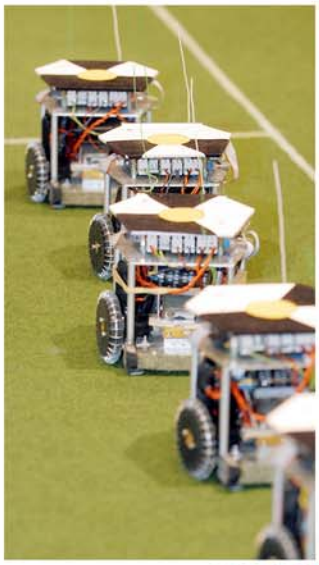
人工知能学会
第25回SIG-Challenge研究会



RoboCup

2007

2007年5月2日
インテックス大阪（大阪市）
ジャパンオープン2007

目次

社団法人 人工知能学会
Japanese Society for
Artificial Intelligence

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-A701-1 (5/4)

災害救助シミュレーションにおける確率モデルによるエージェントの評価方法

An Evaluation Method using Probability Model for Agents in Disaster Rescue Simulation

高橋　友一
Tomoichi Takahashi
名城大学 理工学部 情報工学科
Meijo University, Nagoya
ttaka@ccmfs.meijo-u.ac.jp

Abstract

Agent-based simulation makes it possible to simulate social phenomenon by presenting human behaviors by agents. Taking disaster and rescue simulation as an example, we discuss agent based social simulations from viewpoints whether they can be used for practical use or not. First, possibility in applying agent approaches to social tasks is shown by comparing simulation results with other methods. Next, we propose a method to present agent behaviors with probability model and discuss results of applying the method applied to RoboCup Rescue Simulation data. These will delve into future research topics for developing agent based social simulations to practical ones.

1　まえがき

人間活動を要因とする社会現象は，自然現象と異り再現性の保証がなく，実験することも難しい．従来は，事例研究によるデータからモデルを構築し，評価・考察する手段がとられてきた．最近では，構成要素の自律的な動きをエージェントと呼ばれるプログラムで表現し，社会現象をシミュレーションするアプローチが注目を集めている．

エージェントベースのアプローチは多種多様な分野に適用されている．社会システムの分野においても，シミュレーション実験の新な手法を提供している [2]．一方で，シミュレーション結果を実験データとつきあわせて検証する事が難しく，科学・工学分野と同様に，仮説に基くモデルの作成，モデルに基くシミュレーション・評価などの手法を採ることができない．寺野は，人間活動を含む社会現象をシミュレーションシステムに対する要求事項として以下をあげている [8]．

1. 現実と整合的な結果が得られる事．
2. 既在の理論では説明が困難な現象が示せる事．
3. シミュレーション結果に満足できる事．
4. 結果の妥当性を評価できる事．
5. 既在の理論では説明困難な課題に対して接近できる事．

本論文では，災害救助シミュレーションを例に，エージェントベースのシミュレーション結果のタスクに依存した評価例とタスクに依存しない解析手法について述べる．

2　災害･救助シミュレーションの評価

2.1　自治体における用途

自治体は過去の災害データを元にして，地域にあった被害推定モデルで災害被害を想定している [6]．阪神淡路大震災以降の都道府県・政令都市における地震被害想定調書の特徴として，住民が生活する上で支障になる項目や被害と対策の時間推移に関する予測（シナリオ被害想定）などの項目が追加されている [6]．過去

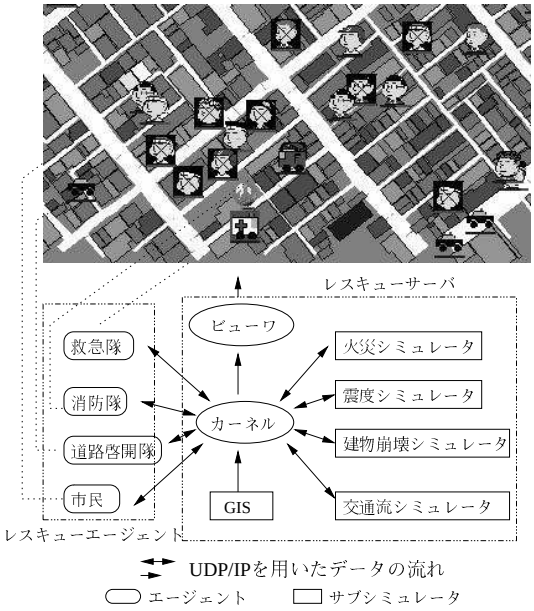


図 1: レスキューシミュレーションのアーキテクチャ

のデータの他に，災害規模や発生時間などの災害状況に加え救助活動や避難行動を含めたシナリオにそった災害救助シミュレーションは自治体の防災対策の上で有効な手段を提供する．

2.2 MASによる災害救助シミュレーション

2.2.1 ロボカップレスキューシミュレーション

ロボカップレスキューシミュレーションシステム (RCRS) は，阪神淡路大震災における事例をもとに設計されたエージェントベースの災害救助シミュレーションシステムである [7]．地震による建物崩壊，火災などの災害と現場で救助・救援活動をする消防隊，市民の活動をエージェントで表現し，災害と救助活動を合せてシミュレーションする (図．1)．

2.3 名古屋市の火災被害想定値との比較

名古屋市はマグニチュード 8 クラスの地震に対して被害想定をしている．昼間に濃尾地震クラスにより発生した火災の想定状況と，シミュレーション結果を表.1，.2 に示す．表.1 は名古屋市が複数地点で火災が発生すると想定した区に対する結果（出火点数は区の面積比で設定），表.2 は地震災害時に特に警戒が必要とされる名古屋駅近くの密集市街地を抱える西区と中村区を対象に想定出火数でシミュレーションした結果を示す．延焼率_nF と延焼率_F は各々，名古屋市の被害想定における消防運用（初期消火活動）なしとありに対応する．シミュレーションでは，消防エージェントの接続の有無が救助活動ありなしに対応している．表から以下の事がいえる．

表 2: 想定出火数を用いたシミュレーション結果

区	出火数		延焼率_nF 名古屋	延焼率_nF RCRS	延焼率_F 名古屋	延焼率_F RCRS
西	夜	30	2.8%	8.5%	2.4%	8.1%
	昼	58	5.8%	13.4%	4.9%	13.0%
中村	夜	22	2.4%	8.9%	1.9%	8.5%
	昼	45	5.1%	15.6%	4.5%	15.2%
相関係数				0.89		0.92

- 消防エージェントによる救助活動の結果，延焼率が減少している．
- 出火条件が異なる表.1 において，消防運用なし（延焼率_nF）と救助活動あり（延焼率_F）ともに名古屋市被害想定値とシミュレーション結果の間には，-0.06, 0.18 と相関関係がない．出火数を合わせた表.2 では，0.89, 0.92 と相関関係がある．
- 名古屋市被害想定における延焼率_nF と延焼率_F の間に相関は高い．[1] シミュレーション結果で両者の間で相関関係があるものの，想定値程高くない．

両者の延焼率の値が異なるのは，火災シミュレータや住宅データの違いによる．一方で，初期条件を合わせると両者の間に相関関係があるので同じ傾向を示している．これらは，寺野が示した事項 1, 3, 4 を満たしているが，2,3 の問題にアプローチするには不十分である．

3 確率モデルによる解析

3.1 マクロ的な解析の必要性

延焼率などのマクロ的な評価だけでなく，自治体の防災担当者は，いくつかのシナリオに基づくシミュレー

[1] 名古屋市被害想定では，消防運用の効果は初期消火だけに影響し，その後は関与しないので，当然，相関値は高い．

表 1: 名古屋市の濃尾地震クラスに対する被害想定とシミュレーション結果

区	対象区の道路網		名古屋市被害想定（昼間）				RCRS シミュレーション結果			
	頂点	辺	建物数	出火数	延焼率_nF	延焼率_F	建物数	出火数	延焼率_nF	延焼率_F
北	6,069	3,870	39,302	22	3.9%	3.4%	9,541	7	1.51%	0.99%
西	6,430	4,122	44,773	58	5.8%	4.9%	10,468	7	1.97%	1.74%
中村	6,044	3,766	41,769	45	5.1%	4.5%	8,994	6	2.16%	1.15%
中	2,026	2,093	18,726	5	0.9%	0.5%	5,396	4	2.03%	1.12%
瑞穂	4,053	2,563	30,092	2	0.5%	0.1%	6,656	4	0.65%	0.47%
熱田	2,609	1,760	17,580	3	1.3%	1.0%	4,309	3	4.32%	1.42%
中川	9,449	6,154	58,612	31	2.6%	1.7%	17,327	13	0.93%	0.87%
複数出火 7 区の被害想定値とシミュレーション結果の相関係数							0.96		-0.06	0.18
延焼率_nF と延焼率_F の相関係数						099				0.59

ション結果から，救助隊や市民などのエージェントの動きを大局的に把握，防災計画を立案する必要がある．ミクロレベルのエージェントシミュレーション結果からマクロレベルの動きの解析方法を提案する．

3.2 エージェント 活動と確率モデル

エージェントの動作モデルを以下に示す [3].

$$action : S^* \to A, \quad env : S \times A \to \mathcal{P}(S),$$

$S = \{s_1, s_2, \dots\}$ は環境の状態，$A = \{a_1, a_2, \dots\}$ はエージェントの行動，$\mathcal{P}(S)$ は S の冪集合を示す．状態 s_i にあるエージェントは，環境情報，他のエージェントとの会話と自分の経験（データ）から自らの判断で a_i を実行する．

エージェント i の動きは，履歴 h_i として表現される．

$$h_i : s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \cdots$$

n 個のエージェントからなるマルチエージェントシステム（MAS）の動きはその集まり $H = \{h_1, \dots, h_n\}$ として表現できる．

災害発生時に，救助エージェントは素早く災害現場をみつけ，その場所に向かう．図. 2 は，2 個所 n_2，n_3 で火災が発生し，救助エージェントが交差点 n_0 から現場に向っている状況を示す．交差点 n_1 でどちらの火災を消火するか，即ち，直進するか，右折するか，あるいは指令を待つかなどの行動を選択する必要がある．その選択結果は，交差点 n_1 から次のステップでどちらの交差点 n_j に移動したかで観測できる．その確率を p_{1j} とすると，救助エージェントの行動パターンを $P_1 = \{p_{10}, p_{11}, p_{12}, p_{13}, p_{14}\}$ で表現できる．

以上，まとめると以下の事がいえる．

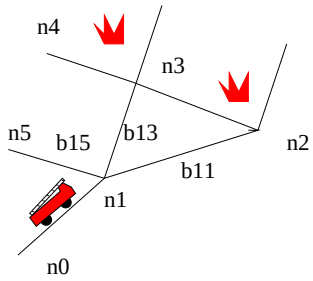


図 2: エージェントの動作選択と確率モデル

- エージェントは環境を観測して次の行動を決める．エージェントの行動によって引き起こされる環境の変化が MAS の出力となる．
- エージェントの振舞いは，状況 i である行動をとった結果，行動 j に移る確率 $\{p_{ij}\}$ の集合で表現できる．同じ状況においては，同じ目的をもつエージェントは似通って行動をとると考えられる．
- 与えられた状況に対し，望ましい結果を導いたエージェントとそうでないエージェントの行動の違いは，行動選択（$\{p_{ij}\}$）の違いで表現できる．

3.3 $\mathbb{F}$（$\mathbb{P}$）による履歴表現と解釈

状態 i から状態 j への移動した回数を f_{ij} を要素とする $\mathbb{F}$（以降，頻度行列と呼ぶ）はシミュレーション結果から求める事ができる．与えられた状況に対し，同じ目的をもったエージェントの履歴のアンサンブル平均をとり，$\mathbb{P}$（正規化の条件：$\sum_j p_{ij} = 1$）とする．

$\mathbb{P}$ の例として，消火場所に移動し，放水，タンクが空になると給水場所に移動し，満タンになると消火活

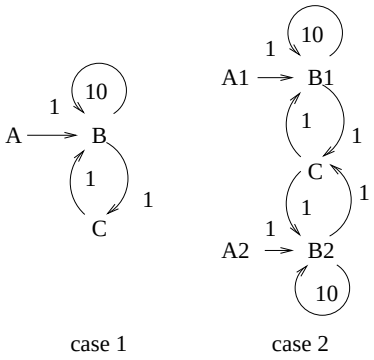


図 3: 救助エージェントの行動パターン例（辺の横にある数字はステップ数）

動を再開する消防エージェントの 2 つの行動パターンを考える（図. 3)．ここで，移動には 1 タイムステップ，満水から消火を続けタンクが空になるまで 10 タイムステップかかるとし，A は初期位置，B は火災発生位置で消火場所，C は給水場所とする．

case1 ：エージェント数は 1 個, B と C の場所も 1 個所とし，放水と給水を n 回繰り返す．

	A	B	C
A		$1/m$	
B		$10n/m$	n/m
C		n/m	

$$\xrightarrow{n\to\infty} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 10/12 & 1/12 \\ 0 & 1/12 & 0 \end{pmatrix}$$

ここで，全体のステップ数 $m = 1+n(10+1+1) = 12n+1$ で，行列 $\mathbb{F}$ のランクは 2 である．

case2 ：2 個所で火災が発生し，2 つのエージェントが分担して消火する．

	A_1	B_1	A_2	B_2	C
A_1		$1/m$			
B_1		$10n/m$			n/m
A_2				$1/m$	
B_2				$10n/m$	n/m
C		n/m		n/m	

$$\xrightarrow{n\to\infty} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 10/24 & 0 & 0 & 1/24 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10/24 & 1/24 \\ 0 & 1/24 & 0 & 1/24 & 0 \end{pmatrix}$$

$m = 24n+2$ で，行列のランクは 3 になる．

固有値，固有ベクトルを計算すると，

case1 ：$\mathbb{F}$ の固有値は 0.842 と-0.008 で，対応する固有ベクトルは $(0, -0.995, -0.09)^t$ と $(0, 0.09, -0.995)^t$ である．最大固有値 0.842 をもつ固有ベクトルで最大成分を持つ第 2 成分は，火災現場 B に対応する．

case2 ： 0.425, 0.417, -0.008 と 2 つの固有値が同じ大きさを持つ．$(0, 0.7, 0, 0.7, 0.14)^t$, $(0, 0.7, 0, -0.7, 0)^t$ が 2 つの固有値に対応する固有ベクトルで，case1 と同様に，大きな値の成分は各々火災現場 B_1,B_2 に対応する．

以上から次の性質がいえる．

性質 1 エージェントの行動履歴を示す $\mathbb{F}$ のランクは，そのエージェントの有効な動きを示す．同じ $\mathbb{F}$ のサイズでランクが大きいほど，エージェントはその範囲で有効に動いている．

性質 2 固有値の大きい固有ベクトルの主要な成分はエージェントがよく訪れた場所（状態）に対応する．

性質 3 複数のエージェントの動きが独立であれば，固有値の大きさはその独立の成分に応じて分散し，固有ベクトルの成分も各々のエージェントの動き対応する．

4 救助シミュレーション結果の解析

4.1 シミュレーション結果の時間的変化

図. 4 は，RCRS で提供されている阪神淡路大震災時の長田区[2] で，市民エージェントとして 85 個，負傷者を病院に避難所に搬送する救急エージェント，道路から瓦礫を取り除くエージェントと火災を消火する消防エージェントそれぞれ 7，11，13 個が災害・救助活動をした状況をしめす．目視から以下の事がいえる．

[2]道路網の頂点数，辺，建物の数は，各々765，820，740 で建物の 2 個所が給水できる避難所 (refuge_0，refuge_1) に指定されている．

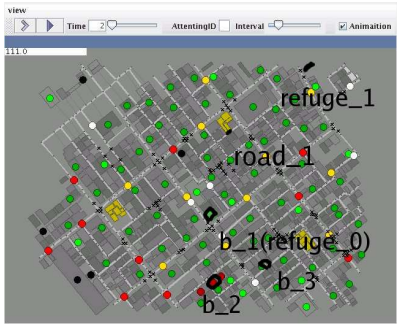
(a) 災害発生直後 (B1,B2,B3 出火地点)

(b) 50 ステップ

(c) 100 ステップ

(d) 150 ステップ

(e) 300 ステップ



(f) 固有ベクトルの主要成分に対応する位置

図 4: 災害の拡がりと救助エージェンの活動の時間変化

(a) 3 個所 ($B1$，$B2$，$B3$) で火災が発生する．

(b) 50 ステップの段階では，$B1$，$B2$ の火災に対して消火する．$B3$ の火災はひろがる．(黒い部分は火災で延焼した部分を示す．)

(c) 100 ステップの段階では，$B3$ の火災に対して消火活動が始まる．

(d,e) 150 ステップ以降，消火活動により延焼を防止している．

4.2 固有値，固有ベクトルからの解釈

50 ステップ刻みの消防エージェントに活動に対する固有値，固有ベクトル[3] を表. 3 に示す．表から，

- 行列の大きさ（ランク）が示す救助エージェントの活動は時間とともに拡がっている．
- 固有値の絶対値の比をみると，時間とともに，他に比べ 1 番値の大きいものが主になる．これは，最初は離れた位置にいた救助エージェントが，時間とともに同じ活動をしている事を示している．
- 頻繁に訪れた場所に対する固有ベクトルの主成分から以下の事がいえる．
  - すべての時間を通じて，給水場所に指定された建物 (主要位置を示す図. 4 (f) で refuge_0 と指定された b_1) が一番訪問されている．
  - 100 から 250 ステップで，B1 に近い道路 (road_1) が 2 番目の場所である．
  - 250 ステップに B3 に近い b_3 が 3 番目に現れ，300 ステップには 2 番目になっている．

がいえ，これらは 4.1 節の延焼率の時間的な変化，目視での観察にはないシミュレーションの経過を説明している．

[3] ここで，$\mathbb{F}$ の大きさは，頂点数，辺と建物の数をあわせた 2325 × 2325 で，消防エージェント数 (13) とシミュレーション時間 (300) からするとスパースな行列である．固有値，固有ベクトルの計算にあたっては，訪れていない場所を除外した行列で計算した．

5 考察とまとめ

エージェントを用いたアプローチは，人間の活動を含む社会シミュレーションに新しい研究手段を提供する．実際問題に適用する時に，シミュレーションモデ

表 3: 延焼率，頻度行列，固有値・ベクトルの時間的変化

時間	延焼率	行列		固有ベクトルの主要成分に対応する位置		
		サイズ/ランク	固有値 (1/2/3)	最大固有値	2 番目の固有値	3 番目の固有値
50	2.6%	155/135	96/36/36	b_1	b_2	road_1
100	4.0%	246/217	343/38/36	b_1	road_1	b_2
150	4.9%	300/271	615/39/36	b_1	road_1	b_2
200	5.0%	355/325	844/39/36	b_1	road_1	b_2
250	5.0%	422/388	1018/39/37	b_1	road_1	b_3
300	5.1%	484/442	1192/40/39	b_1	b_3	road_1

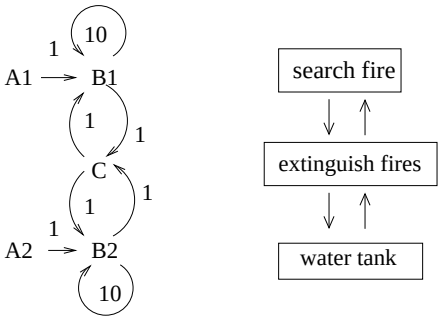


図 5: エージェントの行動と状態の遷移

ルの健全性と結果の妥当性を説明できる事が課題として指摘されている．

本論文では，災害救助シミュレーションとして RCRS を用いその結果が他の方法による自治体の災害想定値にあった結果である事を示した．次に，確率モデルを元にシミュレーション結果を解釈する方法を提案した．図．5 に示す様に，エージェントの位置移動がその状態遷移に対応すると仮定し，シミュレーション結果からエージェントの履歴を確率モデルを基にする行列で表現し，その固有値，固有ベクトルからマクロレベルで妥当な解釈ができる事を示した．

今後，検討すべき項目として以下があげられる．

- 複数のシナリオのシミュレーション結果の比較などを通じて，モデルの健全性と結果の妥当性を高め, 防災計画の評価手段の提供する．
- 災害救助以外に社会現象に対し，確率モデルによる解釈の適用を検討する．

謝辞本研究の一部は，文部科学省科学研究費補助金（課題番号：17500099) の助成により行われた．

最期に，日頃，ご意見を頂くロボカップレスキュー関連の皆様，貴重なご助言を頂いた齊藤 公明氏に感謝します．

参考文献

[1] 名古屋市地震被害想定調査, 3 月 1999.

[2] Peter Stone and Manuela Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, Vol. 8(3), pp. 369–381, 2000.

[3] Gerhard Weiss. *Multiagent Systems*. The MIT Press, 2000.

[4] 高橋, 高橋, 伊藤. 防災計画へのマルチエージェントシステム適用の一考察. In *JAWS06*, 2006.

[5] 高橋, 谷川, 高橋. 公開地図データにおける建物の自動生成方法. 情報処理学会 第 67 回全国大会 1V-8, 2004.

[6] 被害保険料率算定会. 地震保険調査報告 28 地震被害想定資料集, 9 月 1998.

[7] 田所諭, 北野宏明, 高橋友一, 松野文俊, 竹内郁雄. Robocup-rescue：情報科学の緊急災害対応問題への挑戦. 情報処理, No. 4, pp. 412–418, 2000.

[8] 寺野隆雄. エージェント・ベース・モデリング:その楽しさと難しさ. 計測と制御, Vol. 43, No. 12, pp. 927–931, 2004.

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-A701-2 (5/4)

他者の状態価値の推定に基づく協調・競合行動の獲得

Cooperative/Competitive Behavior Acquisition Based on State Value Estimation of Others

野間 健太郎 [1]，**高橋泰岳** [1]，**浅田 稔** [1,2]

Kentaro NOMA[1], **Yasutake TAKAHASHI**[1], **Minoru ASADA**[1,2]

[1] 阪大学大学院工学研究科，[2]JST ERATO 浅田共創知能システムプロジェクト

[1]**Graduate School of Engineering, Osaka University,** [2]**JST ERATO Asada Synergistic Intelligence Project**

{kentaro.noma,yasutake,asada}@ams.eng.osaka-u.ac.jp

Abstract

The existing reinforcement learning approaches have been suffering from the curse of dimension problem when they are applied to multiagent dynamic environments. One of the typical examples is a case of RoboCup competitions since other agents and their behaviors easily cause state and action space explosion. The keys for learning to acquire cooperative/competitive behaviors in such an environment are as follows:

- a two-layer hierarchical system with multi learning modules is adopted to reduce the size of the sensor and action spaces. The state space of the top layer consists of the state values indicating how close to the goals of the individual modules at the lower level, and the macro actions are used to reduce the size of the physical action space, and further,
- to what extent the other agent task has been achieved is estimated by observation and used as a state value in the top layer state space to accelerate the cooperative/competitive behavior learning.

This paper presents a method of modular learning in a multiagent environment, by which the learning agent can acquire cooperative behaviors with its team mates and competitive ones against its opponents. The method is applied to 4 on 5 passing task, and the learning agent successfully obtained the desired behaviors.

1 はじめに

エージェントが複数存在するマルチエージェント環境に強化学習を適用し，協調・競合行動の獲得を行う研究が多くなされている[1, 2, 3, 4, 5]. マルチエージェント環境で強化学習を適用する際の問題点として，自身や対象物の記述だけでなく，複数の他者との関係の記述も必要なため，考慮しなければならい情報が多くなり，探索空間が莫大になるため現実時間で学習するのが困難である．

Shivaram et al.[3]は，ハーフコートのサッカーフィールドで 4 対 5 でパスを行いシュートを決めるタスクで，味方の学習情報を共有することで，学習効率が上がることを示した．しかし，センサレベルの状態変数を使って状況判断をしているため，先に述べたように探索空間が大きく，学習時間が非常に長い．Stefan et al.[1]はマクロ行動を導入することにより，2 台のロボットが協調行動の獲得を実時間で実現している．マクロ行動とは設計者によってあらかじめ決められた行動のことで，モータレベルの行動を学習する必要がないので，効率的に状態空間を探索することができる．彼らは，2 台のロボットがいる環境で行動のみ抽象化することで，実時間で協調行動を獲得できることを示した．しかし，複数のロボットがいるような環境では，センサレベルの情報を用い，行動のみ抽象化するだけでは，現実時間では学習が困難である．他者と協調・競合行動を行う際，他者の行動予測が重要となってくる．これは他者の将来の行動が適切に予測可能であれば，他者の行動を考慮に入れた上で，自身のタスク達成に最適な行動決定を行なうことが可能であるためである．Takahashi et al.[6]は，強化学習の枠組を用い，ゴール状態までの距離を表す状態価値を用いて，他者の行動を推定する手法を提案している．この手法では，相手の行動の違いや視点の差による状態認識の違いが存在しても，ある意図に対応する行為の推定した状態価値の増加・減少によって意図を正しく推定できることが確かめられている．センサ

レベルの情報を用いて現在の状況を判断するのではなく，他者行為の状態価値の推定情報を組み込むことで，探索空間が小さくなり，結果的に学習時間が短縮できると考えられる．

本研究では，センサレベルの情報を抽象化した ” 自己行為の状態価値と他者行為の推定した状態価値 ” に基づく協調・競合行動を速やかに獲得する手法を提案する．RoboCup 中型機リーグに出場しているサッカーロボットを想定したシミューレータを用い，5 対 4 でパス，ドリブル，シュートを行うタスクで実験を行ない，本手法の有効性を示す．

2 強化学習

強化学習は，エージェントが環境との相互作用を通して累積報酬を最大化にする枠組である．エージェントは試行錯誤を行いゴールを見つける．状態価値 $V(s)$ とは，ゴール状態までの累積報酬の期待値である (式 1)．ここで簡単な例を示す．Fig.1 では，エージェントが状態 s_0 から s_5 まで移動し，報酬 0 を受け，s_5 で止まる行動をとり，正の報酬 $+1$ を受け取る様子を示している．この経験により，各状態の状態価値 V は s_5 から s_0 へと減衰率 γ の割合で伝播していく．状態価値関数は Fig.2 のようにゴール状態に向かって山のような形の関数となるので，状態価値の高さはゴール状態への近さを反映する．エージェントは状態価値が高くなるような行動をとる．マルチエージェント環境では，センサレベルの情報を使っての学習は探索空間が大きくなり現実時間では困難である．そこで，センサ情報ではなく，センサ情報を抽象化した状態価値を用いることで，探索空間が抑えることを考える．

$$V(s) = E\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\right\} \qquad (1)$$

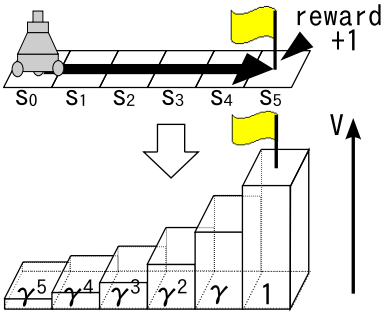


Figure 1: Sketch of state value propagation

3 協調競合行動学習

学習者はいくつかの基本行動モジュールを持っており，これらはあらかじめ強化学習の枠組で状態価値推定と共に獲得されているものとする．また，他者の視点の観測情報

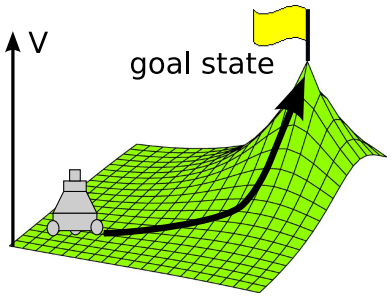


Figure 2: Sketch of a state value function

を推定し，これを基にある行動の状態価値を推定するモジュールも存在する．この状態価値推定は自己の行為と他者の行為両方に関して同じものを利用する．これらの自己と他者の状態価値推定を基に新たに状態価値空間を張り，協調競合行動学習を行う．

システムは，下位層に行動モジュール (action module) と他者行為推定モジュール (V estimation module) があり，上位層に学習器 (Gate) があるマルチモジュール型学習機構である (Fig.3)．行動モジュールは観測情報から各行動に対する状態価値を計算する．一方，他者行為推定モジュールは自分の観測情報を基に，自己中心座標系から他者中心座標系への変換を行い，他者の観測情報を推定する．すでに獲得している自分の行動モジュールに他者推定観測情報を当てはめることで，他者の行為の状態価値を推定する．上位層の学習器は，他者行為推定モジュールと行動モジュールから送られてくる状態価値を状態変数として，どの行動モジュールを選択するかを動的計画法の枠組で学習し，選ばれた行動モジュールに従った行動をとる．センサ情報を使って他者の状況を認識するのではなく，他者の行為を推定することによって，探索空間を抑え学習効率を向上させることができる．

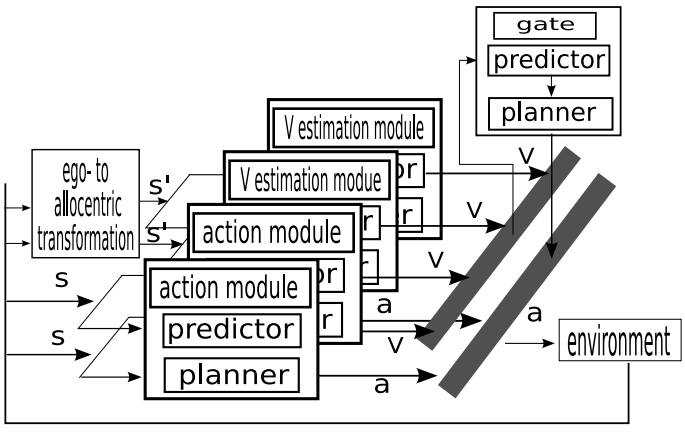


Figure 3: A multi-module learning system

4　タスクと仮定

環境は RoboCup 中型機リーグのフィールドにオフェンスチームが 5 台，デイフェンスチームが 4 台で構成されている．オフェンスチームはパスを回し，シュートをする．ディフェンスチームはオフェンスをマークしながら，ボールが近くにくるとボールをとりにいく．オフェンス (ディフェンス) チームのマーカーの色はマジェンダ (シアン) で，自陣ゴールの色は青色 (黄色) である．ボールに一番近いロボットがパサーとなる．

仮定として，オフェンスチームのパサーのみが学習し，他のレシーバとデイフェンスは固定政策で動くものとする．パサーは基本行動として 4 台のレシーバにパスをするか，ドリブルシュートを用い，状況に応じた適切な行動を学習する．パサーがレシーバに向かって，パスを出した後，パスを受けたレシーバがパサーに，パスを出したパサーがレシーバに切り替わるものとする．１試行が終わるたびに，各ロボットがコミュニケーションを行うことにより，学習情報を共有できるものとする．また，行動モジュールと推定モジュールはあらかじめ，学習しているものとする．

4.1　オフェンスチーム

パサーは他の４台のレシーバのうち，どのレシーバにパスをするか，またドリブル・シュートを選択する．パサーはパスを出した後，ある一定時間だけゴールに向かって移動するパスアンドゴーを行うものとする．レシーバはボールの方を向き，ボールやパサーや他のレシーバに一定距離以上近づかず，長方形を作るように動く (Fig.4)．なお，試行開始時の位置は，自陣でランダムに配置されている．

4.2　デイフェンスチーム

ボールの一番近くにいるオフェンス（パサー）に一番近いデイフェンスがマークをし，残りのディフェンスは一番近いオフェンスをマークする．マークとは，オフェンスの近くでオフェンスと自陣ゴールの間に入ることである (Fig.4)．そして，ボールが近くにくるとボールをとりにくる．また，オフェンスチームの不利にならないように，ペナルティエリアに一定時間入れないものとする．なお，試行開始時の位置は，自陣でセンターサークルに入らないようにランダムに配置されている．

4.3　ロボットと実験環境

RoboCup 中型機リーグに出場しているロボット (Fig.5) を想定したシミュレーションにより実験を行った (Fig.6)．ロボットは，センサに全方位カメラ，前方カメラ，移動機構に全方位移動機構，キック機構を持ったロボットである．Fig.6 の右上は，前方カメラ，右下は全方位カメラの画像を表している．ロボットは，色情報を使って，ボールや他のロボットを認識している．仮定として，ロボットは，全方位カメラから得られるを情報を使って，三次元再構成をし，他者のセンサ情報を推定する．

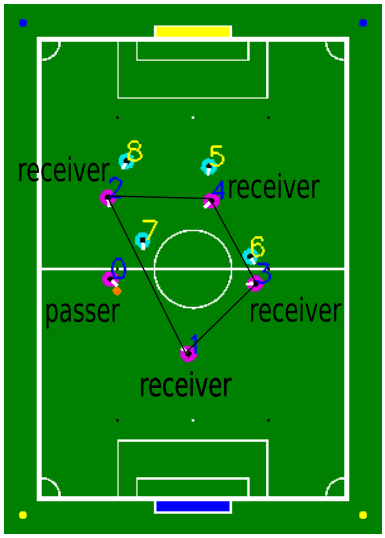


Figure 4: A passer and the defence formation

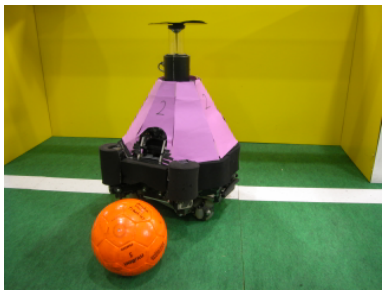


Figure 5: A real robot

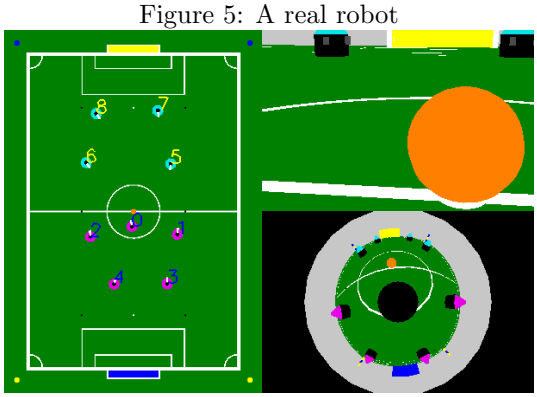


Figure 6: Viewer of simulator

4.4　上位学習器の状態/行動空間

パサー (学習者) の下位層の行動モジュールは，各レシーバに対するパスモジュールを４つ，ドリブル・シュートモジュールを１つ，合計５つある．下位層のレシーバ推定モジュールは，各レシーバーがパスを受けた後のシュートのしやすさの達成度を推定するもので，合計４つある．これらの行動モジュールとレシーバ推定モジュールはあ

らかじめ，上位層の学習の前に獲得しているものとする．下位層の行動空間は，設計者によって設計されたマクロ行動を適用する．マクロ行動は，モータレベルの探索をしないため，探索空間を抑えることができる．パサーの上位層の学習器の状態空間 S は，下位層から送られてくる状態価値から成り立っている．

- 各レシーバに対するパスモジュールの状態価値それぞれ４つ
- ドリブルシュートモジュールの状態価値１つ
- レシーバ推定モジュールの状態価値それぞれ４つ

状態数は，２値化されていて，2^4 x 2 x 2^4=512 である．報酬は次のように与えられている．

- 10 ボールがゴールに入る．(1 試行終了)
- -1 ボールをインターセプトされる．(1 試行終了)
- 0.1 パスが成功する．
- 0.3 ドリブルが成功する．

ボールがフィールドの外に出たり，ある一定時間たつと引きで 1 試行が終了する．以下では，下位層の行動モジュールの詳細を示す．

4.5　パスモジュールの状態空間

パスモジュールの状態空間 S は，全方位カメラ上で，

- レシーバより手前にいるディフェンスの中で，レシーバとのなす角が最も小さいディフェンスとの角度 (θ_1)
- 一番近いディフェンスとレシーバの角度 (θ_2)

である (Fig.7)．二つの角度は，ロボットが見えない状態を含めて，それぞれ 10 個に量子化されている．よって，状態数は 100 である．パスモジュールの状態価値のイメージ図を Fig.8 に示す．Fig.8 はある状況でパサーがパスのしやすさを示したものである．パサーは 3 号機である．各レシーバ (0,1,2,4 号機) の横にあるゲージは各レシーバに対するパスモジュールの状態価値を表していて，ゲージが高いほど状態価値が高い．1，2 号機はディフェンスにパスコースを防がれていないので，状態価値が高い．一方，0，4 号機はディフェンスにパスコースを防がれているので，状態価値が低い．状態価値のマップを Fig. 9 に示す．レシーバとディフェンスの角度が小さい程，状態価値が低い．黒色の領域は未経験の状態である．上位層に送られる状態価値は黄色と赤黒で 2 値化されている．

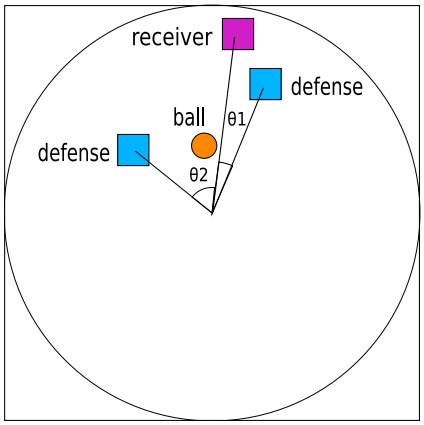


Figure 7: state variable of the pass module

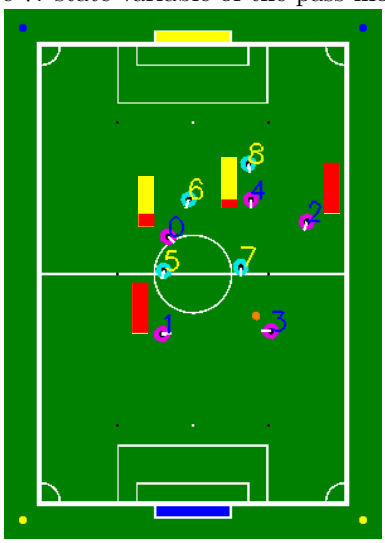


Figure 8: examples of state valuesof the pass module

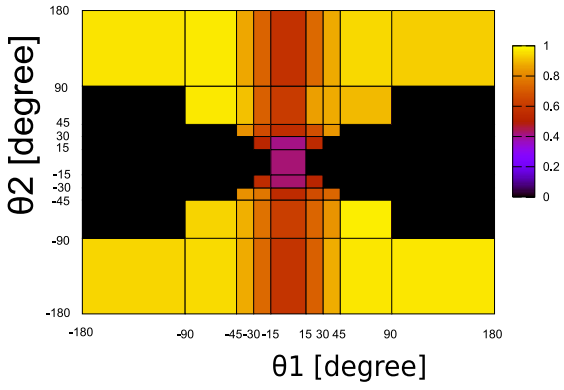


Figure 9: state value map of the pass module

4.6 ドリブル・シュートモジュール

ドリブル・シュートモジュールの状態空間 S は，全方位カメラ上で，

- 一番近いディフェンスと相手ゴールの角度 (θ_1)
- 一番近いディフェンスとボールの角度 (θ_2)
- 一番近いディフェンスの距離 (r)
- 相手ゴールの両エッジの角度 (θ_3)（ゴールまでの距離を表す）

である (Fig.10)．それぞれ，8,8,5,7 に量子化されている．よって，状態数は 8 x 8 x 5 x 7 = 2240 である．パサーのドリブル・シュートモジュールの状態価値のイメージ図を Fig.11 に示す．Fig.11 は，ドリブルシュートのしやすさを示している．左図は，パサー (1 号機) は，ディフェンスが近くにいなく，ゴールに近いので，状態価値が高い．一方，右図は，パサー (3 号機) は，ゴールから遠く，ディフェンスが近くにいるので，状態価値が低い．θ_2 と θ_3 を固定した時の θ_1 と r の状態価値のマップを Fig. 12 に示す．レシーバとディフェンスの角度が小さい程，ゴールから遠い程，状態価値が低い．上位層に送られる状態価値は黄色と赤黒で 2 値化されている．

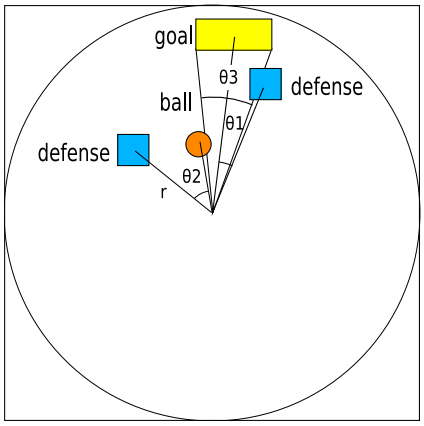


Figure 10: state variables of the dribble and shoot module

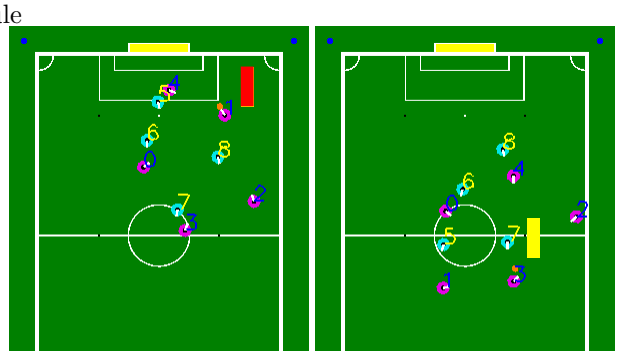


Figure 11: two examples of state values of the dribble and shoot module

4.7 レシーバの推定モジュール

パサーは全方位画像情報から 3 次元再構成をし，レシーバがどのような画像情報を取得しているか計算する．そして，すでに獲得している自分のレシーバモジュールに当てはめてることで，レシーバがパスを受けてからシュートしやすさの推定を行う．レシーバの推定モジュールの状態空間 S は，全方位カメラ上で，

- 一番近いディフェンスの距離 (r)
- 相手ゴールの両エッジの角度 (θ_1)（ゴールまでの距離を表す）

である (Fig.13)．それぞれ，5,7 に量子化されていて，状態数は 5 x 7 = 35 である．レシーバ推定モジュールの状態価値のイメージ図を Fig.14 に示す．Fig.14 に示す．各レシーバ (0,1,3,4 号機) の横にあるゲージは状態価値を表し，状態価値が高いとゲージが高い．0 号機は相手ゴールの近くでディフェンスが近くにいないので状態価値が高い．一方，1 号機は相手ゴールから遠く，ディフェンスが近くにいるので状態価値が低い．レシーバ推定モジュールの状態価値のマップを Fig. 15 に示す．ディフェンスが遠く，ゴールに近いほど，状態価値が高い．黒色の領域は未経験の状態 (ゴールの中に入った) である．上位層に送られる状態価値は黄色と赤黒で 2 値化されている．

