3B2-R-2-6

モジュール組換え型モデルにおけるモジュールの学習とモジュール組換え系列の学習

Learning functions of modules and their free combinations in a module recombination model

坂本裕太*1 坂戸達陽*1 尾関基行*1 岡夏樹*1
Yuta Sakamoto Tatsuya Sakato Motoyuki Ozeki Natsuki Oka

*1京都工芸繊維大学 大学院工芸科学研究科
Graduate School of Science and Technology, Kyoto Institute of Technology

In the study of the intelligent systems with module structure, there have been attempts to solve complex problems by combining simple modules. We proposed a model in which modules are freely combined, and studied the learning of the series of combinations of modules, but the functions of the modules were fixed. We considered that problems which could not be solved by fixed modules will be solved by learning the functions of modules. In this paper, we therefore propose a method in which the learning of the functions of the modules and the learning of the combinations of them occur simultaneously. Experimental results show that problems which cannot be solved only by the learning of module combinations can be solved by the proposed concurrent learning, and that the concurrent learning performs more flexibly than the sequential learning of the functions of modules and their combinations.

1. はじめに

問題解決を行う知的エージェントにおいて，未知の環境や変動する環境に対処するには，ロボットに環境に対応した行動を学習させる必要があり様々な研究が行われている．その中でも，重要な機能をモジュール化し，それらを組み合せることで問題を解く研究がある [小川 04, Wolpert 98, Jacobs 91, 高橋 09]．ロボットにそれらのモジュールの組み合せ方を学習させることで，環境に対応させようとする手法である．その中で我々は，モジュールの組み合せ方を限定しないことを特徴とするモデル [Oka 99] を提案し，設計者の想定外の処理をも獲得できることを示してきた [本多 09, 坂本 10]．また，モジュールの組み合せ方を限定しないことで生じるモジュールの組み合せ爆発という問題に対して，人から与えられる言語教示を利用してモジュールの組換え系列の学習を容易にする方法を提案した [本多 11]．しかし，本多らの研究では，自由なモジュールの組み合せ方を許す一方で，モジュール自体の機能は固定であった．

そこで本研究では，モジュールの機能の学習，および，モジュールの組換え系列の学習を同時に行う手法を提案し，シミュレーションを行い，評価する．障害物回避タスクを用い，モジュールの機能の学習，および，モジュールの組換え系列の学習を逐次的または同時に行い，各場合についてどのように学習が進むかについて比較を行う．同時に学習を行うことで，予想していたモジュールより良いモジュールができたり，予想とは別の仕方でモジュールが分かれたりすることを実験的に示す．

2. モジュール組換え型モデル

本研究で用いたモジュール組換え型モデルの構成を図 1 に示す．モデルは，大きく制御部と機能部に分かれる．機能部に位置するモジュールは，入力部と方策部にそれぞれ 2 つ，出力部に 1 つの合計 5 つから構成され，それぞれがワーキングメモリを介して繋がっている．このモデルは，制御部からの制御信号で機能部の各モジュール間の結合の on/off を操作し，各モジュール結合が on にされることでワーキングメモリを介してモジュールが機能する．以降，場面ごとに on にするモジュールを制御信号によって変えていくことをモジュールの組換えと呼ぶ．入力部はモジュールごとに対応したセンサーの値を処理するモジュールで，値は状態メモリに一時的に記憶される．方策部は状態メモリの値を読み取って，ロボットのとるべき行動を決定するモジュールで，行動は行動メモリに保持される．出力部のモジュールはロボットの行動を出力する．

