RoboCup サッカー 2D リーグの PK モードの学習環境構築と強化学習

奈良女子大学　理学部　情報科学科　4 回生
05251580　新出研究室　上野 範子

概要

我々は RoboCup サッカーシミュレーションを使用し PK を学習することが可能な環境を構築し，キッカの強化学習を行った．本論文では我々の構築した学習環境およびそれによる学習の結果について述べる．

はじめに本研究の目的，RoboCup サッカーについて，そして RoboCup サッカーシミュレーションでの競技内容やシミュレーションの仕組みについてを簡単に説明する．次に，具体的な PK モードの学習環境の構築について説明する．今回，PK モードを作成するにあたって，学習をするためにトレーナエージェントを使用した．現在の環境では既存のトレーナエージェントやプレイヤのプログラムは PK の学習に使用できないため、ともに新しく実装することにした．なお，簡単にするため，左チームは 4 人のプレイヤ (うち 1 人はゴールキーパー) を，右チームはゴールキーパー 1 人で，左チームのプレイヤ 3 人の PK の学習をするような実装にした．学習としては強化学習を用い，学習の手法としては Q 学習を用いた．

1　はじめに

自律エージェント分野のよく知られた課題として，RoboCup サッカー [1] がある．RoboCup にはサッカーをはじめ，以下のような競技がある．

- サッカー
  シミュレーション，小型，中型，四足，ヒューマノイド
- レスキュー
  シミュレーション，実機
- ジュニア
- @Home

我々はサッカーシミュレーションにおける PK の学習を行った．RoboCup サッカーはフォーメーションなどの学習はこれまでにも広く行われているが，PK の学習はこれまでにほとんど手をつけられていない状態であった．PK も試合を左右する要素である．よって，PK についても学習を進めることは重要と考えられる．しかし学習を進めるには現在の環境では PK だけを実行することができないという様々な問題があった．

そこで本研究は PK の学習に必要な環境を構築するとともに，機械学習を行った．学習には強化学習を用い，学習の手法としては Q 学習を用いた．

2　RoboCup サッカーについて

2.1　競技内容

RoboCup サッカー 2D リーグについて簡単に説明する．

2D リーグは高さの概念がない為，物体は平面上を滑るように移動する．しかし，それ以外は実際人間がプレイする様な 11 対 11 のサッカーを再現している．古くから存在しているだけあって，チームプレイの完成度は RoboCup 全競技の中でも群を抜いている．本研究はこの 2D リーグを使用した．

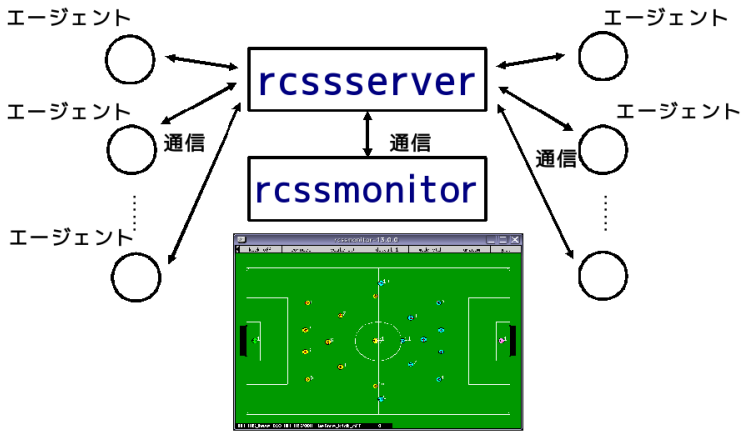

図 1　シミュレーション実行時のプログラムの関係図.

