

A I チャレンジ研究会(第 51 回)

Proceedings of the 51th Meeting of Special Interest Group on AI Challenges

CONTENTS

- RoboCup サッカーにおける共有知を用いたプランニング推測に基づく行動選択…………… 1
鈴木 雄大†, 中島 智晴†, 秋山 英久‡
†大阪府立大学, ‡福岡大学
- RoboCup サッカーにおけるニューラルネットワークによる評価関数モデリング…………… 6
福島 卓弥†, 中島 智晴†, 秋山 英久‡
†大阪府立大学, ‡福岡大学
- RoboCup2D プレイヤーのファジィ方策関数の進化的獲得…………… 14
西野 順二, 佐土瀬 寛
電気通信大学
- VR ヘッドセットを用いたサッカーシミュレーションにおける人間の視野方向取得と分析…………… 18
秋山 英久, 齋藤 峻, 荒牧 重登
福岡大学
- カスケード分類器を用いた照明環境変化にロバストなサッカーボール認識…………… 22
森 のどか, 會澤 要, 鈴木 拓央, 小林 邦和
愛知県立大学
- 直交座標型ロボットへの非接触給電システムの開発…………… 28
大黒 康平†, 植村 渉†, 栗井 郁雄‡
†龍谷大学, ‡株式会社リユーテック

日時:2018年5月4日

場所:ソフトピアジャパンセンタービル 10F 特別会議室

The Center Building of Softopia Japan Center, May 4th, 2018

一般社団法人 人工知能学会

Japanese Society for Artificial Intelligence

RoboCup サッカーにおける共有知を用いたプランニング推測に基づく行動選択

Action Selection based on Planning Estimation using Knowledge Sharing in RoboCup Soccer

鈴木 雄大†, 中島 智晴†, 秋山 英久‡

Yudai SUZUKI†, Tomoharu NAKASHIMA†, Hidehisa AKIYAMA‡

大阪府立大学†, 福岡大学‡

Osaka Prefecture University†, Fukuoka University‡

sya01138@edu.osakafu-u.ac.jp

tomoharu.nakashima@kis.osakafu-u.ac.jp

akym@fukuoka-u.ac.jp

Abstract

The aim of this research is to achieve kicker's planning using the idea of knowledge sharing between players. When a kicker makes a decision he plans action sequences. However, our team opuSCOM has a low success rate of the planning action sequences. It is caused by no consideration of other players' movements in the kicker's planning, which is a difficult task. We employ knowledge sharing to consider the kicker's planning and to select an action. In the experiment, we show effectiveness of the knowledge sharing.

1 はじめに

ロボット工学と人工知能の領域横断型研究プロジェクトとしてRoboCup [1] が知られている。RoboCup には様々なリーグが存在しており、それぞれのリーグにおいて活発に研究、開発が行われている。チーム開発においては、ただ単に勝利するだけでなく、賢く安定して勝利することが望まれている。ランダムや思いつきで作られた戦術を使って勝利することよりも、緻密なデータ分析や機械学習によるモデル化を活用して勝利につながる戦術を生成することがRoboCupの理念と合致する。ドリブル等の行動を洗練する [2], 相手の行動を予測して意思決定を行う [3], 相手チームに合わせた戦術をとる [4], といった研究が様々な観点から進められている。

RoboCup サッカーリーグの一つであるサッカーシミュレーション 2D リーグ [5] では、近年戦術が多様化しており、相手戦術との相性が試合結果に大きな影響を与えている。このような背景から、勝利のために、相手に合わせた戦術に切り替える研究が行われている。しかし、これには相手戦術の見極めや、複数の対抗戦術を用意する必要が

ある。そこで、相手戦術に依存せず、自身の戦術の完成度を高めていく実装に取り組んでいる。

本論文では、オフENS時のプレイヤーの行動選択を研究対象とする。オフENS時、ボールを保持したプレイヤー (Kicker) は、より有効な戦術を実現するために、パスやドリブルなどのプランニングを数手先まで行い、行動候補の中からより評価値の高い行動を選択する。ボールを保持していないプレイヤー (nonKicker) は、フォーメーションに従うことのみを目的としたポジショニングを行い、そこにはKickerのプランニングが考慮されていない。本研究室で開発しているチーム opuSCOM は、Kickerがプランニングを行っているにも関わらず、そのプランニング通りに行動できていない場面が多い。この問題を解決するために、本論文では nonKicker の行動に着目し、共有知を用いたプランニング推測に基づく行動選択を提案する。全プレイヤーが推測したプランニングに基づいて行動することで、プランニング通りの行動を実現させることを目的とする。この目的を達成することで、opuSCOM の性能の向上を図る。数値実験により、共有知の実装前後における opuSCOM の性能の変化を調査する。

2 研究背景

RoboCup サッカーシミュレーション 2D リーグでは、戦術の多様化が進んでおり、各チームは様々な戦術を用いる。それに伴い相手戦術との相性が試合結果に大きな影響を与える。この問題に対して、相手戦術に合わせた戦術への切り替え、または、相手戦術に依存しない戦術の実装が対応策として挙げられる。戦術切り替えでは、Nakashimaら [6] が提案した勝敗予測などから相手戦術の識別を行うことに加え、多様な相手戦術の定義や、自身のチームに複数の戦術の実装が必要である。よって、相手の全戦術に適切な対応を行うことは困難と考えられる。そのため、相手戦術に依存しない有効な戦術を実装することが望ましい。全ての戦術に対応するには、プレイヤーがあらゆる状況が

ら動的にプランニングを行い、有効な行動を選択することが重要である。本研究室で開発しているチーム opuSCOM は、nonKicker や敵の移動など未来の状態を正確に考慮できておらず、プランニングが効果を発揮していない。未来の状態を Kicker が考慮する手法として、Akiyama ら [7] の、Delaunay Triangulation を用いたボール位置に基づくポジショニングを用いて味方プレイヤーの位置を予測する手法が考えられる。しかし、この手法の場合、フォーメーションは予め設定されたものであるため動的なプランニングができていないとは言えない。そのため、敵のあらゆる戦術に対応できないと考えられる。また、敵プレイヤーの位置を予測する研究が山下ら [8] によって提案されたが、精度は十分なものではない。これらのことから、Kicker のプランニングを Kicker による予測のみで有効にするのは困難である。そこで本論文では、Kicker のプランニングではなく、nonKicker の行動選択に着目した。未来の状態を予測するなど Kicker のプランニングの精度を高めるのではなく、nonKicker が Kicker のプランニングを考慮して行動することで、問題の解決を図った。Kicker のプランニングを考慮する方法として、評価関数と行動候補を共有知として用いる。従来では nonKicker はフォーメーションに従い、Kicker のプランニングを考慮していなかった。本論文では、共有知を用いたプランニング推測により Kicker のプランニングを考慮した行動選択を提案する。これにより、Kicker のプランニング通りに行動できるようになり、opuSCOM の性能を向上させることができると考えた。

3 共有知を用いたプランニング推測

3.1 プランニング

プランニングとは、パスやドリブルなどの行動候補から、より評価関数の評価値が高い行動、もしくは行動列を探索することである。Kicker は行動決定の際、より評価値の高い行動を選択するためにプランニングを行う。行動列において、直近の行動を深さ 1、その次の行動を深さ 2 と呼ぶ。本論文では、行動意思決定のモデルとして Akiyama ら [9] の提案した木探索による協調行動プランニングを用いる。このモデルでは、ボールキック時において探索木を生成し探索することにより、行動プランを作成している。本論文では、行動プランを次にプレイヤーが行うべき一定数の長さを持つ行動列と定義する。また、探索木の走査アルゴリズムとして、最良優先探索を用いる。

以下の手順により行動プランを作成する。まず、ルートノードに現在の状態を格納する。そして、ルートノードから行動候補を生成する。このとき、プレイヤーが観測した現在の状態から、自分と他者を含めた複数のプレイヤーによって実行される行動候補（パスやドリブル、シュートなど）を生成する。生成された行動候補の評価値を評価関

数により計算し、行動候補と評価値を探索木へ子ノードとして格納する。すべてのノードが追加された後、評価値が最大のノードを選択し、そのノードからさらに行動候補を生成する。これを繰り返すことで、探索ノード数があらかじめ設定された最大値に達するまで探索木を成長させる。ただし、木の深さがあらかじめ設定した値を越える場合や、行動列の終了条件に設定されている行動（シュート）が生成された場合は、その葉ノードでの子ノード生成は行わないものとする。構築された木構造の中から評価値の高いノードをつなげると、行動列が得られる。探索終了後、生成された行動列の中で評価値が最大の行動列を行動プランとする。

プランニングにより、プレイヤーは、より戦術的価値が高い行動を選択することが可能となる。最良優先探索に基づく行動列の探索例を図 1 に示す。図 1 では、簡略化のためノードには行動の評価値のみを記し、エッジ上に行動を記す。図 1 では、初期状態からパスが 2 つ、ドリブルが 1 つ生成されている。そして、それぞれに対し評価関数により、評価値を計算している。その結果、パスにはそれぞれ評価値 30、評価値 20、ドリブルには評価値 15 が計算されている。この中で最も評価値の高い行動である、評価値 30 のパスから、行動候補をさらに生成し評価値を計算している。深さ 1 でパスを受け取るとプランニングされた nonKicker を nonKickerB、深さ 2 でパスを受け取るとプランニングされた nonKicker を nonKickerA とする。仮に、図 1 の状態で探索が終了されたならば、この探索木では、Kicker が nonKickerB へパスを行い、その後さらに nonKickerB が nonKickerA へパスを行う行動プランが生成される。現在 opuSCOM は、深さ 2 時点でほとんどの場合、行動プラン通りに行動できていない。これは、Kicker のプランニングを nonKicker が知らないことに起因している。

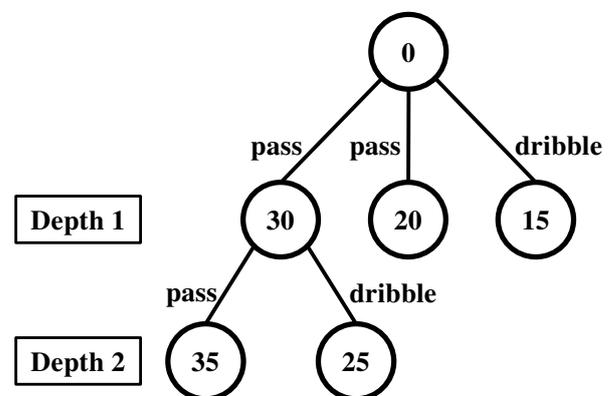


図 1: 最良優先探索に基づく行動列の探索例
Example of best first search in action planning

3.2 共有知

共有知とは、複数のプレイヤーで共有される知識のことである。従来では Kicker のみが持つ評価関数と生成する行動候補を、本論文では全プレイヤーの共有知とし、nonKicker が Kicker のプランニングを推測できるようにする。non-Kicker が Kicker と同様の行動候補を生成できれば、全プレイヤーで同じ行動プランが得られる。従って、Kicker のプランニング推測結果を使って、プレイヤー間の協調行動を獲得できる。評価関数と行動候補を共有知とすることで、試合中に生成される動的な行動プランを複数のプレイヤーが共有することを実現する。nonKicker に Kicker と同様の行動候補生成を可能にする共有知の実装を行い、数値実験によりその効果を検証する。

4 数値実験

4.1 実験設定

opuSCOM に共有知を実装した場合と実装しなかった場合に対して、opuSCOM と実力が近い 10 チームとそれぞれ 300 試合実施する。対戦チームは、2017 年の JapanOpen に出場した Esperanza, Toyosu-Galaxy, WIT の 3 チーム、世界大会に出場した HillStone, Fifty-Storms, ITAndroids, Persiangulf2017, Ri-one2017 の 5 チーム、agent2d[10] と共有知を実装前の opuSCOM の計 10 チームとした。本論文では、Kicker が nonKickerB へパスを行い、さらに nonKickerB が nonKickerA へパスをする、といった行動プラン“ Kicker (深さ 1)nonKickerB (深さ 2) nonKickerA ”を Kicker が意思決定した場合を対象とする。つまり、深さ 1、深さ 2 が共にパスの行動プランを研究対象とする。深さ 1 をパスとした提案理由として、ドリブルが実現されたかどうかの判断が難しいためである。ドリブルに対してパスは、ボールを保持するプレイヤーが移り変わるため、実現されたかの判断が容易であり、本実験を行う上で適していると考えた。

図 2 に実験の概要図を示す。図 2 では、破線がプランニングを、実線が nonKickerA の移動を示す。共有知により、nonKickerA が推測したプランニングが図 2 に示すようなプランニングの場合、矢印の先が示すボール受け取り予測位置へ nonKickerA が移動する。これによりプレイヤーが行動プラン通りに行動できるようになると考えた。本実験では、Kicker と nonKickerA のプランニング共有率、Kicker のプランニング実現率、試合結果から opuSCOM の性能の変化を調査する。プランニング共有率は、パスのみで構成された行動プランのうち、Kicker がプランニングを行った前後 10 サイクル以内に深さ 2 で受け取る nonKickerA が同様のプランニングを行った割合と定義する。実現率は、パスのみで構成された行動プランのうち、Kicker がプランニングを行った後、深さ 1 では 10 サイクル以内、深さ 2 では 20 サイクル以内に行動プランに基づ

くパス先のプレイヤーにボールが渡った割合と定義する。座標は考慮せず背番号のみで一致しているか判断する。また、nonKicker は Kicker の位置座標から行動プランの推測を行うため、ノイズを含む情報の認識ではプランニングの推測は困難である。実験では、共有知の有効性を調査するために、ノイズがなくプレイヤーが他のプレイヤーの位置座標を正確に認識できる fullstate を用いて実験を実施する。

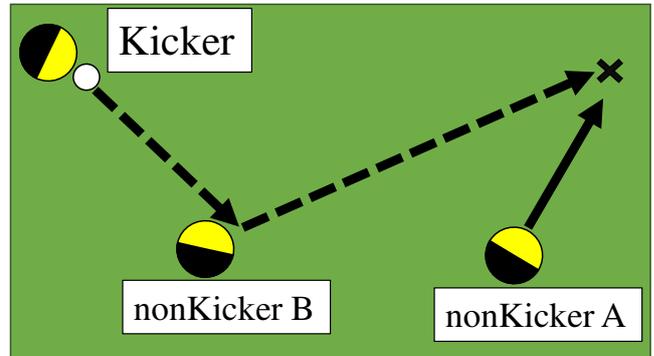


図 2: 実験の概要図
Experimental situation

4.2 実験結果

プランニング共有率、プランニング実現率、試合結果の調査結果を以下に示す。

4.2.1 プランニング共有率

共有知を実装後の opuSCOM と 300 試合行った各チームに対する、1 試合の平均プランニング回数 (#Planning), Kicker と nonKicker のプランニング共有率 (Sharing Rate) を表 1 に示す。表 1 より、どのチームに対しても 90% 程度の高い共有率を示した。このことから、共有知を用いることで nonKicker は Kicker のプランニングを推測できていた。