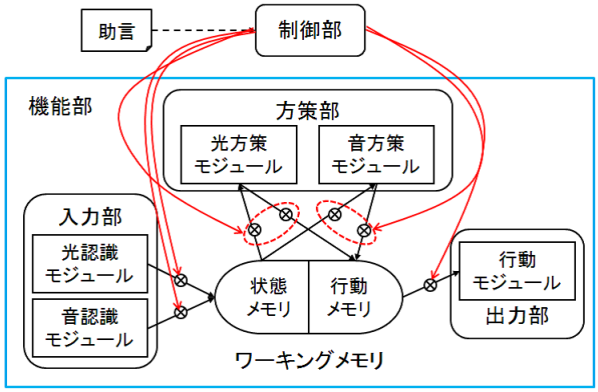


図 1: モジュール組換え型モデル

2.1 制御部

今回のモデルでは，“ 光認識モジュール→状態メモリ→光方策モジュール→行動メモリ→行動モジュール ”または，“ 音認識モジュール→状態メモリ→音方策モジュール→行動メモリ→行動モジュール ”のように，環境に応じて適切な入力部のモジュールを選択し，それに対応した方策部のモジュールにより行動を決定し，行動モジュールに出力することでセンサー入力に対して適切な行動を出力することができる．制御部では，モジュール結合の状態を制御信号によって変えていくことにより，モジュールの組換えを実現する．今回の実験では，入力部，方策部，出力部にある 5 つのモジュールを制御する．

連絡先: 岡 夏樹，nat あっと kit.ac.jp

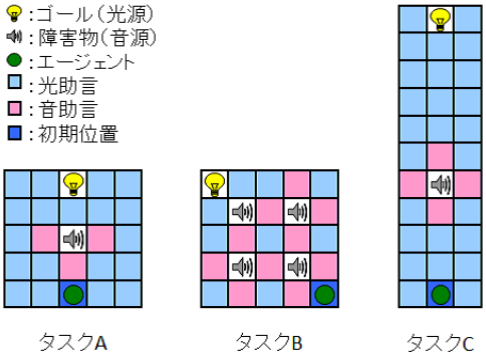


図 2: 実験タスク

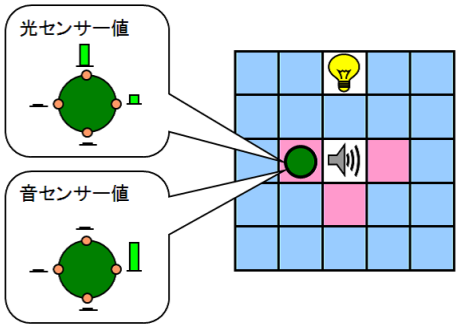


図 3: 入力部が観測するセンサー値

2.2 ワーキングメモリ

状態メモリ： 入力部から受け取った認識された値を保存する．光と音のセンサー情報は別々に保存する．

行動メモリ： 方策部から受け取った行動を保存する．

2.3 入力部

入力部のモジュールは，後述するシュミレーション環境下で現在のエージェントの位置の対象物のセンサー値を認識する．状態メモリとの結合が on になることで，その時点の値を状態メモリに出力する．

光認識モジュール： 現在のエージェントの位置における光センサーの値を認識するモジュール．

音認識モジュール： 現在のエージェントの位置における音センサーの値を認識するモジュール．

2.4 方策部

ワーキングメモリにあるセンサー情報を読み取り，エージェントがとるべき行動を出力する．

光方策モジュール： 状態メモリにある光源のセンサー情報に基づいてとるべき行動を出力する．

音方策モジュール： 状態メモリにある音源のセンサー情報に基づいてとるべき行動を出力する．

2.5 出力部

行動モジュール： 行動メモリにある情報に基づいてエージェントを動かす．

3. 実験方法

3.1 実験タスク

今回のシミュレーションでは，図 2 に示す 3 種類の平面（以下フィールドと呼ぶ）上で，シミュレーション実験を行った．エージェントは，図 2 の各タスクの図に示す初期位置をスタートし，障害物である音源をよけて，ゴールである光源にたどり着くことが目的である．エージェントはフィールド内を上下左右に 1 マスずつ進むことができる．エージェントは，図 2 中の各マス目の色分けで示されているように，光を見て動いて欲しい場面と音を聞いて動いて欲しい場面で異なる助言（それぞれ，光助言，音助言と呼ぶ）を与えられながら移動し，障害物のあるマスか，ゴールのあるマスに侵入するまでを 1 試行とする．ただし，エージェントのフィールド内での移動回数が 30 回を超えるか，エージェント内部でモジュールの組換え回数が 300 回を超えた場合に，その試行は打ち切り次の試行に移る．

