深層強化学習と Deep Neuroevolution による ロボティックスワームの群れ行動生成と解析*

森本 大智[†]・平賀 元彰[†]・大倉 和博[†]・松村 嘉之[‡]

Generating and Analyzing Collective Behavior in a Robotic Swarm by the Use of Deep Reinforcement Learning and Deep Neuroevolution*

Daichi MORIMOTO[†], Motoaki HIRAGA[†], Kazuhiro OHKURA[†] and Yoshiyuki MATSUMURA[‡]

This study proposes a method to apply deep neural networks to controllers of robotic swarms. In a typical approach to design controllers, the designer has to define the features extracted from sensory inputs in advance. By applying deep neural networks with convolution layers, it can automatically extract features from sensory inputs. We applied two methods to train the deep neural networks, i.e., deep reinforcement learning and deep neuroevolution. The controllers were tested in a path-formation task using computer simulations. Compared with deep reinforcement learning, deep neuroevolution was able to generate collective behavior even in sparse reward settings.

1. はじめに

Swarm Robotics (SR) は多数のロボットからなる集団に、ロボット間あるいはロボットと環境間の局所的な相互作用によって所望の群れ行動の生成を目指す研究分野である [1]. SR におけるロボットシステムは一部のロボットの故障に対する頑健性、異なる環境やタスクに対応する柔軟性、異なるロボット台数で動作する拡張性が期待される [2]. SR では所望の群れ行動を生成するためのロボットの制御器をどのように設計するかについて研究が行われている. Behavior-based design method [2] では設計者自身が各ロボットの行動則を定義し繰り返し改良する. 個々のタスクに応じてロボットの行動を試行錯誤するため、効率的なタスク達成が期待されるが、異なる条件に対して制御器の再設計が必要となる. これに対し、Automatic design method [2] では汎用的な計算手法によって制御器を自動的に設計する.

Automatic design method では強化学習や進化計算といった手法が用いられる. とくにロボットの制御器にニューラルネットワークを用い、これを進化計算によって設計する進化ロボティクスアプローチ [3] が多く試みられている. 先行研究では集合行動 [4], フロッキング [5], 協調搬送 [6,7] などの群れ行動が生成されている. このアプローチでは、タスクの達成度やロボットの振舞いをもとに設計した適応度関数を用いて、各ロボットの行動則を明示的に定義することなく群れ行動の生成が可能となる. しかし、ロボットに搭載するセンサとセンサ情報からどのような特徴量を抽出し、制御器への入力とするかを事前に決定する必要がある、ロボティックスワームの性能を大きく左右する要因となる. そのため、これらは設計者が試行錯誤的に決定することが多く、Automatic design method を用いるうえでの難点の一つである.

近年、深層ニューラルネットワーク (Deep Neural Network, DNN) が注目されている. とくに畳み込み層をもつ Deep Convolutional Neural Network (DCNN) は学習によって画像の特徴量を自動的に抽出可能であり、自然画像認識において従来手法を上回る性能を示す [8]. DCNN をロボティックスワームの制御器に用いることで、事前の特徴量の設計を行わずに画像を直接入力とした群れ行動の生成が期待される. DNN によってエージェントの行動を生成する手法として、深層強化学習 (Deep Reinforcement Learning, DRL) [9] が知られている. DRL はさまざまな問題に適用され、良好な成績を

* 原稿受付 2019 年 7 月 12 日

* 第 63 回システム制御情報学会研究発表講演会にて発表 (2019 年 5 月)

[†] 広島大学 大学院 工学研究科 Graduate School of Engineering, Hiroshima University; 1-4-1, Kagamiyama, Higashi Hiroshima, Hiroshima 739-8524, JAPAN

[‡] 信州大学 大学院 総合工学系研究科 Graduate School of Science and Technology, Shinshu University; 3-15-1, Tokida, Ueda, Nagano 386-0018, JAPAN

Key Words: swarm robotics, deep reinforcement learning, deep Q-network, deep neuroevolution.

記録している [9,10]. DRL をロボティックスワームに適用する試みも存在し, 追跡行動 [11] や二点間の往復行動 [12] が生成されている.

