

ロボカップサッカーにおけるシミュレータによる開発の影響

A consideration for developing robot soccer agents by a simulator

尾添健司*¹
Kenji Ozoe

岩川朋弘*¹
Tomohiro Iwakawa

伊藤暢浩*²
Nobuhiro Ito

岩田員典*³
Kazunori Iwata

犬塚信博*¹
Nobuhiro Inuzuka

和田幸一*¹
Koichi Wada

*¹名古屋工業大学
Nagoya Institute Of Technology

*²愛知工業大学
Aichi Institute Of Technology

*³愛知大学
Aichi University

Mixed Reality league is one of RoboCup Soccer competitions. In this league, small real robots are used to play soccer games. A simulator is required to develop agents programs, since all parts of the small robot for this league are breakable. Thus we should use them to adjust their behavior at a final step. However, when we develop agents with a simulator, there is a possibility that robots don't move as same as ones on the simulator. Therefore we developed a simple simulator to develop a soccer agent program for the small robots. We also studied behavior of robots that are developed with the simulator.

1. はじめに

ロボットに関する技術の発展に伴い、エージェントやマルチエージェントシステム [1] に注目が集まっている。このマルチエージェントシステムに関する研究のテストベッドとして RoboCup Soccer [2] がある。RoboCup Soccer は、ロボット工学と人工知能の融合・発展のため、自律移動ロボットによるサッカーを題材として日本の研究者らによって提唱された国際プロジェクトである [3]。RoboCup Soccer には、大きくわけてロボットリーグとシミュレーションリーグがあり、将来的にはそのふたつのリーグの技術は統合されなければならない。そこで、2007 年から Physical Visualization リーグ [4] が導入された。現在はリーグ名が Mixed Reality (以下 MR) リーグと変更されている。

MR リーグはロボットの開発をする際にハードウェアの準備など、様々な問題があり実機のみによる開発は負担が大きい。そこでシミュレータを用いた開発が必要であると考えられる。そこで本研究では、ロボットを使わずに開発可能なシミュレータを開発する。

2. RoboCup Soccer MR リーグ

MR リーグのシステム (以下 MR システム) は、図 1 のような構成となっている。モニタに映し出された仮想的なサッカーフィールド上で実物の小型ロボット [5] を動かし、カメラでロボットの位置を認識して仮想的なボールを蹴ることで試合を行う。ボールの動きは MR サーバ上で計算される。MR サーバは、プレイヤーとなるクライアントのプログラムから送られる動作の命令を赤外線送信機でロボットに送る。そして、カメラから得られたロボットやボールの位置情報をクライアントに送る。クライアントでは、MR サーバから与えられた情報をもとにロボットの動きを決め、その動作の命令を MR サーバに送信する。

MR システムにおいてプレイヤーとなるエージェントプログラムを開発するには以下のような問題があり、繰り返し実験

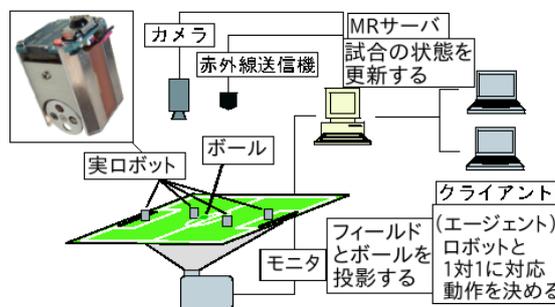


図 1: MR システム概要

を行うのは負担が大きい。

- ロボットの充電電池の消耗が激しい。
- ロボット等実験機具の消耗・摩耗がある。
- 実験機具の準備に時間がかかる。

これらの負担を軽減するために、ロボットをできるだけ使用することなくエージェントのプログラムを開発するためのシミュレータが必要である。しかし、シミュレータで開発を行ったエージェントのプログラムをロボットに適用して動作させた時にシミュレータ上で開発した通りの動作をしない可能性がある。そこでロボットを使わずエージェントのプログラムを開発するためのシミュレータ (以下 MR シミュレータ) の開発を行う。そして開発したシミュレータで作成したエージェントプログラムをロボットに適用するとシミュレータ上の動作とどの程度異なるのか調べる。

