

# 強化学習を用いたサッカーエージェントのボール保持行動 獲得

## A Reinforcement Learning for a Ball Holding Task in the RoboCup Soccer 2D Simulation

秋山英久† 岡山智彦‡ 中島智晴‡

Hidehisa AKIYAMA† Tomohiko OKAYAMA‡ Tomoharu NAKASHIMA‡

福岡大学† 大阪府立大学‡

Fukuoka University†, Osaka Prefecture University‡

akym@fukuoka-u.ac.jp, {tomohiko.okayama@ci.cs, tomoharu.nakashima@kis}.osakafu-u.ac.jp

## 概要

In this paper, we propose a reinforcement learning approach for a ball holding task in the RoboCup soccer 2D simulation environment. We applied the Sarsa( $\lambda$ ) algorithm and evaluated several reward design in order to acquire the ball holding action with practical performance. The experiment results show that our approach can outperform the hand-coded heuristic method.

## 1 はじめに

マルチエージェントシステムにおいて、集団の協調行動のパフォーマンスを向上させるには個々のエージェントの性能向上も重要である。特に、自律的な意思決定を行う敵対的なエージェントが存在する場合、環境は動的であり、最適な行動ルールをハンドコーディングで実装することは難しい。このような環境下でのタスクにおいては、試行錯誤を通じて最適な制御規則を自動的に獲得できる強化学習の適用が有望と考えられる。敵エージェントが存在する動的環境下での制御技術が要求される問題として、サッカーのような対戦型スポーツが考えられる。本研究

では、サッカーにおける基本的なプレイヤーの制御技術である、単体エージェントによるボール保持行動を扱う。実験環境として RoboCup サッカー 2D シミュレータを用い、強化学習によって実用的な性能のボール保持行動を獲得させる。

## 2 関連研究

RoboCup サッカー 2D シミュレーション環境において、エージェントの制御技術や戦術的行動の獲得を強化学習によって試みた研究が数多く行われている。Gabel ら[2]らや Nakashima ら[4] はボールを捕捉する動作の精度向上を強化学習によって実現している。さらに、Gabel らは、ボールを保持する敵エージェントに対する守備行動を実用的な性能で獲得することに成功している[3]。Stone ら[1]は複数の敵味方エージェントが存在する環境でボールを保持することを目標とする Keepaway タスクを提案し、複数エージェントの行動政策の獲得を試みている。Carbalho ら[5]は、1対1でのドリブルタスクへの強化学習の適用している[5]。本稿では、サッカープレイヤーとしてより基本的な技術であるボール保持タスクを取り上げ、動的環境での明確な目標状態を設定できないタスクにおいて、強化学習の適用によって実用的な性能を

得ることを目指す。

### 3 強化学習によるボール保持行動獲得

#### 3.1 ボール保持タスク

本稿では、敵エージェントにボールを奪われないことを目的とした単体のエージェントによるボール保持行動をボール保持タスクと呼ぶ。本稿で扱うボール保持タスクは、ボールを保持するエージェントとボールを奪おうとする敵エージェントとの1対1での環境とする。ボールを保持するエージェントはその場から移動せず、ボールをキックする行動のみ可能とする。

RoboCupサッカー2Dシミュレータでは、エージェントの位置を中心とした半径  $kickable\_area$  の円周内部がキック可能領域となる(図1)。エージェントのキック可能領域内にボールが存在すれば、そのエージェントはボールをキックすることができる。よって、本稿におけるボール保持タスクは、敵エージェントのキック可能領域内へボールを侵入させないように、ボール保持エージェントのキック可能領域内でボールを移動させることが目的となる。これを達成するためには、現在の観測状態に応じて、自分自身のキック可能領域内の適切な位置へボールを移動させなければならない。しかしながら、自律的に行動する敵エージェントを避けるための最適なボール移動位置は明らかではなく、ハンドコーディングでのルール実装は難しい。そこで、本稿では強化学習によってボール保持タスクの解決を試みる。

#### 3.2 強化学習アルゴリズム

本稿では、強化学習アルゴリズムとして  $Sarsa(\lambda)$ [7] を使用する。  $Sarsa$  では1ステップ毎に直前の状態-行動対のみ価値を更新するのに対して、  $Sarsa(\lambda)$  では、数ステップ遡って過去の価値を更新することができる。ボール保持タスクでは、

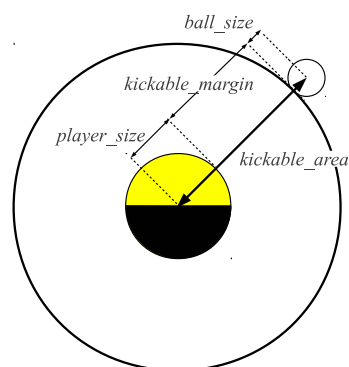


図1: エージェントのキック可能領域。エージェントとボールとの距離が  $kickable\_area$  以下であれば、エージェントはボールを蹴ることができる。エージェントの半径 ( $player\_size$ ) と  $kickable\_margin$  の大きさはエージェントごとに異なる。

ステップ前の行動がボール保持の失敗に影響を与えると予想されるため、  $Sarsa(\lambda)$  が適していると考えられる。行動選択の手法として  $\epsilon$ -greedy を用いる。

ボール保持タスクにおけるエピソードは、ボール保持エージェントのキック可能領域の外へボールが出てしまい、ボールの制御を失った時点で終了とする。RoboCupサッカー2Dシミュレータは離散時間シミュレータであるため、シミュレータの1シミュレーションステップを強化学習における1ステップとして扱う。

#### 3.3 状態空間の設計

RoboCupサッカー2Dシミュレータは連続状態行動空間の環境である。強化学習をサッカーシミュレーション環境へ適用するために、状態空間を近似する手法として次元タイルコーディング[7]を使用する。状態変数として以下の情報を用い、それぞれにタイリングを用意する。各状態変数で用いる座標系は、ボール保持エージェントの現在位置を原点とし、体の向きを  $x$  軸正方向とする。

- ボール保持エージェントと敵エージェントとの距離

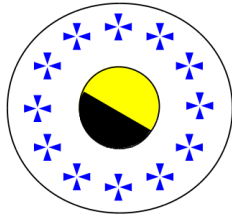


図 2: あらかじめ設定する 12 個のボール移動候補点.

- 敵エージェント位置の方向
- ボール保持エージェントとボールとの距離
- ボール位置の方向
- 敵エージェントの移動速度の大きさ
- 敵エージェントの移動速度の方向
- ボールの速度の大きさ
- ボールの速度の方向

### 3.4 行動空間の設計

サッカーシミュレーション環境では、状態空間と同様に行動空間も連続である。行動空間を離散化するために、エージェントのキック可能領域内にボールの移動候補位置をあらかじめ設定する。本稿ではこの候補位置を 12 箇所限定した。まず、エージェントの体の正面を基準にエージェントの周囲を 12 分割する。分割した方向それぞれに、エージェントの体の中心から  $ball\_size + kickable\_margin \times 0.72 + player\_size$  の距離だけ離れた位置にボールの移動候補位置を設定する (図 2)。

### 3.5 報酬の設計

本稿では、次の 4 種類の報酬設計を用意し、数値実験によって獲得される政策の性能を調査する。

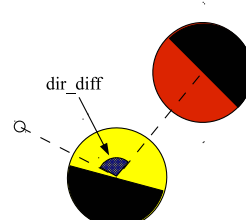


図 3: ボール保持エージェントを中心とし、ボール方向と敵方向のなす角の絶対値を  $dir\_diff$  とする。

#### 3.5.1 報酬設計 1: エピソード終了時の負の報酬のみ

エピソード終了時に、ボールの制御を失ったことへの罰則として報酬-1.0 を与える。

#### 3.5.2 報酬設計 2: 保持成功ステップ数の反映および角度差によるステップごとの報酬

報酬設計 1 のように、エピソード終了時には負の報酬を与えるのは妥当と考えられる。しかし、長時間のボール保持を達成した場合には、ボール保持成功ステップ数の大きさに応じて罰則を小さくすることが望ましいと予想される。そこで、エピソード終了時に与える報酬を  $-\frac{50.0}{steps}$  とすることで、ボール保持成功ステップ数に応じて負の報酬の絶対値を小さくする報酬設計を導入する。 $steps$  はボール保持に成功したステップ数、すなわち、エピソードの長さである。

さらに、エピソード中のステップ毎に報酬を与える。エージェントがボール保持に成功している状態では、ボールは敵から遠い位置に存在する。そこで、図 3 に示される角度差  $dir\_diff$  の大きさに応じた報酬を設定する。ここでは、ステップ毎に  $\frac{dir\_diff - 120.0}{15}$  を報酬として与える。

### 3.5.3 報酬設計 3: 保持成功ステップ数の反映およびステップごとに一定値の報酬

エピソード終了時の負の報酬として  $-\frac{50.0}{steps}$  を使用する。また、エピソードのステップ毎に、敵エージェントがキック可能な状態であれば-1.0、キック不可能な状態であれば 1.0 の報酬を与える。

### 3.5.4 報酬設計 4: 累積エピソード数の反映

エピソードの実行回数を  $episodes$  とすると、エピソード終了時の負の報酬として  $-5.0 - \frac{\sqrt{episodes}}{20}$  を与える。

## 4 実験

### 4.1 実験設定

実験環境として、公式 RoboCup サッカー 2D シミュレータである rcssserver バージョン 15.0.1 を使用する。今回の実験では、観測誤差による影響を小さくするために、ノイズのない完全知覚情報を得られる fullstate 環境を用いる。ボール保持エージェントと敵エージェントの身体能力は同じとする。

本稿で提案した強化学習アルゴリズムに基づいたボール保持行動を RLHoldBall と呼ぶ。強化学習の各エピソードを実行中、ボール保持エージェントは常に RLHoldBall を実行する。敵エージェントは、ボールが自身のキック可能領域に存在すればボールをキックし、そうでなければボールを捕捉するための行動を実行する。ボール保持エージェントとは異なり、敵エージェントはボール捕捉のために移動動作を実行する。本来のシミュレータの仕様ではエージェントの移動には体力を消費する仕組みとなっているが、今回の実験では体力の消費は無いものとする。ボール保持エージェント、敵エージェントのいずれも agent2d-3.1.0[6]を使用する。

表 1: 使用した強化学習パラメータ。

$\alpha$	0.125
$\lambda$	1
$\epsilon$	0.01
$trace\_value$	1

#### 4.1.1 エピソードごとの初期設定

エピソード開始時の初期状態として、次のようにボール、ボール保持エージェント、敵エージェントを配置する。

1. ボール保持エージェントをフィールドの中央に配置する。
2. フィールド中央からの距離が  $ball\_size + kickable\_margin \times 0.72 + player\_size$  で方向がランダムな位置に、停止した状態でボールを配置する。
3. 敵エージェントをフィールド中央から 3.0m だけ離し、フィールド中央に対してランダムな方向に配置する。

エピソード開始後、ボール保持エージェントはボール保持行動を取り、敵エージェントはボール捕捉行動を取る。ボール保持エージェントのキック可能領域の外へボールが出た時点でエピソードが終了する。

#### 4.1.2 強化学習パラメータの設定

今回の実験では、すべて 50000 回のエピソードを繰り返して学習を行う。強化学習のパラメータとして、Carvalho ら[5]と同じ設定を使用する(表 1)。 $\alpha$  は学習率、 $\lambda$  はトレース減衰パラメータ、 $\epsilon$  はランダムに行動を選択する確率である。適格度トレースの更新には入れ替え更新トレースを用いる。 $trace\_value$  は入れ替え後の適格度トレース  $e$  の値である。



表 2: 状態変数の種類, それぞれの値の範囲, タイルの分割数.

状態変数	範囲	分割数
$dist(holder, opponent)$	[0.0, 3.0]	10
$angle(holder, opponent)$	[-180, 180]	12
$dist(holder, ball)$	[0.0, 3.0]	10
$angle(holder, ball)$	[-180, 180]	12
$velNorm(opponent)$	[0.0, 3.0]	10
$velAngle(opponent)$	[-180, 180]	12
$velNorm(ball)$	[0.0, 3.0]	10
$velAngle(ball)$	[-180, 180]	12

#### 4.1.3 状態変数の設定

表 2 に使用する状態変数を示す.  $dist(A, B)$  は  $A$  と  $B$  との距離,  $angle(A, B)$  は  $A$  の体の方向を  $x$  軸方向とした場合の  $B$  位置の相対角度,  $velNorm(A)$  は  $A$  の速度の大きさ,  $velAngle(A)$  は  $A$  の速度の方向とする. 各変数に対してタイリングは 32 層とする. RLHoldBall として, これらの状態変数のうち 6 個あるいは 8 個を使用する 2 種類の設定を用意する. 状態変数の数が 6 の場合, ボールの位置速度情報は状態変数として使用されない.

#### 4.1.4 他手法との比較

提案する強化学習アルゴリズムの性能を比較評価するために, 以下の手法でもボール保持タスクを実行する.

- **RandomHoldBall:**  
ボールの移動先をランダムに決める.
- **HandCoding:**  
ハンドコーディングで設計した関数で移動候補点を評価し, もっとも高い評価値を得られた位置をボールの移動先に決定する. RoboCup2011 準優勝の HELIOS2011 が使用したものと同一のものであ r..

- **FarthestHoldBall:**

敵からもっとも遠くなる位置でボールを保持する.

RandomHoldBall はボール移動候補位置として RLHoldBall と同じ位置を使用する. ただし, ボールの移動先は完全にランダムに決定される. HandCoding では, いくつかのルールとパラメータを人出で調整して評価関数を作成する. RLHoldBall と同様に複数の移動候補位置を生成し, 評価関数が出力する評価値に基づいてボールの移動位置が決められる. FarthestHoldBall は, 現在の敵位置からもっとも遠い位置を候補位置から選択し, ボールを移動させる. ボール移動位置とボール保持エージェントとの距離は RLHoldBall と同じ値を使用する.

## 4.2 実験結果

各手法で 50000 エピソードを実行した結果を図 4 に示す. このグラフはボール保持成功ステップ数の 1000 エピソード毎の移動平均を示している. RLHoldBall には報酬設計 2 を用いている. RLHoldBall は HandCoding を上回る性能を示しており, 人手で設計した評価関数を用いるよりも高い性能を持つ行動政策を強化学習によって獲得できたことが分かる.

ボールの位置と速度を状態変数として追加し, 状態変数の数を 6 から 8 に増やしても, 結果に有意な差は見られない. これは, ボール保持行動の性能に関して, ボールの位置や速度は影響が小さい状態変数であったためと考えられる.

3.5 節の各報酬設計を使用した実験結果を図 5 に示す. グラフ中では, MinusOnly が報酬設計 1, PlusMinusDirDiff が報酬設計 2, PlusMinusKickable が報酬設計 3, PlusMinusSqrt が報酬設計 4 を表す. このグラフは保持成功ステップ数の 1000 エピソード毎の移動平均を示している. PlusMinusDirDiff と PlusMinusSqrt がほぼ同等の性能を示しており, 他の報酬設計を上回る性能を得られた.

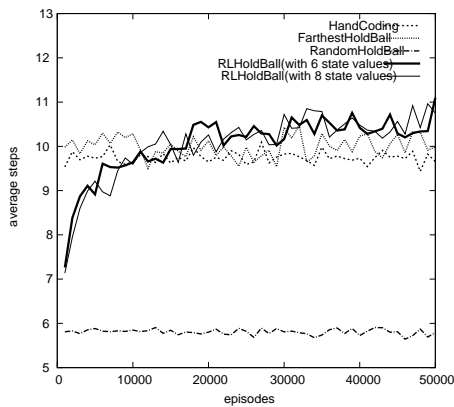


図 4: 報酬設計 2 と他手法との比較. 保持成功ステップ数の 1000 エピソード毎の移動平均.

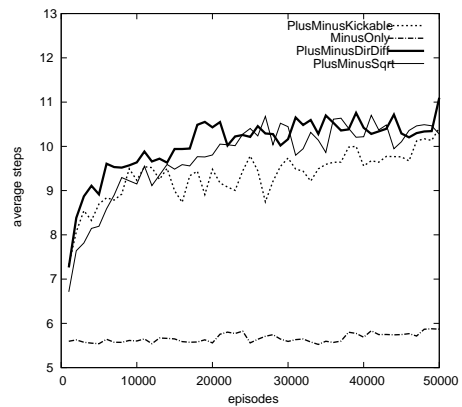


図 5: 各報酬設計の比較. 保持成功ステップ数の 1000 エピソード毎の移動平均.

## 5 まとめと今後の課題

本稿では, RoboCup サッカー 2D シミュレーション環境において, 強化学習を用いてボール保持行動を獲得させた. 数値実験により, 強化学習を用いることによって人出で設計した評価関数を用いる場合よりも高い性能が得られることを示した. 今後の課題として, 複数の敵エージェントへの対応, 敵エージェントの特徴にや状況に応じた政策の獲得などが考えられる.

## 参考文献

- [1] Peter Stone, Richard S. Sutton and Gregory Kuhlmann: Reinforcement Learning for RoboCup-Soccer Keepaway, *Adaptive Behavior*, 13(3), pp. 165-188, (2005).
- [2] Thomas Gabel and Martin Riedmiller: Learning a Partial Behavior for a Competitive Robotic Soccer Agent, *KI Zeitschrift*, vol.20, pp. 18-23, (2006).
- [3] Thomas Gabel, Martin Riedmiller and Florian Trost: A Case Study on Improving Defense Be-

havior in Soccer Simulation 2D: The NeuroHasle Approach. *RoboCup 2008: Robot Soccer World Cup XII*, pp. 61-72, (2008).

- [4] Tomoharu Nakashima, Masayo Udo and Hisao Ishibuchi: A Fuzzy Reinforcement Learning for a Ball Interception Problem, *RoboCup 2003: Robot Soccer World Cup VII*, pp.559-567, (2004).
- [5] Arthur Carvalho and Renato Oliveira: Reinforcement Learning for the Soccer Dribbling Task, *Proceedings of the 2011 IEEE Conference on Computational Intelligence and Games*, pp. 95-101, (2011).
- [6] 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).
- [7] 三上貞芳, 皆川雅章: 強化学習, 森北出版, (2001).

# エージェントの意思決定における振動低減

## Decreasing Oscillations in Decision Making of Agents

成 本 洋 介 † 中 島 智 晴 † 秋 山 英 久 ‡

Yosuke NARIMOTO† Tomoharu NAKASHIMA† Hidehisa AKIYAMA‡

大阪府立大学 † 福岡大学 ‡

Osaka Prefecture University† Fukuoka University‡

tomoharu.nakashima@kis.osakafu-u.ac.jp, akym@fukuoka-u.ac.jp

### Abstract

In this paper, we propose a method that decreases the number of the oscillations in decision making by adjusting the importance of the previous decision making using a weighting function so that the agent tends to continue the previously selected action. With some numerical experiments, it is confirmed that our proposed method decreases the oscillations in decision making. We also investigate how the proposed method influences the performance of the overall team strategy.

## 1 はじめに

近年、動的で不確実な状況下におけるエージェントのプランニングに関する研究が幅広く行われている。例えば、山田は [1] において、ソフトウェアエージェントにおけるプランニングの研究事例としてインターネットエージェントなどを紹介し、プランニングの研究課題について、不確実な状況に対応することや、任意時間で実行可能なプランを出力することを挙げた。神尾ら [2] は、複数のロボットによる協調作業のための経路プランニングに対して、ランダムサンプリングを用いてサブタスクに必要なサブゴールを自動的に生成する経路プランニングアルゴリズムを提案した。また、エージェント同士の協調行動だけでなく、対戦型ゲームにおける人工知能エージェントの意思決定に関する研究も注目を集めている。例えば、Onieva ら [3] は、カーレーシングシミュレーションにおいて、後ろからくる対戦車両に対して、ファジィ制御を用いたブロックング動作を行うカーエージェントを開発した。DeLooze ら [4] は、パックマンゲームにおいて、パックマンエージェントがどのようにゲームをプレイするかを学習するのに、ファジィQ学習が有効であることを示した。

本論文では、RoboCup サッカー 2D シミュレーションにおいて、エージェントが前のサイクルで選択した行動を次

のサイクルの意思決定で考慮することで、エージェントの意思決定の振動を低減する手法を提案する。本論文では、意思決定の振動を、前のサイクルで実行しようとしていた行動に対して、次のサイクルで違う行動を選択することと定義する。意思決定の振動が発生した場合、エージェントは時間を無駄に消費してしまうだけではなく、チーム全体を不利な状況にしてしまうという問題がある。提案手法では、本論文で使用されるエージェントの意思決定において、選択された行動の評価に対して修正式を加えることにより、意思決定の振動を低減する。

## 2 エージェントモデル

2D シミュレーションリーグにおいて、エージェントがどのように意思決定を行うべきかということは戦略開発における重要な要素である。例えば、Ma ら [5] は、専門知識のデータベースであるプランプールの概念を提案し、エージェントの協調行動のための方策探索計画法を適用した。Gspandl ら [6] は、自然言語のテキストから領域知識を自動的に抽出する手法を提案し、その手法をサッカーエージェントの意思決定に適用した。

本論文で使用するエージェントは最良優先探索を用いて、行動連鎖 [7] と呼ばれる木構造を構築することで意思決定を行う。以下に行動連鎖生成のアルゴリズムを示す。

Step 1: 現在の状態をルートノードに入力する。

Step 2: ノードにおいてエージェントが実行可能な、候補となる行動を生成する。  
(pass, dribble, shoot など)

Step 3: 生成された行動の評価値  $e$  を算出し、行動によって達成される状態とともに子ノードに追加する。

Step 4: ノード数が設定された最大数に到達していれば終了する。そうでなければ、 $e$  が最大となるノードを選択して、Step 2 へ戻る。

まず、エージェントが現在置かれている状態をルートノードに入力する。次に、ノードに入力された状態において、実行可能な行動の候補となる行動を生成する。生成された行動に対して評価値  $e$  を算出し、行動によって達成される状態とともに子ノードに追加する。ノードが追加されるごとに、 $e$  が最大となるノードを選択し、ノードにおける状態から再び実行可能な候補となる行動を生成する。これを繰り返すことで、ノード数があらかじめ設定された最大数に到達するまで探索木を成長させる。このとき、葉ノードにおいて以下の条件のいずれかを満たす場合は、その葉ノードでの子ノードの生成は行わないものとする。

- 木の深さがあらかじめ設定した値を越えた。
- ノードに入力された状態から行動が生成できない。
- 行動連鎖の終了条件に設定されている行動 (shoot) が生成された。

構築された木構造のルートノードから葉ノードまでのノード列をつなげると、ある行動連鎖が得られる。最も評価値の高い行動連鎖を選択することで戦略的によいと考えられる行動連鎖を実行することが可能である。

行動連鎖の例を図 1 に示す。図 1 において、ボールを持った 10 番のエージェントは次のような行動連鎖を生成する。まず、10 番のエージェントが 7 番に pass を行う。pass を受け取った 7 番は dribble で前進をして、その後 9 番に pass を行う。pass を受け取った 9 番は相手ゴールに shoot を実行する。10 番のエージェントは以上の流れを考慮して、7 番への pass を実行する。

行動連鎖を生成することで、エージェントは数手先の状況を考慮してより戦略的価値の高い行動を選択することが可能である。しかし、サイクルごとにそれぞれの状況に対する行動連鎖の評価値を計算するため、サイクルが進むにつれて周囲の状況が変化し、最も評価値の高い行動連鎖が不適切に変化する場合がある。エージェントは通常、pass や dribble などの行動に 1, 2 サイクルの準備を必要とするため、選択された行動連鎖が変化すると、行

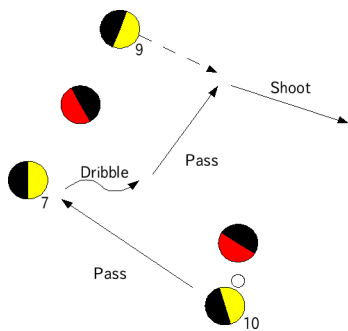


Figure 1: 行動連鎖の例

動の準備がサイクルごとに切り替わってしまう。例えば、100 サイクル目で pass を選択し、pass の準備を始めたのに、101 サイクル目で dribble が選択されると、エージェントは pass の準備を止めて dribble の準備を始める。この様に、選択された行動が変化し、エージェントが無駄なサイクルを過ごしてしまうことを本論文では意思決定の振動と定義する。詳細な定義は次章で述べる。

### 3 意思決定の振動

#### 3.1 振動の定義

本論文では、エージェントの意思決定の振動を以下のように定義する。エージェントがボールを 2 サイクル以上連続して保持しているときに、連続した 2 サイクルで pass などの目標プレイヤーが変更される場合。または、選択した行動の目標ボール座標が、前サイクルで選択した行動のそれと比べてしきい値となる距離以上ずれていた場合とする。ただし、行動のカテゴリが hold から、dribble や pass といった他のカテゴリの行動に変更した場合は、振動ではないものとする。hold とはボールを保持したままその場に止まる行動である。hold は dribble や pass などと違い、最終的には他の行動に変更しなければならないため、hold から他の行動に変更される場合は意思決定の振動から除外する。

振動と判定するための目標位置のずれについては、理論的なずれのしきい値を算出するのは困難である。そのため、ボールの目標座標の距離のずれのしきい値を決定するために以下のような予備実験を行う。

まず、エージェントを 10 回試合させる。ここで、エージェントがボールを 2 サイクル以上連続して保持しているとき、行動の目標座標が連続した 2 サイクルでどの程度の距離ずれるのかを計測する。ただし、目標となるプレイヤーが変更された場合は距離のずれに関わらず意思決定が振動したとみなされ、除外されるものとする。表 1 に計測の結果を示す。ここで、距離は  $x$  軸距離の 2 乗と  $y$  軸距離の 2 乗の和とする。表から、0.6-0.7 の区間までは距離のずれの頻度が小さくなりつつけているが、0.7-0.8 の区間において再び頻度が大きくなっていることが分かる。これを考慮し、本論文では 0.75 を距離のずれをしきい値とする。

#### 3.2 振動の事例

図 2 に意思決定の振動の事例を示す。図 2 の (a) において、左下の赤色の円で囲まれたエージェントは、右上の黄色の円で囲まれたエージェントへの pass を次の行動と決定している。そのために、首や体の向き、kick の準備をしなければならない。しかし、1 サイクル後の (b) では、周囲の状況が変化したため、右下の黒色の円で囲まれたエージェントに pass の対象を変更している。これに伴って、改め

Table 1: 目標位置の距離のずれ

距離 ( $x^2 + y^2$ ) の区間	ずれの頻度
0.0-0.1	518
0.1-0.2	103
0.2-0.3	54
0.3-0.4	25
0.4-0.5	23
0.5-0.6	22
0.6-0.7	5
0.7-0.8	13
0.8-0.9	13
0.9-1.0	34
1.0-1.1	15
1.1-1.2	10
1.2-1.3	10

て首や体の向きの変更, さらにキックの準備をしなければならない. さらに, その次のサイクルである (c) では, パスを実行不可能と判断し, 自分で **dribble** をしようとしている. 従って, 再び首や体の向き, キックの準備を始めなければならない. このように, 状況が少し変化するだけでエージェントの意思決定は大きく変更される. 結果として, エージェントは各行動に対する準備のためサイクルを無駄にしてしまい, 敵に近づかれることにより, ボールを奪われやすくなってしまいう問題が発生している.

### 3.3 振動の統計的調査

意思決定の振動が試合中にどの程度発生するのかを調査するため, エージェントに 20 回試合をさせ, 以下の項目に関して調査を行う.

- 意思決定を行った回数全体
- 2 サイクル以上連続して意思決定を行った回数
- 意思決定が振動した回数
- 意思決定全体に対する振動の割合
- 連続した意思決定に対する振動の割合

ここで, 2 サイクル以上連続して意思決定を行った回数を計測するのは, 意思決定の振動は連続サイクル上で定義されているためである. 結果を表 2 に示す. 表 2 の結果から, 連続した意思決定に対して, 高い頻度で振動が発生していることが分かる. また, これ以降は, 意思決定の振動の割合は, 連続した意思決定における意思決定の振動回数の割合とし, 意思決定を行った回数全体は考慮しないものとする.



(a) 67 サイクル目



(b) 68 サイクル目



(c) 69 サイクル目

Figure 2: 振動の事例

Table 2: 振動の発生回数

意思決定の回数	18119
連続した意思決定の回数	1926
意思決定が振動した回数	1389
全体に対する割合	0.07666
連続したものに対する割合	0.7212

## 4 提案手法

### 4.1 提案手法のアルゴリズム

本論文で対象とするエージェントは, 行動連鎖を生成することで意思決定を行う. 行動連鎖の生成過程では, 候補となる行動に対して評価値を算出し, 最も評価値の高い行動連鎖を採用している. そこで, 提案手法では, 前サイク

ルでの行動連鎖を考慮して評価値に修正を加えることで、意思決定の振動を低減する。以下に、提案手法のアルゴリズムを示す。

- Step 1: 現在の状態をルートノードに入力する。
- Step 2: ノードにおいてエージェントが実行可能な、候補となる行動を生成する。  
(pass, dribble, shoot など)
- Step 3: 生成された行動の評価値  $e$  を算出する。
- Step 4: 前の行動連鎖が記憶されていれば、 $e$  を修正する。そうでなければ、 $e$  は修正しない。
- Step 5: 生成された行動によって達成される状態と  $e$  を子ノードに追加する。
- Step 6: ノード数が設定された最大数に到達していれば  $e$  が最大となる行動連鎖を決定し、Step 7 へ移動する。そうでなければ、 $e$  が最大となるノードを選択して、Step 2 へ戻る。
- Step 7: 前の行動連鎖が記憶されていないか、前の行動連鎖での最終目標地点との距離にしきい値以上のずれがある場合、生成された行動連鎖を前の行動連鎖に上書きして記憶する。
- Step 8: 前の行動連鎖が生成されて一定サイクル経過している場合、削除する。

提案手法では、エージェントが候補となる行動を生成するとき、生成された行動の評価値  $e$  を修正することによって、意思決定の振動を低減する。前の行動連鎖が記憶されている場合、後述する修正式からそれぞれの行動の評価値  $e$  を修正する。生成された行動に対して、前の行動連鎖における同一の深さの行動を使用して、それぞれ修正式を適用する。同一の深さのものがない場合、最後の行動を使用するものとする。記憶された行動連鎖がない場合、評価値の修正は行わない。

最終的な評価値を計算した後、最も高い評価値を持つ行動連鎖が意思決定として選択される。このとき、前の行動連鎖が記憶されていなかった場合、新たに生成された行動連鎖を記憶する。また、新たな行動連鎖の最も深い行動の目標位置が前サイクルの行動連鎖のものとしきい値以上の距離のずれがある場合、新しく生成された行動連鎖を採用し、実行する。これは、しきい値以上の距離のずれがあるのに高い評価値を示している行動連鎖は、状況の変化に対応したよりよい行動連鎖であると考えられるためである。記憶された行動連鎖は、行動連鎖が生成されてから終了するまでにかかると予測されたサイクル数のあいだ持続するものとし、そのサイクルを過ぎた場合、基準となる行動連鎖は削除される。

## 4.2 評価値の修正

評価値  $e$  は以下の式によって修正される。

$$e' = e \times \exp\left(-k \frac{\|p - q\|^2}{(1 + (n - m))^2}\right) \quad (1)$$

ここで、 $e'$  は修正後の評価値である。また、 $n$  と  $m$  はそれぞれ、現在の試合時刻と前の行動連鎖を選択したときの試合時刻である。 $p$  は候補行動連鎖における行動のボールの目標位置、 $q$  は現在実行中の行動連鎖における、同じ深さの行動のボールの目標位置である。さらに、 $k$  は時間と距離の影響を変化させるための非負のパラメータである。

この式は前の行動での目標座標を平均、時間の経過を分散としたガウス分布と類似している。これは目標座標との距離のずれが大きくなるほど評価値が小さくなり、行動が選択され難くなることを意味する。また、現在の時刻と前の行動の時刻が離れるほど、ガウス分布における分散が大きくなり、分布の山がなだらかになる。このことにより、試合時刻が経過するほど修正式の影響は小さくなり、ボールの目標位置が前の行動と大きく異なった行動も選択されるようになる。なぜなら、試合時刻が経過することで状況の変化が大きくなり、距離のずれが大きくても、より高い評価を持つ行動連鎖が採用される必要が出てくるからである。

## 5 数値実験

### 5.1 実験設定

使用するエージェントは、RoboCup2011 の決勝戦で使用された HELIOS2011 に提案手法を組み込んだものとする。行動連鎖の最大の深さを 3 とし、生成される行動数を 500 とする。数値実験では、意思決定の振動が低減されるかどうかを調査するため、提案手法のアルゴリズムを組み込んだチームと、提案手法のアルゴリズムを組み込んでいないチームを、それぞれ agent2d と対戦させる。agent2d[9] は、フリーで公開されているサンプルのチームである。HELIOS2011 と同じように意思決定に行動連鎖を用いており、いくつかの世界大会登録チームによって開発のベースとして利用されている。数値実験では、また、修正式のパラメータ  $k$  の値を変化させることで、時間と距離に対する重みが意思決定の低減にどのように影響するかも調査する。 $k$  の値を 0.1, 0.5, 1, 1.5, 3, 5, 10, 50, 100 と変化させて、20 回ずつ試合を行い、エージェントの意思決定の振動の回数を計測する。

また、同様の実験を agent2d でも行う。行動連鎖の深さは 1 とし、生成される行動数を 500 とする。提案手法を組み込んだ agent2d とそうでない agent2d を対戦させ、意思決定の振動が低減されるかどうかを確認する。修正式のパラメータ  $k$  の値は 0.5, 1, 1.5, 3.0, 5, 10 と変化させ、10 回ずつ試合を行い、エージェントの意思決定の振動の回数を計測する。

## 5.2 実験結果

HELIOS2011の結果を表3, agent2dの結果を表4に示す. 表3と表4から, 両方のチームにおいて, 提案手法を組み込んだ場合は, 提案手法を組み込まなかった場合に比べて意思決定の振動の割合が大幅に減少していることが分かった. また, 表3では, パラメータ  $k$  の値が 1.0 から 5.0 の範囲にあるとき, 振動の割合が最も低くなっていた. ここで,  $k$  の値が大きくなるほど距離に対する重みが大きくなるので, 距離と時間に対する重みが同じか, 距離に対する重みがやや大きい状態が最もよい結果を出すことが分かった. 表4では,  $k$  の値が大きくなるほど振動の割合が減少していた. しかし,  $k$  の値が大きくなるにつれて減少の割合が減っているため, ある程度以上は大きくしても意味がないと考えられる.

## 5.3 性能評価

次に, 意思決定の振動を低減した場合におけるチームの性能評価を行う. 使用するチームは HELIOS2011 とし, その他の条件は数値実験と同様である. 性能評価は人間のサッカーにおけるデータ分析 [8] でも取り入れられている, ボール保持率などに注目して行う. 対戦させるチームは agent2d, AUA2D, Edinferno, HfutEngine, Photon,

RMAS, WrightEagle の 7 チームとする. チームに提案手法を組み込んだ場合と組み込まなかった場合で, それぞれ 20 回試合をさせて性能を評価する.  $k$  の値は 1.0 とする. 結果を表 5 から 9 に示す. ここで, 太字になっている項目は, 提案手法がある場合とない場合で, 優位水準が両側 5% の  $t$  検定で, 平均に有意差があるとみなされたものである. また, 表 5 において, 振動の割合が 1.0 を超えるものがあるが, これは連続した意思決定に対して 2 回以上振動が発生する可能性があるためである. 表 5 では, 提案手法を組み込むことによって, 全てのチームに対して, 意思決定の振動回数が減っている. このことから, 提案手法は相手チームによらず効果があることが分かった. また, 表 6 では, いくつかのチームで平均得点が低下していたが, 表 7 では, いくつかのチームに対してボール支配率が向上していた. これは意思決定の振動が低減することでボールを保持しつづける行動が選択されやすくなり, ボール支配率は上昇しても得点に結びつかなくなってしまったためであると考えられる. また, 表 8 と 9 の結果から, パスの成功回数よりもドリブルの成功回数が向上しており, 提案手法はパスよりもドリブルに対して強く影響していることが分かった.