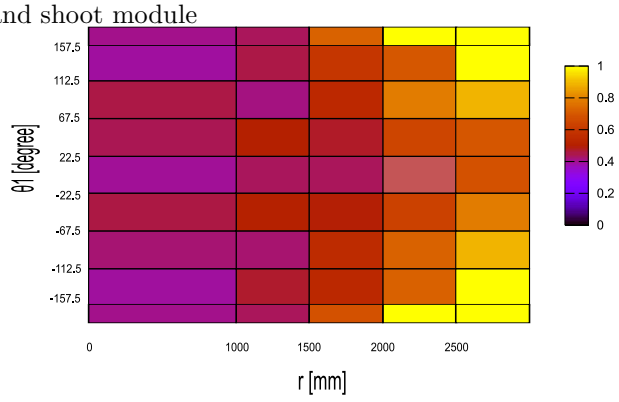


Figure 12: state value map of the dribble and shoot module

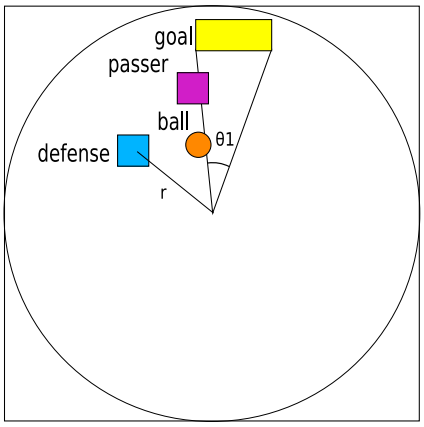


Figure 13: state variables of the receiver module

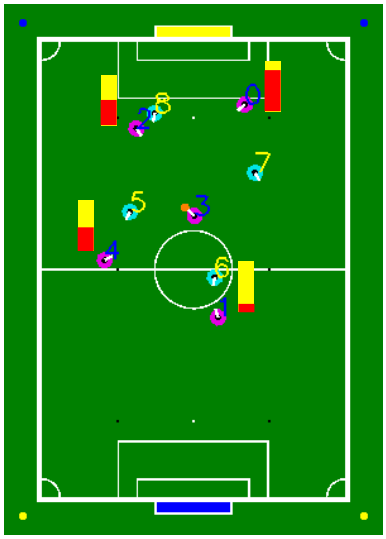


Figure 14: examples of state values of the receiver module

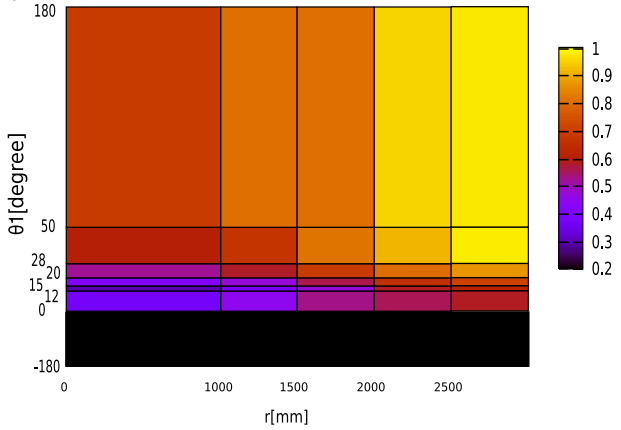


Figure 15: state value map of the receiver module

5 実験結果

成功率を Fig. 5 に示す．下位のモジュール選択を 80% greedy 20%ランダムで行った時のグラフである．900 試行後，成功率が 30%, 失敗率が 70%, 引き分け率が 10%に収束している．Shivaram et al.[3]は，30000 試行後，成功率が 30%程度であり，学習時間は 30 倍程度短くなった．Fig. 16 は，1 試行のパス回数を示している．350 試行以降パス回数が減っている．これは，無駄なパス回しをしていないということである．100%greedy の時の成功率，失敗率，引分率は，それぞれ，55%，35%，10%である．100%random の時は，それぞれ，2%, 97%, 1%である．100%greedy の時の成功率は，80%greedy の時の成功率よりよい．これは，レシーバとディフェンスは固定政策であり，新たな状況がそれほど起こらないからである．獲得された行動の様子の一例を Fig.18 に示す．

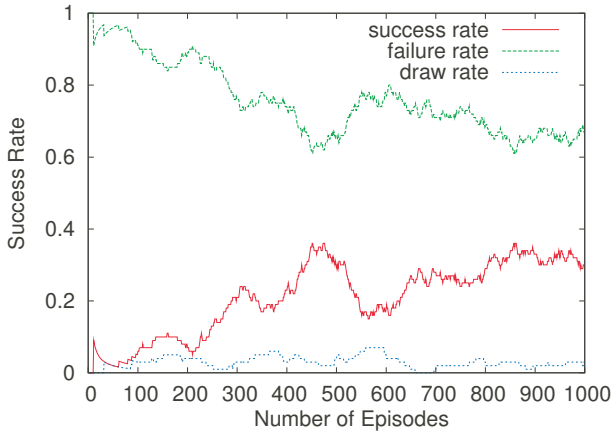


Figure 16: success rate

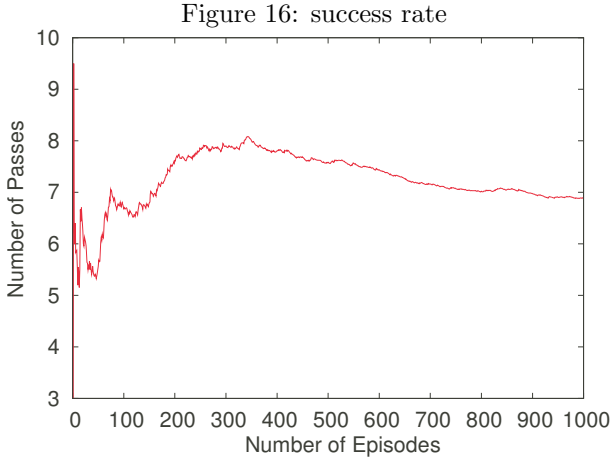


Figure 17: the number of passes

6 考察

学習を加速させるため，センサ情報の代わりに状態価値，モータコマンドの代わりにマクロ行動，そして，レシーバの行動を推定するモジュールを導入した．この結果，学習時間が 30 倍速くなった．比較手法[3]は，30000 試行後，

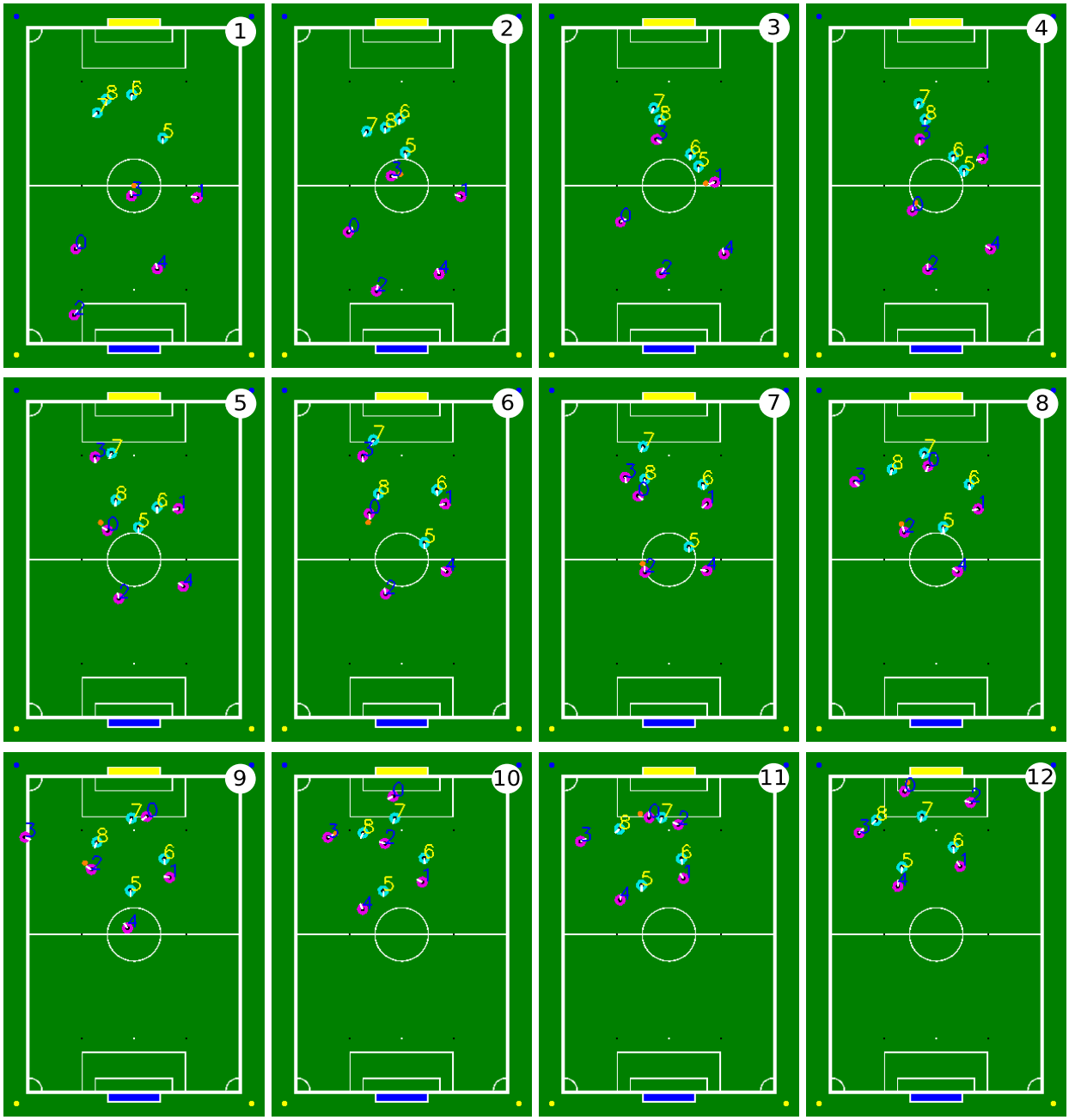


Figure 18: a sequence of a behavior in simulation

成功率がコミュニケーションありで 32%，コミュニケーションなしで 23%に収束している．

本手法では，1 試行中エージェント間でコミュニケーションを行なわないが，レシーバ推定モジュールが同じ役割をしていると考えられる．レシーバ推定モジュールを用いない場合の成功率を Fig. 19 に示す．成功率は 21%程度に収束している．これは比較手法[3]の成功率 23%と近い．状態と行動の抽象化 (状態価値とマクロ行動) は学習時間を抑えることができる．一方で，レシーバ推定モジュールの導入は，チームワークの向上につながる．実機での実験が今後の課題である．

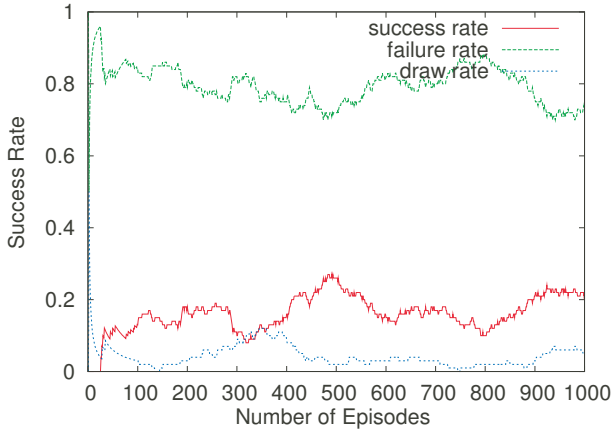


Figure 19: Success rate without the receiver's state inference modules

7 結言

従来，マルチエージェント環境に強化学習を適用する場合，センサレベルの情報を用いて探索すると，状態空間の爆発により現実時間で学習することが困難である問題に直面する．そこで，マルチモジュール学習機構を導入し，センサレベルの情報を抽象化した ” 自己行為の状態価値と他者行為の推定した状態価値 ” を用いて，探索空間を抑えこの問題を解決した．

RoboCup 中型機リーグに出場しているサッカーロボットを想定したシミューレータを用い，5 対 4 でパス，ドリブル，シュートを行うタスクで実験を行ない，本手法の有効性を示した．

参考文献

[1] Stefan Elfwing, Eiji Uchibe, Kenji Doya, and Henrik I. Chirstensen. Multi-agent reinforcement learning: Using macro actions to learn a mating task. , Vol. 13, pp. 3164–2220, 2004.

[2] Shoichi Ikenoue, Minoru Asada, and Koh Hosoda. Cooperative behavior acquisition by asynchronous policy renewal that enables simultaneous learning in multiagent environment. In , pp. 2728–2734, 2002.

[3] Shivaram Kalyanakrishnan, Yaxin Liu, and Peter Stone. Half field offense in robocup soccer: A multiagent reinforcement learning case study. , 2006.

[4] Peter Stone, Richard S.Sutton, and Gregory Kuhlmann. Scaling reinforcement learning toward robocup soccer. , Vol. 13, pp. 2201–2220, 2003.

[5] Yasutake Takahashi, Kazuhiro Edazawa, and Minoru Asada. Multi-module learning system for behavior acquisition in multi-agent environment. In , pp. CD–ROM 927–931, October 2002.

[6] Yasutake Takahashi, Teruyasu Kawamata, and Minoru Asada. Learning utility for behavior acquisition and intention inference of other agent. In , Vol. 1, pp. pp.25–31, 2006.

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-A701-3 (5/4)

Physical Visualization Sub-League: A New Platform for Research and Edutainment

Rodrigo da Silva Guerra [1], **Joschka Boedecker** [1], **Minoru Asada** [1,2]
[1] Graduate School of Engineering, Osaka University, Osaka, Japan
[2] JST Erato Asada Project
{rodrigo.guerra,joschka.boedecker,asada}@ams.eng.osaka-u.ac.jp

Abstract

This work introduces a novel minirobotics system which is to become a new sub-league of the RoboCup Soccer Simulation League, called Physical Visualization. We incorporate mature technology proven efficient over the years in other RoboCup leagues and we introduce new collaborative development concepts into the games of this new sub-league shifting essential research issues from the playing agents themselves to the development of a new versatile research and educational platform. We describe in detail the technical aspects supporting this multi-agent robotic framework, integrating cutting-edge and low cost watch technology in the form of a miniature multi-robot system which mixes reality and simulation. Competition formats and roadmaps are presented and discussed and the advantages for education and research applications are emphasized. Finally we discuss benefits of this new platform in terms of standardization, flexibility and reasonable price and try to characterize and discuss the future possibilities enabled by this project and the place of this new sub-league within the RoboCup community.

1 Introduction

Physical Visualization (PV for short) is candidate to be a new RoboCup Soccer Simulation sub-league. The project is intended for fostering education, research and development together with the RoboCup community. The PV is based on a miniature multi-robot system which mixes reality and simulation through an Augmented Reality (AR) environment. The project has a two-folded focus: research and education. The main goals of the PV are:

- to gradually improve the platform so that it becomes a powerful and versatile standard for multi-agent research and education.
- to explore educational possibilities and real world applications based either on the system as a whole or on some parts of it (e.g. the robots alone).

Since March of 2006 CITIZEN Co. and Osaka University committed themselves to the endeavor of developing together with RoboCup a new miniature, and yet affordable, robotics platform. This comes against the main stream in robotics, where, in general, the solutions are costly. We focused on versatility and affordability, taking advantage of well established industry technologies to allow the development of an inexpensive platform. In order to do that we used the know-how of the cutting-edge and low cost watch technology as a basis for building an affordable miniature multi-robot system mixing reality and simulation. This allows the employment of a large number of robots in a rather reduced space with a very low budget and amazing portability. Both the robots and the system are to be constantly upgraded and improved, being developed together with PV and CITIZEN exclusively for the competitions. Three dominant characteristics of the project are: (a) affordability, (b) standardization and (c) open architecture. These aspects are explained in detail in the next paragraphs.

Affordability: Generally speaking, doing research on robotics is an expensive task, specially when it comes to multi-agent robotics. Even in the most inexpensive real-robot experiments, it is a common sense that one would expect to spend at least several thousands of dollars in order to have a multi-agent setup. For the main reasons, one could surely account for the unavailability of adequate commercial platforms, thus bringing the need for custom robots. In RoboCup the strong competition forces teams to challenge themselves to come up with new design ideas which quite often are translated into more complex and expensive hardware. This last factor also implies that a wider spectrum of technical fields needs to be covered for the complete design of the machines, including issues which are not always related to the research focus originally in mind. Such difficulties may seem inherent to the research track of some institutions, but they most likely come as an obstacle to those

who do not have the man-hour and the money for the journey.

Standardization: Sharing a common standard platform allows the easy comparison of results and concepts into the same grounds. The two-dimensional environment of simulation league [8] is a successful example of such standardization: papers often show comparative results using the same common simulation environment, for example, playing against the code of a good team of former years (e.g. [7, 12, 11]). To a minor degree the four-legged league [15] also shares some of these characteristics as, for instance, the champion teams usually release their source code for the others to build on in the coming year, thus speeding the progress and avoiding the need for newcomer teams to "re-invent the wheel". In our understanding there was still a lack of a standard platform such ours, providing the flexibility of simulations but in real robots and at a reasonable price.

Open architecture: Our platform brings the above standardization in a completely open architecture with room for collaborative improvement. All program codes, including the computer software codes and the robots firmware are being released with the GNU GPL license [10]. Moreover, schematics are already being provided for all electronic circuits of the system.

2 Technical Aspects

The technical aspects of the main system are illustrated in the block diagram of figure 1 and on the simplified drawing of figure 3-a. Robots obey commands sent by a central server through an IR beam, while their actual position and orientation is fedback to the server by a camera located on the top. Meanwhile a number of visual features are projected onto the field by using a flat display. This system merges characteristics and concepts from two of the most mature RoboCup leagues, Simulation and Small-Size [5], and adds a new key-feature: augmented reality.

All the robots are centrally controlled from one CPU but their decision making algorithms run on networked clients, making the robots behave autonomously virtually isolated from each other just like in simulation league. Position feedback is based on colored markers placed on top of the robots which are detected through a vision system in the same way used in small-size league. Robot control is based on strings of commands sent by modulated infrared signals (in this sense resembling U-league to some extent [1]).

One characterizing feature of the system is the unmodelled embodiment dependent representation of the robots. Contrary to the misleading impression the term "physical visualization" might imply, the robots *are not* mere physical visualizations of some sort of internally simulated mechanism of any kind. On the contrary, the system blindly sends client commands to the robots which may (or may not) respond by performing arbitrary movements. In other words, changes on the physical body of the robot would not require changes on the server internal representation of the robot's mechanisms for there is no such a thing.

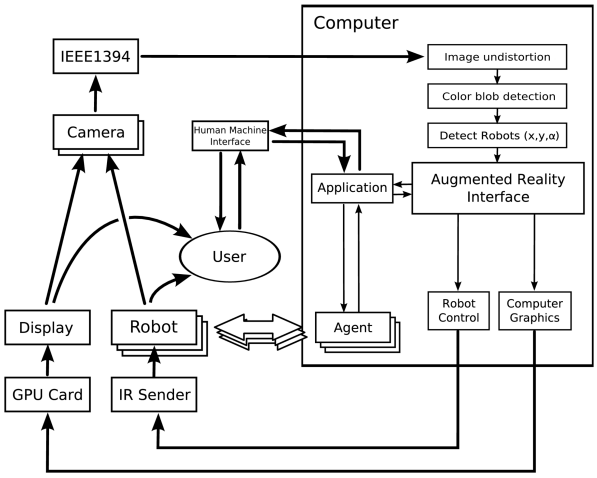


Figure 1: Block diagram illustrating the main system components

2.1 The position feedback

The position of the robots (and eventually other objects, such as ball) is detected from the processing of high-resolution camera images. The computer vision system currently implemented can be divided into three main subsystems: (a) undistortion, (b) blob detection, and (c) identification & orientation. Each one is described in the following paragraphs.

Undistortion: The vast majority of consumer cameras are known to have no significant lens distortion, therefore it is common practice to assume a linear pinhole model. Despite the fact of the PV robots being real three-dimensional objects occupying volume in space, the domain of possible locations for their bodies over the plane of the flat screen is known to be confined into a two-dimensional space. Because of that the calibration problem can be reduced, without loss of generality, to a plane-to-plane linear transformation from the plane of the captured image to the plane of field itself. This transformation is a single linear 3×3 matrix operator which defines a homography in the two-dimensional projective space (see figure 2). In the presence of significant lens distortion the simple addition of a prior step for radial lens undistortion, such as in Tsai's method [14], is likely to be sufficient. Refer [3] for a more extensive review on the projective geometry approach to computer vision.

Blob detection: After undistorted, the image is segmented into blobs of certain colors of interest. These colors are defined by a mask in the three-dimensional $Y \times U \times V$ space. Adjacent pixels, in a 8-neighborhood, belonging to the same color mask configure a single blob. The area (total amount of pixels) and center of mass (average (x, y) coordinates) of the blobs are extracted. Blobs whose mass values are not within a tolerance range from the expected are discarded. This procedure is used

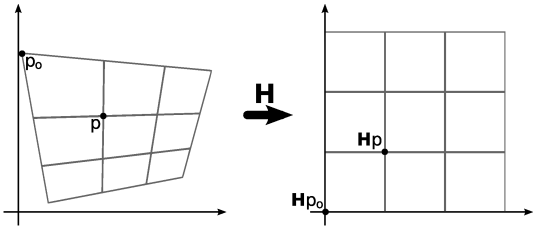


Figure 2: Plane-to-plane projective undistortion based on homography transformation, where **H** is a 3×3 matrix operator and p and p_o are 3-dimensional vectors representing points in the two-dimensional projective space

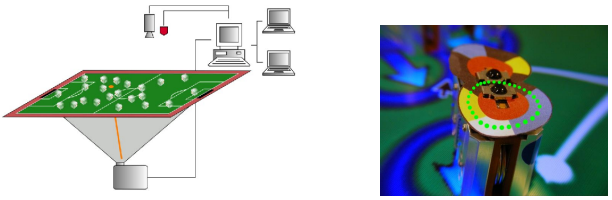
for finding the center of the colored marking patterns on the top of each robot – the red shape seen on figure 3-b.

Identification and orientation: The process here described is inspired on [9]. Once a potential blob is found, a radial pattern of colors is sampled within a pre-defined radius of its center. In figure 3-b these sampling locations are artificially illustrated by a closed path of little green dots. This pattern is cross correlated with a database of stored patterns, each of which uniquely defining a robot's identity. Let's denote $x(i)$ to be the color in the pattern x at the angle i. The cross-correlation r_{xy} is calculated accordingly to the equation 1 for each pattern y the database, and for each $\Delta\alpha$ in the interval $[0^o, 360^o)$. If, for a pattern x, the minimum value of $r_{xy}(\Delta\alpha)$, for any y and $\Delta\alpha \in [0^o, 360^o)$, exceeds a minimum threshold, then the corresponding y gives the identity of a robot, and $\Delta\alpha$ gives its orientation.

$$r_{xy}(\Delta\alpha) = \frac{\sum_{i=0^o}^{360^o} [(x(i) - \bar{x}) \cdot (y(i - \Delta\alpha) - \bar{y})]}{\sqrt{\sum_{i=0^o}^{360^o} (x(i) - \bar{x})^2} \cdot \sqrt{\sum_{i=0^o}^{360^o} (y(i - \Delta\alpha) - \bar{y})^2}} \quad (1)$$

2.2 Augmented reality

The idea about the augmented reality setup is an extension of a previously published similar concept where robot ants would leave visually coloured trails of "pheromones" by the use of a multimedia projector on the ceiling of a dark room in a swarm intelligence study [13]. Huge improvements in versatility, flexibility, and standardization can be introduced by applying that concept into a more customizable system. The figure 3-a shows an illustrative drawing and figure 3-b shows an actual picture of our system in action. Given the reduced size and weight of the PV sub-league robots the application of a conventional flat display as the field becomes feasible – depending on the application, displays as small as 20-inches are more than enough. This adds much versatility to the system without adding much costs and without complicating the required setup. The mixture of reality and simulation enables projections of environmental features surrounding the real robots. By doing so, not only the environment becomes more visually appealing, but also allows for an enormous variety of new applications which would be otherwise impractical extending the possibilities to the limit of one's imagination.



(a) (b)

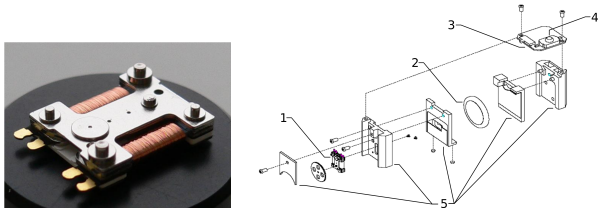
Figure 3: On the left an illustration of the overall system including the feedback control loop (infrared transmitter, camera, server) and the augmented reality screen. On the right an actual close-up picture of two robots playing using such setup. (See last paragraph of subsection 2.1 for explanation about the green dots)

2.3 The miniature robot

Until now, a few developments have been made on very small sized robots, being ALICE one of the most prominent names (see [2] for a survey). In terms of hardware the (current) robot here described is not much different from those many other mini-robots that have been developed so far. We emphasize that it is the unique features brought together by our proposed framework allied to the low cost, robustness and simplicity of the architecture that make this system so attractive.

The first versions of the miniature robot here used were originaly developed by CITIZEN as merchandize devices for demonstrating their new solar powered watch technologies [16]. Since March of 2006 three new prototype versions were already developed for matching the requirements of this project. The most current version of the robot has dimensions of $18 \times 18 \times 22mm$, no sensors, an infrared receiver and is driven by two differential wheels. This first robot was purposely designed to have rather simplistic hardware configuration as a starting point, a seed, to be followed by numerous upgrades in the long term. The main robot components are (numbers in accordance to figure 4-b):

1. Motor – Customized from wristwatch motor unit. See further details in the dedicated sub-section 2.4.
2. Battery – Miniature one-cell rechargeable $3.7V$ lithium ion polymer battery with capacity of $65mAh$.
3. Control board – Currently based on the Microchip $8bit$ PIC18 family of microcontrollers, each robot comes equipped with a PIC18LF1220 which features 4kb of re-programmable flash memory.
4. IR sensor – An IR sensor is used in order to listen for commands from the PC. The sensor operates at the $40kHz$ bandwidth modulation (same of most home-appliance remote controls).
5. Body – The resistant durable body of the robot is micro-machined in aluminum using CITIZEN's high precision CNC machines.



(a) (b)

Figure 4: On the left a close-up picture of the step motor, on the right an exploded view of revealing the robot parts.

Feature	Value
Dimensions (mm)	$7.0 \times 8.5 \times 1.9$
Configuration	2 coils $\times$ 1 rotor
Gear ratio	1 : 240
Torque ($gf \cdot cm$ at $2.8V$)	between 2.0 and 4.0
Current at $200rps$ (mA)	between 4 and 12
Nominal rotation (rpm)	12.000
Direction	standard and reverse

Table 1: Technical specifications of the step motors used in the miniature robots

2.4 The micro step motor

CITIZEN Co. is renowned in what regards to the manufacture of miniature devices. The motors are one of the main features without which it would be hardly possible to achieve such elevated degree of miniaturization. The miniature robots use two motors to drive its differential wheels. For couping with the torque requirements CITIZEN developed a new special class of step motors combining high-speed rotation and nano-scaled geared reduction. The result is highly reliable motor with very low power consumption fitting into the same thin packaging.

2.5 Robot's firmware and control protocol

The current control protocol was programmed in C and compiled using the proprietary MPLAB C18 compiler – sadly, a code port to the open source SDCC compiler has been delayed due to temporary instabilities in that compiler's support for the PIC18 architecture. Eventually the code might become supported by the GNU C compiler if the robot's microcontroller changes in the next comming years (see sub-section 3.1).

All robots share the same firmware but dynamic IDs are be assigned so that commands to an individual robot can be discriminated. Each of the two wheels can be controlled to run at two different speeds, in both directions or stopped (total of 5 possible values). These two speeds can be fine tunned by infrared commands. An extra set of fast ballistic movements is also provided, with duration customizable, again, by infrared commands. Additional firmware features include low-level battery check and special software reset and sleep commands. Because of the physical nature of the infrared light beam, commands have to be sent by the server to one robot at a time, in an ordered fashion. This implies that bigger number of robots result in longer control lags. Therefore the protocol format was designed so that the command could be sent in a very short time. The current command protocol has a length of $12bits$: ID ($5bits$), left command ($3bits$), right command ($3bits$), and bit parity check ($1bit$). Less frequently used instructions are multiplexed from a sequence of two or more commands.

3 Competitions with cooperation

RoboCup is not only a place for robot tournaments and competitions, but also an international effort towards a bigger common goal [6]. Therefore it is crucial for the survival of the new sub-league that its conceptual bases are not redundant with other leagues and in accordance with the RoboCup long term road map. In the strict technical sense, the autonomous playing agents developed for the PV system, just by themselves add no new challenge if compared to other existing RoboCup competitions and *are not* the central aspect around which teams should concentrate their effords. The original and dominant point of the proposed sub-league is its concept of collaboration towards the development of a central platform for the benefit of all. While in other leagues essential research issues are traditionally faced in the playing agents themselves (AI, biped walking, vision, etc.), in the PV the research issues are in the improvement of the system – in the development of the platform and its robots.

Therefore, the PV sub-league multi-agent interface should be simple to use and adequate for educational purposes. Agent code is expected to be developed by seasonal students while permanent members (professor, staff and graduates) pursue longer term projects that contribute for the sub-league itself and for its platform versatility in a variety of fields.

Keeping these ideas in mind we prepared, three competitions:

- Electronics & Firmware Competition;
- Educational Games Competition;
- Undergrad Team Development Competition.

These three competitions form a kind of self-sustainable "ecological" cycle like shown in the figure 5. Arcs 1 and 2 represent respectively the new needs that inspire hardware development and the new possibilities these developments tend to provide (in future years, if they are incorporated into the the standard system). Arcs 3 and 4 work in a similar way, but within the official hardware of the system to which all teams shall have access. Arc 3 represents a selection of interesting ideas from educational games presented in previous years that are put in practice in a big tournament among undergraduates from various teams, bringing again new ideas for even more interesting games in arc 4. Arc 5 and 6 represent the volunteer contribution of the technical committee for "cluing" all pieces together and making the system work. The idea is to maximize the flow in arcs 1, 2, 3, and 4 while minimizing the contributions through arcs 5 and 6, in the form of a self-sustained evolution cycle for the sub-league.

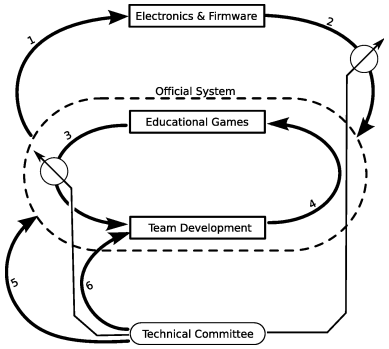


Figure 5: Typical self-sustained development cycle expected

These competitions are indroduced in the next subsections.

3.1 Electronics & Firmware Competition

Goal Allow the evolution of the robot's technology and improve all non-software related aspects of the system.

Summary Teams would have the opportunity to contribute with new ideas for the electronic aspects of the system as well as robot's firmware. Those with background in fields more closely related to the hardware would be able to include in their projects the improvement of certain aspects of the system either for didactic purposes (e.g. class on microcontrollers) or for research. Meanwhile, teams with background in fields more related to computer science would be able to acquire valuable experience by accompanying or even contributing to these projects.

Entries for this competition will consist of documents describing in detail the proposed modification, along with CAD drawings, source code, schematics, etc. Developments could be made on any of the current elements of the system, including the robot, or by introducing a new electronic element to the system. All must be made available to other teams so that they can use and improve at their own. Closed portions will be allowed as part of an entry, but but those parts will not be officially considered during judgement.

Developments on the control circuit of the robots will be required to meet several restrictions regarding the position of mounting holes and pads, the size and shape of the board, the maximum bounding volume, the weight limit of the robot and place of certain components in order to ensure compatibility with current micromechatronic architecture. Within those constraints completely new architectures could be proposed. Optionally, entries could even consist of firmware development only.

3.2 Educational AI Games Competition

Goal Create a pool of interesting didactic software applications in the form of games using the system for educational purposes.

Summary Entrants would come up with different game ideas using the system in which they teach concepts related to common subjects ranging from basic computer programming to very specialized topics related to multi-agent systems and artificial intelligence.

The entries would consist of the proposed games along with their source code, supporting tools or API (if any), documentation and accompanying teaching materials. In order to ensure that other teams could easily profit from these contributions the entries would need to be necessarily based on the current official system only. While the eventual introduction of accessories such as balls maze walls or colored objects would, in general, be permitted, no external specialized electronic devices would be allowed. Live demonstrations and poster presentations would be performed during the game event, and together with prior qualified reviewing would rank the entrant.

3.3 Rapid (Soccer) Team Development Competition

Goal Allow undergraduate students to develop complete teams of their own within the typically limited time window of their courses.

Summary The teams would be based on a simplified didactic game framework allowing easy development requiring only a very limited amount of knowledge. All contestants would have an equally limited amount of time for the development of their teams, thus giving similar advantages to teams with limited time to spare. Game rules and supporting software would be officially released just a predefined amount of months before the games.

This comes to fill the gap between RoboCup Junior and the other RoboCup Senior leagues. Typically RoboCup Junior focus mainly on primary and secondary school children, making its challenges less interesting for the more mature undergraduates of courses more related to specialized subjects. The undergraduation curricula are generally composed of a number of classes that last half a year or less and are taken simultaneously over the course of several years. This makes it very difficult for projects based on RoboCup Senior leagues to be included into the main curriculum. Refer to [1, 4] for a previous account to some of above points, where a new league directed exclusively toward undergraduate students was proposed (the U-league).

In the Rapid Soccer Competition alumni woulb be able to experiment their ideas into a RoboCup environment regardless of their time constraints (i.e. having more time to spare would post no advantage). Competitions would take the form of a tournament which would spam over the duration of the RoboCup event.

4 Discussion

This paper introduced the main technical and conceptual characteristics of a new miniature robotic platform. In particular, it was emphasized in the beginning of section 3 the advantages of shifting of focus from the play-

Figure 6: People get very attracted to the small robots

ing agents to the shared system. Furthermore, the three competitions showed in a more clear way how this collaboration shall be fostered toward the constant development of a versatile system for education and research.

In the last trimester of 2006 we already started some undergradutation class experiments where inexperienced second year students of engineering courses could learn to program and develop whole soccer playing teams in only five sessions in three weeks. Figure 6 gives an idea of the kind of entertaining atmosphere the mini robots produce on people, attracting crowds in the public demonstrations of the system. People feel much more attracted to the miniature robots with their limited behavioral skills than for the virtual agents such as those used in RoboCup soccer simulation league. The small robots seem to attract specially the kids. This is likely to help the use of the system in studies that require interaction of small children in the loop (see figure 1), typical in fields related to developmental cognitive studies such as developmental psychology and cognitive neuroscience.

References

[1] John Anderson, Jacky Baltes, David Livingston, Elizabeth Sklar, and Jonah Tower. Toward an undergraduate league for robocup. In *Proceedings of RoboCup-2003: Robot Soccer World Cup VII*, Lecture Notes In Artificial Intelligence. Springer, 2003.

[2] G. Caprari. *Autonomous Micro-Robots: Applications and Limitations*. PhD thesis, Federal Institute of Technology Lausanne, 2003.

[3] Olivier Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, Cambridge, MA, USA, 1993.

[4] Fredrik Heintz. Robosoc, a system for developing robocup agents for educational use. Master's thesis, Dept. of Computer and Information Science, Linkopings Univ., 2000.

[5] Harukazu Igarashi, Shougo Kosue, Yoshinobu Kurose, and Kazumoto Tanaka. A robot system for robocup small size league:js/s-ii project. In *Proceeding of RoboCup Workshop (5th Rim International Conference on Artificial Intelligence)*, pages 29–38, 1998.

[6] Hiroaki Kitano and Minoru Asada. The robocup humanoid challenge as the millennium challenge for advanced robotics. *Advanced Robotics*, 13(8):723–736, 2000.

[7] Jelle R. Kok and Nikos A. Vlassis. Using the max-plus algorithm for multiagent decision making in coordination graphs. In *RoboCup 2005: Robot Soccer World Cup iX*, Lecture Notes in Artificial Intelligence, pages 1–12. Springer, 2006.

[8] I. Noda, H. Matsubara, K. Hiraki, and I. Frank. Soccer server: A tool for research on multiagent systems. *Applied Artificial Intelligence*, 12:233–250, 1998.

[9] Shoichi Shimizu, Tomoyuki Nagahashi, and Hironobu Fujiyoshi. Robust and accurate detection of object orientation and id without color segmentation. In *Proceedings of RoboCup-2005: Robot Soccer World Cup IX*, Lecture Notes In Artificial Intelligence, pages 408–419. Springer, 2005.

[10] Richard Stallman and Roland McGrath. *GNU make : a program for directing recompilation*. Free Software Foundation, 1996.

[11] Frieder Stolzenburg, Oliver Obst, and Jan Murray. Qualitative velocity and ball interception. In *Proceedings of the 25th Annual German Conference on AI: Advances in Artificial Intelligence*, Lecture Notes In Computer Science, pages 283 – 298, 2002.

[12] Peter Stone. *Layered Learning in Multiagent Systems: A Winning Approach to Robotic Soccer*. MIT Press, 2000.

[13] Ken Sugawara Toshiya Kazama and Toshinori Watanabe. Traffic-like movement on a trail of interacting robots with virtual pheromone. In *Proceedings of the 3rd International Symposium on Autonomous Minirobots for Research and Edutainment (AMiRE 2005)*, pages 383–388, 2005.

[14] R.Y. Tsai. An efficient and accurate camera calibration technique for 3d machine vision. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Lecture Notes In Artificial Intelligence, pages 364–374, 1986.

[15] Manuela Veloso, William Uther, Masahiro Fujita, Minoru Asada, and Hiroaki Kitano. Playing soccer with legged robots. In *Proceedings of IROS-98, Intelligent Robots and Systems Conference*, Victoria, Canada, October 1998.

[16] Kazuhiko Yoshikawa. Eco-be!: A robot which materializes a watch's life. *Nature Interface*, (8):56–57, 2002.

社団法人 人工知能学会
Japanese Society for
Artificial Intelligence

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-A701-4 (5/4)

Successful Teaching of Agent-Based Programming to Novice Undergrads in a Robotic Soccer Crash Course

Rodrigo da Silva Guerra [1], Joschka Boedecker [1], Hiroshi Ishiguro[1,2], Minoru Asada [1,2]
[1] Graduate School of Engineering, Osaka University, Osaka, Japan
[2] JST Erato Asada Project
{rodrigo.guerra,joschka.boedecker,ishiguro,asada}@ams.eng.osaka-u.ac.jp

Abstract

This work describes a method for introducing agent based programming concepts as a hands-on experience targeted at inexperienced students learning C as their first programming language. The approach is based on three main features: (a) a simplified C interface to the RoboCup 2D soccer simulation framework; (b) a strict agent-centered polar vector arithmetics approach for describing soccer skills and agent behaviors; and (3) on a tournament at the end of the course to give students an opportunity to evaluate their programs against one another in a fun environment. We present data of our experience performing a five-day (15h) course with sixty second-year engineering bachelor students who had just barely learned the basics of computer programming. We show how, based on the described scheme, students with very limited computer programming experience became able to develop their own teams of autonomous agents, in some cases including concepts such as dynamic role-assignment, multi-agent coordination, and team formation. We defend the method presented here helps exposing the students to valuable experience ahead of their normal schedule, and boosting overall motivation and performance.

1 Introduction

Even though using real and virtual robots in engineering related courses is known to boost motivation of students [1, 12, 5, 4], care needs to be taken in the course design to minimize the many practical problems of working with the robots or simulation environments. This is especially true for students with little programming experience. In this paper, we present the contents and results from a very short introductory level programming course for undergraduate engineering majors using simulated robotic soccer as a framework to teach agent-based programming, and to give students an opportunity to get hands-on experience putting to work the programming skills they had just barely learned. The approach we chose corresponds to what Lund [8] terms *guided constructionism*, i.e., a combination of traditional constructionist approaches [10, 11] and explicit guidance in forms of lectures and coaching by more experienced students.

We built the course around the 2D soccer simulator [9] which has been widely accepted as a standard research and educational platform for multi-agent applications. Numerous works using this tool have been carried out, including both scientific level research papers (e.g. [7, 14, 13]) and agent programming courses (see e.g. [2, 5]). This helped making this system a robust platform for testing ideas in multi-agent disciplines. However, despite of its wide-spread and general acceptance, the two-dimensional simulation framework is still rather complex, requiring the kind of specialized knowledge typical of programmers with rather mature experience. While such complexity often regards features without which interesting multi-agent problems could not be properly attacked, this same complexity also comes as an obstacle to the non-experienced programming beginners as pointed out in [2, 5].

We believe the limited programming skills of students should not prevent them from experimenting with the basic concepts of agent-based programming. On the contrary, we think allowing these students to experiment with their first codes already in such a dynamic and motivating environment helps boosting their learning of programming concepts as they become necessary for producing more elaborate autonomous soccer teams. Aimplified interface to program the agents was created making it possible to cope with a very narrow time frame of only 5 sessions (each of 3 hours duration) with lectures and programming work plus an additional separate session for a class tournament.

The rest of the paper is organized as follows: Section 2 describes the simplified C-interface around which the course was formulated. Section 3 shows in detail the strict vectorial approach which worked as the cen-

tral conceptual tool for programming the agents. Section 4 present specific details about the organization of the course. Section 5 gives some qualitative impressions from the authors based on the obtained results. Finally, sections 6 summarizes the main contributions of the paper and discusses directions future works.

2 The Simplified C-Interface

We prepared a set of wrapping functions allowing students to implement their code in a very compact and simplified way requiring only the use of standard C code. The idea is similar to that of RoboSoc [6], but with a much stronger focus on simplicity – at the price of loosing generality. These wrapping functions were constructed around the Trilearn base code published in 2002 [3].

The whole environment provided by the Trilearn base code was wrapped into four control functions and a few sense/act functions. Some of the sense/act functions were simple wrapping of existing C++ functions of the original code while others were implemented from scratch or heavily simplified. The four control functions are described below:

- `pv_init` – Includes all initialization procedures that should happen before the main loop in order to prepare the agent and including initial communication with the server. The only argument passed to this function is the desired team name;
- `pv_update` – Includes all the necessary routines for parsing messages received from the server and updating the internal world model of the Trilearn base code. This function is called inside the main loop right at the beginning, so that actualized sensor values can be assured;
- `pv_flush` – Takes all accumulated action commands, assembles them into messages and sends them to the server so that the agent can actually perform them. This eliminates the burden of sending the commands every time an agent takes a decision. Should be called in the last instruction inside the main loop;
- `pv_close` – Performs all procedures necessary for finishing the program, de-allocating resources.

The listing 1 shows an example of a complete agent which is capable of passing the ball to a teammate (variable names are consistent with figure 1). Together with our strict vector approach (which is detailed in section 3) this simplified interface enabled the students with very little programming experience to program a variety of soccer playing behaviors in a very clear and intelligible way.

3 The Vector Arithmetics Approach

Most people learn basic operations with vectors already at school. Vector arithmetics are visually intuitive, especially in the two dimensional space, where calculations can be approximated by sketching simple strokes on a piece of paper. More than that, the analogy of the soccer field as a two-dimensional space gives a very straight

Listing 1: Example of a simple "passing" agent

```
#include <cinterface.h>
int main(int argc, char *argv[])
{
        struct strct_vector c;
        struct strct_vector e;
        pv_init("MyTeam");
        while (pv_update())
        {
                c = pv_getball();
                e = pv_getteammate(1);
                if (pv_cankick())
                {
                        pv_kick(e);
                } else {
                        pv_steerto(c);
                }
                pv_flush();
        }
        pv_close();
}
```

forward interpretation for directions and magnitudes of two-dimensional vectors which can represent forces, accelerations, velocities of players and ball. In fact, this approach is so straight forward that it is very common to see implementations involving two-dimensional vector arithmetics of some kind across the various RoboCup Soccer leagues.

In our approach we developed a complete framework where all essential elements necessary for making a team of soccer playing agents could be implemented by the exclusive use of two-dimensional vector arithmetics and nothing else. Moreover, we took care of describing all necessary components relative to the self in an agent-centered approach excluding completely explicit global coordinates of any sort. This agent-centered perspective allows a deeper understanding on the agent perspective embodied with sensors and immerse in the environment.

In the figure 1 we illustrate how one can take advantage of the two-dimensional vector representation for a very visual strategy planning. Suppose, for instance, your agent were to mark an opponent by placing itself between the opponent and the ball. The vector from the ball to the opponent, according to the figure 1, is given by the expression $\mathbf{c} - \mathbf{d}$. One could simply scale down this vector, let's say, half-way ($\frac{\mathbf{c}-\mathbf{d}}{2}$), and sum to the vector representing the direction to the ball, yielding the expression $\frac{\mathbf{c}+\mathbf{c}-\mathbf{d}}{2}$. This last expression would be the direction the agent should go. Similarly, if an agent were to kick the ball ($\mathbf{c}$) into the center of the goal ($\frac{\mathbf{a}+\mathbf{b}}{2}$), it would need to go towards $\mathbf{c} + (\mathbf{c} - \frac{\mathbf{a}+\mathbf{b}}{2})(r_r + r_b)$, where r_r and r_b are the radius of the robot and the ball respectively (compare with the previous expression).

