学術・技術論文

環境の変化に適応する移動ロボットの行動獲得

港 隆史* 浅田 稔*

Environmental Change Adaptation for Mobile Robot Navigation

Takashi Minato * *and* Minoru Asada *

Most of existing robot learning methods assume that the environment where a robot works does not change, therefore, a robot has to learn from scratch if it encounters new environments. This paper proposes a method which adapts a robot to environmental changes by efficently transferring a learned policy in the previous environments into a new one and effectively modifying it to cope with the environmental changes. The resultant policy (a part of state transition map) does not seem optimal with respect to each individual environment, but may absorb the differences between multiple environments. We apply the method to a mobile robot navigation problem of which task is to reach the target avoiding obstacles based on uninterpreted sonar and visual information. Experimental results show the validity of the method.

Key Words: Mobile robot, Adaptation, Policy transfer, State vector selection

1. はじめに

生物だけでなくロボットのような人工システムにとっても，変化する環境に適応するために学習と発達は重要なプロセスと考えられる．従来の多くのシステムでは環境の変化に対応させるために，設計者が起こり得る環境の変化をすべて想定し，ロボットに個々の環境に対する行動政策（制御モジュール）を学習させ，それらを環境ごとに切り替えさせていた（たとえば[1]にいくつかの例がある）．しかしこのような手法では次のような問題点がある．

- ロボットは各環境に対して独立に学習を行っており，他の環境で学習した経験をバイアスとして学習に再利用していない．すなわち学習時間を短縮させるための情報を利用していない．
- 各環境に対する行動政策は学習した環境に特化されたものであり，汎用性がない．そのため環境ごとに行動政策を持つこととなり，環境の数が増加すれば多大な記憶領域を必要とする．

前者の問題に対してThrun et al. [2]〜[4]は生涯学習（lifelong learning）という枠組みを提案している．この枠組みではロボットはあるクラス内の環境で過去に学習した経験を保持しておき，それを新たな環境における学習時にバイアスとして再利用する．彼らの手法ではタスクに関して複数の環境で不変な知識を事前に獲得し，これをある環境での学習に利用することにより学習時間を短縮させている．具体的には移動ロボットのナビゲーションタスクにおいて，行動とセンサの変化および報酬予測値との関係を示す行動モデルを，説明に基づくニューラルネットワーク（EBNN）によりあらかじめ獲得させ，これを不変知識として用いている．しかしこの手法では事前に獲得した不変知識は固定であり，ロボットが学習中に得た経験は不変知識に反映されない．

これに対してTanaka and Yamamura [5] [6]は，強化学習を組み込んだ生涯強化学習（lifelong RL）という枠組みの中で，ロボットの行動政策をニューラルネットワークで表現し，過去の複数の環境に対してほとんど変化していない重みを不変知識として用いている．この手法では不変知識は学習に伴って更新される．しかし以上の手法で獲得された行動政策は学習した環境に特化されたものであり，ロボットが過去に経験した環境で再びタスクを達成するためには，環境ごとに行動政策を持たなければならず，上述した記憶量の問題は考慮されていない．この手法では不変知識としてニューラルネットの重みを用いているが，汎用性のある行動政策を構築するためには不変知識としてより抽象度の高い表現を用いることが考えられる．

そこで本論文では，強化学習の一手法であるQ学習[7]によって獲得された単一の行動政策を部分的に修正することにより，環境の変化に適応する手法を提案する．本手法ではロボットが持つ行動政策の中で，新たな環境において不都合が生じる部分のみを学習しなおすことにより学習時間を短縮する．不都合が

原稿受付 1999年5月28日
*大阪大学大学院工学研究科
*Guraduate School of Engineering, Osaka University

生じない部分，すなわち環境間で共通にタスク達成に使用される部分が不変知識として残される．Q学習の枠組みで不変知識と考えられるものとして行動価値や行動政策が考えられる．前者は環境変化による変化が後者より大きいと考えられるので，ここでは行動政策を不変知識とすることにより，複数の環境でタスク達成可能な汎用性のある単一の行動政策を構築することを目指す．構築される行動政策は個々の環境において最適な行動を示すとは限らないが，環境間の相違を吸収したものとなる．本論文では提案する手法を，視覚と超音波センサを持つ移動ロボットが障害物を回避し目標物に到達するナビゲーションタスクに適用する．このタスクは以下の2つの問題を含んでいる．

- 異なる種類のセンサ情報の役割が事前に割り当てられていないため，ロボット自身がそれらをどのように扱うかを決定しなければならない．
- 目標物到達と障害物回避という対照的な2つの行動が同時に獲得されなければならない．

