

## リアルタイム適応のためのLEOエージェントモデルの提案

河原林 友美\* 久保 長徳\*\* 高橋 勇\*\*\* 黒岩 丈介\*\*\* 小高 知宏\*\*\*\* 小倉 久和\*\*\*

### LEO Agent Model for Real-time Adaptation

Tomomi KAWARABAYASHI\*, Takenori KUBO\*\*, Isamu TAKAHASHI\*\*\*,  
Jousuke KUROIWA\*\*\*, Tomohiro ODAKA\*\*\*\* and Hisakazu OGURA\*\*\*

(Received January 31, 2007)

We propose a learning agent model for real-time adaptation based on LEO (Learning from Experience and Observation), in which "Learning from Observation (LO)" is incorporated with "Learning from Experience (LE)". In the present paper, the LEO agent model is applied for a soccer agent to adapt its shooting decision to its opponent in the RoboCup simulated soccer games. Through ninety experiments by using the RoboCup Soccer Simulator, the proposed model can improve a success rate of "shot" action to 0.12 from 0.04 (non-learning) and 0.06 (LE model only), suggesting that the LEO agent model works effectively for a real-time adaptation problem.

**Key Words** : Autonomous Agent, Observation Learning, Real-time Adaptation, Soccer Agent and RoboCup

### 1. はじめに

本論文では、ある行動を環境に応じてリアルタイムに適応させるために学習するエージェントのモデルを提案する。我々は、RoboCup<sup>[1][3]</sup>のサッカーシミュレーションリーグにおいて、サッカープレーヤーエージェント<sup>[4][5]</sup>の作成を行っている。サッカーゲームは、そのようなエージェントが11体ずつ2チーム、計22体からなるマルチエージェントシステムである。個々のサッカープレーヤーはそれぞれの特徴をもったエージェントで、走ったり、ボールをドリブルしたり蹴ったりする。パスしたり、シュートして

ゴールしたりしてゲームをする時は、当然ながら相手チームのエージェントの行動から影響を受ける。このような状況下で、リアルタイムで、エージェントが学習する、あるいは相手チームの特徴に適応する、ということは重要なエージェント機能である。

しかし、これまでのエージェントの学習システムについては、リアルタイム性について余り考慮されていない。たとえば、エージェントが経験に基づいてある行動を学習する場合、マルチエージェント環境では他のエージェントと行動が競合し、結果、自分の学習機会が失われる場合がある。これは、学習のための十分な機会が保証されているわけではなく、リアルタイムでは適応のための学習が意図したレベルまで行われぬ可能性が高い。このような学習機会の問題は、マルチエージェント環境におけるエージェントの学習機能上の課題である。

この課題に対して我々は、学習機会を増やし、リアルタイム適応の効果を上げる方法として、他者の行動を観察し学習する仕組みを検討してきた<sup>[6]</sup>。これは、人間が自分の経験から学ぶだけでなく、他人の行動を観察して、その結果を評価し学習する行動に基づいた仕組みである。

\* 工学研究科システム設計工学専攻

\*\* 仁愛大学

\*\*\* 知能システム工学科

\*\*\*\* 工学研究科原子力・エネルギー安全工学専攻

\* System Design Engineering Course, Graduate School of Engineering

\*\* Jinai University

\*\*\* Dept. of Human and Artificial Intelligent System

\*\*\*\* Nuclear Power and Energy Safety Engineering Course, Graduate School of Engineering

本論文では、「経験からの学習 (LE)」と「観察からの学習 (LO)」を併用した「経験と観察からの学習 (LEO)」を導入した LEO エージェントのモデルを提案する。そして、RoboCup サッカーエージェントのシュート行動にこのモデルを川い、RoboCupSoccerServer<sup>[7],[8]</sup>を川いでシミュレーションで試合を行いモデルの有用性を示した。

以下、2章では、LEOを導入した LEO エージェントモデルを記号を川いで定義する。3章では、上記の LEO エージェントモデルを RoboCup サッカーエージェントに実装した例を示す。4章では、提案モデルの有用性を検証するためのサッカーエージェントによるシミュレーション実験と結果を示す。実験では、エージェントのシュート判断をリアルタイム適応の対象とした。5章でまとめを行う。

