計測自動制御学会　東北支部第２１２回研究集会（2003.11.22）
資料番号　212-12

アメーバ型ロボットにおける運動パターンの自己形成

Self-formation of the movement pattern for an amoeba type robot

○中野　彰、大友照彦、大槻恭士、石谷幹夫

○Akira Nakano, Teruhiko Ohtomo, Takashi Otsuki, Mikio Ishitani

山形大学工学部

Faculty of Engineering, Yamagata University

キーワード：強化学習（Reinforcement Learning）, Q-Learning, Profit Sharing, アメーバ型ロボット（Amoeba type Robot）

連絡先：〒992-8510　山形県米沢市城南　4-3-16　山形大学工学部　共通講座　大友研究室

1　はじめに

地球上に生息する全ての生物は、全てその場所・環境に適した形態をもっている。その環境に適応できなかった個体は当然のごとく淘汰され、適応できるように進化したもののみが子孫を残せるという自然界では非常にあたりまえの競争原理に基づいた結果である。

このことから一般に言われているような「退化」という現象はありえない。なぜなら鶏やペンギンが空を飛べないのも、ブラインドケープテトラに目が無いのも、全てその環境に適応した「進化」をした結果であるからだ。もし充分な餌が周りにあり、外敵が存在しないなら大きな翼は走る上では単なる邪魔な存在でしかない。つまり大きな翼を失ったのではなく、小さな翼を手に入れて自分の生存範囲の移動をより快適に行えるように進化したといえる。

同じように車輪から静歩行、さらに動歩行へと進化してきたロボットの移動法にも逆行的な進化が見られる、つまり海月のような群体を思わせる合体・変形を繰り返して移動するユニットの集合体や、アメーバやヒトデといった「柔らかい動き」をするロボットの研究・開発である。

これまでロボットは主に「人間の手助け」を目的に開発されてきたが、最近では月や火星などの地球外環境や、危険地域での活動など人間が入り込めない環境での活動を目的をとした研究も進められている。

そこで注目されるのが前述のようなロボットである。車輪ではよほど大きいかキャタピラでなければ悪路での活動は難しい。また静歩行では重心をキープしつつ移動しているので安定しているが動作が遅い。動歩行は逆に自ら重心を前方にはずしながら（倒れな

がら）移動しているので動作自体はスムーズだが、安定しておらずやはり悪路には対応しきれない。だが群ロボットやアメーバ運動をするロボットならば、大抵の悪路にも対応できるし狭い場所に入っての活動が可能であるなどの利点がある。

本研究では、アメーバ型ロボットのモデルを剛体シミュレータ（Working Model 3D）上に構築し、強化学習を用いた移動法や形状変化の獲得を目標とする。

2　原理

2.1 強化学習（Reinforcement Learning）

強化学習[1]とは，試行錯誤を通じて環境に適応する学習アルゴリズムの一種である。教師付き学習(Supervised learning)とは異なり、状態入力に対して正しい行動出力と照らし合わせるべき教師が存在しない。かわりに報酬という情報を手がかりに学習するが、報酬にはノイズや遅れがあるため、行動を実行した直後の報酬をみるだけでは、学習の主体はその行動が正しかったかどうかを判断できないという困難を伴っている。

しかし強化学習が有用なのは、不確実性のある環境を扱っている点にある。多くの実世界の制御問題では、不確実性の扱いは厄介だが、強化学習では報酬に遅れが存在し、離散的な状態遷移も含んだ段取り的な制御規則の獲得を行うことが可能となる。設計者がゴール状態で報酬を与えるという形で覚えさせたいタスクをエージェントに指示しておけば、ゴールへの到達方法はエージェントの試行錯誤学習によって自動的に獲得される。つまり設計者が「何をすべきか」をエージェントに報酬という形で指示しておけば「どのように実現するか」をエージェントが学習によって自動的に獲得する枠組となっている。

2.2　Q-PSP Learning

強化学習には様々なアプローチが存在し、中でも環境同定型の代表的な手法であるQ-Learningと、経験強化型の強化学習法であるProfit Sharing法（PSP）を組み合わせたものがQ-PSP Learning[2]である。