Table 3: HELIOS2011 の意思決定の振動の回数

$k$	意思決定回数	振動	割合
手法なし	1926	1389	0.7212
0.1	2677	791	0.2955
0.5	3130	709	0.2265
1	3634	594	0.1635
1.5	3661	741	0.2024
3	4047	619	0.1530
5	4085	643	0.1574
10	4414	966	0.2188
50	5264	1012	0.1922
100	4676	963	0.2059

Table 4: agent2d の意思決定の振動の回数

$k$	意思決定回数	振動	割合
手法なし	490	347	0.7082
0.5	1411	522	0.3700
1	1631	539	0.3305
1.5	1900	529	0.2784
3	2021	484	0.2395
5	2055	483	0.2350
10	2085	517	0.2480

Table 5: 意思決定の振動の割合

	提案手法なし	提案手法あり
agent2d	0.7212	<b>0.1635</b>
AUA	1.064	<b>0.4340</b>
Edin	1.051	<b>0.4221</b>
Hfut	1.255	<b>0.4983</b>
Photon	0.8118	<b>0.3147</b>
RMAS	1.082	<b>0.4284</b>
Wright	1.257	<b>0.4721</b>

Table 6: 得失点

	提案手法なし		提案手法あり	
	得点	失点	得点	失点
agent2d	3.1	0.6	3.3	0.5
AUA	<b>15.2</b>	0	11.6	0
Edin	<b>19.85</b>	0	17.15	0
Hfut	<b>19.25</b>	0.05	15.95	0.15
Photon	18.65	0	18.05	0
RMAS	<b>19.7</b>	0	16.7	0
Wright	4.8	0.8	4	1

たないエージェントのインターセプト動作などに振動低減を拡張することが、今後の課題である。

Table 7: ボール支配率

	提案手法なし	提案手法あり
agent2d	0.6578	<b>0.6965</b>
AUA	0.4813	0.4996
Edin	0.6429	<b>0.6840</b>
Hfut	0.5909	0.6014
Photon	0.5454	<b>0.6037</b>
RMAS	0.7720	0.7761
Wright	0.4381	0.4272

Table 8: パスの成功回数

	提案手法なし	提案手法あり
agent2d	102.8	99.95
AUA	113.75	<b>127.75</b>
Edin	119.05	121.05
Hfut	139.35	141.65
Photon	113.35	105.6
RMAS	120.75	122.5
Wright	112.25	104.95

Table 9: ドリブルの成功回数

	提案手法なし	提案手法あり
agent2d	434.05	<b>482.8</b>
AUA	145.5	<b>172.4</b>
Edin	166.05	<b>221.9</b>
Hfut	110.85	<b>165.5</b>
Photon	170.7	<b>314.6</b>
RMAS	165.9	<b>232.85</b>
Wright	93.55	<b>140.6</b>

## 6 おわりに

本論文では、RoboCup サッカーシミュレーション 2D リーグにおいて、プレイヤーエージェントの意思決定の振動を低減させる手法を提案した。提案手法によって、エージェントが意思決定を行うとき、前に選択された行動を考慮して意思決定の評価値を修正することで、前に選択された行動が選択されやすくなり、意思決定の振動の発生が低減された。また、数値実験によって、評価値を修正することで意思決定の振動が低減されていることを確認した。さらに、性能評価により提案手法が適用されたことによるチーム全体への影響を調査し、いくつかのチームに対して平均得点数は減ったものの、ボール支配率が上昇したことを確認した。本論文ではエージェントがボールを持つときの意思決定のみに着目していたが、ボールを持

## 参考文献

- [1] 山田誠二, “ソフトウェアエージェントにおけるプランニング”, 人工知能学会誌, page 623-628, 2001.
- [2] 神尾正太郎, 伊庭齊志, “マルチエージェント協調作業のためのランダムサンプリングを用いた経路プランニングアルゴリズム”, 電子情報通信学会論文誌, D, 情報・システム, pp. 250-260, 2006.
- [3] E. Onieva, L. Cardamone, D. Loiacono and P.L. Lanzi, “Overtaking Opponents with Blocking Strategies Using Fuzzy Logic,” In *IEEE Conference on Computational Intelligence and Games*, pp. 123-130, 2010.
- [4] L.L. Delooye and W.R. Viner, “Fuzzy Q-Learning in a Nondeterministic Environment: Developing an Intelligent Ms.Pac-Man Agent,” In *IEEE Conference on Computational Intelligence and Games*, pp. 162-169, 2009.
- [5] J. Ma and S. Cameron, “Combining Policy Search with Planning in Multi-agent Cooperation,” In *RoboCup 2008: Robot Soccer World Cup XII*, pp. 532-543, 2008.
- [6] S. Gspandl, A. Hechenblaickner, M. Reip, G. Steinbauer, M. Wolfram and C. Zehentner, “The Ontology Lifecycle in RoboCup: Population From Text and Execution,” In *RoboCup Symposium 2011*, pp. 313-324, 2011.
- [7] 秋山英久, “アクション連鎖探索によるオンライン戦術プランニング”, 第 33 回人工知能学会 AI チャレンジ研究会, pp.23-28, 2011.
- [8] 森本美行, 本田にパスの 36%を集中せよ ザック JAPAN vs. 岡田ジャパンのデータ解析, 文春新書, 2011.
- [9] agent2d <http://rctools.sourceforge.jp/pukiwiki/index.php?agent2d>



# ロボットモーションプランニングの自動化に向けての ロボットモーション実行基盤の開発

## Development of Robot Motion Performance Platform for Automatic Generation of Robot Motion Planning

加藤央昌<sup>1\*</sup> 石原裕平<sup>1</sup> 清水優<sup>1</sup> 橋本学<sup>1</sup>

Hiroaki Kato<sup>1</sup> Yuhei Ishihara<sup>1</sup> Masaru Shimizu<sup>1</sup> Manabu Hashimoto<sup>1</sup>

<sup>1</sup> 中京大学大学院情報科学研究科

<sup>1</sup> Graduate School of Computer and Cognitive Sciences, Chukyo University

**Abstract:** In this study, Robot Motion Performance Platform was developed for the automation of Robot Motion Planning using memory-based techniques. This can be used for robots with various arm and leg forms. The developed Robot Motion Performance Platform is used for the motion performance by name-base to a motion command. In this paper, the realization of motion in a robot with two or more different types of control systems is described. In addition, the way in which motion mixing uses low motion data to realize more motions is also discussed.

## 1 はじめに

近年、ロボットの認識技術が向上したため、ロボットの動作計画による動作指令作成 [1]-[5] が詳細に設定されるようになった。しかし、現状はユーザからの要求に対して、コーディングによって設定されている。動作計画の研究は古くから行われており、マニピュレータの障害物回避や、ロボットモーションの生成などがあり、様々な手法で動作指令が算出されている [6]-[25]。このような動作計画が将来的に自動化され、自律ロボットが主流になることを踏まえると、動作計画によって作成された動作指令に対して、命令を解釈し実行するロボット制御基盤技術が必要である。また、ロボットの制御基盤は、様々な脚や腕を持つロボットにおいてモーションが実行可能であり、さらにより多くのモーションが実行可能である必要がある。

そこで、今回筆者らは、様々な脚や腕を持つロボットのモーション実行基盤を開発した。ロボットモーション実行基盤を開発することによって、様々なロボットにおいてモーションの実行が可能であり、さらにより多くのモーションが実行可能である。

前述したが、マニピュレータの障害物回避や、ロボットモーションの生成などにおいて、様々な手法で動作指令が算出されている [6]-[25]。そのため、動作指令の算出方法の数だけロボットの制御方法が存在することになる。この場合、提案された一つのロボットに対し

て一つの制御方法でのみロボットを制御することが可能になる。そのため、制御方法が多数存在することになり、制御方法が煩雑になるという問題がある。そこで、本研究では、動作指令の算出結果の形式を角度情報としたロボットの制御基盤開発に着目した。

ここで、本研究の最終目的について述べる。本研究の最終目的は、ロボットモーションプランニングの自動化に向けて、モーションネーミングルール策定による（作業内容に応じた）機能、特にモーションの発現を可能とするロボットモーション実行基盤の開発である。モーションネーミングルールとは、基本モーションに名前を付け、名前によってモーションを管理する方法である。モーションネーミングルールでは、名前の組み合わせによってモーションの実行やモーションの合成、モーションの検索、名前付けによるモーションの記録等を行うことが可能になる。モーションを名前で管理することによって、ロボットは人間の要求に対して柔軟に対応することが可能になる。例えば、ロボットの知らないモーションの実行が必要な場合、ロボット自身が知っているモーションのうち、近いモーションを検索し実行することで対応する。もしくは、近いモーション同士を合成し、要求されたモーションを作り出すことで対応することが可能になる。例をもう一つ挙げる。ロボットに何か道具を持たせ作業をさせたい場合を考える。道具を使って作業を行いたい場合、道具にあったモーションが必要になる。ロボット自身が実行できるモーションを把握していれば、どの道具が使えるかロボット側から人間に伝えることが出来る。

\*連絡先： 中京大学大学院情報科学研究科  
470-0393 愛知県豊田市貝津町床立 101  
E-mail: hiro1125univ@gmail.com

また、ロボット自身が使える道具を選択し作業を実行することが出来る。モーションネーミングルールによるロボットモーション実行基盤が実現された場合、ロボットの危険地域での自律作業が期待できる。なぜならば、ロボット作業におけるロボットの行動選択や、必要な道具の選択などにおいて、人間が介入する必要性が低くなるからである。また、ロボット開発は単純化されると考えられる。なぜなら、ロボットモーションの生成は、ゼロからではなく近いモーションから生成することが可能になるからである。

本稿では、モーションネーミングルール策定の前段階として開発した、ロボットモーション実行基盤について述べる。これは、動作計画によって作成された動作指令（角度情報）に対して、様々な腕や脚を持つロボットにおいてモーションの実行を可能にするものである。開発したロボットモーション実行基盤は、複数種類の制御系の異なるロボットにおいてモーションの実行が可能である。さらに、少ないモーションデータでより多くのモーションを実行可能にするモーションの合成が可能である。

以下、第2章では関連研究と題して、基盤技術として類似しているRTミドルウェアについて紹介し、また本研究との相違点について述べる。続く第3章ではモーションネーミングルール策定の前段階として開発した、様々な脚や腕を持つロボットのモーション実行基盤について述べる。第4章では、開発したロボットモーション実行基盤による複数種類の制御系の異なるロボットにおいてのモーション実行の実験結果、および、少ないモーションデータでより多くのモーションを実行可能にするモーションの合成についての実験結果について述べる。第5章では本論文のまとめを述べる。

## 2 関連研究

本章では、基盤技術として類似しているRTミドルウェアについて紹介する。RTミドルウェアは、技術の共有と再利用を目的としたロボットシステム開発用のソフトウェア基盤技術のことである。RTミドルウェアは、ロボットの機能部品（センサ、サーボ、モータ等）をモジュール化し、そのモジュール化された機能部品を組み合わせることにより、ユーザの幅広いニーズに合わせた新しいロボットシステムを、設計者が容易に効率よく構築することを可能にするものである[26]-[29]。この技術を用いて実現されたシステムの事例として、移動ロボットシステム、コミュニケーションシステム、マニピュレーションシステム等がある[30]-[32]。また、RTミドルウェアによる技術の共有と再利用を目的としたプロジェクトは、RTミドルウェアプロジェクトと呼ばれており、様々な成果が上げられている[33]-[36]。

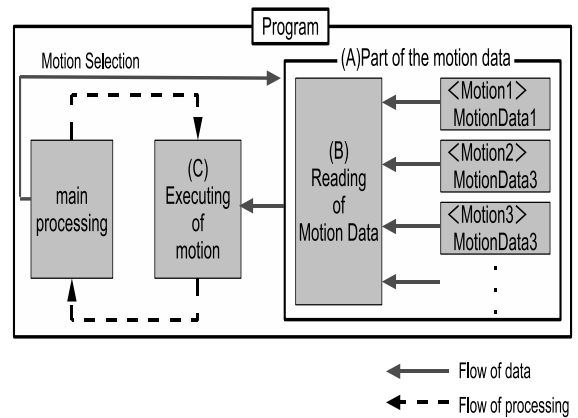


図 1: 開発したロボットモーション実行基盤の構造

本研究は、ロボットモーションプランニングの自動化に向けて、モーションネーミングルールという新しい概念をロボット工学に取り入れ、ネームベースによるモーションの発現を目指したものであるため、RTミドルウェアとは技術的に異なるものである。

## 3 ロボットモーション実行基盤

本研究で開発したロボットモーションの実行基盤は、メモリ・ベースト手法[37]を用いており、図1に示すように、モーションデータ部分とモーションを実行する部分が分離した構造を持つサーボモータ制御プラットフォームである。メモリ・ベースト手法とは、予め用意したモーションデータを場合によって選定し、モーションを実行する方法であり、本研究では、基本的なモーションパターンを複数保持し、それらを合成することによって、より多くのモーションを実現する。また、モーションデータ部分とモーションを実行する部分をつなぐインターフェース部分で、角度情報の通信方式が異なるサーボモータを用いたロボットにおいてもモーション実行が可能となるように、通信方式の情報を追加（コーディング）することによって対応可能としている。これにより、複数種類の制御系の異なるロボットにおいてのロボットモーションの実行が可能である。さらに、開発したロボットモーション実行基盤は、モーションデータ部分とモーション実行部分が分離した構造をしている。そのため、どのモーションデータに対しても一つのモーション実行部分のみでモーションの実行が可能であるため、メモリ効率の良いシステムである。

ロボットモーション実行基盤は、図1に示すように、(A)モーションデータ部分、(B)モーション読み込み部分、(C)モーション実行部分によって構成されている。ロボットモーション実行基盤は、各関節において

モーションのキーとなる点の角度情報を保持している。キーとなる点が始点角度と終点角度の二点の場合、図 2 (a) と図 2 (b) の角度情報を保持していることになる。モーションの実現方法は、点と点の角度の差分を移動時間のサイクル数によって細分化し、細分化された角度 (図 2 (c)) を順次実行することでモーションを実現する。また、モーションの合成は、複数のモーションデータが保持する各関節の角度情報を足し合わせ、その角度情報を用いてモーションを実行する。この際、使用する関節に対してのモーションデータの数にカウントし、足し合わせた角度をカウント数で割ったものを角度情報として使用する。

図 1 の (A), (B), (C) の機能について説明する。(A) はモーションデータ部であり、基本的なモーションパターンのキーとなる始点角度情報や終点角度情報、移動時間のサイクル数を保持する。(B) はモーションデータの読み込み部分であり、(C) はモーション実行部である。モーションデータの読み込み部分 (B) は、モーションデータをモーション実行部 (C) に受け渡すために、最終的なモーションデータへと加工する。モーション実行部 (C) は、モーションデータを図 2 (c) のようにサイクル数分に細分化し、分割された角度情報をもとに滑らかなモーションを実行する。また、モーション実行部 (C) は、通信方式の異なるサーボモータが混在している場合においてもモーションの実行を可能とする拡張可能なインターフェース部分を持つ。モーション実行部 (C) は、図 3 に示すように、角度演算部分と角度演算結果をアクチュエータに出力するインターフェース部分によって構成される。このインターフェース部分は角度制御情報の出力方法について情報を追加 (コーディング) することで拡張することが可能である。現状は、現在主流である PWM 方式のサーボモータとシリアルサーボモータに対応している。最後に、開発したロボットモーション実行基盤は、図 1 に示すように、モーションデータ部 (A) とモーション実行部 (C) が分離した構造になっているため、どのモーションデータに対しても、モーション実行部 (C) のみでモーション実行を行うことが可能である。

## 4 実験

開発したロボットモーション実行基盤が、以下の機能を実現できることを確認する。

1. 複数種類のロボットを制御できること
2. モーションの合成が可能であること

実験に用いたロボットは、2 脚ロボットとアームロボットである。2 脚ロボットは、PWM 方式のサーボモータ 12 個によって構成されている。2 脚ロボットの構成

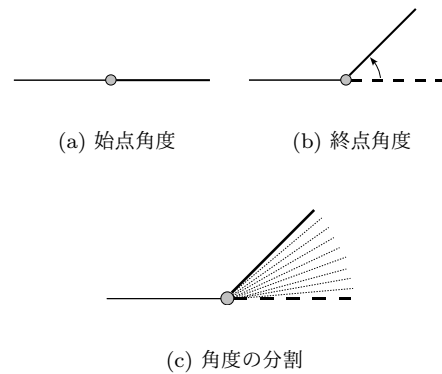


図 2: モーションの実行

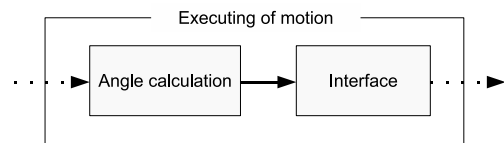


図 3: モーション実行部分の構造

を図 4 に示す。アームロボットは、アーム部分とハンド部分で通信方式の異なるサーボモータを使用している。アーム部分は PWM 方式のサーボモータ 4 個で構成されており、ハンド部分はシリアルサーボモータ 2 個で構成されている。アームロボットの構成を図 5 に示す。制御 CPU はいずれも株式会社ルネサステクノロジ製の SH7145F である。

### 4.1 複数種類のロボットの制御

2 脚ロボットおよびアームロボットを用いて制御系の異なるロボットにおいてモーションが実行可能であることを示す。図 6 は、2 脚ロボットを用いて片脚屈伸モーションを実行した結果である。(a) のスタート位置から (b) (c) (d) (e) へと脚を曲げていき、(e) から (f) (g) (h) (i) へと脚を伸ばしていく様子が見える。続いて、図 7 は、アームロボットを用いてモーションを実行した結果である。(a) のスタート位置から (b) (c) へとアームを持ち上げながらハンドを閉じている様子が見える。続く (d) (e) では、アームを下ろしながらハンドを閉じている様子が見える。さらに、続く (f) (g) では、アームを持ち上げながらハンドを開いていき、(h) (i) ではアームを下ろしながらハンドを開いている様子が見える。以上のことから、PWM 方式のサーボモータを使用した場合、シリアルサーボモータを使用した場合の両方においてロボットのモーションが実行可能であることを確認できた。そのため、PWM

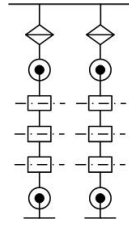


図 4: 2脚ロボットの構成

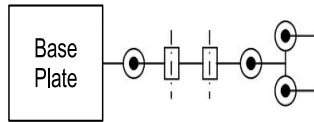


図 5: ロボットアームの構成

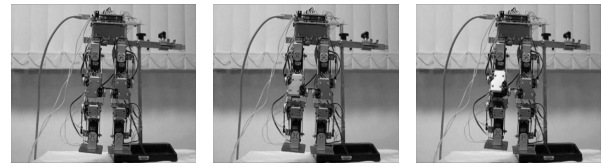
方式のサーボモータおよびシリアルサーボモータで構成された複数種類のロボットにおいてモーションの実行が可能であるといえる。

## 4.2 モーションの合成

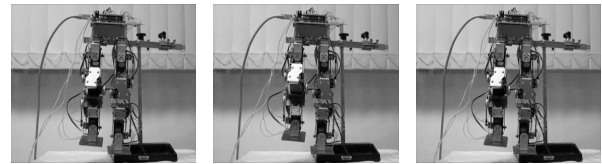
図 6 と図 8 のモーションを用いてモーションの合成が可能であることを示す。図 9 は、図 6 に示す右脚屈伸モーションと、図 8 に示す右脚股関節の開閉モーションを合成した結果である。(a) のスタート位置から (b) (c) (d) (e) へと右脚を外側へ広げながら右脚を曲げていき、(e) から (f) (g) (h) (i) へと右脚を内側に戻しながら右脚を伸ばしていることから、二つのモーションが合成され実行されていることがわかる。以上のことから、少ないモーションデータでより多くのモーションが実行可能となるモーションの合成が可能であることを確認した。

## 5 おわりに

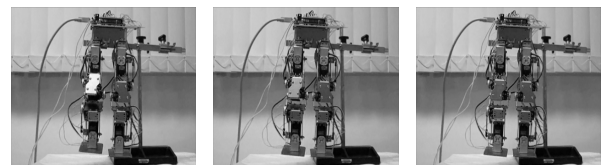
本稿では、モーションプランニングの自動化に向けて必要となるロボットモーション実行基盤の開発結果を述べた。開発したロボットモーションの実行基盤は、実機に搭載するアクチュエータの角度通信方式が異なる場合においてもモーションを実行することが可能である。具体的には、モーション実行部分のインターフェース部分において現在の主流である PWM 方式のサーボモータおよびシリアルサーボモータに対応しており、このインターフェース部分に角度情報の出力方法について情報を追加（コーディング）することで、他のサーボモータで構成されたロボットにおいてもモーションを実行することが可能である。そのため、複数種類の



(a) 1st frame (b) 2nd frame (c) 3rd frame



(d) 4th frame (e) 5th frame (f) 6th frame

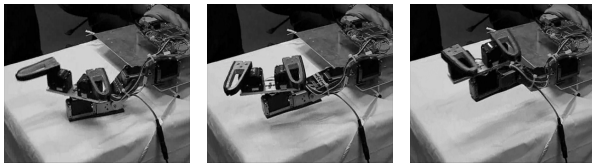


(g) 7th frame (h) 8th frame (i) 9th frame

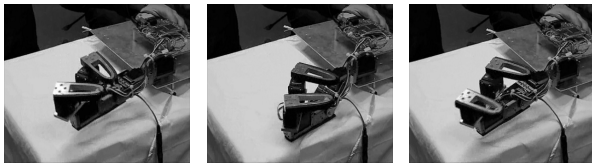
図 6: 屈伸モーションの実行結果

ロボットにおいてモーションの実行が可能である。また、少ないモーションデータで、より多くのモーションが実行可能となるモーションの合成が可能である。

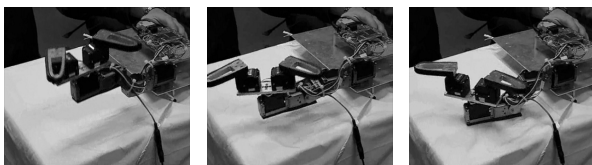
今後は、開発したロボットモーション実行基盤をもとに、モーションネーミングルール策定によるモーション発現の実現に向けて、ロボットモーション実行基盤の改良を行う。具体的には、現状のロボットモーション実行基盤は、センサフィードバックを考慮していない。そこで、センサフィードバックを効率よく行える仕組みを考案し実装する。そして、現在マイコン内で処理している本システムを、PC-マイコン間通信によるシステムへと拡張し、汎用性の高いシステムを実現する。また、本研究で目指しているロボットモーション実行基盤は、実世界のみではない。そのため、仮想世界でのロボットモーション実行基盤を実現する。さらに、実世界と仮想世界において、ロボットモーションプログラミングのコストを軽減した、シームレスなロボットモーション実行基盤を実現する。最終的には、モーションネーミングルールの策定を行い、本システムに適用することで、作業内容に応じた機能、特にモーションの発現を可能とするロボットモーション実行基盤を実現する。



(a) 1st frame (b) 2nd frame (c) 3rd frame

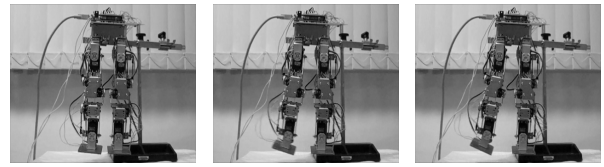


(d) 4th frame (e) 5th frame (f) 6th frame

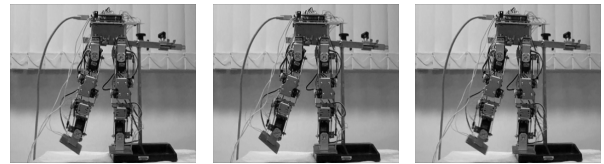


(g) 7th frame (h) 8th frame (i) 9th frame

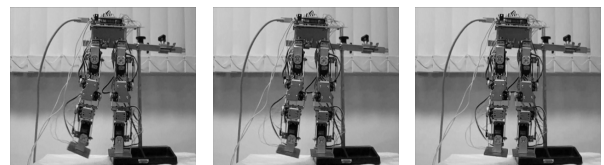
図 7: ロボットアームのモーション実行結果



(a) 1st frame (b) 2nd frame (c) 3rd frame



(d) 4th frame (e) 5th frame (f) 6th frame



(g) 7th frame (h) 8th frame (i) 9th frame

図 8: 股関節の開閉を行うモーションデータ

## 参考文献

- [1] 大須 賀公一: ロボットの制御, 日本ロボット学会誌, Vol. 16, No. 7, pp. 882-885 (1998)
- [2] 米田 完: 脚移動, 日本ロボット学会誌, Vol. 16, No. 7, pp. 897-901 (1998)
- [3] 梶田 秀司: ゼロモーメントポイント (ZMP) と歩行制御, 日本ロボット学会誌, Vol. 20, No. 3, pp. 229-232 (2002)
- [4] 比留川 博久, 加賀美 聡: ヒューマノイドの知能-個としての知能-, 日本ロボット学会誌, Vol. 20, No. 5, pp.478-481 (2002)
- [5] 社会法人 日本機械学会: 生物型システムのダイナミクスと制御, 養賢堂 (2002)
- [6] 比留川 博之, 北村 新三: 安全第一アルゴリズムとポテンシャル関数に基づくマニピュレータの障害物回避法, 日本ロボット学会誌, Vol. 5, No. 3, pp. 171-179 (1987)
- [7] 比留川 博之, 井上 雄紀, 北村 新三, 榊原 聡, 人見 信, 角谷 和俊: 幾何形状のモデリングと処理に基づく自律的マニピュレーションシステム Kraft の開発, 日本ロボット学会誌, Vol. 7, No.5, pp.464-474 (1989)
- [8] 寺崎 肇, 長谷川 勉, 高橋 裕信: 平行 2 指ハンドによる多面体物体の運搬作業のための把握動作計画, 日本ロボット学会誌, Vol. 10, No.2, pp. 273-282 (1992)
- [9] 寺崎 肇, 長谷川 勉: 平行 2 指ハンドによる滑らし操作を利用した知的物体操作のための動作計画, 日本ロボット学会誌, Vol. 12, No. 7, pp. 1056-1065 (1994)
- [10] 寺田 英嗣, 輻形 和幸: 風呂敷包み作業用マルチロボットシステムの運動計画法, 精密工学会誌, Vol. 76, No. 5, pp. 546-551 (2010)
- [11] 佐藤 圭祐: 極小点のないポテンシャル場を用いたロボットの動作計画, 日本ロボット学会誌, Vol. 11, No. 5, pp. 702-709 (1993)
- [12] 池浦 良淳, 中里 央, 猪岡 光: 人間の舞踊動作に基づくダンシングロボットの動作計画, 日本ロボット学会誌, Vol. 15, No. 6, pp. 927-933 (1997)
- [13] 尾崎 弘明, 丘 華: 移動障害回避のためのマニピュレータ軌道の局所修正法, 計測自動制御学会論文集, Vol. 29, No. 1, pp. 122-124 (1993)
- [14] 毛利 彰, 山本 元司, 田代龍次: 複数台マニピュレータの協調動作軌道計画, 計測自動制御学会論文集, Vol. 29, No. 7, pp. 819-825 (1993)
- [15] 佐竹 利文, 林 朗弘, 鬼塚 昭一, 鈴木裕: セル集団の自己組織化を適用したロボットアームの連続的な姿勢変化の生成, 精密工学会誌, Vol. 62, No. 10, pp. 1415-1419 (1996)
- [16] 田中 良幸, 辻 敏夫, 金子 真: 人間の上肢運動を模擬したロボットの軌道生成法, 日本ロボット学会誌, Vol. 18, No. 5, pp. 699-705 (2000)

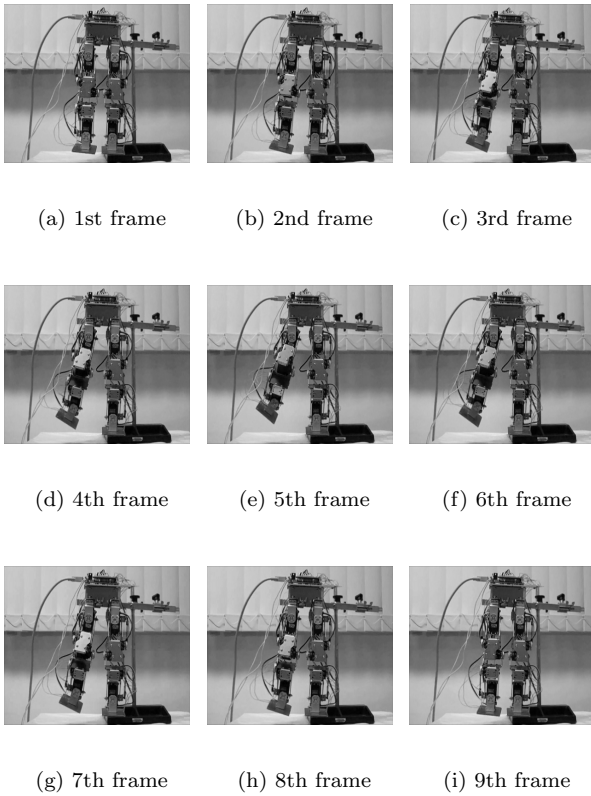


図 9: 図 6 と図 8 のモーションを合成した結果

- [17] 堀内 英一, 谷 和男: 動作計画における幾何学的不確かさに対する確率的アプローチ, 日本ロボット学会誌, Vol. 10, No. 4, pp. 475-480 (1992)
- [18] 新井 民夫, 太田 順: 仮想的なインピーダンスを用いた複数移動ロボット系の動作計画, 日本ロボット学会誌, Vol. 11, No. 7, pp. 1039-1046 (1993)
- [19] 太田 順, 新井 民夫: 群秩序生成可能な追従戦略を用いた複数移動ロボット系の動作計画, 日本ロボット学会誌, Vol. 12, No. 4, pp. 603-608 (1994)
- [20] 太田 順, 新井 民夫, 倉林 大輔: 作業の性質を考慮したロボット群の動作計画, 日本ロボット学会誌, Vol. 13, No. 4, pp. 517-524 (1995)
- [21] 小方 博之, 新井 民夫, 太田 順: 時変環境でユーザ仕様を考慮した移動ロボットの軌道計画法, 日本ロボット学会誌, Vol. 12, No. 6, pp. 905-910 (1994)
- [22] 徳永 紘典, 田中 純一, 加藤 清敬: スプライン曲面を適用したロボットの動作計画, 精密工学会学会誌, Vol. 71, No. 8, pp. 985-989 (2005)
- [23] 内田 雅文, 井出 英人, 横山 修一: 動的環境における障害物領域予測法, 電気学会論文誌 C, Vol. 117-C, No. 2, pp. 198-204 (1997)
- [24] 安藤 昌和, 西 竜志, 小西 正躬, 今井 純: 複数大移動ロボットの搬送経路計画問題に対する自律分散型最適化法, 計測自動制御学会論文集, Vol. 39, No. 8, pp. 759-766 (2003)
- [25] 登尾 啓史, 浪花 智英, 有本 卓: クワッドツリーを利用した移動ロボットの高速度動作生成アルゴリズム, 日本ロボット学会誌, Vol. 7, No. 5, pp. 414-425 (1989)
- [26] 北垣 高成, 末廣 尚士, 神徳 徹雄, 平井 成興, 谷江 和雄: RT ミドルウェア技術基盤の研究開発について-ロボット機能発現のために必要な要素技術開発-, 第 8 回ロボティクスシンポジウム予稿集, pp. 487-492 (2003)
- [27] 神徳 徹雄, 北垣 高成, 安藤 慶昭, 尹 祐根, 末廣 尚士: RT ミドルウェアのソフトウェア開発支援機能の検討, 第 9 回ロボティクスシンポジウム予稿集, pp. 282-287 (2004)
- [28] 安藤 慶昭, 末廣 尚士, 北垣 高成, 神徳 徹雄, 尹 祐根: RT 要素のモジュール化および RT コンポーネントの実装, 第 9 回ロボティクスシンポジウム予稿集, pp. 288-293 (2004)
- [29] 菅野 重樹: RT ビジネスの戦略, 日本ロボット学会誌, Vol. 24, No. 3, pp. 278-283 (2006)
- [30] 松本 吉央: RT ミドルウェアによるロボットアーキテクチャ 移動ロボットシステム, 日本ロボット学会誌, Vol. 28, No. 5, pp. 564-565 (2010)
- [31] 松坂 要佐: RT ミドルウェアによるロボットアーキテクチャ コミュニケーションシステム, 日本ロボット学会誌, Vol. 28, No. 5, pp. 566-567 (2010)
- [32] 末廣 尚士: RT ミドルウェアによるロボットアーキテクチャ マニピュレーションシステム, 日本ロボット学会誌, Vol. 28, No. 5, pp. 568-569 (2010)
- [33] 安藤 慶昭, 中岡 慎一郎, 神徳 徹雄: 分散コンポーネント型ロボットシミュレータ・アーキテクチャ-RT コンポーネントを用いた実機と可換な制御ソフトウェア開発機能-, 日本ロボット学会誌, Vol. 26, No. 5, pp. 407-410 (2008)
- [34] 佐藤 知正, 岡野 克弥: RT ミドルウェアと知能モジュール構築プロジェクト, 日本ロボット学会誌, Vol. 28, No. 5, pp. 546-549 (2010)
- [35] 原 功: RT ミドルウェアによるロボットシステム構築, 日本ロボット学会誌, Vol. 28, No. 5, pp. 562-563 (2010)
- [36] 金広 文男: RT ミドルウェアと OpneHRP3 によるロボットシミュレーション, 日本ロボット学会誌, Vol. 28, No. 5, pp. 556-561 (2010)
- [37] 宮腰 清一: メモリ・ベースト運動制御による 2 足歩行の制御, 日本ロボット学会誌, Vol. 24, No. 5, pp. 623-631 (2006)

# Differentiation within Coordination in Acquisition of Skilled Throwing

Jihoon Park<sup>1</sup>, Yuji Kawai<sup>1</sup>, Takato Horii<sup>1</sup>, Yuji Oshima<sup>1</sup>, Kazuaki Tanaka<sup>1,2</sup>,  
Hiroyuki Mori<sup>1</sup>, Yukie Nagai<sup>1</sup>, Takashi Takuma<sup>3</sup>, and Minoru Asada<sup>1</sup>

<sup>1</sup> Dept. of Adaptive Machine Systems, Graduate School of Engineering,  
Osaka University, Osaka, Japan

<sup>2</sup> CREST, Japan Science and Technology Agency

<sup>3</sup> Dept. of Electrical and Electronic Systems Engineering,  
Osaka Institute of Technology, Osaka, Japan  
robocup@er.ams.eng.osaka-u.ac.jp

## Abstract

Humanoid robots have a large number of degrees of freedom (DoFs). Therefore, motor learning algorithms that explore the optimal parameters of behaviors of these robots are laborious. In contrast, it has been suggested that humans can solve this large-scale problem by coordinating most body parts strongly in the early stage of learning, and then differentiating the movements to optimize a behavior for a task. We propose that heuristic exploration through differentiation within coordination of the DoFs accelerates motor learning of humanoid robots. In this paper, we demonstrate that the process decreases iteration of trials for a throwing skill in soccer. At first, all motors related to the skill are coordinated. Thus the robot explores the optimal timing for releasing a ball in a one-dimensional search space. The DoFs are released gradually, which allows to search for the best timing of the start of each joint's movement. The experiments using a real robot showed that the exploration method was very fast and practical because the solution in a low-dimensional subspace is a good approximation of the optimum.

