強化学習における学習器の多重化に関する研究

２０１９年３月

西 澤 智 恵 子

強化学習における学習器の多重化に関する研究

西 澤 智 恵 子

目 次

図 目 次

表 目 次

第1章　序論

1.1　本論文の背景

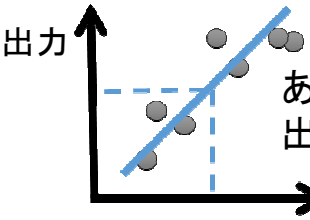
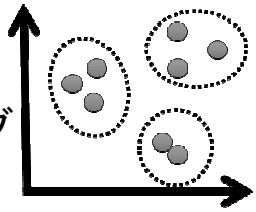
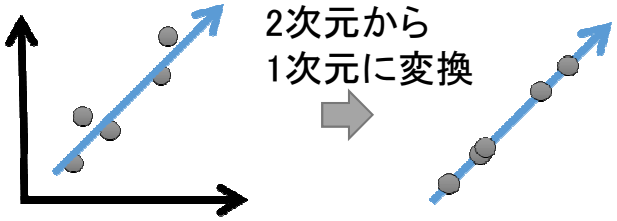
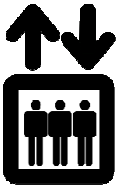
人は，本能的に行動する他に，過去の経験に裏付けられた知識に基づいて行動する．たとえ未経験の状況においても，経験した状況と照らし合わせ，なんらか似ている状況があれば，その際に得た知識を用いて自分にとって，より都合が良い結果となるように行動する．行動した結果，うまく行けば，その行動の確信度を高め，うまく行かなければ，それを新しい経験として，知識を蓄えることができる．これらは，人が経験から置かれている状況ごとに適正な行動を対応付けていく過程を示しており，“学習” と捉えることができる．また，人は，様々な物の見方ができるため，１つの事象に対して様々な視点で知識を蓄えられる．例えば，サッカーボールとバレーボールでは，人は競技に使うときは違うものとして扱う一方で，単純に移動させるときは形状という視点で，球体として転がして同じように扱い，必要に応じて知識を切り替えることができる．すなわち，人は学習の際，適宜，知識を切り替えながら，より良いと考えられる行動から試して，効率的に学習することができる．本研究は，このような状況に応じて知識を適宜切り替えて行動を決めていくという人の学習の仕方を，ロボットなどの機械において実現させることを目指すものである．

ロボットなどの機械において，人が持つ学習能力を実現しようとする手法としては，機械学習が挙げられる．機械学習は，人工知能における研究分野の一つで，人が学習によって得る規則性やルール，つまり知識を，データを解析することで機械

自身で見つけ出すための手法である．機械学習において，学習に用いるデータを教師データと言い，その教師データの与え方の違いで分類すると，“教師あり学習” と “教師なし学習”，それらの間に位置付けられる “強化学習” に分けられる．各手法の比較表を **Table 1.1** に示す．教師あり学習は，教師データとして入力に対する出力の正解を 1 対 1 のセットで与えられると，入力に対してその正解が出力できるようにルールを学習する．画像に写っているものを判断するような分類問題や，株価を予測するような回帰問題などに用いられる．教師なし学習は，教師データとして出力の正解がない入力データのみが与えられると，データの構造や分布を学習する．顧客をカテゴライズするようなクラスタリング問題や，次元削減問題などに用いられる．強化学習は，教師データとして何かの目標を達成したときのみ正解を意味する報酬が与えられると，目標を達成するまでの行動系列を学習する．ドアの開閉のようなロボットの動作獲得問題や，人の持ち時間を減らすエレベーター制御のような最適化問題などに用いられる．強化学習では，状況毎に行動を試行錯誤し，その経験に基づき，状況に応じた適正な行動を割り当てていくため，本研究で対象とする人の行動決定方法の枠組みと一致する．そのため，本研究では，機械学習の中でも強化学習に焦点を当てる．

強化学習をロボットに適用する場合，ロボットの行動設計者は，最終的なロボットの目標を報酬という形式で設定するだけで，ロボット自身が各状況に応じた適正な行動を学習で得られるようになり，設計者の負担は軽減される．このように設計が簡単である反面，強化学習では試行錯誤が必要であるため，学習回数が多くなるという問題がある．強化学習では，状況や行動のパターンが複雑になると，つまりパターン数が増加すると，学習回数がパターン数に対して指数関数的に増加するためである．これは，実環境のような複雑な環境において強化学習を適用する際に大きな障害となる．そこで，本論文では，このような問題に対して，上記で述べた人の

Table 1.1　機械学習の教師データの与え方による分類

項目	教師データ	利用例
教師あり	入力と出力(正解)が対応するデータ	分類問題:  ⇒犬  ⇒猫 回帰問題: 
教師なし	入力データのみで出力(正解)データなし	クスタリラング問題: ある特徴の類似度でデータをクラスタリング  次元削減問題: 
強化学習	目標達成時のみ正解を意味する報酬データ	動作獲得問題: ロボット動作の獲得  最適化問題: 人の待ち時間を減らすエレベータ制御 

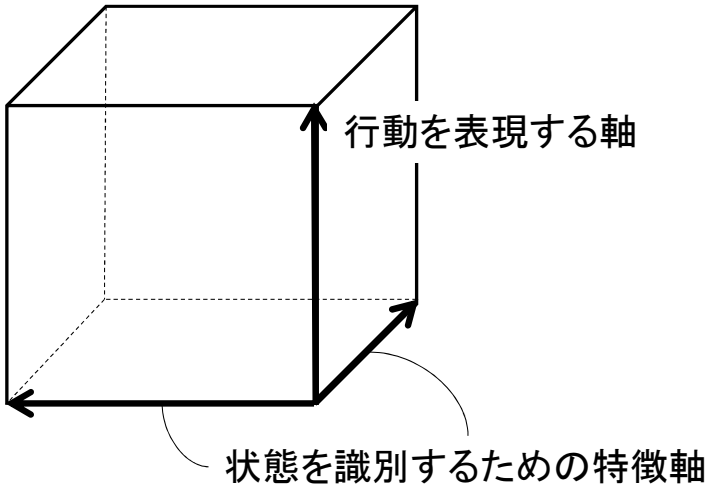


Fig. 1.1　学習空間

知識を適宜切り替えて学習を進める考え方を導入し，知識を保持する学習器を多重化することによる学習回数の低減と学習効率向上に関する研究について報告する．

1.2　強化学習の概要

強化学習において，学習エージェントは，学習空間として状態と行動のペア毎に行動の価値を持っている (**Fig. 1.1**)．ここで，行動の価値が高いことは，その状態においてその行動をとることが目標達成につながることを示し，学習とは状態毎に価値の高い適正な行動を決めていくことを示している．強化学習は，環境との相互作用により学習を進める (**Fig. 1.2**)．学習エージェントは，環境を観測して現在の状態に合わせた行動をとる．行動の結果，目標達成すれば，報酬が与えられて目標達成につながる行動の価値が上がり，学習が進む．どの行動が価値が高いかを確かめるために，学習エージェントは，各状態で行動を試行錯誤する必要がある．そのため，状態数が増えるほど学習回数が増え，学習時間がかかるという課題がある．

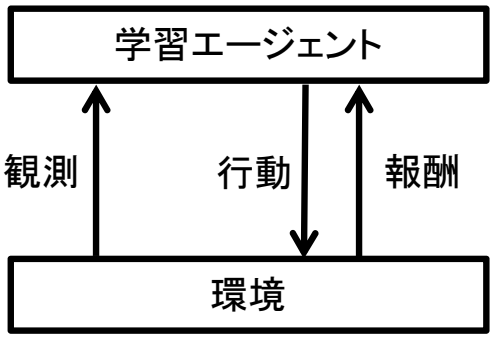


Fig. 1.2　強化学習における学習エージェントと環境の関係

1.3　提案手法の概論

本手法では，強化学習において，人のように適宜視点を切り替えて効率良く学習を進めることを目指す．従来の強化学習では，一つの学習空間を用いて学習を進めるため，環境が複雑になると学習空間が膨大となり，学習に時間がかかるという課題がある．それに対して，本手法では複数の学習空間を用いる．人における視点の違いを学習空間の違いで表現し，それらを同時に学習する．行動選択の際，学習が進んでいると考えられる空間に適宜切り替えることで，学習エージェントが効率良く学習する手法を提案する．例えば，二つの学習空間を用いる場合，一つは，適正行動を学習するのに十分な状態数をもつ学習空間（以下，全体空間と呼ぶ.）を，もう一つは，全体学習空間を圧縮する部分空間（以下，部分空間と呼ぶ.）を用いる (**Fig. 1.3**)．全体空間は，状態数が多いので学習は遅いが，環境に対して適正行動を正確に対応づけできる．部分空間は，状態数が少ないので学習は速いが，環境に対して適正行動を粗く対応付けする．行動選択をする度に，この二つの学習空間のどちらかを使い，更新することで，学習が速く環境にも細かく対応づけする学習を目指す．

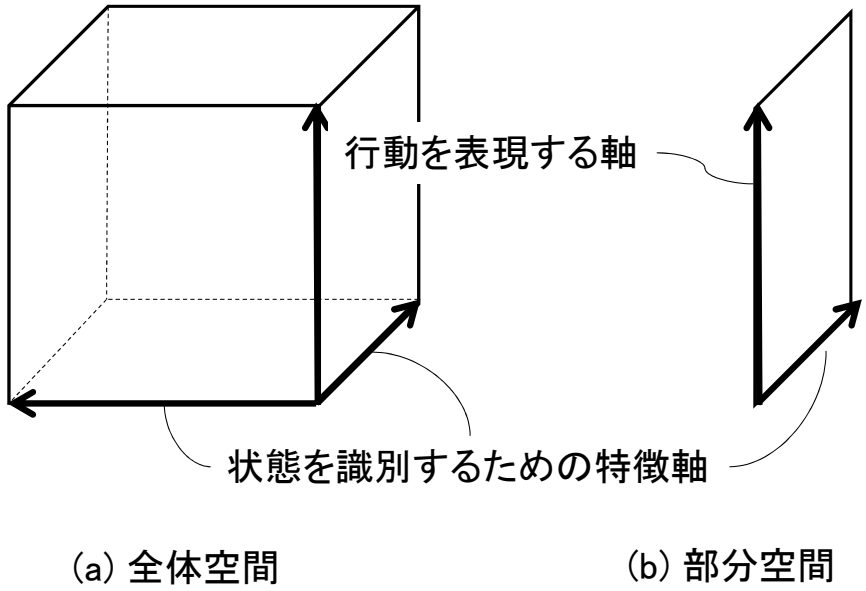


Fig. 1.3　本手法における二つの学習空間の例

1.4　関連研究

強化学習において，学習を効率化する手法としては，(1) 学習空間を適宜分割して学習する手法，(2) 学習空間における状態や行動の価値を関数近似する手法，(3) 既学習情報や設計者が与える情報を事前知識として利用する手法，(4) 複数の学習空間を用いて一定期間学習ののち，学習空間を切り替える手法がある．

(1) の学習空間を分割する手法としては，複雑なタスクを一気に学習させるのではなく，目標達成付近の単純な状況から徐々に複雑な状況へと移行していく [8] や，簡単なタスクから段階的に複雑なタスクへと移行していく [9]，連続状態空間を離散化する際，タスクに応じた適切な状態数を維持することで，過剰な状態が分割されることによる学習の遅延を防ぐ [22]，制御対象の将来の状態を予測する予測モデルと，制御出力を学習する強化学習コントローラの組を，学習空間を分割するように準備し，それらを予測モデルの予測誤差に基づいて切替，または組み合わせて使う

もの [23] や，状態空間は最初は荒く分けて学習が進まない場合にさらに分割，行動空間は学習が進むに従って連続空間に切り替えていくという仕組みを持つ [13]，複雑なタスクを細かいサブタスクに分解した強化学習モジュールを階層的に並べて学習する階層型強化学習 [10, 11] がある．また，大きく複雑なマルコフ決定過程を木構造で表現する algebraic decision diagrams(ADDs) を入力とし，ゴールまでの行動回数を基に，大域的最適性を失わずに階層的クラスをつくり，ゴールまでの最短行動を探索しやすくするもの [12] がある．ただし，これは環境情報を学習しながら得るわけでないため，学習による環境情報の更新による木構造の改変は考慮していない．これらの研究は，環境分割の必要性を適宜確認しながら学習空間をつくりだすことにより，あるいは，全体の学習空間を分割した学習空間を複数用意して切り替えることにより，各学習空間の状態数が膨大となるのを防ぎ，効率良く学習する手法である．

(2) の関数近似を用いる手法としては，状態の評価関数を正規化ガウス関数を基底として近似し，近似の精度によって基底関数を動的に増やすもの [17]，状態価値関数の近似に用いる要素に位置と勾配情報を持たせ，その要素を動的に追加・削除・再配置することで状態価値関数の形状の複雑さに即すもの [18]，行動価値の表現にニューラルネットワークを用いるもの [19]，さらにネットワークの層を増やした深層学習を用いるもの [20][21] がある．これらは，連続空間をそのまま扱うことで，離散化の際に学習空間の状態数が膨大になるのを防ぎ，さらに，近似のためのパラメータを必要に応じて追加・削除したり，近似結果の出力方法を工夫することで学習パラメータが膨大になるのを防ぎ，効率良く学習する手法である．

(3) の事前知識を利用する手法としては，回避行動はほぼ全てのタスクで共通に達成させる動作知識という考えに基づき，回避行動をタスク達成のための学習空間とは別の学習空間にて学習させ，別のタスクでもその学習済み空間を利用するもの

[24]，学習済みの空間を利用して行動を選択，試行回数に対するタスク成功率を測定し，タスク成功率の低下した環境のみ再学習するもの [25] や，設計者が与えた基礎知識を利用して行動を選択し，学習初期のランダム行動を抑制するもの [26]，過去の成功・失敗経験に基づいて探索戦略を変化させる [14][15]，学習初期に事前知識を用い，それが間違っている場合に対応するため，経験した回数に応じて忘却する [16] がある．これらは，既学習情報や設計者が与える情報を事前知識として利用し，学習初期のランダム行動よりもタスク達成可能性が高い行動を選択されやすくし，学習を効率化する手法である．

(4) の一定期間学習ののち，学習空間を切り替える手法としては，学習法やパラメータ数の異なる学習空間を複数並列化して持ち，目標達成毎に得られた報酬から価値を推定，比較して切り替えるもの [27] や，自分の学習結果から同時に学習する他者の状態価値を推定して比較し，より高い方を自分の行動学習に利用するもの [28] がある．これらは，ある一定期間の行動経験によって得られた報酬を基に，複数の学習空間の価値を推定して比較し，より価値の高い空間に切り替える，あるいは，その空間の情報を利用することにより，学習を効率化する手法である．

また，(1)(4) を組み合わせる手法として，学習に必要な全状態を対象とする完全空間と状態を粗視化して状態数を減らした制限空間を並列に学習，学習初期は制限空間を用い，ある程度学習が進んだ後に完全空間に切り替えるもの [29] がある．これは，学習初期は状態を荒く見て状態数を減らして制限空間で学習を進めることで，学習は速く進むが制限空間のみでは性能が悪いため，完全知覚空間でも並列に学習を進め，適当なタイミングで切り替えて学習を効率化する手法である．

これらの手法に対して，本提案手法では，学習の全体空間の使用しない (1) とは違い，全てを学習対象とした全体空間を用いる．全体空間に加えて，学習空間の部分空間を並列化して学習を進め，学習進捗度に応じてそれら空間を切り替えること

で学習の効率化を図る．部分空間は，全体空間より小さいため，学習自体が速く進む．全体空間の学習が進むまでは，学習が進んでいる部分空間を用いて行動を選択し，全体空間が進んでくると，全体空間を用いて行動を選択するようになる．つまり，全体空間をそのまま扱うことで大域的最適性を失わずに，学習初期のランダム行動を，部分空間の学習結果に基づいて，抑制することができる．(2) の学習空間を関数表現する方法は，環境の状況を状態に区切ることなく，その状況の値を関数のパラメータ調整により再現し，別の状況においても，関数に状況を代入すれば，状態価値等の妥当な値を得られるようにするものである．しかし，早く学習するようにパラメータ数を限定すれば近似精度が悪くなり，精度良くしようとすれば，多くのパラメータを必要とし，学習が遅くなる．この関数近似する手法に対して，本提案手法は優劣を比較するものではなく，技術的に全く異なる手法であり，相補間的に組み合わせて性能を高めることが期待できる．例えば，本提案手法に関数近似法を導入すると，パラメータの多いものと少ないもののそれぞれの関数近似表現を持ち，パラメータ学習をそれぞれ同時にする手法となる．本提案手法では，学習の効率化に用いる部分空間が全体空間と同時学習が可能なため，(3) と違い，事前知識や既学習情報を準備する必要はない．また，本手法では，学習進捗度の比較による学習空間の切り替えは，環境の状態を認識して行動選択する度に実施するため，ある一定期間の行動経験が必要な (4) と違い，1 回の行動経験により変化する学習進捗度にも即座に対応できる．本提案手法の全体空間と部分空間を並列に持つ考え方は，(1)(4) を組み合わせた [29] の完全空間と制限空間を並列に持つ考え方と似ているが，[29] では，制限空間から完全空間へ閾値を用いて切り替えること，切り替わった後は，完全空間のみを使う点が異なる．本手法では，全体空間と部分空間のそれぞれの学習進捗度を比較して選択するため，閾値は必要ない，また，各状態で選択するため，目標達成のために学習進捗に応じて全体空間と部分空間を行き来できる．そ

のため，環境変化が起きた場合においても，全体空間のみの場合よりも部分空間がより速く環境変化に対応し，学習を効率化する可能性を持つ．

1.5　本論文の構成

ここでは，それぞれ章の概要と関係を述べる．

本章では，本研究の背景と提案手法の概論を述べた後，関連研究に対する提案手法の位置づけについて述べた．

第 2 章「提案手法」では，学習器を多重化するアルゴリズムについて述べる．

第 3 章「学習器の二重化」では，最も基本的な多重化パターンである全体空間に部分空間を一つ多重化する場合について，提案手法の有効性をシミュレーション実験により検証する．本手法において，切り替え対象である部分空間が，目標達成に対して有用であるかどうかが学習効率に大きく影響する．ここでは，部分空間の有効性を変化させて，提案手法の学習効率を検証する．

第 4 章「学習器の多重化」では，部分空間の数を増やす場合について，提案手法の有効性をシミュレーション実験により検証する．切り替え対象である部分空間について，有用なものが準備できるとは限らない．ここでは，有用でない部分空間が複数多重化された場合や有用なものと有用でないものが混在した場合について，提案手法の学習効率を検証する．

第 5 章「学習器の多重化（包含関係なし）」では，学習器多重化の別応用として，全体空間と部分空間が包含関係を持たない報酬付与に遅れがある環境への応用例を述べる．

第 6 章「実環境における検証」では，実機実験により提案手法の有効性を検証する．シミュレーションでは，学習エージェントの行動によって生じる環境変化やそ

の認識が確定的あるのに対して，実機ではそれらが不確定である．ここでは，実機を用い，実時間において，提案手法が学習効率を向上させるかどうかを検証する．

第 7 章「数理解析」では，最も基本的な多重化パターンである全体空間に部分空間を一つ多重化する場合について，数理解析結果を述べる．

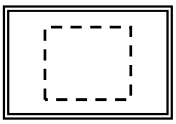
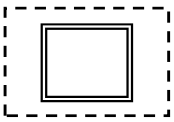
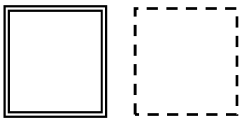
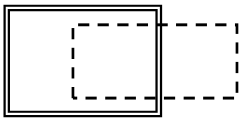
第 8 章「結論」は，各章の結果をまとめて，本論文の結論を示す．

第2章　提案手法

2.1　緒論

強化学習において，あるタスクを持つ学習エージェントは，周囲の環境を観測し，状態を求め，その状態に確率的に対応する行動を実行する．これを繰り返し，最終的にタスク達成した場合に与えられる報酬を元に，各状態における行動実行確率を調整することで学習を進める．本論文では，強化学習で用いる学習空間と，その空間の適性を比較するために正解空間を扱う．学習空間とは，状態と行動のペア毎に行動価値を持つ空間である．学習エージェントは，環境の全状態を認識するわけではなく，エージェント設計者が与えたタスク達成のための行動を決定するのに必要と考える状態を認識する．あるタスク達成に向けて学習が進むと，認識する状態毎に適正な行動が決められるようになる．正解空間とは，タスク達成のための行動を決定するのに必要十分な状態を持つ理想的な学習空間であり，実際に求めることは難しい．二つの空間の関係パターンを **Table 2.1** に示す．**Table 2.1** において，学習空間を二重線，正解空間を破線で表現する．学習空間と正解空間の関係は 5 パターンあり，学習エージェントはあるタスク達成において，No.1 のように学習空間が正解空間の全てを表現でき、かつ一致する場合，最も効率的，かつ正確に学習でき，No.2 のように学習空間が正解空間を含む場合，正確に学習できるものの，No.1 に比べて非効率で学習が遅くなり，No.3 のように学習空間が正解空間の一部のみ表現する場合，一部しか正確に学習できず，No.4 のように学習空間が正解空間を全く表

