局面評価関数を用いたサッカーエージェントの移動先決定

大内 斉[†1]　五十嵐 治一[†1]

概要：本研究では，RoboCup サッカーシミュレーションリーグ 2D の試合において，チームメイトがボールを保持しているときにボールを持っていないプレイヤーが適切な移動先を決定する方式を提案した．本方式は agent2d で提案されているアクション連鎖探索フレームワークをベースとする．アクション連鎖探索フレームワークでは探索木と局面評価関数を用いるが，今回は新たにレシーバのための局面評価関数をヒューリスティクスに基づいて設計した．さらに，観戦者がプレイヤーの行動や試合局面の優劣を評価し，その結果を強化学習の報酬信号として利用することにより局面評価関数中の重み係数をリアルタイムに学習できるシステムを構築した．学習実験の結果は，10 試合の対戦だけでパス回しからの得点パターンが大幅に増加し，対 agent2d の勝率が 52.1%から 73.2%へと向上し，学習の有効性を確認することができた．

キーワード：サッカーエージェント，RoboCup，局面評価関数，方策勾配法，agent2d

Selecting an Advantageous Target Point for Movement of a Soccer Agent Using a State Evaluation Function

HITOSHI OUCHI[†1]　HARUKAZU IGARASHI[†1]

Abstract: This paper proposes an algorithm for a soccer agent to determine an advantageous target point for movement when a teammate holds the ball in RoboCup Soccer Simulation League 2D games. Our algorithm is based on an action sequence search embedded in an open source program called agent2d. The action sequence search needs a search tree and a state evaluation function. We designed a state evaluation function for a receiver agent using heuristic knowledge about soccer. For real-time learning of the weight parameters in the function, we developed a system where a spectator evaluates the plays and the states of either team and gives reward signals to the team's agents to improve their action decisions. The results of our learning experiments show that the number of goal scoring patterns after passes increased drastically and the winning rate against agent2d improved from 52.1% to 73.2%. Those results support the effectiveness of our proposed state evaluation function and the learning algorithm for receiver agents to determine their positions.

Keywords: Soccer agents, RoboCup, State evaluation function, Policy gradient reinforcement learning, agent2d

1. はじめに

RoboCup サッカーシミュレーションリーグ 2D [1][2]は，マルチエージェントシステムにおけるエージェント制御のための標準問題として人工知能の研究対象となってきた．特に，複数の自律分散型エージェント間の協調行動をどのように実現させ，かつ，それをどのように学習させるのかという研究が興味あるテーマの一つとなっていた．

田川らはシミュレーションリーグの試合において複数エージェント間の協調行動をオンライン的な強化学習システムを開発した．このシステムを用いて，実際に観戦者が試合を観戦しながら報酬を与え，リアルタイムに学習を行わせたところ，10 試合程度で効果的なスルーパスを多数回出せるようになった例が報告されている[3].

上記の研究では秋山英久氏が開発・公開した“agent2d”というサッカーエージェントプログラムが使用されている．この agent2d では，ボールを保持したサッカーエージェントの行動決定には，“チェーンアクション”と呼ばれるアクション連鎖探索フレームワークが採用されている．そこでは，局面をノードとし，パスやドリブル，シュートなどの行動を有向辺とする探索木と，ノード局面での優劣を評価した局面評価関数とを用いて，最適と思われる行動を最良優先探索法により求めている[4]. 田川らの研究[3]ではこの局面評価関数を強化学習の一種である方策勾配法を用いて学習を行った．

しかし, agent2d も田川らの研究もいずれもボール保持者の行動決定を対象としていた．すなわち，ボールを持ったサッカープレイヤーがどこへパスを出すべきか，どこへドリブルして行くべきか，あるいはシュートすべきかを判断するための行動決定方式であった．本研究ではこれとは異なり，チームメイトがボールを保持しているときにボールを持っていないプレイヤーがどのような行動をとるべきかを決定するためにチェーンアクションを用いる．例えば，レシーバとしてどこの位置へ移動するのが最適であるかを決定するような問題である．さらに，田川らのオンライン的な強化学習システムを利用して，観戦者が試合中に報酬を与えながらリアルタイムで局面評価関数を学習させた．

†1　芝浦工業大学
Shibaura Institute of Technology

本論文ではこれらの学習方法の提案と実験結果について報告する．

2. RoboCup サッカーシミュレーション 2D リーグ

RoboCup サッカーシミュレーションリーグ 2D では，コンピュータ上に用意された 2 次元の仮想フィールド上でプログラムにより制御されたサッカーエージェント同士によるサッカーの試合を行う．試合はボールや選手の動きを物理モデルに基づいてシミュレートして表示するサッカーサーバ・プログラム(rcssserver)と，各プレイヤーに相当するサッカークライアント・プログラムとが通信を行いながら進められる．サーバはプレイヤーから見た周囲の環境情報を与え，クライアントはその環境情報を基に状況を判断し，行動を決定し，動作コマンドをサーバへ送り返す（図 1）．

サーバから送信される環境情報としては，プレイヤーが見ることができる視覚情報と聞くことができる聴覚情報とがある．一方，クライアントから送信できる動作コマンドはボールを蹴る（kick コマンド）や，走る（dash コマンド）などのかなり基本的な動作コマンドである．したがって，パスやドリブルといった行動は，いくつかの基本コマンドを組み合わせて実現する必要がある．また，競技のルールは基本的に人間のサッカーと同じである．オフサイドなどの反則やプレイヤーのスタミナの概念も取り入れられており，高さの概念が存在しないという点を除けば人間のサッカーゲームをかなりよくシミュレートしている．

