NAIST-IS-MT9951067

修士論文

状態空間を自律的に構成する連続値強化学習法

武田 政宣

2001年2月9日

奈良先端科学技術大学院大学
情報科学研究科 情報システム学専攻

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士 (工学) 授与の要件として提出した修士論文である。

武田 政宣

審査委員：　小笠原 司 教授
伊藤 実 教授
今井正和 助教授

状態空間を自律的に構成する連続値強化学習法*

武田 政宣

内容梗概

強化学習は, ロボットが環境に適応した行動を自律的に獲得する手法として, 近年注目を集めている. 強化学習では状態空間と行動空間を離散化する必要がある. そのため, センサの分解能を無駄にしており, ロボットの動作が滑らかではなかった. 強化学習のなかでも最も広く用いられている Q-learning ではセンサ出力とモータコマンド, それぞれを離散化した状態と行動の組に対して不連続な Q 値を持つ. しかし, 実際にはセンサ出力やモータコマンドは連続値をとなるため, それらに対して連続な Q 値を持つと考えられる.

そこで, ニューラルネットワークや関数近似の手法を用いて状態, 行動,Q 値を連続値として扱う強化学習法が数々提案されている. これらの手法は, 連続な状態に対して最適行動は連続であるという仮定のもとで議論されている. しかし, 実際にはその仮定が成り立たない状態が存在する. この状態において補間などにより連続な行動出力を得ようとすると, 不適切な行動となることがある. これを「最適行動の不連続問題」と呼ぶ.

本研究では線形近似により状態,行動,Q値を連続値として扱うことができる Continuous Valued Q-learning を用いて, 状態空間を適切に構成することによって最適行動の不連続問題に対処できる強化学習法を提案する.

キーワード

強化学習, 状態・行動空間, CVQ-learning, 最適行動の不連続問題

* 奈良先端科学技術大学院大学 情報科学研究科 情報システム学専攻 修士論文, NAIST-IS-MT9951067, 2001 年 2 月 9 日.

Continuous Reinforcement Learning Refineing State Space Automatically*

Masanori Takeda

Abstract

Q-learning, a most widely used reinforcement learning method, normally needs well-defined quantized state and action spaces to obtain an optimal policy for accomplishing a given task. This makes it difficult to be applied to real robot tasks because of poor performance of learned behavior due to the failure of quantization of continuous state and action spaces. To deal with this problem, we proposed a continuous valued Q-learning [11] (hereafter, called CVQ-learning) for real robot applications. This method utilized a function approximation method for representing a action value function. In this paper, we point out that this type of learning method potentially has a discontinuity problem of optimal actions given a state. To resolve this problem, this paper proposes a method for estimating where discontinuity of optimal action takes place and for refining a state space for CVQ-learning. To show the validity of our method, we apply the method to a pendulum swing-up problem. Although the task is simple, the performance is quite impressive.

Keywords:

reinforcement learning, state-action space, CVQ-learning, discontinuity problem of optimal action

* Master's Thesis, Department of Information Systems, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT9951067, February 9, 2001.

目 次

1. はじめに **1**

2. Continuous Valued Q-learning **3**

2.1 状態と行動 3

2.2 行動価値関数 4

2.3 状態価値関数 4

2.4 Q 値の更新 5

3. 最適行動の不連続問題 **6**

3.1 最適行動の不連続問題の例 6

3.2 最適行動の不連続問題の対処法 7

4. Enhanced Continuous Valued Q-learning **8**

4.1 最適行動の不連続問題が起こる状態の推定 8

4.2 代表状態点の追加 8

4.2.1 行動価値習熟率 9

4.3 状態価値習熟率 10

5. 実験 **11**

5.1 1 次元状態・1 次元行動 11

5.1.1 実験方法 11

5.1.2 結果 12

5.1.3 考察 16

5.2 2 次元状態・1 次元行動 - 単振子の振り上げ問題 - 17

5.2.1 実験方法 19

5.2.2 結果 19

5.2.3 考察 20

6. おわりに **36**

謝辞 **37**

参考文献 **38**

図 目 次

1 代表状態ベクトルとその重み 4
2 全方位視覚を持つロボットの状態空間 6
3 シミューレーション画面 / フィールド 11
4 シミュレーション画面 / 全方位画像 12
5 200 試行後の Q 値 13
6 500 試行後の Q 値 13
7 800 試行後の Q 値 14
8 1000 試行後の Q 値 14
9 Q 値の平均の変化 15
10 単振子 17
11 (a) 一定の方向にトルクをかけた場合. (b) 速度と同じ向きにトルクをかけた場合の位置の時間変化 18
12 (a) 一定の方向にトルクをかけた場合. (b) 速度と同じ向きにトルクをかけた場合の位置と角速度 18
13 500 トライアル 状態数 37 セル数 58 CVQ-learning による状態空間と状態遷移 21
14 500 トライアル 状態数 37 セル数 58 CVQ-learning による位置の時間変化 21
15 1000 トライアル 状態数 51 セル数 86 CVQ-learning による状態空間と状態遷 22
16 1000 トライアル 状態数 51 セル数 86 CVQ-learning による位置の時間変 22
17 5000 トライアル 状態数 99 セル数 175 CVQ-learning による状態空間と状態遷移 23
18 5000 トライアル 状態数 99 セル数 175 CVQ-learning による位置の時間変化 23
19 10000 トライアル 状態数 120 セル数 212 CVQ-learning による状態空間と状態遷移 24

20 10000トライアル 状態数120 セル数212 CVQ-learningによる位置の時間変化 24
21 20000トライアル 状態数126 セル数222 CVQ-learningによる状態空間と状態遷移 25
22 20000トライアル 状態数126 セル数222 CVQ-learningによる位置の時間変化 25
23 30000トライアル 状態数130 セル数227 CVQ-learningによる状態空間と状態遷移 26
24 30000トライアル 状態数130 セル数227 CVQ-learningによる位置の時間変化 26
25 40000トライアル 状態数135 セル数237 CVQ-learningによる状態空間と状態遷移 27
26 40000トライアル 状態数135 セル数237 CVQ-learningによる位置の時間変化 27
27 状態数64 Q-learningによる状態価値関数 28
28 状態数256 Q-learningによる状態価値関数 28
29 状態数1024 Q-learningによる状態価値関数 29
30 状態数138 セル数242 CVQ-learningによる状態価値関数 29
31 状態数64 Q-learningによる位置と角速度 30
32 状態数256 Q-learningによる位置と角速度 30
33 状態数1024 Q-learningによる位置と角速度 31
34 状態数138 セル数242 CVQ-learningによる位置と角速度 31
35 状態数64 Q-learningによる位置の時間変化 32
36 状態数256 Q-learningによる位置の時間変化 32
37 状態数1024 Q-learningによる位置の時間変化 33
38 状態数138 セル数242 CVQ-learningによる位置の時間変化 33
39 状態数64 Q-learningによる高さ平均 34
40 状態数256 Q-learningによる高さ平均 34
41 状態数1024 Q-learningによる高さ平均 35

42　状態数 138 セル数 242 CVQ-learning による高さ平均 35

表 目 次

1　学習結果の 100 試行の平均位置 12
2　シミュレーションに用いたパラメータ 19

1. はじめに

複雑な環境に適応する行動を自律ロボットに獲得させるのに, 先見的知識をほとんど必要としない, 強化学習が近年注目されている. 強化学習は, ロボットは環境からのセンサ情報を状態として入力し, その状態に応じた行動をモータコマンドとして出力する. それにより, ロボットの存在する環境が変化し, その環境に応じた報酬がロボットに与えられる. このサイクルを繰り返し, ロボットはより高い報酬を得るように行動することで, 環境に適応していくことができる.

環境がマルコフ性を満たすとき, 必ず最適行動を獲得できることが証明されている Q-learning[1] は, 強化学習で最も広く用いられている手法である. しかし, 一般的な Q-learning では状態空間と行動空間を離散化する必要があるため, 微妙な環境の変化を認識することができず, 実環境でロボットを制御するときに様々な問題が生じる. また, 離散的な行動しか出力できないため, ロボットの動きは滑らかではなく, 環境に対して最適な行動をとれるとは言えない. より最適な行動を獲得するためには, 状態と行動を連続的に扱うことが必要である.

一般の Q-learning では, ある離散化された状態・行動の組に属するものに対しては一定の価値（Q 値）が割り当てられるが, 状態と行動を連続的に扱うためには連続値の状態・行動の組に対しても連続的に Q 値を表現しなければならない. そこで, 状態と行動を細かく離散化する方法が考えられるが, Q-learning では実際に経験した状態・行動の組しか学習することができないため, 経験しなければならない状況が増えることになり, 学習時間が増加してしまう.

これを回避するため小脳モデル神経回路（CMAC）[2] を用いる手法 [3][4] が提案されている. これは, 複数の格子をずらして重ね合わせることによって細かい領域の状態を表現するものである.

その他のアプローチとして, 行動価値を関数近似によって表現する手法があげられる. 関数近似にニューラルネットワークを用いるもの [5] が提案されているが, ネットワークの一部の重みが変化しただけでも全体に影響を与えてしまうため学習がなかなか収束しないという問題を抱えている. そのほか, 粗く離散化された状態・行動の組に対して得られる Q 値を補間し, 任意の状態を複数の状態の重み付き和によって表現する手法 [6][7] がある. 状態を表現する基底関数にガウス

関数を用いる手法として [8][9][10] などがあるが, 複数の関数を組み合わせることによって Q 関数の形状が複雑になってしまい, 行動選択に必要な最大の Q 値を求めるための計算が膨大になることがある.

著者らが提案している Continuous Valued Q-learning [11][12] では, 重み付き和の手法に基づいて任意の状態・行動の組の Q 値を, 代表状態・行動の組がもつ Q 値の線形補間によって定義する. これにより, 学習結果から得られる行動も連続値となるため, ロボットの動きを滑らかにすることができる. しかし, この手法は, 連続な状態に対して最適行動は連続であるという仮定が成り立つことが想定されており, 実際にはその仮定が成り立たない状態が存在する. この状態において補間などにより連続な行動出力を得ようとすると, 不適切な行動となることがある. これを「最適行動の不連続問題」と呼んで指摘する.

本報告では,Continuous Valued Q-learning のアルゴリズムを紹介する. 次に, 重み付き和による連続行動の生成の際の「最適行動の不連続問題」について述べる. さらに,Continuous Valued Q-learning を用い, 状態空間を改良していくことによって, 「最適行動の不連続問題」に対処する手法を提案する.

2. Continuous Valued Q-learning

強化学習で最も広く用いられている手法である Q-learning[1] は, 環境がマルコフ性を満たすとき, 必ず最適行動を獲得できることが証明されている. この Q-learning をもとに, 線形近似を用いることによって状態, 行動,Q 値を連続値として扱えるように改良した,CVQ-learning を紹介する.

2.1 状態と行動

n 次元連続センサ出力 $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ を状態とする. センサ出力の状態空間を適当な間隔で格子状に区切り，格子点が N 個存在したとき, それらの格子点を代表状態ベクトル $\boldsymbol{x}^i = (x_1^i, x_2^i, \cdots, x_n^i)$ $(i = 1, \cdots, N)$ とする．センサ出力の第 k 要素 x_k を挟む二つの代表点の間隔を正規化しておく. 代表状態点 $\boldsymbol{x}^i$ に対する重み w_i^x は次式で計算する.

$$w_i^x = \prod_{k=1}^{n} l_i(x_k)$$

ただし

$$l_i(x_k) = \begin{cases} 1 - |x_k^i - x_k| & (\text{if } |x_k^i - x_k| < 1) \\ 0 & (\text{otherwise}) \end{cases}$$

センサ出力が $2(= n)$ 次元で表されるとき, 図 1 のようにセンサ出力が示す点を含む格子の $4(= 2^n)$ 個の頂点の代表状態ベクトルの重み付き和によって表現する. センサ出力を示す点を通る各軸に垂直な直線を引いたとき, 格子の対角側に構成される長方形の面積が各代表状態ベクトルの重みとなる.

これを用いて, 任意の状態ベクトル $\boldsymbol{x}$ は,N 個の代表状態ベクトル $\boldsymbol{x}^i$ とその重み w_i^x の積の総和

$$\boldsymbol{x} = \sum_{i=1}^{N} w_i^x \boldsymbol{x}^i$$

で表すことができる.

