

簡易コーディングを用いた EANN の倒立振り子制御問題における性能検証

Investigation of Simply Coded Evolutionary Artificial Neural Networks on a Pole-balancing Problem

摂南大 片田 喜章, 中澤 純

Yoshiaki KATADA and Jun NAKAZAWA, Setsunan University

Abstract: One of the advantages of evolutionary robotics over other approaches in embodied cognitive science would be its parallel population search. Due to the population search, it takes a long time to evaluate all robot in a real environment. Thus, such techniques as to shorten the time is required for real robots to evolve in a real environment. This paper proposes to use simply coded evolutionary artificial neural networks for robot control to make genetic search space as small as possible and investigates the performance of them on a pole-balancing problem. The results suggest the ability of the proposed approach.

1. はじめに

ロボットが実環境で振舞うためには、システム自身が高い環境適応能力を有し、環境との相互作用を通して自らの振舞いを獲得しなければならない [1]。このアプローチの一つに多点探索による高い解探索能力が期待できる人工進化を用いて自律ロボットの制御器を設計する進化ロボティクス (Evolutionary Robotics:ER)[2]がある。

ER では、制御器の評価は実環境で実ロボットを用いて行うか [3]、または、シミュレーションを用いて行われる [4]。これらの方法はそれぞれ問題点が指摘されている。実環境で実ロボットを評価するには当然、実時間がかかる。人工進化で要する進化時間は数百世代から数千世代が一般的であり、膨大な時間がかかる。しかしながら、ロボットは振舞うべき実環境で評価されているため、その環境で適切に振舞うことができる。一方、シミュレーションを用いる方法では、ロボットの評価にかかる時間は実環境に比べて非常に短くなる。しかし、シミュレーションで獲得した制御器を実ロボットに搭載した場合、シミュレーションと実環境のギャップによってロボットが必ずしも実環境で適切に振舞えるとは限らない [5]。この問題に対しては多くのアプローチ [4][6][7] が提案されているが、シミュレーションを用いることができる問題は、その環境およびロボットとの相互作用が適切にモデル化できる問題に限られる。ER のアプローチをより多くの問題に適用するには、実環境でロボットを評価する方法、つまり、実環境でロボットを進化させる時間を短縮する方法を考える必要がある。

これに対し、人工進化が解くべき問題の探索空間をできるだけ小さく設定することが考えられる。つまり、遺伝子長を短くすることができれば、どの進化アルゴリズムを採用しても基本的には進化時間を短縮することが可能であると期待できる。我々はこれまで、遺伝

アルゴリズム (GA) の遺伝子長ができるだけ短くなるように、簡易コーディングを用いた進化型ニューラルネットワーク (Simply Coded Evolutionary Artificial Neural Networks: SCEANN) を提案してきた [8]。そして、高い適応度をできるだけ早く獲得すると同時に、より高い適応度を獲得することを考慮して、Simple GA(以下 SGA) および ANN を設計する問題に対しその有効性が確認されている Operon GA[9][10][11](以下 OGA) を SCEANN に適用して、計算機実験における移動エージェント [8]・実機実験における移動ロボットおよび小型飛行船ロボット [12] に搭載し進化実験を行い、その有効性を確認してきた。

本稿では、提案手法である SCEANN の性能をさらに検証するために、制御理論分野や ER および強化学習分野などでも広くテスト問題として使われている倒立振り子制御問題に提案手法を適用する。そして、計算機において進化実験を行い、SCEANN の性能を検証する。なお、実環境でロボットを進化させることを目的としながら計算機実験を行うのは、計算機実験において良いパフォーマンスを示していることが実環境でも良いパフォーマンスを示す前提であること、また、全ての実験に関して実環境で性能検証するには膨大な時間と労力がかかることによる。

2. 簡易コーディングを用いた進化型ニューラルネットワーク (SCEANN)

