実時間性と観測コストを考慮した 4脚ロボットのサッカー行動設計

小林 祐一^{*1} 深瀬 武^{*2} 上田 隆一^{*3}
新井 民夫^{*3} 湯浅 秀男^{*1*3}

Design of Quadruped Robot Soccer Behavior Considering Observational Cost

Yuichi Kobayashi^{*1}, Takeshi Fukase^{*2}, Ryuichi Ueda^{*3},
Tamio Arai^{*3} and Hideo Yuasa^{*1*3}

In this paper, we present a real-time decision making method for a quadruped robot whose sensor and locomotion have large errors, considering the observational cost and the optimality. We make a State-Action Map by off-line planning considering the uncertainty of the robot's location with Dynamic Programming. Using this map, the robot can immediately decide optimal action which minimizes the time to reach a target state at any states. The number of observation, swinging its head, is also minimized by taking the time cost of observation into account. We compress this map for implementation with Vector Quantization. The total loss of optimality through compression is minimized by using the differences of the values between the optimal action and the others. In the simulation, the performance of some soccer behaviors were improved in comparison with current methods. The low computation under the restriction of the memory was verified in the experiment.

Key Words: RoboCup, Quadruped Robot, Observational Cost, Dynamic Programming, Vector Quantization

1. はじめに

ロボカップは1997年の開始以来、実時間性や情報の不完全性・不確実性などのこれまでの人工知能研究例題と異なる特色を持った標準問題を提供している[1]。同大会の脚型ロボットリーグ(Legged Robot League)は、SONYの提供する脚型ロボットERS-2100(Fig. 1)を共通のハードウェアとして用い[†]、プログラムを競うリーグである[2]。脚型ロボットリーグは、ロボカップに共通の上記の問題に加え、

- 計算能力の制約(CPU200 [MHz], メモリ32 [MByte])がある
- 移動の誤差が大きい
- 観測の誤差が大きい
- カメラ視野が狭いため首振りというコストのかかる観測行動が必要である

などの性質を持つ。このような条件下でのサッカー行動を実現

するには、実時間性の要請下でタスクや状況に応じた自己位置精度と観測行動のコストのバランスをとれる行動計画手法が求められる。

脚型ロボットリーグにおける不確実さの対処例として、Hugelらの反射行動ルール設計があげられる[3]。これは、ランドマーク観測と歩容の種類を結びつける単純なルールを効果的に組み合わせることでサッカー行動を実現するものであるが、すべての状態で解を保証するものではなく、また試行錯誤による設計は汎用性に乏しい。また、光永らは4脚ロボットのサッカー行動実現を例にとり、情報量規準に基づいた観測判断可能な行動決定木構造を提案している[4]。しかし、教示情報を元に構成する手法であるため行動設計問題にはかかわりが薄い。

その他にも、不確実さをモデル化した研究は近年多数行われている。Winnerらは、自己位置の不確実さを見積もり、あるしきい値より不確実さが大きく見積もられた場合に観測行動を行う方法を提案し、そのしきい値を実験により求めている[5]。また、車輪型ロボットのナビゲーションにおいて状態の不確実さを考慮した研究としては、抽象化した状態空間にベイズモデルを用いる計画法[6]、プランニングコストを考慮した動作計画法[7] 観測位置と移動のオンライン計画法[8] などがある。しかし、サッカー行動では、ナビゲーションと異なり、自己位置がどのような精度で分かっている必要があるかは状況によって異なり、幾何的・実験的な解析が困難である。

原稿受付 2002年9月13日

^{*1}理化学研究所バイオミメティックコントロール研究センター

^{*2}松下電器産業

^{*3}東京大学工学部

^{*1}Bio-mimetic control research center, RIKEN

^{*2}Matsushita Electric Industrial Co., Ltd.

^{*3}The University of Tokyo

[†]概観は市販のAIBOと同様だが、研究開発用の仕様である。

Fig. 1 ERS-2100

Fig. 2 State variables

本研究では、4脚ロボットによるサッカー行動という必要な自己位置精度が状況により異なるタスクにおいて、実時間性と観測コストを考慮した行動設計手法を提案する。

- 状態空間を自己位置の不確かさのパラメータを含めた空間に拡張して行動計画を行う
- 行動計画に動的計画法[14]を用いて最適性を保証する
- オフラインで計画された状態-行動マップの実装を可能にするためにベクトル量子化による圧縮を用いる
- 移動と観測の不確かさを表すのに一様確率分布を用いるというアプローチをとることにより、計算能力の制約のもとで実時間性と大きな不確かさの問題を解決する。文献[5][6][8]に共通する「不確かさを見積もり、観測行動をとるべきかの判断に利用する」という方法を比較対象として実装し、提案手法の有効性を検証する。

以降、2章では問題設定を述べ、3章で提案行動計画法の全容を示す。4章で4脚ロボットのサッカー行動という問題への適用方法を述べ、5、6章でシミュレーション・実験の結果をそれぞれ示す。

2. 問題設定～脚型ロボットERS2100とフィールド～

本研究で扱う問題は、ロボットがFig. 2に示すフィールド上で、静止するボールに対し適切な方向（敵ゴール方向）に操作（前、左右方向のヘディング）ができるような位置に到達することであり、離散行動による目標状態への最短時間制御問題と捉えられる。条件、仮定を以下のように設定する。

- 初期状態は、ロボットがフィールド上任意の位置、ボールがフィールド上ロボット視野（ロボット正面方向から±90[deg]、2[m]以内の範囲）内の任意の位置とする。
- ロボットの行動は、（前進、左右回転など）8種類の歩容と1種類の観測行動からなる。
- フィールド周囲には色により識別可能な円筒状のランドマークが6個設置されている。
- ロボットは停止して首関節を水平に180[deg]振ることによりランドマークを観測する。これを観測行動と呼ぶ。
- 歩行中のロボットはボールを注視するため、ランドマークを観測できない。
- ロボットによるボール計測は正確に行われる。歩容およびランドマーク観測には大きな誤差が含まれる。