A web-based applet was developed in Java as a tool for helping the students draw their vector arithmetics on the screen of the computer and test their ideas on different game situations. See figure 2 for a screen shot.

To our understanding this simplified vector approach provides a powerful unifying interpretation which can be used across for a variety of very different soccer robots.

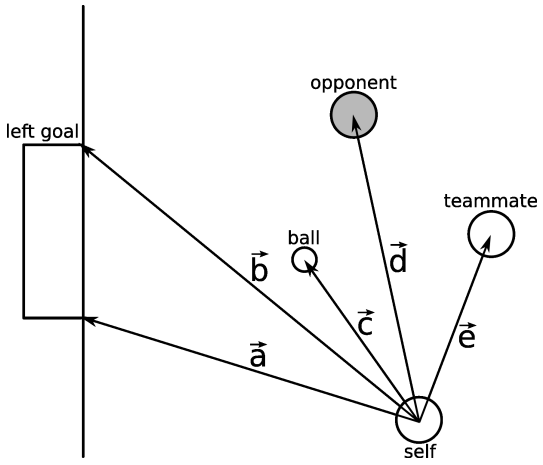


Figure 1: An example of game situation where all necessary elements can be visualized in terms of two-dimensional vectors

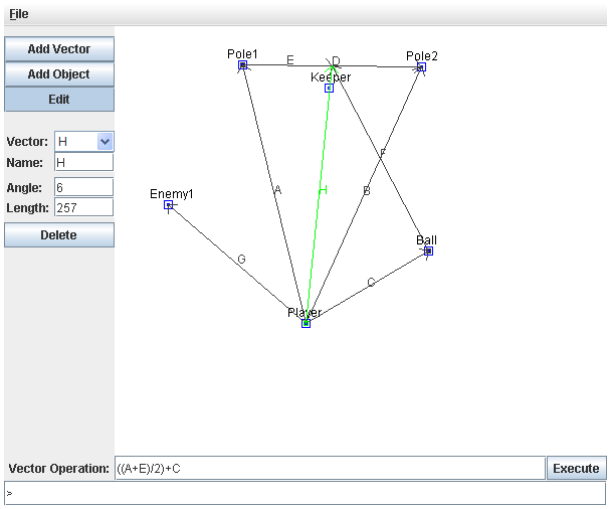


Figure 2: Screen shot of the Java vector applet developed for enabling students to visualise their different strategies and check resulting expressions.

See below some examples of very straight forward interpretation in the case of the most popular platforms.

Omni-directional camera: Suppose the vertical axis of revolution of the omni-camera is normal relative to the horizontal plane (of the field) and fixed in the robot (which is often the case). The center of the revolved image is the origin for the two-dimensional vector representation. Radial distances from the origin in the image have direct mapping into distances from the robot body.

Projective camera: For the sake of simplicity lets assume square pixels, pinhole model, and center of projection on center of image (usually very practical approximation when it comes to non-precision robotics). One can assume a simple mapping from the distances of arbitrary image points to the center of the image into the corresponding horizontal and vertical angles of their respective rays (taking the pinhole as the vertex). See figure 3. The horizontal angle α give the direction of the two-dimensional vector while the vertical angle β gives

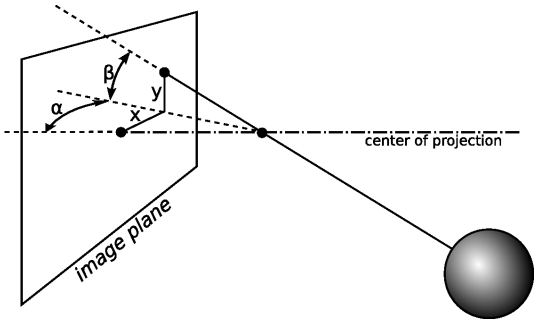


Figure 3: Example of simplified pinhole camera. One can find a simple mapping from the distances x and y to the corresponding angles α and β respectively. In the two-dimensional vector approach the angle α gives the direction and the magnitude is given by $d = f(\beta)$. This relation is derived by simple trigonometry

a direct mapping into distances on the floor (assuming camera is at constant high).

Pan & Tilt: Consider, images on the center of projection of the camera. Pan angles can be used directly as the directions of the two-dimensional vectors while tilt angles map into distances in the floor. The pinhole camera attached to the pan & tilt mechanism can be approximated by simply summing pan and tilt angles with horizontal and vertical camera angles respectively.

Control theorists experienced in mobile robotics would probably still argue it is not so straight forward to derive control laws when you have non-holonomic restrictions in the mobility of the robots (e.g. two wheeled differentially driven robots). Such problems have been focus of constant research in the past years due to the challenging control problem they represent (i.e. closed-form solution does not exist). But here again, this should not come as an obstacle to the learner who is eager to put a robot into movement into a simplistic and practical experiment – otherwise all beginners in robotics would require robots equipped with omni-wheels.

In the original two-dimensional simulation framework agents are moved by the sucessive use of `dash` & `turn` commands, for, respectively, turning and moving forward the agent's position. We implemented an interface which mimics the effect of having differential wheels (i.e. transforming the fictitious wheel velocitys into a corresponding series of `dash` & `turn` commands). This was done in an effort for keeping the virtual agent as closely related as possible to the most typical robotic architectures – at the price of restricting the original (less realistic) maneuverability of the agents.

In our simulated differential driven robot the velocity of each wheel could be set into three different constant values, both backward and forward, or zero (stopped). In such case, for example, $(+3, +3)$ would make the robot go forward, $(+2, -2)$ would make the robot spin clockwise and $(+3, +1)$ would make the robot go in an arc-trajectory to the direction forward/right. In order to face the problem of steering the robot into arbitrary

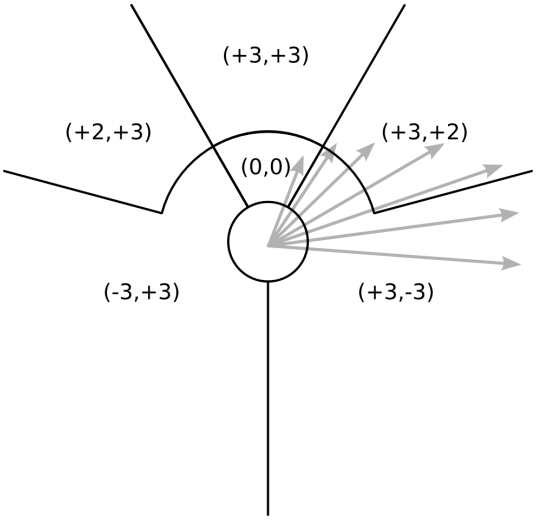


Figure 4: State machine implementing a simple steering algorithm for a two-wheeled differential driven robot

locations we implemented a simple state-machine. See figure 4. This approach was chosen for being simple and effective enough so that the student with their limited experience in programming (and robotics) could easily understand and eventually improve their agent's navigatibility by customizing their own steering methods. This way they not only could understand and use very realistic robots in their experiments but also had contact with a state-machine for decision making in their code.

4 Course Format

The course was performed at the Osaka University during the month of March of 2007 to an audience of 62 second-year engineering students (amoung which only 4 were females). It was composed by five separate lectures of three hours each, given twice a week plus an extra sixth lecture where the tournament was realized. The overall program of the class is summarized in the table 1.

Except for the first class, which was almost completely theoretical, all the other classes were build around the student exercise and practice. In the end of each class a homework was a assigned, which would involve and stretch concepts learned in class, but also bring up issues which would only be formally introduced in the next class. This was done in order for them to experience the needs before trying the solution, so that when the solution was presented its value could be more promptly understood. For instance, students would compile using the command prompt in the first class, but be introduced to Makefiles in the second class, and they would implement different skills (such as passing) inside their main loop in the second class but learn how to create more generic parameterized functions in the third class, and so forth. Despite all the theory being not completely new to them (they came fresh from a theoretical introduction to programming), it takes quite a lot of practice and experimentation until they can really put their knowledge into work in such a dynamical situation.

Day	Content summary
1st	Introduction, review of concepts of vector arithmetics, basic agent programming concepts
2nd	Introduce the most elementary code, explain the use of a simple Makefile, start working on very atomic behaviors (e.g. kick to goal and run to the ball)
3rd	Start generalizing these atomic behaviors with the introduction of simple functions (e.g. find closest teammate)
4th	Make the agent even more versatile introducing dynamic role assignment and general changes of behavior according to things such as own-id, side of play, etc.
5th	Divide the class in groups of four students and start developing their own teams for the final tournament.
6th	Tournament

Table 1: Summary of the course program

5 Results

A questionnaire was formulated in order to help evaluating the evolution of the interests and degree of confidence of the students in three main aspects: (a) programming, (b) robotics and (c) soccer. The questionnaire was distributed twice, firstly before the first class and again later, after the first round of games during the tournament. This questionnaire was extensive and composed by several multiple choice questions. In these questions students had to classify their interests, previous experience and self-confidence on many criteria regarding themes directly or indirectly related to the contents of the course. The speculations discussed in this section are based on the results of this questionnaire, together with the collected homework, the final team code and a final report.

To our surprise, all students were unanimous in that they had never even heard about the term "agent programming" before. On the other hand, when asked to list from the top of their heads names of robots and soccer players, robots like ASIMO and AIBO were cited more often than the most often cited soccer player (which was Ronaldinho). The authors interpret this as an strong indication that robotics is a rather popular subject among young engineering students in Japan.

At the end of the course, despite the fact that we didn't talk about real robots during any of the practical classes, students showed that they have increased significantly their confidence about how much they believed they could make a real robot play soccer. See figure 5.

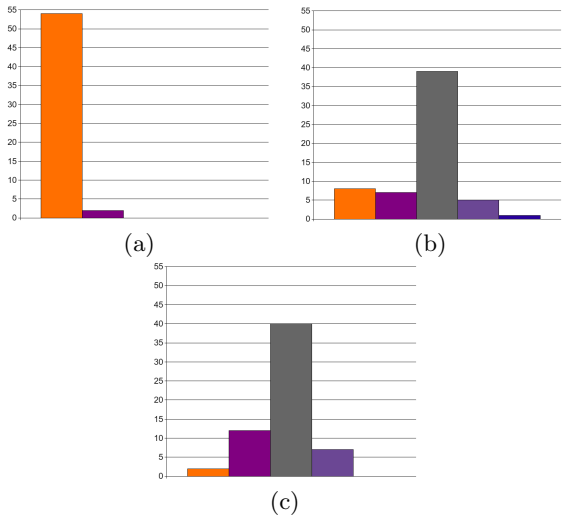


Figure 5: Results of the pool where students were invited to self-evaluate to what extent they believed they could make a real robot play soccer (a) when asked before and (b) when asked after the course, and similarly for making (c) an artificial agent play soccer after the course. The five columns in each chart represent the total number of students that chose each of the corresponding five options, which were, from left to right: (1) can nothing at all, (2) can do almost nothing, (3) can do a little, (4) can do well, (5) can do very well. (the neutral option was explicitly omitted)

Furthermore a much bigger number of students declared to have become more interested in both robot soccer and agent programming. Moreover, in the end of the course we found a very strong correlation between their confidence on their own agent programming skills with their confidence on how much they believed they could make a real robot play soccer (compare charts in figure 5-b and 5-c). On our interpretation the above indicates that, although we worked only with a very simplistic two-dimensional simulation environment, the students could still relate this to a broader concept applicable to real robots.

Furthermore, the authors noticed the positive effects of the tournament on the motivation of the students, which were often verbally expressed on their final reports, and also rather evident when reviewing the videos recorded during the tournament.

6 Discussion

This work presented an effective approach for exercising basic programming skills in the very dynamic environment of soccer simulation in a very short period of time. The method here presented together with the observed results support the idea that limitations on programming skills do not prevent students achieving their goals in the complex and dynamic multi-agent environment.

During the class we collected some strong evidence of the popularity of robotics among the students. For future work we plan to investigate how the (eventual) use of real robots in class would influence their performance if compared to the data collected during this course in which we used simulation only. Moreover, we plan to make our evaluation methods conform to suggested [12] standards thus enabling easy comparison of results.

7 Acknowledgements

The authors would like to thank to Chisato Yoshida for her help on the empirical portions of the work, to Matthias Bohnen for developing the nice Java applet for helping on vector arithmetics, to the teaching assistants for their great help both during the classes and later on analysing the collected data, and to the students themselves for their collaboration and understanding. The authors also want to thank JSPS and JST and the Japanese Ministry for Sports and Education for their financial support.

References

[1] John Anderson and Jacky Baltes. An agent-based approach to introductory robotics using robotic soccer. *International Journal of Robotics and Automation*, 21(2):141 – 152, April 2006.

[2] S. Corradeschi and J. Malec. How to make a challenging ai course enjoyable using the robocup soccer simulation system. In *RoboCup-98: Robot Soccer World Cup II*, Lecture Notes In Artificial Intelligence. Springer, 1998.

[3] Remco de Boer and Jelle R. Kok. The incremental development of a synthetic multi-agent system: the uva trilearn 2001 robotic soccer simulation team. Master's thesis, University of Amsterdam, 2002.

[4] Barry S. Fagin and Laurence Merkle. Quantitative analysis of the effects of robots on introductory computer science education. *ACM Journal of Educational Resources in Computing*, 2(4):1–18, December 2002.

[5] Frederik Heintz, Johann Kummeneje, and Paul Scerri. Simulated robocup in university undergraduate education. In *RoboCup 2000: Robot Soccer World Cup IV*, Lecture Notes In Artificial Intelligence, pages 309 – 314. Springer, 2001.

[6] Fredrik Heintz. Robosoc, a system for developing robocup agents for educational use. Master's thesis, Dept. of Computer and Information Science, Linkopings Univ., 2000.

[7] Jelle R. Kok and Nikos A. Vlassis. Using the max-plus algorithm for multiagent decision making in coordination graphs. In *RoboCup 2005: Robot Soccer World Cup iX*, Lecture Notes in Artificial Intelligence, pages 1–12. Springer, 2006.

[8] Henrik Hautop Lund. Robot soccer in education. *Advanced Robotics*, 13(8):737–752, 1999.

[9] I. Noda, H. Matsubara, K. Hiraki, and I. Frank. Soccer server: A tool for research on multiagent

systems. *Applied Artificial Intelligence*, 12:233–250, 1998.

[10] S. Papert. *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, New York, 1980.

[11] S. Papert. Constructionism: A new opportunity for elementary science education. A proposal to the National Science Foundation. Massachusetts Institute of Technology, Media Laboratory, Epistemology and Learning Group., 1986.

[12] Elizabeth Sklar, Simon Parsons, and Peter Stone. Using robocup in university-level computer science education. *ACM Journal on Educational Resources in Computing*, 4(2), 2004.

[13] Frieder Stolzenburg, Oliver Obst, and Jan Murray. Qualitative velocity and ball interception. In *Proceedings of the 25th Annual German Conference on AI: Advances in Artificial Intelligence*, Lecture Notes In Computer Science, pages 283 – 298, 2002.

[14] Peter Stone. *Layered Learning in Multiagent Systems: A Winning Approach to Robotic Soccer*. MIT Press, 2000.

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-A701-5 (5/4)

四足歩行ロボットによるアドホックネットワークの構築

Development of a Novel Ad-Hoc Network with 4-legged Robots for a Disaster Scene

植村 渉
Wataru UEMURA
龍谷大学理工学部
Ryukoku University
wataru@rins.ryukoku.ac.jp

1 はじめに

地震や火災などの災害現場では，被災者の発見は重要事項の一つである．そのような災害現場において，従来から敷設されているネットワークは，それ自身も被災している可能性が高く，そのようなネットワークを災害救助に用いることは困難である．このような環境において，端末同士でネットワークを構築するアドホックネットワークが注目されている [7, 10, 11]．災害現場においては，被災者を発見するためのロボットが重要である．ロボットは災害現場に入り込み，現場の情報を操作者へ伝える必要がある．ロボットを遠隔操作するためには，無線による通信が期待されるが，そのために操作者は無線通信できる距離から操作する必要がある．しかし現場は危険なため，操作者はできる限り離れていたいというトレードオフが生じる．そこで，ロボットと操作者間の通信は直接通信ではなく，中継器を間に挟むことで距離を延ばす必要がある．このような中継を含む無線通信を**マルチホップ**通信という．本節では，災害現場でのネットワークの端末としてアイボ[1] に着目し，アイボの遠隔操作について議論する．

アイボは，ソニー株式会社が生産した[2] 四足歩行ロボットである．10 万画素のカメラを搭載しており，自律歩行したり首を動かしたりできるため，自分の周辺の情報を映像として獲得することができる．また，ステレオマイクとモノラルスピーカを搭載しているため，周辺の音情報を操作者へ伝えたり，操作者がアイボを通じて呼びかけを行ったりすることも可能であり，被災者との会話が期待できる．アイボの移動は，足の長さの関係もあるため，段差が大きいところは通ることができない．そのため，がれきの多い災害現場では不適切であるが，そうでない災害現場，例えば有害ガスが漏れた場所などでは足元は安定しているため，有効である．本論文では，アドホックネットワークにおける被災者発見のためのロボットシステムのプロトタイプを提案し，そこから生じる問題について検討する．

災害現場において，救助部隊が直接探索をすることができない領域を，被災者発見ロボットが探索を行うことは大変重要である．しかし，ロボットが災害現場の内部奥深くへと進むと，操作者と通信できなくなる場合がある．そのような場合，他のロボットが，現場のロボットと操作者との通信を中継することが必要となる．直接通信のことを**シングルホップ**（single hop）といい，中継を介した通信のことを**マルチホップ**（multi hops）という．アドホックネットワークにおける経路設定では，端末の状態を管理する基地局が存在しないため，ネットワーク全ての端末の情報をリアルタイムに獲得することは難しい．操作者とそれぞれの端末は，通信を行うたびに動的に経路を設定しなければならない [4, 5, 8]．

アドホックネットワークのための経路設定は，大きく二種類に分けられる．一つは**リアクティブ型**であり，もう一つは**プロアクティブ型**である．また，それらを組み合わせた**ハイブリッド型**も存在する．リアクティブ型の経路設定では，経路設定の要求が起こってから，経路設定のための通信を行う．この方法では，経路設定の必要がない場合は通信を行わないため，帯域を有効に使うことができ，またバッテリーも長持ちする．しかし，経路設定の必要が生じてから，経路設定のための情報を交換し合うため，経路設定に時間がかかる欠点がある．代表的なリアクティブ型の経路設定として，AODV(. . . . .)[9] や DSR(. . . . .)[3]，そして DYMO(. . . . .)[1] などが挙げられる．一方，プロアクティブ型の経路設定では，それぞれのノードは常に経路設定のための情報を交換し合う．長所と短所はリアクティブ型と逆になり，すばやい経路設定ができる代わりに，経路設定のための帯域を常に必要とし，電力も必要とする．代表的なプロアクティブ型の経路

[1] ”AIBO”および”AIBO”ロゴはソニー株式会社の商標および登録商標である．
[2] 残念ながら，2006 年 3 月に生産中止となった．

Figure 1: 4-legged Robot AIBO

設定として，TBRPF( )[6] や OLSR( )[2] などが挙げられる．本節では，災害現場での利用を想定するため，長時間のバッテリー駆動を可能とするリアクティブ型の経路設定である AODV を中心に用いる．本論文では，被災者発見のためのアドホックネットワークのプロトタイプを提案する．操作者は，複数のロボットを操作することができ，またセンサ情報を得ることができる．ロボットが操作者の通信可能範囲から離れた場合は，間にいる他のロボットがパケットを中継することで，通信を実現する．現段階では，本システムのロボットは自分で判断して動くことはできず，操作者による遠隔操作による移動のみである．将来的には，各ロボットが自分で判断し，より適切な中継点へ移動したり，以上を操作者へ報告することなどが考えられる．

以下，経路設定アルゴリズムについて説明し，使用するロボットであるアイボの仕様について紹介する．そして，被災者発見システムについて提案し，実験により通信性能を確認する．最後に，まとめと今後の課題を述べる．

2 経路設定プロトコルと四足歩行ロボットについて

ここでは，経路設定のプロトコルと，システムで使用した四足歩行ロボットについて紹介する．

2.1 AODV 経路設定アルゴリズム

AODV（ ）経路設定アルゴリズムは，IETF（ ）[13] の MANET（ ）ワーキンググループによって制定され，リアクティブ型の経

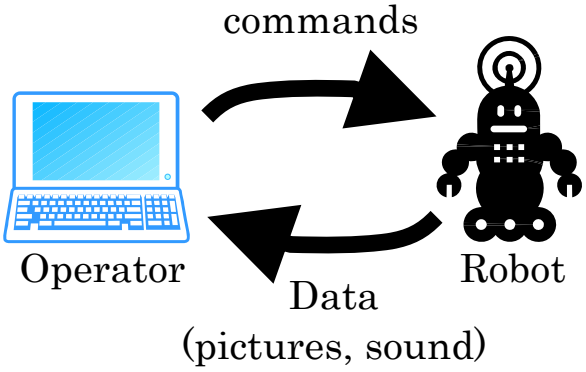


Figure 2: BabyTigers DASH を拡張した遠隔操作システム

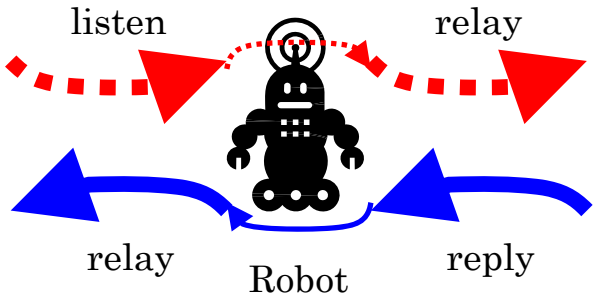


Figure 3: パケットを中継する端末の仕組み

路設定アルゴリズムの一つである．AODV 経路設定では，経路要求が起こってから経路設定の作業を始める．経路設定の必要がない場合は通信を行わないため，帯域を有効に使うことができ，またバッテリーも長持ちする．しかし，経路設定の必要が生じてから，経路設定のための情報を交換し合うため，経路設定に時間がかかる．パケットを送信する端末は，隣接の端末の情報だけを知ればよく，ネットワーク全体の情報を知る必要はない．つまり，経路設定の情報は，ネットワーク内の全ての端末に分散されていると言える．

AODV では，経路設定を要求する端末によって RREQ（ ）メッセージがブロードキャストされる．それを中継する端末は，そのメッセージにコストを付加し，次の端末へ中継する．もし，一度受け取ったメッセージを再び受け取った場合は，中継せずに破棄する．目的とする端末に最初に届いたメッセージが，最もコストが低い経路を経由したことになる．そして，RREQ メッセージを受信した目的端末は，RREP（ ）メッセージを，経路設定を要求した端末に対して送信する．このようにして，経路を確立する．

2.2 四足歩行ロボット

次に，ネットワークの終端と中継の両方の機能を備えた端末として四足歩行ロボットであるアイボを紹介する．アイボは，IEEE 802.11b に準じた無線ネットワークカード

と，10 万画素の解像度を持つデジタルカメラ，ステレオマイクロフォンとモノラルスピーカを持つ（図 1）．そして，8,000Hz のサンプリングレートで wav ファイルを再生することができ，16,000Hz のサンプリングレートで録音することができる．また，各足は二箇所の関節を持ち，それぞれのモータを動かすことで歩くことが可能である．

災害現場において，四足の足を動かし移動し，カメラとマイクによる視聴覚情報より被災者の発見に努めることができる．被災者を発見した後は，操作者はアイボのマイクとスピーカを通して，被災者と会話をすることが可能である．アイボが操作者の通信圏内から離れた場合，直接通信することは不可能になる．そのため，それ以外のアイボを間に移動させ，通信を中継しなければならない．つまり，アイボは，終端端末としての働きと共に，中継端末としての働きをする必要がある．

アイボは CPC（ ）と呼ばれる OPEN-R（ ）[12] ハードウェアから成り立っている．OPEN-R は，ロボット間の構成の違いを吸収し，異なるプラットフォームへの移植を容易にした仕様である．アイボではオブジェクト指向言語である C++言語を用いてプログラムを書くことができる [14]．代表的なアイボの利用方法として，ロボカップ [16] 四足リーグ [17] がある．ロボカップでは，人間が操作することなく，ロボットがサッカーをする競技であり，最終的には人と試合をすることが目的である．その中の四足リーグでは，ロボットを作ることが目的ではなく，同一ハードウェアと言う条件において，各チームがどれだけプログラムのレベルを高められるかを目的として設置されている．なお，単なるプログラムの向上であれば，計算機上だけでサッカーを行うシミュレーションリーグがあるが，四足リーグは実世界でサッカーを行うところに難しさがある．本研究で提案するプログラムは，四足リーグに参加しているチームである BTD( )[15] チームのソースコードをベースとした．このチームは，大阪大学と大阪市立大学，そして龍谷大学の合同チームである．本来は前述の通り，人がロボットを操作することない．しかし，デバッグを行うにあたり，ロボットがフィールド上でどのように物を見ているのか，どのように音を聞いているのかといった情報を遠隔のノートパソコンで確認することは重要であるため，デバッグモードとしてそのような機能を組み込んでいる．その機能を利用し，被災者発見システムへ適用する．アイボと操作者とは，TCP による無線通信でつながり，遠隔操作の制御プロトコルは独自のものを利用している．デバッグのためのプログラムであるため，直接通信しか考慮していない．そこで，制御プロトコルを拡張し，中継通信を実現する．

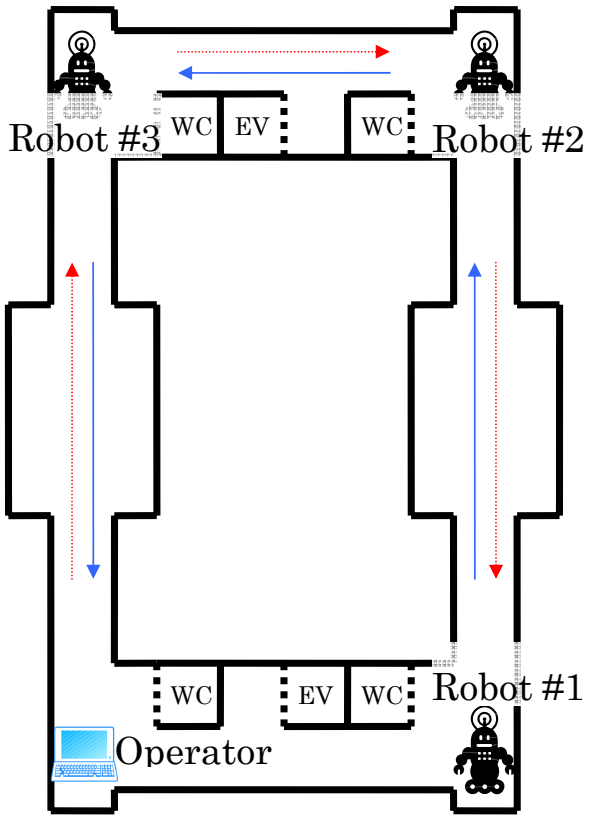


Figure 4: 実験環境

2.3 被災者発見システム

BTD デバッグプログラムは，ポート 11111 を通してアイボと通信をする（図 2）．そのため中継端末は，ポート 11111 の通信で待ちうけを行い，受け取ったパケットを次の端末のポート 11111 へ送信する必要がある．また，次の端末からの返信を受け取り，元の端末へそのパケットを引き渡す必要がある（図 3）．今回，中継端末における転送のためのバッファとして，1024 バイトのメモリを確保する．なおキャッシュメモリとしての役割は持たせていないため，一度に一つずつしか転送をしない．ネットワーク上に一つのメッセージしか流れないため，中継器を多く用意すればするほど，転送にかかる時間は長くなることが予想される．将来的には，緩衝材としてキャッシュメモリを用意し，中継器の数に依存しない中継時間の実現が必要となる．

2.4 通信性能の測定

ここでは，四足歩行ロボットによる作成したプロトタイプの性能を評価する．

図 4 のように，フロアの四隅に三台のロボットと操作者（Operator）を配置する．ロボットは，全台同じプログラムが稼動しており，操作者が中継端末か終端端末かの役割を決定する．いつでも，役割を入れ替えることが可能である．フロアの長辺は約 55m であり，短辺は約 15m

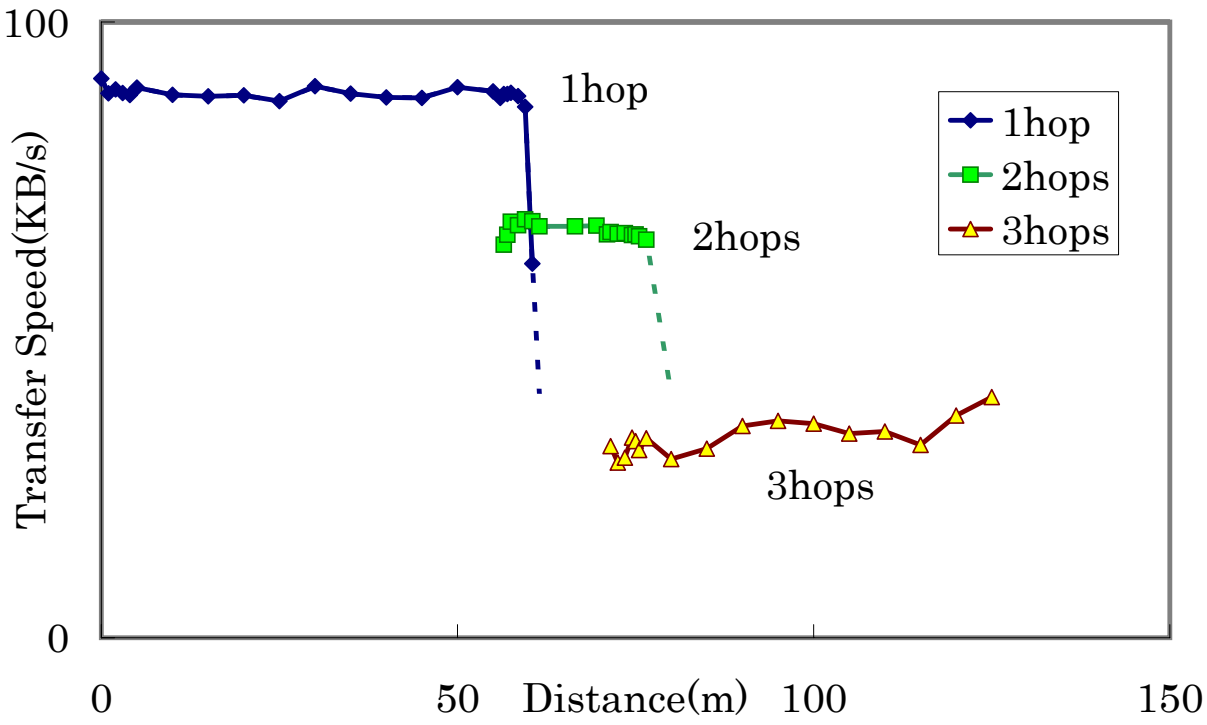


Figure 5: 距離に対する通信速度

である．それぞれのロボットは，同じ辺上にいるロボットとのみ直接通信できる．例えば，ロボット#1 とロボット#3 は，直接通信できない．この配置においては，ロボット#1 と操作者とは直接通信が可能であるが，ここでは直接通信をせず，ロボット#2 とロボット#3 を中継して通信させる．操作者とロボット#3 の通信を 1 ホップ通信とする．これは直接通信となる．ロボット#2，ロボット#1 との通信において，2 ホップと 3 ホップの通信が必要となる．各辺上にロボットを配置し，操作者との総距離に対する通信速度を測定する．各辺の角に到達すると，そのロボットは中継端末とし，次の辺上を新しいロボットが動く．カメラ画像の転送を行い，25 秒間の平均を転送速度として扱う．

結果が図 5 である．それぞれの辺上における転送速度はほぼ一定であり，距離に対する影響は見受けられない．IEEE802.11b の規格においても，室内においては 50m ほど，室外では 150m ほどが到達距離であるため，この環境では見通しがよく十分な通信ができていると考えられる．各角において，中継器をおかずに進める場合，すぐに見通しが悪くなり通信が遮断されるため，転送速度が著しく遅くなることがわかる．また，中継器を増やしホップ数が増えると，ネットワーク上での 1 つのメッセージに対する送通信回数が増加するため，通信速度が遅くなる．3 ホップでは，1 ホップの 1/3 ほどの速さであるが，周りの状況を十分に確認できるカメラ画像が安定して得られた．

2.5 まとめ

災害現場では，被災者の早期発見は最重要タスクのうちの一つである．人間が入ることができない現場も存在する．そのような場合，ロボットが被災者を発見することが重要である．それぞれのロボットは操縦者へ自分の周りの情報を報告しなければならない．ロボットと操作者の距離が離れていたり障害物があるとき，直接通信が困難になる場合がある．このとき，他のロボットが間に入り，通信を中継すればよい．つまり，中継ロボットによってアドホックネットワークが構築される．本節では，ネットワーク端末として四足歩行ロボット，アイボを用いて構築したアドホックネットワークについて提案した．アイボは，搭載しているカメラを用いて写真を撮影することができ，無線ネットワークを通してデータを送信することができた．また，操作者と離れているときは，他のアイボがそのデータを中継することで，離れたアイボの情報を得ることができた．本実験において最大ホップ数は 3 であったが，それは我々の仕様で制限はしていない．アイボに限らず，中継の役割を果たす端末を使えば，ホップ数を増やすことが可能である．

本研究では，アイボは完全に遠隔操作されるだけの端末であった．今後の課題として，アイボが自ら周辺環境の情報を獲得し，よりよい電波環境へ移動したり，「大きな音がした」とか「何かが動いた」といった環境の変化情報を操作者へ伝える仕組みを組み込めれば，被災者の発見に役立つと考えられる．

謝辞

本研究は文部科学省のハイテク事業による私学助成を得て行われた.

参考文献

[1] I.Chakeres, Boeing, C.Perkins, and Nokia: Dynamic MANET On-demand (DYMO) Routing, Internet Engineering Task Force(IETF), Internet-Draft: draft-ietf-manet-dymo-05.txt (June, 2006).

[2] T.Clausen and P.Jacquet: Optimized link state routing protocol(OLST), Internet Engineering Task Force(IETF), Internet-Draft: draft-ietf-manet-olsr-11.txt (July, 2003).

[3] D.B.Johnson, D.A.Maltz, and Y.Hu: The dynamic source routing protocol for mobile ad hoc networks(DSR), Internet Engineering Task Force(IETF), Internet-Draft: draft-ietf-manet-dsr-10.txt (July, 2004).

[4] S.Kikuchi, H.Tsuji, R.Miura and A.Sano: "Mobile-Terminal Localization Based on Local Scattering Model in Multipath Environments," , Vol.J87-B, No.12, pp.2020–2028, 2004.

[5] T.Mukai, H.Murata and S.Yoshida: "Study on Channel Selection Algorithm and Number of Established Routes of Multi-hop Autonomous Distributed Radio Networks," , Vol.J85-B, No.12, pp.2080–2086, 2002.

[6] R.Ogier, M.Lewis, and F.Templin: Topology dissemination based on reverse-path forwarding(TBRPF), Internet Engineering Task Force(IETF), Internet-Draft: draft-ietf-manet-tbrpf-11.txt (Oct, 2003).

[7] M.Okamoto, H.Sugiyama, T.Tsujioka and M.Murata: "Low Power Consumption Type Multi-robot Network System," , CS2005-21, Vol.105, No.280, pp.31–36, 2005.

[8] Osama H.Hussein, Tarek N.Saadawi, and Myung Jong Lee: "Probability Routing Algorithm for Mobile Ad Hoc Networks' Resources Management," , Vol.23, No.12, pp.2248–2259, 2005.

[9] C.Perkins, E.Belding-Royer, and S.Das: Ad hoc on-demand distance vector(AODV) routing, Internet Engineering Task Force(IETF), Internet-Draft: draft-ietf-manet-aodv-13.txt (July, 2004).

[10] Martijn N.Rooker and Andreas Birk: "Combining Exploration and Ad-Hoc Networking in RoboCup Rescue," , LNAI 3276, pp.236–242, 2005.

[11] H.Sugiyama, T.Tsujioka and M.Murata: "Victim Detection System Consists of Networked Mobile Robots," , Vol.46, No.7, pp.1777–1788, 2005.

[12] http://openr.aibo.com/

[13] http://www.ietf.org/

[14] http://www.jp.aibo.com/

[15] http://www.kdel.info.eng.osaka-cu.ac.jp/robocup/

[16] http://www.robocup.org

[17] http://www.tzi.de/4legged/bin/view/Website/WebHome

動的環境におけるエージェント配置手法の提案

Muliti-Agent Positioning Mechanism in the Dynamic Environment

秋山英久 野田五十樹
Hidehisa AKIYAMA and Itsuki NODA
産業技術総合研究所
National Institute of Advanced Industrial Science and Technology
{hidehisa.akiyama, I.Noda}@aist.go.jp

Abstract

This paper describes our novel agent positioning mechanism for the dynamic environment. This mechanism utilizes Delaunay Triangulation to split a environment space with sample data and interpolates output values based on the shading algorithm in 3D computer graphic domain. This method is very simple, but runs fast and has high accuracy and scalability. In the experiment, we compared our method with other function approximation method on the RoboCup Soccer Simulation environment.