本節では、GA で用いるためのニューラルネットワークの簡易コーディング法を示す。

ロボットの制御器として、中間層・出力層が相互結合された ANN を用いる。 i 番目のニューロンの時刻 t での出力は以下の式で表される。

$$x_i(t) = f\left(\sum_j \omega_{ij}x_j(t-1)\right) \quad (1)$$

ここで、 ω_{ij} は j 番目のニューロンから i 番目のニューロンへの結合荷重であり、 $f(x)$ はニューロンの出力関数であり、式 (2) で与えられるシグモイド関数を用いる。

$$f(x) = \frac{1}{1 + \exp(-x/T)} \quad (2)$$

T はシグモイド関数の傾きを決める正定数である。シグモイド関数によるニューロンの出力は比較的大きな同符号の入力が続くことによって容易に飽和してしまう。そこで、そのニューロンの入力値の合計があるしきい値 $\pm x_c$ を越えると、入力値の合計を 0 にする。本稿では、 $|x_c| = 8$ に設定する。これは、ロボットの行動レベルで見ると、同じ行動を永遠に出力し続けることを防ぐ効果がある。

一般的に進化型 ANN ではその変数として、各ニューロン間の結合荷重・ニューロンのしきい値 (シグモイド関数を用いる場合はその傾き)・ニューラルネットの構造・学習ルールが挙げられるが、本稿では遺伝子長を短くするために結合荷重のみを GA の変数とする。そのため、シグモイド関数の傾きは $T = 1$ とする。ストリングへの結合荷重のコーディング法として以下の 3 つを用いる。

SCEANN1

Floreano ら [13] は実ロボットの進化に次のようなコーディング法を用いている。ストリングは複数のブロックから構成され、1 個のブロックは中間層・出力層のニューロンそれぞれに定義される (図 1)。ブロックの最初のビットは入力・中間・出力ニューロンからそのニューロンへの全結合の符号 $\{+, -\}$ を表し、残りのビットは入力・中間・出力ニューロンからの結合それぞれの有無 $\{1, 0\}$ を表す (図 2(a))。結合荷重の大きさは $|\omega| = 1$ とする。入力ニューロン数を N_s 、中間ニューロン数を N_h 、出力ニューロン数を N_o とすると、遺伝子長は $L = (N_h + N_o)(1 + N_s + N_h + N_o)$ ビットとなる。

SCEANN2

Floreano らのコーディング法 (SCEANN1) を拡張したものとして、中間層・出力層のニューロンに入る結合それぞれに符号を設定する。つまり、ブロックは入力・中間・出力ニューロンからそのニューロンへ入る結合の符号 $\{+, -\}$ および結合の有無 $\{1, 0\}$ で構成される (図 2(b))。SCEANN1 よりも ANN の構造に多様性を持たせられる。遺伝子長は $L = 2(N_h + N_o)(N_s + N_h + N_o)$ ビットとなる。

SCEANN3

SCEANN1・SCEANN2 で考えた結合の有無をコーディングせず、入力層・中間層・出力層のニューロンを全結合とする。つまり、ブロックはそのニューロンに入る結合の符号 $\{+, -\}$ のみで構成される (図 2(c))。結合荷重の大きさは $|\omega| = 1$ とする。遺伝子長は $L = (N_h + N_o)(N_s + N_h + N_o)$ ビットとなる。