3. 実時間行動決定手法

実時間行動決定手法には、以下の要求が存在する。

- (1) 経験的な設計ではなく、最適性を用いたシステマティック

な挙動設計ができること

- (2) オンラインでの計算コスト（計算量、メモリ）が小さいこと
 - (3) 観測行動のコスト、自己位置の不確かさを表現できること
- (1) に対して、最適制御の一解法である動的計画法（Dynamic Programming: DP）を用いる。(2) に対し、オフラインでDPにより離散化された全状態に対する行動マップ（以後これを状態-行動マップと呼ぶ）を生成し、オンラインではこのマップを参照する。マップ参照によりオンラインでの計算量を低減できる。さらに、メモリの使用量の制約を満たすために、ベクトル量子化（Vector Quantization: VQ）[19]を用いて状態-行動マップの圧縮を行う。また、本研究の中心課題である(3)に対し、不確かさのパラメータを含めた状態空間の設計を行う。

3.1 DPを用いた行動計画

DPは、最適制御問題の解を求める計算法として知られている。 $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$ を状態変数ベクトル、 $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^m$ を制御入力ベクトルとする。離散時間系での決定論的な系の挙動を

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \quad k = 1, \dots, T-1 \quad (1)$$

と表す。ただし、 T はタスク終了までの時間ステップ数である。制御方策を $\mathbf{u}_k = \pi(\mathbf{x}_k)$ とする。最適制御の目的は、式(1)の制約のもと評価関数

$$L = \sum_{k=1}^T R(\mathbf{x}_k, \mathbf{u}_k) \quad (2)$$

を最大にする最適制御方策 $\pi^*(\mathbf{x})$ を求めることである。ここで、 $R(\mathbf{x}_k, \mathbf{u}_k)$ は各状態・制御入力の組に評価を与える関数である。

状態ベクトル $\mathbf{x}$ と制御入力 $\mathbf{u}$ はそれぞれ s と a に離散化する。ここで、 $\mathcal{S}$ と $\mathcal{A}$ を離散化した状態と行動の集合とする。離散時間・離散空間での（確率的）Bellman方程式は、

$$V^*(s) = \max_a \sum_{s'} \mathcal{P}_{ss'}^a (\mathcal{R}_{ss'}^a + V^*(s')) \quad (3)$$

$$Q^*(s, a) = \max_{a'} \sum_{s'} \mathcal{P}_{ss'}^a (\mathcal{R}_{ss'}^a + \max_{a'} Q^*(s', a')) \quad (4)$$

と表される。ここで、 $\mathcal{P}_{ss'}^a$ は s から s' へ行動 a をとることによって遷移する確率であり、 $\mathcal{R}_{ss'}^a$ は s から s' へ行動 a をとることによって遷移したときに得られる即時評価の期待値である。（最適）状態価値関数 $V^*(s)$ は状態 s の後に最適方策 π^* に基づいて行動決定したときに得られる即時評価の積算期待値である。（最適）行動価値関数 $Q^*(s, a)$ は、同様に状態 s で行動 a をとったのちに得られる報酬の積算期待値を表す。この最適価値関数を逐次計算により推定するのがDPのアルゴリズムである。具体的なアルゴリズムを付録Aに示す。

3.2 不確かさを考慮した最適な行動設計

状態の不確かさを考慮する方法としては、状態を世界座標系での確率分布で表す方法[7][9][13]や、観測と行動の履歴を木構造にして表す方法[4][23]などを挙げることができる[22]。本研究では、(1) 低計算コストで大きな不確かさを扱える(2) 状態空間の連続性を考慮しやすいという理由から、状態の確率分布を一様分布で表現する。Fig. 3に、二次元平面上での自己位置の不確かさの見積もりが歩行によって増大する例を示す。図中

Fig. 3 A state's transition on robot's locomotion

Fig. 4 A state's transition on robot's observation

Fig. 5 State transition in the expanded state space

灰色の部分はロボットが存在する範囲であり、矢印で示した移動により存在するであろう範囲が広がっている。Fig. 4は、観測行動によって不確かさが減少する例である。図中左下のランドマークを観測しランドマークからの距離を得ることで、その距離よりも遠い位置にいる可能性はないという考え方で存在範囲を狭くすることができる[†]。

このように不確かさを考慮に入れて行動計画問題を定式化すると以下のように表すことができる。

$$\begin{bmatrix} \mathbf{x}_{k+1} \\ \boldsymbol{\psi}_{k+1} \end{bmatrix} = \tilde{\mathbf{f}} \left(\begin{bmatrix} \mathbf{x}_k \\ \boldsymbol{\psi}_k \end{bmatrix}, \begin{bmatrix} \mathbf{u}_k \\ \boldsymbol{\omega}_k \end{bmatrix} \right) \quad (5)$$

ここで、 $\boldsymbol{\psi}$ は状態の不確かさを表すベクトル、 $\boldsymbol{\omega}$ は観測行動ベクトルである。この拡張された状態空間 $\tilde{\mathcal{X}} = \{\mathbf{x}, \boldsymbol{\psi} | \mathbf{x} \in \mathcal{X}, \boldsymbol{\psi} \in \Psi\} \subset \mathbb{R}^{\tilde{n}}$ で最適制御問題を解くことで、観測行動が陽に表現される。Fig. 5に状態空間の拡張の概念を示す。図左側では $\mathbf{x}$ の空間内の領域として不確かさを含めた状態表現がなされるのに対し、図右側では拡張された状態空間 $(\mathbf{x}, \boldsymbol{\psi})$ における点により表現される。歩行や観測による不確かさの増大や減少は、 $\boldsymbol{\psi}$ 軸方向への遷移によって表される。

3.3 ベクトル量子化によるマップの圧縮

DPの結果得られる $\pi^*(s)$ は、全状態に対して行動を与える状態-行動マップとみなすことができる。本研究では、ベクトル量子化 (Vector Quantization: VQ) [19] を用いてこれを圧縮する。圧縮によりマップの情報は一部失われるが、この際に失

[†]観測による不確かさ減少のモデルについては、4.2.2項でより詳しく述べる。

^{††}この格子は等間隔でなくてもよい。

Fig. 6 State space and index space

われる情報の損失をDPで求めた最適行動価値関数を尺度として評価する。これにより、最終的な行動計画の最適性を考慮に入れた圧縮が可能になる。

情報圧縮手法にベクトル量子化を用いる利点としては、3.3.2項で述べる歪み尺度を用いた圧縮が可能なこと、オンライン計算量が小さいことなどが挙げられる。ベクトル量子化は、(1)ベクトルの定義、(2)歪み尺度の定義、(3)符号帳の決定・改善、という過程で行う。DP、VQの適用のためには状態空間を離散化する必要がある。以後、拡張された状態空間 $\tilde{\mathcal{X}}$ を各次元について格子状に分割し^{††}、各状態を、 $s(i_1, \dots, i_{\tilde{n}})$ のように各次元の分割番号（以後これをインデックスと呼ぶ）を指定して表す (Fig. 6左側参照)。