一般的に DRL における DNN の学習には確率的勾配降下法が用いられる. これは単点探索型の最急降下法であるため, 多峰性の目的関数における局所解や鞍点に対応することが難しいと考えられる. 目的関数の形状はエージェントに与えられる報酬に対応するため, エージェントの性能は報酬設定に大きく左右される. DRL ではスパースな報酬設定下での学習が難しいことが知られており [9], DRL をロボティックスワームに適用した場合においても複雑な報酬設定が必要となる [12].

これに対して近年, 所望の DNN を進化計算によって獲得する Deep Neuroevolution (DNE) が試みられている [13,14]. DNE は強化学習のベンチマークに適用されており, 一部の問題において DRL に匹敵する性能を記録している [13,14]. DNE は個体群ベースの勾配フリーな探索を行うため, 多峰性の目的関数や報酬設定の変化に対して頑健であることが期待される. DNE をロボティックスワームに適用することで, Automatic design method の汎用性の向上と適用範囲の拡大が期待される.

本研究では DNE によるロボティックスワームの群れ行動生成を行う. SR の進化ロボティクスアプローチにおいて, 事前の特徴量の設計なしに画像を直接入力とした群れ行動の生成を試みる. また, DRL をロボティックスワームに適用した場合との比較を行う. 異なる報酬条件下における DRL と DNE の性能を比較する. また, 得られた振舞いについて詳細な比較を行うために, 振舞いの解析を試みる. 群れ行動の解析手法についてはまだ定石的手法はなく, 群れ行動の定性的, 定量的な評価は難しい. 本論文では比較的簡単に取得可能なロボットの速度ベクトルと局所密度に着目し, 先行研究 [15] を参考にした群れ行動の解析を行う.

本論文は以下のように構成される. 第2章では制御器の設計手法である Deep Q-Network と Deep Neuroevolution について概説する. 第3章では計算機実験とその結果について説明する. 第4章では得られた群れ行動について解析を行う. 第5章では得られた結果について考察を行う. 第6章では本論文の結論と今後の展望を示す.

2. 設計手法

2.1 Deep Q-Network

DQN [9] は DRL の代表的なアルゴリズムであり, 強化学習における Q-Learning と DNN を組み合わせた手法である. 一般的な Q-Learning では行動価値関数 $Q(s, a)$ を近似するテーブルや関数を更新することで学習を進行するが, DQN では DNN によって $Q(s, a)$ を近似する. DNN を用いることで膨大な状態数に対応する $Q(s, a)$ を得ることができる. DQN ではエージェントが獲得した経験に基づいて DNN の学習を進行する. 経験 (s, a, s', r)

は, エージェントが観測した環境の状態 s , 選択した行動 a , 遷移後の環境 s' , 獲得した報酬 r から成る. この経験を基にタイムステップ i における $Q(s, a)$ についての時間的差分誤差の期待値 $L_i(\theta_i)$ を計算し, DNN の重みを更新する.

$$L_i(\theta_i) = \mathbb{E}[(r + \gamma \max_{a'} Q(s', a', \theta_i) - Q(s, a; \theta_i))^2] \quad (1)$$

ここで θ_i は DNN のパラメータ, γ は割引率を示す.

DQN は Mnih ら [9] によって提案され Atari2600 のゲームにおいて人間に匹敵するスコアを記録したことで注目を集めた. DQN はその後も改良が続けられている. 中でも Nair らは複数のエージェントによって経験データを並列に収集する General reinforcement learning architecture (Gorila) を提案した [16]. Gorila による並列処理はロボティックスワームのような多数のエージェントを用いるシステムと相性が良いことが期待される. よって本研究では Gorila を適用した DQN を用いる.

2.2 Deep Neuroevolution

進化計算によって DNN のパラメータを更新する Deep Neuroevolution (DNE) は多点探索型の勾配フリーな手法であるため, 多峰性の目的関数に対して頑健な DNN の最適化が期待される. DNE の先行研究として文献 [13,14] がある. Salimans らは三層の畳み込み層と二層の全結合層からなる DNN を Natural Evolution Strategy をベースにした手法によって学習させた [13]. また Such らは同様の構造をもつ DNN を交叉を用いない Genetic Algorithm によって学習させた [14]. これらの先行研究では DNN の構造を固定し, DNN のパラメータのみを更新する手法を提案している. 本研究においても同様の手法を用いる.