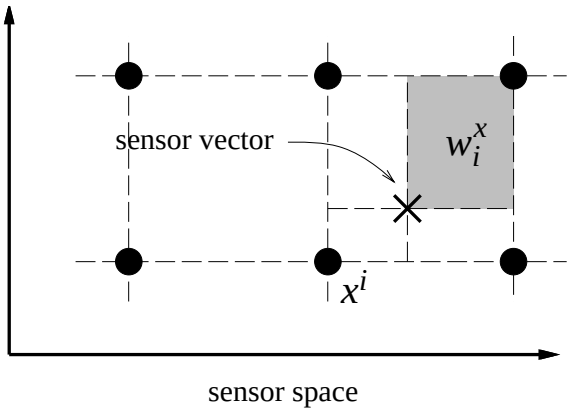

図 1 代表状態ベクトルとその重み

同様に m 次元の行動ベクトル $\boldsymbol{u}$ も M 個の代表行動ベクトル $\boldsymbol{u}^j$ とその重み w_j^u の積の総和で次のように表現される.

$$\boldsymbol{u} = \sum_{j=1}^{M} w_j^u \boldsymbol{u}^j$$

2.2 行動価値関数

ここでは, 各代表状態・行動の組に対する Q 値を線形補間することによって連続的に Q 関数を表現する.

任意の状態 $\boldsymbol{x}$, 行動 $\boldsymbol{u}$ における Q 値は, 代表状態 $\boldsymbol{x}^i$ と代表行動 $\boldsymbol{u}^j$ における Q 値を $Q_{i,j}$ とし, 前節で定義した代表状態・行動に対する重み $\boldsymbol{w}_i^x, \boldsymbol{w}_j^u$ を用いて次式で表現する.

$$Q(\boldsymbol{x}, \boldsymbol{u}) = \sum_{i=1}^{N} \sum_{j=1}^{M} w_i^x w_j^u Q_{i,j}$$

2.3 状態価値関数

行動政策 π に基づいた代表状態 $\boldsymbol{x}_i$ の価値 $V^\pi(\boldsymbol{x}^i)$ は最適行動 $\boldsymbol{u}^*$ が持つ Q 値とし, 最適行動 $\boldsymbol{u}^*$ に対する代表行動の重み $\boldsymbol{w}^{u*} = (w_1^{u^*}, w_2^{u^*}, \ldots, w_M^{u^*})$ を用いて次式

で求められる.

$$\begin{aligned} V^{\pi}(\boldsymbol{x}^i) &= Q(\boldsymbol{x}^i, \boldsymbol{u}^*) \\ &= \sum_{j=1}^{M} w_j^{u*} Q_{i,j} \end{aligned}$$

この離散的な代表状態の価値を状態の重みに応じて線形補間することにより, 状態価値関数 $V^{\pi}(\boldsymbol{x})$ を次式で表現する.

$$\begin{aligned} V^{\pi}(\boldsymbol{x}) &= \sum_{i=1}^{N} w_i^x V^{\pi}(\boldsymbol{x}^i) \\ &= \sum_{i=1}^{N} \sum_{j=1}^{M} w_i^x w_j^{u*} Q_{i,j} \end{aligned}$$

2.4 Q 値の更新

代表状態・行動の重みはそれぞれに対する適合度と考えることができる. 適合度に応じて Q 値の更新率を変化させることによって, Q 値に対する影響を調節しながら,Q 値の更新を行う.

状態 $\boldsymbol{x}$ で行動 $\boldsymbol{u}$ をとり, 次状態 $\boldsymbol{x}'$ へ遷移したときに, 報酬 r が得られたとすると，次式により Q 値の更新を行う.

$$Q_{i,j}^{t+1} = Q_{i,j}^{t} + \alpha w_i^x w_j^u [r + \gamma V^{\pi}(\boldsymbol{x}') - Q^t(\boldsymbol{x}, \boldsymbol{u})] \tag{1}$$

ここで,α は最大の更新率,γ は減衰率を示す.

3. 最適行動の不連続問題

提案手法をはじめ連続な行動を扱う学習方法では, 「連続な状態に対して最適行動は連続である」という仮定を用いているが, 実際には最適行動が不連続になる状態が存在する. 重み付き和により連続値行動を得る場合, 互いに逆符合の行動要素を足し合わせることがある. このとき, もとの行動要素よりも足し合わせて得られた行動の方が出力が小さくなる. もし, この状態における最適行動が出力の大きい行動であれば, 行動要素を足し合わせることは望ましくない. このような状況ではロボットの行動能力を十分に発揮できず, 目標とする状態に到達できない可能性もある. このような問題を「最適行動の不連続問題」と呼ぶことにする.

3.1 最適行動の不連続問題の例

この問題の最も簡単な例として, 全方位視覚を搭載した移動ロボットが, 回転のみの行動によって正面でボールを注視するタスクを考える (図 2 参照).

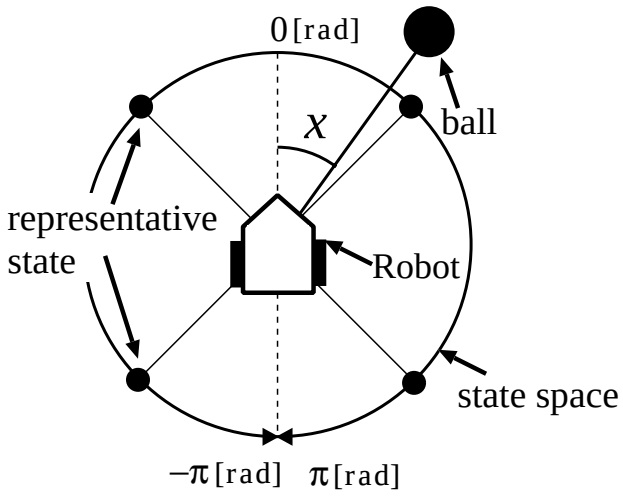

図 2 全方位視覚を持つロボットの状態空間

慣性や摩擦などの影響をを考慮しないとすると, ロボットよりも左側にボールが見えるときは左回転, 右側に見えるときは右回転をすれば最適行動となること

が予測できる. もし, ボールが真後ろで観測された場合, 左右どちらの回転でも最適行動となる. 真後ろにボールがある状態は, 目標姿勢から最も遠い姿勢なので, 目標姿勢により早く近づくようにするためには, 他の状態よりも可能な限り大きな出力によって素速い行動をしなければならないはずである. ところが, 先にも述べたように左回転と右回転という逆の行動要素を足し合わせると, 互いの行動の効果を打ち消し合ってしまい, 最適行動を得ることができない. このように, このタスクにおいてロボットの真後ろにボールが観測される状態では, 最適行動の不連続問題が発生していることがわかる.

3.2 最適行動の不連続問題の対処法

最適行動が不連続になるのに伴い, それらの行動価値関数も不連続になることが予測される. 学習を安定させるためには, このような場合にも対応できるように不連続な行動価値関数も表現できることが望まれる.

それを実現するために, 不連続状態を特定し, そこを越えて影響が及ばないように, 仕切りの状態を付け加えることが考えられる. これにより, 逆符合の行動要素の足し合わせによる出力低下が生じないという利点がある反面, 仕切りを付けることによってその状態を特別扱いしているので, 不連続点の特定を間違っていれば学習に悪影響を及ぼす可能性がある. 完全な解決方法としてはこの方法をを選択するべきであろうが, 不連続点を正確に把握することは難しいため, 実用的ではない.

異なるアプローチとして, 不連続な状態をおおまかに推定し, その付近の状態を細かく表現することによって理想的な行動価値関数に近づける方法が考えられる. 不連続による悪影響の及ぶ範囲を狭くするだけで, その範囲内では問題が残ることになるが, 不連続点の推定が間違っていても状態の表現は他の状態と同じなので, 修正が可能である. 最終的に不連続な状態の領域を限りなく小さくすることによって, この問題に対処することができる.

3.1 章のボール注視の例のように状態・行動空間が低次元の場合は, 設計者が不連続問題が起こる状態は予測できるが, 高次元になると予測することは困難になる. したがって, ロボットが自律的にこの状態を推定することが必要である. また,

その状態の近傍で適切な行動が取れるようにしなければならない.

4. Enhanced Continuous Valued Q-learning

以下では,CVQ-learning を用いて最適行動の不連続問題に対処する手法を述べる.

4.1 最適行動の不連続問題が起こる状態の推定

最適行動の不連続問題は, 隣り合う状態における最適行動が互いに逆符合の行動となる場合に生じる. 最適行動の境界となる状態は, それらの行動価値が等しくなる状態である. CVQ-learning では行動価値関数は線形補間により連続な折れ線で表現される. このとき, 互いに逆符合の行動価値関数との交点でそれらの行動価値が等しくなる. つまり, その付近で最大の行動価値を持つ行動が不連続に変化する可能性があるため, その付近が最適行動の不連続な状態であると推定することができる.

4.2 代表状態点の追加

最適行動が不連続になる境界状態が予測できたとき, この点に新たに代表状態点を加えてさらに学習を進める. 十分に学習し直した後, 再び境界状態を予測して代表状態点を加える. これを繰り返していくことにより, 実際の境界状態を挟む代表状態の間隔が狭まっていくことが期待できる. CVQ-learningでは, 代表状態点を越えた状態の影響は全く受けない. また, 出力行動も代表状態点を越えて影響を及ぼすこともない. 適当な状態に代表点を追加するより, 境界状態の悪影響を受けない学習および行動が期待できる.

代表状態点を追加する状態を決定するためには, 分割された状態領域の頂点に位置する代表状態価値が十分に学習されていなければならない.

マルコフ性が成立する環境では, Q-learningによって得られるQ値は, ある一

定の値に収束することが証明されている. 一般に,Q 値の平均値や合計の変化がある程度小さくなった時点で, 十分な学習によって全ての Q 値が収束したと見なし, 学習を終了することが多い. この方法によって Q 値が収束したと見なし, 新たな代表状態点を追加して再学習を行うと, 次のような問題点が挙げられる.

- **収束判定が学習中の行動選択に依存する**
  選択回数が少ない行動の Q 値が十分に学習されていなくても収束したとみなしてしまう.

- **全ての Q 値が収束するまで収束判定できない**
  なかなか代表点を追加するか否か判定できない.

4.2.1 行動価値習熟率

以上のことから, 代表点を追加するタイミングをはかるために, ひとつひとつの Q 値の習熟率を示す行動価値習熟率 $\sigma(s,a)$ を導入する.

- **更新率が一定のとき**
  Q 値の更新は

$$Q(s,a) <= (1-\alpha)Q(s,a) + \alpha[r + \gamma V^t(s')]$$

  によって行われる. ここで $r+\gamma V(s')$ は, 現在までの学習結果による $Q(s,a)$ の推定値である. これが常に一定の値 Q^* をとった場合の Q 値の更新式は

$$Q_{t+1} = (1-\alpha)Q_t + \alpha Q^* \tag{2}$$

  となる. 初期値を Q_0 としてこの漸化式を解くと,

$$Q_t = (1-\alpha)^t Q_0 + [1-(1-\alpha)^t]Q^* \tag{3}$$

  となる. $0 < \alpha < 1$ なので, 更新する度に Q_0 の係数 $(1-\alpha)^t$ は 0 に漸近し, Q^* の係数 $1-(1-\alpha)^t$ は 1 に漸近するため, Q^t は Q^* に近づいて行くことがわかる.

(3) 式の Q^* の係数 $1-(1-\alpha)^t$ は, Q^* に近づくほど大きくなるため, この値を $Q(s,a)$ の習熟率 $\sigma(s,a)$ として次の式で定義する.

$$\sigma(s,a) = 1-(1-\alpha)^t \tag{4}$$

- **更新率が変動するとき** t 回目の更新率を α_t とすると, Q 値の更新式は (2) 式と同様に

$$Q_{t+1} = (1-\alpha_t)Q_t + \alpha_t Q^* \tag{5}$$

となる. 初期値を Q_0 としてこの漸化式を解くと,

$$Q_t = \prod_{k=1}^{t}(1-\alpha_k)Q_0 + [1-\prod_{k=1}^{t}(1-\alpha_k)]Q^* \tag{6}$$

となる. $0<\alpha_k<1$ なので, 更新する度に Q_0 の係数 $\prod_{k=1}^{t}(1-\alpha_k)$ は 0 に漸近し, Q^* の係数 $1-\prod_{k=1}^{t}(1-\alpha_k)$ は 1 に漸近するため, Q^t は Q^* に近づいて行くことがわかる.