2.2　シミュレータの仕組み

次に，RoboCup サッカーシミュレータについて説明する．

RoboCup サッカーシミュレータとは，その名の通りコンピュータ上でサッカーシミュレーションを実行するためのソフトウェアのことである．英語では The RoboCup Soccer Simulator と書き，これ以降は RCSoccerSim と略すことにする．シミュレータは複数のプログラムを連携させてシミュレーション環境を提供する統合システムとなっている．RCSoccerSim と呼ぶ場合，主に以下のプログラムパッケージが含まれる．

- rcssbase
  RCSoccerSim に含まれる各プログラムが使用する基本ライブラリ
- rcssserver
  シミュレータ本体
- rcssmonitor
  画面表示プログラム

サッカーシミュレータにおいて，実際のシミュレーションを担当するのはサーバープログラムである rcssserver というプログラムである．シミュレータはプログラムをサーバーとクライアントに分け，それぞれのプロセスで役割分担をして処理を行うサーバークライアント方式による分散マルチエージェントシミュレーションを実現している．サーバーは情報を集中管理し，クライアントはその情報を利用するといったネットワークプログラムにおいては極めて一般的なモデルとなっている．シミュレーション実行時の各プログラムの関係は図 1 のようになる．

サッカーエージェントは知覚情報となるメッセージをサーバーから受けとり，それに応じて自分の行動コマンドのメッセージをサーバーに送信し，この送受信によりフィールド上の物体の状態が変化し，シミュレーションが進行する仕組みになっている．さらに，エージェントは個々に独立して通信を行っている．これはサッカーエージェントは全て独立して制御されていることを意味する．公式競技では 1 つのクライアントプログラムが制御可能なのは 1 つのエージェントのみと規定されており，エージェント間の情報共有は rcssserver を介したコミュニケーションでしか許可されていない．このようにして，完全な分散マルチエージェントシミュレーションが実現されている．

2.2.1　フィールドの座標系

rcssserver 内部で使用される座標系は左手座標系である．rcssmonitor でフィールドを表示した場合，フィールド中央を原点とし，右が X 軸正方向，下が Y 軸正方向となる．よって，図で表すとフィールドの座標系は図 2 のようになっている．

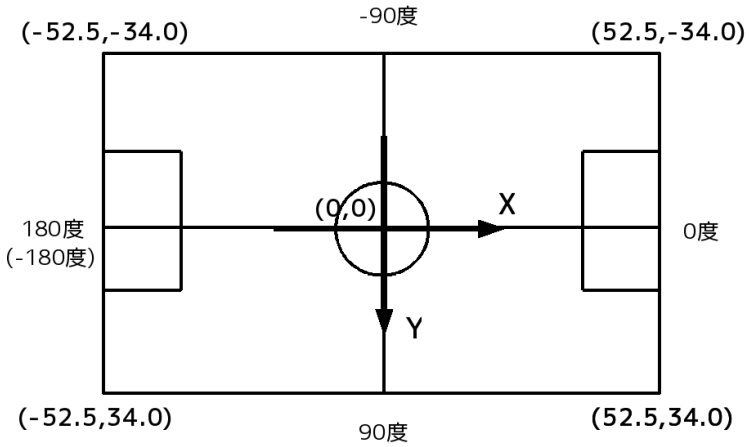

図 2　rcssserver の座標系.

プレイヤエージェントの為の座標系は左サイドのチームは rcssserver 内部の座標系と全く同じで，右サイドのチームは全て反転して使用する.

3　PK モード作成と強化学習

ここでは，我々が PK の学習を行うために構築した環境について述べる.

3.1　チーム開発について

今回の実装にあたって，http://rctools.sourceforge.jp/から以下のものを入手した.

- librcsc-2.0.1
  サッカーシミュレーションのチームプログラムやツールプログラムの開発に使用する基本ライブラリ
- agent2d-1.0.0
  チームとして最小限動く状態にまとめたプログラムソース
  チーム開発を開始する際のテンプレートとして使用する

さらに，http://sourceforge.net/projects/sserver/ から以下のものを同様に入手した.

- rcssbase-12.1.2
- rcssserver-13.0.0
- rcssmonitor-13.0.0

これらについては 2.2 に述べた通りである.
我々は agent2d を編集し，PK モードの学習環境を構築した．これについては以下に述べる.

3.2　トレーナエージェントについて

トレーナエージェントとは，審判同様に試合を制御することが可能なエージェントのことである．主な特徴として，以下のようなものがあげられる.