Q-Learningでは各状態でとり得る行動全てがQ値とよばれる値を持っている。エージェントはこのQ値が最も高い行動を1－εの確率で選択し実行する。またこの値は報酬によって増減する。Q-Learningはマルコフ決定問題においてはある条件の下で最適性が保障されているが、複雑な問題に対しては学習に多数の試行錯誤による行動が必要であり学習速度が遅い。

1．エージェントは環境の状態 s(t)を観測する
2．エージェントは任意の行動選択法によって行動 a(t)を決定する
3．環境から報酬 r(t)を獲得する
4．状態遷移後の状態 s(t+1)を観測する
5．以下の更新式によりQ値を更新する

Q-learningでの学習（現在）

$$Q(s(t),a(t)) \leftarrow (1-\alpha)Q(s(t),a(t)) + \alpha[r(t)+\gamma \max_{a} Q(s(t+1),a(t))]$$

※　α：学習率、γ：割引率

PSPでの学習（過去）

$$Q(past\_s(i),past\_a(i)) \leftarrow (1-\alpha')Q(past\_s(i),past\_a(i)) + \alpha'(\gamma'^{i} \times r(t))$$

※　α'：学習率、γ'：割引率、
i：過去 i Frame 前の状態と行動であることを表す

6．時間ステップ t を t+1 に進めて1へ戻る

図1．Q-PSP Learning アルゴリズム

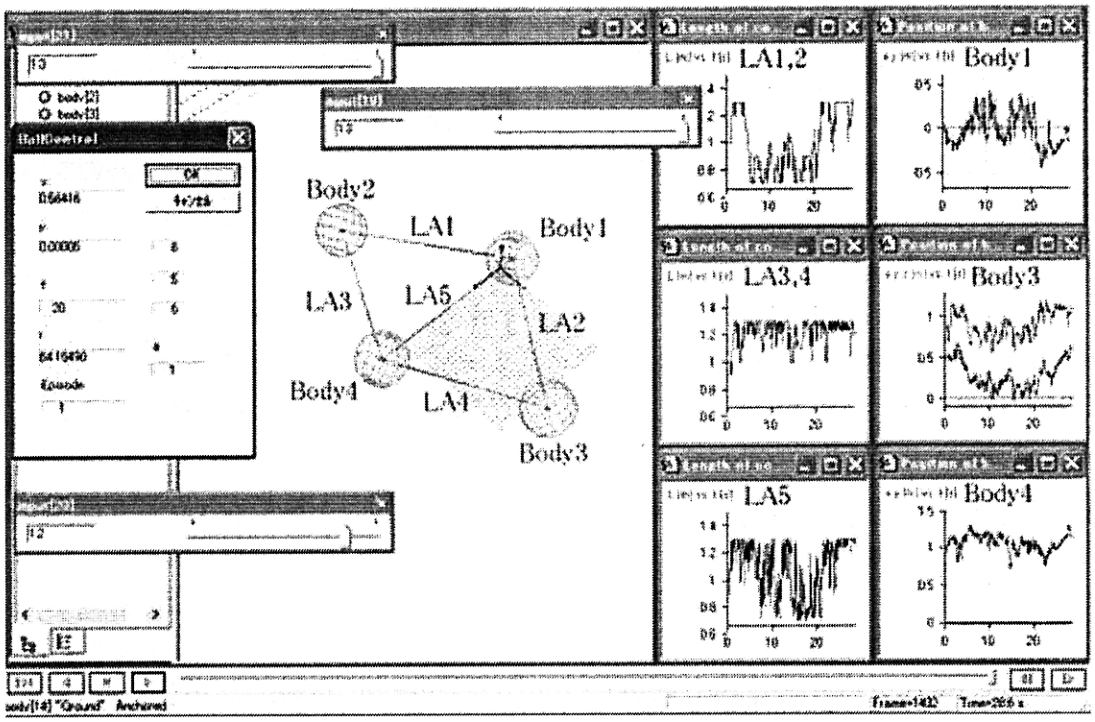

図２．モデルを用いた学習の様子

それに対し PSP では過去の状態とその時の行動（ルール）を past_s と past_a に記憶し、報酬が得られると過去に遡りながら、同時に報酬を減らしながら Q 値を更新していく。本研究では PSP を用いることにしたが、これは「運動パターン」という一連の動作の集合に対し、報酬が一定の過去のルール全てに影響を与える PSP の手法が有効であると考えられるためである。

