強化学習における高次元数の行動空間の扱いについて
ーハッシュと Gibbs-Sampling を用いた行動選択方法の提案ー

○木村 元
九州大学 大学院工学研究院 海洋システム工学部門

Reinforcement Learning with an action-selection scheme using Hash and Gibbs-sampling

○ Hajime Kimura
Dept. of Marine Engineering, Graduate School of Engineering, Kyushu University

Abstract: In real-robot applications, learning controllers are often required to obtain control rules over high-dimensional continuous state-action space. Hash coding is a promising method to deal with high-dimensional state space for representing the state value function. However, there is no standard reinforcement learning scheme to deal with action selection in high-dimensional action space, especially the probability of action variables are mutually dependent. This paper introduces a new action selection scheme using Hash coding and Gibbs sampling, and shows Q-learning and sarsa algorithms for the proposed scheme. We demonstrate it through simulations of a multi-legged robot learning problem.

1 はじめに

強化学習は，ロボットの制御規則を自ら発見し改善していくための学習制御方法として有望である [4]．脚型移動ロボットやヒューマノイド型ロボットなどの多数のアクチュエータを有するロボットにおいて動作規則を学習する場合、状態空間だけでなく行動空間についても高次元で膨大な空間であり、状態の評価方法だけでなく行動空間での行動選択方法および学習方法にも工夫が必要である．

代表的な強化学習法である Q-learning[9] や SARSA[8] は離散的な状態・行動を対象としている．そのため連続な状態空間や行動空間における状態評価関数 (state value function) や状態-行動評価関数 (state-action value function) を表現するために関数近似として CMAC を用いる方法 [8]，ファジィを用いる方法 [2]，過去の経験を用いて補間する方法 [1][6] などが提案されている．特に状態空間の次元数が高い場合，次元の呪いを回避するための有望な手段としてタイルコーディングの一種であるハッシュを使うべきという主張がされている [8]．これは，ランダムに選ばれたいくつかの状態変数で作られる空間中に大きさや位置がランダムなタイルを配置し，そのタイルを，状態価値関数を表現するための特徴ベクトル要素１つに対応させる．状態入力がそのタイル領域内に入ると対応する特徴ベクトル要素の値が１になり，それ以外の場合はゼロになる．このハッシュコーディングによる状態特徴ベクトル生成は，タイルの初期配置が学習性能に大きな影響を与える問題点があるが，依存関係の考慮が必要な状態変数集合とそうでない状態変数を区別したり，重要な状態変数値の組合せをエキスパートの知識から予め与えておくことが容易など多くの利点を有する．このように，高次元空間において状態評価値や状態-行動評価値の汎化については多くの手法が提案されているが，高次元の行動空間における行動選択方法および学習方法については，あまり注目されてこなかった．連続な行動空間を扱うための最も単純な接近法としては，行動空間を全てメッシュに区切り，状態入力に応じて各行動メッシュに対して Q 値に応じた確率を割当てることが考えられるが，行動空間の次元が高くなると「次元の呪い」によってたちまち空間爆発を起こし，記憶容量的にも計算量的にも実行不能になってしまう．また，せっかく膨大な空間を「汎化」によって少ないパラメータで扱おうとしているのに，細かい離散化によってパラメータ数を爆発させては意味がない．だからといって離散化を粗くしすぎてもきめ細かな行動選択ができないため制御の質が落ちるというジレンマがあった．

本論文では，まずハッシュの一種であるランダムタイリングを用いて状態-行動空間を汎化し，Q 値や行動選択確率分布を表現する方法を提案する．次に，多次元で連続な行動空間を細かく離散化するが，Gibbs サンプリングによって空間爆発を回避しつつ行動選択を行う方法を提案する．Q-learning や SARSA など，これまで代表的強化学習手法だったにもかかわらず，行動選択の困難さから膨大な行動空間を持つロボットへ実装することが困難だったが，本論文の手法により，容易に実装できることを示す．