また別の重要な問題として，Q学習におけるロボットの状態構築手法が挙げられる [8]〜[10]．複数のセンサ情報を用いて高次元の状態空間を構築すれば，学習の収束に多くの時間を要する．一方，過度に低次元化すれば目的の行動が獲得できない．したがって適切な状態空間を発見しなければならない．上述した従来研究では設計者が事前に状態空間を与えている．それに対して本論文ではセンサ特徴量から複数の状態変数を用意し，それらの組み合わせの中から適切な状態空間を探索する．

以下では簡単にQ学習について説明した後，提案する手法の基本的な考え方とアルゴリズムを示す．次に適用するタスクを説明した後，計算機シミュレーションと実機システムでの実験結果を示す．

2. Q学習の基本的枠組み [7]

環境とエージェントの相互作用を通して目的の行動を獲得する手法として強化学習は有効な手法であり，その一手法であるQ学習はロボットを含めた環境に関する先験的知識をほとんど必要としないという利点がある．

ロボットが識別することのできる状態の集合を $\boldsymbol{S}$ とし，環境に対してとり得る行動の集合を $\boldsymbol{A}$ とする．環境は現在の状態とロボットの行動によって確率的に遷移するマルコフ過程に従うものとする．状態と行動の組 (s, a) に対しては報酬 $r(s, a)$ が定義される．

一般的な強化学習の問題は，減衰する報酬の積算を最大にする行動政策を見つけることである．そのためにQ学習では状態 $s \in \boldsymbol{S}$ において行動 $a \in \boldsymbol{A}$ をとり，次状態 s' に遷移したとき行動価値関数 $Q(s, a)$ を以下のように更新する．

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(r(s,a) + \gamma \max_{a' \in \boldsymbol{A}} Q(s',a')) \tag{1}$$

ここで α は学習率，γ は減衰係数である．Q 値が与えられると行動政策 $a = f(s)$ は以下のように定義される．

$$f(s) \leftarrow a \text{ such that } Q(s,a) = \max_{a' \in \boldsymbol{A}} Q(s,a') \tag{2}$$

3. 環境の変化への適応

通常のQ学習で得られる行動政策（$\boldsymbol{S}$ から $\boldsymbol{A}$ への写像 $P : \boldsymbol{S} \rightarrow \boldsymbol{A}$）は環境の状態遷移の情報を含んでいる．そのため環境の変化により状態遷移が変化すると，それまでに獲得した行動政策は新たな環境で最適行動を示さない．しかし最適性を犠牲にしてタスク達成だけを考えれば，それまでの行動政策が新たな環境で部分的に適用できる可能性がある．

そこで本手法では過去の環境でQ学習により獲得した行動政策を，新たな環境でタスクが達成できるように部分的に修正する．そのためにはロボットが環境の変化を認識したとき，タスク達成に不都合が生じる状態を探索し，そのような状態の行動政策だけを学習しなおす．ここでそれまでに獲得した行動政策をできるだけ破壊しないために，ロボットがタスクを失敗した状態（負の報酬が与えられた状態）付近の状態を学習しなおす．またロボット自身に環境の変化を認識させるために，タスク成功率の低下により環境の変化を認識させる．ロボットは自身のタスク成功率を常に測定し，タスク成功率の低下によって環境の変化を認識する．したがって，物理的に環境が変化してもタスク成功率が低下しなければロボットにとって環境の変化とは認識されない．

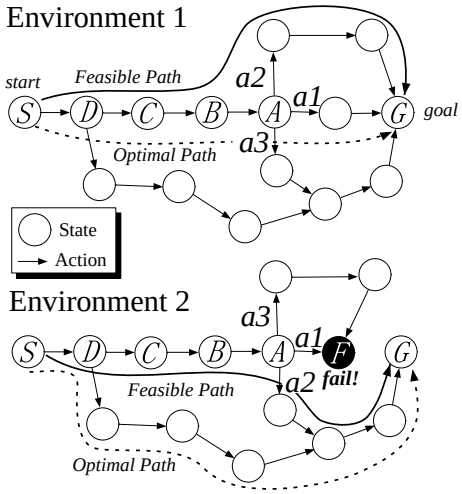


Fig. 1 Policy modification

このことを状態遷移が異なる2つの環境の例（**Fig.1**）を用いて具体的に説明する．まずロボットが環境1で最適行動政策（図中のoptimal path）を獲得したとする．このロボットが環境2に遭遇したときに環境1で獲得した行動政策を適用すると，状態遷移が変化しているために状態 F でタスクに失敗する．ここでロボットはタスク成功率の低下により環境の変化を認識し，F の直前の状態 A の行動政策を学習しなおす．その結果 $A \rightarrow a_2$ と修正された行動政策を獲得する．この場合は，行動政策は最適ではないが，どちらの環境においてもタスクを達成できるものとなっている．ここでロボットは行動政策修正時には現在の環境におけるタスク成功率のみ測定しているため，本当に修正した行動政策が過去の環境においても有効な政策と