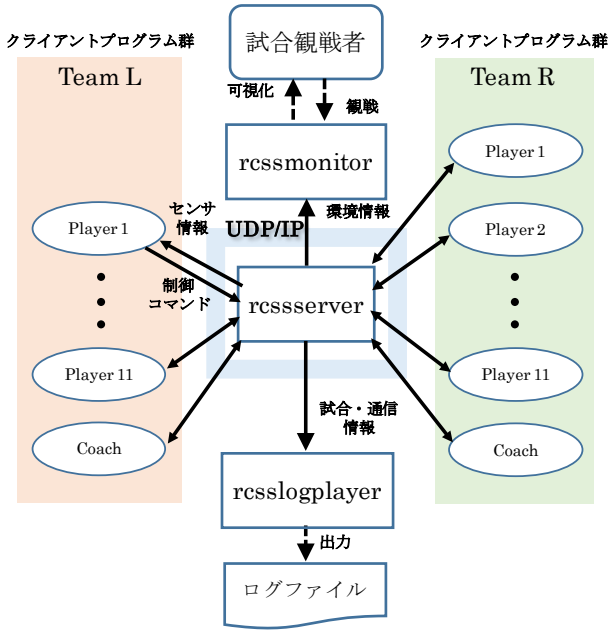

図 1　RoboCup サッカーシミュレーションリーグ 2D におけるプログラム群

Figure 1　Programs operating in the games of RoboCup Soccer Simulation League 2D.

しかし，プレイヤー同士の通信は自由ではなく，定められたコマンドを使用してサーバを介する必要があり，通信量も制限されている．したがって，複数のプレイヤーが連係して行うような“協調行動”を実現させるには何らかの工夫が必要である．

3. agent2d とチェーンアクション

3.1 agent2d とは

agent2d [5]は秋山英久氏らによって開発されている 2D リーグのサンプルエージェントプログラムである．ソースコードは C++で記述され，RoboCup 2010 Singapore 大会で優勝した HELIOS というチームを基にしている．高度な行動決定や戦略は除いてあるが，ドリブルやパスといったサッカーに必要な基本的な行動や，試合を行うのに必要な基本的な戦術や戦略が最初から組み込まれている．そのため，多くのチームがベースプログラムとして agent2d を用いている．

agent2d は，GNU General Public License version.3 [6]として 2006 年に最初に公開された．それ以降，バージョンアップが重ねられ，2016 年 6 月時点での最新バージョンは ver. 3.1.1 である（2012 年 3 月公開．以下，agent2d-3.1.1 と記す）．本研究でもこの agent2d-3.1.1 を使用した．

agent2d は RoboCup サッカーシミュレーション 2 D リーグ用のライブラリ“librcsc”を使用している．このライブラリはチーム開発を支援してくれるソフトウエアであり，解析や幾何計算を行う数学クラス群，rcssserver と通信するためのクラス群，エージェントの内部モデル群,基本的な行動を行うクラス群などで構成されている．この librcsc は agent2d と同じく秋山英久氏らによって開発・公開されている．2016 年 6 月の時点での最新バージョンは 2011 年 5 月に公開された librcsc-4.1.0 である．本研究でも librcsc-4.1.0 を使用した．

3.2 agent2d におけるボール保持者の行動決定

agent2d を起動させると，クライアントプログラムは初期化処理を行った後，①サーバからの環境情報の取得，②環境情報を基にした状況判断と行動決定，③選択した行動コマンドの送信の 3 段階の処理を試合終了まで繰り返す．

②の行動決定においては，プレイヤーがボールを保持した際にはチェーンアクション（chain action）と呼ばれるアクション連鎖探索フレームワークを用いることが 2010 年に公開された ver.3.0.0 から採用された．すなわち，ボール保持者はパスやドリブルといった行動を枝，行動の結果生じる予測状態をノードとする探索木を生成する．各ノードは局面評価関数によって優劣が点数化されている．agent2d では，点数が最も高いノード（必ずしも葉ノードとは限らない）を最良優先探索によって探索し，そのノードへ至る行動を決定論的に選択する（max 戦略）．

これに対し，谷川や田川らの研究[3][7][8]では，上記の過

程において，max 戦略のような決定論的な方策ではなく，Boltzmann 分布を用いた確率的な方策関数を用いることにより，局面評価関数中のパラメータを学習することを可能にしている．本研究でも同様な学習方式を採用する．

3.3 agent2d におけるボール非保持者の行動決定

agent2d では，ボール非保持者が“ホームポジション”を目指して移動することでフォーメーションを維持している．開発者は，フォーメーションエディタ”fedit2”[9]を用いることでフォーメーションを自由に変更することができる．

上記のホームポジションの決定手順は次の通りである[10]．まず，プレイヤーはチーム開発者が予め fedit2 で編集したサンプルデータ（ボールとプレイヤー11 人の座標）が記録されているファイルを読み込む．次に，全サンプルデータのボール座標を頂点とする Delaunay 三角形にフィールドを分割する．もし，ボール座標が与えられると，それを含む三角形の各頂点との相対関係から，各頂点で決まるサンプルデータのプレイヤー座標に対して線形補間を行い，11 人のホームポジションを決定する．