1 はじめに

マルチエージェントシミュレーションにおけるさまざまな問題では，各エージェントの空間的な配置がシミュレーション結果に重要な影響を及ぼすことがある．例えば，predator-pray 問題のように，対象物体の移動に応じてエージェントの移動位置を変化させなければならないタスクでは，入力となる環境の状態と，出力となるエージェントの移動位置とをマッピングしなければならない．同様に，サッカーのように動的な環境でチーム対戦するゲームにおいては，プレイヤとなる各エージェントが現在の状況に応じて個々の判断で移動し続けなければ，チームのパフォーマンスは著しく低下してしまう．

このような問題に対して，エージェントが移動すべき位置を素早く獲得させるためには，人間の観察者からの教示情報を用いる教師あり学習が効果的である．しかしながら，任意の状態に対してエージェントが移動すべき位置を事前に与えておくことは困難であるため，与えられた有限のサンプルから出力値を補間する何らかの内挿関数が必要となる．そこで，本稿では，環境の状態を入力とし，エージェントの移動位置座標を出力とする内挿関数の獲得に効果的なモデルとして，Delaunay 三角形分割と線形補間を組み合わせた手法を提案する．実験では，RoboCup サッカー 2D シミュレータ (RCSS)[3, 4]を用いて，性能を評価する．

```
GetSBSPPosition( Num )
 入力: プレイヤの役割番号 Num
 出力: プレイヤの移動位置
1. 基準位置 := BasePosition( Num );
2. 引力係数 := BallAttract( Num );
3. 移動位置 := 基準位置+ ボール移動量ベクトル * 引力係数
4. 移動可能領域 := MovableRegion( Num );
5. if 移動位置が移動可能領域の範囲外
6.    移動位置を移動可能領域内に調整
7. return 移動位置
```

Figure 1: SBSP の基本アルゴリズム

2 従来手法

サッカーのようなボールゲームにおいては，ボールが最も重要な注目対象であり，その位置を重要な状態変数とする考えは自然である．RoboCup サッカーシミュレーションにおいては，各エージェントの配置をボールの位置に応じて決定する手法として Situation BasedStrategic Position[1]が良く知られている．

2.0.1 Situation Based Strategic Position

Situation Based Strategic Position(SBSP) では，ボールの位置座標を入力とし，プレイヤの移動位置座標を出力する関数を用意する．更に，移動可能領域の制約条件を組み合わせることで，各々のプレイヤの最終的な移動位置座標を決定する．SBSP の基本的な移動位置決定アルゴリズムを図 1 に示す．

1 行目の基準位置とは，ボールがフィールド中央にある場合のプレイヤの移動位置を表す．2 行目の引力係数とは，ボールの移動に追従する割合を表す係数で，通常，$[0, 1]$ の実数である．そして，3 行目の計算によって，プレイヤの移動位置がほぼ決定する．ここでの計算は，基準位置に

対して，ボールの移動量に引力係数を掛けたベクトルを足し合わせるという，単純な線形関数となっている．よって，引力係数の値が 1 に近付くほど，ボールの移動に対してプレイヤが追従することになる．最後に，プレイヤの移動可能領域を調べ，調整を施す．

SBSP はかなり単純なモデルであり，他のプレイヤの存在を考慮していない．しかし，全プレイヤがボールに注目しているという前提であれば，見かけ上チームを上手く機能させることができる．実装も容易であることから，RoboCup サッカーシミュレーションに参加するほぼ全てのチームが SBSP と同一か類似のモデルを採用している．

2.0.2 SBSP の問題

SBSP では，出力値の特性は移動位置決定アルゴリズムで使用される関数に大きく依存している．前節に示したように，SBSP の基本アルゴリズムでは単純な線形関数が使用されており，単一のパラメータセットで実現できる移動特性は非常に単純なものになってしまう．移動特性を大幅に変更するためには，異なるパラメータセットを用意しておき，状況に応じて切替えて使用しなればならない．そのため，パラメータセット間の整合性の調整や，パラメータセットを切替える条件を管理するなどのコストが発生し，利便性は低い．

2.0.3 既存の関数近似モデルの利用

我々の以前の研究において，SBSP で使用される関数を，線形関数から非線型関数へ変更し，既存の関数近似モデルによって獲得させることを試みた[7]．関数近似モデルとして，シグモイド関数を発火関数として持つ 3 層パーセプトロンを使用し，GUI のツールによって人間がサンプルを与える教師あり学習によって近似関数を獲得させた．しかしながら，サンプルの再現精度はあまり高くなく，教示者の思いどおりの配置を実現することは難しかった．より高い近似精度を得るためには，局所的な学習が可能な関数近似モデルの採用が考えられる．

局所的な学習が可能な関数近似モデルの代表的なものとして，動径基底関数 (Radial Basis Funciton : RBF) ネットワーク[5]や正規化ガウス関数ネットワーク (Normalized Gaussian network : NGnet)[6]などがある．しかし，これらのモデルは学習させるパラメータの数が 3 層パーセプトロンよりも多く，サンプルの数が増えるに従って学習にかかるコストが大きくなるという欠点を持つ．また，3 層パーセプトロンと同様に，教示者の意図どおりの学習結果が得られるとは限らず，得られる近似関数がブラックボックス化してしまうという問題も残される．

本稿では，これらの問題を解決し，高い近似精度を実現するモデルを提案する．

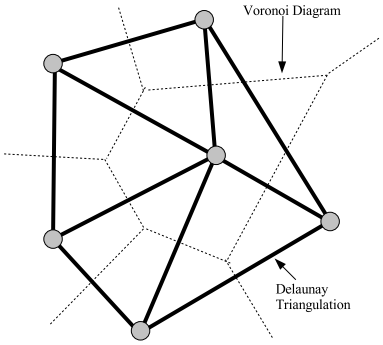


Figure 2: Delaunay 三角形分割

3 Delaunay 三角形分割を利用したエージェント配置手法

教師データとして与えられた各サンプルが局所的に作用し，他のサンプルへの影響を最少限に抑えるには，与えられたサンプルに基づいて空間を分割した上で，各サンプルの影響範囲をその分割領域に限定する，という方法が考えられる．そこで，我々は，対象となる空間を三角形分割し，三角形の各頂点で得られる出力値の線形補間を行う，という手法を提案する．本稿では，三角形分割の手法として Deluanay 三角形分割を，線形補間の手法として単純な内挿法を用いる．

3.1 Delaunay 三角形分割

Delaunay 三角形分割とは，” 平面上の点集合 P の三角形分割 T を構成した場合に，T に含まれる任意の三角形の外接円がその内部に P の点を含まない” 分割のことで，各三角形の最小の内角を最大にする（すなわち，三角形をなるべく細長くしない）という特徴を持つ．図 2 に Delaunay 三角形分割の例を示す．図中には，Delaunay 三角形分割と双対な関係にある Voronoi 図も描かれている．

与えられた点集合の要素数が 3 以上の場合，Delaunay 三角形分割はその点集合に対して一意に求めることができる．Delaunay 三角形分割を計算機上で求めるアルゴリズムはいくつか知られており，最も高速なアルゴリズムの計算量は $O(n \log n)$ である．よって，数百個程度の数の点集合であれば，リアルタイムでの三角形分割の導出が可能である．本稿で用いるプログラムは，もっとも高速なアルゴリズムのひとつとして知られる確率的逐次添加法[2]を用いて実装した．

3.2 線形補間アルゴリズム

線形補間手法としては，実行速度を重視して，単純な 3 点の内挿法を用いる．これは，3 次元コンピュータグラフィックにおける陰影付け手法のひとつであるグローシェーディングアルゴリズム[8]と同じ計算方法である．

図 3 に，グローシェーディングアルゴリズムの計算過程

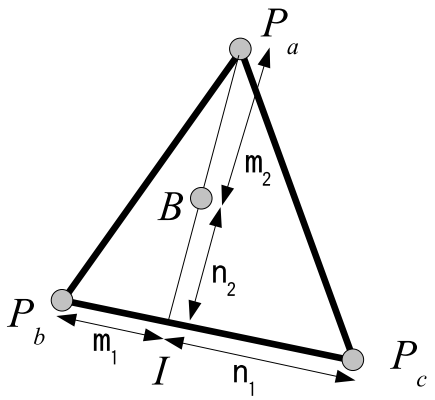


Figure 3: グローシェーディングアルゴリズムによる線形補間

を示す．三角形の頂点 P_a,P_b,P_c から得られる出力をそれぞれ $O(P_a)$,$O(P_b)$,$O(P_c)$ とすると，三角形の内部に含まれる点 B における出力 $O(B)$ は以下の手順で求められる．

1. P_a と B を通る直線と線分 P_bP_c との交点 I を求める．
2. $|\overrightarrow{P_bI}| = m_1$，$|\overrightarrow{P_cI}| = n_1$ とすると，I における出力値 $O(I)$ は，

$$O(I) = O(P_b) + (O(P_c) - O(P_b))\frac{m_1}{m_1 + n_1}$$

3. $|\overrightarrow{P_aB}| = m_2$，$|\overrightarrow{BI}| = n_2$ とすると，

$$O(B) = O(P_a) + (O(I) - O(P_a))\frac{m_2}{m_2 + n_2}$$

3.3 サンプルからのエージェント移動位置の導出

本稿では，教示者が与えるサンプルのボール位置を Delaunay 三角形分割の頂点として扱う．各頂点には，そのボール位置に対してプレイヤが移動すべき位置座標が保持される．ボールがある三角形に含まれたとき，前節で述べた線形補間アルゴリズムによって，プレイヤが移動すべき位置座標が 3 頂点の出力値から補間されて出力される．

3.4 提案手法の特徴

本稿で提案する手法は，以下のような利点を持つ

- 既存の非線形関数近似モデルと同等以上の近似精度を実現．与えたサンプルの入出力関係が完全に保証される．
- 非常にシンプルで高速に動作．与えるサンプルの数が数百個程度であれば，実時間での使用が可能．
- サンプルの追加や修正が行われても，影響が局所的である．追加・修正されたサンプルが属さない三角形領域には全く影響を及ぼさない．
- 獲得された性質を人間が視覚的に把握することが可能で，与えたサンプルが結果にどのような影響を与えているかを容易に推定することができる。
- 高い柔軟性．より滑らかな移動曲線や急激な勾配を持つ移動曲線が必要であれば，サンプルをより多く与えて移動位置を細かく指定するだけで良い．逆に，現在の配置が満足いくものであれば，サンプルの密度を高める必要は無い．
- 高いスケーラビリティ．対象となる空間が拡大・縮小しても，既存のデータに変更を加えることなく対応が可能．
- 完全な再現性．サンプルセットに含まれるデータが全て同一のものであれば，サンプルが与えられた順序に関係無く，完全に同一の結果を得られる．

特に，完全な再現性は非常に重要な利点である．サンプルの追加順序に制約が無くなることで，任意のサンプルを任意のタイミングで変更することが可能となる．これは，既に獲得された配置に対して，任意のタイミングで人間が介入できることを意味する．

4 シミュレーション実験

提案手法を用いてシミュレーションサッカーチームの配置を実際に形成し，RCSS 上で試合を実行してパフォーマンスを評価する．比較対象として，NGnet による関数近似モデルを使用した同様の配置モデルを実装した．提案手法と NGnet には，それぞれ同一のサンプルが与えられ，得られた配置を用いて，同一の対戦相手と試合を実行する．

4.1 NGnet

NGnet は RBF ネットワークの派生手法であり，RBF ネットワークにおける各基底の出力を，全基底の出力の和によって正規化する点が異なる．RBF ネットワークでは，基底間の距離が大きい場合に基底の中間位置付近で出力が 0 に近付いてしまう問題が発生するが，NGnet では正規化によって基底間で必ず補間が行われる．エージェントの配置問題においては，出力が 0 になることは望ましくないため，本稿では RBF ネットワークではなく NGnet を採用した．

x を入力とすると，NGnet の出力層のユニット i からの出力は，以下の式で与えられる．

$$f_i(x, w) = \frac{\sum_{j=1}^{N} w_{ij}\phi_j(x)}{\sum_{j=1}^{N} \phi_j(x)} \quad (1)$$

w_{ij} は中間層から出力層への結合荷重である．$\phi_j(x)$ は中間層の各基底ユニットからの出力であり，以下の式で与えられる．

$$\phi_j(x) = \exp(-\frac{||x - c_j||^2}{2\sigma_j^2}) \quad (2)$$

ここで，c_j は各基底の中心位置，σ は基底の分散パラメータである．

結合荷重 w_i は，以下の更新式による再急降下法で学習させる．

$$w(t+1) = w(t) - \eta\frac{\partial\epsilon}{\partial w} + \alpha(w(t) - w(t-1)) \quad (3)$$

学習率 η を 0.1，修正モーメント法の係数 α を 0.5 とした．基底の中心位置 c_j には，サンプルとして与えたボールの位置座標をそのまま使用する．すなわち，サンプルと基底の数は同一となる．基底の分散 σ には，以下の式で得られるヒューリスティックな値を使用する．

$$\sigma = \frac{1}{N}\sum_{i=1}^{N}||c_i - c_j|| \quad (4)$$

ここで，c_j は c_i の最近傍の基底を意味する．

4.2 FormationEditor

プレイヤの配置の形成においては，人間による俯瞰的な視点からの観察，そしてその観察に基づく直感的な調整が必要である．そこで，我々は，サンプルの編集作業を効率化可能にする GUI ツール，FormationEditor を実装した (図 4)．このツールを使うことで，配置の形成過程を視覚化しつつ，教師データとなるサンプルを編集することができる．

4.3 実験設定

サンプルセットととして 3 つのパターンを用意する．付録 A の図 5 から図 7 にサンプルの分布を示す．それぞれ，図中の×印の位置がサンプルのボール位置を表す．

まず，通常の試合で使用できる程度にまで作り込んだサンプルセットを用意した (図 5)[1]．このサンプルセットでは，フィールド全域に渡ってサンプルが与えられており，両ペナルティエリア内の密度がやや高めになっている．

２つ目のサンプルセットは，1 つ目のサンプルセットから，フィールド四隅，フィールド中央，そしてペナルティエリア内に存在するサンプルのみを残し，他は削除する．(図 6)．このサンプルセットでは，ペナルティエリアとフィールド中盤との疎密の差が大きくなっている．

最後に，2 つ目のサンプルセットから更にサンプルを削除し，フィールド中央とフィールド四隅にのみサンプルを残した最少のサンプルセットを用意する．(図 7)．

これらを教師データとして用いて，提案手法と NGnet のそれぞれにプレイヤの配置を獲得させた．そして，獲得された配置を使って，特定の対戦相手との試合を実行する．実験用プレイヤプログラムとして，我々が開発した RCSS 用ベースチームである agent2d プログラムを使用する [9] [2]．対戦相手には，RoboCup2004 に参加したチームのひとつである UvA Trilearn(アムステルダム大学) [3] を使用した．

4.4 実験結果

各配置データに対して，1 ハーフ 3000 サイクルの試合を 50 回ずつ行った，付録 B の表に，ログファイルから抽出したデータを示す．抽出するデータは，プレイヤの配置の違いによる影響が出やすいであろう指標として，得失点，パスの成功回数，パスカットの回数とした．

表 2 の平均失点に注目すると，偏りのあるサンプルセットの場合に，提案手法は NGnet の 3 分の 1 近くまで失点が抑えらていることが分かる．逆に，得点は非常にわずかだが NGnet の方が多い．次に，表 3 のパスの成功回数に注目すると，提案手法のほうがパス成功回数が多い．特に，偏りありのサンプルセットの NGnet の配置において，もっともパス成功回数が少なくなった．

4.5 考察

実験結果には，偏りのあるサンプルセットを用いて獲得した配置にもっとも大きな違いが現れた．これは，偏りのあるサンプルセットを用いて獲得した配置の特徴に原因があると推察できる．偏りのあるサンプルセットを用いた場合，NGnet で獲得される配置では，ボールがフィールド中盤にある状況では全プレイヤがほとんど静止している．ボールが両ゴールラインへ近付くと，全プレイヤが急激な移動を行い，ペナルティエリア周辺では提案手法とあまり差の無い配置となる．逆に，ゴールライン付近からフィールド中央へボールが戻る際にも，プレイヤの動き出しは鈍い．これに対して，全データの場合と最少データの場合では，いずれもフィールド全域においてボール位置に合わせたプレイヤの移動が確認できる．

NGnet での得点が僅かながらも多い理由は，一旦敵ペナルティエリア付近までボールを持ち込めれば，そのままプレイヤが前方に集まった状態が維持され，得点の機会が増えたためであろうと予想される．逆に，敵に攻め込まれた場合は，フィールド中盤から守備に戻る移動の開始が遅いために，守備が間に合わずに失点するのではないかと予想される．

パス成功回数についても，NGnet で回数が減った原因は，中盤でのプレイヤの移動量の減少が原因であろう．味方プレイヤがボールを持っていても，その動きに合わせて移動しようとしないために，パスコースが少なくなってしまったと予想される．

これらの結果から，NGnet では，対象領域内でのサンプルの密度の差が配置の特徴に大きく影響を与え，結果として，チームのパフォーマンスにも影響を与えやすい

[1] このデータは，RoboCup2006 で 4 位になった TokyoTechSFC が使用したものとほぼ同一である．

[2] http://sourceforge.jp/projects/rctools/ より入手可能．

[3] http://www.science.uva.nl/~jellekok/robocup/ より入手可能．

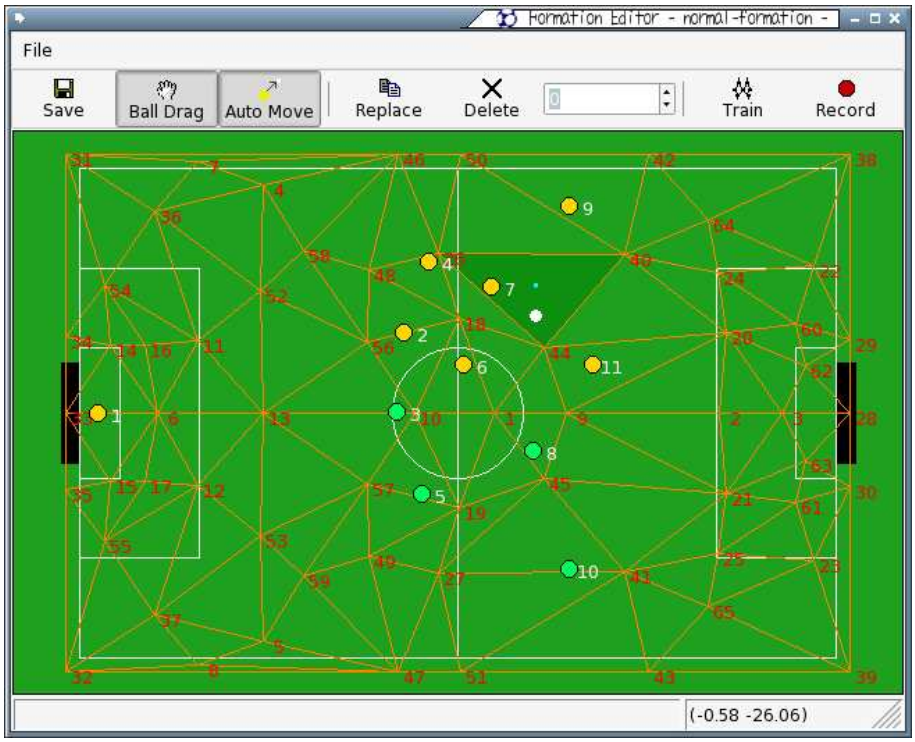


Figure 4: FormationEditor の実行画面

ことが予想される．すなわち，NGnet では，フィールドに対してなるべく一様にサンプルを用意しなければ充分なパフォーマンスを得ることが難しく，言い替えれば，各基底の分散のパラメータの調整が難しいことを意味する．これに対して，提案手法では，サンプルの密度の差に関係無く配置の特徴が安定している．

5 まとめ

本稿では，移動する注目対象に対して複数のエージェントの配置を効率的に獲得させる手法として，Delaunay 三角形分割と線形補間を組み合わせた関数近似モデルを提案し，RCSS を用いた実験によってその効果を示した．

実験で測った指標においてはあまり大きな差が見られなかったが，提案手法は多くの利点を持ち合わせており，非常に扱やすいモデルであるため，今後も有望な手法であると我々は考えている．しかしながら，提案手法には以下の欠点も存在する．

- メモリを大量に消費．NGnet のような関数近似モデルであれば，ネットワークの結合荷重などのパラメータに教師データが吸収されてしまうが，提案手法では教師データを全て保持しておかなければならない．
- ひとつのデータへの修正が，近傍のデータへの修正を要求されることがあるため，教師データの数が増えると，それらの整合性を保つためのコストが高くなる．

今後は，特にサンプルの維持管理コストを削減するために，サンプルの調整の自動化や，不整合なサンプルの検出を支援する手法の開発を試みる予定である．

また，多次元入出力への対応も今後の課題として重要である．現在は入出力がいずれも二次元であるが，実際には入力となる状態空間は無数に近い次元数を有している．出力に関しても，エージェントの移動以外の意思決定に関わる出力を得られるようになれば，より有益である．次元が増えると情報の可視化が困難になるため，上手く次元を圧縮したり，情報を重ね合わせるなどの工夫が必要になるであろう．

参考文献

[1] L. P. Reis et al. *Situation Based Strategic Positioning for Coordinating a Simulated RoboSoccer Team*, Balancing Reactivity and Social Deliberation in MAS, pp. 175–197, 2000.

[2] M. de Berg et al. 浅野哲夫 訳, コンピュータジオメトリ – 計算幾何学：アルゴリズムと応用, 近代科学社, 2000.

[3] The RoboCup Soccer Simulator, http://sserver.sourceforge.net/

[4] I. Noda et al., *Soccer Server and Researches on Multi-Agent Systems*, Proceedings of IROS-96 Workshop on RoboCup, 1996.

[5] T. Poggio et al., *Networks for approximation and learning*, Proceedings of the IEEE, 78, pp. 1481–1497, 1990.

[6] J. Moody et al., *Fast learning in networks of locally-tunned processing units*, Neural Computation, 1, pp. 281–294, 1989.

[7] H. Akiyama et al., *Team Formation Construction Using a GUI Tool in the RoboCup Soccer Simulation*, Proceedings of SCIS & ISIS 2006, 2006.

[8] H. Gouraud, *Continuous shading of curved surfaces*, In Rosalee Wolfe(editor), Seminal Graphics: Pioneering efforts that shaped the field, ACM Press, 1998.

[9] 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.

A 実験で使用したサンプルセット

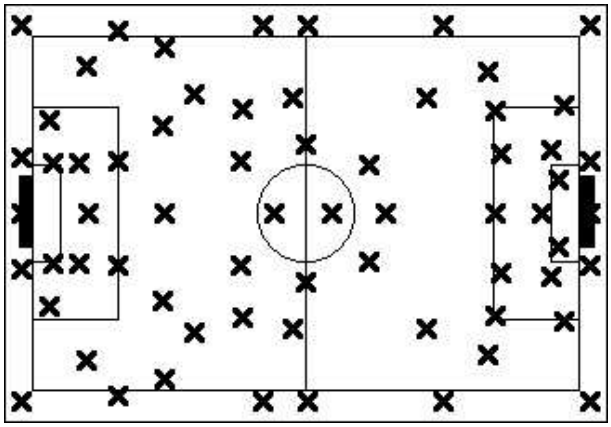


Figure 5: サンプルセット（全データ）

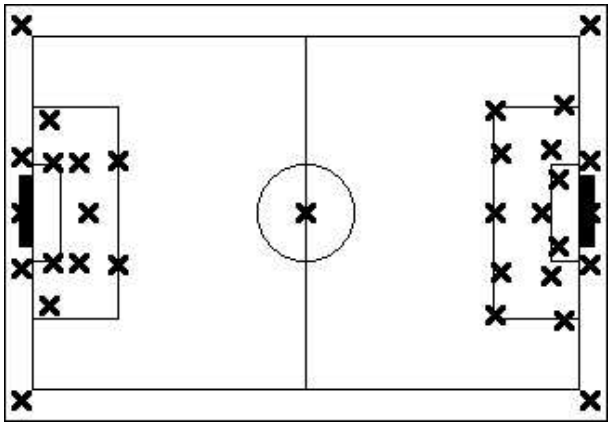


Figure 6: サンプルセット（偏りあり）

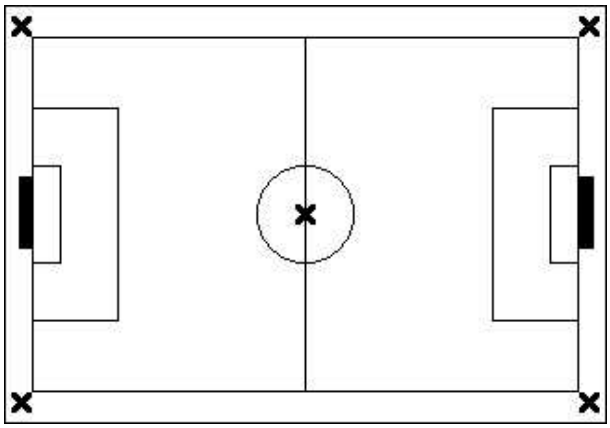


Figure 7: サンプルセット（最少）

B ログから抽出した統計データ

Table 1: 各サンプルセットでの平均得点

	平均得点（標準偏差）	
	NGnet	提案手法
全データ	0.2(0.4)	0.08(0.27)
偏りあり	0.24(0.52)	0.06(0.22)
最少	0.02(0.14)	0.02(0.14)

Table 2: 各サンプルセットでの平均失点

	平均失点（標準偏差）	
	NGnet	提案手法
全データ	0.98(0.89)	0.76(0.9)
偏りあり	1.76(1.24)	0.6(0.74)
最少	3.7(1.97)	3.16(1.67)

Table 3: 各サンプルセットでの平均パス成功回数

	平均パス成功回数（標準偏差）	
	NGnet	提案手法
全データ	99.06(19.08)	110.6(18.79)
偏りあり	67.68(15.07)	91.44(21.35)
最少	83.86(19.1)	84.3(18.48)

Table 4: 各サンプルセットでの平均パスカット回数

	平均パスカット回数（標準偏差）	
	NGnet	提案手法
全データ	20.2(5.06)	17.18(6.06)
偏りあり	18.8(5.52)	19.32(5.29)
最少	12.06(4.17)	14.5(5.21)

マルチエージェントシステムにおける 行動制御アーキテクチャの自己組織化

Self-Organization of Action Control Architecture for Multi-Agent System

○川上皓平, 杉山英輔 (慶應義塾大学大学院理工学研究科)
藤井飛光, 吉田和夫, 高橋正樹 (慶應義塾大学理工学部)
* Kohei KAWAKAMI, Eisuke SUGIYAMA (Graduate School, Keio Univ.),
Hikari FUJII, Kazuo YOSHIDA, Masaki TAKAHASHI (Keio Univ.)

*corre1@a5.keio.jp, sugiyama@yoshida.sd.keio.ac.jp
fujii@sd.keio.ac.jp, yoshida@sd.keio.ac.jp, takahashi@sd.keio.jp

Abstract-This paper deals with a design problem of action control architecture for cooperative action of multi-agent system in dynamic environment and presents a self-organization method based on autonomy and emergence. The architecture consists of a task selector, an action selector and action modules. These are developed by self-organization using an evolutionary computation method of GNP and GA. All agents in the team have similar action control architecture. This enables flexible cooperation in the agent's action based on the global simultaneous evaluation of adaptivity and necessity of the task. The method is applied to the system of the robots for the Middle Size League of RoboCup to verify the effectiveness of the method.

1 序論

近年，情報工学やロボティクス分野の急激な発展に伴い，エージェントを取り囲む環境が大規模になり，エージェントに対する要求も複雑化してきている．これらの問題をマルチエージェントによって協調的に解決する制御手法の研究が盛んに行われている [1]．従来研究では設計者が状況を想定して予めタスクを設定し，そのタスクをいかにエージェントに振り分け解決するのかという役割分担問題に焦点が当てられており，設計者の意図しない環境の変化にエージェントを対応させることが困難であった．

本研究で提案する行動制御アーキテクチャの特徴は，エージェントの制御器を設計者が設計するのではなく，進化論的計算手法を用いて自律性と創発性を持つエージェントが自己組織化して構築するところにある．エージェントが自らタスクを抽出し，選択し，行動することで，設計者が予めタスクや詳細な規則を与えなくても，様々な状況において目標を達成できる柔軟な協調制御が可能となる．

行動制御アーキテクチャの設計手法は大きく分けてトップダウンとボトムアップの 2 手法がある．前者は設計意図を明確に組み込むことができ，後者は創発性を持つエージェントが環境に応じたシステムを自律的に構築することができる．

本研究では両者の特性を活かし，設計意図を取り入れて進化論的計算手法により学習することで準最適なシステムを獲得する手法の構築を図る．具体的には行動制御アーキテクチャを複数の機構に分け，各機構を自己組織化領域，人工設計領域に区切ることができるシステムを設計する．

また，マルチエージェントシステムにおいて，エージェントの動機に基づいたタスク分担をすることで効果的な協調行動を実現する．

本研究は，RoboCup 中型リーグ用のサッカーエージェントを題材とする．複数の未知なタスクが混在する動的環境下においても有効に機能する行動制御アーキテクチャを構築し，その性能を検証する．

2 マルチエージェントシステム

2.1 エージェント間のタスク分担

エージェントの環境に応じたタスク選択はエージェント間の役割の重複を避け，効果的な協調を実現する．マルチエージェントシステムにおけるタスク分担の研究は以下に示すように多くなされている．

情報を一箇所に集めてからタスクを振り分ける AUCTION [2] はそれぞれエージェントのタスクに対する適応度を集めた上で優先度の高いタスクを適応度の高い順に振り分けていく．これは常に他のすべてのエージェントとの相対評価を必要とする．

Parker らが提案した ALLIANCE [3] では各エージェントの持つ impatience と acquiescence という 2 種の動機をもとにチーム内でタスクを動的に振り分ける．他のエージェントを補うようにタスクを分担するため，堅牢な協調制御を実現している．

また，感情によるタスク選択手法は昆虫など生物

の集団行動における役割選択モデルを用いる例が挙げられる．文献 [4] では人の感情の「快-不快」に着目し，好意を持つエージェントに接近し，嫌悪されているエージェントからは離れるという作用によって障害物回避などの行動制御を実現している．

2.2 提案する協調制御手法

マルチエージェントシステムにおいて，環境の変化に柔軟に対応できる協調行動の実現を図る．情報を集中的に管理することなく，タスクのみ共有することでエージェントが所属するチーム全体の協調を行う．藤井，加藤ら [5] は設計者が設定した目標に対する各エージェントの達成度を評価して，チームにおける自身の役割を決定する手法を実現した．これは抽象化された情報を共有することで協調している．

本研究ではエージェントが環境におけるタスク適応度を，チームにおけるタスク必要度と比較することで自身の実行すべきタスクを選択する．タスクと抽象化された評価情報を共有することでチーム全体の目標を達成する．また設計者の意図しない環境の変化に適応するため，タスクは設計者が設定せず，エージェントが環境からタスクを抽出し，タスクを解決する行動系列を獲得する自己組織化を行う．環境に応じた自発的な協調行動を実現するために，行動制御アーキテクチャの機構を分割し，学習を取り入れた分散協調制御のためのシステム設計を行う．

3 行動制御アーキテクチャ

3.1 システムの概要

Figure 1 に本研究の提案する行動制御アーキテクチャの構成を示す．行動決定を行いアクチュエータへの指令値を決定する部分は，環境から情報を取得し，それをもとに自身のタスクを決定し，タスク内の行動選択器において実行すべき行動モジュールを決定する．エージェントは協調のため共通のアーキテクチャを持つ．次節より各機構について説明する．

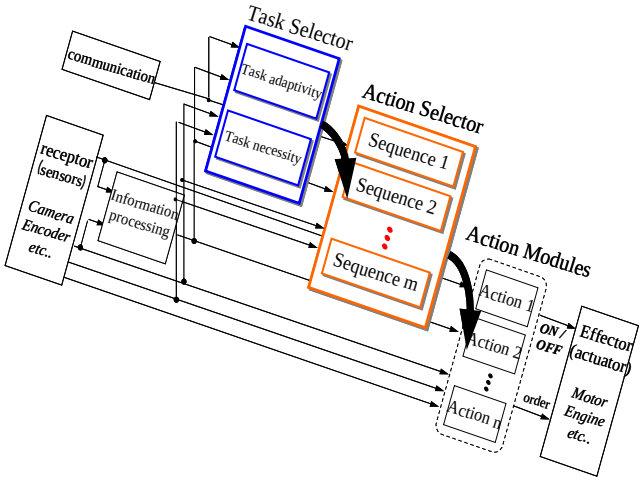


Figure 1: An action controller

3.2 行動モジュール

行動モジュールは実際に行動するために必要な指令値を計算し，アクチュエータへ指令を与える機構である．単純かつ連続性のある行動をモジュール化した．また行動モジュールが実行可能であるかを判定するための判定モジュールを設計した．

3.3 行動選択器

状況に応じて行動モジュールを選択する機構である．3.2 節の判定モジュールと行動モジュールを遺伝的ネットワークプログラミング(GNP:Genetic Network Programming)[6]によりネットワーク状に繋ぎ，行動系列を構築した．概念図を Figure 2 に示す．菱形は判定モジュール，四角は行動モジュール，矢印は行動選択の流れを意味する．

3.4 タスク選択器

タスク選択器は，状況に応じて行動選択器内の実行すべき行動系列を決定する機構である．タスク選択器は，タスク適応度算出器とタスク必要度算出器からなる．タスク適応度とは，環境におけるエージェントのタスクに対する評価値であり，タスク実行のためのエージェントの評価指標といえる．適応度はそれぞれエージェントを中心とした局所的な環境情報から算出される．

タスク必要度は，チームがタスク実行を必要とする程度を表す評価指標といえる．これはタスクの解決に必要となるエージェントの適応度の合計値である．必要度はチーム全体における大局的な環境情報から算出される．

タスク選択は，自身より高いタスク適応度をもつエージェントから優先的にタスクを選択し，自エージェントは最後に必要度が満たされていないタスクを選択する．但し，エージェントのタスク適応度が

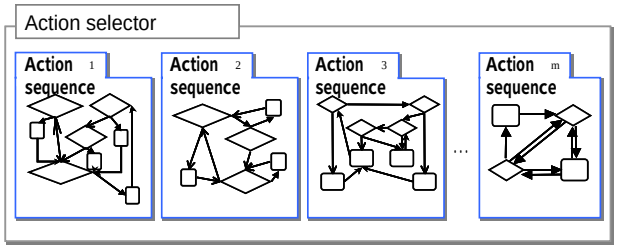


Figure 2: An aciton selector

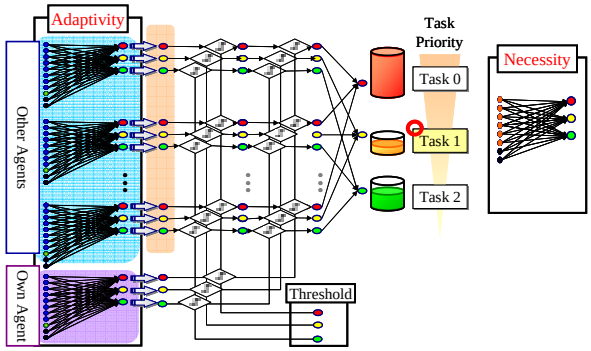


Figure 3: A task selector

閾値以下もしくは必要度が満たされていれば，そのタスクは選択されず，次に優先度の高いタスクに対して同様の処理を行う．タスク番号の小さいものほど優先度が高い．Figure 3 はこの処理の流れを示す．箱の大きさがタスク必要度，各エージェントが持つ箱に入れる中身がタスク適応度を表している．この図では，TASK1 の必要度が他エージェントによって満たされているため，自エージェントは TASK2 を実行する．各エージェントは実行タスクを伝達するのではなく，あくまで抽象化されたタスク毎のタスク適応度を通信によって共有する．

タスク必要度を算出する際の環境情報に対する重み，およびタスクの閾値を遺伝的アルゴリズム(GA:Genetic Algorism)を用いた学習により獲得する．

3.5 タスクの抽出

これまでタスクを評価し選択するためのタスク選択器と，実際にタスクを解決する行動選択器の構築について述べてきた．これらの制御器は，タスクを介して関連している．タスクは，タスク選択器と行動選択器の設計の指標となる．本研究では，タスク選択器と行動選択器の構築を同時に行うことで，環境に存在するタスクそのものを構築しようと試みる．具体的には，各制御器の学習を Figure 4 のようにタスクの性質と解決方法を関連付けて同時学習することにより，設計者に依存せず環境からエージェントに必要なタスクを自ら抽出する．

また，すべてを学習させるのではなく，一部の機構を人工的に設計することで，設計意図を組み込んだ学習を行うことができる．このため，学習の効率化と設計方針を明確にすることができる．

4 実験

4.1 実験環境

本研究では，RoboCup 中型リーグの環境を模擬したクライアント-サーバ方式のシミュレータを用いた．これはプログラムを継承することで実機と同様の動作を実現できる．ボールの反発や摩擦，認識誤差などを再現し，また学習のため実時間よりも高速な動作ができる．Figure 5 に外観を示す．

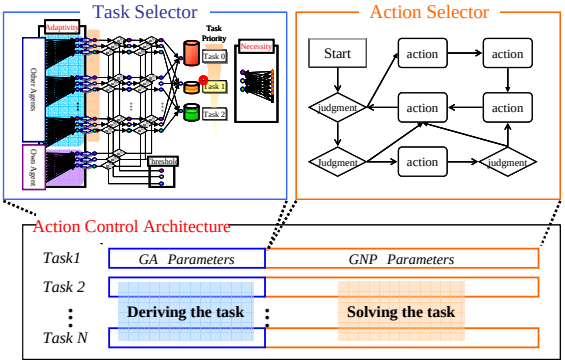


Figure 4: Simultaneous learning

4.2 RoboCup への適用

3 章で述べた行動制御アーキテクチャを RoboCup 中型リーグのロボットに適用した．シュートやパスなどの行動をそれぞれ行動モジュールとし，9 種の判定モジュールと 10 種の行動モジュールの組みあわせを GNP により学習した．エージェントのタスク適応度はボールとの位置関係など自身の環境情報から求め，タスク必要度は得失点差など大局的な情報から求める．この環境情報の重みを GA により学習した．

学習の適応度はゲームで勝つことをチームの大目標としているため，得失点差，ボール支配率，ボールの敵陣存在率を増やし，衝突回数を減らすよう設計した．これを Figure 6 に示す．

4.3 シミュレーション

4.1 節のシミュレータ上で，学習基準となる人工的に行動制御アーキテクチャを設計した相手(2006 年 EIGEN KEIO UNIVERSITY)と試合をするシミュレーションを行った．学習後の個体とは 200 世代学習したものである．世代交代モデルは ER(Elite Recombination)を用いた．次の 3 条件において実験を行った．

(1) タスク選択器を人工的に設計し，行動選択器の行動系列を GNP により学習した．
(2) 設計意図を組み込んだ行動選択器を使用して，タスク選択器の重みを GA で学習した．
(3) 設計意図を与えず，行動選択器とタスク選択器を同時に学習を行った．

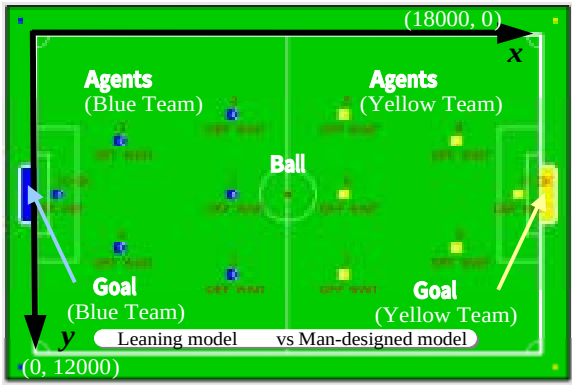


Figure 5: 2D simulator

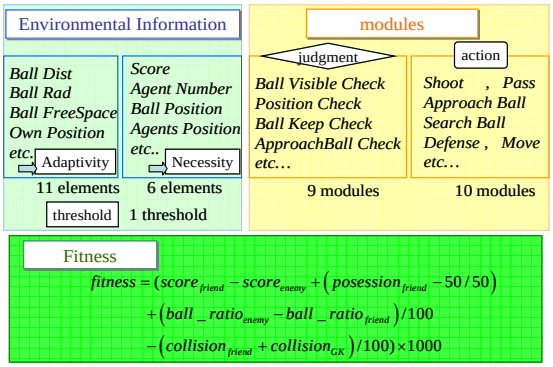


Figure 6: Learning elements

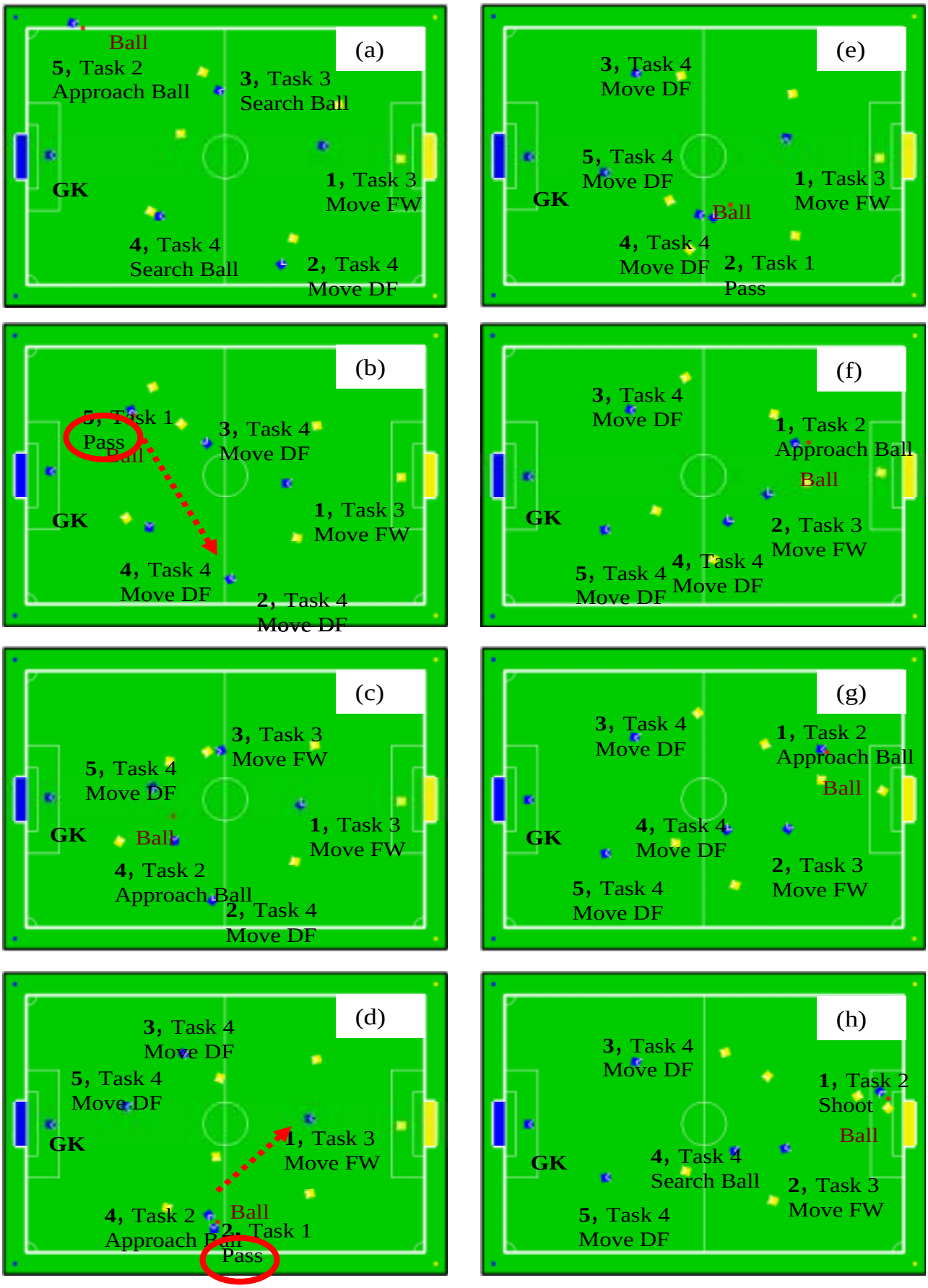


Figure 8: A result of learning with design intention (Blue: learning with pass task)

Figure 9: A result of simultaneous learning (Blue: learning team)

4.4 実験結果

2006 年 EIGEN の行動選択器の一部と, 学習後の(1)の結果を比較したものが Figure 7 である．優先度の最も高いタスクにパスを組み込んだ(2)の学習結果の行動を Figure 8 とする．また(3)の学習結果の行動を Figure 9 に示す．青いエージェント群が学習群である．

4.5 考察

Figure 7 では 7 個の行動モジュールと 6 個の判定モジュールを持つ行動選択器が，学習により各 2 個の行動，判定モジュールで形成された．有効なモジュールだけを用いることで 1/3 ほど大幅にコンパクトで環境に応じた行動選択器が構築できている．

Figure 8 では，1 体がボールにアプローチしている間は他エージェントが補助するようにポジショニングしている．また(b)や(d)を見ると，パスタスクを実行して，味方にパスをすることでボールを敵陣へ運んでいることがわかり，タスクの切り替えが有効に働いているといえる．特に(d)ではゴール前に待機している No.1 のエージェントにボールを渡すことで，相手の守備が整う前に攻めることができ，シュートまで持ち込んでいる．効果的なパスは自群にとって有利なゲーム展開を実現することが確認できる．

Figure 9 は行動選択とタスク選択の両方を学習させた結果である．ボールを保持するタスクとして TASK1 が, TASK2 と 3 がボールを持たないエージェントが待機するタスクとして創発された．まずボールに近いエージェントが最も優先度の高い TASK 1 を実行し，ボールへのアプローチをしている．次に自陣にいるエージェントは TASK2 を実行し，DF 位置につき，そのほかのエージェントは FW に位置取りしている．局所的にエージェントが集中することなく，フィールド全体を使ったポジショニングが実現できている．また(a),(c),(d)など相手側がボールを保持している場合は複数がボールへアプローチすることでボールを獲得している．このため，(c)から(d)のように相手に攻め込まれた場合は迅速な守備ができている．エージェントが協調することで(d)から(e)のように速やかに守備から攻撃へ切り替えることができ，効果的なゲーム展開が実現できている．

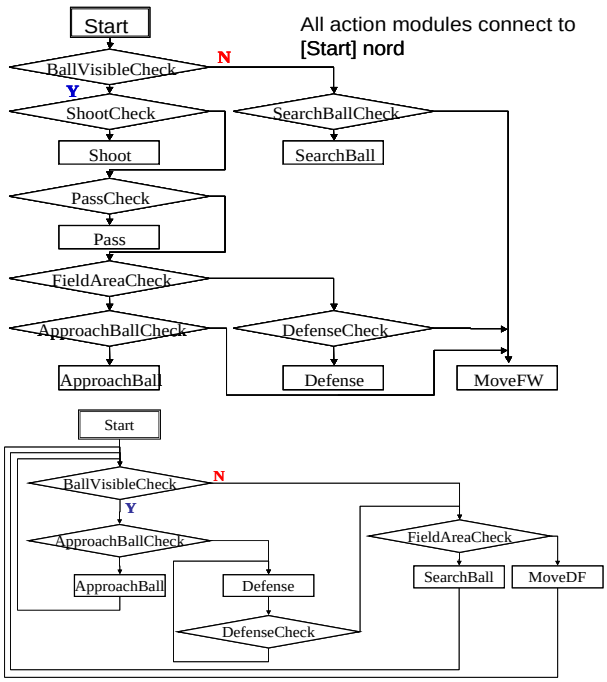


Figure 7: An action selector of man-designed (top) and a learning result (bottom)

5 結論

本研究で提案した行動制御アーキテクチャの自己組織化により，動的環境において，自らタスクを抽出し，動的に切り替えることで協調行動を実現した．本研究は，エージェントの動作環境における大目標を学習の適応度とすることで行動制御アーキテクチャが自己組織化できることを示した．また各エージェントが局所的に得られる抽象的な情報を共有することで動的にタスク決定が行われ，効果的な協調行動を達成できることを示した．

行動制御アーキテクチャを機構ごとに分割して設計することで，環境における進化と設計意図をともに組み込んだシステムが生成された．激しく変化する環境に適応した有効な協調行動の実現を示した．

謝辞

本研究は，文部科学省平成 15 年度 21 世紀 COE プログラム【知能化から生命化へのシステムデザイン】によるものであることを記し，謝意を示す．

参考文献

1. A.Gage, R.Murphy, K.Valanis and M.Long, Affective task allocation for distributed multi-robot teams.
2. Y.Kashimura, A.Ueno and S.Tatsumi, A Distributed Coordination Method for Dynamic Role Assignment in Multi-Robot System, the 20th Annual Conference of the Japanese Society for Artificial Intelligence, 2006
3. L. Parker, ALLIANCE: An Architecture for Fault Tolerant Multirobot Cooperation, IEEE Transaction on Robotics and Automation, vol.14, no.2, pp.220-240, 1998.
4. T.Kusano, A.Nozawa and H.Ide, Emergent of Burden Sharing of Robots with Emotion Model, IEEJTransaction EIS, vol.125,No.7,pp1037-1042, 2005
5. H.Fujii, M.Kato and K.Yoshida, Cooperative Action Control based on Evaluating Objective Achievements, RoboCup International Symposium 2005, 208-218, 2005
6. H.Katagiri, K.Hirasawa and J. Hu, Genetic Network Programming-Application to Intelligent Agents-, Systems, Man, and Cybernetics, 2000 IEEE International Conference on, vol5,3829-3834, 2000

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-A701-8 (5/4)

Sim-3D リーグ Humanoid ロボットに向けたこのへんファジィ制御の提案
KONOHEN-fuzzy Control for Sim-3D league Humanoid robots

西野順二
Junji NISHINO
電気通信大学
Dept. of systems engineering,
The University of Electro-Communications
nishino@se.uec.ac.jp

Abstract

This paper presents KONOHEN-FUZZY reasoning system control method for humanoid robots that have many and redundant joints. We show a simple control exaple model on dance task with humanoid. The task is modeld on torus manifold and 2-dimensional fuzzy sets handle the critical conditions.