Table 2.1　学習空間，正解空間の関係パターン

No.	関係図の例	説明
1		学習空間＝正解空間
2		学習空間⊃正解空間
3		学習空間⊂正解空間
4		学習空間≠正解空間 学習空間∩正解空間＝Φ
5		学習空間≠正解空間 学習空間∩正解空間≠Φ

現しない場合，全く学習ができず，No.5 のように学習空間と正解空間が一部重なる場合，No.4 と No.2，または No.3 との組み合わせとなり，一部が正確に学習できるが，一部は学習できない．強化学習において，正確に行動を学習するためには，学習空間は正解空間を含む必要があり，また，効率的に学習するためには，正解空間と一致していることが望ましい．しかしながら，実環境のように複雑な環境において，学習エージェントの設計者があらかじめ正解空間と一致する学習空間を決定することは困難であり，基本的には，**Table 2.1** の関係パターン No.2 のように，正解空間を含むような学習空間を用いることとなる．本提案手法は，強化学習法の一つである Q-learning[30] を拡張し，様々な学習空間を多重化することで，実質的な学習空間を正解空間に近づけ，学習の効率化を目指す．

2.2　学習空間

一般の Q-learning は，学習空間として，Q-table という状態毎にそれぞれの行動に対する評価に対応する Q 値をもつテーブルを一つだけもつ．それぞれの状態は，環境を量子化し割り当てるものである．それに対して，提案手法は，行動選択数は同じであるが，状態割り当ての異なる複数の Q-table を持つ．それぞれの状態割り当ては，同じ環境 env に対して，下式のように表現できる．

$$ {}^{R}s = func_R(env) \tag{2.1} $$

$$ {}^{D}s = func_D(env) \tag{2.2} $$

ここで，${}^{R}s$，${}^{D}s$ は，各空間の状態をそれぞれ表し，$func_R$,$func_D$ は，エージェントの置かれている環境を各 Q-table のどの状態に対応するかを表す関数である．状態の割り当てが粗い空間の状態を ${}^{R}s$，それよりも細かく割り当てる空間の状態を ${}^{D}s$

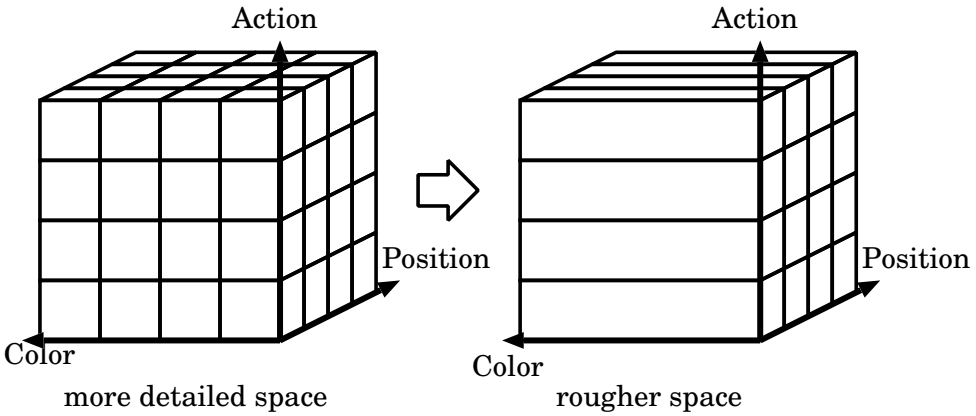


Fig. 2.1　詳細空間 Q-table と粗い空間 Q-table

とすると，${}^{R}s$ は ${}^{D}s$ のいくつかの集合で表される．すなわち，粗い空間の状態に対してそれが表す環境の範囲を rep_R，詳細な空間の状態のものを rep_D とすると，下式のようになる．

$$rep_R({}^{R}s_i) = rep_D({}^{D}s_j) + rep_D({}^{D}s_k) + \cdots \tag{2.3}$$

Fig. 2.1 に詳細空間 Q-table と粗い空間 Q-table の例を示す．環境パラメータは，対象物体の色とその位置であり 2 次元空間中の点である．図の例では，詳細空間 Q-table の状態は，環境である位置を 4 分割，色を 4 分割で，16 状態に分割しており，行動の選択肢が，4 通りあるので，Q 値は 64 個になる．粗い空間 Q-table の状態は，環境である位置のみ 4 分割するだけなので，4 状態に分割しており，行動の選択肢は，同じ 4 通りなので，Q 値は 16 個になる．この例では，粗い空間 Q-table の 1 つの状態に対応する環境の範囲は，詳細空間 Q-table の 4 つの状態に対応する環境の範囲と同じである．

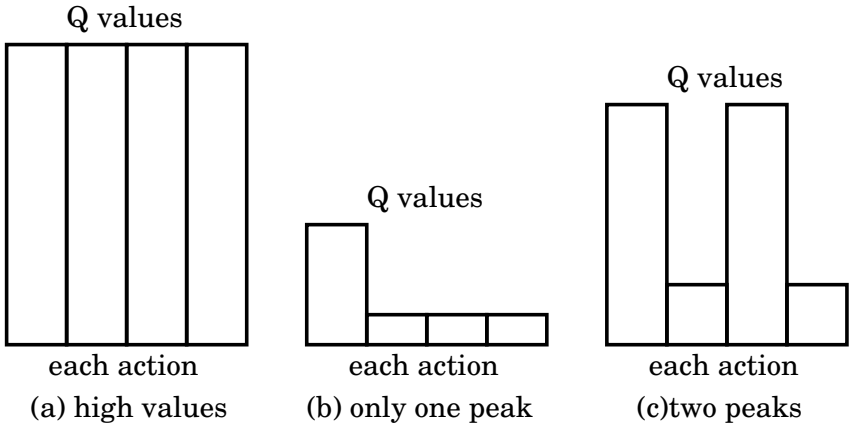


Fig. 2.2　ある状態における各行動の Q 値の分布例

2.3　Q-table の選択

詳細空間 Q-table と粗い空間 Q-table の複数の Q-table を学習しているので，行動選択をどのような方法で決めるかが問題になる．提案手法では，環境を各 Q-table においてそれぞれの状態で表現し，それぞれの状態における行動の Q 値をみて，どちらの Q-table が有用かを判断，有用な Q-table を用いて行動選択する．ここで，**Fig. 2.2** に，ある状態における行動の Q 値の例をあげる．有用な Q-table は，高い報酬を得るためにどの行動をとるべきかについて，確実性の高い情報を持っているものと考える．**Fig. 2.2** の中では，(a) が一番高い Q 値であるが，Q 値が全て同じであるため，行動の選択確率も全て同じとなり，どの行動をとるべきかが分からない．それに対して，(b)(c) は行動の Q 値が異なることで，行動の選択確率に差があり，どの行動をとるべきかが分かる．本論文では，より有用な Q-table は，(a) よりも (b)(c) のような Q-table であり，さらに行動選択確率に偏りがあればあるほど，どの行動をとるべきかについて確実性の高い情報を持つと考える．そこで，行動の選択確率の偏りを評価する指標として下式の平均情報量（情報エントロピー）を用いる．平均

情報量は，状態 s における行動選択確率の偏りが大きくなればなるほど小さくなる．

$$H(s) = \sum_{a \in ACT} p(a \mid s) \log_2 \frac{1}{p(a \mid s)} \tag{2.4}$$

すなわち，行動選択には，それぞれの状態において平均情報量が最も小さい Q-table を用いる．

2.4　Q-table の更新

Q-table の更新は，一般の Q-table の更新アルゴリズムと同様である．ここで，r は報酬，α は学習率，γ は割引率を示す．

$$\begin{aligned} &{}^{i}Q({}^{i}s_t, a) \leftarrow (1-\alpha)\, {}^{i}Q({}^{i}s_t, a) + \alpha(r + \gamma V(s_{t+1})) \\ &i \in \{R : Rough, D : Detail\} \end{aligned} \tag{2.5}$$

ただし，状態価値関数 $V(s)$ は (2.6) を用いる．学習速度を速めるため，行動選択に有用として粗い空間が用いられたならば，詳細空間と粗い空間を合わせて比較し，最大の Q 値を状態価値とする．

$$V(s) = \begin{cases} \max_{i \in \{R,D\}} \max_{a \in ACT} {}^{i}Q({}^{i}s, a) & \text{if } selected\ Q-table = Rough \\ \max_{a \in ACT} {}^{D}Q({}^{D}s, a) & otherwise \end{cases} \tag{2.6}$$

2.5　学習アルゴリズム

前節までで説明した学習アルゴリズムの概念を **Fig. 2.3** に示す NS チャートをもとに整理して説明する．

(1)　全 Q-table の初期化：

学習エージェントが使用する全 Q-table の Q 値を初期化する．

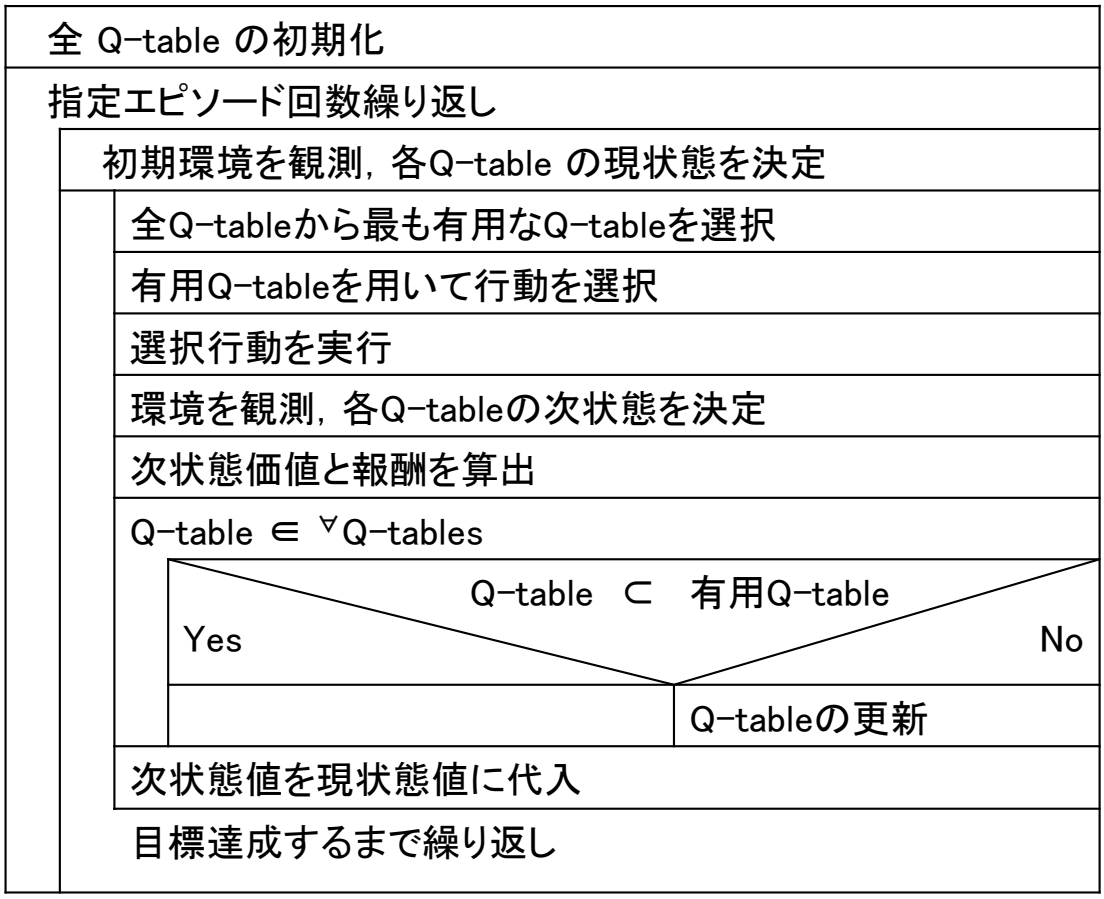


Fig. 2.3　提案手法の NS チャート

(2)　指定エピソード回数繰り返し：

目標達成するまでを 1 エピソードとし，あらかじめ設計者が決めたエピソード回数分学習を繰り返す.

(3)　初期環境を観測，各 Q-table の現状態を決定：

学習エージェントが置かれている環境を観測し，各 Q-table のどの状態に対応するかを決定する.

(4)　全 Q-table から最も有用な Q-table を選択：

学習エージェントがどの Q-table を用いて行動選択するかは，(2.4) で計算できる平均情報量 $H(s)$ を用いて判断する．現状態における各 Q-table の平均情報量を計算し，平均情報量が最も低い Q-table を行動選択のために選ぶ.

(5)　有用 Q-table を用いて行動を選択：

学習エージェントは，平均情報量により選択された Q-table に対し，下式の Boltzmann 選択を使い時刻 t の行動 a_t を選択する.

$$p(a_t \mid s_t) = \frac{\exp(\frac{Q(s_t,a_t)}{T})}{\Sigma_{a \in ACT} \exp(\frac{Q(s_t,a)}{T})} \tag{2.7}$$

(6)　選択行動を実行：

学習エージェントが選択行動を実行する.

(7)　環境を観測，各 Q-table の次状態を決定：

学習エージェントが置かれている環境を観測し，各 Q-table のどの状態に対応するかを決定する.

(8)　次状態価値と報酬を算出：

(2.6) にしたがい，それぞれの状態価値を求める．また，目標達成した場合は，報酬を取得する.

(9)　全 Q-table 分繰り返し：

Q-table 更新のため，(10)(11) を繰り返す.

(10) 更新対象 Q-table が有用な Q-table に包含されるかを確認：

更新対象 Q-table が有用な Q-table に包含される場合は，(9) に戻り，Q-table を更新しない．その他の場合は，(11) に進み，Q-table を更新する．例として，詳細空間 Q-table と粗い空間 Q-table を考える．粗い空間 Q-table は詳細空間

Q-table に包含される．詳細空間が有用な Q-table として選択された場合は，詳細空間 Q-table の更新はするが，粗い空間の Q-table は更新しない．粗い空間が有用な Q-table として選択された場合は，詳細空間 Q-table と粗い空間 Q-table のどちらも更新する．

(11) Q-table の更新：
Q 値の更新式の (2.5) を用いて Q-table を更新する．

(12)　次状態値を現状態値に代入：
(7) で決定した状態値を現状態値として設定する．

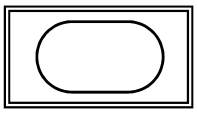
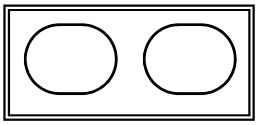
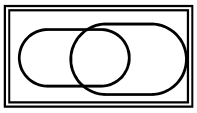
(13)　目標達成するまで繰り返し：
目標達成した場合は，1 回分のエピソード終了として，(2) に戻る．その他の場合は，(4) に戻る．

上記アルゴリズムにて Q 値を更新し，各環境における適正行動を学習する．

2.6　学習空間の多重化パターン

詳細空間と粗い空間の多重化パターンについて，**Table 2.2** に示す．**Table 2.2** において，学習空間のうち，詳細空間を二重線で，粗い空間を実線で表現する．詳細空間と粗い空間の多重化の関係は 3 パターンあり，No.1 は，詳細空間に粗い空間を一つ多重化する最も基本的なパターンで，No.2 は，それぞれ独立した粗い空間を多重化するパターン，No.3 は，一部重なる粗い空間を多重化するパターンである．前節までで述べた学習アルゴリズムが，No.1，2 については適用できる．No.3 は，粗い空間同士の重なり部分について，包含関係が不明であり，Q-table 更新方法の再

Table 2.2　学習空間の多重化パターン

No.	関係図の例	粗い空間の説明
1		単数
2		複数で重ならない
3		複数で一部重なる

検討が必要となる可能性がある．本研究では，最も基本的な多重化パターンである No.1 から順に検証を行い，本論文では，No.1，2 について検証した結果を述べる．

第3章　学習器の二重化

3.1　緒言

本章では，提案手法において最も基本的な多重化パターンである詳細空間に一つの粗い空間を二重化する場合について，提案手法の有効性をシミュレーション実験により検証する．本章において，詳細空間 Q-table を全体空間 Q-table，粗い空間 Q-table を部分空間 Q-table とおく．全体空間 Q-table は，対象となる実験環境を十分正確に学習できるほど細かい状態をもち，部分空間 Q-table は，全体空間の状態を統合して学習に最低限必要な粗い空間表現をする状態空間をもつ．各 Q-table の行動の数と種類は互いに同じである．学習空間を多重化することの影響を検証するため，対象の環境に対してどのような全体空間 Q-table と部分空間 Q-table を用意するかを議論するのでなく，用意した全体空間 Q-table と部分空間 Q-table に対して，シミュレーション環境がどのような場合に学習効率がどうなるかをシミュレーション実験により検証する．具体的には，シミュレーション環境が，(ケース 1) 部分空間 Q-table の状態表現の粗さでも十分表現できる場合，(ケース 2) 部分空間 Q-table では粗すぎ，全体空間 Q-table では細かすぎる場合，(ケース 3) 全体空間 Q-table の状態表現の細かさがないと学習できない場合の三つの場合において本提案手法の有効性を調べる．ここでは，全体空間 Q-table に部分空間 Q-table を多重化する本提案手法と有効性を比較するために，全体空間 Q-table のみと部分空間 Q-table のみを用いる単純な従来手法についてもシミュレーション実験をする．

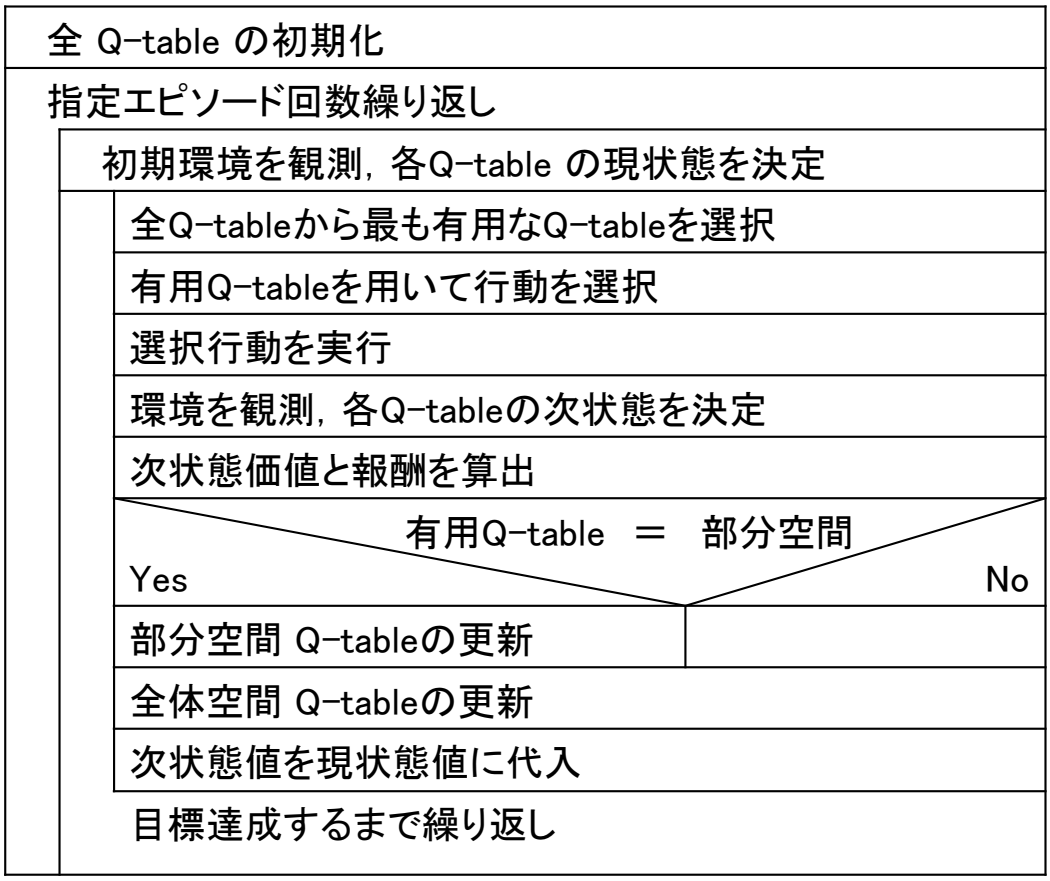


Fig. 3.1　提案手法 (二重化版) の NS チャート

3.2　提案手法 (二重化版)

第 2 章で述べた提案手法を，全体空間に一つの部分空間を二重化する場合に適用する．学習アルゴリズムの NS チャートを **Fig. 3.1** に示す．全体空間と一つの部分空間の多重化では，Q-table の更新部分が単純化される．部分空間は全体空間に包含されるため，行動選択に有用な空間として部分空間が選択された場合は，部分空間と全体空間のどちらも更新し，全体空間が選択された場合は，全体空間のみを更新する．