## 1 Introduction

Skilled behavior of a humanoid robot is important in the robot soccer domain. Soccer skills such as throwing, kicking, and biped locomotion require coordination of the whole body with a large number of degrees of freedom (DoFs). Thus, we must solve a large-scale problem for designing a skilled behavior of a humanoid robot.

There exists much literature on the heuristic exploration approach to solve the problem. In this approach, a robot autonomously optimizes parameters of skills through practice. It has been reported that evolutionary computation (e.g., [1, 2]) and particle swarm optimization (e.g., [3, 4]) enable the robot to acquire faster gait.

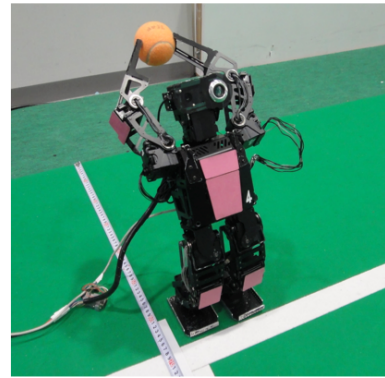


Figure 1: Throwing for exploration of optimal parameters.

Main optimization parameters in most of these studies have been trajectories of limbs or Central Pattern Generators. However, the number of iterations including the evaluation of performance was very large because the search space was vast. Also, degradation and malfunction are problems with using a real robot for a long time. Therefore, optimization methods that require many trials are not useful.

In contrast, Kohl and his colleagues [5–7] have demonstrated that Hill Climbing and Policy Gradient algorithms allow to optimize the parameters for quadruped locomotion and kicking a ball. These algorithms generally require less iterations than evolutionary computation and particle swarm optimization. However, the intrinsic complexity of a humanoid robot's body still causes a large number of iterations.

We take a hint from the progression of skills in humans during behavior acquisition in high-dimensional motor space. Bernstein [8] (see also [9, 10]) suggested freezing and freeing of DoFs in skill acquisition. In the early stage of learning of a motor skill, some DoFs are reduced (frozen). These DoFs are then released (freed) gradually as the learning progresses. These stages of motor learning allow to reduce the search space dimensionality. Yamamoto and Fujinami [11] also found a common organization of acquisition of a periodic skill: differenti-



ation within coordination. They compared clay kneading movements for pottery of experienced subjects and experts. While the experienced subjects tend to coordinate their body parts, slight phase differences between body parts are observed in experts' movements. Their group [12] found similar results for the proficiency of samba dance. A possible interpretation of the coordination of movement in the early learning is that less movement parameters simplify the optimization for the skill.

We introduce this idea of differentiation within coordination to optimization methods, and then aim to apply this to a soccer throwing skill. Most important in the acquisition of skilled throwing is the timing of releasing the ball. A humanoid robot searches for the best timing of the start of each joint based on timing of releasing a ball through practice as shown in Fig. 1. All joints related to the throwing skill are initially coordinated. That is, the robot roughly optimizes the timing of releasing a ball in a one-dimensional space. The joints are then gradually released, which allows the robot to search more optimal parameters. As a result, the robot acquires skilled throwing with only a small number of evaluations.

This paper is organized as follows: In section 2, we explain heuristic exploration using differentiation within coordination of DoFs. Throwing parametrization of a robot and the experimental setting are described in section 3 and 4, respectively. Section 5 then demonstrates that the proposed optimization method results in quicker exploration of optimal parameters. In section 6, the results are discussed, and in section 7 we conclude our research.

## 2 Heuristic exploration through differentiation within coordination

### 2.1 Differentiation within coordination

An example of exploration through differentiation within coordination of parameters is shown in Fig. 2. Here, we assume a two-dimensional search space, that is, the only parameters are  $S_1$  and  $S_2$ . There are two stages of optimization: coordination and differentiation.

#### 2.1.1 Coordination

The search space is restricted to coordination of all parameters, namely,  $S_1 = S_2$ . An initial value is selected in this one-dimensional space (on the dashed line in Fig. 2). The optimal parameter is then explored on the line.

#### 2.1.2 Differentiation

The restriction is gradually lifted after finishing the optimization in the previous search space. The search space is hence extended to multiple dimensions. Initial values are the best parameters in the previous stage. A solution of this algorithm is the optimal set of parameters after all parameters have been freed.

It is expected that this constraint on dimensionality reduces the search space, which accelerates the optimization.

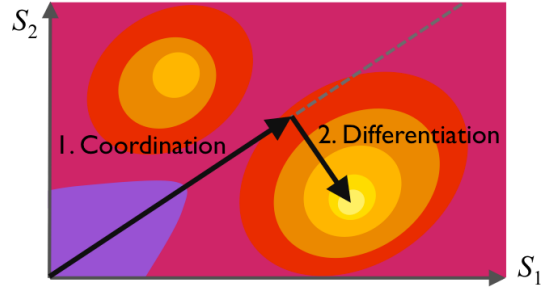


Figure 2: A conceptual example of the proposed exploration. In case of a two-dimensional objective function, the two parameters,  $S_1$  and  $S_2$ , are coordinated at first. These parameters are differentiated after optimization in the one-dimensional space. The constraint on the search space enables faster exploration.

### 2.2 Optimization method

Every time the dimension increases, an optimization method must be applied to the search space. This paper uses a hill climbing algorithm and a modified particle swarm optimization (PSO). These algorithms are widely applied to parameter optimization problems (see [3–5, 7]).

#### 2.2.1 Hill climbing

The Hill Climbing algorithm is one of the simplest optimization methods. It is known that this algorithm explores a solution quickly. An initial value is selected and evaluated in the search space. All neighbors of the initial parameter are evaluated, and the highest-scoring parameter among the neighbors is selected. The selected value is the next center, and then its neighbors' values are evaluated. Repeat the evaluation and the selection until no higher score can be found.

#### 2.2.2 Modified particle swarm optimization

PSO [13] is a probabilistic optimization method similar to genetic algorithms. Initially, a swarm of  $N$  particles is generated in the  $D$ -dimensional search space. Here, we introduce an initial value to this algorithm so that the optimization can inherit the best parameter of the previous search space. Although the existing PSO initializes the particles with random values, this paper sets the initial positions according to a normal distribution, where its mean and variance are the initial value and  $\mathbf{v}$ , respectively. These particles are assigned a position  $\mathbf{x}_i$  ( $1 \leq i \leq N$ ) and a velocity  $\mathbf{v}_i$  ( $1 \leq i \leq N$ ), which are  $D$ -dimensional vectors. Each particle is evaluated by the performance of its parameters. At each iteration, the velocity of each particle is updated depending on two values: the personal best position  $\mathbf{pbest}_i$  ( $1 \leq i \leq N$ ) and the global best position  $\mathbf{gbest}$ .  $\mathbf{pbest}_i$  is the best position that each particle has ever evaluated.  $\mathbf{gbest}$  is the best position that all particles have evaluated. Each velocity  $\mathbf{v}_i^t$  at iteration  $t$  is updated by:

$$\mathbf{v}_i^{t+1} = w\mathbf{v}_i^t + c_p r_p^t (\mathbf{pbest}_i^t - \mathbf{x}_i^t) + c_g r_g^t (\mathbf{gbest}^t - \mathbf{x}_i^t), \quad (1)$$



where,  $w$ ,  $c_p$  and  $c_g$  are weights.  $r_p$  and  $r_g$  are normal random numbers between 0 and 1. We restrict the range of velocity between  $\mathbf{v}^{max}$  and  $-\mathbf{v}^{max}$ , which is determined by:

$$\mathbf{v}^{max} = k \quad \mathbf{x}^{max}, \quad (2)$$

where,  $x^{max}$  is the range of exploration in each dimension, and  $0.1 \leq k \leq 1$ . The next positions of particles  $\mathbf{x}_i^{t+1}$  are calculated by:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^{t+1}. \quad (3)$$

We stop exploring when  $g_{best}$  does not change during  $n$  iterations.

### 3 Throwing parametrization

For our task, a robot searches the optimized combination of the start timing of each joint to throw the ball as far as possible. The VisiON 4G robot a commercial humanoid robot manufactured by Vstone Co.,Ltd. (see Fig. 1) was used for this experiment. Fig. 3 illustrates the robot’s DoFs configuration. The robot has 22 DoFs and each joint is actuated by a VS-SV410 servomotor.

From these 22 DoFs, we selected 4 DoFs for throwing:

Pitch shoulder: throwing the ball overhead.

Roll elbow: holding and releasing the ball.

Pitch waist: achieving more force by the reaction.

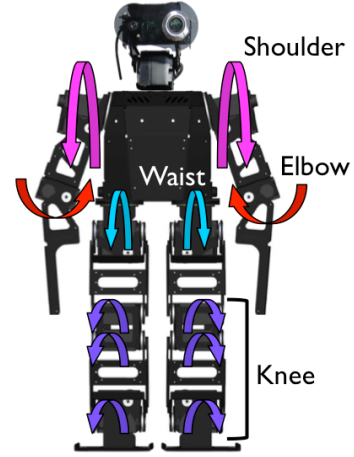
Knee: stretching both knees, which consist of 3 motors each.

Fig. 4 shows the definition of the parameters. We did not use velocities or positions of each Degree of Freedom(DoF) but the timing of movements of each DoF as parameters. In this case, the DoF of the elbow serves as the base of timing because it directly determines the release of the ball and is therefore most important for optimizing skilled throwing. We defined the timing of the start of movement of the shoulder, knee and waist based on the elbow’s timing as  $t_s$ ,  $t_w$  and  $t_k$ , respectively. The robot learns the optimal  $\mathbf{t} = (t_s, t_w, t_k)$  through practice. Initially, the 3 DoFs are coordinated, i.e.,  $t_s = t_w = t_k = t_{init}$ , and the robot optimizes  $t_{init}$  (see Fig. 4(a)). In the last stage of learning, all DoFs are differentiated. Thus the robot searches the optimal  $\mathbf{t}$  in the three-dimensional space (see Fig. 4(b)).

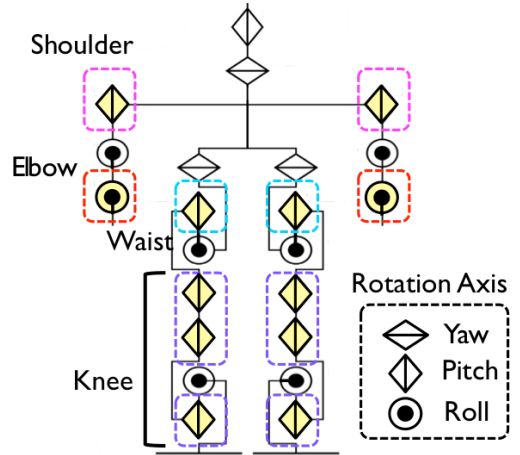
### 4 Experimental setting

In order to validate the proposed optimization method, we conducted experiments using a real robot. The robot explores optimal combinations of  $t_s$ ,  $t_w$ , and  $t_k$ .

The evaluation was determined by the distance between a robot’s toe and the ball point. The throwing distance was measured by visual inspection through video recording with a measuring tape as shown in Fig. 5. We evaluated the distance of throwing regardless of whether the robot fell down during the trials.



(a) Front view of VisiON 4G and its essential DoFs for throwing.



(b) DoFs configuration of VisiON 4G. The yellow joints are used in this experiment.

Figure 3: The number of the substantial DoFs used in the current experiment is 4: the pitch shoulder, the roll elbow, the pitch waist and the pitch knee. We assume symmetry of the motors. The DoF of the knee consists of 6 motors. The elbow affects the holding and releasing the ball.

The robot’s motion started from the same initial pose as shown in Fig. 1 in each trial. We gave the ball to the robot so that the robot could hold the ball with both hands. It took 10 steps to execute a throwing motion, where 1 step was 1/30 sec. The range of exploration was set to  $[-5, +2]$  based on the start timing of the elbow, and the range segmentation was 1 step. The robot rested for 5 minutes after every 10 trials to prevent overheating of the motors.

Optimization experiments were conducted on-line, and were applied to the dataset obtained by exhaustive search in advance. Two trials of the experiment were performed, each of which consisted of 512 different timings tested. The objective function was given the mean of two trials. We tested four optimization algorithms: Hill Climbing and PSO through differentiation within coordination, and existing Hill Climbing and PSO. We then compared the number of evaluations and achieved

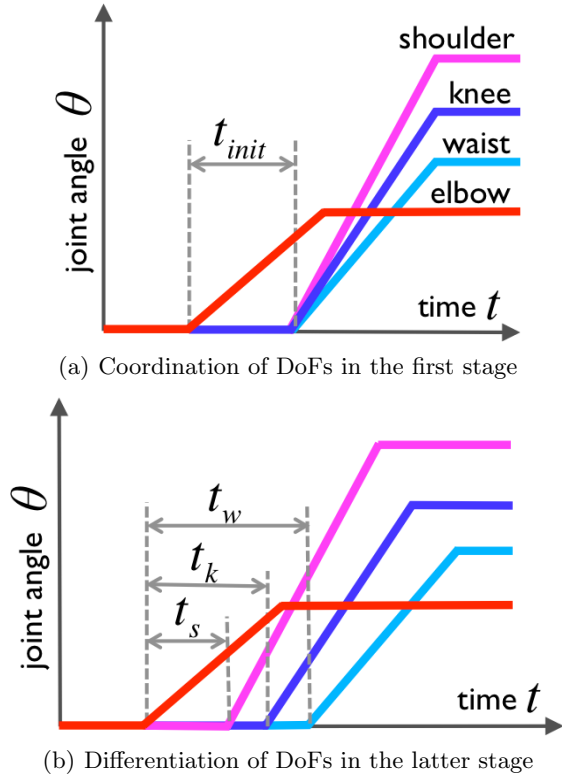


Figure 4: The number of parameters increases gradually during learning. In the early stage of learning (a), the DoF of the shoulder, the knee and the waist are coordinated. The robot explores the optimal  $t_{init}$ , namely timing of releasing of the ball in the one-dimensional space. In the last stage of learning (b), the timing of the start of the each DoF,  $t_s$ ,  $t_w$ ,  $t_k$ , is optimized.

optimal performance.

In the Hill Climbing algorithm, one iteration needs 26 evaluations in the three-dimensional search space. However, we did not count the evaluations of the parameters where the robot once searched. The variables in the PSO were empirically determined: 5 particles were initially positioned according to a normal distribution, whose variance was set to 3. We set  $w = c_p = c_g = 0.5$  in Eq. (1) and  $k = 0.25$  in Eq. (2). The optimization was finished when  $\mathbf{g}_{best}$  did not change for 3 iterations.

The initial parameter was given as an integer between -5 and 2 (i.e., 8 patterns). Each optimization method was conducted 8 times for all initial parameters. The proposed PSO was ran 10 times with each initial parameter setting because PSO includes randomness. In the existing PSO, randomly-selected initial parameters are given, and then we tested it 80 times.

## 5 Result

### 5.1 Number of trials and throwing performance

Fig.6(a) shows the average number of trials for each optimization method. Less trials mean faster exploration, which relieves the robot of load. The blue bars at the left side and red bars at the right side denote the re-

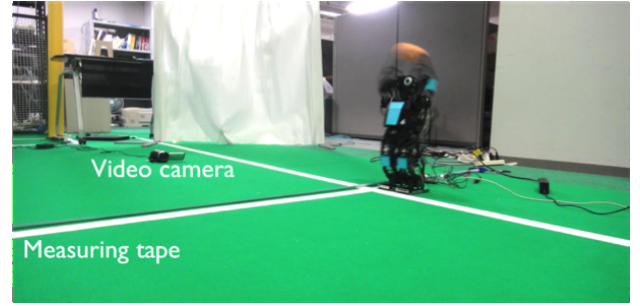
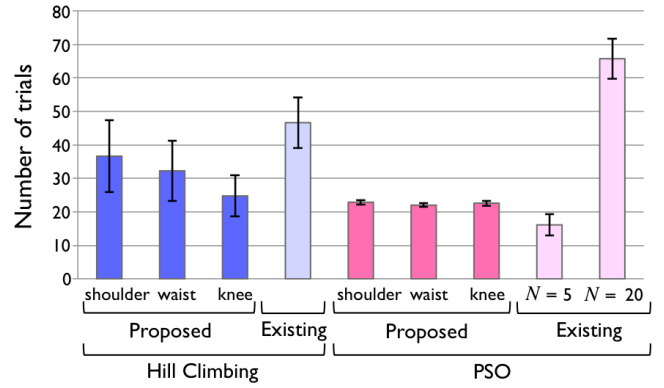
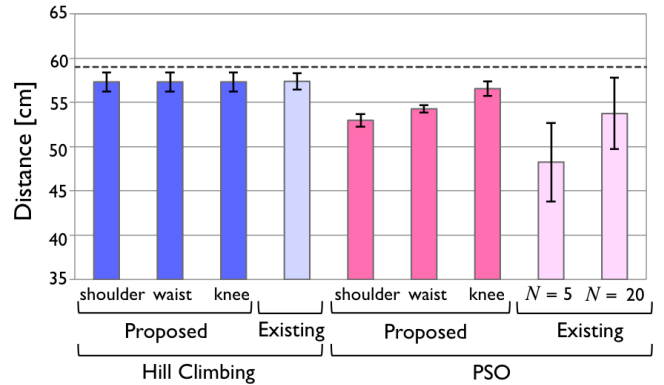


Figure 5: The experimental environment to optimize the parameters for throwing. We record the distance between a robot's toe and a ball fall point.



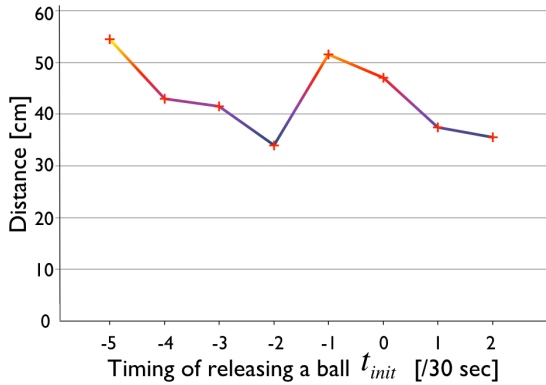
(a) The number of trials



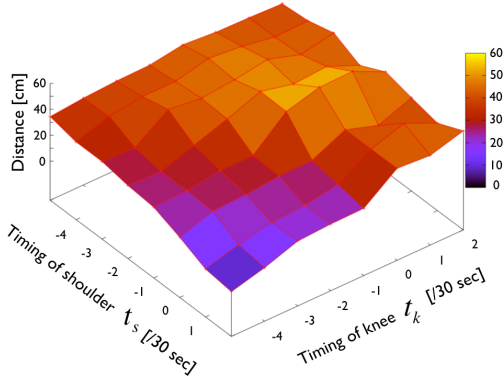
(b) Throwing performance. The dashed line denotes the global optimum (59 cm).

Figure 6: The results of each optimization method. The blue and the red bars indicate the results of the Hill Climbing algorithm and PSO, respectively. The proposed methods differentiate DoF of shoulder (left), waist (middle), knee (right) from other DoFs in the second stage.  $N$  is the number of particles.

results of the Hill Climbing algorithm and PSO, respectively. There are three results in the proposed optimization through differentiation within coordination: shoulder (left), waist (middle) or knee (right) were differentiated from other DoFs in the second stage. We can see that both proposed methods result in less trials than the existing methods. However, the result of PSO using 5 particles ( $N = 5$ ) shows the least number of trials.



(a) one-dimensional search space



(b) two-dimensional search space

Figure 7: Objective function. Proposed algorithm optimizes  $t_{init}$  in the one-dimensional space (a) in the first stage of exploration. In two-dimensional search space with  $t_w = 1$  (b), the global optimal parameter  $\mathbf{t}^{opt}$  is  $(-1, -1, 1)$  which results in a distance travelled of 59cm.

The optimized performance of each method is shown in Fig.6(b). The dashed line indicates the global optimum. We find that all results of Hill Climbing show high performance. The results of the proposed PSO through coordination show less variance and have nearly the same performance as the existing PSO with  $N = 20$ . The existing PSO with  $N = 5$  needs the least number of trials but also performs worst. Therefore, the proposed method can reduce the number of trials while maintaining the high performance of the existing PSO.

## 5.2 Objective function

As shown in Fig. 7, the objective function is obtained by exhaustive search in order to reveal the cause of the above result. Fig. 7(a) illustrates a one-dimensional objective function, where the optimal  $t_{init}$  values are explored in the first stage. There are two local maxima:  $t_{init} = -1, -5$ . The global optimum  $\mathbf{t}^{opt}$  is  $(t_s, t_w, t_k) = (-1, -1, 1)$  as shown in Fig. 7(b). Thus, the result of optimization in the one-dimensional space should be -1 so that the robot can finally find  $\mathbf{t}^{opt}$ . In Hill Climbing, the optimal  $t_{init}$  is -1 if the initial value is more than -2. All particles, however, move toward -5 in PSO. This is why the optimization by PSO with coordination of DoFs was worse than by Hill Climbing (see Fig. 6(b)).

The coordination of the DoF of the shoulder and the

waist makes it easier to reach  $\mathbf{t}^{opt}$  because  $\mathbf{t}^{opt}$  is  $t_s = t_w = 1$ . Thus, the optimization through coordination of the DoF of  $t_s$  and  $t_w$  in the second stage results in the best performance. The proper order of releasing the DoF may be task-dependent.

## 6 Discussion

It is hard for a robot to acquire skilled throwing. Even if we use a faithful simulator or make a dynamical mathematical model of a robot, there exists a gap between the real and the virtual world. One of the differences originates from the environmental complexity. The robot's body interacts with the ball during throwing. The ball deforms slightly and the robot undergoes reaction forces. This interaction seems to influence the performance. Most simulators, however, cannot address detailed touch calculations. The inherent delay of motors from motor commands is also a crucial problem. Many athletic behaviors such as throwing are instantaneous movements. The throwing took only 1/3 sec in this experiment. Thus the motor's tiny delay makes a difference of performance. After all, it is necessary for acquisition of skilled behavior to optimize in high-dimensional space using a real robot.

We demonstrated that the high dimensionality may be reduced by differentiation within coordination of the DoFs. In particular, this optimization method might be applicable for athletic behaviors. Many body parts move instantaneously in parallel moment, which can be regarded as coordination of body parts. However, a little differentiation of movements of each body part is observed in human's skilled motion. In the throwing by an expert the timing to maximum velocities of body parts does not always correspond to the timing of releasing of a ball [14]. It means that there are tiny phase displacements between DoFs for effective movements. The timing optimized in coordination of the DoFs is close to the global optimum. Thus, the local maximum reached in the first stage of exploration is useful even if the space dimensionality increases.

Our proposed optimization method can be applied to other skills. A slight differentiation of the timing of a leg's DoF may be important in high-kicking (kicking the ball as high as possible), which has been an official technical challenge in the RoboCup soccer humanoid league since 2012. We will attempt to optimize these soccer skills by the proposed method. In addition, velocity of body parts is also important for skilled behavior. We will address the acquisition of skilled behavior with more parameters.

## 7 Conclusion

In this paper we presented a practical optimization method through differentiation within coordination of a robot's body parts. All the DoFs related to the skill were coordinated in the first stage of learning. Thus, the robot optimized the timing of the start of releasing the ball in one-dimensional space. The DoFs were then differentiated one by one, which enabled the robot

to explore the optimal timing of the start of each joint's movement. The reduction of the search space dimensionality, consequently, could decrease the number of trials. Our experiments showed that the optimization through coordination of the DoF resulted in as high performance as the result of optimizing without coordination even if less trials were used.

This optimization method may be leveraged when acquiring quick movements such as throwing, kicking and so on. Instantaneous athletic skills can be coordinated behaviors. Thus the optimization of coordination of DoF might be more plausible, i.e., not just a local solution. The robot can reach quickly a valid solution because of usage of the best solution in the previous stage.

## Acknowledgment

The authors gratefully acknowledge the contribution of the team members of the JST ERATO Asada Project(JEAP) for all materials, and also Associate Prof. Tomomichi Sugihara and a former student, Hirotaka Kimura (Currently, the City of Nagoya, Japan) at Osaka University for their suggestion about heuristic exploration.

## References

- [1] GS Hornby, M. Fujita, S. Takamura, T. Yamamoto, and O. Hanagata. Autonomous evolution of gaits with the sony quadruped robot. In *Proc. of the Genetic and Evolutionary Computation Conference*, volume 2, pages 1297–1304, 1999.
- [2] G. Daoxiong, Y. Jie, and Z. Guoyu. A review of gait optimization based on evolutionary computation. *Applied Computational Intelligence and Soft Computing*, 2010, 2010.
- [3] C. Rong, Q. Wang, Y. Huang, G. Xie, and L. Wang. Autonomous evolution of high-speed quadruped gaits using particle swarm optimization. *RoboCup 2008: Robot Soccer World Cup XII*, pages 259–270, 2009.
- [4] N. Shafii, S. Aslani, O. Nezami, and S. Shiry. Evolution of biped walking using truncated fourier series and particle swarm optimization. *RoboCup 2009: Robot Soccer World Cup XIII*, pages 344–354, 2010.
- [5] N. Kohl and P. Stone. Machine learning for fast quadrupedal locomotion. In *Proc. of the 19th National Conf. on Artificial Intelligence*, pages 611–616, 2004.
- [6] M. Saggarr, T. D'Silva, N. Kohl, and P. Stone. Autonomous learning of stable quadruped locomotion. *RoboCup 2006: Robot Soccer World Cup X*, pages 98–109, 2007.
- [7] M. Hausknecht and P. Stone. Learning powerful kicks on the aibo ers-7: The quest for a striker. *RoboCup 2010: Robot Soccer World Cup XIV*, pages 254–265, 2011.
- [8] N.A. Bernstein. The co-ordination and regulation of movements. 1967.
- [9] K.M. Newell and D.E. Vaillancourt. Dimensional change in motor learning. *Human Movement Science*, 20(4-5):695–715, 2001.
- [10] B. Vereijken, R.E.A. van Emmerik, HTA Whiting, and K.M. Newell. Free(z)ing degrees of freedom in skill acquisition. *Journal of Motor Behavior*, 24(1):133–142, 1992.
- [11] T. Yamamoto and T. Fujinami. Hierarchical organization of the coordinative structure of the skill of clay kneading. *Human movement science*, 27(5):812–822, 2008.
- [12] K. Matsumura, T. Yamamoto, and T. Fujinami. A study of samba dance using acceleration sensors. In *Proc. of the 8th Motor Control and Human Skill Conference*, pages 5–4, 2007.
- [13] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proc. of the IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995.
- [14] T. Reilly. *Science and Soccer*. Routledge, 1995.

# Likelihood estimated by two types of Neural Networks

C.-H. Lee, L.-W. Lu, N. M. Mayer

Nat'l Chung Cheng University,

Chia-Yi, Taiwan  
mikemayer@ccu.edu.tw

*Abstract*—The purpose of this contribution is to review two relatively recent developments in the field of neural networks to the robotics audience. The first are deep belief networks and the second are echo state networks. Since much of sensor processing relates to probability estimates and also sampling of data in this work we focus on how to derive directly a probability and likelihood estimates. As it turns out both show complementary features and strengths which make a combination of both highly eligible.

## 1. Introduction

Most out of the box sensor processing approaches, such as Kalman and particle filters[1] either explicitly or implicitly assume certain constraints on the uncertainty distribution of the sensor input and the internal state of a robot. Usually it is assumed that that uncertainty can be expressed by a normal distribution. For everyday applications that works very well although it is common knowledge that real world noise is often not very similar to a Gaussian distribution. This has consequences, such as the fact that many approaches offer a version in which outliers are discarded in a separated step because outliers are covered badly by the Gaussian assumption and have a strong impact on the mean and variance.

Within this work we present two approaches that base on neural networks echo state networks and deep belief networks [2, 3] which in 2 different ways give estimation of a likelihood without any further assumptions on the underlying probability distribution, where the Deep Belief Network as a generative model generates samples (particles with the correct probability distribution).

Deep Belief Networks (DBNs) are one of the mainstream deep learning approaches, which were initially introduced in [4]. DBNs can learn the networks one layer at a time in a biologically plausible way. DBNs have vastly been applied to many types of data in artificial intelligence, such as hand-written digit images of binary pixels in MNIST database[5], windows of mel-spectral coefficients in speech recognition[8] and gray-scaled images in 3D object recognition in NORB dataset[7]. In our paper, we will further examine the ability of DBNs in interpreting

binary encoded data, and we will validate this by showing how dose DBNs perform in learning various distributions.

In our work, we have tried to construct each layer RBMs into individual dynamic linking module, this layer-like property made it easily duplicated and distributed under our Qi's framework[11]. We have successfully implemented the deep believe network on the hand-written digits task using as presented in [4] using the framework. We have also worked on training the binary-represented random numbers by DBNs and try to reproduce an identical distribution from such random number generator.

Sect. 2.1 introduces the structure of Deep Belief Network and Restricted Boltzmann machines (RBMs), the basic component of DBNs. Sect. 2.2 shows how DBNs can learn the binary encoded data by learning certain distributions. In Sect. 2.3, we will propose some ideas we have came up with on combining DBNs with domestic service robot application.

The second approach presented here is the echo state network [22]. Here, we directly attack the concept of likelihood by the idea that all relevant information for the probability estimate in some sense is represented in the reservoir, thus it should be possible to read it out in order to retrieve the best possible probability estimate. We show one example in a experiment and read use this as a benchmark for our approach. The general field of application would be the analysis of time series.

Time series prediction is important to forecast e.g. economical data, and used to make decisions which, in turn, change economy. The term “causality” is used when past values of a time series provide significant information about future values of another time series [26]. One of the possible methods for causality inference is transfer entropy [25], however, it has the disadvantage of requiring a fair amount of data. Granger causality [20], on the other hand, is based on regression and uses less data, but is a linear method (non-linear extensions exist, see also [6] for a comparison between different methods). In this work, we propose a (non-linear) approach based on regression and a recent recurrent neural network learning method, which we

briefly revisit in Sect. 3.1 Recent advances in this area have shown to be successful in time series prediction [4, 5, 3]. However, instead of using predictions of the neural networks directly, we take a different route and describe an approach using the prediction error to detect causal links in time series (Sect. 3.2). Our approach is demonstrated using simulation data (Sect. 3.3), and finally, we discuss our results in Sect. 3.5.

## 2. Numerical Experiments with DBNs

### 2.1 Deep Belief Networks

DBNs are probabilistic generative models composed of multi-layer directed networks of stochastic, latent variables. These stochastic latent variables then form into Restricted Boltzmann Machines (RBMs) a type of neural networks, which are the basic components of DBNs. These networks are "restricted" to a visible layer and a hidden layer, the latter can be viewed as a feature detector capturing correlations that observed at the former.

#### A. Restrict Boltzmann Machine

The Restrict Boltzmann Machines are the type of generative models which can be applied on many types of data such as binary images in MNIST database[5][4], gray-scaled images of 3D objects in NORB dataset[7], colored nature image[6], or mel-cepstral coefficients in speech recognition[8]. To train the RBMs, contrastive diversions[12] are often performed in order the obtain generative weights.

A RBM is a two-layer network in which visible layer and hidden are symmetrically connected and the siblings within one layer are not connected (see Fig. 1).

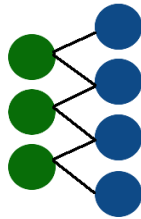


Fig. 1. Structure of RBMs, one RBMs contains a hidden layer and a visible layer.

The network takes the idea of “energy” as described by Hopfield, 1982, in which the joint configuration of a pair  $(v,h)$  has an energy  $E(v,h)$  :

$$E(v, h) = - \sum_{i \in \text{visible}} a_i v_i - \sum_{j \in \text{hidden}} b_j h_j - \sum_{i,j} v_i h_j w_{ij} \quad (1)$$

where  $i$  indicates visible unit and  $j$  indicate hidden unit,  $a_i, b_j$  are their biases respectively and  $w_{ij}$  is the weight between them.

For each possible pair  $(v,h)$ , the energy function is:

$$p(v, h) = \frac{e^{-E(v,h)}}{\sum_{v,h} e^{-E(v,h)}} \quad (2)$$

The probability of visible vector or training image,  $v$ , is given as (3).

$$p(v) = \frac{\sum_h e^{-E(v,h)}}{\sum_{v,h} e^{-E(v,h)}} \quad (3)$$

Hence, we can compute the derivatives of the log probability of data, and the maximum likelihood learning rule for data vector  $v$  can be defined as (4).

$$\frac{\partial \log p(v)}{\partial \omega_{ij}} = \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}} \quad (4)$$

Then we can repeatedly update the weight,  $w_{ij}$ , of visible unit  $i$  and hidden unit  $j$ . Which lead to (5).

$$\Delta w_{ij} = \varepsilon (\langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}}) \quad (5)$$

where  $\varepsilon$  is a learning rate.

A fast learning algorithm is proposed in [4] which has termed *contrastive divergence*. Refer to [4] and [14] for further details.

#### B. Deep Belief Networks

As stated before, DBNs are probabilistic generative models composed of multilayer directed networks of stochastic, latent variables. These stochastic latent variables then form a Restricted Boltzmann Machine, which is the basic components of DBNs. According to [16], DBNs have two most significant properties:

1. “There is an efficient, layer-by-layer procedure for learning the top-down, generative weights that determine how the variables in one layer depend on the variables in the layer above.”
2. “After learning, the values of the latent variables in every layer can be inferred by a single, bottom-up pass that starts with an observed data vector in the bottom layer and uses the generative weights in the reverse direction.”

## 2.2 DBNs on Learning Binary Data



Deep Belief Networks can deal with many different types of sensory input data, including binary colored image, colored nature image, speech phones or even video sequence. In this paper, we want to examine the ability of DBNs on binary encoded sensory input. In this section, we will first depict our experimental setup and we will show that DBNs works well on extracting features from binary encoded input data.

### A. Experimental Setup

In this experiment, we first generate a double precision number of base 10 from a certain distribution, then we encoded this number into binary base according to IEEE-754[17]. (see the shaded area of Fig. 2).

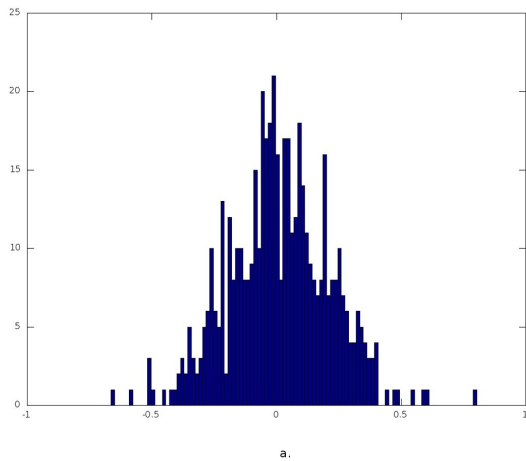
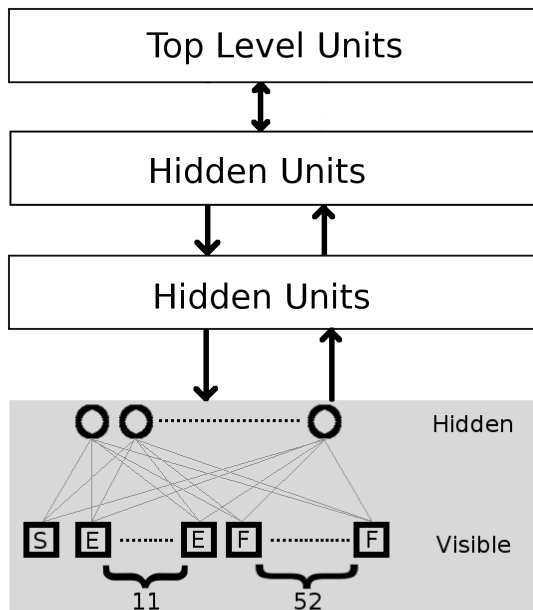


Fig. 2. Structure of our Deep Belief Networks. The shaded area explain how we encoded into binary number. Shape squares stands for each node in visible layer, and circle stands for each node in hidden layer. Character 'S' stands for *sign bit* in IEEE-754, 'E' stands for *exponent bits* and 'F' stands for *fraction bits*. This two layer form a Restricted Boltzmann Machine in Deep Belief Networks.