フローチャートは図１のようになる。

2.3 逆行型 Q-Learning

Q-PSP Learning と同様に過去の状態と行動を記憶しておき、それをもとに報酬を得た時点から、過去に遡って Q 値を更新していく学習法であり、ここではこれを逆行型 Q-Learning と呼ぶことにする。

Q-PSP Learning とは違い、過去の Q 値も以下のように Q-Learning の更新式を用いて学習する。

$$Q(past\_s(i),past\_a(i)) \leftarrow (1-\alpha)Q(past\_s(i),past\_a(i)) + \alpha[\gamma^{'i}\times r(t) + \gamma \max_{a} Q(past\_s(i+1),past\_a(i))]$$

3 実験

3.1 モデルの構築

実験で使用するモデルは Working Model 3D という、剛体シミュレータの一種を用いて構築する。

Working Model 3D は、Windows の‘OLE Automation’技術を用いて、外部プログラムから、より複雑な運動指令を与えてコントロールすることが可能である。

本研究では、Visual C++により OLE Automation を使った外部プログラムを作成し、Working Model 3D をコントロールしている。

Working Model 3D では Body（剛体）、Constraint（拘束部品）、Coord（目印）を用いてモデルを作成する。

・Body

モデルの主要な構造部品。

質量や摩擦係数などの特性を自由に設定できる他、座標や速度などの情報を持つ。

・Constraint

Body を拘束する部品。

Actuator や Motor、Joint などの種類がある。

・Coord

Body につける目印で、Body どうしや Constraint との接合に使う。

これらを用いてのモデルの構築は、図 2 に示すように、動作特性や Working Model 3D の表現能力から網目構造のアメーバ型ロボット[3]とした。これは、４つの球体の Body と５本の Linear Actuator を用いており、基本形は正三角形を二つ組み合わせた形状をしている。

Body に球体を用いたのは、接地面積を小さくし Working Model 3D の計算時間を短くするためと、Linear Actuator の作用点を重心に統合して設置するための設定が容易であるためである。

また、Working Model 3D 内での時間は 1 Frame あたり 0.02 秒である。

3.2 強化学習プラグラムによる制御

上記のモデルを外部プログラムで制御する際、Working Model 3D から得られる情報はモデルの位置（４つの Body の x,y 座標から求めた重心）であり、プログラムから与えるものは Linear Actuator の長さ（Input）である。

図 2 では Linear Actuator が５本あるが全てを別の Input で動かすことは現時点では難しいので、LA１と LA２、LA３と LA４をそれぞれ Input１と２で制御することにする。こうすることでモデルは理論上 x 軸方向にのみ進むようになり、ゴール設定が容易になる。

Working Model 3D のモデルはプログラムでのエージェントに対応しており、３種類の Linear Actuator がとり得る長さの組み合わせが、状態遷移マップになっている。エージェントがマップを移動するときの行動は Linear Actuator の長さを変えることに対応している。

エージェントはこのマップを探索し、より効率的な行動を学習していく。

具体的な流れは以下のようになる。

1] プログラムを実行させると同時に Working Model 3D を起動させシミュレーションに使うファイルを開く

2] Working Model 3D からモデルの位置を取得する

3] 取得した位置情報を過去のものと比較し、それによる報酬を与え Q-Learning と PSP で Q 値を更新する

4] 同時に行動を決定し、状態を遷移させモデルに対して入力を行う

5] モデルがゴールに到達するまで 2~4 を繰り返す

6] 規定の回数ゴールするまでモデルを初期位置に戻して 2~5 を繰り返す

実際の実験を行う時のモデルとエージェントの主なパラメータを表１および２に示す。ただし、α、α' はそれぞれ Q-Learning と PSP の学習率、ε は ε-greedy によるランダム行動選択確率、γ、γ' については報酬の割引率である。また Episode はスタートからゴールに到達するまでの一連の学習過程を行う回数である。