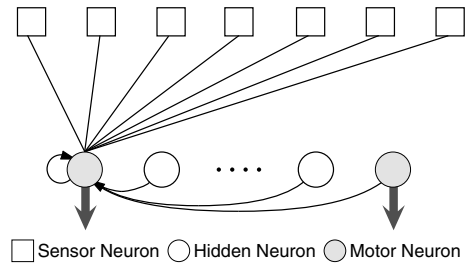


Fig.1 Architecture of ANNs for 1 block of a string

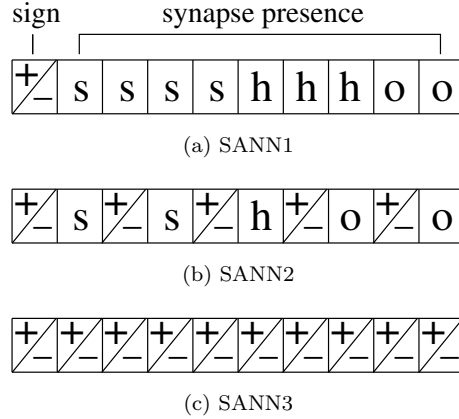


Fig.2 Genetic representation of one block

3. 計算機実験

3.1. 設定

本節では倒立振り制御問題において 2. で示した SCEANN の性能を検証する。図 3 に倒立振り制御問題の概略を示す。倒立振りとは水平方向に動く台車とその中央に回転自由で接続される一本の棒で構成される。台車は環境両端の壁に接触することなく左右に移動することで振子の倒立状態を維持しなければならない。本研究では、文献 [14][15] の設定に従い、台車が中心より ± 2.4 [m] 以内に留まり、かつ、振子が垂直状態から $\pm 36^\circ$ 以内に維持された状態を倒立状態とする。台車は力 P を水平方向に加えて動かすものとする (最大 10[N])。台車の質量を 1.0[kg]、振子の質量を 0.1[kg]、長さ 1.0[m] と設定する。単位時間ステップを 0.02[s] とし、1 ステップ毎に台車に作用させる力 P を決定する。1 個体の試行 (評価) は倒立状態を維持できなくな

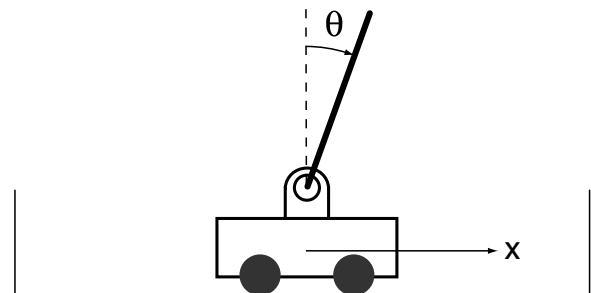


Fig.3 Experimental setup for a pole-balancing problem

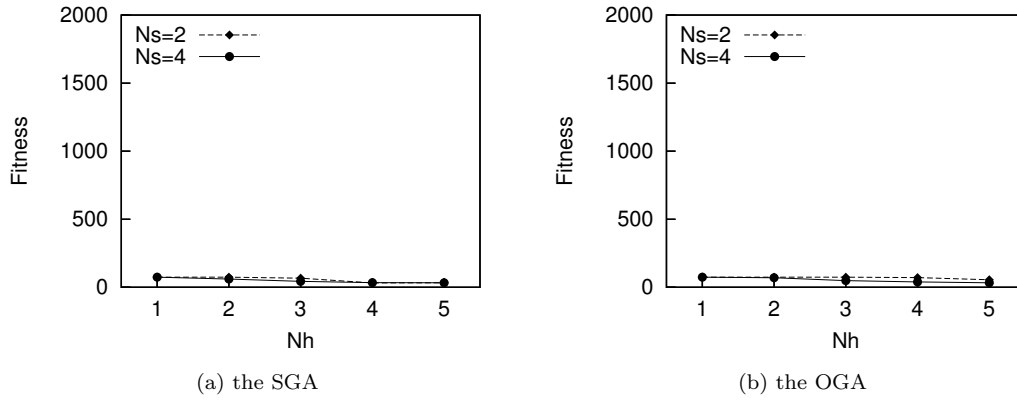


Fig.4 Maximum fitness at the final generation for SCEANN1 on pole-balancing problems

