学術・技術論文

複数の学習器の階層的構築による行動獲得

高　橋　泰　岳[*1]　浅　田　　稔[*2]

Behavior Acquisition by Multi-Layered Reinforcement Learning

Yasutake Takahashi[*1] *and* Minoru Asada[*2]

This paper proposes multi-layered reinforcement learning by which the control structure can be decomposed into smaller transportable chunks and therefore previously learned knowledge can be applied to related tasks in a newly encountered situations. The modules in the lower networks are organized as experts to move into different categories of sensor output regions and to learn lower level behaviors using motor commands. In the meantime, the modules in the higher networks are organized as experts which learn higher level behavior using lower modules. We apply the method to a simple soccer situation in the context of RoboCup, show the experimental results, and give a discussion.

Key Words: reinforcement learning, hierarchical control structure, RoboCup

1. はじめに

近年，環境とエージェントの相互作用を通して学習する手法として強化学習 [9] や記憶に基づく学習法などのロボット学習 [3] が注目されている．しかし実世界において，一つの学習器で複雑な環境，複数のタスクを対象とした場合，学習器が複雑膨大になると予想され，学習時間の増大，学習結果の再利用の困難さなど，さまざまな現象を理解し行動を獲得することは限界がある．

実際には，ロボットに適用する複雑なタスクの多くはいくつかのサブタスクに分解できることが多い．そのため様々な行為の選択や多入力多出力制御に対応するために，複数の制御器を階層的に構成する手法が提案されている．複数の制御器を使えば，一つ一つの制御器は少数のセンサ出力と制御入力を担当すればよい．また層にすることでタスクを各層ごとに分解できるので，それぞれの制御器は簡単で小さなもので十分と考えられ，設計が用意となる．また特定のタスクではなく，様々なタスクを対象とする場合，個々のタスクに対する最適性よりも複数のタスク全体の遂行が優先されるので，階層型制御機構は有効な手法と考えられる．

Albusは各階層がタスク分解，世界モデル，センサ処理モジュールで構成されるシステム (Real-time Control System) [1] を提案した．上位のモジュールは下位のモジュールを利用し，前もって分解されたサブタスクを遂行するシステムである．Kaelbling [4] は HDG アルゴリズムを提案し，上位の層としてランドマークのネットワークを作り，学習を高速化する事に成功している．Stone and Veloso [6] は階層型学習を提案し，マルチエージェントの学習システムに適用している．下位では個々のエージェントのスキルを学習し，上位ではチームワークとしてのスキルを学習する．Morimoto and Doya [5] も二層の階層型強化学習を用い，上位では適切なサブゴールのシーケンスを学習し，下位ではそれぞれのサブゴールへ到達するためのスキルを獲得している．Whitehead et al. [10] は複数のサブゴールが与えられたとき，それぞれのサブゴールに学習モジュールを割り当てることによって学習を速めている．しかし，ここで問題なのは，サブタスク，ランドマーク，スキル，サブゴール等を決定する事である．これまで提案されてきた手法は，その階層構造やそれぞれの階層における役割，それぞれのモジュールへの分担などを全て人間が明示的に構造化した上で，それぞれのモジュールを設計する，もしくは一部を学習させるものが多い．

最近，Tani and Nolfi [8] は明示的に階層ごとの役割を指定することなく，全ての階層において同じ構造のモデルを用いる階層構造を提案し，ロボットのセンサ・モータコマンドシーケンスの予測を可能にしている．学習モジュールはリカレントニューラルネットで構成されており，ロボットが経験するセンサとモータコマンドのシーケンスを学習する．それぞれのモジュールはお互いに異なった範囲のシーケンスを学習し，担当の範囲に入った時，高い活性度を持つ事になる．一方，上の層では別のリカレントニューラルネットワークが下の層の活性度の遷移を学習する．この考え方は興味深いが，提案された手法は，行動選択部分が欠落しており，このままではロボットによる自律的な行

原稿受付
[*1] 大阪大学大学院工学研究科
[*2] 大阪大学大学院工学研究科
[*1] Graduate School of Engineering, Osaka University
[*2] Graduate School of Engineering, Osaka University

動獲得はできない．

本論文では，同一構造の学習器を複数用いて階層的に構築することによる行動獲得法を提案する．下位の層の学習器はそれぞれ異なったサブゴールを担当し，低レベルな行為を学習する．同時に上位の学習器は，下位の学習器を利用し，より高いレベルの行為を学習する．従来の手法とは異なり，それぞれの学習器が担当するサブゴール，タスクは自律的に決定され，また階層も自律的に構成される．提案する手法をロボカップ [2] に出場しているロボットに適用した結果を示す．

2. 行 動 学 習 器

まずはじめに，目的の行動を獲得するための学習器を用意する．ここでは強化学習の一手法として広く利用されている Q 学習 [9] を連続状態行動空間に拡張した Continuous Q learning [7] を用いる．Q 学習は確率的動的計画法に基づいており，環境モデルを必要としない強化学習の一手法で，マルコフ過程が成り立つ環境下で，最適行動を獲得できる．

ここで Q 学習について簡単に述べる．ロボットが識別することができる状態の集合を $\mathbf{S}$ とし，環境に対してとり得る行動の集合を $\mathbf{A}$ 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，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政策を見つけることである．