3.3.1 ベクトルの定義

Fig. 6右側に例を示すインデックス空間を各次元 $\alpha = 1, \dots, \tilde{n}$ について分割間隔 ω_α で格子状に分割した領域を Ω とし、状態を k 番目の領域 Ω_k 内の ℓ 番目の要素として、 $s_\ell^k := s(i_1, \dots, i_{\tilde{n}})$ のように表現し直す。各次元のインデックス $i_1, \dots, i_{\tilde{n}}$ と k, ℓ の関係は次に示す j_α を用いて以下のように表せる。

$$j_\alpha \in \mathbb{Z} \quad \text{such that} \quad j_\alpha \leq \frac{i_\alpha}{\omega_\alpha} < j_\alpha + 1 \quad (6)$$

$$\ell = i_1 + \sum_{\alpha=1}^{\tilde{n}} (i_\alpha - \omega_\alpha j_\alpha) \prod_{\beta=1}^{\alpha-1} \omega_\beta \quad (7)$$

$$k = j_1 + \sum_{\alpha=1}^{\tilde{n}} j_\alpha \prod_{\beta=1}^{\alpha-1} \frac{I_\alpha}{\omega_\alpha} \quad (8)$$

ここで、 I_α はインデックス i_α の最大値であり、 k, ℓ の最大値はそれぞれ $N_\Omega = \prod_{\alpha=1}^{\tilde{n}} \frac{I_\alpha}{\omega_\alpha}$ 、 $N_V = \prod_{\alpha=1}^{\tilde{n}} \omega_\alpha$ となる。この領域 Ω 内の状態に対する最適行動を並べたものを一つのベクトルとする。すなわち、VQに適用するベクトルは $\mathbf{v}_k = [\pi^*(s_\ell^k)]_{\ell=1}^{N_V}$ と定義される。

厳密な最適性の議論のためには領域の分割幅 ω_α はVQアルゴリズム全体の結果が最適になるように定める必要があるが、ここでは計算量を削減するために、ベクトルの次元 N_V を定数に固定した上で次式で定めるエントロピーを最小にするように各 ω_α を定める。

$$H = -\frac{1}{N_\Omega} \sum_{k=1}^{N_\Omega} \log \frac{\mathcal{N}_s(\mathbf{v}_k)}{N_\Omega} \quad (9)$$

ここで、 $\mathcal{N}_s(\mathbf{v}_k)$ は状態空間全域において $\mathbf{v}_k$ と一致するベクトルの数を表す。これは、状態空間により多く同じベクトルが

Fig. 7 Loss of optimality by distortion

存在するようにベクトルを定める方がVQの圧縮効率がよくなるという考えに基づく。

3.3.2 歪み尺度の定義

ベクトル量子化の圧縮・復号に伴ってもとの情報が失われることを歪みと呼ぶ。一般的なデータ圧縮ではデータ再生率を最大にすることが求められるのに対し、本研究で圧縮する状態-行動マップにおいては、歪みを伴ったマップを用いて行動決定する際の最適性の損失を最小にすることが求められる。圧縮に伴う歪みによる最適性の損失をFig. 7で説明する。ある状態 s' における最適行動 $a^* = \pi^*(s')$ が他の行動 a_i に置き換わった場合、その損失は最適価値関数 Q^* により $Q^*(s', a^*) - Q^*(s', a_i)$ と表すことができる。この値を用いれば、図のように置き換わる行動を a_i と a_j とで比較した場合、 a_i に置き換わる方が損失が小さくなることが分かる。行動価値関数 Q^* を用いた歪み尺度を、 $\forall v, \forall w \in \mathcal{V}$ に対して

$$D(v, w) = \sum_{\ell=1}^{N_V} D(v_\ell, w_\ell) \quad (10)$$

のように定義する。ここで、行動間の歪み量 $D(a_m, a_n)$ は、

$$D(a_m, a_n) = \frac{\sum_{k=1}^{N_\Omega} \sum_{\ell=1}^{N_V} \delta_{\pi^*(s_\ell^k), a_m} (Q^*(s_\ell^k, a_m) - Q^*(s_\ell^k, a_n))}{\sum_{k=1}^{N_\Omega} \sum_{\ell=1}^{N_V} \delta_{\pi^*(s_\ell^k), a_m}} \quad (11)$$

で与える。 $\delta_{\alpha, \beta}$ は Kronecker のデルタである。この歪み尺度の導入により、行動計画における最適性を考慮した上でのコンパクトな状態-行動マップ生成が可能になる。

3.3.3 VQのアルゴリズム

VQでは、複数のベクトルを一つの代表ベクトルで代替することにより情報を圧縮する。このとき、一つの代表ベクトルで代替されるベクトルの集合をクラスタと呼び、代表ベクトルの集合を符号帳と呼ぶ。符号帳とクラスタの決定および改善には、一対法と一般化Lloydアルゴリズム (GLA) を用いる [21]。その具体的なアルゴリズムは、付録 B に示す。

4. 4脚ロボットのサッカー行動への適用

本章では、上述の提案行動計画手法を4脚ロボットのサッカー行動に適用する方法を示す。具体的には、状態 S および行動 A の定義、状態遷移確率 $P_{ss'}$ および報酬 $R_{ss'}$ の定義を行い、VQの適用方法を示す。

[†]一つの直方体は (x, y, θ) 空間を離散化した一単位に相当する。

^{††}拡張の単位は上記の直方体の大きさと等しい。

Fig. 8 Discrete state space

Table 1 State discretization

	Division number	Unit size
x [mm]	28	100[mm]
y [mm]	18	100[mm]
θ [deg]	24	15[deg]
r [mm]	10	100 - 600, ∞ [mm]
ϕ [deg]	12	15[deg]

4.1 状態の表現

状態空間は、フィールド座標系でのロボットの位置・姿勢 (x, y, θ) 、ロボットから見たボールの相対位置 (r, ϕ) およびロボットの位置・姿勢の不確かさを表す一次元パラメータ ψ の六次元からなる。各状態変数の離散化方法をTable 1とFig. 8に示す。

