状態推定の不確かさを考慮した実時間行動決定

○上田隆一　新井民夫　坂本浩平　実川達明　竹下和孝 (東京大学)
梅田和昇　大隅久　小村正樹 (中央大学)

Real-Time Decision Making under Uncertainty of State Estimation

*R. Ueda, T. Arai, K. Sakamoto, Y. Jitsukawa, and K. Takeshita (Univ. of Tokyo)
K. Umeda, H. Osumi, and M. Komura (Chuo Univ.)

Abstract— We propose the real-time Q_{MDP} method for decision making of a robot under uncertain state recognition. This method utilizes a particle filter and dynamic programming. This method is applied to total behavior of a goalkeeper for robot soccer. Simulations, experiments and actual games have suggested that the method can decide actions effectively according as uncertain result of state estimation.

Key Words: real-time Q_{MDP} value method, dynamic programming, particle filter, RoboCup

1. 序論

環境中における自身の位置姿勢や，その他行動決定に必要な情報を正確に同定することは，自律移動ロボットにとって非常に困難な課題である．そのため，ロボットが情報の不確かさの中で自身の取るべき行動を選択し，なんらかの作業を遂行するための行動決定問題が研究されている．

情報の不確実さとは，ここでは環境の地図は既知で，未知の可動/移動物体はないが，地図中での自身の位置や物体の姿勢・速度を正確に同定できない場合を指す．情報空間を用いる手法は，このような状況下での行動決定に利用されている手法の一つである [1]．情報空間とは，状態空間（環境やロボット自身の状態を表す変数で張られる空間）を，変数の曖昧さを表す変数で拡張したものである．例えば Roy らは，移動ロボットの位置・方向に，それらの曖昧さを表す変数一つを加えて 4 次元空間中でナビゲーションの問題を解いている [2]．情報空間を用いると，観測等による不確かさの増減が，行動による状態遷移と同様に空間中の点の移動として扱え，既存の行動決定手法を利用できる．一方，もとの物理的な空間よりも高い次元の空間を扱うため，問題を解くときの計算コストが大きくなる．

別の有力な方法としては，Littman らによる Q_{MDP} value method（本稿では Q-MDP 法と呼ぶ）が挙げられる [3]．この手法では，常に状態が既知であると仮定して事前に行動決定問題を解いておき，現在得られている（不確かな）情報と照らし合わせて妥当な行動が選択される．つまり,「今の状況が分かっていれば何をすれば良いのか分かるが，実際は状況が不確かである」という問題を扱う手法である．情報の不確かさは状態空間中の確率分布として表現される．この枠組みでは，事前に状態既知の問題が解けないと適用できないが，解ける場合，問題を解く際の計算コストは情報空間のコストよりも小さい．しかし，Q-MDP 法は，確率分布から行動決定する際の計算量が多く，実時間性の求められる状況で使用することが困難である．

そこで本稿では，パーティクルフィルタを用いて Q-MDP 法を高速に実行する手法を提案する．実時間性の求められるタスクの例として，ロボットサッカーのゴールキーパに提案手法を適用し，試合で使用できることを示す．また，評価のため，シミュレーションによって提案手法の性質を明らかにし，実機実験で不確かさを考慮しない場合との比較を行う．