1 ヒューマノイド化したシミュレーションリーグのサッカーモデル

RoboCup サッカーシミュレーションリーグ[RCS]の 3D 部門は、2007 年から、ヒューマノイドのシミュレーションモデルに移行することとなった。ヒューマノイドロボットの特徴として、歩行などの目的タスクに比して冗長な多数の関節を持つ点がある。このため、従来のシミュレーション 2D リーグが戦略的な高次判断が研究の主眼であったのに対し、新たな 3D リーグにおいては、ロボットの冗長な自由度の利用と制御法を検討する必要が生じる。

本研究は、ヒューマノイドロボットのような冗長自由度をもつシステムの制御に「このへんファジィ」推論[西野 06]を応用することを目的とする。

多数の回転リンクによって構成されるシステムの状態空間は位相構造もユークリッド空間とは同相でない複雑な多様体となり、変数間の関係も非線形である。このため、多くの場合で連続な時不変フィードバック則を求めることもできず、そもそも数理的モデルを一般に求めることも難しい。このため、個々のロボットのコンフィギュレーションに応じて試行錯誤的に数理モデルを構成したり、小脳モデル等による逆動力学モデルの構成や、強化学習による状態空間の分割と制御など、学習機構によって制御則を得ることが行われてきた。

これらの手法による制御則構築、学習則の適用には、十分に精密なロボット動力学モデルの構築が欠かせない。しかし、RoboCup ヒューマノイドリーグで多用されているのは実機ベースで開発が進むホビー用サーボによるロボットであり、費用的にもハードウェア的にも制御則の構築に利用できる高精度モデルの準備が困難である。このため現在までのところ、固定点で表される姿勢を事前かつ試行錯誤的に複数指定し、それらのシーケンス遷移によって行動を規定している。これは一般にモーションコントロールと呼ばれている。

多くの場合、二つの登録姿勢間の補間はすべてのサーボについて平均的に、すなわち線形に補間されている。このため、補間された中間姿勢でときに不安定状態に陥り転倒することもしばしば発生する。そこで、操縦者の知識と熟練によって姿勢の組合せによっては遷移を規制したり、安定な中継点を試行錯誤的に発見し追加することで、転倒問題を回避している。シミュレータを用いるとしても、そもそも試行錯誤的に安定中継点を発見する必要があるのは、ロボットモデルの多変数性と位相構造の複雑さに起因する。

通常で１０自由度を超えるようなヒューマノイドロボットは、制御工学的には複雑なシステムであり、数理的な構造を求めることは難しい。こうしたシステムに対しては、動力学モデルや逆動力学モデルを直接設計できないため、数理モデルを元にした制御則の設計も困難である。そこで、人間や生物を高次に模倣した学習に基づくシステムが研究されている。状態空間を離散的に扱う方法は多変数システムでは組合せ爆発が発生するため、学習等に成功する事例は少ない。

本論文で適用を提案する「このへんファジィ制御」は、任意の構造を持つ多変数空間上で定義される自由形状の多次元ファジィ集合を用いた推論システムである[西野 06]。そこで、この特徴を用いて、非線形性や位相構造をファジィ集合それ自体で対応することによる制御を行い、冗

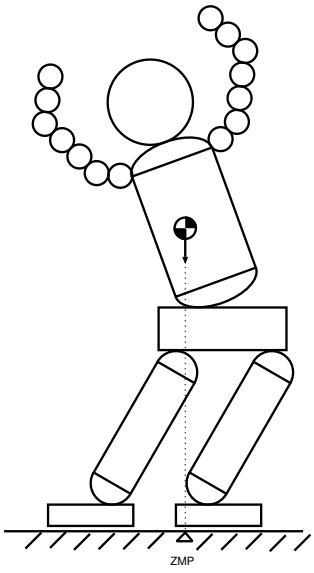


Figure 1: ロボットのダンスタスク

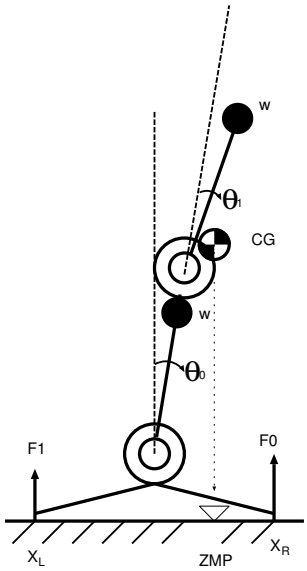


Figure 2: ２リンク表示したダンスロボットの機構モデル

長自由度を持ち、作業空間が複雑な位相構造を持つ制御対象に適用可能である。行動制御では次の式 (1) ような形式の規則を用いる。

$$R_i : if \text{ 作業状態 } is \text{ このへん } then \text{ 行動 } is \text{ 動作 } A \quad (1)$$

ユークリッド空間と同相でない多様体の作業空間あるいは状態空間に対し、このへんファジィの手法では曖昧な部分空間を定義して直接的に規則を構成する。多様体上の曖昧な部分空間はファジィ集合を用いて表現する。

本論文では多次元上のファジィ集合を用いることで、ヒューマノイドに代表される作業空間、状態空間がトーラスなど複雑な多様体構造を持つロボットの制御を行う方法について提案と考察を行う。

2　2リンクダンスモデルへの応用

本論文では、図 1 に示すヒューマノイドモデルを対象とし、このへんファジィによるダンス運動の制御を考える。

簡単のため両足は連動するものとし、足首床面接地部と腰の２関節のみが回転する２リンクシステムとして、図 2 のようにモデル化する。

アクチュエータの回転運動は十分にゆっくりであるとし、角速度 $\dot{\theta}_i$ および角運動量 $I\dot{\theta}_i i$ も十分に小さいものと仮定する。

$$\dot{\theta}_i << \epsilon \quad (2)$$

この仮定の下では ZMP の位置は重心の直下となるため、ロボットは ZMP が足裏内から出ると転倒し、ZMP が足裏内にある間は直立静止を保てるものとなる。

$$x_L < x_{CG} < x_R \quad (3)$$

十分に小さいながらも発散方向に働く角運動量があると仮定し、境界付近では転倒の危険性が発生する。

3　このへんファジィ制御による危険姿勢回避

ロボットの制御構造として、安定制御、行動制御、戦略制御の３階層のモデルを仮定する。本研究で示すこのへんファジィの適用は、中間階層の行動制御でもちいる。行動制御では、最下層において安定性を保証された基本的な運動制御モジュールを、状態認識にもとづいて選択し、サッカーであればドリブルやパスなどのまとまった行動を行うものである。運動制御では、立つ、歩く、走る、止まる、蹴るなどの基本運動ができるものとする。行動制御は式 (1) で構成され、結論部にこれらの運動モジュールを指定する。

このへんファジィを適用するに先立ち、まず静的な安定領域と危険領域および境界を調べてみる。θ i による状態空間上に表現すると図 3 のような形状となる。２リンクロボットを身体に見立てたダンス運動は、この空間上の軌跡となる。静的特性のみを考えると、図 3 のハッチで示した危険な状態に入らないように運動制御しなければならない。

こうしたモーション制御は通常、目的の姿勢ポイントを線形に連結して行われる。しかしながらθ i の空間をユークリッド空間とみなした単純な補間では、図 4 に示すように、ポイント間で安全領域を出てしまうことがしばしば発生する。

このような安全領域逸脱状態を回避するため、通常は図 5 の 2’、3’ のような安定領域でのポイントを増やすことで対応する。これらのポイントは実際のロボットの挙動を見ながら試行錯誤的に求められる。その際には試験的に追加、挙動テストの後、追加点の修正もしくは再追加とテストを繰り返さねばならない。

θ i で表現した作業空間は、回転特性上 $\theta_i = \pi = -\pi$

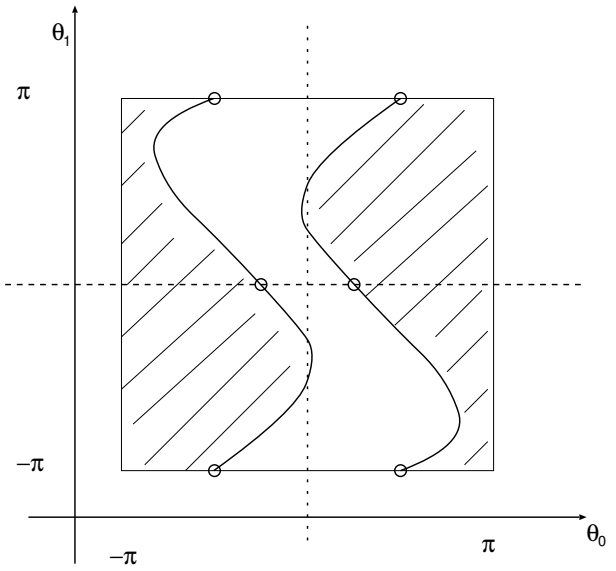


Figure 3: 関節角局所作業空間 (状態空間)

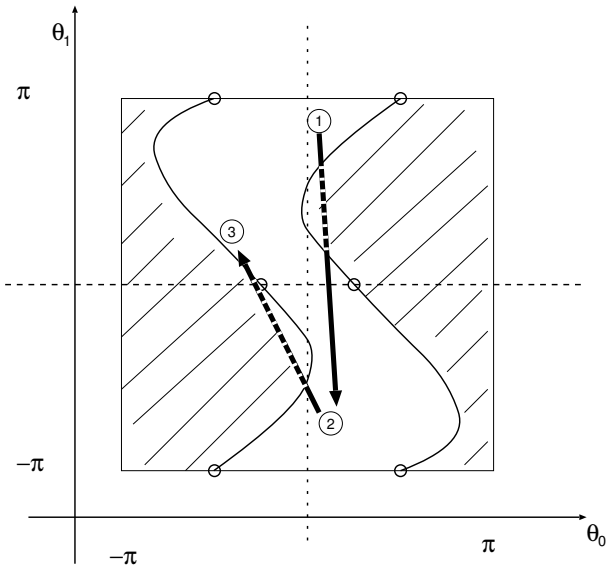


Figure 4: θ i空間で表示したダンス行動の軌跡

と連接している。このため実際には、θ i を局所座標とするトーラスの２次元多様体をなしており、図 6 のように、３次元空間に埋め込んで作業空間を表示することができる。ダンスの行動計画や軌道計画は、このトーラス状多様体上で行わなければならないものである。

モデルでは腰部における上半身の角度に制限を設けず回転できるものとしている。そこで、図 7 に示すような軌跡も動作の解として有効である。しかし、このような行動を実装するためには、設計者による行動の試行錯誤的「発見」が必要である。

このへんファジィ制御では、軌道に対して直接的な目標ポイントを追加するのではなく、並列推論を活かしてつねに安全領域におさまるような行動を追加する。図 8 に危

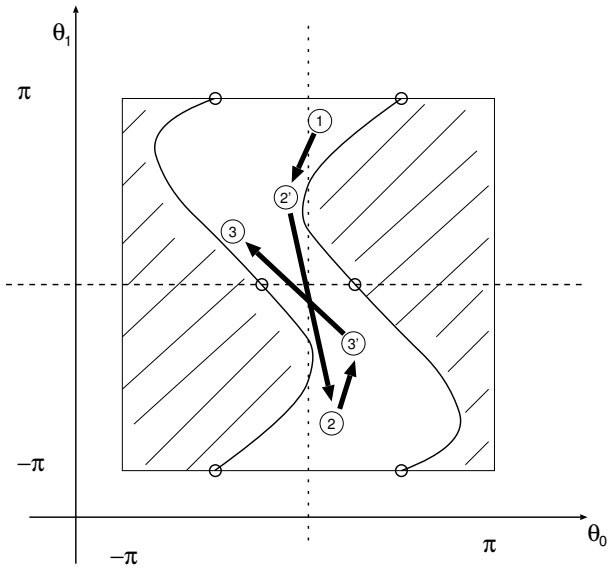


Figure 5: 中継ポイント追加による忌避領域を避けた軌道解の試行錯誤的生成

険領域との界面として、左に行き過ぎ (TooRight)、右に行き過ぎ (TooLeft)、という曖昧な状態を定義しした。このファジィ集合を用いて次のような規則を構成する。

R_L : $if\ (\theta_0, \theta_1)\ is\ TooLeft\ then$ 行動 is 足首 + 回転(4)

R_R : $if\ (\theta_0, \theta_1)\ is\ TooRight\ then$ 行動 is 足首 − 回転(5)

角運動量が微小のとき、足首の角度の加減により図 8 中の矢印で示した方向への軌跡の修正が行われる。危険領域境界という、非線形で形状的複雑性を持った制約を、ファジィ集合によって直接的にカバーしている。

この二つのこのへんファジィ制御則は、危険状態に近付いた場合には安全方向に姿勢を変化させる行動を取る、というメタ知識を表現したものになっている。危険状態という曖昧な観念を右と左の二つに大きく分けただけで、それぞれ一つの「こんな状態」を表す多次元上ファジィ集合で表現する。ダンス動作の基本遷移を線形補間規則で作り、式 (4) 式 (5) の R_L, R_R と並列に用いることで必要に応じて危険を回避する行動を加えることができ、安全な遷移を実現する。

4　ファジィ集合の構築コスト

図 8 に示したの TooLeft, TooRight という２次元空間上のファジィ集合は収集したサンプル点集合によって構成する。実機ロボットにあらかじめ危険な状態を複数与え、各状態量をサンプルする。収集したサンプル点を、このへんファジィ推論エンジンにファジィ集合として登録し利用する。本論文のような２次元作業空間多様体の場合、一つのファジィ集合を定義するために１０　１００点程度の登録で実用上十分である。このファジィ集合を構築する作

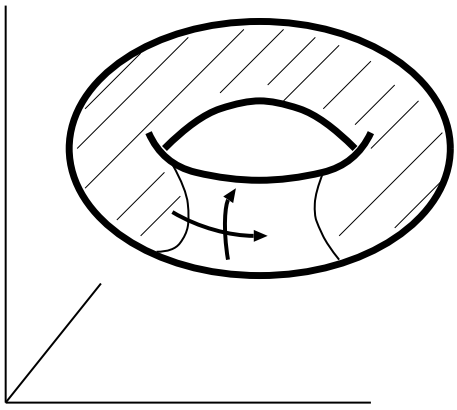


Figure 6: ３次元に埋め込んだトーラスをなす作業空間多様体

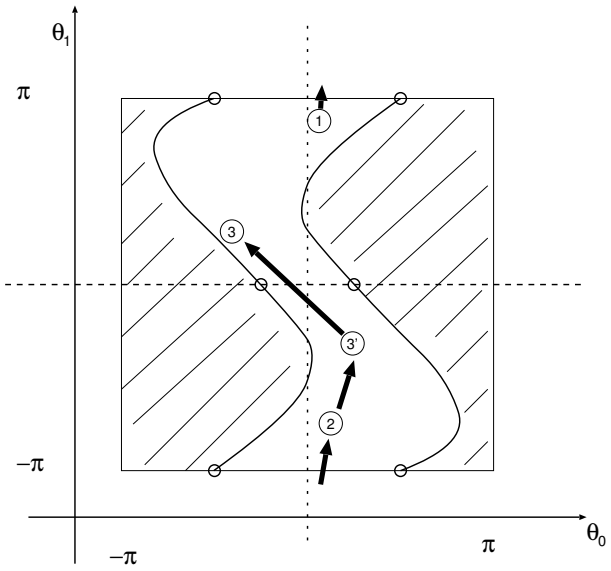


Figure 7: トーラス上を遷移する行動解

業の負荷は、ダンス軌道ごとにその近傍の安定ポイントを探す試行錯誤の負荷と同等程度であると言える。また、軌道生成を多数行う場合には、ポイント探索の試行錯誤が生成する軌道の数に比例して増加するのに比べ、ファジィ集合は一度構成すれば良いのでトータルコストは少ないと考えられる。

5　まとめと今後の課題

このへんファジィは人間が通常持っている直観的な概念をもとに、多次元状態空間に対する行動制御をそのまま表現できる仕組みである。ヒューマノイドロボットに代表される多関節冗長自由度システムの制御においても適用できることを２リンクモデルでのダンスタスクを対象として示した。

新しい形状のロボットを作ってゆくことは、機構的および動力学的な性質の分からない新たなシステムを作る

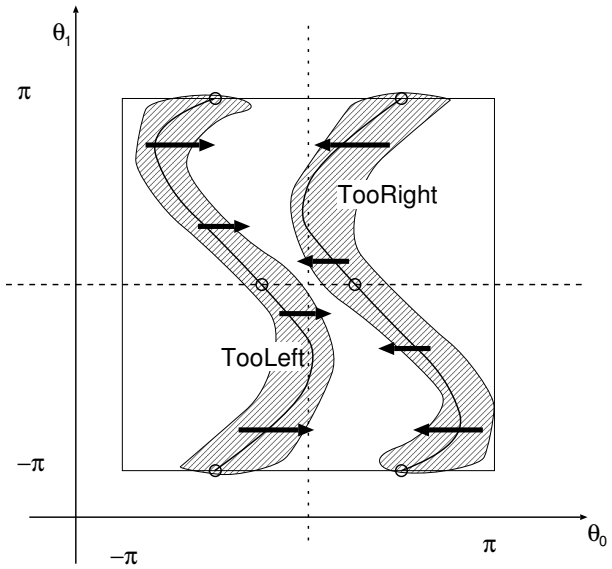


Figure 8: このへんファジィによる忌避状態の回避規則

ことである。このようなロボットシステムを、知識の無い状態から学習によってモデリングする手法に対し、本論文で提案した方法は実物に体する人間の直観的知識を用いる手法である。実際の、とくにホビー的分野でのヒューマノイドロボット制御でも、直観を使う方法と言える試行錯誤による軌道生成が行われている。このへんファジィを援用することで、実体の状態と直観的知識を制御規則として結び、知識による有効な制御が可能となる。新たなロボットシステムを作る上とき、知識無しからすべてを構築するのではなく、このへんファジィを用いた直観的知識と融合した行動則の設計が有効な場面が多数ありこのへんファジィが有効であると考える。

今後のサッカーシミュレーション３Ｄでは、ドリブルや走り込み、ストップなど、動的かつ高速なヒューマノイドモデルの行動制御が必要となる。これらの技術は実ロボットでもまだ完成してしていない。そこで、ヒューマノイドロボットの３Ｄシミュレーションモデルを対象に、直接的空間表現の困難な状態空間に対して、このへんファジィの適用例を示してゆくことが今後の課題である。

なお、本研究に関連するシステムの開発の一部は、独立行政法人情報処理推進機構 (IPA) より、2005 年度上期未踏ソフトウェア創造事業の支援により行われたものである。

参考文献

[RCS] RCSSwiki, : http:// sserver. sourceforgec. net / wiki.

[西野 06] 西野, 久保, 下羅, 中島 : 位置に基づく行動規則を実装した入門用サッカーエージェント作成キット OZED, 第 23 回 SIG-Challenge 研究会講演論文集, pp. 19–24 人工知能学会, 2006.

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-A701-9 (5/4)

3D2Real and other perspectives for the Humanoid League

N. Michael Mayer, Joschka Boedecker,
Masaki Ogino, Sawa Fuke, Kazuhiro Masui, Ayako Watanabe, Takanori Nagura and Minoru Asada
JST ERATO Asada Synergistic Intelligence Project
michael@jeap.org

Abstract

Started in 2002 the Humanoid League has grown and is now one of the major leagues of the RoboCup. The present contribution to the SIG Challenge deals with several engineering problems and the important question how to keep and to improve the status of the league as a meaningful benchmark for science, i.e. for research issues like embodiment and autonomous behaviour and even for communication and development. In order to provide a good basis for research good engineering is essential. Thus, we discuss how this basis can look like and how joint projects with other leagues can be established. As an example we oultline the 3D2Real project.

1 Introduction

Since the start the Humanoid League (HL) underwent a profound development. Competitions and challenges have changed in various ways; rules maturated in many points and gained more focus on the issues that are essential from a technical point of view; and of course the robots became better. In the RoboCup 2005 for the first time regular 2-2 games have been conducted. In 2006 we saw a further improvement of the performance of the teams. For the RoboCup 2007 we have 24 qualified teams in the KidSize League (Height $<$ 60 cm) and 8 teams in the TeenSize league (>80cm).

In the passed some months there has been vivid discussion about the future of the Humanoid League the results are

- Increase the number of players. This has been a very emotional discussion in the past years, because the costs increase significantly with each additional player. Various test games of mixed teams have been conducted during the previous RoboCup competitions. At the moment, we are planning to increase the number of players. The most probable number at the moment is 3 players in the KidsSize in the year 2008, and further increasing numbers in the following years.

- Human-like sensors. In particular the plan is for the later future to ban the omni-vision camera. Vision sensors in other places than the head are already banned by the current rules.

- Foot size. The maximal allowed foot size in the current robots is defined as follows. The smallest rectangle covering one foot should not exceed $H^2/22$. This value has been decreased continuously from $H^2/18$ between 2004 until 2006. A further reduction is planned from 2008.

A useful measure for further milestones is the utopian sounding goal of having finals between the world champion in human soccer and robots, which will be of course humanoids. In order to achieve this target, accomplishments in several leagues have to be merged; one example how this could happen, and what benefits arise from such a merger is the 3D2Real project (see also [1]).

Whereas the KidSize robots evolved rapidly during the past 2-3 years, we expect the same development in the TeenSize yet to come. Typically, TeenSize robots are either derived from KidSize models (typically just on the lower limit of the permitted size of the TeenSize class) or we see that robots participate from initially unrelated fields of research. It is very much to hope that in the near future a TeenSize class with its own profile and own technology evolves.

One of the biggest challenges for the organizers is probably to lead the TeenSize class appropriately into a technological sound development. As already mentioned before different from the smaller robots we have not seen a break through in this area yet. The intention of the organizers is to establish the TeenSize class as a size class of significantly taller robots than those in the KidSize class. At the moment most of the robots participating in the TeenSize class are either non-functional or elongated derivatives of KidSize robots, which are just on the lower height limit of the TeenSize class. A clear profile of the TeenSize class is still missing.

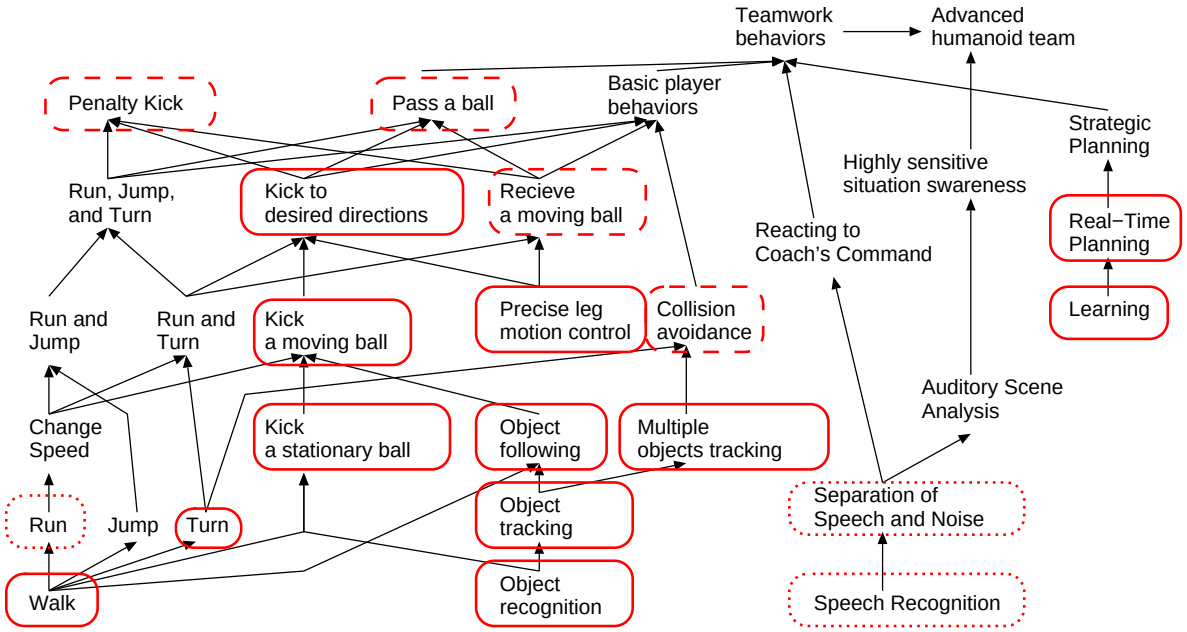


Figure 1: Road map as suggested by Kitano and Asada in 1998 [2]: Red circled are the milestones that have been achieved up to the present; solid lines represent achieved and used in regular games; dashed lines represent milestones that were achieved in some demos or technical challenges within a RoboCup event, dotted lines represent at least partly available technologies that could be integrated into the HL.

The reasons for this unsatisfying situation are multifold: One important issue is the costs. One robot of this size class is typically several times more expensive than a humanoid robot that is designed for the KidSize class. While a robot of the KidSize class can be designed at a price of below USD 10.000 the costs for a functional Teensize robot over 1m can easily reach a multitude of this figure.

The second issue is the control problem. The handling of the control is very different. RC Servos that can handle the typical forces that appear in a robot of a size of above 1m are not available, thus motor controller units have to be designed by the team themselves.

Nevertheless, at each of the last RoboCups there have been one or two promising candidates who had to some extent the potential to serve as prototypes of the TeenSize class. At the RoboCup 2005 the Team Guroo, presented a roughly 30kg heavy robot with a size over 1m. In 2006, a fully functional TeenSize robot was presented by PAL robotics. In addition, the Darmstadt Dribblers Team presented an interesting study for a robot of the TeenSize class[3].

In order to encourage its own profile of the TeenSize class with an technology that is different from the KidSize class, the rules have been modified. Already 2005 different types of competitions from the KidSize class have been introduced. Thus, regular games are currently not conducted and are also not planned within the near future. The most important planned change for the Robocup 2007 is that the minimal height H of a robot in the TeenSize class is going to be increased from 65 to 80 cm. In this way a more distinct separation of KidSize robots from TeenSize robots is intended.

Additional changes are discussed with regard to the handling of the robots. A falling TeenSize robot is more likely to be damaged than a KidSize robot thus it has been suggested to allow robot handlers on the field who can catch a falling TeenSize robot and prevent farther damage.

Different from the TeenSize, the perspective for the next several years is relatively clear in the Kidsize class. If one has followed the discussions during the time span from 2004 to the present stage one can perceive a continuous and ongoing refinement of the discussion for the benefit of the conductance of the competition, and the challenging moment of the competition.

2 Developmental science in the Humanoid League

Often a team for the Humanoid League is funded by some research organization because it is intended to use the Humanoid League competitions as a benchmark for testing a research or engineering paradigm. Reviewing the literature and also for what purpose the humanoid League has been proposed by Kitano and Asada (see Fig. 1), one can see that both goals that relate to engineering and goals that relate to research have been seen as the main challenges for the new league.

Due to the nature of the competition and in spite of significant effords of some teams, the engineering is still by far more important than the science and research issues.

The reason is that the robusteness of the robots is still not achieved to a sufficient extend and the more robust approach is usually the winner against a developmental

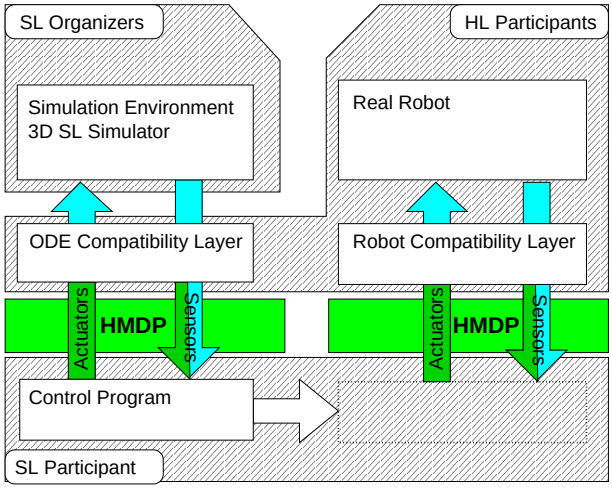


Figure 2: 3D2Real project: Layout of the control architecture. The hatched boxes show how the different leagues contribute to the complete system architecture of the (deleted for blind review) project. The control program for simulation system and real robot system are identical. The graphics includes the proposed role of the HMDP protocol [4].

or learning approach where the team had less time to put their energy into the robustness.

In addition, for developmental and learning approaches long periods or running robots are necessary. Since the lifetime of the mechanical parts is still limited and intensive maintenance of the devices necessary, for many research issues a simulation seems more appropriate than to run the whole developmental and learning process in real robots.

Many teams usually develop an own simulator as a part of their tool chain for behavior prototyping.

Recently the simlation league has started the competitions of a 3D simulatoor that simulates humanoid robots. The intention is to make this simulation (by obeying some constraints as realistic as possible). As a means the 3D2Real project is currently being discussed. As a side effect the resulting simulator can be used as a research platform for developmental science. The resulting behaviors can again be tested in real robots. In the following we outline the concept of the 3d2Real and its current achievements. We conclude with a discussion.

3 The 3D2Real Project

One problem for the RoboCup project is that throughout the leagues a lot of work is duplicated, and collaboration is rather sparse between the different leagues. This is not a desirable situation as know-how is not transferred effectively, and progress is slower than it could be since resources are bound to solve the same problems over and over again. To address this situation, the 3D2Real project [5] was initiated in 2006.

The main idea of this project is to try and use synergy effects from a collaboration between researchers in the Humanoid and the Soccer Simulation League (SSL). This collaboration includes a joint roadmap for the near future of both leagues, as well as the specification of standards and the development of tools that can be used in both leagues.

Traditionally, the SSL and the HL in RoboCup have had rather different research topics. While researchers in the HL mainly worked on the design and the low-level control of their robots, participants in the SSL were concerned with high-level strategies and collaboration. In recent years, however, there have been developments which might bring both leagues closer to each other. On the side of the SSL, there have been continuing efforts to introduce more realism into the rather abstract simulation of the SSL in order to ensure that the developed strategies can be transferred more easily to real robots. Humanoid robot simulation is the preferred choice for many participants of the SSL in order to achieve this. In the HL, on the other hand, the first multi-robot games have been held, and the great progress in controlling the robots allows researchers to approach issues of collaboration and coordination which have been extensively studied in the SSL. In short, both leagues are beginning to come closer to each other, and joint efforts in the development of tools and architectures that allow easier transfer of knowledge and technologies could speed up the mutual progress towards the 2050 goal of RoboCup.

	Soccer Simulation League	Humanoid League
Until RC 2008	real robot type implemented in 3D simulator	
RC 2007	3D SSL technical challenge: 2nd round in real robots	
2008 – 2009	Development of the CPR	
RC 2009	3D SSL finals in real robots (one type)	RoSiML models part of the HL qualification
RC 2010	3D SSL finals with several types of real robots	HL teams commit to the CPR

Table 1: Overview of the current road map towards the milestone of the 3D2Real project.

The joint road map we propose is given in table 1. The goal we envision for the 3D2Real project is to have the finals of the simulation league using real robots by the year 2009. For this ambitious goal several steps are necessary in the next years to create the necessary infrastructure and tools. First, the 3D simulator of the SSL [6] has to be completed, and a real robot prototype has to be implemented as a simulation model. For the description of the robot models, the XML-based format *RoSiML* as used in the *SimRobot* simulator [7] seems promising. According to the proposed road map, a technical challenge would be held at RoboCup 2007 to test the ability to use the agent code of SSL participants on a pre-determined real robot. From 2007 until 2008, we propose the devel-

opment of a *central parts repository* (CPR). This would be a collection of real robot designs, sensor and actuator models, pre-assembled robots, as well as controllers for certain architectures. Participants of both HL and SSL contribute to this repository according to their expertise and interest. The format would again be the *RoSiML* mentioned above. These contributions become a mandatory part for the HL qualification from 2008, and should be continued (at least) until 2009, even after the 3D SSL final has taken place using real robots.

Current State: In the SSL the 3D competition is for the first time conducted in to some extend realistic humanodi robots. For the abstract communication of motion patterns a new type of protocol (HMDP, please cf. [4]) has been proposed.

4 Discussion

If one assumes the Humanoid League as huge evolutionary project this type of design can be seen as the result of the optimization process of the challenges to which the teams have been exposed so far.

On the other hand there are still many things to overcome and to be done in order to go on further. Fig. 1 depicts the initial road map for milestones to be achieved within the Humanoid League. Those were outlined in 1998 [2]. One can see that the league has reached some milestones since it was introduced in the RoboCup of 2002 in Fukuoka. Many milestones related to controlled walking, object recognition and ball handling have been realized.

Some of the missing issues are partly subject to leanring and developmental research issues. For these issues the 3D2Real environment can be useful toolbox to develop behaviors in the simulator and to test them in the real robots.

Acknowledgements

We greatfully acknowledge the support of this work by the Japan Science and Technology Agency (JST), and a fellowship for young scientists from the Japan Society for the Promotion of Science (JSPS).

References

[1] Boedecker, J., Mayer, N.M., Ogino, M., da Silva Guerra, R., Kikuchi, M., Asada, M.: Getting closer: How simulation and humanoid league can benefit from each other. In Murase, K., Sekiyama, K., Kubota, N., Naniwa, T., Sitte, J., eds.: Proceedings of the 3rd International Symposium on Autonomous Minirobots for Research and Edutainment, Springer (2006) 93–98

[2] Kitano, H., Asada, M.: RoboCup Humanoid Challenge: That's One Small Step for A Robot, One Giant Leap for Mankind. In: Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '98). (1998) 419 – 424

[3] Friedmann, M., Kiener, J., Kratz, R., Petters, S., Sakamoto, H., Stelzer, M., Thomas, D., von Stryk, O.: Team Description Paper: Darmstadt Dribblers and Hajime Team (KidSize) and Darmstadt Dribblers (TeenSize). In: RoboCup 2006 Symposium papers and team description papers CD-ROM. (2006)

[4] Mayer, N.M., Boedecker, J., Masui, K., Ogino, M., Asada, M.: HMDP: A new protocol for motion pattern generation towards behavior abstraction. In: RoboCup Symposium. (2007) (accepted).

[5] Mayer, N.M., Boedecker, J., da Silva Guerra, R., Obst, O., Asada, M.: 3d2real: Simulation league finals in real robots. In Lakemeyer, G., Sklar, E., Sorrenti, D.G., Takahashi, T., eds.: RoboCup 2006: Robot Soccer World Cup X. Lecture Notes in Artificial Intelligence, Springer (2006) to appear.

[6] Obst, O., Rollmann, M.: SPARK – A Generic Simulator for Physical Multiagent Simulations. Computer Systems Science and Engineering **20**(5) (September 2005)

[7] Laue, T., Spiess, K., Röefer, T.: SimRobot - a general physical robot simulator and its application in robocup. In: RoboCup 2005: Robot Soccer World Cup IX. Lecture Notes in Artificial Intelligence, Springer (2006)

間引きを用いたシュートモーション学習

Shot Motion Learning Using Thinning-out

小林 隼人 [1] **畑埜 晃平** [2] **石野 明** [1] **篠原 歩** [1]

Hayato Kobayashi[1], **Kohei Hatano**[2], **Akira Ishino**[1], **and Ayumi Shinohara**[1]

[1] 東北大学大学院情報科学研究科 [2] 九州大学大学院システム情報科学研究科

[1]**Graduate School of Information Sciences, Tohoku University**

[2]**Graduate School of Information Science and Electrical Engineering, Kyushu University**

{kobayashi@shino., ishino@, ayumi@}ecei.tohoku.ac.jp and hatano@i.kyushu-u.ac.jp

Abstract

Shooting is one of the most important skills in soccer games, as it directly a ects scoring points. In the four-legged robot league, we must develop accurate and strong shot motions only by changing the parameters of the motions, since the hardware is limited to AIBO. This task consumes much of developers' time and energy and is needed whenever the environment (e.g., friction of the eld) is changed. By using machine learning, we can efficiently reduce these burdens. Existing learning methods, however, take much evaluation time for each trial in the motion learning. In this paper, we propose a new concept, thinning-out, for reducing the number of trials in the motion learning. Thinning-out means to skip over such trials that are unlikely to improve motions, in the same way that gardeners thin-out weak seedlings. We show that our thinning-out technique signi cantly reduces the number of trials. In addition, we show that our virtual robots can acquire a sophisticated motion that is much di erent from the initial motion as the result of utilizing a hybrid learning method combining meta-heuristics and the thinning-out technique.

1 はじめに

ロボットが実世界で機能するには，未知の環境に適応する能力，つまり学習能力が必要である．この能力は，RoboCup サッカーにおいても重要な役割をしめる．特に四足ロボットリーグやヒューマノイドリーグにおける身体ロボット（肢体を持つロボット）は，歩く，シュートする，パスするといった基本技術を獲得するための能力が必要である．この，身体ロボットが基本技術を学習する能力は，**身体学習** (physical learning) として知られており，重要視されている．

ここ数年でも，多くの身体学習に関する研究が存在する．Kim と Uther [2]は，歩行の軌跡を四角形でモデル化して，ロボットに高速な歩行を学習させた．Saggar ら [6]は，歩行の軌跡を楕円形でモデル化して，ロボットに視覚情報を使いながら安定した歩行を学習させた．Fidelman と Stone [1]は，ボールを掴む技術の学習方法を提案した．この学習は，1 層目に歩行学習，2 層目にボールを掴む技術の学習を導入した階層学習である．Kobayashi ら [4]は，強化学習で動いているボールのトラップ技術を 1 次元で学習させた．

本論文では，Zagal と Solar [8]と同様に，シミュレータ上でロボットにシュートモーションを学習させる．素早く強力なシュートは直接得点に結びつくため，シュートモーションの作成は RoboCup サッカーにおいて重要な作業のひとつである．しかしシュートモーションの学習は問題のモデル化が難しいため，上で述べた他の作業と比較すると難しい問題といえる．Zagal と Solar は，動作モデルを仮定することなくシュートモーションを直接学習させたが，彼らの方法は足の関節のみを対象にしており，探索する次元数も固定している．我々も同様にモデルを置かないが，我々の方法はすべての関節を対象としており，次元数も固定しないため，どんなモーションにも適用することができる．しかし，それゆえに学習には多くの試行回数を必要としてしまう．一回の試行に非常に時間がかかるロボットの学習においては，この問題は深刻である．

本論文では，この問題を解決するために，学習過程における試行回数を削減する**間引き** (thinning-out) を提案する．間引きは，決定木学習における**枝刈り** (pruning) と同様の概念であり，園芸家が悪い苗を間引くように，学習

過程において結果に貢献しない試行を省くことを意味する．我々は，間引きを使うことで効果的に試行回数を削減できることを示す．この間引きは，**メモリーベース学習** (memory-based learning) の一種であり，Memory-based Fitness Evaluation Genetic Algorithm (MFEGA) を提案した Sano[7]らと動機は同じである．しかし，我々の提案する間引きは，スコア関数の形を推定しながら試行自体を直接省くという点で完全に彼らの手法とは異なる．

本論文の構成は以下のとおりである．第 2 節では，我々のフレームワークにおけるモーション再生の仕込みとシュートモーションを作成するためのモーションエディタを説明し，本論文で取り扱う学習問題を定式化する．第 3 節では，間引きの方法について説明し，それを実現するための 2 つの手法を提案する．そして，間引きの候補点を選択するためのメタヒューリスティクスについて説明する．第 4 節では，提案した学習手法を用いて，シミュレータ上のロボットにシュートモーションを学習させ，その実験結果について考察する．第 5 節では，全体のまとめと今後の課題について述べる．

2 シュートモーションの学習

2.1 シュートモーション

AIBO のモーションは，首や足の 15 個の関節角度で構成される**フレーム** (frames) を 8 ms ごとに OVirtualRobot へ送り込むことで実現される．OVirtualRobot はプロキシオブジェクトの一種であり，AIBO のソフトウェア開発キット OPEN-R で定義されている．我々のフレームワークでは，これらのフレームは**キーフレーム** (key-frames) によって生成される．キーフレームは，それぞれのモーションの骨組みとなる特徴的なフレームのことである．たとえば，キックモーションはボールを押し出すために足を一度後ろに下げてから前に出さなければならないので，少なくとも 2 つのキーフレームが必要である．我々はそれぞれのキーフレームに対して補間数を指定し，それを元に線形補間で完全なフレームを生成する．それゆえ，我々のモーションを実行するには，補間数を n とすると $8n$ ms が必要である．

2.2 モーションエディタ

我々は，モーションを簡単に作成し調整するために，図 1 のようなスプレッドシートを持つモーションエディタを作成した．このエディタを使うことで，我々はそれぞれのモーションのためのキーフレームとその補間数を指定することができる．しかし，通常はロボットを見ただけではその関節角度の正確な値を推定することができないため，手作業で関節角度のすべてを入力するのは困難である．したがって，エディタにロボットの関節角度をキャプチャする機能を持たせた．我々のロボットプログラムは，肉球（足裏の接触センサ）をクリックすることでその足を脱力させることができ，もう一度クリックすることで固定することができるため，我々はパラパラ漫画やクレイアニメを作成するのと同様に，直感的にさまざまなモーションを作成できる．

2.3 学習問題

本章では，キーフレームを直接探索空間として使うことでシュートモーションの学習を行う．我々は，学習後のモーションの骨組みを決める大雑把なモーションを作成するだけでよい．そうすれば，その骨組みから，その近傍のよいシュートを学習することができる．今回は補間数を学習の対象としないため，キーフレーム数を n とすると，取り扱う探索空間は $15n$ 次元となる．

シュートを評価するためのスコア関数は，ボールの相対距離 r_b, ボールの相対角度 θ_b として以下のように設計した．

$$score(r_b, \theta_b) = r_b \cdot \left(1 - \frac{|\theta_t - \theta_b|}{\theta_c}\right)$$

この式は，シュートの目的角度 θ_t とボールの相対角度との差に反比例して，スコアを線形的に小さくする．θ_c はスコアの減少具合いと決める定数であり，本論文では $\theta_c = 45$ とした．

3 学習方法

3.1 間引き

本節では，不必要な試行を間引く方法について議論する．まず我々は，スコア関数が探索空間上で連続で，ある程度なめらかであることを仮定する．ロボットの動きは連続であり，パラメータの小さな変更はスコアにそれほど影響を及ぼすことはないため，この仮定は妥当なものといえる．この仮定に基づき，我々はスコア関数の局所的な形を推定する．与えられた試行の候補点に対して，我々は試行の履歴中の最近傍点との距離を使うことで候補点のスコアを推定する．もし推定されたスコアの上限が現在の最高点よりも低い場合は，その候補点を間引く．このとき，現在の最高点との距離を考慮にいれていないことに注意されたい．我々は真の最高点が現在の最高点の近くにあることを仮定していない．それゆえ，高いスコアが得られると予測された点は，たとえ現在の最高点よりも遠かったとしても試行される可能性がある．したがって，我々の方法は局所解に陥りにくいという意味で頑健であるといえる．

ここで我々は，**リプシッツ条件** (Lipschitz condition) を持ちいて，スコア関数の局所的ななめらかさを定義する．リプシッツ条件は，微分積分の標準的な教科書にあるように定数 c に対する c-リプシッツ連続を定義するものだが，本論文ではその自然な拡張として関数 g に対する g-リプシッツ連続を定義する．

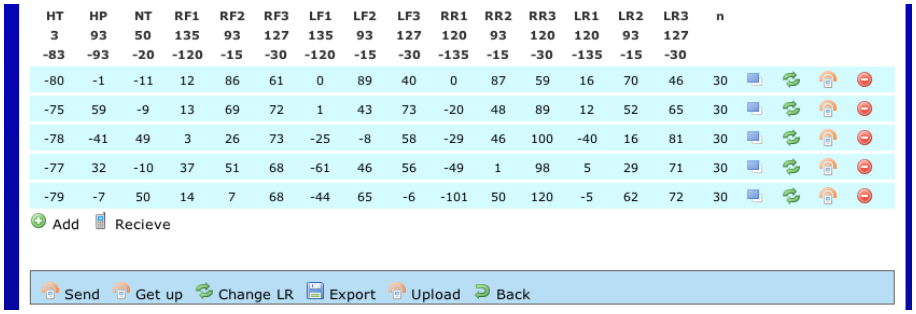


Figure 1: 作成したモーションエディタの一部分．スプレッドシートの 1 行がキーフレームを意味する．HT, HP, ..., LR3 の列はそれぞれの関節角度を意味し，n の列はそのキーフレームまでの補間数を意味する．

