進化型移動ロボットの検定に基づくシミュレーションの更新法 An Update Method of Computer Simulation for an Evolutionary Mobile Robot Based on Statistical Tests

片田 喜章後藤 龍彌Yoshiaki KATADATatsuya GOTOH損南大学損南大学Setsunan UniversitySetsunan Universitykatada@ele.setsunan.ac.jp

Abstract: One of advantages of evolutionary robotics over other approaches in embodied cognitive science is its parallel population search. However, it generally takes an unrealistically long time to evaluate all candidate solutions by using a real robot. Thus, it makes difficult to use physical robots for evolution without any consideration, especially in the case of tasks with human intervention. One approach to overcoming this problem would be the use of simulations. Computer simulations may be helpful to reduce the amount of experimental time to evaluate individuals in a population. However, the controllers evolved in a simulated environment do not always work well in the real one because of uncertain effects, e.g., noise and differences in the electronics and mechanics of robots. Therefore, the validity of simulations is a particularly relevant problem. In this paper, in order to overcome this problem, we propose an update method of simulated robot/environment interaction dynamics based on statistical tests during evolution. Then we conduct a series of experiments with a mobile robot in order to investigate the effectiveness of the proposed method.

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.

進化ロボティクスでは , 制御器の評価は a) 実環境で実 ロボットを用いる [3],または,b)シミュレーションを用いて行われる [4].a)の方法ではロボットは振舞うべき実 環境で進化するため,その環境で適切に振舞うことができ る.しかしながら,集団中のすべての個体を実環境で進化 させるには膨大な実時間がかかる.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, さらに人の 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 進化させる研究がある[3].このように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, すべてのロボットで適切に振舞う制御器を獲得する ことは難しいと予想される.また,ロボットを複数台用意 するためのコストもかかる.片田ら [5] は,可能な限り人 工進化が解くべき問題の探索空間を小さくする, つまり 遺伝子長を短くし,進化時間を短くする方法を提案してい る.しかし,この方法は設計変数の定義域を粗くコーディ ングしていることになり,望ましい振舞いを示す個体もし くは実行可能個体を獲得できない可能性がある

一方,b)の方法では,ロボットの評価にかかる時間を非常に短くすることができるが,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[6].この問題に対してJakobi らは Minimal Simulation を提案している[4].これは,望 ましい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らはあらかじめすべてのセン サ・アクチュエータに関して,細かい精度でサンプルデー タを取得し,精度の高いシミュレーションモデルを構築し た[7].しかし,この方法はすべてのサンプルデータを事前 に用意しているため,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.

上述の問題点に対し、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 [8] . この考え方に基づき、高崎ら [9] は、個体集団 中で実環境とシミュレーションで評価する個体を分け、実 環境で評価している間に実環境の入出力に関するデータを 取得し、オンラインでシミュレーションモデルを更新する 方法を提案している.しかし、高崎らの方法では、実環境 とシミュレーションモデルの整合性は十分に保証されてい ないという問題がある (3.4 節で詳述).そこで、本研究で は高崎らの方法に改良を加え、実環境とシミュレーション モデルの整合性が十分に保証される更新法を提案する.提 案手法の有効性を、移動ロボットを用いたゴール到達問題 において検証する.

2 シミュレーションモデルの更新法

実環境におけるセンサ入力-モータ出力関係のサンプルデー タ集合を R(s,a)(s:センサ入力,a:モータ出力)とし,シミュ レーションにおける入出力データ集合を M(s,a)と表す.



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

ここで, R(s,a)をもとにM(s,a)をオンラインで構成・更新する.この方法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).

- ステップ1 個体集団を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.
- ステップ2 実環境で個体を評価する間に, R(s,a)を取得し, M(s,a) に加える.
- ステップ3 実環境での評価後, *R*(*s*,*a*)をもとに検定を行 い, *M*(*s*,*a*)を更新する. 更新された *M*(*s*,*a*)を用い て, シミュレーションで残りの個体を評価する.
- ステップ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.

ステップ5 終了条件に達するまで1→4を繰り返す.

ステップ 3 で, R(s,a) と M(s,a) の差が有意でなければ R(s,a) と M(s,a) は等しいとする.もし有意であれば,有意でなくなるまで M(s,a) を繰り返し修正する.

