

## AI チャレンジ研究会(第 44 回)

*Proceedings of the 44th Meeting of Special Interest Group on AI Challenges*

### CONTENTS

- 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 ..... 1  
三船哲史\*1, 中島智晴\*1, 秋山英久\*2  
\*1 大阪府立大学, \*2 福岡大学
- RoboCup における SIRMs ファジィモデルを用いた敵位置予測と 1 対 1 の守備への応用 ..... 7  
中出知成\*1, 中島智晴\*1, 秋山英久\*2, 関宏理\*3  
\*1 大阪府立大学, \*2 福岡大学, \*3 大阪大学
- パーティクルフィルタを用いた人型ロボットの立脚姿勢時の安定性を考慮した実時間模倣 ..... 13  
近藤耀, 山本将平, 高橋泰岳  
福井大学
- RoboCup 小型リーグにおける行動系列に基づく相手戦略の学習 ..... 17  
安達勇介, 伊藤正英, 成瀬正  
愛知県立大学
- ロボカップ小型リーグロボットのモデル予測制御における設計パラメータと性能との関係 ..... 24  
日下部宏幸, 伊藤正英, 成瀬正  
愛知県立大学
- 物体認識における背景の有無による学習効果に関する一考察 ..... 30  
津田諒太, 植村渉  
龍谷大学

日時: 2016年3月26日

場所: 愛知工業大学 八草キャンパス 新2号館2階201室

*Aichi Institute of Technology, Aichi, March 36, 2016*

社団法人人工知能学会

Japanese Society for Artificial Intelligence

# 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

## Switching Game Tactics using Kick Distribution in RoboCup Soccer

三船 哲史<sup>\*1</sup>, 中島 智晴<sup>\*1</sup>, 秋山 英久<sup>\*2</sup>

Satoshi MIFUNE<sup>\*1</sup>, Tomoharu NAKASHIMA<sup>\*1</sup>, Hidehisa AKIYAMA<sup>\*2</sup>

大阪府立大学<sup>\*1</sup>, 福岡大学<sup>\*2</sup>

Osaka Prefecture University<sup>\*1</sup>, Fukuoka University<sup>\*2</sup>

{satoshi.mifune@cs, tomoharu.nakashima@kis}.osakafu-u.ac.jp, akym@fukuoka-u.ac.jp

### Abstract

In RoboCup soccer simulation 2D league, two dimensional agents play a soccer game on a 2D field. It is generally known that there is no perfect strategy that defeats any others. Even if a strategy is better than one particular strategy, it could be always inferior to another strategy. Therefore, it is necessary to have multiple strategies available and switch them according to the opponent's one. Although it is not possible to exactly know the strategy that is used by an opponent team, that strategy is generally well represented by how the opponent's players manipulate the ball during the game. This paper proposes a method that switches team's strategies according to the kick distribution of the opponent team.

### 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、それぞれにおいて活発な研究、開発が行われている。RoboCup サッカーでは、競技で勝利することが重要視されているが、ただ単に勝利するだけではなく、ボールを支配し、確実に勝利することが望まれている。そのためには、パスやドリブル等の行動を洗練すること、相手の行動を予測して意思決定を行うこと、相手チームに合わせた戦術をとることなどが必要となっている。本論文では相手チームとの相性を考慮して戦術を切り替える方法に焦点をあてる。相手チームとの相性を調べるためには、相手チームを分析しなければならない。そのために、本研究では試合ログを用いて相手チームをクラスタリングする。RoboCup サッカーシミュレーション

2D リーグでは、試合終了後にログが出力される。試合ログには、プレイヤーやボールの位置情報、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。試合ログは主に過去の大会の傾向の分析などに用いられている。例えば、Gabelら [2] は、過去の大会の傾向の分析を行い、チームの戦術を定量的に評価した。Abreuら [3] は、ログの情報をを用いてロボットのサッカーと人間のサッカーを比較した。

RoboCup サッカーシミュレーション 2D リーグでは、毎年各チームが変化を加えることにより、フィールドの片側に選手を固め攻撃する戦術や守備を偏重する戦術など、多種多様な戦術が存在している。戦術には相性があるため、1つの戦術で全てのチームに勝つことは困難である。そのため、試合中に自チームと相手チームとの相性を判断し、適切な戦略を選択することで試合を有利に進める必要がある。RoboCup サッカーでは、相手チームの戦術や行動を識別するための研究が積極的に行われている。例えば、Rileyら [4][5] は、フィールドを分割し、相手プレイヤーのポジションやパス、ドリブルを記録することにより、相手チームを識別する手法を提案した。Iglesiasら [6] は、トライ木を用いてチームの行動を比較する手法を提案している。しかし、相手チームがとっている戦術を確実に判断することはできていない。そこで本論文では、試合中のキック情報に注目し、戦術をキック分布として表現する。そして、試合中に相手チームとの相性を判断し、戦術を切り替える手法を提案する。提案手法では、Earth Mover's Distance (EMD) [7] をキック分布の類似度として、戦術をクラスタリングする。そして、ハーフタイムにクラスタリング結果と前半戦のキック分布から戦術の切り替えの判断を行う。数値実験では、JapanOpen2015に出場したチームと対戦を行い、得られた試合ログをクラスタリングする。クラスタリング結果と前半戦のキック分布から戦術を切り替えるかの判断を行うことにより、チームの性能がどのように変化するのかを調査する。

## 2 RoboCup サッカーシミュレーション 2D リーグ

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。本論文では、RoboCup サッカーシミュレーションを研究の対象とする。サッカーシミュレーションでは、実機を使用せずに、コンピュータ内に用意された仮想フィールド上でサッカー競技を行う。サッカーシミュレーションには 2D リーグと 3D リーグがある。本論文では 2D リーグを扱う。Figure 1 に 2D リーグの試合の様子を示す。2D リーグでは、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のベクトルとして表される。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。各プレイヤー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おり、制限された視覚情報や聴覚情報からドリブルやパス等の行動選択を行う。

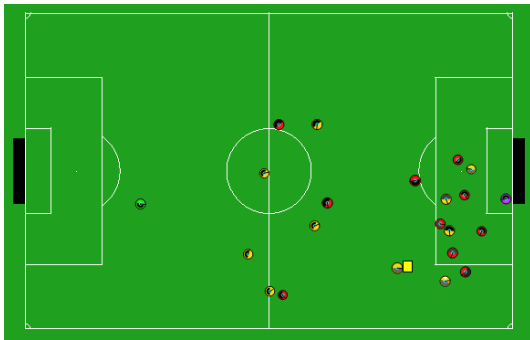


図 1: Soccer simulation 2D league

RoboCup サッカーシミュレーション 2D リーグでは、試合終了後にサーバーから試合ログが出力される。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、試合ログを使うことにより終了した試合を再生することができる。

### 3 戦術のクラスタリング

本論文では、チーム戦術がキック分布に表れるものと仮定する。チーム戦術をグループ分けするために、キック分布をクラスタリング手法により分析する。キック分布をクラスタリングするための距離尺度として Earth Mover's Distance (EMD) を用いる。

### 3.1 キック分布

本論文では、チーム戦術を表す特徴としてキック分布を考える。キック分布とは試合中にプレイヤーがキックした位置とボールの移動量との組の集合である。ボールの移動量を重みとしてキック位置に割り当てる。抽出するキックはパス、ドリブルのみとする。ボールをフィールドの外に出したキックや、相手チームにインターセプトされたパスはキックに含まれない。ボールをキックしたプレイヤーの位置 ( $x$  座標,  $y$  座標) をベクトル  $p_i$  とし、そのキックによってボールが動いた距離をベクトルの重み  $w_{p_i}$  をとする。Figure 2 は opuSCOM 対 agent2d の試合ログから得られたキック分布である。Figure 2 において、縦棒の座標はプレイヤーがキックした位置を示し、高さはベクトルの重み、すなわちそのキックによってボールが動いた距離を表している。

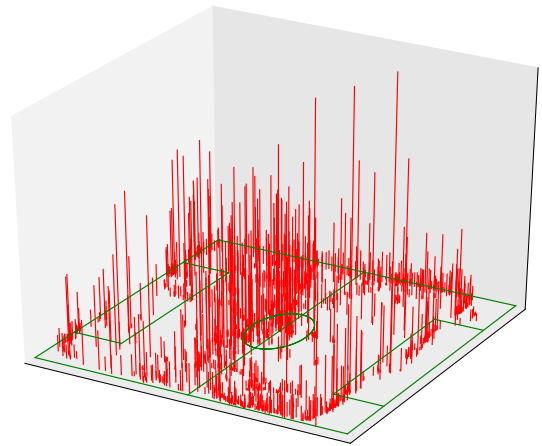


図 2: An example of kick distribution

### 3.2 Earth Mover's Distance (EMD)

EMD は分布間の距離を表すものであり、類似画像検索や類似音楽検索、文書分類 [8] などの分野で用いられている。EMD は分布間の距離の計算を輸送問題として定式化し、一方の分布を各場所における供給量、他方の分布を需要量として最小輸送コストを分布間の距離と定義する。輸送問題とは、複数の供給地と需要地があり、需要を満たすように供給地から需要地に輸送を行うときの最小コストを求める問題である。EMD を求める際、分布は重み付き集合として表現される。一方の分布  $P$  を集合として表現すると、 $P = \{(p_1, w_{p_1}), \dots, (p_m, w_{p_m})\}$  となる。分布  $P$  は  $m$  個の特徴量で表現されており、 $p_i$  は特徴量ベクトル、 $w_{p_i}$  はその特徴量に対する重みである。同様に、もう一方の分布  $Q$  も集合として表現すると、 $Q = \{(q_1, w_{q_1}), \dots, (q_m, w_{q_m})\}$  となる。EMD は、2 つの分布の特徴量の数が異なる場合でも計算が可能であるという特徴を持っている。

表 1: Definitions of kick distributions

No.	Weight of feature	Extracted kick	Target team
①	Distance	Dribbles and passes	Both
②	Distance	Dribbles and passes	opuSCOM
③	Distance	Dribbles and passes	Opponent
④	$x$ -distance	Dribbles and passes	Both
⑤	$x$ -distance	Dribbles and passes	opuSCOM
⑥	$x$ -distance	Dribbles and passes	Opponent
⑦	Distance	Passes	Both
⑧	Distance	Passes	opuSCOM
⑨	Distance	Passes	Opponent
⑩	$x$ -distance	Passes	Both
⑪	$x$ -distance	Passes	opuSCOM
⑫	$x$ -distance	Passes	Opponent

### 3.3 EMD によるキック分布間の距離計算

本論文では、キック分布間の距離を計算するために EMD を用いる。2 つのキック分布  $P, Q$  間の EMD を求めることを考える。EMD を適用するためには、キック分布を重み付き集合として表現する必要がある。  $P = \{(p_1, w_1), (p_2, w_2), \dots, (p_m, w_m)\}$  とすると、  $P$  は  $m$  個のキックで構成されており、特徴量ベクトル  $p_i$  はプレイヤーが  $i$  回目にキックした位置、重み  $w_i$  はそのキックによってボールが動いた距離を表す。同様に、もう一方のキック分布  $Q$  も集合として表現すると、  $Q = \{(q_1, w_1), (q_2, w_2), \dots, (q_n, w_n)\}$  となる。ここで、  $p_i$  と  $q_j$  の距離計算にはユークリッド距離を用いる。

### 3.4 クラスタリングの手順

キック分布をグループに分けるため、階層的クラスタリングを用いる。階層的クラスタリングの手順を以下に記す。

- Step 1: 個々のキック分布を 1 つのクラスタと設定。
- Step 2: 全てのクラスタ間の距離を計算。
- Step 3: 最も距離が小さい 2 つのクラスタを併合。
- Step 4: クラスタが 1 つになれば終了。クラスタが 2 つ以上ある場合は Step 2 へ

クラスタ間の距離は群平均法によって求める。群平均法とは、2 つのクラスタの全ての要素の組み合わせについて距離を求め、その平均値をクラスタ間の距離として用いる手法である。クラスタリング終了後、デンドログラムを作成する。クラスタの数を 2 から 5 まで変化させて、得られたクラスタに属する試合ログの勝敗を確認する。その中で最も勝敗のエントロピーが小さいクラスタ数を使用する。

## 4 戦術の切り替え

Figure 3 に戦術の切り替えの概要を示す。事前に試合データを収集し、3 章で述べたキック分布のクラスタリング結果と各クラスタの勝敗予測ラベルをデータベースに保存しておく。勝敗予測ラベルは、各クラスタ内の勝利数を調

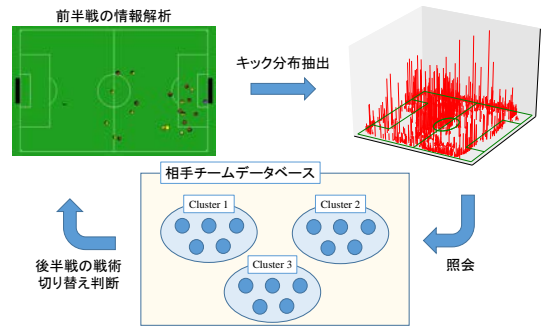


図 3: Overview of switching game tactics

べ、勝利数が過半数を占める場合は「勝利」、同数の場合は「不明」、それ以外を「敗北」とラベル付けを行う。試合前に 2 つの戦術を用意し、どちらか一方を前半戦に使用する戦術とする。次に、前半戦のキック分布を抽出する。ハーフタイムに、抽出したキック分布と、各クラスタとの距離を計算する。距離が最も近いクラスタにそのキック分布が属するものとする。キック分布が属するクラスタのラベルが「勝利」であれば、後半戦においても前半戦と同じ戦術を使用する。「不明」または「敗北」であれば、後半戦で使用する戦術を切り替える。

## 5 数値実験

数値実験では、実際の試合ログを用いて分類実験を行う。次に、前半戦のキック分布から勝敗予測を行うことによって、クラスタリング結果を評価する。また、提案手法を組み込んだチームを使って試合を行い、チームの性能がどのように変化するかを調査する。数値実験において、キック分布は特徴量の重みとして用いるボールの移動距離とキックの種類、抽出するキックを行ったチームの組み合わせによる 12 種類を使用する。使用するキック分布を Table 1 に示す。

### 5.1 戦術のクラスタリング

opuSCOM と agent2d, KU\_BOST, TsubameGaeshi, Ri-one\_A, Ri-one\_B, FiftyStorms, Toyosu-Galaxy の 7

表 3: The correct answer rate (%)

Opponent	①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑪	⑫
agent2d	83	80	90	89	80	75	82	93	94	68	90	87
KU_BOST	74	80	31	50	80	35	77	41	40	74	33	59
Ri-one_A	54	52	50	49	52	44	43	50	56	40	47	41
Ri-one_B	76	87	22	32	87	28	87	48	41	81	51	79
TsubameGaeshi	91	76	94	86	76	76	79	94	93	78	91	87
FiftyStorms	79	54	77	69	54	70	60	77	78	72	62	78
Toyosu-Galaxy	83	74	87	70	74	81	73	86	84	69	81	70
Average	77.1	71.9	64.4	63.6	71.9	58.4	71.6	69.9	69.4	68.9	65.0	71.2

チームを 10 試合ずつ対戦させ、試合ログを作成する。opuSCOM は本研究室で開発を進めているチームである。agent2d [9] はベースチームとして公開されているチームであり、他のチームは JapanOpen2015 に参加したチームである。試合ログからキックした位置毎の  $x$  座標,  $y$  座標, キックによってボールが動いた距離を抽出し, 12 種類のキック分布を作成する。全てのキック分布の組み合わせの EMD を計算する。求めたキック分布間の距離を用いて, 階層的クラスタリングを行う。

## 5.2 勝敗予測

クラスタリング結果を用いて, 前半戦のキック分布から試合の勝敗予測を行う。勝敗予測に使用する情報は前半戦のキック分布のみとし, スコア情報などの他の情報は使用しない。まず, 各クラス内で勝敗予測対象のチームが勝利した数を調べる。勝利数が過半数を占める場合は「勝利」, 同数の場合は「不明」, それ以外を「敗北」とクラスにラベル付けを行う。次に, 勝敗予測対象の試合ログにおける前半戦のキック分布を抽出する。抽出したキック分布と, クラスタリングにより得られた各クラスとの距離を計算する。距離が最も近いクラスにそのキック分布が属するものとする。キック分布が属するクラスのラベルを予測結果とする。勝敗予測に使用する試合ログは opuSCOM と agent2d, KU\_BOST, TsubameGaeshi, Ri-one\_A, Ri-one\_B, FiftyStorms, Toyosu-Galaxy の 7 チームとの対戦を新たに 100 試合ずつ行って作成したものである。キック分布が属するクラスのラベルが試合結果と一致しているかどうかで勝敗予測の精度を調査する。

## 5.3 チームの性能評価

提案手法を組み込んだ opuSCOM (キック分布の種類毎) と agent2d, KU\_BOST, TsubameGaeshi, Ri-one\_A, Ri-one\_B, FiftyStorms, Toyosu-Galaxy の 7 チームを新たに 100 試合ずつ対戦させる。提案手法を組み込んだチームでは, あらかじめ 2 つの戦術を用意し, 前半戦に使用する戦術を固定する。そして, ハーフタイムにコーチエージェントがもう一方の戦術を切り替えるかどうかの判断を行う。従来のチームと試合結果を比較し, チームの性能を調査する。

## 6 実験結果

### 6.1 戦術のクラスタリング

opuSCOM と agent2d, KU\_BOST, TsubameGaeshi, Ri-one\_A, Ri-one\_B, FiftyStorms, Toyosu-Galaxy の 7 チームを 10 試合ずつ対戦させ, 作成した試合ログを 12 種類のキック分布毎にクラスタリングした。クラスタリング結果より, 使用するキック分布によって差があるが, opuSCOM と相性の悪いチームや特徴的な攻撃を行う FiftyStorms などが分類されていることがわかった。