## 2. LEO エージェントモデル

LEO エージェントモデルは、センサ情報処理部、意志決定部、動作部、学習部からなるエージェントのシステムの学習部に、自分の行動結果から学習を行う LE と他人の行動を観察し自分に置き換えて学習をする LO と併用した LEO を組み込んだモデルである。本論文では、このモデルを記号を川いで定義する。

例えばサッカーエージェントの場合、このシステムのセンサ情報処理、意志決定部、動作部は、以下のように動作している。センサ情報処理部では、「ボール、味方、敵の位置や速度」、「ゴールの位置」などのセンサ情報入力を得る。意志決定部では、センサ情報入力と「体の正面と頭の正面の角度差」や「ボールをキックできる範囲」など自分の身体の状態や能力を表す情報である内部状態、「自身が推定した味方や敵のエージェントモデル」である他エージェントモデルを川いで、「ボールが蹴れる」、「ゴールが近い」、「シュートコースが空いている」などの環境状況を認識する。更に、前件が状況で後件が「シュート」などの行動の if-then ルールで構成される行動ルールを複数持っており、この中からルールをひとつ選択することで、「パス」、「ドリブル」、「シュート」などの行動を表す行動ラベルを選択する。動作部では、その行動ラベルを「kick」、「dash」、「turn」などの動作コマンドに変換し、出力する。

### 2.1 LEO エージェントモデルの定義

このエージェントが存在する環境には、複数のエージェントと、これらのエージェントが行動できる物理的なフィールドから構成される。また、これらのエージェントは、少なくとも、環境を知覚し行動できる。

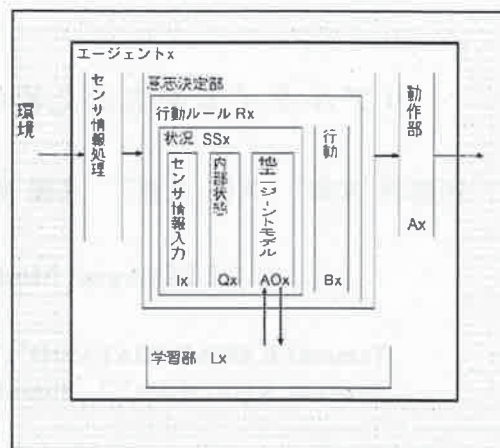


図1: LEO エージェントの枠組

エージェントは、環境を感知するセンサ情報を入力として、環境を認識し環境状況を得る。この環境状況より行動ルールを選択して行動ラベルを決定し、それを動作に変換して環境に対して出力する。また、学習を行なうための学習ルールを持っている。

全てのエージェントの集合を *Agents* とする。エージェント  $x \in Agents$  を、次の8つ組で定義する。

$$x = \langle I_x, Q_x, AO_x, SS_x, B_x, A_x, R_x, L_x \rangle \quad (1)$$

ここで、 $I_x$  はエージェント  $x$  のセンサ情報入力の集合、 $Q_x$  はエージェント  $x$  の内部状態集合、 $AO_x$  はエージェント  $x$  が推定する他エージェントモデルの集合、 $SS_x$  はエージェント  $x$  の環境状況集合、 $B_x$  はエージェント  $x$  の行動ラベル集合、 $A_x$  はエージェント  $x$  の動作集合、 $R_x$  は行動ルール集合、 $L_x$  は学習ルール集合である。学習ルール集合は行動ルールを書き替えるルールである。

図1は、この8つ組の関係をエージェントのシステムに対応付けたものである。エージェントのシステムは、センサ処理部、意志決定部、動作部、学習部からなる。意志決定部には行動ルール集合  $R_x$  が、動作部には動作集合  $A_x$  が、学習部には学習ルール集合  $L_x$  が対応する。行動ルールは、状況と行動の組合せからなるが、これには、環境状況集合  $SS_x$  と行動ラベル集合  $B_x$  が対応している。また、状況は、センサ入力、内部状態、他エージェントモデルによって判断されるが、これには、センサ入力集合  $I_x$ 、内部状態集合  $Q_x$ 、推定された他エージェントモデル集合  $AO_x$  が対応する。一般には、学習によって、行動ルール内の状況と行動の組合せを変更することが可能であるが、ここでは行動ルー

