，これらの手法の中間的な領域に位置する手法も提案されている．例えば適格度トレース (第 C 章参照) を用いる TD 手法が，こうした中間的な手法に相当する．

DP 手法に対する TD 手法の明らかな利点の 1 つは，TD 手法が環境のモデル，つまり報酬と次の状態の確率分布を必要としない点である．DP 手法をもとにした古典的アプローチでは，各状態 (あるいは状態行動対) のバックアップが，状態 (あるいは状態行動対) 空間全体に対して発生する．規模の大きなタスクにおいては，オンラインで完全なバックアップを完了する時間すらないであろうから，この方法は問題が多い．

多くのタスクにおいて，ほとんど大多数の状態は，非常に貧弱な方策のもとで，あるいは非常に低い確率でのみ訪問されるので，状態間の関連性がない．枚挙型

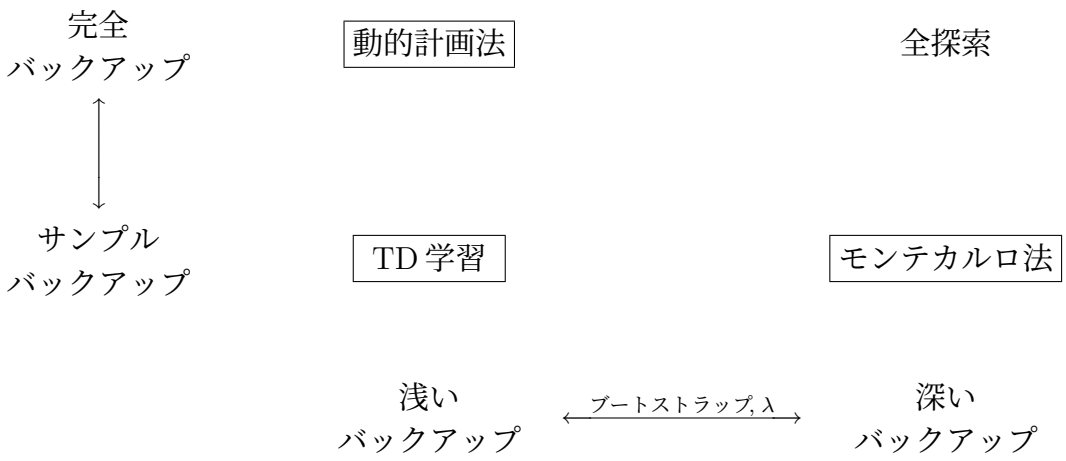

図 D.1: MDP 問題の解法の統一化された見方．動的計画法 (DP)，モンテカルロ法 (MC) 及び TD 手法は，2 つの軸によって特徴付けが可能である．本図は，[35] 図 10.1 を基に作成した．

の処理は，必要とされるところに対象集中する (focusing) のではなく，状態空間全体に対して同等に計算リソース (時間) を割り当てる．枚挙型の処理を行うことと，全ての状態を同等に扱うことは，必ずしも DP 手法に必要とされる性質ではないが，実際上は枚挙型の処理が用いられることが多い．

Sutton は，ある計算リソース (時間) が与えられた場合に，少数の状態行動対における完全バックアップと，より多い状態行動対におけるサンプル・バックアップという 2 つの条件で，学習効率を比較している．この結果から，正確に解くには状態数が多過ぎる問題に対しては，サンプル・バックアップが完全バックアップ比べ優れていると主張する [35].

謝　辞

主指導教官の藤波努助教授，所属講座の國藤進教授には，日頃から様々な面で有益な御指導・御助言をいただきました．この場をお借りして，御礼申し上げます．また，審査をいただいた，江尻正員先生，知識科学研究科の吉田武稔教授，林幸雄助教授には，大変貴重なご指摘をいただきました．有難うございました．さらに，在学中お世話になりました，北陸先端科学技術大学院大学の諸先生方にも，御礼申し上げます．本研究を進めるにあたり，終始御指導をいただきました，慶應義塾大学理工学部の櫻井彰人教授に，心より感謝の意を表します．

参考文献

[1] J. S. Albus. *Brains, Behaviour, and Robotics*. BYTE Books Subsidiary of McGraw-Hill, 1981.