Later, we construct the DBNs as in Fig. 2, there are four hidden layer and one visible layer(input layer), the size of each hidden layer is set as following respectively, 256, 512, 256 and 64. For each RBMs, we train each layer for 100 epochs, each epoch training through the whole data set once. After training, in order to examine the result, we let the network runs as generative mode for the exact the same amount of data as training set.

### B. Experiment Result

#### 1. Normal Distribution

In this experiment, we have tested two normal distribution, one with mean at 0 and another with mean at 10. We generate the number from the function of these two distributions as input data for DBNs respectively. Fig. 3 shows that DBNs can learn the exact mean and deviation of normal distributions and reconstruct the identical distribution.

#### 2. Poisson Distribution

In experiment 1, we have tested the **continuous** distribution. This experiment test test the networks with Poisson distribution, which has the **discrete** property. Fig. 4 shows DBNs can also

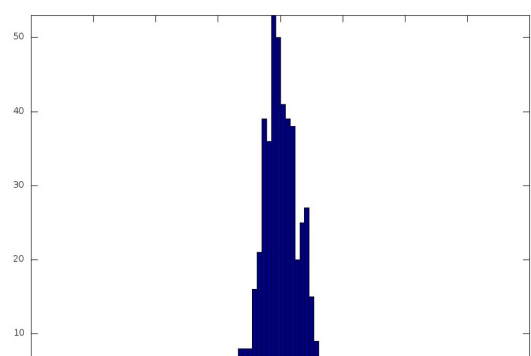
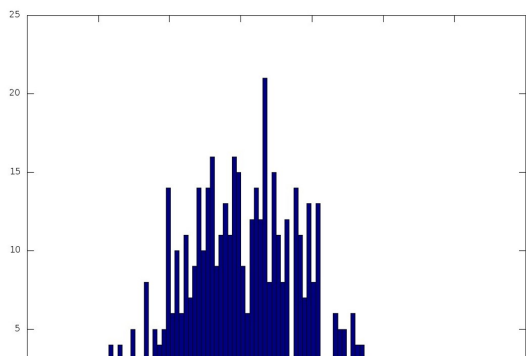
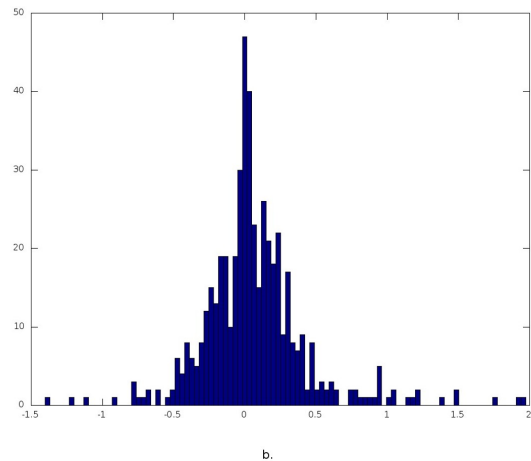


Fig. 3. The result of learning normal distribution with mean 0 and 10. Picture a is the normal distribution with mean 0, b is the learning result of reconstructed distribution. Picture c is the normal distribution with mean 10, d is the result of reconstructed distribution learning from c.

learn the discrete distribution.

### 3. Random Number Generator

In the last experiment, we examine with a uniform distribution. We tested the network with 5000 random numbers. The result is presented in Fig. 6, one can see that most of the reconstructed numbers lay within the range between 0 and 1. This result needs more fine tuning, for example, there's a small gap near 0 with extremely low probability compared to the region nearby and not every reconstructed number is between 0 and 1. In conclusion, the network can get a approximation of our test set.

## 2.3 Combing DBNs with Domestic Service Robots

Among the ideas, we took the advantages of great performance of deep autoencoder[8]. In the service robot competitions, the robots must operate in everyday human

environments, where they have to perform a given set of service tasks, such as following instructions like gesture, voice or even remote control signals. In voice recognition task, we first train the users voice using CMUSphinx[9], the output sentence of CMUSphinx need further treatment since we can not assure zero error in speech recognition. In natural language, one or more mistakes in one sentence may lead to the meaningless task. Hence, we feed those results into our DBNs modules and train the network to categorize semantics into certain task. Another application could be categorizing environment. We take kitchen as example, one could easily imagine what types of objects may appear inside the kitchen environment such as a pot, oven or stove. We will leave the topic as an open issue here, additional works need to be done in the future. Thus, we see good reason to investigate for the service robot tasks. In the following passage we give a example in phone recognition as present in [4]. By doing above experiments, we have shown that Deep Believe Networks can successfully learn the probability distribution with binary encoded dataset. With this property, we can fur-

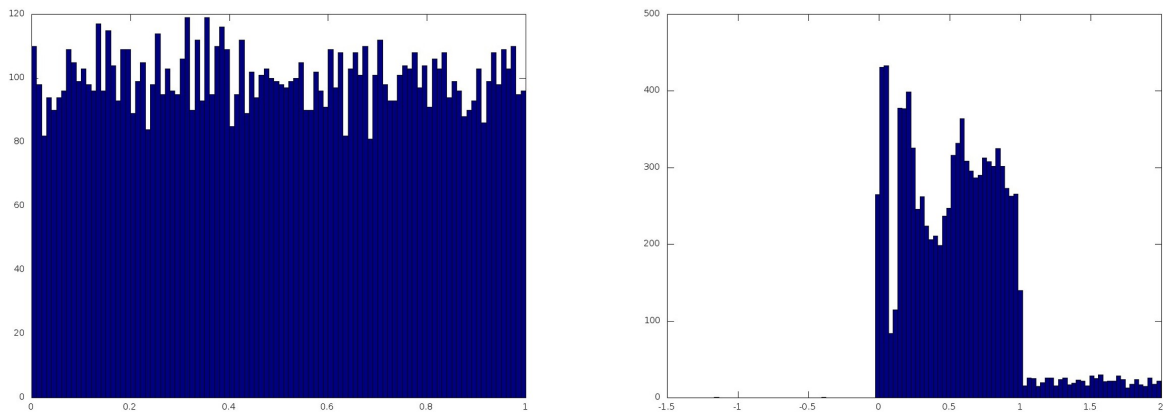


Fig. 4. The result of learning Uniform distribution from random number generator. Left picture shows the original distribution, on the right hand side is the reconstruction of it.

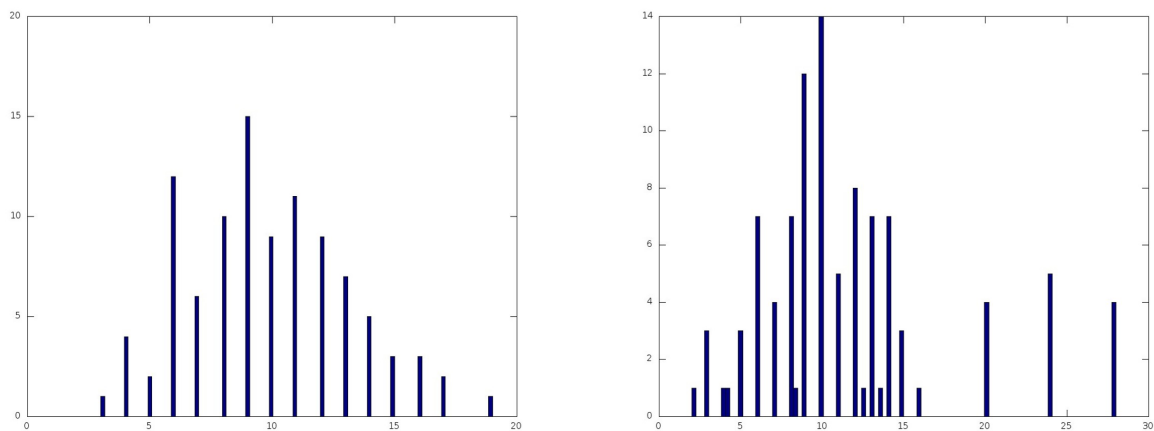


Fig. 5. The result of learning Poisson distribution. Left picture shows the original distribution, on the right hand side is the reconstruction of it.



extend DBNs into state prediction application. In robotics, we often are confronted with the problem of state prediction, for instance, ball prediction. In domestic service robot, particularly in mapping and tracking, state prediction of self-localization and state of the tracked object, one may use DBNs to calculate the posterior of Bayesian filters.

### 3. ESN as a likelihood estimator

## 3.1 Echo State Networks

Echo State Networks (ESN) are an approach to address the problem of slow convergence in recurrent neural network learning. ESN consist of three layers (see Fig. 6): a) an input layer, where the stimulus is presented to the network; b) a randomly connected recurrent hidden layer; and c) the output layer. Connections in the output layer are trained to reproduce the training signal. The network dynamics is defined for discrete time-steps  $t$ , with the following equations:

$$\mathbf{x}_{in,t+1} = \mathbf{W}\mathbf{x}_t + \mathbf{w}^{in}\mathbf{u}_t \quad (6)$$

$$\mathbf{x}_{t+1} = \tanh(\mathbf{x}_{in,t+1}) \quad (7)$$

$$\mathbf{O}_t = \mathbf{w}^{out}\mathbf{x}_t \quad (8)$$

where the vectors  $\mathbf{u}_t$ ,  $\mathbf{x}_t$ ,  $\mathbf{o}_t$  are the input and the neurons of the hidden layer and output layer respectively, and  $\mathbf{w}^{in}$ ,  $\mathbf{W}$ ,  $\mathbf{w}^{out}$  are the matrices of the respective synaptic weight factors.

Connections in the hidden layer are random but the system needs to fulfil the so-called echo state condition. Jaeger [22] gives a definition; in the following a slightly more compact form of the echo state condition:

Consider a time-discrete recursive function  $\mathbf{x}_{t+1} = \mathbf{F}(\mathbf{x}_t, \mathbf{u}_t)$  that is defined at least on a compact sub-area of the vector-space  $\mathbf{x} \in \mathbb{R}^n$ . and where  $\mathbf{x}_t$  are to be interpreted as internal states and  $\mathbf{u}_t$  is some external input sequence, i.e. the stimulus.

The definition of the echo-state condition is the following: Assume an infinite stimulus sequence:  $\bar{\mathbf{u}}^\infty = \mathbf{u}_0, \mathbf{u}_1, \dots$  and two random initial internal states of the system  $\mathbf{x}_0$  and  $\mathbf{y}_0$ . To both initial states  $\mathbf{x}_0$  and  $\mathbf{y}_0$  the sequences  $\bar{\mathbf{x}}^\infty = \mathbf{x}_0, \mathbf{x}_1, \dots$  and  $\bar{\mathbf{y}}^\infty = \mathbf{y}_0, \mathbf{y}_1, \dots$  can be assigned.

$$\mathbf{x}_{t+1} = \mathbf{F}(\mathbf{x}_t, \mathbf{u}_t) \quad (9)$$

$$\mathbf{y}_{t+1} = \mathbf{F}(\mathbf{y}_t, \mathbf{u}_t) \quad (10)$$

Then the system  $\mathbf{F}(\cdot)$  fulfils the echo-state condition if independent from the set  $\mathbf{u}_t$  and for any  $(\mathbf{x}_0, \mathbf{y}_0)$  and all real values  $\epsilon > 0$  there exists a  $\delta(\epsilon)$  for which

$$d(\mathbf{x}_t, \mathbf{y}_t) \leq \epsilon \quad (11)$$

for all  $t \geq \delta$ . The ESN is designed to fulfil the echo state con-

dition.

### 3.1.1 Online learning using recursive least squares

ESN can be trained using either an offline or an online learning procedure. For our approach, we are online learning the output layer using the recursive least square method (RLS). The combination of ESN and RLS has first been published by Jaeger [23]. The following update rule was used:

$$\alpha_t = \mathbf{s}_{teach}^t - \mathbf{w}_t^{out} \cdot \mathbf{o}_t, \quad (12)$$

$$\mathbf{g}_t = \mathbf{p}_t \cdot \mathbf{o}_t / (\lambda + \mathbf{o}_t^T \cdot \mathbf{p}_t \cdot \mathbf{o}_t), \quad (13)$$

$$\mathbf{p}_t = 1/\lambda \cdot \mathbf{p}_t - \mathbf{g}_t \cdot \mathbf{o}_t^T \cdot \mathbf{p}_t / \lambda, \quad (14)$$

$$\mathbf{w}_t^{out} = \mathbf{w}_t^{out} + (\alpha_t \cdot \mathbf{g}_t^T), \quad (15)$$

where  $\alpha_t$  represents the linear error vector and  $\mathbf{p}_t$  the inverse of the autocorrelation,  $\lambda$  is close to 1 and is used as forgetting factor.

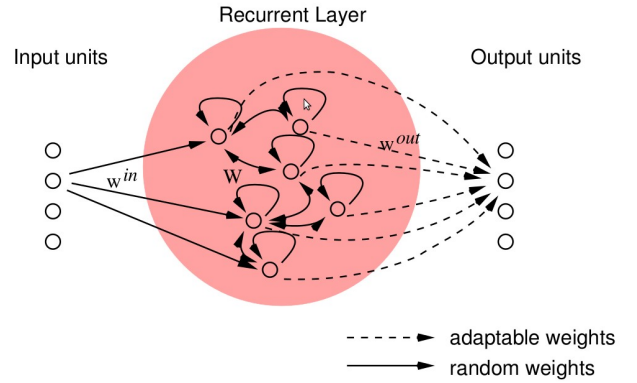


Fig. 6. ESN networks: Principle setup

## 3.2 Modelling probability distributions by using the mean square error

In the following we reproduce one idea that was outlined in the tutorial of Jaeger [22] that we also used as the basis in our previous publication[3]. Instead of training the output  $\omega_t \in \Omega$  directly, we model a probability that a specific event has occurred with regard to the output. In other words: the aim is to train the network in that way that each of the output units represents the probability of an event. As one simplest way to do this we teach the output or of the network to reproduce the probability that the as a range of the – statistical – output variable  $\Omega_r \subset \Omega$  that is of interest for the given task. The task of network is to find  $p(\Omega_r | \mathbf{x}_t(\bar{\mathbf{u}}^\infty))$  in the following written short  $p(\Omega_r)$ . We define  $\mathbf{u}$  the teaching signal  $\mathbf{d}_r$  as:

if  $(\omega_t \in \Omega_r) d_r = 1$   
else  $d_r = 0$

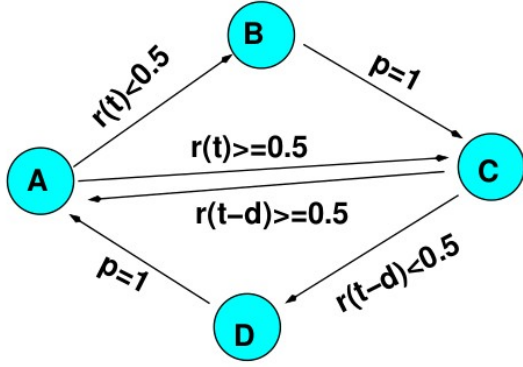


Fig. 7. Test model set-up.

The mean square error (MSE) is the

$$E_{\text{mse}} = \langle (d_{r,t} - o_r)^2 \rangle = p(\Omega_r)(1 - o_r)^2 + (1 - p(\Omega_r))o_r^2 \quad (16)$$

The derivative  $\partial E_{\text{mse}} / \partial o_r$  set equal to zero yields the point at which  $E_{\text{mse}}$  is minimal:

$$o_r - p(\Omega_r) = 0 \quad (17)$$

Thus, the MSE is reached when  $o_r = p(\Omega_r)$ ; we can assume

$$o_r \rightarrow p(\Omega_r), \quad (18)$$

for sufficiently long learning sequences. Since – with the common restrictions of reservoir computing– the full information of the input history is encoded in the activity state of the reservoir. Thus, –without additional efforts in the hidden layer– more information about statistical variables can be retrieved from additional output units: Because the optimal solution (absolute minimum of the MSE) can be derived, the network is going to find the true probability as far as it is detectable by linear regression from the current state of the reservoir. Usually, the quality of the network performance and the learning progress can be checked by measuring  $E_{\text{mse}}$ , where values close to zero represent a good network performance. It should be noted that for the learning rule outlined above the theoretical limit is above zero. Under the assumption that the  $p(\Omega_r)$  is the true probability we get:

$$E_{\text{min}}(\Omega_r) = \min_{o_r}(E_{\text{mse}}) = p(\Omega_r)(1 - p(\Omega_r)). \quad (19)$$

However, since in fact the true value  $p(\Omega_r)$  is unknown, it is not a good idea to use  $E_{\text{mse}} - E_{\text{min}}$  as a measure. However, it can be used to find out if the output node is deterministic (i.e.

the output node takes either 0 or 1). In this case the minimal error is in fact 0 again.

Instead one could go the following way in that we can get a set of outputs that covers a complete range of the random variable in the way that for a range  $r \in \mathbb{R}$ :

$$\bigcup_{r \in \mathbb{R}} \Omega_r = \Omega, \quad (20)$$

$$\Omega_i \cap \Omega_j = \emptyset, \quad (21)$$

for all  $i \neq j$ . Obviously, we have  $r \in \mathbb{R} p(\Omega_r) = 1$ . We can test the constraint in the network. We test the quality of the network output by testing measuring  $r$  or which should be close to one if the network has adapted sufficiently.

Basing on the plausibility constraint it is very easy to define an error function for the network. In the simplest case, it can be assumed to train 2 outputs, of which the first represents the occurrence of an event  $e \in \Omega_x$ , whereas the second output is trained to record the non-occurrence  $e \in \Omega_x$ . Thus, I train  $(d_1 = 1, d_2 = 0)$  in the case  $e \notin \Omega_x$  and else  $(d_1 = 0, d_2 = 1)$ . In this case it can be assumed that the cost function

$$E_{\text{total}} = (o_1 + o_2 - 1)^2 \quad (22)$$

approaches zero after a sufficient long learning process since from Eq. 18 we get

$$E_{\text{total}} \rightarrow (p(\Omega_x) + p(\Omega_x) - 1)^2 = (p(\Omega_x) + (1 - p(\Omega_x)) - 1)^2 = 0. \quad (23)$$

Thus, this energy function may serve better as an estimator how well the network has adapted to the particular current input history.

#### 4 .Simulation details

We demonstrate the approach on a prediction task. Our test model cycles between four states (A, B, C, D). Figure 7 outlines the transition probabilities. Every time in which the state A is reached a random number  $0 \leq r(t) < 1$  is drawn. In every time step the model transfers from one state to the next state. Is the current state the state A and the random value  $r(t)$  smaller then 0.5 the model goes to state B else the model transfers to state C. From state B the model transfers to C. From C a the next step is either D if  $r(t - d) < 0.5$  else the model returns to state A, where  $d$  is a positive or zero delay constant (0 delay indicates that the transition C-A happens in the same cycle as A-C, i.e. the complete cycle becomes A-C-A. Is the model in state D it always transits back to A.

The task of the network is to predict the next state from the previous. Had the states A, B, C, D been interpreted directly as Markov states the transitions from C to the next state would appear random with equal probability either to state A or D, i.e. in

that interpretation the model is non-Markov. However, a perfect hidden Markov model (HMM) with 2d hidden states would – presumably – be able to detect the fact that the choices of the transitions from A and those from C are linked.

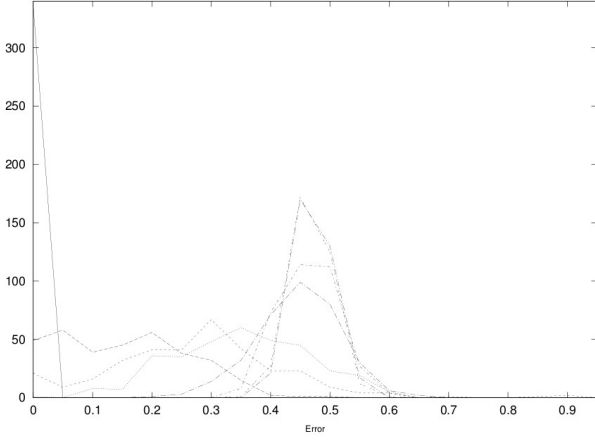


Fig. 8. Histograms of errors for different delays. Depicted is the MSE of probability of the transition to state D in the event of initial state C. If the network cannot detect the causality of transitions from C the error is at best 0.5. Full lines depicts the error at delay 0, dashes delay 1, small dashes delay 2, dotted delay 3, dash-dotted delay 4, double dashed delay 5, small dash-dotted delay 6.

The model is presented as an 8 dimensional vector in the following way to the network:

$$\begin{aligned} A &= [1, 0, 0, 0, 0, 1, 1, 1] \\ B &= [0, 1, 0, 0, 1, 0, 1, 1] \\ C &= [0, 0, 1, 0, 1, 1, 0, 1] \\ D &= [0, 0, 0, 1, 1, 1, 1, 0] \end{aligned}$$

The second half of each vector represents the inverse of the first. Thus, it can serve to find the cost function according to Eq. 23. Obviously, the task becomes more complex as the delay constant increases. The inverse correlation matrix was set to  $p = 0.0001 \cdot I$  where  $I$  is the identity matrix. The forgetting factor  $\lambda$  was set to 1.0, which sets the RLS into the non-forgetting mode. The recurrent matrix was set to random orthonormal matrix which was multiplied by 0.98 (Fig. 8) and 1.14 (Fig. 9), which gives a slightly over-critical network. Since the input practically is never close to 0, the network stays non critical. The online learning was performed from the 1000th step onwards up to 18000th step. The different MSEs were recorded in the last 100 steps of each simulation.

## 5 .Results

We tested the network for several delays and network

sizes. To estimate the network performance we used the MSE error at the transition from node C, in following  $E_{MSE,C}$ . A network that is able to detect the causality between the history and the transition from node C can reach zero MSE, whereas for network that cannot detect the causality the transition appears to be stochastic with equal probability to state D and A. The MSE in this case can be determined by Eq. 19.

A histogram of errors for different delays is depicted in Fig. 8. For sake of simplicity we used the MSE of transitions from A (stochastic,  $E_{MSE,A}$ ) and B (deterministic,  $E_{MSE,B}$ ) in Fig. 9. The plot depicts the values of

$$E_{norm} = (E_{MSE,C} - E_{MSE,B}) / (E_{MSE,A} - E_{MSE,B}). \quad (24)$$

Thus, values of  $E_{norm}$  around 0 can be interpreted in that way that the network is able to detect the causality relation between transitions from A and C. Fig. 8 results for different network sizes and delays. Each line represents different the performance of one network and different delays. The network sizes are 10,20,30,50,100,150, and 200 neurons in the hidden layer.

The value of  $E_{total}$  (cf. Eq. 23)) shows a very fast convergence to the final range almost immediately after the learning starts.

## 6 .Discussion

In the case of DBNs we investigated several ways to insert continuous input to the network and in this way to retrieve samples with correct probability distributions. The results are promising though we were not able to exactly achieve the correct distribution ranges. In the case of ESN we demonstrated an approach able to detect causalities in time series by using the mean square error of an ESN on-line learning procedure. Our current results indicate that the ability of this approach is limited to a few cycles–equivalent some dozens of steps. It can be expected that our results can be further improved by adapting the reservoir to the stimulus statistics. It would be very good if both approaches could be connected in order to combine virtues of both ESNs and DBNs.

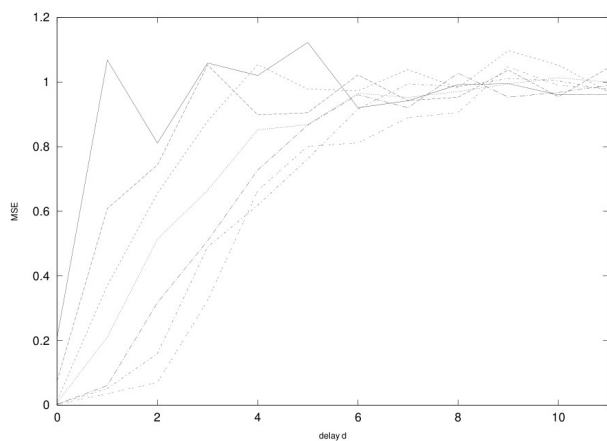


Fig. 9. Normalised errors for different network sizes at the hidden layer as a function of the delay. Full line 10 neurons, dashed 20 neurons, small dashes 30 neurons, dotted 50 neurons, dash-dotted 100 neurons, double dashed 150 neurons, and small dash-dotted 200 neurons.

[1] S. Thun, W. Burgard, D. Fox, Probabilistic robotics, MIT Press, 2006

[2] L-W. Lu, C-H. Lee, N. M. Mayer, Combining Deep Believe Networks with Domestic Service Robot, The 43<sup>rd</sup> Intl. Symp. on Robotics (ISR2012), Taipei, Taiwan, Aug. 29-31, 2012 (in preparation)

[3] Norbert Michael Mayer, Oliver Obst, Chang Yu-Chen, Time Series Causality Inference Using Echo State Networks. Latent Variable Analysis and Signal Separation - 9th International Conference, LVA/ICA 2010, St. Malo, France, September 27-30, 2010. Proceedings; 01/2010

[4] Hinton, G. E., Osindero, S. and Teh, Y., "A fast learning algorithm for deep belief nets," Neural Computation 18, pp 1527-1554., 2009.

[5] THE MNIST DATABASE of handwritten digits, Yann LeCun, Corinna Cortes, <http://yann.lecun.com/exdb/mnist/>

[6] Ranzato, M., Krizhevsky, A. and Hinton, G. E., "Factored 3-way restricted Boltzmann machines for modeling natural images," Proc. Thirteenth International Conference on Artificial Intelligence and Statistics, 2010.

[7] THE NORB DATASET of 3D object recognition from shape, Fu Jie Huang, Yann LeCun, Courant Institute, New York University, 2004, <http://www.cs.nyu.edu/~ylclab/data/norb-v1.0/>

[8] Mohamed, A. R., Dahl, G. E. and Hinton, G. E., "Deep belief networks for phone recognition," NIPS 22 workshop on deep learning for speech recognition, 2009.

[9] Deng, L., Seltzer, M., Yu, D., Acero, A., Mohamed A. and Hinton, G., "Binary Coding of Speech Spectrograms Using a Deep Auto-encoder," Interspeech, Makuhari, Chiba, Japan, 2010.

[10] Carnegie Mellon University, The cmu sphinx group open source speech recognition engines, <http://cmusphinx.sourceforge.net/html/cmusphinx.php>

[11] Qi is a tool for rapid prototyping of integrated behavior systems on a team of distributed embedded systems we developed.

[12] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," Neural Computation, vol. 14, pp. 1771-1800, 2002

[13] K.-F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden markov models," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 37, no. 11, pp. 1641-1648, 1989.

[14] G. E. Hinton, "A practical guide to training restricted Boltzmann machines", 2010

[15] Arel, I.; Rose, D.C.; Karnowski, T.P.; , "Deep Machine Learning - A New Frontier in Artificial Intelligence Research [Research Frontier]," Computational Intelligence Magazine, IEEE, vol.5, no.4, pp.13-18, Nov. 2010, doi: 10.1109/MCI.2010.938364

[16] Geoffrey E. Hinton (2009), Scholarpedia, 4(5):5947. [http://www.scholarpedia.org/article/Deep\\_belief\\_networks](http://www.scholarpedia.org/article/Deep_belief_networks)

[17] IEEE Computer Society (August 29, 2008), IEEE Standard for Floating-Point Arithmetic, IEEE, doi:10.1109/IEEESTD.2008.4610935, IEEE Std 754-2008

[18] T. Lee and D. Mumford, "Hierarchical Bayesian inference in the visual cortex," J. Opt. Soc. Amer., vol.20, pt.7, pp.1434-1448, 2008

[19] Joschka Boedecker, Oliver Obst, N. Michael Mayer, and Minoru Asada. Initialization and self-organized optimization of recurrent neural network connectivity. HFSP J., 3(5):340-349, 2009. doi: 10.2976/1.3240502.

[20] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. Econometrica, 37(3):424-438, 1969.

[21] Barbara Hammer, Benjamin Schrauwen, and Jochen J. Steil. Recent advances in efficient learning of recurrent networks. In ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning, pages 213-226, 2009.

[22] H. Jaeger. The 'echo state' approach to analysing and training recurrent neural networks. In GMD Report 148, GMD German National Research Institute for Computer Science, 2001.

[23] Herbert Jaeger. Adaptive nonlinear systems identification with echo state networks. In Advances in Neural Information Processing Systems; Proceedings of the NIPS 15, pages 609-615, 2003. AA14.

[24] E. Pereda, R. Quiñero, and J. Bhattacharya. Nonlinear multivariate analysis of neurophysiological signals. Progress in Neurobiology, (77):1-37, 2005.

[25] Thomas Schreiber. Measuring information transfer. Physical Review Letters, 85(2):461-464, July 2000.

[26] Takashi Shibuya, Tatsuya Harada, and Yasuo Kuniyoshi. Causality quantification and its applications: structuring and modeling of multivariate time series. In KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 787-796, New York, NY, USA, 2009. ACM.

# RoboCup 小型ロボットリーグにおけるキック検出手法の検討

A new detection method of kick actions from logged data of SSL games

安井興太郎、村上和人、成瀬正

Kotaro Yasui, Kazuhito Murakami and Tadashi Naruse

愛知県立大学 情報科学研究科

Graduate School of Information Science and Technology,

Aichi Prefectural University

im121007@cis.aichi-pu.ac.jp, {murakami, naruse}@ist.aichi-pu.ac.jp

## Abstract

This paper proposes a new algorithm that detects line segments of the ball trajectories, whose purpose is to identify the behaviors of the opponent robots, in the logged data of the small size league games. Line segments are important because they appear in the kick actions. We have already proposed such an algorithm in [1]. It was based on the smoothing technique to reduce the effects of noise. In the algorithm, there were some weaknesses such as detecting two distinct line segments as one line segment. In the new algorithm, we adopt two levels of processing, a line element detection and a connection of the line elements, instead of using the smoothing processing, and realize the high detection rate. Experimental results show that 93.4 % of straight lines are detected correctly compared with 63.1 % detection of the old one.