なお ε は学習後期において獲得した運動パターンを実行する際の妨げになるので、Episode が進むにつれ減少させる。

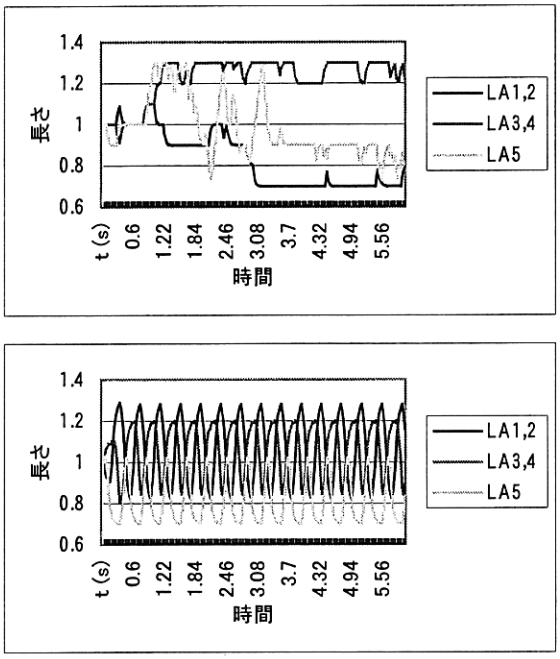

図３．学習初期（上）と後期（下）
のLAの長さの変化

4　シミュレーション結果

表3、４ではQ-PSP Learning及び逆行型Q-Learningと Q-Learningの学習速度の違いを表している。これは最も学習に時間のかかる1Episode目を終えるのに要した時間の比較であり、Q-PSP Learning、逆行型Q-Learningともに Q-Learningのみの学習に比べて学習時間は短縮されている。

また図 3 に学習初期と後期の Linear Actuatorの長さの変化の様子を示す。これより、学習初期では試行錯誤を繰り返し、状態遷移マップ内を探索しているが、学習が進むにつれ、獲得した運動パターンを繰り返していることがわかる。

学習結果は大部分が2Frame 周期で状態を変化させ、細かく振動しながら進むパターンとなったが、中には図４に示すようにある程度形を変化させながら進むパターンを獲得したものもあった。

表１．モデルのパラメータ

Bodyの半径[m]	0.2
Bodyの重量[kg]	33.5
Linear Actuatorの初期長と最小・最大[m]	1.0 [0.7,1.3]
Linear Actuatorの入力による変化[m]	-0.1、0.0、+0.1
Inputの数	3
スタート座標[m]	x＝0.5、y＝0.0
ゴール領域[m]	x≧1.5

表２．エージェントのパラメータ

行動数	3×3×3
状態数	7×7×7
α	0.2
α'	0.4
ε	0.1
γ	1.0
γ'	0.9
Episode	10

表3．PSP有・無での学習の違い

学習形態	Q-Learningのみ	Q-PSP Learning	Q-PSP Learning
記憶ルール数	－	10	30
Frame	14404	7753	8617
時間(秒)	288.1	155.1	172.3

表４．通常と逆行型Q-Learningの学習の違い

学習形態	Q-Learningのみ	逆行型Q-Learning	逆行型Q-Learning
記憶ルール数	－	10	30
Frame	14404	5891	6019
時間(秒)	288.1	117.8	120.4