(6) 式の Q^* の係数 $1-\prod_{k=1}^{t}(1-\alpha_k)$ は, Q^* に近づくほど大きくなるため, この値を $Q(s,a)$ の習熟率 $\sigma(s,a)$ として次の式で定義する.

$$\sigma(s,a) = 1-\prod_{k=1}^{t}(1-\alpha_k) \tag{7}$$

4.3 状態価値習熟率

状態価値 $V(s)$ は, 状態 s を引数とする全ての行動価値 $Q(s,a)$ を用いて定義されている. したがって, 状態価値習熟率は行動価値習熟率を用いて次のように定義する.

$$\bar{\sigma}(s) = \frac{1}{n}\sum_{k=1}^{n}\sigma(s,a_k) \tag{8}$$

ただし,n は代表行動 a_k の個数.

この値が十分高くなった時点以降にその部分に新たな代表状態点を追加するか否かの判定を行なうことができる.

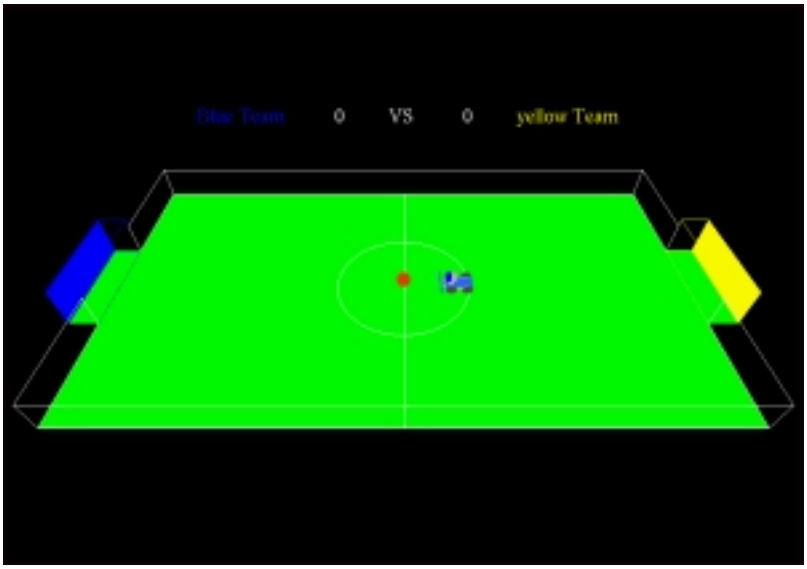

図 3 シミューレーション画面 / フィールド

5. 実験

5.1 1 次元状態・1 次元行動

3 章で述べた全方位視覚を搭載したロボットがその場回転することによってボールを正面で注視するタスクのシミュレーションを行った.

5.1.1 実験方法

3 章図 2 のように 4 点の状態代表点を配置し, 代表行動は左大回転, 左小回転, 右小回転, 右大回転の 4 種類とする. 初期姿勢はランダム,1 試行 50 ステップで, 毎ステップ Q 値の更新を行っている.

報酬は正面を 0[rad] として, 次の式で与えた.

$$r(\boldsymbol{x}) = \begin{cases} 1 - \frac{\pi}{4}|\boldsymbol{x}| & (\text{if } |\theta| < \pi/4) \\ 0 & (\text{otherwise}) \end{cases}$$

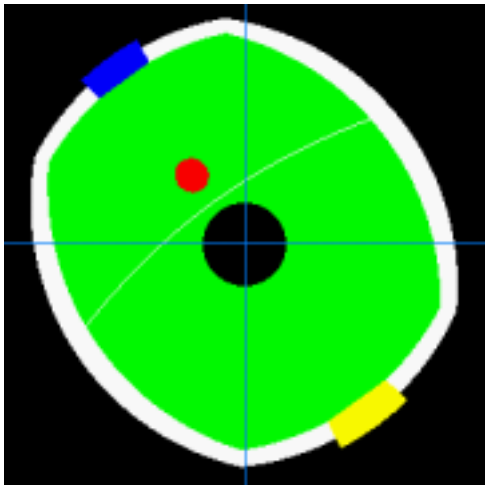

図 4　シミュレーション画面 / 全方位画像

5.1.2　結果

CV Q-learning によって学習した 200,500,800 試行後の Q 値をそれぞれ図 5,6,7 に, Q 値の平均値の変化を図 9 に示す. なお,200,500 試行後に不連続問題が起こる可能性のある点, すなわち左大回転と右大回転の Q 値の交点（図中の星印）となる状態に, 新たな代表点を加えている. また,200,500,800 試行後に常に最適行動を選択するようにしたときの 100 試行の平均位置を表 1 に示す.

表 1　学習結果の 100 試行の平均位置

試行回数	状態点数	平均位置 *[rad]*
200	4	0.1881
500	6	0.0256
800	8	0.0212
1000	10	0.0138

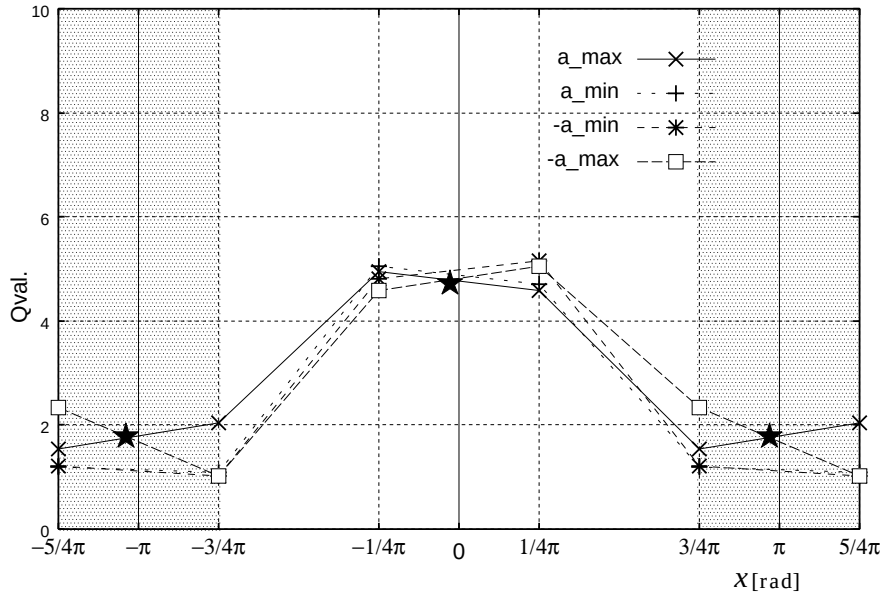

図 5　200 試行後の Q 値

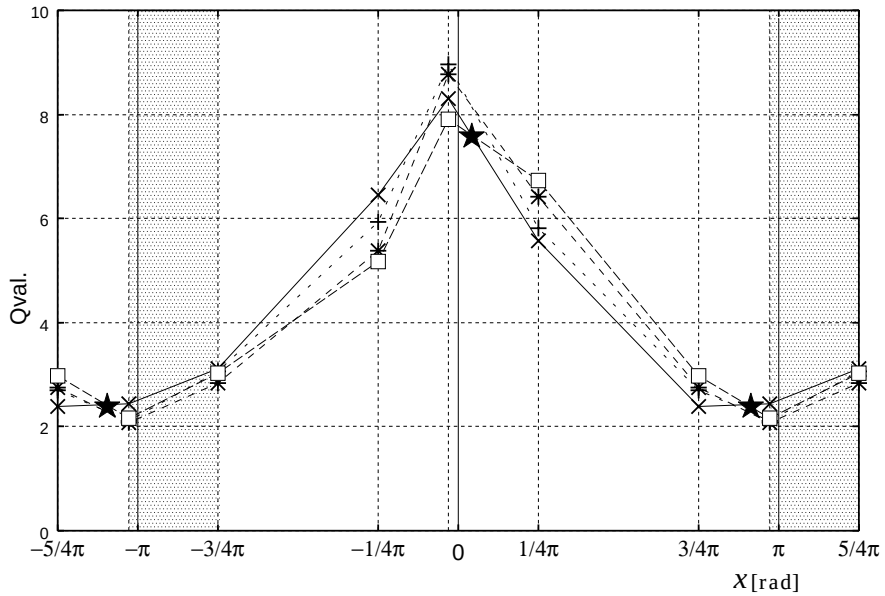

図 6　500 試行後の Q 値

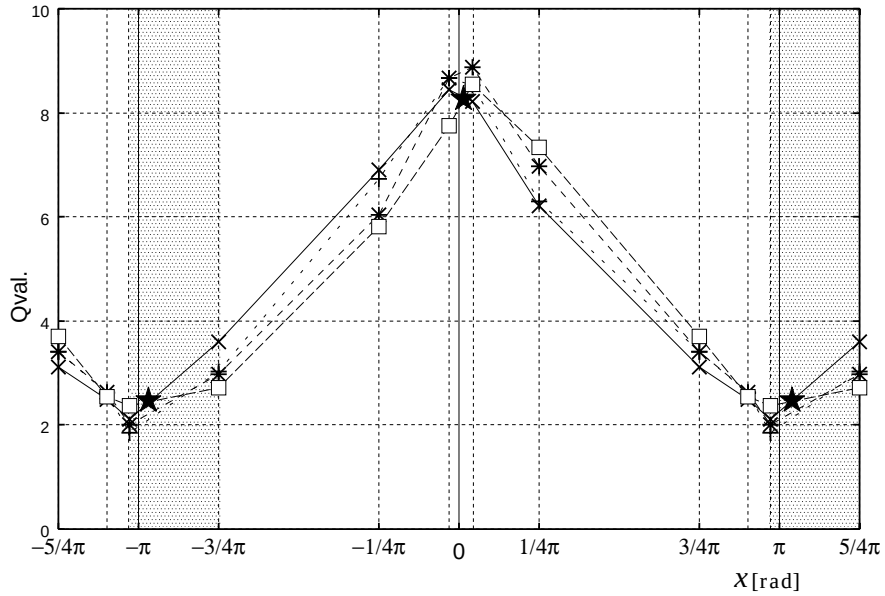

図 7　800 試行後の Q 値

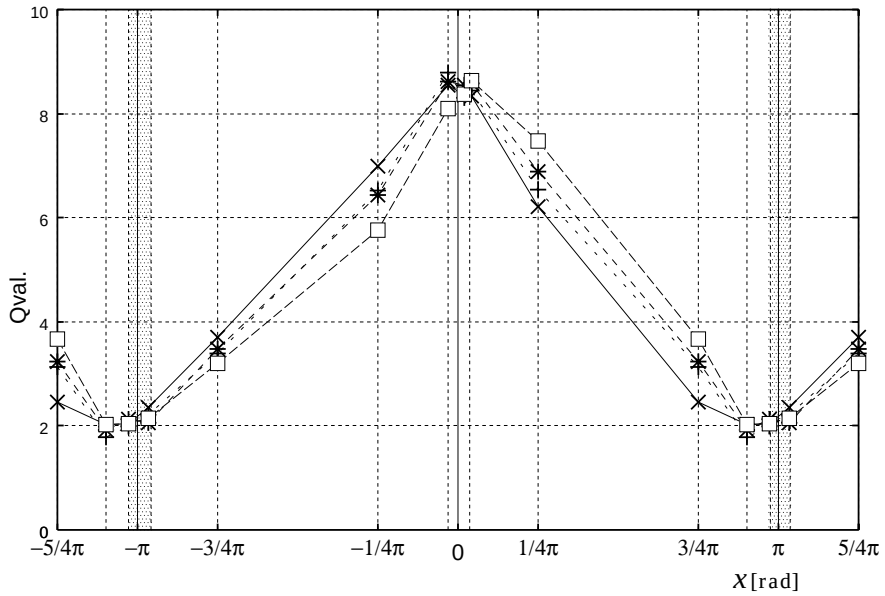

図 8　1000 試行後の Q 値

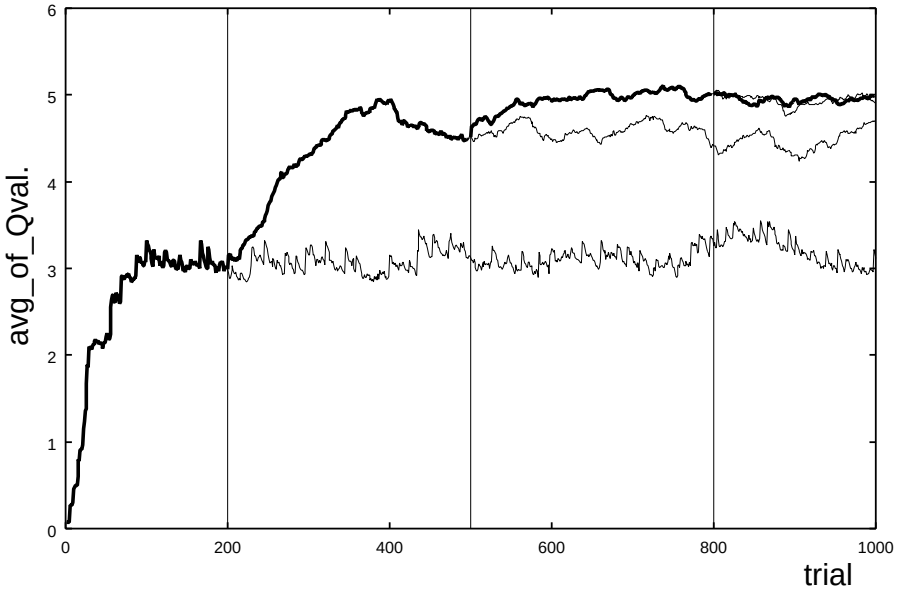

図 9　Q 値の平均の変化

5.1.3 考察

表 1 より, 代表点の追加により性能が改善されていることがわかる. 図 6,7 より 500 試行後に追加した代表状態点は π[rad] を挟むのを失敗しているが, 800 試行後には成功していることがわかる. このように, 最適行動の不連続問題が起こる領域も次第に狭まっており, 好ましい代表状態点の配置になっている. また, 図 9 より代表点が追加されると Q 値の平均値の振動が小さくなっていることがわかる. これにより, 学習が安定化していることが言える.