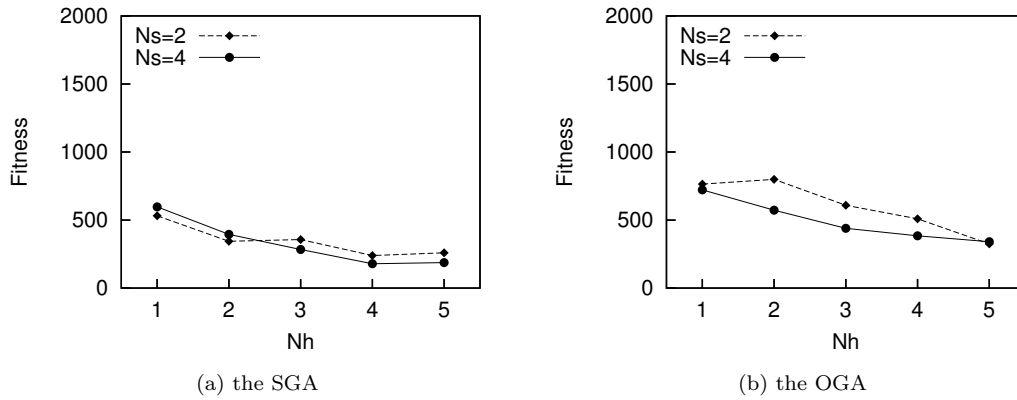


Fig.5 Maximum fitness at the final generation for SCEANN2 on pole-balancing problems

るか、または最大 3000 ステップに達したときに終了とする。本倒立振り子システムの状態は、台車の位置 x 、速度 \dot{x} 、振り子の垂直状態からの角度 θ および角速度 $\dot{\theta}$ で決定される。振り子の初期角度を $1[^\circ]$ 、その他の初期値は 0 とする。

倒立振り子の制御器として、2. で示した SCEANN を用いる。入力ニューロン数は $N_s = \{2, 4\}$ 、中間ニューロン数は $N_h = \{1, 2, \dots, 5\}$ と変化させる。出力ニューロン数は $N_o = 1$ とする。ここで、 $N_s = 4$ のときは、速度情報を含む 4 つの状態変数 $x, \dot{x}, \theta, \dot{\theta}$ が入力として与えられ、 $N_s = 2$ のときは速度情報を含まない x, θ の 2 つだけに限定する。なお、状態変数は $[0.0, 1.0]$ の範囲に正規化され、ANN に入力される。中間ニューロン数は、一般的に設計者が問題ごとに決定するが、本研究では 1. で述べたように遺伝子長を短くすることが目的であるため、比較的小さい数で実験を行う。単位ステップ毎に台車に作用させる力 P を出力ニューロンの出力値より $[-10.0, 10.0]$ の範囲の実数として決定する。適応度は、倒立状態を維持したステップ数で与える。

2. で述べたコーディングに対し進化手法として SGA および OGA を用いる [10][11]。OGA では遺伝的操作として点突然変異に加えて 5 つの遺伝的操作を用いる。これらは結合、分割、重複、欠失、逆位とよば

れ、各パラメータは文献 [10] における推奨値を参考に、 $(g_{con}, g_{div}, g_{dup}, g_{del}, g_{inv}) = (0.3, 0.3, 0.6, 0.3, 0.3)$ を使用する。各遺伝子座における値のリスト長は 6 とする。SGA では遺伝的操作として点突然変異のみを用いる。両 GA ともに点突然変異率を $q = 1/L$ とする。文献 [10] と同様の理由により、交叉は両 GA ともに用いない。選択法としてトーナメント選択を用い、エリート戦略¹ を適用する。SGA では弱い選択圧が、OGA では強い選択圧が推奨されていることから、SGA ではトーナメントサイズを 2 に、OGA では 6 に設定する。個体数を 25、世代数を 50 とし、各計算回数を 50 回とする。

3.2. 実験結果と考察

3.2.1 最終世代における適応度

各コーディング法を用いた場合の最終世代における (N_s, N_h) に対する各 GA の最大適応度 (50 試行の平均) を示す。図 4 に SCEANN1 の結果を示す。このコーディング法ではどの条件下でもステップ初期に倒立状態を維持できなくなっている。図 5 に SCEANN2 の結果を示す。 N_h が増加すると適応度は減少する傾向にあることがわかる。また、OGA は SGA よりも良いパフォーマンスを示している。図 6 に SCEANN3