Q 学習では，状態 $s \in \mathbf{S}$ において行動 $a \in \mathbf{A}$ をとり，次状態 s' に遷移した時，行動価値関数値 $Q(s, a)$ を以下のように更新する．

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma V(s') - Q(s,a)), \quad (1)$$

$$V(s) = max_{a' \in \boldsymbol{A}} Q(s,a'), \quad (2)$$

ここで，α は学習率，γ は減衰係数である．$V(s)$ は状態価値関数と呼ばれる．十分な経験を積んだ後，Q 値が得られると，各状態 s に対して $Q(s, a)$ が最大となる行動 a を選ぶことによって政策が定義される．

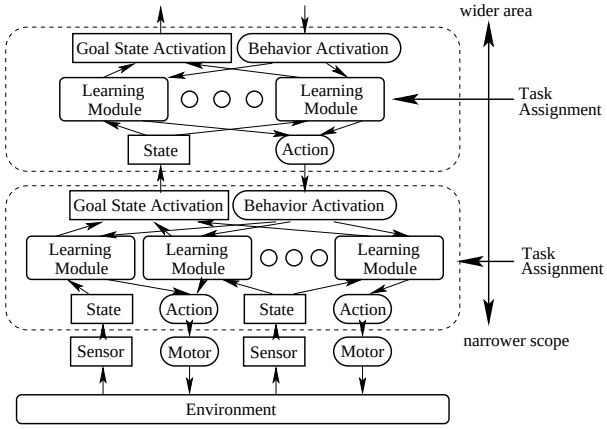
通常，報酬はゴール状態になった時のみロボットに与えられることが多い†．このとき $V(s)$ はゴール状態までの近さを表現していると直観的に考えることができる．なぜなら $V(s)$ はロボットが最適の行動をとったときに得られる報酬の減衰和を予測している値だからである．

3. 階層型学習機構

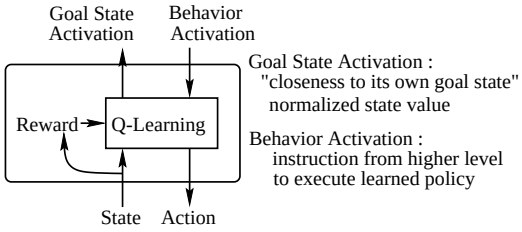
前節で述べた行動学習器を複数並べて層を作り，これを階層的に構築する．この学習器の階層構造はタスク分解の役割を担っていると見なすことができる．下位の学習器は，与えられた環境下で狭い範囲を探索し，より低レベルで基本的な行動を獲得する．一方，上位の学習器は下位の学習器を利用することにより，より広い範囲を探索し，より高レベルで抽象化された行動を獲得する．

複数の学習器を実際のタスクに適用する際に問題になる点は，

†吸収ゴール (absorbing goal) と呼ばれている



(a) A whole system



(b) A behavior learning module

Fig. 1 A hierarchical learning architecture

それぞれの学習器に対してサブタスクをどのように指定し，分配するかである．これまで提案されてきた手法の多くは，サブタスクを設計者が彼らの経験や直観を頼りにあらかじめ決定してきた．

本手法において，タスクをサブタスクに分解することは，ある層の学習器にそれぞれゴール状態を分配することに等しい．なぜなら，各々の学習器にとって自分自身に割り当てられたゴール状態へ遷移することが，割り当てられたサブタスクを遂行することになるからである．

それぞれの学習器へゴール状態を自律的に分配するための方針は,「状態空間の中でそれぞれのゴール状態を一様に分布させる」ことである．それぞれの学習器が，学習すべき自分のゴール状態を他の学習器のゴール状態から離れるように移動させれば，結果としてこの方針に沿うことになる．これにより，従来の手法のようにサブゴールを設計者が前もって設計しておく必要がなくなる．以下では本手法の詳細を述べる．

3.1 アーキテクチャ

Fig.1 に提案する階層型学習機構を示す．Fig.1(a) は階層構造を示しており，Fig.1(b) は行動学習器を示している．それぞれの学習器は状態と行動を認識し，それぞれのゴール状態により報酬を発生させ，これをもとに Q 学習を用いて行動を学習する．最下位の層の学習器の状態，行動はロボットの持つセンサ情報やモータコマンドを使って構成する．

上位の学習器と情報のやりとりをするために，**ゴール状態活性度**と**行動活性度**を用意する．ゴール状態活性度は正規化した (ゴール状態を 1 とした) 状態価値 $V(s)$ であり，この値は下位のそれぞれの学習器が各々のゴール状態にどれだけ近いかを示

しており，この情報が上位の学習器に伝わる．行動活性度は，その学習器が獲得した行動を出力するための上位の学習器から与えられる指令であり，これを受け取った学習器は Q 学習によって得られた状態行動価値関数から最適行動を計算し，下の層に出力する．

学習器の状態価値 $V(s)$ は先で述べたように，その学習器自身のゴール状態への近さを表現しているので，下位の層の学習器のゴール状態活性度のパターンが上位の層の状況を表現していると考えることができる．すなわち，上位の学習器は下位の学習器のゴール状態活性度のパターンを基に状態空間を張る．また上位の学習器の行動空間は下位の学習器への行動活性度パターンで構成する．詳しくは 3.2.2 で述べる．これにより上位の層の学習器は下位の学習器が獲得した行動を利用し，より視野の広い長期的な行為を獲得することができる．