不確かさの表現には、一様確率分布を用いる。すなわち、自己位置の確率分布は確率零の領域と非零の領域で表され、非零の領域では等確率であるものとする。不確かさを表すパラメータ ψ は、 (x, y, θ) 空間の直方体[†]の集まりとして表現する。直方体集合は一様の非零の存在確率の領域を表す。この直方体集合は三次元楕円体を $(100$ [mm], 100 [mm], 90 [deg]) の範囲で拡張^{††}、roll, pitch, yaw角について 0.5 [rad] の単位で回転させ、重複を除いた結果811種類得られる。これが ψ の離散化に相当し、全状態数は $N_S = 28 \times 18 \times 24 \times 10 \times 12 \times 811 = 1,177,182,720$ となる。

4.2 状態遷移確率の定義

4.2.1 歩行による状態遷移

フィールド座標原点において行動 a_i をとったときのロボットの位置・姿勢の存在範囲 $M^{a_i}(\bar{x} \pm \delta_x, \bar{y} \pm \delta_y, \bar{\theta} \pm \delta_\theta)$ を、Table 2に示す。ここで、 $(\bar{x}, \bar{y}, \bar{\theta})$ は平均値を表し、 $(\delta_x, \delta_y, \delta_\theta)$ は最大誤差を表す。すなわち、 x 軸正方向を向いて原点に位置するロボットが“forward”つまり前進を実行した場合、ロボットは $(70$ [mm], 0 [mm], 0 [deg]) に移動しその際の誤差の範囲はそれぞれ $(\pm 30$ [mm], ± 15 [mm], ± 5 [deg]) である。また、歩行による状態遷移確率は状態に依存せず行動のみで決まるとする。歩行による状態遷移確率の計算法は付録 C に示す。

4.2.2 観測行動による状態遷移 (観測モデル)

ロボットは、色認識によって抽出したランドマークの大きさと方向の情報から存在範囲を特定し、状態遷移する。大きな誤差に対処するために、ランドマーク観測に関しては、「画像中で対象物の一部が欠けるなどして実際より小さく見えることは

Table 2 Actions and their parameters

i	Action	M^{a_i} [mm],[deg]			R^{a_i} [msec]
1	forward	(70 ± 30)	0 ± 15	0 ± 6	-768
2	backward	(-40 ± 40)	0 ± 15	0 ± 6	-768
3	rightside	(0 ± 20)	-60 ± 30	4 ± 4	-896
4	leftside	(0 ± 20)	60 ± 30	-4 ± 4	-896
5	rightforward	(10 ± 10)	37.5 ± 17.5	-14.5 ± 6.5	-832
6	leftforward	(10 ± 10)	-37.5 ± 17.5	14.5 ± 6.5	-832
7	roll_right	(35 ± 15)	-35 ± 15	11.5 ± 6.5	-832
8	roll_left	(35 ± 15)	-35 ± 15	-11.5 ± 6.5	-832
9	observation	(0 ± 0)	0 ± 0	0 ± 0	-2800

Fig. 9 Uniform distribution

あるが、実際より大きく見えることはない」という仮定をおき、この情報のみを用いる。

観測したランドマークの大きさから推定されるランドマークの距離を d 、実際のランドマークとの距離を λ とする。観測の確率分布には一様分布を用いる。つまり、ある距離 λ で観測 d を得る確率を $P(d|\lambda)$ とすると、Fig. 9 に示すようにある観測 d_k を得る確率 $P(d_k|\lambda)$ は、一定距離以内では λ によらず一定であり、それ以上では零になる。これが上記の仮定に相当する。

この表現方法は、Gauss 分布などの確率分布のパラメータを実験的に求める方法と比較して、単純化のため一度の観測によって得られる情報が少なくなるが、

- 確率分布を求めるための実験負荷、必要計算量が小さい
- 照明条件の変化やオクルージョンによりランドマークが小さく観測されるなど想定しにくい誤差の影響が小さい

という利点より、多数回・多種類の観測が行える場合に有効である [13]。また、ランドマーク観測方向の誤差分布は、同様に ± 15 [deg] の範囲の一様分布であるとする。この誤差モデルは各ランドマークについて独立であるとする。

4.3 報酬 $R_{ss'}^{a_i}$ の定義

行動に対する報酬には各行動 a_i の所要時間 $R^{a_i} (< 0)$ を用いる (Table 2 参照)。これにより、重みなしの報酬最大化問題は最短時間問題と等価になる。また、所要時間は行動のみに依存するとする。

4.4 VQ による状態-行動マップの圧縮

作成された状態-行動マップの容量は約 588 [MByte] である。これをロボットに実装するために、12 [MByte] まで圧縮する。本問題設定では、圧縮による歪みは到達時間の期待値の増加を意味する。VQ の計算により、これが最小化される。状態空間 S を、不確かさのパラメータ ψ によって二つずつ分割し、分割された各領域について独立に前述の VQ アルゴリズムを適用する。

Fig. 10 State-action map

Table 3 Initial positions of the robot and the ball

	r [mm]	ϕ [deg]	x [mm]	y [mm]	θ [deg]
Cond.1	2100	30	-1000	-600	0
Cond.2	1800	0	1000	0	180
Cond.3	2100	60	1000	-600	90

Fig. 11 Initial conditions

4.5 オフライン計算結果

オフライン計算でできたマップの一部を Fig. 10 に示す。図は、ロボット位置・方向をフィールド中央に固定し、ボールの位置を変化させたときのマップである。計算時間は、Pentium III 866 [MHz] PC による計算で、4.2 節で述べた状態遷移確率計算には 30 分程度、3.3, 4.4 節で述べた VQ による圧縮には約 3 日を要した。

5. シミュレーション

5.1 目的・条件

シミュレーションの目的は、以下の 2 点である。

- (1) VQ による状態-行動マップの圧縮の影響の評価
- (2) 提案手法の従来法との比較

すべての状態について、DP により最適性の実現されているかを評価することは困難であるため、Table 3 および Fig. 11 に示す三つの初期条件で各 10 回試行した結果を評価に用いる。また、ロボットは初期位置・姿勢を正確に知っているものとし、移動、観測に関するモデルには、4 章で用いた条件 (Table 2) と同じものを用いる。(1) の目的のため、圧縮前のマップと圧縮後のマップの比較を行う。