2 問題の定式化

状態空間を $\mathcal{S}$，行動空間を $\mathcal{A}$，上下界を持つ実数の集合を $\mathcal{R}$ と表す．各時刻 t でエージェントは状態観測 $s_t \in \mathcal{S}$ に基づいて行動 $a_t \in \mathcal{A}$ を実行し，状態遷移に伴う報酬 $r_t \in \mathcal{R}$ を得る．本論文が環境のモデルとして仮定するマルコフ決定過程 (MDP) では，一般に次の状態や報酬は確率的で，その分布は s_t と a_t にのみ依存する．MDP では次の状態 s_{t+1} は遷移確率 $T(s_t, a, s_{t+1})$ に従って決まり，報酬 r_t も期待値 $r(s_t, a)$ によって与えられる．エージェントは予め $T(s_t, a, s_{t+1})$ や $r(s_t, a)$ についての知識を持っていない．強化学習の目的はエージェントのパフォーマンスを最適化する政策を得ることである．無限期間のタスクにおける自然な評価規範として，以下のような割引報酬の合計がある．

$$V_t = \sum_{k=0}^{\infty} \gamma^k \, r_{t+k} \ , \tag{1}$$

割引率 $0 \leq \gamma \leq 1$ は未来に得るであろう報酬の現時点での重要度を表し，V_t は時刻 t の評価値 (value) を表す．MDP

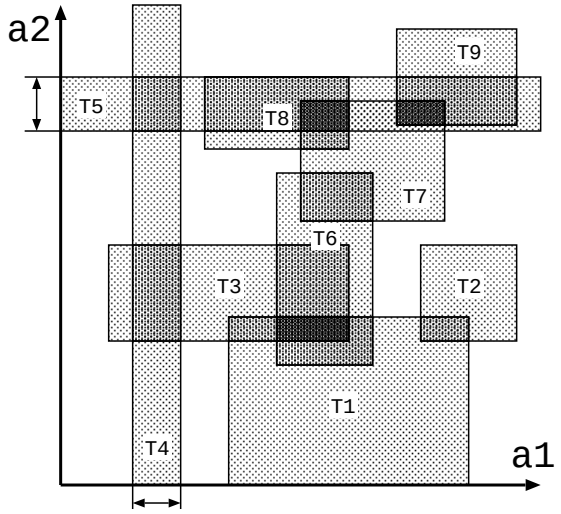

Figure 1: 2次元行動空間での 9 個のタイルによるランダムタイリングの例．タイル T4 および T5 はそれぞれ部分空間 a1, a2 の矢印で示される区間で定義されるタイルのため，それ以外の空間では全領域をカバーするタイルになる．

では評価関数は以下に定義される．

$$V^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t r_t \middle| s_0 = s, \pi\right] , \qquad (2)$$

ただし $E\{\cdot\}$ は期待値を表す．MDP における学習の目的は，各状態 s において式 2 で定義される評価値を最大化するような最適政策を見つけることである．本論文で扱う環境では，状態空間 $\mathcal{S}$ および行動空間 $\mathcal{A}$ は多次元空間で表されるものとする．

3 ランダムタイリングによる高次元状態-行動空間の汎化方法の提案

高次元空間において関数を汎化する方法は多種多様であるが，本論文では多数のランダムタイルを用いる方法を提案する．これは状態空間における汎化のためのタイルコーディングの一種であるハッシュに基づいたアイデアだが，行動空間については情報の流れ方が逆方向になっている．あるランダムタイル $f(x)$ は，入力空間 x を構成する次元のうち，任意の 1 次元以上の複数次元で定義される部分空間中のある矩形領域を表し，入力された座標 x がその矩形領域である場合 $f(x) = 1$ を出力し，それ以外の場合 $f(x) = 0$ である．矩形領域を構成する次元や矩形の範囲・大きさなどは，タイル毎にランダムに与える．このように，ある 1 つのタイルは空間を構成する次元の一部分で構成される空間の，とある領域として処理するが，このタイルを空間全体からみると，定義される部分空間中では範囲の限定された矩形だが，それ以外の空間では全領域をカバーするタイルと同義である．このようなランダムタイルを多数用いて重み付け線形和することにより，空間全体に対して関数近似（汎化）を行う．本論文では，状態空間に対するランダムタイル $f_j(s)$（ただし $j \in \{1, 2, \ldots T_s\}$）および行動空間に対するランダムタイル $f_k(a)$（ただし $k \in \{1, 2, \ldots T_a\}$）の 2 種類のタイル集合による特徴量ベクトルを生成する．特徴量ベクトルを用いて関数近似を行う場合，絶対和ノルムが一定値（理想的には 1）であることが望ましいので，1 から各タイルの値を引いた値を要素とするベクトルを生成し，これを付加する．すなわち，特徴量ベクトルはタイル数の 2 倍の要素になる．さらに，特徴量ベクトルは絶対和ノルムが 1 になるよう正規化する．状態に対するタイル特徴量 $f_j(s)$ と行動に対するタイル $f_k(a)$ の間は重み変数 w_{jk} で接続する．状態 s，行動 a に対する状態-行動評価値（Q 値）は以下のように計算する：

$$Q(s,a) = \sum_{j=1}^{2T_s} \sum_{k=1}^{2T_a} f_j(s)\, f_k(a)\, w_{jk} \qquad (3)$$