3.2 アルゴリズム

3.2.1 状態行動の連続的表現と行動学習

各学習器は **2** 節で述べた Q 学習を拡張した Continuous Q learning を用いて行動を獲得する．ここで Continuous Q learning [7] について簡単に述べる．

まず状態行動空間を適当な粗さで離散化し，離散化された点をそれぞれ代表状態，代表行動とする．Q 学習における状態，行動をそれぞれ各代表状態，各代表行動の適合度ベクトル $(w_1^s, \cdots, w_n^s)$，$(w_1^a, \cdots, w_m^a)$ として表現する．これらの適合度ベクトルはそれぞれの代表状態，代表行動への近さを表しており，合計が 1 になるように正規化する．

代表状態 s_i と代表行動 a_j における Q 値を $Q_{i,j}$ とすると，任意の状態，行動に置ける Q 値は次式で表現される．

$$Q = \sum_{i=1}^{n}\sum_{j=1}^{m} w_i^s w_j^a Q_{i,j} \qquad (3)$$

代表状態 s_i が与えられた場合，最適代表行動は $\arg\max_j Q_{i,j}$ で計算される．任意の状態 $\boldsymbol{s}$ における最適行動 $\boldsymbol{a}^*$ は次式で与えられる．

$$\boldsymbol{a}^* = \boldsymbol{w}^{a*} = \sum_{i=1}^{n} w_i^s e(\arg\max_j Q_{i,j}) \qquad (4)$$

ただし $e(k)$ は第 k 要素だけが 1 である m 次元のベクトルである．最適行動を式 (4) で決定するため元の Q 学習の $\max Q$ の計算は次式が妥当である．

$$\max Q = \sum_{i=1}^{n}\sum_{j=1}^{m} w_i^s w_j^{a^*} Q_{i,j} \qquad (5)$$

したがって，状態 $\boldsymbol{s}$ で行動 $\boldsymbol{a}$ をとり次状態 $\boldsymbol{s}'$ へ遷移し，報酬 r が得られたとき，Q 値の更新は式 (1) を修正した次式となる．

$$Q_{i,j} \leftarrow Q_{i,j} + \alpha_t w_i^s w_j^a (r + \gamma V(\boldsymbol{s}') - Q(\boldsymbol{s}, \boldsymbol{a})) \qquad (6)$$

ただし $V(\boldsymbol{s})$ は状態 $\boldsymbol{s}$ において最適行動 $\boldsymbol{a}^*$ をとったときの Q 値である．

3.2.2 状態/行動空間の構成

最下位層の学習器は通常の強化学習と同様に，ロボットの持つセンサ情報/モータコマンドを離散化し，状態/行動空間を構成する．本研究では [7] と同様にそれぞれのセンサ空間/モータコマンド空間を離散化し，代表状態/代表行動の適合度ベクトルを計算した．

最下位層よりも上の層の学習器の場合，下位の層の学習器のゴール状態活性度/ 行動活性度が上の層の学習器のそれぞれ代表状態/代表行動の適合度に 1 対 1 に相当する．つまり下位の層の学習器のゴール状態活性度/行動活性度パターンを正規化したものが上位の層の学習器の代表状態/代表行動の適合度ベクトルになる．

3.2.3 ゴール状態の自律的振り分け

基本方針は「ある層の全状態空間をその層に割り当てられた学習器で一様に振り分ける」である．

このときの問題は

① 全状態空間を事前に定義できない．

② 状態間の距離をあらかじめ知ることができない

であり，特に最下位より上の層で問題となる．

そこで，学習器のゴール状態活性度をお互いの学習器間の距離関数として用いることでこれらの問題に対処する．なぜならば，ゴール状態に到達したときにのみ正の報酬が与えられ，それ以外はゼロの報酬が割り当てられる場合,「学習の結果得られたある状態の Q 値の最大値 (状態価値) はその状態からゴールまでの距離の近さを表現している」と考えられるからである．

それぞれの学習器のゴール状態を一様に振り分けるための具体的な手順を Fig.2 を用いて説明する．ここでは簡単のために一次元の状態空間を考えるが，これは多次元になっても同様である．縦軸にそれぞれの学習器のゴール状態活性度を示す．Fig.2(a) では三つの学習器があるが，これは学習の初期段階なので分布が偏っている．そこでゴール状態活性度が低い所はその状態を担当する学習器が無いと判断し，その状態をゴール状態として持つ新しい学習器を追加する (Fig.2(b))．また学習器の密度の大きい所では，それぞれのゴール状態活性度が高いので，ある学習器のゴール状態が他のゴール状態から離れるように，つまり他の学習器の状態価値が小さくなる方向にゴール状態を移動させる (Fig.2(c))．それでも尚，密度が高い場合は学習器を削除する手続きを用意する．この結果，学習が進めば，Fig.2(d) のようにそれぞれの学習器のゴール状態は状態空間上に一様に分布するようになる．以下に，その手続きを示す．