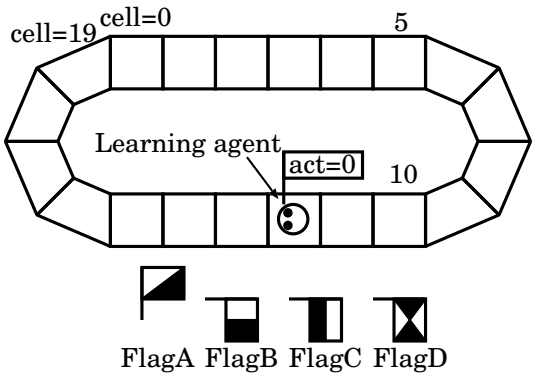


Fig. 3.2　シミュレーション環境

3.3　シミュレーション環境

シミュレーションに用いる環境として，部分空間がどれだけ環境を正確に表現しているか分かりやすい恣意的な環境を用意する．具体的には，**Fig. 3.2** に示す環境を考える．この環境では，$CellN$ 個のセルが円環状に並べられる．学習エージェントは，時刻 $t = \{0, 1, 2, \cdots\}$ が変わるごとに一つずつセル $cell_t$ を時計回りに移動する．学習エージェントは，時刻 t において，必ず 0 から $ActN - 1$ の整数のいずれかの値 n_t の札を上げる．学習エージェントからみえるところに，$ClrN$ 種類の旗があり，時刻ごとに掲げる旗 clr_t がランダムに切り替わる．**Fig. 3.2** では，FlagA が立てられている状況である．学習エージェントが上げる札 n_t が，そのときに立てられている旗の種類とセルの場所で予め決められる行動 $n(cell_t, clr_t)$ と一致すると，その場で段を上がり，そうでない場合，一番下の段へ下りる．そして，$StageN$ 段分上がると報酬がもらえる．すなわち，旗の種類とセルの場所で予め決められる行動を，連続して $StageN$ 段分実行できると報酬がもらえる．この環境において，行動選択に失敗した場合，一番下の段となる．しかしながら，セルは円環上に並べられており，行動選択に失敗しても，時刻が変わると自動的に隣のセルに移動して状態

が変わるため，Q 値の更新式において，一番下の段にいる場合にも，次状態の評価値として (2.6) で算出する値を用いるのは適切でない．一番下の段にいる場合は，次状態の評価値を 0 として Q 値を更新する．学習エージェントが，時刻 t のときにいる $cell_t$ は，0 時刻のときに，0 セルにいるとすると，$cell_0 = 0$ と記述され，時刻 t の $cell_t$ では，下式で記述される．

$$cell_t = t \bmod CellN \tag{3.1}$$

また，時刻 t の報酬 R_t は，下式で記述される．

$$\begin{aligned} R_t &= \begin{cases} 1 & \text{if } \left(\sum_{i=t-m+1}^{t} tf_i\right) = StageN \\ 0 & otherwise \end{cases} \\ tf_i &= \begin{cases} 1 & \text{if } n(cell_i, clr_i) = n_i \\ 0 & otherwise \end{cases} \end{aligned} \tag{3.2}$$

上記の環境において，$StageN$=4 として，4 連続で適正行動を実行できると報酬がもらえるものとし，全体空間と部分空間を以下のように定義する．

- 全体空間 Q-table：
  学習エージェントの場所 (20) と旗 (4) と段数 (4)，学習エージェントの行動 (4) の 4 軸で表現される．

- 部分空間 Q-table：
  学習エージェントの場所 (20) と段数 (4)，学習エージェントの行動 (4) の 3 軸で表現される．

この環境では，全体空間 Q-table を用いれば，どのセル位置にいても，どの旗が揚がっても，各状態ごとに適正行動を学習することができる．しかし，部分空間 Q-table のみを用いると，どのセル位置にいても，同じセル位置なら旗の種類に関わら

ず適正行動が一意に決められていないと，各状態ごとに適正行動を学習することができない．同じセル位置でも，掲げる旗により適正行動が異なると，確率的にしか適正行動を学習することができない．そこで，本研究では，同じセル位置において，一意に適正行動の決まる環境から旗の種類ごとに適正行動の異なる環境まで，部分空間 Q-table の有用度を変更することで，本提案手法の多重 Q-table の有効性がどのように変化するかを観測する．学習に用いるパラメータは，Q 値の初期値を 0.0，報酬 r を 1.0，学習率 α を 0.08，割引率 γ を 0.8 に設定する．行動選択手法として Boltzmann 選択を用い，温度 T を 0.10 に設定する．

3.4　比較のための実験条件

環境の全てが部分空間 Q-table のみで学習できるケース 1，環境の半分を部分空間 Q-table のみで学習できるケース 2，環境を部分空間 Q-table では学習できないケース 3 の合計 3 種類の環境でシミュレーション実験をし，本提案手法の特性を確認する．3 つの環境とも，1000 回目のエピソードから環境が変化する．

ケース 1：学習環境の全てを部分空間のみで表現できる場合 (部分空間による推論が完全に有効な場合)

この環境では，学習エージェントの適正行動の全てが部分空間のみで表現できる．具体的には，**Fig. 3.3** に示すように，偶数セルにいるときは，行動 0，奇数セルのときは，行動 1 をとればよく，学習エージェントの適正行動は全て，旗に関係なく学習エージェントの位置情報のみで決まる．また，環境変化後も，偶数セルと奇数セルの設定が入れ替わるだけで，位置情報のみで決まる．言い替えると，エージェントの適正行動はすべて状態をセル位置のみで記述する部分空間 Q-table のみで表現できる．この場合，部分空間を用いる推論が完全に有効であると考えらる．

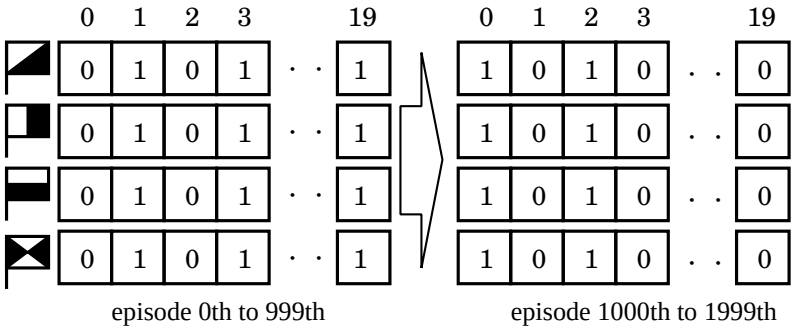


Fig. 3.3　ケース 1：学習環境の全てを部分空間のみで表現できる場合の環境

ケース 2：学習環境の半分を部分空間のみで表現できる場合 (部分空間による推論が半分有効な場合)

この環境では，学習エージェントの適正行動のうち，半分が部分空間のみで表現でき，残り半分は部分空間では確率的にしか表現できない．具体的には，**Fig. 3.4** に示すように，学習エージェントが偶数セルにいるときは，行動 0，奇数セルにいるときは，旗により，行動 0 か，行動 1 が適正行動であると決まる．言い替えると，学習エージェントの適正行動のうち，半分は部分空間 Q-table のみで表現できるが，残り半分は，部分空間 Q-table のみでは確率的にしか決められず全体空間 Q-table を必要とする．ここでの環境変化は，部分空間 Q-table に対しては変化がなく全体空間 Q-table に対してのみ変化のあるものとする．この場合，部分空間 Q-table による推論のうち，半分は有効であると考えられ，提案手法が他の 2 つの従来法と比べて有効であることが期待できる．

ケース 3：学習環境を部分空間では表現できない場合 (部分空間による推論が全く有効でない場合)

この環境では，学習エージェントの適正行動は部分空間のみでは表現できない．具体的には，**Fig. 3.5** に示すように，学習エージェントの適正行動は全て，学習エー

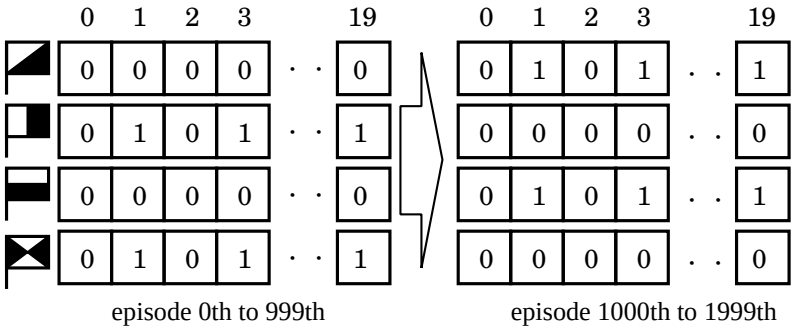


Fig. 3.4　ケース 2：学習環境の半分を部分空間のみで表現できる場合の環境

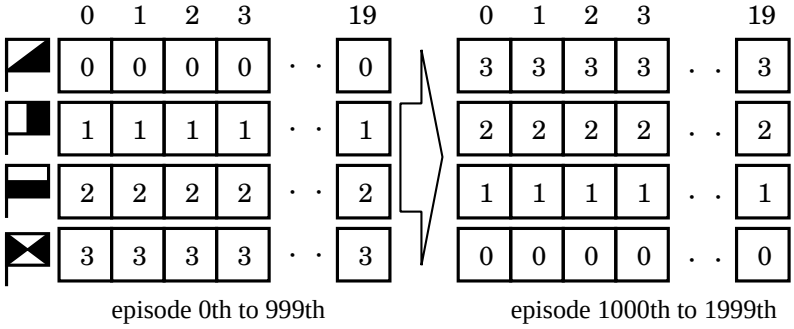


Fig. 3.5　ケース 3：学習環境を部分空間では表現できない場合の環境

ジェントがいるセルの位置と旗により決まる．ここでの環境変化は，部分空間のみでは学習できない別の環境へ変化する．言い替えると，学習エージェントの適正行動は全て部分空間 Q-table のみで表現できず，全体空間 Q-table を必要とする．この場合，部分空間 Q-table による推論は全く役に立たないと思われる．

3.5　シミュレーション実験結果と考察

部分空間 Q-table と全体空間 Q-table の 2 つの Q-table を持つ提案手法と部分空間 Q-table のみの従来法と全体空間 Q-table のみの従来法との比較シミュレーションの

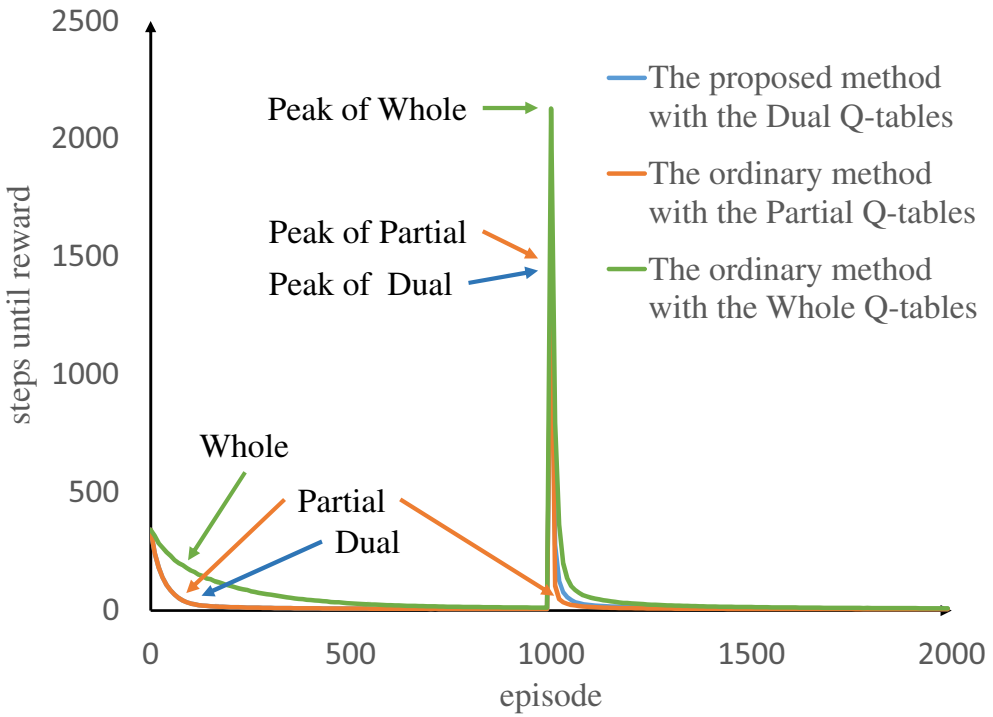


Fig. 3.6　シミュレーション実験結果：ケース 1

実験結果を示す．ケース 1 から 3 までのそれぞれの実験結果のグラフは，1000 エピソード目に環境を変更する 0 から 1990 エピソードまでの 1000 試行シミュレーション実験の平均の結果である．

Fig. 3.6 にケース 1 のシミュレーション実験結果を示す．ケース 1 で，提案手法と部分空間 Q-table のみの従来法，および全体空間 Q-table のみの従来法の 3 手法とも適正な値を得ることができた．部分空間 Q-table のみの従来法と提案手法はほぼ同じ早さで，全体空間 Q-table のみの従来法よりも早く収束した．部分空間 Q-table が環境に対して適しているため，効率良く環境を学習することができた．全体空間 Q-table は同じ状態とみなして良い環境に対して余分に行動を試すため，効率良く環境を学習することができなかった．提案手法は，初期状態からの学習は，グラフ上

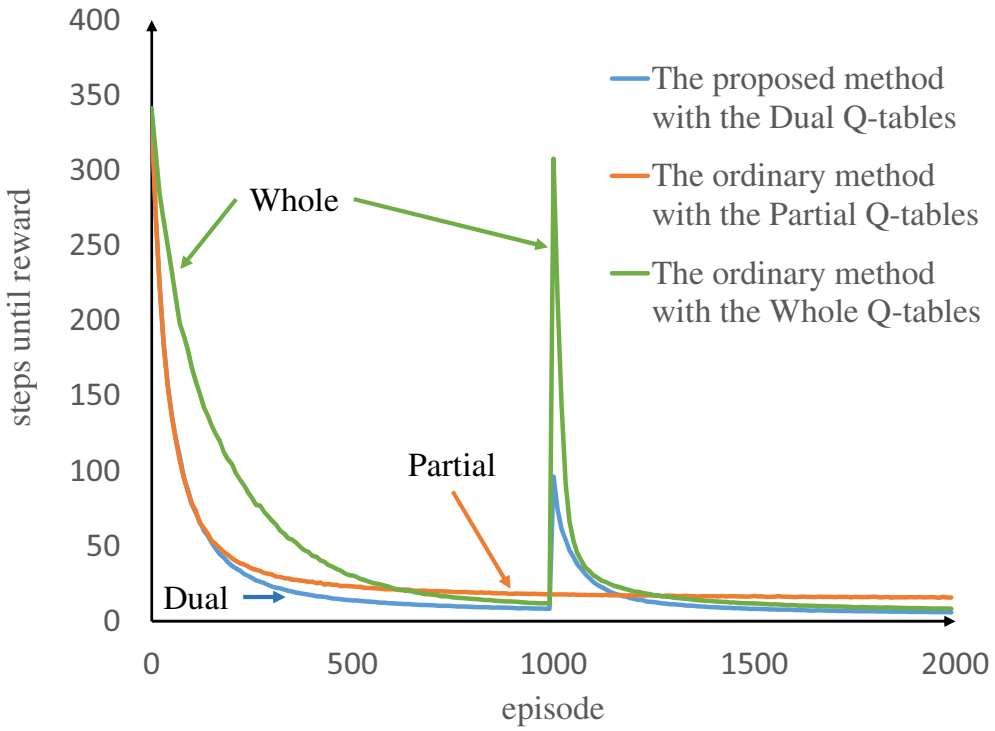


Fig. 3.7　シミュレーション実験結果：ケース 2

で部分空間 Q-table のみの従来法と区別がつかないほどの速度で収束した．しかし，環境変化に関しては，全体空間 Q-table のみの従来法よりは良いものの，部分空間 Q-table のものよりも幾分遅くなった．この結果より，部分空間の推論が完全に正しい，ケース 1 において，提案手法は，全体空間 Q-table のみの従来法よりも，部分空間のみの学習に近く有効であることを示した．

Fig. 3.7 にケース 2 のシミュレーション実験結果を示す．ケース 2 で，提案手法と全体空間 Q-table のみの従来法の 2 手法は最適な値を得ることができた．しかし，部分空間 Q-table のみの従来法は，最適な値を得ることができなかった．これは，半分の環境は部分空間で表現できるため，学習エージェントは適正行動を決められるが，残り半分の環境は全体空間でないと表現できないため，学習エージェントは適正行

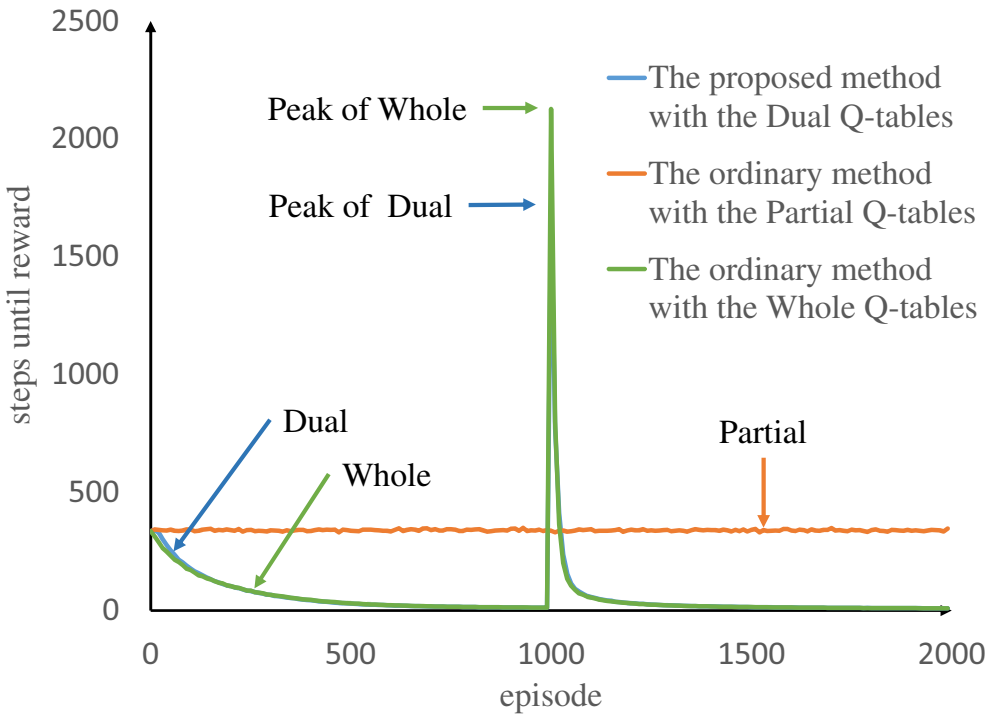


Fig. 3.8　シミュレーション実験結果：ケース 3

動を決められずランダムに行動を選択した結果である．提案手法は，学習初期から部分空間 Q-table に頼り知識から推論し，行動を選択するため，全体空間 Q-table のみの従来法よりも早く収束した．そして，全体空間 Q-table でしか表現できない環境は全体空間 Q-table に頼り，行動を選択するため，最適な値をとることができた．ケース 2 において，提案手法は，全体空間 Q-table のみの従来法よりも，効率良く学習できることを示した．

Fig. 3.8 にケース 3 のシミュレーション実験結果を示す．ケース 3 で，提案手法と全体空間 Q-table のみの従来法の 2 手法は最適な値を得ることができた．しかし，部分空間 Q-table のみの従来法は，最適な値を得ることができなかった．これは，全ての環境は全体空間でないと表現できないため，学習エージェントは適正行動を決

められずランダムに行動を選択した結果である．提案手法は，部分空間 Q-table より全体空間 Q-table を多く採用することで環境を学習することができた．ほとんど同じ学習曲線を描く，提案手法と全体空間 Q-table 手法との間で，環境変化において，提案手法が全体空間 Q-table よりも，騙されるピークを低くできたのは，部分空間 Q-table のおかげであると考えられる．これは，変化後の環境に適応している途中は，嘘を答える Q-table よりもランダムを答える方が役に立つためであると考えられる．ケース 3 において，提案手法は全体空間 Q-table の従来法とほぼ同じくらい有効であることを示した．

環境設定を代表する 3 つのケース，すなわち，学習環境の全てを部分空間で表現できるケース，学習環境の半分は部分空間で表現でき残り半分は確率的でないと表現できないケース，学習環境の全てを全体空間でないと表現できないケースにおいて，全体空間 Q-table のみの手法よりも，提案手法が劣ることはないことが検証できた．これにより，用意する部分空間 Q-table にかかわらず提案手法が全体空間 Q-table のみの手法よりも有用であることをシミュレーションにおいて示せた．

