

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

An Evolutionary Autonomous Mobile Robot with An On-line Model Update Mechanism

高崎真也 (神戸大学) 片田喜章 (神戸大学)
大倉和博 (神戸大学) 田浦俊春 (神戸大学)

SHINYA Takasaki, Kobe University, Rokkoudaichou1-1. Nada-ku, Kobe, Hyougo
YOSHIAKI Katada, Kobe University
KAZUHIRO Ohkura, Kobe university
TOSHIHARU Taura, Kobe University

One of advantages of evolutionary robotics over other approaches in embodied cognitive science would be its parallel population search. Due to the population search, it takes an unrealistically long time for some kinds of task to evaluate all robot in a real environment. Thus, a technique of computer simulation is one of the most important topics in evolutionary robotics. However, a simulated robot must be synchronously consistent with the real one during its evolution for achieving appropriate behavior. In this paper, we propose an update method of a simulated robot based on distributions of a real robot's displacements during evolution, and investigate whether an evolved robot shows appropriate behavior.

Key words Embodied Cognitive Science, Evolutionary Robotics, Genetic Algorithm, Omni-directional Camera

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、環境とのインタラクションを通して、自らの振舞いを獲得しなければならない。このフレームワークに基づく研究領域は身体性認知科学 (Embodied Cognitive Science)[1] と呼ばれている。

Embodied Cognitive Science の一分野として、人工進化を用いて自律ロボットの制御機構を設計する進化ロボティクス (Evolutionary Robotics: ER) がある。しかし、ER では解探索集団中の個体を全て実環境で進化させるには膨大な実時間がかかり、人の介入を必要とする類のタスクでは実質的に進化させることは不可能になる。この問題に対し、実時間を削減させる研究として、同じ構造のロボットを複数台用意し、並行して進化させる研究 [2] や、シミュレーションを用いる研究等がある。ロボットを複数台用意する場合、同じ構造でもセンサーやアクチュエーターなどの部品によっては、機械的・電気的な性能の違いが生じるため、全てのロボットで適切に振舞う制御機構を獲得することは困難である。また、シミュレーションを用いた場合、ノイズ等の影響により実環境のセンシングとアクチュエーションを正確にモデル化できず、シミュレーションで獲得された振舞いが必ずしも実際のロボットに反映されるとは限らない [3] という問題がある。この問題に関して Jakobi らは Minimal Simulation を提案した [4]。これは、求める振舞いの創発に関連するロボットと環境の相互作用のみをモデル化するという方法であるが、実験者の経験に大きく依存する。また、Miglino らはあらかじめ全てのセンサ、アクチュエーターに関して、細かい精度でサンプルデータを用意し、精度の高いシミュレーションモデルを構築した [5]。これは、あらかじめサンプルデータを用意しているため、時間の経過によるセンサ・アクチュエーターの機械的・電気的な変化に対応できないという問題がある。この問題に対して、オンラインでサンプルデータを取得していきシミュレーションモデルを構築する方法が考えられる。

本稿では上述の問題点に対し、集団中の数個体を実環境で評価し、残りをシミュレーション上のモデル環境で評価するというアプローチを取る。実環境の特徴をオンラインでモデル環境に反映させるため、実環境で個体を評価している間に実環境のデータを取得し、シミュレーション上のモデル環境を構築する。また、実環境との整合性を保つために、仮説検