3. MR シミュレータの開発

ロボットの開発側から提示されている資料と、実際にロボットの動作を計測したデータにしたがってシミュレータを次のようにする。

まず、シミュレータを MR システムで用いられる MR サーバをベースに開発する。図 1 において、カメラ、赤外線送信

連絡先: 尾添 健司, 所属 名古屋工業大学 和田研究室,
住所 愛知県名古屋市御器所町, 電話番号 052-375-5000, Fax 番号 052-735-5009, 電子メールアドレス
kenji@phaser.elcom.nitech.ac.jp

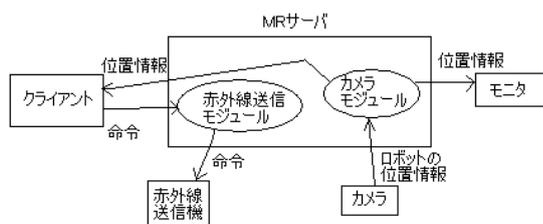


図 2: MR サーバ簡略図

機, ロボットなどのハードウェアを用いずに動作できるようにする. MR サーバの簡略図を図 2 に示す.

MR サーバにおけるハードウェアをコントロールするモジュールを代替ソフトウェアモジュールに置き換えることで MR シミュレータとする. MR シミュレータの簡略図を図 3 に示す.

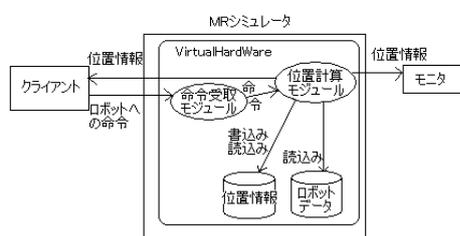


図 3: MR シミュレータ簡略図

赤外線送信モジュールの代替となっている命令受取モジュールではクライアントから送られてくるロボットへの動作の命令を受け取り, 受け取った命令を解析して位置計算モジュールに送信する. そしてカメラモジュールの代替となっている位置計算モジュールでは命令受取モジュールから受信した情報を元にロボットの新しい位置を計算し, 記憶しているロボットの位置情報を更新する. そして新しいロボットの位置情報をクライアントやモニターに送信する. ロボットの位置情報の計算には, 開発側から提示されているロボットの資料と, ロボットの動作を実測したデータを使用することにする. 以上のようなシミュレータを Ubuntu8.04 上で実現した.

4. 実験

作成したシミュレータを用いて, エージェント(ロボット)プログラムを開発したとき, そのプログラムが実際にシミュレータ上で同様に動くとは限らない. そこで, 本研究ではどのような違いが出るのかを検討する.

エージェントプログラムの開発では, 我々は次の 3 つの段階に分けて考える必要がある.

1. 動作するロボット: 1 台, 環境: 変化しない
2. 動作するロボット: 1 台, 環境: 自分自身の動きで変化する
3. 動作するロボット: 2 台以上, 環境: 自分以外の影響でも変化する

本研究ではそれぞれの段階に相当する行動についてシミュレータでエージェントプログラムを開発し, 実験を行なうことにした. 実験で取り扱う行動は以上の段階に対応させた, ロボット 1 台のみの行動である 8 の字走行, ロボット 1 台とボールを含む行動であるドリブル, ロボット 2 台とボールを含む行動であるパス&シュートである. 以上の行動についてエージェントプログラムを開発し, それぞれシミュレータ上とロボットを用いた実験をそれぞれ 30 回ずつ行ない以下の項目を比較・考察する.

- 行動の成功率
- 成功時の平均時間
- 行動軌跡

4.1 8 の字走行

8 の字走行は動作するロボットが 1 台のみで環境が変化しない状況での行動である. 味方ロボット 1 台, 敵ロボット 2 台を配置して, 敵ロボット 2 台の間を味方ロボットが 8 の字を描くように走行する.

行動目的

片方の敵ロボットの周囲を時計回りに走行, もう片方の敵ロボットの周囲を半時計回りに走行, を繰り返すこと.

初期配置

図 4 のように, センタースポットを中心にお互い点対称の位置に敵ロボットを配置する. 味方ロボットをその中央付近に配置する.