[2] M. Asada, S. Noda, and K. Hosoda. Action-based sensor space categorization for robot learning. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems 1996*, pp. 1502–1509, 1996.

[3] J. A. Bagnell, K. L. Doty, and A. A. Arroyo. Comparison of reinforcement learning techniques for autonomous behavior programming. In *Conference on Automated Learning and Discovery*, 1998.

[4] A. G. Barto and S. Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete Event Systems Journal*, 13(Special Issue on Reinforcement Learning):41–77, 2003. .

[5] R. Bellman. *Adaptive Control Processes: a Guided Tour*. Princeton University Press, Princeton, N.J., 1961.

[6] R. A. Brooks. New approaches to robotics. *Science*, 253:1227–1232, September 1991.

[7] D. Chapman and L. P. Kaelbling. Input generalization in delayed reinforcement learning: An algorithm and performance comparisons. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI91)*, pp. 726–731, 1991.

[8] C. Chow and J. Tsitsiklis. The complexity of dynamic programming. *Journal of Complexity*, 5:466–488, 1989.

[9] Cyberbotics. *Webots 2.0 User Guide*, 1999.

[10] Cyberbotics. *Webots 2.0 Refernce Manual*, 1999.

[11] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, second edition, 2001. (邦訳 : パターン識別, 尾上 監訳, 新技術コミュニケーションズ, (2001)).

[12] F. Even-Dar and Y. Mansour. Learning rates for Q-learning. In *The Fourteenth Annual Conference on Computational Learning Theory*, 2001.

[13] 深尾, 大村, 足立. Q-learning における適応空間の適応的分割法. 計測自動制御学会論文集, 37(3):242–249, 2001.

[14] 原田, 今井. まず「トーイ」より始めよ. 日経エレクトロニクス, No. 747, pp. 123–140, 1999.

[15] H. Ishiguro, R. Sato, and T. Ishida. Robot oriented state space construction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems 1996*, pp. 1496–1501, 1996.

[16] 石井, 佐藤. オンライン EM アルゴリズムによる動的な関数近似. 信学技報, NLP97-142, NC97-94, 1997.

[17] 金子. ペット型ロボット 喜び悲しむ、本物そっくりメカトロニクス・AI 駆使. 日経ビジネス, pp. 73–76, 1999. 1999 年 7 月 12 日号.

[18] 川人, 銅谷, 春野. ヒト知性の計算神経科学 第 5 回 その 1 モザイクの拡張とコミュニケーション. 岩波科学, 71:197–204, 2001.

[19] 木村, L. P. Kaelbling. 部分観測マルコフ決定過程下での強化学習. 人工知能学会誌, 12(6):822–830, 1997.

[20] 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):618–623, 1999.

[21] B. J. Kröse and J. W. van Dam. Adaptive state space quantization for reinforcement learning of collision-free navigation. In *Proceedings of 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1327–1332, 1992.

[22] S. Mahadevan. To discount or not to discount in reinforcement learning: A case study comparing R learning and Q learning. In *Proceedings of the 10th International Conference on Machine Learning*, pp. 298–305, 1994.

[23] S. Mahadevan. Average reward reinforcement learning: Foundations, algorithms, and empirical results. *Machine Learning*, 22:159–196, 1996.

[24] S. Mahadevan and J. Connell. Automatic programming of behavior-based robots using reinforcement learning. *Artificial Intelligence*, 55(2–3):311–365, June 1992.

[25] M. L. Minsky and S. A. Papert. *Perceptron – Expanded Edition*. MIT Press, 1969. (邦訳 : パーセプトロン (改訂版), 中野, 阪口 訳, パーソナルメディア, (1993)).