3.6　結言

本章では，提案手法における詳細空間を全体空間，粗い空間を部分空間とおき，最も基本的な多重化である全体空間に一つの部分空間を二重化する場合について，提案手法の学習効率をシミュレーション実験により検証した．用意した全体空間 Q-table と部分空間 Q-table に対して，(ケース 1) 部分空間 Q-table の状態表現の粗さでも十分表現できる場合，(ケース 2) 部分空間 Q-table では粗すぎ，全体空間 Q-table では細かすぎる場合，(ケース 3) 全体空間 Q-table の状態表現の細かさがないと学習できない場合，の条件でシミュレーション環境を変化させた．これらの条件は，部分空

間に観点をおくと，部分空間として，(ケース 1) 空間内の要素すべてが学習環境を表現していて有用なもの，(ケース 2) 一部の要素のみが学習環境を表現して有用なもの，(ケース 3) いずれの要素も学習環境を表現しておらず有用でないもの，に言い換えられる．つまり，本章では，提案手法の学習効率を，部分空間の有用性を変化させて検証した．結果より，提案手法では，部分空間が有用であるほど学習効率が向上する一方で，部分空間が有用でなくとも学習効率がほとんど低下しないことを確認した．

第4章　学習器の多重化

4.1　緒言

本章では，詳細空間に複数の粗い空間を多重化する場合について，提案手法の有効性をシミュレーション実験により検証する．最も詳細な空間の Q-table を全体空間 Q-table，それよりも粗い空間の Q-table を部分空間 Q-table とおく．全体空間 Q-table は，対象となる実験環境を十分正確に学習できるほど細かい状態をもち，部分空間 Q-table は，全体空間の状態を統合して学習に最低限必要な粗い空間表現をする状態空間をもつ．各 Q-table の行動の数と種類は互いに同じである．強化学習において，あるタスク達成のために必要十分な学習空間を求めることは難しく，複数の部分空間を多重化する際にも，有用な空間ばかりが設定できるとは限らない．本章では，有用でない空間がさらに増えた場合や有用な空間と混在した場合の環境において，提案手法の学習効率をシミュレーション実験により検証する．

4.2　提案手法 (多重化版)

第 2 章で述べた提案手法を，全体空間に複数の部分空間を多重化する場合に適用する．学習アルゴリズムは，第 2 章で述べた NS チャート (**Fig. 2.3**) の通りである．

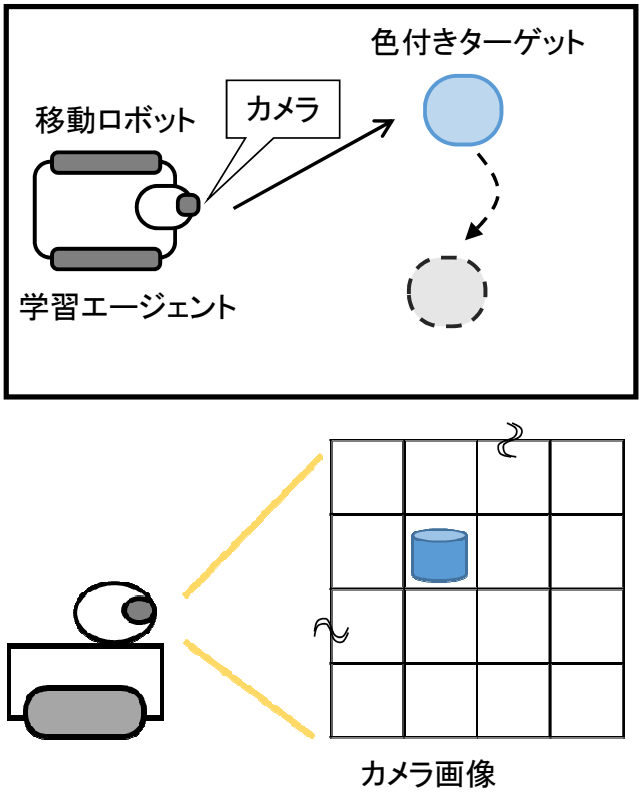


Fig. 4.1　シミュレーション環境

4.3　シミュレーション環境

ここでは，学習エージェントであるカメラ付きの移動ロボットが，その取得画像の情報に基づき行動することを想定する．ロボットのいる環境には，色つきのターゲットが常に一つのみ存在する．ターゲットの中心位置がカメラ画像を分割したうちの指定位置となるとき，ロボットに報酬が与えられる．ロボットが目標達成すると，ターゲットの位置と色がランダムに変化する (**Fig. 4.1**)．学習のための状態軸は，ターゲットの位置と色の 2 軸である．全体の学習空間 W は，位置軸と色軸に対する行動軸をもつ空間であり，基本の部分空間 Pc，Pp は，色軸に対する行動軸，位置軸に対する行動軸をもつ空間とする（**Fig. 4.2**）．ここで，色の種類数＝ 4，位

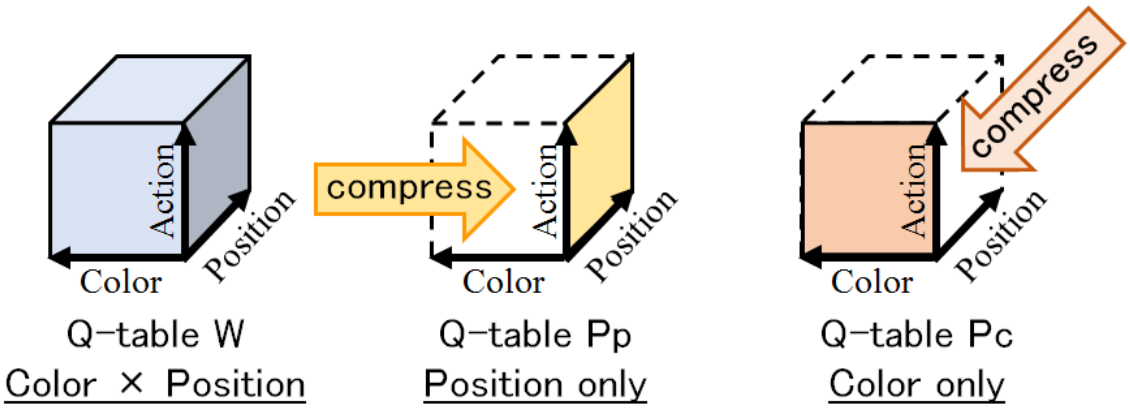


Fig. 4.2　全体空間 Q-table と部分空間 Q-table

置の種類数＝ 50，行動の種類数＝ 8 とする．目標達成時の各ターゲットの画像上の指定位置を **Fig. 4.3** に，ロボットの行動パターンを **Fig. 4.4** に示す． ターゲットの目標指定位置は色が 4 色のあるうち，2 色ずつを同じ位置にしており，どの色も画像上の中心位置の左右どちらかの位置にしているため，目標達成のために，ターゲットを画像の周辺から中心方向に移動させる行動戦略は，部分空間 Pp のみで表現できる．また，ロボットは，ターゲットが指定位置に写るように移動する必要があり，ターゲットの位置を無視する部分空間 Pc のみでは，目標達成のための行動戦略を表現できない．言い換えると，部分空間 Pp は有用であり，部分空間 Pc は有用でない．本実験では，さらに，有用でない部分空間 Pc の部分空間 $Pcc01$，$Pcc02$，$Pcc03$ を有用でない学習空間として用いる．$Pcc01$〜03 は，Pc の色軸の表現数を半分にしたものである．Pc では，[0,1,2,3] と 4 種類表現できるのに対して，$Pcc01$〜03 では，[0,1] の 2 種類とする．$Pcc01$ では，Pc において物体の色が 0 または 1 の場合は 0 に，2 または 3 であれば 1 と表現する (**Fig. 4.5**)．$Pcc02$ では，Pc における 0，2 を 0 に，1，3 を 1 と表現し，$Pcc03$ では，Pc における 0，4 を 0 に，1，2 を 1

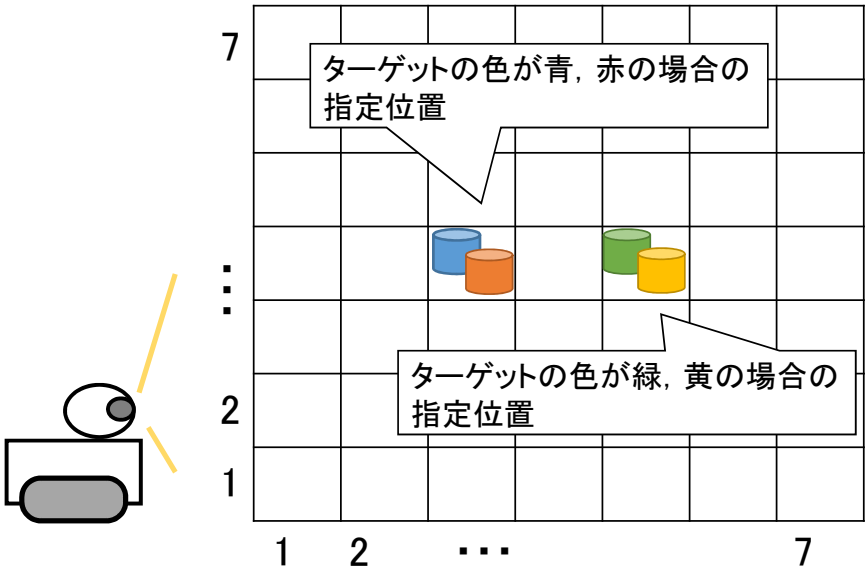


Fig. 4.3　目標状態

と表現する．学習に用いるパラメータは，Q 値の初期値を 0.0，報酬値 r を 1.0，割引率 γ を 0.8，学習定数 α を 0.08，ボルツマン選択の温度係数 T を 0.5 に設定する．

4.4　シミュレーション実験結果と考察

全体空間 W に有用でない部分学習空間 Pc，$Pcc01$〜$Pcc03$ をそれぞれ多重化した場合のシミュレーションの結果を **Fig. 4.6** に示す．グラフは，横軸を報酬が得られた回数，縦軸をロボットの累積ステップ数とした学習曲線であり，グラフの傾きの変化が学習進捗度合いを示す．グラフの傾きが一定なら，報酬ごとの増加ステップ数が一定であることを示し，学習が収束したと見なせる．学習収束までにかかるステップ数が少ないほど，効率的に学習したことを，また，学習収束後のグラフの傾きが小さいほど，次の報酬が得られるまでの増加ステップ数が少ないため，よい行動方策を学習したことを示す．

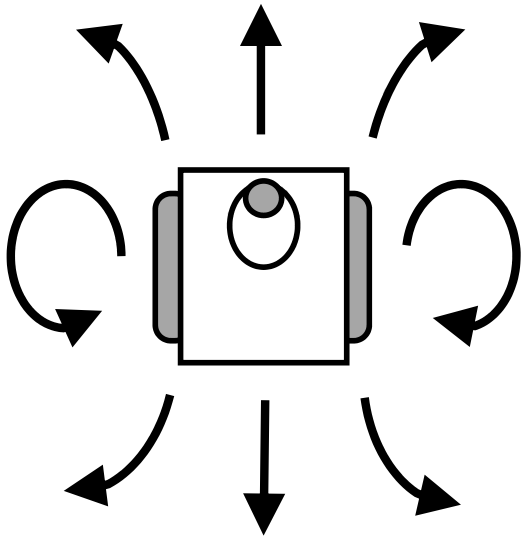


Fig. 4.4　自律移動ロボットの行動パターン

全体空間 W のみの学習が基本手法である．$W + Pc$，$W + Pcc01$，$W + Pcc02$，$W + Pcc03$ では，W と同等の行動方策が得られたものの僅かに学習効率が低下しており，多重化させた部分学習空間が有用でないことを確認した．また，**Fig. 4.6** の右図は，学習収束後の学習曲線を一部拡大したグラフである．その右図からは，$W + Pc$ は，$W + Pcc01$，$W + Pcc02$，$W + Pcc03$ よりも，学習効率が低下していることが読み取れる．これは，部分空間 Pc の状態数が $Pcc01$〜$Pcc03$ より多いためである．

次に，全体空間 W に有用でない部分学習空間 Pc，$Pcc01$〜$Pcc03$ を全て同時に多重化した場合のシミュレーション結果を **Fig. 4.7** に示す．$W + Pc + Pcc01$〜$Pcc03$ では，W と同等の行動方策が得られ，学習効率の低下は単独で多重化させた場合 $W + Pc$ とほぼ同等，つまり，学習の低効率化は累積されないという結果が得られた．

ここで，この結果を考察する．本件では，全体学習空間の次元数を固定した上で，有用でない空間として，色軸のみを持つ部分空間をさらに圧縮した部分空間を作成し

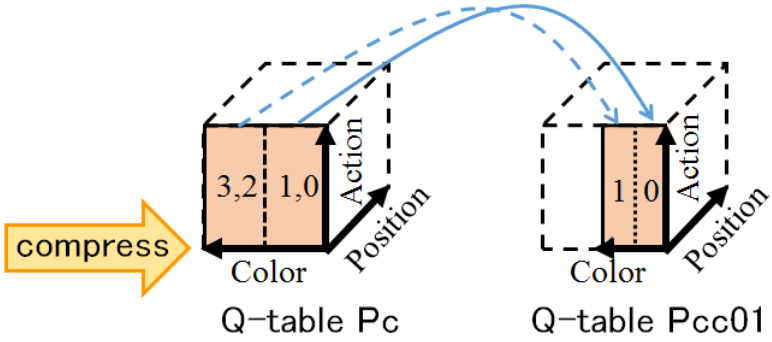


Fig. 4.5　有用でない部分空間の作成方法

た．これらの部分学習空間同士は類似性を持つ．我々は，多重化する部分空間同士が類似性を持つ場合，学習の効率化に与える影響が累積されないとの仮説を立てた．ある部分学習空間と類似性が最も高い学習空間は，それと同じ空間である．仮説検証のため，同じ部分学習空間を複数回多重化した場合のシミュレーション結果を **Fig. 4.8** に示す．$W + Pc$ に同じ部分学習空間 Pc をさらに 10 回多重化した $W + Pc + 10Pc$ は，$W + Pc$ と同じ結果，つまり，仮説の通り，学習の低効率化は累積されないという結果が得られた．

これらの結果について，本手法における Q 値の更新方法からも述べる．学習中の行動選択は，学習進捗度によって Q-table を選択し，その Q-table を用いて行う．Q 値の更新は，行動選択に用いた Q-table 以外の学習空間の Q 値も全て同時に実施する．これにより，学習空間の類似性が高ければ高いほど，同じように Q 値が更新され，学習が進む．従って，一方の学習空間の有用性が判断できるようになると，他方の学習空間でもそれができるようになるため，学習の低効率化が累積されないと考えられる．$W + Pc$ と $W + Pc + Pcc01$〜$Pcc03$ の比較においても，部分学習空間 Pc と $Pcc01$〜$Pcc03$ は類似性を持っており，低効率化が累積されない結果が得られたと考えられる．しかしながら，$W + Pc$ と $W + Pc + Pcc01$〜$Pcc03$ では，極僅か

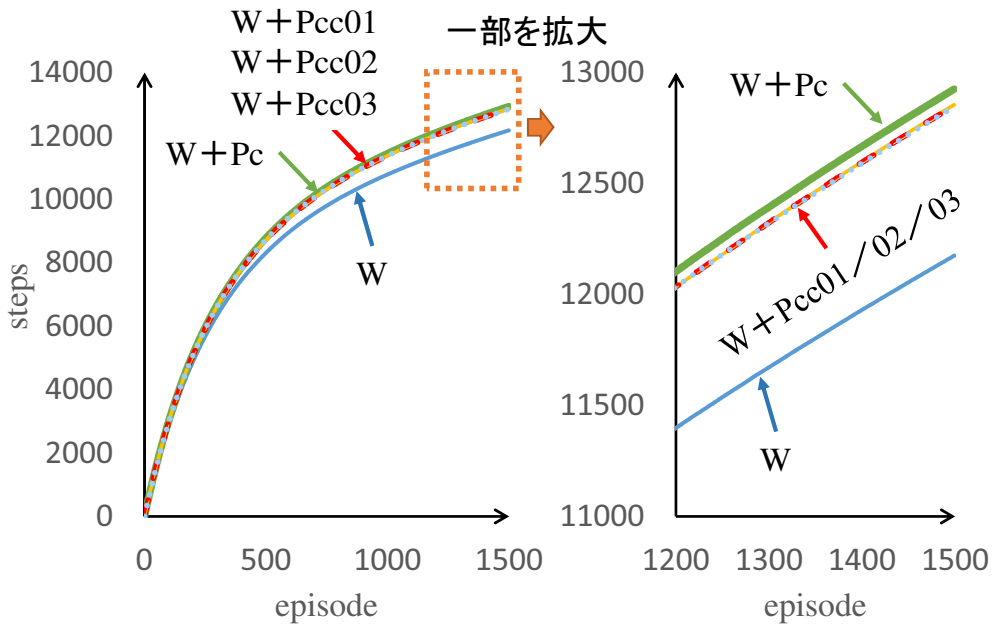


Fig. 4.6　シミュレーション実験結果：有用でない部分空間を多重化

に学習曲線は異なる結果となった．上記でも述べたように，本手法では学習進捗度によって試行行動を決定する Q-table を選択する．そのため，類似性は持っていても，学習進捗度の異なる部分学習空間が複数多重化された場合は，単独の場合とは異なる順序で試行行動が決定される可能性を持つ．学習中の試行行動順序の違いが，極僅かに学習曲線に現れたと考えられる．

ここで，有用でない複数の部分学習空間の他に，1 つの有用な部分学習空間を多重化した場合のシミュレーション結果を **Fig. 4.9** に示す．$W+Pc+Pcc01$〜$Pcc03+Pp$ では，有用でない部分学習空間が複数あっても，1 つでも有用な部分学習空間があれば，学習が大幅に効率化できることを示している．

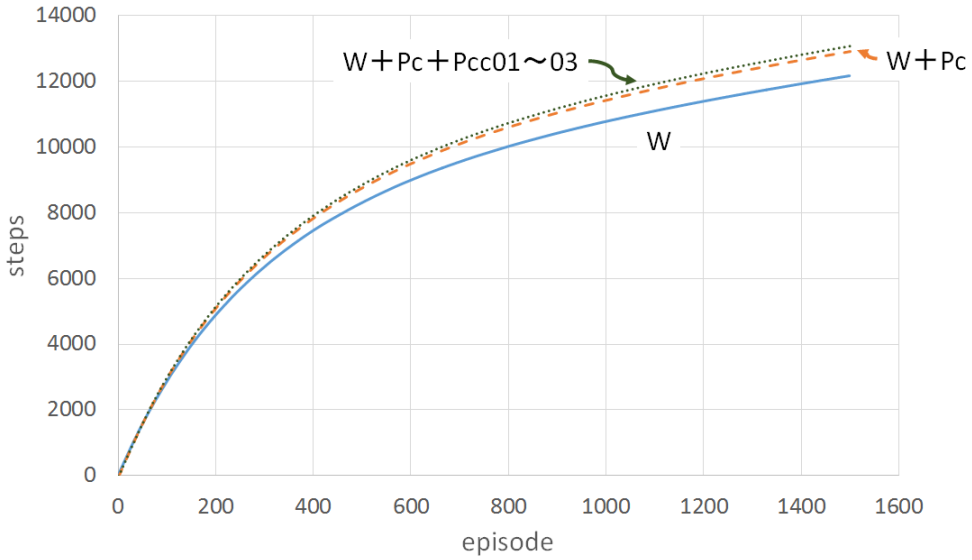


Fig. 4.7　シミュレーション実験結果：有用でない部分空間を複数多重化 (異なる部分空間)

4.5　結言

本章では，提案手法における詳細空間を全体空間，粗い空間を部分空間とおき，一般的な多重化である全体空間に複数の部分空間を多重化する場合について，提案手法の学習効率をシミュレーション実験により検証した．第 3 章の部分空間の有用性を変化させたシミュレーション実験結果では，全く有用でない部分空間を多重化させても，ほとんど学習効率は低下しないものの，若干の低効率化が見られていた．ここでは，有用でない部分空間がさらに増えた場合や，有用な空間と混在した場合の環境において，シミュレーション実験した．結果より，有用でない部分空間が複数あっても，それらが交互作用的に学習の低効率化を引き起こさないこと，一つでも有用な部分空間が含まれていれば，学習が効率化できることが確認できた．第 3 章で述べたことと考え合わせ，部分空間設計指針として,「用意した部分空間の中に