表 1: プランニング共有率

Planning sharing rates		
Opponents	#Planning	Sharing Rate(%)
agent2d	95	89.40
HillStone	89	89.64
Esperanza	84	90.16
Toyosu-Galaxy	30	92.20
WIT	51	89.49
Fifty-Storms	100	90.48
ITAndroids	84	89.81
PersianGulf2017	73	87.45
Ri-one2017	65	89.72
opuSCOM (Before)	114	90.29

4.2.2 プランニング実現率

共有知を実装後の opuSCOM と 300 試合行った各チームに対する、1 試合の平均プランニング回数 (#Planning) と全プランニングのうち深さ 1 の実現率 (D1)、深さ 1 が実現したプランニングのうちの深さ 2 のプランニングが実現した割合 (D2) を表 2 に示す。表 2 より、深さ 1 が実現したプランニングのうちの深さ 2 のプランニングが実現した割合は実装前後で 2 % ~ 11 % 上昇した。比率の差の検定により、有意水準 5 % で、共有知を実装前後のプランニング実現率に差があると言えた。このことから、共有知を用いたプランニング推測によって、深さ 2 時点でプランニング通りに行動することができた割合は上昇した。

表 2: プランニング実現率
Planning achievement rates

Opponents		#Planning	D1 (%)	D2 (%)
agent2d	Before	101	73.37	57.40
	After	95	73.39	64.25
HillStone	Before	94	71.92	54.52
	After	89	70.20	65.23
Esperanza	Before	89	72.49	60.05
	After	84	73.34	66.70
Toyosu-Galaxy	Before	29	74.46	40.96
	After	30	75.41	49.22
WIT	Before	51	74.06	59.90
	After	51	73.57	64.65
FiftyStorms	Before	104	67.80	55.34
	After	100	68.73	62.58
ITAndroids	Before	91	73.63	48.36
	After	84	73.95	54.97
PersianGulf2017	Before	71	70.29	56.54
	After	73	68.67	58.61
Ri-one2017	Before	67	73.71	48.84
	After	65	73.06	50.68
opuSCOM (Before)	Before	120	71.13	57.70
	After	114	72.43	65.10

4.2.3 試合結果

試合結果を表 3 に示す。項目は、勝ち (Win)、負け (Lose)、引き分け (Draw) の割合をそれぞれ示す。表 3 より、勝率に着目すると勝率が向上したチームもあるが、比率の差の検定により、有意水準 5 % で共有知を実装前後で勝率に差があるとは言えなかった。つまり、有意に勝率が向上したとは言えなかった。

4.3 考察

プランニング実現率が向上したにも関わらず、試合結果で勝率が向上しなかった。この原因として、敵陣では味方プレイヤーの数が少なくパス パスのプランニングが生成されにくいことが原因として挙げられる。図 3 は、agent2d との 1 試合の行動プランを全てプロットした図である。座

表 3: 試合結果

Game results

Opponents		Win (%)	Lose (%)	Draw (%)
agent2d	Before	59.33	29.67	11.00
	After	56.33	30.67	13.00
HillStone	Before	24.67	60.67	14.67
	After	24.00	64.67	11.33
Esperanza	Before	33.00	52.33	14.67
	After	33.67	47.33	19.00
Toyosu-Galaxy	Before	83.33	2.33	14.33
	After	81.33	5.67	13.00
WIT	Before	74.33	14.33	11.33
	After	79.33	10.33	10.33
FiftyStorms	Before	24.00	65.67	10.33
	After	23.67	64.33	12.00
ITAndroids	Before	5.00	86.33	8.67
	After	6.33	87.00	6.67
PersianGulf2017	Before	40.00	46.67	13.33
	After	41.67	48.00	10.33
Ri-one2017	Before	21.33	61.33	17.33
	After	20.67	65.67	13.67
opuSCOM (Before)	Before	37.33	48.00	14.67
	After	41.14	39.13	19.73

標はサッカーフィールドを表し、右側が敵陣、左側が自陣を示す。赤線が成功した行動プラン、青線が失敗した行動プランを示す。図 3 より、敵陣内での行動プランが少ないことが分かる。よって、研究対象とした行動プラン自体が敵陣内で少なく、プランニング実現率が上昇しても勝率への影響が少なかったことが考えられる。また、プランニング実現率は上昇したものの、未だ深さ 2 時点での値は高くないことも原因として考えられる。

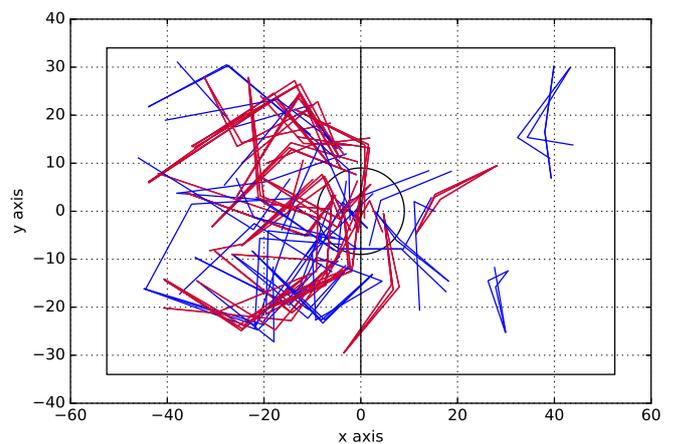


図 3: 1 試合の全行動プラン

All planning during the game

5 おわりに

本論文では、共有知を用いたプランニング推測に基づく行動選択を提案した。共有知によって nonKicker が Kicker のプランニングを推測することができた。これにより、non-Kicker がプランニングに従った行動を選択することができた。実験結果として、プランニング実現率は上昇したが、勝率の向上は見られなかった。今後の課題として、敵陣での攻撃的な行動生成の実装、コーナーキックなど特定の場面への限定化など共有知を活かした実装、または、プランニング実現率がより上昇するような工夫への取り組みが挙げられる。

参考文献

- [1] Hitoaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawa and Hitoshi Matsubara, “RoboCup: A Challenge Problem for AI,” *AI Magazine*, Vol. 18, No. 1, pp. 73-85, 1997.
- [2] Martin Riedmiller, Roland Hafner, Sascha Lange and Marti Lauer, “Learning to dribble on a real robot by success and failure,” *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2207-2208, 2008.
- [3] Tomonari Nakade, Tomoharu Nakashima, Hidehisa Akiyama and Seki Hirosato, “Position Prediction of Opponent Players by SIRMs Fuzzy Models for RoboCup Soccer 2D Simulation,” *Proc. of 8th IEEE International Conference on Soft Computing and Intelligent Systems (SCIS) and 17th International Symposium on Advanced Intelligent Systems*, pp. 224-228, 2016.
- [4] 三船 哲史, 中島 智晴, 秋山 英久, Tomonari Nakade, Tomoharu Nakashima, Hidehisa Akiyama and Seki Hirosato, “RoboCup サッカーにおけるキック分布を用いたゲーム戦術の切り替え,” *Proc. of the JSAI Technical Report SIG-Challenge*, pp.1-6, 2016.
- [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, 2006.
- [6] Tomoharu Nakashima, Satoshi Mifune, Jordan Henrio, Oliver Obst, Peter Wang and Mikhail Prokopenko, “Kick Extraction for Reducing Uncertainty in RoboCup Logs,” *Proc. of 17th International Conference on Human Interface and the Management of Information*, pp. 622-633, 2015.
- [7] Hidehisa Akiyama and Itsuki Noda, “Multi-Agent Positioning Mechanism in the Dynamic Environ-
- ment,” *Proc. of the 12th RoboCup Symposium*, pp. 377-384, 2008.
- [8] 山下 雄大, 三船 哲史, 中島 智晴, 秋山 英久, “RoboCup に対する SIRMs を用いた位置予測モデルの開発,” 第 25 回ソフトサイエンス・ワークショップ講演論文集, 下関市立大学, 山口, 4 ページ, 3 月 10-11 日, 2015.
- [9] Hidehisa Akiyama, Shigeto Aramaki and Tomoharu Nakashima, “Online Cooperative Behavior Planning using a Tree Search Method in the RoboCup Soccer Simulation,” *Proc. of 4th IEEE International Conference on Intelligent Networking and Collaborative Systems(INCoS)*, pp. 170-177, 2012.
- [10] Hidehisa Akiyama and Tomoharu Nakashima, “Helios Base: An Open Source Package for the RoboCup Soccer 2D Simulation,” *Proc. of Robot Soccer World Cup XVII*, pp. 528-535, 2013.

RoboCup サッカーにおけるニューラルネットワークによる評価関数モデリング

Evaluation Function Modeling by using Neural Networks for RoboCup Soccer Simulation

福島 卓弥 †, 中島 智晴 †, 秋山 英久 ‡

Takuya FUKUSHIMA†, Tomoharu NAKASHIMA†, Hidehisa AKIYAMA‡

大阪府立大学 †, 福岡大学 ‡

Osaka Prefecture University†, Fukuoka University‡

takuya.fukushima@edu.osakafu-u.ac.jp

tomoharu.nakashima@kis.osakafu-u.ac.jp

akym@fukuoka-u.ac.jp

Abstract

This paper discusses the construction of an evaluation function by using neural networks in RoboCup Soccer Simulation. For this purpose, four-layered neural networks are employed to model the evaluation function. Supervised learning and reinforcement learning are considered for the learning of the evaluation function. For the training of the neural networks, we generate training data from game logs. We define the successful episodes and extract them from the game logs. In the learning of the neural networks, first, the parameters of neural networks are learned by supervised learning. Then, reinforcement learning is applied to fine-tune the neural networks. We investigate the performance of neural networks tuned by supervised and reinforcement learning through the computational experiments. As a result, it is shown that the performance of the trained neural networks is the same as ones with hand-coded rules.

1 はじめに

近年の人工知能は、状態や行動の質を評価関数に従って点数付けし、得点の高い予測状態になるように行動選択したり、得点の高い行動を選択することで意思決定を行う。そのため、人工知能において評価関数は非常に重要な要素である。この評価関数は一般的に人間が自身の知識に基づいて行動のルールを設定することで調整される。しかし、人間が全ての局面に対して最適なルールを設定することは時間や労力がかかる。また、ルールを設定する際に人間自身がその問題に対してある程度の知識を持って

いる必要がある。これらの問題点から、評価関数を自動的に獲得する研究が進められている。また、評価関数をうまく調整できれば、優れた探索手法を組み合わせることで人間の思考能力を超えるコンピュータプログラムを作成することが可能となる。その一例として、DeepMind社が開発したAlphaGo [1, 2] が挙げられる。AlphaGo [1] はまず人間のエキスパートの棋譜からニューラルネットワークを教師あり学習し、その後、強化学習によってさらに洗練された行動を探索する手順で学習を進める。

ロボット工学と人工知能の領域横断型研究プロジェクトとしてRoboCup [3] が知られている。RoboCupには様々なリーグが存在しており、それぞれのリーグにおいて活発に研究、開発が行われている。その中の一つであるRoboCup サッカーリーグでは、ただ単に勝利するだけではなく、賢く安定して勝利することが望まれている。ランダムや思いつきで作られた戦術を使って勝利するよりも、緻密なデータ分析や機械学習によるモデル化を活用して勝利につながる戦術を生成することがRoboCupの理念と合致する。

本論文では、評価関数を自動で獲得するためにAlphaGoで用いられた実験手順をRoboCup サッカーシミュレーション環境に応用することを試みる。そのために、評価関数を4層のニューラルネットワークでモデル化する。ボールをペナルティエリアに持ち込むことができた行動の軌跡を試合ログから抽出し、それをトレーニングデータとして扱うことで、教師あり学習を行う。その後、強化学習によって重みが調整されたニューラルネットワークを用いて、強化学習を行う。教師あり学習を行うことで、さらに微調整を行うことが可能となる。実験では、教師あり学習後のニューラルネットワーク、強化学習後のニューラルネットワークを用いた評価関数を用いて試合を行い、性能を評価する。

2 関連研究

近年は、様々な問題領域に対して深層学習 [4, 5] が用いられている。深層学習では教師あり学習や強化学習を用いてニューラルネットワークの重みを自動で調整することが可能である。

例えば、マルチエージェント問題において、Hong ら [6] はマルチエージェントシステムを対象とした Q -network を提案した。このモデルではサッカーフィールドをグリッド状に仕切った離散空間が使用されている。しかし多くの現実問題では、このような離散的な環境ではなく、連続的に表現される場合がほとんどである。

Liu ら [7] や Hausknecht ら [8] は連続的に全状態が表現される RoboCup サッカーシミュレーション環境において、深層学習を適用した。しかし、これらの手法はプレイヤーの人数を少人数、もしくは 1:1 に限定している。これは RoboCup サッカーシミュレーションにおける多くの制約問題に起因している。

一方で、評価関数に関する研究として、Warnell ら [9] は Atari のボウリングゲームに対して、人間のトレーナーを用いることでより早いニューラルネットワークの学習を可能にし、高い性能を残した。Stanescu ら [10] は Deep Convolutional Neural Network を Real-time Strategy Game に応用する手法を提案している。また、Silver ら [1, 2] は碁において人工知能が人類を超えるほどの探索手法と評価関数を提案した。

本論文では、RoboCup サッカーシミュレーション環境において、ニューラルネットワークを用いて評価関数をモデル化し、学習することを試みる。

3 RoboCup

3.1 RoboCup サッカー

RoboCup は、ロボット工学と人工知能の発展を目的とした、自律移動型ロボットによるサッカーなどを題材とした研究プロジェクトである。RoboCup には「西暦 2050 年までに、サッカーの世界チャンピオンチームに勝てる自律型ロボットチームを作る」という目標があり、この目標に向けて盛んに研究が行われている。RoboCup にはサッカー以外にも、大規模災害への対応のシミュレーションや災害現場で活躍するロボットの開発を促進するレスキューリーグ、日常生活で人間を支援する自律ロボットによる競技を通じて、人とコミュニケーションしながら役に立つロボットの実現を目指す@ホームリーグの他に、次世代のロボット技術者育成を目的としているジュニアリーグも存在する。本論文では、RoboCup サッカーシミュレーションリーグを研究の対象とする。サッカーシミュレーションはモデル化の形式によって 2D リーグと 3D リーグに分けられる。図 1, 2 に 2D リーグと 3D リーグの試合の様子

を示す。本論文では、図 1 の 2D リーグを扱う。

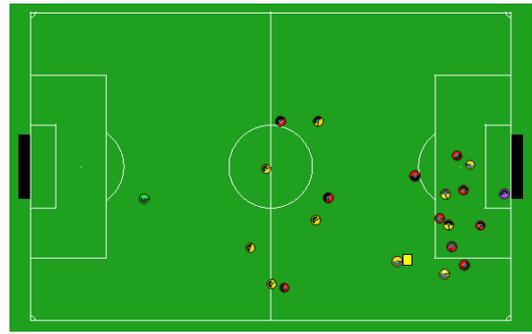


図 1: Soccer simulation 2D league



図 2: Soccer simulation 3D league

3.2 RoboCup サッカーシミュレーション 2D リーグ

本研究では、RoboCup サッカーシミュレーション 2D リーグを対象とする。シミュレーションリーグは RoboCup 創設当初から存在する最も古いリーグの 1 つである。2D リーグでは実機を使用せず、コンピュータ内に用意された二次元平面を仮想サッカーフィールドとし、円形のエージェントをプレイヤーとして競技を行う。また、プレイヤーやボールの位置と速度は全て二次元ベクトルとして表される。2D リーグでは、kick, dash, turn などの抽象化されたコマンドを基本行動とする。試合は前後半 3000 サイクルずつ合計 6000 サイクルからなる。1 サイクルは 0.1 秒で離散化されている。

