

## アニボットの協調行動の獲得

北海道大学 大学院情報科学研究科 ○川上 祥, 鈴木 育男, 山本 雅人, 古川 正志

### 要旨

これまでに、3次元の仮想空間内において物理法則に従った挙動を自律的に獲得するソフトウェア・ロボットとしてアニボットを提案し、そのモデリングとシミュレーションを行うためのツールの開発を行ってきた。本論文では、アニボットの協調行動の獲得を目的とする。そのためにアニボットをマルチエージェント問題として定式化した。更に、2台のロボットが協力して荷物を運ぶ行動獲得を行い、その結果を報告する。

### 1. はじめに

これまでに、3次元の仮想空間内において物理法則に従った挙動を自律的に獲得するソフトウェア・ロボットとしてアニボットを提案し、そのモデリングとシミュレーションを行うためのツールの開発を行ってきた。

ツールの目的として、生物などのモーションの自動生成による3Dアニメーション技術への応用や、自律ロボットのソフトウェアベースで設計・動作制御がある。また、モデルのコピー機能を実装することで群のモデリングを可能とし、これまでにいくつかの協調行動の獲得実験を行ってきた。

本研究では、アニボットが協力して目的を果たす協調行動の獲得実験を行い、その結果を報告する。

### 2. アニボット

アニボット (Anibot : Animated Robot) は、物理法則に従って自律行動するソフトウェアロボットである。センサ、アクチュエータ、そして意思決定のためのコントローラを持つ。また、質量や材料等の物理特性等も兼ね揃えており、現実の物理法則に従って動作する。なお、シミュレーションにおける物理演算には、NVIDIA社のPhysX物理エンジンを利用する。

アニボットの構成は、球、立方体、任意形状のメッシュ体からなるプリミティブと、プリミティブ間を繋ぐジョイントを組み合わせて成る。プリミティブにはセンサを、ジョイントにはアクチュエータの属性を持たせる事ができ、これらによって行動制御を行う。

### 3. 協調行動

協調行動は、自然界においては鳥や魚の群れの行動や捕食動物の狩猟行動など、動物の群に見られる。また、近年では、センサネットワークや群ロボットなど、工学的なアプローチにも協調行動が用いられている。

協調行動獲得の手法としてBoidやFlock, Schoolなどのルールベースで得る方法と、マルチエージェントの方法がある。本研究では、マルチエージェントの方法を用いて協調行動の獲得を行う。

### 4. マルチエージェントによる協調行動獲得

#### 4.1. 数理モデル

アニボットをマルチエージェント問題 (MAP) として以下のように定式化する。

$$MAP = \langle A, E, C \rangle (1)$$

ここで、 $A$ はエージェント (アニボット)、 $E$ は環境、 $C$ は評価を表す。環境が変化しない場合、 $E$ は固定であるので、エージェント $A$ によって評価 $C(E, A)$ は決定する。

獲得したい行動に合わせて評価 $C(E, A)$ を設定し、それを最大化するようなエージェント $A$ を決定することによって、協調行動の獲得ができる。

#### 4.2. アニボットの表現

アニボットをそれぞれ $A_1, A_2$ のように表すと、 $A$ は以下の式で表せる。 $N$ はアニボットの数とする。

$$A = \{A_i : i = 1, 2, \dots, N\} (2)$$

また、各アニボットを

$$A_i = \langle I_i, O_i, S_i, F_i, G_i \rangle (3)$$

と定義する。ここで  $I_i, O_i, S_i, F_i, G_i$  は、それぞれアニボット $i$ の入力、出力、状態、状態遷移関数、出力関数である。以下にこの要素を述べる。

##### 4.2.1. 入力 $I_i$

入力  $I_i$  は、「センサの入力  $E_i$ 」及び「アクチュエータの状態  $C_i$ 」が与えられ、 $I_i = \{E_i, C_i\}$  と表す。

##### 4.2.2. 出力 $O_i$

出力  $O_i$  には、アクチュエータへの出力 (トルク  $T_d$  や目標角速度  $\omega_d$ ) で与えられ、 $O_i = \{T_d, \omega_d\}$  と表す。

##### 4.2.3. 状態 $S_i$

時刻  $t$  におけるアニボット  $A_i$  の状態を以下の式で表す

$$S_i(t) = \langle P_i(t), V_i(t), \Theta_i(t), \Omega_i(t) \rangle (4)$$

各要素はアニボットの位置、速度、角度、角速度のベクトルとなる。

##### 4.2.4. 状態遷移関数 $F_i$

状態遷移関数は、入力と状態から次のステップの状態を得る関数である。

$$F_i : \sum_{i=0}^N (I_i \times S_i) \rightarrow S_i (5)$$

ここで、 $N$ はアニボットの数である。 $F_i$ は物理シミュレーションの1ステップの計算であり、PhysXの物理演算によって計算される。

##### 4.2.5. 出力関数 $G_i$

出力関数  $G_i$  はニューラルネットワーク相当する。アニボット  $A_i$  の出力  $O_i$  は、入力  $I_i$  からニューラルネットワークによって計算される。

$$G_i : I_i \rightarrow O_i (6)$$

#### 4.3. コントローラ (ニューラルネットワーク)

コントローラとして人工ニューラルネットワーク適用し、アニボットに対する制御量を決定する。入力信号として、センサからの情報やアクチュエータの状態が与え

られ、出力信号として、アクチュエータに与えるトルク及び角速度が出力される。

目標行動の獲得は、遺伝的アルゴリズム (GA) を用いて、各ニューロンをつなぐウエイトを最適化することで行う。本研究で用いる3層ニューラルネットワークのコントローラNCの式を以下に示す。

$$NC = \langle X, Y, H, W, V \rangle \quad (7)$$

ここで、 $X = \{x_i\}$ ,  $Y = \{y_j\}$ ,  $H = \{h_k\}$ ,  $W = \{w_{ik}\}$  及び  $V = \{v_{kj}\}$  であり、これらは、 $X$ は入力信号、 $Y$ は出力信号、 $H$ は隠れノード、 $W$ ,  $V$ はそれぞれ  $X \rightarrow H$ ,  $H \rightarrow Y$ のシナプスのウエイトである。 $W$ と $V$ の値を最適化することで、目標とする行動を獲得する。