なっているかどうかは検証しない．このようにして過去の環境における行動政策をできるだけ破壊せずに，現在の環境でタスクを達成できる単一の行動政策を構築する．

この例では修正する状態をFの直前の状態としたが，タスクを達成するためにはさらに修正する状態の範囲が広がる場合も考えられる．しかし修正する状態の範囲を広げると行動政策が破壊されるため，ここでは現在の環境における目標成功率を設定し，その値が得られるまで修正する状態の範囲を広げながら学習する．そのためにロボットがタスクを失敗したときに経験した状態時系列において，失敗状態のn_rステップ前までの状態を再学習状態をとし，n_rの値を1から1づつ増やすことにより再学習状態の範囲を拡大する．Fig.1の場合，再学習状態は$n_r = 1$のときA，$n_r = 2$のときA, Bのようになる．さらに再学習状態の範囲を広げるために状態空間上でそれらの状態と隣接する状態も再学習状態とする．ここではできるだけ過去の環境に有効な行動政策を破壊しないために，再学習状態を失敗状態近傍に設定したが，タスクによっては失敗状態から離れた状態を再学習する必要がある場合も考えられる．この場合はn_rの値が大きくなるまで再学習範囲を拡大することにより対処する．以上を手続き化したものを以下に示す．

(1) 初期環境で通常のQ学習を行う．得られた行動政策を$P : \boldsymbol{S} \rightarrow \boldsymbol{A}$とする．
(2) タスク成功率Rが低下しない限り行動政策Pを使い続ける．Rが低下すれば$n_r = 1$として(3).へ．
(3) 行動政策を修正する状態集合$\boldsymbol{S}_r \subset \boldsymbol{S}$を探索する．ここで$\boldsymbol{S}_r$は失敗状態から$n_r$ステップ前までの状態，および状態空間上でそれらと隣接する状態の集合である．
(4) Q学習を行う．学習中の状態sにおける行動は
$s \notin \boldsymbol{S}_r \bigcup \boldsymbol{S}_u$なら$P$
それ以外なら通常のQ学習における探索行動
に従う．ここで$\boldsymbol{S}_u \subset \boldsymbol{S}$は未経験状態集合である．
これにより$\boldsymbol{S}_r \bigcup \boldsymbol{S}_u$における行動政策だけが修正される．
(5) 目標成功率R_dが得られれば(6)へ．得られなければ$n_r \leftarrow n_r + 1$として(3)へ．
(6) 得られた行動政策をPとして(2)へ．

ここで目標成功率R_dは，環境の変化時に低下した成功率を回復させる割合βによって決定される．

$$R_d = R_c + \beta(R_p - R_c) \qquad (3)$$

上式中のR_c, R_pはそれぞれ現在の環境での成功率，前環境での成功率である．βは行動政策をどれだけ破壊せずに維持するかを示す値とみなすことができる．設計者がロボットに与えるパラメータはβのみであり，βを与えることによりロボットは自律的に環境の変化を発見し，環境の変化に適応する．

4. タスク設定と状態空間の構成

4.1 ロボット

ロボットは左右輪を独立に駆動できる独立二輪操舵（PWS）機構を持つ．実験では各車輪への指令値を前進，停止，後退の3段階に離散化する．したがってロボットは合計9 通りの行動を持つ．

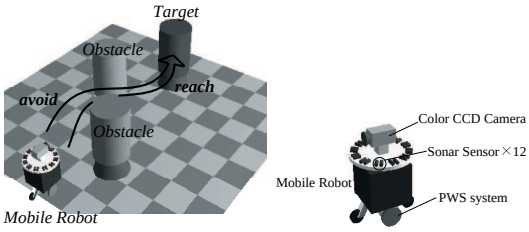
またロボットはセンサとしてカラーCCDカメラと環状に配置した12個の超音波センサを持つ．**Fig.3**(a)に示すようにカメラの画角は60[°] で，カメラ画像からは色情報により目標物が識別できる．超音波センサの測定可能距離は3[m]で，**Fig.3**(b)に示すように測定可能範囲角は30[°]である．

4.2 タスク

ロボットのタスクは**Fig.2**(a)に示すように，静止障害物を回避しながら指定された静止目標物に到達することである．これは1節で述べたように2つの解決すべき問題を含んだタスクである．環境の変化については，本手法では環境が変化してもロボットが持つ状態変数自体は変化させないため，次節で定義した状態変数で対応できる変化として次の2つの変化を想定した．