報酬は，方策部にはゴールしたときに 10，障害物にぶつかったときに -10，1 回移動するごとに -1 を与える．エージェントがフィールドの外に出る行動を行ったときには，エージェントは移動せずに -2 を与える．制御部には，ゴールしたときに 100，障害物にぶつかったときに -100，一度のモジュールの組換えを行うごとに -1 を与える．移動回数や組換え回数の上限で試行が打ち切られた場合，打ち切られたことによる報酬は与えない．

エージェントが行動した場合，ワーキングメモリに記憶されている全ての情報はクリアされる．行動メモリに情報がある状態で方策部が起動されると，行動メモリの情報は上書きされる．

センサー値は，図 3 のようにエージェントの前後左右に付いている強い指向性を持つ認識センサーで，対象物の情報を獲得する．センサー値は，逆 2 乗の法則を元に計算を行い，0〜1 の値をとり，対象物から離れるとセンサー値が下がるように決めた．対象物が複数存在する場合は，重ね合わせの原理に基づき各対象物からのセンサー値の和を用いた．後述する方策部の強化学習においては，状態空間として，センサー値をタイルコーディングで表現したものを用いた．

3.2 エージェントが行う学習

本実験では，モジュールの機能の学習を方策部で行い，モジュールの組換え系列の学習を制御部で行う．そして，それぞれの学習について，逐次的に学習を行う方法と，同時に学習を行う方法を提案し，各場合についてどのように学習が行われるかを比較する．

制御部が行う学習（組換え系列の学習）

制御部の学習では，Sarsa(λ) を用い，各パラメータは，学習率 $\alpha = 0.1$，割引率 $\gamma = 0.9$，$\lambda = 0.9$，行動選択には $\epsilon = 0.1$ の ϵ-greedy を用いた．制御部は，与えられた助言と現在のモジュール間結合を状態として，次の時点のモジュール間結合（これが行動）の価値を学習する．

方策部が行う学習（モジュール機能の学習）

方策部の学習では，Q 学習を用い，各パラメータは，学習率 $\alpha = 0.1$，割引率 $\gamma = 0.9$，行動選択には $\epsilon = 0.1$ の ϵ-greedy を用いた．方策部は，入力部から得たセンサー値を状態として，各状態におけるエージェントの各行動（前，後，左，右への移動）の価値を学習する．