ル内における推定された他エージェントモデル ( $AO_x$ ) を変更する場合を対象とする。

## 2.2 LEO エージェントのセンサ情報処理と環境状況の認識

エージェント  $x$  は、自分の内部状態集合  $Q_x$  及びエージェント  $x$  が想定する他エージェントモデル集合  $AO_x$  を用いて、センサ情報入力集合  $I_x$  を解釈し、環境状況集合  $SS_x$  として認識する。また、エージェント  $x$  は内部状態を複数個持っており、 $i$  番目の内部状態を  $q_x^i \in Q_x$  と表現する。さらに、エージェント  $x$  は自身が想定する他エージェントモデルを複数個持っており、 $j$  番日の他エージェントモデルを  $Asm(x, j) \in AO_x$  ( $j \in Agents, j \neq x$ ) と表現する。これらの認識の結果、得られる環境状況の集合を  $SS_x$  とする。また、引数を明示して、 $SS_x(I_x, Q_x, AO_x)$  とも表す。

エージェント  $x$  が想定する  $j$  番日の他エージェント  $Asm(x, j)$  は次のような7項組である。

$$Asm(x, j) = \langle asm(x, I_j), asm(x, Q_j), asm(x, SS_j), asm(x, B_j), asm(x, A_j), asm(x, R_j), asm(x, L_j) \rangle \quad (2)$$

これら、 $asm(x, I_j)$ ,  $asm(x, Q_j)$ ,  $asm(x, SS_j)$ ,  $asm(x, B_j)$ ,  $asm(x, A_j)$ ,  $asm(x, R_j)$ ,  $asm(x, L_j)$  は、それぞれ「エージェント  $x$  が想定したエージェント  $j$ 」の環境状況集合、センサ情報入力集合、内部状態集合、行動ラベル集合、動作集合、行動ルール集合、学習ルール集合である。これらはエージェント  $j$  の実際の構成と関係はない。

## 2.3 LEO エージェントのルール

### 行動ルール

エージェントの行動ルールは、if-then ルールからなり、後件は行動を表す行動ラベル、前件はその行動が実行されるための状況集合である。エージェント  $x$  は、行動ルール集合  $R_x$  から行動ルールを1つ選ぶことで行動を選択する。行動ルールが複数選択された場合は競合解消を行なって1つを選択する。

エージェント  $x$  の認識結果が  $k$  番日の行動ルール  $r_x^k \in R_x$  の前件  $S_x^k \in SS_x$  に合致し、選択されたとする。その結果、エージェント  $x$  の  $k$  番目の行動ラベル  $b_x^k \in B_x$  が選択される。エージェント  $x$  の  $k$  番日の行動ルール  $r_x^k$  を

$$r_x^k = \text{if}(S_x^k(I_x, Q_x, AO_x)) \text{then}(b_x^k) \quad (3)$$

と表す。ここで、 $S_x^k \in SS_x$  は、エージェント  $x$  の  $k$  番日の行動ラベル  $b_x^k$  が選択される条件を表す状況集合である。この  $S_x^k$  を

$$S_x^k = \{s_x^{km}\} \quad (4)$$

と表現する。ここで、 $s_x^{km}$  は、エージェント  $x$  の  $k$  番目の行動の条件を表す状況集合の  $m$  番目の状況である。引数を明示して次のようにも表す。

$$s_x^{km} = s_x^{km}(I_x, Q_x, AO_x) \quad (5)$$

### 動作出力

エージェント  $x$  は、 $k$  番日の行動ラベル  $b_x^k \in B_x$  を選択した結果、動作  $a_x^{kn} \in A_x^k$  を環境に対して出力する。 $A_x$  はエージェント  $x$  のすべての動作集合、 $A_x^k$  はその部分集合である。 $a_x^{kn}$  はエージェント  $x$  の  $k$  番日の動作集合  $A_x^k$  中の  $n$  番目の動作である。たとえば、エージェント  $x$  をサッカーエージェント、 $a_x^{kn}$  をボールを蹴る動作、行動ラベル  $b_x^k$  をシュート行動ラベルとすると、この時、シュート行動を実現するための蹴る動作に関する角度と強さの組み合わせの集合は  $A_x^k \in A_x$  のように表せる。 $a_x^{kn}$  は、シュートを実現するための蹴る動作集合  $A_x^k$  のうちの1つを表している。