この Q 値を温度パラメータ T で除したボルツマン分布に従い，行動を確率的に選択する．各重み変数の値を調節することにより，Q 値および行動選択確率分布が変化する．本手法には以下の特徴がある：

- ランダムタイルが定義される行動空間の部分空間によって行動変数間の依存関係が表現できる．
- 定義されている行動部分空間が互いに干渉していないランダムタイルが存在することにより，互いの空間において独立に行動を出力することが必要なタスクを学習できる．例えば，ある行動変数の集合は固定された特定の値だけ，別の行動変数の集合では 0〜1 の値を一様に出力することが求められるような場合にも対処できる．
- ある状態入力において，行動空間中のある特定の領域と，別の特定の領域の行動のどちらかを半々くらいの確率で出力することが求められる場合，それぞれに対応するタイルの重みを大きくすることで対応できる．

このように興味深い特徴が期待されるが，高次元の行動空間全体に対して定義された重み関数のボルツマン分布に従う行動選択は，そのまま実行しようとすると組合せ爆発を起こしてしまうため工夫が必要である．

4 高次元空間での複雑な確率分布によるサンプリングと学習

4.1 従来手法：フラットな行動選択

ランダムタイリングによって高次元空間において複雑な状態-行動評価関数（Q 関数）や行動選択確率分布関数が表現できるが，この分布に従って行動を選択するための最も原理的に単純な方法は，フラットな行動選択法である．これは，行動空間を全て細かい格子状に離散化し，各超矩形領域に対して確率を割当て、ルーレット選択によっていずれかの超矩形領域に相当する行動を選択するものである．本論文では，連続な行動空間を各次元毎に等しく D 分割して格子状に離散化する．よって行動空間を N 次元と仮定すると D^N コの行動 a_i $(i = \{1, \cdots, D^N\})$ へ離散化さ

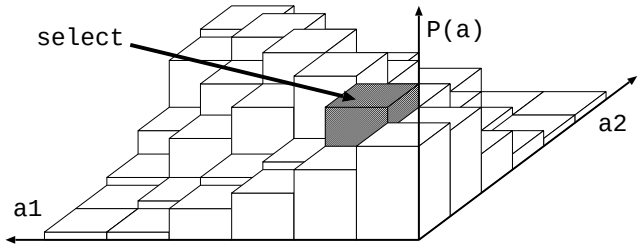

Figure 2: 2次元行動空間を格子状に分割し，各メッシュの持つ確率に応じてフラットに行動選択する様子．$a1-a2$ は行動空間を表し，たて軸は確率を表す．

れる．行動は，各行動 a_i 毎に定義された確率 $P(a_i)$ に従って，どれか１つが選ばれる．Fig.2 は 2 次元の行動空間におけるフラットな行動選択の様子を示す．この例では 2 次元空間 $a1-a2$ で離散化された全ての格子について確率を計算し，その比率に基づいて格子を一つ選択して行動出力としている．

本論文では，行動の確率 $P(a_i)$ は，一般に状態 s において行動 a_i に割り当てられた Q 値 $Q(s,a_i)$ に応じた以下のボルツマン分布とする：

$$P(a_i) = \frac{\exp(Q(s,a_i)/T)}{\sum_{j=1}^{D^N}\exp(Q(s,a_j)/T)} \tag{4}$$

ただし T は正の値をとる温度パラメータであり，値が大きいと全行動の確率が均一に近づき，ゼロに近づくと大きな Q 値を持つ行動の確率が大きくなる．このとき，各行動の $Q(s,a_i)$ はランダムタイルの値と重み変数の線形和から計算される．

このフラットな行動選択方法は，式 4 の示すとおり全行動空間の Q 値を調べる必要があるため，行動次元数が 2〜3 次元程度ならばある程度実用になるが，次元数が高くなると離散化した行動空間が指数的に増大するため，記憶容量的にも速度的にも実行不能になる．