(1) 現時刻で閾値 T よりも適合度が大きい代表状態を集め，これを $s^{neighbor} \in S^{neighbor}$ とする．
(2) もし $||S^{neighbor}|| = 0$ ならば，終了．ただし $||\cdot||$ は要素の数を表す．
(3) $S^{neighbor}$ 中にゴール状態を持つ学習器 $module_{query}$ を探す．
(4) $module_{query}$ 以外の学習器の中で最大のゴール状態活性度の分布 $V_{\max}^{noquery}(s^{neighbor})$ を求める．
(5) $module_{query}$ がなく，かつ $V_{\max}^{noquery}(s^{neighbor})$ の最大値が小さければ新しい学習器を生成して，終了．
(6) $module_{query}$ があり，かつ $V_{\max}^{noquery}(s^{neighbor})$ の最小値が大きければその学習器を削除して，終了．

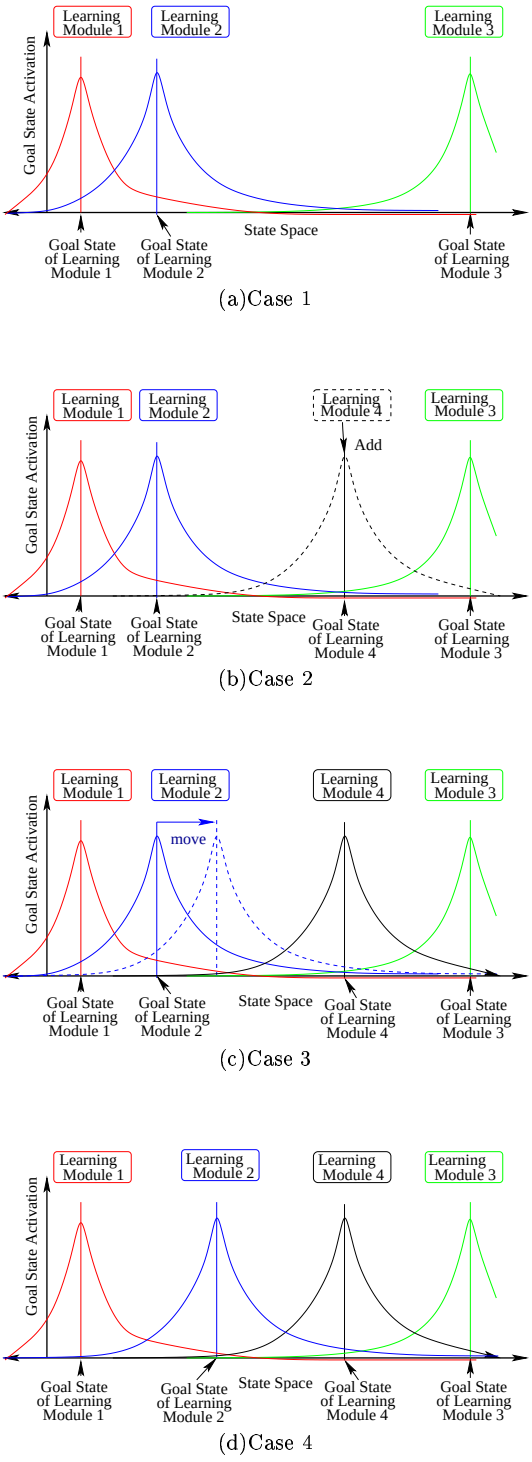


Fig. 2 Example of the assignment of the goal state among learning modules

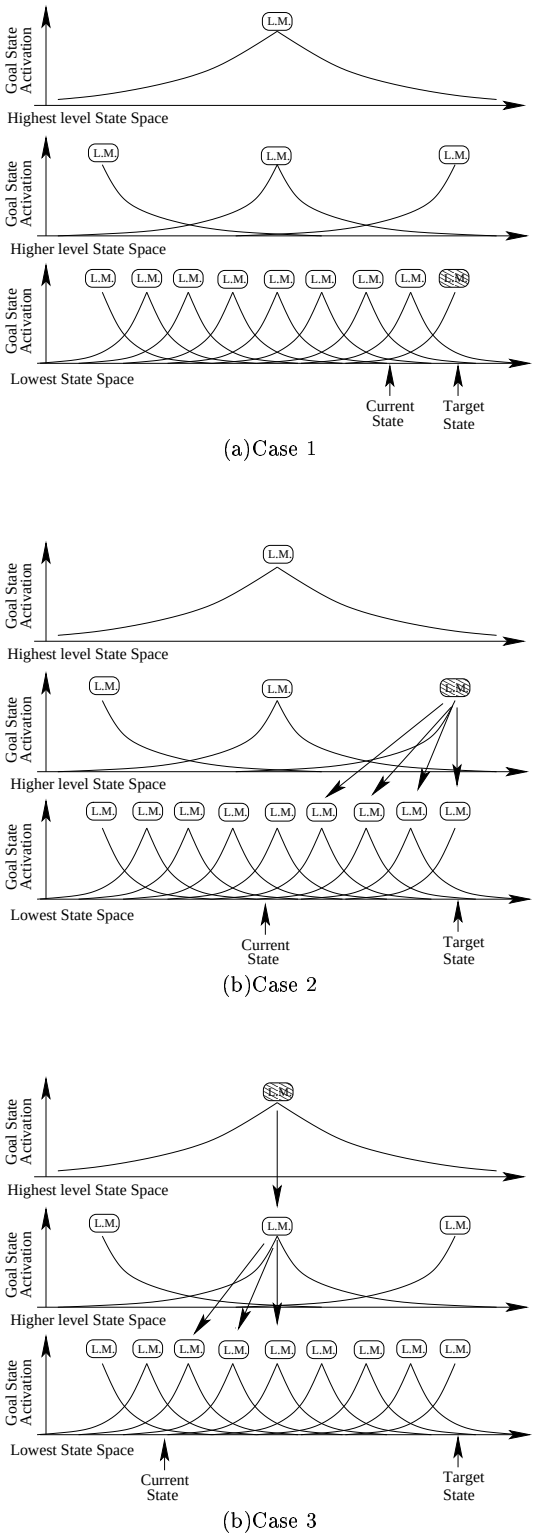


Fig. 3 Strategy in the multi-layered control structure. L.M. stands for learning module

(7) $S^{neighbor}$ の中で $V_{\max}^{noquery}(s^{neighbor})$ が最小となる状態 $s_{\min}^{neighbor}$ を探す．$module_{query}$ があり，そのゴール状態が $s_{\min}^{neighbor}$ でなければゴール状態を $s_{\min}^{neighbor}$ へ移動する．

3.2.4 階層の構築

各層における学習器の分配については前節で述べた．この層を複数重ねて階層化する．上位の層の学習器は下位の層の学習器を基に状態/行動空間を張る．状態数よりも少ない学習器が割り当てられるので，状態の数は上位の層の方が少ない．また上位の層にいく程，学習器の数が減っていき，最終的には一つになる．この時点で階層化が終了する．

3.2.5 タスク遂行時の階層型学習機構内での行動戦略