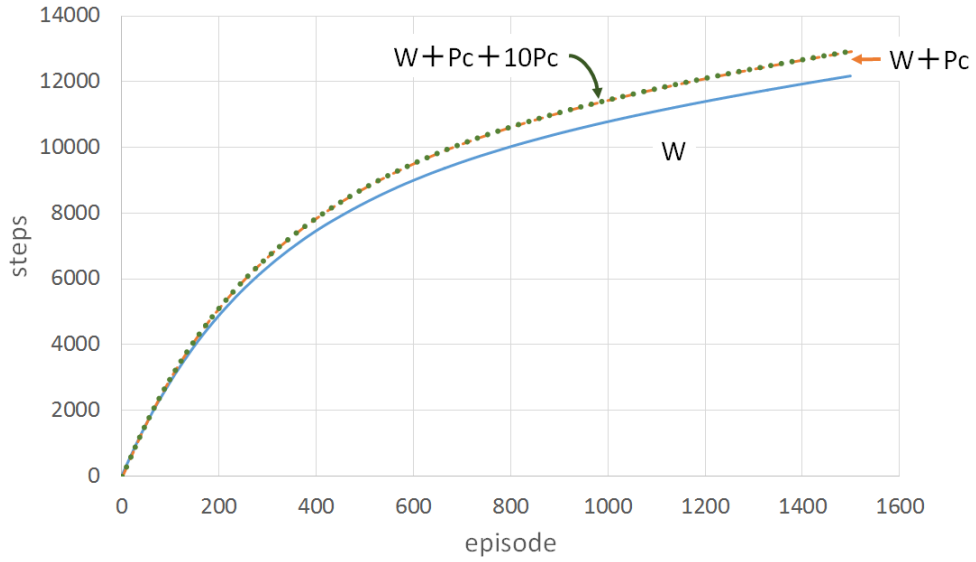


Fig. 4.8　シミュレーション実験結果：有用でない部分空間を複数多重化 (同じ部分空間)

有用でない空間が含まれていても学習効率は低下しない．その一方で有用な部分空間が含まれていればそれだけ学習効率は高くなる．したがって，予め部分空間の有用性を推測して，有用性の有無に応じて取捨選択する必要はない.」が導かれる．

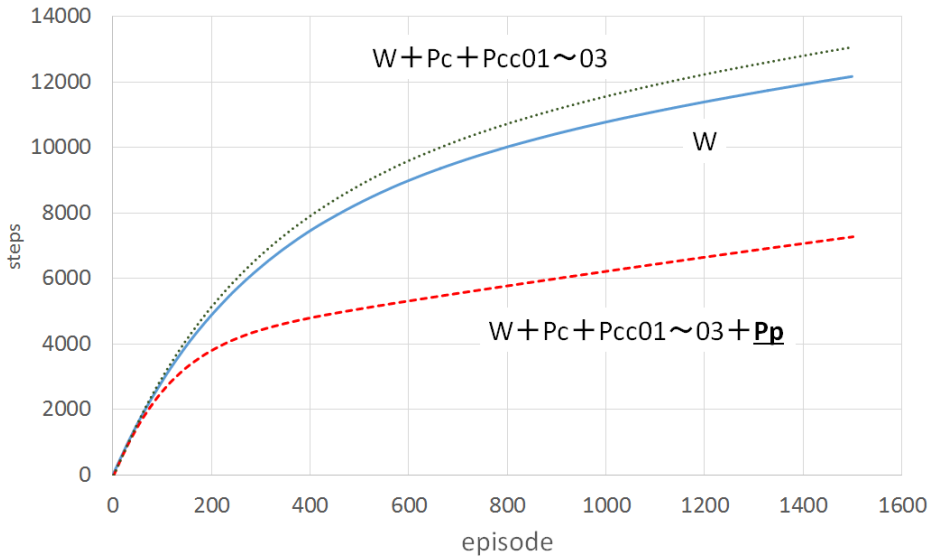


Fig. 4.9　シミュレーション実験結果：1 つの有用な部分空間と複数の有用でない部分空間を多重化

第5章　学習器の多重化(包含関係なし)

5.1　緒言

第3章や第4章では，包含関係をもつ詳細空間と粗い空間を用いる多重化について検証した．本章では，提案手法における複数の学習空間の多重化を，包含関係をもたない学習空間同士を用いる場合に応用し，提案手法の有効性をシミュレーション実験により検証する．強化学習では，一般に，報酬は目標達成時に即座に得られる．報酬付与に遅れがある場合，報酬の対象となる状態と行動の関係に不確定性が生じ，学習を効率的に進めることができない．具体例としては，プロセス制御システムにおけるむだ時間系や人とロボットとのインタラクションによる学習がある．この問題に対して，報酬付与の遅れ時間を事前実験にて予め計測し，その分前の時間の状態・行動に報酬を与える方法 [38] がある．また，行動決定から実行までの時間を長くとるなど，学習程度に応じて調整することで，報酬付与の遅れを緩和する方法 [39] もある．これらに対して，報酬付与の遅れ時間を考慮する学習器を多重化することにより，学習を効率化する手法を提案する．

5.2　提案手法(包含関係なし)

第2章で述べた提案手法を，包含関係を持たない学習空間同士を多重化する場合に適用する．学習アルゴリズムのNSチャートを **Fig. 5.1** に示す．包含関係を持た

Fig. 5.1　提案手法 (包含関係なし) の NS チャート

ない学習空間同士の多重化では，Q-table の更新部分が単純化される．含有関係がないため，全ての学習空間を更新する．

5.3　シミュレーション環境

本シミュレーションでは，グリッドワールドである学習環境に学習エージェントとターゲットが 1 体ずつ存在し，常に同じ行動をとるターゲットを，エージェントが捕まえることを想定する．**Fig. 5.2** に，学習環境を示す．**Fig. 5.2** の左図のように，エージェントは上下左右と斜め方向に移動でき，**Fig. 5.2** の右図のように，エージェントとターゲットの位置が同じとなれば，目標達成とする．基本の学習空間は，

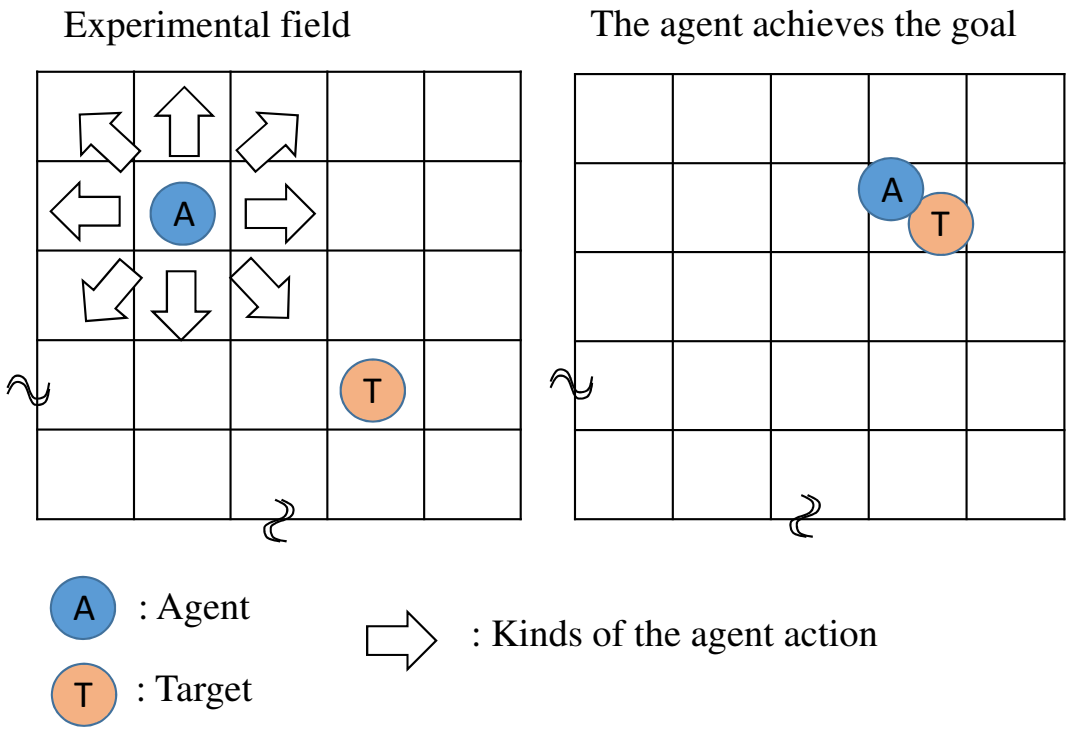


Fig. 5.2　シミュレーション環境

エージェントとターゲットの位置に対する行動軸を持つ空間とし，従来手法ではこの空間を 1 つのみ持つ．提案手法では，基本空間の他に，1 ステップ分，2 ステップ分，…，τ ステップ分の報酬付与遅れを考慮する学習空間を持つ (**Fig. 5.3**)．環境条件は，位置の種類数=50，行動の種類数=8，学習パラメータは，報酬 r=1.0，割引率 γ=0.8，学習定数 α=0.08，ボルツマン選択の温度係数 T=0.5 とする．

5.4　報酬付与の遅れ時間の条件

報酬付与の遅れとは，目標達成直後ではなく，そこから何ステップ分か遅れて報酬を受け取ることを意味し，報酬付与の遅れ時間とは，目標達成してから報酬を受

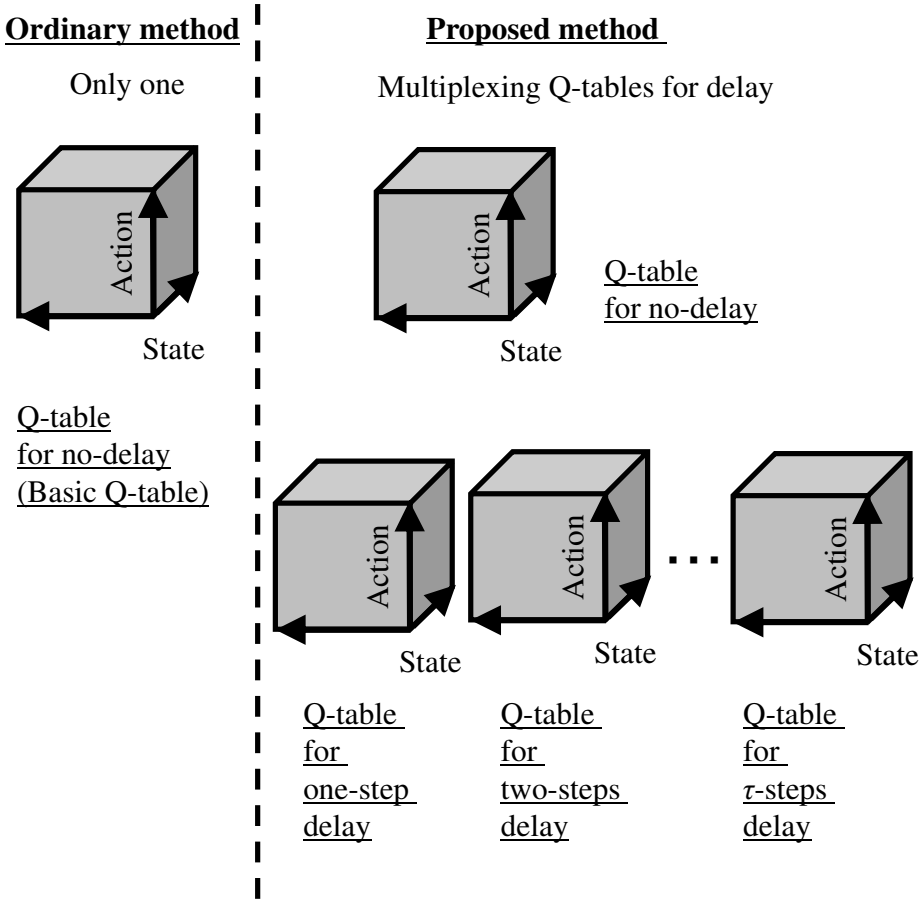


Fig. 5.3　報酬取得遅れを考慮する Q-table の多重化

け取るまでのステップ数のことを言う．報酬付与の遅れ時間が固定なケース 1 と，ランダムなケース 2 の 2 種類の環境でシミュレーション実験をし，本提案手法の特性を確認する．

ケース 1：遅れ時間が固定

この環境において，学習エージェントは，目標達成してから常に固定ステップ分遅れて報酬を受け取る．提案手法では，各遅れ時間を考慮する学習空間を多重化する．言い替えると，実際の遅れ時間を考慮する学習空間が多重化される環境である．

ケース 2：遅れ時間がランダム

この環境において，学習エージェントは，目標達成してからランダムステップ数分遅れて報酬を受け取る．ランダムステップとは，遅れのステップ数のパターンが複数あり，それらがランダムに切り替わることを意味する．提案手法では，各遅れ時間を考慮する学習空間を多重化する．言い替えると，実際の遅れ時間を考慮する学習空間が多重化されない環境である．

5.5　シミュレーション実験結果と考察

報酬付与の遅れ時間が固定のケース 1 と，ランダムなケース 2 のそれぞれについて，基本の学習空間の Q-table のみをもつ従来手法と，報酬付与の遅れ時間を考慮した学習空間の Q-table を多重化する提案手法のシミュレーション実験結果を示す．

Fig. 5.4 に，ケース 1 の場合の従来手法について，実験条件と結果を示す．グラフの横軸はエピソード数，縦軸は，エージェントが目標を達成するまでのステップ数であり，1000 回のシミュレーションの平均をとったものである．学習が進むにつれ，目標達成に必要なステップ数が減少し，それがほぼ一定となると学習が収束したと見なせる．学習収束までにかかるエピソード数とステップ数が少ないほど，効率的に学習したことを，また，収束時点で目標達成に必要なステップ数が小さいほど，良い行動方策を学習したことを示す．グラフの右側には，学習に用いる Q-table の図と，目標達成と報酬取得のステップを表にした学習条件を示す．表内の「(G)」が目標達成したステップ,「(R)」が報酬を受け取り，エピソードが終了したステップである．ここでは，報酬付与の遅れ時間が 1 ステップ，2 ステップで固定の場合と，比較のために報酬付与の遅れがない場合についてシミュレーションした．従来手法

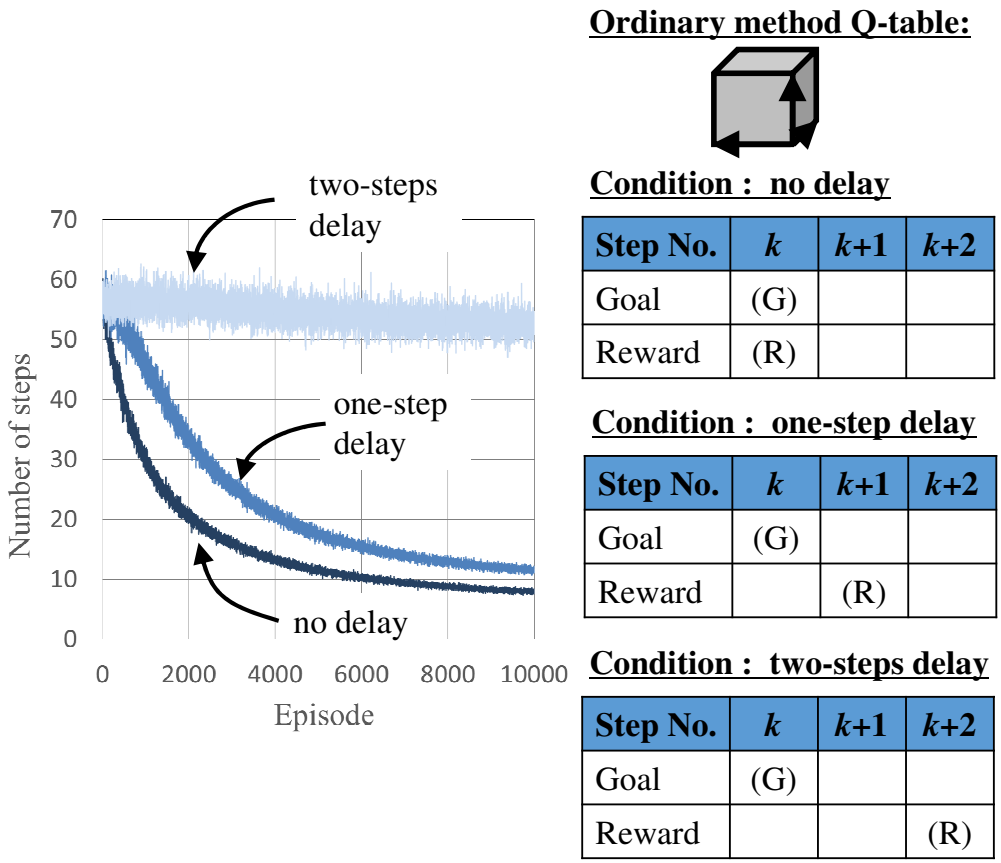


Fig. 5.4　シミュレーション実験条件と結果：ケース 1 の場合の従来手法

の学習結果より，報酬付与の遅れ時間が増加するに従い，学習が進まなくなっており，報酬付与遅れが学習に悪影響していることが分かる．次に，**Fig. 5.5** に，ケース 1 の場合の提案手法について，実験条件と結果を示す．提案手法では，遅れ時間の考慮なしの基本の学習空間 Q-table に，1 ステップ分，2 ステップ分のそれぞれの遅れ時間を考慮する学習空間 Q-table を多重化する．提案手法の学習結果より，報酬付与の遅れ時間が増加しても，遅れのない場合と同様に学習が進んでおり，悪影響を受けずに学習できることが確認できた．ケース 1 において，提案手法は，基本空間 Q-table のみの従来手法よりも効率良く学習できることを示した．ここで，報

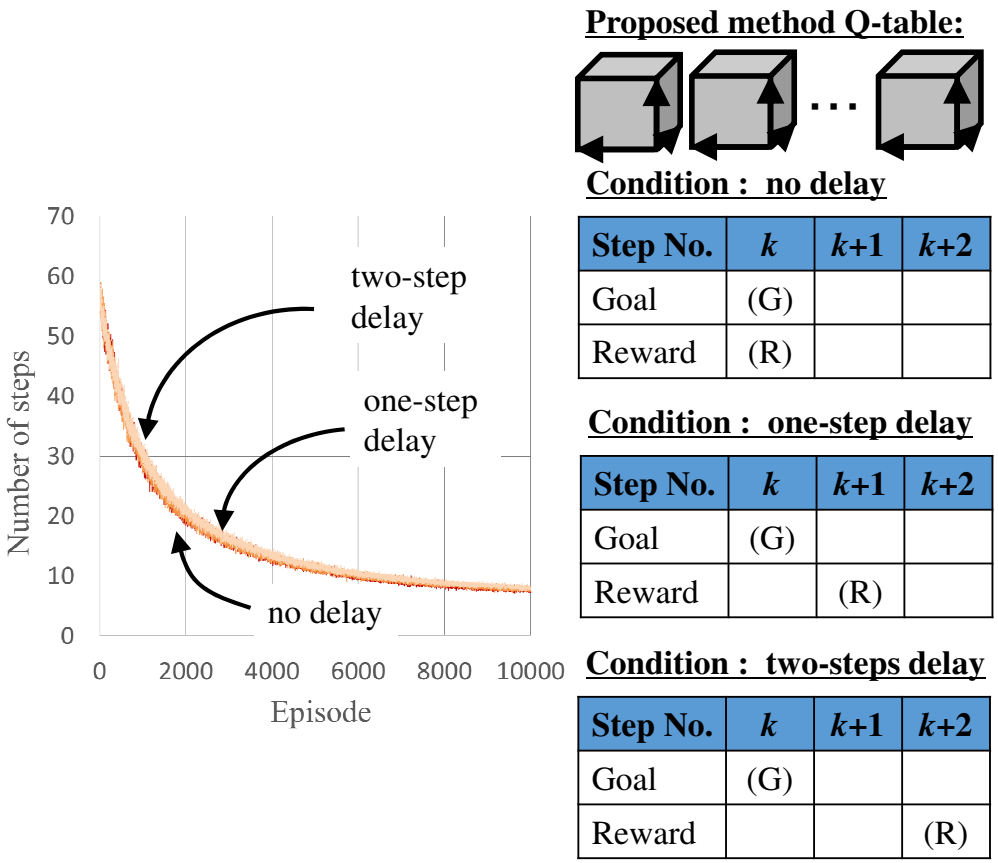


Fig. 5.5　シミュレーション実験条件と結果：ケース 1 の場合の提案手法

酬付与の遅れ時間が 2 ステップの場合の環境において，提案手法での Q-table の選択と平均情報量の変化を **Fig. 5.6** にて確認する．左側のグラフは，横軸をステップ数，縦軸を各 Q-table が選択された回数とし，100 ステップ毎に 1 つプロットしている．右側のグラフは，横軸をステップ数，縦軸を平均情報量とし，1 ステップ毎に 1 つプロットしている．左グラフより，学習が進むに従い，平均情報量が同じ場合にデフォルトで選択される遅れ考慮なしの Q-table から，環境の報酬付与遅れと同じ 2 ステップ分だけ遅れを考慮する Q-table に切り替わっており，提案手法において，適切な Q-table が選択されていることが確認できた．さらに，右グラフより，平均