- 目標物と障害物の相対的配置の変化．
- 障害物の個数の変化．

他にも物体の形状の変化，通路環境の存在などが考えられるが，それらはこの2つの変化で表現することが可能である．



(a) task (b) robot

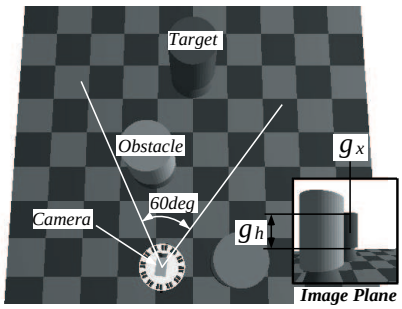
Fig. 2 Task and mobile robot

4.3 状態空間の構成

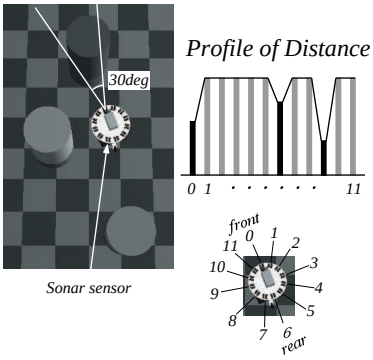
上述したように，強化学習における状態空間構成は単一の環境においても重要な問題である．Nakamura et al. [10] は本論文と同様の設定でQ学習を行っているが，状態空間の構成にタスクの指標が入っておらず，タスクに有効な状態空間が構成されている保証がない．

そこで本論文では学習結果が高い能力を示す状態変数を探索することにより，タスクに有効な状態空間を構成する．状態空間を構成する状態変数の候補として，カメラ画像からはFig.3(a)に示した画像上の目標物の水平位置g_xおよび高さg_h，そして超音波センサ測定値パターンからは**Fig.4**および**Table 1**に示した8つの特徴量を選んだ．Fig.4においてd_{limit}は超音波センサが反応していないことを示している．

以上の10個の状態変数の組み合わせにより構成される複数の状態空間の中から，Q学習を行った結果ロボットが最大のタスク成功率を示すものを選び出す．状態空間の次元を抑えるため状態変数は最高4つとし，さらに組み合わせ方法は，1あるいは2個の画像変数(g_*)，0あるいは1個の距離変数(d_*)，0あるいは1個の方向変数(θ_*)の組み合わせとした．次元の異なる状態空間を比較するため，次元の小さい状態空間では離散化を細かくすることにより各状態空間をほぼ状態数が等しくなるよう



(a) sonar



(b) vision

Fig. 3 Sensory information

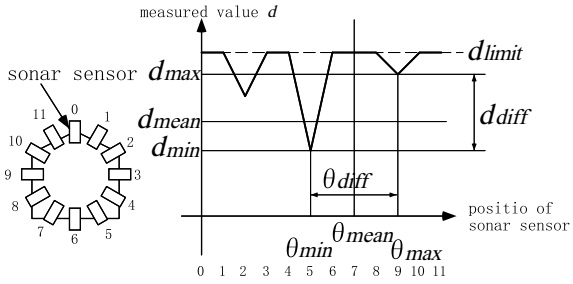


Fig. 4 Primitive features from sonar profile

Table 1 Sonar features

range feature	
d_{min}	minimum range value observed
d_{max}	maximum range value observed
d_{mean}	mean range value observed
d_{diff}	$d_{max} - d_{min}$
direction feature	
θ_{min}	direction of d_{min}
θ_{max}	direction of d_{max}
θ_{mean}	mean between θ_{min} and θ_{max}
θ_{diff}	width between θ_{min} and θ_{max}

に離散化した．Q学習はFig.2(a)の環境（物体，ロボットをともに直径0.4[m]の円柱で近似）でコンピュータシミュレーションにより行った．Q学習の設定は以下の通りである．

- 学習率 $\alpha = 0.25$，減衰係数 $\gamma = 0.9$.
- 報酬はロボットが目標物に到達したときに1，その他は0を与える．
- 1試行はロボットが目標物に到達したとき，障害物に衝突したとき，決められた時間を超えたときのいずれかを満足したときに終了する．

各状態空間を持つロボットの学習後のタスク成功率を**Fig.5**に示す．ここでタスク成功率は（目標物到達回数／試行回数）である．(a)はg_x，距離変数，方向変数からなる状態空間の結果，(b)はg_h，距離変数，方向変数からなる状態空間の結果，(c)はg_x, g_h，距離変数，方向変数からなる状態空間の結果を示している．変数の添字$none$はその変数を用いていないことを示す．この結果から変数θ_{min}を含む状態空間を持つロボットが高い成功率を示しており，ロボットに最も近い物体の方向が重要な情報であることがわかる．Fig.5の全ての状態空間の中からタスク成功率の高いものを選び出すと，$g_x, g_h, \theta_{min}, d_{mean}$および$g_x, g_h, \theta_{min}, d_{min}$が選ばれる．タスク成功率が最高のものでも100%に達していない理由は，状態空間の離散化による知覚的見せかけ問題[11]が生じているためである．ここでは直観的に理解しやすいため，ロボットの状態空間を表現する状態ベクトル$\boldsymbol{x}$の要素として次の4つの変数を選択した．

$$\boldsymbol{x} = \begin{pmatrix} g_x & g_h & \theta_{min} & d_{min} \end{pmatrix}^T \qquad (4)$$