[26] S. Nolfi and D. Floreano. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press, 2000.

[27] R. Pfeifer and C. Scheier. *Understanding Intelligence*. MIT Press, 1999. (邦訳 : 知の創成—身体性認知科学への招待, 石黒, 小林, 細田 監訳, 共立出版, (2001)).

[28] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, 1994.

[29] S. J. Russell and P. Norvig. *Artificial Intelligence: Modern Approach*. Prentice-Hall, Inc., 1995. (邦訳 : エージェント アプローチ 人工知能, 古川 監訳, 共立出版, (1997)).

[30] 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):144–154, 1999.

[31] A. Schwartz. A reinforcement learning method for maximizing undiscounted rewards. In *Proceedings of the 10th International Conference on Machine Learning*, pp. 298–305, 1993.

[32] 塩瀬, 椹木, 片井. 環境にナビゲートされた自律移動体の創発的行動形成と教示戦略. 計測自動制御学会論文集, 34(9):1255–1262, 1998.

[33] S. P. Singh and R. S. Sutton. Reinforcement learning with replacing eligibility traces. *Machine Learning*, 22:123–158, 1996.

[34] R. C. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I. J. Cox and G. T. Wilfong eds., *Autonomous Robot Vehicles*, pp. 167–193. Springer-Verlag, 1990.

[35] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998. (邦訳 : 強化学習, 三上, 皆川 訳, 森北出版, (2000)).

[36] P. Tadepalli and D. Ok. H-learning: A reinforcement learning method to optimize undiscounted average reward. Technical Report 94-0-01, Oregon State University, 1994.

[37] 高橋, 浅田. 階層型学習機構における状態行動空間の構成. 日本ロボット学会誌, 21(2):164–171, 2003.

[38] Y. Takahashi, M. Asada, and K. Hosoda. Reasonable performance in less learning time by real robot based on incremental state space segmentation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems 1996*, pp. 1518–1524, 1996.

[39] M. Tan. Learning a cost-sensitive internal representation for reinforcement learning. In *Proceedings of Eighth International Workshop on Machine Learning (ML91)*, pp. 358–362, 1991.

[40] E. Uchibe and K. Doya. Competitive-cooperative-concurrent reinforcement learning with importance sampling. In *Proceedings of the Eighth International Conference on the Simulation of Adaptive Behavior*, pp. 287–296, 2004.

[41] 畝見. 強化学習. 人工知能学会誌, 9(6):40–46, 1994.

[42] C. J. C. H. Watkins and P. Dayan. Technical note: Q-learning. *Machine Learning*, 8:279–292, 1992.

[43] A. M. S. Zalzala and A. S. Morris eds. *Neural Networks for Robotic Control : Theory and Applications*. Ellis Horwood, 1996.

本研究に関する発表論文

1. 査読付論文誌

[i] 石川 浩一郎，櫻井 彰人，藤波 努，國藤 進，“強化学習におけるオンラインセンサ選択”，電気学会論文誌 C，Vol. 125, No. 6, pp. 870–878, 2005.

[ii] 石川 浩一郎，櫻井 彰人，藤波 努，國藤 進，“複数の状態行動価値表を用いた R 学習の高速化”，電気学会論文誌 C，(投稿中).

2. 査読付国際会議

[iii] K. Ishikawa, T. Fujinami, A. Sakurai “Integration of Constraint Logic Programming and Artificial Neural Networks for Driving Robots”，Proceedings of 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS01)，pp. 1011–1016 (2001).

[iv] K. Ishikawa, A. Sakurai “A Target Following Robot System Built on an Autonomous Mobile Platform”，Proceedings of the 5th International Symposium on Artificial Life and Robotics (AROB00), pp. 371–374 (2000).

3. 国内発表

[v] 石川 浩一郎，櫻井 彰人 “アメニティ・ロボットの構築に向けて — 人間追随行動の実装”，人工知能学会基礎論研究会資料 SGI-FAI-9904-1.