

# 複数の報酬とゲート機構を用いた モジュール型強化学習アルゴリズム

吉田 裕昭<sup>†</sup> 中村 真吾<sup>‡</sup> 橋本 周司<sup>‡</sup>

早稲田大学 先進理工学部応用物理学科<sup>†</sup> 早稲田大学 理工学術院<sup>‡</sup>

## 1. はじめに

強化学習[1]は、エージェントが環境との試行錯誤により行動の目標に応じた報酬の総和を最大にするような行動則を獲得するための枠組みである。制御結果に対する評価だけを用いて学習し、制御対象に対する事前知識を必要としないため、幅広い制御対象に適用できる可能性がある。しかし、入出力数が多い複雑なシステムの最適な制御器を獲得しようとする、状態空間が指数関数的に拡大し、膨大な学習時間が必要となってしまう。この問題の解決策として複数の単純な制御器を用意し、系の制御方法を学習するモジュール型強化学習が提案されている[2][3]。しかし、いずれかの単純な制御器を選択するだけの従来手法では、結局のところ、単純な制御しか行うことができない。

そこで、本研究では複数の制御器に対して更にゲートを設け、制御器ごとに報酬を与える手法を提案する。これにより、状態空間の爆発を抑えつつ複雑な系の制御則を獲得することが期待できる。実験では、テレビゲームのキャラクターの操作制御に提案手法を適用し、その有効性を確認した。

## 2. 提案手法

### 2.1 アルゴリズムの概要

制御目的に応じて複数の制御モジュールを用意する。モジュール  $m$  は対応した状態  $s_m$  を観測し、行動  $a_m$  を出力する。行動  $a_m$  の  $k$  番目の要素  $a_{m,k}$  をゲート  $G_k$  に渡し、ゲートの選択則に従ってどのモジュールの行動要素を選択するか決定する。このようにして各モジュールの行動要素を組み合わせた行動  $a$  がシステム全体の制御出力となり、より複雑な制御を可能とする。各モジュールの報酬  $r_m$  は制御目的に応じて決定される。図1にモジュール数2、行動要素数3の場合のアルゴリズムの概要を示す。

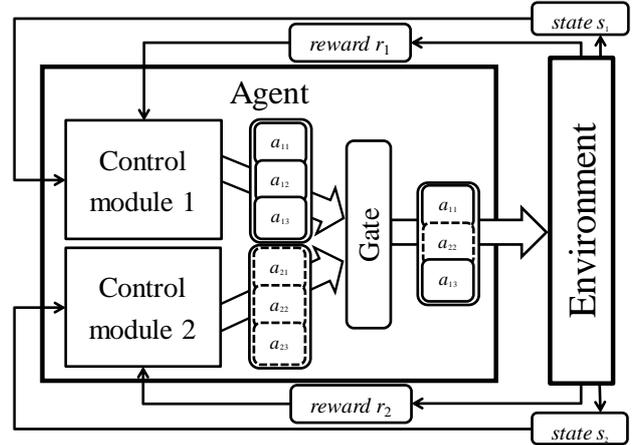


図1 提案学習アルゴリズムの概要

### 2.2 Q学習

各モジュールは Q-Learning: 方策オフ型 TD(0)法により学習を行う。つまり、行動価値  $Q_m$  は以下の(1)式によって更新される。

$$Q_m(s_m, a) \leftarrow Q_m(s_m, a) + \alpha [r_m + \gamma \max_{a'} Q_m(s'_m, a') - Q_m(s_m, a)] \quad (1)$$

ここで、 $s'_m$  はモジュール  $m$  に渡される次のステップの状態、 $\alpha$  は学習率、 $\gamma$  は割引率である。

### 2.3 行動出力則

各モジュール  $m$  はソフトマックス行動選択によって行動  $a_m$  を出力する。つまり、行動  $a_m$  が選択される確率は

$$P(a_m) = \frac{e^{Q(a_m)/\tau}}{\sum_{b_m} e^{Q(b_m)/\tau}} \quad (2)$$

とする。ここで  $\tau$  は定数である。

### 2.4 ゲート選択則

各モジュールから得られた行動の選択方法として、優先選択と多数決選択の2つの方法で実験を行い評価することとした。

優先選択では各モジュールに予め優先順位を付け、優先度の高いモジュールの行動を優先的に選択するようにする。ただし、優先度の高いモジュールの行動価値  $Q_m$  が他の行動を取る時に比べて低い場合には、次に優先度の高いモジュールについて同様の処理を行う。一方、多数決選択では各モジュールが最も多く出力した行動を選択する。

Modular reinforcement learning algorithm using multiple rewards and gates.