プレイヤーやコーチはそれぞれ独立したエージェントとしてプログラムされている。各プレイヤーには実際の人間と同様に視野が設定されており、自身の視野内で認識できた情報に基づき、視覚情報が形成される。また、他のプレイヤーやコーチがメッセージにフィールドの情報を含めることで、視野情報を補完できる。これらの視覚情報や聴覚情報からフィールド情報を形成し、これに基づきドリブルやパスなどの意思決定を下す。しかし、視覚情報にはノイズが含まれ、正確な情報を獲得することができない。聴覚情報についても確実に受信できるわけではなく、コーチの

```
(player_type (id 17)(player_speed_max 1.05)(stamina_inc_max 51.6181)
(player_decay 0.459447)(inertia_moment 6.48617)(dash_power_rate 0.00489698)
(player_size 0.3)(kickable_margin 0.643989)(kick_rand 0.0439887)(extra_stamina
85.5322)(effort_max 0.857871)(effort_min 0.457871)(kick_power_rate 0.027)
(foul_detect_probability 0.5)(catchable_area_1_stretch 1.03085))
(playmode 1 kick_off_)
(team 1 opuSCOM NEO_FS 0 0)
(show 1 ((b) 0 0 0 0) ((l) 1 0 0x9 -49.1109 0.0076 -0.0444 0.003 -92.301 90 (v h
180) (s 8000 1 1 130555) (c 0 1 69 0 1 71 1 1 0 0 1)) ((l2) 11 0x1 -18 -5 0 0 7.368
-3 (v h 180) (s 8000 0.941673 1 130600) (f1 11) (c 0 0 50 0 1 51 1 0 0 0 1)) ((l3) 8
0x1 -18 5 0 0 -38.621 26 (v h 180) (s 8000 0.805717 1 130600) (f1 11) (c 0 0 50 0
1 51 1 0 0 0 1)) ((l4) 7 0x1 -18 -14 0 0 -5.5 7 (v h 180) (s 8000 0.944233 1 130600)
(f1 11) (c 0 0 50 0 1 51 1 0 0 0 1)) ((l5) 16 0x1 -18 14 0 0 7.083 2 (v h 180) (s 8000
0.876201 1 130600) (f1 11) (c 0 0 50 0 1 51 1 0 0 0 1)) ((l6) 4 0x1 -15 0 0 0 29.599
1 (v h 180) (s 8000 0.996398 1 130600) (f1 11) (c 0 0 50 0 1 51 1 0 0 0 1))
```

図 3: Game log

メッセージは通常プレイ時には到達までに遅延も発生する。そのため、プレイヤーはフィールド上の正確な情報を保持することはできない。一方で、コーチはフィールド上のすべての情報をノイズなしで取得することができるため、正確なフィールド情報を保持することができる。通常プレイ時におけるプレイヤーへの意思伝達には遅延が発生し、回数も制限されているが、ハーフタイム時には、プレイヤーに即時に情報を伝達することが可能である。また、試合毎に異なる能力を持つプレイヤーのセットが与えられ、各チームがポジションの割当を行う。

上記のように RoboCup にはランダムな要素が多く、プレイヤーが獲得する情報や物体の移動にノイズがかかることから、同一の対戦相手であっても、試合毎に結果や内容が異なる。

3.3 試合ログ

試合ログは、RoboCup サッカーシミュレーション 2D リーグにおいて、試合終了後にサーバから出力されるファイルである。試合ログには、各プレイヤーの最高速度やキックできる範囲等といったパラメータ、ゲームの状態、サイクル毎のプレイヤーとボールの位置や速度の情報、プレイヤーの行動、プレイヤーやコーチ間の情報の伝達等といった試合中の全ての情報が含まれている。そのため、試合ログを用いることで終了した試合を再生することができる。実際の試合ログを図 3 に示す。図 3 のように、試合ログは試合中の情報が文字列で表現されている。そのため、分析に用いる際は必要な情報のみを抽出する。

3.4 状態評価

プレイヤーは意思決定を行う際に、各行動や状態に対して評価値を付ける。評価値は評価関数によって算出される。プレイヤーが観測した現状態や予測状態を入力として、評価関数を用いることで状態評価を可能にする。状態評価は行動選択時に用いられ、行動選択手法と組み合わせることで、プレイヤーは数手先の状況を考慮し、より戦術的価値が高い行動を選択することが可能となる。

現在、評価関数は人間の知識に基づいて行動のルール

を設定することで調整されている。行動のルールを設定するにあたって、評価方法を考慮する必要がある。評価の指標として、ボールとゴールの距離、ボールと敵プレイヤーの距離を用いる場合や、プレイヤーの行動を評価値の指標とする場合が考えられる。これらの特徴量は開発者の勘や経験によって決定され、またその評価値の計算方法も手作業によって調整される。そのため、開発者の意図通りにプレイヤーエージェントを制御するには、開発者の評価関数調整に関する熟練した知識と多数の繰り返しが現状では必要である。

3.5 行動選択

本論文では、行動意思決定のモデルとして木探索による協調行動プランニングを用いる [11]。このモデルでは、ボールキック時において探索木を生成することにより、行動プランを作成している。行動プランを次にプレイヤーが行うべき一定数の長さを持つ行動列と定義する。本論文で使用するチームでは、探索木の走査アルゴリズムとして、最良優先探索を用いる。以下の手順により行動プランを作成する。まず、ルートノードに現状態を格納する。そして、ルートノードから行動候補を生成する。このとき、プレイヤーが観測した現状態や予測状態を入力とし、自分と味方プレイヤーを含めた複数のエージェントによって実行可能な行動（パスやドリブル、シュートなど）を生成する。この時、実現可能な行動がどうかを計算し、不可能だと判断した行動は削除されるため、確実性のある行動のみ生成される。生成された行動に対して評価関数により評価値を計算し、行動と状態、評価値を探索木へ子ノードとして格納する。すべてのノードが追加された後、評価値が最大のノードを選択し、そのノードにおける予測状態からさらに行動の候補を生成する。これを繰り返すことで、探索木を成長させ行動プランニングを実現する。ただし、木の深さがあらかじめ設定した値を越える場合や、ノードの予測状態から行動が生成できない場合、行動列の終了条件に設定されている行動（シュートなど）が生成された場合には、その葉ノードでの子ノード生成は行わないものとする。構築された木構造の中からノード列をつなげることで、行動列を得る。

最良優先探索に基づく行動列探索の例を図 4 に示す。簡略化のためノードには行動の評価値のみを記し、エッジ上に行動を記している。図 4 では、初期状態からある地点へのドリブル動作が 1 つ、パス動作が 2 つ生成されている。そして、それぞれに対し評価関数により、評価値を計算する。その結果、ドリブルには評価値 30、パスにはそれぞれ評価値 20、評価値 10 が計算されている。この中で最も評価値の高い行動である、評価値 30 のドリブル行動後の予測状態から、実行可能な行動をさらに生成し評価値を計算している。仮に、図 4 の状態で探索が終了された

ならば、この探索木では、ある地点までドリブル動作を行い、その後ドリブル動作を行う行動プランが生成される。

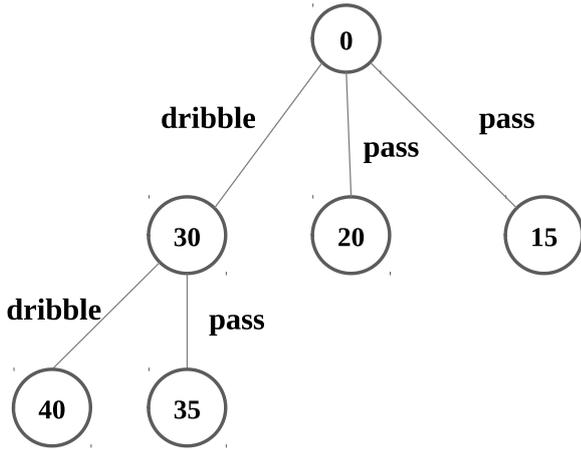


図 4: Example of action planning

4 提案手法

本論文では、行動プランニングにおける行動決定の要素となる評価関数に着目する。チームを強くするためには、各プレイヤーが的確に行動選択する必要がある。現在、評価関数は開発者の勘と調整の繰り返しによって定められた評価項目とパラメータによって設定されている。そのため、本当に適切な行動を選択できているかどうか不明であるうえに、手動による調整ではコストや性能に限界がある。そこで、本論文では行動の適切な評価を行うために、教師あり学習と強化学習を組み合わせることで評価関数を自動で獲得することを目的とする。

本論文では、評価関数を構築するために、AlphaGoの手法をRoboCup環境に応用する。サッカーシミュレーション2Dリーグでは、プレイヤーエージェントは0.1秒で意思決定を行わなければならない。そのため、複雑な構造のモデルでは、計算時間が長く意思決定を行うことができない可能性がある。このことから、今回は複雑すぎない構造の4層ニューラルネットワークを評価関数のモデルとして用いる。ニューラルネットワークは高い近似能力を持つため、評価関数のモデルとしてふさわしいと考えた。さらに、ニューラルネットワークは評価関数のモデルとして非常に優れた性能を持っており、構造を変更することも比較的容易であることから、ニューラルネットワークを本実験では用いることとする。

本論文では2種類のニューラルネットワーク(NN1, NN2)を用いる。ニューラルネットワークの構造の概要を表1に示す。ニューラルネットワークの入力として、予想ボール位置 (x_p, y_p) の2入力の場合と、現在ボール位置 (x_c, y_c) と予想ボール位置 (x_p, y_p) の4入力の場合を用意した。すべての層においてシグモイド関数を活性化関数

として用いる。そのため、ニューラルネットワークの出力値の範囲は $[0,1]$ となる。

表 1: Neural networks for experiments

Neural Network	NN1	NN2
Input Layer	2	4
Hidden1 Layer	100	
Hidden2 Layer	100	
Output Layer	1	
Activation Function	Sigmoid	
Learning Rate	0.1	

4.1 教師あり学習

教師あり学習によって学習させた評価関数の性能を評価する。本論文では、敵のペナルティエリアに至るまで自チームがボールを保持し続けることができた一連の行動を成功エピソードとして定義する。教師信号は成功エピソード中の行動に対しては1、その他のエピソード中の行動に対しては0とする。図5の赤線は成功エピソードを、青の点線は失敗エピソードを示している。多くのチームのベースチームとして用いられるAgent2D[12]に対して、エキスパートを対戦させ、試合ログを収集する。その試合ログからエキスパートのパスやドリブルなどの行動を抽出し、成功エピソードと失敗エピソードに分け、入力情報と教師信号を付加した学習用データを生成する。エキスパートとして、本論文では、HELIOS[13]とWrightEagle[14]を用いる。両チームともに世界大会で複数回の優勝経験をもつチームである。

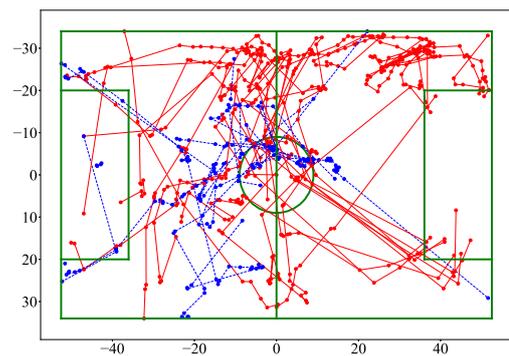


図 5: Example of positive episodes (red lines) and negative episodes (dotted blue line)

4.2 強化学習

強化学習によってニューラルネットワークの重みを更新して得られた評価関数の性能を評価する。ランダムに初期

化された重みを用いず、教師あり学習によって学習された重みを用いる。4.1節のときと同様に、ログファイルから成功エピソードとその他のエピソードを抽出する。強化学習における、評価関数の更新式を式(1)に表す。

$$V^{new}(s) = (1-\alpha)V^{old}(s) + \alpha * \{0.5 + \gamma^t(V(s_{next}) + R - 0.5)\} \quad (1)$$

ここで、 V は評価値、 s は状態、 R は報酬、 α は学習率、 γ は減衰率とする。また、 t はエピソードの最後の行動を1として、行動を遡るにつれて1ずつ増加させる。

また強化学習の実験では、本研究室が開発しているチーム opuSCOM を用い、敵チームは Agent2D とする。

5 数値実験

5.1 実験設定

数値実験では、教師あり学習、強化学習によって重み調整されたニューラルネットワークを評価関数のモデルとした際の性能の変化を調査する。教師あり学習、強化学習ともに NN1, NN2 を用いる。また学習用データとして、HELIOS の試合ログのみを用いたもの、WrightEagle の試合ログのみを用いたもの、両チームの試合ログを用いたものを用意する。ニューラルネットワークの重みを全く学習していない初期値のままのものも比較対象として用いることで、ニューラルネットワークの学習が成功しているのかどうかを評価する。強化学習においては、プレイヤーがすべての視覚情報を正確に得ることができる、full state 環境下においても実験を行う。すべての実験設定を表2に示す。Default とは、人間によって設定されたルールの集合で表現された評価関数である。Simple はゴールまでの距離が短いほど良い評価値をもつ単純な評価関数でありそれ以外のルールを持たない。これは Agent2D が用いているものと同等のものである。また性能の評価指標として、本論文では平均得点、勝率、平均ペナルティエリア侵入回数を調査する。

5.2 実験結果

HELIOS の試合ログを用いて学習したニューラルネットワークの評価値の遷移を図6-11に示す。各図において、 $x > 0$ の領域は敵陣を、 $x < 0$ の領域は自陣とする。NN1については、 (x_p, y_p) を入力として与えることで評価値を可視化している。NN2の場合は、まず9つの (x_c, y_c) を入力として与える。それぞれ、 $(-45.0, -30.0)$, $(0.0, -30.0)$, $(45.0, -30.0)$, $(-45.0, 0.0)$, $(0.0, 0.0)$, $(35.0, 0.0)$, $(-45.0, 30.0)$, $(0.0, 30.0)$, $(45.0, 30.0)$ である。その後NN1と同様に、 (x_p, y_p) を入力として与えることで、9箇所の (x_c, y_c) における (x_p, y_p) の評価値を可視化した。

図6-11から、ニューラルネットワークはエキスパートの行動をうまく学習していることがわかる。ボールをペナルティエリアに持ち込むことができるようなボールの

軌跡は高い評価値をもっているが、一方で、たとえゴールから近くてもボールをペナルティエリアに持ち込むことが難しい行動は評価値が低くなるように出力されている。

Agent2D との試合結果を図12-14に示す。各項目名は表2に示している。図12-14から、ほとんど全てのニューラルネットワーク評価関数は Default や Simple の評価関数に匹敵することがわかった。一方で、それらを上回る性能を残すことはなかった。これらの結果は、実験設定が原因であると考えられる。例えば、強化学習において学習率 α が固定であることや、式(1)は、学習において強い影響を持っており、本設定ではうまく学習できないことがわかった。そのうえ、ボール位置だけをニューラルネットワークの入力にするには不十分であるため、敵プレイヤーや味方プレイヤーの位置を考慮可能なニューラルネットワークを検討する必要がある。