(2) の比較対象として、不確かさの見積もりを用いて観測を判断する二つの方法を用いる。これは [5] などで行われている考え方と同等であり、ロボットは自己位置とその不確かさの見積もりの情報を持っている。

- (a) 不確かさしきい値法：自己位置の不確かさの見積もりがあるしきい値を超えた場合に観測を行う。
- (b) 歩数しきい値法：一定歩数 (8 歩) 歩いたら観測を行う。

(a) での観測行動判断のしきい値は $(x_{th}, y_{th}, \theta_{th}) = (600$

Fig. 12 Trajectory

Table 4 Influence of compression

	Obs. #	Walk #	Time[sec]	Success rate
Before compression				
Cond.1	2.4	27.3	30.8	10/10
Cond.2	3.5	31.4	36.1	10/10
Cond.3	2.6	30.8	33.7	10/10
After compression				
Cond.1	2.6	27.5	31.4	10/10
Cond.2	3.6	32.5	37.6	10/10
Cond.3	2.8	32.2	35.9	10/10

[mm], 500 [mm], 60 [deg]) とし、どれか一つでも超えた場合に観測行動をとる。(b)の手法は観測の判断をより単純化したものである。両者とも行動決定には、自己位置の不確かさを考慮しない五次元の状態空間でDPによる計画を行った状態-行動マップを用いる。その他の条件は以下のように定める。

- 終了条件：ロボットの見積もる状態が目標状態に含まれるとき。
- タスク成功の判定：ロボットの実際の目標状態への到達。ロボットの見積もりで目標状態に到達したと判断しても、実際のロボット位置が目標状態に達していない場合に、失敗と判定される。

5.2 結果・考察

圧縮前のマップと圧縮後のマップをそれぞれ利用した行動の結果を Table 4 に示す。所要時間は、実行された行動ステップを時間に換算したものである。圧縮による再生率は 75% であった。すなわち、全状態のうち 25% の状態では行動情報が圧縮前のマップのものとは異なる。それに対して、平均到達時間の増加率は 4.3% であった。圧縮に伴って多くの行動置き換えが起こったにもかかわらず、到達時間への影響は小さいということが分かる。

次に、従来法との比較を Table 5 に示す。提案手法の場合はすべての試行において成功したのに対し、両しきい値法による場合には数度の失敗があった。本研究のタスクでは、目標状態到達の前に観測を行って確実に目標に到達できるようにする必要がある、自己位置によって観測できるランドマーク数が異なり、より不確かさが減少する位置で観測を行う必要がある、という理由から、観測行動を行うタイミングが重要な意味をもっている。両しきい値法に対する優位性から、自己位置の不確かさを考慮に入れた最適制御を用いる本手法が有効であることが分かる。

[†]状態遷移確率モデルに相当し、ロボットの存在範囲を表現できる手法であれば何を用いてもよい。本実験では、Uniform-MCL [13] を用いる。

Table 5 Comparison with alternatives

	Obs. #	Walk #	Time[sec]	Success rate
Variance threshold method				
Cond.1	4.0	28.1	34.4	10/10
Cond.2	4.6	33.6	41.5	8/10
Cond.3	3.6	32.4	38.6	9/10
Walkstep threshold method				
Cond.1	3.1	25.7	32.4	8/10
Cond.2	3.6	30.2	35.4	6/10
Cond.3	3.3	29.1	35.2	7/10

Table 6 Experimental results

	Obs. #	Walk #	Time[sec]	Success rate
Remote brain				
Cond.1	2.3	25.5	30.1	10/10
Cond.2	3.1	29.2	35.8	9/10
Cond.3	2.7	28.9	34.5	9/10
Completely autonomous				
Cond.1	1.4	23.3	27.5	9/10
Cond.2	3.0	29.8	42.0	6/10
Cond.3	2.3	32.2	41.1	8/10

Fig. 12 に、シミュレーションで得られたロボットの軌跡の例を示す。図中の矢印はロボットの位置および方向を表している。軌跡から、移動速度の遅い横方向への移動を多用するなど、直観的な解とはやや異なる結果が得られていることが分かる。このような結果が自動的に求まることも本手法の利点の一つである。

6. 実 験

本章では提案手法を実機環境に実装した結果を示す。実験はリモートブレイン方式と完全自律方式の2通り行う。それぞれの目的は以下の通りである。

- (1) リモートブレイン方式：シミュレーションとの比較により、オフライン計画の移動・観測のモデル化の妥当性を検証する。
- (2) 完全自律方式：自己位置同定法[†]との組み合わせにより完全自律で動作させる。オフライン計画結果の実装可能性と実時間動作性を検証する。

初期条件および終了条件はシミュレーションの場合と同様であるが、成功判定では、目標状態への到達だけでなく、ボールに意図せず接触した場合を失敗として扱う。

実験結果を Table 6 に示す。表におけるリモートブレインの時間は、状態遷移および行動決定を行う外部PCとの通信時間の問題から、行動のデータより換算したものを表示している。完全自律型の時間は実測値である。

リモートブレイン方式

観測回数・歩行回数ともにシミュレーションより効率がよい。これは、実験環境の移動・観測の誤差がオフライン計画に用いたモデルよりも小さかったためと考えられる。オフライン計算で不確かさを大きく見積もることの悪影響は小さく、本適用で

Fig. 13 Example of motion sequence

のモデル化が妥当であるということがいえる。

完全自律方式

リモートブレインの場合と比較しても、さらに観測回数・歩行回数が小さくなっている。これは、ボールを注視している際にもランドマークが視界に入ることがある、ゴール情報を利用する、不確かさの表現力が高いなどの Uniform-MCL を用いることの利点のためであると考えられる。

また、到達時間の増大の原因は行動決定および歩行制御の処理時間であり、そのうち歩行制御は実機では不可避の要因である。歩行制御に要する時間が約 100 [msec] であるのに対して行動決定の処理時間はその 1/10 以下であった。マップ参照という行動決定法がオンライン計算量を削減するという要求に当たっていることが分かる。

タスク失敗の原因

タスクの失敗は、目標状態付近でロボットがボールに意図せずに接触してしまう場合がすべてであった。シミュレーションの従来法における失敗は目標状態に到達できない場合であり、実験の場合とは異なる。これは、足元付近でのロボットのボール位置測定誤差によるものと考えられる（足元付近にあるボールはカメラ視野のほとんどの領域を占めてしまうため、大きな誤差を生むことがある）。この失敗は、連続して歩行する完全自律方式では振動が大きいことより顕著であった。