ここで得られた状態ベクトルは設計者がトップダウン的に与えたとしても充分考えることができるものである．しかしこれが実際にタスクに有効であることを検証したことにこの実験の意義がある．

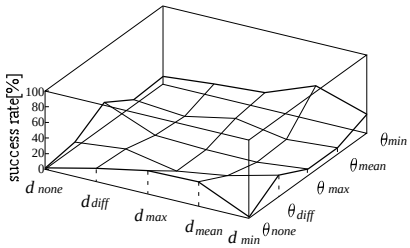
5. 実験結果と考察

提案した手法の有効性を検証するため，まずコンピュータシミュレーションによる実験結果を示し，得られた行動政策に関して考察する．次に実機による実験結果を示し，最後に本手法における不変知識に関して考察する．

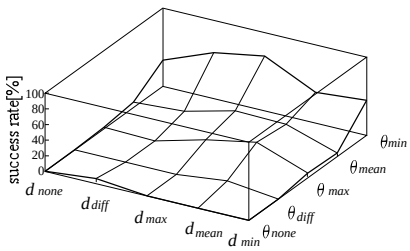
5.1 シミュレーション

Fig.6に示す5個の環境を用意してコンピュータシミュレーションにより実験を行った．図中の各環境の一番上の円が目標物，その他の円が障害物，多角形が移動ロボットを示している．ロボットは初期環境を環境$E1$とし，$E2$から$E5$へと順に適応する．学習中に環境は変化しないと仮定する．ロボットの状態空間は式（4）を用い，3060の状態に離散化した．学習の設定は4.3節と同様の設定とし，成功率を回復させる割合 β を0.8とした．失敗状態は障害物に衝突した状態および決められた時間が経過したときにロボットがいる状態とする．後者はロボットがデッドロックに陥った状態を検出するために用意した．またロボットは成功率が10%以上低下したときに環境が変化したと認識する．

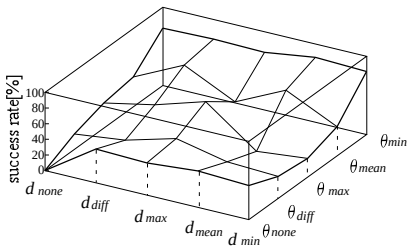
各環境で獲得された行動政策を用いたときのタスク成功率を**Table 2**に示す．i番目の環境Eiで獲得された行動政策がPi



(a) g_x, θ_*, d_*



(b) g_h, θ_*, d_*



(c) g_x, g_h, θ_*, d_*

Fig. 5 Performance of each state space

で，各成功率は行動政策Piを環境Eiに適用した結果である．() 内の成功率は参考に測定したものである．また$||\boldsymbol{S}_r||$は行動政策Piを獲得する際に実際に行動政策が変更された状態数で() 内の数値はn_rを示している．

Table 2 Success rates of each policy [%]

policy	$\|\|\boldsymbol{S}_r\|\|$	$E1$	$E2$	$E3$	$E4$	$E5$
$P1$	-	90.9	61.0	(48.2)	(47.3)	(61.5)
$P2$	61(2)	[92.4]	90.8	45.2	(50.2)	(69.5)
$P3$	75(2)	[88.2]	[88.8]	93.3	82.7	(68.9)
$P4$	76(2)	[83.0]	[83.5]	[83.5]	94.6	52.7
$P5$	59(2)	[73.3]	[87.0]	[92.7]	[92.2]	93.3

ロボットが新たな環境に遭遇するとタスク成功率が低下するが，本手法により新たな環境およびそれまで経験した環境でタスクを達成可能な単一の行動政策を獲得している．たとえば環境$E1$で行動政策$P1$を獲得したロボットが環境$E2$に遭遇すると，タスク成功率が90.9%から61.0%に低下する．ここでロボットは環境の変化を認識し，再学習状態を探索して行動政策を修正する．このときは61の状態の行動政策が修正されている．そしてその結果得られた行動政策$P2$を用いると，タスク成功率は61.0% から90.8%に回復する．また行動政策$P2$を環境$E1$に用いるとタスク成功率は92.4%となっており，$P2$は2つの環境$E1, E2$でタスク達成可能なものとなっている．最終的に獲得された行動政策$P5$は，それまで経験したすべての環境においてほぼタスクを達成している．ここで表中の[] 内の成功率は獲得した行動政策を過去に経験した環境に適用した結果である．Fig.6に示したロボットの移動軌跡は行動政策$P5$を各環境に適用した結果である．Table 2においてタスク成功率が設定したβの値より大きく回復している理由は，再学習状態範囲の拡大が離散的に行われているためである．