¹適応度が最も高い個体を次世代の個体の 1 つとして引き継ぐ。

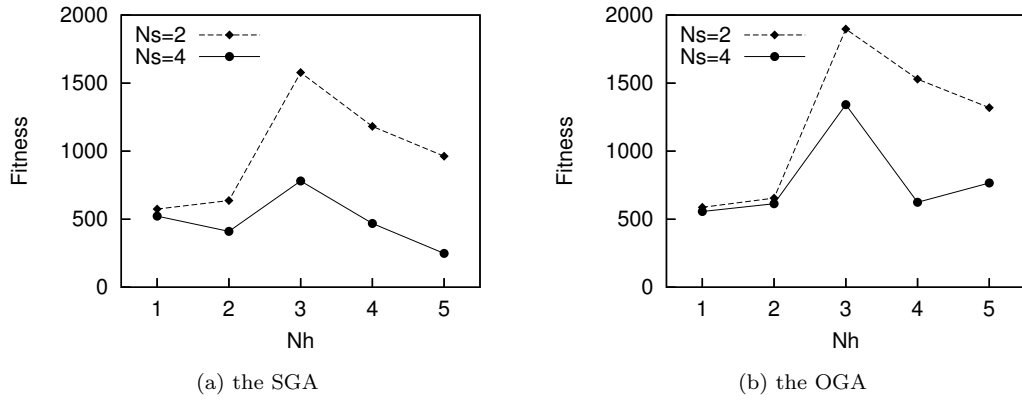


Fig.6 Maximum fitness at the final generation for SCEANN3 on pole-balancing problems

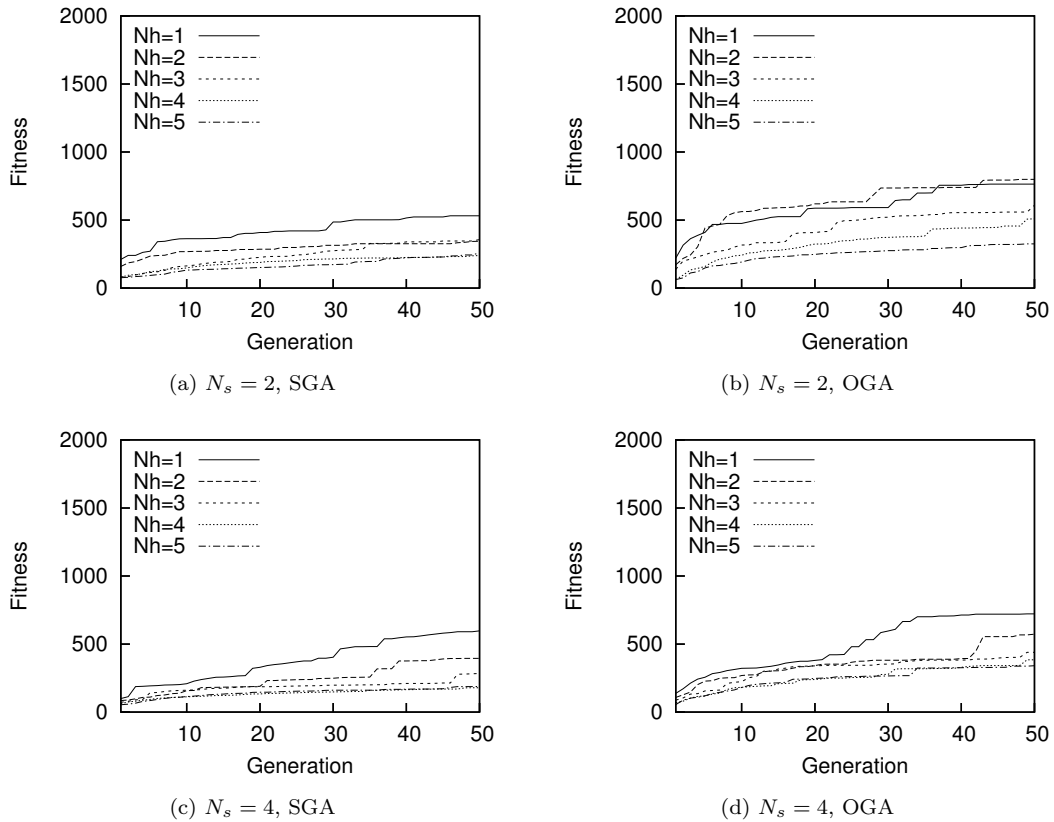
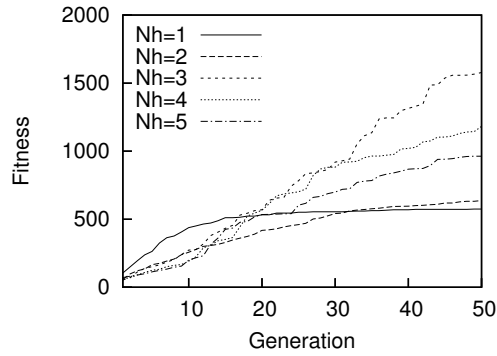


