

新しい環境下におけるロボットの行動学習に関する取り組み

An Approach to Learning a Robot Controller under an Unknown Environment

曾我 紗知子*1 小林 一郎*2

Sachiko Soga

Ichiro Kobayashi

*1 お茶の水女子大学大学院人間文化創成科学研究科理学専攻

Advanced Sciences, Graduate School of Humanities and Sciences, Ochanomizu University

Even though a robot encounters an unknown working environment, it is required to achieve its tasks properly in the environment. Recently, evolutionary robotics using evolutionary computing in the process of learning a robot controller has been a main approach to achieve the requirement. In this study, we observe a robot's behaviors on a simulator at the three basic environments which the robot has not encountered before, and then discuss the learning of a robot controller using evolutionary computing.

1. はじめに

実世界で作業を行うロボットは、新しい環境に直面した時でも自律的に合理的な行動をすることが求められる。近年、そのようなロボットの行動を制御するコントローラを学習するのに、進化計算を利用する進化ロボティクスが盛んに研究されている。進化ロボティクスとは、遺伝的アルゴリズム、遺伝的プログラミングなどの進化計算や強化学習、ニューラルネットワークなどを用いて環境に対して柔軟な行動を選択できる、ロボットコントローラを設計する技術である。

本研究では、ニューラルネットワークと遺伝的アルゴリズムを用いシミュレータ上の様々な環境に応じて適切な行動を行うようになる、ロボットの行動規則の学習について考察を行う。

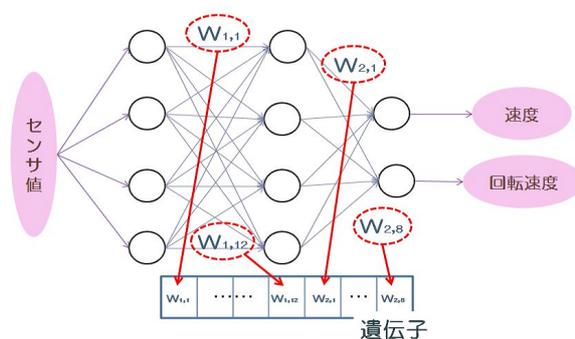


図 1: ロボットコントローラ

2. ロボットコントローラ

2.1 学習環境

本研究では、Web 上でオープンソースとなっている、進化ロボティクスの研究・教育用 3D シミュレータ Simbad[1] を利用して、ロボットコントローラの学習を行う。Simbad は、ロボットの他、壁や箱などの障害物を設置することができ、様々な環境の下でロボットコントローラの学習を行うことができる。ロボットには、視覚センサ（カラー単眼カメラ）、ソナー、光センサ、衝突検出用のバンパーが取り付け可能となっており、制御対象としては、左右の車輪の速度、回転速度、回転角度である。今回のシミュレーションで利用するのは、センサ（ソナー）と速度、回転速度である。

2.2 構成

本研究では、図 1 に示すニューラルネットワークで構成されているコントローラを遺伝的アルゴリズムによって進化させる。コントローラは、単一方向にのみ信号が送られるフィードフォワード型の入力層 4、中間層 4、出力層 2 のニューラルネットワークで構成する。ロボットが取得するセンサ値を入力とし、速度と回転速度を出力とする。遺伝的アルゴリズムを利用す