4.2 Gibbs サンプリングによる行動選択法の提案

高次元空間における確率分布に従って効率良くサンプリングするための方法の１つである Gibbs サンプリングを用いる行動選択法と適正度の計算法を提案する．Gibbs サンプリングとは，高次元の確率変数において，注目している次元以外の次元の変数の値を固定し，そのときの条件付確率分布を用いて 1 次元ずつサンプルを行っていく処理を全ての次元に対して十分な回数繰返し，最終的に得た値をサンプルとする Markov chain Monte-Carlo 法（MCMC 法）の一種である [3]．Fig.3 は 2 次元の行動空間における Gibbs サンプリングの様子を示す．2 次元空間 $a1-a2$ で離散化された全ての格子における確率は Fig.2 のフラット選択の場合と同じだが，1 次元ずつサンプルを行っていくので，Fig.2 のような全行動空間における確率を全て計算するような処理が不要であることが分かる．Fig.3 の右上 (1) は $a2$ を固定した条件付確率により $a1$ をサンプルし，左下 (2) では (1) で選んだ $a1$ を固定した条件付確率により $a2$ をサンプルしている．これで反復 1 回分であり，さらに右下 (3) では (2) で選んだ $a2$ を固定した条件付確率により再び $a1$ を選んでいる．このような反復を十分な回数繰返した結果得られた $a1, a2$ を最終的な行動出力とする．Gibbs サンプリングでは，反復回数についての理論的な下限についてはまだ明らかにはされておらず，実験的に決められているのが実情である．

Fig.3 で示した例題のように 2 次元程度の空間を 6×6 に粗く分割した程度では，フラットな行動選択との計算量的な差はあまりないが，行動次元数や分割数が増加すると，その差は顕著に現れる．例えば行動空間が 8 次元で各次元を 10 分割する場合，フラットな行動選択においては各メッシュにおける重みの計算を 10^8 =1 千万回行わなければならないが，Gibbs サンプリングでは $10\times 8\times$（反復回数）程度であり，反復回数を数十回程度に抑えればフラットな行動選択法に比べて１万分の１程度の計算量で済むことになる．

Gibbs サンプリングでは行動選択確率分布自体は式 4 と同じだが，各行動座標軸毎に処理を行うため，記法を以下のように対応させる．フラットな行動選択の場合同様，行動の次元数 N で，行動空間は各次元毎に D 分割により離散化されているものとする．ある多次元行動 a について，各行動次元毎に分解して次のように表す:$a=(a^1,a^2,\ldots a^N)$ ただし各次元の要素 a^n (ただし $n\in\{1,2,\cdots N\}$) は $a^n\in a_d^n$ ただし $d\in\{1,2,\cdots D\}$ である．ある状態 s 行動 a に対する Q 値は，$Q(s,a)=Q(a^1,a^2,\ldots a^N|s)$ と表す．この Q 値は，フラット選択における値と同一である．Gibbs サンプリングの t 回目の反復においてサンプルされた行動要素を $a^1(t),a^2(t),\ldots a^N(t)$ と記する．このとき，$t+1$ 回目の反復における行動要素は以下の確率分布に従ってサンプルされる：

$$\begin{aligned}
a^1(t+1) &\sim P(a^1|a^2(t),a^3(t),\ldots a^N(t))\\
a^2(t+1) &\sim P(a^2|a^1(t),a^3(t),\ldots a^N(t))\\
&\vdots\\
a^N(t+1) &\sim P(a^N|a^1(t),a^2(t),\ldots a^{N-1}(t))
\end{aligned}$$

ここで，条件付き確率 $P(a_i^n|a^1,a^2,\ldots a^N)$ は，以下のボルツマン分布で与えられる：

$$P(a_i^n|a^1,a^2,\ldots a^N) = \frac{\exp(Q(a^1,a^2,\ldots a_i^n,\ldots a^N|s)/T)}{\sum_{d=1}^{D}\exp(Q(a^1,a^2,\ldots a_d^n,\ldots a^N|s)/T)} \tag{5}$$

よってフラットな行動選択と比べると，考慮すべき行動が１次元ずつになっただけで式 4 と同じような処理を繰り返し行うようになっただけである．このとき，全行動空間について Q 値を調べる必要がないのが大きな利点である．