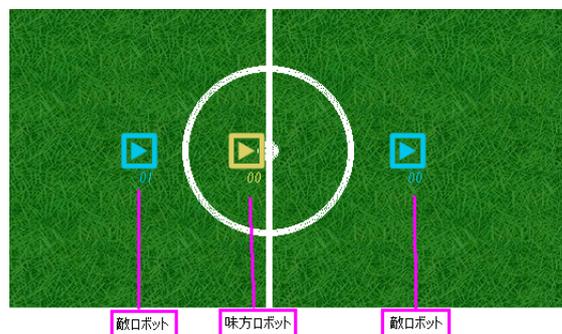


図 4: 8 の字走行の初期配置

行動アルゴリズム

- 味方ロボット
敵ロボットの一方を基準とし, 自分からその敵ロボットへの向きを 0 度としたときに +90 度の方向に進む. 円軌道の半径を, 2 台の敵ロボット間の距離の 0.6 倍とする. 自分がその半径よりも離れた場合, +81 度の方向に進む. 8 の字の交差する点付近に来たら, 基準点をもう片方の敵ロボットに変更し, 今度はそのロボットに対して -90 度の方向に進む. 基準となる半径以上離れた場合は -81 度方向に進む.
- 敵ロボット
動かない.

4.2 ドリブル

8の字走行と比べて、ボールが動くという環境が変化する状況での行動である。味方ロボットがセンターライン手前からボールをドリブルし、3台の敵ロボットをかわしながらゴールまで進む。

行動目的

ボールをフィールドの外に出さずにゴールさせる。敵ロボットに正面から衝突した時に失敗とする。

初期配置

図5のように、ハーフラインからゴールの間に3台の敵ロボットを配置する。味方ロボットはセンターラインよりすこし手前に配置する。

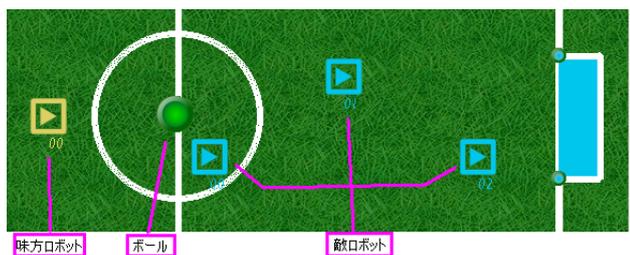


図 5: ドリブルの初期配置

行動アルゴリズム

- 味方ロボット
ボールが自分のkickできる距離にある場合、ボールが敵ロボットの近く(150pixel以内)ならばその敵ロボットからゴールに向かう側の90度方向にボールをkickし、かわす。敵ロボットから離れたら、ボールをゴールの方向にkickする。ボールをkickできる距離でない場合、ボールに向かう。
- 敵ロボット
動かない。

4.3 パス&シュート

ボールと自分以外のロボットが動くという、環境が自分以外の行動によっても変化する状況での行動である。2台の味方ロボットを用い、片方のロボットがコーナーからパスを出し、もう片方のロボットが受けてシュートする。

行動目的

味方ロボットがキックしたボールがゴールに入ればよい。ボールがフィールドの外に出た場合、またはロボット同士が正面衝突した場合は失敗である。

初期配置

図6のように、コーナーエリアからパスを出すロボットと、中央で受けてシュートをするロボットを配置する。

行動アルゴリズム

- パスを出すロボット
ボールに向かい、シュートを行なうロボットに向かってボールをkickする。

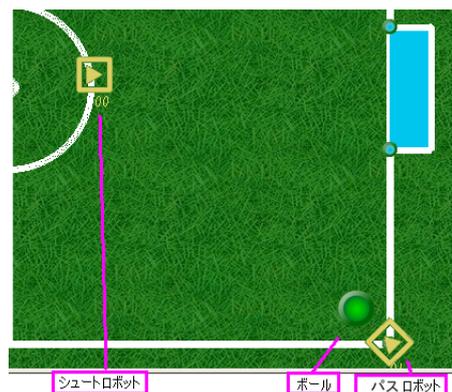


図 6: パス&シュートの初期配置

- シュートを行なうロボット
ボールに向かい、ゴールの中心に向けてボールをkickする。

5. 結果と考察

5.1 8の字走行

MR シミュレータとMR システムでの結果を表1に示す。

表 1: 8の字走行の結果

実験環境	成功率	成功時の平均時間
MR シミュレータ	100 %	32.37 秒
MR システム	73 %	37.92 秒

図7は8の字走行を行ったロボットのMR シミュレータ上での軌跡(緑)と実際のロボットの軌跡(赤)を実験した30回分プロットした物である。

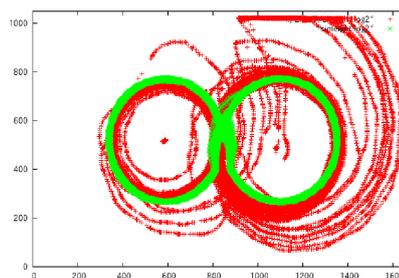


図 7: 8の字走行プロット

実際のロボットを使った実験では図7のように、開発したエージェントプログラムで想定させる軌道を大きく外れて失敗してしまうことがしばしば見られた。その理由として、ロボットの向きや位置を誤認識することがあることをフィールド上の表示から分かった。図7にあるように、8の字の円の中心にプロットがあるのは、カメラが回転の基準となる円の中心の敵ロボットを誤認識して自身がその位置にいると認識してしまっているためである。これらのことから、カメラの誤認識によっ