したがって，agetn2d ではあらかじめ用意したサンプルデータを基に移動先が決定され，状況に応じたフォーメーションの変更を考慮していない．したがって，agent2d のフォーメーションに対してパスコースを塞ぐようなポジショニングを行う相手には対応できないという問題点があるが，次章で提案する移動先の決定方式には，このホームポジションの情報もある程度は考慮することにする．

4. チェーンアクションを用いたボール非保持者の行動決定

4.1 ボール非保持者の探索木の構成

agent2d ではボール保持者が探索木を構築する．その探索木のルートノードから延びる枝は保持者自身のパスやドリブルなどの行動であるが，その下の階層における枝はボールを受け取った味方プレイヤーのパスやドリブルなどの行動である．ノードはこれらの行動完了後の予測状態（予測局面）を表している．この予測には，ボールとレシーバ以外は静止しているという仮定が置かれている．

一方，本研究では，ボールを持っていないプレイヤーの行動決定を対象としている．この場合，プレイヤーが行う行動としては，ボールを持っている相手プレイヤーへのタックル，相手のパスのインターセプト，味方からのパスを受けるためのレシーバとしての移動などが考えられる．これらの中で，最も連係プレーなどの協調行動に関係するのはレシーバとしての移動であろう．そこで，ルートノード（現局面）から行う深さ 1 の行動を自身の現在地から他の場所への移動に限定する．その際，他のプレイヤーとボールは静止していると仮定する．深さ 2 の行動としては，自身のさらなる移動やレシーブ後のパスやシュート，チームメイトの行動など様々な可能性が考えられる．今回は計算量の観点から探索木の深さは 1 にとどめ，1 段階の移動行動だけを考えることにする（図 2）．

図 2 において s_0 が現在の局面，a_1~a_4 がそれぞれ移動先候補 1~4 への移動行動，s_1~s_4 が移動後の予測局面である．また，各ノードの右下の数字は局面の評価値である．図 2 の例では s_4 の評価値が最も大きいため，プレイヤーは移動先候補 4 を目指して移動する．以後，チェーンアクションを用いて移動先の探索を行っているプレイヤーを“探索プレイヤー”と呼ぶ．

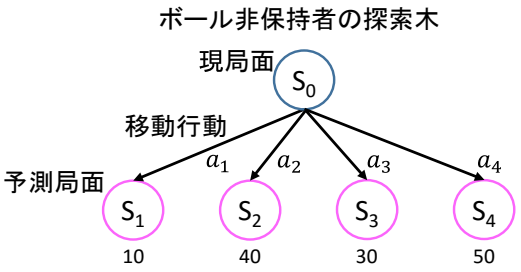

図 2　ボール非保持者が行動決定に用いる探索木
Figure 2　A search tree used for determining an action of an agent that is not holding the ball.

4.2 移動行動と予測局面

プレイヤーが体の向きを変える turn コマンドとプレイヤーが前進するための dash コマンドを用いて，探索木の枝に相当する“移動行動”を定義する．今回は，0~1 回以上の turn コマンドとその直後の 1 回以上の dash コマンドの組を移動行動 a と定義した．

探索プレイヤーは動的な移動先探索によってホームポジションにとらわれず自由に位置取りをさせるが，ある程度は agent2d 本来のフォーメーションを維持することとする．そこで，次のような移動先候補の生成ルールを考案した．

【移動先の候補地点】
- 探索プレイヤーの周囲 24 方向，各方向に対して 2n 回 (n=0,…,10) の dash コマンドで到達する地点

ただし，次の条件 1～4 を満たす移動先候補は取り除く．

【移動候補から除外するための条件】
- x 座標が-51.5 以下または 51.5 以上，もしくは y 座標が-33.0 以下または 33.0 以上の地点[a]（条件 1）
- オフサイドラインを越える地点（条件 2）
- ボールに極めて近い地点やボールを跨いだ先の地点

a) 仮想的なサッカーフィールドは中央を原点とする．原点から相手ゴールへ向かう方向を x 軸の正方向，x 軸と直交して右へ向かう方向を y 軸の正方向とする．また，フィールドのサイズは，68[m]×105[m]である．

（条件 3）

- 探索プレイヤーのホームポジションから 10m 以上離れる地点（条件 4）

上記の条件 1～3 により除外された候補地点の例を図 3 に，条件 4 により除外された候補地点の例を図 4 に示す．

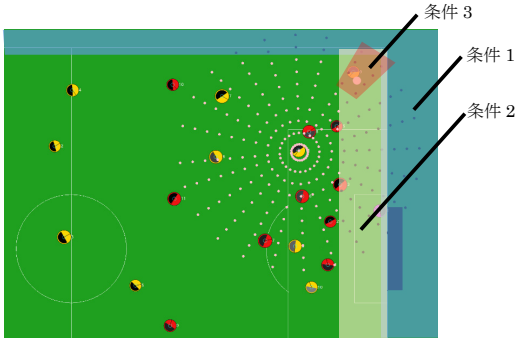

図 3　移動先候補の除外の例（条件 1～3 の場合）

Figure 3　Removing candidates for target positions of a receiver's moving (case 1~case 3).