4.3 強化学習アルゴリズム

ランダムタイルによる空間汎化と Gibbs サンプリングによる行動選択を強化学習アルゴリズムと組み合わせる．ここでは Q-learning 法と sarsa アルゴリズムを取り上げる．

状態空間に対するランダムタイル $f_j(s)$（ただし $j\in\{1,2,\ldots T_s\}$）および行動空間に対するランダムタイル $f_k(a)$（ただし $k\in\{1,2,\ldots T_a\}$）の２種類のタイル集合により

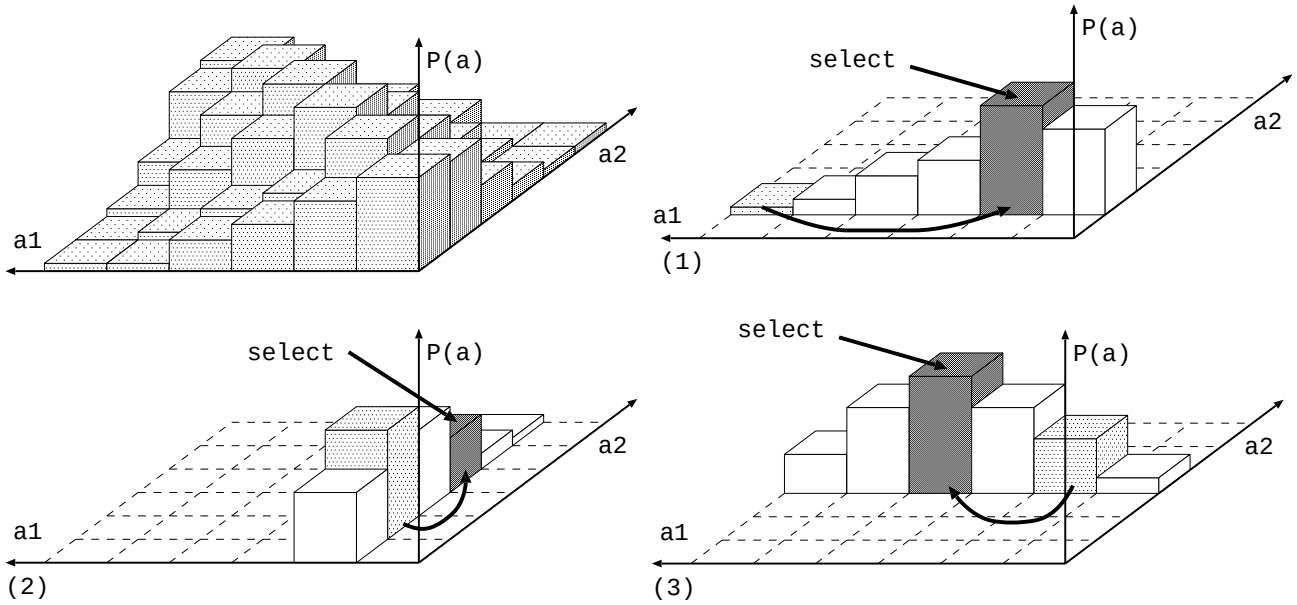

Figure 3: 格子状に分割された 2 次元行動空間を Gibbs サンプリングする様子．左上の図は行動選択確率分布，右上 (1) は $a2$ を固定したときの条件付確率により $a1$ を選ぶ様子，左下 (2) は (1) で選んだ $a1$ を固定した条件付確率により $a2$ を選ぶ様子，右下 (3) は (2) で選んだ $a2$ を固定した条件付確率により再び $a1$ を選ぶ様子．$a1 - a2$ は行動空間を表し，たて軸は確率を表す．

特徴量ベクトルを生成する．状態に対するタイル $f_j(s)$ と行動に対するタイル $f_k(a)$ の間は重み変数 w_{jk} で接続し，状態 s，行動 a に対する状態-行動評価値（Q 値）は以下のように計算する：

$$Q(s,a) = \sum_{j=1}^{T_s} \sum_{k=1}^{T_a} f_j(s)\, f_k(a)\, w_{jk} \tag{6}$$

エージェントは状態 s を観測し，以下のボルツマン分布に従って行動 a_i を選択し実行する：

$$P(a_i) = \frac{\exp(Q(s,a_i)/T)}{\sum_{j=1}^{D^N} \exp(Q(s,a_j)/T)} \tag{7}$$

ただし行動は式 5 の Gibbs サンプリングにより選択するため，全行動について Q 値を計算する必要は無い．行動選択後，報酬 r と遷移先の状態 s' を観測する．遷移先での状態 s' における Q 値を使い，もとの状態 s で実行した行動 a_i の Q 値を更新する：

$$Q(s,a_i) \leftarrow Q(s,a_i) + \alpha \left(r + \gamma \max_a Q(s',a) - Q(s,a_i) \right) \tag{8}$$