- 試合前の訓練や実験の為に利用される
- 機械学習を行う為のエピソード繰り返し実行の制御が可能
- プレイヤエージェントの行動内容の評価を自動化することが可能
- 実際の試合には使用不可

今回は学習するために agent2d に添付されていた既存のサンプルのトレーナを利用しようとしたが，これは PK に使用すると内部の整合性を保つことができなくなるため，サーバー組込みの PenaltyKickState コマンドが使用できないという問題が起こった．このため，トレーナは新たに実装した．実装は，agent2d のサンプルトレーナがテンプレートとして用意しているアクション関数 sampleAction を，図 3 のものに置き換えることで行った．なお，プレイヤ，ボールの初期位置は以下のようにした．トレーナは全て左手座標系で表す．背番号が 1 のプレイヤはゴールキーパーを表している．

- uniform number : 1 (right team)
  $(51.5, 0.0)$
- uniform number : 1 (left team)
  $(50.5, 25.0)$
- uniform number : 2 (left team)
  $(32.0, 0.0)$
- uniform number : 3 (left team)
  $(31.0, 0.0)$
- uniform number : 4 (left team)
  $(30.0, 0.0)$
- ball
  $(41.5, 0.0)$

3.3 PK モードの学習環境構築

agent2d に添付されている既存の PK モードでは，サーバー組込みの PenaltyKickState コマンドを使用していたため，PK を行うフィールドが左右どちらになるかはランダムであった．今回の実装では，右のフィールドで PK を行うように設定した．

また，プレイヤのプログラムにゲームモードが PlayOn になるとこの新しく実装した PK モードのプログラムが動くように設定した．

さらに，rcssmonitor を起動する際に左チーム 4 人のプレイヤ (うちゴールキーパーが 1 人)，右チーム 1 人のプレイヤ (ゴールキーパー) が PK に参加するようにした．今回は左チームのプレイヤ 3 人の学習を実装した．

実装した PK モードの画面は図 4 のようになる．

PK モードのプレイヤの実装は，図 5 に示す．ゲームモードが PlayOn のとき，関数 execute が呼ばれる．

3.4 強化学習

3.4.1 強化学習について

強化学習 [2] とは，目標指向型の学習と意志決定を理解する為の計算的アプローチである．正しい行動を直接与えて教示するのではなく，実行した行動の評価を訓練情報として利用するという特徴を持っている．強化学習では，学習と意志決定を行う者を「エージェント (agent)」，エージェントが相互作用を行う対象を「環境」，環境から発生するものを「報酬」と呼ぶ．報酬は数値で表される．エージェントは時間の経過の中でこの報酬を最大化することを目標とする．各時間ステップ t において，エージェントは何らかの「状態 (state)」の表現 $s_t \in \mathcal{S}$($\mathcal{S}$ は可能な状態の集合) を受け取り，これに基づいて「行動」$a_t \in \mathcal{A}(s_t)$ を選択する ($\mathcal{A}(s_t)$ は状態 s_t において選択することが可能な行動の集合)．1 時間ステップ後に，エージェントはその行動の結果として数値化された報酬 $r_{t+1} \in \mathbb{R}$ を受け取り，新しい状態 s_{t+1} にいることを知る．状態と行動から，その行動を取る確率への関数のことを「方策 (policy)」と呼び，π_t と表す．図 6 にエージェントと環境との相互作用を示す．

sampleAction

状態を表す state を 0, 蹴る順番を表す kick_number を 2 とする

if 2 チームが参加する場合

ボールをペナルティマークへ移動させる

ゲームモードを BeforeKickOff から PlayOn に変える

if ゲームモードが AfterGoal_に変わる場合

if ゲームモードが FreeKick_に変わる場合

if ある一定のサイクルが経過する場合

のいずれかが起きると state を 1 にする

if kick-number がプレイヤの人数以上になる場合

kick_number を 2 に初期化する

switch(state):

state が 0

トレーナがプレイヤに kick_number を言う

state が 1

プレイヤ全員のスタミナを回復させる

ゲームモードを PlayOn に変える

プレイヤ，ボールを初期位置に移動させる

kick_number を 1 増やす

state を 0 に初期化する

break;

図 3　トレーナエージェントのアルゴリズム.