3.3 同時学習と逐次学習

本実験では，以下の 4 種類の学習方法を比較する．

組換え学習のみ： 方策部は作りこみのものを用い，制御部の学習のみ（すなわち，組換え学習のみ）を行う．

同時学習： 各方策部の学習（モジュール機能の学習），および，制御部の学習（組換え学習）を同時に行う．

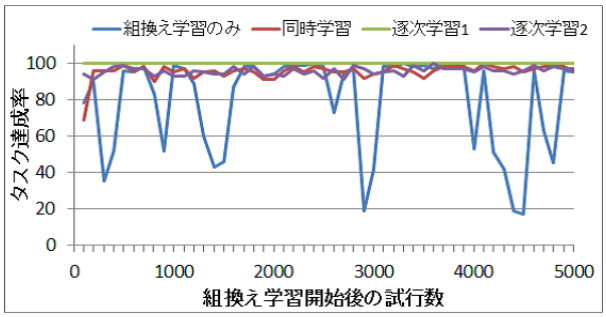


図 4: タスク A でのタスク達成率

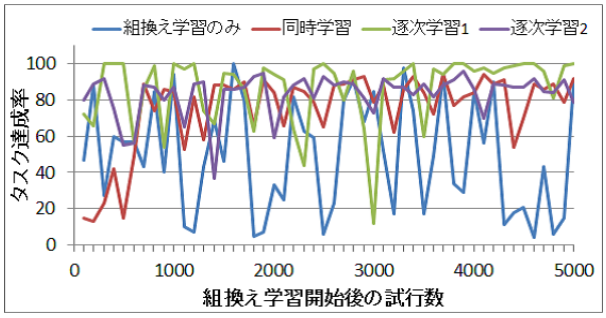


図 5: タスク B でのタスク達成率

逐次学習 1： 制御部は作りこみとして方策部の学習を行った後，学習した各方策部の機能を固定して制御部の学習を行う．

逐次学習 2： 制御部は作りこみとして方策部の学習を行った後，方策部の学習と制御部の学習を同時に行う．

作りこみの方策部は，光方策モジュールはゴールに向かう行動を選択し，音方策モジュールは障害物を避ける行動を選択する．具体的には，光方策モジュールは，センサー値が一番大きい値の方向に，音方策モジュールは，センサー値が一番大きい値の方向以外に向かう行動を選択する．作りこみの制御部は，各方策モジュールそれぞれ個別に学習を行う．具体的には，光方策モジュールの学習を行う場合は，“ 光認識モジュール→光方策モジュール→行動モジュール ”と起動し，光方策モジュールの学習を行う．音認識モジュールの場合も，同様に学習を行う．

4. 結果と考察

4.1 実験結果

図 2 の各タスクについて，100 試行ごとのタスク達成率を各手法で比較したグラフを図 4，図 5，図 6 に示す．各グラフの横軸は，組換え学習開始後の試行数であるので，逐次学習 1,2 においては，グラフのプロット開始前に方策モジュールの学習が行われていることに注意して欲しい．組換え学習開始前の方策モジュールの学習は，2500 試行分行った [坂本 12]．

4.2 考察

4.2.1 組換え学習のみの場合

図 4，図 5，図 6 より，組換え学習のみでは概ね学習できていることがわかるが，途中，達成率が下がることがある．これは，制御部の探索行動によって起こると考えられる．学習がうまくいっているとき，エージェントは光助言で光方策モジュールを用いてゴールに向かうのだが，探索行動によって音方策モジュールで偶然ゴールすると，光助言で音方策モジュールを用

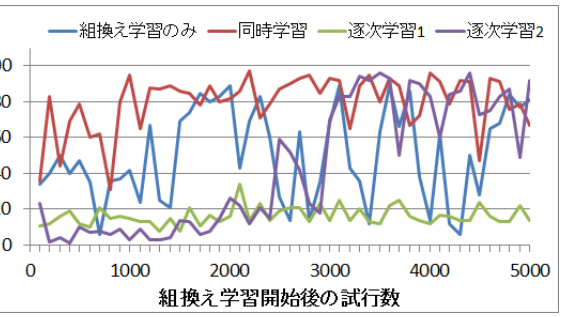
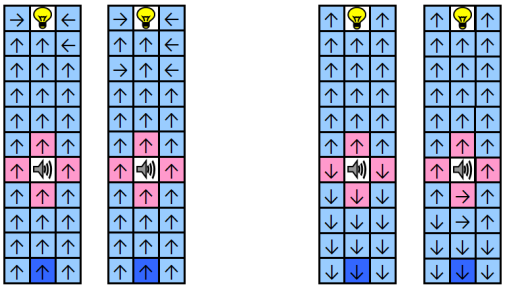


図 6: タスク C でのタスク達成率



(a) 個別　(b) 同時　(c) 個別　(d) 同時

光方策モジュール　音方策モジュール

図 7: タスク C で方策部が獲得した行動（(a)，(c):先に制御部作りこみで個別に学習を行った場合，(b)，(d):制御部と同時に学習を行なった場合）

いるようになる．音方策モジュールはゴールに向かうように作り込まれていないので，ゴールできずに達成率が下がる．そのため，光助言で光方策モジュールを用いるようになるには，時間がかかり，途中，大きく達成率が下がってしまうと考えられる．しかし，モジュールの組換えによる小さな負の報酬が与えられるので，次第に光助言で光方策モジュールを用いることが増え，再び達成率は上がったと考えられる．