定義 1 (リプシッツ条件，Lipschitz condition) $\boldsymbol{X}$ を探索空間とする．$f : \boldsymbol{X} \to \boldsymbol{R}$ を探索空間上のスコア関数とする．ある関数 $g : \boldsymbol{R} \to \boldsymbol{R}$ について，どんな $x_1, x_2 \in \boldsymbol{R}$ に対しても以下の条件を満たすとき，f は g-リプシッツ連続 (g-Lipschitz continuous) であるという．

$$|f(x_2) - f(x_1)| \leq g(|x_2 - x_1|).$$

f を g-リプシッツ連続とする．そのとき，どんな x_1, x_2 に対しても，次のように $f(x_1)$ の上限が得られる．

$$f(x_1) \leq f(x_2) + g(|x_2 - x_1|).$$

これより，我々はスコア関数 f が，ある g に対して g-リプシッツ連続であると仮定する．我々の間引き戦略は，候補点のスコアの上限が得られるように，スコア関数 f を特徴づける関数 g を推定することである．もしもその上限が現在の最高点よりも小さいとき，我々はその候補点を試行する必要はない．関数 g を推定する方法の詳細は後に述べる．我々の間引き条件は以下のように定式化される．

定義 2 (間引き条件，Thinning-out condition) x_b を現在の最高点とする．x_c を試行の候補点とする．x_n をその候補点の最近傍点とする．ある推定関数 $\hat{g} : \boldsymbol{R} \to \boldsymbol{R}$ について，次の条件を満たすとき，x_c は $\hat{g}$ に関して間引き条件 (thinning-out condition) を満足するという．

$$f(x_n) + \hat{g}(|x_c - x_n|) \leq f(x_b)$$

ここで，我々は g を推定する 2 つの手法を提案する．

最大傾斜法 (Max Gradient Method)

まず，X 上の任意の 2 点 x_1，x_2 について，スコア関数 f の最大の傾斜

$$c = \max_{x_1 \neq x_2,\ x_1, x_2 \in X} \frac{|f(x_2) - f(x_1)|}{|x_2 - x_1|}$$

を知っていると仮定する．このとき，$g(d) = c \cdot d$ によって定義される関数によって，f が g-リプシッツであることは容易にわかる．それゆえ，g は安全に候補を間引くことができる．しかし実際には，c 自身を知ることはできないので，我々は過去の試行におけるすべての 2 点における最大傾斜でそれを置き換える．これはおそらく，十分な試行のあとには c のよい近似となるはずである．したがって我々はこの方法を最大傾斜法 (MG) と呼ぶ．

この方法は，スコア関数上で起伏の一番大きい場所を基準とする単純なものだが，一番安全な方法である．しかし，関数 g が返す値は多くの場合大きすぎるため，起伏の大きいスコア関数上ではほとんど候補を間引くことはできないと予想される．

差分収集法 (Gathering Differences Method)

差分収集法 (GD) は，試行の履歴中の任意の 2 点において，それらの差の小さなものから昇順で，候補点と最近傍点との距離以上になるまで，それらのスコアの差分を足し合わせることで関数 g を推定する．Algorithm 1 にこの方法のアルゴリズムを示す．

この方法は，現在の最高点付近のスコア関数の局所的な形を推定するので，多くの候補点を間引くことができる．しかし，この方法は単なるヒューリスティクスであり理論的妥当性はないため，間違って間引く可能性は MG よりも高いと予想される．

3.2 メタヒューリスティクス

我々が提案した間引きは単に候補点を省く手法であるため，その候補点を取ってくるための効率のよいサンプリング手法が必要である．シュートモーションの学習問題ではスコア関数が未知であるため，いくつかのメタヒューリスティクスを利用する．Kohl と Stone [5]は，遺伝的アルゴリズム (Genetic Algorithm: GA)，山登り法 (Hill Climbing: HC)，政策勾配法 (Policy Gradient: PG)，アメーバ法 (Amoeba) の 4 つのメタヒューリスティクスを用いて，四足歩行の学習をおこなった．本研究でも，これらのメタヒューリスティクスを利用するが，アメーバ法だけは除いた．これは，アメーバ法が決定的アルゴリズムであるため，間引きを直接適用することができないからである．それゆえ，アメーバ法の代わりに，よく知

Algorithm 1: Gathering de erences method

input : distance and the set Hist of pairs $(x, f(x))$ observed so far

output: An inferred value of g(distance)

initialize Diff as a map from $\mathcal{R}$ to $\mathcal{R}$;
foreach key1 value1 Hist **do**
  foreach key2 value2 Hist **do**
    Diff [|key1 − key2|] ← |value1 − value2|;
  end
end
sum_diff_key ← 0;
sum_diff_val ← 0;
foreach key value Diff key **do**
  sum_diff_key ← sum_diff_key + key ;
  sum_diff_val ← sum_diff_val + value ;
  if sum_diff_key ≥ distance **then**
    return sum_diff_val ;
  end
end
return ∞ ;

られたメタヒューリスティクスの一つである疑似焼き鈍し法を (Simulated Annealing: SA) を実験に加えた．GA, HC, PG に関しては Kohl と Stone のアルゴリズムと同じものを使用するため，SA のアルゴリズムの詳細に関してだけ述べる．

疑似焼き鈍し法 (Simulated Annealing)

SA は，Kirkpatrick ら [3]によって提案された金属工学における焼き鈍し処理を模した最適化手法である．Algorithm 2 に，間引きを用いた SA のアルゴリズムを示す．我々の提案する間引きはメタ戦略であるため，他の手法に関しても，同様の方法で簡単に間引きを適用することができる．

4 実験

4.1 シュートモーションの学習

本節では，間引きが身体学習に適応できることを確かめるために，物理シミュレーションによるシュートモーションの学習を行う．シミュレータとして，Zaratti ら [9]が開発した 3D シミュレータを拡張して使用した．このシミュレータは，非常に使いやすいユーザインターフェイスを持っており，実ロボットと同じように各関節角度を送り込むことで簡単に仮想ロボットを動かすことができる．このシミュレータは完全な実環境を実現するものではないが，ロボットを傷つけないだけではなく，実ノイズに悩まされることなく再現性のある実験ができるため，新しい手法の性能を検証するには敵した環境である．

Algorithm 2: Simulated annealing with thinning-out

input : num_trials, initial_motion, initial_temperature, cooling_facter

output: an optimized motoin

state ← initial_motion ;
Hist [state] ← Evaluate(state);
temperature ← initial_temperature ;
while num_trials **do**
  new_state ← a random perturbation of state ;
  while new_state **do**
    new_state ← a random perturbation of state ;
  end
  new_score ← Evaluate(new_state);
  if new_score < score **then**
    if $exp\left(-\frac{\text{score}-\text{new\_score}}{\text{temperature}}\right)$ **then**
      state ← new_state ;
      score ← new_score ;
    end
  end
  temperature ← temperature · cooling_facter ;
  Hist [new_state] ← new_score ;
end
return Max(Hist);

本研究では，右前足で左斜め方向にボールを打つシュートモーションを初期モーションとして用いた．このモーションは 5 個のキーフレームを使用するため，探索空間は 75 次元となる．実験はメタヒューリスティクスにより 50 個の候補点を選ぶことを 10 回行い，間引き率とその失敗率の平均を出した．(今回は失敗率を出すために間引かれた候補点も評価した.）本実験はシミュレータ上で行うが，物理シミュレーションには複雑な計算が必要であるため，それでもかなりの時間が必要となる．実際，今回の一回の実験には数十時間を要した．それゆえ，時間のかかる試行自体を削減する間引きは，実環境だけではなく，仮想環境においても有効に働く．

表 1 に，2 種類の間引き (MG, GD) を適用した各種メタヒューリスティクス (GA, SA, HC, PG) による実験結果を示した．表の MG と GD を比較すると，MG は比較的安全だがあまり候補点を間引けず，GD は失敗を恐れずに多くの候補点を間引けることが分かる．また，GD は 9 割以上の成功率で試行回数を通常の半分程度にできることが分かる．今回は 50 個の候補点での結果であるため間引き率はそれほど高くないが，候補点が多いほど間引き率を高くできることが分かっている．

MG と GD の性質を比較するために，SA にそれぞれの

手法を適用した実験結果のグラフを図 2 に示す．青い丸は実際に試行された候補，黄色い四角は正しく間引かれた候補，赤い十字は間違えて間引かれた候補を表す．表 1 と同様に，MG の結果 (a) では高い確率でスコアが低い候補点だけが間引かれており，GD の結果 (b) ではある程度高いスコアを持つ候補点も間引かれていることが分かる．GD の結果 (b) に表れているように，誤って間引いたとしても，未来の試行でより高いスコアを持つ候補点を見つける可能性がある．

図 3 に，初期モーションと学習の結果得られたモーションを示す．初期モーション (a) はほとんど左前足だけしか使っていなかったが，学習したモーション (b) は体全体を使っている．また，モーション (c) は体重を乗せて，足を降り下ろすモーションである．これらの学習後のモーションはどちらも体全体を動かしているため，次元数を減らすためにモデル化するような既存の学習方法では獲得することが困難である．学習前，学習後それぞれのモーションの動画は，我々のサイト (http://www.jollypochie.org/papers/) で閲覧可能である．

5 まとめと今後の課題

本論文では，それぞれの試行に多くの評価時間を要する問題に対して有効な間引きを提案した．また，間引きで利用する関数 g を推定するための 2 つの手法 (MG と GD) を提案した．シュートモーション学習の実験により，本手法がさまざまなメタヒューリスティクスに関して有効であり，汎用的なメタ戦略であることを確かめた．

今後は，間引きが実ノイズに耐えうるかどうかを確かめるために，実ロボットでの実験が必要である．しかし，実環境でのシュートモーション学習は，毎回の試行後に転がったボールの距離や角度などを測定し，それを注意深く元の場所に戻す作業を行わなければならないため，非現実的である．それゆえ我々は，Kobayashi ら[4]のように，ロボットに**自律学習** (autonomous learning) を行わせることを検討している．前方シュートの自律学習は，彼らの手法をそのまま利用することができる．他の方向のシュートに関しても，天井カメラなどを用いれば可能である．自律学習という概念は，通常は現実的ではなかったシュートモーション学習などを実現するだけではなく，試合中に，つまりオンラインで身体動作を学習し，向上させる可能性を秘めている．実機リーグにおいては，戦略のオンライン学習はいくつか報告されているが，身体動作のオンライン学習はいまだ行われていない．オンライン学習能力は，未来の RoboCup に必要とされる重要な能力であるため，取り組みがいのある問題だといえる．

また，本章で提案した 2 つの手法は一長一短であったため，我々はより精度の高い関数 g の推定方法を考える必要がある．我々は，重みつき平均を用いた手法など，2 つの手法以外にもいくつかの推定方法を思いついたが，どれもうまくいかなかった．今後は，個々の問題に依存したヒューリスティクスをうまく活用することによって，もっとよい推定ができないかを考えたい．

参考文献

[1] P. Fidelman and P. Stone. The Chin Pinch: A Case Study in Skill Learning on a Legged Robot. In , LNAI. Springer-Verlag, 2007. to appear.

[2] M. S. Kim and W. Uther. Automatic Gait Optimisation for Quadruped Robots. In , pp. 1–9, 2003.

[3] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by Simulated Annealing. , 220(4598):671–680, May 1983.

[4] H. Kobayashi, T. Osaki, E. Williams, A. Ishino, and A. Shinohara. Autonomous Learning of Ball Trapping in the Four-legged Robot League. In , LNAI. Springer-Verlag, 2007. to appear.

[5] N. Kohl and P. Stone. Machine Learning for Fast Quadrupedal Locomotion. In , pp. 611–616, 2004.

[6] M. Saggar, T. D'Silva, N. Kohl, and P. Stone. Autonomous Learning of Stable Quadruped Locomotion. In , LNAI. Springer-Verlag, 2007. to appear.

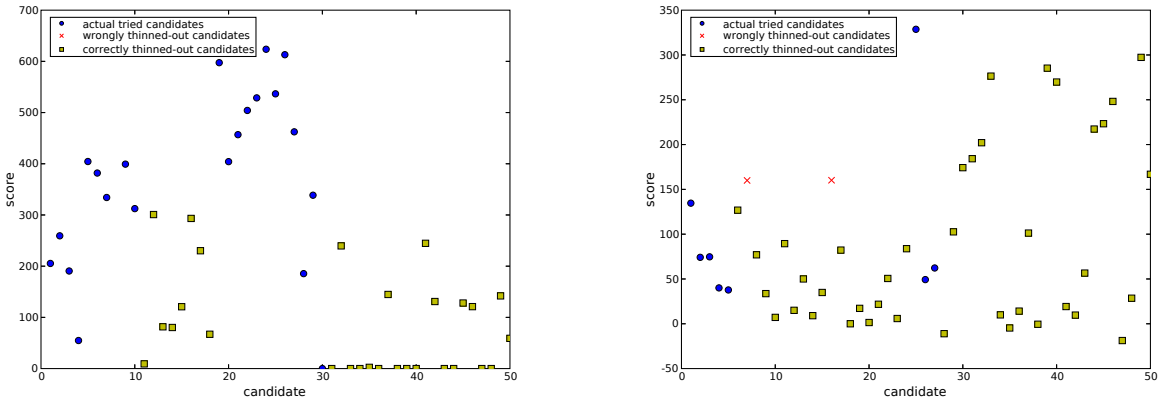
[7] Y. Sano and H. Kita. Optimization of Noisy Fitness Functions by means of Genetic Algorithms using History of Search with Test of Estimation. In , pp. 360–365, 2002.

[8] J. C. Zagal and J. R. del Solar. Learning to Kick the Ball Using Back to Reality. In , Vol. 3276 of , pp. 335–347. Springer-Verlag, 2005.

[9] M. Zaratti, M. Fratarcangeli, and L. Iocchi. A 3D Simulator of Multiple Legged Robots based on USARSim. In , LNAI. Springer-Verlag, 2007. to appear.

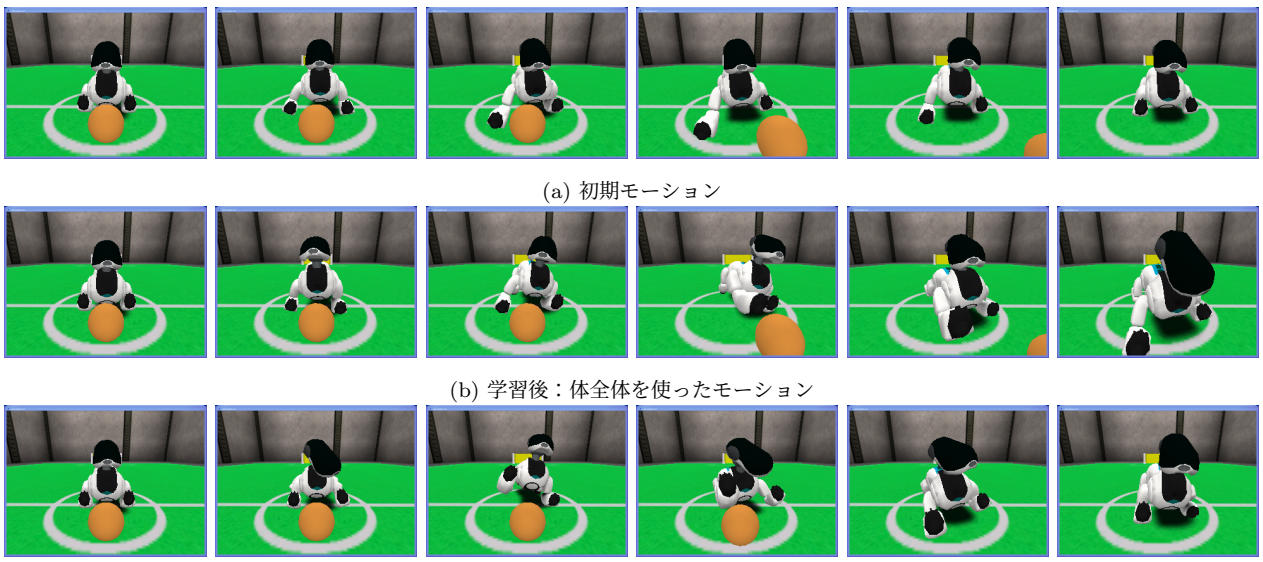
Table 1: シュートモーション学習の実験結果．actual trial は間引き率，error rate は失敗率を表す．失敗率は，間引かれた候補の中で，学習の性能を落とす原因となった候補の率を表す．

	actual trial		error rate	
	MG	GD	MG	GD
GA	37.6%	49%	0.56%	7.69%
SA	66.2%	76.2%	0.75%	7.00%
HC	7.2%	57.4%	0.00%	2.88%
PG	10.6%	47.6%	0.71%	3.54%



(a) 最大傾斜法　　(b) 差分収集法

Figure 2: 疑似焼き鈍し法における最大傾斜法と差分収集法の実験結果



(a) 初期モーション

(b) 学習後：体全体を使ったモーション

(c) 学習後：体重を使ったモーション

Figure 3: 初期モーションと学習後のモーション

ニューラルネットとヒューリスティックを用いたドリブルスキルの開発

Developing Dribble Skills Using Neural Networks and Heuristic Knowledge

中島 智晴，荘司 悠希男，石渕 久生

Tomoharu NAKASHIMA, Yukio SHOJI, Hisao ISHIBUCHI

大阪府立大学大学院 工学研究科

Graduate School of Engineering, Osaka Prefecture University

nakashi@cs.osakafu-u.ac.jp, shoji@ci.cs.osakafu-u.ac.jp, hisaoi@cs.osakafu-u.ac.jp

Abstract

In this paper we show the developmente of dribble skills using neural networks and heuristics. We divide the dribble skill into two parts such as driving part and heading part. In the driving part the dribbling agent moves straight towards its heading while kicking the ball. The predicted position of the ball is calculated from the physical movement specified by the soccer server. For the heading part we use a neural network that is trained to mimic the direction of a target dribbling agent. We also use heuristic rules to determine the heading of the dribbling agent when it is in the opponent's penalty area.

1 はじめに

ロボカップサッカーシミュレーションは，エージェントの物理的な設計の必要が無く，自律的なサッカー行動をおこなうための研究が行いやすい環境である．サッカー行動は大きく二つに分けることができる．一つは，エージェント個人の技能により関係する低レベル行動であり，もう一つは，チーム全体の協調行動により関係する高レベル行動である．

洗練された戦略の開発により重点がおかれている．しかし，どんなにすぐれた戦略を開発することができたとしても，

2 ドリブルスキル

ドリブルスキルは，キックとダッシュを組み合わせてボールとエージェント自身を移動させる行動とみなすことができる．さらに，目的の方向と移動方向が異なっている場合には体の向きを変更させることが必要となる．そこで，本論文では，ドリブルスキルを図 1 のような構造を持つ意思決定として取り扱うことにする．

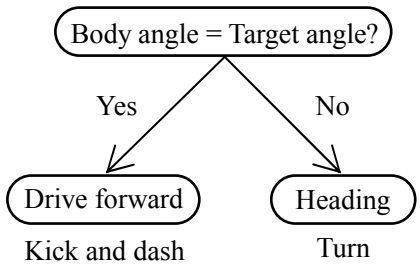


Figure 1: ドリブルスキルにおける意思決定の構造

図 1 では，ドリブルを行う際，まずはじめに体の向きが目標角度と一致しているかどうかを調べる．もし一致していれば前進のための意思決定（キックもしくはダッシュ）を行い，一致していなければ方向変換（ターン）の意思決定を行う．

3 前進移動

前進移動はキックによりボールを前方へ移動させ，ダッシュにより自身を前方に移動させるという行動の組み合わせである．ボールをキックするためには，ボールがエージェントキッカブルエリア内になければならない（図 2）．従って，キックする際には，次にキックをする時点でのボールの位置を計算しておかなければならないことになる．

本論文では，あるキックから次のキックまでの行動を一連のエピソードとする．次のキックまでに何回ダッシュするかが決められると，キックのパラメータ（パワーと角度）がそれに応じて計算される．なお，以下の計算では，エージェントやボールの移動にノイズは考慮されていない．

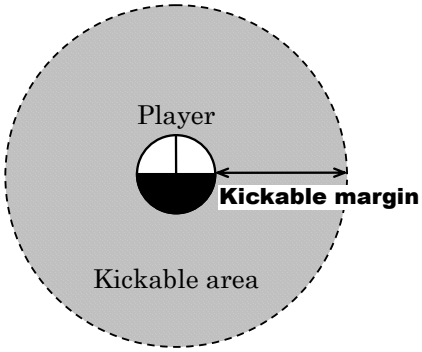


Figure 2: キッカブルエリア

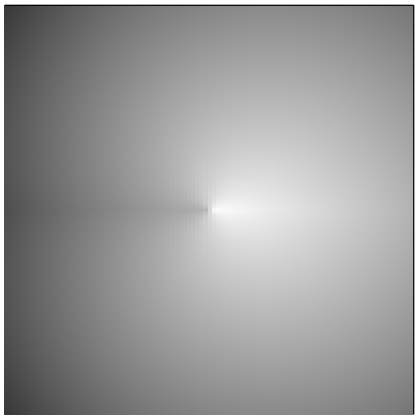


Figure 3: ボール位置が加速度に与える影響

3.1 キックパラメータの決定

キックの後に n 回のダッシュを行い，一つのエピソードが終了すると仮定する．時刻 t におけるキックのパラメータは，時刻 $t+n+1$ にボールが目標地点にあるように計算される．

時刻 t におけるボールの位置ベクトルと速度ベクトルをそれぞれ X_b^t，V_b^t，キックにより与えられたボールの加速度ベクトルを A_b^t とする．このとき，時刻 $t+n+1$ におけるボールの位置 X_b^{t+n+1} は以下で求められる．

$$X_b^{t+n+1} = X_b^t + \sum_{k=0}^{n}(V_b^t + A_b^t) * decay_{\mathrm{b}}^k \tag{1}$$

ここで，$decay_{\mathrm{b}}$ はボールに対する減衰係数であり，定数である．

一方，エージェントは時刻 t においてキックした後 n 回ダッシュを行い，時刻 $t+n+1$ で再びキックをする．ダッシュのパラメータ（パワー）を $dashPower$ とする．一つのエピソード内では，$dashPower$ は一定であるとする．時刻 t におけるエージェントの位置ベクトルと速度ベクトルをそれぞれ $V^t = (v_x^t, v_y^t)$，を $X_p^t = (p_x^t, p_y^t)$，エージェントのスタミナを $stamina^t$ とすると，キックの後連続して n 回ダッシュした際の時刻 $t+n+1$ におけるエージェントの位置 X_p^{t+n+1} は以下で求められる．

$$\begin{aligned} X_p^{t+n+1} = & \\ & X_p^t + V^t \\ & + \sum_{k=1}^{n}\{V_p^t * decay^l + \sum_{l=1}^{k} A_p^{t+l} * decay_{\mathrm{p}}^{l-1}\} \end{aligned} \tag{2}$$

ここで，A_p^t は，時刻 t にプレーヤに与えられた加速度である．ただし，式 (1) と式 (2) の両方において，サッカーサーバで定められている最高速度を超えることはないものとする．

時刻 $t+n+1$ におけるボールの位置は，エージェントの位置より少しずれた位置になければならない（衝突するため）ので，ボールの目標位置を $X_p^{t+n+1} + \phi$ とすると，キックによりボールに与える加速度は

$$X_b^{t+n+1} = X_p^{t+n+1} + \phi \tag{3}$$

を式変形することにより得られる．A_b^t が決定された後，ボールに加速度 A_b^t を与えるキックのパラメータであるパワーと角度を決定する．

キックがボールに与える加速度ベクトルの大きさは，キックパワーとキックした時点でのボールとエージェントの位置関係に依存している．図 3 に，ボールの位置関係がキック力に与える影響を図示している．ボールの位置関係がキック力に与える影響を ρ とすると，ρ は以下で求められる．

$$\rho = 1.0 - 0.25 * \frac{\delta_{dir}}{180} * 0.25 * \frac{\delta_{dist}}{kickableMargin} \tag{4}$$

ここで，δ_{dir} はエージェントの体の向きに対するボール位置の角度，δ_{dist} はエージェントからボールまでの距離，$kickableMargin$ は，サッカーサーバで決められているキッカブルマージンであり，定数である．

図 3 では，図の中心にエージェントがおり，向かって右側が前方であると仮定している．色が明るい部分ほどキックパワーがそのままボールの加速度に反映されることを示し，暗い部分ほどキックパワーが加速度に反映されないことを示している．図 3 より，ボールがエージェントの前方にあるほど，さらにボールがエージェントに近いほどキックパワーがボールの加速度に与える影響が高いことがわかる．

キックパワーとボール位置が加速度に与える影響が与えられたという過程のもとで，ボールに与えられる加速度 A_b の大きさは以下で決定される．

$$|A_b| = kickPower * 0.027 * (1.0 - \rho) \tag{5}$$

式 (5) より，ドリブル中，時刻 t でキックする際のパラメータ $kickPower^t$ は以下となる．

$$kickPower = |A_b^t|/0.027/(1.0 - \rho) \tag{6}$$

キック角度は，$|A_b^t|$ の角度とエージェントの体の向きとの差から求められる．

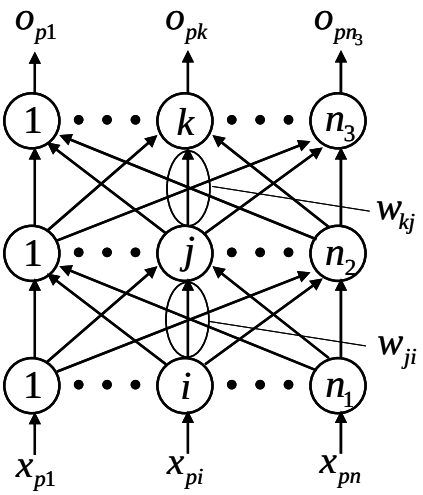


Figure 4: 3層階層型ニューラルネット

3.2 ダッシュパワー

3.1 の仮定により，ダッシュパワーは一度決定されるとエピソードが終わるまで固定である．キック後に連続して n 回のダッシュを行う実装は agent2d[1]の Intention クラスを用いて行われる．

4 方向決定部

ドリブルの目標方向決定は，ニューラルネットにより行われる．ニューラルネットは，3層階層フィードフォワード型（図 4）を用いる．特に，本論文では，2つの入力層ユニット，5つの中間層ユニット，1つの出力層ユニットから構成されるニューラルネットワークを用いて方向を決定することにする．ニューラルネットワークへの入力はボールの位置，ニューラルネットワークからの出力はドリブルの移動方向とする．ただし，ニューラルネットワークの出力は 0 から 1 までの実数であるので，これをドリブルスキルへ適用する際には− 90 から+90 へのスケール化を施すことにする．

ニューラルネットの学習用パターンとして，インターネット上で公開されているチームのドリブル方向を利用する．学習用パターンの生成手続きは以下になる

Step 1: ターゲットとするチームを用いて試合を行う．

Step 2: ターゲットチームがドリブルをしている時刻とエージェント番号を記録する．

Step 3: 記録した時刻のボールの位置をニューラルネットの入力とし，そのときのエージェントの体の向きをニューラルネットの教師出力とする．

図 5 に，上記の手続きによって得られたドリブルの例を示す．この図は，2005 年の世界大会に出場したチーム HELIOS のエージェントから得られたドリブルの軌跡である．ドリブルの軌跡から，ボールの位置とエージェントの体の向きをニューラルネットの教師出力として利用する．なお，ニューラルネットの学習用パターンの生成はすべて手作業で行われる．図**??**で，ドリブルの移動方向が変更されているのは，近傍に静止している敵がいるため，回避している様子を示している．

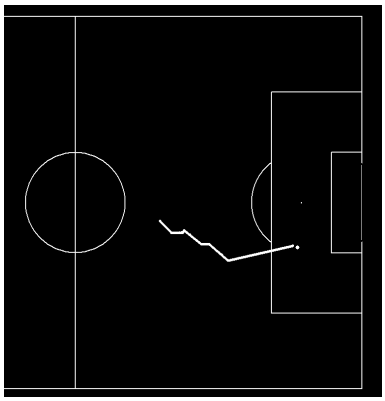


Figure 5: 抽出されたドリブル軌跡の例

なお，ニューラルネットの学習はサッカーの試合中には行わず，オフラインで行われる．したがって，本論文の仕様では，いったん試合が始まるとドリブルの移動方向はボールの位置によって一意に決まる．ボールを奪いにくる敵に対して回避行動をとる必要があるが，本論文では考えられておらず，今後の課題である．

5 数値実験

数値実験では，主にニューラルネットによって得られたドリブル方向について考察する．学習用パターンの抽出に使用したチームは TokyoTech06[3], TokyoTech05[2], YowAI06yowai06, UvA_Trilearn03[4]である．それぞれのチームに対して，試合中まったく動かないエージェントからなるチームを敵チームとして 6000 サイクル分の試合を実行した．図 6 に敵チームの静止位置を示す．

試合中，適宜手作業でボールの位置を変更し，なるべくたくさんのポイントからドリブルが始められるようにした．図 7 に，獲得されたドリブルの軌跡を示す．学習前のニューラルネットの重みはランダムに生成されるため，何も学習していない状態のニューラルネットを用いると，ゴール方向には向かわない．その一方で，学習済みのニューラルネットワークを用いた方向決めの場合，ゴールに向かってドリブルを進めている様子がわかる．また，ただ直線的に進むのではなく，静止している敵位置の近傍ではそれを避けるような方向が得られている．入力がボールの位置のみであるにもかかわらず，敵エージェントを避けることができるのは，学習用パターンの生成時に手本となるチームが静止している敵エージェントを避けながらドリブルしていたからである．学習したニューラルネッ

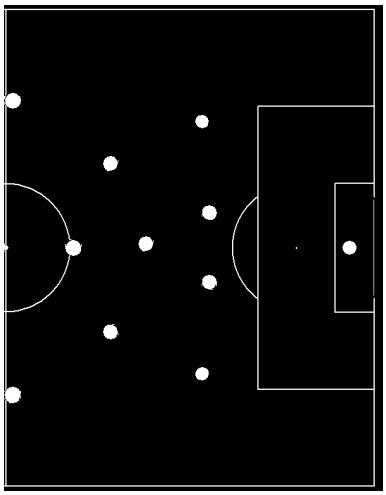


Figure 6: 敵チームの静止位置

トには，このような明確に含まれていない情報も考慮したうえで方向付けを行うことができる．

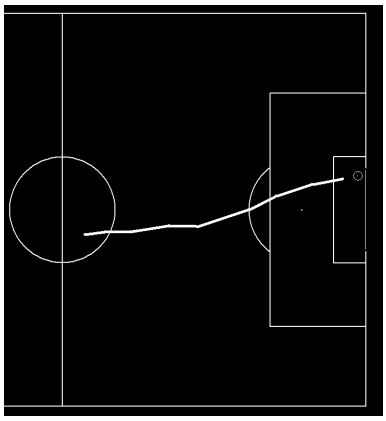


Figure 7: 開発したドリブルの軌跡 1

6 おわりに

本論文では，ニューラルネットとヒューリスティックを駆使したドリブルスキルの開発を行った．ドリブルスキルは個人技能の中でも重要な役割を占めており，優れたドリブルを行うことは，強いチームを構築するための絶対条件ともいえる．本論文では，サッカーサーバの仕様を領域知識としてヒューリスティックなキックパラメータの決定に用い，一方でチームの個性がよりよく出るドリブル方向の決定をニューラルネットを用いて行った．ニューラルネットの学習用パターン生成には，他のチームのゲームログから週出されたドリブル情報が用いられた．したがって，ボールを前に運ぶ技術は我々独自のものであるが，ボールを運ぶ方向については他チームを参考にしているとみなすことができる．本論文では，ニューラルネットの学習がいったん終了すればその後は変更していないが，ゲーム中に敵の動きに合わせて移動方向を適応的に変化させる技術も必要である．この技術については，近い将来の課題である．

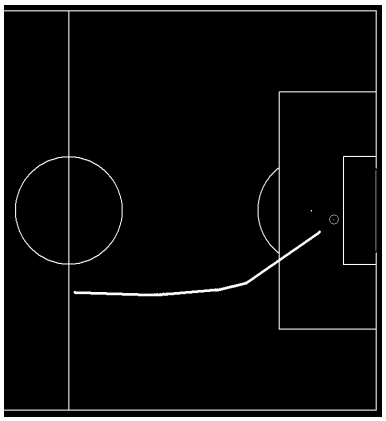


Figure 8: 開発したドリブルの軌跡 1

参考文献

[1] 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.

[2] 東京工業大学 TokyoTech05 download page, http://www.ntt.dis.titech.ac.jp/ akiyama/robocup/download.html

[3] 東京工業大学 TokyoTech06 download page, http://www.ntt.dis.titech.ac.jp/ akiyama/robocup/download.html

[4] アムステル大学 Uva_Trilearn2003 download page, http://staff.science.uva.nl/ jellekok/robocup/2003/trilearn_rc2003_bin.tar.gz

[5] 東京大学 YowAI 2006 download page, http://ssil.uni-koblenz.de/RC06/2D/Binaries/

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-A701-12 (5/4)

Incremental Behavior Acquisition Based on Reliability of Observed Behavior Recognition

Tomoki NISHI[1], **Yasutake TAKAHASHI**[1], **Minoru ASADA**[1,2]
[1]**Osaka University,** [2]**JST ERATO Asada Synergistic Intelligence Project**
{tomoki.nishi,yasutake,asada}@ams.eng.osaka-u.ac.jp

Abstract

We propose a novel approach for acquisition and development of behaviors through observation in multi-agent environment. Observed behaviors of others gives fruitful hints for a learner to find a new situation, a new behavior for the situation, necessary information for the behavior acquisition. RoboCup scenario gives us a good test-bed multi-agent environment where a learner can observe behaviors of others during practices or games. It is more realistic, practical, and e cient to take advantages of observation of skilled players than to discover new skills and necessary information only though the interaction of a learner and an environment. The learner automatically detects state variables and a goal of the behavior through the observation based on mutual information. Reinforcement learning method is applied to acquire the discovered behavior suited to the robot. Experiments under RoboCup MSL scenario shows the validity of the proposed method.

1 Introduction

Imitation is one of the most significant ability of a person. The young children increase their repertoire of behaviors and keep advancing it through the imitation of the observed behaviors with interest. Meltzoff et al. insists that the ability of imitation has borne important role for understanding intention and feelings of the other person.

In the environment involving robots and/or humans, it is less necessary to discover new tasks/behaviors only with self exploration than through observations of behaviors of others with skills. The latter is also much useful and realistic in practical view. Furthermore, it is a very useful ability to utilize understanding of intention of humans in order to generate cooperative behaviors with them.

Baldwin et al. [1] showed a fact that a young child of 10-11 months already has a segmentation ability of a behavior through some experiments. In addition, they insist that young children seem to have a low level model with continuity of trajectory of the body and segment the observed trajectory where it deviates from the model as the break of the behavior because of the fact that they can do the behavior segmentation even if the observation is the first time for them. Itti and Baldi [2] did the experiment regarding visual features which induce the gaze of a person. A vision image is divided spatially into small regions and temporally short periods and the feature quantity such as difference of color strength of red and green are modelled dynamically in regard to the each regions. Then, they showed that the human gaze is induced by the regions where the changes of parameter values of the model are large. From those insights, segmentation of an observed behavior seems to be done by detecting a remarkable point as a break point of the behavior where parameter values of the model change largely.

We propose a novel approach for incremental acquisition and development of behaviors through detection of remarkable points of observed behaviors. and apply it to our robots. A local linear model is introduced to check continuity of trajectory concerning each sensor value and

a point with a big error of this model is regarded as remarkable point for segmentation of the observed behavior. A new behavior learning module is assigned autonomously to a novel segmented behavior. Reinforcement learning method is applied to acquire the new behavior suited to the robot. Experiments under RoboCup MSL scenario show the validity of the proposed method.

2 Related Work

Research regarding imitation through observation has been done so far [3, 4, 5]. Almost conventional work focuses on e cient reproduction of observed behaviors by following trajectory of an observed behavior of a demonstrator. Those proposed imitation methods have applicability limitation as a trajectory of its imitated behavior becomes almost same of the instructed behavior because the imitated behavior is evaluated not by the intention of the behavior but by the similarity of the trajectory itself. The imitation with reproducing the observed trajectory is called mimicry as known as the most primitive imitation.

Expanded definition of imitation of the young child includes this mimicry, emulation, and narrow defined imitation. Emulation is when after observing an action, the observer jumps to conclusions and performs only those actions that will lead it to the goal, without caring about the exact methods of the demonstrator (although observed methods biases future actions). Finally, imitation is the crowning of copying, the sophisticated capability of reenacting sequence of actions to detailed levels, with the agent clearly aiming for the same objective as the demonstrator's.

Capability of emulation is useful for intention recognition because it is important to reproduce the result of the observed behavior but not about the exact trajectory of the observed behavior. It is unrealistic in the real world to acquire precise trajectory of an observed behavior because of the sensor/actuator noises or any possible differences in the parameters of body between the observer and the demonstrator and/or objects. Takahashi et al. [6] proposed a method of emulation that does not use similarity of the trajectory and does infer the intention based on the increase and decrease of achievement of the observed behavior. They showed the validity of the proposed method to infer the intention of other even if the trajectory of observed behavior is different from the one of own behavior of itself.

Reinforcement learning [7] has been studied well for motor skill learning and robot behavior acquisition. It generates not only an appropriate policy (map from sensor outputs to motor commands) to achieve a given task but also an estimated discounted sum of reward that will be received in future while a learning agent is taking an optimal policy. But it is known well that learning time and the required computational resources for a simple application to a real robot tends to be too huge and almost unpractical. One of the potential solutions might be application of so-called "mixture of experts" proposed by Jacobs et al. [8], in which a whole state space is decomposed to a number of areas so that each expert module can produce good performance in the assigned area, and one gating system weights the output of the each expert module for the final system output. This idea is very general and has a wide range of applications [9, 10, 11]. Therefore, emulation can be achieved based on reinforcement learning so that the observed behavior is divided into a number of modules and a reward is given when the result of behavior is reproduced. In general, it is a di cult problem to design appropriate combination of behaviors beforehand and it is desirable to be done autonomously by the observer itself.

Many kinds of modular learning systems with autonomous behavior segmentation mechanisms are proposed so far. Samejima et al. [4] arranged modules of a linear prediction model and a controller of reinforcement learning method as group in parallel, changed those assignment adaptively based on prediction error of the prediction models. Taniguti et al. [12] also proposed a system with a set of reinforcement learning modules in parallel that splits and merges among them based on the prediction error of reinforcement signal. In these systems, the state space, the space which describes the relationship between a learning agent and an environment, and a reward function have to be defined beforehand. Unfortunately, it is also di cult in general to design a state space and a reward function appropriately because it depends on not only behaviors the learner will observe and acquire in future but also the sensors and the motion mechanism equipped on it. We need some mechanism to find an appropriate state space and reward function automatically for each segmented behavior through observation of instructed behaviors.

3 Observed Behavior Segmentation based on Remarkable Points

3.1 Basic Idea

From the insights of Baldwin et al. [1] and Itti and Baldi [2], it seems to be possible to segment an observed behavior properly at remarkable points where parameter values of a simple model changes largely during the observation as a human tends to look at the points carefully. Since a trajectory of one single behavior tends to have stable direction and speed, then a local linear model can be applied to fit the trajectory. On the other hand, because linearity of a trajectory breaks when continuity of trajectory breaks, the remarkable point can be easily found by measuring change of reliability of the local linear model parameters. We call this measurement as "degree of attention" in this paper. In addition, an important space and a target state in order to explain the observed behavior can be found based on the mutual information between the remarkable points and the state in the space because a remarkable point is also a position of the target state of the behavior.

On the basis of argument above, our method,

1. finds remarkable points of observed behavior based on degree of attention,
2. segments observed behavior based on the remarkable points, and
3. assigns a behavior learning module to each segment of the observed behavior.

If observed behavior can be emulated by an appropriate module in a behavior repertory, then the degree of attention is suppressed so that acquisition of only novel behaviors for the learner can be focused on. An hierarchical learning system integrates a number of small time-scale behaviors so that the observer can emulate a long time-scale behavior. The hierarchical learning system has reinforcement learning modules that acquire purposive action policy through trial and error manner. Another advantage of this hierarchical learning system is e cient re-usability of behaviors learned before and enables the observer to keep learning new behaviors while it accumulates useful ones.

3.2 Algorithm Overview

A learner tries to emulate observed behavior and cumulatively acquire behaviors by the procedure below:

1. Observe behaviors of other,
2. Detect remarkable points in the observed behavior
3. If there is at least one remarkable points, then, go to next, else, go to 6.
4. Segment the observed behavior into smaller ones based on the remarkable points,
5. Find a state space and a goal state for the segmented behavior based on degree of attention.
6. If there are more than one observed behavior, then generate another learning module at higher layer to coordinate them.

We adopt a hierarchical learning system proposed by Takahashi et al. [9] Because of limited space of paper, we eliminate the details of the hierarchical learning system and concentrate on the acquisition of new behaviors from the observation.

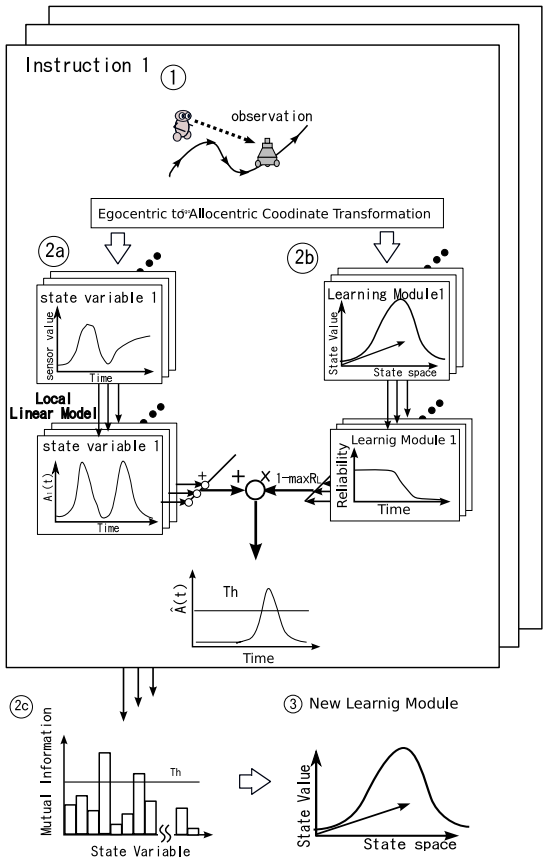


Figure 1: Sketch of Algorithm of New Behavior Detection

Fig. 1 shows a sketch of algorithm of new behavior detection and acquisition. From a number of observation data, degree of attention and reliabilities of existing behavior are calculated. The degree of attention suppressed by the reliabilities is used for selection of state

variables related to a new behavior acquisition based on mutual information. A reinforcement learning modules is assigned with a new generated state space with the selected state variables and a goal state, then, acquires the behavior through trials and errors. The details are explained in following sections.

3.3 Remarkable Point and Degree of Attention

As mentioned about the insights from the work of Baldwin et al. and Itti and Baldi before, we introduce a concept of degree of attention. The degree of attention should be large if the continuity of the trajectory breaks. We adopt a local linear model to detect a remarkable point of the continuity of the observed trajectory.