### 6.2 勝敗予測

opuSCOM と agent2d, KU\_BOST, TsubameGaeshi, Ri-one\_A, Ri-one\_B, FiftyStorms, Toyosu-Galaxy の 7 チームを新たに 100 試合ずつ対戦させ, 作成した試合ログから前半戦のキック分布を抽出した。12 種類のキック分布毎にクラスタリング結果と前半戦のキック分布から, 勝敗予測を行った。勝敗予測に使用した試合の勝敗を Table 2 に示す。また, キック分布の種類毎の正答率を Table 3 に示す。Table 3 より, キック分布①を使用したとき最も正答率が高いことが読みとれる。勝率が 5 割に近い Ri-one\_A との対戦の勝敗はどのキック分布を用いても上手く予測することができなかった。

表 2: Game results

Team	Win	Lose	Draw
agent2d	95	0	5
KU_BOST	19	68	13
Ri-one_A	51	22	27
Ri-one_B	12	74	14
TsubameGaeshi	96	2	2
FiftyStorms	80	9	11
Toyosu-Galaxy	88	7	5

### 6.3 チームの性能評価

提案手法を組み込んでいない opuSCOM と提案手法を組み込んだ opuSCOM (キック分布の種類毎) で agent2d, KU\_BOST, TsubameGaeshi, Ri-one\_A, Ri-one\_B, FiftyStorms, Toyosu-Galaxy の 7 チームと 100 試合ずつ対戦を行った。提案手法を組み込んでいない従来のチームで 100 試合ずつ対戦した際の勝敗を Table 4 に示

す。戦術の切り替えを行った際、後半戦において使用する戦術での勝敗を Table 5 に示す。Table 5 より、従来のチームと相性の悪い KU\_BOST, Ri-one\_B に対して従来より勝率が高い戦術であることが読みとれる。一方で、他の相手に対しては従来のチームのほうが勝率が高いことがわかる。提案手法を組み込んだチーム（キック分布の種類毎）を使ったときの勝敗を Table 6~17 に示す。Table 6~17 より、提案手法を組み込んだチームでは、どのキック分布を使用したときも KU\_BOST, Ri-one\_B に対して勝率が向上していることがわかる。これは相性が悪いチームに対して戦術の切り替えが有効に働いていることを示している。また、切り替え先の戦術と相性の悪い FiftyStorms に対して勝率を維持できていることが多く、相手チームの戦術に適した戦術を選択できていることがわかる。

表 4: Game results: previous team

Team	Win	Lose	Draw	AG	AGA
agent2d	94	2	4	5.64	1.26
KU_BOST	18	69	13	1.00	2.23
Ri-one_A	58	25	17	2.08	1.43
Ri-one_B	9	59	32	0.65	1.57
TsubameGaeshi	97	1	2	5.98	1.16
FiftyStorms	75	17	8	5.33	2.75
Toyosu-Galaxy	87	5	8	4.89	1.41

(AG: Average Goals, AGA: Average Goals Against)

表 5: Game results: the strategy after switching

Team	Win	Lose	Draw	AG	AGA
agent2d	84	7	9	4.77	1.86
KU_BOST	68	14	18	3.53	2.14
Ri-one_A	38	38	24	2.01	2.34
Ri-one_B	45	22	23	2.41	1.90
TsubameGaeshi	88	3	9	5.63	1.58
FiftyStorms	51	37	12	5.63	4.19
Toyosu-Galaxy	82	7	9	4.49	1.71

表 6: Game results: kick distribution①

Team	Win	Lose	Draw	AG	AGA
agent2d	93	1	6	5.75	1.39
KU_BOST	45	37	18	2.35	2.18
Ri-one_A	40	42	18	2.14	2.12
Ri-one_B	33	43	24	1.44	1.69
TsubameGaeshi	95	2	3	5.71	1.37
FiftyStorms	81	9	10	5.58	2.79
Toyosu-Galaxy	87	4	9	4.55	1.36

表 7: Game results: kick distribution②

Team	Win	Lose	Draw	AG	AGA
agent2d	93	2	5	5.45	1.35
KU_BOST	41	39	20	2.27	2.31
Ri-one_A	52	31	17	2.45	1.83
Ri-one_B	28	50	22	1.33	1.97
TsubameGaeshi	90	5	5	5.41	1.41
FiftyStorms	68	24	8	5.46	3.65
Toyosu-Galaxy	80	5	15	4.61	1.59

表 8: Game results: kick distribution③

Team	Win	Lose	Draw	AG	AGA
agent2d	89	3	8	5.53	1.37
KU_BOST	23	56	21	1.61	2.34
Ri-one_A	42	45	13	2.22	2.40
Ri-one_B	16	60	24	1.05	2.08
TsubameGaeshi	90	4	6	5.92	1.39
FiftyStorms	75	16	9	5.24	3.05
Toyosu-Galaxy	86	3	11	4.85	1.50

表 9: Game results: kick distribution④

Team	Win	Lose	Draw	AG	AGA
agent2d	90	4	6	5.73	1.38
KU_BOST	39	47	14	2.16	2.38
Ri-one_A	46	43	11	2.29	1.99
Ri-one_B	18	48	34	1.03	1.80
TsubameGaeshi	96	1	3	6.17	1.04
FiftyStorms	74	14	12	5.78	3.54
Toyosu-Galaxy	82	11	7	4.84	1.58

表 10: Game results: kick distribution⑤

Team	Win	Lose	Draw	AG	AGA
agent2d	91	3	6	5.36	1.26
KU_BOST	35	40	25	2.06	2.29
Ri-one_A	45	33	22	2.07	1.65
Ri-one_B	25	44	31	1.17	1.97
TsubameGaeshi	94	2	4	5.81	1.25
FiftyStorms	69	17	14	5.23	2.94
Toyosu-Galaxy	85	4	11	4.85	1.69

表 11: Game results: kick distribution⑥

Team	Win	Lose	Draw	AG	AGA
agent2d	91	1	8	5.78	1.24
KU_BOST	24	57	19	1.68	2.50
Ri-one_A	33	37	30	2.05	2.12
Ri-one_B	18	49	33	0.92	2.12
TsubameGaeshi	87	9	4	5.67	1.42
FiftyStorms	80	14	6	5.59	3.07
Toyosu-Galaxy	82	7	11	4.47	1.59

表 12: Game results: kick distribution⑦

Team	Win	Lose	Draw	AG	AGA
agent2d	90	5	5	5.51	1.12
KU_BOST	35	37	28	2.11	2.30
Ri-one_A	47	35	18	2.27	1.97
Ri-one_B	24	52	24	1.04	1.73
TsubameGaeshi	92	4	4	5.56	1.23
FiftyStorms	73	20	7	5.57	3.38
Toyosu-Galaxy	87	2	11	4.48	1.54

表 13: Game results: kick distribution⑧

Team	Win	Lose	Draw	AG	AGA
agent2d	94	4	2	5.43	1.26
KU_BOST	29	45	26	2.06	2.27
Ri-one_A	41	39	20	2.19	1.98
Ri-one_B	19	54	27	1.18	2.15
TsubameGaeshi	93	4	3	5.55	1.25
FiftyStorms	67	20	13	5.41	3.53
Toyosu-Galaxy	86	5	9	4.51	1.33

戦術を自動的に作成することなどが挙げられる。

表 14: Game results: kick distribution⑨

Team	Win	Lose	Draw	AG	AGA
agent2d	92	3	5	5.22	1.34
KU_BOST	27	39	34	1.90	2.46
Ri-one_A	49	34	17	2.09	1.73
Ri-one_B	14	63	23	0.98	2.26
TsubameGaeshi	95	1	4	5.78	1.12
FiftyStorms	73	15	12	5.20	3.26
Toyosu-Galaxy	83	8	9	4.46	1.39

表 15: Game results: kick distribution⑩

Team	Win	Lose	Draw	AG	AGA
agent2d	89	2	9	5.45	1.47
KU_BOST	36	51	23	2.13	2.49
Ri-one_A	53	36	11	2.40	2.04
Ri-one_B	23	56	21	1.14	1.99
TsubameGaeshi	95	2	3	5.74	1.06
FiftyStorms	67	21	12	5.29	3.24
Toyosu-Galaxy	83	3	11	4.69	1.40

表 16: Game results: kick distribution⑪

Team	Win	Lose	Draw	AG	AGA
agent2d	92	3	5	5.23	1.33
KU_BOST	26	54	20	1.91	2.35
Ri-one_A	44	32	24	2.03	1.71
Ri-one_B	18	63	19	1.09	2.21
TsubameGaeshi	97	1	2	5.60	1.04
FiftyStorms	63	23	14	5.23	2.65
Toyosu-Galaxy	86	7	7	4.58	1.56

表 17: Game results: kick distribution⑫

Team	Win	Lose	Draw	AG	AGA
agent2d	90	4	6	5.88	1.54
KU_BOST	33	53	14	2.15	2.43
Ri-one_A	53	26	21	2.26	1.66
Ri-one_B	26	57	17	1.25	2.40
TsubameGaeshi	96	1	3	5.53	1.31
FiftyStorms	54	35	11	5.05	4.06
Toyosu-Galaxy	90	5	5	4.88	1.42

## 7 おわりに

本論文ではチームの戦術をキック分布により表現した。そして、キック分布の類似度として EMD を用いて、実際の試合ログをクラスタリングした。また、提案手法をチームに組み込み、試合中に相手チームとの相性を判断し、戦術の切り替えを行った。数値実験では、提案手法を組み込んだチームで対戦を行い、性能を調査した。相性の悪いチームに対しての勝率が上がったことから、戦術の切り替えが有効であることを示した。今後の課題として、分類や勝敗予測の精度を向上させることや、各クラスに適した

## 参考文献

- [1] Hiroaki 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] Thomas Gabel, Martin Riedmiller, “On Progress in RoboCup: The Simulation League Showcase,” *The 14th RoboCup 2010 Symposium*, pp.36-47, Springer, Berlin Heidelberg, 2010.
- [3] Pedro Abreu, João Moreira, Israel Costa, Daniel Castelão, Luis Reis, Júlio Garganta, “Human Versus Virtual Robotics Soccer: A Technical Analysis,” *European Journal of Sport Science*, Vol.12, No.1, pp.26-35, Taylor & Francis, 2011.
- [4] Patrick Riley, Manuela Veloso, “On Behavior Classification in Adversarial Environments,” *Distributed Autonomous Robotic Systems*, pp.371-380, Springer, 2000.
- [5] Patrick Riley, Manuela Veloso, “Recognizing Probabilistic Opponent Movement Models,” *RoboCup-2001: Robot Soccer World V*, Vol.2377, pp.205-245, Springer, 2002.
- [6] Iglesias, J.A., Ledezma, A., Sanchis, A.: “A Comparing Method of Two Team Behaviors in the Simulation Coach Competition”, *Modeling Decisions for Artificial Intelligence*, Vol.3885 of LNCS, pp.117–128, Springer, 2006.
- [7] Yossi Rubner, Carlo Tomasi and Leonidas J.Guibas, “The Earth Mover’s Distance as a Metric for Image Retrieval”, *International Journal of Computer Vision*, Vol.40, No.3, pp.130-137, 2000.
- [8] 柳本 豪一, 大松繁, “Earth Mover’s Distance を用いたテキスト分類”, 人工知能学会全国大会論文集, 2007.
- [9] 秋山 英久, RoboCup サッカーシミュレーション 2D 必勝ガイド, 秀和システム, 2006.



# RoboCupにおける SIRMs ファジィモデルを用いた敵位置予測と1対1の守備への応用 Opponent's Position Prediction using SIRMs Fuzzy Models

and Its Application to One-to-One Defense for RoboCup

中出 知成<sup>\*1</sup>, 中島 智晴<sup>\*1</sup>, 秋山 英久<sup>\*2</sup>, 関 宏理<sup>\*3</sup>

Tomonari NAKADE<sup>\*1</sup>, Tomoharu NAKASHIMA<sup>\*1</sup>, Hidehisa AKIYAMA<sup>\*2</sup>, Hirosato SEKI<sup>\*3</sup>

大阪府立大学<sup>\*1</sup>, 福岡大学<sup>\*2</sup>, 大阪大学<sup>\*3</sup>

Osaka Prefecture University<sup>\*1</sup>, Fukuoka University<sup>\*2</sup>, Osaka University<sup>\*3</sup>

{swa01200@edu, tomoharu.nakashima@kis}.osakafu-u.ac.jp, akym@fukuoka-u.ac.jp, seki@sys.es.osaka-u.ac.jp

## Abstract

RoboCup サッカーでは、敵プレイヤーの動きに対して適切に対応することが重要である。特に、1対1の守備では味方プレイヤーが一人で対応しなければならないため、敵がどのような行動を取るかを予測することが重要になる。本論文では、RoboCup サッカーシミュレーション 2D リーグを題材として、SIRMs ファジィモデルを用いた1対1の守備における敵位置予測の手法を提案する。数値実験では、教師信号による予測精度の検証を行い、さらに提案手法を組み込んだチームの1対1の守備における性能を調査する。

## 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 サッカーでは、競技で勝利することが重要視されているが、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る。そのためには、敵チームの攻撃に対し適切な守備を行うことが重要である。特に、1対1の場面における守備は試合中の機会も多く、その完成度が試合の勝敗にも影響する。

チーム開発においては、守備の動きを実装するために開発者が手作業で調整を繰り返しているのが実情である。そのため、敵チームの攻撃に対して適切な守備が行えていない可能性がある。また、どのようにすれば1対1の守備で適切な守備を行えるかは敵プレイヤーの行動に大きく影響されるため、想定される全ての攻撃について手作業で調整を行うとコストがかかってしまう。そこで、相手の行動を予測し、適切な守備を選択する手法が必要となる。

例えば、Gabelら[2]は、NeuroHassleと呼ばれる手法を提案し、1対1の守備を繰り返し、学習させることで守備性能の向上を行った。

本論文では、SIRMs (Single Input Rule Modules:SIRMs) ファジィモデル[3]を用いた敵位置予測とそれを用いた1対1の守備への応用を提案する。1対1の守備では敵の動きが重要であるが、もし敵が移動する位置を予測できれば、味方の行動決定を容易にすることができる。そのための手法として、SIRMs ファジィモデルを用いる。数値実験では、SIRMs ファジィモデルによる敵位置予測を行い、予測精度について検証を行う。また、敵位置予測が正しく行えたとしても適切なタイミングの位置を使っていなければ有効な守備にならない。そこで、提案手法を組み込んだチームの1対1の守備性能について、予測タイミングを変更して評価を行う。

## 2 RoboCup

### 2.1 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．Figure 1, 2 に 2D リーグと 3D リーグの試合の様子を示す．2D リーグでは，基本的な動作はコマンドとして実装されている．そのため，2D リーグでは，高レベルな意思決定を主な研究対象としている．一方，3D リーグでは，エージェントはヒューマノイドロボットで形成されているため，基本的な動作を関節から制御する必要があり，基本的な動作が非常に重要である．本論文では 2D リーグを扱う．

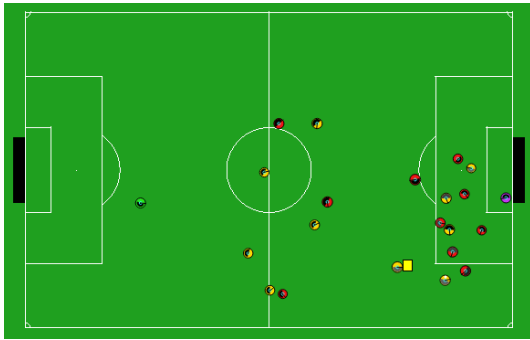


図 1: 2D Simulation League



図 2: 3D Simulation League

## 2.2 RoboCup サッカーシミュレーション 2D リーグ

シミュレーション 2D リーグは RoboCup 創設当初から存在する最も古いリーグの一つである．2D リーグでは，実機を使用せずに，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のベクトルとして表される．2D リーグでは，低レベルの行動に関しては，kick, dash, turn などの抽象化されたコマンドにより実装されている．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

チがメッセージとして発している情報を聴覚情報として使用できる．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，視覚情報や聴覚情報にはノイズが含まれる．視覚情報には遠距離のプレイヤーやボールの情報が一部欠落し，聴覚情報についても確実に受信できる訳ではなく，コーチの発生するメッセージはプレイオン時にはプレイヤーに到達する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リアルタイムで保持することができない．

## 3 ドリブルに対する守備

本論文ではボールを持っている敵プレイヤーに対し，味方プレイヤーが 1 人で対応する動作を 1 対 1 の守備と定義する．Figure 3 に 1 対 1 の守備の例を示す．Figure 3 において，ボールを持っている敵プレイヤーに対しては 1 人で守備を行っており，他の味方プレイヤーはゴールの守備やパスに備えて他の敵プレイヤーのマークを行っている．マークを行う際に重要視されることは敵のパスを防ぐことである．そのため，自分がどのプレイヤーをマークし，どこに位置取ることでパスに備えるかなどを考える必要がある．例えば，マークするプレイヤーとボールを持ったプレイヤーの間に入るように位置し，パスを防ぐ動きを行う，あるいはパスは防がず，プレイヤーとゴールの間に入るように位置し，パスを受けた敵プレイヤーの行動を制限させるといった戦術が考えられる．これに対し，1 対 1 の守備では敵のドリブルを防ぐことが重要視される．例えば Figure 3 では，敵がドリブルで目の前にいるプレイヤーを突破すると，敵プレイヤーとゴールの間にはキーパーしか存在せず，ゴールに接近され，失点する危険性が高くなる．また，1 対 1 の守備では後方や横方向へのパスを防ぐことが困難である．そのため，敵のドリブルを止めてもパス回しにより，全体での敵の攻撃が途切れないことがある．これを 1 対 1 の守備で防ぐには敵のボールを取り，ボールを味方のものにする必要がある．1 対 1 の守備を行う際，パスを防ぐことは比較的重要なものではないが，敵の前方向へのパス

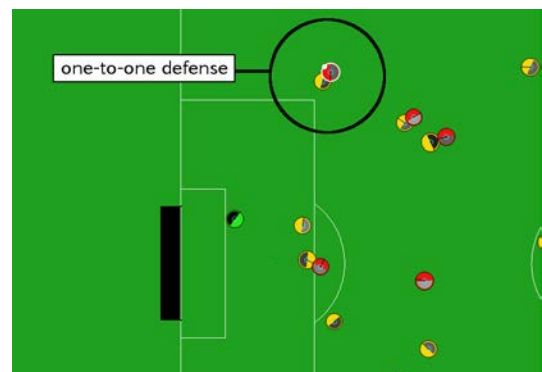


図 3: One-to-One Defense

には注意をはらう必要がある．通常，1対1の守備を行っている場合は，パスルートは制限され，敵プレイヤーにとって前方向へのパスは難しいものとなる．しかし，敵プレイヤーはドリブルで守備を行っているプレイヤーを引き離すことで，パスルートを作りパスを行うことができる．そのため，敵のパスを防ぐという観点から見ても，敵のドリブルを防ぐことは重要になる．