ただし γ は割引率，α は学習率（$0 \leq \alpha \leq 1$）である．ここで Q 値は式 6 で表されているので，式 8 で示される Q 値の更新は，重みパラメータ w_{jk} を以下のように更新することで実現する：

$$w_{jk} \leftarrow w_{jk} + f_j(s)\, f_k(a_i)\, \alpha \left(r + \gamma \max_a Q(s',a) - Q(s,a_i) \right) \tag{9}$$

ここで $\max_a Q(s',a)$ を探す必要があるが，これはボルツマン分布の温度パラメータ T をゼロに近づけて Gibbs サンプリングによって得た行動の Q 値で代用する．

次に sarsa アルゴリズムだが，Q 値の更新方法が異なるだけである：状態 s において Q-learning 同様ボルツマン分布で行動 a_i を選択後，遷移先での状態 s' において行動 a' を選択し，以下のように Q 値を更新する：

$$Q(s,a_i) \leftarrow Q(s,a_i) + \alpha \left(r + \gamma Q(s',a') - Q(s,a_i) \right) \tag{10}$$

ここで Q 値は式 6 で表されているので，式 10 で示される Q 値の更新は，重みパラメータ w_{jk} を以下のように更新することで行う：

$$w_{jk} \leftarrow w_{jk} + f_j(s)\, f_k(a_i)\, \alpha \left(r + \gamma Q(s',a') - Q(s,a_i) \right) \tag{11}$$

このように遷移先の状態で最大の Q 値を探索する必要がないため，sarsa アルゴリズムの方が Q-learning よりも実装が簡単である．

5 実験

Fig.4 に示す仮想的なほふくロボットに本手法を適用する．学習目標は，ロボットを前進させるために，アームを足のように作用させる動作の獲得である．関節は位置制御のサーボモーターによって角度を制御される．各時間ステップにおいて，エージェントは 8 つの関節モータの角度および 4 つの足先のタッチセンサの状態という 12 個の状態量を要素とした 12 次元ベクトルを観測する：関節角度 $\phi_1, \ldots \phi_8$ は 8 次元連続空間で定義され，タッチセンサ $\phi_9, \ldots \phi_{12}$ は 0 または 1 の 2 値である．

行動は，関節角度の目標値を指示し，8 次元ベクトル $(a^1, \ldots a^8)$ の各要素がそれぞれ関節角度を表す．行動ベクトルの各要素は $[0,1]$ の範囲に限定される．行動が選ばれると，モータは指示された目標位置へ動きはじめる．関節

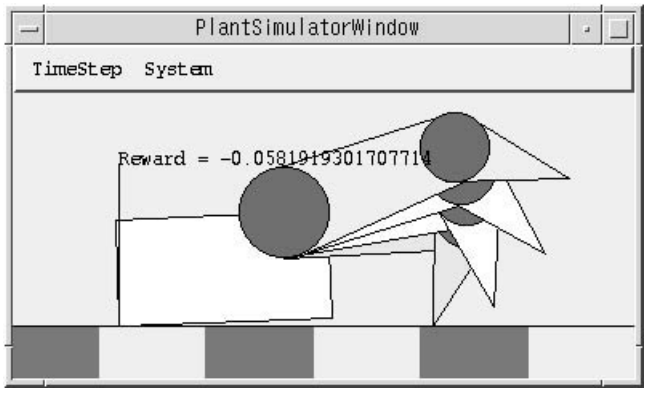

Figure 4: 多自由度ほふくロボットシミュレータ．右側が前方．足は 4 本で，各足にはモータ 2 個ずつ計 8 個取り付けられている．各足の先端にはタッチセンサが付いている．状態は 8 関節の角度と 4 個のタッチセンサの値の計 12 次元で，行動は 8 個の関節の目標値で計 8 次元である．