3. 計算機実験

DRL および DNE によるロボティックスワームの群れ行動生成実験を行う. シミュレーション環境は Unity3D ゲームエンジン [17] を用いて構築している.

3.1 二点間報復タスク

ロボティックスワームのベンチマークとして二点間往復タスクを用いる. このタスクでは環境内の二つのターゲットエリア間を往復することを目的とする. 実験環境を Fig. 1 に示す. 環境の両端にシアンおよび黄色のランドマークを配置する. ランドマークは高さ 2.0m, 横幅 7.5m であり, ランドマークの中心から半径 3.75m の範囲がターゲットエリアである. 各ロボットは目標ターゲットエリアへ到達後, もう一方のターゲットエリアを目指す. ロボット台数は 36 台とする. また, ロボットの視界を制限し, タスクの難易度をあげるため環境内に霧を発生させる. 霧は周囲の壁の高さまで発生し, ロボットは約 8.0m 先まで環境を視認できる.

3.2 ロボット

実験に用いるロボットを Fig. 2(a) に示す。ロボットは左右の車輪の回転数を独立に制御することで駆動する差動駆動型移動ロボットである。本実験ではロボットの行動を停止、直進、左回転、右回転の4種類とする。また、ロボットの最高移動速度は1.0m/sとする。ロボットはカメラ、IR センサ、LED を搭載している。カメラは 128×128 pixels の RGB 画像を生成する。Fig. 2(b) にカメラから得られる画像の例を示す。IR センサは測定範囲内に存在する物体までの距離を検出する。これらセンサの測定範囲を Fig. 2(c) に示す。ロボット上部の LED は後方のみ点灯する。ロボットが黄色のランドマークを目指す場合は赤色、シアンの場合は青色に点灯する。

3.3 制御器

ロボットの制御器として、先行研究 [12] を参考にした DNN を用いる。制御器の構造を Fig. 3 に示す。制御器は入力層、4層の畳み込み層、全結合層、出力層からなる。制御器への入力にはカメラ画像、IR センサの情報、目標ランドマーク情報であり、それぞれ過去4タイムステップ分の情報が入力される。入力画像を畳み込み層で処理することで得られる特徴マップと他のセンサ情報が全結合層に入力され、ロボットの各行動に対応した $Q_1 \sim Q_4$ を出力する。畳み込み層におけるフィルタにはサイズが 4×4 のものを64種類用意する。畳み込み時のストライドは2とし、これらフィルタに関する設定は4層の畳み込み層で共通とする。また、各畳み込み層の直前に Batch Normalization 層 [18] を加える。

3.4 実験設定

ここでは DQN と DNE で共通の設定を述べる。実験は1エピソードを1000タイムステップとし、500エピソードを最終エピソードとする。1タイムステップは0.2秒に相当し、ロボットは毎タイムステップ行動を選択する。制御器は24個とし、DQN ではこれらの制御器を独立に学習させる。DNE では進化アルゴリズムによって制御器のパラメータを更新する。またロボットに対する報酬は以下のものを用いる。

$$r_{d,i,t} = 1.0 \times (d_{i,t-1} - d_{i,t}) \quad (2)$$

$$r_{e,i,t} = 5.0 \quad (3)$$

$r_{d,i,t}$ はタイムステップ t におけるロボット i がターゲットへ近づく動作への報酬を示す。 $r_{d,i,t}$ はロボットのカメラにターゲットが映る場合のみ有効となる。 $d_{i,t}$ はタイムステップ t におけるロボット i とターゲットの重心との距離を表す。 $r_{e,i,t}$ はロボットが目標ターゲットエリアに到達した際に与えられる。これらの報酬を組み合わせた二つの条件下で実験を行う。一つ目の条件では $r_{d,i,t}$ 、 $r_{e,i,t}$ の和で報酬を決定する。 $r_{d,i,t}$ は与えられる頻度が比較的高いため、DQN がタスクを達成しやすい設定と