One dimensional local linear model $x = at + b$ is introduced and we assume that variation of measurement values follows normal distribution, then, variance of error of parameter is presumed as below.

$$\sigma_0^2 = \frac{1}{N \quad 2}(\sum x_i^2 \quad b\sum x_i \quad a\sum t_i x_i) \quad (1)$$

$$\sigma_a^2 = \frac{N}{N\sum t_i^2 \quad (\sum t_i)^2}\sigma_0^2 \quad (2)$$

$$\sigma_b^2 = \frac{\sum t_i^2}{N\sum t_i^2 \quad (\sum t_i)^2}\sigma_0^2 \quad (3)$$

where N is number of samples during observations. We define a reliability $R_m(t)$ of the parameter of the local linear model m at time t as follows:

$$\sigma^2(t) = \sqrt{(\sigma_a^2(t))^2 + (\sigma_b^2(t))^2} \quad (4)$$

$$R_m(t) = \begin{cases} \sigma^2(t) & \text{if} \quad \sigma^2(t) < 1 \\ & \text{else} \end{cases} \quad (5)$$

This reliability has a high value if the observed data has good linearity. Then we apply one dimensional local linear model on each state variable and define the degree of attention $A(t)$ at of time t as total sum of the changes of the reliabilities of all models. That is,

$$A(t) = \sum_{m \in M} |R_m(t+1) \quad R_m(t)| \quad . \quad (6)$$

We define the remarkable point where the degree of attention $A(t)$ is larger than a threshold k as the part with big change of the parameters of the all models.

3.4 Suppression of Degree of Attention

If a behavior has been already assigned before, then the assignment of a new behavior module should be suppressed even if the degree of attention is high. State value $V(t)$ which is utilized with reinforcement learning represents the closeness to a goal of the behavior. If a demonstrator follows to a policy of the behavior, the state value keeps rising, while it shows a movement in the opposite direction, then, the state value tends to decrease. Reliability that has higher value when the state value is rising and lower else is introduced here. The degree of attention is suppressed by this reliability (Fig.1 2b)

The reliability $R_l(t)$ of a learning module l at of time t is defined as follow:

$$R_l(t) = \frac{1}{1 + \exp(\quad k_1 e(t))} \quad (7)$$

$$e(t) \quad += \begin{cases} 0 & \text{if} \quad e(t \quad 1) > k_2 \\ & \text{or} \quad e(t \quad 1) < \quad k_2 \\ V(t) \quad V(t \quad 1) & \text{else,} \end{cases} \quad (8)$$

where k_1 and k_2 are a gradient factor of sigmoid function and maximum value of $e(t)$, respectively. Initial value of the reliability is 0.5, that is, $e(0) = 0$ in this paper.

This reliability $R_l(t)$ can evaluate how the observed behavior follows the policy of the module. In other words, if the reliability of this learning module is high, this means that the observed behavior has been already acquired in advance. Then degree of attention is suppressed as below on the basis of the reliability of the existing learning modules. The suppressed degree of attention $\hat{A}(t)$ is calculated as

$$\hat{A}(t) = (1 \quad \max_{l \in L} R_l(t))A(t) \quad , \quad (9)$$

where $\max_{l \in L} R_l(t)$ is maximum of reliabilities of all existing learning module acquired.

3.5 Selection of a state space and a goal state

Fig. 2 shows a diagram of selection of state space and a goal state for learning a new behavior based on the suppressed degree of attention $\hat{A}(t)$. The mutual information $I(X;Y)$ is information gain of the phenomenon Y when the phenomenon X is observed and shows depth of the relation of two phenomena. A new state space for an observed behavior is selected as the space with most mutual information gain between the phenomena "$\hat{A}(t)$ is higher than a threshold" and the one "The state s takes place in the state space S". Then, a new behavior module is assigned to the state space. The concrete procedure of selection of a state space and a goal state is shown below:

1. Create a histogram H_1 of appearance frequency of state visited through observed behaviors

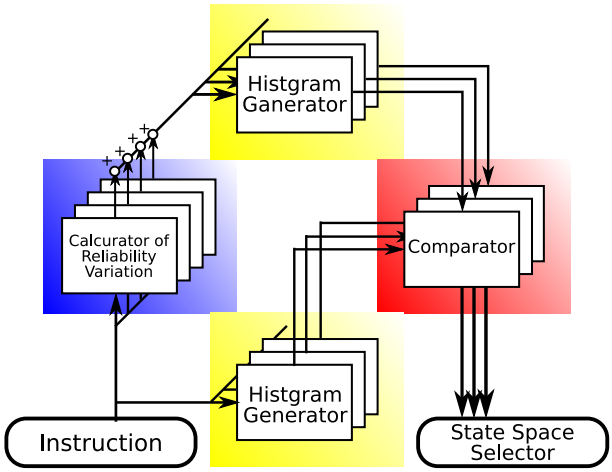


Figure 2: Sketch of a system which selects state space

2. Create a histogram H_2 of appearance frequency of state where the $\hat{A}(t)$ is larger than a threshold

3. Calculate the mutual information gain between phenomena "$\hat{A}(t)$ is higher than the threshold" and the one "the state s takes place in the state space S" based on the histogram H_1 and H_2

4. Assign a new behavior learning module with the state space and the goal state which appear most frequently in the histogram if the mutual information gain is larger than a threshold

4 Experiments

4.1 Experimental setup

The proposed method was verified with real robots under a scenario of RoboCup MSL. A robot has one omnidirectional vision system and detects objects around it in all direction simultaneously. It also has a omnidirectional vehicle to go to all direction and turn around on floor. Additionally, a kick mechanism is attached on the robots. There are a passer, who shows instruction behaviors, a receiver, a ball, two goals, and an observer (learner) in the environment.

4.2 Passing Behavior Observation and Acquisition

In our experiments, the learner observed a passing behavior 41 times from different viewpoints. 41 times is experimentally enough for the task. Fig. 3(a) and (b) show an example of the observation situation and a sequence of major sensor values during the observation respectively. Red, blue and the green lines in figure Fig. 3(b) indicate distance between the ball and the receiver, relative angle between the ball and the receiver from the viewpoint of the passer, distance between the ball and the passer, on the image of the observer, respectively. These values are normalized accordingly. From this figure, the passer starts dribbling from around 150th step toward the receiver and kicks a ball to the receiver at approximately 220th step. Then, the receiver received the ball at around 230th step.

Degree of attention $\hat{A}(t)$ is calculated through all 41 times observations. Fig. 4(a) shows a sequence of degree of attention during the observation of the instruction. $\hat{A}(t)$ is calculated with all observed behavior and the mutual information between each state variable and the region where the $\hat{A}(t)$ is over than a threshold 0.2 is graphed in Fig. 4(b). Two new behavior modules, LM1 and LM2, that have state spaces and goal states where are highly related with a space where $\hat{A}(t)$ is larger than a threshold based on mutual information is assigned. Table 1 shows the state space and the goal state. LM1 acquired a behavior of approaching a ball while LM2 acquired a behavior of turning around the ball and facing them in front of the body. Fig. 5(a) and (b) show examples of the acquired behaviors by learning modules LM1 and LM2. Whole pass behavior is acquired by integrating these behavior modules using a hierarchical reinforcement learning mechanism. One example of the acquired passing behavior is shown in Fig. 5(c).

4.3 Shooting Behavior Observation and Acquisition

As the second instruction, the learner observed shooing behavior 41 times, again. The relationship between a state space and a region which $\hat{A}(t)$ is larger than a threshold was calculated based on mutual information, again. Fig. 6(a) shows an example behavior acquired by a new module with the new state space. Fig. 7(a) and (b) show a sequence of degree of attention $\hat{A}(t)$ during the observation and the mutual information between each state variable and the region where the $\hat{A}(t)$ is over than a threshold, respectively. A behavior of approaching a ball is necessary for this observed behavior and this behavior has been already acquired as LM1 through passing behavior, then, LM1 does not need additional learning stage. A new behavior module LM3 is generated with an appropriate state space and a goal state shown in Table 1. The shooting behavior is acquired as the integration of LM1 and LM3 with hierarchical reinforcement learning mechanism. One example of the

Table 1: List of state variables and goal state in acquired learning modules

learning module	state variable	center of goal state
LM1	distance to ball	0.05
	angle to ball	0.00
LM2	angle between ball and receiver	0.00
	angle to ball	0.00
	distance to ball	arbitary
LM3	angle between ball and goal	0.00
	angle to a ball	0.00
	distance to a ball	arbitrary

(a) behavior of LM1 (b) behavior of LM2 (c) integrated behavior

Figure 5: Examples of acquired behaviors of LM1 and LM2 and integrated one

behavior is shown in Fig. 6(b).

5 Conclusion and Future work

We proposed a novel approach for acquisition and development of behaviors through observation in multi-agent environment. The learner automatically detected state variables and a goal of the behavior through the observation based on mutual information. Experiments under RoboCup MSL scenario showed the validity of the proposed method. The learner observed passing and shooting behaviors and tried to imitate them by incremental skill acquisition such as LM1, LM2, and LM3. Future work will investigate more number of typical behaviors like "obstacle avoidance", "receiving a ball", "interfering the opponents" and so on in RoboCup games.

参考文献

[1] Dare A. Baldwin, Jodie A. Baird, Megan M. Saylor, and M. Angela Clark. Infants parse dynamic action. , Vol. 72, No. 3, pp. 709–717, May/June 2001.

[2] Laurent Itti and Pierre Baldi. A principled approach to detecting surprising events in video. In , June 2005.

[3] Yuichiro Yoshikawa, Minoru Asada, and Koh Hosoda. View-based imitation learning by conflict resolution with epipolar geometry. In , pp. 1416–1427, 2001.

[4] Kazuyuki Samejima, Kenji Doya, and Mitsuo Kawato. Mosaic reinforcement learning architecture: Symbolization by p redictability and mimic learning by symbol. , Vol. 19, No. 5, pp. 551–556, 2001.

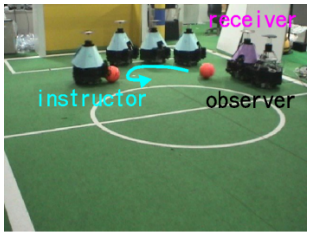
[5] Aude Billard and Maja J. Mataric. Learning human arm movements by imitation: evaluation of a biologically inspired connectionist architecture. In , Vol. 941, pp. 1–16, 2001.

[6] Y.Takahashi, Kawamata, and M.Asada. Learning utility for behavior acquisition and intention inference of other agent. In , pp. 25–31, 2006.

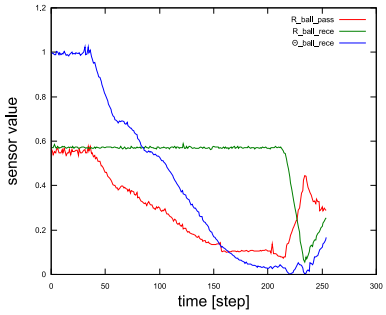
[7] R. Sutton and A. Barto. . MIT Press, 1998.

[8] R. Jacobs, M. Jordan, Nowlan S, and G. Hinton. Adaptive mixture of local experts. , Vol. 3, pp. 79–87, 1991.

[9] Y.Takahashi and M.Asada. Multi-controller fusion in multi-layerd reinforcement learning. In



(a)



(b)

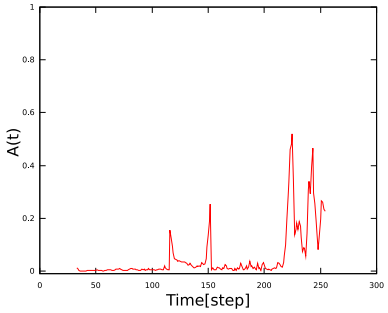
Figure 3: (a)Instruction of passing behavior, (b)Example sequence of the major sensor values ;((red line): distance between receiver and ball, (blue line): angle between receiver and ball, and (green line): distance between passer and ball)

, pp. 7–12, 2001.

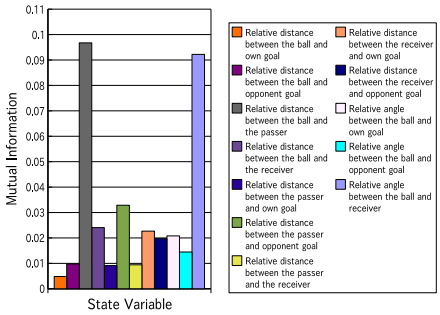
[10] Jun Morimoto and Kenji Doya. Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning. In , pp. 623–630, 2000.

[11] Satinder P. Singh. The e cient learning of multiple task. , Vol. 4, pp. 251–258, 1992.

[12] T. Taniguchi and T. Sawaragi. Incremental acquisition of behavioral concepts through social interactions with a caregiver. In , 2006.

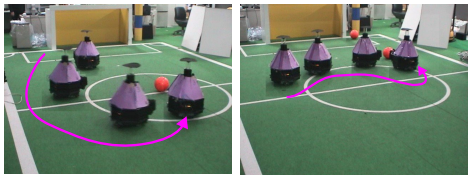


(a)



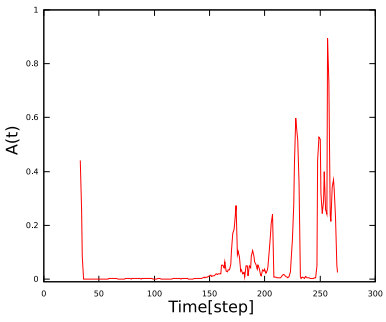
(b)

Figure 4: (a) Sequence of degree of attention during observation and (b) Mutual information in each state variable

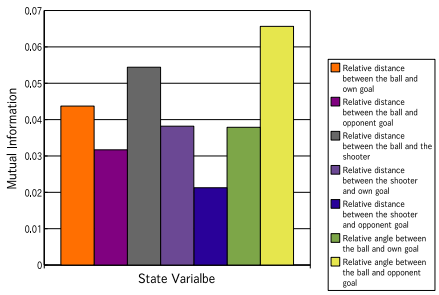


(a) behavior of LM3 (b) integrated behavior

Figure 6: Example of acquired behavior of LM3 and integrated one for shooting a ball



(a)



(b)

Figure 7: (a) Sequence of degree of attention during observation of shooting behavior and (b) Mutual information in each state variable

Q 学習を用いた制約付巡回セールスマン問題の解法

A Solution to the Traveling Salesman Problem with Additional Constraints Using Q-learning

○伏島 優　　(芝浦工業大学工学部)
五十嵐 治一 (芝浦工業大学工学部)
石原 聖司　(近畿大学工学部)

* Yutaka FUSEJIMA(SIT) , Harukazu IGARASHI(SIT), Seiji ISHIHARA (Kinki Univ.)

Abstract— Simulated annealing (SA) and Genetic Algorithm (GA) are known as quick approximate solution techniques for solving combinatorial optimization problems. They give optimal or nearly optimal solutions in rather short computation time. However, appropriately adjusting several parameters to avoid being trapped in local optimal solutions and wasting computation time is not easy. In this paper, we tried to learn the control of parameters in the objective function used in SA algorithms by Q-learning. We tested and verified our method by applying it to traveling salesman problems with additional constraints.

1.はじめに

組合せ最適化問題は実社会のいたるところに現れ，高速で最適解を導き出すことが重要課題となっている．そのため組合せ最適化問題の近似解法として，シミュレーテッド・アニーリング(SA)や遺伝的アルゴリズム(GA)等が考案された[1]．これらは比較的短時間で最適解，もしくは準最適解を得ることが可能であるが，パラメータの設定が必要であり，その設定を誤ると無駄な探索の増加や局所解に陥るという問題点がある．

本研究では Q 学習を用いて SA で使用する目的関数中のパラメータの調整を行い，得られる解の精度向上を目的とする．今回は例題として制約付きの巡回セールスマン問題をとりあげた．

2.制約付巡回セールスマン問題

一般に巡回セールスマン問題は，n 個の都市全てを一度ずつ巡り戻って来る経路の中から，総距離が最小になる経路を求める問題である[2]．これに訪問順の制約を加えた問題を「制約付巡回セールスマン問題」と本稿では定義する．今回は都市数を 20 とし，円周上に等間隔で配置し，さらに制約条件として訪問する都市間に先行関係を設けた．この制約条件は 7 つ設定した．Fig.1 に都市配置と先行関係を示す．

次に，この問題の解法のための定式化を行う．まず経路 l の総距離を $E_1(l)$，制約違反数を $E_2(l)$とし，経路 l の関数として目的関数 $E(l)$を次のように定義する．

$$E(l) \equiv E_1(l) + \omega E_2(l) \qquad (1)$$

(1)中の ω は制約項 $E_2(l)$の重み係数である．本論文では経路 l は 20 個の都市の訪問順序を表す 1 次元リストで表現する．$E_2(l)=0$ という制約を満たし(実行可能解)，総距離 $E_1(l)$ができるだけ小さくなるような経路(最適解)を見つけることが探索の目的である．

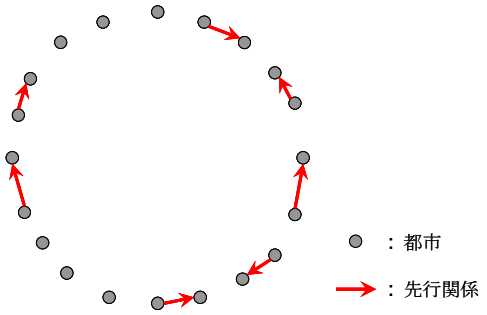


Fig.1　City arrangement and precedence relation

3.提案方式

3.1 処理フロー

SA は統計物理学における熱平衡状態を人工的に作り，徐々に温度を下げることで，問題に対して設定された目的関数の大局的最小点を見つける手法である．SA は比較的短時間で最適解，もしくは準最適解を得ることができるが，パラメータ(2.の例では ω)の値の設定次第では局所解や実行不可能解に陥りやすくなる．そこで本研究では，SA 実行中に目的関数中の重みを Q 学習により自動的に調整できる方式を考案した．本研究での解探索(SA+重み調整)の流れを Fig.2 に示す．Q 学習については次節で述べる．

まず，①の初期設定では初期温度 T_0 および初期重み係数 ω_0 の値を設定する．②の試行変形操作では，現在の状態 x（都市の訪問順序）からランダムに 2 つの都市を選択し，その 2 つの都市を入れ替えて次状態 x'を作る．③の受理判定では 2 つの都市を入れ

替える前の状態(x)と後の状態(x')を比較し，次状態 x' を受理するかどうかを判定する．今回は受理判定の方法としてメトロポリス法を用いた．メトロポリス法においては，現在の状態 x の目的関数値を E(x)，次状態の目的関数値を E(x')，その差を $\Delta E \equiv E(x') - E(x)$ としたときに，次状態 x'の受理確率 P(x,x')は以下のように表される．

$$P(x,x') = \begin{cases} 1 & \Delta E < 0 \\ \exp(-\dfrac{\Delta E}{T}) & \Delta E \geqq 0 \end{cases} \quad (2)$$

T は現在の温度である．

⑤の冷却判定では，あらかじめ設定された回数だけ試行変形，受理判定，状態遷移を繰り返したかどうかを判定し，温度 T を下げる(⑥，ただし $0 < \beta < 1.0$)．その際に制約項の重み係数 ω の調整も同時に行う．ω の変化量 $\Delta\omega$ は次節で述べる Q 学習により学習する．温度 T が一定の値より小さくなったら探索を終了する（⑦終了判定）．試行変形，受理判定，状態遷移，冷却判定，T および ω の更新操作をまとめて 1 つの step と呼ぶことにする．

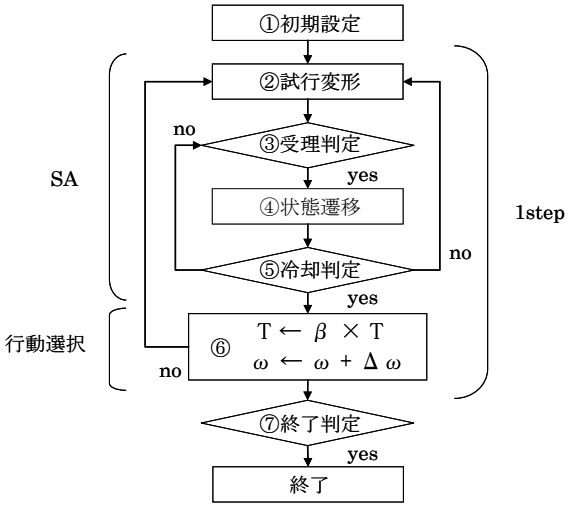


Fig.2　Flow of solution search

3.2 Q 学習における学習則

強化学習では，数値化された報酬信号を最大にするための行動決定の方法(方策)を試行錯誤を通して獲得する[3]．本研究では強化学習の手法として Q 学習を用いる．Q テーブルは，時刻 t において状態 s で行動 a を取り，以後方策 π に従った場合の期待報酬として次のように定義される．

$$Q_\pi(s,a) \equiv E_\pi\{R_t \mid s_t = s, a_t = a\} \quad (3)$$

R_t は割引報酬であり，次のように定義されている．

$$R_t \equiv \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \quad (4)$$

γ は割引率と呼ばれるパラメータ($0 \leq \gamma \leq 1$)である．割引率は将来の報酬が現在においてどれだけ価値があるかを決定する．Q 学習における Q 関数の更新式を以下に示す．

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a' \in A(s')} Q(s',a') - Q(s,a)\right] \quad (5)$$

Q(s',a')は遷移先の状態 s'において行動 a'をとるときの価値を示し，r は状態 s から s'への遷移において得られる報酬，A(s')は状態 s'で実行可能な行動全体の集合である．また α ($0 < \alpha \leq 1$)は学習率である．(5)の Q(s,a)は使用した方策によらずに，最終的に(3)の最大値

$$Q^*(s,a) = \max_\pi Q_\pi(s,a) \quad (6)$$

に収束する．最適方策は Q*(s,a)から決定論的に求めることができる．

4.重み係数の学習実験

4.1 状態と行動

3.2 の Q 学習を用いるには，状態 s と行動 a を定義する必要がある．今回の実験では，状態 s を現在の温度 T と重み係数 ω，したがって $s=(T,\omega)$ と表し，行動 a を $a=(a_1,a_2)$ と定義する．ただし a_1 は重み変化量 $\Delta\omega$，a_2 は温度変化量 ΔT であるが，a_2 はあらかじめ，操作：$T \leftarrow \beta \times T$ に固定し，a_1 のみを確率的に選択する．今回の実験では以下に示すボルツマン選択を使用した．

$$P(a \mid s) = \frac{\exp(Q(s,a)/\tau)}{\sum_a \exp(Q(s,a)/\tau)} \quad (7)$$

P(a|s)は状態 s で行動 a を取る確率である．実験では $\Delta\omega$ は，−50 から 50 の間で値を 5 刻みで離散化した−50，−45，−40 … 40，45，50 の計 21 種の中から(7)に従い決定する．

4.2 報酬

次に報酬 r を設定する．解探索終了時の最終状態を出力解とし，その解に報酬を与える．本研究の目的は出力解の精度向上なので，出力解が実行可能解で($E_2=0$)，最適解($E_1=260$)に近いほど高報酬を与えることにした．報酬の具体的な値を Fig.3 に示す．

実行可能解($E_2(l)=0$)	報酬 r
$E_1(l) = 260$ (最適解)	200
$260 < E_1(l) \leqq 280$	100
$E_1(l) > 280$	0
実行不可能解($E_2(l) \neq 0$)	−100

Fig.3　Reward

4.3 実験条件

学習は計 20 万回行った．初期温度 T_0=500.0 と設定し，初期重み係数 ω_0 は 0，10，20，30 の中からランダムに決定する．1step 中の試行変形回数は 500 回，冷却速度 β=0.8 とし，終了条件は T<1.0 とした．これにより，解探索 1 回あたりのステップ数は 29 となる．ただし，29 ステップ目は終了状態であるため，解の探索は行われない．よって学習 1 回における総試行変形回数は 500×28=14000 回である．ボルツマン選択時のパラメータ τ は，τ=5.0 に設定した．Q テーブルはすべて 0.0 で初期化し，割引率は γ=0.9，学習係数 α=0.1 とした．なお，コンピュータは Pentium(R) 4 CPU 3.4GHz，メモリは 2GB のものを使用し，20 万回の学習に要する時間は約 2 時間であった．

4.4 学習実験の結果

学習 1000 回ごとに得られる報酬 r の平均値 $\overline{r_{1000}}$ の推移を Fig.4 に示す．Fig.4 より，学習が進むにつれて $\overline{r_{1000}}$ の値が増加しており，学習の有効性がわかる．

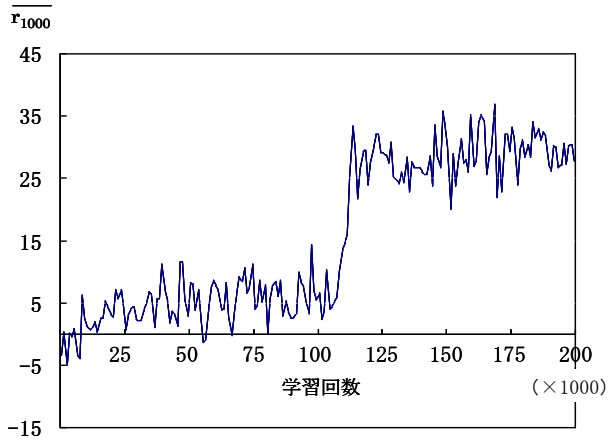


Fig.4　Change of $\overline{r_{1000}}$

5.評価実験

5.1 学習回数ごとの評価実験

4.の学習実験より得た Q 値を用いて，Fig.2 の流れに沿って評価実験を行い，学習回数 2 万 5 千回ごとに性能を比較した．評価実験に使用した実験条件は 4.3 に述べたものと同じであるが，行動は各状態 s において Q(s,a)を max とする行動 a を選択させるようにした．出力解も学習実験とは異なり，探索中に得た実行可能解の中で $E_1(l)$が最も小さいものとした．探索は各々1000 回行い，その中で発見した実行可能解の E の平均値 $\overline{E_{fea}}$，実行不可能解の出力回数 N_{inf} をカウントした．結果を Fig.5 に示す．

5.2 ωを手動で設定した場合の評価実験

ω を固定した従来の SA の性能評価実験を行った．SA では初期 ω を ω=10，15，20，25，30，35 に設定し，各々解探索を行う．探索回数，初期温度 T_0，初期重み係数 ω_0，試行変形の回数，冷却速度 β，終了条件，出力解は 5.1 で使用したものと同じになるよう設定した．結果を Fig.6 に示す．

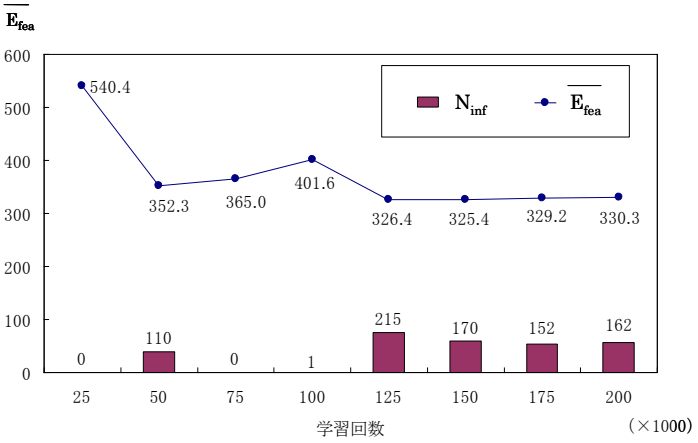


Fig.5　Performance comparison

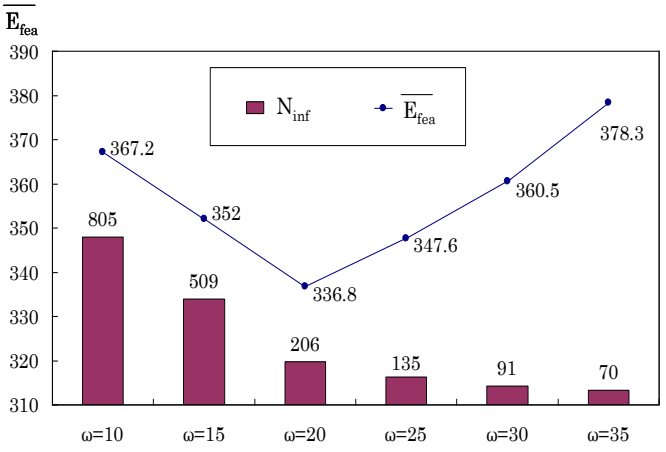


Fig.6　Results of SA when ω is fixed

5.3 ωの推移

20 万回学習後の Q 値を用いたときの，ある解探索中の ω の挙動を観察した．初期重み係数 ω_0 の値を 30 に設定したときの推移の例を Fig.7 に示す．横軸はステップ数 n(1≦n≦29)である．

ω の変化の仕方としては，探索開始直後の n=2 で ω=0 まで下げ，その後 ω=0 を保ち，探索終盤（24≦n≦26）に ω=35 に増加後，終了間際（27≦n≦29）で ω を 75 まで急増させている．この推移の傾向は初期重み係数 ω_0 の値を 0，10，20 に設定しても同じであった．

次に上記の解探索の各ステップごとに，取った状態 s における学習のエントロピー S_{ent} を計算し，行動決定の不確定さを調べた．その結果を Fig.8 に示す．各状態 s での学習のエントロピー $S_{ent}(s)$は次式により求めている．

$$S_{ent}(s) = -\sum_{a} P(a \mid s) \log_{10} P(a \mid s) \qquad (8)$$

P(a|s)は(7)で与えられ，エントロピーの計算の際は τ=0.5 と， τ の値は低く設定した．

Fig.8 より高温では $S_{ent}(s)$が高く，低温になるにつれ $S_{ent}(s)$が低くなっていくことがわかる．$S_{ent}(s)$が高いということは，状態 s において Q(s,a)の差があまりないことを表している．実際に 20 万回学習した後の Q テーブルの値を観察すると，高温時の各状態では，どの行動に対しても Q 値はほぼ同じであった．評価実験では，各状態において Q(s,a)を max とするような行動 a を選択させるようにしたため，Fig.7 のようにω=0 となったが，実際は高温時のωの値は必ず 0 にする必要はなく，行動選択にボルツマン選択を用いると，ωの値が高くなることも観測された．

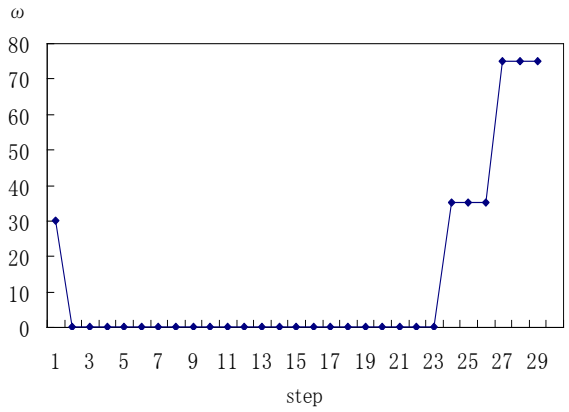


Fig.7　Change of ω

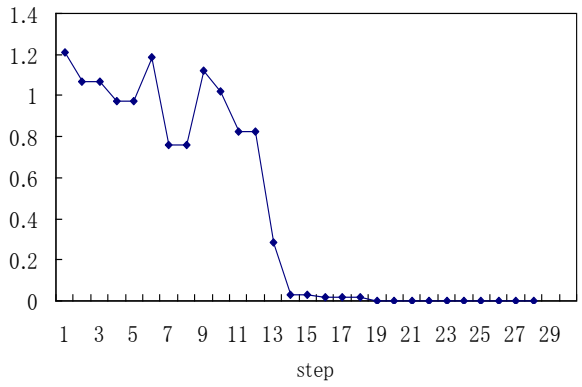


Fig.8　Change of entropy

7.考察

Fig.6 からわかるように，SA の場合，ωが小さすぎると制約を重視せず，実行不可能解が多く出現してしまう．そのためなかなか実行可能解を見つけられず，出力解は悪くなる．一方，ωが大きすぎると実行不可能解の出力回数は減るが，局所解に陥りやすくなるため出力解は悪くなる傾向がある．今回は，Fig.6 よりωを固定した場合の SA の中で実行可能解の E の平均値が一番良いのはω=20 のときで，$\overline{E_{fea}}$= 336.8 であった．これと同程度の性能は，学習 12.5 万回以降で得ることが可能であった(Fig.5)．さらに実行不可能解の出力回数は，ωを固定した場合の SA が N_{inf}=206 であるのに対し，20 万回学習後では N_{inf} =162 と，実行不可能解の出力回数をωを固定した場合の SA よりも約 20%抑えることができた．

5.3 のωの挙動の仕方については，次のように考えている．まず探索序盤(1≦n≦13)では，ω=0 まで下げているが，Fig.8 より高温時はωの値はあまり重要ではない．これは高温の場合，ωの値によらず局所解から抜け出すことが容易なためと考えられる．

次に探索中盤(14≦n≦23)であるが，学習のエントロピー$S_{ent}(s)$がかなり低いことから，確実にωを 0 にしようとする行動を選択していることがわかる．これはω=0 として制約違反を気にせずに広く探索したほうが良いと学習したためと考えられる．ただしω=0 のままでは制約違反を考慮していないため，実行不可能解を出力してしまう可能性が高い．よってωの値をどこかで上げる必要があるが，そのタイミングが n=24 であると考えられる．その後さらにωを 75 まで急上昇させているが，これについては，制約違反を完全になくすように学習したためであると考えられる．実際，実験中に得られた最適解もしくは準最適解は step24～26 で求まっていることが多く，それ以降のステップでは，その解から実行不可能解への遷移をしないようにωを大きくしたのではないかと思われる．

8.まとめ

本研究では SA で使用する目的関数中のパラメータωの調整法を学習し，ωを固定した場合の SA よりも実行不可能解の出力回数を約 20%減らすことができた．しかし学習後の解の精度が格段に良くなっているわけではない．そこで今後はさらに出力解の精度をあげることを目指したい．その一つの方針として，今回はωのみを学習し，温度は常に一定の割合で下げていったが，今後は温度変化の学習も行うことを考えたい．特に温度を上げることはレプリカ交換モンテカルロ法でも行われている[4]．この手法を取り入れることを検討したい．

また，状態 s の定義についても温度 T と制約項の重み係数ωだけでは不十分である可能性がある．そのため，目的関数 E のゆらぎを状態量として考慮するなど再検討が必要であると考えている．

参考文献

[1]白石洋一：組合せアルゴリズムの最新手法，丸善株式会社，pp.43-161 (2002).
[2]柳浦睦憲，茨城俊秀：組合せ最適化－メタ戦略を中心として－，朝倉書店 (2004).
[3]三上貞芳，皆川雅章 訳，Richard S.Sutton 他著：強化学習，森北出版，pp159-161(2000).
[4]伊庭幸人 他：計算統計Ⅱ，岩波書店，pp.74-78 (2005).

方策勾配法を用いたサッカーエージェントの学習
～フリーキック時の壁パス～
Learning of Soccer Player Agents Using Policy Gradient Methods
~Wall Pass after Free Kicks~

○福岡 仁志 (芝浦工業大学工学部)
中村 浩二 (芝浦工業大学工学部)
五十嵐 治一 (芝浦工業大学工学部)
石原 聖司 (近畿大学工学部)

* Hitoshi FUKUOKA(SIT), Koji NAKAMURA(SIT), Harukazu IGARASHI(SIT),
Seji ISHIHARA(Kinki Univ.)

Abstract— The RoboCup Simulation League is a test bed for research of multi-agent learning. As an example, we dealt with a learning problem between a kicker and a receiver when a direct free kick is awarded just outside the opponent's penalty area. In such a situation, where should the kicker kick the ball? We previously proposed a function that expresses heuristics that evaluate how advantageous a target point is for sending/receiving a pass safely and contributing to scoring. The evaluation function makes it possible to handle a large space of states consisting of the following positions: kicker, receiver, and a pair of opponents. A target point of a free kick is selected by the kicker using Boltzmann selection with an evaluation function. Parameters in the function can be learned by a kind of reinforcement learning called the policy-gradient method. The point to which a receiver should run to receive the ball is simultaneously learned in the same manner. In this paper, we raise the learning of the passing action between kicker and receiver at free kicks to a more intelligent level. As one advanced tactic, we adopted and realized the learning of a wall pass between two players immediately after free kicks.

1. はじめに

近年，人工知能の分野ではマルチエージェント環境下での協調行動，実時間処理，不完全知覚といった複雑な問題が研究されている[1]．これらの問題を含んだ研究材料としてロボットサッカーの競技会であるRoboCupが提唱されている[2]．これまでに著者らは，ゴール前でのフリーキックの場面においてpasserとreceiverによる協調行動の学習という研究を行ってきた[3][4]．これは，フリーキックという限られた場面ではあるが，状態数によらない方策表現を用いた強化学習法を採用することにより，精度の高い協調行動の実現を目指したものであった．しかし，この研究では２人のプレーヤー（passerとreceiver）が行動決定を同時刻に各々１回のみ(passerのパス先決定, receiverの移動先決定)を行うというものに留まっていた．

そこで本研究では，フリーキックを行った後に，さらにパス行動を複数回行うという協調行動の拡張を目指した．これにより，フリーキックからの壁パスの実現や，実際のゲームでのスムーズなパスワークの実現へと繋がることを期待している．

2. 学習方式

2.1 方策勾配法の適用

本研究では強化学習の一種である方策勾配法を用いて学習を行う．方策勾配法とは，報酬の期待値が最大になるように方策パラメータを更新する学習法である．このときの最大化の手段として確率的勾配法を用いる．方策勾配法は数学的な基礎がはっきりしており，理論的に取り扱いやすい．また，方策として if-then 型のルールや，ポテンシャルなどの様々な関数が利用できるので，方策への知識表現が容易であるという長所がある．元々は，Williams により提案された手法[5]であるが，著者らも追跡問題やカーリングゲームに適用し，有効性を確認している[6][7].

2.2 目的関数

本研究では，次章で述べるフリーキックの問題のために，以下のような行動決定方式を用いる[3][4].まず，一般的なパス行動を考える．フィールドを格子状の長方形セルの集合に区切る．セル k へパスを出す/移動するという行動 a_k の価値を次のような目的関数で表す.

$$E(a_k;\omega) = -\sum_i \omega_i \cdot U_i(a_k) \qquad (1)$$

この関数は，パス先のセル k を決める上で有効と思われる状態の特徴量（ヒューリスティクス）U_i の線形和である．ただし，目的関数 E の値が小さい方が行動としての価値が高くなるように設計する．また，重み ω_i は学習により決定する.

式(1)では敵プレーヤーの数や配置，味方プレーヤーの位置など，その場面に応じて各セルへのパスを出す行動の価値を計算しており，プレー中の環境の変化も考慮している．勿論，広大な状態空間の保存を必要としていない.