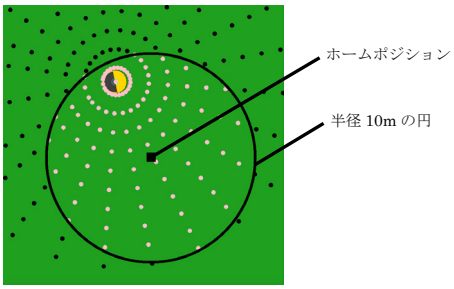

図 4　移動先候補の除外の例（条件 4 の場合）

Figure 4　Removing candidates for target positions of a receiver's moving (case 4).

4.3 局面評価関数の設計

図 2 の探索木のノードは，各移動行動により生成される予測局面を表している．探索木を用いた行動決定を行うには，この予測局面において探索プレイヤーが属するチームがどの程度優勢であるかを数量化する必要がある．これまで，探索プレイヤーがボール保持者である agent2d では，ボールの位置座標だけに依存する評価関数を採用していた．これは単にボールが相手ゴールに近ければ自分のチームは優勢であるという極めてシンプルな評価であった[b]．また，田川らはヒューリスティクスに基づいたより複雑な局面評価関数を用いていた[3][7][8].

しかし，いずれも決定すべき対象となる行動がパスやドリブルなどのボールを移動させる行動であった．したがって，ボールの移動により生ずる評価値の変動を陽に表現する評価項目が用いられていた．例えば，ボール保持者と相手チームのプレイヤーとの距離，ボール保持者から見て相手ゴール側にいる相手チームと味方チームのプレイヤーの人数比，ボールと両ゴールの距離，ボールと両チームのプレイヤーの距離，ボール周辺の敵味方のプレイヤー分布などである．

ところが，本研究で考えている探索プレイヤーの行動は探索プレイヤー自身の移動であり，ボールや他のプレイヤーは静止していると仮定している．したがって，移動行動の評価には上記のようなボールの位置座標やボール周辺のプレイヤーの分布などに関する評価は適していない．むしろ，移動後の探索プレイヤー自身の位置がレシーバとしていかに適切であるかどうかを直接的に評価した方が良いであろう．そこで，次の 7 つの評価項の線形和で表される局面評価関数を考えた．

$$E_s(s;\omega)=\sum_{i=1}^{7}\omega_i U_i(s) \qquad (1)$$

ただし，s は探索プレイヤーが移動した後の評価対象となる局面，$U_i(s)$($\in$[-10,10])は評価項，ω_i($\geqq$0)は$U_i(s)$の重みを表している．評価項の内容を表 1 に示す．

表 1　評価項目$U_i(s)$

Table 1 Evaluation terms $U_i(s)$

	評価内容
$U_1(s)$	探索プレイヤーと相手ゴールの間の x 方向の距離
$U_2(s)$	探索プレイヤーと相手/味方ゴールの間の直線距離
$U_3(s)$	パサーからのパスの通りやすさ
$U_4(s)$	味方プレイヤーからのパスの通りやすさ
$U_5(s)$	味方プレイヤーへのパスの通りやすさ
$U_6(s)$	スルーパス受け取り後の相手ゴールまでの距離
$U_7(s)$	この地点からのシュートコースの広さ

本研究では，味方がボールを得たときにフォワード(FW)やミッドフィルダー(MF)が的確にパス回しを行い，パス回しからの高い得点力を実現することを評価関数の基本的な設計方針とした．表 1 のU_1~U_2は探索プレイヤーが相手ゴールへ近づく（攻める）ことを評価する項目であり，U_3~U_6は探索プレイヤーを中心としたパスの通りやすさを評価した項目，U_7は探索プレイヤーがシュート可能かどうかを評価した項目である．3 種類とも味方ボール時の攻撃面における評価項目である．以下，これらの説明を簡単に述べる．

b) agent2d での評価値は，*point=ball_pos_X*+max(0,40-*dist_from_ball_to_goal*)で与えられる．ただし，*ball_pos_X* はボールの x 座標，*dist_from_ball_to_goal* はボールと相手ゴール中央との間の直線距離である．ただし，ボールがフィールド外や味方ゴール内にある場合は極端に大きな負の値が，シュート可能位置や敵ゴール内にある場合は極端に大きな正の値が加算される．

なお，各評価項の具体的な表式は付録に記載した．

U_1は探索プレイヤーのx座標を評価する．ボール非保持者が積極的に相手ゴール側へ移動することで，攻めが継続される．U_2は探索プレイヤーと相手ゴール間の距離と，探索プレイヤーと味方ゴール間の距離の 2 つの距離を評価する．相手ゴールとの距離が短く，味方ゴールとの距離が長いほど高く評価する．U_1と似ているが，y 座標も評価に含める点が異なる．