以上より、ボール近傍での状態や観測のモデル化を除いて、本提案手法の適用方法は妥当であったということがいえる。また、Fig. 13 に実際のロボットの移動を示す。図中①が初期位置で、左側が目標ゴールである。③、④の位置で観測行動をとり、⑤、⑥のオペレーションによりボールをゴール方向に転がすことができていくことが分かる。

7. 考 察

本研究での DP では、圧縮の際の歪み尺度定義に行動価値関数を用いることから、収束が保証されていることが知られている。価値反復法と状態を完全に離散化する方法を用いた [16]、自律的な状態空間の離散化方法 [11] [12] や連続空間を扱う方法 [17] も近年盛んに研究されており、それらとの組み合わせにより状態表現の効率化を図ることが期待できる。

また、本提案手法を他のロボットの存在する環境に直接適用することは困難であるが、1 台のロボットによるサッカー行動の状態-行動マップを協調行動の設計に利用することは可能であり、より高次元の問題に対して利用可能な部分問題の解となることが期待される。

実機におけるタスク失敗は、2 章での仮定が実機に適合していなかったためであり、対策としては、以下の二つが考えられる。

- 実験により足元付近のボールとの接触領域を測定し、より精密な状態遷移確率を用いて DP 計算を行う
- 足元付近のボールの相対位置を、自己位置と同様に不確かさを含めて状態表現を行う

8. お わ り に

本研究では、ロボット位置の不確かさを含めた状態-行動マップを DP によるオフライン計算により構成し、VQ により圧縮する方法を提案した。圧縮の歪み量の定義に DP により得られる行動価値関数を用いることで、メモリに制約のあるロボットに実時間性を損なわずに実装した。シミュレーションにおいて、本提案手法の従来型のしきい値法に対する優位性を示し、実験において、実機に実装可能であり有効であることを示した。観測行動のコストや情報の不完全性というロボカップ 4 脚ロボットの問題設定の持つ性質は、今後も人工知能の重要な課題の一つであると考えられる。

参 考 文 献

- [1] 浅田稔, 北野宏明: “ロボカップ戦略: 研究プロジェクトとしての意義と価値”, 日本ロボット学会誌, vol.18, no.8, pp.1081-1084, 2000.
- [2] 藤田雅博: “4 脚ロボットリーグの技術課題と展望”, 日本ロボット学会誌, vol.20, no.1, pp.15-19, 2002.
- [3] V. Hugel, P. Bonnin and P. Blazevic: “Reactive and Adaptive Control Architecture Designed for the Sony Legged Robots League in RoboCup 1999,” Proc. of IROS-2000, 2000.
- [4] 光永法明, 浅田稔: “移動体の意思決定のための情報量規準に基づく観測対象選択戦略”, 日本ロボット学会誌, vol.19, no.6, pp.793-800, 2001.
- [5] E. Winner and M. Veloso: “Multi-Fidelity Robotic Behaviors: Acting With Variable State Information,” Proc. of the Seventeenth National Conference on Artificial Intelligence, Austin, Aug. 2000.
- [6] A. R. Cassandra, L. P. Kaelbling and J. A. Kurien: “Acting under Uncertainty: Discrete Bayesian Models for Mobile-Robot Navigation,” Proc. of IROS-96, 1996.
- [7] 三浦純, 白井良明: “プランニングコストと視覚の不確かさを考慮した移動ロボットの視覚と移動のプランニング”, 人工知能学会誌, vol.13, no.4, pp.558-596, 1998.
- [8] 文仁赫, 三浦純, 白井良明: “不確かさを考慮した観測位置と移動のオンライン計画手法”, 日本ロボット学会誌, vol.17, no.8, pp.1107-1113, 1999.

- [9] D. Fox, W. Burgard, F. Dellaert and S. Thrun: "Monte Carlo Localization: Efficient Position Estimation for Mobile Robots," Proceedings of Sixteenth National Conference on Artificial Intelligence (AAAI-99), Orlando, Florida, 1999.
- [10] T. Fukase, M. Yokoi, Y. Kobayashi, R. Ueda, H. Yuasa and T. Arai: "Quadruped Robot Navigation Considering the Observational Cost," Proceedings of RoboCup symposium, 2001.
- [11] 高橋泰岳, 浅田稔: "実ロボットによる行動学習のための状態空間の漸次的構成", 日本ロボット学会誌, vol.17, no.1, pp.118-124, 1999.
- [12] 小林祐一, 湯浅秀男, 新井民夫: "自律分散型の適応アルゴリズムによる強化学習のための関数近似", 計測自動制御学会論文集, vol.38, no.2, pp.219-226, 2002.
- [13] R. Ueda, T. Fukase, Y. Kobayashi, T. Arai, H. Yuasa and J. Ota: "Uniform Monte Carlo Localization - Fast and Robust Self-localization Method for Mobile Robots," Proc. of ICRA-2002.
- [14] R. Bellman and S. Dreyfus: Applied Dynamic Programming. Princeton University Press, 1962.
- [15] R. S. Sutton and A. G. Barto: Reinforcement Learning: An Introduction. The MIT Press, 1998.
- [16] D. P. Bertsekas and J. N. Tsitsiklis: Neuro-Dynamic Programming. Athena Scientific, 1996.
- [17] D. Ormonet: "Kernel-based reinforcement learning," Machine Learning, vol.49, pp.161-178, 2002.
- [18] G. J. Gordon: "Stable Function Approximation in Dynamic Programming," Proc. of the International Conference on Machine Learning, pp.261-268, 1995.
- [19] A. Gersho and R. M. Gray: Vector Quantization and Signal Compression. Kluwer Academic Publishers, 1992.
- [20] 田崎三郎他: 情報源符号化—歪みのあるデータ圧縮. 培風館, 2000.
- [21] W. H. Equitz: "A new vector quantization picture coding," IEEE Trans. Acoust. Speech Signal Process., pp.1568-1575, Oct. 1989.
- [22] S. M. LaValle: "Robot Motion Planning: A Game-Theoretic Foundation," Algorithmica, vol.26, pp.430-465, 2000.
- [23] L. P. Kaelbling, M. L. Littman and A. R. Cassandra: "Planning and Acting in Partially Observable Stochastic Domains," Artificial Intelligence, vol.101, pp.99-134, 1995.