2.3 確率的な方策

この目的関数を用いて，プレーヤーは次のボルツ

マン選択による確率的な方策を用いてパス先/移動先のセル k を決定する．

$$\pi(a_k;s)\equiv\frac{e^{-E(a_k;s)/T}}{\sum_x e^{-E(x;s)/T}} \quad (2)$$

ボルツマン選択は温度パラメータ T を大きくするほどランダムに行動を選択するようになり，小さくするほど最も大きい価値の行動を選択しやすくなる．特に，T→0 では決定論的な行動決定となる．なお，s は全系の状態（i.e. 全プレーヤー，ボールの位置）を表している．

2.4 学習則

一般に，マルチエージェント系全体の状態を s，行動を a とする．それぞれ，各エージェントの状態と行動とを要素とするベクトルであることに注意する必要がある．行動 a に対する目的関数を E(a;s,θ)とし，方策が式(2)のようなボルツマン選択である場合に，そのまま方策勾配法を適用すると，パラメータ θ に関する学習則は，

$$\Delta\theta=\varepsilon\cdot r\sum_{t=0}^{L-1}e_\theta(t) \quad (3)$$

で表される[6]. ε(>0)は学習係数，r はエピソード終了時に与えられる報酬，L はエピソード長である．$e_\theta(t)$ は，離散時刻 t における特徴的適正度(characteristic eligibility)で，次の式で定義されている．

$$e_\theta(t)\equiv\frac{\partial}{\partial\theta}\ln\pi(a;s,\theta) \quad (4)$$

2.5 自律分散的な行動決定と学習

次に，行動決定と学習とを自律分散的に行うことを考える．そこで，系全体の方策を，エージェント i ごとの方策関数 π_i の積で近似する[6]．すなわち，

$$\pi(a;s,\theta)\approx\prod_i\pi_i(a_i;s,\{\theta_{ij}\}) \quad (5)$$

と仮定する．ここで，a_i はエージェント i の行動であり，θ_{ij} は π_i に含まれる j 番目のパラメータである．さらに，各 π_i は，各エージェントに定義される目的関数 $E_i(a_i;s,\{\theta_{ij}\})$を用いたボルツマン分布とする．したがって，式(2)中の右辺の目的関数 E は，厳密にはエージェントごとに定義された目的関数 E_i である．

なお，(5)の近似は，エージェント間の行動選択の相関を無視していることに相当している．また，他のプレーヤーの位置情報も分かっているという前提に立っているので，各 π_i には系全体の状態 s が使用されている．したがって，各エージェントは味方や敵の行動選択とは無関係に行動を選択するが，彼らの存在（位置情報）までを無視しているわけではないので，このような近似を行っても協調行動を学習することはある程度までは可能と考えられる．

一方，報酬 r の期待値 E[r]をパラメータについて微分すると，形式的には，

$$\frac{\partial E[r]}{\partial\theta_{ij}}=E\left[r\sum_{t=0}^{L-1}e_{\theta_{ij}}(t)\right] \quad (6)$$

となる．(5)の近似を用いて(6)の右辺の適正度を計算すると，

$$e_{\theta_{ij}}(t)\equiv\frac{\partial}{\partial\theta_{ij}}\ln\pi(a;s) \quad (7)$$

$$\approx\sum_i\frac{\partial}{\partial\theta_{ij}}\ln\pi_i(a_i;s,\theta_{ij}) \quad (8)$$

さらに，各エージェントの目的関数中のパラメータは互いに独立であると仮定すると(7),(8)は，

$$e_{\theta_{ij}}(t)\approx-\frac{1}{T}\left[\frac{\partial E_i}{\partial\theta_{ij}}-\left\langle\frac{\partial E_i}{\partial\theta_{ij}}\right\rangle\right] \quad (9)$$

と表される．なお，<--->は(2)の分布による期待値操作である．一方，パラメータ θ_{ij} の学習則は(3)の導出と同様に，(6)を用いて確率的勾配法を適用すると，

$$\Delta\theta_{ij}=\varepsilon\cdot r\sum_{t=0}^{L-1}e_{\theta_{ij}}(t) \quad (10)$$

となる．ただし，(5)の近似により(10)の右辺の適正度は(9)で与えられる．(3)が系全体の目的関数 E(a;s)に含まれるパラメータに関する学習則であるのに対し，(10)はエージェント i が自己の行動 a_i を選択するために使用する目的関数 E_i の中に含まれるパラメータの学習則であり，かつ，自己の目的関数 E_i だけを用いている点で，自律分散的な学習であると言える．

以上より，本稿における式(1)のパラメータ ω_i の学習則は(9)，(10)より，状態 s において実際に選択された行動 a_k に対して，

$$\Delta\omega_i=r\cdot\varepsilon\cdot\frac{1}{T}\left(U_i(a_k)-\sum_a U_i(a)\cdot\pi(a;s)\right) \quad (11)$$

となる．

3. 対象とする課題

3.1 フィールドの分割

実験では図１に示すようにペナルティエリア付近を一辺５m の正方形セルで４×８に分割したフィールドを用いる．さらに均等にゴールも３分割する．これにより，「ゴールにパスをする」ことが「シュートをする」ことを実現している．

使用するエージェントと配置場所を以下に，その配置例を図１に示す．なお，SoccerServer ではフィールドの中心（センターマーク）を原点とし，タッチラインに平行に X 軸，ゴールラインに平行に Y 軸をとる．プレーヤーは３種類で，味方プレーヤーが２人，敵チームのディフェンダー１人とゴールキーパー１人である．また，便宜上味方プレーヤーは最初にパスを行うものをＡ，最初にレシーブを行うものをＢとする．

A を配置する際に配置場所の X 座標はゴールラインから 24m の位置で固定，Y 座標はランダムな位置とする．配置場所はオフサイドポジションにならないランダムな位置とする．

敵ディフェンダーはゴールを決めさせないためにパスカットやプレスを行う．配置場所はゴール３セルを除くセルの中でランダムな位置とする．ゴールキーパーはゴールエリア内のランダムな位置に配置する．

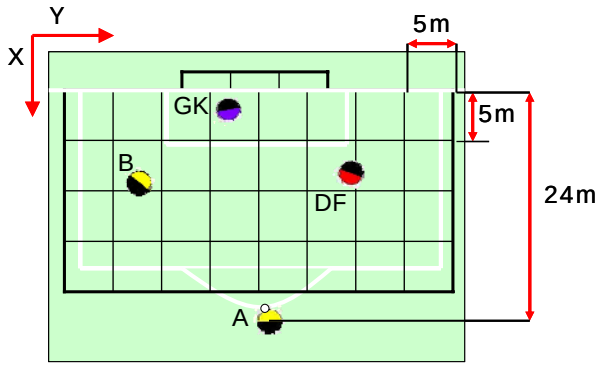


Fig. 1 Arrangement of players

プレーヤーの他に trainer というエージェントも使用する．trainer はプレイモードを kick_off から free_kick に変更したり，プレーヤーとボールを移動したりすることができる．そして，フィールドを監視し，パスは通ったのか，シュートは決まったのかをプレーヤーに伝えることができる．

3.2 本研究におけるエージェントの学習

本研究では２人のエージェントA, Bがフリーキックの後に複数回のパスを行うという協調行動を考える．複数回の行動決定を行うために，エージェントごとに，かつ，行動決定ごとにそれぞれ目的関数(1)を用意する．エージェントの行動決定としては，パス先の決定と移動先の決定(ポジショニング)の２つを考える．

また，複数回の行動決定を含むあるエピソードを定義し，エピソード終了時にプレーに対して報酬を与える．そして(1)のω_iを(11)の学習則により学習する．

4. 実験

4.1 処理の流れ

エピソード中の処理の流れを図２に示す．なお，行動決定の機会をA1~A4，B1~B2で表す．まず，passerであるプレーヤーAはフリーキック前にパス先を決定する(A1)．フリーキック後，Aは移動先を決定(A2：ポジショニング)してそこへ移動し，決定した２つの行動内容（A1とA2）をBへ通知する．receiverであるBはその情報に基づいて移動して(B1)ボールを受け取った後はpasserとなり，逆にAはreceiverへと役割を変更する．次に，新たにpasser となったBはパス先を決定して(B2)パスを行い，A に行動内容を通知する．Aはその情報を基に移動しボールを受け取るが，その際Bは移動先の決定と移動を行わず，パスを受けたAは必ずシュートを行うものとする(A4)．今回，これらの行動決定のうちでA1，A2，B2を学習の対象とした．

なお，１エピソードを70シミュレーションサイクル(7sec)とし，１エピソード終了前にゴールを決めるか70シミュレーションサイクルが過ぎると，エピソードはそこで終了とし，エピソード終了時に報酬を与える．報酬の与え方については4.4で述べる．

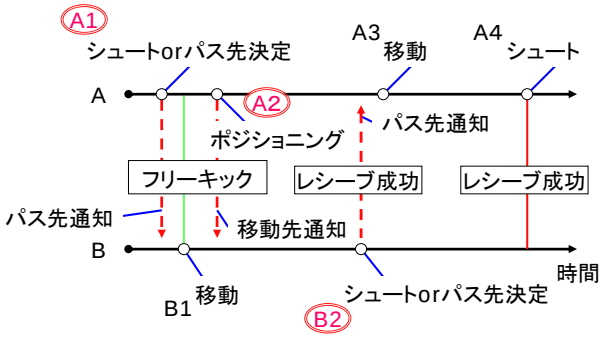


Fig.2 Actions of players A and B

4.2 実験条件

複数回の行動に対して学習をする前準備として，A1 のみを学習する予備的な学習実験を行った．この結果を本実験において，A1 の重みの初期値として使用することで，学習時間の短縮ができる．予備実験の際のパラメータは T=10.0, ε=0.04 とし 2000 エピソード学習する．

次に本実験を行う．ここで学習が期待されるのはA→B→A とパスが渡り A4 でシュートを行うまでの，いわゆる壁パスが通った後にシュートを行う協調プレーである．実験の際のパラメータは T=10.0, ε=0.04 とし，2000 エピソード学習する．なお，予備実験，本実験共に対戦相手として Trilearn Base を使用した[8]．Trilearn Base とは 2003 年に優勝したアムステルダム大学チーム UvA Trilearn 2003 から高度な行動決定や戦略を除いてソースコード形式で配布されているチームである．よって，UvA Trilearn 2003 よりは弱くなっているが最低限の行動は保証されている．

4.3 ヒューリスティクスの設計

実験では， パス行動 A1，B2 においてゴール前でのフリーキックに有効と思われる 1)～4)の４つのヒューリスティクス U_1~U_4 と，ポジショニング A2 において有効と思われる 5)～10)の６つのヒューリスティクス U_5~U_{10} を用いた．なお，どの関数 U_i も攻撃側にとって値が大きいほど価値があるとする．

まず，パス先の決定において使用するヒューリスティクス U_1~U_4 を以下に示す．

1) パスコースにおける敵の有無：

$$U_1 = \begin{cases} 10.0 & \text{敵がいない場合} \\ 2.0 & \text{敵がいる場合} \end{cases} \qquad (12)$$

実際にパスをするにあたって，パスコースに敵が

いない方がパスは通りやすい．図３a のようにパスコースに敵がいないセル（α の範囲内のセル）がパス先であれば 10，敵周辺と敵の後方のセル（β の範囲内のセル）がパス先であれば 2 を返す．

2) パス先とゴールとの距離：

$$U_2 = \left(-\left(X_G + Y_G\right) + 37.5\right)/3.5 \qquad (13)$$

パス先とゴールとの距離が近い方が，パスを受け取った時にゴールに結びつきやすい．式(13)の X_G と Y_G は図３b のようにそれぞれパス先とゴールとの距離ベクトルの X 成分，Y 成分である．なお，式(13)は正規化してあり，取りうる範囲は 0.5～10.0 である．

3) パス先と味方との距離：

$$U_3 = \left(-\left(X_r + Y_r\right) + 50.0\right)/5.0 \qquad (14)$$

パス先と味方(receiver)との距離が近いほど，パスは通りやすい．式(14)の X_r と Y_r は図３c のようにそれぞれパス先と味方との距離ベクトルの X 成分，Y 成分である．なお，式(14)は正規化してあり，取りうる範囲は 0.0～10.0 である．

4) パス先と最近接の敵との距離：

$$U_4 = \left(X_O + Y_O\right)/5.0 \qquad (15)$$

パス先と敵との距離が遠いほど，そのパス先にはスペースがあるということである．つまり，フリーでシュートを打ちやすくなる．式(15)の X_O と Y_O は図３d のようにそれぞれパス先と最近接の敵との距離ベクトルの X 成分，Y 成分である．なお，式(15)は正規化してあり，取りうる範囲は 0.0～10.0 である．

次に，移動先の決定（ポジショニング）において使用するヒューリスティクス U_5~U_{10} を以下に示す．

5)移動先とパス先との距離：

$$U_5 = \left(X_n + Y_n\right)/5.0 \qquad (16)$$

移動先とパス先が近いと，連続パスなどのプレーを効果的に行いにくくなるため，遠い方がよい．式(16)の X_n と Y_n は図３eに示すように，移動先とパス先との距離ベクトルのX成分，Y成分である．なお，式(16)は正規化してあり，取りうる範囲は0.0～10.0である．

6)移動先とゴールとの距離：

$$U_6 = \left(-\left(X_G + Y_G\right) + 37.5\right)/3.5 \qquad (17)$$

2)の(13)と同じ式で目的も同様だが，「移動先」とゴールの間の距離である点が異なる．なお，式(17)は正規化してあり，取りうる範囲は0.5～10.0である．

7)移動先と味方との距離：

$$U_7 = \left(X_r + Y_r\right)/5.0 \qquad (18)$$

移動先と味方との距離が近いと，ボールを交換するだけの短いパスといった無駄なパスに繋がりやすくなるため遠い方がよい．式(18)の X_r と Y_r は，それぞれ「移動先」と味方との距離ベクトルのX成分，Y成分であり，3)と考え方が似ているが，距離が遠い方がよいとする点，パス先ではなく「移動先」と味方との距離としている点が異なる．式(18)は正規化してあり，取りうる範囲は0.0～10.0である．

8)移動先と最近接の敵との距離：

$$U_8 = \left(X_O + Y_O\right)/5.0 \qquad (19)$$

「移動先」と敵との距離が遠い方がパスをうけるのに望ましい．式(19)は4)の(15)と同じ式で目的も同様だが，「移動先」とパス先の間の距離である点が異なる．式(19)は正規化してあり，取りうる範囲は0.0～10.0である．

9)移動先と現在位置との距離：

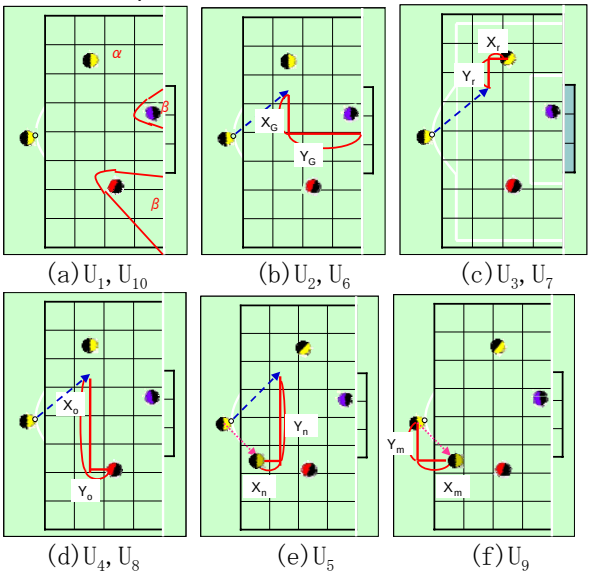
$$U_9 = \left(-\left(X_m + Y_m\right) + 5.0\right)/5.95 \qquad (20)$$

移動先と現在位置との距離が遠いと，移動が完了する前に味方がパスを出してしまい，パス先にたどり着けない可能性が高くなってしまう．式(20)の X_m と Y_m は図３fのように移動先と現在位置との距離ベクトルのX成分，Y成分である．なお，式(20)は正規化してあり，取りうる範囲は0.0～10.0である．

10)移動先とパス先の間の敵の有無：

$$U_{10} = \begin{cases} 10.0 & \text{敵がいない場合} \\ 2.0 & \text{敵がいる場合} \end{cases} \qquad (21)$$

パスが通った先からパスが行われると考えられるので，パス先と移動先の間に敵がいない方がパスは通りやすい，1)と同様に選択したセルがα内のセルであれば10，β内のセルであれば２を返す．



(a) U_1, U_{10}　(b) U_2, U_6　(c) U_3, U_7

(d) U_4, U_8　(e) U_5　(f) U_9

Fig.3 Heuristics$\{U_i\}$(i=1~10)

4.4　報酬

4.2 の学習では，行動決定 A1，A2，B2 ごとに目的関数を用意する．プレーヤーA は A1，A2 の，プレーヤーB は B2 の目的関数を各行動決定に用いる．したがって，重み係数も別々に$\{\omega_{A1}\}$，$\{\omega_{A2}\}$，$\{\omega_{B2}\}$と

用意する．また，それぞれの重み係数の学習に対し，学習則(11)において与える報酬をそれぞれ r_{A1}，r_{B2}，r_{A2} とし，1 エピソード終了時に図 4 のように設定した．それぞれの重みに与えられる報酬のパターンとしては 6 種類あるが，行動決定の段階(本稿の図 2 における A1~B2，B2~A4，A4 以降の 3 段階)と結果に応じていずれか 1 種類の報酬が与えられる．

A1のパス成功

	r_{A1}	r_{B2}	r_{A2}
A1でのパス失敗	−20		
直接シュート成功	60		

B2のパス成功

	r_{A1}	r_{B2}	r_{A2}
B2でのパス失敗	0.5	−20	−20
B2でシュート成功	80	80	

	r_{A1}	r_{B2}	r_{A2}
A4でのシュート失敗	2	2	2
A4でシュート成功	100	100	100

Fig.4 Rewards r_{A1},r_{B2},and r_{A2}

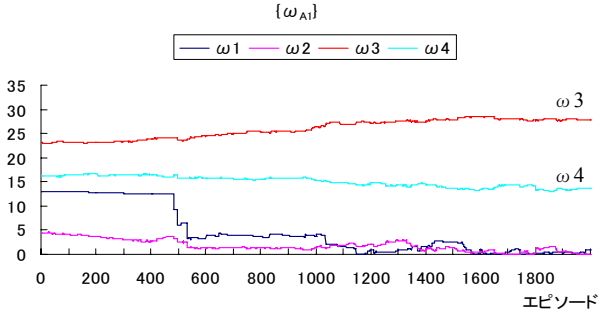
4.5 学習実験

4.2 の条件で学習実験を行った．実験中の重みの変化を図 5 に，学習実験中の A1，A2，B2 の報酬の期待値 $\overline{r_{A1}}$，$\overline{r_{A2}}$，$\overline{r_{B2}}$ を図 6 に示す．

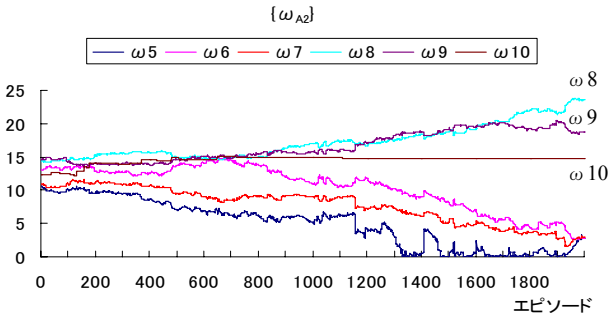
図 5 a より，行動決定 A 1 でパス先を決定する際に，味方の近くのセルを重要視する U_3 の重み ω3 と敵から遠いセルを重要視する U_4 の重み ω4 の割合が大きい．したがって，最初に A が行うパス(A1)では味方の近くで敵から遠い場所，つまり B が安全にパスを受けることのできる場所へ蹴ることを学習している．

同様にして，図 5 b より，行動決定 A2 で移動先を決定（ポジショニング）を行う先を決定する際に，敵から遠いセルで(U_8)，現在位置から近く(U_9)，かつ，A1 で決定したパス先の地点からパスコースのあいているセル(U_{10})を選択することが分かる．つまりパスカットの行われにくい近くの場所へと移動することを学習している．

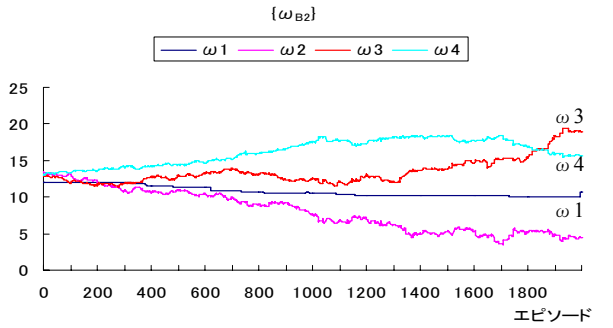
図 5 c からは，行動決定 B2 では ω3 と ω4 が大きくなっており，それぞれ味方の近くのセルと敵から遠いセルを重要視していることが分かる．したがって，B は A が安全にパスを受けることのできる場所へパスすることを学習している．



(a)Action A1

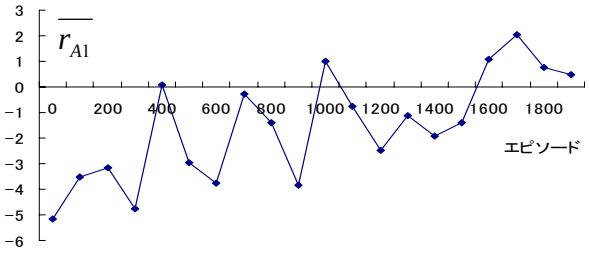


(b)Action A2

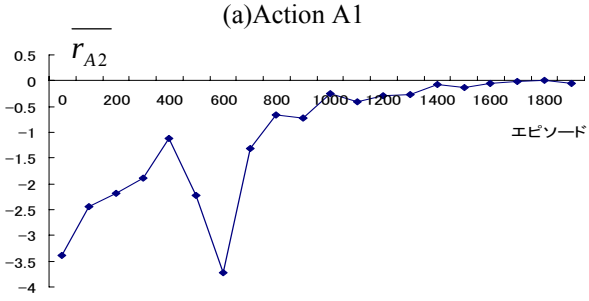


(c)Action B2

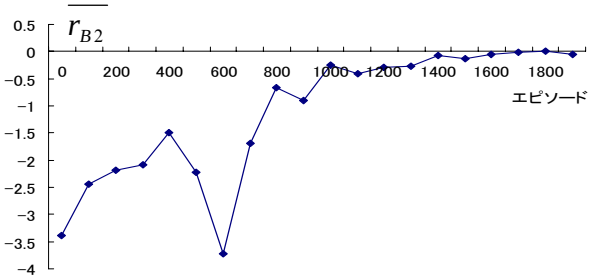
Fig.5 Change of weights $\{\omega_{A1}\}$,$\{\omega_{A2}\}$,and $\{\omega_{B2}\}$



(a)Action A1



(b)Action A2



(c)Action B2

Fig.6 Expected value of rewards $\overline{r_{A1}}$，$\overline{r_{A2}}$，and $\overline{r_{B2}}$

4.6 評価実験

学習実験途中に得られた 100 回ごとの重み{ω_{A1}}, {ω_{A2}}, {ω_{B2}}を使用し，各々500 エピソードだけプレーを行わせた．実験を行う際の温度パラメータ T は 0.1 と低く設定した．評価実験で得た A1，B2 におけるパス成功率，A1，B2，A4 のシュート成功率を図 7 に，A1，B2，A4 におけるシュート試行回数を図 8 に示す．

図 7 と図 8 を見ると，学習回数が 1400 回以降では，A4 でシュートを行う回数が増えているにもかかわらず，試行回数あたりの得点率が減少傾向にある．これは，壁パスが成功しても，シュートを行う場所が悪く，GK や DF にカットされてしまうためと考えられる．そこで図 9 に 2000 エピソード学習した後の B2 におけるパス先の割合を示した．なお，図 9 のセルの位置関係は図 1 のセルと対応している．

図 9 を見ると，B2 でパスを出す際，ゴールに向かって右 45° と左 45° の最後方のセルにパスを出す割合が非常に高いことが分かる．これでは壁パスが通ってもバックパスに近い安全パスといったものになり，その場でシュート(A4)を行っても，シュートが決まらない確率は当然高くなると考えられる．

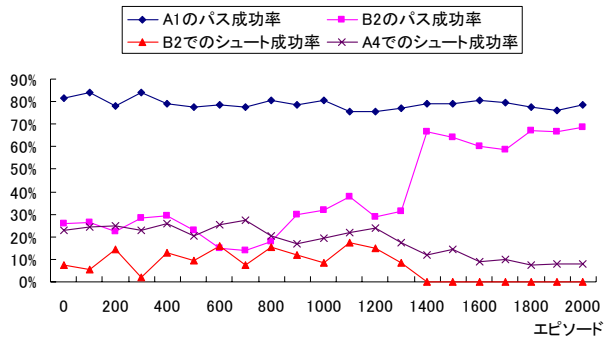


Fig.7 Pass/Goal Success rate

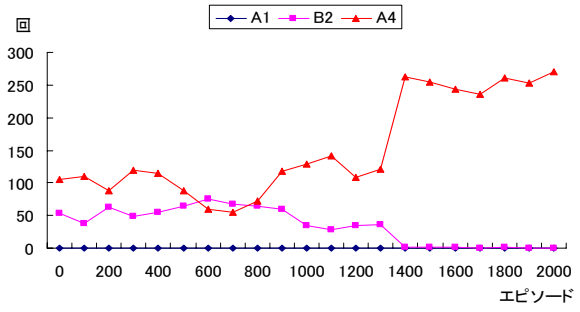


Fig.8 Number of shots

Fig.9 Target of action B2

5. まとめ

本研究では，複数パス交換へと協調行動を拡張することを目的とし，壁パスの実現には関して学習の有効性を確認することができた．しかし，安全なパスを目指すあまりに得点率の向上は達成できなかった．得点率の向上をさせるためには，報酬の設計を再検討する必要がある．例えば，バックパスを行う場合は，パス成功に対する報酬を減少させ，安全パスを行うよりは途中(本稿でのA1，B2)でシュートを行わせることが考えられる．しかし，バックパスのことを考慮する際，必ずしもバックパスを行うことが悪いとは限らないため，報酬の設計は慎重にしなければならないであろう．また，得点するための個人技術の向上も必要である．

また，著者らはフリーキックに限らず，本研究で示した方策勾配法による学習方式をフルゲームのボールキープの関数に適用し，MF と DF のボールキープ力を高める学習実験を行い，学習の有効性を確認している．今後もフルゲームの各種の協調プレーへと適用し，それらを結合することを考えていきたい．

参考文献

[1] 高玉圭樹 : マルチエージェント学習－相互作用の謎に迫る－，コロナ社(2003)

[2] RoboCup Official Site 〈http://www.robocup.org/〉

[3] 中村浩二，五十嵐治一，石原聖司：“方策勾配法を用いたサッカーエージェントの学習～フリーキックにおけるキッカーとレシーバ～”，第23 回 SIGchallenge研究会予稿集,pp.7-12(2006)

[4] 中村浩二，五十嵐治一，石原聖司：“方策勾配法を用いたサッカーエージェントの学習～フリーキック時のキッカーとレシーバの相互作用～”，数理モデル化と問題解決シンポジウム論文集(情報処理学会シンポジウムシリーズ, Vol.2006, No.10, ISSN 1344-0640),pp.119-126（2006.10.24-25, 名古屋）

[5] Williams, R.J. : Simple Statistical Gradient- Following Algorithms for Connectionist Reinforcement Learning, Machine Learning, Vol.8, pp.229-256 (1992)

[6] 石原聖司，五十嵐治一 : マルチエージェント系における行動学習への方策こう配法の適用－追跡問題－，電子情報通信学会論文誌 D-I，Vol.J87-D-I, No.3，pp.390-397(2004)

[7] 五十嵐治一，石原聖司，野原 勉：方策勾配法を用いた運動方程式中のパラメータ学習~２ストーン系のカーリングゲーム~，ロボティクス・メカトロニクス講演会’05(ROBOMEC’05)講演論文集, 1A1-N-028(pp.1-4)(2005.6.10-11，神戸，主催：日本機械学会ロボティクス・メカトロニクス部門)

[8] The Universiteit van Amsterdam 〈http://staff.science.uva.nl/~jellekok/robocup/〉

背景色を利用したマーカ色抽出と全方位移動型ロボットの制御
～RoboCup 小型リーグ～
Marker Color Extraction Using Background Color and Control of an Omnidirectional Robot
～RoboCup Small-Size League～

○長谷川 卓也　（芝浦工業大学工学部）
脇本 耕平　（芝浦工業大学工学部）
五十嵐 治一　（芝浦工業大学工学部）
田中 一基　（近畿大学工学部）

*Takuya HASEGAWA(SIT), Kohei WAKIMOTO(SIT), Harukazu IGARASHI(SIT),
Kazumoto TANAKA(Kinki Univ.)

Abstract—In this paper, we describe two improvements of our robot system made for competing in the RoboCup Small-Size League. First, we propose a color-recognition method that can be applied under the non-uniform lighting conditions observed in actual games. Our proposal's basic idea uses the value of green lightness as a key for searching a database to find proper thresholds in the HLS color space, because the carpet surrounding the robot is green. Second, we propose a method for controlling omnidirectional moving robots based on a binary search for an appropriate value of argument ω in moving command MOVE. The command MOVE has three arguments incleding argument ω, which specifies the robot's rotational velocity. Our experiments demonstrated the effectiveness of employing these two methods.

1. はじめに

マルチエージェントシステムにおける制御や学習と，自律移動型ロボットシステムの評価を行う試みの一つとして，ロボットサッカー競技会 RoboCup が 1997 年から開催されている[1]．競技部門はいくつかに分かれているが，小型リーグ部門はグローバルビジョンカメラの使用が許されているのが特徴であり，かつ，実機を使用する部門の中では情報処理とロボット移動とにおいてスピードを最も必要とされている．初期の頃は，スピードを重視する余り，制御に正確性を欠き，衝突や暴走する場面がしばしば見られたが，最近では画像処理や制御の精度が上がり，サッカー競技らしい試合が多く見受けられるようになった．これに伴い，研究の重点も，自然環境に近い照明条件下におけるロバストなマーカ認識（＝ロボットの位置と姿勢の認識），協調行動や戦略レベルでの制御・学習へとより高度なものに徐々に移行しているようである．

我々は，このような小型リーグのロボットシステムを構築する際に生じた，以下の 2 つのテーマを対象として研究を行ってきた[2]．第 1 のテーマは，マーカ認識におけるロバストな色抽出であり，第 2 のテーマは，本ロボットシステムでも使用している全方位移動型ロボットの正確な走行制御である．第 1 のテーマについては，背景色（＝床面の緑色）の HLS 値をキーとする閾値データベースを用いる方式をこれまで提案してきた．第 2 のテーマについては，ロボットの走行特性を強化学習の代表的な手法である Q 学習を用いて学習する方法を提案し，直進走行という簡単な事例について適用してきた．本研究では、第 1 の色抽出については従来方式に対して閾値データベースの検索キーとマーカ判定方式に改良を加えた．第 2 の走行制御に関してはロボットの姿勢制御のために適切なコマンドパラメータを探索することを試みた．

2. ロボットシステムの全体構成

本システムではベースとして，近畿大学で開発されたロボットシステム KU-Boxes++[3]を用いている．システム構成を Fig.1 に示す．

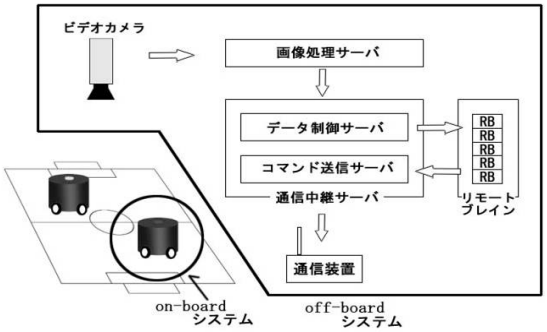


Fig.1 System overview of KU-Boxes++

このシステムは，実際にフィールド上で試合を行う移動ロボット（on- board システム）と，それを制御するシステム（off-board システム）とに分かれている. 後者の off-board システムは, 画像処理サーバ,

通信中継サーバ，リモートブレインから構成されている．小型リーグの規約では，フィールド上方に設置したビデオカメラによるグローバルビジョンの使用が許されており，ロボット上面に貼り付けたマーカを認識することにより，5 台のロボットの識別と位置・姿勢情報を取得することが可能である．

3. 背景色を利用した色抽出方式

Fig.2 に 2005 年に開催された世界大会の小型リーグで用いられた競技用フィールド上での照明の状態の一例を示す．このようにフィールド上の明るさは一様ではなく，はっきりとした影の領域も存在する．したがって，マーカ認識においては，実環境での明るさの変動の影響を受けて，色を正しく認識できないことがある．また，通常，各マーカの認識に使われる閾値の設定が競技前にしか行われないために，競技中での明るさの時間的変動に対応できないといった問題点もある．すなわち，空間的・時間的に照明が変動する環境下でも正しくマーカの色を抽出することが必要になってくる．本研究では前者の空間的に照明が変動する環境下でのマーカ色の抽出に焦点を絞ることにする．

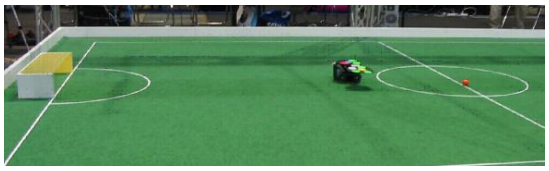


Fig.2 Robot field used in RoboCup 2005

そこで，フィールド内の異なる複数の場所で HLS 表色系（H：色相，L：明度，S：彩度）における閾値を設定し，データベースとして保存しておく．これを閾値テーブルと称する．さらに，マーカの背景色（ここでは床面の緑色）の HLS 値の平均値を保存しておき，色抽出時には閾値テーブルからの検索時にキーとして利用する[2].

4. マーカ認識

4.1 全体の流れ

本ロボットシステムにおけるマーカ配置例を Fig.3 に示す．このマーカの認識処理は，①色抽出，②マーカ判定③，ID・姿勢検出からなる．①ではチームカラーの青・黄，ボール色の橙に加えてさらにサブマーカの色である白の計 4 色の色抽出を対象とする．①,②についてそれぞれ 4.2 と 4.3 で簡単に説明する．

4.2 色抽出

9 分割したフィールド(1.8m×2.1m)の領域毎に抽出すべき色と緑(床面)の計 5 色の閾値データベースを作成する．これまで色抽出時には緑の HLS 値をデータベースの検索キーとして使用していた[2]が，今回は明るさの変動には最も敏感である輝度 L の値のみを検索キーとして用いることにより精度向上と計算量の削減を図った．

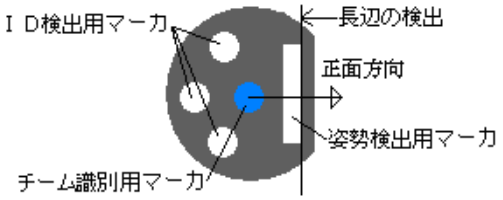


Fig.3 Example of placing markers

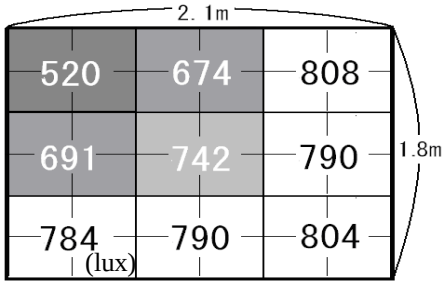


Fig.4 Illuminance of the robot field used for our color extraction experiment

4.3 マーカ判定方式

これまでは，色抽出の後二値化により 2×2 ドット以上のラベル領域をマーカとして認識していた[2]. しかし，白色を検出対象に加えると複数色を持つ画素が出現しやすいため正常な認識ができない事があった．また画像のにじみによってマーカの輪郭から別の色を認識する場合もあった．

そこで，本方式では複数色として抽出された画素を含むラベル領域に対しても，そのラベル領域の重心付近の画素を調べて最も抽出頻度の高い色を判定結果とする方式を考案した．

4.4 色抽出実験

Fig.4 のフィールドを 36 分割し，各領域内に各色 1 つずつマーカを設置し，以下の 4 方式(1)～(4)で 1000 回の色抽出を行った．その結果を Table1～4 に示す．

(1)従来方式[2]：閾値データベースの検索キーとして背景色である緑の HLS 値すべてを用い，かつ 2×2 ドット以上のラベル領域を単にマーカと判定する方式.

(2)本方式 1：従来方式から閾値データベースの検索キーを L 値のみとしたもの（4.2 で説明）.

(3)本方式 2：従来方式からラベル領域の重心付近の画素を用いてマーカの判定を行うようにしたもの（4.3 で説明）.

(4)本方式 1+2：従来方式から検索キー，マーカ判定方式の両方を改良したもの.

Table1～4 に示すとおり，黄の抽出は従来方式において白との誤認識が多かった．しかし，Table2～4 から，本方式のどちらも有効であり，約 21%あった白との誤認識を最終的には約 1%まで抑えることができた．白の抽出については本方式 2 が有効であり，認識率が約 5 ポイント上昇し，青については本方式 1 が有効であり，認識率が約 4 ポイント上昇した．橙については，従来方式でも本方式でも誤認識は殆ど発生しておらず，2 つの改良による悪影響は無かった．

Table1 Recognition rate by method in the past

(%)	正認識	抽出失敗	重複認識	誤認識			
				青	黄	橙	白
青	95.25	3.03	1.71		0.00	0.00	0.00
黄	79.38	0.02	0.03	0.00		0.00	20.58
橙	99.60	0.38	0.01	0.00	0.01		0.00
白	94.40	0.00	0.00	0.00	5.60	0.00	

Table2 Recognition rate by Present method 1

(%)	正認識	抽出失敗	重複認識	誤認識			
				青	黄	橙	白
青	99.33	0.41	0.26		0.00	0.00	0.00
黄	92.99	0.00	0.02	0.00		0.00	6.99
橙	100.00	0.00	0.00	0.00	0.00		0.00
白	94.57	0.00	0.00	0.00	5.43	0.00	

Table3 Recognition rate by Present method 2

(%)	正認識	抽出失敗	重複認識	誤認識			
				青	黄	橙	白
青	95.25	3.03	1.71		0.00	0.00	0.00
黄	95.93	0.00	0.04	0.00		0.00	4.02
橙	99.11	0.04	0.76	0.00	0.09		0.01
白	99.14	0.00	0.00	0.00	0.86	0.00	

Table4 Recognition rate by Present method 1+2

(%)	正認識	抽出失敗	重複認識	誤認識			
				青	黄	橙	白
青	99.33	0.41	0.26		0.00	0.00	0.00
黄	98.95	0.00	0.00	0.00		0.00	1.05
橙	100.00	0.00	0.00	0.00	0.00		0.00
白	99.60	0.00	0.00	0.00	0.40	0.00	

5. 走行制御

本ロボットシステムでは移動型ロボットとして，ロボス社製のオムニ Robo-E を使用している．オムニ Robo-E はオムニホイールと呼ばれる特殊な車輪を 4 個用いることにより，姿勢を保持したまま全方位移動が可能である．しかし，オムニホイールは多数の小さな車輪を周辺に取り付けた車輪であり，すべりや引っかかりなどのため床面の材質等に走行特性が依存し，フィードフォワード的な制御だけでは正確な走行制御は難しいとされている．

したがって，実際の走行時にはグローバルビジョンからの視覚情報を利用したフィードバック制御による走行方向の補正処理を欠かすことができない．我々は，このグローバルビジョンによる補正処理の負担を軽減するために，その走行環境におけるオムニ Robo-E の走行特性を学習し，学習結果をより正確な走行制御に利用することができないかと考えた．

5.1 MOVE コマンドによる走行例

制御にはロボス社から提供された MOVE コマンドを使用する．MOVE コマンドは進行方向 φ (0～360[degree])，速度 v (0～1800 [mm/s])，回転速度 ω (-128～127 [degree/s])の 3 つの引数を指定する．φ はロボットの正面方向を基準とし，ω とともに反時計回りに大きくなっていく．

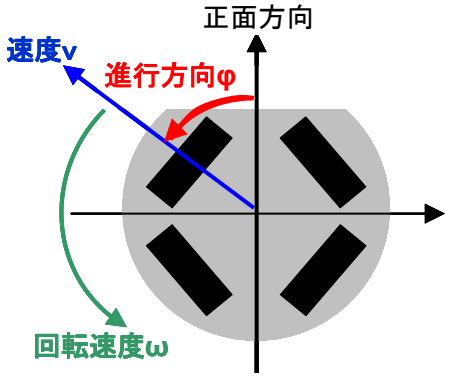


Fig.5 Parameters of command MOVE

これまで著者らは 3 つの引数のうち，φ の値のみを Q 学習により学習することで，指定した目的地への直進性を改善することができた[2]．しかし，このときの回転速度 ω は 0 で固定としていたにも関わらず，実際の動作では回転が生じていた．そこで本研究では回転速度 ω に対する補正値を自動的に推定し，その推定値を用いることにより姿勢制御することを試みた．

まず，Fig.6 にロボットの動作の実測値を示す．ただし，v =500，ω =0 と固定し，φ を 0～330° まで 30° ごとに指定した場合の走行結果の軌道が示されている．なお，ロボットの走行時間は 2 秒間で，500ms ごとの位置が示されている．この走行実験においては ω =0 と固定しているため，理想的には指定した φ の方向に姿勢を保持したまま平行移動するはずである．しかし，実際には図のように，φ =0，φ =180 といった前後の動作の場合はその方向へほぼ直進しているが，それ以外の φ の値を指定した場合にはずれが生じ，軌道が変形していることがわかる．また，姿勢も保持されていない．

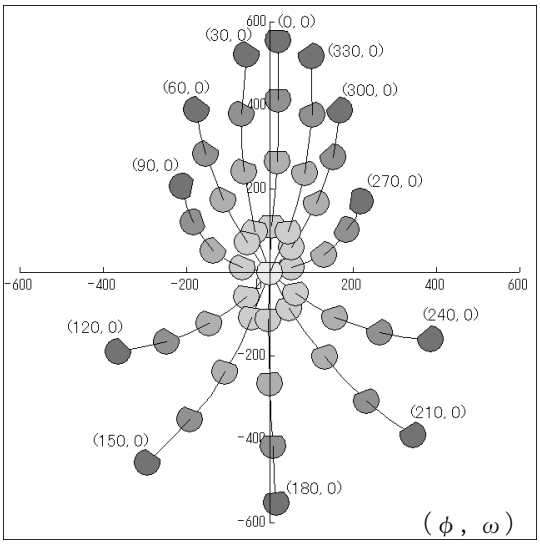


Fig.6 Robot traces by command MOVE where ω=0

5.2 回転速度ωの補正

MOVE コマンドの引数(v,ϕ,ω)を指定した場合，ロボットは常に正面方向から一定方向ϕに進もうとする．これに加えて自転が発生すると，合成された結果は円運動となる．Fig.6 にはこのような現象が見られる．しかし，逆に走行中の自転を打ち消すようなωの値を指定すればロボットは直進するのではないかと考えた． そこで，1 つの MOVE コマンドを送ってから 2 秒間姿勢を保持し，回転が生じないようにωの値を補正する．ϕの値を 0～330° まで 30° ずつ変化させていき，それぞれのϕに対して目標の姿勢（ここでは初期姿勢）となるようなωの値を 2 分探索法により求めた．求めたωの補正値ω'を MOVE コマンドに指定し，実際に 2 秒間走行させた結果を示したのが Fig.7 である．Fig.7 を見ると補正値ω'を用いたことにより，ロボットがほぼ直進するようになったことがわかる．

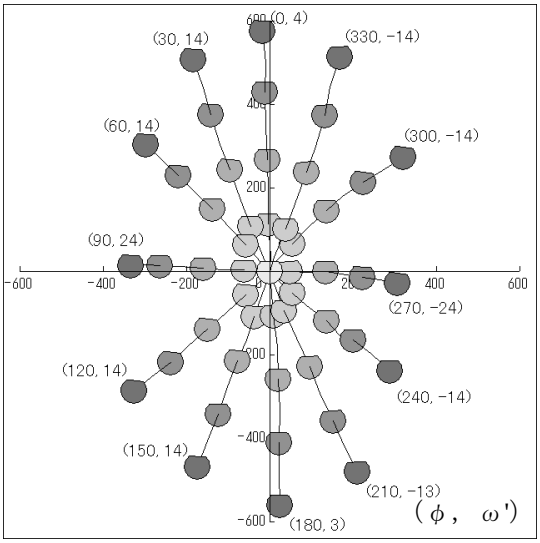


Fig.7 Robot traces using the corrected values ω's

また，Fig.6，Fig.7 における動作の最終時刻（移動開始から 2 秒後）における平行移動時の姿勢のずれを Fig.8 に示す．Fig.8 では，横軸は MOVE コマンドで指定したϕの値，縦軸は初期姿勢からのずれである．ω=0 と固定した場合，-100° ～+80° の大きな姿勢のずれが生じているが，補正値ω'を用いたことにより，-5° ～5° に抑えることが出来た．

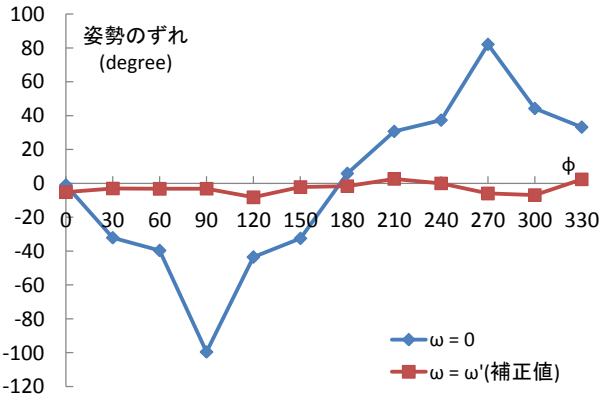


Fig.8 Comparison of gap of posture at translation

6. 今後の課題と展開

試合中など時間的に照明が変動する環境下での安定した色抽出と ID・姿勢検出についてのアルゴリズムについては検討中である．

今後は，色抽出のための閾値設定の自動化や，ロボットの走行コマンドの引数値の自動調整，移動障害物の存在する環境下での経路計画，さらには音声認識と自然言語処理を用いたコーチシステムの構築を目指す．

参考文献

[1]RoboCup Official Site <http://www.robocup.org/>

[2] 五十嵐治一，梁島昇弘，吉田翔，村木俊幸，岩男卓哉，“背景色を利用したマーカー色抽出と Q 学習による全方位移動型ロボットの制御～RoboCup 小型リーグ～”，ROBOMEC2006,2P1-B05(2006.5.26-28, 東京).

[3]三好威士他，“RoboCup 小型リーグ向けロボットシステムの構築”平成 15 年度近畿大学工学部卒業論文.

[4]五十嵐治一：“強化学習を用いた自立移動型ロボットの行動計画方の提案”電子情報通信学会論文誌 D-I, Vol.J84-D-I, No.3, pp.294-3-2(2001).

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

(本研究会についてのお問い合わせは下記にお願いします.)

AI チャレンジ研究会

主 査

奥乃 博

京都大学大学院 情報学研究科
知能情報学専攻 音声メディア分野
〒 606-8501 京都市左京区吉田本町
Tel: 075-753-5376 Fax: 075-753-5977
okuno@i.kyoto-u.ac.jp

担 当 幹 事

浅田 稔

大阪大学大学院 工学研究科
知能・機能創成工学専攻 先導的融合工学講座
〒 565-0871 大阪府吹田市山田丘 2-1
Tel: 06-6879-7349 Fax: 06-6879-7348 /
JST ERATO 浅田共創知能システムプロジェクト
asada@ams.eng.osaka-u.ac.jp

担 当 幹 事

光永 法明

(株) 国際電気通信基礎技術研究所
知能ロボティクス研究所
〒 619-0288 京都府相楽郡精華町光台 2-2-2
Tel: 0774-95-1401 Fax: 0774-95-1408
mitunaga@atr.jp

幹 事

中臺 一博

(株) ホンダ・リサーチ・インスティチュート・ジャパン
〒 351-0114 埼玉県和光市本町 8-1
Tel: 048-462-5219 Fax: 048-462-5221 /
東京工業大学大学院 情報理工学研究科
nakadai@jp.honda-ri.com

Executive Committee

Chair

Hiroshi G. Okuno

Dept. of Intelligence Sci. and Tech.,
Graduate School of Informatics,
Kyoto University
Yoshida-honmachi Sakyo-ku,
Kyoto, 606-8501, JAPAN

Secretary in Charge

Minoru Asada

Dept. of Adaptive Machine Systems,
Graduate School of Engineering,
Osaka University
2-1 Yamada-oka, Suita,
Osaka 565-0871, JAPAN /
JST ERATO Asada Synergistic Intelligence Project

Secretary in Charge

Noriaki Mitsunaga

Dept. of Intelligent Robotics and Communication Laboratories,
Advanced Telecommunications Research Institute International
2-2-2 Hikaridai, Seika-cho, Souraku,
Kyoto 619-0288, JAPAN

Secretary

Kazuhiro Nakadai

Honda Research Institute Japan
8-1 Honcho, Wako,
Saitama, 351-0114, JAPAN /
Dept. of Math. and Comp. Sci.,
Graduate School of Engineering,
Tokyo Institute of Technology

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