6 おわりに

本論文では、教師あり学習や強化学習を用いて評価関数を学習する手法を提案した。機械学習手法を用いて自動で微調整した評価関数は、ボードゲームだけでなくサッカーでも用いることが可能であることがわかる。今後の課題として、異なる構造のニューラルネットワークを用いて調査することや、敵プレイヤーの位置情報などを考慮できるように入力を変更することが挙げられる。

参考文献

- [1] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel and Demis Hassabis, "Mastering the game of Go with deep neural networks and tree search," Nature, Vol. 529, pp. 484-489, 2016.
- [2] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel and Demis Hassabis, "Mastering the game of Go without human knowledge," Nature, Vol. 550, pp. 354-359, 2017.
- [3] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, Eiichi Osawai and Hitoshi Matsubara, "RoboCup: A Challenge Problem for AI and

表 2: Abbreviation of experimental settings

Abbreviation	Experimental settings				
	Evaluation function	Initial weights	Learning	Training data	Environment
NN1_SL_HE	NN1	Random	Supervised	HELIOS	Not full state
NN1_SL_WE	NN1	Random	Supervised	WrightEagle	Not full state
NN1_SL_HE	NN1	Random	Supervised	HELIOS+WrightEagle	Not full state
NN1_INIT	NN1	Random	-	-	-
NN2_SL_HE	NN2	Random	Supervised	HELIOS	Not full state
NN2_SL_WE	NN2	Random	Supervised	WrightEagle	Not full state
NN2_SL_HE+WE	NN2	Random	Supervised	HELIOS+WrightEagle	Not full state
NN2_INIT	NN2	Random	-	-	-
NN1_RL_HE+WE	NN1	Trained by HELIOS+WrightEagle	Reinforcement	opuSCOM	Not full state
NN1_RL_HE+WE_FULL	NN1	Trained by HELIOS+WrightEagle	Reinforcement	opuSCOM	Full state
NN2_RL_HE+WE	NN2	Trained by HELIOS+WrightEagle	Reinforcement	opuSCOM	Not full state
NN2_RL_HE+WE_FULL	NN2	Trained by HELIOS+WrightEagle	Reinforcement	opuSCOM	Full state
Default	With hand-coded rules	-	-	-	-
Simple	Without hand-coded rules	-	-	-	-

Robotics,” Robot Soccer World Cup, pp. 1-19, Springer, 1997.

- [4] Juergen Schmidhuber, “Deep learning in neural networks: An overview,” Neural networks, Vol. 61, pp. 85-117, 2015.
- [5] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, Fuad E.Alsaadi, “A survey of deep neural network architectures and their applications,” Neurocomputing, Vol. 234, pp. 11-26, 2017.
- [6] Zhang-Wei Hong, Shih-Yang Su, Tzu-Yun Shann, Yi-Hsiang Chang and Chun-Yi Lee, “A Deep Policy Inference Q-Network for Multi-Agent Systems,” *arXiv:1712.07893*, 2017.
- [7] Yaxin Liu and Peter Stone, “Value-Function-Based Transfer for Reinforcement Learning Using Structure Mapping” Proc. of the 21st National Conference on Artificial Intelligence, pp. 415-420, 2006.
- [8] Matthew Hausknecht and Peter Stone, “Deep Reinforcement Learning in Parameterized Action Space,” *arXiv:1511.04143*, 2015.
- [9] Garrett Warnell, Nicholas Waytowich, Vernon Lawhern and Peter Stone, “Deep TAMER: Interactive Agent Shaping in High-Dimensional State Spaces,” *arXiv:1709.10163*, 2017.
- [10] Marius Stanescu, Nicholas A. Barriga, Andy Hess and Micheal Buro, “Evaluating Real-Time Strategy Game States Using Convolutional Neural Networks,” Proc. of the IEEE Conference on Computational Intelligence and Games (CIG), pp. 1-7, 2016
- [11] Hidehisa Akiyama, Shigeto Aramaki and Tomoharu Nakashima, “Online Cooperative Behavior Planning using a Tree Search Method in the RoboCup Soccer Simulation,” Proc. of 4th IEEE International Conference on Intelligent Networking and Collaborative Systems (INCoS), pp. 170-177, 2012.
- [12] Hidehisa Akiyama and Tomoharu Nakashima, “Helios base: An open source package for the robocup soccer 2D simulation,” Robot Soccer World Cup XVII, pp. 528-535, Springer, 2013.
- [13] Hidehisa Akiyama, Tomoharu Nakashima, Sho Tanaka and Takuya Fukushima, “HELIOS2017: Team Description Paper,” RoboCup2017, Nagoya, Japan, 2017.
- [14] Xiao Li, Rongya Chen and Xiaoping Chen, “WrightEagle 2D Soccer Simulation Team Description 2015,” RoboCup2015 Hefei, China, 2015.

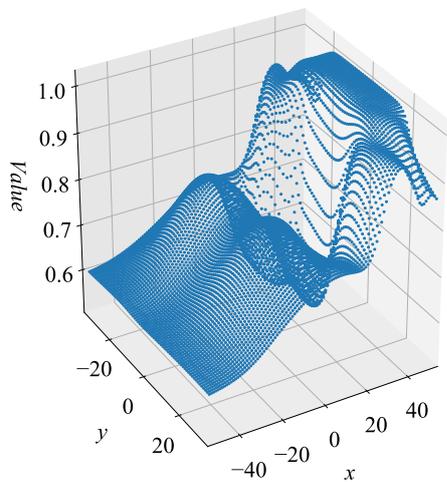


Figure 6: Input-output mapping of NN1 trained by supervised learning

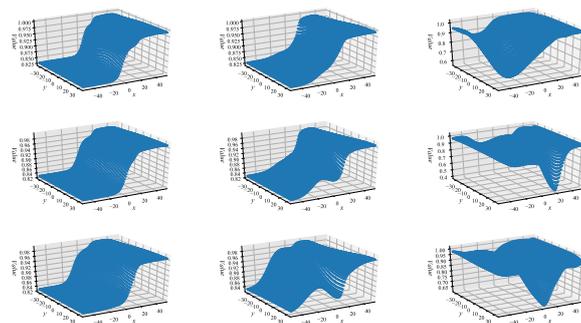


Figure 8: Input-output mapping of NN2 trained by supervised learning

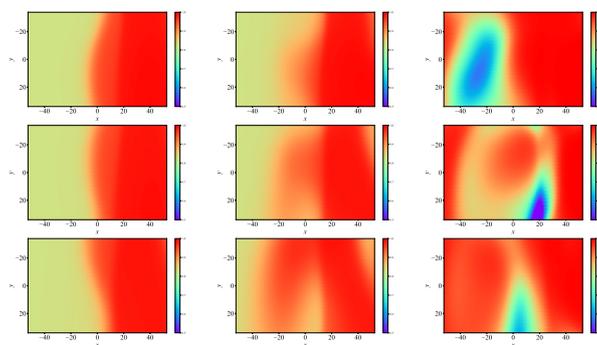


Figure 9: Heatmap of the evaluation value from NN2 trained by supervised learning

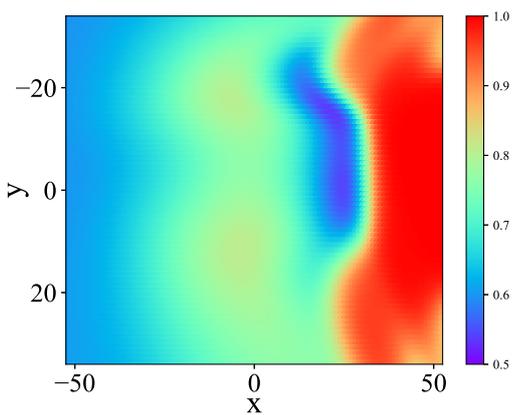


Figure 7: Heatmap of the evaluation value from NN1 trained by supervised learning

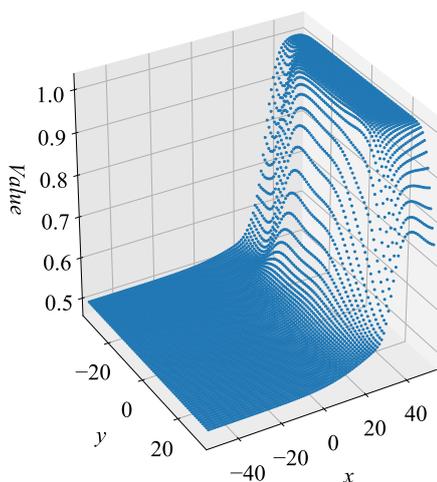
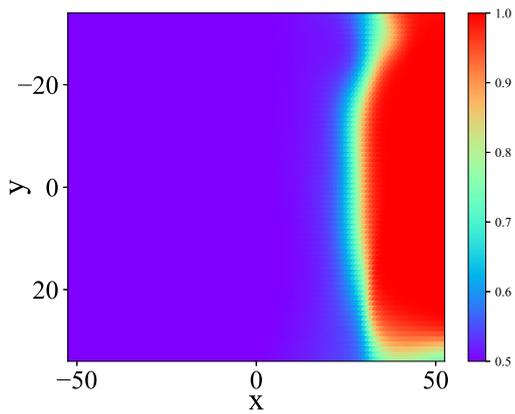
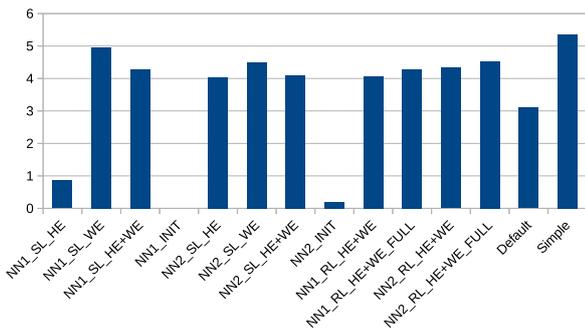


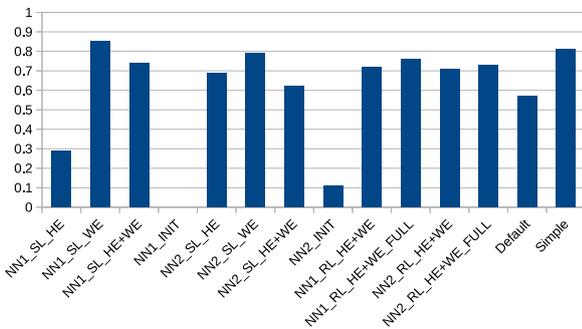
Figure 10: Input-output mapping of NN1 trained by reinforcement learning



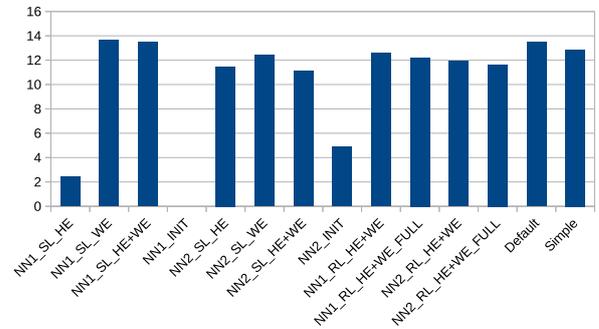
☒ 11: Heatmap of the evaluation value from NN2 trained by reinforcement learning



☒ 12: Average scored goals



☒ 13: Average win rates



☒ 14: Average number of times the ball entered into the opponent penalty area

RoboCup2D プレイヤのファジィ方策関数の進化的獲得

Fuzzy Rule Genetic evolution for RoboCup simulation 2D player

西野順二、佐土瀬寛

Junji NISHINO, Kan SADOSE

電気通信大学

The University of Electro-Communications

nishinojunji@uec.ac.jp

Abstract

サッカープレイヤーの行動決定手法としてファジィ方策関数を提案し、その方策パラメータであるファジィ集合を GA を用いて進化的に獲得した。獲得実験では、世代集団、親個体群、子個体群の個体数をそれぞれ 810、10、20 と設定した。ランダムな初期方策から始めて agent2d との対戦結果を適応度とし 1200 世代以上の進化実験を行った結果、世代ごとの適応度である累計得失点の平均値が約-2.5 点から +2.0 点超まで増加し、強いチームを得ることができた。この計算を実行するため、40 ワークからなるサッカーシミュレーションクラスタを構築し用いた。獲得したファジィ方策関数を実装したプログラムと agent2d による対戦実験の結果、学習前のプログラムの勝率 23 % と比較して 54.9 % と高い勝率であったことから、GA による進化的獲得の有効性を明らかにした。

1 はじめに

強化学習の枠組みでは環境に対する行動戦略を方策関数 π で表し、 θ によるパラメータ化を行い $\pi(\theta)$ とおく。最適なパラメータの発見を目指すことが行われている。例えばニューラルネットであれば θ はリンクの重み変数に相当し、重み付きのルールベースシステムならば、ルールの重みとルールの記述パラメータに相当する。このとき $\Delta\theta$ を明示的に計算することで更新する学習測が用いられる。

しかしながら サッカープレイヤーの協調行動のように、状態空間が大きく、報酬がスパースかつ遅延が大きいという複雑な問題の強化学習での解決は非常に困難である。

DQN[Mnih 13] は、状態空間をニューラルネットによって近似しながら学習を行うことで、巨大な状態空間への

対応を行い簡単なゲームでは高い学習性能をあげている。しかし、報酬がスパースで遅延するものは学習がうまくできていないことと、学習ができた場合でも獲得されたニューラルネットの解釈・分析が事実上できないという課題がある。

ファジィ方策関数は、複雑な問題の大局的な解法知識をファジィ推論によって表現することで、こうした困難な問題を可読性のある形で解決することを目指している。ファジィ推論規則で表現した方策関数はニューラルネットに比べて構造的な記述能力が高いという特徴がある。一方でファジィ推論の非線形性と非単調性(多峰性)、いたるところで微分不可能という特性から、 $\Delta\theta$ を明に得ることができず、勾配を用いた最適化によって方策を求められない。

そこでヒューリスティックなメタ探索である遺伝的アルゴリズム [北野 93] を用いてファジィ方策関数を獲得することとした。

本研究では疎なサッカーシミュレーション用クラスタを構築し 40 試合を並列で行うことでより進化世代数を大幅に増やすことを試みた。近年、強化学習では試行をより多数、この場合 GA の進化世代数をより長期間行うことで、よりよい解を得られることが示されている [Silver 17]。遺伝的アルゴリズムによるサッカールール獲得は、2006 年ころまで積極的に行われてきた。このころ Taylor らは 420 時間相当(約 2,500 試合)[Taylor 06]、Luke らは 10,000 試合相当の学習 [Luke 98] を行っている。計算資源の制約によりこの程度の実験が行われていた。本実験では、大幅に長期間にあたる 8,000 時間相当(48,000 試合)の進化反復を専用のサッカークラスタを構築して実施した。