### 学習ルール

学習ルールを用いて行動ルールを操作する場合、一般には行動ルールの取り換え、行動ルールの前件と後件の組み合わせの変更、行動ルールの組み合わせはそのまま前件や後件に修正変更を加えるような学習方法が考えられる。ここでは、学習ルール  $l_x^1, l_x^2, l_x^3, l_x^4 \in L_x$  を用いて、行動ルールの前件部分を修正変更する。LE では、自らの行動と結果から、対応する行動ルールの前件部を評価修正する。LO では、自分が学習対象としている行動を第三者が行っている場合、その状況と行動と結果を観察し、対応する自分の行動ルールの前件部を評価修正する。また、学習のタイミングであるが、LE は自ら行動したとき、LO は他者の行動を観察したときにそれぞれの学習ルールが発動するため、これらの2つの学習は同時には行われない。

### 学習対象

エージェント  $x$  の行動ルール  $r_x^k$  の前件部のある状況  $s_x^{km}$  に用いられている  $x$  が想定しているエージェント  $y \in Agents$  ( $y \neq x$ ) の  $t$  番日の内部状態  $asm(x, q_y^t)$  を学習対象とする。これらの関係を次のように記述する。

$$r_x^k = \text{if}( S_x^k(I_x, Q_x, AO_x) ) \text{then}( b_x^k ) \quad (6)$$

$$s_x^{km} = s_x^{km}(I_x, Q_x, asm(x, q_y^t)) \quad (7)$$

### LE のルール

LE では、エージェント  $x$  が自分の行動の成否によって学習を行う。成功時は学習ルール  $l_x^1$ 、失敗時は学習ルール  $l_x^2$  に従う。

$$l_x^1 = \text{if}( s_x^{km}(I_x, Q_x, asm(x, q_y^t)) = T \quad \text{and } Res(b_x^k) = S )$$

$$\text{then}( asm(x, q_y^t) \leftarrow Adj(asm(x, q_y^t)) = F(asm(x, q_y^t)) ) \quad (8)$$

$$l_x^2 = \text{if}( s_x^{km}(I_x, Q_x, asm(x, q_y^t)) = T \quad \text{and } Res(b_x^k) = F )$$

$$\text{then}( asm(x, q_y^t) \leftarrow Adj(asm(x, q_y^t)) = F'(asm(x, q_y^t)) ) \quad (9)$$

$Res(b_x^k)$  は行動  $b_x^k$  の結果を, S, F は成功, 失敗を表す。 $F(asm(x, q_y^t))$  と  $F'(asm(x, q_y^t))$  はそれぞれ学習の具体的操作を表す式である。 $Adj(asm(x, q_y^t))$  は,  $asm(x, q_y^t)$  の操作結果を表す。 $\leftarrow$  は, 値を代入し更新することを表す記号である。学習ルール  $l_x^1, l_x^2$  の後件部では, 行動ルール  $r_x^k$  の前件部  $s_x^{km}$  の  $asm(x, q_y^t)$  の値を基に,  $Adj(asm(x, q_y^t))$  を求め, それを再び行動ルール  $r_x^k$  の前件部  $s_x^{km}$  の  $asm(x, q_y^t)$  に代入している。

### LO のルール

LO では, エージェント  $x$  が学習したい行動を, 第三者であるエージェント  $z \in Agents$  ( $z \neq x, y$ ) が行った場合, その状況と行動を観察し, 自分に置き換え, その行動に対応する自分の行動ルールの前件部を評価する。この評価結果とエージェント  $z$  の行動結果が異なった場合, 学習ルール  $l_x^3$ , または  $l_x^4$  に従って学習を行う。

$$l_x^3 =$$

$$\text{if}( Perc(x, a_z^{cd}) = T \quad \text{and } a_z^{cd} = a_x^{ke} \quad \text{and } b_z^c = b_x^k \quad \text{and } s_x^{km}(asm(x, I_z), Q_x, asm(x, q_y^t)) = F \quad \text{and } Res(b_z^c) = S )$$

$$\text{then}( asm(x, q_y^t) \leftarrow Adj_{obs}(asm(x, q_y^t)) = F_{obs}(asm(x, q_y^t)) ) \quad (10)$$

$$l_x^4 =$$

$$\text{if}( Perc(x, a_z^{cd}) = T \quad \text{and } a_z^{cd} = a_x^{ke} \quad \text{and } b_z^c = b_x^k \quad \text{and } s_x^{km}(asm(x, I_z), Q_x, asm(x, q_y^t)) = T \quad \text{and } Res(b_z^c) = F )$$

$$\text{then}( asm(x, q_y^t) \leftarrow Adj_{obs}(asm(x, q_y^t)) = F'_{obs}(asm(x, q_y^t)) ) \quad (11)$$