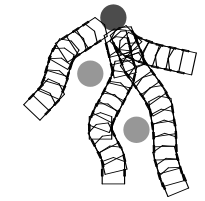
次に本手法により学習収束時間が通常のQ学習より短縮される結果を示す．そのために環境$E2$において次の2つの方法により学習収束時間を比較した．

- ケース1:環境$E2$において通常のQ学習を行う．
- ケース2:環境$E1$で行動政策$P1$を獲得したロボットが，本手法により環境$E2$で学習を行う．

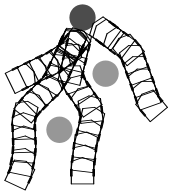
その結果，学習の収束までにケース1では5400試行，ケース2では1900試行それぞれ要した．この結果では本手法により学習収束時間が約65%短縮されている．

5. 2 得られた行動政策に関する考察

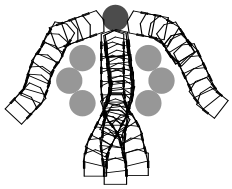
本手法ではタスク失敗により行動政策を修正する状態が決定されるため，獲得される行動政策はロボットの過去の経験，すなわち過去に作業した環境によって変化する．そのことを示すため，同様の実験をロボットが遭遇する環境の順序を逆にして行った．その結果を**Table 3**に示す．この実験ではロボットは環境の変化を2度しか認識しておらず（$E5 \rightarrow E4$と$E3 \rightarrow E2$），2度の行動政策修正によりすべての環境でタスク達成可能な行動政策を獲得している．$P5$も$P2'$も同じ環境を経験して獲得した行動政策であるが，成功率の違いから分かるように異なったものとなっている．また$P2'$を環境$E5$に適用した結果（**Fig.7**）



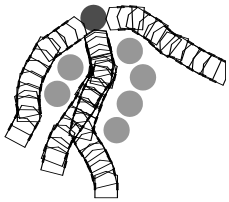
(a) Environment $E1$



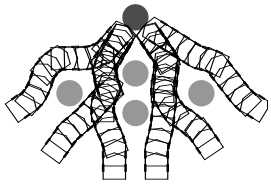
(b) Environment $E2$



(c) Environment $E3$



(d) Environment $E4$



(e) Environment $E5$

Fig. 6 Environments and successful trajectories

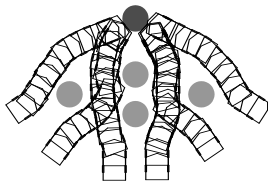


Fig. 7 Successful trajectories (policy $P2'$)

を見れば，ロボットの移動軌跡がFig.6(e)と異なることが確認できる．

Table 3 Another result [%]

policy	$\|\|\boldsymbol{S}_r\|\|$	$E5$	$E4$	$E3$	$E2$	$E1$
$P5'$	-	94.4	78.0	(85.3)	(55.3)	(63.9)
$P4'$	37(2)	[92.8]	91.8	95.2	57.9	(65.2)
$P2'$	45(1)	[89.1]	[91.2]	[95.6]	87.9	86.9

本手法が従来手法と異なる点は，本手法が各環境における最適行動政策を求めるのではなく，それまで持っていた行動政策をできる限り破壊せず，かつタスクが達成できるように修正する点である．このことは環境間で共有できるタスク達成可能な行動政策を求めていると考えることもできる．本手法で獲得された行動政策は最終的にどの環境においてもタスク達成可能である反面，最適でない行動政策であってもタスク達成に影響を及ぼさなければ保存されるため，各環境において行動の最適性は失われている．Fig.6の環境$E5$の一番左のロボットの移動軌跡がその例である．最適行動をとればロボットは障害物を回避したのち目標物に向かって前進するはずであるが，この例では右に遠回りしている．