本研究では RoboCup Soccer simulation 2D リーグを対象とし、サッカープレイヤーのファジィ方策関数を提案しそのパラメータであるファジィ集合を GA を用いて学習しファジィ方策関数を獲得することを目的とする。

2 RoboCupSoccer2D プレイヤにおける ファジィ方策関数

2.1 RoboCupSoccer2D シミュレーション

本研究では、RoboCup Soccer simulation 2D リーグのプレイヤーを対象とする。simulation 2D リーグのシミュレータは、サーバクライアント方式で構築され、rcsserver と呼ばれる一つのサッカーサーバプログラムと、両チーム 11 ずつのプレイヤーに相当する、のべ 22 クライアントプログラムで 1 つの試合を実行する。試合時間は 6000 ステップ、実時間で 10 分である。

サッカーサーバでは 2 次元平面で高さのない仮想サッカーフィールドを用意し、ボールやプレイヤーの物理的な運動のシミュレーションを行う。サーバは、各クライアントのサッカーエージェントプログラムへ、仮想の視覚・聴覚情報の送信、物理計算、状態更新など 100 ミリ秒単位で行う。サッカーエージェントはこの知覚情報等を元に状況を判断し、100 ミリ秒以内に行動を決定し、行動コマンドをサーバへ送信する。

2.2 ベースプログラム agent2d

本研究では ベースプログラムとして agent2d を用いた。これは秋山らによって開発された 2010 年の世界大会優勝チーム HELIOS の基本的な行動をライブラリ化したキットのサンプルエージェントプログラムである。

ドリブルなどの基本行動や、状態予測と探索に基づく行動計画、ボール位置を反映したチームフォーメーション、基本的なセットプレー、ペナルティキックなどのその他の基本行動がビルトインされている。優勝チーム HELIOS にごく近く、サンプルでありながら、そのままの状態、標準的なサッカーチームとしては十分な強さを持っている。

2.3 chain action 探索

agent2d では ボール保持者の行動を決定する方策決定として chain action 探索を用いている。

探索では以下のアルゴリズムによって行動決定を行う。

1. 初期状態を入力とし行動の候補を生成する。
2. 成功と予測される行動を生成し 行動とその結果の予測局面を組としたノードを作る。
3. ノードを生成するときに予測局面を評価し 各ノードは評価値を保持する。
4. 最良優先探索により終了条件を満たすまで木を成長させる。
5. 探索木の生成が終了した後 評価値が最も高いノードにつながるアクション連鎖を決定する。

ここで生成される探索木を図に示す。

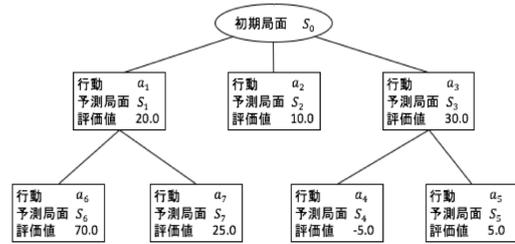


図 1: chain action の探索木

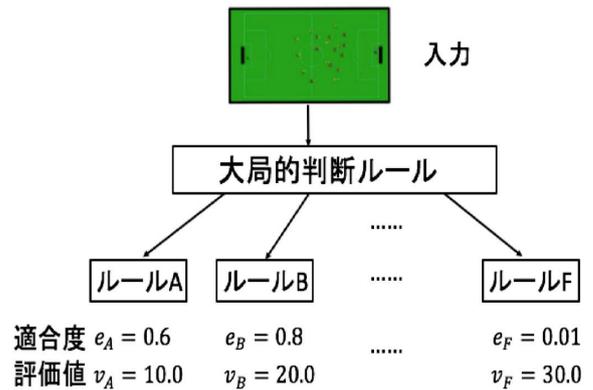


図 2: ファジィ局面評価関数

2.4 ファジィ方策関数

ファジィ方策関数 $\pi(s)$ は、状態から行動を決定するためのアルゴリズムとして、ファジィ推論を組み込んだ方策関数である。状態に対して行動を確定的に決定する。

谷川ら、田川らは chain action 探索内で用いる評価関数を人間の知識に基づく状態特徴量の線形和で表現し、これに対して強化学習を行った [谷川 13, 田川 15]。

ファジィ方策関数は、人の直感的知識を推論構造に反映し、線形和に比べて表現力が高い特徴がある。

ファジィ方策関数は、谷川らと同様に、chain action 探索を基本形として採用し、探索で用いる評価関数を多段ファジィ推論で表現した。これは、ボール保持者の行動を決定するとき、大局的な状況判断とその状況に応じた状態評価の二段階に対応している。

ファジィ推論の構造を図 2 に示す。

3 ファジィ方策関数の進化的獲得

ファジィ方策関数は、ファジィ推論に依拠する非線形な複雑さにより、一般に多峰的であり、確率の方策勾配法など一般的な強化学習手法によってパラメータを学習することが困難である。本研究では ファジィ方策関数のファジィ推論パラメータを GA を用いて獲得することを提案した。GA の世代交代モデルには JGG を、交叉手法には ファジィ推論パラメータが実数値であることから、REX

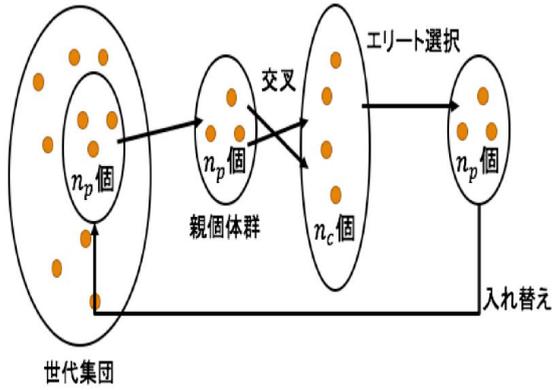


図 3: 世代交代モデル JGG (Just Generation Gap)

を採用した [小林 09]。

3.1 世代交代モデル JGG (Just Generation Gap)

JGG (Just Generation Gap) 法にもとづく世代交代の手順は以下のとおりである。

1. 与えられた初期化領域に n_{pop} 個の個体をランダムに生成し初期集団とする。
2. 集団からランダムに n_p 個の親個体を非復元抽出する。
3. 親個体集団に交叉を繰り返し適用し n_c 個の子個体を生成する。
4. 子個体の適応度を評価する。
5. 子個体集団を適応度にしたがってソートし、上位 n_p 個体を親個体と置き換えて次世代の集団とする。

この流れを図 3 に示す。

3.2 交叉 REX (Real-coded Ensemble Crossover)

交叉 REX では式 (2) および 式 (1) にしたがって子個体 $\mathbf{x}^c \in \mathcal{R}^n$ を生成する。 \mathbf{p}^g を親個体群の重心ベクトルである。

$$\mathbf{p}^g = \frac{1}{n_{pop}} \sum_{i=1}^{n_{pop}} \mathbf{p}^i. \quad (1)$$

$$\mathbf{x}^c = \mathbf{p}^g + \sum_{i=1}^{n_{pop}} \xi_i (\mathbf{x}^i - \mathbf{p}^g), \quad \xi_i \sim \phi \quad (2)$$

4 ファジィ方策関数の進化的獲得実験

世代の個体数などの設定を表 1、子個体の評価に関する設定を表 2、のとおりにしてファジィ方策関数の進化的獲得実験を行った。

表 1: 集団の設定

次元数	54
世代集団数 n_{pop}	810
親個体群の個体数 n_p	10
子個体群の個体数 n_c	20

表 2: 子個体評価の設定

適応度	累計得失点差
評価試合数	2 試合
対戦相手	agent2d

4.1 サッカー GA クラスタ

GA によって進化実験をするとき、評価を子個体数だけ行う必要がある。世代数を千回、子個体を 10 として、1 試合で評価をするとしても 10,000 回の試行が必要となる。一方 RoboCup Soccer simulation 2D の試合は、1 試合のシミュレーションを 10 分で行うため、10,000 回では約 70 日を必要とする。

このため、本研究ではクラスタを構築し、進化計算の高速化を行った。ここで、子個体評価のためのサッカーシミュレーションが違いに影響しない疎な関係にあるため、独立な計算機を 40 台並列に用いた。設定は子個体数を 20 個体、評価は 2 試合であるため 1 世代での試合数は 40 となる。

本クラスタを用いることで、1 世代分のサッカーシミュレーションを一度に計算することができ、世代をまとめる GA 操作には時間がかからないため、全体として効率はほぼ 40 倍であった。詳細には、個々の試合ごとにラグタイムがあるため、各世代単位で終了待ちの同期を取った。シミュレーション全体への影響はほとんどなかった。

4.2 GA による進化と最良個体評価

以上の実験設定で、1200 世代超まで進化実験を行った。合計試合数は $1200 \times 20 \times 2 = 48,000$ 試合である。

世代ごとの適応度平均の推移を図 4 に示す。個体集団平均で -2.5 点程度の得失点差だったが 1.0 点程度まで上昇し agent2d 相手に強くなったことがわかる。

いくつかの世代のクライアントを取り出し agent2d と対戦させたところ、学習前 (初期世代集団) の個体の勝率は 23.2 %、学習後の最良個体は 54.9 %、手作業によるファジィ方策関数を用いたクライアントは 51.2 % の勝率であった。

以上の結果から、強いファジィ方策関数を獲得できたと言える。

また、評価の変化はまだ上昇をつづけている傾向があり、収束したとは言えず、従来より増やした 1200 世代でも進化には十分とはいえないことがわかった。

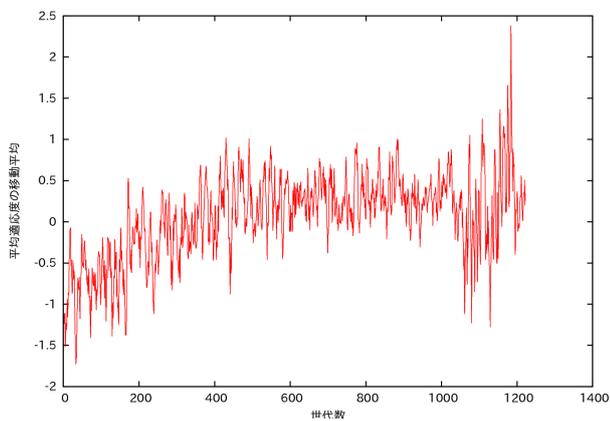


図 4: 進化過程 (1200 世代) 縦軸は 2 試合での累計得点差

5 おわりに

本研究では、強化学習による行動学習が難しい問題に対し、人の直感を構造的に生かしたファジィ方策関数を設定し、そのパラメータを GA を用いて進化的に獲得する手法を提案し、実験した。

従来より多数の、1200 世代、48,000 試行以上の個体評価を専用クラスタを構築して行って進化実験をしたところ、適応度が世代を繰り返すごとに増加したことから GA によるファジィ方策関数の獲得の可能性を確かめることができた。

学習後のファジィ方策関数を用いたプログラムは基準の agent2d に対し勝率 54.9 % であり強いプログラムが得られている。

今後の課題として、まだ収束しきっていない進化をさらに続けることで、何世代まで進化させるのが適当であるかの知見を得ることが望まれる。

謝辞本研究の一部は科学研究費補助金基盤研究 C(26330273) および電気通信大学共同研究 (株式会社 QUANTUM) により支援されたものである

参考文献

- [Luke 98] Luke, S., et al.: Genetic programming produced competitive soccer softbot teams for robocup97, *Genetic Programming*, Vol. 1998, pp. 214–222 (1998)
- [Mnih 13] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M.: Playing atari with deep reinforcement learning, *arXiv preprint arXiv:1312.5602* (2013)
- [Silver 17] Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al.: Mastering Chess and Shogi by Self-Play with a General

Reinforcement Learning Algorithm, *arXiv preprint arXiv:1712.01815* (2017)

- [Taylor 06] Taylor, M. E., Whiteson, S., and Stone, P.: Comparing evolutionary and temporal difference methods in a reinforcement learning domain, in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pp. 1321–1328 ACM (2006)
- [小林 09] 小林重信: 実数値 GA のフロンティア, *人工知能学会論文誌*, Vol. 24, No. 1, pp. 147–162 (2009)
- [谷川 13] 谷川俊策, 五十嵐治一, 石原聖司 他: 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, *ゲームプログラミングワークショップ 2013 論文集*, pp. 106–109 (2013)
- [田川 15] 田川諒, 五十嵐治一 他: サッカーエージェントにおける局面評価関数の強化学習, *ゲームプログラミングワークショップ 2015 論文集*, Vol. 2015, pp. 78–83 (2015)
- [北野 93] 北野宏明: 遺伝的アルゴリズム, 産業図書 (1993)

VRヘッドセットを用いたサッカーシミュレーションにおける人間の視野方向取得と分析

Acquiring and Analyzing Human's View Direction using VR Headsets in the Soccer Simulation

秋山英久, 齋藤峻, 荒牧重登

Hidehisa AKIYAMA, Ryo SAITOU, Shigeto ARAMAKI

福岡大学

Fukuoka University

akym@fukuoka-u.ac.jp

Abstract

In this paper, we propose a method to acquire human's view direction in the simulated soccer environment. We use a VR headset in order for human to immerse in the virtual soccer field. Human's head direction in the virtual soccer field can be measured by VR headsets. In the experiments, we analyze the differences between human's head direction and simulated soccer player's one.