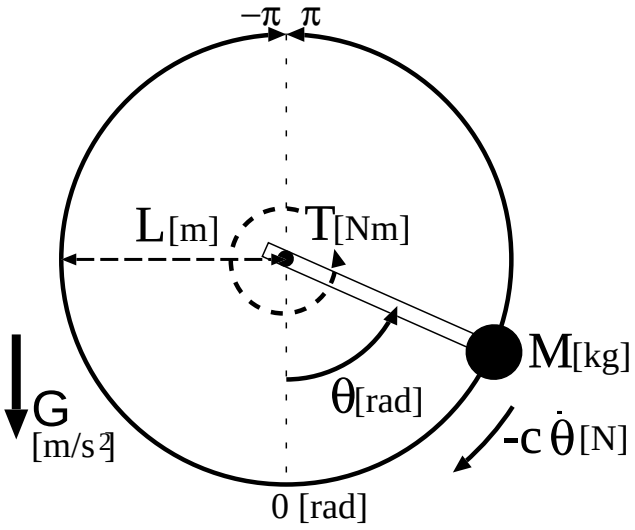

図 10 単振子

5.2 2 次元状態・1 次元行動 - 単振子の振り上げ問題 -

Fig.10 のような単振子の振り上げ問題のシュミレーションを行う.
この単振子の運動方程式は次のようになる.

$$ML^2\ddot{\theta} = -c\dot{\theta} + MGL\sin\theta + T$$

従って, 次の数列により角度, 角速度, 角加速度を求める.

$$\begin{aligned}
\theta_n &= \Delta t\dot{\theta}_{n-1} + \frac{\Delta t^2}{2}\ddot{\theta}_{n-1} \\
\dot{\theta}_n &= \Delta t\ddot{\theta}_{n-1} \\
\ddot{\theta}_n &= \frac{-c\dot{\theta}_{n-1} + MGL\sin\theta_{n-1} + T}{ML^2}
\end{aligned}$$

一番下の位置 ($\theta = 0$) で静止した状態を初期状態とし, (a) 一定の方向に最大のトルクをかけた場合は最も高い位置 ($\theta = \pm\pi$) に到達することができないが, (b) 速度と同じ向きに最大のトルクをかけた場合は到達できるように各パラメータを設定した. Fig.11,12 参照

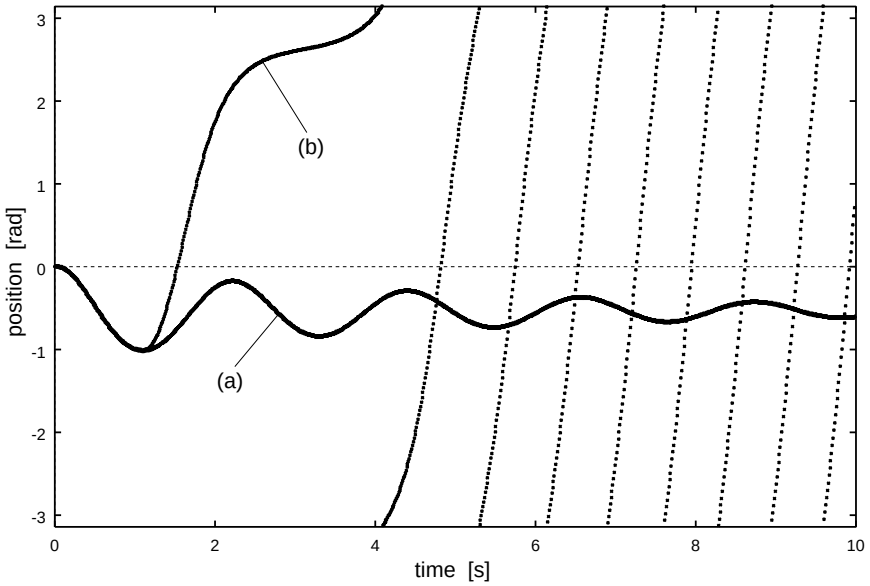

図 11 (a) 一定の方向にトルクをかけた場合. (b) 速度と同じ向きにトルクをかけた場合の位置の時間変化

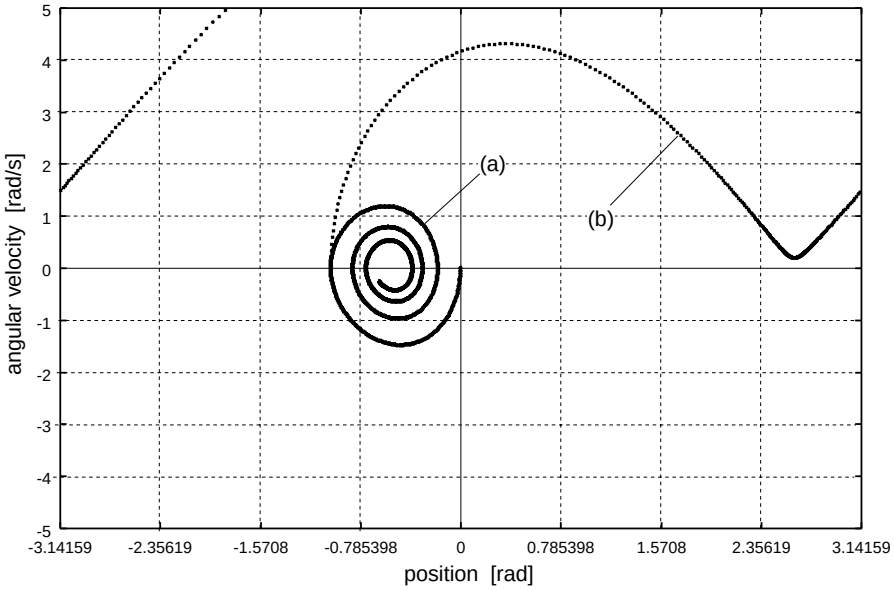

図 12 (a) 一定の方向にトルクをかけた場合. (b) 速度と同じ向きにトルクをかけた場合の位置と角速度

表 2 シミュレーションに用いたパラメータ

サンプリングタイム	:	$\Delta t = 0.01\ [s]$
おもりの質量	:	$M = 1.0\ [kg]$
腕の長さ	:	$L = 1.0\ [m]$
角速度摩擦係数	:	$c = 0.5\ [Ns/rad]$
重力加速度	:	$G = 9.8\ [kgm/s^2]$
トルク出力	:	$-5.0 \leq T \leq 5.0\ [Nm]$

5.2.1 実験方法

本研究で提案している手法とセミマルコフモデルによる Q-learning を用いて学習した. 状態空間は, 角度, 角速度の二次元とし, 行動空間はトルクの一次元とし, 状態の読みとり, 行動の指令は 0.1s 毎に行える事とした. 報酬関数はおもりの高さを正規化した次の式で与えた.

$$r(\theta) = \frac{1 - \cos\theta}{2}$$

CVQ-learningでは代表状態点を頂点とする三角形のセルを形成し, そのセル内の状態は, 頂点の 3 つの代表状態の線形結合で表現する. 2 章と同様に, 代表状態の重みは入力ベクトルと頂点を結ぶ線分で分割された三角形の面積の割合とする. Q 値の更新は状態が存在するセルが変化したときに行う. 更新率を調整する状態と行動の重みは, そのセル内での平均値とした. 各セル内の代表状態点の状態習熟率が 0.9 を越えたときにセルを分割するか否かの判定を行う. セルを分割した場合, そのセル内の代表状態点の状態価値は変化すると考えられるので, 状態習熟率に 0.9 をかけて減少させた.

5.2.2 結果

以下に CVQ-learning によって状態代表点が追加されていく様子と, 各トライアル終了後の一番下の位置 $\theta = 0$ からの挙動を図 13〜26 に示す.

また,CVQ-learningとセミマルコフモデルのQ-learningによる50,000トライアル終了時の学習結果を図27〜 42に示す.

5.2.3 考察

図27〜30は学習によって獲得された状態価値関数である. 図27〜29のQ-learningでは状態を細かくしていく事で, 状態価値関数の精度が向上していることが分かる. 図30のCVQ-learningは, 状態数が138と少ない割には状態価値関数の特徴をうまく表現している.

図31〜??は状態空間と獲得された行動の挙動を示す. Q-learningで, 状態数64のときは離散化された状態が大きすぎて挙動も滑らかではないため, 目標位置で静止することが出来ていないが, 状態を細かくしていくにつれ, 描かれる軌跡が滑らかになり, スムーズに目標状態に到達している. 図30のCVQ-learningは, 先に示したように学習中に状態点を追加することによって獲得された. 行動の出力は連続値なので軌跡は滑らかであり, 状態数は比較的少ないが目標状態に到達することができている.

図35〜38は獲得された行動による位置の時間変化を示す. 図35の状態数64の場合は勢いを付けて最高点まで到達できているが, 最高点付近に留まることが出来ていない. 状態数が増えると, 勢いを付けるために振れる回数も減り, より早く最高点付近に近付いており, そのまま留まることが出来ている. 図38のCVQ-learningでは状態数256,1024のQ-learningよりは勢いを付けるために振れる回数が1回多く, 時間がかかるが最高点付近に留まることが出来た.

図39〜42は学習中のに獲得された行動で, 最下点で静止した状態から始めたときの高さの100トライアルごとの平均を示す. 状態数が少ない程, 振動が大きいことがわかる. これは状態が大きいことにより状態遷移先が広範囲に分布するのでQ値が不安定になっていると思われる. いずれの場合もほぼ一定の値に収束しているが, 状態数が多い程, 収束が遅くなっている. 図41の状態数1024の場合はQ値の変動が小さく, 収束後の性能が最も良いが, 目標状態に到達するまでに通過しなければならない状態数が多く, その分, 学習しなければならない状態数が多いこととなり, 学習時間が増大する. 本研究では, 比較的低次元の状態空間を扱ってい