図 4　実装後の PK モードの実行画面.

```
execute
    if ゴールキーパーである場合
        if 右チームである場合
            if ボールの速度が 0.5 より大きい場合
                return doGoalie;     /*ボールを追いかける*/
            else return doGoalieSetup;     /*右チームのゴールキーパーはゴール前に移動*/
        else return doGoalieWait;     /*左チームのゴールキーパーは待機*/
    if トレーナが言った背番号のプレイヤである場合
        return doKicker;     /*ボールを蹴る*/
    else return doKickerWait;     /*初期位置に待機*/

doKickerWait
    if 初期位置にいる場合
        ボールの方へ首を向ける
    else 初期位置に移動する
    return true;

doKicker
    if ボールが勝手に動いていた場合
        return false;
    if ある一定のサイクルより小さい場合
        ボールの近くに移動する
        return true;
    else if ボールとの距離が十分に近い場合
        シュートする
        return true;
    else ボールの距離が十分に近くなるまで移動する
        return true;

doGoalieWait
    if 初期位置にいる場合
        ボールの方へ首を向ける
    else 初期位置に移動する
    return true;

doGoalieSetup
    初期位置 (ゴールの前) に移動する
    体や首の向きを変える
    return true;

doGoalie 以下は既存の PK と同様.
```

図 5　PK モードのプレイヤのアルゴリズム.

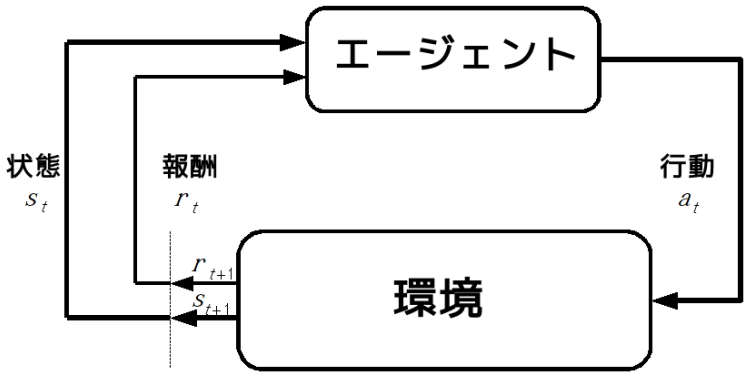

図 6　強化学習におけるエージェントと環境間の相互作用.

$Q(s,a)$ を任意に初期化
各エピソードに対して繰り返し:
　s を初期化
　エピソードの各ステップに対して繰り返し:
　　Q から導かれる方策 (例えば Q に対する ϵ グリーディ方策) を使って，s での行動 a を選択する
　　行動 a を取り，r, s' を観測する
　　$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$
　　$s \leftarrow s'$;
　s が終端状態ならば繰り返しを終了

図 7　Q 学習: 方策オフ型 TD 制御アルゴリズム.

3.4.2　Q 学習について

今回の実装では Q 学習を使用した．Q 学習とは，経験から直接学習することが可能であり，最終結果を待たずに他の推定値の学習結果を一部利用し，推定値を更新する TD 学習 (時間的差分学習;Temporal Difference Larning, 以下 TD 学習と呼ぶ) の 1 つである．

Q 学習は方策オフ型手法である．方策オン型手法では，方策を制御に用いる一方で，方策の価値を推定するのに対し，方策オフ型手法はこれら 2 つの機能を分離しているのが特徴である．

最も簡単な形式は 1 ステップ Q 学習と呼ばれ，次の様に定義される．

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1},a) - Q(s_t,a_t)]$$

アルゴリズムは図 7 に示す．我々は方策として ϵ グリーディ手法を用いた．

3.4.3　ϵ グリーディについて

グリーディ (貪欲;greedy) な行動とは，価値の推定値を最大とする様な行動のことである．ϵ グリーディ手法とは，ほとんどいつも貪欲に振る舞うが，たまに小さい確率 ϵ で，行動価値推定量とは無関係に，一様に任意の行動を選ぶ様な方法のことである．ϵ グリーディ手法の利点は，プレイの数を増やし，極限に至るならば，全ての行動が無限回試され，したがって全ての a に対して k_a(行動 a に対する回数) $\rightarrow \infty$ を保証している点である．