守備における様々な状況における対処は，開発者が手作業による調整を繰り返すことで実装の完成度を上げているのが実情である．また，敵位置予測も速度や体の向きから計算を行っている．そのため，1対1の守備が適切に行えていない可能性があり，世界大会においても1対1の守備が失敗する場面が見られた．そこで，SIRMs ファジィモデルによって敵位置予測を行うことで，1対1の守備性能を向上させる．

## 4 提案手法

### 4.1 SIRMs ファジィモデル

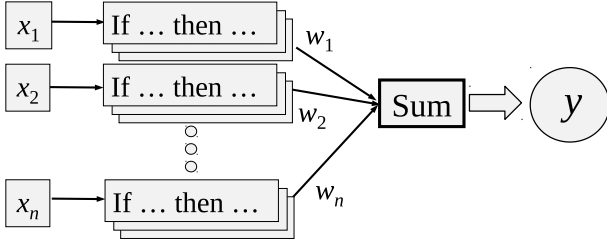


図 4: Overview of an SIRMs fuzzy model

ファジィモデルの一つとして単一入力ルール群 (Single Input Rule Modules : SIRMs) ファジィモデル [3] がある．SIRMs ファジィモデルの概形を Figure 4 に示す．各入力項目に対してルール群が一つ用意される．ルール群に含まれるファジィIf-Then ルールには対応する入力項目だけが前件部変数に含まれる．各ルール群には重視度と呼ばれる実数重みが割り当てられている．ルール群のファジィ推論結果の重視度付き総和を最終出力とする．SIRMs ファジィモデルは入力項目数が多い場合に，従来のファジィ推論モデルよりも大幅にファジィルール数とパラメータ数を削減することが可能であり，様々な制御問題へ応用されている．Figure 4 において， $x_i (i = 1, \dots, n)$  は  $i$  番目の入力項目に対応する前件部変数， $y_i (i = 1, \dots, n)$  はそのルール群の推論結果である． $h_j^i (i = 1, \dots, n, j = 1, \dots, m_i)$  は  $i$  番目のルール群における  $j$  番目のルールの前件部適合度， $c_j^i (i = 1, \dots, n, j = 1, \dots, m_i)$  は  $i$  番目のルール群における  $j$  番目のルールの後件部の実数出力値を意味する．また， $w_i (i = 1, \dots, n)$  は各入力項目の重視度， $y$  は SIRMs ファジィモデルの最終出力を表す．入力ベクトル  $(x_1, \dots, x_n)$  が与えられたとき， $h_j^i, y_i$  は以下の式で求め

られる．

$$h_j^i = A_j^i(x_i) \quad (1)$$

$$y_i = \frac{\sum_{j=1}^{m_i} h_j^i \cdot c_j^i}{\sum_{j=1}^{m_i} h_j^i} \quad (2)$$

$A_j^i (i = 1, \dots, n, j = 1, \dots, m_i)$  は  $i$  番目のルール群における  $j$  番目のルールの前件部変数  $x_i$  のメンバシップ関数であり，以下の式のガウス関数とする．

$$A_j^i = \exp\left(-\frac{(x_i - a_j^i)^2}{b_j^i}\right) \quad (3)$$

ここで， $a_j^i, b_j^i$  は  $A_j^i$  のガウス関数の形状を決めるパラメータである．最終出力  $y$  は以下の式のようにルール群の推論結果の重視度付き総和として求められる．

$$y = \sum_{i=1}^n w_i \cdot y_i \quad (4)$$

本論文では，SIRMs ファジィモデルのパラメータの学習に最急降下法を用いる．最急降下法は，評価関数の勾配情報を利用して評価値が最小となるようにモデルパラメータを修正する手法である．SIRMs ファジィモデルで学習するパラメータは，前件部変数のメンバシップ関数のパラメータ  $a_j^i, b_j^i$ ，後件部の実数出力値  $y_i$ ，および入力項目の重視度  $w_i$  である．入力パターン  $(x_1, \dots, x_n)$  に対して，理想の出力が  $y^T$ ，SIRMs ファジィモデルからの出力が  $y^0$  であったとする．SIRMs ファジィモデルの性能評価関数  $E$  を以下の式で表す．

$$E = \frac{1}{2}(y^T - y^0)^2 \quad (5)$$

式 (5) より，重視度  $w_i$ ，後件部の実数出力値  $c_j^i$ ，前件部変数のメンバシップ関数のパラメータ  $a_j^i$  と  $b_j^i$  の修正量は， $t$  を現在の学習回数， $\alpha, \beta, \gamma, \eta$  を学習係数とすると以下の式で求められる．

$$\Delta w_i(t+1) = \alpha \cdot (y^T - y^0) \cdot y_i(t) \quad (6)$$

$$\Delta c_j^i(t+1) = \beta \cdot w_i(t) \cdot (y^T - y^0) \cdot \frac{h_j^i(t)}{\sum_{j=1}^{m_i} h_j^i(t)} \quad (7)$$

$$\Delta a_j^i(t+1) = \gamma \cdot w_i(t) \cdot (y^T - y^0) \cdot (c_j^i(t) - y_i(t)) \cdot \frac{h_j^i(t)}{\sum_{j=1}^{m_i} h_j^i(t)} \cdot \frac{2 \cdot (x_i - a_j^i(t))}{b_j^i(t)} \quad (8)$$

$$\Delta b_j^i(t+1) = \eta \cdot w_i(t) \cdot (y^T - y^0) \cdot (c_j^i(t) - y_i(t)) \cdot \frac{h_j^i(t)}{\sum_{j=1}^{m_i} h_j^i(t)} \cdot \left( \frac{x_i - a_j^i(t)}{b_j^i(t)} \right)^2 \quad (9)$$

#### 4.2 位置予測手法

SIRMs ファジィモデルは 1 出力であるのに対し、プレイヤーの位置は  $x$  座標と  $y$  座標の 2 次元で表現される。そのため、SIRMs ファジィモデルを 2 つ用意し、それぞれに  $x$  座標と  $y$  座標の値を学習させる。入力項目は以下の 16 項目とした。

- ボールの位置  $(x_b, y_b)$  と速度  $(v_{bx}, v_{by})$
- 味方プレイヤーの位置  $(x_p, y_p)$  と速度  $(v_{px}, v_{py})$
- 敵プレイヤーの位置  $(x_o, y_o)$  と速度  $(v_{ox}, v_{oy})$
- 敵プレイヤーに 2 番目に近い味方プレイヤーの位置  $(x_s, y_s)$  と速度  $(v_{sx}, v_{sy})$

各入力データは正規化された後に SIRMs ファジィモデルに入力される。位置の  $x$  座標は次のように正規化される。ここで  $PitchWidth$  は 115 である。これはフィールドの縦の幅 105m に、10m の余裕をもたせたものである。

$$x_{input} = \max\left\{\frac{x}{PitchWidth} + 0.5, 1.0\right\} \quad (10)$$

$$input = \{b, p, o, s\} \quad (11)$$

位置の  $y$  座標は次のように正規化される。ここで、 $PitchLength$  は 78 である。これはフィールドの横の幅 68m に 10m の余裕をもたせたものである。

$$y_{input} = \max\left\{\frac{y}{PitchLength} + 0.5, 1.0\right\} \quad (12)$$

$$input = \{b, p, o, s\} \quad (13)$$

速度は、 $x$  成分と  $y$  成分ともに次のように正規化される。実験中に速度の  $x$  成分、 $y$  成分が 3 を超えたことはなかったが、余裕を持たせるため速度が 4 を超えることはないと仮定し、各成分について以下の正規化を行う。

$$v_{input} = \max\left\{\frac{v}{4} + 0.5, 1.0\right\} \quad (14)$$

$$input = \{bx, by, px, py, ox, oy, sx, sy\} \quad (15)$$

各入力項目に 5 個のガウス型メンバシップ関数を用意し、初期のガウス型メンバシップ関数を隣同士がグレード 0.50 で交差するように式 (3) の  $b_j^i$  の初期値を設定する。また、後件部実数値の初期値をすべて 0.00 とする。各入力項目の重視度の初期値を 0.25 に設定する。

## 5 数値実験

数値実験では、試合ログから敵位置の学習を行い、教師信号による予測精度の検証を行う。さらに、提案手法を組み込んだチームで試合を行い、1 対 1 の守備における性能調査を行う。

### 5.1 敵位置予測の検証

まず、SIRMs ファジィモデルによる敵位置予測の検証を行う。HELIOS2015 と WrightEagle15 の 100 試合を学習用データ収集の対象とした。HELIOS2015 は本研究室と福岡大学が共同開発しているチームである。WrightEagle15 は世界大会である RoboCup2015 の優勝チームである。試合後、生成されたログファイルから学習に必要なデータを抽出し、SIRMs ファジィモデルを用い学習を行った。以下の条件を満たした状況を 1 対 1 の守備と定義し、学習対象とした。

- 味方プレイヤーの半径 5 メートル以内に敵プレイヤーが一人だけいる
- 直前にボールをキックしたプレイヤーが対象の敵プレイヤーである
- ボールが対象の敵プレイヤーの半径 5 メートル以内に存在している
- 対象の味方プレイヤーが対象の敵プレイヤーに 1 番近いプレイヤーである

教師信号は 1, 2, 3, 5 サイクル後の敵の位置とした。学習後、HELIOS2015 と WrightEagle15 の学習時に用いたものとは異なる 100 試合をテストデータとして用い、予測精度について検証を行った。Table 1 に予測サイクル数での平均誤差を示す。

表 1: Average error for each prediction cycle

Prediction cycle	1	2	3	5
Average error (m)	0.6165	1.3105	2.4487	4.6011

Table 1 より、予測サイクル数が大きくなるに従い、平均誤差も大きくなることわかる。予測サイクル数が大きくなるほど敵の到達可能な範囲が大きくなるため、予測が難しくなり、誤差が大きくなったと考えられる。また、1, 2 サイクル後の平均予測誤差では約 0.7 メートルほどの差であるのに対し、2, 3 サイクル後の平均予測誤差では約 1.1 メートル以上の差があることがわかる。予測サイクル数が小さければ、敵の速度などから予測が容易なのに対し、予測サイクル数が大きいと敵がどのように行動するかによって状況が変化することが起きやすいため、予測が難しくなり、誤差が大きく増加したと考えられる。

表 2: Defense performance

	Conventional	Prediction cycle			
		1	2	3	5
pass	4899	4358	4201	4482	4257
advance with pass	298	217	190	289	219
advance with dribble	219	242	193	298	285
opponent's setplay	61	53	33	26	48
our setplay	5	7	5	5	2
our kick	24	23	32	11	25

## 5.2 提案手法を用いた守備の性能調査

次に、提案手法を HELIOS2015 に組み込んだ場合の守備の性能調査を行う。提案手法を組み込む前の HELIOS2015 では手作業による調整で 1 対 1 の守備を行っており、敵位置予測は敵プレイヤーの体の向きや速度から計算し、繰り返し調整することで行っている。本実験ではこの敵位置予測を SIRM s ファジィモデルによって行い、提案手法なしの場合と守備性能を比較する。また、提案手法はポジションがサイドバックのプレイヤーにのみ実装した。実装するプレイヤーにサイドバックを選んだ理由は、他のポジションよりも、サイドバックが多く 1 対 1 の守備の機会があったことによる。HELIOS2015 におけるサイドバックのポジションを Figure 5 に示す。

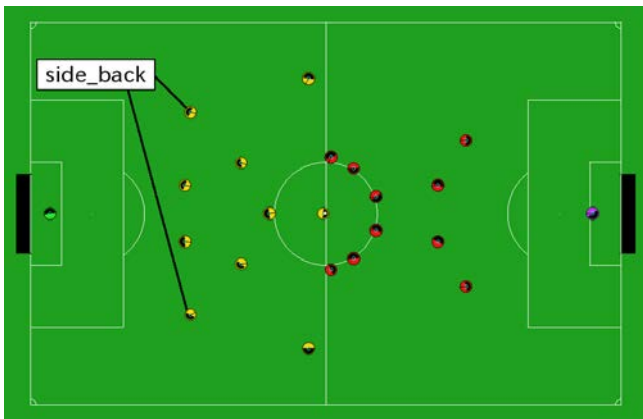


図 5: Side back

守備性能の評価項目を以下の 6 項目とし、1 対 1 の守備においてどのような結果になったかを評価する。

### パス (pass)

敵がパスを行った回数

### パスによる前進 (advance with pass)

パスによってボールが  $x$  軸方向に 5m 以上自陣側に前進した回数

### ドリブルによる前進 (advance with dribble)

ドリブルによってボールが  $x$  軸方向に 10m 以上自陣側に前進した回数

### 敵のセットプレイ (opponent's setplay)

敵によるセットプレイになった回数

### 味方のセットプレイ (our setplay)

味方によるセットプレイになった回数

### 味方のキック成功 (our kick)

味方プレイヤーがボールをキックした回数

「パスによる前進」「ドリブルによる前進」「敵のセットプレイ」は敵にとって良い結果となるため、増加すると 1 対 1 の守備が悪くなったと考えられる。「味方のセットプレイ」「味方のキック成功」は味方にとって良い結果となるため、増加すると 1 対 1 の守備が良くなったと考えられる。

提案手法を用いていないチームと、1, 2, 3, 5 サイクル後の敵位置を予測する SIRM s を組み込んだ 4 チームの計 5 チームを用意し、WrightEagle15 と 100 回試合を行った。Table 2 に結果を示す。1 サイクル後の敵位置予測を用いた場合は、パスによる前進が減少したが、ドリブルによる前進が増加してしまっただとされる。これは 1 サイクル後の敵位置予測は、従来の体の向きや速度から計算することが比較的容易であるためだと考えられる。例えば、敵が急にドリブル方向を変更した場合、提案手法では予測精度が低いことがあったが、従来手法だと敵の体の向きを考慮するため、1 サイクル後であればある程度の予測が可能である。よって、従来手法に比べて予測精度が悪くなったと考えられる。2 サイクル後の敵位置予測を用いた場合は、敵のパスによる前進とセットプレイが減少し、味方がキックをした回数が増加しており、最も良い結果となった。3 サイクル後の敵位置予測を用いた場合は、パスとドリブルによる前進が増加しており、味方のキック成功数も減少し、最も悪い結果となった。これは予測精度が悪くなったためだと考えられる。5 サイクル後も同様に予測精度が悪いため、ドリブルによる前進が増加しているが、パスによる前進は減少している。これは敵にドリブルによって前進されつつも、敵をマークする動きが多かったためと考えられる。以上のように、提案手法は予測するサイクル数によって、1 対 1 の守備性能を向上させることができる。

## 6 おわりに

本論文では、SIRMs ファジィモデルにより 1 対 1 の守備における敵位置予測を行い、予測結果の検証を行った。また、SIRMs ファジィモデルを実際にチームに組み込み、1 対 1 の守備性能について評価を行い、提案手法によって 1 対 1 の守備性能が向上したことを示した。今後の課題としては、より正確な予測を行うことや、他のチームにも対応できる位置予測を行うことなどが挙げられる。

### 参考文献

- [1] Hiroaki 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] Thomas Gabel, Martin Riedmiller and Florian Trost, “A Case Study on Improving Defense Behavior in Soccer Simulation 2D: The NeuroHassle Approach,” *RoboCup2008: Robot Soccer World Cup XII*, pp.61-72, 2008.
- [3] 湯場崎 直義, 易建 強, 廣田 薫, “複数入力ファジィ制御のための単一入力ルール群結合型ファジィ推論モデルの提案,” Vol.18, No.5, pp.669-709, 1997.
- [4] 秋山 英久, *RoboCup サッカーシミュレーション 2D 必勝ガイド*, 秀和システム, 2006.

# パーティクルフィルタを用いた人型ロボットの立脚姿勢時の安定性を考慮した実時間模倣

Real-time Imitation of Humanoid Robot based on Particle Filter with Standing Stabilization Control

近藤耀 山本将平 高橋泰岳

Yo KONDO, Shohei YAMAMOTO, and Yasutake TAKAHASHI

福井大学

University of Fukui

{ykondo,shyamamoto,yasutake}@ir.his.u-fukui.ac.jp

## Abstract

This paper proposes a posture imitation method based on particle-filter-based joint angle estimation with stabilization control for a humanoid robot. An inexpensive camera with depth sensor enables us to observe joint positions of a human demonstrator with reasonable accuracy in a fast and comfortable way. In order to mimic the motion observed with the depth sensor based motion capture system by a humanoid robot, inverse kinematics should be solved to acquire the joint angle of the robot. However, It is tough to solve the inverse kinematics, in fact, to calculate the joint angles of the robot from the position of the joints of the human demonstrator, because of redundancy and difference of body link structure between a human demonstrator and a humanoid robot. The proposed method provides a realistic solution of the estimated sequence of the joint angle for the humanoid robot imitation in real-time.

法 [1, 2] があるが, 人の人間動作の妨げになるなど日常生活においてそのような装置を用いるのは困難である. そこで本研究では深度センサを利用した簡易モーションキャプチャシステムを用い, より自然な人間動作中の各関節位置を取得する. 各関節位置情報から関節角を推定することが出来ればロボットの模倣学習に活用出来る. 先行研究として臼井らは, 関節と関節を結ぶ剛体をリンクと呼び, そのリンク姿勢誤差と関節角の変化が最小となる関節角を求める手法を提案した [3]. また, 高橋と榊原はリンク姿勢から関節角の推定に柔軟な計算が行え, リアルタイムに推定を行うことができるパーティクルフィルタを用いる手法を提案した [4]. しかし, 前者は関節角の算出時に人間の特異姿勢を考慮する必要があり, 探索を全探索で行うためリアルタイムで推定を行えないなどの問題があった. 後者は, ヒトが呈示する姿勢を模倣するためのロボットの関節角をリアルタイムに推定できることを示したが, ロボットの安定性は考慮していない. 特に, 後者の手法では足首関節の推定が行われていないため, 立脚姿勢時の安定性が補償されていない. そこで, 本報告ではパーティクルフィルタを用いて関節角をリアルタイムに推定し, 立脚姿勢時の安定性を確保するための脚の各関節の関節角度制御を行う.