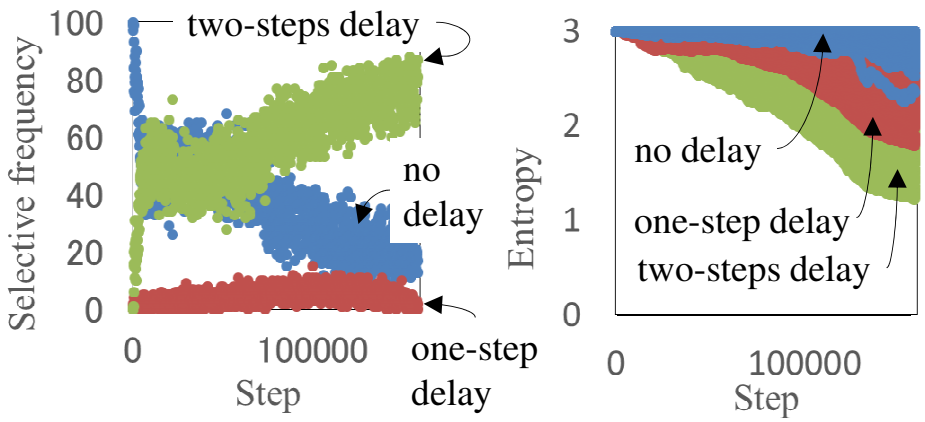


Fig. 5.6　シミュレーション実験結果：Q-table 選択時の平均情報量

情報量は，Q-table が考慮する遅れステップ数が，環境の報酬付与遅れ時間に近いほど小さく，Q-table 選択の指標としての有用性が確認できた．

Fig. 5.7 に，ケース 2 の場合の従来手法と提案手法について，実験条件と結果を示す．提案手法では，遅れ時間の考慮なしの基本の学習空間 Q-table に，1〜4 ステップ分のそれぞれの遅れ時間を考慮する学習空間 Q-table を多重化する．ここでは，報酬付与の遅れ時間が 2 ステップと 3 ステップでランダムに切り替わる場合と比較のために報酬付与の遅れがない場合についてシミュレーションした．実験結果より，従来手法はほとんど学習できないが，提案手法では報酬付与の遅れがない場合に比べると学習効率は低下するものの，学習が進んでいることが確認できた．ケース 2 においても，提案手法は，基本空間 Q-table のみの従来手法よりも効率良く学習できることを示した．

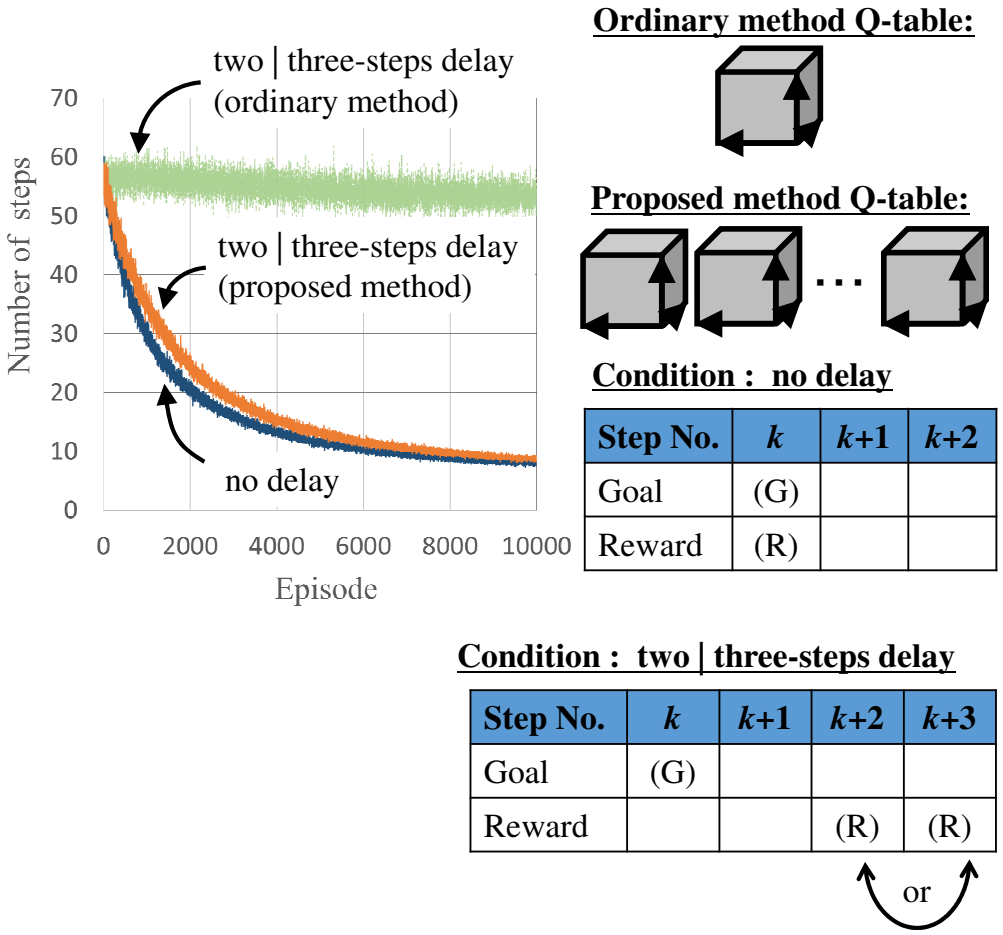


Fig. 5.7　シミュレーション実験条件と結果：ケース 2

5.6　結言

本章では，報酬付与の遅れに，遅れ時間毎の多重学習器を用いることで対応する強化学習を提案し，提案手法の有効性をシミュレーション実験により検証した．結果より，従来手法ではほとんど学習できない環境においても，提案手法では行動選択に適切な学習空間を選択して，学習することを確認した．また，学習空間選択に用いる平均情報量は，環境の遅れ時間に一致する学習空間ほど低くなり，指標として有用であることが確認できた．

第6章　実環境における検証

6.1　緒言

本章では，最も基本的な多重化パターンである全体空間に部分空間を一つ多重化する場合について，実機実験により，提案手法の有効性を検証する．コンピュータシミュレーションにおいて，実機で起こる現象をシミュレートしきれているわけではない．シミュレーションでは，学習ロボットの行動によって生じる環境変化やその認識が確定的あるのに対して，実機ではそれらが不確定である．そこで，本章では提案手法がコンピュータシミュレーションと同様の振る舞いをみせることを確認することで，提案手法が実環境でも有効であることを示す．第3章のシミュレーション実験と同様に，提案手法は，全体空間 Q-table と部分空間 Q-table の 2 つを用い，それぞれの Q-table は行動の数と種類が同じで状態の数と種類が異なる．シミュレーション実験のケース 2 のように，学習環境の一部が部分空間のみで表現できる場合について実験する．全体空間 Q-table のみと部分空間 Q-table のみの単純な従来手法の 2 つと提案手法の学習効率を比較し，シミュレーションの結果と同様に提案手法の有効性を示す結果が得られるかを検証する．

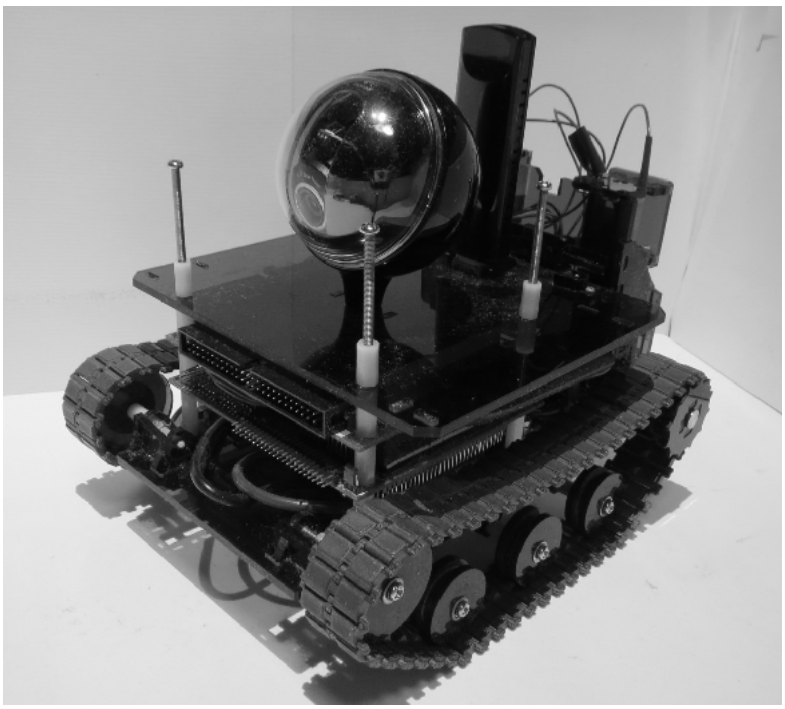


Fig. 6.1　自律移動ロボット：MieC

6.2　実機実験環境

実機は，**Fig. 6.1** に示す自律移動ロボット MieC を用いる．MieC は三重大学大学院工学研究科機械工学専攻メカトロニクス研究室で開発された自律移動ロボットで，移動機構として 2 本の無限軌道を用いる．2 本の無限軌道は，2 つのモータにより，それぞれ独立に駆動される．外部センサとしては，CCD カメラを搭載している．外部通信には無線 LAN を用いる．さらに，CPU カードと FPGA カードを搭載しており，画像処理などは CPU カードが処理を担当し，モータ制御などの処理は FPGA カードが担当する．ロボットが動作する環境は，大きさ 1.20[m] × 0.91[m] のプラ船を用い，内側には白色のプラスチック版を張り付ける．タスクは，ロボットの動作環境に表示された色を持つターゲットを，ロボットのカメラ画像の指定位置

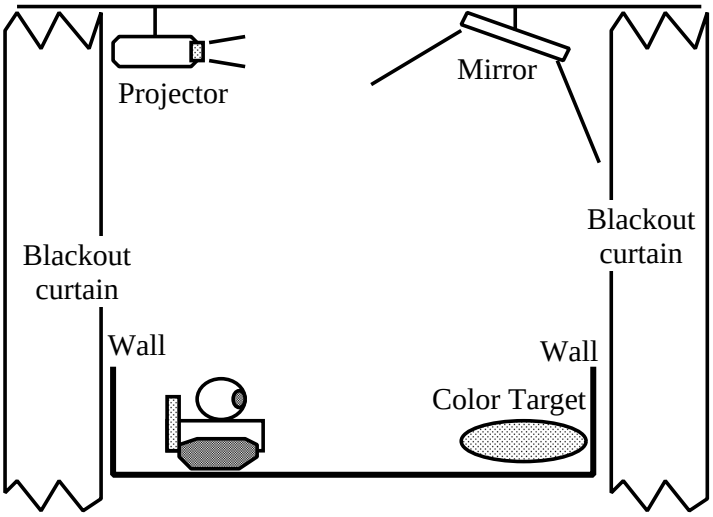


Fig. 6.2　実機実験環境

で認識することを目標とする．ターゲットは，**Fig. 6.2** のようにプロジェクターと鏡を用いて，動作環境に投影することで表示し，動作環境に 1 つのみ存在する．ロボットが目標達成すると，ターゲットの投影位置は規則的に，色は不規則に変化させる (**Fig. 6.3**)．ターゲットの投影位置はロボット動作環境の四隅，色は赤・青・緑・黄色の 4 色のいずれかである．**Fig. 6.4** に，実験の様子を実機の上部から撮影した写真を示す． 報酬の取得状態について説明する．報酬は，ロボットのカメラ画像において，ターゲットの中心位置がカメラ画像を分割したうちのあらかじめ決められる位置となるとき，ロボットに与えられる．**Fig. 6.5** のように，カメラ画像の分割数は縦 3 ×横 5 とし，左上から順に番号を割り振る．実機実験においても，学習途中での目標達成条件を変更して環境を変化させる．849 エピソードまでの環境変更前は，赤色・青色のターゲットは 6 番の位置に，緑色・黄色のターゲットは 8 番の位置に映ったときに，850 エピソード以降の環境変更後は，赤色・青色のターゲットは 8 番の位置に，緑色・黄色のターゲットは 6 番の位置に映ったときに，ロボッ

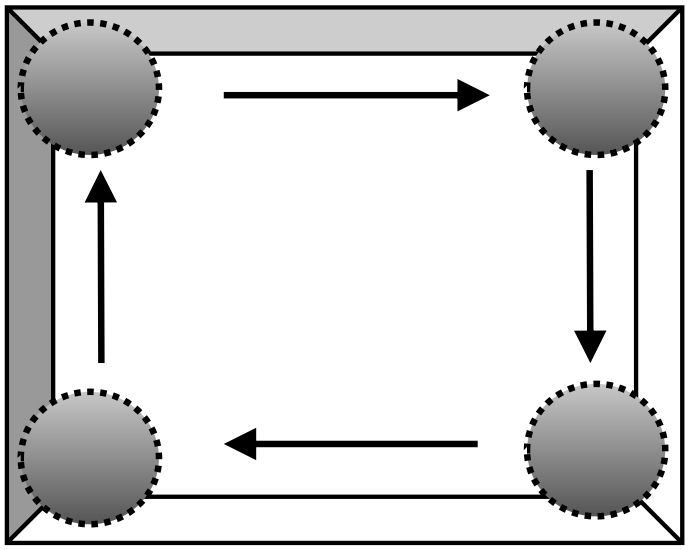


Fig. 6.3　色付きターゲットの移動パターン

トは報酬を得る (**Fig. 6.6**)．カメラ画像の横方向の分割については，カメラ画像取得処理の遅れ等で各色の目標位置が隣り合う状態と誤認識されないように，つまり，7 番の位置を確実に認識させるため，横幅を他の 3 倍としている．縦方向の分割については，上下方向動作の学習難易度を上げるため，目標位置 6，8 番を含む 5〜9 番の高さを他の 2 分の 1 としている．ロボットの行動集合は，**Fig. 6.7** の矢印のパターンで示すように，8 つの行動で構成する．動作速度は一定とし，その行動により，環境が別の状態に移るまで続け，移るまでを一ステップとする． 上記の環境において，全体空間と部分空間を以下のように定義する．全体空間と部分空間共に位置情報として画像分割数の 15 に 1 を加えるのは，ターゲットが画像に映らない状態を表現するためである．

- 全体空間 Q-table：

  ロボットのカメラ画像にターゲットが映る位置 (15+1) とターゲットの色 (4)，ロボットの行動 (8) の 3 軸で表現する．

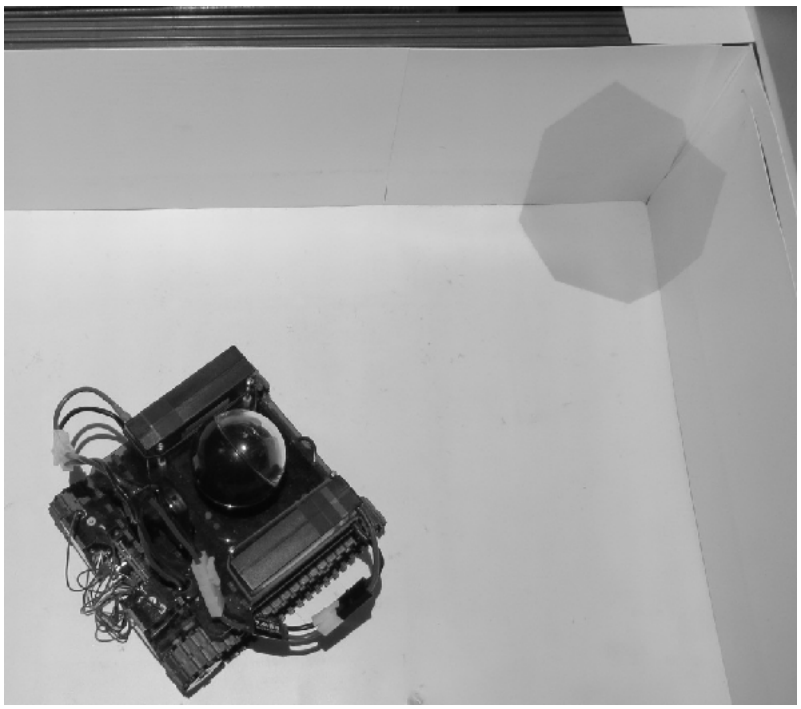


Fig. 6.4　実験環境の様子

・部分空間 Q-table：

ロボットのカメラ画像にターゲットが映る位置 (15+1)，ロボットの行動 (8) の 2 軸で表現する．

第 3 章のシミュレーション実験と同様，全体空間を用いれば，各状態ごとに適正行動を学習することができるが，部分空間のみを用いると，確率的にしか適正行動を学習できない．また，ターゲットの目標位置は色が 4 色のあるうちの 2 色ずつを同じ位置にしており，どの色も画像の中心位置の左右どちらかの位置にしているため，目標達成のために，ターゲットをカメラ画像の周辺から中心方向に移動させる行動戦略は，部分空間のみで表現できる条件となる．4 色それぞれを区別する全体空間では細かすぎ，色を区別できない部分空間では荒すぎる条件である．本実験では，第

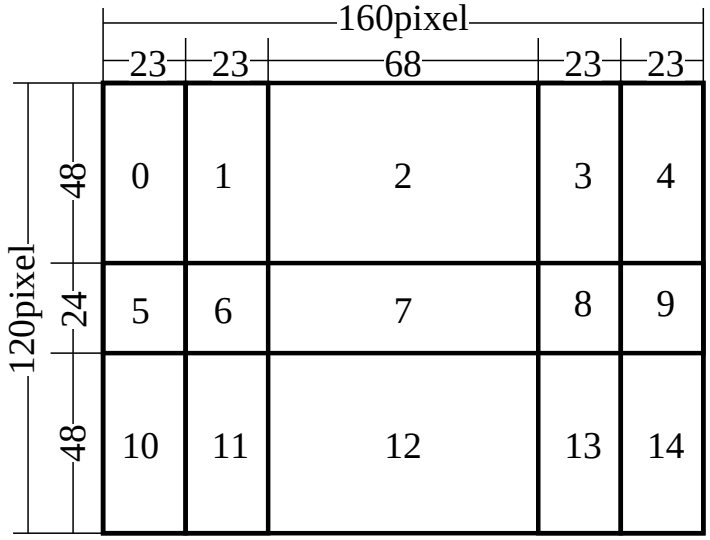


Fig. 6.5　カメラ画像における状態分割

3 章のシミュレーション実験のケース 2 と同等の条件において，提案手法の振る舞いが同様であることを確認することで，実環境での有効性を検証する．学習空間の Q 値の初期値を 0.0，報酬 r は正の報酬を 1.0，負の報酬を-1.0，学習率 α を 0.1，減衰率 γ を 0.8 に設定する．ロボットが動作環境であるプラ船の壁にぶつかって動作し続けるのを防ぐため，一定時間行動しても状態が推移しない場合に負の報酬を与える．行動選択手法として Boltzmann 選択を用い，温度 T を 0.1 に設定する．

6.3　実機実験結果と考察

部分空間 Q-table と全体空間 Q-table の 2 つの Q-table を持つ提案手法と部分空間 Q-table のみの従来法と全体空間 Q-table のみの従来法との比較結果を示す．グラフは，それぞれの学習法で 1700 エピソードまで 3 試行分実施し，ステップ数の累積値の平均をとった結果である．環境変化として，850 エピソードからターゲットの目

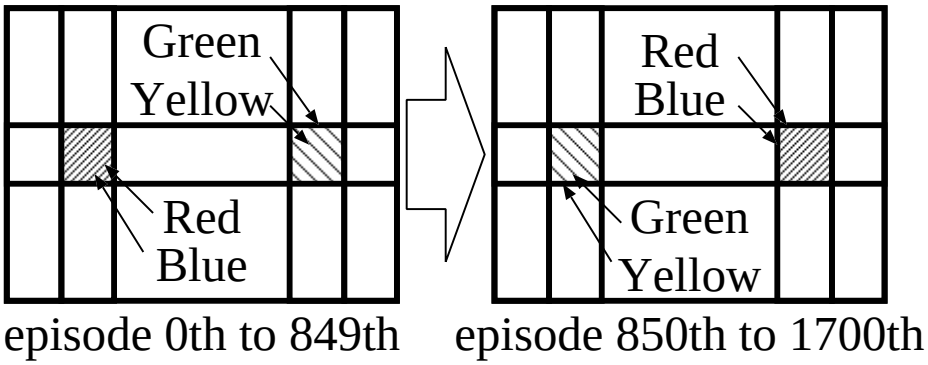


Fig. 6.6　目標状態

標認識位置の変更を実施した．

Fig. 6.8〜**Fig. 6.10** に実験結果を示す．横軸はエピソード数，縦軸は行動ステップ数の累積を示す．**Fig. 6.8** が全エピソード分，**Fig. 6.9** が 0 から 300 エピソード分，**Fig. 6.10** が 850 から 1700 エピソード分で，かつ，学習状況を比較しやすくするために 850 エピソードからステップ数の累積値をカウントしたグラフである．ステップ数の累積値のため，グラフの最終的な傾きが目標達成に必要なステップ数，傾きが一定になるまでが学習収束の速さを示す． まず，環境変化前の 849 エピソードまでの結果を考察する．最終的なグラフの傾きより，提案手法と全体空間 Q-table のみの従来法の 2 手法は最適な値を得ることができたが，部分空間 Q-table のみの従来法は，速く収束するものの最適な値を得ることができなかった．これは，ターゲットの色に関係なくカメラ画像の中心方向，目標位置近くに移動させる適正行動は速く獲得できるものの，色によって指定された目標位置に移動させる適正行動が確率的にしか決められないためである．提案手法は学習初期は部分空間を用い，全体空間でしか表現できない環境は全体空間を用いることで，全体空間のみの従来法よりも速く収束し，かつ，最適な値を得ることができた．次に環境変化後の 850 エピソード以降の結果を考察する．ターゲットの色を区別しない部分空間 Q-table の