U_3はボール保持者(パサー)が，探索プレイヤーの足元へパスを送る場合の通りやすさを評価する．パスが敵に奪われにくいほど高く評価する．U_4では各味方プレイヤーがボールを持っていると仮定し，それぞれのプレイヤーから探索プレイヤーの足元へパスを送る場合を考えた時の，パスの通りやすさの合計値を計算する．パスが通る人数が多く，パスが通りやすいほど評価値が高い．U_5では探索プレイヤーから各味方プレイヤーの足元へパスを送る場合を考え，各パスの通りやすさの合計値を計算する．パスが通る人数が多く，パスが通りやすいほど評価値が高い．U_4とU_5を考えたのは，パスコースを複数確保することが素早く的確なパス回しの実現につながると期待したからである．U_6では探索プレイヤーと相手ゴール間の距離，スルーパス受け取り予測地点と相手ゴール間の距離の 2 つの距離を評価する．パッサーがスルーパスを出すことができる位置に探索プレイヤーがいる場合にのみ評価を行い，その他の場合では最低評価値(-10)を返す．スルーパスを受け取る地点と相手ゴールとの間の距離が短いほど評価値が高い．U_6を考えたのは，足元へのパスによるパス回しだけでなく，スルーパスを用いた素早い攻めも高く評価したいからである．

U_7はシュートコースの広さを評価する．探索プレイヤーの足元にボールがあると仮定して，そこからの相手ゴールへのシュートコースの角度が広いほど評価値が高い．この項を考えたのは，レシーバがシュートしやすい場所へ移動することで，パス回しからのシュートにより得点力が高まると考えたからである．

5. 局面評価関数の学習

5.1 学習則

学習には方策勾配法[11][12]を用いた．まず，学習するエピソードを定義し，エピソード終了後にその時点での局面やエピソード全体を評価して報酬を与える[c]．次に，エピソードあたりの報酬の期待値を極大化するために，確率的勾配法を用いて(1)の各ω_iを更新する．すなわち，方策勾配法による学習則は次のように表される．

$$\Delta\omega = \varepsilon \cdot r \sum_{t=0}^{L-1} e_\omega(t) \qquad (2)$$

$$e_\omega(t) \equiv \partial \ln \pi(a(t)|s(t);\omega)/\partial\omega \qquad (3)$$

ここで，rは報酬，$s(t)$は時刻tにおける局面，$a(t)$は時刻tで選択した行動，Lはエピソード長，$\varepsilon(>0)$は学習係数である．

また，(3)の$\pi(a)$は確率的な方策である．状態$s(t)$において行動 a を実行した遷移先の状態 s_a を(1)の局面評価関数$E_s(s;\omega)$により評価する．この局面評価値を目的関数とする次の Boltzmann 分布関数を方策$\pi(a)$として用いる[12]．

$$\pi(a|s(t);\omega) \equiv e^{E_s(s_a;\omega)/T} / \sum_x e^{E_s(s_x;\omega)/T} \qquad (4)$$

ただし，T(>0)は温度パラメータである．(4)を(3)へ代入すると，

$$e_\omega(t) = \frac{1}{T}\left[\frac{\partial}{\partial\omega}E_s(s_{a(t)};\omega) - \sum_x \pi(x|s(t);\omega)\frac{\partial}{\partial\omega}E_s(s_x;\omega)\right] \qquad (5)$$

となり，(1)を代入すると，最終的な学習則

$$\Delta\omega_i = \frac{\varepsilon r}{T}\sum_{t=0}^{L-1}\left[U_i(s_{a(t)}) - \sum_x \pi(x|s(t);\omega)U_i(s_x)\right] \qquad (6)$$

を得る．

5.2 学習システム

本研究では田川らが考案・開発した学習システム[3]を，ボール非保持者用に拡張して使用した．学習システムの概要を図 5 に示す．図 5 で試合観戦者はモニタを通して試合を観戦しながら投票画面を操作し，学習プログラムに報酬を与える（投票）．学習プログラムは探索プレイヤーの探索木に関する情報と与えられた報酬を基に(6)に従ってエピソードごとに重みを更新する．学習は試合中にリアルタイムで行われる．

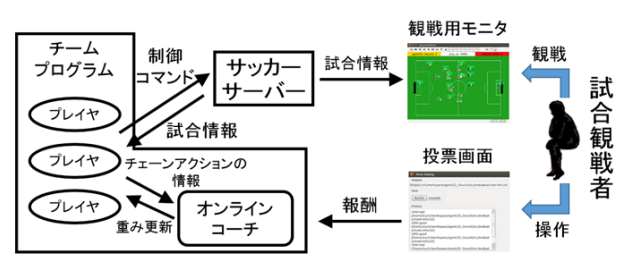

図 5　学習システムの概要

Figure 5 Overview of the learning system.

5.3 エピソード

本研究では学習対象の探索プレイヤーはボール保持者がパスを送った先のプレイヤー（レシーバ）である．また，5.1 で述べたようなエピソードごとの学習を行うので，エピソードを定義する必要がある．今，投票があった時刻から遡って，パスが出た時刻 t_{end} にレシーバがいた位置へのレ

c) エピソード終了時の状態だけではなく，エピソード内の状態・行動列を評価して報酬を与える場合でも適用可能なことが証明されている[13].