Fig.7 Maximum fitness at each generation for SCEANN2

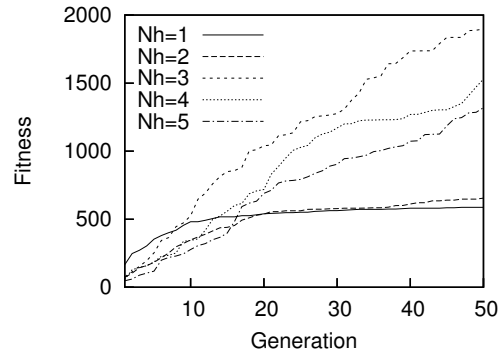
の結果を示す． N_h が増加すると適応度は増加するが， $N_h = 4$ 以上になると適応度が減少する傾向にあることがわかる． $N_s = 2$ と $N_s = 4$ の場合では，SCEANN2よりも差が明確になっている．倒立振り子制御問題において，入力に速度情報がない場合は速度情報がある場合よりも問題が難しくなることが一般的に知られている [15]．しかしながら，得られた結果は $N_s = 2$ の方が $N_s = 4$ よりも良いパフォーマンスを示している．また，SCEANN2と同様に，OGAはSGAよりも良いパフォーマンスを示している．

3.2.2 適応度の推移

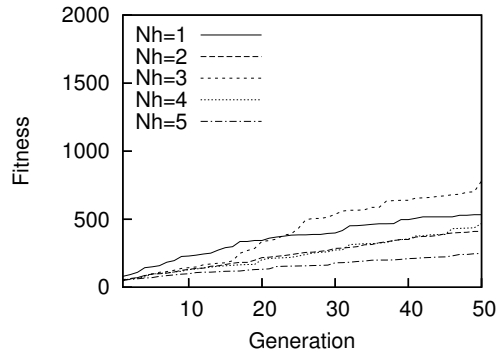
次に，最終世代までに適応度の増加が観察されたSCEANN2とSCEANN3に関して，世代数に対する各GAの最大適応度の推移を図7,8に示す．SCEANN2を用いた場合，世代数に対する適応度の増加の仕方は非常に遅い．また，最終世代数に達しても適応度が500を越える程度に留まっている(図7)．一方，SCEANN3を用いた場合，最終世代でパフォーマンスが良かった $N_s = 2$ かつ $N_h = \{3, 4, 5\}$ においてSGA・OGAともに世代が経過するにつれて適応度が増加していることが確認できる(図8)．最終世代付近においても適応度がさらに増加しており，50世代では進化しきれてい