Fig. 1 Experimental field

Fig. 2 Robot settings

Fig. 3 Robot controller

なる。タイムステップ t におけるロボット i への報酬 $R_{i,t}$ は以下の式で与えられ、これを「報酬設定 (i)」とする。

$$R_{i,t} = r_{d,i,t} + r_{e,i,t} \quad (4)$$

二つ目の条件では $r_{e,i,t}$ のみで報酬を決定する。 $r_{e,i,t}$ は与えられる頻度が比較的低いため、DQN がタスクを達成しにくい設定となる。報酬 $R_{i,t}$ は以下の式で与えられ、これを「報酬設定 (ii)」とする。

$$R_{i,t} = r_{e,i,t} \quad (5)$$

また、各ロボットは $Q_1 \sim Q_4$ に対し ϵ -greedy 法により行動を選択する。replay memory size のタイムステップが経過するまでは $\epsilon = 1.0$ とし、ロボットは完全にランダムに行動を選択する。また、この期間中は制御器の学習を行わない。replay memory size 以降では $\epsilon = 0.1$ とする。以上の実験を DQN、DNE 共に3試行ずつ行う。

3.5 制御器の学習

3.5.1 Deep Q-Network

DQN のパラメータを Table 1 に示す。DQN では各ロボットの経験を Experience replay によって再生することで制御器の学習を進行する。ロボットの経験は、制御器への入力となる行動選択以前のセンサ情報、選択した行動、得られた報酬、行動選択後のセンサ情報、1エピソードの終了判定を示すバイナリ値からなる。一つの経

Table 1 Parameters of DQN

time steps/episode	1000
total episodes	500
policy	ϵ -greedy $\epsilon = \begin{cases} 1.0 & \text{(initial exploration)} \\ 0.1 & \text{(final exploration)} \end{cases}$
optimizer	RMSPropGraves
discount factor	$\gamma = 0.95$
minibatch size	32
replay memory size	36×400

Table 2 Parameters of DNE

time steps/episode	1000
total episodes	500
policy	ϵ -greedy $\epsilon = \begin{cases} 1.0 & \text{(initial exploration)} \\ 0.1 & \text{(final exploration)} \end{cases}$
population size	24
tournament size	2
mutation power	0.02
elite size	1

験セットは過去 4 タイムステップ分の情報からなる。各ロボットは replay memory size 分の経験セットを保有する。また、Gorila アーキテクチャ[16]を導入し、全ロボットの経験を集約して制御器の学習を行う。optimizer として RMSPropGraves[19] を用いる。RMSPropGraves のパラメータは深層学習ライブラリである Chainer[20] のデフォルト値を用いる。

3.5.2 Deep Neuroevolution

DNE のパラメータを Table 2 に示す。DNE では全ロボットが 1 エピソードで獲得した報酬の総和を制御器の適応度として、各エピソードの終了時に制御器の更新を行う。適応度 F は以下の式で表される。

$$F = \sum_t \sum_i R_{i,t} \quad (6)$$

制御器の進化アルゴリズムとして先行研究[14]を参考にした遺伝的アルゴリズムを用いる。このアルゴリズムでは選択した親個体に対し突然変異操作を行うことで子個体を生成する。突然変異操作は以下の式で表される。

$$\theta^n \leftarrow \theta^{n-1} + \sigma \epsilon \quad (7)$$

ここで θ は DNN のパラメータベクトル、 n は世代数、 ϵ は θ と同じ要素数をもつノイズベクトル、 σ はスケールパラメータである。 ϵ の各要素は標準正規分布よりサンプリングを行う。 θ は結合荷重値と Batch Normalization 層におけるアフィン変換のパラメータとする。

親個体の選択は各エピソードの終了時にトーナメント

Fig. 4 Experimental result on reward setting (i)