4.2.2 同時学習と逐次学習 1,2 の比較

タスク A〜C それぞれについて考察を行う．タスク A では，同時学習，逐次学習 1,2 いずれにおいても，光方策，音方策の両モジュールとも，障害物をよけゴールに向かう行動を学習した．これは，行動を出力した方策モジュールには，ゴールと障害物のどちらの場合も報酬を与えたため，片方のセンサー情報だけでフィールド上の自分の位置が特定できる場合は，そのセンサー値だけでも十分に行動できたと考えられる．また，達成率は組換え学習のように途中で下がることはなかった．これは，制御部が探索行動を行っても，どちらの方策部でも障害物をよけゴールに向かうことができたからだと考えられる．

タスク B では，音方策モジュールは障害物が複数存在し，センサー値が複雑になるためうまく学習できなかった．しかし，光方策モジュールは障害物をよけゴールに向かう行動を学習できたので，途中，達成率が下がることがあるが，3 つの学習方法のいずれにおいても学習できていることが分かる．

タスク C では，同時学習はうまく学習でき，逐次学習 1 はうまくいかず，逐次学習 2 は試行数を重ねることで次第に達成率が上がっている．ここで，同時学習と逐次学習 1,2 での方策部の学習に注目すると以下のような違いがあることがわかる．

同時学習： 制御部と同時に学習を行なっている間に方策モジュールの学習を行う

逐次学習 1： 先に作りこみの制御部でそれぞれ個別に方策モジュールの学習を行う

逐次学習 2： 最初は逐次学習 1 と同様に学習を行い，その後制御部と同時に学習を行う中で方策モジュールの学習を行う

これより制御部の学習で用いている方策部は，逐次学習 1 や逐次学習 2 の前半のように (i) 先に制御部作りこみで個別に学習を行った方策モジュールを用いる場合と，同時学習や逐次学習 2 の後半のように (ii) 制御部と同時に学習を行なった方策モジュールを用いる場合に分けることができる．(i) と (ii) の場合で獲得した行動を図 7 に示す．図 7 より，エージェントは 1 つの方策モジュールで障害物を避けながらゴールに向かう行動は学習していないことがわかる．これは，タイルコーディングを用いているからだと考えられる．タイルコーディングでは連続値から有限個の離散値にコード化するときに，値の差がとても小さい場合，同じ値にコード化されることがある．タスク C の光センサー値は，ゴール周辺の各マスではゴールに近いため各マスのセンサー値が高い値だが，ゴールから遠いマスについては，センサー値がとても低い値となる．そのため，ゴールから遠いマスのセンサー値は同じ値にコード化されてしまい，同じ行動をとったと考えられる．以上から，ゴールから遠く障害物に近いマスでは，障害物の位置が分からないので，障害物をよける行動がとれなかったと考えられる．音方策モジュールも同様に考えられる．そのため，エージェントは 2 つの方策モジュールをうまく使い分けてゴールしなければならない．(i) の方策部では，図 7(a)，7(c) より，2 つの方策モジュールを助言によってうまく使い分けてもゴールできないので，達成率が上がらなかったと考えられる．一方，(ii) の方策部では，図 7(b)，7(d) より，音方策モジュールが障害物周辺で障害物を避けながらも，ゴール方向へ向かう行動を学習している．これにより，エージェントはこの 2 つの方策モジュールを助言によってうまく用いることで，ゴールすることができる．