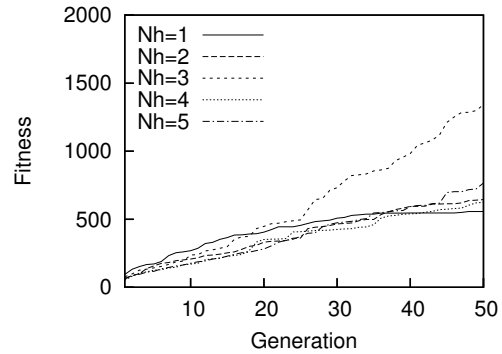
(a) $N_s = 2$, SGA



(b) $N_s = 2$, OGA



(c) $N_s = 4$, SGA



(d) $N_s = 4$, OGA

Fig.8 Maximum fitness at each generation for SCEANN3

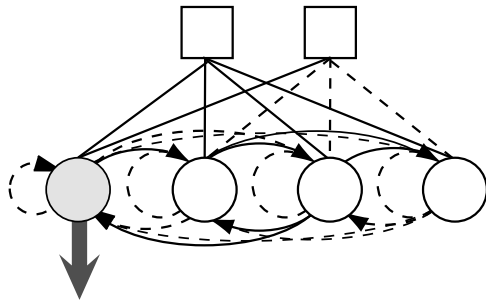


Fig.9 A genetically determined controller for SCEANN3 with $(N_s, N_h) = (2, 3)$ by the OGA on a pole-balancing problem: the solid lines indicate excitatory connection weights and the dashed lines inhibitory.

ない試行があったと推測される。

3.2.3 獲得した最良個体の構造と振舞い

本実験では、OGA を用いた SCEANN3 $((N_s, N_h) = (2, 3))$ (図 9) が最終世代において最も良い適応度を獲得した。この ANN の振舞いの中で倒立状態を最大ステップ数まで維持できた振舞いを図 10 に示す。台車が左右に移動することで、倒立状態を維持していることがわかる。また、台車の移動幅も非常に小さく、両端の壁から十分離れている。

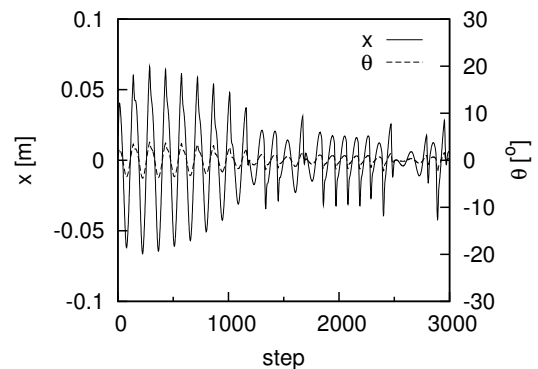


Fig.10 Transitions of the position of the cart and the angle of the pole

3.2.4 タスク成功率

前節で述べたように OGA を用いた $(N_s, N_h) = (2, 3)$ における SCEANN3 が本実験では最も良い適応度を獲得した。しかし、全ての試行において設定した最大ステップ数まで倒立状態を維持できたわけではない。表 1, 2 に SCEANN3 を用いた場合の 50 試行におけるタスクの成功率 (最終世代における最良個体が最大ステップ数まで倒立状態を維持できた割合) を示す。最も良い適応度を獲得した $(N_s, N_h) = (2, 3)$ の場合でも 50% を下回っており、提案手法および用いた設定で本倒立振り子問題を十分に解けたとは言い難い。

Table 1 Success rate (%) of SCEANN3 for the SGA in 50 runs

| $N_s \backslash N_h$ | 1 | 2 | 3 | 4 | 5 |
|----------------------|---|---|----|----|----|
| 2 | 0 | 0 | 34 | 26 | 20 |
| 4 | 0 | 0 | 12 | 4 | 0 |

Table 2 Success rate (%) of SCEANN3 for the OGA in 50 runs

| $N_s \backslash N_h$ | 1 | 2 | 3 | 4 | 5 |
|----------------------|---|---|----|----|----|
| 2 | 0 | 0 | 44 | 36 | 26 |
| 4 | 0 | 0 | 26 | 8 | 6 |