2. 実時間 Q-MDP 法

2·1 問題設定

次のような部分観測マルコフ決定過程を考える．

1. ロボットやその周囲の状況が n 個の変数 $x_1, x_2, \ldots, x_n$ で張られる状態空間 $\mathcal{X}$ 中の点 $\boldsymbol{x}$ として表現できる．
2. あるタスクに対し，それが達成された状況を $\mathcal{X}$ の部分集合 $\mathcal{X}_{\mathrm{f}}$ で表す．$\mathcal{X}_{\mathrm{f}}$ 内の状態を終端状態と呼ぶ．
3. ロボットは m 種類の行動からなる集合 $\mathcal{A} = \{a_1, a_2, \ldots, a_m\}$ から単位時間あたり行動を一つ選択する．
4. 状態 $\boldsymbol{x} \in \mathcal{X}$ から，ある行動 $a \in \mathcal{A}$ を実行後，$\boldsymbol{x}' \in \mathcal{X}$ に状態が遷移することに対する確率密度 $p^a_{\boldsymbol{x}\boldsymbol{x}'}$ が任意の $\boldsymbol{x}, \boldsymbol{x}', a$ の組に対して既知であり，時不変である．
5. 上記 $\boldsymbol{x}, \boldsymbol{x}', a$ の組に対して報酬 $r^a_{\boldsymbol{x}\boldsymbol{x}'} \in \Re$ を与える．
6. 制御の目的は，ある状態 $\boldsymbol{x} \in \mathcal{X}$ から $\boldsymbol{x}_{\mathrm{f}} \in \mathcal{X}_{\mathrm{f}}$ に至るまでの報酬の和（$\boldsymbol{x}$ の価値と呼び $V(\boldsymbol{x})$ と表す）を最大とする行動を決定することである．場合によっては，$\mathcal{X}_{\mathrm{f}}$ 中の各終端状態に価値 $V(\boldsymbol{x}_{\mathrm{f}})$ を設定し，報酬の和と $V(\boldsymbol{x}_{\mathrm{f}})$ を足した値を価値 $V(\boldsymbol{x})$ とする．
7. ロボットは，状態を $\mathcal{X}$ 中の確率密度関数 $b : \mathcal{X} \to \Re$ として知覚する．状態が $Y \subset \mathcal{X}$ 中に存在する確率は，$B(Y) = \int_Y b(\boldsymbol{x}) d\boldsymbol{x}$ と定義される．

このとき，項目 7 を無視して状態が常に同定できると仮定すると，この問題設定はマルコフ決定過程（MDP）となり，動的計画法 [4] 等で解ける．多くの場合，$\mathcal{X}$ は有限個の離散状態の集合 $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ に分割され，各状態に対して最適な行動を与える写像（方策）$\pi^* : \mathcal{S} \to \mathcal{A}$ と，その方策に基づいた場合の価値を与える関数（最適状態価値関数）$V^* : \mathcal{S} \to \Re$ が得られる．

2·2 Q-MDP 法

項目 7 の存在下では，たとえ π^* が分かったとしてもロボットが最適な行動を選択できるわけではない．Q-

MDP 法 [3] は，センサなどによる観測から，状態 $\boldsymbol{x}$ が離散状態 s に属する確率 $B(s)$ から，次のような値

$$Q_{\mathrm{MDP}}(B,a)=\sum_{s}B(s)\sum_{s'}\mathcal{P}^{a}_{ss'}[\mathcal{R}^{a}_{ss'}+V^{*}(s')] \quad (1)$$

を求め，それを最大化する方策 Π:

$$\Pi(B)=\operatorname*{argmax}_{a}Q_{\mathrm{MDP}}(B,a) \quad (2)$$

で行動決定を行う手法である．ここで，$\mathcal{P}^{a}_{ss'},\mathcal{R}^{a}_{ss'}$ は，それぞれ $p^{a}_{\boldsymbol{x}\boldsymbol{x}'},r^{a}_{\boldsymbol{x}\boldsymbol{x}'}$ から，離散状態間の遷移確率と報酬の平均値を求めたものである．値 $Q_{\mathrm{MDP}}(B,a)$ は，確率分布 B（あるいは確率密度関数 b）に対する価値関数とみなすことができる．