1 はじめに

マルチエージェントシステムは人間の集団行動に起こりうる事象の分析を目的としており、災害時の避難誘導やスポーツにおけるチーム戦術を題材とした研究が行われている。マルチエージェントシステムにおいて、人間が取得しえない情報に基づいてエージェントが意思決定した場合、シミュレーションの結果を実世界へ適用できない可能性がある。シミュレーションの精度を向上させるには、エージェントが周辺環境から取得する情報を、人間のものとより近づけることが必要である。

マルチエージェントシステムのテストベッドとしてRoboCupサッカー2Dシミュレーションが知られており、サッカーシミュレーション上のプレイヤーはサッカーエージェントとも呼ばれている。従来は、サッカーエージェントの視野方向の決定において、人間が同環境に置かれた場合にどこに注目するかというデータは考慮されていなかった。「人間が自身の意思決定に必要な情報を得るために注目している対象」といったデータを収集できれば、より実世界に近づけたシミュレーションの実現が期待できる。

本研究では、VRヘッドセットを用いた3次元ビューを用いて、サッカーシミュレーション上での人間の視野方向データを収集する手法を提案する。実験では、人間と

サッカーエージェントの視野方向にどのような差が観られるか、また、人間の視野方向データをサッカーエージェントへ反映することでシミュレーションのリアリティを向上させうるか、という観点で分析を試みる。

2 関連研究

RoboCupサッカー2Dシミュレーションはマルチエージェントシステムのテストベッドとして知られており、高さの概念が存在しない2Dフィールド上で各チーム11体のプレイヤーが現実とほぼ同様のルールで試合を行う[Noda 96]。RoboCupサッカー2Dシミュレーションにおけるチームの強化には個々のサッカーエージェントの性能向上が不可欠であり、従来研究には様々な観点から個々の選手の性能向上を試みるものが見られる[Nakayama 02, Kok 03, Yao 02, Gabel 08]。これらの研究では、センサ情報からより精度よく世界の情報を推定する手法や、サッカー選手としての基本的な行動の性能を向上させることが目指されてきた。

プレイヤーによる世界の認識や体の動きに関する性能向上は数多く試みられてきた。しかしながら、世界の状態を認識する上で、特に視野方向は非常に重要な要素であるにも関わらず、より効率よく世界の情報を収集するための研究は十分に行われてきていない。本研究では、サッカーエージェントの視野方向に注目し、人間とエージェントの視野方向を比較、分析することで、人間の視野方向データをエージェントの意思決定へ反映させるに足る特徴を発見することが目的となる。

3 VRヘッドセットを用いた人間の視野方向取得

人間と同様に、サッカーエージェントが周辺環境から得る情報の大部分は視覚情報である。よって、効率よく視野を動かすことが良質な内部モデルの構築に繋がる。本研究では、サッカーエージェントの意思決定への応用を将来的に

想定し、人間の視野方向のデータを収集することを目的とする。これを実現するために、人間をサッカーエージェントと同環境に置き、試合を追体験させることで人間が意図する視野方向を記録、分析する。

3.1 VRヘッドセットを用いた3次元ビューワ

今回開発するシステムの構成を図1に示す。サッカーシミュレーションの仮想フィールドに人間を没入させるために、VRヘッドセットを用いる。VRヘッドセットの画面では3次元ビューワでサッカーシミュレーションの試合ログが再生される。人間の視野の中心となる3次元ビューワ内でのカメラ位置は、仮想フィールド上の指定したプレイヤーの位置に設定される。2次元平面のサッカーであるため、シミュレーション上ではエージェントの視野に高さは設定されていない。しかし、人間の感覚に近づけるために、3次元ビューワでは視野の高さを設定した。これによって、VRヘッドセットを装着した人間は、仮想フィールド上の特定プレイヤーとほぼ同じ視野で試合を追体験することができる。

ヘッドマウントディスプレイを装着した人間は図2のような試合風景を観察できる。今回開発する3次元ビューワは試合ログ再生機能のみであるため、人間は自分の意思で移動はできず、エージェントが試合中に移動した位置を辿るのみである。ただし、人間は自分の頭の動きと連動して視野方向を変更可能である。

今回開発したシステムでは、VRヘッドセットとしてHTC Viveを用いた(図3)。また、3次元ビューワの開発には、ゲームエンジンであるUnityを採用した。

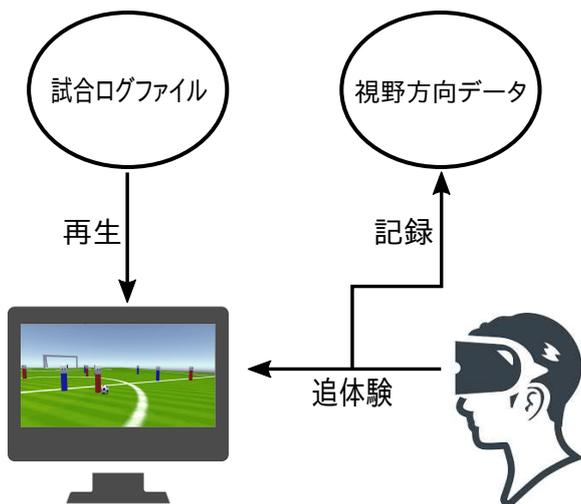


図1: システム構成。

3.2 試合ログ再生

3次元ビューワの試合ログ再生に関する仕様は以下の通りである：

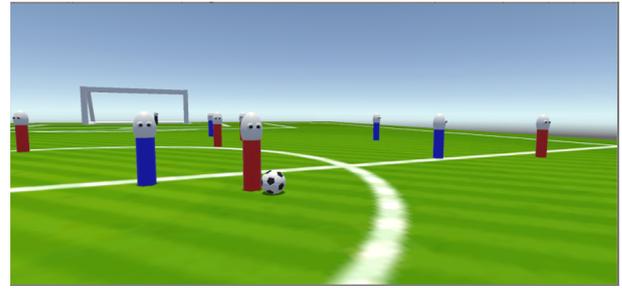


図2: 3次元ビューワの再生画面。



図3: 使用したVRヘッドセット: HTC Vive。

- RoboCup サッカー 2D シミュレータが記録した試合ログファイルから、各シミュレーションサイクルにおけるボール及びサッカーエージェントの位置座標、体の向きを取得する。
- RoboCup サッカー 2D シミュレーションの1サイクルは100ミリ秒であるため、試合ログをそのまま再生するとフレームレートは10fpsとなる。これでは人間の目には十分に滑らかな描画とならない。そこで、サイクル間の物体の位置座標を補間して、フレームレートが20fpsとなるようにする。
- ボールおよびサッカーエージェントを描画するためのオブジェクトに位置座標と体の向きなどを与え、画面表示する。試合ログにはサッカーエージェントの頭の向きも記録されているが、今回の3次元ビューワではサッカーエージェントの頭の向きは使用されない。

3.3 人間の視野方向データの記録

本研究で開発したシステムでは、人間の頭の向きの変化をVRヘッドセットで計測することで、シミュレーション空間内での人間の視野方向を記録する。実際の人間の視野方向は眼球の動きも影響するが、今回は人間の頭の向きのみを視野方向とし、眼球の動きは考慮しない。

本研究で開発した3次元ビューワの座標系を図4に示す。ただし、VRヘッドセットによる視野方向の変更はサッカーエージェントと同じ範囲に限定し、z軸回転で-180度

から+180度までとする．試合ログの再生時，各サイクルの人間の視野方向が記録される．

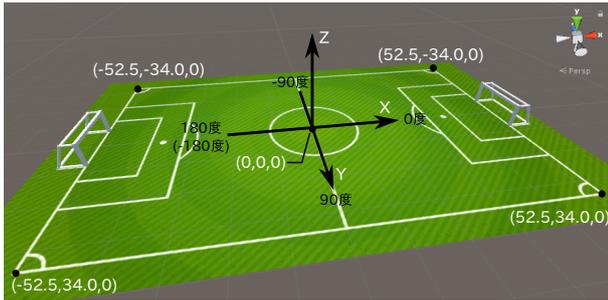


図 4: 3次元ビューワの座標系.

4 実験

サッカー経験者 a (経験年数 10 年)，経験者 b (経験年数 3 年)，未経験者の計 3 名の視野方向データを収集・分析する実験を行う．

4.1 実験設定

試合開始から一定時間経過するまでをデータ収集のための計測時間とする．今回開発したシステムを長時間使用することの身体的疲労が大きかったため，被験者の集中力低下を抑えるために，一回の計測時間を 1000 サイクル (100 秒) とした．ただし，その間に得点，フリーキック，コーナーキックの状況が含まれる試合を対象とした．実験に使用した試合ログは，RoboCup2017 に出場した HELIOS2017 と PersianGulf2017 の対戦である．

RoboCup サッカー 2D シミュレーションの試合ログには，得点やファウルによって試合が中断してからリスタートするまでの，試合が止まっている状況も記録されている．よって，厳密には，データ収集を行う計測時間は，試合開始時点から 1000 サイクル経過時点までの間に起こった得点 (リスタートまで 51 サイクル) 1 回とファウル (リスタートまで 30 サイクル) 4 回の経過時間を加算した 1171 サイクル分 (約 117 秒) となった．

被験者が追体験するサッカーエージェントとして，攻守ともに関わる機会が多いボランチのポジションに当たる選手を採用した．データの収集にあたり，被験者は自由に視野を動かして良いものとし，その他特別な指示はしない．

以下の方法で視野方向データの分析を行う:

実験 1 : サッカー経験者と未経験者の視野方向の比較

実験 2 : サッカー経験者とサッカーエージェントの視野方向の比較

4.2 実験 1

サッカー経験者と未経験者の視野方向を比較し，人間の視野方向に個人差が観察できるかを確認する．視覚機能

の中でも特にスポーツと関係が深い周辺視野の使い方は，スポーツの練度に影響される．そのため，経験者と未経験者の視野方向データには何らかの差異が予想される．

4.3 実験 1 結果

被験者 3 名の視野方向データをグラフ化したものを図 5 に示す．図中の縦軸は z 軸中心での視野方向角度であるため，絶対値 180 度で循環している．図 5 では 400 サイクル前後に大きな振動が観られるが，実際には絶対値が 180 度の周辺へ視野が向けられている．



図 5: 被験者 3 名の視野方向 .

事前の予想と異なり，被験者 3 名の視野方向に大きな差は観られなかった．これは，被験者全員が常にボールを視野内に入れておくことを意識してしまったためと考えられる．しかし，視野方向の振れ幅に注目すると，未経験者の視野方向は経験者 2 名に比べ外側に振れており，ボールを目で追いつつ視野を制御する能力がサッカーの練度に影響を受けていると予想される．また，得点が起こる 900 サイクル前後では，被験者 3 名の視野方向に差が観られた．

4.4 実験 2

サッカー経験者とサッカーエージェントの視野方向の差をグラフ化し比較，分析する．

4.5 実験 2 結果

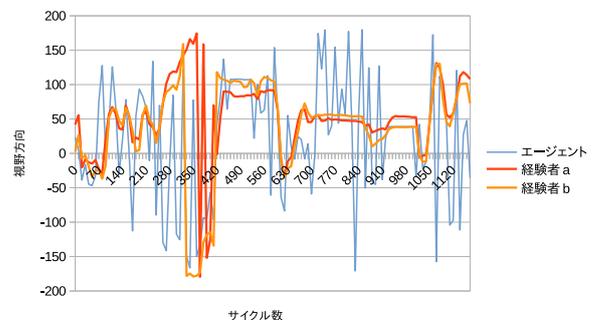


図 6: 経験者とエージェントの視野方向 .

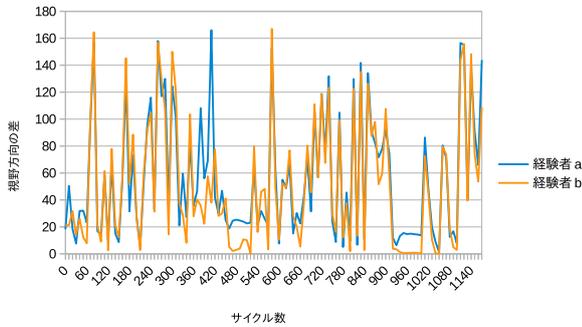


図 7: 経験者とエージェントの視野方向の角度差。

サッカー経験者とサッカーエージェントの視野方向をグラフ化したものを図 6 に示す。図より、試合全体を通してサッカー経験者の視野方向とサッカーエージェントの視野方向には大きな差があることが分かる。この結果から、人間はボールを視野に入れつつ他の選手の位置を確認しているのに対して、サッカーエージェントは試合中にボールから目を離し、フィールド全体の敵や味方の位置を確認していることが分かる。一方で、コーナーキックや得点により試合が中断してからリスタートするまでの間は、サッカー経験者 2 名とサッカーエージェントの視野方向はほぼ一致している。これは、試合が中断している間は他の選手の位置が大きく変化することが無く、サッカーフィールド全体を見渡さなくとも周辺の情報を取得することができるためと考えられる。

図 7 は、経験者 a, b の視野方向とサッカーエージェントの視野方向の角度差をそれぞれ示したものである。コーナーキック（500 サイクル前後）や得点（900 サイクル前後）の部分を除いて、約 100 サイクル（約 10 秒）毎に視野方向の差が大きく増加と減少を繰り返していることが分かる。さらに、人間とエージェントの視野方向の差が急激に増加するのに要する時間は約 10 サイクル（約 1 秒）と短く、サッカーエージェントが一定周期かつ短時間で周辺の情報を取得しようとしていることが分かる。

4.6 考察

今回の実験結果から、RoboCup サッカー 2D シミュレーションにおけるエージェントの視野方向は人間の視野方向と大きく異なり、エージェントは人間に比べ短時間で広範囲から情報を取得していることが分かる。この結果は、RoboCup サッカー 2D シミュレーションにおけるエージェントが、人間には不可能な早さで周囲の情報を取得していることを示していると考えられる。

5 まとめ

本稿では、人間とエージェントが周辺環境から取得する情報に共通点や差異が見られるかという点に主眼を置き、人間にエージェントと同じ立場で試合を追体験させるとい

う手法を用いて、人間とエージェントの視野方向の収集・分析を行った。実験によって、RoboCup サッカー 2D シミュレーションにおけるエージェントが、人間に比べ短時間で広範囲から情報を取得でき、人間には不可能な早さで周囲の情報を取得している可能性があることを確認した。

本研究で収集した視野方向データはまだ少なく、人間の視野方向データをエージェントに学習させることでシミュレーションのリアリティが向上すると結論付けるには不十分である。まず、人間がより試合へ没入し、サッカーエージェントと同様の意思決定をすることを意識づけた上で実験が必要である。その上で、より多くのデータを収集・分析し、人間の視野方向データをエージェントの視野方向に反映させることが今後の課題である。

参考文献

- [Noda 96] Itsuki Noda, Hitoshi Matsubara: Soccer Server and Researches on Multi-Agent Systems, Proc. of IROS-96 Workshop on RoboCup, pp.1-7, 1996.
- [Nakayama 02] Koji Nakayama, Takuya Ako, Toshihiro Suzuki, Ikuo Takeuchi: Team YowAI-2002, RoboCup 2002: Robot Soccer World Cup VI, 2003.
- [Yao 02] , Jinyi Yao, jiang Chen, Yunpeng Cai, Shi Li: Architecture of TsinghuaAeolus, RoboCup 2001: Robot Soccer World Cup V, pp.491-494, 2002.
- [Kok 03] Jelle R. Kok, Remco de Boer, Nikos Vlassis, F.C.A. Groen: UvA Trilearn 2002 team description, RoboCup 2002: Robot Soccer World Cup VI, 2002.
- [Gabel 08] Thomas Gabel, Martin Riedmiller and Florian Trost: A Case Study on Improving Defense Behavior in Soccer Simulation 2D: The NeuroHasle Approach. RoboCup 2008: Robot Soccer World Cup XII. pp. 61-72, 2008.