4. おわりに

実ロボットを実環境で進化させるには膨大な時間がかかる。本稿では、この実時間を短縮するために、人工進化が解くべき問題の探索空間をできるだけ小さくすることを目的として、簡易コーディングを用いた進化型ニューラルネットワーク (SCEANN) を提案した。また、高い適応度をできるだけ早く獲得すると同時に、より高い適応度を獲得することを考慮して、2種類のGAをSCEANNに適用し、倒立振子制御問題においてSCEANNの性能を検証した。得られた結果について次にまとめる。(1) 両GAとも、 $N_h = 3$ におけるSCEANN3が最も良いパフォーマンスを示した。さらに、入力として速度情報を含む場合よりも含まない場合の方が良い結果を示した。(2) 全ての (N_s, N_h) の組み合わせにおいて、OGAを適用したSCEANNはSGAのそれよりも良いパフォーマンスを示した。以上より、 $N_h = 3$ をもつSCEANN3にOGAを適用することがパフォーマンスおよび進化速度の観点から有効であることを確認した。しかしながら、多くの試行で最大適応度を得る(最大ステップ数まで倒立状態を維持する)には至らなかった。したがって、文献[8]で得られた知見および本稿で得られた結果から、SCEANNは比較的簡単なタスク(例えば移動エージェントによるゴール到達問題)では十分なパフォーマンスを示すことがわかる。今後、本稿で得られた知見をもとに、さらに実環境における進化ロボティクスの問題や進化型マルチエージェントシステムの協調問題に挑戦したい。

参考文献

[1] R. Pfeifer and C. Scheier, Understanding Intelligence, The MIT Press, 1999.
 [2] Stefano Nolfi and Dario Floreano: Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines, MIT Press, (2000)

[3] R. A. Watson, S. G. Ficici and J. B. Pollack, Embodied Evolution: Embodying an Evolutionary Algorithm in a Population of Robots, Proc. of Congress on Evolutionary Computation, 335/342 (1999)
 [4] N. Jakobi, Half-baked Ad-hoc and Noisy: Minimal Simulation for Evolutionary Robotics, Proc. of the Fourth European Conference on Artificial Life, 348/357 (1997)
 [5] R. A. Brooks, Artificial Life and Real Robots, Proc. of the First European Conference on Artificial Life, 3/10 (1992)
 [6] O. Miglino, H. H. Lund and D. Nolfi, Evolving Mobile Robots in Simulated and Real Environments, Artificial Life 2, 417/434 (1995)
 [7] 高崎, 片田, 大倉, 田浦: 進化型自律移動ロボットのオンラインモデル更新に関する一手法, 第47回自動制御連合講演会, CD-ROM 予稿集, 404, (2004)
 [8] 片田, 高田: 簡易コーディングを用いた進化型ニューラルコントローラの性能評価, 第34回知能システム・シンポジウム資料, 149/154 (2007)
 [9] 大倉, 上田: 中立突然変異型GAによる騙し問題の最適化, 計測自動制御学会論文集, vol. 32, no. 10, 1461/1469 (1996)
 [10] 片田, 大倉, 上田: Neutral Networksを含む適応度景観における遺伝的アルゴリズムの進化ダイナミクス, システム制御情報学会論文誌, vol. 17, no. 5, 187/195 (2004)
 [11] 片田, 大倉: ニュートラルネットワークを含む適応度景観における遺伝的アルゴリズムの進化ダイナミクス-進化ロボティクスでの検証, システム制御情報学会論文誌, vol. 21, no. 2, 31/39 (2008)
 [12] 片田, 永松, 森川: 簡易コーディングを用いた進化型ロボットの実環境における行動獲得, 第17回インテリジェント・システム・シンポジウム (FANシンポジウム) 講演論文集, 237/242 (2007)
 [13] D. Floreano and C. Mattiussi: Evolution of Spiking Neural Controllers, Proc. of Evolutionary Robotics -From Intelligent Robots to Artificial Life (ER'01), 38/61, Springer-Verlag (2001)
 [14] D. E. Moriarty and R. Miikkulainen: Efficient Reinforcement Learning through Symbiotic Evolution, Machine Learning, Kluwer Academic Publishers, vol. 22, 11/33 (1996)
 [15] F. J. Gomez and R. Miikkulainen: Solving Non-Markovian Control Tasks with Neuro-Evolution, Proc. of The International Joint Conference on Artificial Intelligence (IJCAI99), 1356/1361 (1999)