選択によって行われる。また、エリート選択によりエピソード中で最高の適応度を記録した制御器を突然変異操作を行うことなく次エピソードに引き継ぐ。

3.6 結果

3.6.1 報酬設定 (i) に基づく結果

Fig. 4 に実験 3 試行分の到達回数 の平均値と最大値の推移を示す。グラフは 72 個の制御器のデータを示し、平均値は標準偏差とともに示す。DRL のデータを破線、DNE のデータを実線で示す。DRL では 50 エピソード付近まで到達回数が急激に上昇し、その後は比較緩やかに上昇している。DNE では初期から到達回数が比較的緩やかに上昇している。この傾向は平均値、最大値に同様に見られ、制御器の学習、進化時ではおおむね DRL が優れた性能を示す。また、DRL および DNE によって獲得した制御器を用いてロボティックスワームの行動生成を行った様子を Fig. 5, Fig. 6 に示す。DRL の場合、ロボットは二つのランドマーク間の最短経路付近を移動する振舞いを示す。DNE の場合、ロボットは環境内を輪を描くように移動する。DRL と DNE で異なる傾向をもつ振舞いが観測された。

3.6.2 報酬設定 (ii) に基づく結果

Fig. 7 に実験結果を示す。グラフの形式は報酬設定 (i) の場合と同様である。Fig. 7 より、DRL では報酬設定 (i) と比較して性能が低下している。一方、DNE では報酬設定 (i) の場合とほぼ同様の到達回数の推移を示す。DNE が、DRL の性能が低下するスパースな報酬設定下においてタスクを達成していることがわかる。また、報酬設定 (i) の場合と同様に、両手法によって獲得した制御器を用いてロボティックスワームの行動生成を行った様子を Fig. 8, Fig. 9 に示す。DRL の場合、多くのロボットが壁面付近に停止する。DNE の場合、ロボットは報酬設定 (i) の場合と同様に環境内を輪を描くように移動する。DRL では報酬設定の変化に伴い性能が低下した。一方、DNE では報酬設定の変化に対しロボットが同様の振舞いと性能を示した。

4. 群れ行動の解析

DRL, DNE によって生成された群れ行動について、より詳細な比較のために解析を行う。群れ行動の解析に

Fig. 5 Snapshot of behavior generated by DQN

Fig. 6 Snapshot of behavior generated by DNE

Fig. 7 Experimental result on reward setting (ii)

はいまだ定石的手法がなく、群れの定性的、定量的評価は難しい。ここでは比較的簡単に取得可能なロボットの速度、密度を用いた解析を行う。モデル化された群れの解析を行った先行研究 [15] を参考に、二つの指標として速度ベクトルの総和 $\psi(t)$ 、局所密度の総和 $\rho(t)$ を導入する。

4.1 制御器の選定

報酬設定 (ii) の DRL では多数のロボットが壁付近で停止しており、指標による評価が困難である。よって DRL, DNE でロボットが十分に移動している報酬設定 (i) の制御器を用いる。解析には DRL, DNE の各試行においてテスト時の性能が最も優れた制御器を用いる。制御器の学習、進化時において 50 エピソードごとに最高の到達回数を記録した制御器を保存する。各実験において $500/50 \times 24 = 240$ 個の制御器が保存される。この中から到達回数が多い順に 10 個ずつ制御器を選択し、ロボットの行動生成を行う。行動生成時は 1 エピソードを 3000 ステップとし、 $\epsilon = 0.05$ とする。行動生成を 10 エピソード分行い、到達回数の平均値が最も高い制御器を各試行における最良の制御器として解析に用いる。

4.2 速度ベクトルの総和 $\psi(t)$

第一の指標である $\psi(t)$ は各ロボットの速度ベクトルの総和であり、以下の式で表される。

$$\psi(t) = \sum_{i=1}^N \mathbf{v}_i(t) \quad (8)$$

ここで $\mathbf{v}_i(t)$ はロボット i の時刻 t における速度ベクトルであり、 N はロボット台数である。 $\psi(t)$ は各制御器について 3000 タイムステップ分算出される。速度ベクトルは各ロボットの座標情報から算出する。そのため最終タイムステップにおける速度は 0 となる。