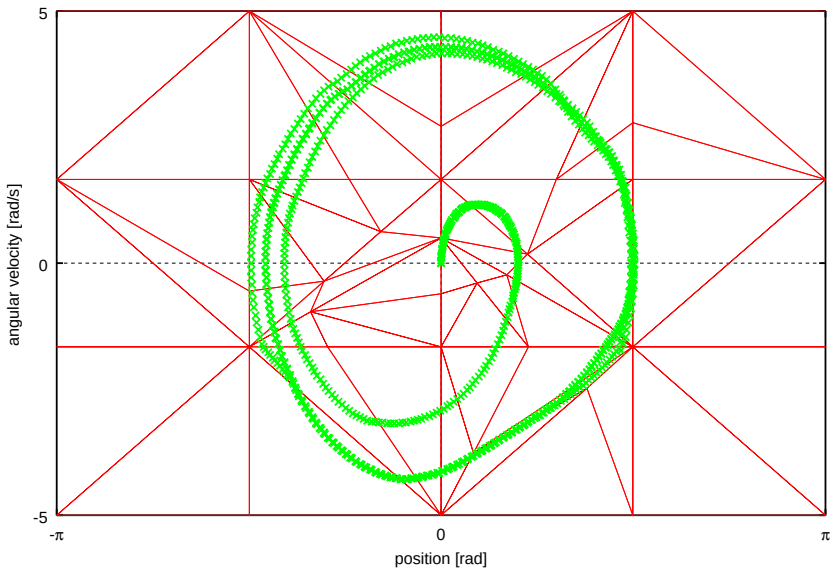

図 13　500 トライアル 状態数 37 セル数 58 CVQ-learning による状態空間と状態遷移

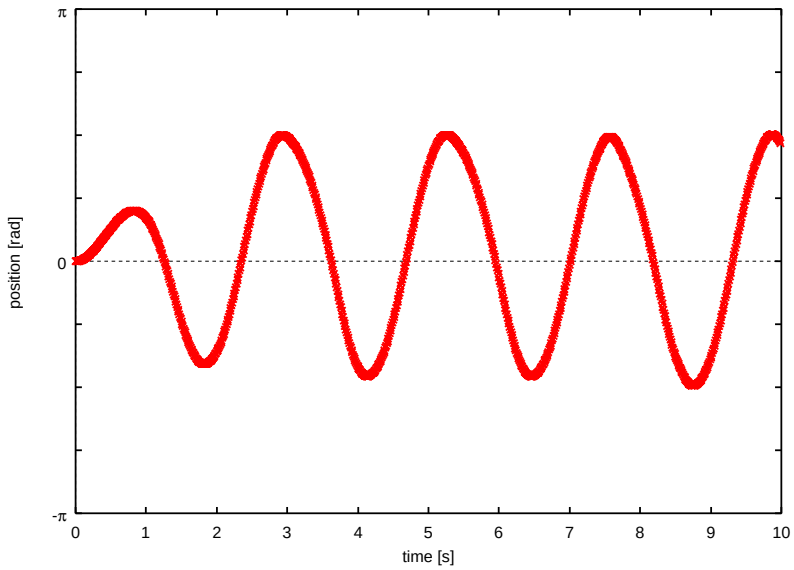

図 14　500 トライアル 状態数 37 セル数 58 CVQ-learning による位置の時間変化

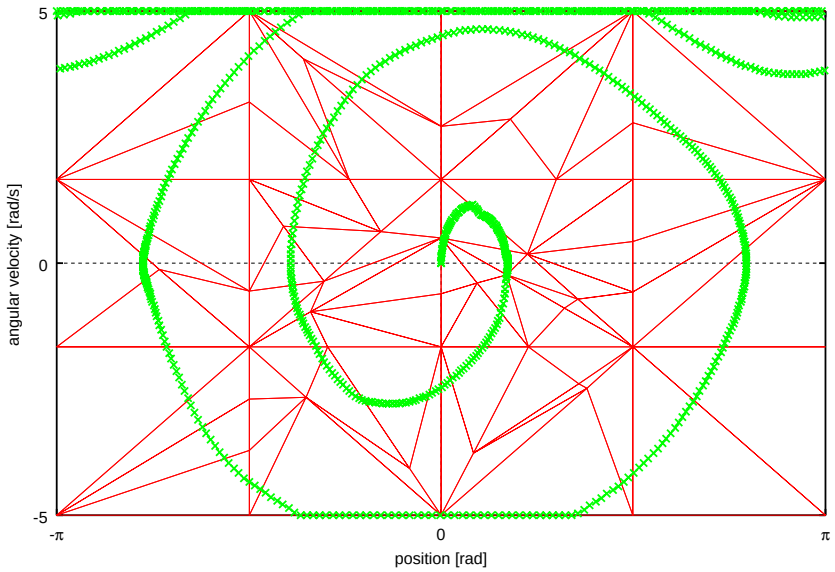

図 15　1000 トライアル 状態数 51 セル数 86 CVQ-learning による状態空間と状態遷

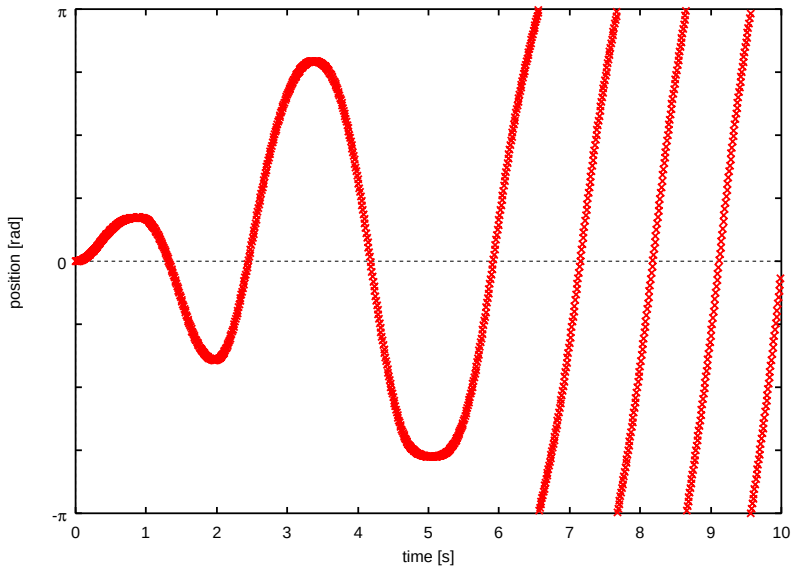

図 16　1000 トライアル 状態数 51 セル数 86 CVQ-learning による位置の時間変

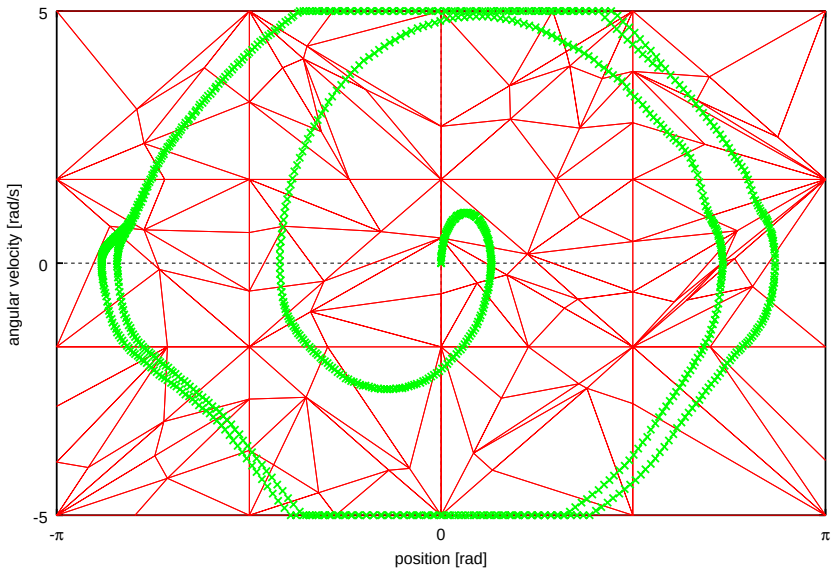

図 17　5000 トライアル 状態数 99 セル数 175 CVQ-learning による状態空間と状態遷移

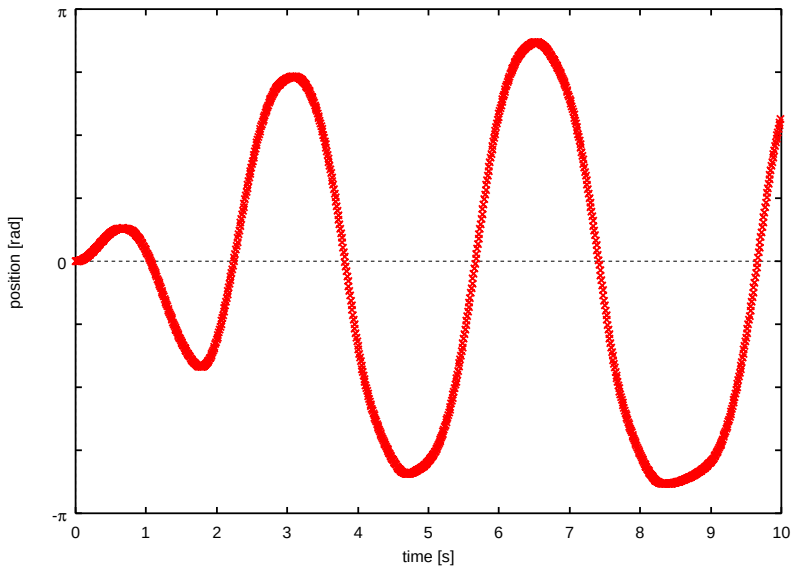

図 18　5000 トライアル 状態数 99 セル数 175 CVQ-learning による位置の時間変化

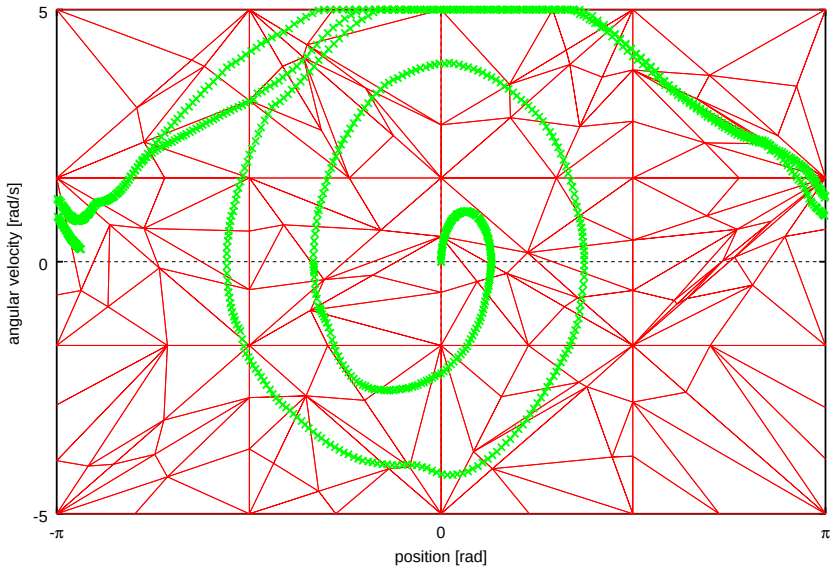

図 19　10000 トライアル 状態数 120 セル数 212 CVQ-learning による状態空間と状態遷移

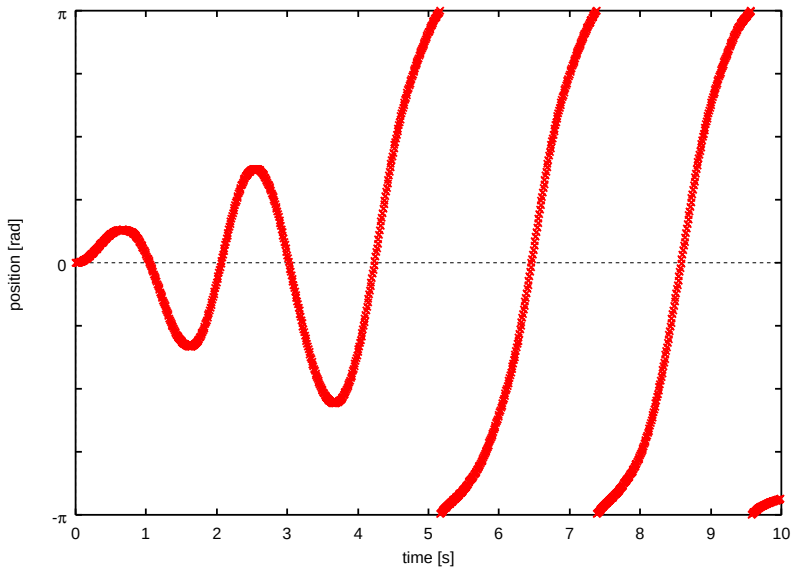

図 20　10000 トライアル 状態数 120 セル数 212 CVQ-learning による位置の時間変化

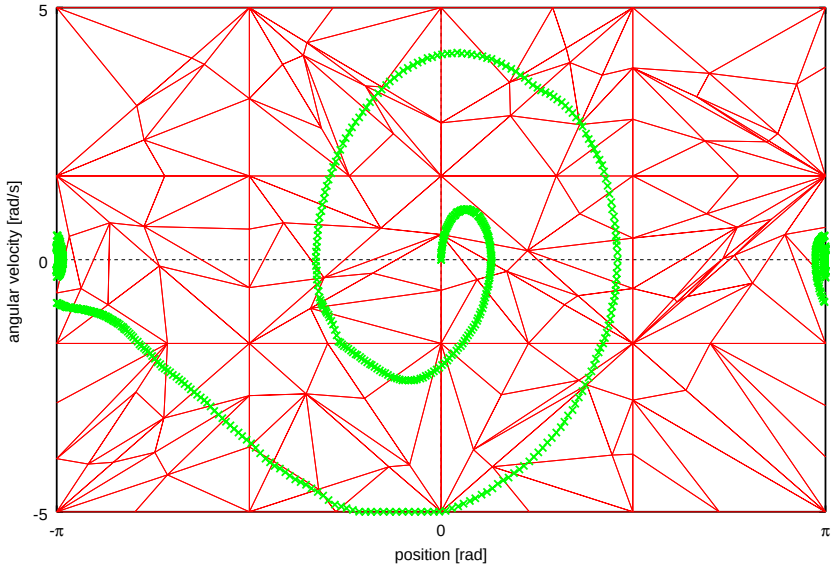

図 21　20000 トライアル 状態数 126 セル数 222 CVQ-learning による状態空間と状態遷移

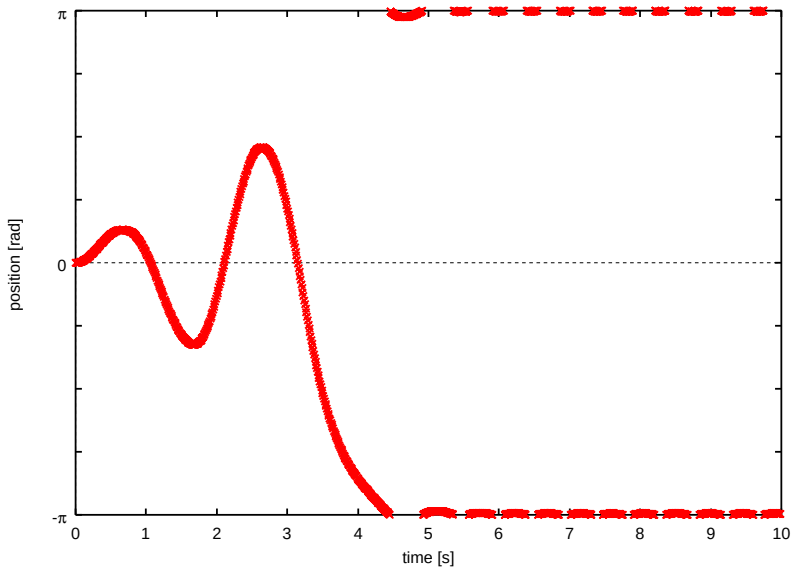

図 22　20000 トライアル 状態数 126 セル数 222 CVQ-learning による位置の時間変化

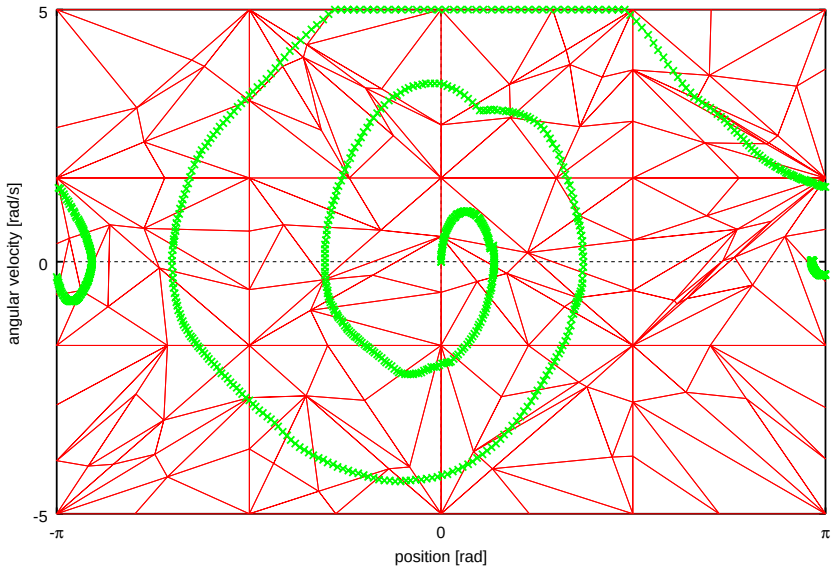

図 23　30000 トライアル 状態数 130 セル数 227 CVQ-learning による状態空間と状態遷移

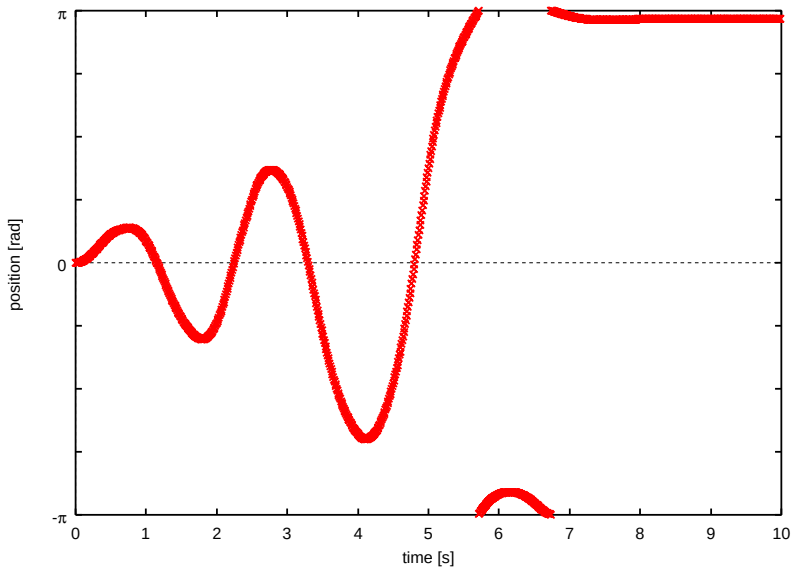

図 24　30000 トライアル 状態数 130 セル数 227 CVQ-learning による位置の時間変化

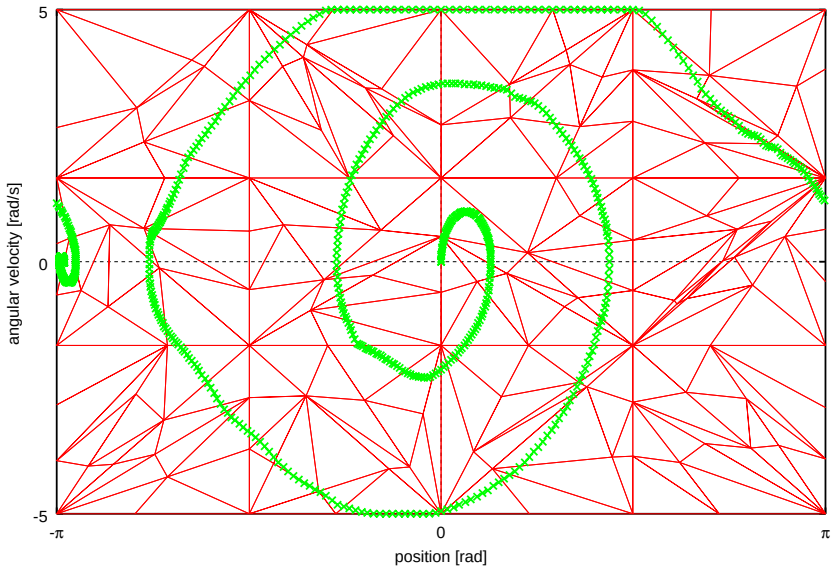

図 25　40000 トライアル 状態数 135 セル数 237 CVQ-learning による状態空間と状態遷移