カスケード型分類器を用いた照明環境変化にロバストなサッカーボール認識

A Robust Soccer Ball Recognition against Lighting Fluctuation by Using Cascade Classifier

森のどか，會澤要，鈴木拓央，小林邦和

Nodoka MORI, Yo AIZAWA, Takuo SUZUKI and Kunikazu KOBAYASHI

愛知県立大学 大学院 情報科学研究科

Graduate School of Information Science and Technology, Aichi Prefectural University

kobayashi@ist.aichi-pu.ac.jp

Abstract

現在ロボカップサッカーの標準プラットフォームリーグ (SPL) では、白地に黒色の五角形のテクスチャ模様がある、いわゆるサッカー柄のボールを使用し、自然光が差し込むフィールドでの精度が高いボール認識が要求されている。本稿では、そのような照明環境が変動する環境に対してロバストかつ、リアルタイムにボール認識を行うため、画像の特徴量によるカスケード型の分類器を用いた手法を提案する。そして、最良の精度が得られる特徴量を調べるため、複数の特徴量を用いて比較検証を行った。

オレンジ色のボールを使用していたときは、画像内のオレンジ色の領域を探索すればよかったが、白地に黒色の模様のボールに変わり、単にボールの色を探索するだけでは認識できなくなった。しかし、その手法ではボールの影や、内部に生じるエッジによりズレが生じたり、NAOの関節などをボールと誤検出してしまふ (例: 図 3,4)。

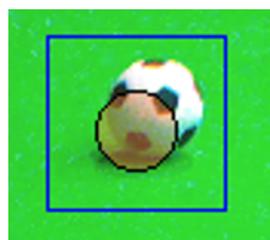


図 3: 影によって誤認識した図

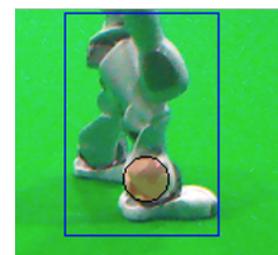


図 4: NAO をボールと誤認識した図

1 はじめに

ロボカップサッカーの標準プラットフォームリーグ (SPL) は、全チームが同一のロボットを用いてプログラミングの優劣を競うサッカーリーグである [1]。現在は Soft-Bank Robotics 社のヒューマノイド型ロボット、NAO が用いられている。また、SPL のルールは年々変更されており、人間がサッカーを行う環境に近づき、難易度も増している。その中でもボールは、以前はオレンジ色のボール (図 1) を用いられていたが、ルール変更により白地に黒色の五角形のテクスチャ模様のボール (図 2) に変わり、フィールド環境も自然光の中で行うよう変更された。



図 1: 以前使用していたオレンジ色のボール



図 2: 現在使用している白地に黒色の模様のボール

また、NAO は計算能力が限られており、その中でボールや NAO の認識、自己位置の計算等をリアルタイムで行わなければならない。

近年、物体認識の手法として、機械学習を用いる手法が注目されている。この手法は、人が物体の特徴を設定したりする必要がなく、複雑な物体でも認識することが可能である。有名なものとして、Deep Learning を用いた手法が挙げられる。これは、現在物体認識において、認識精度が最も高い手法であり、2012 年の画像認識コンテストで 2 位の手法と圧倒的な差をつけて優勝したことにより注目されるようになった。しかし Deep Learning を用いた手法は、計算量が莫大であるため、NAO の CPU ではリアルタイム処理が難しい。そこで本研究では、機械学習の内、Boosting されたカスケード型分類器を用いることで、NAO でリアルタイムなボール認識を実現させた。

2 先行研究

2.1 Boosting

ブースティングとは、弱分類器を組み合わせることで、より精度の高い分類器を作り上げる手法である (図 5)[2] .

本研究では、複数ある Boosting の手法の内、AdaBoost という手法を用いた . AdaBoost とは、Adaptive Boosting の略であり、学習過程で、分類器の認識率に対して適応的に重み付けをし再学習することで精度の良い分類器を作る手法である [3] .

図 5 において、 $h_T(x)$ は、 T 個目の特徴量を指し、 α は重みを指す .

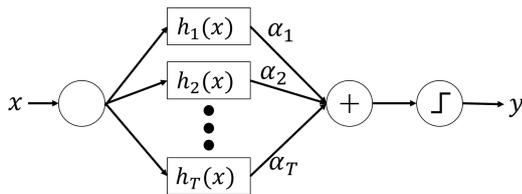


図 5: AdaBoost

2.2 画像特徴量

カスケード型分類器は、学習時に特徴量を抽出しそこから分類器を作成する . 今回は以下の 3 つの特徴量について比較した .

- Haar-like 特徴量
- Local Binary Pattern (LBP) 特徴量
- Histogram of Oriented Gradients (HOG) 特徴量

次項からそれぞれの特徴量の説明を記す .

2.2.1 Haar-like 特徴量

Haar-like 特徴量とは、画像の輝度の差を利用したものである [4][5] . この手法は、予め Haar-like 特徴と呼ばれる矩形領域のパターンを複数用意する (図 6) .

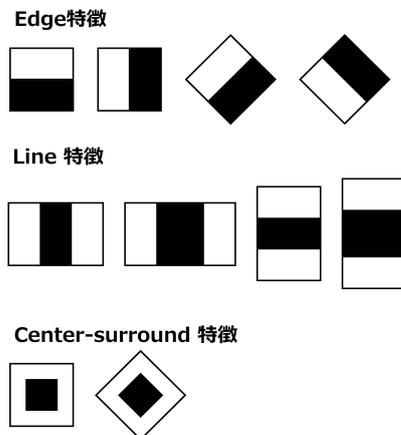


図 6: Haar-like 特徴

次に、入力画像上の一部分を切り取り、そこにパターンをそれぞれ当てはめ、式 (1) のように計算し、その値を特徴量とする .

$$H(r_1, r_2) = S(r_1) - S(r_2) \quad (1)$$

ここで、 $S(r_1)$ と $S(r_2)$ がそれぞれ用意したパターン中の白と黒の 2 つの領域の平均輝度値である .

この手法は、顔認識によく用いられており、年齢、人種関係無い特徴を得ることができる .

2.2.2 LBP 特徴量

LBP 特徴量とは画像のヒストグラムを利用したものである [6][7] . この手法は回転変動などには弱いですが、画像の照明変化の影響を受けにくく、また高速計算が可能である .

LBP の計算は 3×3 ピクセルサイズの画素領域に着目し行う . まず中心の輝度値と周辺 8 近傍の画素の輝度値を比較する . その 8 近傍の内、輝度値が中心の輝度値以上のとき 1、それ以外を 0 とする . ここにマスクを掛け合わせ、その総和を求めこの値が中心画素の輝度値と置き換える . マスクとは、左上から時計回りに 2^n の重みを割り振ったものである (図 7 中、右から 2 番目の図) . この作業を全画素に対して行い、できた画像を LBP 画像と呼ぶ . こうして求めた LBP 特徴量を用いて物体を認識する .

図 7 の場合、上記のように計算し、式 (2) より LBP 特徴量は 241 となる .

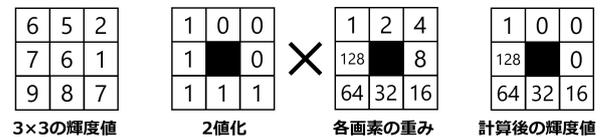


図 7: LBP の計算方法

$$LBP = 1 + 16 + 32 + 64 + 128 = 241 \quad (2)$$

2.2.3 HOG 特徴量

HOG 特徴量とは、局所領域の輝度の勾配方向をヒストグラム化したものである [8] . HOG 特徴量の求め方は、まず局所領域の輝度の勾配方向と強度を算出し、ヒストグラムを作成する . そして局所領域ごとで正規化を行うことで求めることができる .

この特徴量は、画像内の対象物体の大きさ変化や回転には弱いですが、対象の物体が一定の枠内に収まっている画像等では、照明の変化などにも強く、人物検出として用いられている .

2.3 Attention Cascade

Attention Cascade とは、強分類器を複数連結させた構造をしている (図 8)[9] . こうすることで、上位で関係ない領域を早めに削除していくため、効率よく認識するこ

とができる．各ステージの強分類器は AdaBoost で学習させている．



図 8: カスケードと学習器の構成



図 10: ポジティブ画像 (屋外)

3 提案手法

本研究は，画像の特徴量を用いてカスケード型分類器を作成し，ボール認識を行った．カスケード型分類器は，画像群を学習データとして学習させ作成する．用いる学習データセットとして，ボールが写っている画像 (ポジティブ画像) と，ボールが写っていない画像 (ネガティブ画像) をそれぞれ用意した．ポジティブ画像には，室内で撮影したものと屋外で撮影したものを (例：図 9,10)，ネガティブ画像に，画像内にボールが写っていない画像を用いた (例：図 11)．これらの画像データセットは公開しており，誰でも利用可能である¹．

また，機械学習で精度を向上させるためには大量のデータが必要となるため，同じ SPL チームの SPQR が公開しているデータセットも用いた [10]²．これらの画像データを用いて，カスケード型分類器を作成する．



図 9: ポジティブ画像 (室内)



図 11: ネガティブ画像 (屋内)

また，画像全体を探索すると処理に時間がかかってしまうため，まずボールの候補領域を抽出し，その領域候補を分類器に適用する．このボール候補領域の抽出には，SPL のチーム，B-Human のコードリリース 2016 内の『Ball-Region』というモジュールを使用した [11]³．このモジュールでは，まず画像全体を垂直方向に探索し，白色の領域をボールの候補として検出している．

本研究では，このモジュールにより得られたボール候補をカスケード型分類器に入力し，1 つに絞った．

4 実験

本研究では，前節で説明したそれぞれの特徴量のカスケードを作成し，性能を比較する．評価するポイントは次の 3 点である．

- ボールの認識率
- 照明環境の変化に対するロバスト性
- 処理速度

¹<http://www.ist.aichi-pu.ac.jp/lab/robocup-spl/en/publications.html>

²<http://www.dis.uniroma1.it/~labrococo/?q=node/459>

³<https://github.com/bhuman/BHumanCodeRelease/tree/coderelease2016>

実験条件は表 1 の通りである．実験で使ったプログラムコードは github で公開している⁴．今回，ステージ数を 14 にしたのは，それぞれの特徴量でカスケードを作成した際に，最も少ないステージ数で候補を絞れたカスケードのステージ数に揃えたためである．

表 1: 実験条件

画像処理ライブラリ	OpenCV 2.4.13.1
Boosting type	Gentle AdaBoost
プログラミング言語	C++
画像サイズ	640×480
カスケードのステージ数	14

また，精度を比較するにあたり，認識率の値とボールではない部分をボールだと誤認識した数を用いた．誤認識の判断は，ボール全体を囲っていない，又は，囲った内側とボールの間に隙間がないものとした．

4.1 実験用カスケード作成

まず，実験を行うために Indoor カスケードと Outdoor カスケードの 2 種類のカスケードを作成する．本研究では，カスケードの作成に OpenCV (Open Source Computer Vision) のライブラリを用いた．

学習に用いるデータ画像は，NAO に備わっているカメラで撮影した．NAO のカメラは額の中心と口元の 2 箇所にあり，この内，額の中心にあるカメラを用いた．この画像サイズは 640×480 であり，フレームレートは 30fps である．

Indoor カスケードは屋内で撮影した画像のみで学習させ作成する．学習に用いたデータの内訳は表 2 に示す．

表 2: Indoor カスケードの作成に用いた画像枚数

ポジティブ画像	700 枚
ネガティブ画像	300 枚

Outdoor カスケードは，屋外で撮影した画像と屋内で撮影した画像を両方用いて学習し作成した．学習に用いたデータの内訳は表 3 に示す．

表 3: Outdoor カスケードの作成に用いた画像枚数

ポジティブ画像	屋内画像	700 枚
	屋外画像	210 枚
ネガティブ画像		300 枚

4.2 ボールの認識精度の比較

屋内で撮影したボール 100 個分の画像群を用いて，それぞれの特徴量における認識精度を比較する．この実験では Indoor カスケードを使用した．実験の結果は表 4 の通りとなった．

表 4: ボールの認識精度

特徴量	認識率 [%]	誤認識数 [個]
Haar-like	95	44
LBP	97	197
HOG	86	5

この結果から，LBP 特徴量は認識率が高いが誤認識した数も多く，HOG 特徴量は他の特徴量と比べると，認識率は低い誤認識した数は少ないことが分かった．

4.3 照明環境の変化に対するロバスト性の比較

次は，屋外の画像群をボール 100 個分用いて，それぞれの特徴量における精度を比較する．今回の実験では，Indoor カスケードと Outdoor カスケードそれぞれで精度比較を行う．実験の結果は表 5 の通りとなった．

表 5: 照明環境の変化に対するロバスト性

カスケード	特徴量	認識率 [%]	誤認識数 [個]
Indoor	Haar-like	41	78
	LBP	52	405
	HOG	63	11
Outdoor	Haar-like	83	191
	LBP	60	1184
	HOG	85	17

この結果から，照明環境の変化に対して最もロバスト性が高いのは HOG 特徴量となった．そして，全ての特徴量に対して屋内の画像のみで作成したカスケードよりも，屋外の画像も合わせて作成したカスケードの方が認識精度が向上したことも分かった．

4.4 処理速度の比較

最後に，Indoor カスケードを用いて，それぞれの特徴量における処理速度を比較する．今回はネガティブ画像を 2000 枚用いて処理速度を計測し誤認識した数を計 3 回ずつ測定した．実験の結果は表 6 の通りとなった．

この結果から，Haar-like 特徴量が最も処理時間が早く，逆に HOG 特徴量が一番時間がかかることが分かった．しかし HOG 特徴量は誤認識した数が最も少なく，LBP 特徴量が誤認識した数が最も多かった．

⁴<http://github.com/CamelliaDragons/CamelliaDragonsCodeRelease>

表 6: 処理速度

特徴量	処理速度 [ms]	誤認識数 [個]
	8.78×10^4	
Haar-like	8.08×10^4	11902
	8.79×10^4	
LBP	10.14×10^4	39352
	10.09×10^4	
	10.16×10^4	
HOG	19.00×10^4	1687
	18.90×10^4	
	18.20×10^4	

4.5 考察

4.2~4.4節の結果から、それぞれの特徴量に対し次のことが分かった。

Haar-like 特徴量

Indoor カスケードで屋外でのボール認識の精度が良くないが、Outdoor カスケードでは屋外でのボール認識の精度は高いため、学習データに対応する環境のデータを加えることで対応できることが分かる。また、処理速度が3つの特徴量の中で最も早い。

LBP 特徴量

誤認識した数が多く、また Indoor カスケードを用いて屋内でボール認識を行ったときは認識率は一番高いが、Outdoor カスケードを用いたときの屋外のボール認識精度が低いため、より複雑な環境下での精度を高めるには大量の学習データやカスケードのステージ数を増やす必要があると考えられる。

HOG 特徴量

照明変化に強く、誤認識した数も低いが、処理速度が他の特徴量を用いた場合よりも圧倒的に遅い。

これらのことを考えると、SPL におけるリアルタイムボール認識には、3つの特徴量の中では Haar-like 特徴量が適していると考えられる。

最後に、実際に Haar-like 特徴量を用いてボール認識を行った結果を示す。以下に示す画像において、黒い輪郭で赤いフィルターでマークされたものがボールと認識されたものである。

まず屋内にて、3m 離れたところにボールを置いた場合と、4.5m 離れたところにボールを置いた場合でも図 12、13のように正確に認識することができた。



図 12: 3m 離れたボールの認識



図 13: 4.5m 離れたボールの認識

次にボールをフィールドの白線上に置き、認識が困難な場合でも、ボールを正確に認識することができている(図 14)。



図 14: 白線上のボールの認識

また自然光が当たる屋外における環境においても、ボールの認識が正確にできている(図 15)。

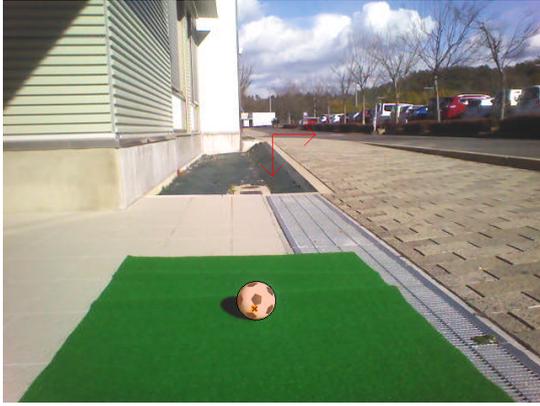


図 15: 屋外でのボールの認識

5 おわりに

本稿では、画像特徴量を弱識別器とした Adaboost を用いたカスケード型分類器によるボールのリアルタイム認識を行った。機械学習により、物体の特徴を自動で抽出できるため、人間にとって特徴を定義するのが困難な場合でも特徴を捉え識別できることが分かった。また前節の結果より、SPL においては、Haar-like 特徴量を用いた物体認識が適していることが分かった。

今後の課題として、静止画像に対して分類器を作成したため、動いているボールの認識精度はあまり高くない。強豪チームはキックしたボールの速度が速いため、このような状況への対処方法も検討していきたい。

参考文献

- [1] Robocup standard platform league. <http://spl.robocup.org/>.
- [2] 藤吉弘亘. 局所特徴量と統計学習手法による物体検出. <http://www.vision.cs.chubu.ac.jp/cvtutorial/pdf/03objectdetection.pdf>.
- [3] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, Vol. 55, No. 1, pp. 119–139, 8 1997.
- [4] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. pp. 511–518, 2001.
- [5] Rainer Lienhart and Jochen Maydt. An extended set of haar-like features for rapid object detection. In *IEEE ICIP 2002*, pp. 900–903, 2002.
- [6] 長谷川修. Local binary pattern, a review. 研究報告グラフィクスと CAD (CG), Vol. 2012, No. 3, pp. 1–6, nov 2012.