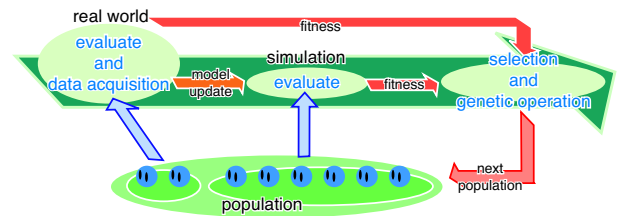


Fig. 1: Evolution based on on-line model update

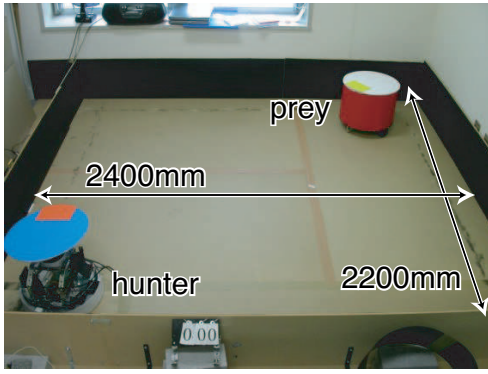
定を用いてモデル環境を更新する方法を提案し、その有効性を検証する。

2 オンラインモデル更新を用いた進化

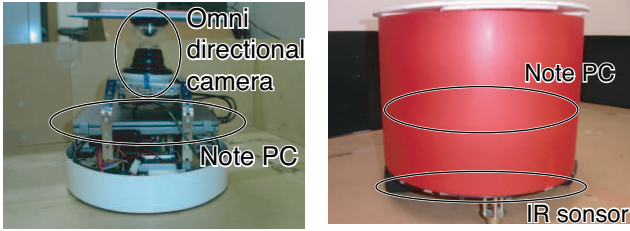
ER では、ノイズ等の影響により、センサ入力 s と、モータ出力 a で構成されるシミュレーション上のモデル環境 $M(s_M, a_M)$ と実環境 $R(s_R, a_R)$ との間にギャップが生じる。また、時間の経過に伴い $R(s_R, a_R)$ が変化する。これらの問題に対し、 $R(s_R, a_R)$ を基に $M(s_M, a_M)$ をオンラインで構成・更新することを考える。このオンラインモデル更新を用いた進化計算プロセスは次のようにまとめられる (Fig.1)。

1. 実環境で評価する個体とシミュレーションで評価する個体に分ける。
2. 実環境で評価する間に、 $R(s_R, a_R)$ から、センサ入力とモータ出力のサンプルデータを取得する。
3. サンプルデータを用いて $M(s_M, a_M)$ を更新し、シミュレーションで評価する。
4. 実環境とシミュレーションで得た評価値に基づき選択を行い、遺伝的操作により次世代の個体を生成する。
5. 終了条件に達するまで 1→4 を繰り返す。

ここで、 $M(s_M, a_M)$ を更新する際に、 $R(s_R, a_R)$ との整合性を検定するために t 検定を用いる。検定の方法は以下のように行う。



(a) Experimental field



(b) Hunter robot with omnidirectional camera

(c) Prey robot with IR sensors

Fig. 2: Experimental environment

$M(s_M, a_M)$ 用のサンプルデータ (データ数: n) の平均を μ_M , 標準偏差を σ_M , $R(s_R, a_R)$ から得られたデータの平均を μ_R としたとき, $\mu_M = \mu_R$ という仮説に基づいて有意水準 $\alpha = 0.05$ で検定する. 仮説が棄却されたとき, $M(s_M, a_M)$ 用のサンプルデータと $R(s_R, a_R)$ から得られたデータとの間に差が生じているため, サンプルデータを補正し, μ_M を μ_R に近づけ, 差を埋めていく. サンプルデータの補正方法については 3.3.1 で詳述する.

3 シミュレーションモデルの構築

3.1 実験環境

本稿では自律移動ロボットを用いた追跡問題を取り扱う. 実験環境を Fig.2(a) に, 実験に用いるハンターロボットを Fig.2(b) に, 獲物ロボットを Fig.2(c) に示す. 実験環境は平面で, 2400mm×2200mm の長方形であり, 四方は壁で囲まれている. 進化の対象はハンターロボットとする.

ハンターロボットは全方向カメラを搭載し, 画像処理とロボットの制御を行うノート PC を搭載している. 直径 350mm, 独立駆動型の 2 輪式で, コントローラとして隠れノードと出力ノードが完全相互結合したリカレント型の Pulsed Neural Networks(PNNs) を用いる. PNNs の入力ノード数は 64, 隠れノード数は 1, 出力ノード数は 2 とする.

獲物ロボットは周囲を観測するセンサとして IR センサを 8 個 (前方 6 つ, 後方 2 つ) 搭載し, ロボットの制御を行うノート PC を搭載している. 形状は赤色の円筒形とし, 直径 350mm, 独立駆動型の 2 輪式で, コントローラはブライテンベルグビーグル [1] を基に IR センサを用いて障害物回避行動をするように設定した (Fig.3).