Q 学習は全ての行動が試されれば行動価値関数 $Q_t(s,a)$ が最適行動価値関数 $Q^*(s,a)$ に収束し，結果として ϵ グリーディ手法なら収束が保証される．

3.4.4 学習の実装

PK を学習するにあたり，プレイヤの行動は，初期位置からある地点への移動と，そこの地点からゴールのどこかを狙ってのシュートの 2 種類に限った．移動する地点としては図 8 のように，ある範囲を 3×4 の格子状に分割した地点から選択することにした．この範囲に限定した理由は，サッカー選手が実際に蹴るときに選ぶ場所がこの範囲に含まれていることが知られているからである．また，左右対称になるため Y 座標が正の値の部分である点線で囲まれた範囲は省略した．座標は以下のようにした．

- $(37.0, -1.0)$
- $(37.0, -0.5)$
- $(37.0, 0.0)$
- $(38.0, -1.0)$
- $(38.0, -0.5)$
- $(38.0, 0.0)$
- $(39.0, -1.0)$
- $(39.0, -0.5)$
- $(39.0, 0.0)$
- $(40.0, -1.0)$
- $(40.0, -0.5)$
- $(40.5, 0.0)$

また，シュートで狙う位置は図 8 に見られるように，ゴールを一定間隔で区切った地点の 1 つを選択することにした．X 座標は 52.5 とし，Y 座標は以下のようにした．rcsc::ServerParam::i().goalHalfWidth() は組込みのパラメータで，ゴールの幅を表している．

- $y =$ rcsc::ServerParam::i().goalHalfWidth()
- $y =$ rcsc::ServerParam::i().goalHalfWidth()−1.0
- $y =$ (rcsc::ServerParam::i().goalHalfWidth())/2
- $y =$ (rcsc::ServerParam::i().goalHalfWidth()−1.0)/2
- $y = 0.0$
- $y = -$(rcsc::ServerParam::i().goalHalfWidth()−1.0)/2
- $y = -$(rcsc::ServerParam::i().goalHalfWidth())/2
- $y = -$(rcsc::ServerParam::i().goalHalfWidth()−1.0)
- $y = -$(rcsc::ServerParam::i().goalHalfWidth())

移動する場所は青い × 印，シュートする場所は黄色の × 印で示した．

報酬はゴールが決まったときに 1 を与え，それ以外は 0 を与えるようにした．

図 9 は Q 学習を使用し，新しく実装した PK モードの強化学習のアルゴリズムである．なお，行動価値関数 $Q(s, a)$ の γ(割引率: $0 \leq \gamma \leq 1$) は 1 とし，α (ステップサイズ・パラメータ: $0 < \alpha \leq 1$) は $\frac{1}{k+1}$ とした (k はその状態でその行動が選択された回数)．

ここで，割引率とは将来の報酬が現在においてどれだけの価値があるのかを決定するものである．$\gamma = 0$ ならば，エージェントは即時報酬 (immediate reward) のみに関心を持つという意味で近視眼的であり，γ は 1 に近いほど将来報酬を重視する．今回はエピソード的タスクなので，将来の報酬の割引を行う必要のないため，$\gamma = 1$ とした．ステップサイズ・パラメータは，行動 a に対する k 番目の報酬を扱う場合に用いられる．これを定数とすると振動し，真の値が変わったときに常に新しい値に影響されやすくなる．今回 $\alpha = \frac{1}{k+1}$ としたことにより，行動価値関数の推定としては単にそれまでの報酬の単純平均を用いることになる．

また，スコアが更新されるときには既にキッカの順番が変わっているので，新しく作成したプログラムでは移動する行動の前に，以前のスコアと現在のスコアを比較し，現在のスコアが大きければ報酬を 1 与え，行動