また、 $\psi(t)$ は各タイムステップにおいて計算されるためロボットの短期的な行動に反応する。ロボットの長期的な行動戦略に対する $\psi(t)$ の変動を確認するためローパスフィルタを適用する。シミュレーション上においてロボットの位置の記録は 0.2 秒ごとに行われ、これが $\psi(t)$ のサンプリング周期となる。実験時の各試行の最良の制御器によって生成された振舞いにおいて、カットオフ波長を $f_c = 0.01 [\text{Hz}]$ としてローパスフィルタを適用した $\psi(t)$ を Fig. 10, Fig. 11 に示す。破線が $\psi(t)$ の x 方向成分 ψ_x を示し、実線が y 方向成分 ψ_y を示す。

実験環境を俯瞰した場合、DRL ではロボットがおもに左右に移動するため Fig. 10 において ψ_y の値が 0 に近くなる。Fig. 10 の DRL では第 1, 第 2 試行において 1500 タイムステップ付近までに正弦波に近い波形が見られる。これは左に移動するロボットが多い状態と右に移動するロボットが多い状態が周期的に遷移することを示す。しかし 1500 タイムステップ以降ではこのような波形は見られない。また、第 3 試行では 2000 タイムステップ以降に比較的正弦波に近い波形が見られる。DRL では正弦波に近い波形が比較的短期間維持されている。

DNE ではロボットが環状の経路を形成する。ロボッ

Fig. 8 Snapshot of behavior generated by DRL in reward setting (ii)

Fig. 9 Snapshot of behavior generated by DNE in reward setting (ii)

Fig. 10 ψ passed through low pass filter (DRL) Fig. 11 ψ passed through low pass filter (DNE)

トが経路上に均等に分布し、それを維持するよう移動すれば ψ_x , ψ_y の値は 0 に近くなる。 ψ_x , ψ_y の値が 0 から離れるほど、経路上でロボットの密度が高くなる箇所が存在することを示す。 Fig. 11 では ψ_x , ψ_y が正弦波に近い波形を示す。 よって、経路上でロボットの密度が高くなる箇所が存在し、これが経路上を一定の速度で伝

Fig. 12 Summation of local robot densities ρ

播していると考えられる。 また、 DRL に対して、 ψ_x , ψ_y の周期性が比較的長期間維持される傾向が見られる。

4.3 局所的密度の総和 $\rho(t)$

第二の指標である $\rho(t)$ は各ロボットについての局所的密度の総和である。 ここで局所的密度を「あるロボットを中心とした半径 R の範囲内に存在するほかのロボットの数」として定義する。 $\rho(t)$ は以下の式で表される。

$$\rho(t) = \sum_i^N \sum_{j \neq i}^N c_{i,j}(t) \quad (9)$$

$$c_{i,j}(t) = \begin{cases} 1 & \text{if } (x_i - x_j)^2 + (y_i - y_j)^2 \leq R^2 \\ 0 & \text{otherwise} \end{cases}$$

i, j はロボットの番号、 N はロボット台数、 x_i , y_i はロボット i の位置座標を示す。 $\rho(t)$ は群れ全体の密集度を示し、値が大きいほどロボットが密集していることを示す。 $R=3$ として $\rho(t)$ を算出した結果を Fig. 12 に示す。 DNE では $\rho(t)$ の値が低く、変動も小さい。 $\rho(t)$ の値は

Fig. 13 Box plot for number of arrivals

常に 100 程度であり、これは平均して一台のロボットの周囲 3m に 2.8 台のロボットが存在することを示す。よって DNE ではロボット間の距離がある程度保たれており、混雑が発生しにくいと考えられる。DRL ではロボットがより狭い範囲に集まるため $\rho(t)$ の値が高くなる。また $\rho(t)$ の変動が大きく、ピーク時には 300 程度の値を示す。これは平均して一台のロボットの周囲 3m に 8.3 台のロボットが存在することを示す。よって、狭い範囲にロボットが密集し、混雑が発生しやすいと考えられる。