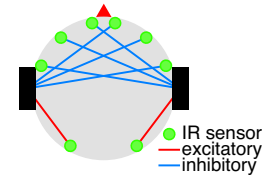
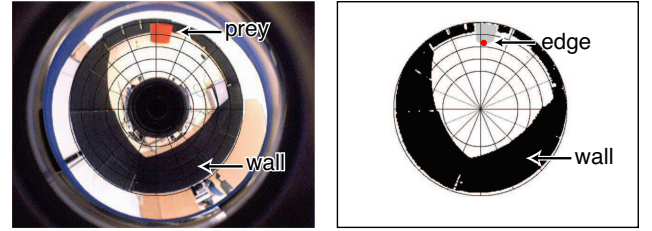


Fig. 3: Control architecture of a prey robot



(a) Camera image

(b) Binarized image

Fig. 4: Omni-directional image plane

3.2 入力モデル

3.2.1 ハンターロボット

全方向カメラを用いる場合, 一般的に角度情報は信頼性が高く計算も容易である一方, 距離情報は計算が複雑になる. しかし, 平坦な環境では距離情報は計算が容易であり [6], 取得した画像の中心から同心円上では, 全方向カメラの中心からの距離がほぼ一定となる. そこで, Fig.4(a) に示すように円周方向に一定の角度 (22.5°) で 16 分割し, 半径方向に同心円で 4 分割して計 64 個のセルに画像を分割した. 各セルにおける獲物ロボットと壁の存在情報を PNNs の入力とする.

[獲物ロボットの位置情報]

1. 全方向カメラの画像に対し, 2 値化, ラベリング, 重心算出の画像処理を行い, 獲物ロボットと床面とが接しているエッジ部分を検出する (Fig.4(b)).
2. 検出されたエッジ部分の位置に対応するセルが円周方向に a 番目, 半径方向に b 番目のとき, そのセルと, そのセルより半径方向に外側のセルに対応した入力ノードに入力値を与える (式 (1)).

$$\begin{cases} Cell_{h,k} = 1.0 & \text{if } h = a, k = b, \dots, 4 \\ Cell_{h,k} = 0.0 & \text{otherwise} \end{cases} \quad (1)$$

[壁の位置情報]

1. 壁は黒色で示す (Fig.2(a)). 全方向カメラの画像に対し 2 値化の処理を行い, 画像上の壁の面積 (ピクセル数) を計算する (Fig.4(b)).
2. 壁のピクセル数が閾値以上のセルが存在する時, そのセルに対応した入力ノードに入力値を与える (式 (2))

$$\begin{cases} Cell_{h,k} = 1.0 & \text{if } P_{W(h,k)} \geq \phi \cdot P_{C(h,k)} \\ Cell_{h,k} = 0.0 & \text{otherwise} \end{cases} \quad (2)$$

$P_{W(h,k)}$: 円周方向に h , 半径方向に k のセルの壁のピクセル数

$P_{C(h,k)}$: 円周方向に h , 半径方向に k のセルの総ピクセル数

ϕ : 閾値の割合 (=0.1)