付録 A. 動的計画法における価値反復法

価値反復法 (value iteration) による π^* の導出は以下のアルゴリズムによる. ε_d は正の定数, N_S は状態数である.

```

 $V(s_i) \leftarrow 0 \ (i = 1, \dots, N_S)$ 
do
   $\Delta \leftarrow 0$ 
  for  $i = 1$  to  $N_S$ 
     $v \leftarrow V(s_i)$ 
     $V(s_i) \leftarrow \max_{a_k} \sum_{j=1}^{N_S} \mathcal{P}_{s_i s_j}^{a_k} [\mathcal{R}^{a_k} + V(s_j)]$ 
     $\Delta \leftarrow \max\{\Delta, |v - V(s_i)|\}$ 
  next
loop until  $\Delta < \varepsilon_d$ 
 $\pi^*(s_i) \leftarrow \arg \max_{a_k} \sum_{j=1}^{N_S} \mathcal{P}_{s_i s_j}^{a_k} [\mathcal{R}^{a_k} + V(s_j)] \ (i = 1, \dots, N_S)$ 

```

付録 B. 一対法と GLA

B.1 一対法

一対法 (Pairwise Nearest Neighbor algorithm: PNN) では, 3.3.1項で定義したベクトル $\mathbf{v}_k \in \mathcal{V} (k = 1, \dots, N_\Omega)$ を, 最終

的に N_P 個の部分集合 (クラスタ) $R_i \subset \mathcal{V}, i = 1, \dots, N_P$ にクラスタリングする. その際, 各部分集合 R_i に代表ベクトル $\mathbf{c}_i$ を以下の方法で割り当てる.

$$C_e(R_i) := \mathbf{c}_i \text{ such that } \sum_{\mathbf{v}_k \in R_i} \mathcal{D}(\mathbf{c}_i, \mathbf{v}_k) \leq \sum_{\mathbf{v}_k \in R_i} \mathcal{D}(\mathbf{w}, \mathbf{v}_k) \ \forall \mathbf{w} \in \mathcal{V}, \mathbf{w} \neq \mathbf{c}_i \quad (\text{B.1})$$

これは, 3.3.2項で定義した歪み尺度の部分集合にわたる総和を最小にするベクトルを与える関数である. さらに, クラスタ R_i, R_j に対して d_i, d_{ij} を

$$d_i = \sum_{\mathbf{v}_k \in R_i} \mathcal{D}(C_e(R_i), \mathbf{v}_k) \quad (\text{B.2})$$

$$d_{ij} = \sum_{\mathbf{v}_k \in R_i} \mathcal{D}(C_e(R_i \cup R_j), \mathbf{v}_k) + \sum_{\mathbf{v}_k \in R_j} \mathcal{D}(C_e(R_i \cup R_j), \mathbf{v}_k) \quad (\text{B.3})$$

のように定義する. PNN は, クラスタの併合を繰り返し, 部分集合の総数を N_Ω から N_P に以下の方法で減少させる.

$R_k = \{\mathbf{v}_k\} (k = 1, \dots, N_\Omega), n \leftarrow N_\Omega$

do

$$(i, j) \leftarrow \arg \min_{(\alpha, \beta)} (d_{\alpha\beta} - d_\alpha - d_\beta)$$

$$R_i \leftarrow R_i \cup R_j$$

$$R_j \leftarrow \emptyset$$

$$n \leftarrow n - 1$$

until $n = N_P$

各部分集合の代表ベクトルを集めたものを符号帳と呼び, ここで得られた N_P 個の代表ベクトルの集合を初期符号帳とする.

B.2 GLA

GLA は以下のアルゴリズムにより, 初期符号帳と初期クラスタから開始して符号帳とクラスタの改善を繰り返す. ここで, ε_g は正の定数, N_P は B.1 で生成したクラスタ数である.

$\mathbf{c}_i = C_e(R_i) (i = 1, \dots, N_P), \sigma' = 0$

do

$$R_i \leftarrow \{\mathbf{w} | \mathcal{D}(\mathbf{w}, \mathbf{c}_i) \leq \mathcal{D}(\mathbf{w}, \mathbf{c}_j) \ \mathbf{w} \in \mathcal{V}, \forall i \neq j\}$$

$$\mathbf{c}_i \leftarrow C_e(R_i)$$

$$\sigma \leftarrow \sigma'$$

$$\sigma' \leftarrow \sum_{i=1}^{N_P} d_i$$

$$\Delta \leftarrow (\sigma' - \sigma) / \sigma$$

loop until $\Delta < \varepsilon_g$

このアルゴリズムは, 初期符号帳, 初期分割から始め, 「符号帳を固定して, この符号帳に対する最適な分割を求める」「分割を固定して, その分割に対する最適な符号帳を求める」という繰り返しからなる.

付録 C. 状態遷移確率の計算法

確率的に表現されるロボット位置と決定論的に表すボール位置とを分けて考えるために, 拡張された状態空間 $\tilde{\mathcal{X}}$ を,

$\tilde{\mathcal{X}} = \tilde{\mathcal{X}}^p \otimes \mathcal{X}^b$ のようにロボット位置の空間 $\tilde{\mathcal{X}}^p$ とボール位置の空間 $\mathcal{X}^b$ に分解する. また, $\tilde{\mathcal{X}}^p = \mathcal{X}^p \otimes \Psi$ とし, 各状態空間 $\tilde{\mathcal{X}}^p, \mathcal{X}^p, \mathcal{X}^b$ をそれぞれ離散化した状態の集合を $\tilde{\mathcal{S}}^p, \mathcal{S}^p, \mathcal{S}^b$ と表す. 本適用では, $\tilde{\mathcal{S}}^p$ の要素は, $\mathcal{S}^p$ の要素の集合として表される. すなわち, $p \in \mathcal{S}^p, \tilde{p} \in \tilde{\mathcal{S}}^p, p \subset \tilde{p}$ の関係が成り立つ[†]. 状態 $s \in \mathcal{S}$ は, ロボット位置を表す状態 $\tilde{p} \in \tilde{\mathcal{S}}^p$ とボール状態を表す状態 $b \in \mathcal{S}^b$ の組と一対一に対応する. これを, $s \leftrightarrow (\tilde{p}, b)$ のように表す.