システムに与えられる目標状態は最下位層の状態空間の中で与えられる．最下位層の中で与えられた目標状態に一番近いゴール状態を持つ学習器を探す．この学習器で現在の状況から目標状態に到達できるなら，この行動活性度を上げることで起動させる (Fig 3(a)).「タスクを達成できる」か「できない」かの判断は，その学習器の Q 値によって判断する．つまり Q 値が高ければその状態からゴール状態へ行くパスがあり，逆に低ければこの状態が学習されていないか，ゴール状態から非常に離れているかになる．ここでは，Q 値がある閾値を越えれば「タスクを達成できる」と判断する．

最下位層のなかで与えられた目標状態 s_{target}^0 に一番近いゴール状態を持つ学習器 $module_g^0$ で対応できない場合，一つ上の層で $module_g^0$ に対応する状態を目標状態 s_{target}^1 とし，これに一番近い学習器 $module_g^1$ を探す．この学習器 $module_g^1$ で現在の状況から目標状態 s_{target}^1 まで到達できるなら，この行動活性度を上げることで起動させる．この学習器 $module_g^1$ は最下位層の学習器に対して行動活性度という形で指令を送ることで，自分のゴール状態へ移動する (Fig.3(b))．最下位層の学習器 $module_g^0$ で対応できる状態まで遷移し，Fig.3 の Case 1 の状況になれば，さきほどと同様に学習器 $module_g^0$ を起動し，与えられた目標状態へ移動する．

学習器 $module_g^1$ で対応できない場合は，更に上の層にいき，おなじことを繰り返す (Fig.3(c))．また学習器間の競合が起こらないように行動活性度が立ち上がる学習器は一つだけにする．これは式 (4) で求められる行動の適合度ベクトル $\boldsymbol{w}^{a*}$ のなかで最大の適合度を持つ行動に対応する学習器の行動活性度だけを上げることで実現する．

4. 実　　験

4.1 問題設定

提案する手法がどのように各学習器を分配し，階層を作り，与えられたタスクを遂行するかを検証するために，移動ロボットの簡単なナビゲーション実験を行った．目標状態は，実際に移動ロボットを目的地まで移動させ，センサ情報を読み込ませることで与える．

Fig.4 に使用するロボットとボール，ゴールを示す．Fig.5 にロボットの簡単なシステムを示す．ロボットはセンサとして広角レンズを装着した CCD カメラと全方位ミラーを装着したカメラを持ち，二枚の画像処理ボードを使って実時間でボールやゴールの重心を抽出する．カメラの搭載位置のため，広角レンズを装着したカメラはロボット前方を，全方位ミラーを装着したカメラはロボットの側方と後方を観測することになる．また移動機構は左右独立駆動機構である．