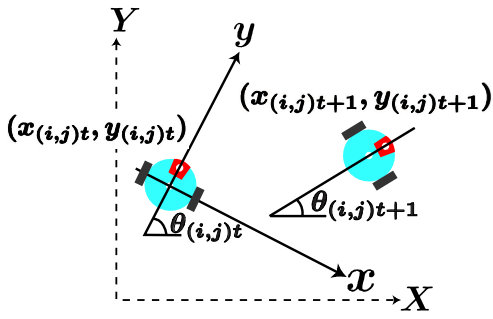


Fig. 5: Displacement of a mobile robot

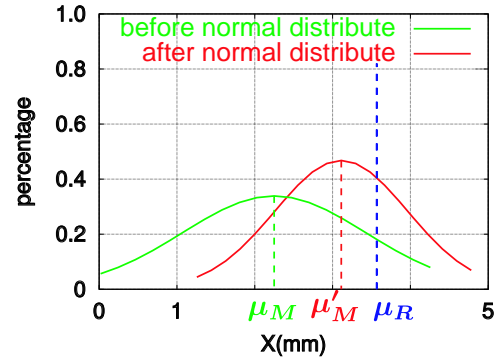


Fig. 6: Data correction

シミュレーションではこの入力モデル s_M は一定であると仮定し、更新は行わない。

3.2.2 獲物ロボット

実環境であらかじめ IR センサのサンプルデータを取得し、サンプルデータを基に正規分布を用いてシミュレーションの入力モデルとした。シミュレーションではこの入力モデル s_M は一定であるものと仮定し、更新は行わない。

3.3 出力モデル

3.3.1 ハンターロボット

左右の各モータコマンドは、前進 10 通り、後進 10 通りの計 20 通りの出力が可能であり、 20×20 の計 400 通りのモータコマンドにより行動する。出力モデルは各モータコマンド (i, j) (i : 左モータコマンド, j : 右モータコマンド) に対する移動量 $(x(i, j), y(i, j), \theta(i, j))$ とする。これらの移動量は、実環境から得られたサンプルデータを基に正規分布を用いてシミュレーション内でモデル化する [7]。次に出力モデルの更新について述べる。

1. 実験環境の上方に CCD カメラを設置し、画像処理ボード (日立 IP5005) を用いて、ロボットの重心位置を取得する。
2. $0.5s$ の間隔で移動した距離を重心位置から測定し、そのときのモータコマンドに対する実環境での移動量 $(x(i, j)_R, y(i, j)_R, \theta(i, j)_R)$ を取得する (Fig.5)。

$$\begin{cases} x(i, j)_R = x(i, j)_{t+1} - x(i, j)_t \\ y(i, j)_R = y(i, j)_{t+1} - y(i, j)_t \\ \theta(i, j)_R = \theta(i, j)_{t+1} - \theta(i, j)_t \end{cases} \quad (3)$$

3. 実環境の評価が終わった後、取得した移動量のデータを加える (式 (4))。

$$\begin{aligned} (X(i, j), Y(i, j), \Theta(i, j))_{g+1} \Leftarrow \\ (X(i, j), Y(i, j), \Theta(i, j))_g + (X(i, j)_R, Y(i, j)_R, \Theta(i, j)_R) \end{aligned} \quad (4)$$

$(X(i, j), Y(i, j), \Theta(i, j))_g$: g 世代目においてモデル環境を構築している各移動量のデータ集合

$(X(i, j)_R, Y(i, j)_R, \Theta(i, j)_R)_g$: g 世代目に実環境で得られた各移動量のデータ集合

4. 2. に基づき、データ補正を行う。仮説が棄却されたとき、過去 g' 世代の間に実環境から得られたデータ $\sum_{g-g'}^g (X(i, j)_R, Y(i, j)_R, \Theta(i, j)_R)_g$ の平均 μ_R に $(X(i, j), Y(i, j), \Theta(i, j))_g$ の平均 μ_M が近づくように補正し、 $\sum_{g-g'}^g (X(i, j)_R, Y(i, j)_R, \Theta(i, j)_R)_g$ と $(X(i, j), Y(i, j), \Theta(i, j))_g$ との差を埋める。Fig.6 のように $\mu_R \neq \mu_M$ の場合、 μ_M を大きくするために、モデル環境を構築しているサンプルデータの中から値の小さいデータを削除する。補正後の平均 μ'_M と μ_R について再び検定をし、仮説が採択されるまでこれを繰り返す。同様に $\mu_R \neq \mu_M$ の場合、値の大きいサンプルデータを削除する。

5. 各移動量の平均 μ と標準偏差 σ を計算し、 $(x(i, j)_M, y(i, j)_M, \theta(i, j)_M)$ を求める (式 (5))。

$$\begin{cases} x(i, j)_M = N(\mu(X(i, j)_g), \sigma(X(i, j)_g)) \\ y(i, j)_M = N(\mu(Y(i, j)_g), \sigma(Y(i, j)_g)) \\ \theta(i, j)_M = N(\mu(\Theta(i, j)_g), \sigma(\Theta(i, j)_g)) \end{cases} \quad (5)$$