式 (1) の計算量は，離散状態数 n と行動の種類 m に比例するため，一回の行動決定に制限時間を設定すると，それに対する nm の上限が必ず存在する．

2·3 実時間 Q-MDP 法

そこで，Q-MDP 法を n が大きい状態でも実時間実行できるように，パーティクルフィルタを用いて $Q_{\mathrm{MDP}}(B,a)$ 値を計算することを提案する．パーティクルフィルタを推定に適用することが大きな状態数に対して有効性であることは，自己位置推定の研究で示されてきた [5]．パーティクルフィルタは，パーティクルと呼ばれる重みつきの点 $\xi^{(i)}$ $(i=1,2,\ldots,N)$ を空間に散布し，その空間中の確率分布を近似する手法である．各パーティクル $\xi^{(i)}$ は，パラメータとして重み $w^{(i)}$ と空間中での位置 $\boldsymbol{x}^{(i)}=(x_1^{(i)},x_2^{(i)},\ldots,x_n^{(i)})$ を持つ．状態空間 $\mathcal{X}$ 中の状態 $\boldsymbol{x}$ を推定する問題を考えると，$\boldsymbol{x}$ がある領域 $s\subset\mathcal{X}$ に含まれる確率は，s 内のパーティクルの重みを合計した値

$$B(s)=\sum_{i=1}^{N}w^{(i)}\delta(\boldsymbol{x}^{(i)}\in s) \quad (3)$$

で近似される．ここで，$\delta(\cdot)$ は括弧内が真ならば 1，偽ならば 0 をとる関数とする．パーティクルの重みは状態に関する情報が入ったときにベイズの定理で更新され，状態が遷移したときには，遷移前のパーティクルの重みと位置に基づいて新たなパーティクルの分布が作成される．

実時間 Q-MDP 法では，パーティクルフィルタを用い，以下に説明する二つの過程で m と n の大きさの影響を受けずに，情報の不完全さを考慮して行動決定することを実現する．

2·3.1 投票

提案手法では，まず π^{*} を用いて，m 種類の行動から c 個の候補を選出する．各行動に対して次の値

$$g(a)=\sum_{i=1}^{N}w^{(i)}\delta(\pi^{*}(s^{(i)})=a) \quad (4)$$

を計算し，値の大きいものから c 個選択する．ここで，$s^{(i)}$ は $\boldsymbol{x}^{(i)}$ が属している離散状態である．この過程を実装したときの計算量はパーティクルの数 N に比例する．

2·3.2 Q_{MDP} の計算

次に，選択された c 個の行動に対して Q_{MDP} を計算する．パーティクルを利用すると Eq.(1) は以下の式

$$Q_{\mathrm{MDP}}(B,a)=\sum_{i=1}^{N}w^{(i)}[\mathcal{R}^{a}_{s^{(i)}s'}+V^{*}(s')] \quad (5)$$

で近似計算できる．ここで，遷移先の状態 $\boldsymbol{x}'\in s'$ は，各パーティクルの位置 $\boldsymbol{x}^{(i)}$ から，確率密度関数 $p^{a}_{\boldsymbol{x}\boldsymbol{x}'}$ に従って選択する．

c 個の行動に対して式 (5) を計算するとき，その計算量は c とパーティクルの数 N に比例する．離散状態の数 n と必要なパーティクルの数 N には相関があることには留意すべきであるが，計算量は直接的には行動数 m にも離散状態の数 n にも影響を受けない．

3. ゴールキーパータスクへの適用

評価のための例題として，RoboCup 4 足ロボットリーグ [6]（2003 年度の環境）のためのゴールキーパー行動を扱う．キーパーロボットが自己位置の不確かさに対して提案手法で柔軟に対応できることを示す．

3·1 ロボカップ 4 足ロボットリーグ

この競技ではサッカーロボットとして，Fig. 1(a) に見られる自律型 4 脚ロボット ERS-210 を用いている．このロボットは各脚と頭部に 3 自由度を有する多自由度ロボットで，計算資源として MIPS 192 MHz と 32 MB の DRAM が搭載されている．主要なセンサは頭部の CMOS カメラ（18 万画素）であり，このカメラから 40[ms] 毎に 176×144 画素，256 階調の YCrCb カラー画像が DRAM に転送される．4 足リーグの公式試合では，Fig. 1(c) に見られるフィールド上で，ERS-210 が 4 対 4 でサッカーを行う．