## 1 はじめに

近年, 人間との共生環境下において活動する自律ロボットの研究・開発が進められている. 人間とロボットが共生するためには, ロボットに人間の動作をあらかじめ設計しておくことが考えられるが, 人間の日常環境においてロボットに必要とされる全ての動作を事前に設計することは困難である. そこで, ロボットに人間の動作を呈示し, その動作をロボットに学習をさせる模倣学習の研究が行われている.

教示動作を取得するために従来の手法では複数のカメラやマーカ, 装着型のモーションキャプチャを使用する方

## 2 パーティクルフィルタによる関節角の推定

Microsoft 社の KINECT センサ (以下, KINECT) を用いてモーションキャプチャを行い, マーカやセンサ類を装着せずに人間の骨格を検出し, 15 点の関節位置座標時系列データを記録する (図 1). 記録した教示動作データから関節角を推定するにあたり, ロボット座標系  $\Sigma_r$  を設定し, 観測されたカメラ座標系  $\Sigma_c$  における教示動作の各関節座標をロボット座標系へ変換する (図 2). リンク姿勢とは関節と関節を結ぶ剛体をリンクとし, リンクの単位方向ベクトルをリンク姿勢と呼ぶ. 本実験で使用する NAO は, Aldebaran 社が開発した身長 58cm の人型ロボットで,



図3は、NAOの関節を示したものである。Kinectから得られた人間の関節位置からリンク姿勢を考慮しパーティクルフィルタを用いてロボットの関節の角度を推定する。関節  $l$  と  $m$  を結ぶリンク  $j$  のリンク姿勢を式(1)で定義する。

$$\varphi^j = \frac{p_l - p_m}{\|p_l - p_m\|} \quad (1)$$

$p_l$  と  $p_m$  はそれぞれ関節  $l$  と  $m$  のロボット座標系での位置座標で、 $\|\cdot\|$  はユークリッドノルムとする。本実験に利用するパーティクルフィルタのアルゴリズムを Algorithm 1 に示す。まず、ロボットの関節角のセットをパーティクルとし、複数用意して初期化する。3行目で各パーティクルにおける各関節角の更新を行うが、ここではガウス分布にもとづいてランダムに移動すると仮定している。Algorithm 1の4行目でヒトのリンク姿勢とロボットの関節角にもとづく尤度関数を用いる。尤度関数はロボットが制御する関節角から得られるロボットのリンク姿勢と観察したヒトのリンク姿勢の類似度とし、具体的には次式で定義する。

$$h(h\varphi_t^1, \dots, h\varphi_t^N | \theta_t) = \exp\left(-\frac{(1-s)^2}{\sigma}\right) \quad (2)$$

$$s = \frac{1}{N} \sum_{j=1}^N h\varphi_t^j \cdot r\varphi^j(\theta_t) \quad (3)$$

$h\varphi_t^j$  は KINECT で取得したリンク  $j$  の時刻  $t$  における姿勢、 $r\varphi^j$  はロボットのリンク姿勢を表す。 $s$  は人間のリンク姿勢とロボットのリンク姿勢の内積であり、尤度はその内積が1に近いほど高い値を示す。 $\sigma$  は類似度関数の広がり決定する係数である。6行から9行目で尤度の高いパーティクルが優先的に保存され、尤度が低いパーティクルを排除する。

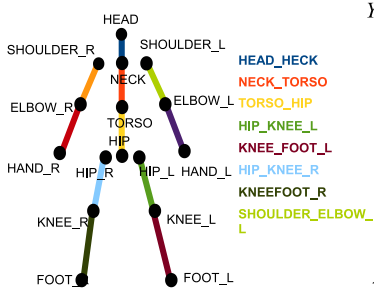


図1: KINECTで取得する関節とリンク

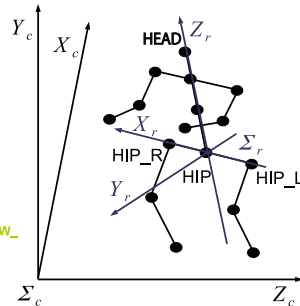


図2: カメラ座標系  $\Sigma_c$  とロボット座標系  $\Sigma_r$

### 3 立脚時の安定性を考慮した腰・膝・足首関節角制御

立脚時の安定性を考慮し、足裏が常に水平に保たれるよう足首を制御する。図4に制御時に用いる関節角を示す。ロボットの足裏を支持面に接地させ続けるには、図4足

#### Algorithm 1 Particle filter

- 1: Initialize particles  $\Theta_t = (\theta_t^{[1]}, \theta_t^{[2]}, \dots, \theta_t^{[M]})$
- 2: **for**  $m = 1$  to  $M$  **do**
- 3: Update particles with the motion model:  
 $\theta_t^{[m]} = \theta_{t-1}^{[m]} + \mathcal{N}(0, \Sigma) \Delta t$
- 4: Calculate the belief of each particle with the measurement model:  
 $w^{[m]} = h(h\varphi_t^1, \dots, h\varphi_t^N | \theta_t)$
- 5: **end for**
- 6: **for**  $m = 1$  to  $M$  **do**
- 7: draw  $m$  from  $\Theta_t$  with probability  $\propto w^{[m]}$
- 8: add  $\theta_t^{[m]}$  to  $\Theta_{t+1}$
- 9: **end for**
- 10: return  $\Theta_{t+1}$

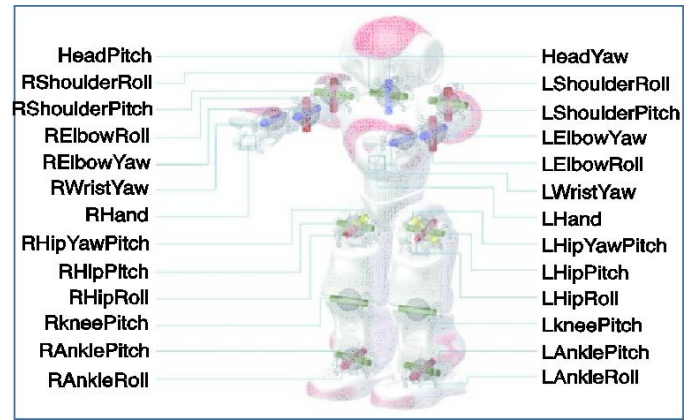


図3: 人型ロボット (NAO) の関節

<http://qiita.com/Ryo87/items/8a17d2c8f81237f2a01f> より引用・改変

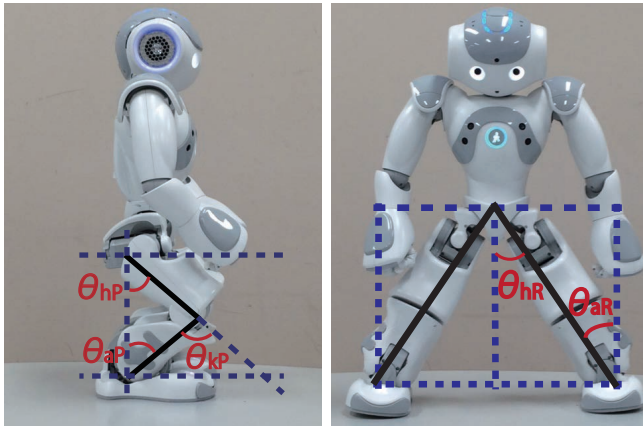
首関節のピッチ角  $\theta_{aP}$ 、ロール角  $\theta_{aR}$ 、腰関節のピッチ角  $\theta_{hP}$ 、ロール角  $\theta_{hR}$  を制御する必要がある。本実験で用いる人型ロボット (NAO) は腰関節から膝関節までの長さ、膝関節から足首関節までの長さがほぼ等しいため、腰関節、膝関節、足首関節を結ぶ三角形は、二等辺三角形と考える。足首関節のロール角は腰関節のロール角と同じ角度を維持することによって、胴体を垂直に保つ。足裏を常に水平に保ちながら、お辞儀のように腰を曲げる動作も考慮する。これらの知見をもとに設計した  $\theta_{aP}$ 、 $\theta_{aR}$  の制御、 $\theta_{HP}$  (パーティクルフィルタで推定した腰関節) を用いた  $\theta_{hP}$ 、 $\theta_{aP}$  の制御式を以下に示す。

$$\theta_{aR} = \theta_{hR} \quad (4)$$

$$\theta_{hP} = \frac{\theta_{kP}}{2} + \theta_{HP} \quad (5)$$

$$\theta_{aP} = \frac{\theta_{kP}}{2} + \frac{\theta_{HP}}{2} \quad (6)$$





(a) 横から見た下半身の関節角 (b) 正面から見た下半身の関節角

図 4: 立脚時安定性を確保するため制御する関節角

## 4 実験

ヒトの呈示する動作を KINECT で観測し，パーティクルフィルタを用いて推定した関節角の値を実機に入力し，模倣する動作を屈伸動作と左右揺動動作に分けて実験を行った．パーティクルの数は 400 個に設定した．実験方法は KINECT で取得した人間動作のリンク姿勢からパーティクルフィルタを用いて関節角を推定し，推定した関節角の値をリアルタイムに NAO に送信し，姿勢を模倣させる．

図 5, 6 に，提案手法を用いて前後・左右方向の人間の姿勢を模倣している様子を示す．足首が床面に対して水平を保つことで，安定して人間の姿勢を模倣できていることがわかる．図 7(a), 7(b) は，それぞれ図 5, 6 に示した，立脚時の安定性を考慮した脚の関節角制御の有無時の重心の時間変化を表したグラフである．どちらもパーティクルフィルタにのみで制御した方は，立脚時の安定性を考慮した関節角制御を用いた方法に比べ重心の変化が激しく，その後転倒している．立脚時の安定性を考慮した関節角制御を行う場合は，軌道が安定しバランスを崩していないことがわかる．

## 5 おわりに

本報告では，人型ロボットに人間の日常動作の模倣を行わせるため，大掛かりな装置を必要とする従来の研究とは異なり，容易にモーションキャプチャを行うためのモーションキャプチャシステムに KINECT を用いたことで自然な人間の日常動作を取得することができた．KINECT から得られた関節位置座標からリンク姿勢を計算しパーティクルフィルタを用いて関節角をリアルタイムに推定しロボットに模倣させた．重心の位置や ZMP を考慮せずとも，簡単な計算でリアルタイムにロボットの足首・腰関節角の制御を行い，立脚時の姿勢を安定化させることができた．今後の課題として，より多くの人間の日常動作を模倣できるようにするため，安定した片足立ちや斜面上で

人間の動作を模倣できるように制御をかけることが挙げられる．

## 参考文献

- [1] Jonas Koenemann, Felix Burget, and Maren Bennewitz. Real-time imitation of human whole-body motions by humanoid. In *International Conference on Robotics & Autonomous*, pp. 2806–2812, 2014.
- [2] Christopher Stanton, Anton Bogdanovych, and Edward Ratanasena. Teleoperation of a humanoid robot using full-body motion capture, example movements, and machine learning. In *Australasian Conference on Robotics and Automation*, 2012.
- [3] 臼井和廉. ヒューマノイドロボットのための対話型強化学習システムの開発. 福井大学 卒業論文, 2013.
- [4] 高橋泰岳, 榊原洗太. 人間動作の観察から動作模倣を行う人型ロボットのためのパーティクルフィルタを用いた関節角度時系列推定. 第 42 回人工知能学会 AI チャレンジ研究会 (SIG-Challenge) 予稿集, pp. 21–23, 2015.05.

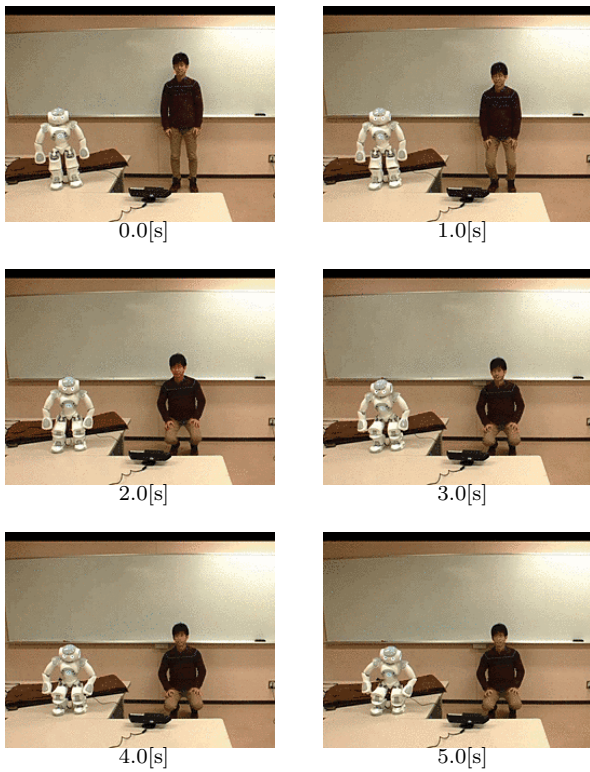
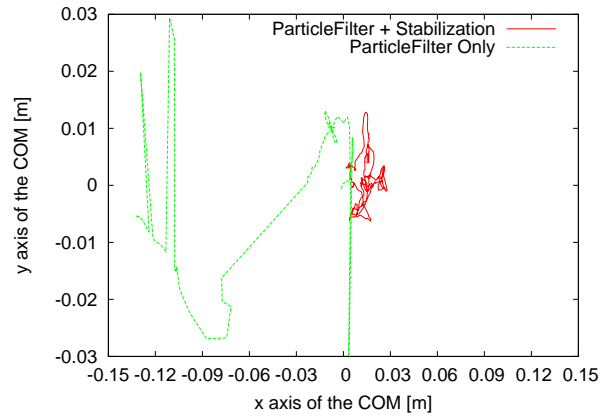


図 5: 屈伸動作の模倣



(a) 屈伸動作時の重心変化

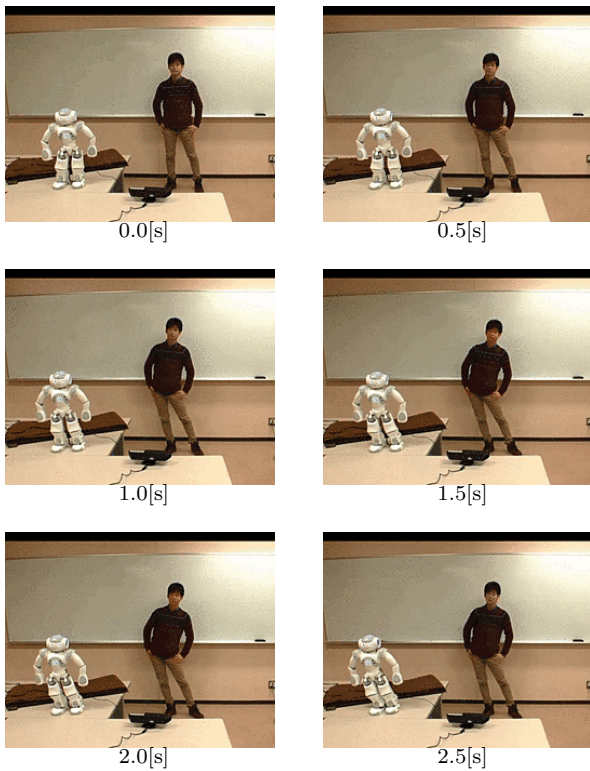
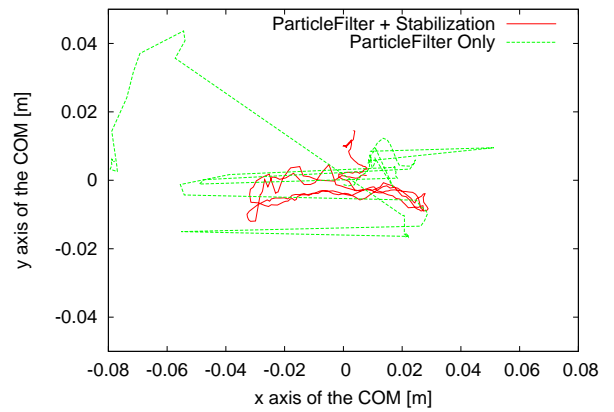


図 6: 左右揺動動作の模倣



(b) 左右揺動動作時の重心変化

図 7: 反射運動の有無における重心の変化

# RoboCup 小型リーグにおける行動系列に基づく相手戦略の学習

Learning an opponent strategy based on a sequence of actions in RoboCup SSL

安達勇介, 伊藤正英, 成瀬正

Yusuke Adachi, Masahide Ito and Tadashi Naruse

愛知県立大学 情報科学部

School of Information Science and Technology, Aichi Prefectural University

is121004@cis.aichi-pu.ac.jp, {masa-ito, naruse}@ist.aichi-pu.ac.jp

## Abstract

RoboCup Soccer 小型リーグでは, 近年サッカー戦略が著しく進歩しており, 相手の行動を予測した上で対策行動を取ることが重要になってきている. このことは, 相手行動の学習という問題の解決を図る必要性を示唆している. 我々は, セットプレーを対象にして, ロボットの動き (配置) の類似性から相手戦略の分類を行う手法を検討してきた [2]. この手法には, セットプレーというトリガ点を与えること, および, (セットプレー後の) プレー時間が長い場合に処理時間が長くなる, という問題点がある. そこで本論文では, これらの制約を取り払うために, ロボットの一連の動作をキックやマークなどの基本行動の系列で表し, これを用いた相手戦略の分類を試みる. 分類結果と正解データを比較した結果, 我がチーム RoboDragons に対しては 4 試合中 3 試合において Rand Index が 0.878 を超え, 相手の 4 チームに対しては 0.802 以上の高い分類性能を得ることができた.