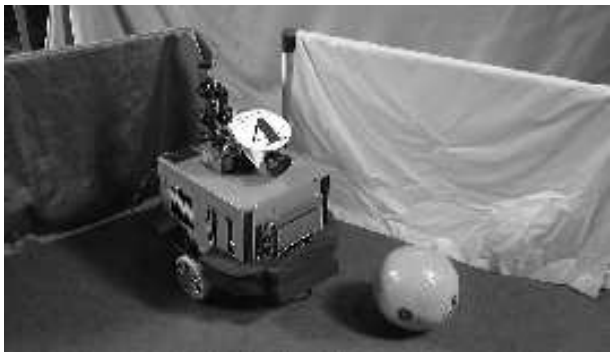


Fig. 4 A mobile robot, a ball and goals

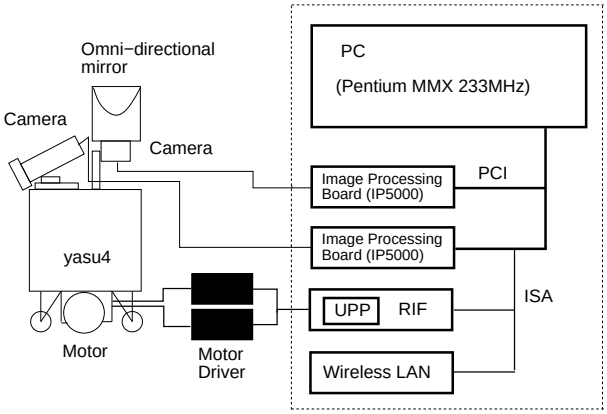


Fig. 5 An overview of the robot system

この実験では，簡単のため一つのゴールだけを知覚するようにした．Fig.6 に示すように，最下位層の学習器の状態空間は二つのカメラから得られた画像上のゴールの座標で構成され，それぞれの空間を 9×9 で離散化した．また行動空間は左右の車輪のモータコマンドに与える指令値で構成され，これも 3×3 で離散化した．前方を担当するカメラに物体が写っているときは，全方位ミラーつきカメラからの情報を無視し，前方担当のカメラからのセンサ情報を優先した．従って，最下位層の代表状態数は $162(9 \times 9 \times 2)$，代表行動数は $9(3 \times 3)$ である．それより上の層の状態や行動は，自分より一つ下の層の学習器のゴール状態活性度と行動活性度によって構成される．それは自律的に割り当てられる学習器によって最終的に決定される．

4.2 実験結果

実験は，学習段階と，学習結果を用いたタスク遂行の段階の二つの段階からなる．まず，ロボットは約 2 時間，環境の中をランダムに移動し学習した．Fig.6 に示すように，最上位層に一つの学習器を持つ 4 層の階層が得られた．ここでは一番下の層から最下位層，中間層，上位層，最上位層と呼ぶ．この実験では最下位層に 40，中間層に 15，上位層に 4，最上位層に 1 の学習器が割り当てられた．Fig.7，8 にそれぞれゴールが前方担当カメラ，全方位ミラー付きカメラの画像上に写ったときの最下

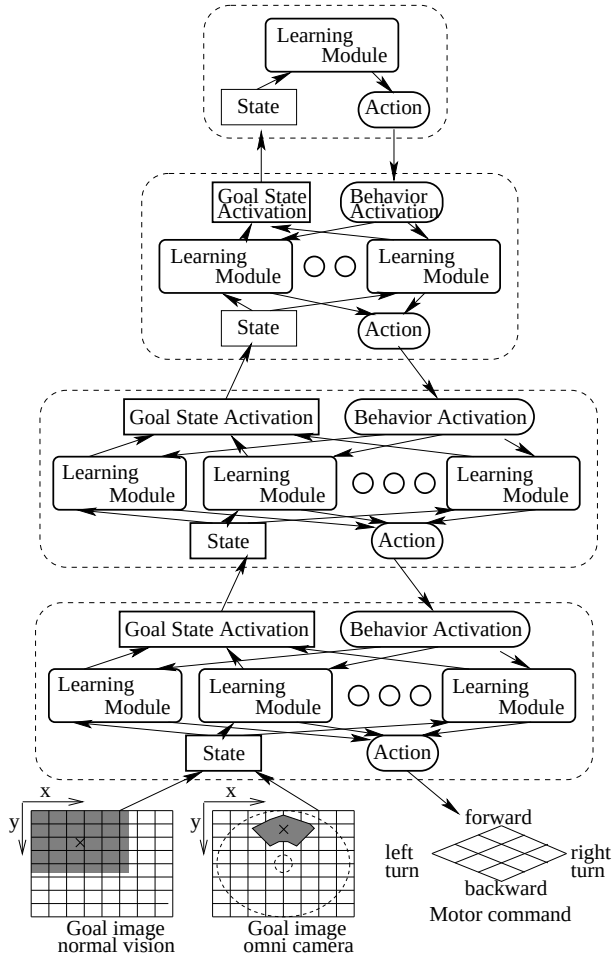


Fig. 6 A hierarchy architecture of learning modules

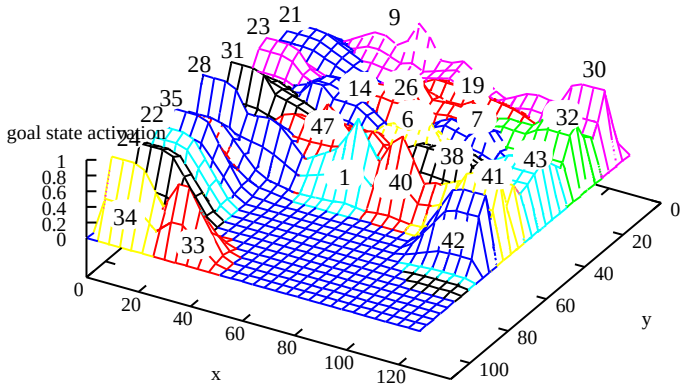


Fig. 7 The distribution of learning modules at lower layer on the normal camera image