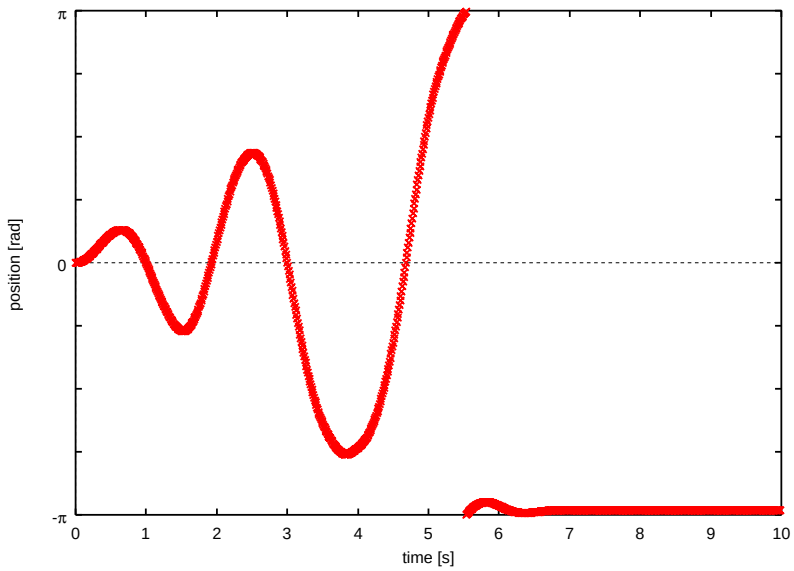

図 26　40000 トライアル 状態数 135 セル数 237 CVQ-learning による位置の時間変化

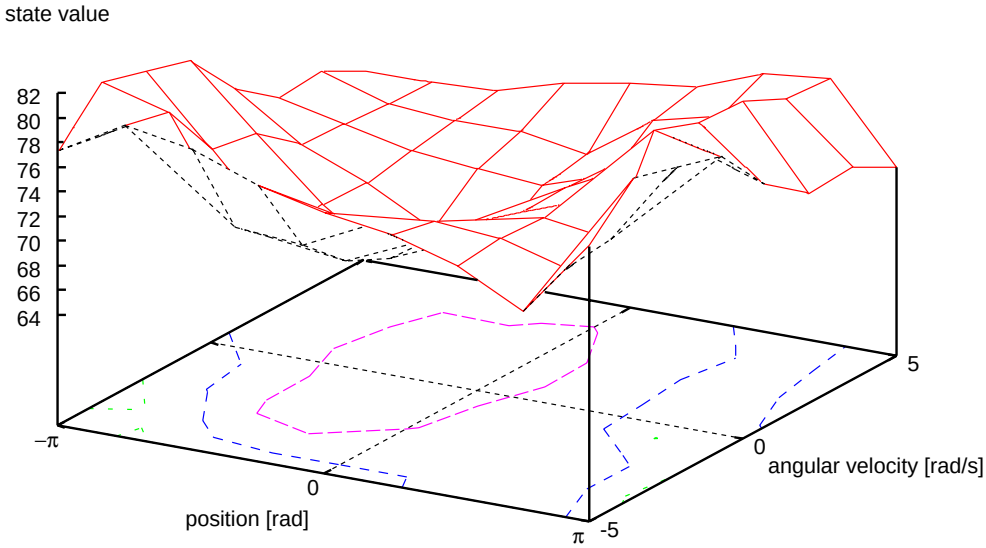

図 27　状態数 64 Q-learning による状態価値関数

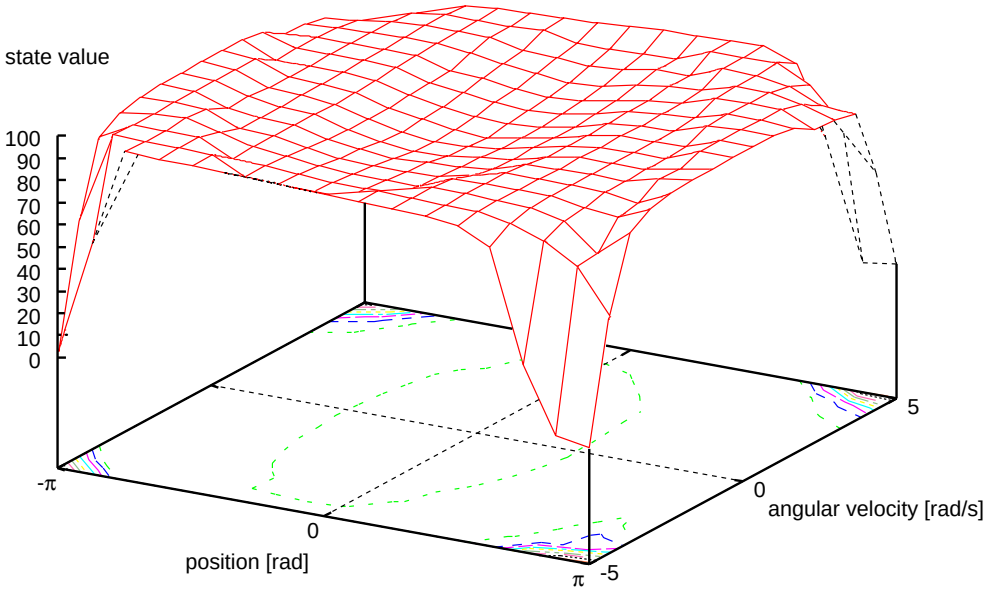

図 28　状態数 256 Q-learning による状態価値関数

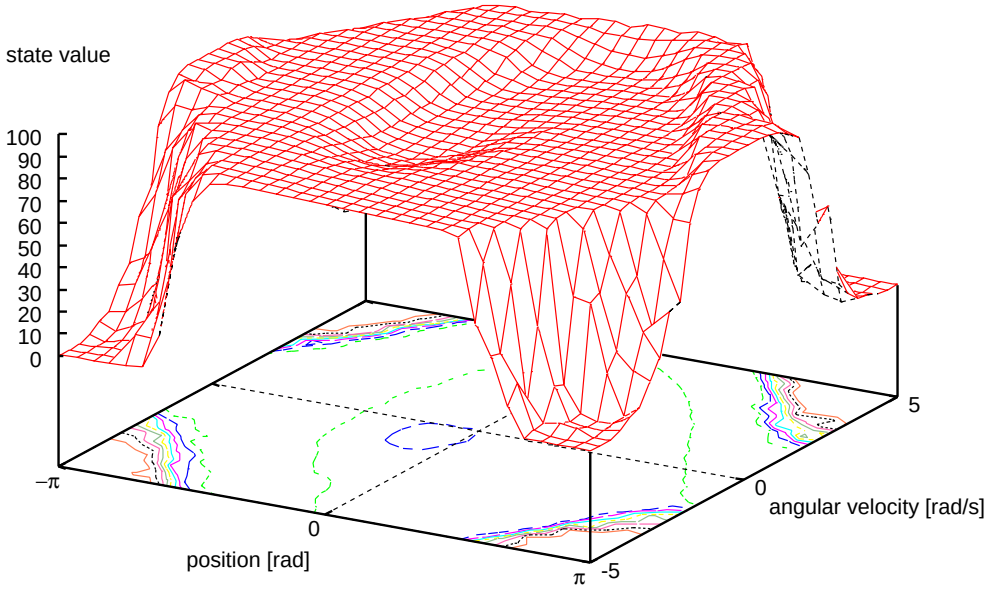

図 29　状態数 1024 Q-learning による状態価値関数

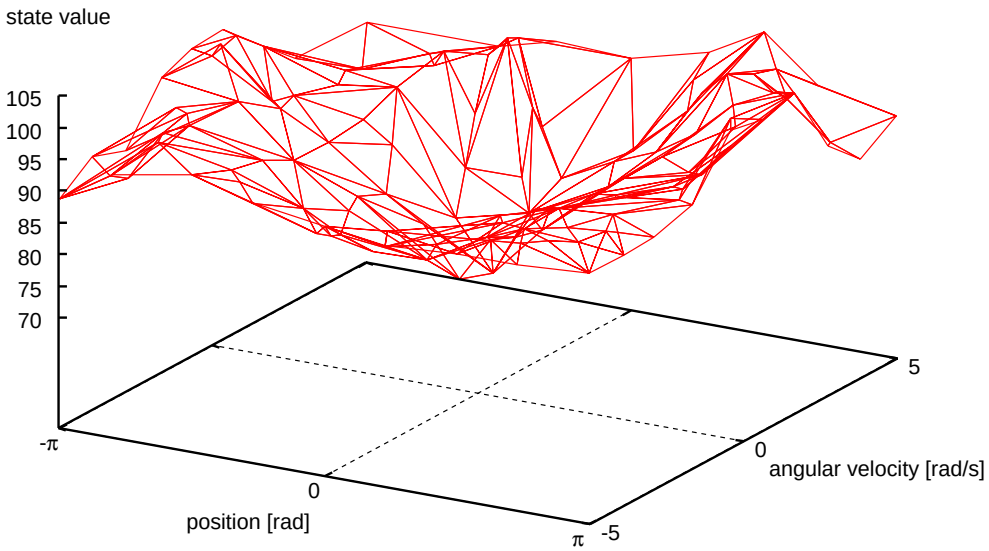

図 30　状態数 138 セル数 242 CVQ-learning による状態価値関数

図 31　状態数 64 Q-learning による位置と角速度

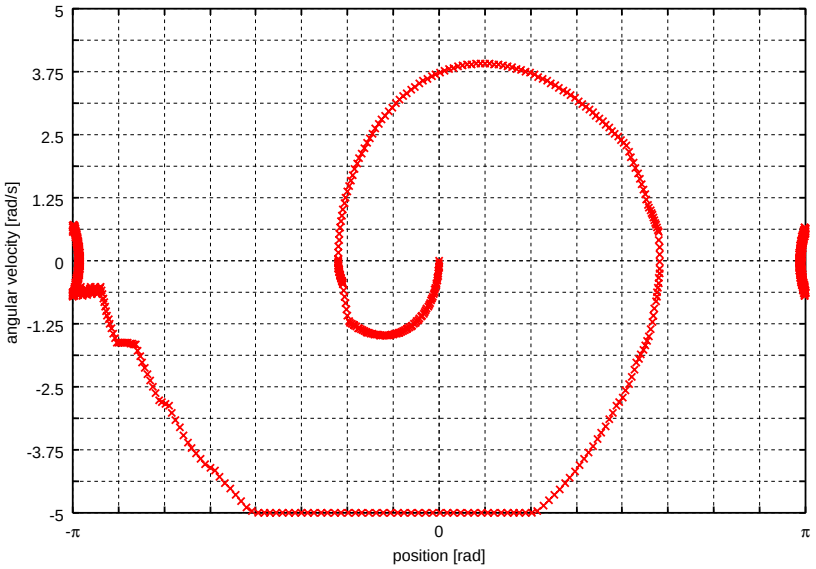

図 32　状態数 256 Q-learning による位置と角速度

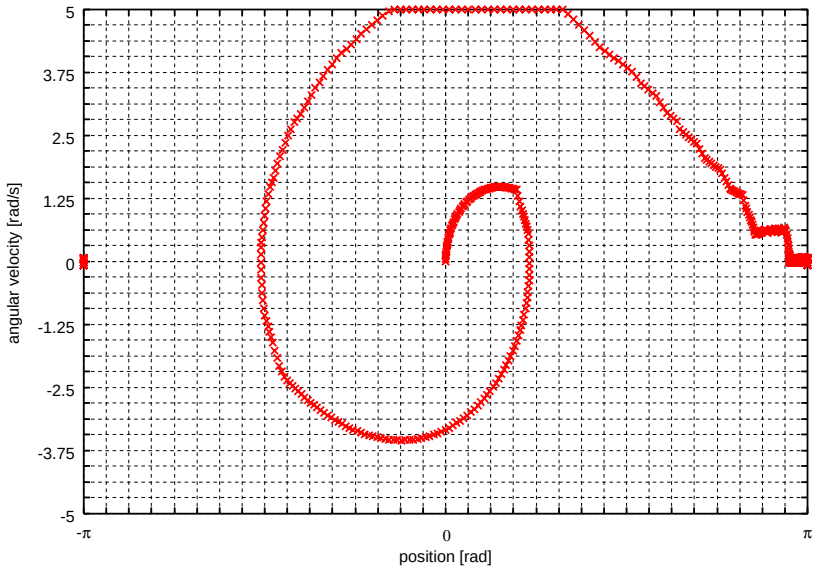

図 33　状態数 1024 Q-learning による位置と角速度

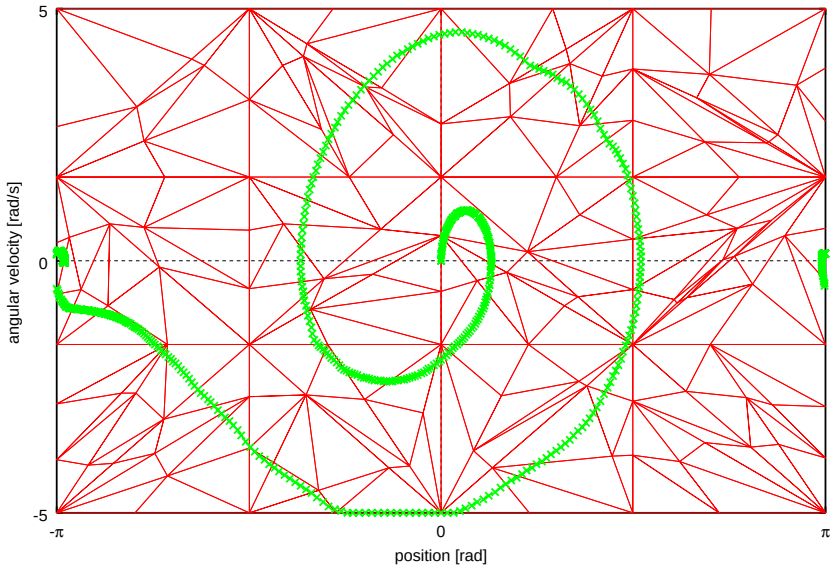

図 34　状態数 138 セル数 242 CVQ-learning による位置と角速度

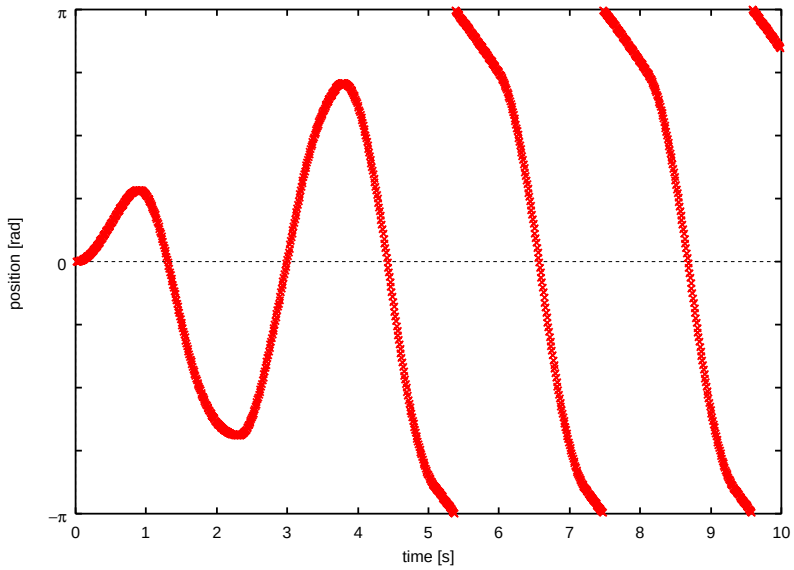

図 35　状態数 64 Q-learning による位置の時間変化

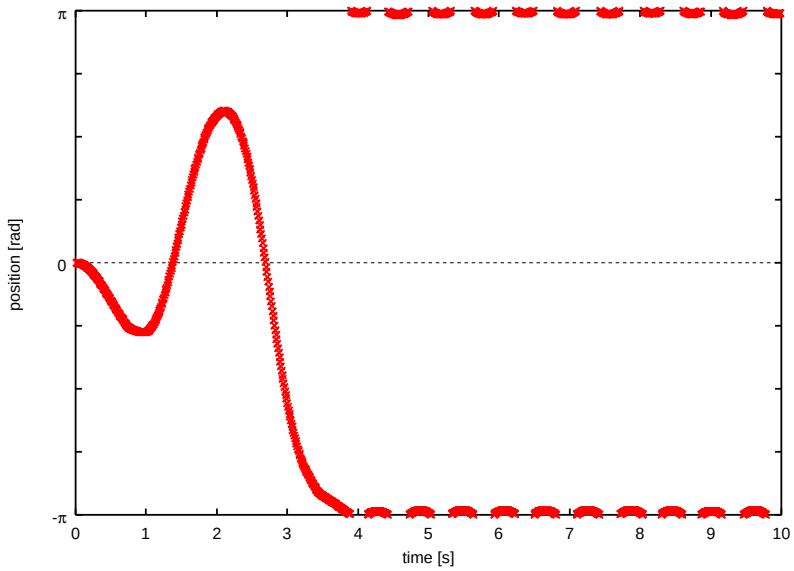

図 36　状態数 256 Q-learning による位置の時間変化

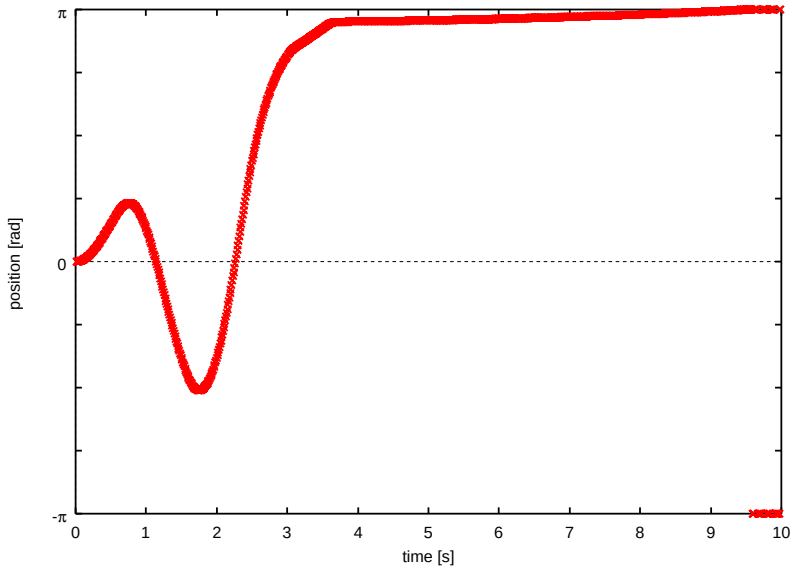

図 37　状態数 1024 Q-learning による位置の時間変化

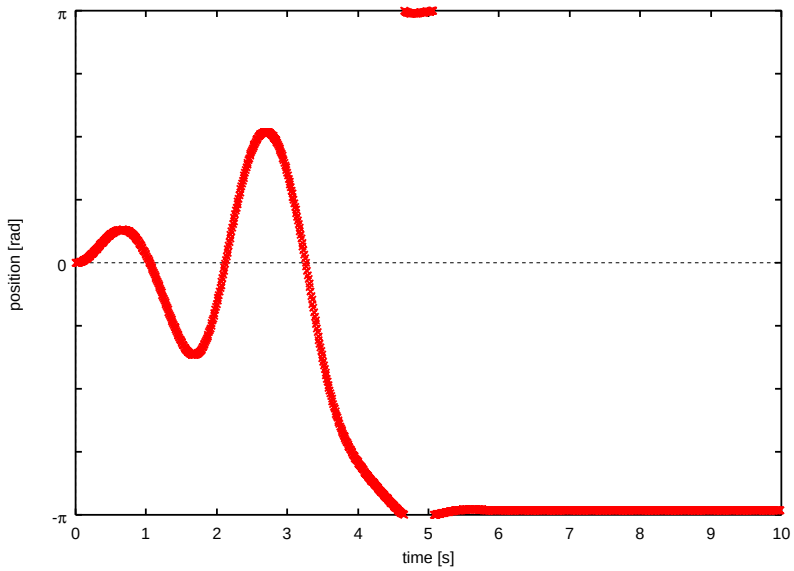

図 38　状態数 138 セル数 242 CVQ-learning による位置の時間変化

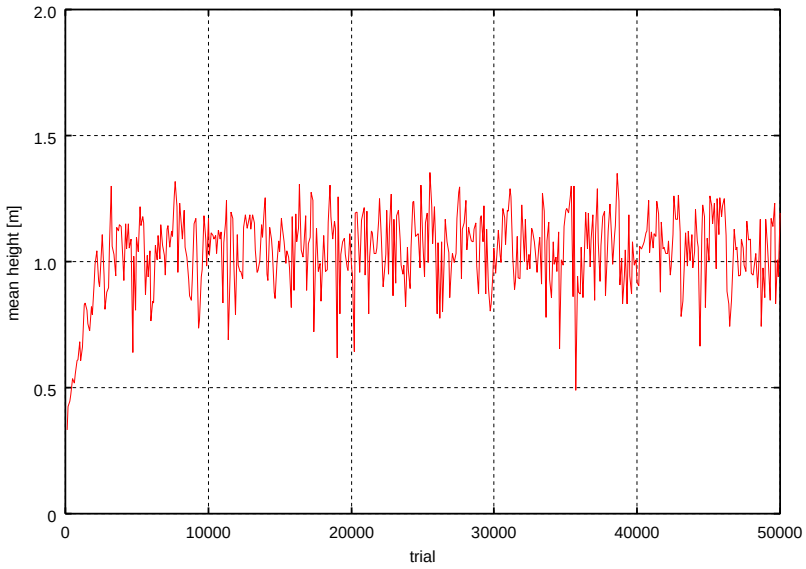

図 39　状態数 64 Q-learning による高さ平均

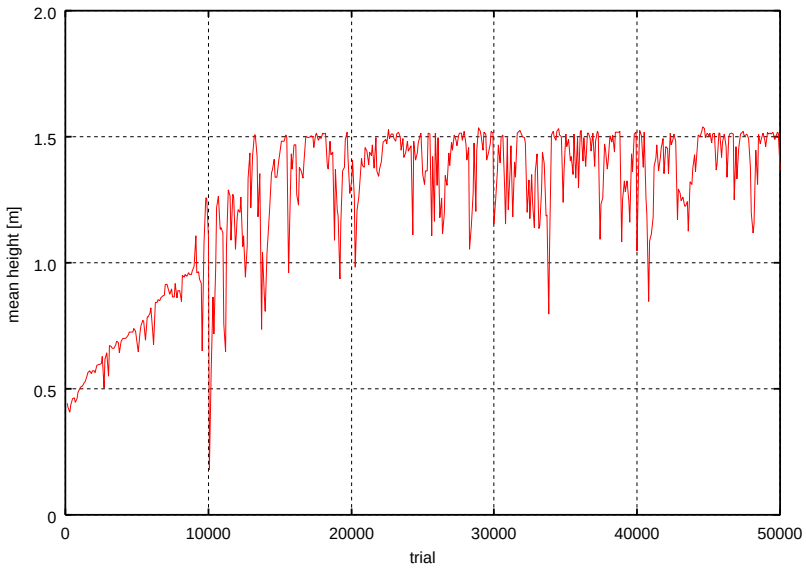

図 40　状態数 256 Q-learning による高さ平均

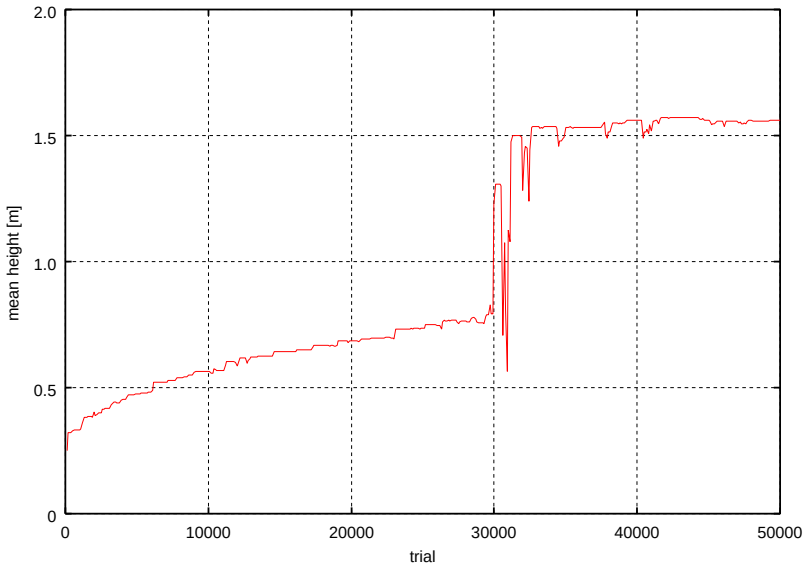

図 41　状態数 1024 Q-learning による高さ平均

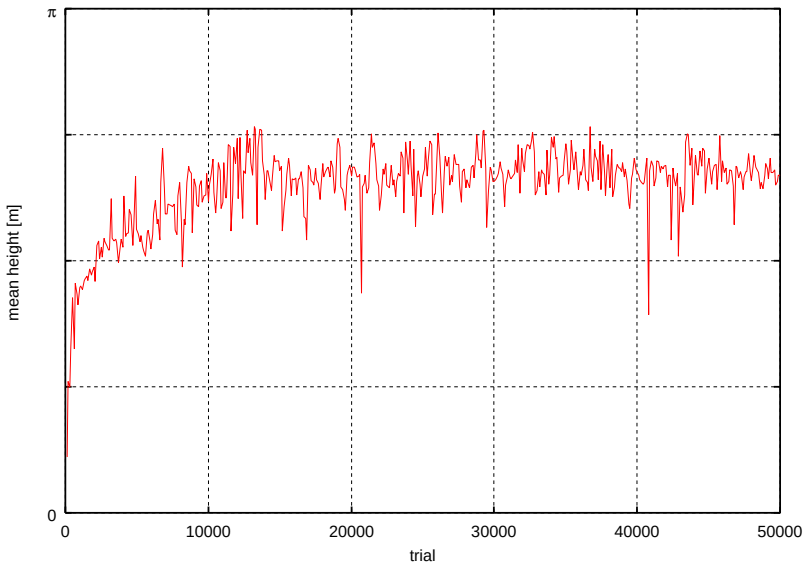

図 42　状態数 138 セル数 242 CVQ-learning による高さ平均