3 移動ロボットのシミュレーションモデル

3.1 実験環境

本稿ではロボット制御問題の1つであるゴール到達問題に おいて2節で示した提案手法の有効性を検証する.

実験環境を Fig. 2(a) に,実験に用いる移動ロボットを Fig. 2(b) に示す.実験環境は平面で,3000 mm×3000 mm の正方形であり,周囲は壁で囲まれている.移動ロボット は全方位カメラを搭載し,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380 mm,左右二輪独 立駆動型である.ロボットが全方位カメラ画像を通して認 識できる目標物(ゴール)の形状は赤色の円筒形(直径300 mm・高さ290 mm)とする(Fig. 2(c)).

3.2 センサ入力モデル

全方位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が,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[10], 取得した画像の中心から同心円上では,全方位カメラの中 心からの距離がほぼ一定となる.そこで,Fig.3(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7分割して計112個のセルに画像を分割



(a) real environment



Fig. 2: Experimental setup for a real environment



Fig. 3: Omni-directional image plane

した.4節で説明する目標物検出法に基づき,各セルにお いて目標物の存在を検出する.上で述べたように全方位カ メラを用いた場合,平坦な環境では比較的高い精度で距離 情報が得られることから,本研究ではセンサ入力モデルの 更新は行わず一定とする.

3.3 モータ出力モデル

文献 [11] から,左右モータへの速度入力に対する移動ロ ボットの実環境での移動量 (x_R, y_R, θ_R) (Fig. 4) は,実環 境で計測した移動量データから求まる平均 $(\mu_{x_R}, \mu_{y_R}, \mu_{\theta_R})$ と分散 $(\sigma_{x_R}^2, \sigma_{y_R}^2, \sigma_{\theta_R}^2)$ の正規分布で表されることが報告 されている ((1)式).

$$\begin{aligned} x_R &= N(\mu_{x_R}, \sigma_{x_R}^2) \\ y_R &= N(\mu_{y_R}, \sigma_{y_R}^2) \\ \theta_R &= N(\mu_{\theta_R}, \sigma_{\theta_R}^2) \end{aligned}$$
(1)

本研究で用いる移動ロボットの任意の速度入力(デュー ティ比:4節で述べる)に対する移動量は正規分布に従うこ とを予備実験において確認していることから,(1)式をモー タ出力モデルとして採用する.



Fig. 4: Displacement in a relative coordinate

3.4 仮説検定に基づく出力モデル更新

3.2 節で述べたように,入力モデルに関しては一定と仮定 しているため実験の間更新しないが,出力モデルに関して は実環境の移動量データを基に更新する.このとき,はじ めにで述べたように,時間経過に伴い,実環境で得られた 移動量データとシミュレーションモデルを構成している移 動量データとの間にギャップが生じてしまう可能性がある そこで,両データの整合性を保証するため,仮説検定に基 づく更新を行う.シミュレーションモデルの更新法の概略 についてはすでに2節で述べた.本節では2節の進化計算 プロセス(ステップ2,3)について詳述する.

3.3 節で述べたように, モータへの速度入力に対する移動量は正規分布に従うことから, 平均と分散を考慮し, t 検定および χ^2 検定を用いる(はじめにで述べたように, 文献[9]の方法では平均しか考慮されていない).以下にその方法を示す.

- 実環境の上方に CCD カメラを設置し,ロボットの重 心位置と前方位置を取得する.このとき,ロボット の重心位置は Fig. 2(b) で示すロボットの上に取りつ けた青いマーカー (Fig. 4参照)の中心で,前方位置 は赤いマーカーの中心である.
- 2. Fig. 5 に示すように, ロボットの重心位置と前方位置 から絶対座標におけるロボットの位置・姿勢 (x_t, y_t, θ_t) を計算し, 同様に1秒後のロボットの位置・姿勢 $(x_{t+1}, y_{t+1}, \theta_{t+1})$ を計算し,速度入力に対する実環 境での相対移動量 (x_R, y_R, θ_R) (Fig. 4)を取得する ((2) 式).

$$\begin{cases} x_R = x_{t+1} - x_t \\ y_R = y_{t+1} - y_t \\ \theta_R = \theta_{t+1} - \theta_t \end{cases}$$
(2)

モデル更新は x_R , y_R , θ_R の各座標軸および各速度 入力値の組合わせに対し独立に行う.表記の簡略化 のため,以降,相対移動量 x_R についてのみ説明を行 い, y_R , θ_R については省略する.