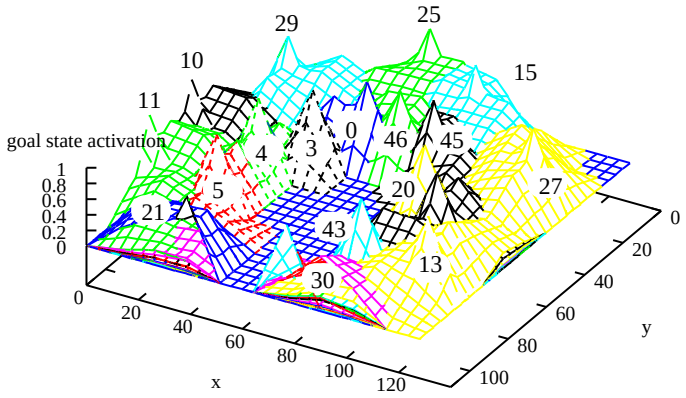


Fig. 8 The distribution of learning modules at lower layer on the omni-directional camera image

位層のそれぞれの学習器のゴール状態活性度の分布を示す．それぞれ x, y 座標は画像上のゴールの中心座標に対応している．図中の数字はそれぞれ学習器に割り振られた番号である．それぞれの学習器はとくに偏ったりせずに振り分けられていることがわかる．

この階層型学習機構を使って，指定の場所までいくタスクを遂行させた．ロボットをゴールと反対の方向を向かせ，ゴールから遠方に置いた状況を初期状態とし，ゴールに近付いてそれを正面に見る状況を目標状態として与えた．Figs.9，10，11 にそれぞれ最下位層，中間層，上位層のゴール状態活性度と行動活性度の遷移を示す．最上位層は学習器が一つで，一度も行動活性度が起動していないので，ここでは割愛した．行動活性度は図の上部に線分図として表示している．Fig.9 の数字はそれぞれ最下位層の学習器に割り振られた番号であり，Figs.7，8 の番号と同じ学習器が対応している．Fig.12 に最下位層，中間層，上位層における状態の遷移，下位の学習器への指令などの概略図を示す．この図は Figs.9，10，11 と対応している．丸はそれぞれ学習器であり，丸に囲まれた数字は，それぞれの層の学習器に割り当てられた番号を表している．上に向かっている点線の矢印は，矢印のもとの下位の学習器に対応する状態を矢印で指された上位の学習器がゴール状態としているということを表す．実線の細い矢印は，タスク遂行中のゴール状態活性度が高くなった学習器の遷移を表している．下に向けられた実線の太い矢印は，上位の学習器が下位の学習器を起動している様子を表す．初期状態では，最下位層では学習器 25，中間層では学習器 10，上位層では学習器 1 のゴール状態に近い．与えられた目標状態は，最下位層では学習器 1，中間層では学習器 7，上位層では学習器 0 のゴール状態が近い．まず，ロボットは最下位層の学習器 1 を起動しようとするが，この学習器の担当範囲外なので，中間層の学習器 7 を次に起動しようとする．しかしこの学習器も担当範囲外なので，更に上位の学習器 0 を起動する．約 40 ステップまで，上位層の学習器 0 は中間層の学習器 15 を起動し，この学習器が更に最下位層の学習器 27，13 を起動する．中間層の学習器 7 が対応できる状況まで来たら，約 360 ステップまで，この学習器が最下位層の学習器 30，26 を起動する．最終的に最下位層の学習器 1 が担当でいる範囲に来ると，この学習器だけが起動し目標状態へ遷移する．

5. お わ り に

本論文では同一構造の学習器を複数用いて階層的に構築することによる行動獲得法を提案し，この手法を実機に適用した結果を示した．今後は階層をより多層化し，より複雑なタスクに適用しすることで本手法の有効性を確かめたい．また複数のタ

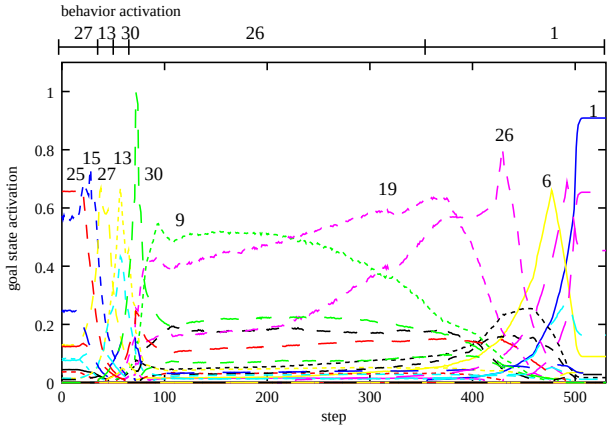


Fig. 9 A sequence of the goal state activation and behavior activation of learning modules at lower layer