ここで,  $a_z^{cd}$  は, エージェント  $z$  の行動ラベル  $b_z^c$  に関する動作集合の  $d$  番目の動作  $A_z^d$  の要素を表す。 $Perc(x, a_z^{cd})$  はエージェント  $x$  が,  $a_z^{cd}$  を知覚したかどうかを表す。 $asm(x, I_z)$  は, エージェント  $x$  が推定するエージェント  $z$  のセンサ入力情報集合である。 $Res(b_z^c)$  は行動  $b_z^c$  の結果を, S, F は成功, 失敗を表す。 $F_{obs}(asm(x, q_y^t))$  と  $F'_{obs}(asm(x, q_y^t))$  はそれぞれ学習の具体的操作を表す式である。 $Adj_{obs}(asm(x, q_y^t))$  は,  $asm(x, q_y^t)$  の操作結果である。 $\leftarrow$  は値を代入し更新することを表す。

学習ルール  $l_x^3, l_x^4$  の後件部では, 行動ルール  $r_x^k$  の前件部  $s_x^{km}$  の  $asm(x, q_y^t)$  の値を基に  $Adj_{obj}(asm(x, q_y^t))$  を求め, それを再び条件部の  $asm(x, q_y^t)$  に代入している。この  $Adj_{obs}(asm(x, q_y^t))$  は, 他者の行動を観察することにより間接的に求めた値であって, 自分で行った行動から求めた値  $Adj(asm(x, q_y^t))$  とは一般には異なる。したがって, それを再び条件部の  $asm(x, q_y^t)$  に代入する際には, 何らかの変換操作を行う必要がある。しかし, ここでは簡単のためそのまま代入する。

### 3. サッカーエージェントのシュート判断への適用

LEO エージェントモデルをサッカーエージェントに適用し, シュート判断を適応的に行うエージェントを構築した。

サッカーの試合においては、対戦相手のインターセプトの能力によって、シュート時における対戦相手の配置などが全く同じであっても、シュートの成功、失敗が異なる。対戦相手の能力は事前に情報が収集できない場合や試合中に変化する可能性がある。したがって、入りそうなのにシュートしない、取られるのにシュートしてしまうという状況避けるために、シュートが成功するかどうかの状況判断をリアルタイムに適応的にする必要がある。また、シュート行動は、他の行動に比べて多くは行われず、他のエージェントが実行してしまうと、自分で学習するチャンスが減る。したがって、リアルタイムに適応するためには学習機会が少いことを補う必要がある。

### 3.1 シュート行動ルールと状況のモデル化

エージェント  $p$  のシュート行動ルール  $r_p^{\text{shoot}}$  を次のように表現する。

$$r_p^{\text{shoot}} = \text{if}(S_p^{\text{shoot}}(I_p, Q_p, AO_p)) \text{then}(b_p^{\text{shoot}}) \quad (12)$$

$$= \text{if}(s_p^{\text{shoot}1} \text{ and } s_p^{\text{shoot}2} \text{ and } s_p^{\text{shoot}3}) \text{then}(l_p^{\text{shoot}}) \quad (13)$$

また、シュート行動ルールの前件部、後件部は具体的には次のような状況や行動として表現する。

$$s_p^{\text{shoot}1} = (\text{I can kick the ball}) \quad (14)$$

$$s_p^{\text{shoot}2} = (\text{I am in the shooting distance}) \quad (15)$$

$$s_p^{\text{shoot}3} = (\text{A shooting space is open}) \quad (16)$$

$$b_p^{\text{shoot}} = (\text{Shoot}) \quad (17)$$

このうち  $s_p^{\text{shoot}3} = (\text{A shooting space is open})$  という状況の評価は、次のような認識の過程を経て行なう。ゴール周辺にいる対戦相手の位置などから、シューターがゴールを臨む領域から (図2 領域T)、インターセプトされそうな領域 (図2 領域U) と重なる部分を (図2 領域V) を差し引いて、開いている領域 (図2 領域W) があればシュート可能と判断する。