るが, 高次元の状態空間になると状態数が爆発的に増加し, 実用的でない程学習時間の増大が顕著になる. 以上の観点より, 状態数は多すぎても少なすぎても問題であるが, 図 42 の CVQ-learning の場合は, 学習収束後のパフォーマンスでは状態数が多いものと比べると多少劣るものの, 状態数が少ないにもかかわらず, 十分な行動が獲得されている. また, 学習の初期段階においては状態数が少ないため立上りが早くなっており, 収束も早い.

6. おわりに

本研究では連続値強化学習で問題となる最適行動の不連続問題を指摘し, 学習しながら状態点を追加していく事により, この問題に対処する手法を提案した. また, シミュレーションにより本手法の有効性を示した.

近年, 多くの強化学習法が提案されているが, 比較的低次元のタスクでしか検証されていないため, 獲得された行動の最適性を追求しているものが多く, 多次元になり状態数が増加した場合にも実用可能な程度の学習時間であるかという議論はあまりなされていない.

提案手法では, 代表状態点おける最大の Q 値を持つ行動を補間することにより連続な行動を得ているため, 最適性を犠牲にしているが, 状態空間を改良していく事によってその欠点を補っている. また, 考察でも述べた通り, 学習時間を短縮することができ, さらに状態数も少ないため, 多次元状態のタスクへの拡張も十分期待できる.

謝辞

本研究を終えるに臨み，終始暖かい御指導，御鞭撻を賜わり，さらに本論文遂行にあたって並々ならぬ御援助を賜わりました小笠原司教授に深く感謝の意を表します.

本論文の作成にあたり, 多くの助言を頂いた伊藤実教授に深く感謝の意を表します.

本研究の遂行にあたり，日頃から多くの御指導を賜わりました今井正和助教授に深く感謝の意を表します.

また，本研究の遂行にあたり，多くの助言助力を頂いた中村恭之助手, 松本吉央助手に深く感謝の意を表します.

最後に，本研究の遂行にあたり多くの御協力，助言を頂いた奈良先端科学技術大学院大学ロボティックス講座一同の方々に深く感謝致します.

参考文献

[1] C.J.C.H.Watkins, P.Dayan: "Technical note: Q-learning". *Machine Learning*,8:279-292,1992.

[2] J.S. Albus: "A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC)". *Journal of Dynamic Systems, Measurement, and Control, Trans. ASME*,97(3):220-227, 1975.

[3] F. Saito, T. Fukuda: "Learning Architecture for Real Robotic Systems-Extension of connectionist Q-Learning for Continuous Robot Conrtol Domain". *Proceedings of IEEE International Conference on Robotics and Automation*, pp.27-32, 1994.