シーバの移動行動を a とする．この移動行動 a の開始時刻を t_{start} とすると，時刻 t_{start} から t_{end} までをエピソードと定義する．

図 6 にエピソードの例を示す．図 6 において，$t = t_6$で観戦者の投票があったとする．この時点から時間を遡って，$t = t_4$はパスが出た時刻である．この際，t_4の turn の後から t_5までの移動 b はパスされたボールへの反射的な追跡行動であるので，レシーバの位置取りのための移動行動としては扱わない．したがって，$t = t_1$からt_3までの移動行動 a が $t_1(= t_{start})$から$t_4(=t_{end})$までのエピソード内の移動行動として扱われる．今回は，強化の即時性を優先して，1 エピソード内での移動行動は 1 つに限った．この移動行動を決定したレシーバの方策が本学習の対象である．

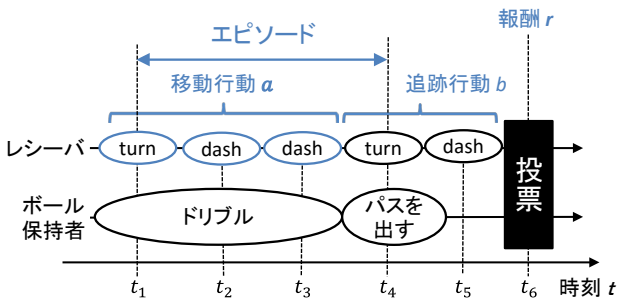

図 6　エピソードの例

Figure 6 Example of an episode.

5.4 報酬

観戦者が試合を観戦しながら学習チームのプレイヤーの行動を評価し，投票画面にある「Good」と「Bad」のボタンを押す．「Good」または「Bad」の投票 1 回ごとにそれぞれ報酬が 10 または-10 加算される．また，最初の投票後 10 サイクル以内であれば 2 回まで報酬を加算することができる[d]．さらに，学習チームの得点直前の移動行動には自動で報酬を 100 加算した．これは，得点直前の移動行動は得点につながった特に良い行動であると考えたからである．

6. 実験

6.1 学習実験

学習チームは agent2d-3.1.1 をベースにし，味方ボール時にボールを持っていないセンターフォワード(CF), サイドフォワード(SF), オフェンシブハーフ(OH)のプレイヤーがチェーンアクションを行うよう変更した[e]．学習対象となる探索プレイヤー数は，CF 1 人，SF 2 人，OH 2 人の計 5 人である．学習時の実験条件は以下の通りである：

- ポジション（役割）ごとに共通の重みωとする．
- 重みの初期値はすべて 1 とする．
- 温度 T=50，学習係数 ε =0. 1 と設定する．
- agent2d-3.1.1 を相手に 10 試合行う．
- 被験者 1 名が投票を行う．
- 相手チームがボールを保持している時や自陣内にいる時は agent2d-3.1.1 と同様に行動する[f].

表 2 に学習後のポジションごとの重み$\{\omega_i\}$を示す．表 2 から次のことが言える．CF はω_7の値がかなり大きくなっているので，シュートコースの広い地点へ移動してシュートを狙う動きを学習している．次にω_3の値が大きいことから，パサーからのパスの受け取りやすい位置へ移動しようとする．さらに，ω_6の値も比較的大きいので，相手ゴールに近い位置でスルーパスを受け取ろうとしている．したがって，積極的なストライカーとしての動きを学習したと解釈できる．

SF も CF と同様にω_7の値が最も大きく，ω_3の値も次に大きいのでストライカー的な傾向があるが，相手ゴール側への移動$(\omega_1$と$\omega_2)$と味方選手へパスを出しやすい位置への移動(ω_5)も考慮している．したがって，積極的にサイドから前進し，パサーからのパスを受取り，シュートをするか味方プレイヤーへパスを回すことに適した位置取りを学習したと解釈できる．

上記の CF と SF に対して，OH はω_7の値が大きくならず，シュートのできる位置へは行こうとはしない．ω_3, ω_4とω_5の値が大きいので，他の味方プレイヤーとのパスコースの確保を優先的に行う．したがって，前線からは一歩下がって，パス回しの中継的役割を担うような位置取りを学習したと解釈できる．

表 2　学習後の重み

Table 2 Weight parameters after learning.

ポジション	ω_1	ω_2	ω_3	ω_4	ω_5	ω_6	ω_7
CF	0.30	0.01	2.06	0.14	0.18	0.97	6.22
SF	1.23	1.71	1.96	0.00	1.77	0.76	4.50
OH	1.05	1.03	1.28	1.51	1.85	1.03	0.61

6.2 評価実験

学習チームの強さを確認するために agent2d-3.1.1 と対戦させる評価実験を行った．表 3 に評価実験の結果を示す．試合数はいずれも 1000 試合で，勝率の計算には引分け数は除いてある．

d)　サッカーシミュレータの 1 サイクルは実時間では 100ms である．

e)　ボールが原点（フィールド中心点）にあるときには，SF は CF の両側に，OH は CF の後方の両側に位置を取るように設定している．

f)　したがって，学習チームの守備力は agent2d-3.1.1 とほぼレベルである．

表 3　評価実験の結果
Table 3 Results of the evaluation experiments.