インターセプト領域の広さは対戦相手によって異なる。この領域の広さを表現するためにインターセプト能力推定値指数  $\alpha = b/a$  を設定する (図2)。 $a$  はインターセプターがボールを捉える事のできる最短地点までのボールの移動距離、 $b$  はインターセプターがボールを捉える事のできる最短距離である。行動能力はエー

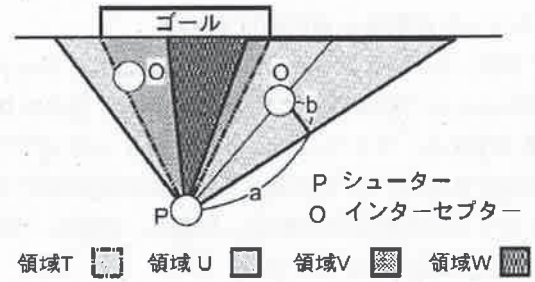


図2: シュートスペース認識のモデル化

ジェントごとに異なるが  $\alpha$  は、初期値が  $\alpha_0$  で、最小値  $\alpha_{min}$  から最大値  $\alpha_{max}$  までの実数とする。 $\alpha$  の値が低いと対戦相手のインターセプト領域は小さく、高いとインターセプト領域は大きいと判断する。

ここでは、簡単のため各インターセプターを区別せず同一の  $\alpha$  を用いる。この  $\alpha$  は、エージェント  $p$  が推定するインターセプター  $o$  の内部状態集合の一つであるので、これを  $h$  番目の内部状態とし、 $\alpha = \text{asm}(p, q_o^h)$  とする。よって、 $s_p^{\text{shoot}3}$  を次のように記述する。

$$s_p^{\text{shoot}3} = s_p^{\text{shoot}3}(I_p, Q_p, \alpha) \quad (18)$$

### 3.2 シュート行動ルールにおける LE と LO

#### (1) シュート行動ルールにおける LE

LE では、エージェント  $p$  が自分のシュートの結果によって学習を行う。成功時は学習ルール  $l_p^1$ 、失敗時は  $l_p^2$  に従う。

$$l_p^1 = \text{if}(s_p^{\text{shoot}3}(I_p, Q_p, \alpha) = \text{T} \text{ and } \text{Res}(b_p^{\text{shoot}}) = \text{S}) \text{ then}(\alpha \leftarrow \text{Adj}(\alpha) = F(\alpha) = \alpha - \delta) \quad (19)$$

$$l_p^2 = \text{if}(s_p^{\text{shoot}3}(I_p, Q_p, \alpha) = \text{T} \text{ and } \text{Res}(b_p^{\text{shoot}}) = \text{F}) \text{ then}(\alpha \leftarrow \text{Adj}(\alpha) = F'(\alpha) = \alpha + \delta) \quad (20)$$

ここで、 $\text{Res}(b_p^{\text{shoot}})$  は行動ラベル  $b_p^{\text{shoot}}$  が示す行動の結果で、S、Fはその成功、失敗を表す。 $F(\alpha)$  と  $F'(\alpha)$  はそれぞれ学習の具体的な操作を表す式である。 $\text{Adj}(\alpha)$  は、 $\alpha$  を操作した値である。 $\delta$  は、 $\text{Adj}(\alpha)$  を求めるために  $\alpha$  を調節する実数定数である。 $\leftarrow$  は値を代入し更新することを表す。

## (2) シュート行動ルールにおける LO

LOでは、エージェント  $p$  が第3者であるエージェント  $u$  のシュート行動と状況と結果を観察し、自分の状況に置き換える。そして、シュート行動ルール  $r_p^{\text{shoot}}$  の前件部である  $s_p^{\text{shoot}3}$  を評価する。この評価結果とエージェント  $u$  の行動結果が異なった場合、学習ルール  $l_p^3$ 、または  $l_p^4$  に従って学習を行う。