[4] R.S Sutton: "Generalization in Reinforcement Learning: Successful Examples Using Sparse Coarse Coding". *Advances in Neural Information Processing Systems 8*, pp.1038-1044, 1996.

[5] Lin, Long-Ji: "Self-Improving Reactive Agents: Case Studied of Reinforcement Learning Frameworks", *from Animals to Animats 3*, pp.225-245, 1992.

[6] J. Boyan and A. Moore: "Generalization in Reinforcement Learning: Safely Approximating the Value Function". *Proceedings of Neural Information Processing Systems 7*,1995.

[7] 杉村, 増本, 長田: "強化学習における状態空間の逐次的精密化法", 人工知能学会全国大会（第 12 回）論文集, pp.67-68,1998.

[8] Atkeson, C.G., Moore A.W., Schaal S.: "Locally Weighted Learning", *Artificial Intelligence Review,* 11, pp.11-73, 1997.

[9] 石井, 佐藤: "オンライン EM アルゴリズムを用いた強化学習法", 信学技法, NC98-83, pp.41-48,1999.

[10] 堀内, 藤野, 片井, 椹木: “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.

[11] Y.Takahashi, M.Takeda, M.Asada: “Continuous Valued Q-learning for Vision-Guided Behavior Acquisition”, *Proc. of 1999 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp.255-260, 1999.

[12] 武田, 高橋, 浅田: “状態と行動を連続値で評価するQ学習 Continuous Valued Q-learning for Vision-Guided Behavior Acquisition”, 第 17 回日本ロボット学会学術講演会予稿集 3, pp.975-976, 1999.