ではなぜ，同時に学習を行うと音方策が音助言の場面で，障害物を避けながらゴールに向かう行動が学習できるのかというと，(i) の場合，音方策モジュールはタスク A では分かっていたゴールの位置を距離が遠いため認識できないので，ゴールする行動が学習できない．そのため障害物周辺で，ゴール方向に向かっても正の報酬が得られない．それどころか，障害物に近づくことで大きな負の報酬を得るかもしれない．そのため，ゴール方向に向かい障害物に近づく行動よりも，障害物から離れる行動をとろうとする．よって，(i) の場合，音方策モジュールは障害物を避けながらゴールに向かう行動は学習できない．一方，(ii) の場合，エージェントは，光教示の場面で光方策を用いてゴールすることが可能で，音方策モジュールは音助言の場面でゴールに向かう行動をとれば良い．エージェントがゴール可能となった後，制御部の探索行動によって音方策モジュールでゴールすることでゴール周辺の行動価値が高くなり，音方策モジュールがゴールに向かう行動をとることができる．その結果，同時に学習を行った場合，音助言の場面で音方策は障害物をよけながらゴールに向かうという行動を獲得できる．

同時学習と逐次学習 1,2 を比較すると，タスク A やタスク B ではあまり差が出なかったが，タスク C では違いが出た．同時に学習を行うことで，タスク C のような一つの方策モジュールがうまく行動できない場合も，制御部が 2 つの方策モジュールを助言に応じてうまく使い分け学習を行うことでタスク達成できるようになった．以上より，今回実験を行ったタスクでは同時学習が一番良い結果となった．

5. おわりに

本研究では，モジュールの組み合わせ方の自由度が高い，モジュール組換え型モデルについて，モジュールの機能の学習とモジュールの組換え系列の学習を同時に行う手法を提案した．そして，学習を同時に行うことで，モジュールの機能はタスクに応じてカスタマイズされ，モジュールの組換え系列の学習だけを行った場合より，エージェントがより安定した行動を学習できることを示した．また，モジュールの機能の学習，および，モジュールの組換え系列の学習を逐次的に行う方法と，同時に行う方法を比較し，単独モジュールではうまくいかない場合にも良好な学習結果を得られるのは，同時に学習を行った場合であることを，実験的に示した．

今後の課題として，今回の実験で得られた同時学習が優れているという結果が，どの程度一般的であるかを調べることが挙げられる．本実験では作りこみの方策モジュールをそれぞれ１つしか用意しなかったが，異なる作りこみ方策を用いて実験を行い調べる必要がある．また，学習中に，ゴール位置をランダムに設定したり，タスク自体の変更を行うとどうなるかについて実験を行いたい．さらに，性質の異なるさまざまなタスクについての実験も行いたい．

参考文献

[小川 04] 小川昭利, 大森隆司: 機能部品組合せモデルによるナビゲーション行動学習処理の獲得方式の提案, 信学論 (D), vol.J87-D-II, no.4, pp.987-998, (2004).

[Wolpert 98] D. M. Wolpert and M. Kawato: Multiple paired forward and inverse models for motor control, Neural Network, vol.11, pp.1317-1329, (1998).

[Jacobs 91] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton: Adaptive mixture of local experts, Neural Coputation, vol.3, pp.79-87, (1991).

[高橋 09] 高橋泰岳, 枝澤一寛, 野間健太郎, 浅田稔 :モジュール型学習機構を用いたマルチエージェント環境における競合行動獲得, 日本ロボット学会誌, vol.27, no.3, pp.350-358, (2009).

[Oka 99] N. Oka: Apparent “free will” caused by representation of module control, No matter,Never mind: Proceedings of Toward a Science of Consciosness: Fundamental Approaches, pp.243-249, (1999).

[本多 09] 本多透, 板舛尚樹, 岡夏樹: ロボットの内部情報処理に対する言語教示可能性, 第 23 回人工知能学会全国大会論文集, (2009).

[坂本 10] 坂本裕太, 本多透, 尾関基行, 岡夏樹: モジュール組換え型アーキテクチャを用いた行動学習法の検討, 情報処理学会関西支部大会講演論文集 2010, (2010).

[本多 11] 本多透, 坂本裕太, 尾関基行, 岡夏樹: モジュール組換型アーキテクチャにおける再利用可能な内部情報処理の学習, 第 38 回知能システムシンポジウム資料, pp.111-116, (2011).

[坂本 12] 坂本裕太: 自由度の高いモジュール組み換えとモジュール機能の同時学習, 京都工芸繊維大学修士論文, (2012).