チーム	①勝-②負-③分	勝率 ①/(①＋②)	平均得失点
未学習	411-378-211	52.1%	2.23 - 2.16
学習	624-229-147	73.2%	3.35 – 2.22

表 3 から分かるように，重みがすべて 1 である未学習チームでも agetn2d-3.1.1 に対して，若干ではあるが勝ち越している．これは 4.で述べたボール非保持者の移動先決定方式，特に表 1 に示した評価項目$\{U_i\}$が妥当であることを示している．さらに，10 試合の学習により，評価項目の重み係数$\{\omega_i\}$が適切な値へと更新され，対 agent2d の勝率が 52.1%から 73.2%へ向上している．これは，平均失点には殆ど変化がなく，平均得点が 2.23 から 3.35 へと大きく増加していることから，学習により得点能力が向上したことによると考えられる．

さらに，学習後のチームの試合内容を観察すると，相手ゴール前でのパス回しの回数が格段に増えていることが見て取れた．そこで，対 agent2d-3.1.1 の 10 試合を観察し，得点シーンにおいて得点が入った直前のプレー内容を「パス回し」と「ドリブル」とに分類した．その割合を表 4 に示す．表 4 から，agent2d も未学習チームもドリブルからシュートを放ち得点するパターンが全体の約 5 割～6 割と非常に多い．それに対し，学習チームは 9 割以上がパスを回しながら得点を決めている．これは本研究で提案した学習方式により，パス回しのために適した位置取りが可能になったためであると考えられる．

表 4　対 agent2d との 10 試合における得点直前のプレーの割合(%)
Table 4 Plays before goal scoring in ten games against agent2d.

得点直前のプレー	agent2d	未学習チーム	学習チーム
パス回し	53.3	38.9	92.6
ドリブル	46.7	61.1	7.41

7. まとめ

本研究では，RoboCup サッカーシミュレーションリーグ 2D の試合において，チームメイトがボールを保持しているときにボールを持っていないプレイヤーが適切な移動先を決定する方式を提案した．本方式は agent2d で提案されているアクション連鎖探索フレームワークをベースとする．アクション連鎖探索フレームワークでは探索木と局面評価関数を用いるが，今回は新たにレシーバのための局面評価関数をヒューリスティクスに基づいて設計した．さらに，観戦者がプレイヤーの行動や試合局面の優劣を評価し，その結果を強化学習の報酬信号として利用することにより局面評価関数中の重み係数をリアルタイムに学習できるシステムを構築した．学習実験の結果は，10 試合の対戦だけでパス回しからの得点パターンが大幅に増加し，対 agent2d(ver.3.1.1)の勝率が 52.1%から 73.2%へと向上し，学習の有効性を確認することができた．

今後は学習回数や被験者数，対戦チームの種類を増やして学習実験を行い，提案手法の有効性をさらに調べる必要がある．報酬にも勝敗や得失点など客観的な評価も加えて学習にフィードバックさせることが望ましい．探索に関しては，レシーブ後のパスやシュートまでも考慮した深さ 2 以上の行動計画の立案や，ノード生成時におけるボール静止の仮定の見直しが考えられる．後者は，レシーバとパサーの行動計画においてノード生成時の仮定条件を同じくするということである．局面評価関数を同種のものに統一でき，連係プレー時の合意形成に役立つ可能性がある．また，最も勝率の向上に貢献する報酬の与え方の研究や，チームメイトがボールを保持していないときのプレイヤーの行動，すなわち，守備時の非ボール保持者の行動決定方式の研究も行いたいと考えている．

さらには，強化学習だけではなく，観戦者が具体的に指示した行動を教師信号とする教師有り学習も同じ枠組みに組み入れると，人間の与える教示内容を無駄なく有効にエージェントの行動決定へ反映することができる．このように強化学習と教師有り学習を同時併用して，リアルタイムで高速な学習が可能となることも目標の一つとしたい．

謝辞　本研究は JSPS 科研費（課題番号 26330419）の助成を受けた．謹んで感謝の意を表する．

参考文献

[1] Noda, I. and Matsubara, H.. Soccer Server and Researches on Multi-Agent Systems. Proc. of IROS-96 Workshop on RoboCup, 1996, p.1-7.
[2] “RoboCup Soccer Simulation League”. http://wiki.robocup.org/wiki/Soccer_Simulation_League
[3] 田川 諒，五十嵐治一．サッカーエージェントにおけるスルーパスの強化学習．第 15 回情報科学技術フォーラム (FIT2016)，2016，講演番号 6107.
[4] 秋山英久．アクション連鎖探索によるオンライン戦術プランニング．人工知能学会研究会資料，2011，SIG-Challenge-B101-6, p. 23-28.
[5] “agent2d ver3.1.1 の配布ページ”．http://sourceforge.jp/projects/rctools/releases/
[6] “GNU General Public License v.3.0”. http://www.gnu.org/licenses/gpl.html.
[7] 谷川俊策，五十嵐治一，石原聖司．RoboCup サッカーシミュレーションリーグ 2D における局面評価関数の学習．第 18 回ゲーム・プログラミング・ワークショップ 2013 予稿集，2013, p.106-109.
[8] 田川 諒，谷川俊策，五十嵐治一．agent2d のチェーンアクションにおける評価関数の重み調整．第 13 回情報科学技術

フォーラム講演論文集(FIT2014)，2014，第 2 分冊，p.285-288.
[9] “RoboCup tools”. https://osdn.jp/projects/rctools/releases/.
[10] 秋山英久．サッカーシミュレーションリーグ第 2 回，情報処理，2010, vol. 51, no. 10, p.1303-1316.
[11] R.J. Williams. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. Machine Learning, 1992, vol.8, p.229-256.
[12] 石原聖司，五十嵐治一．マルチエージェント系における行動学習への方策こう配法の適用―追跡問題―．電子情報通信学会論文誌 D-I, 2004, vol.J87-D1, no.3, p.390-397.
[13] 五十嵐治一，石原聖司，木村昌臣．非マルコフ決定過程における強化学習―特徴的適正度の統計的性質―．電子情報通信学会論文誌 D, 2007, vol.J90-D, no.9, p.2271-2280.