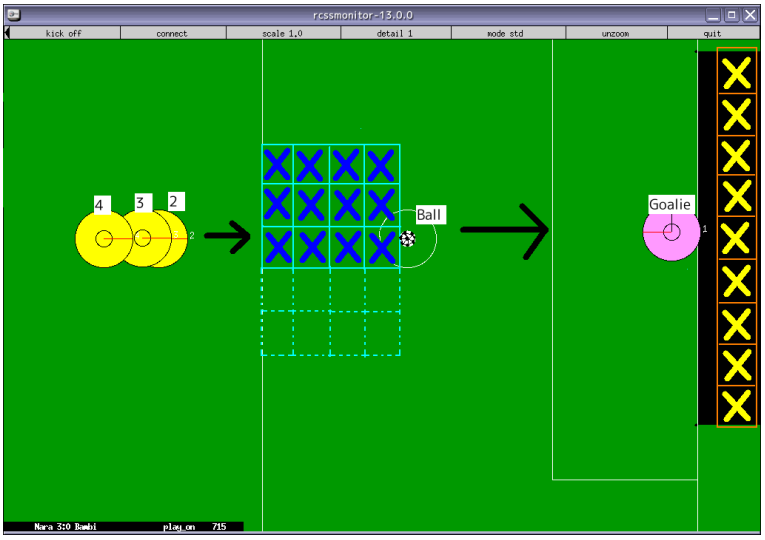

図 8　移動場所とシュートの場所.

行動価値関数を初期化する
s = 初期状態
繰り返し:
　ϵ グリーディ手法を使い，移動する行動を 1 つ選択してそれを a とする
　a で行動をする
　報酬を 0 与える
　s' = 次の状態
　s' での行動のうち価値が最大となるものを選択する
　その行動での価値を計算し，s での行動 a の価値を更新する
　$s = s'$

　ϵ グリーディ手法を使い，シュートをする行動を 1 つ選択してそれを a' とする
　a' でシュートする
　if ゴールが決まる場合
　　報酬を 1 与える
　else 報酬を 0 与える
　s での行動 a' の価値を更新する　　/*エピソードタスクのため，次の状態での行動価値は計算しない*/
　s = 初期状態
ゲームモードが TimeOver になると終了

図 9　PK モードの強化学習.

価値関数を計算するように実装した.

3.4.5　考察

図 10 は，ϵ を 0, 0.1, 0.01 にしたときのシュートの回数と，$\frac{\text{スコア}}{\text{シュートの回数}}$ の割合を出したものである.

また，表 1 はそれぞれの ϵ に対し，最初の 500 回蹴った回数と最後の 500 回でのスコアとシュートの回数の比を計算した結果を示したものである.

グラフから，$\epsilon = 0.1$ の場合はより早く最適行動を見出すことが可能であることがわかる．しかし，ϵ を小さ

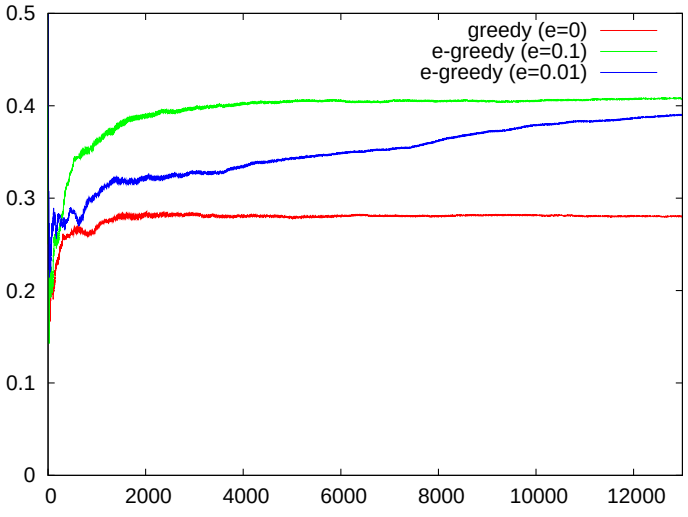

図 10　$\epsilon = 0, 0.1, 0.01$ のときのシュートの回数と $\frac{\text{スコア}}{\text{シュートの回数}}$ の割合.

表 1　それぞれの 500 回蹴ったときのスコアとシュートの回数の比.