C.1 自己位置の遷移確率計算

状態 $\tilde{p} \in \tilde{\mathcal{S}}^p$ において行動 a_j をとった結果 (決定論的に) 遷移する状態 $k(\tilde{p}, a_j) \in \tilde{\mathcal{S}}^p$ を, 以下のように近似する. ここで, S は $p \in \mathcal{S}^p$ の集合, M^{a_j} は 2 で述べた行動 a_j による状態遷移を表す集合である.

init $S \leftarrow \emptyset$

do

choose $[x, y, \theta] \in \tilde{p}$ randomly

choose $[\tilde{x}, \tilde{y}, \tilde{\theta}] \in M^{a_j}$ randomly

find $p' \in \mathcal{S}^p$ such that $p' \ni [x + \tilde{x}, y + \tilde{y}, \theta + \tilde{\theta}]$

$S \leftarrow S \cup \{p'\}$

loop sufficiently

$k(\tilde{p}, a_j) = \arg \min_{\tilde{p}_i \in \tilde{\mathcal{S}}^p} |S \cup \tilde{p}_i - S \cap \tilde{p}_i|$

[†]状態 $p, \tilde{p}, b$ は, 離散化されたシンボルの集合 $\mathcal{S}^p, \tilde{\mathcal{S}}^p, \mathcal{S}^b$ の要素であるが, 同時に $\mathcal{X}^p, \tilde{\mathcal{X}}^p, \mathcal{X}^b$ の部分空間に対応する. 混乱しない限り, 同じ記号を用いて $p \subset \mathcal{X}^p, b \subset \mathcal{X}^b$ のように各状態を部分空間としても扱う.

これは, 多数の点の遷移情報から構成された集合 S に対し, 最も近い $\tilde{p}_i \in \tilde{\mathcal{S}}^p$ を採用することを意味している.

C.2 ボール位置の状態遷移確率計算

次に, 状態 $b \in \mathcal{S}^b$ において行動 a_j をとった結果状態 $b_l \in \mathcal{S}^b$ に遷移する確率 $p_{bb_l}^{a_j}$ の計算法を以下に示す.

$n_\ell \leftarrow 0$ ($\ell = 1, \dots, |\mathcal{S}^b|$)

do

choose $[r, \phi] \in b$ randomly

choose $[\tilde{x}, \tilde{y}, \tilde{\theta}] \in M^{a_j}$ randomly

calculate $[r', \phi']$ from $[r, \phi], [\tilde{x}, \tilde{y}, \tilde{\theta}]$

$n_\ell \leftarrow n_\ell + 1$ such that $b_\ell \ni [r', \phi']$

loop sufficiently

$p_{bb_\ell}^{a_j} \leftarrow n_\ell / \sum_\ell n_\ell$

アルゴリズム中 calculate の計算は, 二次元座標系 Σ_0 において $[r, \phi]$ で極座標表示された点を, Σ_0 を $[\tilde{x}, \tilde{y}]$ 平行移動し, $\tilde{\theta}$ 回転させた座標系 Σ_1 から見たときの極座標 $[r', \phi']$ を与える.

C.3 状態遷移確率の導出

以上で求めた $k(\tilde{p}, a_j), p_{bb_l}^{a_j}$ を用いて, 状態 s_i において行動 a_k をとった結果状態 s_j に遷移する確率 $P_{s_i s_j}^{a_k}$ を表す.

$$P_{s_i s_j}^{a_k} = \begin{cases} p_{bb'}^{a_k} & \text{for } q = k(\tilde{p}, a_k) \\ 0 & \text{for } q \neq k(\tilde{p}, a_k) \end{cases} \quad (\text{C.4})$$

$$(\text{C.5})$$

ここで, $s_i, s_j \in \mathcal{S}$ は, $s_i \leftrightarrow (\tilde{p}, b), s_j \leftrightarrow (\tilde{q}, b')$ をそれぞれ満たすものとする.

小林祐一 (Yuichi Kobayashi)

1975 年 1 月 8 日生. 1997 年東京大学工学部精密機械工学科卒業, 1999 年同大学大学院工学系研究科精密機械工学専攻修了. 2002 年東京大学大学院工学系研究科精密機械工学専攻博士課程修了. 同年, 理化学研究所バイオ・ミメティックコントロール研究センターフロンティア研究員, 現在に至る. 博士 (工学). 強化学習, 自律分散システムの研究に従事.

(日本ロボット学会正会員)

上田隆一 (Ryuichi Ueda)

1979 年 2 月 16 日生. 2001 年東京大学工学部精密機械工学科卒業, 2003 年東京大学大学院工学系研究科精密機械工学専攻修士課程修了. 同年, 同専攻博士課程進学, 現在に至る. ロボカップ4足ロボットリーグに関する研究に従事.

(日本ロボット学会学生会員)

湯浅秀男 (Hideo Yuasa)

1961 年 4 月 2 日生. 1986 年名古屋大学大学院工学研究科博士前期課程修了 (情報工学専攻). 同年名古屋大学工学部情報工学科助手. 同大学工学部電子機械工学科講師を経て, 1999 年東京大学大学院工学系研究科助教授. また, 1993 年より理化学研究所バイオ・ミメティックコントロール研究センターフロンティア研究員を併任 (非常勤). 2002 年 9 月 18 日逝去. 博士 (工学). システム理論, ロボット工学の研究に従事.

(日本ロボット学会正会員)

深瀬 武 (Takeshi Fukase)

1977 年 2 月 2 日生. 2000 年東京大学工学部精密機械工学科卒業, 2002 年東京大学大学院工学系研究科精密機械工学専攻修士課程修了. 現在, 松下電器産業に勤務. カーオーディオの開発に従事.

新井民夫 (Tamio Arai)

1947 年 8 月 4 日生. 1970 年東京大学工学部精密機械工学科卒業. 1972 年同大学大学院工学系精密機械工学課程修士課程修了. 1977 年同大学大学院工学系精密機械工学課程博士課程修了. 1977 年同大学工学部講師. 1979 年同助教授. 1987 年より東京大学工学系研究科精密機械工学専攻教授, 現在に至る. 博士 (工学). 自動組立, 複数移動ロボットなどの研究に従事.

(日本ロボット学会正会員)