る際に必要となる遺伝子には、入力層、中間層、出力層全てのノード間の 24 個の結合荷重の値を遺伝子座に入れる。

2.3 学習過程

図 2 にロボットコントローラの学習の概要を示す。

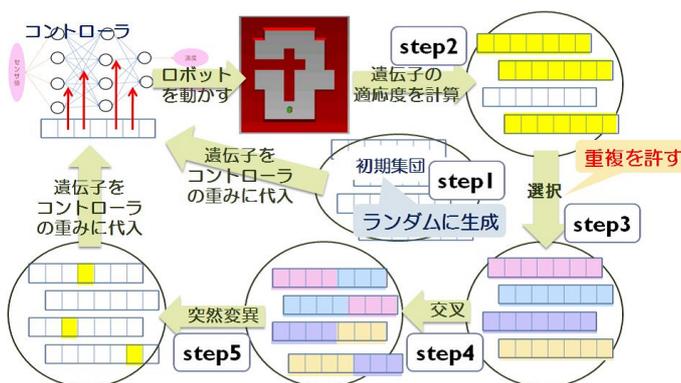


図 2: ロボットコントローラの学習

以下図 2 の概要に沿って、本研究におけるコントローラの学習プロセスとパラメータの設定を示す。

step 1. 初期集団の生成

$[-1.0, 1.0]$ の実数値を結合荷重の数 24 個をランダムに発生させ、1 つの遺伝子の中に代入する。この個体を 50 個

連絡先: 曾我紗知子, お茶の水女子大学大学院人間文化創成科学研究科理学専攻情報科学コース小林研究室, 〒112-8610 東京都文京区大塚 2-1-1, Tel.03-5978-5708, soga.sachiko@is.ocha.ac.jp

生成する。

step 2. 適応度の計算

コントローラのノード間の結合荷重に1つの個体の値を代入し、シミュレータ上でロボットを動作させ、設定した適応度関数に従って適応度を計算する。これを50個全ての個体で行う。

step 3. 選択

ルーレット選択を行い、遺伝子集団から重複を許して次世代の個体候補を選び、新しい個体集団を生成する。新しい集団も元の個体集団の個数を保持する。また、ここでは最も適応度の高い1個体に対してエリート保存選択をしている。

step 4. 交叉

交叉率を0.6とし、一点交叉を行う。

step 5. 突然変異

突然変異率0.1で1つずつ遺伝子座の値を突然変異させるかを決め、突然変異させる際には、その遺伝子座の値を $[-1.0, 1.0]$ のランダムな値に変更する。これを全ての遺伝子に対して行う。

step 6. step 2.へ戻る

step2~step5を設定された世代数になるまで繰り返す。設定された世代数になったら終了する。

3. コントローラ学習実験

本研究では、単純な環境の中で壁に衝突しないで動作することを学習し、より複雑な環境でもロボットが同じように動作できるようにすることを目的とする。

3.1 学習仕様

適応度関数を以下の様に設定する。

$$f = \sum \{t + (\pi - |r|) + s + k\} + 10 \times d \quad (1)$$

f : 適応度

t : 速度

$|r|$: 回転速度の絶対値

s : センサ値の平均

k : センシング1ステップ間の移動距離

d : 総移動距離

適応度関数は、ロボットが大きく回転せず、高速に前進し、次のセンサ情報が入力されるまでの移動距離と、総移動距離が長くなる場合に、適応度が高い値をとるように設定される。コントローラは、センシングの1ステップ毎に入力値を代入し、出力値を計算している。それによって、適応度関数も1ステップ毎に累積している。設定したステップ数が終了するか、ロボットの速度が0、または1ステップ間の移動距離が0になった時点で、適応度関数の累積計算を終了し、最後に総移動距離を足し合わせている。長い距離を移動した場合に適応度が高くなるよう、経験的に移動距離に10の値を掛けている。また、壁に衝突した際には適応度を0にし、終了条件が満たされるまで計算を続ける。終了条件の1つであるステップ数は2500と設定した。

3.2 ケース1: 環境変更

最初に単純な環境(図3参照)での学習を行った。適応度関数は式(1)に示したものを採用した。この環境では、100世代まで学習を行うと、ロボットは壁に衝突せずに巡回するようになったが、未知の環境に適応するには進化せず、図3の環境のみに適したコントローラが獲得された。この実験結果より、未知の複雑な環境においても、動作を可能にするには、学習環境を工夫しなければならないことがわかった。

次に、図3よりも少し複雑な図4の環境を構築した。窪みや突起を作り、ロボットが突き当りに行き着くと、引き返さなければ、環境を一周できないようになっている。こちらの学習で利用した適応度関数も、3.1節のものである。この環境では、コントローラを学習させる世代数を100世代と200世代に設定し、実験を行った。