## 1 はじめに

昨今のRoboCup 小型ロボットリーグでは、主に「ボールの位置・速度」「味方ロボットの位置・速度」「敵ロボットの位置・速度」「レフェリー信号」の4つの情報をもとに、自チームの戦略を決定している。一方、人間のサッカーでは、各プレイヤーは「ボールの位置・速度」「味方プレイヤーの位置・速度」「敵プレイヤーの位置・速度」「審判のコール」の4つの情報に加えて、味方・敵が過去にどのような行動をしたのかという、「敵・味方両プレイヤーの過去の行動情報」を認知した上で、これらをもとに、行動の決定を行っている。

そこで、ロボットのサッカーでも、人間と同じように、味方や敵の過去の行動を認識し、それによって戦略を変えることができれば、より効果的な戦略でゲームを支配していくことができると考えられる。過去の行動に基づいた戦略の決定を行うためには、SSL-Vision[2]から送られてきた情報をもとに、キックやパス、シュートなどの基本的な行動の検出を行う必要がある。本論文では、とくにキックの検出を行うための手法について検討する。

キック検出を行うためには、SSL-Vision から一定間隔で送られてくるボールの位置情報をもとに、ボールがどのフレームからどのフレームまで、直線運動をしたかを検出することが必要である。直線検出としては、ハフ変換を用いた手法[3]が代表的であるが、これは与えられた静止画像中に実際にある直線を検出するというものであり、与えられた複数の点からボールが直線運動をしている部分を抜き出す、という今回の問題とは性質が異なる。一方、与えられた複数の点から1つの直線を当てはめる手法としては、最小二乗法[4]が代表的であるが、あてはめ対象となる点の集合を特定することに対して、最小二乗法は解答を与えない。[5]では人間のサッカーの試合映像から、ボールを検知し、パスやシュートを検出するアルゴリズムが提案されているが、詳しい手法については記述されておらず、この手法がRoboCup 小型ロボットリーグにおいても通ずるものであるかは確認することができない。したがって、直線運動の検出を行う新しいアルゴリズムを考える必要がある。

我々は2008年に既に直線運動の検出を含む、独自のキック検出手法を提案している[1]。しかし、この既存手法には問題点があったため、検出精度の向上を目的として、本論文で改良手法を提案する。

## 2 キック検出

### 2.1 定義

30fps あるいは 60fps(これはカメラの性能による)で SSL-Vision から送信されてくるログデータには、ロボットとボールの位置情報が格納されている。ここで述べるキックの検出とは、静止または移動しているロボットが、静止または移動しているボールをキックし、ボールが直線運動を始めたフレームと、ボールが直線運動を終えたフレームの 2 点を、このログデータから求めるものである。本論文では、チップキックについては考察の対象としない。

### 2.2 既存手法

我々が既に提案しているキック検出手法[1]では、1) ボールの直線運動を検出し、2) 検出された直線運動がキックによって起こされたものであるか否かを判定する、という 2 つの段階からなる。

#### 2.2.1 1) 直線運動検出アルゴリズム

フレーム  $i$  でのボールの二次元座標を  $B_i$  とする。 $B_i$  を始点とし、 $B_{i-1}$  を終点とするベクトルを  $\vec{a}_i$ 、 $B_i$  を始点とし、 $B_{i+1}$  を終点とするベクトルを  $\vec{b}_i$  として、ボールの移動角度差  $\theta_i$  を eq.(1) のように定義する。

$$\theta_i = \left| \pi - \cos^{-1} \frac{\vec{a}_i \cdot \vec{b}_i}{|\vec{a}_i| |\vec{b}_i|} \right| \quad (1)$$

次に、フレーム  $i$  からフレーム  $i+n-1$  の間におけるボールの区間平均速度  $v_i$  とボールの平均移動角度差  $\theta_i$  を平滑化のためのフレーム数  $n$  を使って eq.(2) のように定義する。

$$\bar{v}_i = \frac{1}{n} \sum_{m=i}^{i+n-1} v_m, \quad \bar{\theta}_i = \frac{1}{n} \sum_{m=i}^{i+n-1} \theta_m \quad (2)$$

$v_m$  は、フレーム  $m$  におけるボールの移動速度 [mm/s] である。

#### 直線検出アルゴリズム

Step0  $i \leftarrow 1$  とする。

Step1  $i$  が終了フレーム番号の場合、計算を終了する。そうでない場合は eq.(3) を計算する。

$$\{\bar{\theta}_i \leq f(\bar{v}_i)\} \wedge \{\bar{v}_i \geq T_1\} \quad (3)$$

$f$  は  $\bar{v}_i$  が高くなるほど低い値を返す関数で ( $B = f(A)$  のとき  $B \propto A^{-1}$ )、その値を閾値に用いる。 $T_1$  はあらかじめ与えられた閾値である。eq.(3) が偽ならば、 $i \leftarrow i+1$  として Step1 を繰り返す。eq.(3) が真ならば、フレーム  $i$  が直線運動開始フレームである。 $i \leftarrow i+1$  として、Step2 へ。

Step2 eq.(4) を計算する。

$$\{\bar{\theta}_i \leq f(\bar{v}_i)\} \wedge \{\theta_{i+n} \leq T_2\} \quad (4)$$

$f$  は eq.(3) における  $f$  と同じ関数である。 $T_2$  はあらかじめ与えられた閾値である。eq.(4) が真ならば、 $i \leftarrow i+1$  として Step2 を繰り返す。eq.(4) が偽ならば、フレーム  $i+n$  が直線運動終了フレームである。得られた直線を出力し、 $i \leftarrow i+1$  として、Step1 へ戻る。

#### 2.2.2 2) キック判別アルゴリズム

次のようなアルゴリズムで、検出されたそれぞれの直線運動がキックによって起こされたものであるか否かを判別する。

##### キック判別アルゴリズム

直線運動開始フレームを  $s$  とする。 $s$  フレーム目において、ボールに最も近接しているロボット  $R_s$  を求める。また、 $k$  フレーム目における  $R_s$  とボールとの距離を  $D_k$  とする。eqs.(5), (6) を計算する。

$$\{D_s \leq T_3\} \wedge \{(D_s < D_{s+1}) \wedge \dots \wedge (D_{s+n-1} < D_{s+n})\} \quad (5)$$

$$\max_{0 \leq k \leq n-2} \{(D_{s+k+2} - D_{s+k+1}) - (D_{s+k+1} - D_{s+k})\} \geq T_4 \quad (6)$$

$T_3, T_4$  はあらかじめ与えられた閾値である。

eqs.(5), (6) を共に満たす時、この直線運動はロボット  $R_k$  のキックによって起こされたものと判定する。eq.(5) を満たすが、eq.(6) を満たさない場合は、この直線運動はロボット  $R_k$  による跳ね返りによって起こされたものと判定する。どちらも満たさない場合は不明な直線運動と判定する。

#### 2.2.3 既存手法の問題点

この手法を提案した浅野は、90%強の高い精度で正しくキック検出を行うことができると主張している。しかしながら、当該論文[1]には、1) の段階で 1 本の直線運動をノイズ等の影響で分割された複数の直線運動として検出してしまったものに関してどういう扱いをするか、ということについては何の記述もない。このようなケースを誤検出と考えて追試実験を行った場合、精度が 50%弱に落ちってしまった。よって、当該論文ではこのようなケースも正しい検出結果と判定していた可能性が高いと予想できる。検出した直線運動情報をもとにした行動分析、及び新戦略の決定の際に、直線運動が分割されてしまっていると、情報として扱いつらく、不都合が生じることは容易に想像できる。そこで、このようなケースを誤検出と考えても高い精度が得られるよう、既存手法のステップ 1 の部分に改良が必要である。既存手法では、平滑化フレーム  $n$  を大きくすることでノイズの影響を小さくできるため、ノイズ

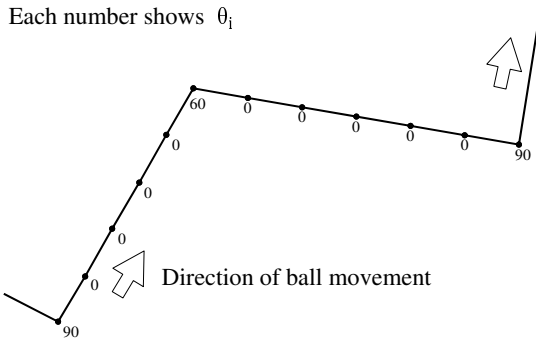


Figure 1: 移動角度差  $\theta_i$  の一例

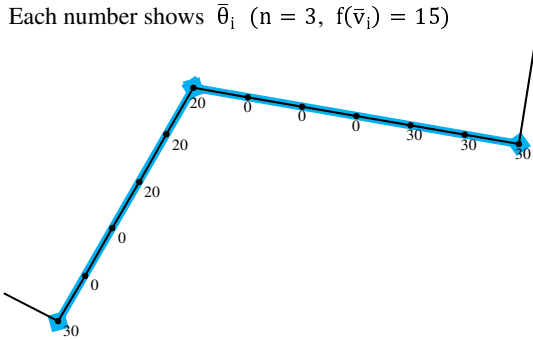


Figure 2: Fig.1 から計算した  $n = 3$  の時の平均移動角度差  $\bar{\theta}_i$  と検出される直線運動

による直線の分割を抑えることができる。しかしながら、 $n$  が大きいと、次のような新しい問題が発生する。

今、ボールが Fig.1 のような軌道で左から右へ 90 度、60 度、90 度の順に 3 回曲がって動いたと仮定する。点は各フレームのボールの位置であり、点の隣の数値は eq.(1) で定義したボールの移動角度差  $\theta_i$  である。この場合、平滑化フレーム数  $n = 3$  として eq.(2) の平均移動角度差  $\bar{\theta}_i$  を求めると、Fig.2 のようになる。同様に、 $n = 5$  として  $\bar{\theta}_i$  を求めると Fig.3 のようになる。

「平均移動角度差  $\bar{\theta}_i$  が 15 度以下の場合には直線運動をし

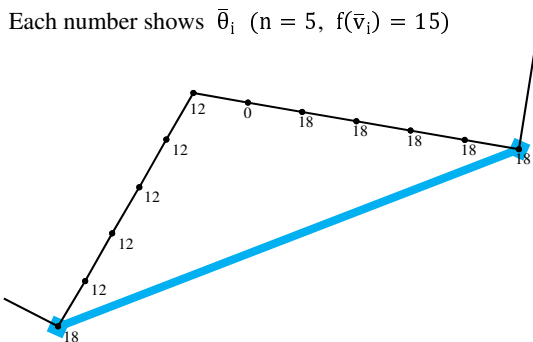


Figure 3: Fig.1 から計算した  $n = 5$  の時の平均移動角度差  $\bar{\theta}_i$  と検出される直線運動

ている」と仮定しよう。Fig.2 では、 $\bar{\theta}_i$  が 15 度より大きい 20 度、30 度の部分 (2 回目、3 回目に曲がった部分) で終点が見つかるため、直線は正しく 2 本検出される。しかし、Fig.3 では、2 回目に曲がった部分の  $\bar{\theta}_i$  が 15 度以下の 12 度であるため、直線運動がまだ続いているとみなされ、2 本の直線が 1 本の直線となって検出されてしまう。

これは、直線運動中か、そうでないかを識別するための特徴量である移動角度差  $\theta_i$  の平滑化を行ってしまったために、特徴量の変化が小さくなってしまい、終点を検出できなかったからである。

このような問題が生じるために  $n$  を大きくすることは検出精度を上げることの根本的な解決にはならない。加えて、この手法では  $n$  フレーム以上続いている直線運動しか検出できないので、 $n$  は小さいほど短い直線運動を検出でき、ノイズの影響を無視できるならば平滑化は行わないほうが良い。

そこで、移動角度差  $\theta_i$  の平滑化は行わずに、既存手法と同じような方法でより確実に直線運動をしている部分のみをまず検出し、その後でノイズで分割されてしまった直線運動を結合する処理を行う、というプロセスで、問題解決を行う。

### 2.3 新たに提案する直線運動検出手法

ボールが直線運動をしているかどうかの判定式として、eqs.(3), (4) の代わりに、eq.(7) を定義する。

$$\{\theta_i \leq T_5\} \wedge \{v_i \geq T_6\} \quad (7)$$

$T_5, T_6$  はあらかじめ与えられた閾値である。既存手法とは異なり、移動角度差  $\theta_i$  の平滑化を行わないようにしている。また、Fig.4 のように、ボールの速度が遅いと、直線運動をしていてもノイズの影響で移動角度差  $\theta_i$  が大きくなることもあるため、既存手法では eqs.(3), (4) で閾値関数  $f$  を使用していたが、新手法ではより確実に直線運動をしている部分のみを検出するために、 $f$  を使用しないようにしている。

直線運動  $L$  の開始フレーム番号を  $L(s)$ 、終了フレーム番号を  $L(e)$  とする。 $B_{L(s)}$  を始点とし、 $B_{L(e)}$  を終点とするベクトルを  $\vec{L}$  と書くことにする。最後に検出した直線運動  $L_{last}$  が、 $n$  番目に検出した直線運動  $L_n$  と同一の直線運動かどうかを判定する式として、eq.(8) を定義する。

$$\{L_{last}(s) - L_n(e) \leq T_7\} \wedge \left\{ \cos^{-1} \frac{\vec{L}_{last} \cdot \vec{L}_n}{|\vec{L}_{last}| |\vec{L}_n|} \leq T_8 \right\} \quad (8)$$

$T_7, T_8$  はあらかじめ与えられた閾値である。左側の条件式はボールが直線運動  $L_n$  をした後、直線運動  $L_{last}$  をするまでにかかったフレーム数が  $T_7$  以下であるかどうかを判定する式、右側の条件式は直線運動  $L_n, L_{last}$  の直線の角度のずれが  $T_8$  以下であるかどうかを判定する式である。



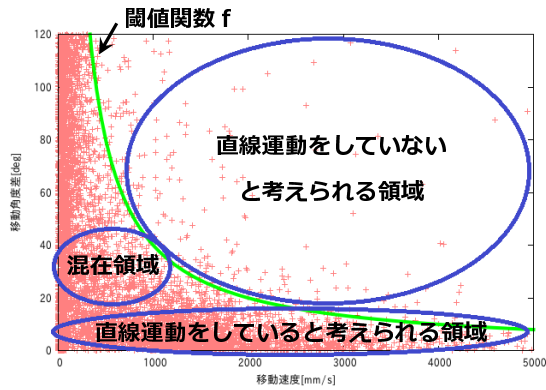


Figure 4: ある試合中のボールの移動速度  $v_i$  と移動角度差  $\theta_i$  の分布

### 新しく提案する直線運動検出アルゴリズム

**Step0**  $i \leftarrow 1, N \leftarrow 0$  とする。  $N$  は検出した直線運動の総本数である。

**Step1**  $i$  が終了フレーム番号の場合、計算を終了する。 eq.(7) が偽ならば、  $i \leftarrow i + 1$  として Step1 を繰り返す。 eq.(7) が真ならば、フレーム  $i - 1$  が直線運動開始フレームである。  $i \leftarrow i + 1$  として、Step2 へ。

**Step2** eq.(7) が真ならば、  $i \leftarrow i + 1$  として Step2 を繰り返す。 eq.(7) が偽ならば、フレーム  $i$  が直線運動終了フレームである。  $i \leftarrow i + 1$  として、Step3 へ。

**Step3** 今までに検出した直線運動  $\{L_0, \dots, L_{N-1}\}^1$  のうち、 eq.(8) を満たす直線運動  $L_n$  が存在する場合は、直線運動  $L_n$  と  $L_{\text{last}}$  は同一の直線運動と判定し、  $L_{\text{last}}(s) \leftarrow L_n(s)$  として直線運動の結合処理を行って、  $L_n \sim L_{N-1}$  の検出直線運動を削除するために  $N \leftarrow n$  とする。

**Step4**  $L_N \leftarrow L_{\text{last}}, N \leftarrow N + 1$  とする。 Step1 へ戻る。

## 3 実験

既存手法と新手法で、2011年世界大会の準々決勝戦 RoboDragons vs Skuba の試合中に起こったボールの直線運動を検出し、その精度を比較する実験を行った。各手法には、パラメータを与える必要がある。実験には、次の値を用いた。

#### 既存手法

- 平滑化フレーム数  $n = \{1, 3, 5\}$  とする
- 閾値関数  $f$  については  $\bar{\theta}_i = f(\bar{v}_i) = 40000/\bar{v}_i$  [degree] で固定する
- $T_1$  については 500 [mm/s] で固定する
- $T_2 = \{15, 25, 35\}$  [degree] とする

#### 新手法

<sup>1</sup>  $N = 0$  の場合、  $\{L_0, \dots, L_{N-1}\}$  は空集合  $\phi$  とする

Table 1: 検出直線運動の分類

A	検出直線運動のうち、正解直線運動と等しいもの本数
B	検出直線運動のうち、正解直線運動の(全体ではない)一部分であるもの、または1本の正解直線運動が複数本に分かれて検出されたもの本数
C	検出直線運動の(全体ではない)一部分が正解直線運動の全体あるいは一部分であるもの本数
D	正解直線運動のうち、検出直線運動にないもの本数
E	検出直線運動のうち、正解直線運動にないもの本数
F	除外本数(非常に低速な直線運動、あるいはチップキックでボールが空中に浮いている状態での直線運動の本数)

- $T_5$  については、15 [degree] で固定する
- $T_6$  については 500 [mm/s] で固定する
- $T_7 = \{10, 15, 20\}$  とする
- $T_8 = \{10, 15, 20\}$  [degree] とする

### 3.1 評価

RoboDragons のログデータ再生ソフト simulator[6]を用いて、試合中のボールの動きを観測したとき、評価者によって検出した直線運動を正解直線運動と呼ぶことにする。既存手法、新手法のそれぞれについて、検出した直線運動を Table.1 のようにクラス A ~ F に分類する。求めた A ~ F の本数をもとに、次のような適合率  $P$ 、再現率  $R$ 、F 値を計算、これを用いて評価する。

$$\text{適合率 } P = \frac{A + 0.5 \cdot B + 0.25 \cdot C + 0 \cdot E}{A + B + C + E} \quad (9)$$

$$\text{再現率 } R = \frac{A + 0.5 \cdot B + 0.25 \cdot C + 0 \cdot D}{A + B + C + D} \quad (10)$$

$$F \text{ 値} = \frac{2PR}{P + R} \quad (11)$$

適合率  $P$  は、検出直線運動のうち、正解直線運動の占める割合を意味し、再現率  $R$  は、正解直線運動のうち、検出直線運動の占める割合を意味する。  $F$  値は適合率  $P$  と再現率  $R$  の調和平均であり、  $P$  と  $R$  が共に高い時に、一番高い値をとる。こういった情報検索の問題は、適合率をあげれば再現率が下がり、再現率を上げれば適合率が下がる傾向にあるため、有効な評価尺度である。



Table 2: 既存手法実験結果 (1)

$n$	1	1	1	3	3	3	5	5	5
$T_2$	15	25	35	15	25	35	15	25	35
$A$	3	13	14	13	21	23	7	16	18
$B$	52	42	41	33	25	21	35	22	19
$C$	0	0	0	6	6	8	8	12	12
$D$	2	2	2	5	5	5	7	7	7
$E$	1	3	4	1	2	4	3	7	10
$F$	594	409	330	304	205	156	310	209	158

Table 3: 既存手法実験結果 (2)

$P$ $R$ $F$ 値	$T_2 = 15$	$T_2 = 25$	$T_2 = 35$
$n = 1$	0.518	0.586	0.585
	0.509	0.596	0.605
	0.513	0.591	0.595
$n = 3$	0.585	0.648	0.634
	0.596	0.614	0.623
	0.564	0.631	0.628
$n = 5$	0.500	0.526	0.517
	0.465	0.526	0.535
	0.482	0.526	0.526

Table 4: 新手法実験結果 (1)

$T_7$	10	10	10	15	15	15	20	20	20
$T_8$	10	15	20	10	15	20	10	15	20
$A$	46	46	49	49	51	52	50	51	52
$B$	9	8	4	6	3	1	4	3	1
$C$	0	1	2	0	1	2	1	1	2
$D$	2	2	2	2	2	2	2	2	2
$E$	1	1	2	1	1	2	1	1	3
$F$	98	83	73	98	82	70	81	82	66

### 3.2 実験結果

既存手法によって検出した直線運動のクラス分類結果を Table.2 に示す。Table.2 から求めた適合率  $P$ 、再現率  $R$ 、 $F$  値を Table.3 に示す。

次に、新手法によって検出した直線運動のクラス分類結果を Table.4 に示す。Table.4 から求めた適合率  $P$ 、再現率  $R$ 、 $F$  値を Table.5 に示す。

### 3.3 考察

既存手法に関しては、次のようなことが分かった。

- $n$  あるいは  $T_2$  が小さい時に  $B$  だった直線運動が、 $n$  あるいは  $T_2$  を大きくすることで  $A$  になっているものが多くあった。反面、 $A$  や  $B$  が  $C$  や  $E$  になってしまったものも多々あった。これは、 $n$  あるいは  $T_2$  を大きくすることでノイズに強くなって一つの直線の分割を防ぐことができるが、二つの異なる直線を同一の直線と誤検出することも多くなるためである
- Table.2 から分かるように、 $n$  が大きいほど、 $D$  が増えた。これは、既存手法では  $n$  フレーム以上続いている直線運動しか検出できないため、 $n$  が大きい

Table 5: 新手法実験結果 (2)

$P$ $R$ $F$ 値	$T_8 = 10$	$T_8 = 15$	$T_8 = 20$
$T_7 = 10$	0.902	0.897	0.904
	0.886	0.882	0.904
	0.894	0.889	0.904
$T_7 = 15$	0.929	0.942	0.930
	0.912	0.925	0.930
	0.920	0.934	0.930
$T_7 = 20$	0.933	0.942	0.914
	0.917	0.925	0.930
	0.925	0.934	0.922

と、短い直線運動を検出できないためである

- 実験を行った中で、既存手法で最も良かったパラメータの組み合わせは、 $n = 3, f = 40000/\bar{v}_i, T_1 = 500, T_2 = 25$  であり、この時の  $F$  値は 0.631 であった

新手法に関しては、次のようなことが分かった。

- $T_7$  あるいは  $T_8$  が小さいほど、分割された直線運動を結合できない場合が多くなり、 $B$  が増えてしまった
- $T_7$  あるいは  $T_8$  が大きいほど、誤った結合をしてしまう場合が多くなり、 $C, E$  が増えてしまった
- 実験を行った中で、新手法で最も良かったパラメータの組み合わせは、 $T_5 = 15, T_6 = 500, T_7 = \{15, 20\}, T_8 = 15$  であり、この時の  $F$  値は 0.934 であった

既存手法と新手法を比較すると、tables 2 ~ 5 から、新手法では、どのパラメータ設定でも既存手法で最も高かった  $F$  値 0.631 を超えていることが分かる。また、新手法の最高  $F$  値 0.934 と既存手法の最高  $F$  値 0.631 との差は 0.303 で、48.0%向上しており、既存手法よりも新手法の方が、この問題に有効な手法であると言える。

### 3.4 キック検出への適用

改良したアルゴリズムで検出した直線運動に対して、section2 で述べたキック判定アルゴリズムを適用したところ、その精度は 60%弱であった。これは、[1] に記述されている検出精度 97.9%よりも低い。この主な理由として、

- ビジョンシステムが SSL-Vision に変わり、当時のものよりもオブジェクトの位置検出精度が落ちた
- SSL-Vision 対応のカメラに変えたことで、カメラの性能が落ち、オブジェクトの位置検出精度が落ちた

などが考えられる。我々は今後、これらの問題の解決に、積極的に取り組んでいく予定である。

## 4 おわりに

本論文では、相手行動の学習と目的に既に提案されている RoboCup 小型リーグにおけるキック検出手法の精度上の問題を改善するために、ボールの直線運動検出手法の改良を行った。新たに提案した手法では、以前の手法よりも高い精度で実際の試合中のボールの直線運動検出を行うことができ、新手法の有効性を確認した。

一方、改良した手法を用いて検出した直線運動に対して、キック判定アルゴリズムを適用したところ、既存手法提案時よりも、検出精度が落ちてしまった。今後は、キック判定アルゴリズムにも改良を施し、精度を上げることが求められる。また、これらの手法を用いて検出したロボットのキック行動情報をもとに、味方だけでなく、相手行動の分析を行い、より効果的な戦略の決定を行う方法を検討していくことも、大きな課題である。

## 謝辞

本研究は、愛知県立大学理事長特別研究費の支援を受けて行われた。記して感謝する。

## 参考文献

- [1] K. Asano, K. Murakami and T. Naruse, "Detection of basic behaviors in logged data in RoboCup Small Size League", RoboCup 2008: Robot Soccer World Cup XII, LNCS 5399 pp. 439-450, Springer, 2009
- [2] Stefan Zickler, Tim Laue, Oliver Birbach, Mahisorn Wongphathi and Manuela Veloso, "SSL vision: The Shared Vision System for the RoboCup Small Size League", RoboCup 2009: Robot Soccer World Cup XIII, Lecture notes in artificial intelligence, LNAI 5949, pp. 425 - 436, Springer, 2010
- [3] Hough, P.V.C. "Method and means for recognizing complex patterns", U.S.Patent No.3069654, 1962
- [4] Derek York, "Least squares fitting of a straight line", Canadian Journal of Physics, 44, 1079-1086, 1966
- [5] X. Yu, C. Xu, H. W. Leong, Q. Tian, Q. Tang, and K. Wan, "Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video", in Proc. ACM Int. Conf. Multimedia, 2003
- [6] Stefan Zickler, James Bruce, Joydeep Biswas, Michael Licitra, and Manuela Veloso "CMDragons 2009 Extended Team Description", <http://www.cs.cmu.edu/~mmv/papers/09robocup-cmdragons.pdf>

## 不均一なモジュールの組み合わせによるロボットの構築 Robot construction by combination of non-uniform modules

足立 匠 (中部大学大学院工学研究科)  
長坂 保典 (中部大学工学部)  
藤井 隆司 (中部大学全学共通教育部)  
佐伯 守彦 (中部大学工学部)  
柴田 祥一 (中部大学工学部)  
岡崎 明彦 (中部大学工学部)

\* Takumi ADACHI(Chubu Univ.), Yasunori NAGASAKA(Chubu Univ.),  
Takashi FUJII(Chubu Univ.), Morihiko SAEKI(Chubu Univ.),  
Shoichi SHIBATA(Chubu Univ.), Akihiko OKAZAKI(Chubu Univ.)

[te11002-7372@sti.chubu.ac.jp](mailto:te11002-7372@sti.chubu.ac.jp), [any@isc.chubu.ac.jp](mailto:any@isc.chubu.ac.jp), [fujii@cs.chubu.ac.jp](mailto:fujii@cs.chubu.ac.jp), [saeki@isc.chubu.ac.jp](mailto:saeki@isc.chubu.ac.jp),  
[shibata@isc.chubu.ac.jp](mailto:shibata@isc.chubu.ac.jp), [okazaki@cs.chubu.ac.jp](mailto:okazaki@cs.chubu.ac.jp)

*Abstract*—We present robot construction concept by introducing combination of non-uniform functional robot modules. It has two merits, the higher availability and reliability come from modular system, and the higher adaptability and wide range of available work come from problem specific module design. We defined several type of modules. The control module has two functions. They are communication with outside of the robot, and sending commands to other modules and controlling them as a robot brain. The locomotion module gives a locomotive function to the robot. The arm module works like arms or legs of animals. It can mount other modules on the top of itself. The "other" module includes any other functional modules, like sensors, actuators, special devices. We built three different prototype robots as combination of modules. They have common body and different locomotion, four omni-wheel drive mechanism, two wheel drive mechanism, biped walking. In experiments we investigated whether those robots can be controlled by single series of commands. As a result, we confirmed that each robot moved correctly by single series of commands.

### 1. はじめに

一般にロボットを設計する際、その制御方法により、集中型制御と分散型制御に分類される。集中型制御は一つの高速なプロセッサでロボット全体の制御をしており、その特徴は各制御対象を直接制御するため、複雑な動作や特殊な機材を扱えることにある。しかし、全体を制御しているメインシステムに障害が発生した場合、全機能が停止する可能性がある。一方、分散型制御方法は、ロボットの制御を複数の構成要素のプロセッサに分散して一つの処理を行う方式であり、複数のシステムが分かれているため、一つのシステムが機能停止した場合でも全体に及ぼす影響は低い。しかし、一定の共通な規格が決められており、全てのシステムがその規則に準じていなければならないという欠点を持つ。また、複数のシステムを管理するためのシステムが必要となる。分散型制御を利用したロボットは多く存在する。

一般に、モジュール型ロボットと呼ばれるロボットも分散型制御を採用したロボットであり、代表的なロボットとして産業技術総合研究所の M-TRAN[1, 2] や Lund 氏が開発した ATRON[3] が挙げられる。これらのロボットは均一なモジュールで構成されており、個々のモジュールにアクチュエータが搭載されている。そして、このモジュールを共通化された接続部で連結し様々な形状・動作を可能にしている。

また、制御部も個々のモジュールに搭載されているため、一部のモジュールにトラブルが発生しても、システム全体への影響は少ない。その場合、正常なモジュールと交換することにより修復することが可能である。しかし、均一なモジュールだけで構成されたモジュール型ロボットは、モジュールの機能が限定されるため、可能な作業は集中型制御と比較すると少なく、動作のパターンも限られてしまう。それを補うように、モジュールに様々な機能を搭載した場合、モジュール自身の巨大化やシステムが複雑化してしまう。

本報告では分散型制御でありながら均一なモジュールではなく、不均一なモジュールで構成されたモジュールロボットシステムを提案する。均一なモジュールでは、先に述べた問題により、特殊な機能をモジュールに搭載することはできないが、不均一にすることによりそれぞれ独自の機能を搭載することが可能となる。機能ごとにモジュールを製作することにより、状況に応じて、必要なモジュールだけでロボットを構成することが可能となり、現場の環境や課題に対し、優れた適応が図れると考える。また、モジュール型を採用することにより、トラブル発生によるシステム全体への影響を最小限に抑えると共に、独自の機能を搭載したモジュールの組み合わせによる現場や環境への高い適応度を実現することを目的とする。実証実験として本報告では基本的なモジュールに移動モジュールとロボットアームを使用して RoboCup サッカーのサッカーロボットに適用する。

実験では同じ命令系を使用して異なる構成のロボットを動作させることにより、不均一なモジュールの組み合わせによるロボットの構築が可能かを検証する。

## 2. モジュール

分散型制御により、それぞれの機能単位毎にマイコンやアクチュエータが組み込まれている。これらを構成する1つの集合をモジュールとする。

モジュールは大きく分けて、コアモジュールとエンドモジュールに分けられる。コアモジュールは、システム全体を管理するモジュールであり、エンドモジュールは実際に作業をするモジュールを指す。

全体の構成は Figure 1 に示すように、一つのコアモジュールに、複数のエンドモジュールを接続する。

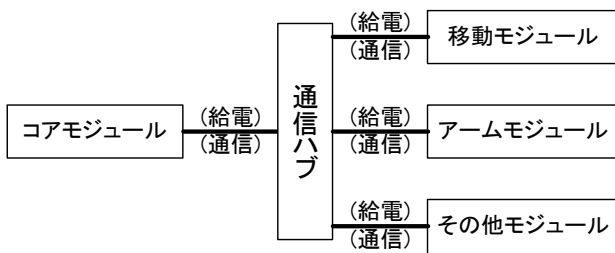


Figure 1: Structure of whole robot

一つのコアモジュールへの最大接続数は、給電される電源の容量とコアモジュールに搭載されたマイコンの処理能力により決定される。今回使用するコアモジュールは電源の容量が小さく、システムを単純化することから、接続数の上限を 10 モジュールとする。

### 2-1. コアモジュール

コアモジュールは、他のモジュールを管理するためのモジュールであり、ホストから受信する命令を判断して、各モジュールへ送信する役割をもつ。コアモジュールの構成を Figure 2 に示す。

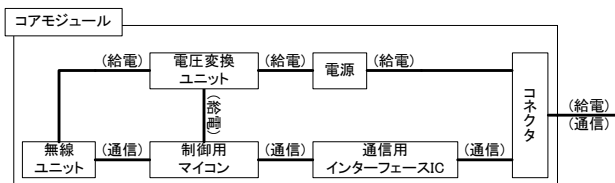


Figure 2: Structure of core module

コアモジュールは、ホストと通信するための無線ユニットが搭載されており、制御用マイコンへと接続されている。また、エンドモジュールと通信するための通信用インターフェース IC が搭載されており、通信ハブを経由してエンドモジュールへ通信や給電を行う。

### 2-2. エンドモジュール

エンドモジュールは Figure 3 に示すように、ベースユニットと制御対象（モータ、センサなど）から構成される。

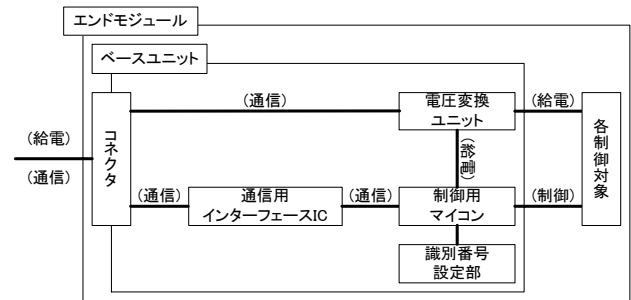


Figure 3: Common controller of modules

ベースユニットは、コアモジュールと通信するための通信用インターフェース IC、エンドモジュールを制御する制御用マイコン、個々を判別するための識別番号設定部、給電される電源の電圧を変換し、マイコンなどに供給する電圧変換ユニットで構成される。本報告ではエンドモジュールの例として移動モジュール（歩行型、4 輪型、2 輪型の 3 種類）、アームモジュール、その他モジュールを製作した。

#### ・移動モジュール

移動モジュールはコアモジュールから命令を受信し、ロボット全体を指定された目的地まで移動するためのモジュールである。歩行型移動モジュールの構成を Figure 4 に示す。

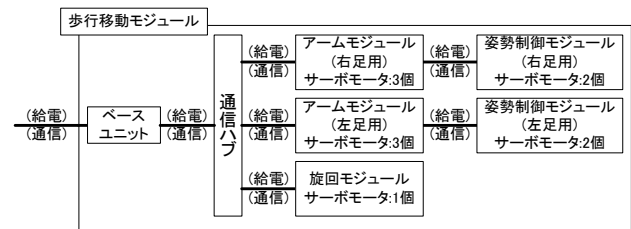


Figure 4: Structure of locomotion module

歩行型移動モジュールは 2 つのアームモジュールとその先端に姿勢制御モジュール、移動モジュールの上部に設置されているモジュールを旋回させるための旋回モジュールで構成される。本実験では、移動モジュール全体の構成を簡易化するため、これらモジュール群を 1 つの移動モジュールとした。これにより歩行型移動モジュールは、ベースユニット部分 1 つと 11 個のサーボモータとなる。

#### ・アームモジュール

アームモジュールは、主にその先端に取付けた別のモジュールを指定された目的地まで移動させるモジュールである。Figure 5 にアームモジュールの構成を示す。

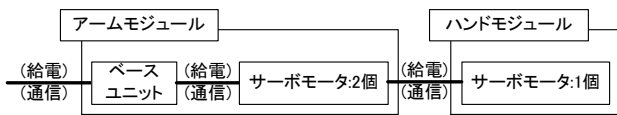


Figure 5: Structure of arm module

アームモジュールはベースユニットとアームの駆動を行うサーボモータ 2 個で構成されている。アームの先端には、その他のモジュールが接続可能となっており、給電や通信方式はすべて同じ規格となっている。本実験では簡単な構造のハンドモジュールが接続されており、ベースユニットは簡略化されているが、複雑な機構のモジュールを接続する場合は、ベースユニットとそれぞれのデバイスを接続する必要がある。

• その他のモジュール

その他のモジュールは、前で述べたモジュール以外のモジュールを指しており、コアモジュールを通して外部からの固有の命令によって動作するモジュールである。本報告では、その他モジュールとして、ソレノイドのプッシュ機構を利用したキックデバイスモジュールを用いる。このモジュールは RoboCup サッカー小型ロボットリーグの 4 輪型ロボットでも使用されているキックデバイスである。Figure 6 にキックデバイスモジュールの構成を示す。

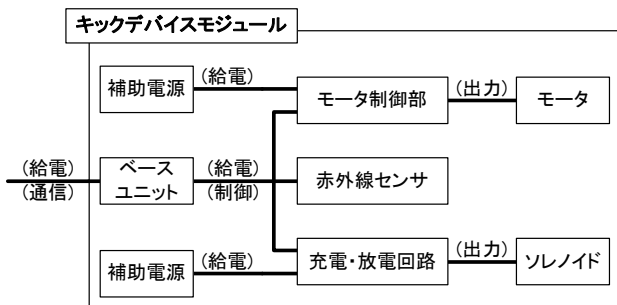


Figure 6: Structure of kick device module

キックデバイスモジュールが制御するデバイスは、ボールを保持するホールディングデバイスのモータ、ボールが保持されたかを検出するための赤外線センサ、ボールをキックするためのソレノイドを駆動する充電・放電回路の 3 つで構成される。このデバイスはコアモジュールから供給される電源のみでは必要な電力を賄うことができないため、別途専用の補助電源を搭載している。それぞれの電源供給はマイコンが管理しており、コアモジュールを通して外部操作が可能となっている。

3. モジュールの動作

コアモジュールはシステムの管理および、受信した命令を判断しエンドモジュールに伝送する。エンドモジュールはモジュールの種類ごとに決められた抽象的な共通命令を実行する。本報告では、ロボットの移動命令とアームへの命令を用いる。ロボット

の移動命令では、進行方向と速度を示すベクトルと回転の情報が与えられ、アームへの命令には、モジュールの取り付け位置からの先端座標の相対座標が送信される。

3-1. コアモジュールの動作

コアモジュールは接続されたエンドモジュールの種類、機能の把握を行う。エンドモジュールの把握は、接続時に行われる通信によってコアモジュール内にエンドモジュールの情報が記録されることにより行われる。その通信内容はモジュールの提供可能な操作を指示するものであり、今回の例では「全体移動」「ロボットアームによるモジュールの移動」「特殊な命令による動作」の 3 つに分けられる。

Figure 7 のように、ホストとなる PC からロボットに命令を送信すると、通信ユニットを搭載しているコアモジュールがロボット全体への命令か、特定のモジュールへの命令か判断する。全体への命令の場合、命令内容が実行可能であるモジュールに対して命令を送信する。特定のモジュールを対象とした命令は、指定されたモジュールへ命令を送信する。エンドモジュールは受信した命令が動作可能か判断し、もし実行不可能な場合はエラーとしてコアモジュールへ送信する。エラーを受信したコアモジュールは代用可能なモジュールを判断し、存在しない場合はその旨をホストへ送信する。

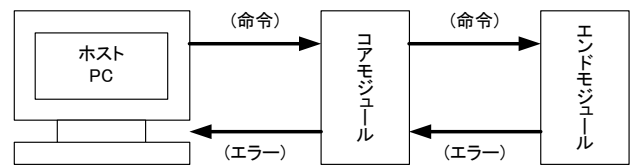


Figure 7: Communication process

3-2. 移動モジュールの動作

移動モジュールは各モジュールに進行可能な角度が記録されている。受信したベクトルの角度が動作可能か判断し、不可能な場合は微少旋回動作を行う。Figure 8 に歩行型、4 輪型、2 輪型の進行可能角度を表した単位円を示す。

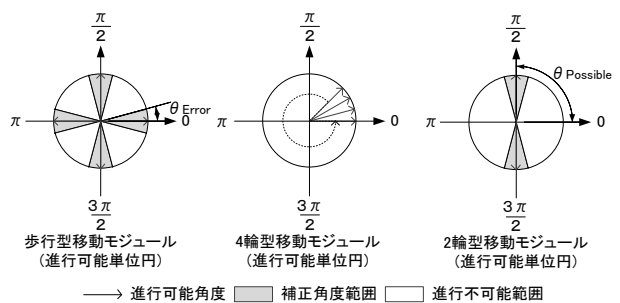


Figure 8: Unit circles indicate the angles which each locomotion module can move

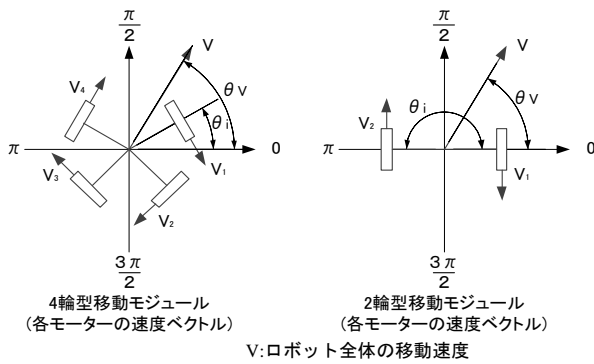
### ・歩行型移動モジュール

歩行型移動モジュールの動作は、前後左右の移動に対応するモーションがモジュールに登録されており、対応する移動可能範囲内に命令されたベクトルが入ることにより、該当するモーションが再生される。移動速度はモーションに依存するため、歩行型移動モジュールでの移動は一定の速さとなる。