5. 考察

制御器の選定時における到達回数の比較を Fig. 13 の箱ひげ図に示す。各箱はそれぞれ 30 個の制御器を用いて得られた 300 個の到達回数のデータから構成される。中央値、四分位範囲で比較すると、DNE がおおむね優れた性能を示す。制御器の選定時は 1 エピソードが 3000 タイムステップであるため、到達回数の期待値は学習時の最大値の 3 倍程度となる。DNE では中央値が期待値に近い値を示す。一方、DRL では期待値の約 45% 程度であり、学習時と比較して性能が低下している。この原因として DRL によって生成される振舞いの性質が考えられる。解析の結果から DRL による群れ行動では、行動がもつ周期性が維持されにくく、ロボットの混雑が発生しやすいと考えられる。とくにランドマーク付近でロボット同士の接触による渋滞が発生し、性能が低下する傾向が見られる。DNE では行動の周期性が維持されやすく混雑が起きにくいと考えられる。よって DNE で生成された行動は比較的長期の行動生成に有利であると考えられる。

他の原因として DRL のアルゴリズムが考えられる。学習時において制御器は各エピソードの終了時点のものが保存される。DNE では 1 エピソード中で同一の結合荷重値を用いるのに対し、DRL では毎タイムステップ結合荷重値が更新される。よってエピソード終了時の性能がエピソード中と乖離する可能性が考えられる。

DRL と DNE は同一の報酬条件下で異なる傾向の振舞いを生成した。DRL では各ロボットが収集した経験を用いて制御器の学習を行う。報酬設定 (i) ではロボットが直進し、ターゲットエリアに接近、到達することで報酬

が得られる。この経験を基にした学習を行うことで各ロボットがターゲットエリアを目指す振舞いが得られたと考えられる。また、報酬設定 (ii) では報酬がターゲットエリアへの到達のみであるため、ターゲットエリアに到達した後の回転、直進といった行動が報酬と結びつかず、タスクを達成できなかったと考えられる。DNE では各エピソードの終了時に各制御器の適応度を用いて制御器の更新を行う。実験の初期に生存しやすい制御器は直進行動を出力し、比較的高い適応度を得やすい制御器だと考えられる。つぎにその中から、適度に回転行動を出力し衝突を回避する制御器が生存すると予想される。このような過程を経て、直進と回転を組み合わせた輪を作る振舞いが得られたと考えられる。

6. おわりに

本論文では DRL および DNE によるロボティックスワームの群れ行動の生成と解析を行った。DNE により、SR の進化ロボティクスアプローチにおいて、事前の特徴量の設計なしに高次元データである画像を直接入力とした二点間往復タスクの達成が可能であることを示した。また DNE は DRL によるタスク達成が困難であるスパースな報酬設定下においてタスクの達成が可能であることを示した。加えて、DRL と DNE が、同様の報酬設定下で異なる振舞いを生成することを示し、群れ行動の解析を行った。解析の結果、DNE によって生成された振舞いは長期の行動生成に有利である可能性を示した。

今後、より多様な報酬条件下での実験を行う。とくに、ロボット間の協調を明示的に示す報酬を用いた際にどのような振舞いが生成されるか実験を行う。また、複雑な景観やノイズが生じやすい環境などを導入することでタスクの難易度を上げ、将来的には実環境での実験を行う。

参考文献

- [1] E. Şahin: Swarm robotics - from sources of inspiration to domains of application; *Swarm Robotics* (E. Şahin, W. M. Spears, Eds.), Springer, pp. 10–20 (2004)
- [2] M. Brambilla, E. Ferrante, M. Birattari and M. Dorigo: Swarm robotics - a review from the swarm engineering perspective; *Swarm Intelligence*, Vol. 7, No. 1, pp. 1–41 (2013)
- [3] S. Nolfi and D. Floreano: *Evolutionary Robotics - The Biology, Intelligence, and Technology of Self-organizing Machines*, MIT press (2000)
- [4] M. Dorigo, V. Trianni, E. Şahin, R. Groß, T. H. Labella, G. Baldassarre, S. Nolfi, J.-L. Deneubourg, F. Mondada, D. Floreano and L. M. Gambardella: Evolving self-organizing behaviors for a swarm-bot; *Autonomous Robots*, Vol. 17, Nos. 2–3, pp. 223–245 (2004)
- [5] G. Baldassarre, S. Nolfi and D. Parisi: Evolving mo-