$$\begin{aligned}
l_p^3 = & \\
& \text{if} ( \text{Perc}(p, a_u^{\text{shoot}f}) = T \\
& \quad \text{and } a_u^{\text{shoot}f} = a_p^{\text{shoot}g} \\
& \quad \text{and } b_u^{\text{shoot}} = b_p^{\text{shoot}} \\
& \quad \text{and } s_p^{\text{shoot}3}(asm(p, I_u), Q_p, \alpha) = F \\
& \quad \text{and } Res(b_u^{\text{shoot}}) = S ) \\
& \text{then}(\alpha \leftarrow Adj_{obs}(\alpha) = F_{obs}(\alpha) = \alpha - \delta')
\end{aligned} \tag{21}$$

$$\begin{aligned}
l_p^4 = & \\
& \text{if} ( \text{Perc}(p, a_u^{\text{shoot}f}) = T \\
& \quad \text{and } a_u^{\text{shoot}f} = a_p^{\text{shoot}g} \\
& \quad \text{and } b_u^{\text{shoot}} = b_p^{\text{shoot}} \\
& \quad \text{and } s_p^{\text{shoot}3}(asm(p, I_u), Q_p, \alpha) = T \\
& \quad \text{and } Res(a_u^e) = F ) \\
& \text{then}(\alpha \leftarrow Adj_{obs}(\alpha) = F'_{obs}(\alpha) = \alpha + \delta')
\end{aligned} \tag{22}$$

ここで、 $a_u^{\text{shoot}f}$  は、エージェント  $u$  のシュート行動に関連する動作集合  $A_u^{\text{shoot}}$  の  $f$  番目の動作を表す。 $\text{Perc}(p, a_u^{\text{shoot}f})$  は、エージェント  $p$  が  $a_u^{\text{shoot}f}$  を知覚したかどうかを表す。 $a_p^{\text{shoot}g}$  は、エージェント  $p$  のシュート行動のための動作集合  $A_p^{\text{shoot}}$  の  $g$  番目の動作を表す。 $asm(p, I_u)$  は、エージェント  $p$  が推定するエージェント  $u$  のセンサ情報入力を表す。 $Adj_{obs}(\alpha)$  は、 $\alpha$  を操作した値である。 $F_{obs}(\alpha)$  と  $F'_{obs}(\alpha)$  はそれぞれ学習の具体的な操作を表す式である。 $\delta'$  は、 $Adj_{obs}(\alpha)$  を求めるために  $\alpha$  を調節する実数定数である。 $\leftarrow$  は値を代入し更新することを表す。

この  $Adj_{obs}(\alpha)$  は、他者の行動を観察することにより間接的に求めた値であって、自分で行った行動から求めた値  $Adj(\alpha)$  とは異なる。したがって、それを再び条件部の  $\alpha$  に代入する際には、何らかの操作を行う必要がある。しかし、ここでは簡単のためそのまま代入する。また、 $\delta'$  も簡単のため、 $\delta$  と同じ値とする。

表1: チーム Neko に対するチーム 1, 2, 3 の試合結果 (30 試合平均)

チーム	得点	シュート	得点率	センタリング	観察回数
1	0.8	18.3	0.04	0.7	—
2	1.3	20.0	0.06	3.7	—
3	3.0	25.0	0.12	11.5	11.87

得点率 = 得点 / シュート試行回数

## 4. サッカーシミュレータを用いた試合形式の実験

サッカーシミュレーションの試合を行い、提案手法である LEO エージェントモデルを適用したサッカーエージェントのシュート判断のリアルタイム適応効果を示す。

検証項目は以下の通りである。リアルタイム適応効果として、LEO エージェントの得点率が、「学習なし」、LE だけを用いた LE エージェントより高いかを見る。また、リアルタイム適応の結果どのような事が起こったかを「得点」、「シュート試行回数」、「ラストパス回数」から検証する。

## 4.1 実験概要

実験は、学習のタイプを変えた3つのチーム1, チーム2, チーム3と実験用の対戦チーム neko をそれぞれ30回ずつ対戦させることで行なう。

## サッカーエージェントチーム

チーム1, 2, 3は我々が開発したチーム RaiC を基に作成した。チーム1は学習を用いないエージェントからなるチーム、チーム2はFW (フォワード) 2体と攻撃的MF (ミッドフィルダー) 1体にのみLE エージェントモデルを用いたチームである。チーム3はFW2体と攻撃的MF1体にのみLEO エージェントモデルを用いたチームである。IOを併用したエージェントの観察の回数やタイミングは一切制御していない。