ϵ の値	最初の 500 回	最後の 500 回
ϵ=0	0.276	0.292
ϵ=0.1	0.334	0.410
ϵ=0.01	0.284	0.440

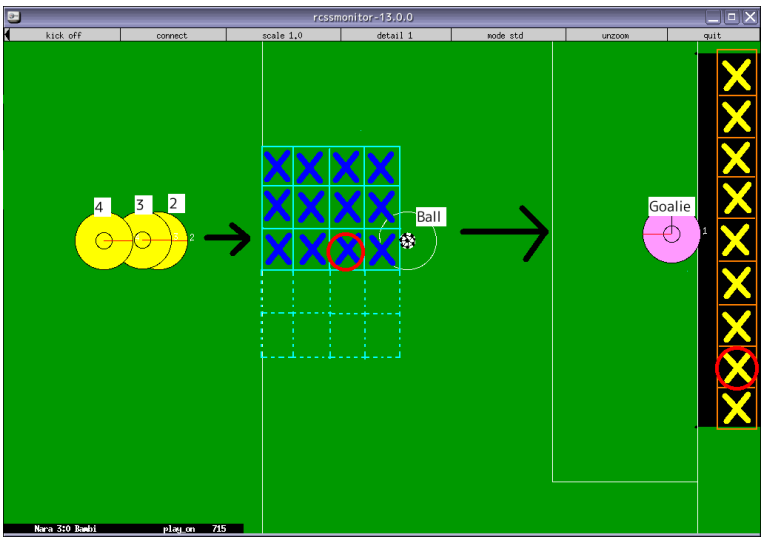

図 11　よく選択された行動.

くしていくにつれ，比較的ゆっくりと改善が行われるが，性能評価尺度において，最終的には $\epsilon = 0.1$ の場合よりも高い能力を示していることが表 1 からわかる.

実験の結果，もっとも推定価値の良かった行動は図 11 に ◯ 印で示す．シュートで狙う最良の位置がゴールの端ではないことが注目される．これは，シミュレーションでは特定の場所を狙って蹴ってもノイズのために少し外れることがあるので，端を狙うと外す確率が増えるためであると考えられる.

4 まとめ

本研究では，新たに PK モードを作成し，キッカを学習することが可能となった．また，この PK モードは PK 以外にも試合の中で利用するシュートの強化学習も可能である．強化学習を行うことにより，どの場所でどこにシュートをすれば PK の試合において有利になることがわかった．

今後の展望として，現在は蹴る位置とシュートで狙う位置の刻みがまだ粗いが，刻みをもっと細かくすることでよりよい行動を得られる可能性がある．また，今回学習後の 1 回あたりの平均スコアが 0.4 程度に留まったのは，学習で選択する行動が偏ったためである可能性もあるので，オプティミスティック初期値や，ϵ を最初は大きくしておいて次第に減らすなどの手法で学習を改善することも考えられる．なお，オプティミスティック初期値は行動価値の推定値がその初期値に影響されてしまうのを防ぐため行動価値の推定値を大きくしておき，初期の段階で全ての手の探索を促す手法のことである．

さらに，ゴールキーパーの強化学習も考えられる．今回作成した PK モードのプログラムのゴールキーパーの動きは，既存の PK モードと同じ動きをしているので，より熟練したゴールキーパーを相手にした場合に対応できない可能性がある．したがって，ゴールキーパーも学習で行動を改善することにより，キッカとゴールキーパーの双方について，より良い行動の獲得が期待できる．

謝辞

本研究を進めるにあたって，大変多くの方々にご協力，ご指導をいただき，ここで感謝の気持ちを述べたいと思います．

本研究のテーマである PK モードの環境構築にあたって，「RoboCup サッカーシミュレーション 2D リーグ必勝ガイド」[1] の著者の秋山さんには rctools-users というメーリングリスト [3] で，ご多忙の中，沢山の質問に 1 つ 1 つ丁寧に返信して下さり，有益なご指導やご助言を沢山頂戴して下さり，篤くお礼を申し上げ，感謝いたします．

参考文献

[1] 秋山英久 (著),「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，2006.

[2] Richard S. Sutton・Andrew G.Barto (著)，三上貞芳・皆川雅章 (翻訳),「強化学習」，森北出版，2000.

[3] Rctools-users 保存書庫 http://lists.sourceforge.jp/mailman/archives/rctools-users/