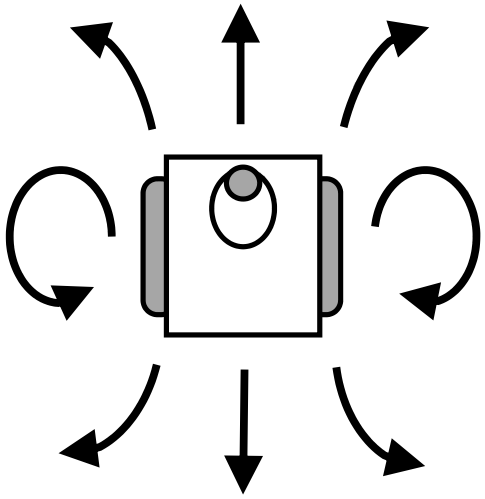


Fig. 6.7　自律移動ロボットの行動パターン

みの従来法にとっては，今回の環境変化が起こってもその環境変化を認識できない，つまり関係ないため，グラフの傾きに変化はなかった．環境変化後も環境変化前と同様に，提案手法の方が速く収束し，かつ，最適な値を得ることができた．環境変化前と後のどちらも，収束の速さと結果の正確性について，提案手法と 2 つの従来手法は，第 3 章のシミュレーション実験のケース 2 と同様の結果となった．つまり，実機実験においても，提案手法は，有用性があるといえる．

6.4　結言

本章では，最も基本的な多重化パターンである全体空間に部分空間を一つ多重化する場合について，実機実験により，提案手法の有効性を検証した．結果より，提案手法が実環境でもシミュレーションと同様に振る舞い，学習効率を向上させることを確認した．

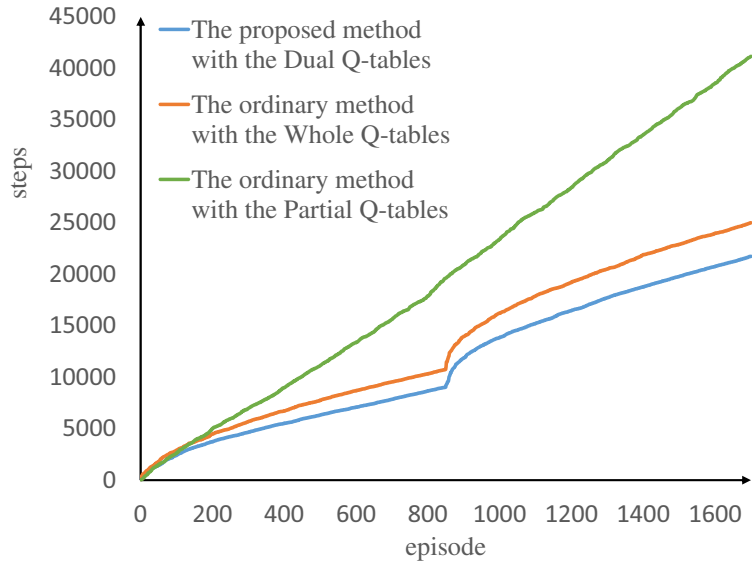


Fig. 6.8　実機実験結果：0 から 1700 エピソードまでのステップ数累積値

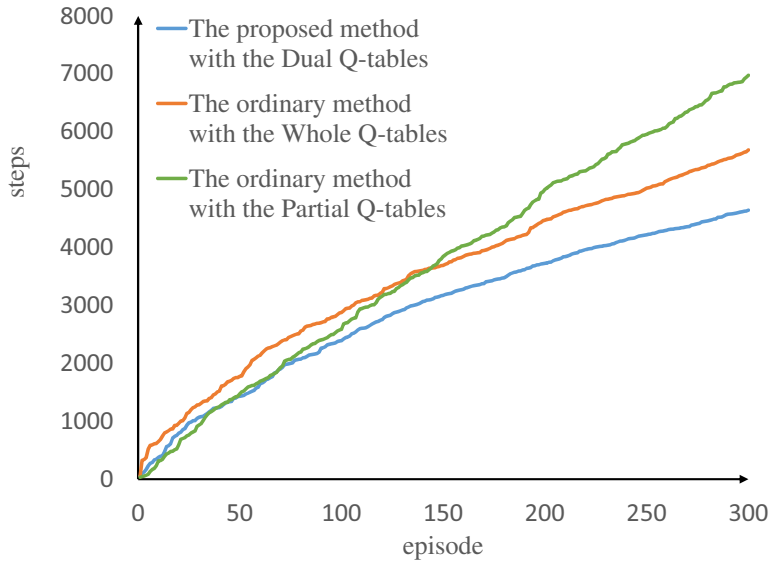


Fig. 6.9　実機実験結果：0 から 300 エピソードまでのステップ数累積値

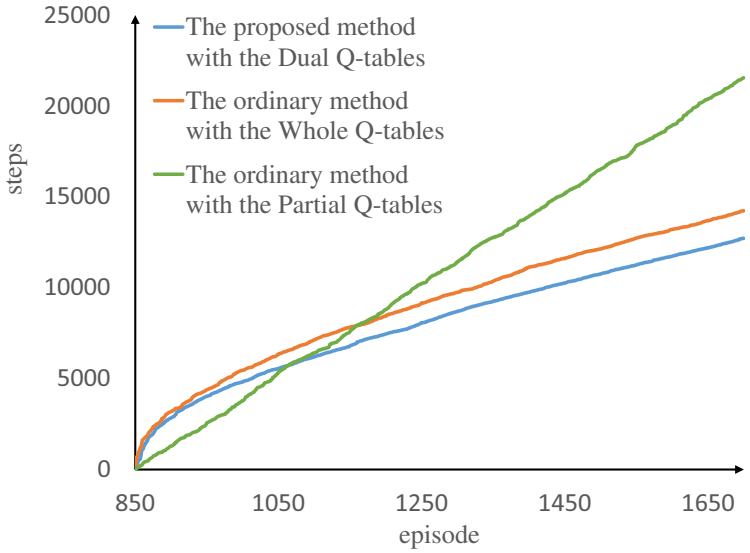


Fig. 6.10　実機実験結果：850 から 1700 エピソードまでのステップ数累積値 (850 エピソード時からの累積値)

第7章　数理解析

7.1　緒言

本章では，提案手法の学習の収束速度に着目し，第3章で用いた3つの学習空間において従来手法と比較する．ケース1では部分空間のみの従来手法に対して全く遅れがないこと，ケース3では全体空間のみの従来手法に対してほとんど遅れがないこと，ケース2ではケース1とケース3の間になることを説明する．本解析では，ケース1と2についてはアルゴリズムの流れからの説明のみとし，ケース3のみ数理的に説明する．また，本論文では学習環境の変化が起きる前までについてを解析する．

7.2　学習環境を部分空間のみで表現できる場合

ケース1における提案手法の学習の収束速度が，部分空間のみの従来手法に対して，全く遅れがないことをアルゴリズムの流れから説明する．

i，jは，適切な整数，kは，jとは異なる整数とする．(2.3)で示すように，部分空間のある一つの状態${}^{p}S_i$が，全体空間の複数の状態${}^{w}S_j$，${}^{w}S_k$，...に対応すると考える．ケース1の場合，部分空間の状態${}^{p}S_i$と全体空間の状態${}^{w}S_j$，${}^{w}S_k$，... の適正行動が一致する．ゆえに，部分空間と全体空間のそれぞれの状態において，Q値の期待値は，その状態を経験する回数が同じならば一致する．また，Q値の期待値は，

経験する回数が多いほど，Q 値の収束値に近づく．経験する回数が多いほど，その状態における適正行動の Q 値が大きくなり，Q 値の偏りが大きくなる．部分空間の状態 ${}^{p}S_i$ を経験する回数は，全体空間のそれぞれの状態 ${}^{w}S_j$，${}^{w}S_k$，... を経験する回数の和に等しい．ゆえに，部分空間の状態 ${}^{p}S_i$ の経験回数は，どの全体空間の状態 ${}^{w}S_j$，${}^{w}S_k$，... よりも常に多く Q 値の偏りが大きい．Q 値の偏りが大きくなると，Q 値から計算する行動の選択確率も偏りが大きくなる．提案手法においては，行動選択確率の偏りが大きい空間を用いて行動を選択するため，ケース 1 の環境では，常に部分空間が行動選択に用いられるので，実質部分空間のみの従来法と同じ枠組みになり，同じ速さで収束する．

7.3　学習環境を部分的に部分空間で表現できる場合

ケース 2 における提案手法の学習の収束速度が，ケース 1 とケース 3 の間になることを簡単に説明する．

部分空間の状態 ${}^{p}S_i$ が全体空間の状態 ${}^{w}S_j$，${}^{w}S_k$，... に対応するとする．また，状態 ${}^{w}S_j$ の方が，${}^{w}S_k$ より先に学習するとする．ケース 2 においては，部分空間 ${}^{p}S_i$ と全体空間の状態 ${}^{w}S_j$ に対応する状況で学習した結果が，部分空間 ${}^{p}S_i$ と全体空間の状態 ${}^{w}S_k$ に対応する状況において必ずしも適正行動となるわけではない．適正行動でない場合は，全体空間の状態 ${}^{w}S_k$ で再学習する必要があり，学習効率はケース 1 より下がる．また，部分空間 ${}^{p}S_i$ と全体空間の状態 ${}^{w}S_j$ に対応する状況で学習した結果が，部分空間 ${}^{p}S_i$ と全体空間の状態 ${}^{w}S_k$ に対応する状況において必ずしも不適正な行動となるわけでもない．適正行動となる場合は，部分空間の学習結果を用いて，全体空間の状態 ${}^{w}S_k$ で適合行動が学習強化され，学習効率はケース 3 より上がる．ゆえに，学習速度はケース 1 と 3 の間になる．

7.4　学習環境を部分空間では表現できない場合

ケース 3 における提案手法の学習の収束速度が，全体空間のみの従来手法に対してほとんど遅れがないことを説明する．ケース 3 の学習環境において，部分空間はすべての状態において適正行動が学習できないため，部分空間に誤誘導される分だけ，提案手法は全体空間のみからなる従来手法よりも学習の収束が遅くなる．ここでは，提案手法の行動選択空間が部分空間から従来手法と同じ全体空間に切り替わり，その間の誤誘導は学習初期に起きるのみであることを数理的に説明する．

7.4.1　学習空間

解析に用いる学習空間は 3 章と同様とするが，簡単化のため，セル数 $CellN$ は 1 個，各セル旗の種類 $ClrN$ は 2 種類，報酬取得のために上がる段数 $StageN$ は 1 段，すなわち，状態に対する適正行動を 1 回すると報酬が取得できる．**Fig. 7.1** に示すように，旗によって学習エージェントの適正行動が異なるため，部分空間では適正行動を学習できない．学習時の Q 値の初期値を 0，報酬 r を 1，学習率 α を 0.08，割引率 γ を 0.8，行動選択手法に Boltzmann 選択を用い，温度 T を 0.2 に設定する．

本解析では，提案手法とは異なるが，簡単化のため，以下の 3 つを実施する．1 つ目は，本設定では任意の状態から 1 ステップの適正行動で報酬が与えられるので，ほとんど影響のない Q 値の更新式の次状態の価値を 0 とする．2 つ目は，誤誘導を維持することとなるが，部分空間の誤誘導により選択行動が失敗しても選択確率を変更しない．ただし，実際には行動選択失敗により Q 値は低下し，平均情報量は変化するので，目標達成時においては行動選択空間の切り替わりを考慮する．3 つ目は，最終的な行動選択空間の切り替わり判断は，平均情報量で比較する本手法より悪く見積もり，部分空間における各行動の目標達成回数が一致するときとする．

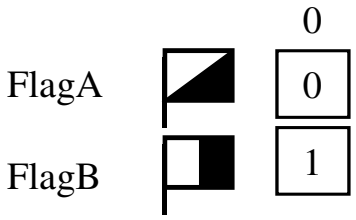


Fig. 7.1　数理解析時のシミュレーション環境

7.4.2　切り替わり時の Q-table

1 エピソード目は，全体空間，部分空間とも全 Q 値が初期値で平均情報量が同じため，提案手法における行動選択には部分空間 Q-table が用いられる．本解析では，1 エピソード目に FlagA の状況にて目標達成した後，2 エピソード目以降で誤誘導されながら，全体空間の全ての状態において行動選択空間が部分空間から全体空間に切り替わるエピソード回数の期待値を算出する．1 エピソード目で FlagB の状況にて目標達成した場合においても，以降の説明で FlagA と FlagB が入れ替わるのみで算出方法は同じである．

提案手法においては，2 エピソード目以降の目標達成するまでの旗の系列は，行動選択空間の切り替わり方により 3 つに分類される．3 つの系列は，正規表現である，”.”は 1 つの記号，”+”は直前の記号が 1 回以上，”∗”は 0 回以上続くことを示すものを用いると，FlagA が立つ状況のみを経験して目標達成する ($A+$ 系列)，FlagB が立つ状況を 1 回は経験して最後は FlagA で目標達成する ($.*B.*A$ 系列)，最後は FlagB で目標達成する ($.*B$ 系列) と表現される．($A+$ 系列) の場合，部分空間にて高い Q 値を持つ行動 0 の失敗が一度も起こらず，部分空間の平均情報量は変化しないため，目標達成時の行動選択空間は部分空間であり，全体空間の Q 値のみでなく，部分空間の Q 値が強化される．($.*B.*A$ 系列) の場合，FlagB の状況にて部分空間で高い Q 値を持つ行動 0 の失敗が起き，部分空間の平均情報量が増加するため，目

標達成時の行動選択空間は全体空間である．この場合，部分空間の Q 値は更新されず，FlagA の状況，つまり全体空間の一部の状態において行動選択空間が全体空間に切り替わる．そのため，これ以上 FlagB の状況にとって誤誘導となるように部分空間が強化されることはない．$(.*B$ 系列$)$ の場合，部分空間と比べて，FlagB の状況において全体空間には高い Q 値をもつものもなく Q 値の偏りは小さいため，目標達成時の行動選択空間は部分空間である．この場合，部分空間の Q 値が更新され，行動 1 の Q 値が行動 0 に近づき偏りが減る．この 3 種類の系列の繰り返しにより，やがては，全体空間の全ての状態において，行動選択空間が部分空間から全体空間に切り替わる．具体的には，1 エピソードは，必ず $(.*A$ 系列$)$ であり，FlagA の状況で部分空間の Q 値が強化されなくなるのは，エピソード系列：$(.*A$ 系列$)$ $((A+$ 系列$))*$ $((.*B.*A$ 系列$)\mid(.*B$ 系列$))$ のときである．ここで，”|”は”または”を意味する．このエピソード系列の後，1 エピソード目とエピソード系列中の $(A+$ 系列$)$ の回数分だけ $(.*B$ 系列$)$ で報酬を得るまでは，FlagB の状況において行動選択空間が部分空間から全体空間へ切り替わらず，誤誘導される可能性がある．具体例として，2 エピソード終了時の Q-table 更新結果を **Fig. 7.2** に示す．図中の $Q(N_{Goal})$ は，N_{Goal} 回目標達成して更新された Q 値を，P_{Goal}(旗の系列) は，旗の系列を経験して目標達成する確率を示す．図中の $(.*A$ 系列$)$ $(A+$ 系列$)$ 後では，部分空間の Q 値は強化され，FlagB の状況で 2 回報酬を得るまで行動選択空間が部分空間になり，FlagB の状況において行動 0 が適正行動であると誤誘導される．さらに，3 エピソード目が $(A+$ 系列$)$ だと，FlagB の状況で 3 回報酬を得るまで誤誘導される．$(.*A$ 系列$)$ $(.*B.*A$ 系列$)$ 後では，FlagA の状況においては，行動選択空間が全体空間に切り替わる．これにより，部分空間はもうこれ以上誤誘導となる方向へ強化されず，FlagB の状況で 1 回報酬を得れば誤誘導されなくなる．$(.*A$ 系列$)$ $(.*B$ 系列$)$ 後では，FlagA と FlagB のいずれの状況，すなわち，全体空間の全ての状態において行

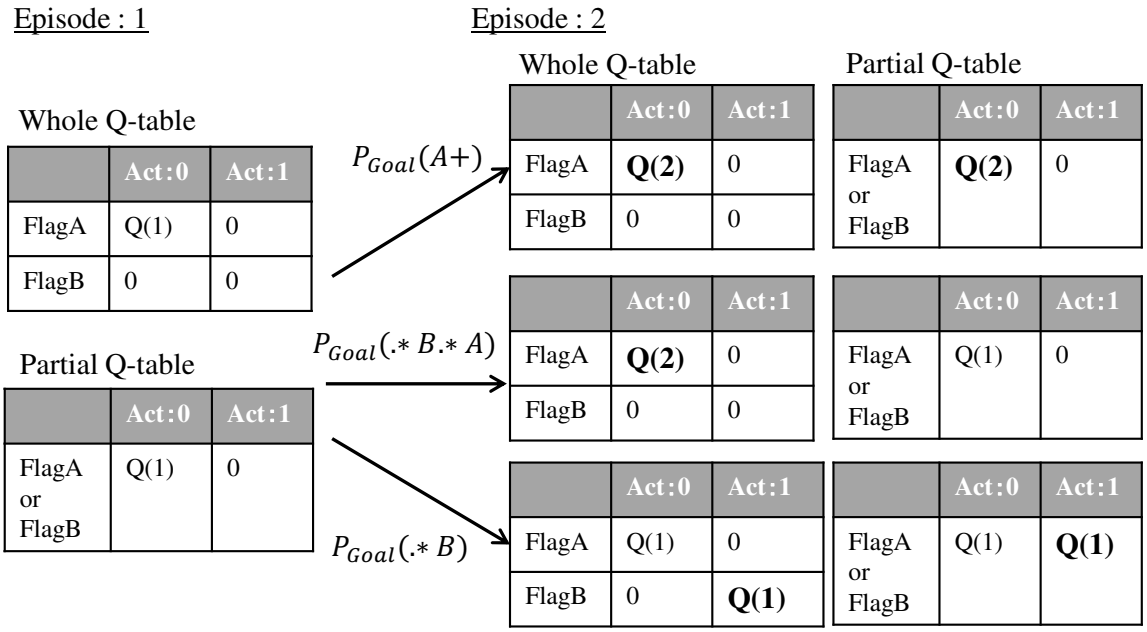


Fig. 7.2　二重 Q-table の更新

動選択空間が全体空間に切り替わる．ゆえに，もう誤誘導されない．各系列で目標達成する確率は無限級数の公式を用いて，(7.1)，(7.3)，(7.4) のように算出できる．

$$\begin{aligned} P_{Goal}(A+) &= P_A P(0|A) \sum_{k=0}^{\infty} \{P_A(1-P(0|A))\}^k \\ &= \frac{P_A P(0|A)}{1-P_A(1-P(0|A))} \end{aligned} \tag{7.1}$$

$$\begin{aligned} P_{Goal}(.*A) &= P_A P(0|A) \\ &\quad \sum_{k=0}^{\infty}(1-P_A P(0|A)-P_B P(1|B))^k \\ &= \frac{P_A P(0|A)}{P_A P(0|A)+P_B P(1|B)} \end{aligned} \tag{7.2}$$

$$P_{Goal}(.*B.*A) = P_{Goal}(.*A) - P_{Goal}(A+) \tag{7.3}$$

$$P_{Goal}(.*B) = \frac{P_B P(1|B)}{P_A P(0|A) + P_B P(1|B)} \tag{7.4}$$

$$P_{Goal}(B+) = \frac{P_B P(1|B)}{1-P_B(1-P(1|B))} \tag{7.5}$$

ただし，P_A は各ステップにおいて FlagA が立つ確率，P_B は FlagB が立つ確率，$P(0|A)$ は FlagA の状況にて適正行動 0 をとる確率，$P(1|B)$ は FlagB において適正行動 1 をとる確率である．ここでは，P_A=P_B=0.5 とする．