3. 取得したモータコマンドに対する相対移動量 x_R の集 合 X_R をシミュレーション構成用の移動量データ集 合 X_M に加える.

$$X_M \leftarrow X_M + X_R \tag{3}$$

また, X_R を現在の世代から 500 世代前までに実環 境で得られた移動量データの集合 X_R, に加える (同



Fig. 5: Displacement in an absolute coordinate

時に現世代から 500 世代以前のものは X_{R'} から削除 する).

$$X_{R'} \leftarrow X_{R'} + X_R$$
 (4)
検定回数を表すフラグを $count = 0$ とする.

- 4. シミュレーション構成用の移動量データ集合 X_M の 平均を μ_{X_M} ,分散を $\sigma^2_{X_M}$,実環境から得られた各移 動量データ集合 $X_{R'}$ の平均を $\mu_{X_{R'}}$,分散を $\sigma^2_{X_{R'}}$ と する. $\sigma^2_{X_M} = \sigma^2_{X_{R'}}$ を仮説 1 とし,有意水準 $\alpha = 0.05$ で χ^2 検定を行う.
- 5. 仮説 1 が棄却されたとき $X_M \ge X_{R'}$ の間に差が生じ ているため, X_M を補正し, $\sigma_{X_M} \ge \sigma_{X_{R'}}$ に近づけて いく.ここで, $\sigma_{X_M} < \sigma_{X_{R'}}$ のとき, X_M に関して 明確な方針の下で補正を行えないため, X_M をその まま $X_{R'}$ に置き換える. $\sigma_{X_M} \ge \sigma_{X_{R'}}$ のとき, X_M のデータ補正を行う.ここで, $\mu_{X_M} < \mu_{X_{R'}}$ のとき, X_M において値のもっとも小さいデータを削除する ことで $\sigma_{X_{R'}}$ に近づける.逆に, $\mu_{X_M} \ge \mu_{X_{R'}}$ のと き, X_M において, 値の最も大きいデータを削除す ることで $\sigma_{X_{R'}}$ に近づける.このとき,補正するデー タ数は 1 ずつとする.4. へ戻る. 仮説 1 が採択されたとき, 6 へ進む.
- μ_{XM} = μ_{XR} を仮説 2 とし,有意水準 =0.05 でt検 定を行う.
- 7. 仮説 2 が棄却されたとき X_M と $X_{R'}$ の間に差が生じ ているため, X_M を補正し, μ_{X_M} を $\mu_{X_{R'}}$ に近づけて いく.ここで, $\mu_{X_M} < \mu_{X_{R'}}$ のとき, X_M において 値のもっとも小さいデータを削除することで $\sigma_{X_{R'}}$ に 近づける.逆に, $\mu_{X_M} \ge \mu_{X_{R'}}$ のとき, X_M におい て, 値の最も大きいデータを削除することで $\sigma_{X_{R'}}$ に 近づける.6. へ戻る.

仮説2が採択されたとき,count = count + 1とする.

8. count = 1 のとき, 4. へ戻る. count = 2 のとき, 更新を終了.

補正後のデータを基に,シミュレーション上での移動量 モデル ((5) 式) を求める.

$$\begin{aligned}
x_M &= N(\mu_{X_M}, \sigma_{X_M}^2) \\
y_M &= N(\mu_{Y_M}, \sigma_{Y_M}^2) \\
\theta_M &= N(\mu_{\theta_M}, \sigma_{\theta_M}^2)
\end{aligned} (5)$$

4 実験設定

4.1 タスクと適応度関数

ロボットは,より早くゴールに到達することを目的とする. ロボットの初期位置は環境(Fig.2(a))の左下,初期姿勢は 上向きと右向きの2通りとする.ゴールの位置は右上とす る.実環境での評価は,ロボットの各初期姿勢において, それぞれ1試行ずつ計2試行行う.1試行はロボットが目 標物に到達するか,40ステップで終了とする.このとき, 1ステップは約1秒と設定する.なお,ロボットのゴール 判定は搭載されている全方位カメラを用いて行う.