## 1 はじめに

近年, ロボットの動きを指令する戦略プログラム (以下, 戦略) は高度化しており, 相手の戦略を予測して対処することが重要課題となっている. 相手の戦略を学習しようという試みは, セットプレー戦略に関しては Erdogan ら [1] や 安井ら [2] の研究で行われている. しかしこれらはロボットの位置座標を元にした連続データのみを使用しており, 学習後の対策行動を直接求められないことや, ノイズへの弱さ, ロボットの軌道の照合の関係上 1 回のプレー (セットプレーからインプレーに移行し, 再びセットプレーが行われるまでの間) 全体へ範囲拡張することの困難さがあげられる.

本研究では, これらの課題を解決するため, 新たにロボットの行動を単位とし, 行動の系列によって戦略を分類することを試みる. 本手法は, 細かい座標データの動きにとらわれずにより大きな概念から分類を行い, また過去の行動連続データとの比較で, 次に来る行動そのものを予測することで対策行動を立てやすくする利点がある. 以下, 本論文では, ロボットの行動の抽出法について述べる. 次いで, 行動系列の非類似度を定義し, 非類似度計算アルゴリズムおよびクラスタリング手法を示す. 最後に, 実験結果を示し, 提案手法の有効性を示す.

## 2 関連研究

Erdogan ら [1] は, セットプレーの開始からボールが外に出るか相手に取られるまでの間の攻撃ロボットの軌道を取得し, これらを用いてセットプレー戦略の分類を行った. 一方, 安井ら [2] はセットプレー開始までのロボットの配置の連続データを使用してセットプレー戦略の分類を行っている.

本論文で提案する手法は, 新たにロボットの行動を使用してセットプレー及び任意の範囲でのプレーの分類を試みるものである.

ここで, ロボットの行動の検出に関しては浅野ら [3] や 安井ら [4] による先行研究が存在する. 浅野らはキックされたボールの直線運動を検出し, これの目的を判別すること, 及びマーク (パス妨害型マーク, シュート妨害型マーク, ボール保持者へのマーク) を検出することを行っている. 安井らは浅野らのボールの直線運動検出法の改善案を示した. 本論文ではこれらの手法を継承しつつ, 今回の目的に合わせるための修正, 機能追加を行い, 行動抽出を行う.

## 3 ロボットの行動の抽出方法

小型ロボットリーグでは, 相手の行動を分類する手法が浅野ら [3] によって, キック行動の検出が安井ら [4] によって

提唱されている。本研究ではこれらの手法を継承しつつ、現在の試合状況に則した手法へと改良を行った。

### 3.1 マーク行動の抽出

浅野らはマークの行動をパス妨害型マーク、シュート妨害型マーク、ボール保持へのマークの3種類に分類し、それぞれの検出方法を示した。ここでは浅野らの手法の紹介と、変更を加えた部分を示す。

#### 3.1.1 浅野らによるマーク検出方法

以下に浅野ら [3] の手法を示す。

- $R, (x_i, y_i)$  : 味方ロボット  $R$  と時刻  $i$  での位置座標
- $R_j, (x_{ij}, y_{ij})$  : 相手ロボットと時刻  $i$  で位置座標
- $L : ax + by + c = 0$  , パス妨害型マークの場合, ボールと味方ロボット  $R$  を結ぶ線. シュート妨害型マークの場合, ロボット  $R$  と相手ゴール中心を結ぶ線. ボール保持者へのマークの場合,  $R$  をボール保持者として,  $R$  とボールを結ぶ線.
- $T$  : しきい値

として, 次のアルゴリズムを実行する.

Step1  $i = 1$  とする.

Step2 各  $j$  に対して次の式を計算する.  $E1$  は敵ロボットと味方ロボットとの距離である.

$$E1_{ij} = \sqrt{(x_i - x_{ij})^2 + (y_i - y_{ij})^2} \quad (1)$$

相手ロボット  $R_j$  から  $L$  に下ろした垂線が  $L$  に交わるかどうかで,  $R_j$  と  $L$  との距離を表す  $E2$  の値を以下のように決定する.

$$E2_{ij} = \begin{cases} \frac{|ax_{ij} + by_{ij} + c|}{\sqrt{a^2 + b^2}} & (\text{垂線と } L \text{ が交わる}) \\ \frac{|ax_{ij} + by_{ij} + c|}{\sqrt{a^2 + b^2}} \cdot \gamma & (\text{垂線と } L \text{ が交わらない}) \end{cases} \quad (2)$$

2つの評価値を重み  $\alpha, \beta$  を使用して足し合わせたものを  $E_{ij}$  とする.

$$E_{ij} = \alpha \cdot E1_{ij} + \beta \cdot E2_{ij} \quad (3)$$

この  $E_{ij}$  の  $n$  フレーム平均を取る.

$$\bar{E}_j = \frac{1}{n} \sum_{k=i}^{i+n-1} E_{kj} \quad (4)$$

$$CoveringByRobot_j = \begin{cases} 0 & (\bar{E}_j > T) \\ 1 & (\bar{E}_j \leq T) \end{cases} \quad (5)$$

もし  $CoveringByRobot_j = 1$  であった場合,  $i$  から  $i+n-1$  の間  $R$  は  $R_j$  にマークをされたとする.  $i = i+n$  として, 試合ログデータの終わりまで Step2 を繰り返す.

### 3.1.2 パス妨害型マーク

浅野らの手法ではパス相手の存在の有無が考慮されておらず, 単にボールからロボットへのコース上を妨害しているかどうかの評価式であった. ボールを追っているだけの状態のようなケースを除外し, より忠実にパス妨害を判定できるようにしたものが次のアルゴリズムである.

$T_{Nearest}$  はパス相手であり, そのフレームにおいて, マーク対象となっているロボット以外のうち, 直線  $L$  との距離が  $D$  未満<sup>1</sup> である中で, 最小距離の味方ロボットとする.

- $\vec{T}_i = (x_{ti}, y_{ti}) (i = \{0, 1, \dots, 5\})$ : フレーム  $f$  における味方ロボット  $T_i$  の位置ベクトル
- $\vec{O}_j = (x_{oj}, y_{oj}) (j = \{0, 1, \dots, 5\})$ : フレーム  $f$  における敵ロボット  $O_j$  の位置ベクトル
- $\vec{B}$  : ボールの位置ベクトル
- $\vec{V}_1 = \vec{B} - \vec{T}_i$  :  $T_i$  からボールへのベクトル
- $\vec{V}_2 = \vec{O}_j - \vec{T}_i$  :  $T_i$  から  $O_j$  へのベクトル
- $\vec{T}_{Nearest} = (x_n, y_n)$  : フレーム  $f$  で,  $\vec{V}_1$  の延長線上に最も近い  $T_i$  以外の味方の位置ベクトル
- $\vec{V}_3 = \vec{T}_{Nearest} - \vec{T}_i$  :  $T_i$  から  $T_{Nearest}$  へのベクトル
- $\vec{V}_4 = \vec{T}_i - \vec{T}_{Nearest}$  :  $T_{Nearest}$  から  $T_i$  へのベクトル
- $\vec{V}_5 = \vec{O}_j - \vec{T}_{Nearest}$  :  $T_{Nearest}$  から  $O_j$  へのベクトル

Step0 フレーム番号  $f, j = 0, i = 0$  とする.

Step1  $T_i$  と  $O_j$  に対して以下の式を計算する

$$Dot_1 = \frac{\vec{V}_1}{|\vec{V}_1|} \cdot \vec{V}_2 \quad (6)$$

$$Dot_2 = \frac{\vec{V}_4}{|\vec{V}_4|} \cdot \vec{V}_5 \quad (7)$$

$$Val_{fj} = \begin{cases} \left| \frac{\vec{V}_3}{|\vec{V}_3|} \times \vec{V}_2 \right| & (Dot_1 \geq 0 \wedge Dot_2 \geq 0) \\ \gamma_p + \left( \left| \frac{\vec{V}_3}{|\vec{V}_3|} \times \vec{V}_2 \right| \right) & (Dot_1 < 0 \vee Dot_2 < 0) \end{cases} \quad (8)$$

$Val_{fj}$  は味方ロボット  $T_i$  と  $T_{Nearest}$  を結ぶ線分に  $O_j$  の中心から下ろした垂線の長さである. 味方ロボット2台よりも外側に存在する場合, それはパス経路の妨害をしていないことになるので, 重みを加算して除外する.

$$MarkPass_j = \begin{cases} 0 & \left( \frac{1}{n} \sum_{k=f}^{f+n-1} Val_{kj} > T_p \right) \\ 1 & \left( \frac{1}{n} \sum_{k=f}^{f+n-1} Val_{kj} \leq T_p \right) \end{cases} \quad (9)$$

<sup>1</sup>今回は 400(mm) を使用した.

評価平均値が閾値  $T_p$  よりも小さければ、 $T_i$  は  $f$  から  $n$  フレームの間は敵ロボット  $O_j$  によってマーク行動が行われたとし、行動を MarkPass とする<sup>2</sup>。

Step2  $i$  をインクリメントする。もし  $i$  が 5 ならば、 $j$  をインクリメントし、 $i = 0$  として Step1 に戻る。 $j$  が 5 ならば、このフレームでのマーク計算は終了である。

### 3.1.3 シュート妨害型マーク

シュート妨害型マークは、浅野らの手法ではボール保持者との距離が評価式に含まれていたが、シュート妨害は離れた位置で行われることが多いためこれを削除した。具体的には式 3 の  $\alpha = 0, \beta = 1$  とし、また修正したパス妨害型マークと同様に垂線が  $L$  に交わらない場合は完全に除外するように設計した。シュート妨害型マークの状態のとき、行動を MarkShoot とする<sup>3</sup>。

### 3.1.4 ボール保持者へのマーク

ボール保持者へのマーク行動検出は、ボールの後ろやロボットの後ろから接近している場合もボール保持者へのマークと考えられるため、式 2 を  $\gamma = 1.0$  とした。つまり、位置関係による除外を行わない。ボール保持者へのマークの状態であるとき、行動を MarkBall とする<sup>4</sup>。

## 3.2 パス待ち行動の抽出

今回新たにパス待ちの検出を設計した。基本的に  $n$  回の平均を取るアイデアは浅野によるものを採用し、パス待ちという行動の評価値を新たに設計したことになる。

Step0 最もボールに近い敵ロボットを探索し、 $O_{ball}$  とする。

Step1 フレーム  $k$  で、 $O_{ball}$  以外の敵ロボット  $O_j$  について、もし  $O_{ball}$  であるか、相手の陣地奥に存在する場合、空き角度  $\theta_{kj}$  に負の定数を入れ、パス待ちの判定から除外する。

そうでない場合、 $O_j$  から見た自ゴールに対しての空き角度  $\theta_{kj}$  - 障害物がおらず、シュートが狙える空き角度 - を計算する。

Step2 以下のように  $n$  フレーム間での平均値を取り<sup>5</sup>、それがしきい値  $T_w$  よりも大きい場合、そのロボットはパス待ちをしているとし、行動を WaitPass とする<sup>6</sup>。

$$WaitPass_j = \begin{cases} 1 & \left( \frac{1}{n} \sum_{k=f}^{f+n-1} \theta_{kj} \geq T_w \right) \\ 0 & \left( \frac{1}{n} \sum_{k=f}^{f+n-1} \theta_{kj} < T_w \right) \end{cases} \quad (10)$$

<sup>2</sup>実験では  $T_p = 400(\text{mm})$  を使用した。

<sup>3</sup>実験ではしきい値  $T = 400(\text{mm})$  を使用した。

<sup>4</sup>重み  $\alpha = \beta = 1/2$ 、しきい値  $T = 400(\text{mm})$  を使用した。

<sup>5</sup>平滑化フレーム数はマークとパス待ち検出で  $n = 3$  を使用した。

<sup>6</sup>今回の実験では  $T_w = 0.139626(\text{rad})(8 \text{度})$  を使用した。

## 3.3 キック行動抽出とキックの種類判別

ロボカップサッカーにおいて、マーク行動とともに重要なものがキック行動である。キックそのものの検出については安井ら [4] が行っているため、本研究ではこれによって検出されたキックの種類を判別する仕組みを新たに設計した。浅野ら [3] が行ったキック結果判別と目的とするところは類似しているが、ここでは目的にのみ注目し、シュート、パス、クリアのみをキックの種類とする。また、キックの延長線とオブジェクトとの接触だけではなく、近年の試合に多く見られるゴール近辺でのシュート、パスの分類ができるよう設計を行った。アルゴリズムを次に示す。

Step0 キック行動の種類 Kick を、

Kick = {KickBall, KickShoot, KickPass, KickClear} とする。KickBall はシュート、パス、クリアのどれにも該当しないキックであり、KickShoot はシュート、KickPass はパス、KickClear はクリアを目的としたキックである。

Step1 キック開始地点とボールの直線運動終了地点を結ぶベクトル  $\vec{P}_b$  を求める。

Step2  $\vec{P}_b$  の終点の座標がフィールドのタッチライン(サイドライン)よりも外であった場合、クリア (KickClear) とする。

Step3  $\vec{P}_b$  の終点の座標が自チームのゴールラインよりも外であった場合、シュート (KickShoot) とする。

Step4 キック開始点から他の相手ロボットへのベクトル  $\vec{P}_{oi}$  を求め、

$$\left| \vec{P}_{oi} \times \vec{P}_b \right| < D_1 \quad (11)$$

を満たしている場合、パス (KickPass) とする<sup>7</sup>。

Step5 キック開始点から、自チームのゴール右端と左端にマージンを加えた地点へのベクトル  $\vec{P}_{gr}, \vec{P}_{gl}$  を用意する。キックの始点から終点へのベクトルの角度が、 $\vec{P}_{gr}, \vec{P}_{gl}$  の角度の間に収まる場合、シュートの方向に蹴られたことがわかる。このとき、 $\vec{P}_b$  の終点の座標と自分のゴール中心との距離  $d$  がしきい値  $D_G$  よりも小さい場合、ゴールを狙った途中で何かにあたったと考えられるので、これはシュート (KickShoot) とする<sup>8</sup>。

Step6 どの目的にも該当しない場合、KickBall とする。

## 4 行動抽出の全体の流れ

これまでの行動抽出アルゴリズムを次のように動作させる。

<sup>7</sup>本研究では  $D_1 = 800(\text{mm})$  を使用した。

<sup>8</sup> $D_G = 2000(\text{mm})$  を実験では使用している。

Step1 現在のロボットやボールの位置が保存されたフレームデータを取得する。

Step2 キック検出，パス待ち検出，マーク検出を行う。

Step3 各ロボットについて，検出された行動をキューに保存していく．検出された行動が以前に現れた行動の続きである場合，その行動を保存した場所のデータを更新する．Step1 に戻る．

行動は，現れた順に保存されていく．優先度は高いものからキック，パス待ち，マークの順としており，上位の行動が継続している間は下位の行動は検出されず，下位の行動継続中に上位の行動が検出された場合はその下位の行動は終了し，上位の行動に切り替わる．

#### 4.1 得られた行動系列に対する前処理

行動系列が得られた段階では，ノイズと考えられる行動データも含まれている．これらを次のように取り除く．

- 数フレーム<sup>9</sup>で切り替わった行動はノイズと判断するべきで，このような行動に対しては除去を行う．ただし，キックの場合はすぐにフィールド外に出たような場合も重要な特徴となり得るため，前後にキックの行動がない場合はこの処理は行わない．
- 同じ行動が途切れ，再開された場合，これは同一の行動として扱うべきであるので，統合する処理を行う．ただし，キックの場合，連続してキックするということは重要な意味を持つので，一定フレームの差<sup>10</sup>がなく同一と考えられる場合のみ統合を行う．
- 特に短い範囲で行動抽出をした場合，行動が1つも抽出できない場合があるが，その場合に限り行動 No-Data を追加する．

## 5 行動の非類似度

前章までの行動抽出法により，シュートマーク，パスマーク，ボール保持者へのマーク，パス待ち，キック，パス，シュート，クリアの8種類 (NoData を含めると9種類) の行動が抽出できることを示した．この行動の種類を  $action$  とし，特にキック4種類を Kick グループと定義する．試合中にはこれらの開始点，終了点，継続フレーム数を加えて，次の形で行動データが保存されていく．このとき， $P$  回目のプレーにおいて，6 台のロボットのうち  $n$  番目のロボットに対して  $t$  回行動抽出した行動系列  $A_P[n]$  は，行動開始点  $\vec{p}_s$ ，行動終了点  $\vec{p}_e$  と継続フレーム数  $frame$  を合わせて，

次のような形となる．

$$A_P[n] = \left[ \begin{array}{c} action_{n1} \\ \vec{p}_{sn1} \\ \vec{p}_{en1} \\ frame_{n1} \end{array} \right], \dots, \left[ \begin{array}{c} action_{nt} \\ \vec{p}_{snt} \\ \vec{p}_{ent} \\ frame_{nt} \end{array} \right] \quad (12)$$

この系列のうち， $k$  番目の1要素を取り出したものを  $A_P[n][k]$  と定義する．

$$A_P[n][k] = \begin{pmatrix} action_{nk} \\ \vec{p}_{snk} \\ \vec{p}_{enk} \\ frame_{nk} \end{pmatrix} \quad (13)$$

2つの行動  $A_{P_1}[n_1][t_1]$ ,  $A_{P_2}[n_2][t_2]$  の非類似度  $d_0$  を，重み  $\alpha, \beta, \gamma$  を用いて，次のように定義する．それぞれの項は行動の種類的一致，位置的一致，系列の長さの一致を表している．行動の種類が一致している場合，行動継続期間の全体に占める割合で一致具合を表現する．Kick は重要な行動であるため，Kick と他の行動のマッチングに対しては距離が離れるように設計した．

$$d_0(A_{P_1}[n_1][t_1], A_{P_2}[n_2][t_2]) = \begin{cases} \alpha \cdot frame\_diff + \beta \cdot p\_distance + \gamma \cdot diff\_size\_cost \\ (action_{n_1 t_1} = action_{n_2 t_2}) \\ \alpha \cdot 2.0 + \beta \cdot p\_distance + \gamma \cdot diff\_size\_cost \\ (action_{n_1 t_1} \in Kick \wedge action_{n_2 t_2} \notin Kick) \\ \alpha \cdot 1.0 + \beta \cdot p\_distance + \gamma \cdot diff\_size\_cost \\ (action_{n_1 t_1} \notin Kick \wedge action_{n_2 t_2} \in Kick) \\ \alpha \cdot 1.0 + \beta \cdot p\_distance + \gamma \cdot diff\_size\_cost \\ (otherwise) \end{cases} \quad (14)$$