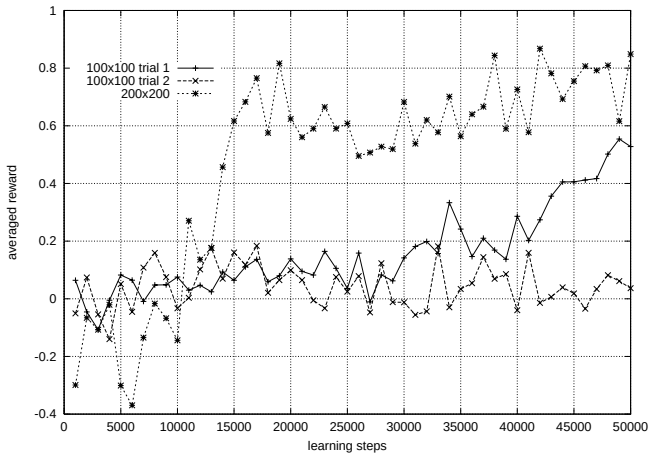

Figure 5: ほふくロボット（状態入力 12 次元，行動出力 8 次元）における学習の様子．横軸は学習ステップ，縦軸は 1000 ステップ毎の平均報酬を表す．

角度が指示された位置まで動くか，あるいはタッチセンサの値が変化すると，状態遷移の結果として報酬が与えられ，次の時刻へ進む．関節のモータが目標位置まで動く途中でセンサの値が変化すると，そこで意思決定イベントが発生して動きが打ち切られるため，次のステップでの関節角度は行動として出力された目標角度には一致しない．よって状態遷移には不確実性が存在する．報酬は，ボディが前進した距離与えられる．ロボットが後退した場合，報酬の値は負値になる．

強化学習エージェントは，12 次元の状態入力のうち関節角度 $\phi_1, \ldots \phi_8$ については連続値を持つ各要素を 10 分割して離散化する．タッチセンサ $\phi_9, \ldots \phi_{12}$ についてはそのまま離散値として扱う．また 8 次元の行動 $(a^1, \ldots a^8)$ については，各要素を 10 分割して離散化する．よって離散空間の強化学習問題としては状態数 $10^8 \times 2^4$ =1 億 6 千万，行動数 10^8 =1 千万，状態-行動空間は $2^4 \times 10^{16}$ という膨大な空間になる．この状態空間に対し，ランダムタイルを 100 個生成して状態特徴ベクトル $f_j(s)$ とする．状態のランダムタイルは，各状態次元についてタイル矩形領域の部分空間を構成する要素として選択する確率を 0.3 としてランダムに作成する．ただし要素を１つも選択しなかったタイルは除外する．ランダムタイルの矩形領域については，離散化された空間の境界に合わせて一様な乱数で生成する．行動のランダムタイル $f_k(a)$ も同様に 100 個生成する．強化学習法として，sarsa アルゴリズムを適用する．Gibbs サンプリングの反復回数は 40 回，割引率 $\gamma = 0.9$，ボルツマン分布の温度パラメータ $T = 0.02$ で一定，学習率 $\alpha = 0.4$ に設定する．

Fig.5 に学習の様子を示す．ここでは比較的学習に成功した例と失敗例を示している．成功例 (Fig.5 中の trial 1) においては，行動出力する際，およそ 3〜8 個の行動空間のランダムタイルがアクティブになっている．ランダムタイルの設定によっては，ほとんど学習しない場合 (Fig.5 中の trial 2) もあり，その時アクティブになる行動空間のランダムタイルは 0〜1 個程度だった．状態-行動ランダムタイリングを 200x200 に増やした場合が Fig.5 中の 200x200 である．タイルを増やすと学習が安定する傾向が見られる．これらの学習により得られた動作規則や，その評価値については，従来のガウス分布を用いた Actor-Critic 法 [5] には及ばなかったが，ランダムタイルの配置を工夫したり，学習率やボルツマン分布温度パラメータを適切にスケジューリングすることにより，性能向上が期待できる．

6 考察

- 実験ではいくつかの種類のランダムタイリングを用いて実験したが，ランダムタイルの生成パターンによって，学習性能が大きく変わることがあった．これは 12 次元という高次元の状態に対してたった 100 個のタイルにて特徴量を抽出するのでは少な過ぎるものと考えられる．あまりに少ないタイルで状態-行動空間を表現することは，不完全観測問題に類似した問題を引き起こしてしまうため，学習不安定の要因になる．もっと十分な個数のランダムタイルを生成すれば，性能は安定するものと考えられる．