## パラメータ

$\alpha$  は初期値0,  $0 \leq \alpha \leq 1$  とした。 $\alpha$  の値を変更する  $\delta$  の値は、0.125 とした。学習を行わないエージェントの  $\alpha$  値は0で固定とした。neko は学習を一切しないチームである。

## 4.2 実験結果

チーム1, 2, 3のFW2体と攻撃的MF1体がnekoから獲得した1試合平均の得点、シュート試行回数、得点率、ラストパス回数を1に示す。1試合の試合平均の得点はチーム3がチーム1の3.75倍、チーム2の

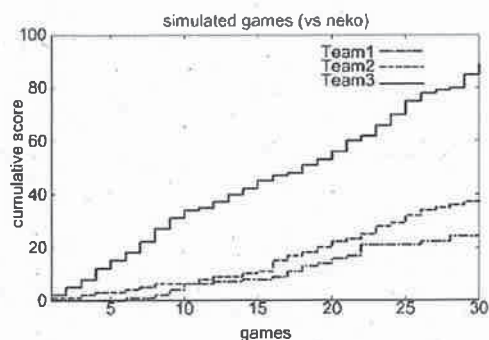


図 3: 30 試合累計得点 (対チーム Neko)

2.31 倍得ている。また、図 3 に、チーム 1, 2, 3 の累積得点結果を示す。チーム 1 がチーム 2, 3 と比べて試合毎に得点を取っていることが分かる。得点率もチーム 3 がチーム 1 の 3 倍、チーム 2 の 2 倍高い。得点率はそのときの状況に応じた適切なシュート判断を行っているか評価するための重要な指標である。学習なしよりも LE エージェントの方が、LE エージェントよりも LEO エージェントの方が、より対戦相手に適応した判断を行い、得点を取るという目的をより達成しているといえる。1 のセンタリングの回数もチーム 3 がチーム 1 の 16.1 倍、チーム 2 の 3.11 倍多い。インターセプト能力推定指数をリアルタイムで得ることによって、「シュートコースが開いている」という状況認識が適応的に変化し、行動ルール選択においてシュートが選択されず、センタリング行動が選択されたということである。得点、得点率、センタリング回数ともにチーム 3 がチーム 1, 2 よりも高い。1 試合中 11.87 回の I.O の回数が LE を補ったと考える。これは LEO によって、より適応的なシュート行動選択、目的の達成ができていているということである。実践的環境において LEO エージェントモデルが有用であるといえる。

## 5. まとめ

マルチエージェント環境におけるエージェントのリアルタイム適応を目的として、「観察からの学習 (LO)」を「経験からの学習 (LE)」と併用する「経験と観察からの学習 (LEO)」を用いた LEO エージェントモデルを提案した。

RoboCup サッカーシミュレータを用い、サッカーエージェントのシュート判断に提案手法を実装し、コンピュータシミュレーション実験を行った。結果、LEO エージェントモデルを用いたエージェントは、LO が LE を補ってシュート状況判断適応の効果が上がったことを示した。

## 参考文献

- [1] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Ozawa.: RoboCup: The Robot World Cup Initiative, Proc. of IJCAI-95 Workshop on Entertainment and AI/Alife (1995)
- [2] D.Nardi, M.Riedmiller, C.Sammut, J.Santos-Victor(Eds.): RoboCup 2004: Robot Soccer World Cup VIII, Springer-Verlang (2005)
- [3] <http://www.robocup.org/>
- [4] 伊藤暢浩, 高橋友一: RoboCup ではじめるエージェントプログラミング, 共立出版 (2001)
- [5] 秋山英久: ロボカップサッカーシミュレーション 2D リーグ必勝ガイド, 秀和システム (2006s)
- [6] T. Kawarabayashi, T. Kubo, T. Morisita, J. Nishino, T. Odaka, and H. Ogura: Real-time Adaptive Learning from Observation for RoboCup Soccer Agents, Proc. of the 6th International Conference on Distributed Autonomous Robotic Systems, Vol.5, pp.207-214, Springer-Verlag (2002).
- [7] I. Noda, H. Matsubara, K. Hiraki, and I. Frank. Soccer server: A tool for research on multi-agent systems, Journal of Applied Artificial Intelligence, Vol.12, pp.233-250 (1998)
- [8] <http://sserver.sourceforge.net/>