## 5. 協調行動の獲得実験

本論文では協調行動獲得のために、一台では達成できない荷運び行動の獲得問題を設定した。

### 5.1. 実験モデル

図1に示す荷運びロボットを2台用いて上部に乗せた円盤を光源 (目的地) まで運ぶ行動を獲得する。それぞれのタイヤが正転と逆転が可能なので、前進、後退、旋回の行動ができる。

#### 5.1.1. 入力 $I_i$

時刻 $t$ における $A_i$ の具体化した入力を以下の式で表す。

$$I_i(t) = \langle \theta_{iL}(t), \theta_{iD}(t), d_{iD}(t), \theta_{ij}(t), d_{ij}(t), v_{ik}(t) \rangle \quad (8)$$

各要素はそれぞれ荷運びロボット $i$ における、光源への角度、円盤中心への角度、円盤中心までの距離、他のロボットへの角度、他のロボットへの距離、他のロボットとの相対速度となる。

#### 5.1.2. 出力 $O_i$

時刻 $t$ における $A_i$ の具体化した出力を以下の式で表す。

$$O_i = \{O_{ik} : k = 1, 2\} \quad (9)$$

$$O_{ik}(t) = \Omega y_k(t) \quad (10)$$

ここで、 $\Omega$ は定数、 $y_j$ はニューラルネットワークの $j$ 番目の出力であり $[-1, 1]$ の値をとる。 $O_{ij}$ は $j$ 番目のアクチュエータの目標角速度となる。

### 5.2. 学習条件

実験開始時のアニボットと円盤の配置は、図2の右側のように設定する。ニューラルネットワークの設定は、入力ノード数10、中間ノード数5、出力ノード数2とした。GAの条件を個体数20、世代数200、交叉率40%、突然変異率5%とした。

学習は以下の評価式 $C(S)$ を最大化するようにコントローラを学習することで行う。

$$C(S) = -\{d_t + P_f(t)\} \quad (11)$$

ここで、 $d_t$ は円盤の中心から光源までの距離、 $P_f$ は荷運びロボット同士が衝突した際のペナルティである。この評価下では、ロボット同士が衝突しないように、かつ素早く円盤を光源まで運ぶほど良い評価が得られる。また、 $P_f(t)$ は途中で円盤を落とした場合以下のペナルティである。その場合シミュレーションを終了する。

$$P_f(t) = \begin{cases} -\alpha d_t (STEP - t) : t < STEP \text{かつ円盤が落下} \\ 0 : \text{otherwise} \end{cases} \quad (12)$$

ここで、 $\alpha$ は調整用変数で、値が大きいほど円盤を落としたときのペナルティが高くなる。 $STEP$ はシミュレーションステップ数である。



fig1. A load-carrying robot

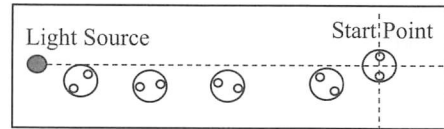


Fig2. A trajectory of load-carrying robots (After learning)

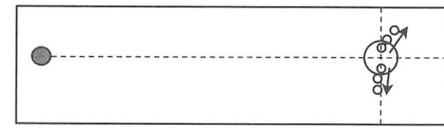


Fig3. A trajectory of load-carrying robots (Before learning)

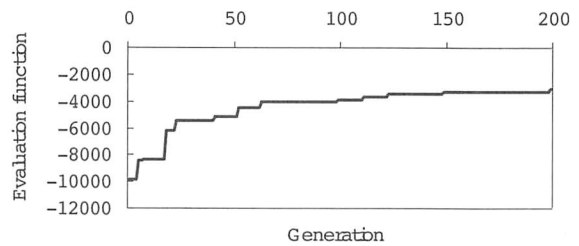


Fig4. A trajectory of load-carrying robots

### 5.3. 実験結果

学習後の荷運びロボットが移動した軌跡を図2に示す。学習前は、各ロボットが自分勝手に移動し円盤をすぐに落としていた。学習後は、それぞれのロボットが互いに距離を保ったまま光源まで移動した<sup>2)</sup>。また、学習前は図3のように個々のロボットが自分勝手に行動しすぐに円盤を落としてしまう。図4は、世代ごとの評価値である。世代が進むにつれて、荷運び行動が獲得できていることが確認できる。

ただし、各ロボットがそれぞれ光源までの移動軌跡を学習しただけとも言えるので、互いにどのような影響をあたえているかの検証が必要である。

## 6. おわりに

本論文では、アニボットの協調行動として荷運び行動の獲得を行った。今後は、荷運びロボットの数を増やしより複雑な協調行動の獲得実験を行う。また、以下の点についても検証していく。

- (1)アニボット同士が協調しているかの検証
- (2)ノイズのある環境下での行動獲得
- (3)より難易度の高い行動 (障害物回避など) の獲得

### 参考文献

- 1)川上 祥, 鈴木 育男, 山本 雅人, 古川 正志: アニボットの協調行動の獲得. 社団法人日本機械学会 機械力学・計測制御部門 Dynamics and Design Conference 2009
- 2)<http://junji.complex.eng.hokudai.ac.jp/researches/physics-modeling/movies/kawakami/>