7.4.3　切り替わりエピソード回数の期待値

部分空間による誤誘導は，全体空間の全ての状態において行動選択空間が，部分空間から全体空間に切り替わるまで起こる可能性がある．そこで，簡単化のため，誤誘導されるエピソード回数の期待値を，初回から全体空間に切り替わるまでのエピソード回数の期待値として見積もる．切り替えは，1 エピソード目と $(A+)$ 系列でゴールし続けるエピソード回数の和 N_{AG} に対して，$(.*B)$ 系列で N_{AG} 回ゴールするエピソード数が必要である．これは，$(A+)$ 系列でゴールし続けないと，FlagA での行動選択空間が全体空間に切り替わり，その後は，$(.*B)$ 系列で最大同数ゴールすれば，FlagB での行動選択空間も全体空間に切り替わるので，全体空間だけで学習する従来手法と同じになるまでにかかるエピソード回数を求めるものである．$P_{Goal}($系列$)$ の確率は，N_{AG} の数にも依存するので，ここから $P_{Goal}($系列$, N_{AG})$ と記述する．すると，$(.*B)$ 系列で N_{AG} 回ゴールするのに期待されるエピソード数の概数 N_{exBG} は，$N_{exBG} = \frac{N_{AG}}{P_{Goal}(.*B,N_{AG})}$ とかける．求めるべき切り替えのエピソード回数の期待値 Nep は次式となる．

$$N_{ep} = \sum_{N_{AG}=1}^{\infty} P_{contA}(N_{AG})\ (N_{AG} + \frac{N_{AG}}{P_{Goal}(.*B, N_{AG})}) \tag{7.6}$$

ここで，$P_{Goal}(.*B, N_{AG})$ の代わりに $P_{Goal}(B+, N_{AG})$ を用いて多めに見積る．$P_{Goal}(.*B, N_{AG}) = P_{Goal}(B+, N_{AG}) + P_{Goal}(.*A.*B, N_{AG})$ であり，$P_{Goal}(.*A.*B, N_{AG})$ は，学習が進むと FlagA の状況でゴールするようになって 0 に近づく．また，$P_{contA}(n)$ は，1 エピソードで $(.*A$ 系列$)$ でゴールし，2 エピソード以降に連続で $(A+$ 系列$)$

のみで n エピソード目までゴールし，$n+1$ エピソード目は，($A+$ 系列) 以外でゴールする確率とする．これは $P_{Goal}(A+,n)$ を用いて計算できる．

$$P_{contA}(n) = (1 - P_{Goal}(A+, n+1)) \prod_{k=2}^{n} P_{Goal}(A+, k) \tag{7.7}$$

ただし，($A+$ 系列) 以外でゴールする確率 $(1 - P_{Goal}(A+, n+1))$ は，$P_{Goal}(.*B.*A, n+1)$+$P_{Goal}(.*B, n+1)$ である．$P_{contA}(n)$ は指数関数的に減り，ここでの設定パラメータでは $P_{contA}(1)$=0.63, $P_{contA}(2)$=0.22, $P_{contA}(3)$=0.087 となり，$P_{contA}(20)$ では 2.9×10^{-7} と非常に小さくなる．そこで，N_{AG} を 1 から 19 までの和 Nep_{1-19} と 20 以降の和 Nep_{20-} に分けて計算する．Nep_{1-19} は 1 エピソードずつ (7.5)，(7.6)，(7.7) を用いて数値計算すると Nep_{1-19}=9.6 となる．Nep_{20-} は (7.6) を展開して求めていく．$P_{contA}(N_{AG})$ は，(7.1) より $P_{Goal}(A+)$ の最大が 1/2 であることから，少なくとも 1 エピソード毎に 1/2 倍ずつされていく．$P_{Goal}(.*B, N_{AG})$ は，行動選択確率をボルツマン選択を用いて算出することから最小値を持つ．今回のパラメータにおける最小値 6.6×10^{-3} を用いる．(7.6) の第 1 項は，第 2 項に比べて 1/1000 程度と十分小さいため無視できる．(7.6) の第 2 項は次式に展開でき，無限級数の和の公式を用いて求めると 0.0029 となる．

$$\begin{aligned}
&Nep_{20-} \approx \textstyle\sum_{N_{AG}=20}^{\infty} P_{contA}(N_{AG}) \, \frac{N_{AG}}{P_{Goal}(.*B, N_{AG})} \\
&= P_{contA}(20) \, \frac{20}{P_{Goal}(.*B, N_{AG})} \, (1 + \tfrac{21}{20} \tfrac{P_{contA}(21)}{P_{contA}(20)} + \cdots) \\
&\leq 2.9 \times 10^{-7} \, \tfrac{20}{6.6 \times 10^{-3}} \{1 + (\tfrac{1.05}{2})^1 + (\tfrac{1.05}{2})^2 + \cdots\} \\
&= 0.0029
\end{aligned} \tag{7.8}$$

ゆえに，Nep=Nep_{1-19}+Nep_{20-}　9.6 となる．ここで，Q-table の学習収束を適正行動の選択確率が 99% 以上になることとおくと，今回の学習条件において，その確率が 99% 以上になるまで Q 値が上がるには，FlagA と FlagB の各状況にて 30 回ずつ目標達成する必要があり，学習収束には 60 エピソードが最低限必要となる．それ

に対する切り替えエピソード回数の概算期待値は 16% 程度である．すなわち，提案手法において誤誘導の可能性を持つエピソード回数は，学習初期の数エピソード程度であることを数理的に確認できた．

7.5　結言

本章では，提案手法の学習の収束速度に着目して第 3 章で用いた 3 つの学習空間において従来手法と比較し，ケース 3 の有用でない部分空間を用いた場合に，学習効率がほとんど低下しないことに関して数理解析した．有用でない部分空間は適正行動が学習できないため，提案手法ではその部分空間に誤誘導されることになり，その分だけ，全体空間のみを持つ従来手法よりも学習の収束が遅くなる．その一方で，提案手法では，全体空間も同時に学習し，Q-table の学習進捗度を比較してより良い方を行動選択に用いる．その結果，部分空間の学習は進まない一方で，全体空間の学習が進むことから，行動選択には全体空間を用いるようになる．一旦，行動選択空間が全体空間に切り替われば，それ以降の学習は従来手法と変わらない．数理解析により，有用でない部分空間から全体空間への切り替わりが学習の初期段階で発生することを証明した．全体空間に切り替われば目標は達成できることから，提案手法は適正行動を学習できることが導かれる．

第8章　結論

本研究では，強化学習の問題である学習回数の多さに対して，学習器の多重化による学習効率化を提案した．提案手法では，複数の学習空間を並列に持ってそれぞれ学習し，行動選択の度に適切な空間に切り替えることで，学習回数を減らし，効率的な学習を実現する．

本論文の第1章「序論」では，本研究の背景と提案手法の概論を述べた後，関連研究に対する提案手法の位置づけについて述べた．

第2章「提案手法」では，強化学習において，最も効率的で正確に学習できる理想的な正解空間と実際に学習に用いる学習空間との関係パターンを整理した後，複数の学習空間を適宜切り替える提案手法の学習アルゴリズムについて述べた．

第3章「学習器の二重化」では，最も基本的な多重化パターンである全体空間に部分空間を一つ多重化する場合について，部分空間の有用性を変化させ，シミュレーション実験により提案手法の学習効率を検証した．結果より，提案手法では，部分空間が有用であるほど学習効率が向上する一方で，部分空間が有用でなくとも学習効率がほとんど低下しないことを確認した．

第4章「学習器の多重化」では，部分空間の数を増やす場合について，特に，有用でない部分空間が増える場合の提案手法の学習効率を，シミュレーション実験により検証した．結果より，有用でない部分空間が複数あっても，それらが交互作用的に学習の低効率化を引き起こさないこと，一つでも有用な部分空間が含まれていれ

ば，学習が効率化できることが確認できた．第 3 章で述べたことと考え合わせ，部分空間設計指針として,「用意した部分空間の中に有用でない空間が含まれていても学習効率は低下しない．その一方で有用な部分空間が含まれていればそれだけ学習効率は高くなる．したがって，予め部分空間の有用性を推測して，有用性の有無に応じて取捨選択する必要はない.」が導かれる．

第 5 章「学習器の多重化（包含関係なし）」では，学習器多重化の別応用として，報酬付与の遅れに，遅れ時間を考慮する学習器を多重化することで対応する強化学習を提案し，その有効性をシミュレーション実験により検証した．結果より，従来手法ではほとんど学習できない環境においても，提案手法では行動選択に適切な学習空間を選択して，学習することを確認した．

第 6 章「実環境における検証」では，実機においても，有用な部分空間が多重化される場合，第 3 章「学習器の二重化」のシミュレーション実験と同様に学習効率が向上するかを検証した．結果より，提案手法が実環境でもシミュレーションと同様に振る舞い，学習効率を向上させることを確認した．

第 7 章「数理解析」では，提案手法の学習の収束速度に着目して第 3 章で用いた 3 つの学習空間において従来手法と比較し，ケース 3 の有用でない部分空間を用いた場合にも，学習効率がほとんど低下しないことに関して数理解析した．結果により，有用でない部分空間から全体空間への切り替わりが学習の初期段階で発生することを証明した．全体空間に切り替われば目標は達成できることから，提案手法は適正行動を学習できることが導かれる．

最後に，学習空間の次元数が多い場合を考える．本論文では，提案手法の有効性検証に用いた学習環境の次元数は数次元程度と少ないものの，提案手法において，学習空間は 1 状態をそれぞれ比較して切り替えるため，環境の次元数が増減しても同じアルゴリズムが使え，結果も同様になると考えられる．具体的には，第 3 章，第 4

章にて検証した通り，多重化する部分空間が有用であるほど学習効率が向上し，第 7 章にて検証した通り，全く有用でない部分空間を用いる場合にも学習効率がほとんど低下しない．次元数が増えると，状況毎に冗長な次元が存在する可能性や，全体空間と部分空間の次元数の比，すなわち，部分空間の圧縮率が高くなる可能性が考えられる．本提案手法では，冗長な次元を圧縮した部分空間を多重化することで，状況毎に，より適切な学習空間を自動的に選択して学習を進めることができるため，冗長な次元を持つ状態数が多くなればなるほど，学習が効率化される．さらに，その状態では，部分空間の圧縮率に比例して学習が効率化される．つまり，次元数が増えるほど，提案手法はより効果的に働き，全体空間のみの従来手法に比べて大幅な学習効率化が期待できる．また，学習設計者は冗長な軸を圧縮した部分空間を用意するのみで，あらかじめ状況毎の冗長性を検討する必要はない点も，提案手法の利点の一つと考えている．

今後は，より複雑な学習器の多重化パターンである，学習空間同士が一部重なり，包含関係が不明な空間同士を多重化する場合について，アルゴリズムの検討が必要である．

参考文献

[1] R.S. Sutton, A. Barto: “Reinforcement Learning: An Introduction”, A Bradford Book, The MIT Press, 1998.

[2] 今福啓: “環境の変化に適応するマルチエージェントの学習法”, 人工知能学会論文誌, vol.21, no.2, pp.153–166 2006.

[3] 森山甲一, 沼尾正行: “環境状況に応じて自己の報酬を操作する学習エージェントの構築”, 人工知能学会論文誌, vol.17, no.6, pp.676–683 2002.

[4] S. Mikami, Y. Kakazu and T. C. Fogarty: “Co-operative Reinforcement Learning By Payoff Filters, in Proc”, 8th European Conference on Machine Learning ECML-95(Lecture Notes in Artificial Intelligence 912), pp.319–322, Heraclion, Crete, Freece, 1995.

[5] 小堀訓成, 鈴木健嗣, パトヨハルトノ, 橋本周司: “尤度情報に基づく温度分布を用いた強化学習法”, 人工知能学会論文誌, vol.20, no.4, pp.297–305 2005.

[6] 松井藤五郎, 犬塚博久, 世木博久, 伊藤英則: “強化学習結果の再構築への概念学習の適用”, 人工知能学会論文誌, vol.17, no.2, pp.135–144 2002.

[7] 阪口豊, 高野光雄: “環境変化への適応と文脈切替え”, 第 16 回生体・生理工学シンポジウム論文集, pp.157–160, 2001.

[8] 浅田稔, 野田彰一, 俵積田健, 細田耕: “視覚に基づく強化学習によるロボットの行動獲得”, 日本ロボット学会誌, Vol.13, No.1, pp.68–74, 1995.

[9] 前田陽一郎, 花香敏: “Shaping 強化学習を用いた自律エージェントの行動獲得支援手法”, 知能と情報, 日本知能情報ファジィ学会誌, Vol.21, No.5, pp.722–733, 2009.

[10] 高橋泰岳, 浅田稔: “階層型学習機構における状態空間の構成”, 日本ロボット学会誌, Vol.21, No.2, pp.164–171, 2003.

[11] 内部英治, 銅谷賢治: “複数報酬のもとでの階層強化学習”, 日本ロボット学会誌, Vol.22, No.1, pp.120–129, 2004.

[12] J. L. Barry, L. P. Kaelbling, T. Lozano-Perez: “DetH*: Approximate Hierarchical Solution of Large Markov Decision Processes”, 22nd International Joint Conference on Artificial Intelligence, IJCAI-11, AAAI Press, pp.1928–1935, 2011.

[13] M. Nagayosi, H. Murao, H. Tamaki: “Developing reinforcement learning for adaptive co-construction of continuous high-dimensional state and action spaces”, Journal of Artifical Life and Robotics, pp.204–210, 2012.

[14] 増山岳人, 山下淳, 淺間一: “変換不変性を用いた経験の抽象化と内発的動機づけに基づく強化学習”, 日本機械学会論文集C編, 79 巻, 798 号, pp.289–303, 2013.

[15] 齋藤雅矩, 瀬古沢照治: “エージェントの行動履歴を活用した Q-learning アルゴリズムの提案”, 電気学会論文誌C, 136 巻, 8 号, pp.1209–1217, 2016.

[16] K. Terashima, H. Takano, J. Murata: “Acceleration of Reinforcement Learning by Controlled Use of Options Given as Prior Information”, SICE Journal of Control, Measurement, and System Integration, Vol.6, No.4, pp.252–258, 2013.

[17] 森本淳, 銅谷賢治: “強化学習を用いた高次元連続状態空間における系列運動学習 –起き上がり運動の獲得–”, 電子情報通信学会論文誌, Vol.J82-D-2, No.11, pp.2118–2131, 1999.

[18] 小林祐一, 湯浅秀男, 新井民夫: “自律系分散系の適応アルゴリズムによる強化学習のための関数近似”, 計測自動制御学会論文集, 38 巻, 2 号, pp.219–226, 2002.

[19] 小林高彰, 澁谷長史, 森田昌彦: “選択的不感化ニューラルネットを用いた連続状態行動空間における Q 学習”, 電子情報通信学会論文誌, Vol.J98-D, No.2, pp.287–299, 2015.

[20] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller: “Playing Atari with Deep Reinforcement Learning”, NIPS 2014 Deep Learning Workshop, pp.1–9(arXiv:1312.5602v1), 2013.

[21] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis: “Human-level control through deep reinforcement learning”, Nature, Vol.518, pp.529–533, 2015.

[22] 濱上知樹, 小圷成一, 平田廣則: “適応的な状態分割を行う Q-Learning における状態数の調整方法”, 電子情報通信学会論文誌, Vol.J86-D-1, No.7, pp.490–499, 2003.

[23] 鮫島和行, 片桐憲一, 銅谷賢治, 川人光男: “複数の予測モデルを用いた強化学習による非線形制御”, 電子情報通信学会論文誌, Vol.J84-D-2, pp.2092–2106, 2001.

[24] 山口明彦, 杉本徳和, 川人光男: “回避行動の再利用メカニズムを備えた強化学習手法と多関節ロボットの全身運動学習への応用”, 日本ロボット学会誌, Vol.27, No.2, pp.209–220, 2009.

[25] 港隆史, 浅田稔: “環境の変化に適応する移動ロボットの行動獲得”, 日本ロボット学会誌, vol.18, no.5, pp.706–712 2000.

[26] 片山謙吾, 輿石尚宏, 成久洋之: “強化学習エージェントへの階層化意志決定法の導入–追跡問題を例に–”, 人工知能学会論文誌, Vol.19, No.4, pp.279–291, 2004.

[27] 内部英治, 銅谷賢治: “重点サンプリングを用いた複数強化学習器の同時学習”, 電子情報通信学会技術研究報告, NC, pp.179–184, 2003.

[28] 高橋泰岳, 田村佳宏, 浅田稔: “価値システムに基づく他者行為観察と自己行動学習の循環的発達”, 日本知能情報ファジィ学会誌, Vol.21, No.5, pp.640–652, 2009.

[29] 伊藤昭, 金渕満: “知覚情報の粗視化によるマルチエージェント強化学習の高速化 – ハンターゲームを例に –”, 電子情報通信学会論文誌, Vol.J84-D-1, No.3, pp.285–293, 2001.

[30] C.J.C.H. Watkins, P. Dayan: “Technical Note:Q-Learning”, Machine Learning, vol.8, pp.279–292, 1992.

[31] 日置智恵子, 松井博和, 野村由司彦: “未知環境において行動指数を利用する強化学習法 – 同一空間上に対して複数の学習テーブルを用いる方法 –”, 第 44 回人工知能学会 人工知能基礎論研究会資料, pp.31–34, 2001.

[32] 西澤智恵子, 松井博和: “多重学習器を用いる強化学習の検討”, 第 31 回 日本ロボット学会 学術講演会予稿集, RSJ2013AC3I3-05, 2013.

[33] C., Nishizawa, H., Matsui, Y., Nomura: “Investigation of reinforcement learning with multiplex learning spaces”, Australian Journal of Basic and Applied Sciences, 8 (4) Special, pp.455–458, 2014.

[34] 西澤智恵子, 松井博和: “多重学習器を用いる強化学習 –有用でない学習空間を増加させた場合の学習効率低下の考察–”, 日本機械学会 ロボティクス・メカトロニクス講演会講演概要集, 1A1-X07, 2014.

[35] C., Nishizawa, H., Matsui, Y., Nomura: “Reinforcement learning with multiplex learning spaces:- consideration of the learning inefficiency in a case that all the partial spaces are ineffective and are not similar each other -”, Proceedings of the twentieth International Symposium on Artificial Life and Robotics, pp.49–52, 2015.

[36] C., Nishizawa, H., Matsui, Y., Nomura: “Reinforcement learning with multiplex learning space:Q-learning with an inconstant delay time”, Proceedings 2nd International Conference on Information Technology, pp.126–130, 2016.

[37] 西澤智恵子, 松井博和: “多重学習器を用いる強化学習 –報酬付与に遅れがある学習環境への適用–”, 日本機械学会 ロボティクス・メカトロニクス講演会講演概要集, 1P1-04b3, 2016.

[38] 廣川 暢一, 鈴木 健嗣: ”教示者による学習支援に基づくエージェントのオンライン行動獲得”, 人工知能学会論文誌, vol.25, no.6, pp.694–702 2010.

[39] 田中一晶，尾関基行，荒木雅弘，岡夏樹: ”ロボットへの教示場面における「間」の重要性：ロボットの行動の遅れは学習効率を向上させ教えやすい印象を与える”，人工知能学会誌, 25 巻 6 号 SP-D, pp.703-711, 2010.

[40] C.Nishizawa, H.Matusi, Y.Nomura: ”Reinforcement learning with multiplex learning spaces – Consideration of the learning inefficiency in a case that all the partial spaces are ineffective and are not similar each other –”, Proceedings of the twentieth international symposium on Artificial Life and Robotics (2015)

[41] Osamu NISHIMURA, Hirokazu MATSUI, Chieko HIOKI,Yoshihiko NOMURA: “Reinforcement Learning with Self-Instruction by using dual Q-tables”, AROB 11th, 2006. 5

[42] 西澤智恵子, 松井博和: “多重学習器を用いる強化学習の検討”, 第 31 回 日本ロボット学会学術講演会，2013. 9

謝辞

本論文は，三重大学大学院工学研究科システム工学専攻において，機械工学専攻野村由司彦教授，松井博和助教の御指導のもとに研究し，日本ロボット学会誌等に公表した論文をまとめたものである．本研究を遂行するにあたり，長期間にわたり御指導と御鞭撻を賜りました野村由司彦教授，松井博和助教に対して，謹んで深謝の意を表します.

本論文をまとめるにあたり，同研究科情報工学専攻の若林哲史教授，ならびに電気電子工学専攻の高瀬治彦教授に，懇切なる御指導と御助言を頂き，深く感謝申し上げます.

私が所属するプロセス解析研究室において，坂本良太助教や学生の方々と意見交換をさせていただき，大変感謝しております。また，実験に協力してくださったメカトロニクス研究室の方々に深謝致します.

同研究科分子素材工学専攻の久保雅敬教授には，国際会議における発表において大変お世話になり，心より感謝申し上げます.

社会人ドクターとして研究活動をする中，業務調整してくださったトヨタテクニカルディベロップメント株式会社の上司，同僚の皆様に深謝致します．最後に，心より応援，サポートしてくれた夫をはじめ，家族のみんなに感謝致します.

皆様，本当にありがとうございました.