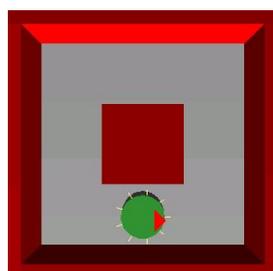


図3: 最初の学習環境

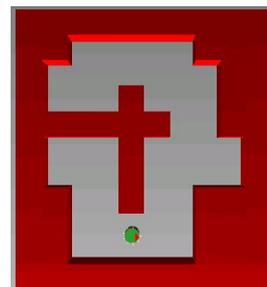


図4: 学習環境

100世代進化させたコントローラは、センサからの情報を利用して壁を回避するのではなく、学習した環境(図4参照)のスタート地点から壁に衝突しない半径で回転をするだけの学習をした。ここで得られたコントローラでは、複雑な環境(図5参照)で動作させても、壁に衝突はしなかったが、図4の環境での動作と同じように壁に衝突しない半径での回転をする動きを見せた。このことから、100世代では、まだ十分な学習がなされていないことがわかる。

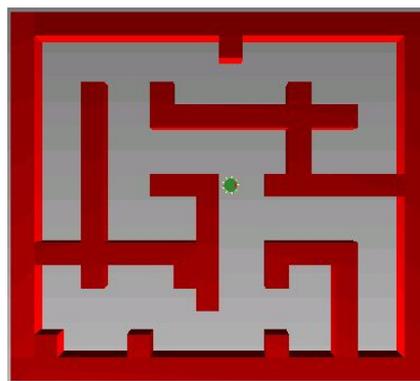


図5: 複雑な環境

同様に、図4の環境で200世代まで学習を続けると、図4の環境の中で壁に衝突せずに動作し続けていた。また、図5に示す環境に移しても、壁に衝突せずに動作することが観察された。

次に、衝突回避を学習したコントローラを図5よりも複雑な環境で動作させたが、壁に衝突し、正しい動作はしなかった。このことから、図3の環境での実験結果と合わせて考えると、

未知の環境において正しく動作するためには、その環境を想定した学習環境の設定が重要であることがわかった。

3.3 ケース 2：通過点の設定

また、ロボットが通過する地点を設定したい場合、その地点を通過させることが可能であるかを判定する実験を行った。

適応度関数は式 (1) で設定したものと同じものに対し、図 6 の環境で白丸の地点を通過した場合に、適応度を 1000 ずつ上げることとした。

$$f = \sum \{t + (\pi - |r|) + s + k\} + \sum 1000 + 10 \times d \quad (2)$$

ロボットがスタートしてから、左上、右下、左上というように 2 つの白丸の地点を交互に通過するごとに適応度に 1000 を足す。ここでは 500 世代まで学習を行うと、その地点を通過するように学習した。図 7 にその軌跡を示す。

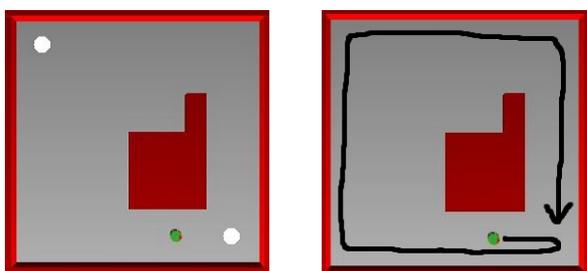


図 6: 通過点を設定した学習環境 1 図 7: 通過点を設定した学習後の動作軌跡 1

同じ環境設定で、白丸の位置を変更した場合 (図 8 参照) の実験を行った。41 世代まで学習を行い、設定した地点を通過するようになった。その軌跡を図 9 に示す。図 8 の環境では、先に学習した壁に衝突しないで移動する行動の軌跡に近い場所に白丸を設定したため、学習が早く終わったと考えられる。