- 実験において，状態空間に対するランダムタイルの中には，全く使用されていないものが 2〜3 割存在した．これは，実験で用いたシミュレータでは足先のタッチセンサが同時に 2 個以上反応する状態が存在しなかったり，タッチセンサが反応し，かつ対応する足が上に持ち上げられている状態が存在しないなどの事情による．本論文では固定されたランダムタイリングを用いているため上記のような問題が生じているが，新しい未知の状態入力があるたびに，それに対応して適応的に新しいランダムタイルを生成するなどの工夫が考えられる．また，学習初期は大きな領域を持つランダムタイルによって学習し，学習進行に応じて重要と思われる領域へ適応的に小さめのタイルを追加していくことにより，関数近似の精度を上げていくことも考えられる．

- ランダムタイリングによる状態-行動空間の汎化法および Gibbs サンプリングによる行動選択方法は，ど

ちらも計算処理が軽く，リアルタイム動作に向いた方法である．

- ボルツマン分布に従って行動選択は可能であるが，本提案手法を用いて最大の Q 値を探すような計算を厳密に行うことは困難であり，温度パラメータ T をゼロに近づけて近似的に行っている．しかしあまり T をゼロに近づけ過ぎると，単なるローカルサーチになるが，今度は局所解に陥る可能性が出てくる．
- Gibbs サンプリングによって行動を選んだとしても式 4 のボルツマン分布に従った選択になっている．本方法は Actor-Critic 法にも使えるが，式 4 のボルツマン分布について適正度（eligibility）を計算するには，やはり全行動空間の重みを調べて合計する必要があるため，適正度の計算において空間爆発を起こしてしまう問題がある．そこで，Gibbs サンプリングによって行動要素を選択していく過程で適正度を計算して合計していくことにより，この計算を近似することが考えられる．
- 状態-行動空間の汎化方法としてランダムタイリングを提案したが，この部分は別の汎化手法に置き換えることも可能である．

7 おわりに

本論文では，高次元の状態-行動空間における汎化および行動選択方法としてハッシュの一種であるランダムタイリングと，Gibbs サンプリングによる行動選択法を提案した．本手法を sarsa アルゴリズムと組合せ，12 次元の状態および 8 次元の行動空間を持つロボット学習問題へ適用し，高次元の状態-行動空間において学習することを示し，学習はランダムタイルの初期配置に大きく依存する問題があることを示した．学習状況に応じて適応的にタイルを追加・削除する方法の検討は今後の課題である．

References

[1] 深尾 隆則，稲山 典克，足立 紀彦：正則化理論を用いた連続的状態と行動を扱う強化学習，システム制御情報学会論文誌，Vol.11, No.11, pp.593–599 (1998).

[2] 堀内 匡，藤野 昭典，片井 修，椹木 哲夫：連続値入出力を扱うファジィ内挿型 Q-learning の提案，計測自動制御学会論文集, Vol.35, No.2, pp.271–279 (1999).

[3] Jordan, M. I.: Learning in Graphical Models, The MIT Press, (1999).

[4] 木村 元，宮崎 和光，小林 重信：強化学習システムの設計指針，計測と制御, Vol.38, No.10, pp.618–623 (1999).

[5] 木村 元，山下 透，小林 重信：強化学習による４足ロボットの歩行動作獲得，電気学会 電子情報システム部門誌, Vol.122-C, No.3, pp.330–337 (2002).

[6] Santamaria, J. C., Sutton, R. S. & Ram, A.: Experiments with Reinforcement Learning in Problems with Continuous State and Action Spaces, *Adaptive Behavior 6 (2)*, pp.163–218 (1998).

[7] Sutton, R.S.: Generalization in reinforcement learning: Successful examples using sparse coarse coding, *Advances in Neural Information Processing Systems 8 (NIPS8)*, pp.1038–1044 (1996).

[8] Sutton, R.S. & Barto, A.: Reinforcement learning: An introduction, *A Bradford Book*, The MIT Press (1998).

[9] Watkins, C. J. C. H. & Dayan, P.: Technical Note: *Q*-Learning, *Machine Learning 8*, pp.279–292 (1992).