3·2 最適方策の作成

Fig.2 のようにフィールド座標系 Σ_{f} とロボット座標系 Σ_{r} を設定する．Σ_{f} の x 軸の先にあるゴールが，ロボットの守るゴールである．状態は，Σ_{f} でのロボットの位置・向きを表す (x,y,θ) と，Σ_{r} でのボールの位置で定義する．ボールの位置は，Σ_{r} からのボールの距離 r

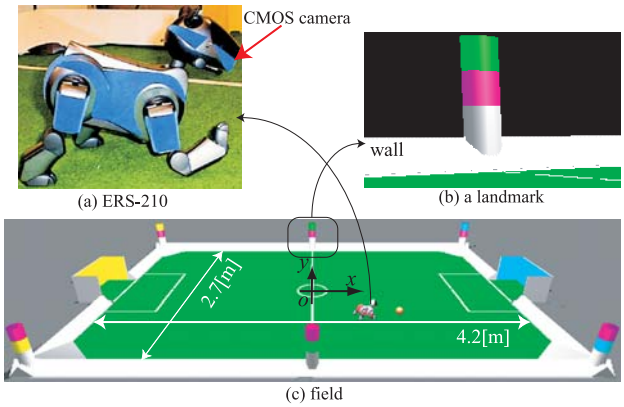

Fig.1 The Field and Robot for RoboCup Four Legged Robot League

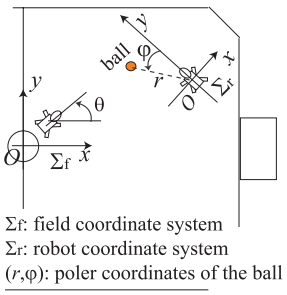

Fig.2 State Variables

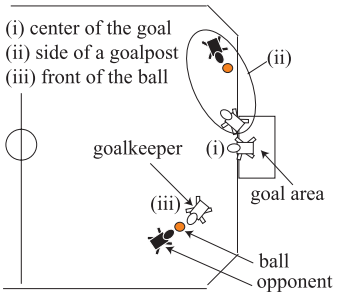

Fig.3 Appropriate Poses

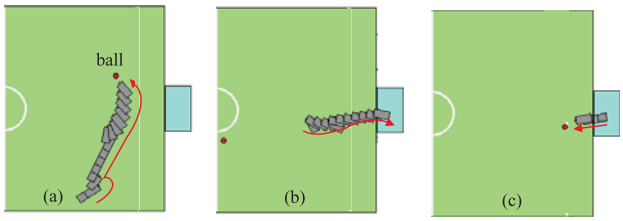

Fig.4 Behavior of The Robot with π^*

と，y 軸からのボールの偏角 φ で表す．つまり，状態変数は $x, y, \theta, r\varphi$ の 5 個である．状態空間は，Table 1(a) で設定した各区間の直積で定義する．ただし，ボールがここで設定した区間の外にある場合，CMOS カメラでボールを観測できなくなるため，(r, φ) は特別な値 *ball invisible* をとることし，*ball invisible* の際の $xy\theta$ 空間を $\mathcal{X}$ に加える．

ロボットに 40 種類の歩行行動を実装する．各歩行行動は，ロボットの位置と向きを決められた分だけ移動させる．40 種類中には，100[mm] 程度の前進・後退，30[deg] 程度の回転の他，斜め方向に移動するための歩容，歩幅の小さい前進・後退・回転等が含まれる．

方策，状態価値関数を計算するため，Table 1(a) に見られる区間の幅，個数で状態空間を離散化する．ボールが見える離散状態の個数は $2,973,348$，*ball invisible* では $6,804$ で，離散状態の数は合計 $2,980,152$ となる．