### ・車輪型移動モジュールの計算モデル

車輪型移動モジュールの動作は、モジュールにより進行可能角度が限定されており、その角度を中心に補正角度範囲が設定されている。この補正角度範囲内にコアモジュールから送信されてきた進行方向が入っている場合、進行方向とロボットの角度がそろったと判断し、移動を実行する。また、進行方向が補正角度範囲内に入っていない場合は、補正角度範囲内に入るまで旋回動作を行う。4輪型移動モジュールは全方向移動が可能のため、全ての角度が進行可能となる。

移動に対する各モータの制御パラメータへの変換は、それぞれを Figure 9 のようなモデルとし、以下の計算モデルにより変換される [5]。



V: ロボット全体の移動速度  
 $\theta_v$ : ロボット全体の回転速度  
 $V_i$ : 各モータの回転速度 ( $i=1,2,3,4$ )  
 $\theta_i$ :  $V_x$  軸と回転軸のなす角 ( $i=1,2,3,4$ )  
 $L$ : 中心からオムニホイールまでの長さ

Figure 9: Motor rotation speed and moving direction of locomotion module

4 輪型移動モジュールの各モータの回転速度は次の式で求められる。

$$V_i = V_x \cos\left(\theta_i - \frac{\pi}{2}\right) + V_y \sin\left(\theta_i - \frac{\pi}{2}\right) + \theta_v L$$

$$= V_x \sin\theta_i + V_y \cos\theta_i + \theta_v L \quad (1)$$

(1) 式の  $i$  に ( $i=1, 2, 3, 4$ ) を代入する。

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} \sin\theta_1 & -\cos\theta_1 & L \\ \sin\theta_2 & -\cos\theta_2 & L \\ \sin\theta_3 & -\cos\theta_3 & L \\ \sin\theta_4 & -\cos\theta_4 & L \end{bmatrix} \begin{bmatrix} V_x \\ V_y \\ \theta_v \end{bmatrix} \quad (2)$$

同様に、2 輪型移動モジュールの 2 個のモータの回転を求める。

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} \cos\theta_1 & L \\ \cos\theta_2 & L \end{bmatrix} \begin{bmatrix} V \\ \theta_v \end{bmatrix} \quad (3)$$

### 3-3. アームモジュールの動作

アームモジュールの動作は逆運動学を用いて数式による解析的方法で各関節の角度を決定する。しかし、関節の構造によっては、各関節の角度の解が複数個ある場合や存在しない場合があり、手先の姿勢パラメータ (ロール、ピッチ、ヨー) を自由に決めるには 6 自由度以上なければならない [6]。よって、これらの問題を解決するため、以下の条件を使用し各関節の角度を決定する。

- それぞれの関節の回転角度が相対的に最も少ないパターンを選択する
- 手先の位置を手先の姿勢より優先とし、それぞれ位置、姿勢において X、Y、Z の順に優先度とする。

アームモジュールの共通命令は、付け根部分を原点として先端の相対座標と相対回転角とする。この共通命令を受信したモジュールは、実際にその座標・回転角が実現可能か判断し、実現が不可能な場合はエラーとして本体へその旨を伝え、実現が可能な場合は、それぞれのアクチュエータの回転可能範囲内かを確認する。回転範囲外の場合はエラーとしてコアモジュールへ送信され、回転可能な場合命令が実行される。

### ・簡易マニピュレータ計算モデル

今回使用したアームのモデルを Figure 10 に示す。

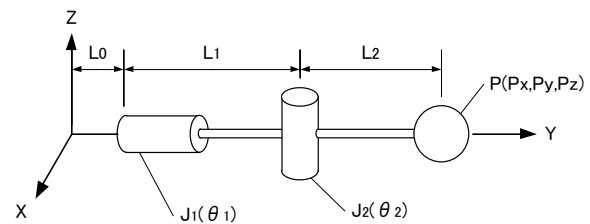


Figure 10: Joint model of arm module

このアームモジュールは、目標座標 P の X 軸座標、Y 軸座標、Z 軸座標の中から 2 つの座標が決まると残り座標の 3 つ目が決まる。このことから、各入力要素の優先順位より、X 軸座標、Y 軸座標を優先し、式(4)により Z 軸座標を演算により求める [6]。

$$P_z = \sqrt{L_2^2 + P_x^2 + (P_y - (L_0 + L_1))^2} \quad (4)$$

このロボットアームはピッチに関係する関節は  $\theta_1$  だけなので、 $\theta_1$  はアームの先端位置  $P_x, P_z$  を使用し式(5)で求められる。

$$\theta_1 = \tan^{-1}\left(\frac{P_z}{P_x}\right) \quad (5)$$

$\theta_2$  は次式により求める。

$$\theta_2 = \tan^{-1}\left(\frac{P_y - (L_0 + L_1)}{\sqrt{P_x^2 + P_z^2}}\right) \quad (6)$$



### 3-4. その他モジュールの動作

その他モジュールは、共通命令の定義はなく、独自の命令プロトコルで動作する。命令を受信したその他モジュールは、実行可能か判断し、実行不可能な場合はエラーを返信する。本実験では RoboCup 小型サッカーリーグで用いられている、キックデバイスモジュールを使用した。キックデバイスモジュールは、モータ、赤外線センサ、ソレノイドの複合モジュールで構成される。命令はモータの ON/OFF 命令によるホールドディングデバイスの制御とソレノイドを用いたキックデバイスの制御の 2 種類あり、キックデバイスについてはセンサの情報を元に実行の判断をする。

## 4. 動作実験

実験では、1. 構造を変更した際に同じ命令系で動作させた場合の各モジュールの動作を確認、2. 各モジュールのエラーに対応する動作の確認を行う。

### 4-1. 実験 1 (構造を変更した際の同じ命令での動作検証)

コアモジュール、アームモジュール、移動モジュールを組み合わせ、3 種類のサッカーロボットを構成し、同じ命令を送信することにより、異なるモジュール構成であっても動作をするのかを検証する。操作は、パソコンから全体移動やキックなどの命令を無線で送信するものとする。

#### • 歩行型タイプの構成

Figure 11 に歩行タイプロボットの外観を示す。

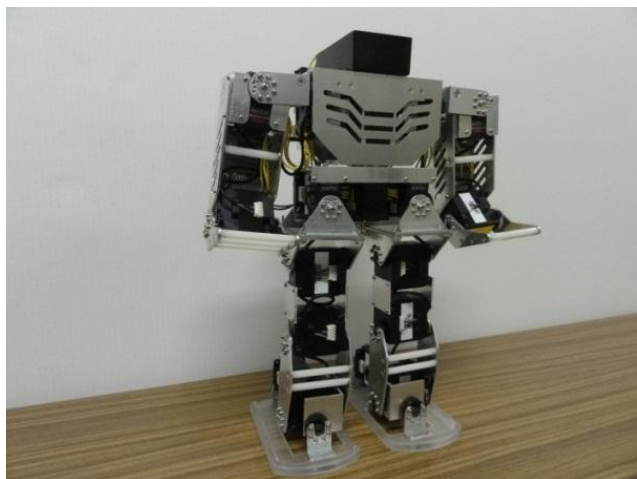


Figure 11: Structure of biped walk type locomotion module

上半身はコアモジュールとアームモジュールで構成されており、下半身は歩行型移動モジュールで構成されている。ボールをキックする動作については車輪型の構成とは異なりキックデバイスモジュールを搭載していないが、脚部(アームモジュール)の移

動によりキックを行う。

#### • 4 輪型タイプの構成

Figure 12 に 4 輪型タイプロボットの外観を示す。

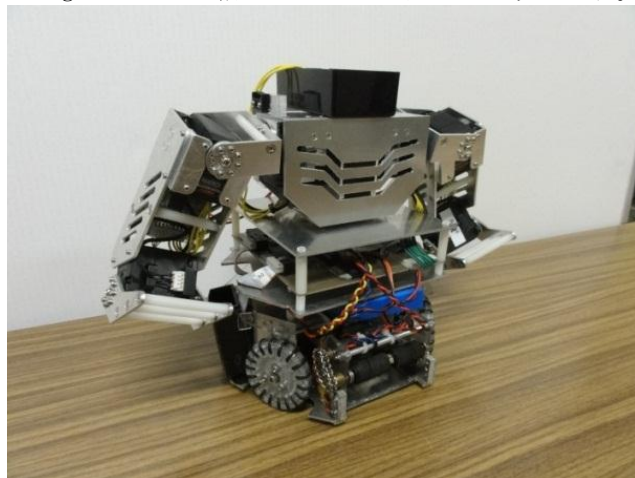


Figure 12: Structure of 4-wheel type locomotion module

上半身はコアモジュールとアームモジュールで構成されており、下半身は全方向移動型の 4 輪型移動モジュールとキックデバイスモジュールで構成されている。ボールのキック機構はキックデバイスモジュールに搭載されているソレノイドによってキックする。

#### • 2 輪型タイプの構成

Figure 13 に 2 輪型タイプロボットの外観を示す。上半身は 4 輪型ロボットと同じ構成となっており、下半身は 2 輪型移動モジュールとキックデバイスモジュールとなっている。

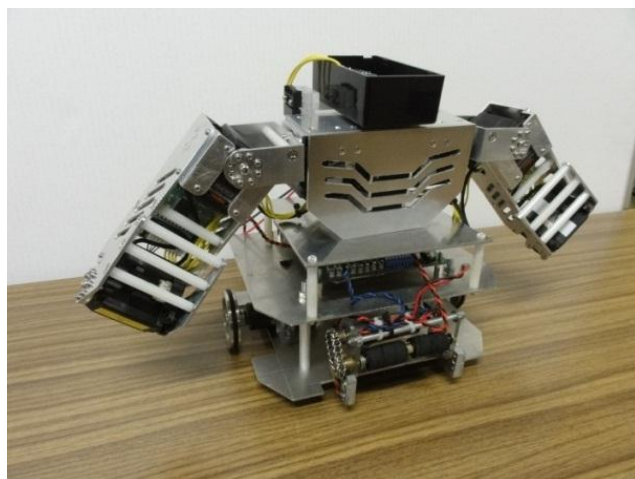


Figure 13: Structure of 2-wheel type locomotion module

#### • 結果

はじめに、それぞれの構成において進行可能な 90° 方向 (前方) への直進命令を送信した。この命令に対し歩行は前進モーションを再生し、4 輪型の構成と



2 輪型の構成は各モータを制御し前進した。よって、命令したベクトルを進行可能範囲内であると判断し、指定された方向への移動動作を実行することを確認した。

次に、ロボットへ送信する命令のベクトルを  $90^\circ$  から  $0^\circ$  へ微小角度ずつ変化させた。歩行型の構成と 2 輪型の構成は、命令のベクトルを  $90^\circ$  から数度傾けた場合でも前進動作を実行したが、一定角度を境に旋回動作を実行した。よって、補正角度範囲内は指定された方向への移動動作を実行し、進行不可能範囲内では旋回動作を実行することを確認した。

一方、4 輪型の構成の場合は、 $90^\circ$  から  $0^\circ$  への変化に対し、指定されたベクトル方向へ進行した。よって、進行不可能範囲を持たない 4 輪型の構成は旋回動作を実行せず、命令のベクトルを進行可能範囲だと判断し、指定された方向への進行動作を実行した。

以上より、異なる構成であっても同じ命令で操作することが可能であることを確認した。

#### 4-2. 実験 2(各モジュールエラーに対する動作検証)

本来は接続されているモジュールに対してエラーの状態を発生させ、どのような動作となるかを検証した。その方法は、ロボットの構成を上半身はコアモジュールとアームモジュール、下半身を歩行型移動モジュールにし、仰向けもしくはうつ伏せにして配置する。その状態で、直立をするように起き上がりモーションを命令する。その時、コアモジュールの左右に接続されているどちらかのアームの配線を切り離した際、コアモジュールが、その内容を判断し、適切に動作するか検証をする。

##### ・結果

左右ともにアームの配線が接続されている場合は、左右の腕を前後に伸ばし、立ち上がることができた。左右のアームのいずれか一方の配線を切り離した場合、断線しているアームからエラーが検出され、接続されている方のアームで代行し立ち上がった。左右ともにアームの配線を切り離したとき、左右のアームからエラーが検出され、コアモジュールから動作不可能というエラーがホスト PC に送信された。

以上より、モジュールでトラブルが発生した場合でも、その内容を判断し対応することが可能であることを確認した。

#### 5. まとめ

本報告では、分散型制御を用いたロボットの一種である、モジュール型ロボットを基礎とし、モジュールの不均一化によるロボットの拡張性の向上を提案した。コアモジュールとエンドモジュールを組み合わせることでロボットを構成し、異なる構成でも共通の命令で動作することを確認した。また、意図的にモジュールにエラーを発生させ、モジュールが

使用不可能な状況に陥った場合でも、動作可能なモジュールを判断しエラーに対応することを確認した。

以上より、集中型制御の複雑な制御の可能性と分散型制御のシステムの安全性・保守性を兼ね合わせることが可能なことを確認することができた。

このシステムは、広範な環境や課題が存在する災害現場など緊急時に即座に対応できることも期待できる。

今後の課題は、アームの状態とロボット全体の姿勢の演算、モジュール同士の衝突回避が必要である。また、外界センサを取り付け、環境から受ける外乱に対しコアモジュールに判断をゆだねるのではなく各モジュールで判断、動作をする人間の“反射”に類似した研究をテーマにしていきたい。

#### 謝辞

本研究は、中部大学情報科学研究所および中部大学特別研究費(CP)の支援を得て行われた。ここに記して謝意を表す。

#### 参考文献

- 1) 神村, 村田, 吉田, 黒河, 富田, 小鍛: 自己組立可能なモジュール型ロボットシステムに関する研究, 日本機械学会論文集(C編) Vol. 68 No. 667, pp. 886~892, 2002.
- 2) 黒河, 吉田, 神村, 富田, 村田, 小鍛: 変形し移動する自立モジュール型ロボット, 日本ロボット学会誌 Vol. 21 No. 8, pp. 855~859, 2003.
- 3) M. W. Jorgensen, E. H. Ostergaard and H. H. Lund: "Modular ATRON: Modules for a self-reconfigurable robot," Proc. of the 2004 IEEE/RSJ International Conference on Intelligent
- 4) 川内, 稲葉, 福田: セル構造化ロボットシステムに関する研究(適応性、自己組織性、自己進化性のあるロボットを目指して)、日本ロボット学会誌 Vol. 12 No. 1, pp. 1116~132, 1994
- 5) 渡部, 王碩, 石田, 井上, 河田, 水野, 木村: 全方向移動型歩行訓練機の経路追従の制御、日本機械学会 Vol. 2008 No. 46, pp. 423~424, 2008
- 6) 出村 公成, 簡単! 実践! ロボットシミュレーション、森北出版株式会社

## 距離センサを用いたロボット間通信に関する一考察 A study on communication between robots using distance sensors

小田翔平 植村渉

Shohei Oda and Wataru Uemura

龍谷大学

Ryukoku University

oda@vega.elec.ryukoku.ac.jp wataru@rins.ryukoku.ac.jp

Abstract — Robot technologies have been developed so much. Due to the developed technologies, the problems, more complex and various request for robot's work, are caused. To solve this problem, multi-robots should cooperate with each other. In order to cooperate with each other, robots need to share information by communication. Usually the communication method between robots is wireless communication. Because it is possible to twine, if several robots work with cable. However, wireless communication may fall by interference, access point trouble, and so on. Then, robots cannot share information, and cooperation. Therefore, we noticed mounted sensors. Mounted sensors are less affected by environment than wireless communication. Generally, robots often have distance sensors. To solve the problem that wireless communication falls, we propose the robot communication method using distance sensors.

### 1. はじめに

RoboCup 世界大会に 2012 年度から新たに加わったリーグとして Logistics League sponsored by Festo がある。このリーグは 2010 年、2011 年に Festo Logistics Competition という形でデモンストラーションを行い、2012 年から、正式種目となった。

この競技は Festo 社製の Robotino というロボットを使用する。このロボットは、OS として Ubuntu 9.04 を搭載し、全方向に移動が可能であり、多数のセンサを多数搭載している。また、RoboView という GUI のプログラミング環境が用意されており、直観的なプログラムの作成ができる。競技エリアを生産工場と見立て、材料（書き換え可能な無線タグである RFID が乗ったパック）の加工をいくつかの作業にて繰り返し、指定された配送エリアに運ぶことが最終的な目的である（図 1 参照）。作業を終えると作

業に応じた点数を得る。作業によって、配点が大きく異なるので、効率よく点を得るには、お互いのロボットの行動が干渉するときには配点の高い作業をしているロボットを優先させる方が良い。また試合の途中でボーナスゲームとして特急便課題が用意されており、専用のパックを 1 回加工して配送エリアに運ぶと、高い点が得られる。これを上手く利用すると、劣勢を逆転することも可能である。



図 1. 競技エリア

材料の加工は、図 1 の正方形の黒い枠に囲まれた 10 カ所の作業場で行う。作業場に置かれた生産装置は、シグナルにより材料の受け入れ状態を表す。どの生産装置がどのような加工をするかは試合ごとにランダムで決められているので、ロボットは試合中に生産装置のシグナルから、どの種類が割り当てられているかを推測しなければならない。この生産装置の情報は無線 LAN を介してロボット間通信を行うことで、ロボット同士が共有し、加工手順を決めることや、作業を優先するロボットを決めることができる。ルール上、競技の途中で無線 LAN が 100 秒間途切れる。ロボット間通信ができないため協調動作ができなくなる。このとき、ロボットは次にどの作業場へ材料を運ぶべきか、どのロボットを優先して作業させるべきかわからなくなるため、競技の進行に支障が生じる。ただし 2011 年の世界大会では、

無線 LAN の切断は適用されなかった

本研究では、ロボットが標準で搭載している距離センサを用いることにより、無線 LAN が使えない環境のためのロボット間通信を提案する。

## 2. 提案手法

この競技で高得点を狙うには、無線 LAN が切断されても、ロボット間で情報共有することが要求される。本研究では、無線 LAN が使えないときの通信手段として、距離センサを用いてロボット間通信を行う方法を提案する。送信側のロボットは、受信側に対する距離を基準とした3箇所を近い順に position\_short, position\_middle, position\_long として選び、position\_middle を中心にどちらに動くかで0と1の1ビットの情報を送信する。1ビット送るたびに position\_middle に戻り、ビットの区切りを伝える。また、通信を始める前に、position\_short と position\_long で一旦静止し、受信側にその距離を知らせる。図2に1ビットの送信方法を示す。

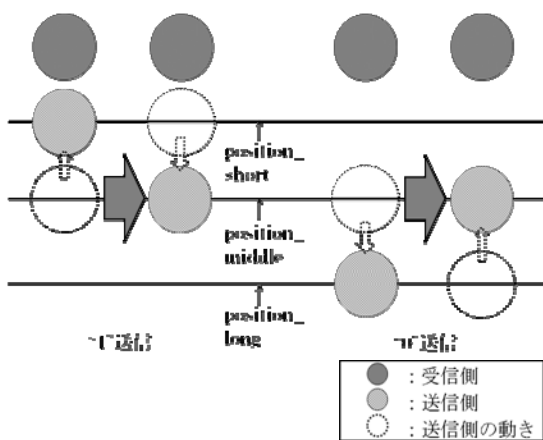


図2. 1ビットの送信方法

受信側は、距離センサが出力する電圧値から、送信側との距離が把握できる。ただし、電圧値と距離の関係は図3に示すように線形ではない。そのため、position\_short と position\_long 間の距離を3等分すると、D1 から D2 を position\_short, D2 から D3 を position\_middle, D3 から D4 を position\_long とおくと、D3 から D4 間での電圧変化は非常に小さいものになる。D1 は最近接から少し離れた位置でとり、予備実験では6cm で受信率が良かったので、

6cm で固定する。このときロボット内の回路や外のノイズにより、出力電圧が変化する場合があることを考えると、電圧はある程度幅があるべきであると考えられる。

本研究では、position\_short と position\_long で取る電圧を3等分して距離を決めた。

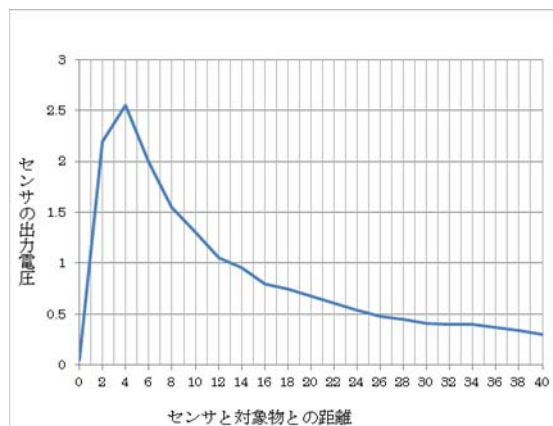


図3. 距離センサの特性

## 3. 実験

競技で使用するロボットには、全9方位に距離センサが設置されており、ロボットの周りの障害物を検知できる。作業場の座標はあらかじめわかっているため、作業場に近づいたときに距離センサが反応した場合、その作業場には既に作業しているロボットがいると考えられる。

このとき、相手のロボットに対して試合環境の情報を伝えることを考える。試合の情報を伝えるには最低でも6ビットの情報が必要である。生産装置の受け入れ状態は機械的な故障を除き3パターンある。そして場所は競技エリア内10箇所である。生産装置の状態を表すために2ビット、場所を表す4ビット、合計6ビットの通信ができれば、無線が切れる前後の差分を埋めることができると考える。

そこで、実験では6ビットの情報を送り、そのときのビットの損失、および、ビット誤りを測定する。ロボットの移動距離と誤りの関係を得た。情報を正しく受信できた割合を図4に示す。

また、結果はシミュレータ上で動作したものである。

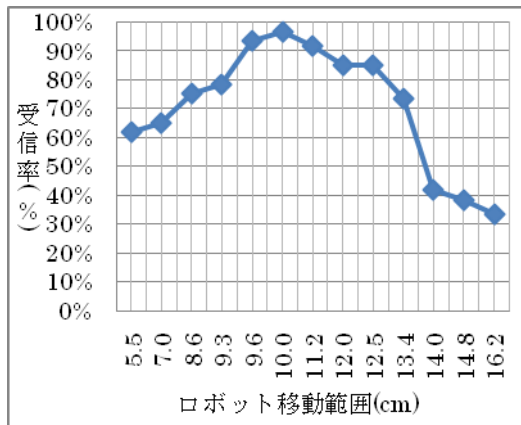


図4. 受信できた割合

グラフから、ロボットの移動範囲が9.6cm から11.2cm 程度の時が、良く受信できることがわかった。

送信側が position\_short を伝えるとき、送信側自身の距離センサ値を用いる。 position\_long を伝えるときは position\_short からの距離で判断する。

position\_short を伝えるとき、送信側と受信側はお互いの距離センサの向きが分からないため、それぞれの距離センサは異なる値を示す。例えば、送信側が電圧値 V1 から V2 の間を用いて送信したとき、受信側では V1-a1 から V2-a2 の間で距離センサが変化する。

これにより、送信側が2節の D2 付近を移動しているにも拘わらず、受信側では D2 から D3 を移動しているように受け取るといった誤認識が生じ、受信率が下がった原因と考えられる。

#### 4. おわりに

本研究では、ロボットに搭載された距離センサの変位を用いて、ロボット間通信を行った。送信ロボットが9.6cm から11.2cm 程度の幅で移動したときが、効率がよいことがわかった。競技で用いるならこの範囲がよいことがわかる。ただし、受信率が100%ではないので、今後は、プログラムを見直し、受信率100%を目指すことが課題である。

#### 参考文献

[1] 自立分散型ロボット間通信のための適応型メデ

ィアアクセス制御プロトコル。

荒井 順平, 小山 昭夫, バロリ・レオナルド.  
 情報処理学会研究報告. マルチメディア通信と分散  
 処理研究会報告, 2003(87), pp. 117-122, 2003

[2] 研究活動とロボットコンテスト(RoboCup).

浅田稔, 國吉康夫, 野田五十樹, 北野宏明  
 日本ロボット学会誌, Vol. 15 No. 1, pp. 13-16,  
 1997

# 自律移動ロボットの環境地図とセンサ情報に基づく 隠れ領域を考慮したリアルタイム速度制御 Real Time Speed Control for Autonomous Mobile Robots Considering Occluded Regions Based on Map and Sensor Data

○萬 礼応 (慶應義塾大学大学院)  
森口 智規 (村田機械株式会社)  
高橋 正樹 (慶應義塾大学)

\* Ayanori YOROZU(Keio Univ.), Toshiki MORIGUCHI(Murata Machinery Co., Ltd.),  
Masaki TAKAHASHI(Keio Univ.)

[ayanoriyorozu@a5.keio.jp](mailto:ayanoriyorozu@a5.keio.jp), [toshiki.moriguchi@drw.muratec.co.jp](mailto:toshiki.moriguchi@drw.muratec.co.jp), [takahashi@sd.keio.ac.jp](mailto:takahashi@sd.keio.ac.jp)

**Abstract**—This paper describes a real time speed control method for autonomous mobile robots considering safety in the coexistence space of human and the robots. Recently various autonomous mobile robots are developed for practical use. In a real environment, there are occluded regions of sensor, such as corners and crossroads. In order to ensure the safety for humans and moving objects appear from these occluded regions, it is necessary to perform the real time speed control method. In addition, in the narrow environment, it is necessary to limit the speed for safety. In this paper, the novel real time speed control method considering occluded regions and movable area based on map and sensor data is presented. To verify the effectiveness of the proposed method, several numerical simulations are carried out.

## 1. 緒言

近年、案内や荷物の搬送など、人間の生活空間での稼働を目的とした自律移動ロボットの研究・開発が行われ、公共施設などへの導入例も報告されている[1]-[3]。著者らも、Fig. 1に示すような病院内搬送ロボット MKR-003(Muratec Keio Robot)を開発し、実環境での実験・検証を行っている。これらの自律移動ロボットが実環境で稼働する際に必要な要素技術としては、環境地図の作成、経路計画、自己位置推定、障害物回避などが挙げられる。特に、人間の生活空間で稼働する場合には、レーザレンジファインダ(LRF: Laser Range Finder)などの測域センサの情報に基づいてリアルタイムに安全な行動生成を行う必要がある。しかし、実環境には十字路や曲がり角などセンサの隠れ領域が生じる状況がある。そこで、Fig. 1(b)に示すように、隠れ領域から現れる人間や移動物体などに対して安全性を確保するために、ロボットの速度制御を行う必要がある。また、病院内での実機検証の際に、狭い通路など移動可能な領域が小さい環境で、広い環境と同様の速度で移動を行った場合に、ロボットに対して恐怖心を感じるというアンケート結果を得た。そこで、移動領域の大きさに応じて、移動速度を調整する必要がある。

このような速度制御に関しては、未知環境においてセンサで検出した障害物や未観測領域への移動時

にその距離に応じて移動速度の制御を行う手法が提案されている[4]。しかし、ロボットの隠れ領域に対する危険度は、距離だけではなく隠れ領域の大きさに依存する。そのため、距離に加えて隠れ領域の大きさ、つまり、ロボットと隠れ領域の位置関係を考慮した速度制御が必要である。また、事前に取得した環境地図を用いて、障害物の近傍に仮想的な領域を付加することで安全性の向上を図るオフラインでの経路計画が提案されている[5][6]。しかし、事前情報に基づく速度制御であるため、事前情報にない障害物など環境が変化した場合に対応できないといった問題がある。

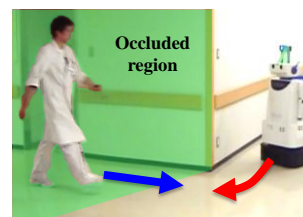
そこで本研究では、事前に取得した環境地図（占有格子地図）とセンサ情報に基づいて、以下の二点を考慮したリアルタイムな速度制御手法を提案する。

- ① 隠れ領域の大きさ
- ② 移動領域の幅

隠れ領域の大きさに応じて速度制御を行うことで、曲がり角などでは隠れ領域が大きくなる手前で減速を行い、曲がり角に差し掛かりセンサ視界が開けて隠れ領域が小さくなると加速を行うといった人間の動きに近い速度制御を行うことが可能になる。また、事前に取得した環境地図とセンサ情報を逐次比較することで、曲がり角などにおける隠れ領域に加え、事前情報にない障害物による隠れ領域にも対応することが可能になる。さらに、センサによって移動領域の幅を測定し、移動領域幅に応じて速度を制限することで、狭い通路などで安全かつ人間に恐怖心を与えない移動を実現する。



(a) MKR-003



(b) An example of situation that require speed control.

Fig. 1 MKR-003: a mobile robot for transport applications in hospital domain.



## 2. 隠れ領域の大きさに基づく速度制御

本章では、ロボットが十字路や曲がり角などセンサの隠れ領域が生じる環境へ進行する場合に、隠れ領域から現れる人間や移動物体に対して安全性を確保するためのリアルタイムな速度制御の設計法①について述べる。

ロボットは事前に作成した環境地図（占有格子地図）を所有し、LRFなどの測域センサを搭載していると仮定する。ここで占有格子地図とは、環境を小さな格子（グリッド）毎に分割し、グリッド毎に障害物の存在確率を保有するものである。この事前取得した環境地図と自己位置、現在のセンサに基づく周囲環境までの距離情報を比較する。そして、Fig. 2に示すようにリアルタイムな行動制御手法で決定した移動方向から $\pm\theta$ の範囲に存在する隠れ領域の大きさに応じて、ロボットの最大速度指令値 $v_{max}^{ref}$ を以下のように決定する。

$$v_{max}^{ref} = v_{max} \times \frac{R_{recognize}(\theta)}{R_{map\_free} \cap R_{sensor}(\theta)} \quad (1)$$

ここで、 $R_{map\_free}$ は占有格子地図上で障害物が存在しない、すなわちロボットが移動可能な領域、 $R_{sensor}(\theta)$ 、 $R_{recognize}(\theta)$ はそれぞれ進行方向から $\pm\theta$ の範囲のセンサ領域と実際にセンサで取得している領域である。また、 $v_{max}$ はロボットの最大移動速度である。

このように、事前取得した環境地図とセンサ情報を比較することで、曲がり角などにおける隠れ領域に加え、事前情報にない障害物による隠れ領域にも対応することが可能になる。また、距離に基づいた速度制御ではなく、隠れ領域の大きさ、つまり、ロボットと隠れ領域の位置関係に応じて速度制御を行うことで、曲がり角などの手前で減速を行い、曲がり角に差し掛かり隠れ領域が小さくなると加速を行うといった人間の動きに近い速度制御を実現可能となる。

## 3. 移動領域幅に基づく速度制御

狭い通路など移動領域が狭い場合に、接近した物体に対して過敏な回避行動を行うと人間に対して恐怖感を与える場合がある。そこで、本章では通路幅や移動領域の大きさに応じた速度制御を行うための設計法②について述べる。

まず、センサ情報から移動領域の幅 $w$ をFig. 3(a)に示すように、ロボットの進行方向に対して検出した障害物間の垂直方向の距離として求める。そして、 $w$ に基づく関数 $f(w)$ からロボットの最大速度指令値を以下のように決定する。

$$v_{max}^{ref} = v_{max} \times f(w) \quad (2)$$

ここで、 $f(w)$ はFig. 3(b)に示すような以下のシグモイド関数として定義する。

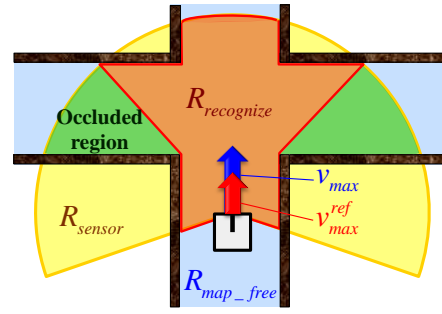


Fig. 2 An example situation of speed control considering occluded regions based on map and sensor.