<sup>†</sup>Hiroaki Yoshida, Department of Applied physics, Waseda University

<sup>‡</sup>Shingo Nakamura, Shuji Hashimoto, Faculty of Science and Engineering, Waseda University

### 3. 評価実験

#### 3.1 実験対象

オープンソーステレビゲーム『INFINITE MARIO BROS』[4]に提案手法を適用し、性能を評価した。このゲームはスクロール型アクションゲームで、制御するキャラクタをステージ右端まで進めることを目的とする。ステージには所々に段差や落とし穴、敵等の障害物が設置されており、地形の状態の多さや、時間の経過により刻々と状態が様々に変化するという点で複雑な制御を必要とする。

ゲームステージは制御キャラクタの幅の150倍とし、9度グラウンドレベルが変化するものとした。更に、落とし穴を2つ設置し、踏み倒す以外に接触してはいけない敵キャラクタの数を1に設定した。

#### 3.2 モジュール構成

実験では制御目的ごとにモジュールを3つ用意した。各モジュールが扱う状態は、 $s_1$ は段差の $x, y$ 座標と高さ、 $s_2$ は穴の $x, y$ 座標と幅、 $s_3$ は敵の $x, y$ 位置と向きとし、これらにキャラクタの状態( $x, y$ 方向の速度、ジャンプ状態)を加えて、各モジュールに渡す。表1に各モジュールの制御目的と扱う状態数を示す。行動は、ゲームを制御するジャンプ・ダッシュ・方向の3種類のボタンのON/OFFとし、全部で8種類とした。各モジュールに与える報酬値を行動結果ごとに表2に示す。この時、学習率 $\alpha=0.3$ 、割引率 $\gamma=0.9$ 、 $\tau=5.0$ とした。

#### 3.3 結果

提案手法を評価するために通常の強化学習も行った。選択則のモジュール優先順位の付け方を、落とし穴飛び越えモジュール優先、敵撃破モジュール優先を用意した。加えて多数決選択も適用し全部で4つの手法を比較した。 $Q$ 値の更新は3フレームごとに行われ、全モジュールの更新回数の総和が1000に達するごとに、ステージクリア率を求めた結果を図2に示す。

穴飛び越えモジュール優先選択と敵撃破モジュール優先選択は、通常強化学習よりも早い段階で高いステージクリア率を示していることが確認できる。これらの間であまり差異が生じなかったのは穴と敵の2つの障害がゲームの性質

表1 モジュールの制御目的と状態数

モジュール	制御目的	全状態数
1	段差を越えて先に進む	3780
2	落とし穴を飛び越える	3528
3	敵を倒す、または避ける	4860

表2 報酬定義

種類	行動結果	値
$r_1$	右に進む	1ステップに進んだ距離
	壁に衝突	-20
$r_2$	落とし穴を飛び越える	+100
	落とし穴に落ちる	-100
$r_3$	敵を倒す	+100
	敵と接触	-100
	敵を飛び越える	+10

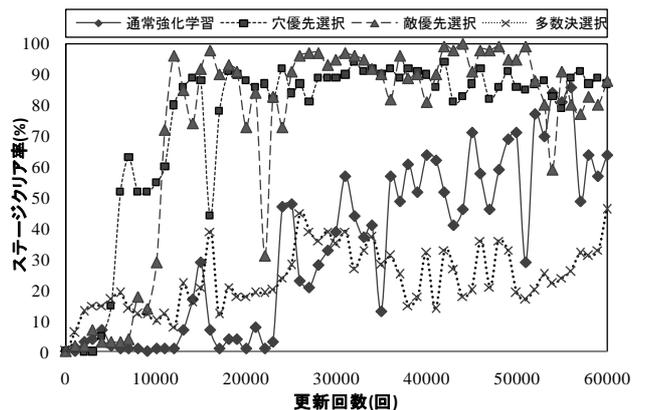


図2 更新回数とステージクリア率の関係

上、同時に発生することがほとんどないためと思われる。一方、多数決選択ではあまり良い結果が得られなかった。

### 4. まとめ

複数の報酬とゲート機構を用いた学習アルゴリズムを提案し、テレビゲームのキャラクタ制御に適用することでその有効性を確かめた。今後さらに制御モジュールを増やし、テレビゲーム以外のロボットプラットフォームなどに提案手法を適用することで、本手法の汎用性を確認したいと考えている。

### 謝辞

本研究の一部は、早稲田大学ヒューマノイド研究所、グローバルCOEプログラム「グローバル ロボット アカデミア」、科学技術振興機構CREST研究「人を引き込む身体的メディア場の生成・制御技術」の研究助成を受けて行われた。

### 参考文献

- [1] Richard S.Sutton and Andrew G.Barto, "Reinforcement Learning: An Introduction", The MIT Press, 1998.
- [2] 山田訓, "モジュール型強化学習", 信学技報, NC97(623), pp.139-146, 1998.
- [3] 中間隼人ら, "3種類のセンサを持つロボット制御へのモジュール型強化学習の適用", 電子情報通信学会, NC108(480), pp.301-306, 2009.
- [4] M.persson, "INFINITE MARIO BROS", Available: <http://www.mojang.com/notch/mario/>