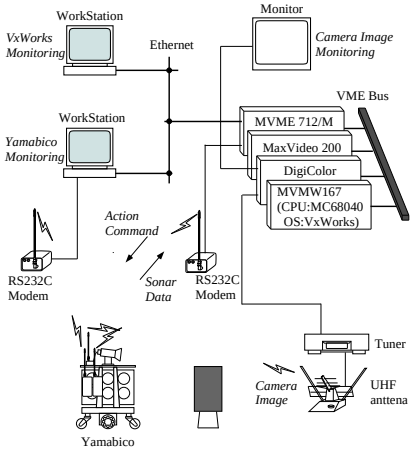


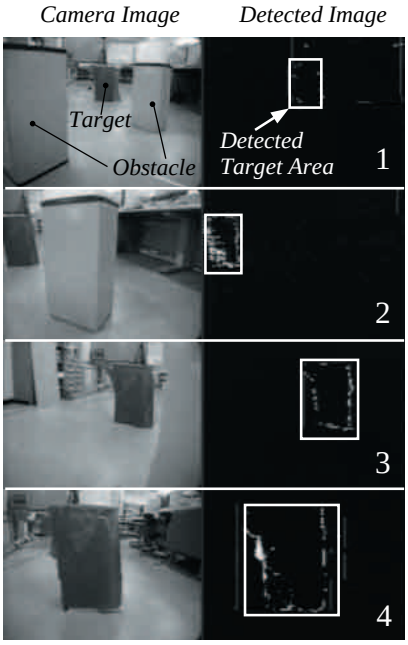
Fig. 8 Experimental system

5.3 実機での実験

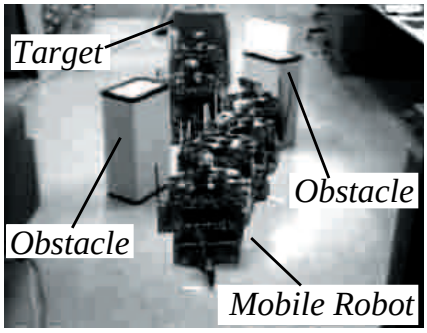
環境の変化としてシミュレーション環境から実環境への変化を設定し，獲得した行動政策がそれらの間の相違を吸収したものとなっていることを検証する．そのために，Fig.6の環境$E1$および$E2$に似た実環境を用意し，行動政策$P5$を実ロボットに適用した実験を行った．移動ロボットとして「山彦」を使用し，4.1節で述べたセンサと同等のセンサを取り付けた．実験システム全体の概要を**Fig.8**に示す．画像処理等の主な処理はホストコンピュータ（OS:VxWorks）で行われ,「山彦」とのデータ通信はすべて無線で行われる．画像処理はリアルタイムパイプライン画像処理装置MaxVideo200で行われる．

Fig.6(a)に示す環境$E1$に似た実環境に対して適用した結果を**Fig.9**に示す．Fig.9(a)はロボットのカメラ画像および目標物を抽出した画像のシーケンス，(b)はロボットの移動軌跡で

ある.「山彦」が障害物であるごみ箱を回避し，目標物のごみ箱に到達する結果が得られた．このときタスク成功率は低下しておらず，ロボットは新たな環境であると認識することなくタスクを達成している．またFig.6(b)に示す環境$E2$に似た実環境においても「山彦」は新たな環境と認識することなくタスクを達成した.



(a) Real input images



(b) Successful trajectory

Fig. 9 Experimental result of real robot

5.4 不変知識に関する考察

本手法では環境が変化したときに行動政策（式（2）における$f(s)$）のみを残しており，行動価値（式（2）における$Q(s,a)$）は残していない．ここではその理由について述べる.

一般にQ学習では学習が収束すると**Fig.10**の左図のように目標状態に近い状態ほど高い行動価値を持つ．ここで環境がE_iからE_jに変化したときに，状態遷移がFig.10の右図のように変化したとする．ここで環境E_iで獲得した行動価値をそのまま初期値として利用すると目標状態から遠い状態が高い状態価値を持つことになる．このような状態価値を初期値とすると次のような問題が生じる.

- 初期値がすべて0の場合より悪い影響を与えるバイアスとなる.
- 局所最適解を多数発生させ，学習を正しく収束させない.

例として5.1節と同様の条件で，環境が変化したときに前環境で獲得された行動価値を新たな環境での学習の初期値として用いる実験を行った．このとき環境$E4$において**Fig.11**に示すように成功率が収束しない現象が見られた．成功率をみると振動しており，局所最適な行動価値が生じていることが推測される.

以上のように行動価値を直接再利用すると学習時間の増大や学習未収束が生じるため，本手法では行動政策のみを再利用する．そのため新たな環境では行動価値は0から計算されるが，行動政策を修正しない状態では政策の探索を行う必要がないため，学習時間は通常のQ学習と比較して短縮される.

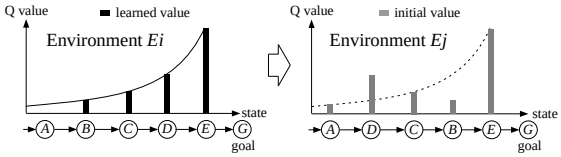


Fig. 10 Transferring Q-values

6. おわりに

本論文では，ロボットが過去の環境で獲得した行動政策を新たな環境でタスクが達成できるように修正することにより，環境の変化に適応する手法を提案した．本手法では最終的に複数の環境の相違を吸収した行動政策が獲得されることをシミュレーションおよび実機による実験により検証した.