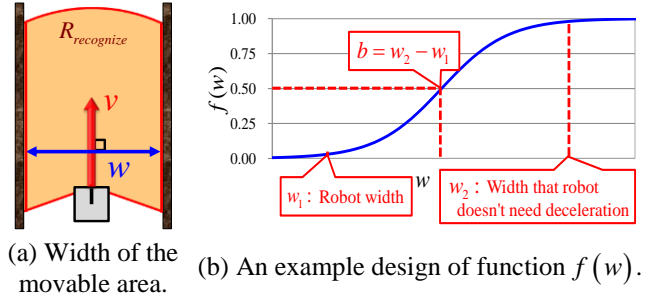


Fig. 3 An example situation of speed control considering movable area.

$$f(w) = \frac{1}{1 - \exp(-a(w-b))}, b = w_2 - w_1 \quad (3)$$

このとき、 $w_1$ をロボットの幅、 $w_2$ をロボットが減速不要な移動領域の幅として設定する。

また、本研究では最終的に前章で述べた隠れ領域も考慮し、以下の式によって環境地図とセンサ情報に基づくリアルタイムな速度制御①+②を実現する。

$$v_{max}^{ref} = v_{max} \times \frac{R_{recognize}(\theta)}{R_{map\_free} \cap R_{sensor}(\theta)} \times f(w) \quad (4)$$

## 4. シミュレーション

本章では数値シミュレーションにより、提案手法の有効性の検証を行う。本研究では病院内搬送ロボットMKR-003を想定し、ロボットは全方位移動機構を搭載し、最大移動速度 $v_{max} = 0.5\text{m/s}$ 、最大加速度 $0.5\text{m/s}^2$ で移動可能であるとする。また、機体前方にLRFを搭載し、前方を $0\text{deg}$ として、 $100\text{ms}$ 毎に $\pm 120\text{deg}$ 、 $4.0\text{m}$ までの距離情報を取得可能であるとする。ロボットは事前に環境の占有格子地図（今回はグリッドサイズを $5\text{cm}$ とした）を取得しているものとする。手法①の進行方向に対して隠れ領域を考慮する角度幅 $\theta$ を $90\text{deg}$ とし、進行方向の決定等はファジィポテンシャル法(FPM: Fuzzy Potential Method)[7]を用いて行い、その最大速度指令値 $v_{max}^{ref}$ を

本提案手法により決定を行うものとする。また、手法②の移動領域幅に対して速度制御を行うためのパラメータはロボットサイズ等を考慮してそれぞれ、 $w_1 = 0.70\text{ m}$ 、 $w_2 = 2.0\text{ m}$ 、 $a = 4.9$ とする。

#### 4.1 ロボットと隠れ領域の位置関係に基づく最大速度指令値の検証

本節では、Fig. 4(a)に示すように、ロボットと隠れ領域までの距離をすべて1.5 mとし、その位置関係が異なる状況における隠れ領域の見え方の違いや、提案手法①による最大速度指令値  $v_{max}^{ref}$  を検証する。そこで、ロボットをFig. 4(a)における  $\varphi$  を0から60 degまで5 deg毎変化させた際の各位置で、提案手法による最大速度指令値を算出する。ただし、各位置でのロボットの進行方向はすべてロボット正面方向(y軸正方向)であるとする。

Fig. 4(b)に各位置での提案手法①による最大速度指令値  $v_{max}^{ref}$  を示す。また、 $\varphi = 20\text{ deg}$ 、 $60\text{ deg}$ におけるロボットから見た隠れ領域の範囲をそれぞれFig. 4(c)、Fig. 4(d)に示す。Fig. 4(c)とFig. 4(d)を比較して分かるように、ロボットと隠れ領域までの距離は同じであっても、位置関係が異なるとロボットから見る隠れ領域の範囲も異なるため、それに応じた速度制御が必要である。これに対して、提案手法①では、隠れ領域の大きさを考慮することで、Fig. 4(b)に示すように、状況に応じて最大速度を決定することが可能であることが確認できる。

#### 4.2 十字路における速度指令値の検証

まず、Situation 1としてFig. 5に示すような通路幅2.3 mの十字路をロボットが直進する際の速度指令値に関する検証を行う。次に、Situation 2として、事前情報にない幅0.60 m、奥行0.80 mの障害物が(-1.45 m, 4.5 m)の位置に存在する場合、最後にSituation 3として、通路幅が1.5 mであった場合の提案手法②による速度指令値の検証を行う。

Situation 1の結果をFig. 5に示す。Fig. 5(a)、5(c)より、隠れ領域を考慮して速度制御を行わない場合には、十字路を最大速度のまま通過している。これに対して、Fig. 5(b)、5(c)より、提案手法①を適用することで、十字路に侵入する前で減速を行い、十字路侵入後は隠れ領域や障害物がないため、加速するという移動が実現できることが確認できる。

Situation 2の結果をFig. 6に示す。事前に取得した環境地図を用いて、オフラインで速度制限領域を設定する手法[5][6]では、事前情報のない障害物によって生じる隠れ領域を考慮した速度制御を行うことはできない。これに対して、本提案手法①では事前に取得した環境地図とセンサ情報を比較することで、リアルタイムに隠れ領域を検出することが可能である。このため、Fig. 6に示すように、事前情報にない障害物による隠れ領域に対しても減速を行った移動を実現できることが確認できる。

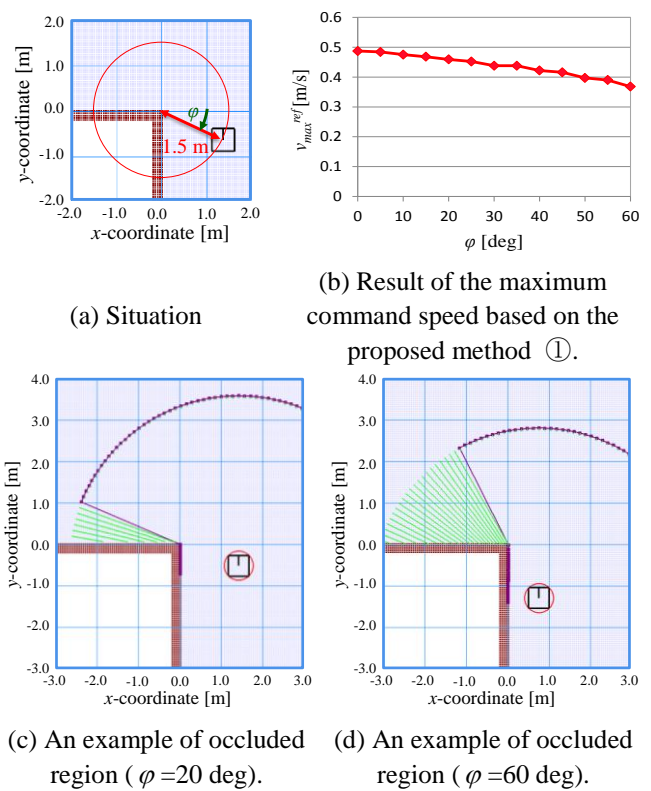


Fig. 4 Verification of the speed control considering occluded regions based on map and sensor in several situations.

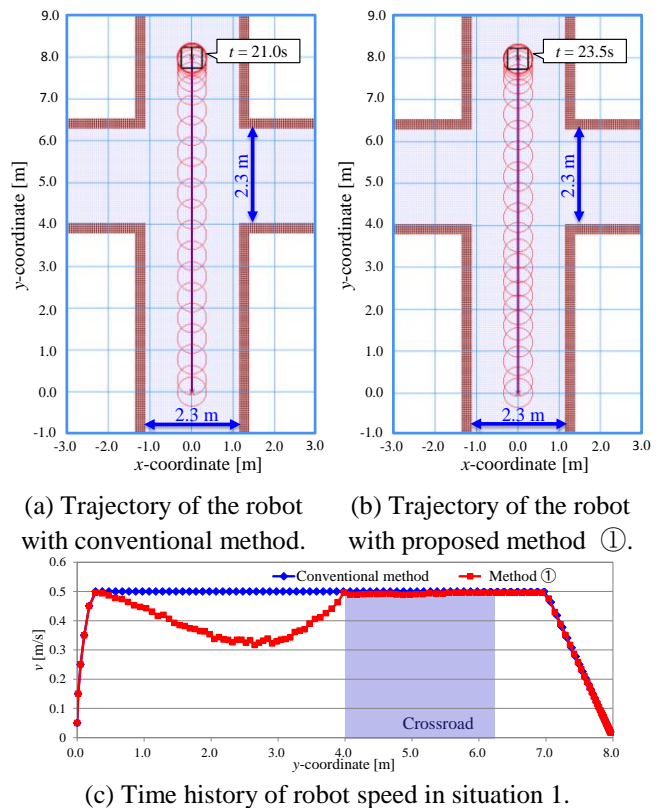
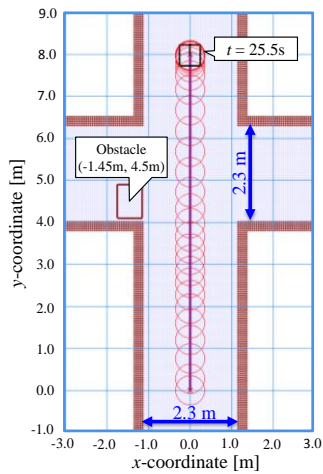
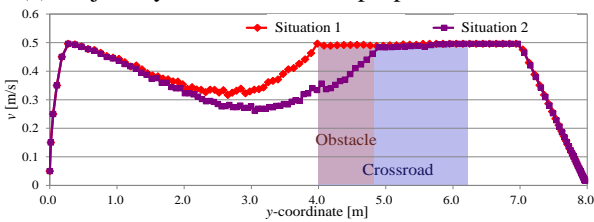


Fig. 5 Verification of the speed control considering occluded regions based on map and sensor at crossroad (Situation 1).





(a) Trajectory of the robot with proposed method ①.



(b) Time history of the robot speed in situation 2.

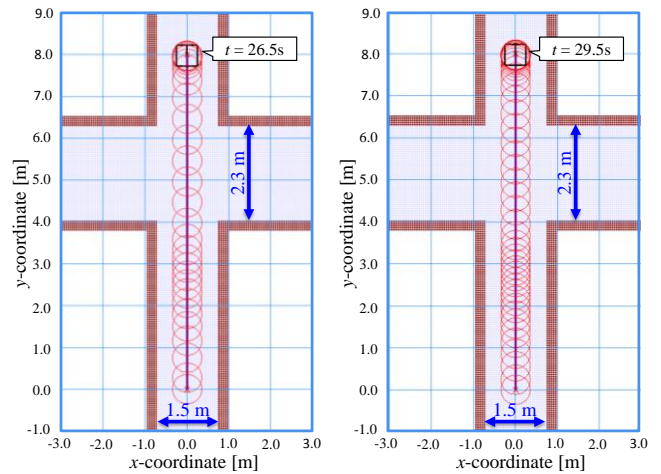
Fig. 6 Verification of the proposed method ① for the obstacle that the robot has no priori information (Situation 2).

最後に、Situation 3の結果をFig. 7に示す。ここでは、移動領域の幅を用いて速度制御を行う手法②を統合することで、Fig. 7(b), (c)に示すように、ロボットが幅の狭い通路などを移動する際に、移動速度を抑えた移動が実現できることを確認した。

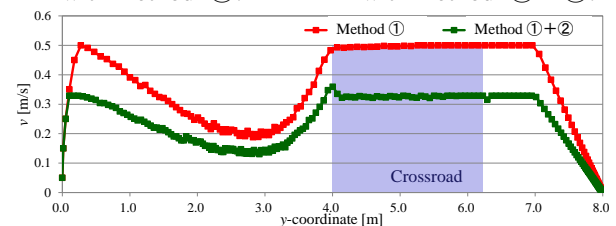
以上より、数値シミュレーションにおいて、隠れ領域の大きさや移動領域の幅を考慮したリアルタイムな速度制御手法の検証を行った。隠れ領域の大きさを考慮した速度制御を行うことで、ロボットから隠れ領域までの距離ではなく、それらの位置関係によって速度を決定することができることを確認した。また、事前に取得した環境地図とセンサ情報から隠れ領域を検出することで、事前情報にない障害物による隠れ領域なども考慮した速度制御が実現できることを確認した。さらに、移動領域幅に基づいた速度制御を加えることで、狭い通路など移動可能な領域が小さい場合に、最大移動速度を抑制することが可能であることを示した。

## 結言

本研究では、RoboCup サッカー中型リーグから派生し、現在病院内など実環境での稼働を目的とした自律移動ロボットの安全性向上のために、環境地図とセンサ情報に基づく隠れ領域を考慮したリアルタイムな速度制御手法を提案した。ロボットの隠れ領



(a) Trajectory of the robot with method ①. (b) Trajectory of the robot with method ①+②.



(c) Time history of robot speed in situation 3.

Fig. 7 Verification of the speed control considering the movable area at the narrow passage (Situation 3).

域に対する危険度は、距離だけではなく隠れ領域の大きさ、つまり、ロボットと隠れ領域の位置関係に依存する。そこで、隠れ領域の大きさを考慮した速度制御を行うことにより、曲がり角などでは隠れ領域が大きくなる手前で減速を行い、曲がり角へ進行してセンサ視界が開けて隠れ領域が小さくなると加速するといった、人間の動きに近い速度制御が実現できることを確認した。また、事前に取得した環境地図と現在のセンサ情報を用いることで、事前情報に無い障害物などによる隠れ領域も考慮した速度制御が可能であることを確認した。さらに、移動領域幅に基づく速度制御を統合することで、狭い通路など移動可能な領域が小さい場合に、移動速度を抑制した移動が実現できることを確認した。

本稿では静的環境において、隠れ領域と移動領域を考慮した速度制御を行った。今後は、本速度制御に基づく障害物回避手法の検討と移動物体の存在する動的環境への拡張を行い、実機実験によって検証・評価することを課題とする。

## 参考文献

- [1] Zhao Tiejun, Tan Dalong and Zhao Mingyang, "The Development of a Mobile Humanoid Robot with Varying Joint Stiffness Waist," Proceedings of the IEEE International Conference Mechatronics and Automation, Vol. 3, pp. 1402-1407, 2005.
- [2] B. Krishnamurthy and J. Evans, "HelpMate: A Robotic Courier for Hospital Use," IEEE International Conference

on Systems, Man and Cybernetics, Vol. 2, pp. 1630-1634, 1992.

- [3] J. M. Evans, "HelpMate: An Autonomous Mobile Robot Courier for Hospitals," Proceedings of the IEEE/RSJ/GI International Conference on Intelligent Robots and Systems, Vol. 3, pp. 1695-1700, 1994.
- [4] 三浦純, 根岸善朗, 白井良明, "地図と移動の不確かさを考慮した未知環境における移動ロボットの適応的速度制御," 日本ロボット学会誌, Vol. 24, No. 1, pp. 47-55, 2006.
- [5] Woojin Chung, Seokgyu Kim, Minki Choi, Jaesik Choi, Hoyeon Kim, Chang-bae Moon and Jae-Bok Song, "Safe Navigation of a Mobile Robot Considering Visibility of Environment," IEEE Transactions on Industrial Electronics, Vol. 56, No. 10, 2009.
- [6] 後藤清宏, 根和幸, 松野文俊, "速度制約領域を考慮した自律移動ロボットの行動計画," 日本ロボット学会誌, Vol. 28, No. 8, pp. 930-937, 2010.
- [7] 津崎良一, 吉田和夫, "ファジィポテンシャル法に基づく全方位視覚を用いた自律移動ロボットの行動制御," 日本ロボット学会, Vol. 21, No. 6, pp. 656-662, 2003.

# ロボカップジュニアサッカーチャレンジへ参加することを目標とした ロボットプログラミングテキストの試作

光永 法明, 山形 慎平

Noriaki Mitsunaga and Shinpei Yamagata

大阪教育大学

Osaka Kyoiku University

mitunaga@cc.osaka-kyoiku.ac.jp

## Abstract

本論文では、読者がロボカップジュニアサッカーチャレンジに参加することを目標とし、ロボットのプログラミングを通して外界への関心を高められるように配慮した、ロボットプログラミングのテキストを試作したので報告する。まずテキストの製作方針と構成を紹介する。そして試作したテキストについてロボットスクール受講生による評価を示す。その中で受講生から後輩のために「あるといい」という回答を得た。

## 1 はじめに

現在広く利用されているデジタル計算機にはプログラムが必要である。プログラミングとは、プログラミング言語の文法に従って、目的とする計算機の動作を実現する表現をプログラムとして生成することである。プログラミングには、プログラミング言語についての知識だけでなく、実世界の数値あるいは記号による表現についての理解も必要となる。そのため、子どもたちが計算機をプログラミングして何かを実現しようとするとき、外界への関心が高まり学びにつながるという報告がある [三宅, 1985]。そのとき重要なのはプログラミング言語を教えプログラムを作らせることではなく、子ども自身が興味を持って何かを達成しようとすること（子どもの主体性）ではないかと指摘されている。

ところで、ロボカップジュニア (RoboCupJunior) はロボットと人工知能に関するグランドチャレンジ RoboCup から派生した国際ロボット競技会である。現在では 30 カ国以上で活動があり、RoboCupJunior 2011 には 251 チーム、955 名が参加した。参加者は 19 歳以下であり、参加者が自ら作成した自律型ロボットにより競技をする。競技はチャレンジとよばれ、サッカー、レスキュー、ダンスの 3 種が行われている。ロボカップジュニアでは、ロボッ

トが自律して動作することが求められ、競技中にはロボットの操作が認められない。そのため、ロボットの動作を決めるプログラミングが不可欠となっている。また参加者自身がロボットのハードウェア、ソフトウェアを作成することを求めている。したがって、ロボットに興味を持ち競技に参加しようとする子どもたちにとって、プログラミングを通して主体的に外界への関心を高める場となると期待される。

一方、子どもたちがロボカップジュニアに関心を持ち参加したいと考えたとき、必ずしも身近にロボットのプログラミングに精通した指導者を見つけられるとは限らない。また子ども向けの書籍も限られている [子供の科学編集部, 2002, 黒木, 2006] のが現状である。

そこで、読者がロボカップジュニアサッカーチャレンジへ参加することを目標とし、ロボットのプログラミングを通して外界への関心を高められるように配慮したテキストを試作したので本論文で報告する。以下では、ロボカップジュニアサッカーチャレンジについてルールの概要を説明する。次にテキストの作成指針と内容を紹介する。そしてロボットスクールに参加する子どもたちにテキストについてアンケートを実施したので報告し、最後にまとめと今後の課題を述べる。

## 2 ロボカップジュニアサッカーチャレンジ

ロボカップジュニアサッカーチャレンジではサッカーを模した競技をする [RoboCupJunior Soccer TC, 2012]。1 チームのロボットは 2 台までである。競技フィールドには SOCCER A と B があるが競技する広さは 122cm × 183cm で共通である。SOCCER A のフィールドを Fig. 1 に示す。フィールド上のゴールは青と黄色で色分けされるので、色によりフィールド内でのロボットの向きを判別できる（地磁気を使うことも多い）。赤外線を発光するボールを競技に用いることでロボットからのボールの判別を容易にしている。ロボットとロボットのプログラムは参加

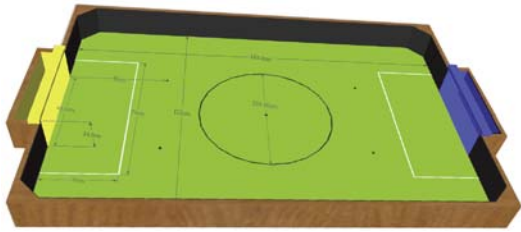


Figure 1: RoboCupJunior Soccer field (SOCCER A) quoted from the RoboCupJunior soccer challenge's rule book. The size of the field is 122cm by 183cm.

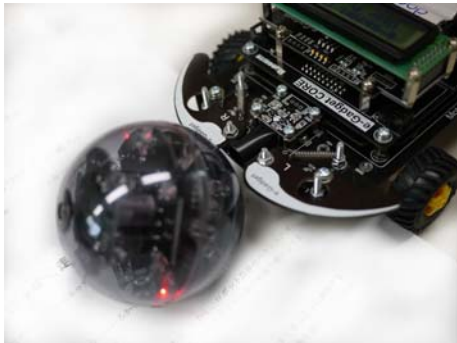


Figure 2: e-Gadget TT robot by Daisen Electronic Industrial Co. Ltd.

者自身が製作することが求められており、市販のキットなどを用いる場合にも外観で他の参加者と同じロボットと判断されないようにする必要がある。

### 3 ロボットプログラミングテキスト

#### 3.1 想定する読者とロボット

テキストの読者としてはロボカップジュニアサッカーチャレンジに興味があるが、いままでに移動ロボットのプログラミング経験のない子ども（小学校中学年から中学生程度）とその保護者や助言者を想定する。テキストを読む前に、ロボットの組立や動作確認、プログラミング環境のインストールまでは終了しているとする。テキストの試作にあたり、ロボカップジュニア参加者の入門や製作のベースに使われている、ダイセン電子工業の e-Gadget TT (Figure 2) と、そのグラフィカルプログラミング環境 C-Style (Figure 3) を想定することにする。C-Style はアイコンをマウス操作で配置することでプログラミングする、キーボード操作を極力不要としたプログラミング環境である。モータ制御、時間待ち、タイマ関数などを実現するアイコンや C 言語の if, while, for などに相当する制御アイコンなどがある。

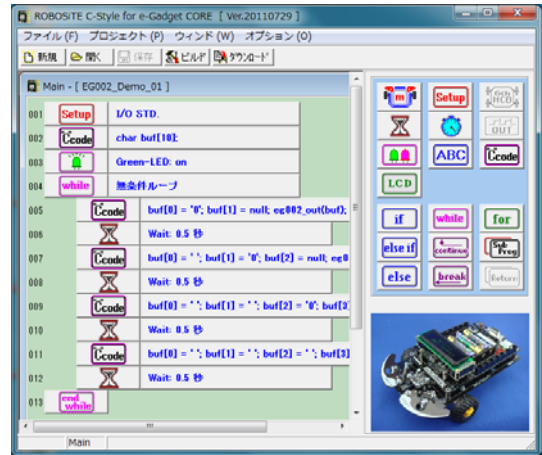


Figure 3: C-style programming environment by Daisen Electronic Industrial Co. Ltd.

#### 3.2 テキストの製作方針

これまでにロボットプログラミングの経験のない著者の一人が実際にプログラムを作成し、その過程を基にテキストを製作する。これにより、年齢によらずロボットプログラミング経験の浅い初心者にとっての理解の鍵となるポイントを記載できると考える。

そのような例には次のようなものがあつた。C-style は手続き型のプログラミング言語であるので、アイコンの表す処理が順次実行される。そして if 文 (に相当するアイコン) は、その時点での値 (たとえばセンサの値) に基づいて処理の分岐をする。ところが、if 文を順次実行処理の一部ではなく、ある条件が成立したときに実行したい内容を書く手段に間違えることがある。たとえばボールセンサ (赤外線センサ) を使ってボールが正面にあるときには直進、それ以外は回転という動作をさせたいとする。この場合に if 文を使うことは早い段階で気づくが、繰り返し if 文で判定するようループを使うことに気づかない場合がある。このようなときに while 文を使うとよいとヒントを与えることが考えられる。

また身体を持つロボットのプログラミングでは実際のロボットをよく観察し理解することが重要である。たとえばロボットを完全に左右対称に製作することは難しく、直進させたい場合に左右のモータに同じ電圧を与えても直進からのずれが発生する。同じ場所へ移動するにも無数の経路がある、質量を持つ物体が急な加減速は難しい、どこで衝突 (ロボットがボールをドリブルしたり蹴るといった場合に起きる) するかというのが重要である、といったことに気づき、考え、理解することが重要であると考える。一方で、プログラムが動いただけで満足してしまうと気づきにつながらない場合もあると考えられる。そこでテキストの中で問いかけを与えることで気づきを促す。

Table 1: Themes in our text book for RoboCupJunior Soccer beginners.

Theme #	Theme title
1	まっすぐ走らせてみよう
2	2m 先でピタリ止ませよう
3	ボールを見つけよう (近く)
4	ボールを見つけよう (1m)
5	フィールドの中でボールを見つけよう
6	ボールを追いかけよう
7	ドリブルをしよう
8	ゴールを見つけよう
9	ゴールを守ろう
10	試合をしよう

### 3.3 テキストの構成

テキストは表紙、目次、凡例のあとに、ロボカップジュニアサッカーチャレンジのフィールドで簡単なロボットサッカーの試合が出来るまでを 10 の段階に分け (Table 1), それぞれのテーマについてヒントとプログラム例を示している. 1~2 段階目がロボットの基本動作, 3~7 段階目がボールを発見し操作する動作, 8, 9 段階目がゴールを守る動作 (ゴールキーパプログラムの作成), そして 10 段階目で総合してフォワードのプログラムを作成する.

テキストの 1 ページを Figure 4 に示す. まず挑戦するテーマがあり, 問いかけと動かししている様子の写真を示す. 次にテーマを実現するために必要と考えられるプログラミング要素 (アイコンまたはアイコンを並べたブロック) を説明する. そしてヒントとして文章でプログラムがどのように動作すれば実現出来るか説明する. また複数の方法で実現が可能であったり, 実世界の理解につながりそうなポイントについては, さらに問いかける. たとえば 2m 先まで走らせるテーマでは, 速度, 時間, 距離の関係について, 気づきを持たせたい. そこで「時間を調節してみよう」「速度を調節してみよう」と働きかけ, 気づきを促している. ほかの働きかけを Table 2 に示す. ヒントの次にそのテーマを実現するプログラムの例を載せ, 一つのテーマについての説明が終了する.

### 4 ロボットスクール受講生による評価

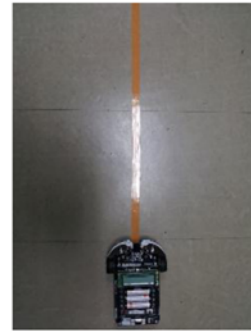
試作したテキストをロボットスクール (教室) を訪れた児童・生徒に見せ, 質問紙によりテキストの評価を尋ねた. ロボットスクールでは C-style を利用している. 回答したのは小学校 2 年生から中学校 2 年生までの 14 名 (男子 13 名, 女子 1 名) で, そのうち 11 名が小学校 4 年生と 5 年生である.

質問紙の質問と回答の選択肢を Table 3 に回答を Table 4 に示す. テキストのテーマのプログラム作成の経験のない

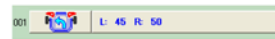
## ①まっすぐ走らせてみよう

考えてみよう

ロボットをまっすぐ走らせるにはどうしたら良いでしょうか?



使うアイコン



このアイコンでモータの回転を指示します。  
LとRに続く値で速度を指定します。  
(説明書 p. 3)

ヒント

左右のタイヤの速度を調節してみよう

LとRに続く値を同じにするとまっすぐ走るように思われます。けれども、左右のタイヤのちょっとした向きの違いやモータのばらつきなどがあるためまっすぐ走らないことがあります。そういう時は左右の値を少し変えてみましょう。短い距離だとまっすぐかわかりにくいいため、ある程度長い距離を走らせてみましょう。

解答例

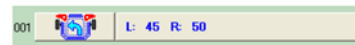


Figure 4: The page of the first theme “まっすぐ走らせてみよう.”

2 名をのぞいた回答を見ると, テキストのテーマのうち, やったことがあるテーマの数は平均で 6.4, できるテーマの数は 4.5, やってみたいテーマの数は 1.2 であった. 回答者にはロボットプログラミングの経験がある程度あり, 自由回答のコメントとあわせるともう少し発展的なテーマについて知りたいという要求があると考えられる.

経験のあるテーマについて, アイコンの説明が分かりやすいとの回答が 5, ヒントが分かりやすいとの回答が 7, 後輩のために「あるといい」との回答が 7 ある. やってみたいテーマについてアイコンの説明が分かりやすいとの回答が 7, ヒントが分かりやすいとの回答が 8 で, ここまでの質問に対する残りの回答は「どちらでもない」である. やってみたいテーマのプログラムについては, なんとなくわかるが 6, たぶんわかるが 5, むずかしそうが 1 である. 残りの 1 名は, やってみたいテーマに関する質問すべてに無回答である.

プログラム作成の経験のない 2 名のうち 1 名は, アイコンの説明やヒントについて「わかりにくい」と回答し, 後輩のために「あるといい」, やってみたいテーマのプログラムは「たぶんわかる」と回答した. もう 1 名はアイコンの説明に「わかりやすい」, ヒントについて「どちらでもない」と回答し, 後輩のために「あるといい」, やっ



Table 2: Hints and questions to readers to enhance understanding of real world

Theme #	Hints and questions	aiming
1	床に直線を描く 左右のタイヤの速度を調整してみよう	直進の難しさへの気づき 上記への対応
2	直進する時間を調節してみよう 直進する速度を調節してみよう	速度, 時間, 距離の関係への気づき 同上
3	無条件ループをヒントとして与える 回転して色々な方向を向こう 回転する速度を調整しよう	手続きとして繰り返しが必要なことへの気づき センサの反応範囲への気づき センサの応答速度への気づき
4	(テーマ) 広い範囲で動かそう 回転する円の大きさを調節しよう	センサの反応範囲(光の到達距離)への気づき 速度と軌跡の関係 同上
5	(テーマ)	複数のセンサ処理, プログラムの実行時間
6	ボールに当たる速度を調節しよう	力積(と摩擦)
7	ボールに当たった時に進む方向を決めよう	反射(衝突した物体の動き)
8	回転する速度を調節しよう	センサの応答速度への気づき
9	ゴールから離れすぎないようにしよう	速度, 時間, 距離, 加速度の関係への気づき
10	(テーマ)	複数の処理の組み合わせ, プログラムの実行時間

てみたいテーマのプログラムは「むずかしそう」と回答した。

回答の傾向をみると, 年齢(学年)の影響はなく, ロボットのプログラミング経験がある(やったことのあるテーマの数が多い)ほど, テキストの説明をわかりやすく感じる傾向があるように見える。とくにやってみたいテーマについて, プログラムがなんとなくわかると回答した6名の, やったことのあるテーマ数は平均7.8と高い。一方で, むずかしそうと回答した2名は経験のあるテーマ数が1または0であった。

したがって, テキストはおおむね理解しやすいが, 未経験の部分については難しく感じる傾向があるといえる。また14名中9名が後輩のために「あるといい」と回答しており, テキストの意義が認められたといえる。未経験の部分について難しく感じるのはこのテキストに限らないと考えるが, 難しく感じても取り組みを続けるためには, このようなスタイルのテキストだけではなく, ロボット教室への参加や, 文書であれば面白さ・経験を伝える物語形式のものが必要かもしれない。

## 5 まとめ

本論文ではロボカップジュニアサッカーチャレンジに参加することを目標としたロボットプログラミングのテキストを試作した。テキストについてロボットスクール受講生に質問紙による評価をし, 後輩のために「あるといい」という評価を得た。一方で未経験の内容について難しさを感じる傾向があることもわかった。今後は実際にテキストを利用することにより有効性を検証するとともに改

良していきたい。

## 謝辞

試作したテキストについてのアンケートにご協力いただいた, 金沢工業大学 出村公成教授ならびにロボカップジュニア石川ブロックのDKT ロボットスクール受講生に感謝する。

## 参考文献

- [三宅, 1985] 三宅なほみ編: 教室にマイコンをもちこむ前に, 新曜社, 1985.
- [子供の科学編集部, 2002] 子供の科学編集部編: ロボカップジュニアガイドブック: ロボットの歴史から製作のヒントまで, 誠文堂新光社, 2002.
- [黒木, 2006] 黒木啓之: RoboDesigner ロボット製作入門-自律型ロボットの作り方-, カットシステム, 2006.
- [RoboCupJunior Soccer TC, 2012] RoboCupJunior Soccer Technical Committee: RoboCupJunior Soccer Rules 2012, ([http://rcj.robocup.org/rcj2012/soccer\\_2012.pdf](http://rcj.robocup.org/rcj2012/soccer_2012.pdf)), 2012.

Table 3: Questions and choices on the questionnaire to ask the evaluation of the text

質問	選択肢
C-sytle のプログラムを書けるか	かけない, すこしかける, まあまあ, よくかける
C-sytle のプログラムを読めるか	わからない, すこしわかる, まあまあ, よくわかる
テキストのテーマのうち, やったことのあるテーマ番号すべてに できるテーマ番号すべてに やってみたいテーマ番号すべてに	1, 2, 3, 4, 5, 6, 7, 8, 9, 10 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
やったことがある/できるテーマについてテキストの アイコンの説明が分かりやすいか ヒントが分かりやすいか あなたの後輩のためであるとよいか	わかりにくい, どちらでもない, わかりやすい わかりにくい, どちらでもない, わかりやすい なくていい, どちらでもない, あるといい
やってみたいテーマについてテキストの アイコンの説明が分かりやすいか ヒントが分かりやすいか プログラムが分かりやすいか	わかりにくい, どちらでもない, わかりやすい わかりにくい, どちらでもない, わかりやすい むずかしそう, わかりそう, たぶんわかる, なんとなくわかる, わかる

Table 4: Answers to the questionnaire

学年	(経験) テーマ数	出来る テーマ数	(挑戦) テーマ数	(経験) アイコン	(経験) ヒント	後輩のために	(挑戦) アイコン	(挑戦) ヒント	プログラム
小5	10	10	0						なんとなくわかる
小5	10	0	1						なんとなくわかる
小2	9	8	1						なんとなくわかる
小5	8	7	2						たぶんわかる
小6	8	0	2						なんとなくわかる
小4	7	3	0						なんとなくわかる
小5	7	7	1						たぶんわかる
小4	6	5	1						たぶんわかる
中2	6	6	0						
小5	3	5	3						なんとなくわかる
小5	2	1	1						たぶんわかる
小5	1	2	2						むずかしそう
小5	0	0	10						むずかしそう
小5	0	0	0	×	×		×	×	たぶんわかる

(経験) はやったことがある/できるテーマについての回答

(挑戦) はやってみたいテーマについての回答

はわかりやすい/あるといい, はどちらでもない, ×はわかりにくい/なくていいを表す



# 人間とロボットのサッカーゲーム実現に向けた考察と開発環境の提案

A proposal new platform based on USARSim toward Realization of RoboCup Dream

和田 拓也<sup>†</sup>, 高橋 友一<sup>†</sup>, 清水 優<sup>‡</sup>

Takuya Wada<sup>†</sup>, Tomoichi Takahashi<sup>†</sup>, Masaru Shimizu<sup>‡</sup>

名城大学<sup>†</sup>, 中京大学<sup>‡</sup>

Meijo University<sup>†</sup>, Chukyo University<sup>‡</sup>

e0830115@ccalumni.meijo-u.ac.jp, ttaka@meijo-u.ac.jp, shimizu@sist.chukyo-u.ac.jp

## Abstract

Since RoboCup began in 1997, the robots have been improving to achieve the dream: a team of fully autonomous humanoid robot soccer players will play soccer games with human players by 2050. In this study, we survey the features of RoboCup soccer leagues and show problems that we have to solve toward realizing human versus robot soccer games complying with the official FIFA rules. We propose a platform based on USARSim that are useful to check new rule setting and sensor algorithms necessary to achieve the dream.