付録

表 1 の評価項目 $U_i(s)$(i=1,2,…,7) $(-10 \le U_i(s) \le 10)$の定義を以下に記す.

● U_1：探索プレイヤーの x 座標値が大きいほど高く評価.

$$U_1(s) = 20 \cdot \sigma(pos\_x) - 10 \qquad \text{(A.1)}$$

$$\sigma(x) = \frac{1}{1 + e^{-(x-\theta)/\tau}} \qquad (0 \le \sigma(x) \le 1) \qquad \text{(A.2)}$$

pos_x は x 座標値，θとτはそれぞれ 0 と 5.25 と設定した.

● U_2：探索プレイヤーと敵ゴールの距離と，探索プレイヤーと味方ゴールの距離とを評価する．敵ゴールとの距離が短く，味方ゴールとの距離が長いほど高く評価する． U_1とは y 座標も評価に含める点で異なる.

$$U_2(s) = 10 \cdot \{\sigma(dist_o) - \sigma(dist_m)\} \qquad \text{(A.3)}$$

dist_m は探索プレイヤーと味方ゴール中央との距離を，*dist_o* は探索プレイヤーと敵ゴール中央との距離を表す．また，θとτはそれぞれ 34 と 5 である.

● U_3：ボール保持者(*Passer*)が探索プレイヤーの足元へパスを送る場合の通りやすさを評価する．パスが敵に奪われにくいほど高く評価する.

$$U_3(s) = 20 \cdot \sigma\left(cycle_o - cycle_p\right) - 10 \qquad \text{(A.4)}$$

$cycle_p$ はボールが探索プレイヤーの足元へ届くまでの予測サイクルを，$cycle_o$ はそのパスを敵が拾う場合の最短サイクルを表す．また，$\sigma(x)$のθとτはそれぞれ-3 と 0.8 である.

● U_4：各味方プレイヤーがボールを持っていると仮定し，探索プレイヤーの足元へパスを送る場合にパスが通る人数と各パスの通りやすさを評価する.

$$U_4(s) = 20 \cdot \sigma\left(\sum_{i=1}^{num} pass\_quality(i)\right) - 10 \qquad \text{(A.5)}$$

pass_quality(*i*)は味方プレイヤーi から探索プレイヤーの足元へパスを送る場合のパスの通りやすさを表す．値は 0（通らない）,0.8（通りにくい）,1（通りやすい）,1.2（通る）の 4 段階である[g]．*Num* はパスを送る味方プレイヤーの人数を表し，本研究では agent2d-3.1.1 のポジション（ロール）におけるセンターフォワード(CF)1 人，サイドフォワード(SF)2 人，オフェンシブハーフ(OH)2 人の計 5 人を考慮する．$\sigma(x)$の θとτはそれぞれ 1.2 と 1 である.

● U_5：探索プレイヤーがボールを持っていると仮定し，探索プレイヤーから味方プレイヤーの足元へパスを送る場合に，パスが通る人数と各パスの通りやすさを評価する.

$$U_5(s) = 20\sigma\left(\sum_{i=1}^{num} pass\_quality(i) \cdot role\_bonus\right) - 10 \qquad \text{(A.6)}$$

pass_quality(*i*)は探索プレイヤーから味方プレイヤーi へパスを送ると仮定した場合の通りやすさを示す．*num* はパスが送られる味方プレイヤーの人数を表し，U_4と同様に CF1 人，SF2 人，OH2 人の 5 人を考慮した．また，プレイヤーのロールごとに *role_bonus* は異なり，CF と SF は 1.5，OH は 1 に設定し，$\sigma(x)$のθとτはそれぞれ 1.5 と 1 と設定した.

● U_6：探索プレイヤーと敵ゴールの距離，スルーパス受け取り予測地点と敵ゴールの距離の 2 つの距離を評価する．パサーがスルーパスを出すことができる位置に探索プレイヤーがいる場合にのみ評価を行い，その他の場合では最低評価値(-10)を返す．スルーパス受け取り予測地点と敵ゴールの距離が短いほど評価値が高い.

$$U_6(s) = 20 \cdot \sigma(dist\_current - dist\_through) - 10 \qquad \text{(A.7)}$$

dist_current は現局面における探索プレイヤーと敵ゴールの距離，*dist_through* はスルーパス受け取り予測位置と敵ゴールの距離．$\sigma(x)$のθとτはそれぞれ 10 と 3.5 と設定した.

● U_7：シュートコースの広さを評価する．探索プレイヤーの位置からのシュートコースの角度が広いほど評価値が高い．また，探索プレイヤーと敵ゴールが 17m 以上離れていれば，シュート不可能と判断して最低評価値-10 を返す.

$$U_7(s) = 20 \cdot \sigma(shoot\_margin - 15) - 10 \qquad \text{(A.8)}$$

shoot_margin はシュートコースの角度で，シュートコースが複数ある場合には最も広い角度を採用する．$\sigma(x)$のθとτはそれぞれ 1 と 2 と設定した.

g) agent2d では味方／敵プレイヤーがボールへ追いつく最短時間を比較してパスが通るかどうか(0/1)の 2 値の判定をしていたのを修正して用いた