本手法ではタスクに必要な状態変数が変化するような環境の変化は考慮されていない．すなわち環境クラス（あるいは環境集合）はタスクに必要な状態変数によって規定されている．本論文では(4)式の状態変数を持つロボットが適応可能な環境の変化として，障害物の配置，個数の変化を設定した．しかし，たとえば移動障害物が存在する環境では，状態変数に移動障害物の速度を表現する成分が必要と考えられるため，(4)式の状態変数を持つロボットはそのような環境の変化には適応できないと考えられる．このような環境の変化に適応するためには，行動政策を部分的に修正するだけではなく，状態変数を（部分的に）変化させる必要がある.

さらに本手法で獲得された不変知識は，異なる状態空間の間では不変とならないため，どのように不変知識を伝達するべきかが問題となる．この問題に関しては異なる状態空間を持つ行動政策を統合したときに，再学習領域と再利用領域を分割する手法[12]が応用できる可能性がある．また新たな環境クラスに対する状態変数の構成方法も問題となる．さらに異なる状態空間の間での知識伝達が可能になれば，別のロボットへの知識伝達の可能性も生まれる．これらは今後の課題である.

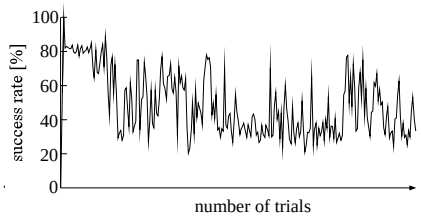


Fig. 11 Transition of success rate

参 考 文 献

[1] J. H. Connel and S. Mahadevan: "Robot learning," Kluwer Academic Publishers, 1993.
[2] S. Thrun and T. Mitchell: "Lifelong robot learning," Technical Report IAI-TR-93-7, University of Bonn, Dept. of CS III, 1993.
[3] S. Thrun: "A lifelong learning perspective for mobile robot navigation," Proc. of the IEEE/RSJ/GI Conference on Intelligent Robots and Systems, pp. 23–30, 1994.
[4] S. Thrun and J. O'Sullivan: "Discovering structure in multiple learning tasks: The tc algorithm," Proc. of the thirteenth International Conference on Machine Learning, 1996.
[5] F. Tanaka and M. Yamamura: "An approach to lifelong reinforcement learning through multiple environments," 6th European Workshop on Learning Robots, pp. 93–99, 1997.
[6] 田中，山村："Lifelong agentの強化学習"，日本機会学会，ロボティクス・メカトロニクス講演会'98講演論文集, 1998.
[7] C. J. C. H. Watkins and P. Dayan: "Technical note: Q-learning," Machine Learning, vol. 8, pp. 279–292, 1992.
[8] 浅田，野田，細田："ロボットの行動獲得のための状態空間の自律的構成"，日本ロボット学会誌，vol. 15, no. 6, pp. 886–892, 1997.
[9] 高橋，浅田："実ロボットによる行動学習のための状態空間の漸次的構成"，日本ロボット学会誌，vol. 17, no. 1, pp. 118–124, 1999.
[10] T. Nakamura, J. Morimoto and M. Asada: "Direct coupling of multisensor information and actions for mobile robot behavior acquisition," Proc. of 1996 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration, pp. 139–144, 1996.
[11] S. D. Whitehead and D. H. Balard: "Active perception and reinforcement learning," Proc. of Workshop on Machine Learning, pp.179–188, 1990.
[12] E. Uchibe, M. Asada and K. Hosoda: "Behavior coordination for a mobile robot using modular reinforcement learning," Proc. of 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp 1329–1336, 1996.

港 隆史 (Takashi MINATO)
1974年2月13日生まれ．1998年大阪大学大学院工学研究科博士前期課程修了．現在，同大学博士後期課程在学中．知能ロボットの行動獲得に関する研究に従事．（日本ロボット学会学生会員）

浅田 稔 (Minoru ASADA)
1982年大阪大学大学院基礎工学研究科後期課程修了．同年，大阪大学基礎工学部助手．1989年大阪大学工学部助教授．1995年同教授．1997年大阪大学大学院工学研究科知能・機能創成工学専攻教授となり現在に至る．この間，1986年から1年間米国メリーランド大学客員研究員．知能ロボットの研究に従事．1989年，情報処理学会研究賞，1992年，IEEE/RSJ IROS'92 Best Paper Award受賞．1996年日本ロボット学会論文賞受賞．博士（工学）．ロボカップ国際委員会副委員長，ロボカップ日本委員会委員長，電子情報通信学会，情報処理学会，人工知能学会，日本機械学会，計測自動制御学会，システム制御情報学会，IEEE R&A, CS, SMC societiesなどの会員．（日本ロボット学会正会員）