報酬（この問題設定ではコストと呼ぶのが適当）については Table 1(b) のように 4 種類与える．表中の (1) は一回行動を行う際に生じる時間のコストを -1 としている．(2) はボールを見失う，(3) はボールもゴールも観測不可能で自殺点の恐れがあること，(4) は $\mathcal{X}$ から状態が外れてしまう（壁にぶつかる，ボールに不用意に触れる）状態に対応している．(2)-(4) の値に関しては，筆者らの経験等から与えた値であり，キーパー行動に最適な値とは限らない．

終端状態については，Fig.3 のように，キーパーが (i) ゴールに待機する位置，(ii) ゴールポストを閉める位置，(iii) ボールを確保できる位置，の 3 種類を定める．これらに対応する状態変数の条件と価値を Table 1(c) のように設定する．

上記設定で動的計画法を適用し，方策と状態価値関数を得た．計算に要した時間は，3.6 GHz CPU を有する計算機で 49 分であった．得られた方策，状態価値関数を最適とみなし，それぞれ π^*, V^* とする．Figure 4 は，π^* から状態が既知であることを前提として得られたロボットの行動の例である．

Table 1 Parameters for Value Iteration

(a) State Space

state variable	domain	width of a cell	# of cells
x[mm]	$[1000, 2400)$	100	14
y[mm]	$[-1350, 1350)$	100	27
θ[deg]	$[-180, 180)$	20	18
r[mm]	$[120, 2020)$	100	19
φ[deg]	$[-92, 92]$	8	23

(b) Reward (cost) $\mathcal{R}^a_{ss'}$

case	reward
(1) Action a is executed.	-1
(2) $s : \lvert\varphi\rvert \leq 92$ and $s' : \lvert\varphi\rvert > 92$	-5
(3) s' satisfies $\lvert\varphi\rvert > 92$[deg], $\lvert\varphi_{\rm G}\rvert > 30$ and $x < 2100$.	-5
(4) s' : out of the domain	-250

(c) Final States

case (see Fig.3)	value
(i) *ball invisible* & $\lvert x - 2200\rvert < 100$ & $\lvert y\rvert < 100$ & $\lvert\theta\rvert > 150$	-15
(ii) $2100 < x < 2200$ & $\lvert\theta\rvert > 150$ & $\{\lvert\varphi\rvert < 12$ or $(\varphi < 0$ & $y > 50)$ or $(\varphi > 0$ & $y < -50)\}$	-10
(iii) $\varphi < 45$ & $r < 200$ & $\{\lvert\theta\rvert > 135$ or $\lvert\varphi_{\rm G}\rvert > 135\}$	0

3·3 提案手法の実装と行動決定に要する時間の計測

パーティクルフィルタとして，uniform Monte Carlo localization [7] を用い，実時間 Q-MDP 法を実装した．このときのパーティクルの数は $N = 1000$ とした．投票では $c = m$ として，$g(a) \neq 0$ の行動のみに対して $Q_{\rm MDP}$ を計算するという実装方法をとったが，このとき ERS-210 の 192MHz CPU では，一回の行動選択で 63[ms] 以下の計算時間であった．

実際に使用している様子を示すため，試合でのキーパーの行動例を Fig.5 に示す．このキーパーは，上記の実装に加え，ボールが付近にある場合にボールをはじく動作が実装されている．この試合では 20 分間に渡って常に攻撃を受ける展開であったが，許した失点は 2 点であった．ロボットは，相手がボールを移動すると，それに応じて Fig.3 の 3 種類の終端状態間を移動し，シュートを防いだ．