なお、予備実験として、全モータコマンドに対する移動量のサンプルデータを 10 ずつ用意し、初期世代の出力モデル $(x(i, j)_M, y(i, j)_M, \theta(i, j)_M)$ を構築する。本稿では、上記の手法を 50 世代毎に適用し、 $g' = 500$ として $(x(i, j)_M, y(i, j)_M, \theta(i, j)_M)$ を更新していく。

3.3.2 獲物ロボット

左右の各モータコマンドは、前進 4 通り、後進 4 通りの計 8 通りの出力が可能であり、 8×8 の計 64 通りのモータコマンドにより行動する。出力モデルの更新に関しては 3.3.1 と同様とする。

4 実験

4.1 実験設定

実環境・シミュレーションともにハンターロボットの初期姿勢を Fig.2(a) における上向きと右向き の 2 パターンに設定する。獲物ロボットの初期位置は環境におけるハンターロボットの対角線上で、初期姿勢は各試行毎にランダムな向きとする。各個体につき 2 試行を行い、1 試行はハンターロボットが獲物ロボットを捕獲するか、40 ステップで終了とする (1 ステップは約 $1s$)。進化計算に遺伝的アルゴリズムを用い、ハンターロボットの PNNs の結合係数としきい値を変数とする。遺伝子長 2040、個体数 50 とし、トーナメントサイズ 2 のトーナメント選択を行い、エリート戦略 (エリート数: 1) を用いる。突然変異率は 0.00049 とし、交叉は用いない。世代数を 2000 世代とし、50 世代毎に前世代において評価値の高い上位 2 個体を実環境上で評価し、残りの個体と他の世代をシミュレーションで評価する、評価値は次のように設定する。

$$\Phi = \left\{ \sum_{i=1}^{Numtrial} (1.0 - Step/MaxStep) \right\} / Numtrial \quad (6)$$

$Step$: ハンターロボットが獲物ロボットを捕獲したステップ数

$MaxStep$: 40

$Numtrial$: 試行回数 (2)

4.2 実験結果

評価値の推移を Fig.7 に、各 50 世代毎に評価値の上位 2 個体を実環境で評価した値を Fig.8 に示す。進化の初期では

実環境で獲物ロボットを捕獲できない振舞いが見られたが、1000 世代以降は獲物ロボットを追跡する振舞いが多く見られるようになった。また、初期の世代では壁に衝突したまま壁から離れる振舞いは見られないが、最終世代では Fig.9, Fig.10 に見られるように、壁に接近した際に回転して壁から遠ざかる振舞いがシミュレーションと実環境の双方で見られるようになった。

仮説検定を用いてモデル環境を更新する提案手法の有効性を検証するために、オフラインモデル進化として、初期世代のモデル環境を用いて、シミュレーションのみで 2000 世代まで進化させて得られた最終世代の最良個体と、提案手法で得られた最終世代の最良個体を実環境のロボットに載せた際のタスク達成率を比較した。それぞれ、上向き 10 試行、右向き 10 試行の計 20 試行ずつ、60 ステップで行った。Table.1 に示すように、オフラインモデル進化で得られた最良個体は、獲物ロボットを上向きに対して 50%、右向きに対して 60%、全試行に対して 55%の確率で捕獲した。また、提案手法であるオンラインモデル更新で得られた最良個体は、獲物ロボットを上向きに対して 50%、右向きに対して 90%、全試行に対して 70%の確率で捕獲した。この結果より、提案した手法を用いて得られた最良個体の方が、オフラインモデル進化を用いて得られた個体よりも達成率が高く、提案手法の有効性が確認された。