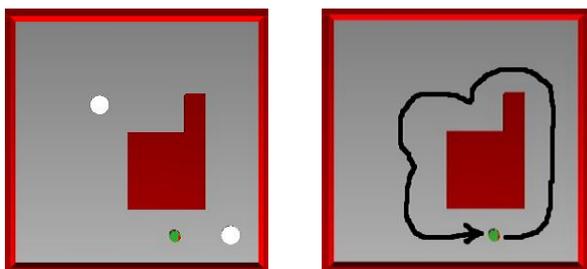


図 8: 通過点を設定した学習環境 2 図 9: 通過点を設定した学習後の動作軌跡 2

3.4 ケース 3：複数台のロボット

さらに、複数台のロボットを対象に、壁に衝突しないことに加え、お互いに衝突せずに動作する学習の実験を行った。実験環境を図 10 に示す。この環境は壁と壁との距離を、ロボット 2 台並べた直径の約 1.7 倍と、ロボット 2 台がすれ違うのに十分な長さにとっている。適応度関数は 3.1 節で示した同じものを使用している。ロボット同士の衝突も検知するように設定し、衝突した際は適応度を 0 にする。

実験においては、実験環境に存在する 4 台のロボットのうち、3 台のロボットでは、ニューラルネットワークの結合荷重を $[-1.0, 1.0]$ のランダムな値に設定したコントローラで動作

させ、残りの 1 台のロボットコントローラを壁にもロボットにも衝突しない様に学習させる。学習をしない 3 台のロボットは、壁に衝突してもランダムに方向転換をし、学習しているロボットがリセットされるまで動き続けるように設定した。

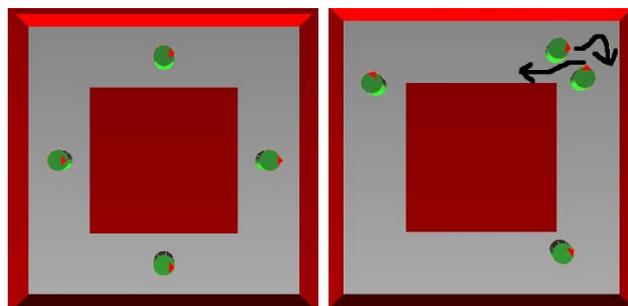


図 10: 複数台ロボットの環境 図 11: 衝突回避

100 世代までの学習では、ケース 1 と同様に、学習した環境 (図 10 参照) のスタート地点から壁に衝突しない半径で回転する学習をしていた。

200 世代まで学習させ、その学習したコントローラ 1 つを 4 台全てのロボットに組み込んで動作確認をしたところ、ロボットは壁とロボット同士の両方に衝突しないで動作した。ロボット同士が衝突しそうになると、速度を緩め、お互いに横に避けて進む動き (図 11 参照) を見せた。

200 世代以上の学習を行うと、100 世代までの学習と同じ様に、ロボットは回転を続ける動作を引き起こしてしまった。これは、コントローラが過学習をしてしまったものと思われる。また、100 世代の時より速度が速くなっており、2500 ステップの中で適応度をより高くしようと、総移動距離を長くするために、その場で回転を続けることと、速度を速くすることを学習したのだと考えられる。

4. まとめ

本研究では、単純な環境でロボットコントローラを進化計算により学習させて、その環境に適したコントローラを得た。その後、未遭遇のより複雑な環境においても単純な環境で学習した目的を遂行できることを確認した。また、通過点を設定して学習させる実験から、ロボットに望む動作を学習させることが可能であることを確かめられた。そして、ロボットを複数台にし、ロボットがお互いに衝突しない、という動作を学習したことから、静止している物体だけではなく、動き続けている物体を避ける学習ができることを認められる。しかし、得られたコントローラは学習した環境に依存したものになってしまうため、学習に 2 つ以上の目的を持たせたい場合は、適応度関数を 1 つにまとめることが難しい。今後は、適応度関数を複数にし、多目的遺伝的アルゴリズムを適用して、より複雑なタスクを遂行できるコントローラへと進化させることを考えている。

参考文献

- [1] <http://simbad.sourceforge.net/>
- [2] Louis Hugues, Nicolas Bredeche, Simbad : an Autonomous Robot Simulation Package for Education and Research, Simulation of Adaptive Behavior, Rome : Italy(2006)
- [3] S.Nifi and D. Floreano, Evolutionary Robotics, MIT Press, 2000.