3·4 自己位置の不確かさを考慮しない場合との比較

まず，不確かさを考慮することの効果を示すため，実装した手法と不確かさを考慮しない方法を比較する．不

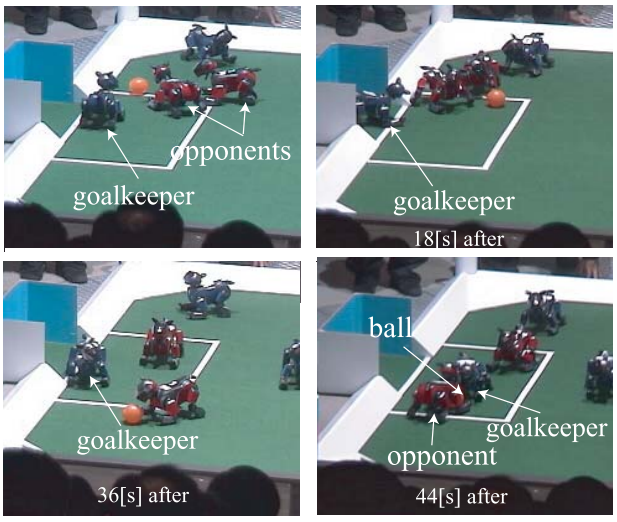

Fig.5 Scenes in a Game (Team ARAIBO 0-2 Kyushu Institute of Technology)

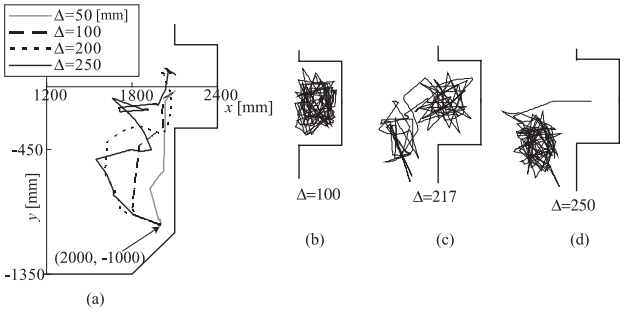

Fig.6 Behavior of The Robot with Q_{MDP}

確かさを考慮しない方法として，パーティクルの姿勢の重み付き平均で現在の状態を推定し，π^* を用いて行動決定する手法を 2 種類実装した．一方（実装 1）は，提案手法の実装と同様，不確かでも行動選択を行い，他方（実装 2）は，無駄な歩行で壁に衝突することを避けるため，パーティクルの分布の標準偏差が閾値（x, y 軸方向でどちらかが 1[m]，あるいは θ 軸で 90[deg]）を超えると歩行を止めるコードを加えた．どの実装でも，ロボットは左右に CMOS カメラを振り，ランドマーク（Fig.1(b)）観測から自己位置推定しながら行動する．

比較の指標として，ロボットをある未知の姿勢に置き，そこからゴール（$x > 2100$ のエリア）に戻るまでの時間を用いる．この実験では，前述のキーパー行動の方策を用いず，フィールド全域からゴールへ帰還するための方策を作成し，それを用いた．初期姿勢は次の 10 点：$(x, y, \theta) = (1000, 0, 45), (1000, 0, 135), (1000, 500, \pm 45), (1000, 500, \pm 135), (1000, 1000, \pm 45), (1000, 1000, \pm 135)$ を選び，各点について 2 回ずつ試行を行って終端状態までの秒数を求めた．結果，実時間 Q-MDP 法では 44[s]，実装 2 では 56[s]，実装 1 だと 20 回のうち 180[s] 以上たってもゴールに到達できない試行があり，それを除外して平均をとっても 76[s] であった．また，実時間 Q-MDP 法を用いた試行のほとんどでは，ロボットは試行の早い段階でゴール前に到達し，長い時間ゴール前でランダムに歩行してから突然ゴールに戻るという行動を示した．これは，壁に衝突せずにゴールに入るために必要な情報が得られるまでゴール前に待機していたものと考えられる．

3.5 不確かさの程度の変化による行動パターンの変化

自己位置の不確かさを考慮した行動が選択されているか，シミュレーションで挙動を調べる．Fig.6 は，ロボットの真の位置 (x, y) を中心とする矩形内に一様にパーティクルを散布して，その散布された幅を変化させてロボットの挙動を調べる．パーティクルの方向は，常に真の θ と一致させた．