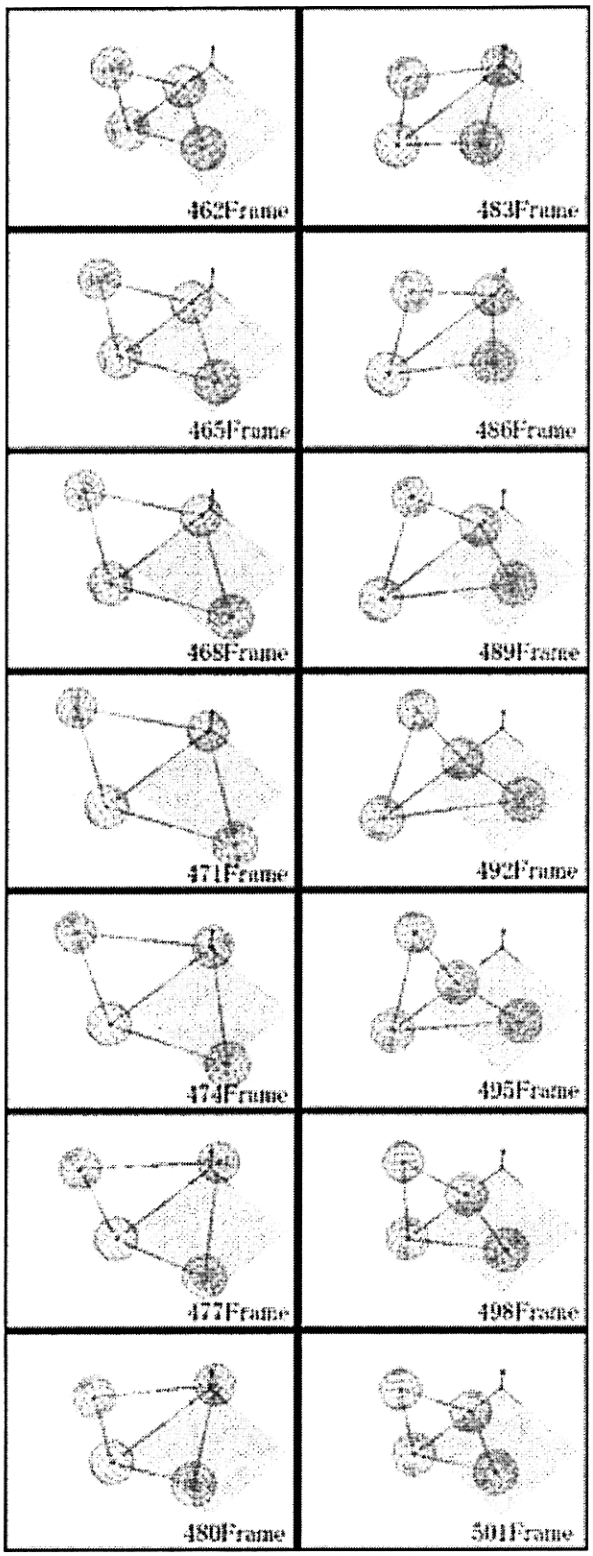

図 4. 獲得した運動パターンの例

5 考察

運動パターンはその周期が増えるに従い、パターン数は飛躍的に増加するが、実際に有効なパターンは限られてくる。そこでエージェントは周期を短くすることで、最適ではないにしろ前進しやすい振動パターンでの解を得ているものと考えられる。これはつまり、今までの学習法では学習時間は短くできても、局所最適を回避しにくいということを表している。

6 まとめ

剛体シミュレータ上にアメーバ型ロボットを構築し強化学習による運動パターンの自己形成について検討した。その結果、2 Frame 周期で状態を変化させて振動するパターンが得られた。

今後は、局所最適に陥らないようにするため、効果的に行動列として評価できるようにするなど、学習法や報酬の与え方などを改善し、形状変化のための行動パターンの獲得なども検討するつもりである。

7 参考文献

(1) 木村 元，宮崎和光，小林重信：強化学習とは？(What is Reinforcement Learning?)、http://www.fe.dis.titech.ac.jp/~gen/edu/RL_intro.html#Section1

(2) 堀内 匡、藤野昭典、片井 修、椹木哲夫；経験強化を考慮した Q-Learning の提案とその応用、計測自動制御学会論文集、Vol.35,No.65,645/653（1999）

(3) 長井 隆、羽倉 淳、横井浩史、嘉数侑昇：アメーバ様ロボットの構築に関する研究、日本機械学会［No.９８－４］ロボティクス・メカトロニクス講演会‘９８講演論文集