- bile robots able to display collective behaviors; *Artificial Life*, Vol. 9, No. 3, pp. 255–267 (2003)
- [6] R. Gross and M. Dorigo: Towards group transport by swarms of robots; *International Journal of Bio-Inspired Computation*, Vol. 1, Nos. 1–2, pp. 1–13 (2009)
- [7] R. Groß and M. Dorigo: Evolution of solitary and group transport behaviors for autonomous robots capable of self-assembling; *Adaptive Behavior*, Vol. 16, No. 5, pp. 285–305 (2008)
- [8] A. Krizhevsky, I. Sutskever and G. E. Hinton: Imagenet classification with deep convolutional neural networks; *Advances in Neural Information Processing Systems*, pp. 1097–1105 (2012)
- [9] V. Mnih, et al.: Human-level control through deep reinforcement learning; *Nature*, Vol. 518, No. 7540, p. 529 (2015)
- [10] S. Gu, E. Holly, T. Lillicrap and S. Levine: Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates; *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, pp. 3389–3396 (2017)
- [11] M. Hüttenrauch, S. Adrian and G. Neumann, et al.: Deep reinforcement learning for swarm systems; *Journal of Machine Learning Research*, Vol. 20, No. 54, pp. 1–31 (2019)
- [12] Y. Wei, X. Nie, M. Hiraga, K. Ohkura and Z. Car: Developing end-to-end control policies for robotic swarms using deep Q-learning; *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 23, No. 5, pp. 920–927 (2019)
- [13] T. Salimans, J. Ho, X. Chen, S. Sidor and I. Sutskever: Evolution strategies as a scalable alternative to reinforcement learning; *arXiv preprint arXiv:1703.03864* (2017)
- [14] F. P. Such, V. Madhavan, E. Conti, J. Lehman, K. O. Stanley and J. Clune: Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning; *arXiv preprint arXiv:1712.06567* (2017)
- [15] C. Huepe and M. Aldana: New tools for characterizing swarming systems: A comparison of minimal models; *Physica A: Statistical Mechanics and its Applications*, Vol. 387, No. 12, pp. 2809–2822 (2008)
- [16] A. Nair, et al.: Massively parallel methods for deep reinforcement learning; *arXiv preprint arXiv:1507.04296* (2015)
- [17] Unity Technologies: Unity - game engine; <https://unity3d.com/>
- [18] S. Ioffe and C. Szegedy: Batch normalization: Ac-

celerating deep network training by reducing internal covariate shift; *arXiv preprint arXiv:1502.03167* (2015)

- [19] A. Graves: Generating sequences with recurrent neural networks; *arXiv preprint arXiv:1308.0850* (2013)
- [20] Preferred Networks: Chainer - A flexible framework for neural networks; <http://chainer.org/>

著者略歴

もりもと だいち (学生会員)

2019年3月広島大学工学部第一類(機械システム工学系)卒業, 同年4月広島大学大学院工学研究科博士課程前期に進学し, 現在に至る。

ひらがもと あき

2016年3月広島大学工学部第一類(機械システム工学系)卒業, 2018年9月広島大学大学院工学研究科博士課程前期修了, 同年10月広島大学大学院工学研究科博士課程後期に進学し, 現在に至る。

おおくら かずひろ (正会員)

1990年3月北海道大学大学院工学研究科情報工学専攻修士課程修了。同年(株)富士通研究所, 1993年神戸大学助手, 1998年英国サセックス大学客員研究員, 2000年神戸大学助教授, 2006年広島大学教授。EC, RL, MAS, SR, 生産システムなどの研究に従事。博士(工学)。日本機械学会, 計測自動制御学会, IEEEなどの会員。

まつむら よしゆき (正会員)

1998年神戸大学工学部機械工学科卒業, 2002年同大学大学院自然科学研究科博士後期課程修了。2000年日本学術振興会特別研究員(DC1)。2003年信州大学繊維学部講師, 英国バーミンガム大学名誉研究員。2009年信州大学繊維学部准教授, 中国蘇州大学特別客座教授。EANN, ER, Cloud, 複雑ネットワークなどの研究に従事。博士(工学)。SICE, 精密工学会, 繊維製品消費科学会などの会員。