てMRシミュレータと実際のロボットの間で大きな動作の違いが生じることが分かった。本来ならばロボットが円の内側にまわり込まなければいけない場所にいるときに、ロボットの位置を違う場所として認識し、内側に回り込む行動を行わないといったことが起こったと考えられる。

また、図7で向かって右側の円ばかり失敗している。これは、このプログラムでは向かって左の円を先にまわることになっているので、左の円をまわっている時に生じたズレが右側の円で大きく影響しているからだと考えられる。

5.2 ドリブル

MRシミュレータとMRシステムでの結果を表2に示す。

表 2: ドリブルの結果

実験環境	成功率	成功時の平均時間
MRシミュレータ	70%	18.29秒
MRシステム	45%	13.8秒

実験における味方ロボットの動きを見ると、敵ロボットに衝突してしまう失敗が多かった。この行動は、ボールを使った細かい動作が必要になってくるため、行動と行動の間に生じるタイムラグの影響が顕著に現れている。さらに赤外線通信の欠落にも影響され、行動に遅れが生じ、敵ロボットに衝突したりボールに追いつけないといったことがあった。

また成功時の平均時間に関しては、MRシミュレータ上ではタイムラグがなく細かい動きができる分、一度ボールから離れてしまってもゴールできることが多いため、平均時間が長くなってしまった。その一方で、MRシステム上では一度ボールから離れてしまうと失敗になってしまうことが多いためこの様な結果になった。

5.3 パス&シュート

MRシミュレータとMRシステムでの結果を表2に示す。

表 3: パス&シュートの結果

実験環境	成功率	成功時の平均時間
MRシミュレータ	90%	3.44秒
MRシステム	80%	9.25秒

成功率の差は他の行動よりも小さいため、成功時の平均時間の差に着目した。ロボットではパス、シュートともにkickのときにうまくボールに加速が伝わらないという失敗が目立った。つまり、MRシミュレータ上ではパスロボットとシュートロボットが1度ずつのkickで成功しているが、MRシステム上では2,3度kickしなければ味方ロボットあるいはゴールに届かないといったことが多くあるため、その分時間がかかっている。平均時間が長くなっているのもこのためである。ボールに加速が伝わらない理由として考えられるのは、カメラの誤認識である。kickは、ボールに加速を与えるものなので、ロボットがkickできる範囲にボールがある間、加速を与えることができる。MRシミュレータ上では、その加速がロスなしに伝えられるが、8の字走行の項目でも述べたように、MRシステム上では本当はkickできる場所にボールがあってもカメラがロボットの位置を誤認識することがある。よって違う場所にいるという情報がクライアントに伝えられ、kick命令が送られる回数が少なくなる、といったことが起こっていると考えられる。

6. まとめ

本研究ではMixed Realityリーグにおけるエージェントプログラムの開発にかかる負担を軽減するためにシミュレータを開発し、その動作について考察を行なった。

その結果カメラの誤認識、赤外線通信の欠落、動作のタイムラグがエージェントプログラムの開発に影響を与える大きな要素であることがわかった。今後は、これらの要素をシミュレータに導入し、さらに必要な要素を見つけ出ししていく必要がある。

参考文献

- [1] 大内東, 山本雅人, 川村秀憲.
”マルチエージェントシステムの基礎と応用” コロナ社.
- [2] RoboCup Official Site
<http://www.robocup.org/>
- [3] RoboCup Japan National Committee
<http://www.robocup.or.jp/about.html>
- [4] Rodrigo da Silva Guerra, Joschka Boedecker, Norbert Mayer, Shinzo Yanagimachi, Yasuji Hirose, Kazuhiko Yoshikawa, Masaaki Namekawa and Minoru Asada
”Introducing Physical Visualization Sub-league”
- [5] Eco-Be!
<http://www.eco-be.jp/>