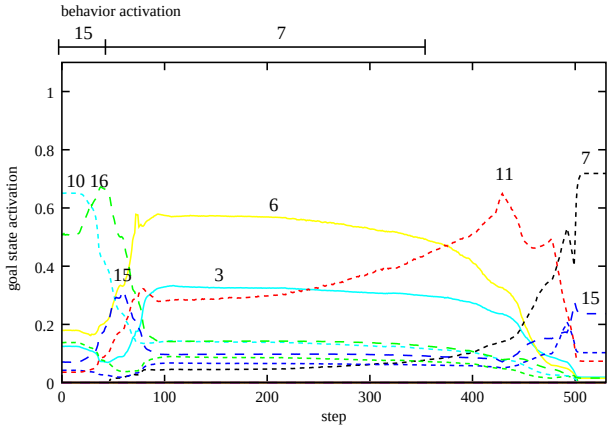


Fig. 10 A sequence of the goal state activation and behavior activation of learning modules at middle layer

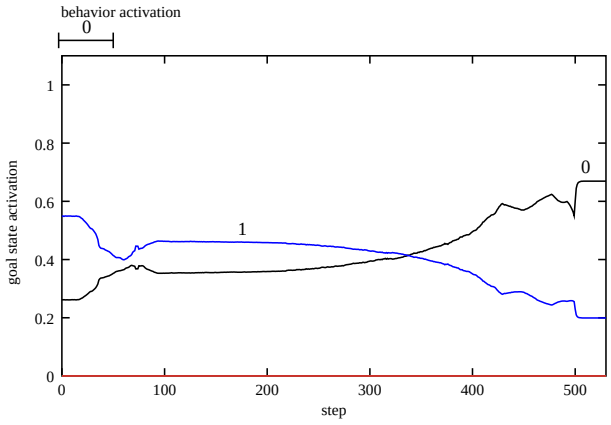


Fig. 11 A sequence of the goal state activation and behavior activation of learning modules at upper layer

スクを学習するとき，以前の学習結果を再利用する事による有効性を検証予定である．

謝辞 本研究は科学技術振興事業団の戦略的基礎研究推進事業「脳を創る」中村プロジェクトの援助を受けた．

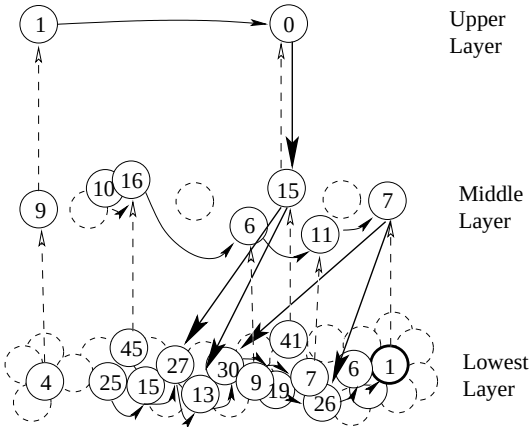


Fig. 12 A rough sketch of the state transition on the multi-layer learning system

参 考 文 献

[1] James S. Albus. The engineering of mind. In *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (From Animals to Animats 4)*, pp. 23–32. MIT Press, 1996.

[2] M. Asada, H. Kitano, I. Noda, and M. Veloso. Robocup: Today and tomorrow – what we have learned. *Artificial Intelligence*, pp. 193–214, 1999.

[3] J. H. Connel and S. Mahadevan. "Introduction to robot learning". In J. H. Connel and S. Mahadevan, editors, *Robot Learning*, chapter 1. Kluwer Academic Publishers, 1993.

[4] Leslie Pack Kaelbling. Hierarchical learning in stochastic domains: Preliminary results. In *Proceedings of the Tenth International Conference on Machine Learning*, 1993.

[5] Jun Morimoto and Kenji Doya. Hierarchical reinforcement learning of low-dimensional subgoals and high-dimensional trajectories. In *The 5th International Conference on Neural Information Processing*, Vol. 2, pp. 850–853, 1998.

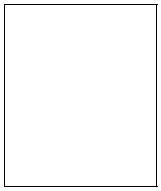
[6] Peter Stone and Mamuela Veloso. Layered approach to learning client behaviors in the robocup soccer server. *Applied Artificial Intelligence*, Vol. 12, No. 2-3, 1998.

[7] Yasutake Takahashi, Masanori Takeda, and Minoru Asada. Continuous valued q-learning for vision-guided behavior acquisition. In *Proceeding of the 1999 IEEE International Conference on Multisenso Fusion and Integration for Intelligent Systems*, pp. 255–260, 1999.

[8] J. Tani and S. Nolfi. Self-organization of modules and their hierarchy in robot learning problems: A dynamical systems approach. Technical report, Sony CSL Technical Report, SCSL-TR-97-008, 1997.

[9] C. J. C. H. Watins and P. Dayan. "Technical note: Q-learning". *Machine Learning*, Vol. 8, pp. 279–292, 1992.

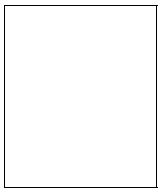
[10] Steven Whitehead, Jonas Karlsson, and Jsho Tenenberg. Learning multiple goal behavior via task decomposition and dynamic policy merging. In Jonathan H. Connell and Sridhar Mahadevan, editors, *ROBOT LEARNING*, chapter 3, pp. 45–78. Kluwer Academic Publishers, 1993.



高橋 泰岳

1972 年 12 月 13 日生．1994 年大阪大学大学院学研究科博士前期課程修了．2000 年同大学博士後期課程中退，同年同大学大学院工学研究科助手．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

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）．日本ロボット学会，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）