$frame\_diff$  は，注目する行動が行われたプレーのフレーム数  $frame\_play$  に対して，注目する行動がどれだけの割合を占めるかの非類似性を表す評価値である．0 から1までの値を取る．

$$frame\_diff = \left| \frac{frame_{n_1 t_1}}{frame\_play_{n_1 t_1}} - \frac{frame_{n_2 t_2}}{frame\_play_{n_2 t_2}} \right| \quad (15)$$

$p\_distance$  は，開始位置と終了位置の距離の0から2の評価値を表し，次の式で定義する． $FieldLength$  はフィールド長手方向の長さである． $x$  軸 (自ゴール中心から相手ゴール中心へ正の軸) に対して座標の対称性を考慮するため， $y$  座標 ( $y$  軸は  $x$  軸の90度左回転) を反転させたものと，そうでないものとで値が小さい方を採用する．

$$p\_distance = \min \left\{ \frac{|(\vec{p}_{sn_1 t_1} - \vec{p}_{sn_2 t_2})|}{FieldLength}, 1.0 \right\} + \min \left\{ \frac{|(\vec{p}_{en_1 t_1} - \vec{p}_{en_2 t_2})|}{FieldLength}, 1.0 \right\} \quad (16)$$

<sup>9</sup>今回は6フレーム (0.1s) とした．

<sup>10</sup>今回は60フレーム (1.0s) と設定した．



diff\_size\_cost は、対象とする行動が含まれる行動系列の大きさの差を追加コストとして与える評価値である。最小は 0 であり、1:4 の差がある場合に 1 を取り、1:7 以上の差がある場合に 2 を取る。長い方の系列の大きさを long\_size、短い方を small\_size とすると次のようになる。

$$\text{diff\_size\_cost} = \min \left\{ \frac{1}{3} \left( \frac{\text{long\_size}}{\text{small\_size}} - 1.0 \right), 2.0 \right\} \quad (17)$$

ここで、あるロボット  $n_1$  とあるロボット  $n_2$  の行動系列全体  $A_{P_1}[n_1], A_{P_2}[n_2]$  同士の非類似度  $d_1(A_{P_1}[n_1], A_{P_2}[n_2])$  を定義する。行動系列の長さは同じとは限らないため、短い方を基準とし、長い方の中にどれだけ中身が一致するかを調べ非類似度とする。系列と系列の非類似度  $d_1$  は次のアルゴリズムにより決定する。

**Step1** 系列の長さを比較し、短い方を small、長い方を long とする。それぞれの長さは small\_size、long\_size と表現する。long の中に含まれる行動のうち Kick の数をカウントし、これを *kick\_num* とする。small と long の探索用カウンタを  $i, j$  とし、初期値は 1、long の探索開始点を  $start\_j = 1$ 、long の探索終了点  $limit\_j = 1$  とする。  $d_1 = 0$  として Step2 を開始する。

**Step2** まず、small で選択中の  $i$  番目の行動に対して、long 中の探索範囲を決定する。これは長さの比を元に次のように決定する。なお、小数点以下は切り捨てを行う。

$$\begin{aligned} ls &= \text{long\_size}/\text{small\_size} \\ limit\_j1 &= i + ls \\ limit\_j2 &= \min(start\_j + ls, \text{long\_size}) \\ limit\_j &= \max(limit\_j1, limit\_j2) \end{aligned} \quad (18)$$

small の  $i$  番目の行動と、long の  $start\_j$  から  $limit\_j$  までの行動とで、行動の種類 *action* が一致するまで探索を行う。

**Step3** 一致する行動が存在した場合、

$$d_1 = d_1 + d_0(A_{P_1}[n_1][i], A_{P_2}[n_2][j])$$

を計算し、 $start\_j = j + 1$  とする。一致する行動が存在しない場合、

$$d_1 = d_1 + d_0(A_{P_1}[n_1][i], A_{P_2}[n_2][i])$$

の計算を行う。  $i < \text{small\_size}$  の場合は  $i = i + 1$  として Step2 へ、そうでない場合 Step4 に進む。

**Step4** 最後に、Kick の数 *kick\_num* のうち使用されずに残った数 *kick\_unuse* を追加コストとして加算する。

$$d_1 = d_1 + \text{kick\_unuse}$$

最後に、あるプレーで得られたロボット 6 台分の行動系列と、別のプレーで得られたロボット 6 台分の行動系列の対応関係を加味してプレー同士の非類似度  $d_2$  とする。 $P_\sigma$  は置換行列である。

$$d_2(A_{P_1}, A_{P_2}) = \min_{\sigma \in S_6} \{ \text{Tr}(FP_\sigma) \} \quad (19)$$

$$F = [f_{ij}] \quad (20)$$

$$f_{ij} = \{d_1(A_{P_1}[i], A_{P_2}[j])\} \quad (21)$$

ここで定義した類似度を用いて群平均法 [5] でクラスタリングを行う。

## 6 クラスター数の決定

本研究では、安井ら [2] による研究で用いられた手法によりクラスター数を決定する。この手法は、あらかじめクラスター数の範囲を指定する必要がある Davies Bouldin index[6] などとは異なり、範囲指定する必要がない。相手戦略といったクラスター数が不明なものを分類する場合には有効な手法である。安井らが提案した手法は次の通りである。

$X_k$  を  $k$  回目のプレー、 $A_{P_k}$  をそのときの行動系列として、

$$W(K) = \sum_{i=1}^K \sum_{X_k \in C_i} \sum_{X_l \in C_i} d_2(A_{P_k}, A_{P_l}) \quad (22)$$

を定義する。この式は、 $K$  個のクラスターに分類されたときの、それぞれのクラスター内の要素（プレー）同士の距離の総和を表している。つまり、クラスター内の他の要素と比べて非類似度の高い（距離の遠い）要素がクラスター内に存在する場合、この評価値は要素間の距離の和を取るため、値が大きくなる仕組みとなっている。この  $W(K)$  を用いて、

$$W'(K) = W(K)/W(1) \quad (23)$$

としたとき、

$$\arg \max_{1 \leq K \leq N} (W'(K) \leq h) \quad (24)$$

となるようなクラスター数  $K$  を採用する。

## 7 試合ログデータを用いた実験結果と考察

RoboCup2015 で我々のチーム RoboDragons の行った公式戦 4 試合について、RoboDragons の行動がどのように分類されるか実験を行った。この実験では非類似度の重みを  $\alpha = \beta = \gamma = 1/3$  とした。クラスター数決定のしきい値  $h$  については、0.06 を使用する。この値は先行研究 [2] で最も良い結果が出たときの値である。

今回は人の目で行動の分類の行いやすいセットプレーを対象にする。ここでのセットプレーは、ボールの再配置から、ボールが相手に取られるかフィールド外に出るまでの間とする。それぞれの試合についてクラスタリングを



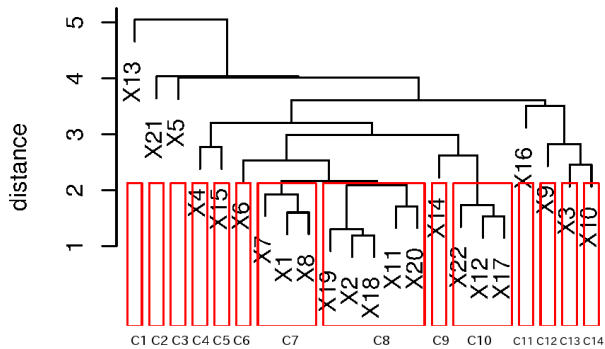


Figure 1: Dendrogram for match No.1  
(group average clustering; the same clustering is used in the following figures)

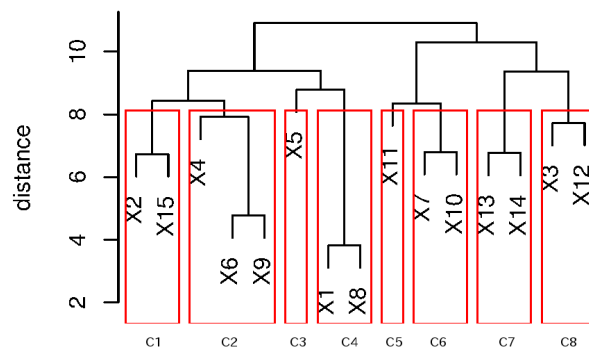


Figure 3: Dendrogram for match No.3

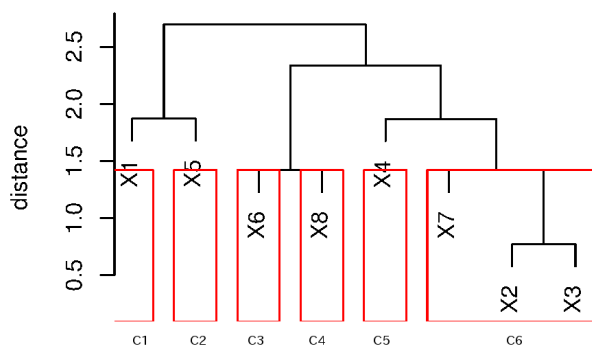


Figure 2: Dendrogram for match No.2

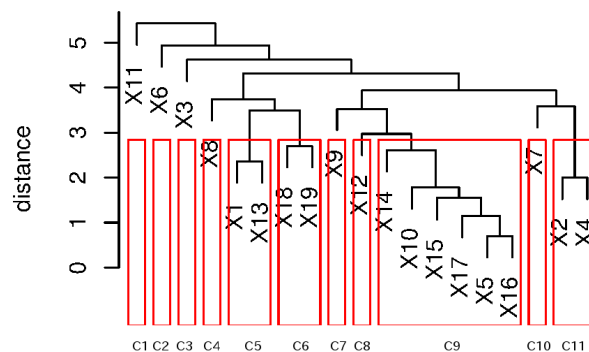


Figure 4: Dendrogram for match No.4

行った結果が図 1 から図 4 である。これら分類結果と、筆者が行動に着目して行った分類結果とを Rand Index[7] で評価を行った。その結果が表 1 である。第二試合の結果を除くと正解データとした筆者の分類結果と高い水準で類似していることがわかる。第二試合に関して調査したところ、相手が不調により動かなかった場合があり、マーク行動の判定が消滅したことによる分類悪化が起きていることがわかった。0.878 や 0.879 という水準で止まった試合については、正解データとして用意したクラスターが提案手法では 2 つに分離される現象が起きており、これが大きな原因と考えられる。実用上では同じ行動が 2 つのクラスターに分かれていても問題は少ないため、十分役立つのではないかと考えている。

Table 1: Rand Index (RoboDragons)  
(computer clustering vs. human clustering)

	No.1	No.2	No.3	No.4
Rand Index	0.879	0.750	0.990	0.878

## 8 他のチームでの結果

RoboDragons の場合を前章で見たが、公式戦 4 試合の相手チームに対して適用した場合どうなるのかを本章で示す。結果が表 2 であり、各々のクラスタリング結果が図 5 から図 8 である。4 試合とも Rand Index は 0.80 以上の値となった。今回の結果も自チームのときと同様に、正解として用意したクラスターが 2 つ以上に分裂する現象が起きていることがわかった。この問題を解決するには、類似したプレーを分離させずにまとめる手法を考案することが挙げられ、現状の郡平均法を使用したクラスタリング部を再考する必要があると思われる。例えば、最短距離法で一定距離までを結合する、などが考えられるが、今後検討を進める必要がある。

## 9 実行時間

最も多い 35 回のセットプレーが行われた 4 戦目の対戦相手の場合について、クラスタリングの部分における実行時

Table 2: Rand Index (opponents)

	No.1	No.2	No.3	No.4
Rand Index	0.853	0.861	0.850	0.802

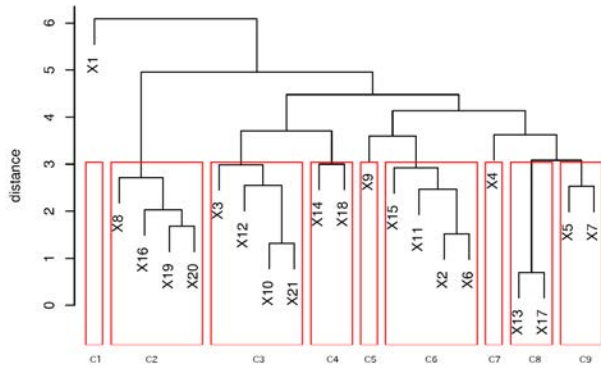


Figure 5: Dendrogram for match No.1(opponent)

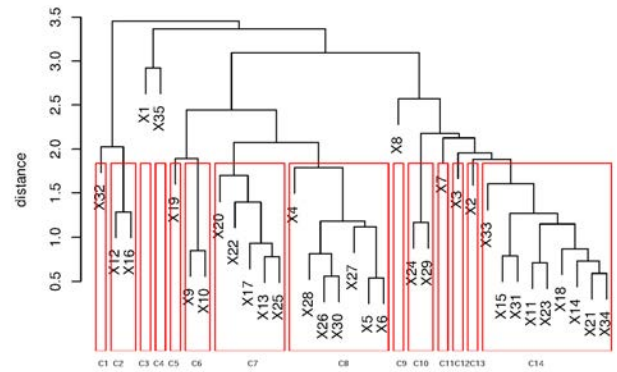


Figure 8: Dendrogram for match No.4(opponent)

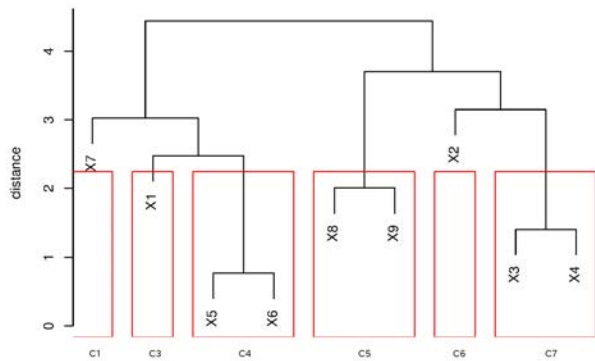


Figure 6: Dendrogram for match No.2(opponent)

間の計測を行った。この部分では行動系列の前処理を行い、距離行列の作成と、郡平均法で全て単独クラスターの状態から1つのクラスターになるまで結合を繰り返す処理を行う。(1回目では単独クラスター1つしかないのでそのまま終わり、35回目では35回目と、今までの34回分のプレー全てと距離の計算を行い、その後クラスタリングを行う。)この結果が表3であり、この結果から十分リアルタイム性を持って計算が行えることがわかる。

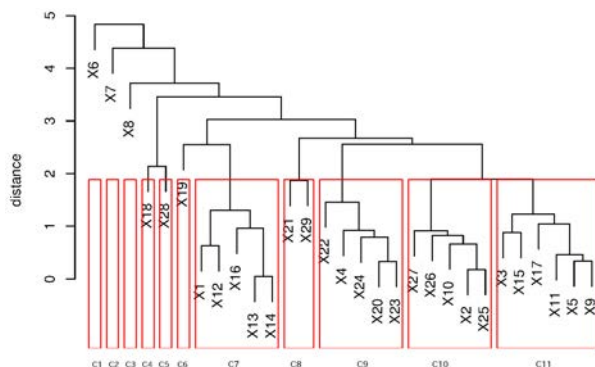


Figure 7: Dendrogram for match No.3(opponent)

Table 3: clustering time(AMD A10 7800)

	average time(ms)	max time(ms)
No.4(35 setplays)	0.67	1.82

## 10 おわりに

本研究では新たに行動を単位として相手の戦略を学習する仕組みを設計した。正解データとの間で Rand Index を計算すると RoboDragons に対して1試合の例外を除いて0.878以上、相手チームに対しては0.802以上の評価値が得られ、十分正解に近い結果が得られていると言える。実行時間に関しても実時間で計算できることを示した。今後は高精度化及び、より広いプレー範囲での検証や、過去の行動系列データとの照合による対策行動の生成、自チームの戦略へのリアルタイム反映といった課題に取り組んでいきたい。

## 参考文献

- [1] Can Erdogan, et al. "Action Selection via Learning Behavior Patterns in Multi-Robot Domains", IJCAI, 2011
- [2] 安井興太郎, 他, RoboCup 小型リーグにおけるリアルタイム学習のための相手行動の分類, 電子情報通信学会 論文誌 VOL. J97-D NO.8, 2014
- [3] Koshi Asano, et al. "Detection of basic behaviors in logged data in RoboCup Small Size League", RoboCup Symposium, 2008
- [4] 安井興太郎, 他, RoboCup 小型ロボットリーグにおけるキック検出手法の検討, 人工知能学会 AI チャレンジ研究会 SIG-Challenge, 2012
- [5] B.S.Everitt et al. "Cluster Analysis 5th Edition", Wiley, 2011
- [6] David L. Davies and Donald W. Bouldin, "A Cluster Separation Measure", IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI- 1(2), pp.224-227, 1979
- [7] W. M. Rand, "Objective criteria for the evaluation of clustering methods", Journal of the American Statistical Association (American Statistical Association) 66 (336): 846-850, 1971

# ロボカップサッカー小型リーグロボットに対するモデル予測制御の 設計パラメータと制御性能との関係

Relation between Control Performance and Tuning Parameters of Model Predictive Control for  
Robots in RoboCup Soccer Small Size League

日下部 宏幸・伊藤 正英・成瀬 正

Hiroyuki KUSAKABE, Masahide ITO, and Tadashi NARUSE

愛知県立大学 情報科学部

School of Information Science and Technology, Aichi Prefectural University

is121032@cis.aichi-pu.ac.jp, {masa-ito, naruse}@ist.aichi-pu.ac.jp

## Abstract

ロボカップサッカー小型リーグは現在、より動的なパワークへと発展する途上にあり、個々のロボットには高精度な運動制御が要求されている。予測を取り入れた高精度な制御手法の一つとしてモデル予測制御があるものの、一般に設計パラメータの調整指針が明らかでない。本稿では、小型リーグロボットの運動制御に対し後退ホライズン方策に基づくモデル予測制御器を設計し、数値実験に基づく検討によって、設計パラメータと制御性能との関係を明らかにした。