5 おわりに

本稿では、自律移動ロボットの追跡問題をタスクとし、実環境のデータを用いてシミュレーションモデルを構築・更新

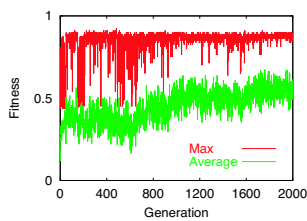


Fig. 7: Fitness for each generation in the simulated environment

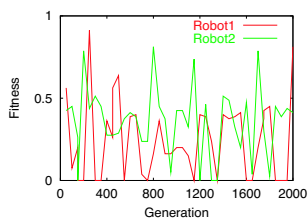


Fig. 8: Fitness for each 50 generation in the real environment

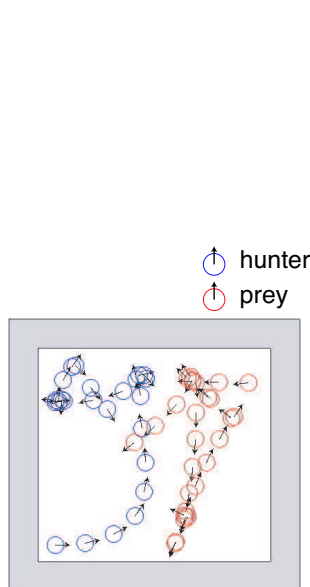


Fig. 9: Behaviour of the best evolved robot in the simulated environment

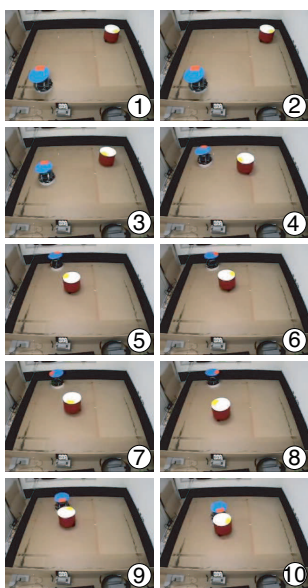


Fig. 10: Behaviour of the best evolved robot in the real environment

	upward	rightward	total
<i>Off-line model</i>	50%	60%	55%
<i>On-line model update</i>	50%	90%	70%

表 1: Success rate(%) for each method

することで、実環境とシミュレーション上のモデル環境のギャップを埋める方法を提案した。提案した手法で最終世代まで進化させて得られた最良個体を実環境のロボットで評価したところ、タスクを達成する振舞いを獲得していることから、提案したシミュレーションモデルの構築方法の有効性を確認した。また、シミュレーション上のモデル環境と実環境の整合性を検定するために t 検定を用いたデータ補正によるモデル環境の更新方法を提案した。実験結果より、実環境でロボットを評価した場合、あらかじめ用意したシミュレーションモデルのみで進化させるより、本稿で提案した手法の方が、タスク達成率が高いことを確認した。これより、本稿で提案した入力モデルの構成と出力モデルの構成・更新方法の有効性を示すことができた。今後は追跡問題において、獲物ロボットも進化する共進化問題を取り扱い、本稿で提案した手法を用いて実験を行っていく予定である。

参考文献

- [1] R. Pfeifer and C. Scheier, Understanding Intelligence, The MIT Press, 1999.
- [2] R. A. Watson, S. G. Ficci and J. B. Pollack, Embodied Evolution: Embodying an Evolutionary Algorithm in a Population of Robots, Proceedings of Congress on Evolutionary Computation, pp.335-342, 1999.
- [3] R. A. Brooks, Artificial Life and Real Robots: Proceedings of the First European Conference on Artificial Life, pp.3-10, 1992.
- [4] N. Jakobi, Half-baked Ad-hoc and Noisy: Minimal Simulation for Evolutionary Robotics, Proceedings of the Fourth European Conference on Artificial Life, pp.348-357, 1997.
- [5] O. Miglino, H. H. Lund and D. Nolfi, Evolving Mobile Robots in Simulated and Real Environments, Artificial Life 2, pp.417-434, 1995.
- [6] 森 信人, 家田 純一, 松井 渉, 全方位移動機構と全方位視覚を有する小型ロボットによるサッカー競技の実現 -チーム OMNI の戦略-, 人工知能学会 第 6 回 SIG-Challenge 研究会論文集, Vol.1, pp.42-47, 2000.
- [7] 小森谷 清, 大山 英明, 谷 和男, 移動ロボットのためのランドマーク観測計画, 日本ロボット学会誌, Vol.11, No.4, pp.533-540, 1993.