まず，ロボットを初期位置 $(x, y, \theta) = (2000, -1000, 0)$ におき，ロボットがゴールに入る（$x > 2100$ となる）までの挙動を記録したものを Fig.6(a) に示す．図中の Δ は，真の位置から矩形の辺までの距離である．$\Delta = 200, 100, 50$[mm] のときは，Δ が大きいほど壁を避けてゴールに戻る行動が見られる．$\Delta = 250$[mm] では軌道が乱れているが，これは，Table 1(b) のコスト (3),(4) の影響である．

次に，ロボットの初期位置をゴールの中 $(x, y, \theta) = (2200, 0, 0)$ にしてその後のロボットの行動を 180 歩分記録してゴール付近の xy 平面にプロットしたものを Fig.6(b)-(d) に示す．$\Delta = 250$[mm] のときは，ロボットはゴールから出て，ゴール方向を向きながら待機する行動をとった．この行動がキーパーにとって最適かどうかは不明だが，壁への衝突と自殺点のコストが影響して発現した行動と考えられる．

4. 結論と今後の展望

実時間 Q-MDP 法を提案し，ロボットサッカーのゴールキーパーに適用した．3 百万の離散状態（ロボットの姿勢に対しては 6,804 状態）に対して得られた最適方策，最適状態価値関数と，ロボットの姿勢の不確かさを表現するパーティクルフィルタ（パーティクル数 1,000）を利用し，実時間 Q-MDP 法を実行した．結果，192MHz CPU を有するロボットで 63[ms] 以下で Q-MDP 値が計算できた．実機実験では，提案手法を実装したキーパーが，情報の不確かさを考慮しない場合に比べて 2 倍（実装 1 と比較），1.3 倍（実装 2 と比較）短時間でゴールに戻ることを確認した．また，シミュレーションでは，キーパーが不確かな自己位置を考慮して待機位置を決めることを確認した．

一方で様々な課題が残されている．Q-MDP 法が機能するためには状態推定に関する確率分布が実際を良く表している必要がある．一方，通常のパーティクルフィルタでは，状態を同定するために確率計算しているにすぎず，その計算過程での確率分布が実際をよく表現しているかどうかはあまり問題とされない．本稿で使用された自己位置推定法では，確率分布を狭くしすぎない工夫 [7] がされており，このような問題は生じなかったが，状態を一つに同定する能力は高くない．そこで，状態の同定を重視して最適方策 π^* を用いる場合と，確率分布の表現を重視して Q-MDP 法を用いる場合の比較が必要となる．また，観測行動を含めた行動決定を Q-MDP 法と組み合わせて実現することも必要である．この際，Q-MDP 法の単純な数式の枠組みを崩さずに，ロボットを知的に振舞わせることは，興味深い課題である．

参考文献

[1] Jérôme Barraquand and Pierre Ferbach. Motion Planning with Uncertainty: The Information Space Approach. In *Proc. of ICRA*, pp. 1341–1348, 1995.

[2] Nicholas Roy, et al. Coastal Navigation - Mobile Robot Navigation with Uncertainty in Dynamic Environments. In *Proc. of ICRA*, pp. 35–40, 1999.

[3] Michael L. Littman, et al. Learning Policies for Partially Observable Environments: Scaling Up. In *Proceedings of International Conference on Machine Learning*, pp. 362–370, 1995.

[4] Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

[5] Dieter Fox. Adapting the Sample Size in Particle Filters Through KLD-Sampling. *International Journal of Robotics Research*, Vol. 22, No. 12, pp. 985–1004, 2003.

[6] 大橋健. 4 足ロボットリーグのとりくみ. 日本ロボット学会誌, Vol. 20, No. 1, pp. 45–46, 2002.

[7] Ryuichi Ueda, et al. Uniform Monte Carlo Localization – Fast and Robust Self-localization Method for Mobile Robots. In *Proc. of ICRA*, pp. 1353–1358, 2002.