## 1 背景と目的

ロボカップサッカー小型リーグでは、近年の活発な研究・開発の結果、ロボットは高速で正確な動きが可能となり、戦略もますます高度化している。特に、複数台ロボットによる連携プレーに着目すると、各ロボットが受け身でボールを回すというこれまでの主流から、動きながらのボール回しに現在シフトしつつある。この発展にはより高精度な運動制御が要求される。

モデルに基づく予測を取り入れた制御手法として、モデル予測制御 (Model Predictive Control; MPC) [1] が注目されている。ロボカップサッカーでは小型リーグと中型リーグにおいてモデル予測制御の適用例がある [2, 3]。それらの適用例では、従来の制御手法と比べて制御性能が向上したという結果は示されているものの、設計パラメータと制御性能との関係性が明らかになっていない。そこで本研究では小型リーグで用いられている車輪型全方向移動ロボットを対象として、設計した MPC 制御器のパラメータと制御性能との関係を明らかにすることを目的とする。

## 2 小型リーグシステムの概要

小型リーグでは一般的に、オムニホイール 4 輪から構成される全方向移動ロボットが用いられる。ロボットやボールの幾何情報は、各フィールドの全体を俯瞰するように取り付けられた 4 つのカメラによって観測される。一方、ロボットの動きはホスト PC から無線通信によって指令される。

カメラから得られた画像データは SSL-Vision サーバにおいて適切に画像処理され、各ロボットの位置と姿勢、ボール位置などが得られる。それらの情報は各チームのホスト PC へ配信され、試合状況の分析や戦略の決定が行われたのち、ロボット個々の目標軌道が計画される。各ロボットには目標軌道に基づいた速度指令値が無線通信によって送信され、ロボット内部では指令値に対するフィードバック制御が行われる。なお、ホスト PC で速度指令値が送信されてから実際のロボットがそれに応じた動作が開始するまでには、一定のむだ時間が生じる。

## 3 車輪型全方向移動ロボットの運動モデル

車輪型全方向移動ロボットの速度に基づく運動モデルは連続時間表現として次式で与えられる [2]。

$$\begin{cases} \dot{x} = \alpha_x v_x(t - t_0) \\ \dot{y} = \alpha_y v_y(t - t_0) \\ \dot{\theta} = \alpha_\omega \omega(t - t_0) \end{cases} \quad (1)$$

ただし、 $(x, y, \theta)$  はロボットの位置・姿勢、 $(v_x, v_y, \omega)$  は各速度を表している。 $t_0$  は通信によるむだ時間を表しており、入力の影響が実際にロボットの出力に表れるのは時刻  $t_0$  後ということの意味する。速度のスケールパラメータ  $(\alpha_x, \alpha_y, \alpha_\omega)$  はロボットを実際に走行させ、入力として指令した速度と発生した速度との比較から同定することができる。状態ベクトルを  $\mathbf{z} = (x, y, \theta)^T$ 、入力ベクトルを  $\mathbf{v} = (v_x, v_y, \omega)^T$  とすると式 (1) は次式のように表現し

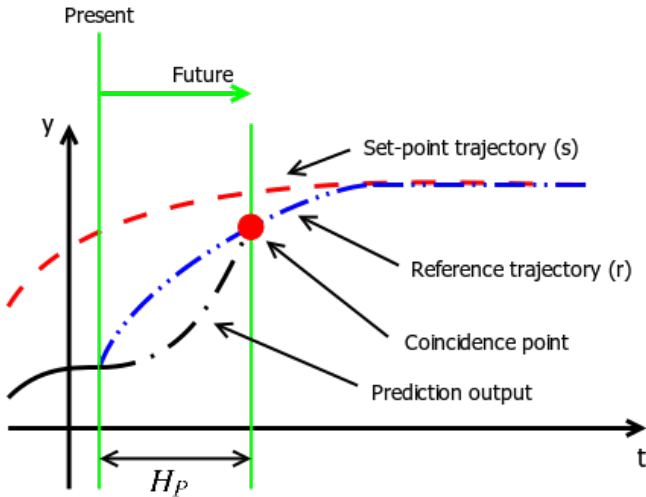


Figure 2: Parameters of MPC

直せる。

$$\dot{z} = \alpha v(t - t_0) \quad (2)$$

#### 4 モデル予測制御

近年注目されている MPC は、制御対象のモデルを用いて将来の動きを予測し、制御入力を決める制御方法である。基本的な考え方は、有限の未来の時間区間の動きを予測し、その動きを最適化するように現在の入力を決定するというものである。また、外乱補償器の付加が容易で、むだ時間やモデル誤差にロバストな制御系を構成することができる。

本稿では、離散時間モデルに基づく最も基本的な MPC として後退ホライズン方策 [1] による制御器を設計する。Figure 1 にブロック線図を示す。

##### 4.1 MPC の設計パラメータ

MPC で用いる 2 種類の軌道と設計パラメータとを Figure 2 に示す。設定値軌道 (set-point trajectory) は、ロボットに与える最終的な所望軌道である。一方、参照軌道 (reference trajectory) は現在の制御対象の出力から設定値軌道に向かうまでの軌道である。参照軌道と制御対象の予測出力が一致する点を一致点 (coincidence point; CP) といい、MPC 制御器はこの一致点に向かうような入力を生成する。MPC 制御器の性能は、以下の設計パラメータに依存する。

- $T_{ref}$  : 設定値軌道に対する参照軌道の時定数。
- $H_p$  : 予測ステップ数の長さ。何ステップ先まで予測を行うか。これは許容される計算時間に依存する。
- 入力変化の数 :  $H_p$  ステップ間で入力が何ステップ変化するか。

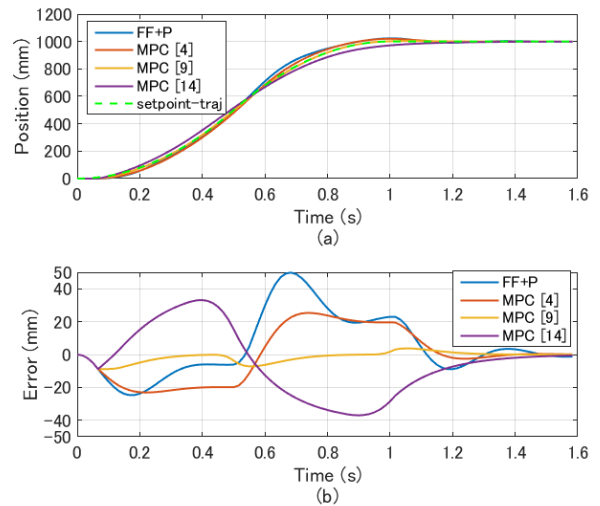


Figure 3: Comparison between FF+P and MPC

- 一致点の数 :  $H_p$  ステップ間で制御対象の予測出力と参照軌道との一致点をいくつ持つか。

##### 4.2 MPC で使用するモデル

MPC 制御器内部には制御対象の出力を予測するためにモデルが必要になる。制御対象としているロボットには第 3 節に述べたとおりむだ時間  $t_0$  が存在するが、これはモデル化誤差と考える。つまり、式 (2) においてむだ時間  $t_0$  を  $t_0 = 0$  としたものをモデルとして考えることにする。

式 (2) の連続時間モデルを離散化すると、

$$z(k+1) = z(k) + \alpha T_s v(k) \quad (3)$$

となる。ただし、サンプリング周期を  $T_s$ 、離散時間を  $k$  と表している。

##### 4.3 設計した制御器の性能

設計した制御器の性能を数値実験により示す。ここでは、 $x(0) = 0 \text{ mm}$  から  $x(t_f) = 1000 \text{ mm}$  までの移動を考え、この 2 点間をつなぐ多項式軌道を設定値軌道とする。ただし、 $t_f$  は理想的な移動し終える時間とする。 $t_f = 1.0 \text{ s}$ ,  $T_{ref} = 5/60 \text{ s}$ ,  $H_p = 4, 9, 14$ , 入力変化の数を 1、一致点の数を 1 としたときの結果を Figure 3 に示す。

比較のため、従来制御手法の一つとして、フィードフォワード補償器つき比例 (FF+P) 制御による結果も合わせて示してある。Figure 3(a) はロボットの実際の出力と設定値軌道の時間による変化、(b) は設定値軌道と実際の出力との誤差を示している。

Figure 3(b) から  $H_p$  の値が大きくなると誤差は小さくなっているが、 $H_p = 14$  の場合は逆に誤差が増えることがわかる。したがって、MPC には最適な予測区間が存在すると思われる。

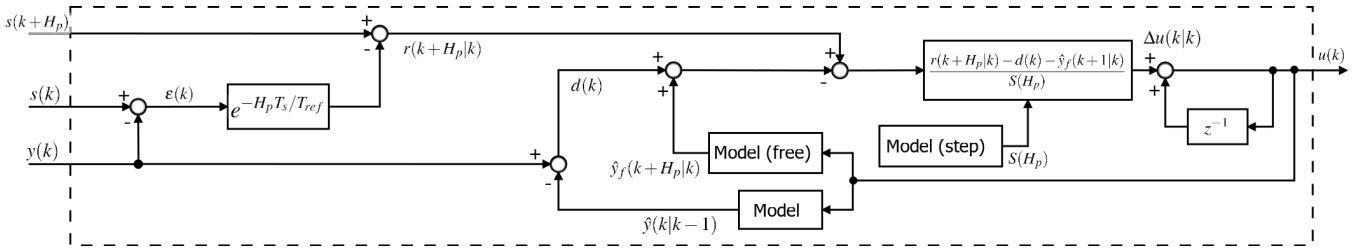


Figure 1: Block diagram of MPC

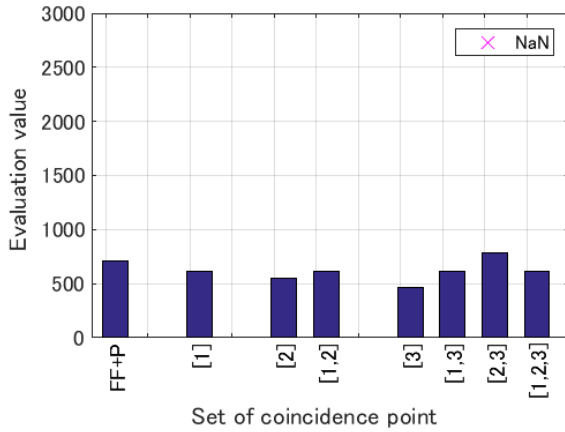


Figure 4: Performance with  $H_p = 1, 2, 3$  steps

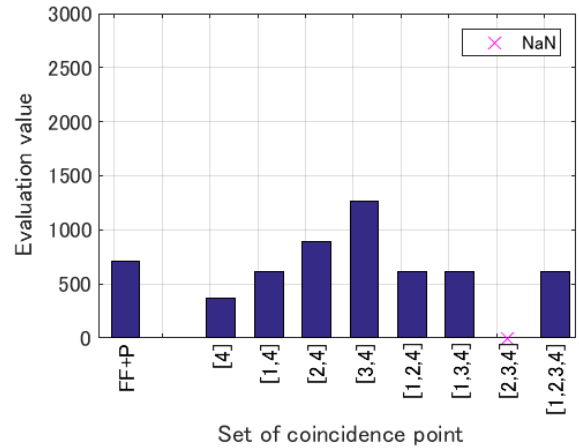


Figure 5: Performance with  $H_p = 4$  steps

## 5 設計パラメータと制御性能との関係

第 4.3 節と同じ問題設定のもと MPC 制御器の設計パラメータと制御性能との関係を明らかにする。評価値として、ロボットの実際の軌道と設定値軌道との差の二乗和を移動にかかる時間ステップ数で規格化したものを考える。誤差の二乗和を評価値としているので評価値が小さいほどよい制御性能が得られていることを意味する。

式 (1) において、むだ時間  $t_0$  の値は文献 [4] の推定法により  $t_0 = 0.07206$  と求まった。また、スケールパラメータ  $\alpha$  は予備的な実験により  $\alpha = 0.9$  とした。

本節では一致点の数と予測区間中に許される入力変化の数とが等しい場合に限定して、1) 一致点のパターンと制御性能との関係、2) 予測区間の長さとの関係、3) 時定数と制御性能との関係に着目する。なお、本節では一致点のパターンの表記について次のように定義する。

### 一致点の表記

ある時刻  $k$  において、予測区間が  $N_n$  ステップ。一致点の時刻が  $k + N_1, k + N_2, \dots, k + N_n$  であるとき、一致点の集合を  $[N_1, N_2, \dots, N_n]$  で表す。

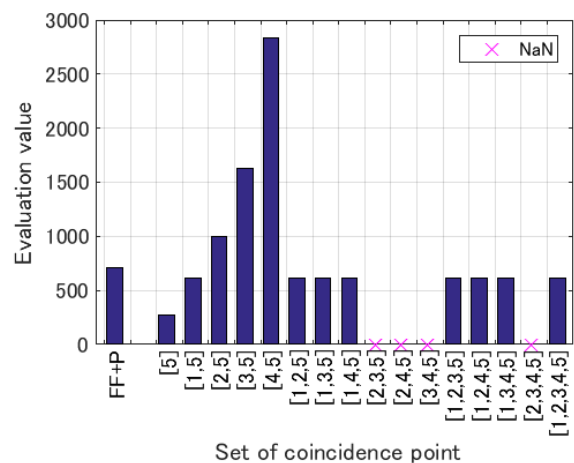


Figure 6: Performance with  $H_p = 5$  steps



### 5.1 一致点のパターンと制御性能との関係

予測区間の長さを5ステップまでとし、その中で考えられる一致点のすべてのパターンに対し、評価を行った。結果を Figures 4~6 に示す。図中の「NaN」は制御で入力を求めるための行列が特異となり、値が不定であることを示している。Figures 4~6 から次の2点がいえる。

- (A) : 一致点の集合に [1] を含む場合、評価値はすべて同じである。
- (B) : 一致点が複数個あり、一致点の集合に [1] を含まない場合、最初の一致点の位置に比例して評価値が大きくなる。

(A) に関して、一致点の集合が [1] と [1,3] の場合を例として考える。このときの入力生成の様子を Figure 7 に示す。Figure 7 のように、MPC では設定値軌道に収束するような参照軌道が時刻  $k$  において生成されるが、設定値軌道と制御対象の出力が等しければ一致点の集合によらず、生成される参照軌道は同じである。参照軌道は同じであるから、一致点の位置が同じであれば Figure 7 のように1ステップ目の入力は一致する。よって、入力が一致するので制御対象の出力も一致し、誤差も等しくなる。Figure 8 に一致点が [1] と [1,3] の入力と出力の変化を示す。Figure 8 から入力と出力が一致しているのが分かる。

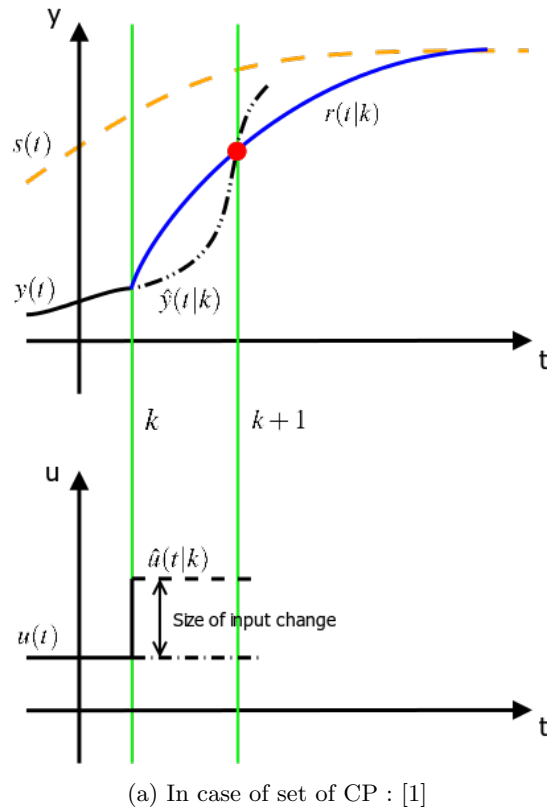
(B) は、例として一致点が [3,4] の場合を考える。このときの MPC 制御器による入力生成は Figure 9 のようになると考えられる。一致点の集合が [3,4] の場合、3, 4 ステップ目で参照軌道と制御対象の予測出力が一致するよう、最初の2ステップ目に変化する入力が生成される。しかし、制御対象に実際印加される入力は1段階目の入力だけである。その結果、Figure 9 のように1段階目の入力のみによる出力と1段階目と2段階目の入力による出力では差が生じる。1段階目の入力のみでは一致点で参照軌道と制御対象の予測出力は一致しないため誤差が生じたと考えられる。最初の一致点の位置が伸びるほど誤差が大きくなるのは1段階目の入力のみによる出力によって生じる誤差がステップが進むにつれ大きくなるからだろうと思われる<sup>1</sup>。

予測区間の長さごとに誤差を見ると、どの予測区間の長さにおいても一致点の数が1つの場合が最も評価値が小さくなる結果となった。

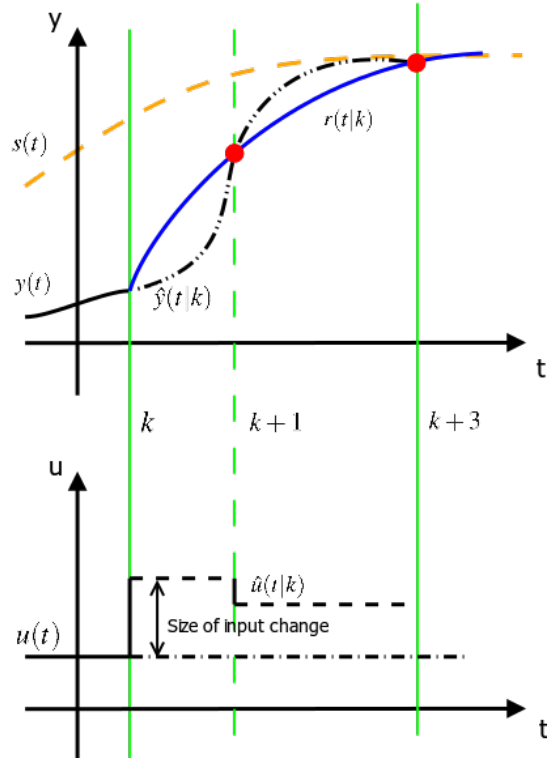
### 5.2 予測区間の長さとの関係

一致点の数を一つに限定して予測区間の長さについて評価を行った結果を Figure 10 に示す。Figure 中から予測区間の長さが9ステップまでは評価値が単調減少する一方、10ステップ以降は評価値が大きくなる結果になった。