4.2 ロボット設定

4.2.1 センサ入力

3.2 節で述べたように, ロボットは全方位カメラを用い て目標物の位置を以下のように検出する.

- 1. 全方位カメラの画像 (Fig. 3(a)) に対し,2値化,ラベ リング,重心算出の画像処理を行い,目標物と床面が 接しているエッジ部分を検出する.目標物は赤色の円 筒形をしているため,赤色の物体を抽出するようにし きい値を設定する.2値化処理した画像を Fig. 3(b) に示す.
- 2. 検出されたエッジ部分の位置に対応するセルから,半 径方向に沿って外側のセル全てに対して入力値1を 与える.

4.2.2 制御器

ロボットの制御器として,出力関数にシグモイド関数を 用いた人工神経回路網(ANN)を用いる.中間層と出力層 のニューロンは完全相互結合(非対称)しており,自己結 合をもつ.このとき,112に分割されたセル(3.2節参照) を入力値として扱うため,入力ノード数を112とする.ま た,左右のモータコマンドを決定するために出力ノード数 を2とし,中間ノード数を1に設定する.

4.2.3 モータ出力

ロボットの左右の DC モーターは, PWM 制御によりそ れぞれ 9 段階 (前進方向) の速度制御 (デューティ比: $D \in \{60, 65, \cdots, 100\}$ [%]) が可能である.速度入力値であるデ ューティ比は ANN の出力値を 9 段階に割り当てることで 決定される.従って,左右モーター速度の組み合わせによ リ 9 × 9 = 81 通りの行動を取り得る.この 81 通りのモー タ出力に対して,モデル更新を行う.

4.3 進化計算設定

進化手法として拡張型 GA の 1 つであり ANN の進化に対し 有効性が確かめられている Operon GA(OGA)[12, 13, 14] を用いる.OGA の個体表現は, ANN の結合係数を変数と し,10 ビットのバイナリコーディングを行う.ここで,入 カノード数を N_s 個,中間ノード数を N_h 個,出力ノードを N_o 個としたとき,結合係数の総数 $(N_h + N_o)(N_s + N_h + N_o)$ 個となり,遺伝子長は $L = 10(N_h + N_o)(N_s + N_h + N_o)$ ビットとなる.前節で述べたように $N_s = 112, N_h =$ $1, N_o = 2$ と設定するので L = 3450 となる.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 [13] 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.また, 点突然変異率を1/Lとする.各遺伝子座にお ける値のリスト長は6とする.選択法としてトーナメント 選択(トーナメントサイズ: 6)を用い, エリート戦略を適 用する.個体数を50,世代数を1000とし.適応度関数は 次のように設定する.

$$Fitness = \frac{1}{NumTrials} \sum_{k=1}^{NumTrials} (1 - \frac{Step}{MaxStep}) \quad (6)$$

ここで, NumTrials は試行回数 (2) であり, MaxStep は 40 である.そして, 2 節で提案した更新法に従い, 50 世 代ごとに前世代の適応度上位 2 個体について,初期姿勢 上向き1 試行・右向き1 試行の2 試行のみ実環境で評価 する.実環境での評価の間,ロボットの移動量データを取 得することで,シミュレーションのモデル更新を行う.残 りの個体および他の世代 ($g \neq 50n, n = 0, 1, 2, \cdots, 20$) で はシミュレーションのみで評価する.なお,初期世代のシ ミュレーションでは,実験前にあらかじめ実環境で取得し た各モータへの速度入力に対する移動量データ 15 個をも とにモデルを構成する.

5 実験結果

世代数に対する適応度の推移を Fig. 6 に示す.最大適応度 および平均値が初期世代から大きく上昇し,100 世代辺り から最終世代まで安定した適応度を示していることが確認 できる.次に,各世代に対するデータ補正回数を Fig.7 に 示す.450 世代まではデータ補正がなく,500 世代で急激 に増加し.その後は少しずつ補正を行っていることが確認 できる.