- [7] 寺島裕貴, 喜田拓也. 勾配情報を用いた local binary pattern の改良. 第 6 回データ工学と情報マネジメントに関するフォーラム論文集, Vol. 2014, , 3 2014.
- [8] 藤吉弘亘. Hog 特徴量と boosting を用いた人検出. http://www.vision.cs.chubu.ac.jp/joint_hog/pdf/HOG+Boosting_LN.pdf.
- [9] 皆川卓也. Opencv による物体検出徹底解説. <https://www.slideshare.net/takmin/20141008-40019861>.
- [10] Domenico Bloisi, Francesco Del Duchetto, Tiziano Manoni, and Vincenzo Suriani. Machine learning for realisticball detection in robocup SPL. *CoRR*, Vol. abs/1707.03628, , 2017.
- [11] Thomas Röfer, Tim Laue, Jonas Kuball, Andre Lübken, Florian Maaß, Judith Müller, Lukas Post, Jesse Richter-Klug, Peter Schulz, Andreas Stolpmann, Alexander Stöwing, and Felix Thielke. B-Human team report and code release 2016, 2016. Only available online: <http://www.b-human.de/downloads/publications/2016/coderelease2016.pdf>.

直交座標型ロボットへの非接触給電システムの開発 The Contactless Power Supply System for a Cartesian Robot

大黒 康平
Kohei OHGURO
龍谷大学

t18m009@mail.ryukoku.ac.jp

植村 渉
Wataru UEMURA
龍谷大学

wataru@rins.ryukoku.ac.jp

栗井 郁雄
Ikuo AWAI
株式会社リユーテック

awai@ryutech.com

Abstract

直交座標型ロボットには、その頻繁な動作により電力ケーブルが疲労切断したり、電力ケーブルが大きな空間を占めることにより、設置範囲や組み合わせが制限されるといった問題点がある。このような問題を解決するためには、非接触給電が有効であるが、駆動用サーボモータの運動変化に伴う抵抗値の変動による、動作電力の変化に対応する必要がある。本論文ではこの問題に対して電界結合方式での非接触給電システムを提案し、検討を行う。そのシステムの中心部は1段あたり3つの π 型配置の容量で表すことができる。電源から負荷側を見ると並列共振と直列共振があるが、本論文では直列共振を用いて電力伝送を行っている。直列共振周波数は後ろ側のシャントのCのみの影響を受けるため、そのキャパシタンスを小さくする構造を考案する。本システムの電力分配特性を確認すると変動負荷に供給する電力は抵抗値の変動に追従し、それ以外の負荷に供給する電力は一定となったことより、直交座標型ロボットへの複数負荷に対する非接触での電力供給が可能であることを示す。

1 はじめに

直交座標型ロボットとは2軸または3軸の直交するスライド軸により構成される産業用ロボットを指し、高速性、正確性、巧緻性、信頼性、柔軟性といった特徴を持つ。それにより、シーリング、部品組み立て、搬送といった人間の労働に置き換わりつつある。しかし、多数回の高速摺動に伴う給電ケーブルの屈曲によってそれが疲労切断[1]したり、多数の給電ケーブルの束が太いために、設置範囲や組み合わせが制限されるといった問題がある。そこで、本論文では各サーボモータへの給電を非接触で行うことにより、ケーブルの欠点を解決することを目的とする。

以下、2章では非接触給電システムの技術的課題とその解決の方針について述べ、3章にて提案する非接触給電回路とその構造を説明する。その回路の電力分配特性を4章にて確認し、5章で本稿をまとめる。

2 非接触給電の方式と技術的課題

本論文では位置合わせの自由度が高い電界結合方式を用いて非接触給電を行う[2]。共振用のインダク

タを装荷し、1MHzで共振させる。

しかし、このシステムの実現には次のような技術的課題が存在する。直交座標型ロボットの駆動用にはサーボモータが用いられている。サーボモータの運動変化は抵抗値の変動によって決められる。本論文では変動負荷に対して、常に要求する電力を供給できる回路を考案し、検討を行う。

3 提案する非接触給電システム

3.1 共振回路について

伝送システムの等価回路図をFig.1に示す。 C_1 、 C_2 の各2枚の極板が非接触となっており、入出力合わせて4枚の極板を用いているが、両者はGNDを共有しているため、1段あたり、3つの π 型配置の容量で表すことができる。次にFig.1の1段を取り上げ、負荷をショートにした時の電源から負荷側を見たときのCとLのみのリアクタンス X_{in} を式(1)に示す。リアクタンスが0になるかその逆数のサセプタンスが0になることが共振の条件であり、式(1)より、 C_1 、 C_{11} 、 C_{12} 、 L_1 での並列共振と、 C_1 、 C_{12} 、 L_1 での直列共振があることがわかる。本稿では共振状態が安定しやすい直列共振を用いて、電力伝送を行う。

$$X_{in} = \frac{1 - \omega^2 L_1 (C_1 + C_{12})}{j\omega \{C_1 + C_{11} - \omega^2 L_1 (C_1 C_{11} + C_{11} C_{12} + C_{12} C_1)\}} \quad (1)$$

次に、1段目を取り上げ、1MHzで共振させた際に、 C_{12} を変動させた場合のリアクタンス X_{in} の変動を回路シミュレータで確認する。結果をFig.2に示す。Fig.2より、 C_{12} を変動させた場合にはリアクタンス X_{in} は大きくずれることが確認できる。従って、本論文で用いる直列共振では後ろ側のシャントのキャパシタンス C_{12} の最小化が必要である。

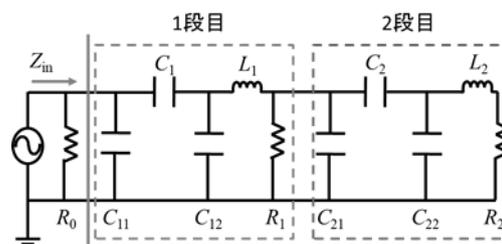


Fig.1 伝送システムの等価回路

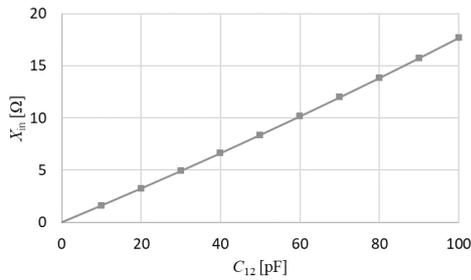


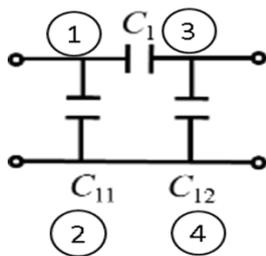
Fig.2 シャント C と X_{in} の関係

3.2 後ろ側のシャント C を減らす構造

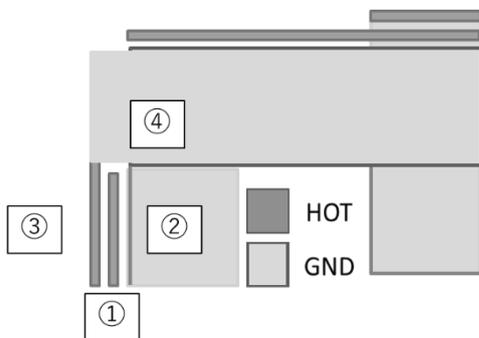
今回使用するシステムの等価回路図と概略図を Fig.3 に、作製した実験機を Fig.4 に示す。前節で、各段の後ろ側のキャパシタンス C_{12} 、 C_{22} が悪影響を与えることがわかったため、 C_{12} 、 C_{22} が生まれにくい構造を考えた。Fig.3, Fig.4 の丸数字はそれぞれ対応しており、③と④の極板を垂直に配置することで生まれにくくし、②と③で生じるキャパシタンスは①の極板がブロックする。得られた各キャパシタの値を表 1 に示す。共振用に L を装荷し、1MHz で共振させる。

4 電力分配特性

回転の変化によって抵抗値が変動するサーボモータを想定し、変動負荷に対して、要求する電力を供給できているかを確認するために、電力伝送を行う。変動負荷以外の抵抗の値は 100Ω に固定し、変動負荷の範囲は 100Ω から 1000Ω の 10 倍とする。変動負荷と各負荷に供給している電力の関係を Fig.5 に示す。



(a) 等価回路図



(b) 概略図

Fig.3 等価回路図と概略図

表.1 素子値

	1段目		2段目
C_1	427.2 pF	C_2	449.8 pF
C_{11}	33 pF	C_{21}	31 pF
C_{12}	2 pF	C_{22}	12 pF

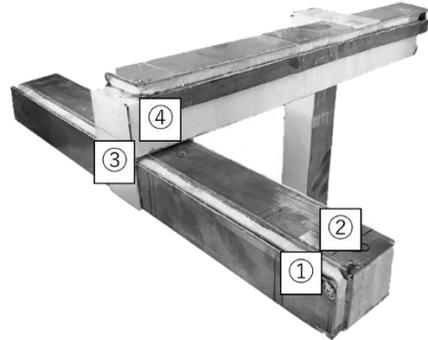


Fig.4 作製した実験機

実線が実測値、破線が回路シミュレーション値となっており、 P の添え字は Fig.1 の R の添え字と対応している。変動負荷に供給する電力は抵抗値の変動に追従し、それ以外の負荷に供給する電力は一定となった。

5 まとめ

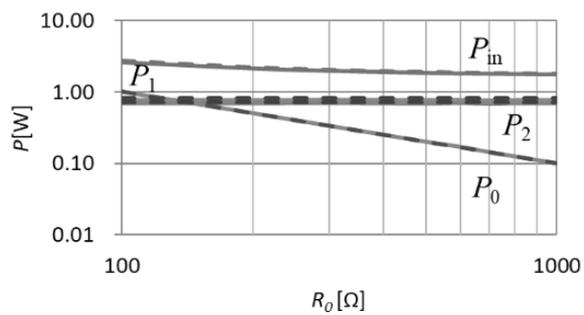
本稿では、直交座標型ロボットへの抵抗値が変動する複数負荷に対し、電力供給することを目指し、原理確認をすることができた。

今後、サーボモータの抵抗値が現実に変動する範囲の調査、情報の伝送についての検討を行い、実機への実装を目標とする。

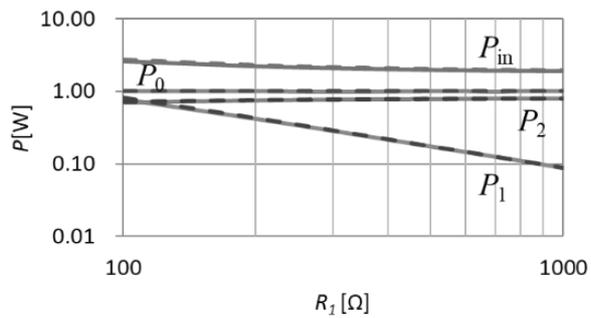
参考文献

[1] 川辺健太郎, 張陽軍, 粟井郁雄, “ディスクリピータを用いた多関節アームロボットへの新しい非接触給電法”, 信学技報, WPT2016-23, pp.19-24, 2016

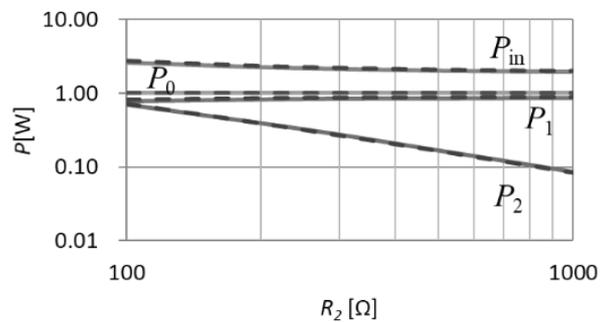
[2] 大黒康平, 植村渉, 粟井郁夫, “直交座標型ロボットへの電界結合 WPT システムの開発”, 電子情報通信学会総合大会, p.506 2018 年 3 月 21 日



(a) R_1 を変動



(b) R_1 を変動



(c) R_1 を変動

Fig.5 各負荷に供給する電力

© 2018 Special Interest Group on AI Challenges
Japanese Society for Artificial Intelligence
一般社団法人 人工知能学会 AI チャレンジ研究会

〒162 東京都新宿区津久戸町 4-7 OS ビル 402 号室 03-5261-3401 Fax: 03-5261-3402

(本研究会についてのお問い合わせは下記をお願いします。)

AI チャレンジ研究会

主 査

公文 誠

熊本大学 大学院先端科学研究部

Executive Committee Chair

Makoto Kumon

Faculty of Advanced Science and
Technology, Kumamoto University
kumon @ gpo.kumamoto-u.ac.jp

主 幹 事 / 担 当 幹 事

光永 法明

大阪教育大学 教員養成課程 技術教育講座

Secretary

Noriaki Mitsunaga

Department of Technology Education,
Osaka Kyoiku University

担 当 幹 事

植村 涉

龍谷大学 理工学部 電子情報学科

Wataru Uemura

Department of Electronics and Informat-
ics, Faculty of Science and Technology,
Ryukoku University

幹 事

鈴木 麗璽

名古屋大学 大学院情報学研究科 複雑系科学専攻

Reiji Suzuki

Department of Complex Systems Science,
Graduate School of Informatics,
Nagoya University

中 臺 一 博

(株) ホンダ・リサーチ・インスティテュート・
ジャパン / 東京工業大学 工学院
システム制御系

Kazuhiro Nakadai

Honda Research Institute Japan Co., Ltd.
/ Department of Systems and Control
Engineering, School of Engineering,
Tokyo Institute of Technology