<sup>1</sup>Figure 9 の時刻  $k+3$  と  $k+4$  では後者のほうが誤差が大きい。



(a) In case of set of CP : [1]



(b) In case of set of CPs : [1,3]

Figure 7: MPC with set of CPs [1] and [1,3]

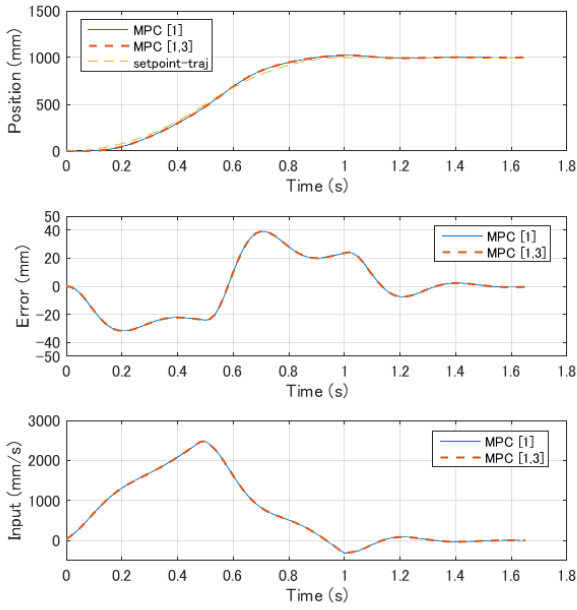


Figure 8: MPC with set of CPs : [1] vs [1,3]

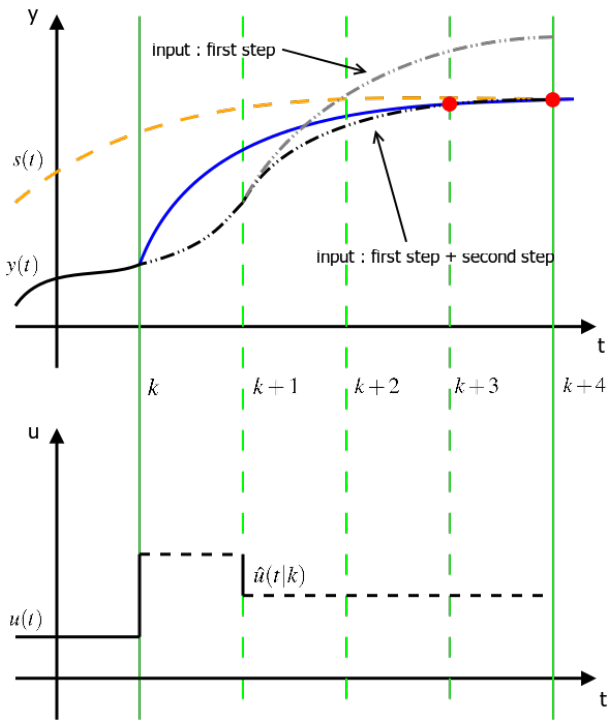


Figure 9: MPC with set of CPs [3,4]

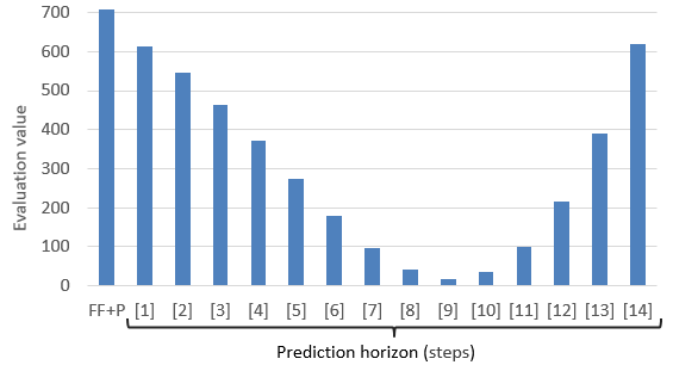


Figure 10: Performance w.r.t. various prediction horizon

この誤差は以下に示す二つの要因から生じるものだと分析することができる (Figure 11(a))。第一の要因は、設定値軌道  $s$  と参照軌道  $r$  から生じる誤差  $r - s$  である。参照軌道が設定値軌道に収束する速さは MPC の設計パラメータの一つである時定数によって決まる。時定数の値が大きいと参照軌道が設定値軌道へ収束するのに時間がかかるため、予測区間の終点では参照軌道と設定値軌道が一致せずその差が誤差となって生じてしまう。第二の要因は実際の出力  $y_p$  と参照軌道  $r$  から生じる誤差  $y_p - r$  である。この差はむだ時間などモデル化誤差が影響に起因する。

これら二つの誤差を予測区間の長さごとに示したのが Figure 11(b) である。Figure 11(b) から 2 つの誤差の合計が最も少ない予測区間の長さ 9 のときが Figure 10 で最も誤差が少ない場合と一致しているのがわかる。また、誤差  $r - s$  は予測区間の長さが伸びれば参照軌道は設定値軌道に収束していくので、予測区間が長くなるほど誤差が少なくなっていることがわかる。誤差  $y_p - r$  について見ると、予測区間の長さが 10 以降で急激に誤差が増えていることがわかるが、これは一致点を予測区間の終点の一点のみとしているため、予測区間中の軌道追従については考慮されず、そこから誤差が生じるためである。

### 5.3 時定数と制御性能との比較

予測区間の長さを 1 ステップとし、時定数を変化させて評価した結果が Figure 12 である。Figure 12 から評価値は時定数が小さくなるほど減少する傾向にあるが、 $T_{ref} = 0.02s$  では大幅に増加している。このとき MPC 制御器は制御対象を制御できておらず出力が発散振動をしていた。この結果から制御対象の特性に依存して臨界点が決まっていると考えられる。しかしながら、具体的にどんな点が原因となっているかについては今後、検討を進める必要がある。



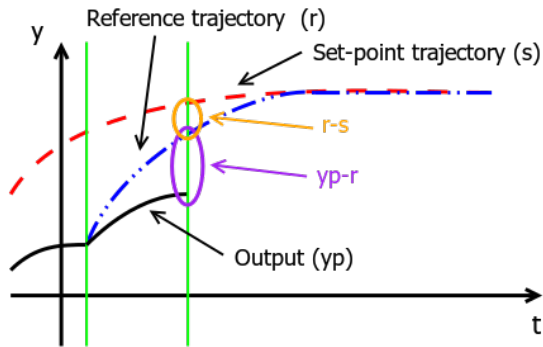
## 6 おわりに

本稿では、設計した MPC 制御器に対して様々な設計パラメータを与え数値実験を行ったことによって、本稿で考えた運動モデルとロボットに対して、最適な予測区間が存在するなどの設計パラメータと制御性能との関係を示すことができた。また、従来の制御手法と比べて軌道追従誤差が減るなどのより高精度な制御が可能であることを示すことができた。これによって従来より高精度なボールのパス回しが可能となるだろう。

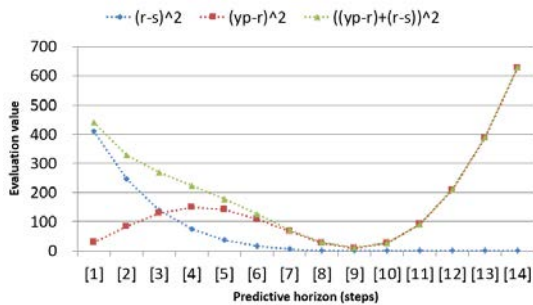
今後は、実機システムへの実装、パス精度での評価などが課題である。

## 参考文献

- [1] J. M. Maciejowski : “モデル予測制御 — 制約のもとでの最適制御—”, 足立・菅野 訳, 東京電機大学出版局, 2005.
- [2] M. Zarghami, A. Fakharian, A. Ganjali Poudeh and A. Adhami-Mirhosseini: “Fast and precise positioning of wheeled omni-directional robot with input delay using model-based predictive control,” Proceedings of the 33rd Chinese Control Conference (CCC'14), pp.7800–7804, 2014.
- [3] K. Kanjanawanishkul and A. Zell: “Path following for an omnidirectional mobile robot based on model predictive control,” Proceedings of IEEE International Conference on Robotics and Automation (ICRA'09), pp.3341–3346, 2009.
- [4] Y. Wu, X. Qiu, G. Yu, J. Chen, X. Rie, Y. Wu, and R. Xiong: “Extended TDP of ZjuNlict 2009,” Proceedings of RoboCup2009, pp.14–16, 2009.



(a) Causes of tracking error



(b) Decomposition of tracking error

Figure 11: Analysis of tracking error

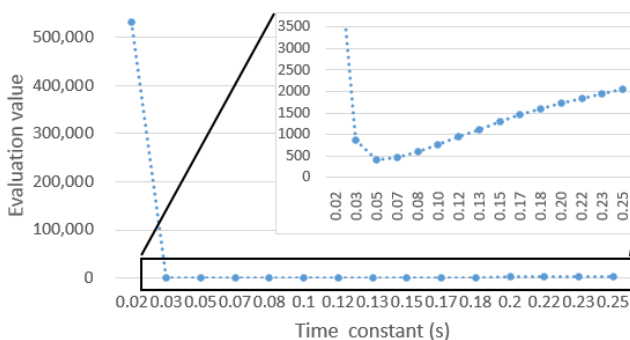


Figure 12: Performance w.r.t. various time constants

# 物体認識における背景の有無による学習効果に関する一考察

Recognizing Objects Without a Back Ground Image Using Stereo Cameras.

津田 諒太 , 植村 渉

Ryota Tsuda and Wataru UEMURA

龍谷大学

Ryukoku University

tsuda@vega.elec.ryukoku.ac.jp, wataru@rins.ryukoku.ac.jp

## Abstract

ファクトリーオートメーションを課題とした RoboCup リーグとして Logistics League がある。このリーグでは、工場内の加工機械を模した MPS と呼ばれる装置を用い、指示に応じて素材を加工する必要がある。MPS は高さ約 1m の直方体であり、上面は 70cm × 35cm であり、機械の種類によって上に載っている装置が異なる。ここで、上に載っている装置の違いから、MPS の種類を見分ける必要がある。現在は、MPS の長辺側の側面に ALVAR の AR タグを取り付けて、識別できるようにしているが、テクニカルチャレンジとしてマーカ無しでの識別が求められている。

本研究では、ニューラルネットワークを用いた学習を用いることで物体識別を行う。このとき、ステレオカメラの視差による距離情報を用いて、MPS の上のみを切り出すことで、識別率に与える影響を調べる。

## 1 はじめに

近年、ファクトリーオートメーションが注目されている。研究課題の一つとして工場における生産ラインの変化への適応があげられる[1]。このような工場では加工機械の位置や種類が頻繁に変化する。機器の情報を人の手で設定する代わりに、ロボット自身がカメラなどの入力機器を用いて加工機械の形状及び種類を識別する方法が必要とされている。

このような画像による物体識別には、画素マッチングや画像学習が用いられる。画素マッチングは入力画像と比較用画像内の同一座標の画素値が近似するかどうかで物体識別を行う。機械の種類ごとに大量の画像を保持する

必要があり、また明暗の変化に弱い。画像学習は対象の画像を多数用意し、画像内に共通する要素を見つけ出す。学習後はそれら要素のみを利用するため画像を保持しなくて良い。しかし学習に必要な画像は数千枚の規模であり、数千枚の画像の撮影には膨大な時間がかかる。工場での撮影を行うことは現実的でない。本研究では少ない枚数しか撮影できない環境を想定し、学習が効果的に進むように画像情報から学習に必要な部分のみを抜き出して用いることを考える。ここではカメラを 2 つ用意し、ステレオマッチングにより距離を推定することで、背景画像を除いた画像を用意しその学習効果について評価を行う。

## 2 ニューラルネットワーク

ニューラルネットワークとは脳構造を模した学習方法の一つであり、入力層、中間層、出力層に分かれている。層のそれぞれは前の層の出力に重み付けした総和を取り、その値によって自分が出力を行うかを決定する。この際の重みを変化させることで学習を行う。

今回のニューラルネットワークは多層パーセプトロンを用い、中間層は 1 層、学習アルゴリズムは逐次誤差逆伝搬アルゴリズムを用いる。ニューラルネットワークには複数のパラメータがあり、パラメータによって学習の良し悪しが変化する。今回は表 1 のパラメータで総組み合わせ (32 パターン) の学習を行う。

中間層数	100, 1000
最大繰り返し数	100, 1000
学習打ち切り最小誤差	0.000001, 0.00001
重み変更度合い	0.05, 0.1
重み変更抑制	0.05, 0.1

Table 1: ニューラルネットワークの設定パラメータ

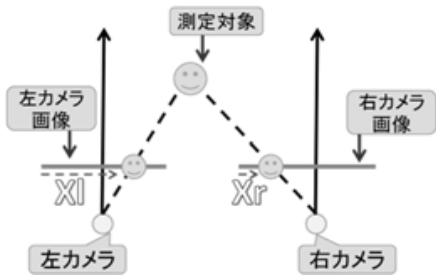


Figure 1: ステレオマッチングの原理

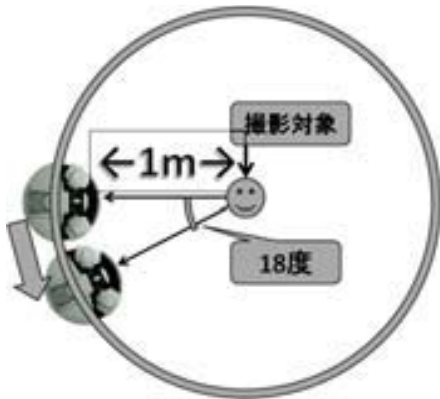


Figure 2: 撮影状況

### 3 ステレオマッチング

ステレオマッチングとは、2台のカメラを設置し、2枚の画像の視差を取ることで画像内の物体までの距離を推定する方法である。

基準となる左の画像内の要素に対応する要素を右の画像から探し、その視差 ( $X_l - X_r$ ) から距離を算出する (図1)。

### 4 実験と結果

距離情報により背景を除いた場合の学習効果を確認するために、背景あり学習画像での学習と、背景なし学習画像での学習を行い、学習が収束した後に両者の誤認識率を比較する。

画像撮影は Festo 社から販売されている全方位移動可能ロボット Robotino を用いる (図2)。

Robotino は撮影対象から一定の距離を保ちながら対象の方向を向きながら移動し、撮影を行う。撮影は18度ごと、計20回撮影を行った。識別対象物として1) 持運びTV、2) スピーカ、3) コーヒーメーカーの3つを対象とした。背景あり画像は左カメラのものを使用した。背景なし画像は、左カメラ画像を基準に作成した距離画像の閾値以上の座標を元画像で黒く塗りつぶすことで背景を取り除いた。背景を除いた影響を測定するため、学習用画像と評価用画像の撮影は別の場所で行った。画像の明暗の影響

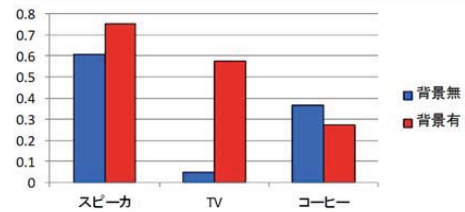


Figure 3: 背景の有無による誤認識率の違い

を排除するため、学習画像及び評価画像はエッジ抽出を行った。学習画像をニューラルネットワークに入力して、学習回数が最大繰り返しに達するか、学習結果の誤差が打ち切りの値以下になるまで学習を行った。完成した識別器に評価画像を入力し、識別結果を出力した。認識率と誤認識率の一部を図3に示す。

### 5 おわりに

大量の画像撮影ができない環境下において、少数の画像でも有効な学習結果を出力するために、ノイズとなりうる背景部分を除いての学習を提案した。背景有り画像での学習は、殆どの学習パラメータで有効な結果が出ず、最も優秀な結果だったものでも識別にはかなり不十分であった。一方、背景なし画像での学習は、背景なしと比較して優れた結果が出力された。複数枚の評価画像があれば十分識別可能な識別器が作成できた。今後の課題は、パラメータの設定に左右されない深層学習への適用が必要である。

### 謝辞

本研究は JSPS 科研費 15K16313 の助成を受けたものである。

### 参考文献

- [1] Frederik Zwillig, Tim Niemueller, Gerhard Lake-meyer, "Simulation for the RoboCup Logistics League with Real-World Environment Agency and Multi-level Abstraction", RoboCup 2014: Robot World Cup XVIII, pp 220 – 232, 2014.

© 2016 Special Interest Group on AI Challenges  
Japanese Society for Artificial Intelligence  
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にお願いします.)

---

## AI チャレンジ研究会

### 主査

光永 法明

大阪教育大学 教員養成課程 技術教育講座

### 主幹事

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 大学院  
情報理工学研究科

### 担当幹事

植村 渉

龍谷大学 理工学部 電子情報学科

### 幹事

公文 誠

熊本大学 大学院 自然科学研究科

### 幹事

中村 圭佑

(株) ホンダ・リサーチ・インスティテュート・  
ジャパン

## Executive Committee

### Chair

Noriaki Mitsunaga

Department of Technology Education  
Osaka Kyoiku University  
mitunaga @ cc.osaka-kyoiku.ac.jp

### Secretary

Kazuhiro Nakadai

Honda Research Institute Japan Co., Ltd.  
/ Graduate School of Information  
Science and Engineering,  
Tokyo Institute of Technology

### Wataru Uemura

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

### Makoto Kumon

Graduate School of Science and  
Technology,  
Kumamoto University

### Keisuke Nakamura

Honda Research Institute Japan Co., Ltd.

---

SIG-AI-Challenges web page; <http://winnie.kuis.kyoto-u.ac.jp/SIG-Challenge/>