最終世代で得られた最良個体の実環境での振る舞いを初 期姿勢右向きの場合について Fig. 9 に示す.ロボットは, 対角線上の目標物に対し,左に動き過ぎては右へ,右に動 き過ぎては左へと進行方向を変え,左右のモータ出力をう まく調節しながら目標物に到達している (Fig. 9).初期姿 勢上向きの場合も,右向き姿勢同様にモータ出力を調節し ながら目標物に到達する振舞いを示している (Fig. 10).実 環境でタスクを達成する振舞いを示したことから,実環境 を反映したシミュレーションモデルの構築ができたといえ る.また,実験時間は約3時間で終了したことから,実環 境のみで評価を行う場合と比べて大幅に短縮できている.



Fig. 6: Maximum and average fitness at each generation



Fig. 7: Number of data corrections at each generation



Fig. 8: Fitness at each trial on the best evolved robot with and without update

6 考察

提案手法の有効性を検証するため、シミュレーションのみ で進化させた個体との比較実験を行う.まず、オフライン 進化として、4節で述べた初期世代のシミュレーション用 の移動量データを用い、モデル更新を行わず、シミュレー ションのみで1000世代まで進化させる.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 を実環境の移動ロボットに制御器として搭載し、評価を行 う.両個体ともに、ロボットの初期姿勢を右向き1回、上 向き1回を1試行とし、合計40試行行う.各試行に対す る適応度をFig.8に示す.提案手法を用いた場合は、全 試行で安定して良い適応度を示したのに対し、シミュレー ションのみの場合は、試行によって安定して良い適応度は 得られなかった.

次にタスク達成率 (ゴール到達率)を Table1 に示す.モ デル更新で得られた最良個体はシミュレーションのみで進 化させた最良個体よりも,タスク達成率が良い結果が得 られた.さらに,モデル更新で得られた最良個体は,右向 き・上向きともに100%のタスク達成率という結果になっ た.以上の結果より,提案手法の有効性が確認された.

7 おわりに

本稿では,移動ロボットのゴール到達問題をタスクとし, 実環境のデータを用いてシミュレーションモデルを構成・ 更新することで,実環境とシミュレーションにおけるモデ

Table 1: Success rate(%) for each method

Initial orientation	Rightward	Upward	Average
Update method	100	100	100
Simulation only	95	60	75

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確認 したことから、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保証するために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. 今後は移動ロボットに関する他の問題[15]や、本稿とは異 なる確率モデルをもつロボットの制御問題[16]において、 本稿で提案した手法の有効性を検証していきたい.

参考文献

- R. Pfeifer and C. Scheier: Understanding Intelligence, MIT Press, 1999.
- [2] S. Nolfi and D. 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, In Proceedings of Congress on Evolutionary Computation, pp. 335-342, 1999.
- [4] N. Jakobi: Half-baked Ad-hoc and Noisy: Minimal Simulation for Evolutionary Robotics, In Proceedings of the Fourth European Conference on Artificial Life, pp. 348-357, 1997.
- [5] 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, pp. 149-154, 2007.
- [6] R. A. Brooks: Artificial Life and Real Robots, In Proceedings of the First European Conference on Artificial Life, pp. 3-10, 1992.
- [7] O. Miglino, H. H. Lund and D. Nolfi: Evolving Mobile Robots in Simulated and Real Environments, Artificial Life 2, pp. 417-434, 1995.
- [8] D. Keymeulen, M. Iwata, K. Konaka, R. Suzuki, Y. Kuniyoshi and T. Higuchi, Off-line Mode-free and On-line Model-based Evolution for Tracking Navigation Using Evolvable Hardware, In Proceedings of the First European Workshop on Evolutionary Robotics, Springer-Verlag, Paris, 1998.
- [9] 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.



Fig. 9: Behavior of the best evolved robot in the real environment from the initial rightward orientation

- [10] 森,家田,松井: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.
- [11] 小森谷,大山,谷: 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.
- [12] 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, pp. 1461-1469, 1996.
- [13] 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, pp. 187-195, 2004.
- [14] 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, pp. 31-39 2008.



Fig. 10: Maximum and average fitness at each generation from the initial upward orientation

- [15] 片田,西村:進化型マルチエージェントによる協調荷 押し行動獲得に関する一考察;第52回システム制御 情報学会研究発表講演会論文集,pp. 469-470 2008.
- [16] S. Thrun, W. Burgard and D. Fox: Probabilistic Robotics, MIT Press, 2005.