## 1 はじめに

RoboCup はロボットと人間によるサッカーゲームの実現を夢として、関連の研究技術の進歩を目的にしている。その目的のために各種リーグが開催されている。例えば中型リーグでは 1997 年から比較するとコートが拡大される、コート横の壁がなくなる、自分の位置を計算するためのランドマークが取り外されるなど、確実にロボカップの夢に向けて進歩している (図 1)。一方で、人間とサッカーをする視点から見ると、FIFA のルールでサッカーをするためにはコートの大きさや人間型ロボットの開発に加え、ロボット同士では使用できるセンサーが使用できないなど課題は多い。

RoboCup Rescue で使用されている Urban Search And Rescue Simulation(USARSim) は、フィールドの

設定に加え、レンジセンサやカメラなどの各種センサモジュールをもったシミュレータである [1]。現在 Mixed Reality Simulation(MR) リーグや小型リーグでは、ロボットはフィールド全体を写すグローバルビジョンからすべてのロボットの位置情報を得る。フィールド内のロボットの位置情報を取得するグローバルビジョンシステムや、中型リーグで使用されている全方位カメラ<sup>1</sup>も、人間とサッカーをするロボットのセンサとしては中間点であり、最終的にはロボットはそれぞれのロボット本体に装着されたカメラから得られる前方の画像を元に行動する必要がある。

今回我々は最終的な人間対ロボットのサッカーゲームを意識し、アルゴリズムの検討を行う環境として USARSim を利用したサッカーシミュレータを提案する。その特徴は、フィールドサイズはロボットの性能に応じ変化を付ける、実際のロボットに則したセンサーをロボットに搭載できるなど、FIFA のルールに則った人間とロボットのサッカーゲームを実現するにあたってのリーグのルール検討やセンサアルゴリズムの開発に使用できることにある。本報告では (1)USARSim の ImageServer を用いて、実機リーグのロボットと同様にそれぞれのロボットにつけられたカメラからの画像からロボットがまわりのロボットの位置情報を把握し行動するクライアントプログラムと、(2)MR リーグ相当のサッカーゲームを実現した。それにより、今後人間とロボットがサッカーゲームをするために必要とな

<sup>1</sup>FIFA のルールに従えば、全方向の情報が得られるカメラは使用センサとして許可されない。

る実機リーグのルール設定やロボットアルゴリズムの開発環境の有効性を示した。

## 2 背景

RoboCup2007Atlanta 大会で、中型リーグのロボットと人間のサッカーゲームのデモンストレーションが実施された [2]。その内容は初期に提案された RoboChallenge の課題に対して、この 10 数年の進歩を示すものであった。一方で、2050 年までに FIFA ルールで人間とロボットでサッカーゲームをするという RoboCup の夢を実現するには多くの課題が残っていることを示した。

現在 Robocup サッカーリーグでは様々なリーグが開かれている。表 2 に人間のサッカーリーグ、実機リーグ、シミュレーションリーグの比較表を示す。 $P_{number}$  はプレイヤー 1 名あたりのフィールドの広さ、 $P_{space1,2}$ ,  $P_{grid}$  はプレイヤーの体の大きさを 1 単位としたときのフィールドの縦横のサイズ、 $P_{time}$  は人間の走力に対するフィールドの大きさの相対的な長さを表している。中型リーグと小学生のサッカーで、 $P_{time}$  の値が同じである。Robocup の目標は、ロボットと人間の試合を実現することである。この様な相対値は、ロボットのサッカーゲームと人間のゲームでは同じレベルが望ましい。

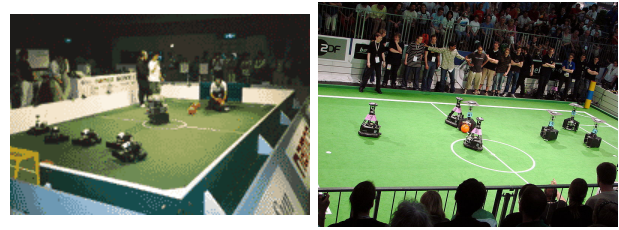
現在のサッカー実機リーグにおいて、この値を人間リーグと同じにしようとするには問題がある。ロボットを大きくするにもフィールドを広くするにも多くのコストがかかる。実現するにも、大学などの施設内で準備できる範囲を越えている。一方、シミュレーションならば人間と同じサイズのフィールドとロボットを用意し、動きを検証することができる。

ロボカップにおいても、シミュレーションリーグを始め、多くのシミュレーション環境がある。そこではセンサ環境やロボットの動きはモデル化され、実機とは隔たりがある。

## 3 提案するプラットフォーム

### 3.1 USARSim と ImageServer

図 2 に提案する開発環境を示す。下段が USARSim 環境、上段が Mixed Reality リーグの環境である。ま



(a)1997 年

(b)2011 年

図 1: 中型リーグの試合

た、図 3 に提案する環境に対応する実際の MR リーグのシステム構成図を示す。図 2 の左の Client (ロボット)、Soccer Server からみると両システム同じインタフェースである。

MR リーグ、または USARSim を利用した Mixed Reality Simulator では、上に設置したカメラからフィールド全体を撮影していた。このグローバルビジョンでロボットの位置を取得し、それを各ロボットのクライアントプログラムに送っていた。USARSim では、カメラをシミュレーション環境の任意の位置、又可動ロボットに取り付けることができる。そのカメラ画像は ImageServer を介して各クライアントに送信する。ImageServer とは USARSim の中に設定されたカメラからの画像をキャプチャすることのできるアプリケーションである。我々は ImageServer を使って、ロボットが得た画像をそれぞれのクライアントが取得できる環境を実装した。

### 3.2 MR リーグとの比較

MR リーグで使用されているロボット、CITIZEN EcoBe を用い、11 対 11 のサッカーゲーム環境を構築した。表 2 に MR リーグとの比較を示す。提案するプラットフォームで MR リーグと変更した点を以下に示す。

- フィールドサイズを大きくすることで、プレイヤー数とフィールドサイズの比率を合わせた。
- 実機では実装されていないレンジセンサとカメラを前面に装備し、ローカルなレンジデータとカメラ画像をロボットが取得できるようにした<sup>2</sup>。

<sup>2</sup>両センサとも、大きさ、重さとも 0 の仮想的なセンサーとして装着した。

表 1: 人間のサッカーリーグと RoboCup リーグの比較

$$Pnumber = \frac{\text{Field Area}}{\text{Player Area}}, Pspace1 = \frac{\text{Long Side of Field}}{\text{Player Width}}, Pspace2 = \frac{\text{Short side of Field}}{\text{Player Depth}}, Pgrid = \frac{Pspace1 \times Pspace2}{\text{Players}}, Ptime = \frac{\text{Field long side}}{\text{Player Speed}}$$

Leagues	Field Size (m*m)	Players	Metrics							
			Player Size (W*D(*H) cm)	Speed (m/s)	Pnumber	Pspace1	Pspace2	Pgrid	Ptime (s)	
Human League	TopLeague	105*68	11 vs. 11	50*30*180	9.09	325	210	227	2164	12
	JuniorHighSchool	70*50	11 vs. 11	30*20*130	6.25	159	233	250	2652	11
	ElementarySchool	68*50 (halfsize)	8 vs. 8	30*20*100	6.25	213	227	250	3542	6
Real Robot League	Middle (2010)	18*12	5 vs. 5	(30-50)*(40-80)	3.00	22	36	30	108	6
	Middle (2003)	10*5			3.00	5	20	13	25	6
	F180 (2010)	6.05*4.05	5 vs. 5	18*14	3.00	2.5	34	23	76	2
	F180 (2003)	2.8*2.3			1.00	0.6	16	13	20	6
	SPL/Humanoid	6*4	3 vs. 3	10*5*58	0.10	4.0	60	80	800	60
	MR	0.928*0.523 (42inch)	5 vs. 5	2.5*2.7*2.8	0.04	0.05	37	19	72	23
Simulation League	2D	105*68	11 vs. 11		1.20	325				88
	3D	105*68	11 vs. 11			325				
	USARSimMR	0.928*0.523 (42inch)	5 vs. 5	2.5*2.7*2.8	0.04	0.05	37	19	72	23

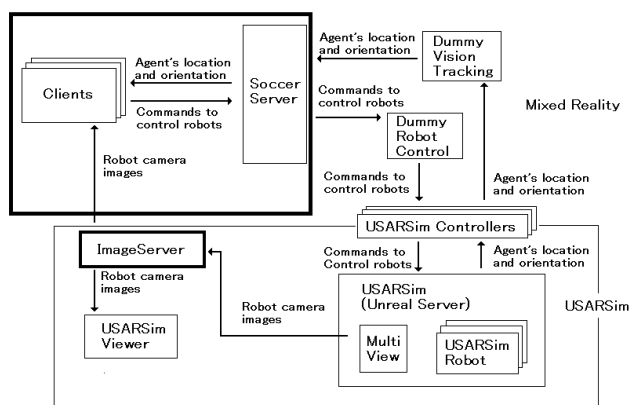


図 2: 提案するサッカーロボット開発環境

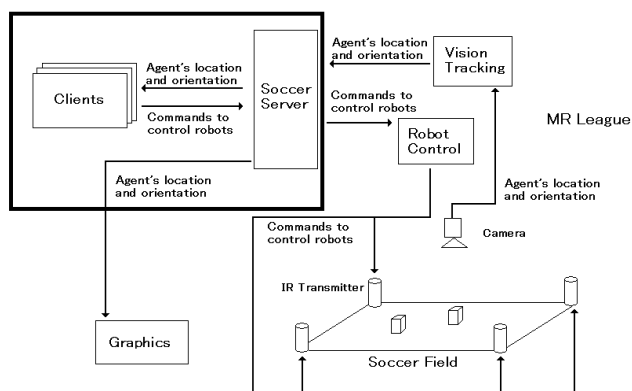


図 3: Mixed Reality リーグ構成図

表 2: MR リーグとの比較

	MR リーグ	提案プラットフォーム
ロボット数	5vs5	11vs11
Field の大きさ [m×m]	0.934×0.525	1.868×1.050
センサ	無	レンジセンサ カメラ

## 4 実現システムと考察

### 4.1 MR リーグベースの実験

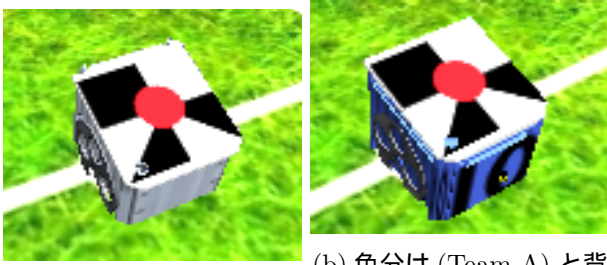
MR リーグではロボットの位置、チームを識別するために、ロボット上面に個体識別と向きを検出可能なマークを貼っている (図 4)。提案するシステムでは、ロボットに人間プレイヤーと同じ背番号とし、本体の色 (青と黄色) を用いて識別する。

ローカルビジョンを用いたロボットの画像処理の内容を以下に示す。

1. 青黄チームを画素の RGB 情報から識別する。
  - 各画素の Red, Green, Blue の値を比べる。
    - RGB 値の入力
    - ラベリングによる領域抽出
2. ラベリングされた塊の画素数が 500<sup>3</sup>以上のものをロボットであると判断する。

<sup>3</sup>カメラサイズは 320\*240 であり、フィールドの中央から端にいるロボットを写すと 25\*25 程度で写るため、数字の部分を検討し閾値を 500 とした。

図 5 に黄色チームの一台のロボットのカメラの入力画像と、それを画像処理した結果を示す。処理前の画像から、向こう側に相手チームの左から 7,11,3 番のプレイヤーと味方チームの 2,6 番のプレイヤーがいることがわかる。処理後の画像では左側に敵が、右側に味方がいることがわかるが、敵の 3 番と 11 番の領域が繋がってしまい、敵の数が 2 体になっている。



(a) オリジナル

(b) 色分け (Team A) と背番号

図 4: システムに使うロボット : EcoBe



(a) ImageServer からの画像

(b) ローカルでの画像処理

図 5: 敵 (7,11,3 番) と味方 (2,6 番) のラベリング

## 4.2 実験内容と考察

2 種類のチームを用いてサッカーゲームを行った。両チームとも FW3 体, MF3 体, DF4 体, KP1 体で構成される。チームの味方位置の取得方法の違いを表 3 示す。Team A は味方の位置情報をグローバルビジョンで得るのに対し、Team B はローカルビジョンから味方の位置情報を得る。両チームとも、自分のフィールドに対する位置はグローバルビジョンから得る。また、Soccer Server から自分とボールの位置関係が送られる。以下に例として FW クライアントのアルゴリズムを示す。



図 6: USARSim 上で動作した 11vs11

```

senser() //位置情報を得る
if ボールを持っていない
    if ボールが自陣にある
        if ボールとの距離がコート全体のサイズの 1/6 よりも近い
            Go to ball.
        else
            Back to home position.
    else
        Go to ball.
else
    if 自分の蹴る範囲にゴールが無い
        if 自分よりも前にフリーの味方がいる
            Pass for teammate.
        else
            Dribble.
    else
        Shoot.

```

表 4 に、グローバルビジョンの方が強いという結果が見える。表 5 に 1 ループあたりにかかる時間を示す。ローカルビジョンはグローバルビジョンと比較して 2.9 秒と 5 倍弱処理時間がかかる。一方、Soccer Server がロボットに命令を送る 1 ループは 66 ミリ秒で、現在のところ、クライアントからロボットに対する命令が送られてくるまでは、最後に送られてきた命令を繰り返しロボットに送る。senser() によって位置情報を得たときと命令を送ったときでは表 5 に示す時間分ラグが生じる。このことより、Team A と Team B の対戦結果は命令送信サイクルの差から当然である。

このシミュレーションでは 2.9 秒間にロボットが 10cm 以上進んでしまう。ロボットのサイズが 2.5cm\*2.7cm なので、誤差をロボット 1 台分まで減らすことを目標とすると、1 ループを 0.7 秒以下に収める必要がある。

表 3: 2 チームの比較

	Team A	Team B	
自分以外の位置 を得る Vision	Global	Global	自分の position
		Local	まわりの position

表 4: 試合結果

試合内容	結果
Team A vs Team A	1-0
Team A vs Team B	6-0

表 5: プログラムの 1 ループあたりの時間

	プログラム	平均時間
クライアントプログラムが サーバに命令を送信する	Team A	0.602[s]
	Team B	2.881[s]
サーバがロボットに 命令を送信する	Soccer Server	0.066[s]

グローバルビジョンとローカルビジョンにおいては `senser()` 関数で得られる情報が全体と部分という相違がある。小型リーグにおいて、当初はローカルビジョンのチームがあったが、現在グローバルビジョンのチームだけであるように、特に小型のロボット対ロボットのサッカーゲームでは現在使用されていない [5]。しかし、人間対ロボットのサッカーゲームを、ローカルビジョンを用いたプラットフォームを小型ロボットに対しても提供できる。

## 5 まとめ

人間とロボットがサッカーゲームをするという夢を実現するにあたり、実機を用いては検証が困難なテーマが多々ある。同じロボット同士のサッカーゲームであれば、ゲームをスムーズにするためにセンサに工夫することは当然である。一方で人間とロボットが FIFA のルールの下でサッカーゲームをするとなれば、人間と同じ条件をルール上に明記する必要がある。本論文ではその中でも特に自分以外のロボットを認識するにあたって、特殊なマークやシステムを使用しない状況を実現するために USARSim プラットフォームを提案した。

そして、試作したプラットフォームを用いた 11 対 11 のゲームを用い、ロボットに搭載したローカルビジョンによるゲーム結果を示した。このプラットフォームにより、人間とのサッカーゲームで必要となる色やパ

ターンによるチーム識別、番号 (顔や体型) によるプレイヤー識別の研究を、MR リーグに限らず、中型、小型リーグにおける次の段階のルールにあわせ、検討できる可能性を示した。

## 参考文献

- [1] <http://sourceforge.net/projects/usarsim/>
- [2] <http://www.robocup-us.org/Old/robocup-2007/index.html>
- [3] 'RoboCup' Soccer Match Is a Challenge for Silicon Rookies  
<http://www.sciencemag.org/content/277/5334/1933.full>  
2012/4/9
- [4] RoboCup Middle Size League - Wikipedia, the free encyclopedia  
[http://en.wikipedia.org/wiki/RoboCup\\_Middle\\_Size\\_League](http://en.wikipedia.org/wiki/RoboCup_Middle_Size_League)  
2012/4/9
- [5] 高橋友一, 秋田純一, 渡辺正人. 小型ロボットの基礎技術と製作 -RoboCup 小型リーグへの挑戦 共立出版, 2003.

# ロボットオペレータチーム訓練環境の一提案

A study of the exercising simulator for team robot operators

清水 優, 高橋 友一

Masaru Shimizu, Tomoichi Takahashi

中京大学 情報理工学部, 名城大学 理工学部

Chukyo University, Toyota shimizu@sist.chukyo-u.ac.jp

Meijo University, Nagoya ttaka@meijo-u.ac.jp

## Abstract

Remote controlled robots have been used in Fukushima nuclear plants. The tasks at Fukushima present necessary issues on robots. Drills of operators who manipulate robots at disaster areas are necessary as well as developing robots. We proposed a mechanism that creates dynamical changing environments. In this paper, we proposed an exercising environment using the mechanism for people drill team robot operations under dynamical changing environments. System architecture is described and experiment results of using the system are discussed.

## 1 はじめに

近年日本では、大規模自然災害が数年に1度発生し、多くの人命が失われている。多くの人命を救うため、倒壊家屋内部に取り残されている要救助者（以下 VICTIM）を探索し、救助計画の基礎情報となる倒壊家屋内地図を作成するレスキューロボットの研究が盛んに行われている [1]。

レスキューロボットの開発とともに、研究者間で共通の性能評価用フィールドを利用できなかったり、ロボットの比較実験が困難である等、評価の標準化が難しいという実機ならではの問題点もある [2]。又、現在はレスキューロボットオペレータの訓練も必要とされてきている。福島原発では、レスキューロボットが人間が入れない環境を調査するために使用された。そのときのオペレータの話から [3]、(1) 瓦礫が散乱してい

るような特殊な環境でのロボット遠隔操作はあらかじめ訓練が必要であること、(2) ロボットが無いと訓練できない（ロボットを操縦できる環境にある人が世界中に多くいるわけではない）ことが述べられている。

実際の福島原発の現場のように、予期せぬ移動や崩壊がある、無線接続状況の変化など動的な環境変化をシミュレーションに導入する事で、より現実に近いロボットのテスト環境、オペレータの訓練環境を用意できる。

この動的環境変化は、シミュレータ外部からのコマンドによる制御だけでなく、あらかじめ動的環境変化を設定したシナリオによる制御も可能になる。併せて、任意のタイミングで災害イベントを発生できるシナリオベースの動的環境下において、ロボットオペレータの訓練環境を提案する。2章では、ロボットオペレータ訓練の問題点を示す。3章では、提案するレスキューロボットオペレータ訓練環境について述べる。4章では、チーム訓練の評価について述べる。5章では、実際に行ったオペレータ訓練の結果とその妥当性を検討する。6章で全体をまとめる。

## 2 UARSim とロボットオペレータ訓練

### 2.1 USARSim

レスキューロボットで用いられる代表的なシミュレータに、市販ゲームソフトウェア Unreal Tournament 2004(以下 UT2004) を元に開発された USAR-

Sim(Urban Search and Rescue Simulation)がある[4]. USARSimは、瓦礫、火災、被災者、レスキューロボット(本体や移動機構、センサ等)の基本モデル、レーザレンジファインダなどの距離センサなどの実在する部品も多数用意されている。また、動力学演算によるリアリティのある3Dシミュレーション画像がカメラ画像として得られる。そのリアルな画像は、ロボットビジョンや環境認識、行動計画やロボットオペレータの訓練に利用できる。

USARSimは、図2に示すように、USARSimは物理演算ユニット(Karma Engine)とイベント処理ループを中核とするUnreal Engineを持っており、物理演算など必要な演算は自動的に行う。ロボットユーザインタフェースプログラムは、ネットワーク接続によるロボット操作コマンドやセンサによる計測データを送受信するクライアントプログラムを記述する事で、レスキューロボットとのユーザインタフェースを構築できる。

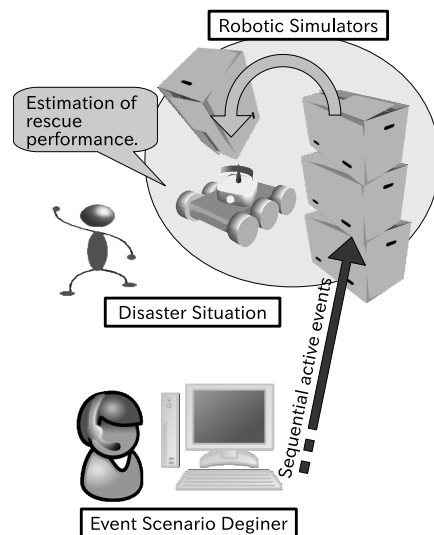


図1: レスキューロボットシミュレータフレームワーク。レスキューロボット開発者は、動的環境変化によってロボットの挙動を評価できる

## 2.2 ロボット訓練での問題点

これまでのレスキューロボットシミュレーションでは、崩れない瓦礫、変化しない火災、温度分布、CO<sub>2</sub>濃度、電波環境、位置が固定されたVICTIM等「静的な環境」であった。実際の災害現場は環境が変化する場所であり、図1に示すように、レスキューロボットオペレータの訓練のために、動的環境変化を与えられるシミュレータが必要である。

自動車や建設作業機械と同様に、ロボットオペレータもレスキューロボットを操作して訓練する必要がある。ロボットを使用する予定の環境での訓練や、チームミッションの場合はチームでの訓練が必要となる。2011年の福島原発事故での作業員は、以下の問題点を指摘している[3]。

- 訓練時間不足(ロボットの台数不足, ロボット充電中は訓練できない)
- ロボットが用意できない場合は, 訓練できない
- チームでの訓練不足

シミュレータを用いる一般的な利点として、実機ロボットが無くても訓練可能、ロボットが壊れる危険が

無い、という点に加え、以下の点もレスキューロボットオペレータの訓練という観点では重要である。

- レスキューロボットが活動する環境を構築しやすい
- 複数の視点を設定でき、容易に視点を切り替えられる
- 遠隔地に分散しているオペレータによるチーム訓練を再現可能である

## 3 レスキューロボットオペレータ訓練環境

### 3.1 訓練用ロボットとユーザインタフェース

動的環境変化が可能なシミュレータを用いて、ロボット操縦訓練環境を図3に示す。(a)は使用したUSAR-Simに標準搭載されている車輪型ロボットP2ATを示す。



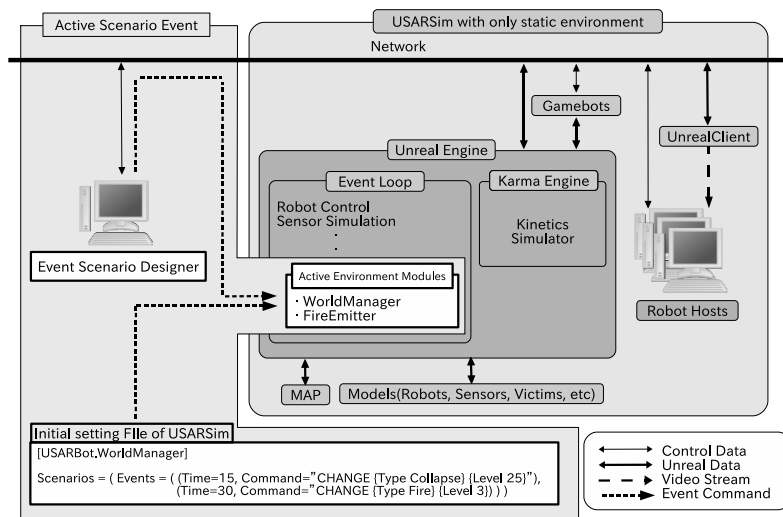


図 2: USARSim の動的環境変化コマンド/データ流れ図. 白枠は動的環境変化のための新しいモジュール

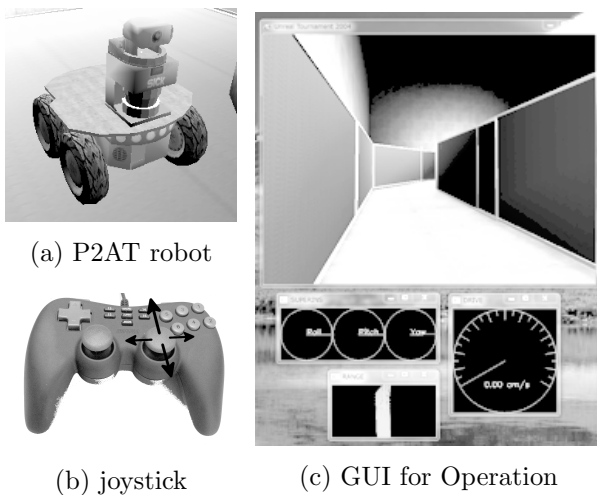


図 3: 訓練に使用したロボットの外観, 操縦用ジョイスティック, メータ付 GUI

P2AT<sup>1</sup>は、カメラ、傾斜計、速度計、レンジファインダ（ロボット前方 180 度を計測）を搭載している。(b)は操縦用アナログジョイスティックで、オペレータがスティックを倒すと、ロボットは倒した方向に進む。また、スティックを倒した傾きに応じて、速度を増減する。(c)の操縦用 GUI は、ロボットに搭載したカメラからの映像と傾斜計、速度計、レンジファインダ表示で構成した。図上半分がロボットカメラからの映像、

<sup>1</sup>P2AT は、挙動が分かりやすい車輪型移動機構と訓練に必要なセンサを備えているため、今回の訓練で使用することとした。

その下左側が傾斜計、右側が速度計、最下部がレンジファインダの表示である。オペレータは、ロボットカメラの映像を見て、センサ表示を確認しながらロボットを操縦する訓練を行った。

### 3.2 訓練用コースによるロボット操作習熟

全くの初心者も存在する事を考慮して、初心者向けコースで、ロボットの機体幅の感覚を体得してもらい、その後訓練コースへ進みロボット操作訓練を行った [5].

図 4 (a) は初心者向けコースの全体を上空から見た図である。オペレータはロボットを操作して、12 時の位置からスタートして、時計回りにこの単純な周回コースの中を移動する。

図 5 は、でこぼこの地面や坂道、岩や壁等の障害物によって難易度が高くなっている訓練用コースの 1 例である。初心者コースではコースの曲がり角にロボットが引っかかることがあったが、ここではさらに岩や壁などが積極的にロボットの走りやすいコースを遮るように配置されている。

図 6 は、図 5 のコースのでこぼこや坂道の寸法図である。コース最初の曲がり角を越えると、斜度 22.5° の小さい山が 2 つある。それを過ぎると、斜度 11.3° の長い坂を上るように配置した。

図 7 は、コースの一部に崩れる箱を設置した動的環

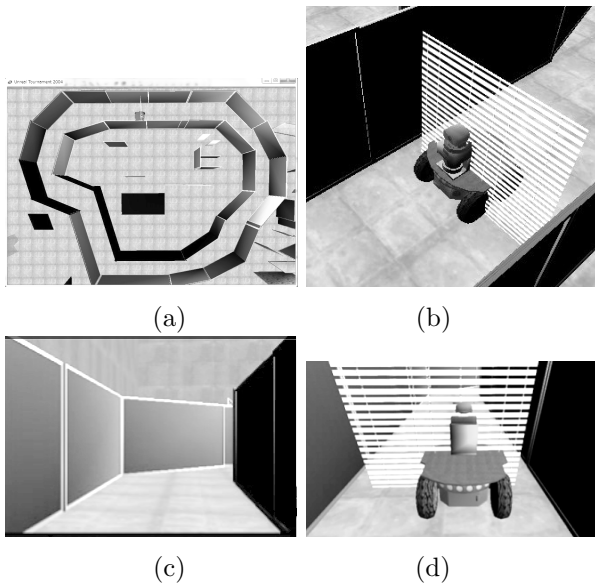


図 4: 初心者向け低難易度コース. (a) シンプルなコースの全体; (b) スタート地点; (c) ロボット前方カメラからの映像; (d) ロボット後部設置カメラからの映像, ロボットが写っている.

境変化を導入した訓練コースを示す. 図 7 (a) は, 図 5 のコースの右上部分である. 動的環境変化の一つである崩れる箱を使って, 崩れる壁 (図 7 (a) 上方) と崩れる道 (図 7 (a) 下方) を構成した. 崩れる壁は, シナリオ機能によりロボットがちょうど横を通り過ぎるときに, 自動的に崩れるように設定した. 図 7 (b) の崩れる道は, ロボットがこの部分で立ち止まると崩れる. オペレータは, 画面の揺れや傾斜計の動きで地面が不安定である事を知ったならば, 止まらずに通らなければならないことをこの訓練コースから学ぶ

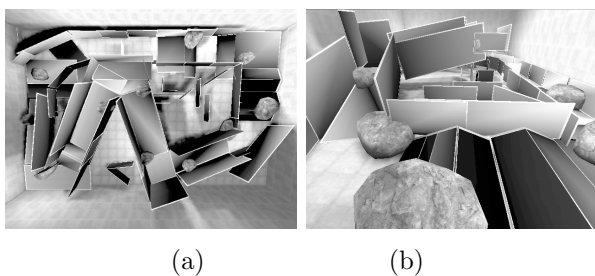


図 5: 訓練コース (a) 天井からのコース全体像; (b) (a) の右方から見たコース側面, 数カ所に斜面がある.

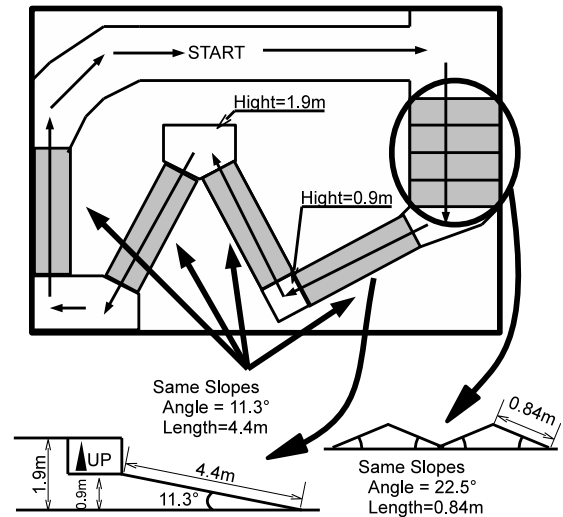


図 6: 訓練コース解説. ロボットは時計回りにコースを進行する. 最初の右折の後, 2つのでこぼ道がある. 1つの斜面は, 斜度  $22.5^\circ$ , 長さ  $0.84\text{ m}$  である. 2つの小さな丘の後, 長く高い道がある. この道は4つの長い坂道で構成されている. 1つの坂道は斜度  $11.3^\circ$ , 長さが  $4.4\text{ m}$  である. 最も高い部分は, 地面から  $1.9\text{ m}$  である.

事ができる. 図 7 (c) は, 壁が崩れた状況を示す. オペレータは, 場合によっては障害物を避ける操作をしなければならない. 図 7 (d) は, 崩れる道上でロボットを停止させた瞬間であり, すでに道は崩れ始めている事がわかる.

#### 4 チームによるオペレーション評価

福島原発のロボット投入にあるように, 災害現場ではロボット1台ではなく複数台がチームを組む. 表1は, 2名のオペレータによって, チームミッションの訓練を行った結果である. チームミッションであるので, 2台ともがスタート地点に帰還した時間を計測した. チームミッションの必要性を構成する為に, 高所コースの壁を削除した. これにより, 「ロボットが高所から落下する可能性がある」という心理的ストレスも増加する. チームミッションの目的は, お互いの安全を確認しながら, かつ手際よくミッションを遂行する (周回時間が短くなるように努力する) こととした. また, チームミッションの場合, チームリーダーの善し悪

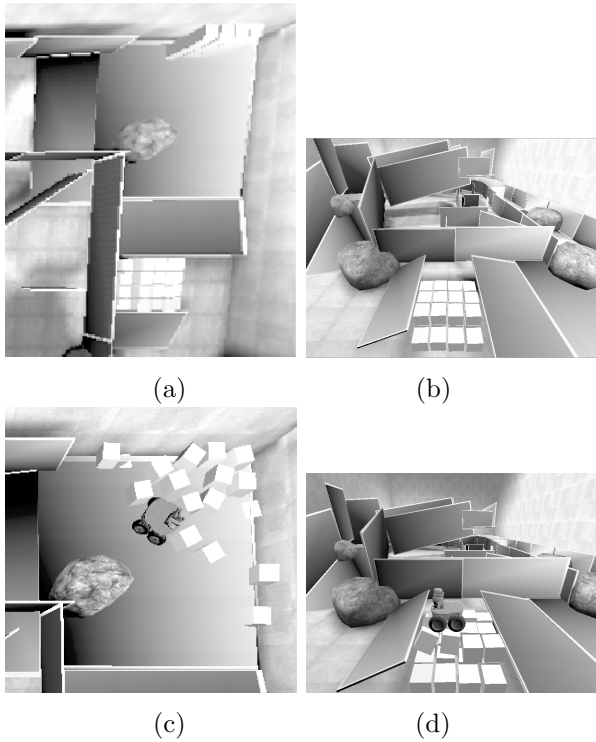


図 7: 動的環境変化を付加した訓練コース. (a) 動的環境変化オブジェクト (白い箱) の配置を上から見た図; (b) 崩れる床 ((a) 下部の拡大図). (c) シナリオにより自動的に崩れた白い箱; (d) ロボットが通過する事により崩れた白い箱.

しがチーム全体の成績を左右することはよく知られている. そこで, 以下のケースで周回時間を計測した.

- OP<sub>1</sub> がリーダーで, OP<sub>2</sub> が後続かつ OP<sub>1</sub> の補助を行う場合
- OP<sub>2</sub> がリーダーで, OP<sub>1</sub> が後続かつ OP<sub>2</sub> の補助を行う場合
- OP<sub>1</sub> と OP<sub>2</sub> がその時の状況で, 指示を出すオペレータを決めた場合 (途中で変更可能)

さらに, 計器の表示がある場合とない場合での違いも計測した.

後続のオペレータが行う補助は, 主に先頭ロボットの全体および周囲の画像を撮影し, 先頭オペレータに見せる事であった. チームミッション中の会話では, チームリーダーから「私のロボットの左側をあなたのロボットのカメラで撮影してください」等, 共同作業を

表 1: チームミッションの結果.

conditions		time (s)		num.of	
GUI panel	robots	avg.	std.	pauses	failures
-	OP <sub>1</sub>	117.1	3.2	4	0
	OP <sub>2</sub>	108.3	2.5	6	0
	both	112.7	5.2	10	0
indicator	OP <sub>1</sub>	122.8	7.1	5	0
	OP <sub>2</sub>	92.7	2.3	0	0
	both	107.7	15.9	5	0

“robots” column shows which operator control the robot.

円滑に行う為の指示が出されていた. 表 1 からは, 以下の事がわかる.

- OP<sub>2</sub> が先頭するとき, 周回時間が短い
- センサがあっても OP<sub>1</sub> がリーダーのときは周回時間が長くなっている
- センサがある時, OP<sub>2</sub> がリーダーでは周回時間が短くなっている

ここまでの訓練で, OP<sub>2</sub> は OP<sub>1</sub> よりも 10 秒~20 秒以上短い周回時間であった. つまり, OP<sub>2</sub> のように能力に余裕があるオペレータがリーダーを務めるとチームとして良い結果が得られる. 提案しているシミュレーション環境は, 個人別の習熟度を評価でき, さらにチームミッションにおいても誰がチームリーダーになればミッションを安全かつ円滑に進める事ができるのかを示す事ができる. また, オペレータ間の支持や返答は, 共同作業に習熟するに従って短くなっていった. すなわち, シミュレーションでも, チームミッションとして連携が円滑に行える程に訓練できたことがわかった. これらは, ロボットオペレータの訓練やチームミッションでの訓練に, 提案するシミュレーション環境が有効である事を示している.

## 5 考察

我々が提案したシミュレーション環境でのレスキューロボットオペレータの訓練から判明した事柄を, 以下

に示す。

1. 提案した訓練用シミュレーション環境は、現実同様の訓練フィールドとして使用できる。現実の訓練と同等の効果は、シミュレーションを利用した訓練でも再現された：

- 繰り返しの訓練は、オペレータを早くロボット操作に習熟させた
- 一旦習熟した後、坂道で脱輪させるなど慣れによる致命的なミスも見られた
- 動的な環境変化（突発的事態の発生）によって、オペレータは周囲を注意深く観察するようになるため、周回時間が長くなった

2. チームミッションの訓練では、当初 OP<sub>1</sub> と OP<sub>2</sub> は音声による意思疎通を行っていた。数回の訓練の後、彼らは自動的にそれぞれの役割を実行していた。

- OP<sub>1</sub> は、自身のロボットを写した映像が必要なとき、自動的に2台目のロボットの到着を待った
- OP<sub>1</sub> は、2台目のロボットが難所にさしかかった時、自動的に2台目ロボットとその周辺画像を撮影した
- OP<sub>2</sub> は、1台目ロボットの姿が映像に入るように、自動的に速度を調整した

チームミッションの訓練では、我々が提案するシミュレーション環境はロボット開発やロボットオペレータ訓練に貢献できる可能性を示した。特に福島原発事故で報告があるように [3]、作業環境や使用するシステム、ロボットオペレータのチームメンバが制限されることがある。事故処理では事態収拾に一刻を争うので、平常時からチームミッション訓練やオペレータ組み合わせの選定を行っておく必要がある。それには、提案しているシミュレーション訓練の活用が適している。

## 6 まとめ

本報では、シミュレーションによるレスキューロボットオペレータの訓練環境を提案した。動的環境変化を付加し、突発的な事態への対処も訓練できるシミュレー

ション環境を提供した（動的環境変化のロボットオペレータ訓練環境への適用例を示した）。

動的環境変化を付与した USARSim によってロボットオペレータ訓練環境を構築し、段階的な訓練環境の例およびチームミッション訓練結果を示した。その結果、ロボットオペレータに欠かせない「突発的事態への対処能力」も訓練可能であることがわかった。シナリオによる動的環境変化は、複数のオペレータに対して共通の「突発的事態」を与える事もでき、また対応に習熟するまで繰り返すことも可能である。このように、本提案による訓練用シミュレーション環境は、ロボットオペレータの訓練に有用である。

## 参考文献

- [1] 田所 諭, 北野宏明: ロボカップレスキュー - 緊急大規模災害救助への挑戦, 共立出版 (2000).
- [2] 田所 諭, 高橋友一: ロボカップレスキュー (特集) 自律分散型サッカーロボット, 日本ファジィ学会誌, Vol. 14, No. 6, pp. 593-601 (オンライン), <http://ci.nii.ac.jp/naid/110002912622/> (2002-12-15).
- [3] GUIZZO, E.: Fukushima Robot Operator Writes Tell-All Blog, Technical report, <http://spectrum.ieee.org/automaton/robotics/industrial-robots/fukushima-robot-operator-diaries> (2011.10.23).
- [4] Carpin, S., Lewis, M., Wag, J., Balakirsky, S. and Scrapper, C.: Bridging the gap between simulation and reality in urban search and rescue, *Lecture Notes in Computer Science*, Vol. 4434, p. 1 (2007).
- [5] Shimizu, M. and Takahashi, T.: Simulation Platform for Performance Test for Robots and Human Operations, *2011 AAI Fall Symposium Series* (2011).