

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

曾我 紗知子 (指導教員：小林 一郎)

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。

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。

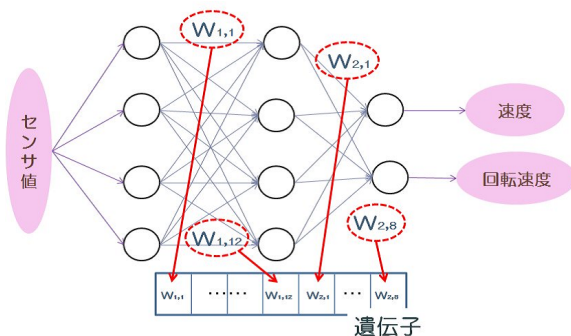


図 1: ロボットコントローラ

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。

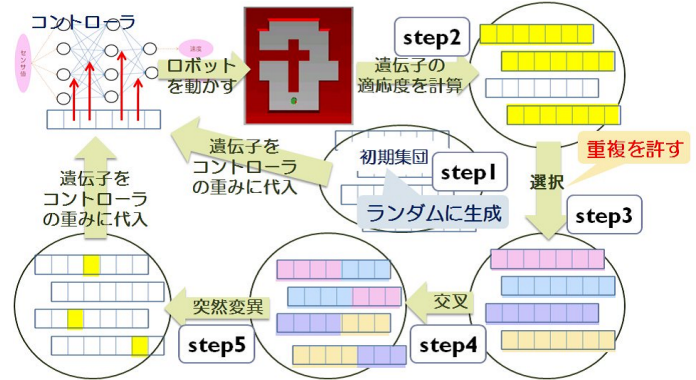


図 2: ロボットコントローラの学習

以下に図 2 の概要に沿って、コントローラの学習プロセスを示す。

step 1. 初期集団を生成

-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 つの遺伝子の中に入れて 50 個生成する。

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 0.6 で一点交叉を行う。

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 -0.1 ~ 0.1 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

単純な環境として、図3に示す環境でロボットコントローラの学習を行った。

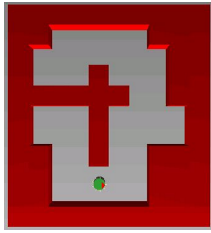


図 3: 学習環境

適応度関数を以下の様に設定する。

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

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。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にする。

コントローラを学習させる世代数を100世代と200世代にして実験を行った。コントローラを進化させる前には動き始めてからすぐに壁に衝突していたのに対して、進化させた後には、環境の中で壁に衝突せずに動くことができるようになった。

3.2 ケース1：環境変更

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3参照)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4参照)で動作させても、壁に衝突はしなかったが、図3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。

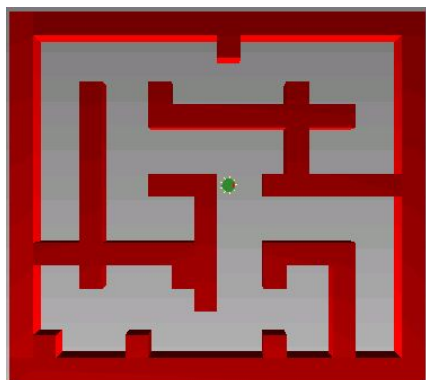


図 4: 複雑な環境

同様に、図3の環境で200世代まで学習を続けると、図3の環境の中で壁に衝突せずに動作し続けていた。また、図4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。学習した環境とは違う環境になってはいるものの、図3の環境において壁に衝突しないということを学習したため、環境が複雑になっても、センサからの情報を得て壁を回避して進むようになった。このことから、世代数を増やして学習したことにより、100世代のときと比べ、複雑な環境でも壁に衝突しないで巡回するコントローラが獲得された。

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の実験を行った。適応度関数は3.1節で設定したものと同じものを使用し、それに加え、図5の環境で白丸を通過した場合に、適応度を上げることとした。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つの白丸を交互に通過するごとに適応度に1000を足す。

通過点を設定すると、その地点を通過するように学習した。図6にその軌跡を示す。

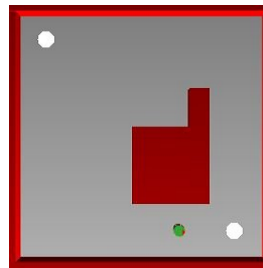


図 5: 通過点を設定した学習環境

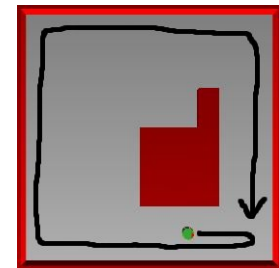


図 6: 通過点を設定した学習後の動作軌跡

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認した。今後、様々な異なる環境において学習を行い、その環境それぞれに適したコントローラを構築し、新しい環境にロボットが置かれた際、学習によって獲得されたコントローラ群の中からその環境に合ったものをロボット自身が選択して、動作できるようにしていくつもりである。

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.