# 目次

第１章　序論……1
1.1　背景……1
1.2　本研究の目的……2
1.3　本論文の構成……2

第２章　強化学習……3
2.1　強化学習の概要……3
2.2　行動評価手法……4
2.3　行動選択手法……5
2.4　まとめ……5

第３章　強化学習における状態の設定……6
3.1　強化学習の問題点……6
3.1.1　実機のロボットを扱う時の問題点……6
3.2　問題点の解決策……7

第４章　従来システムの検証……8
4.1　実験内容……8
4.2　実験設定……8
4.2.1　データ作成……8
4.2.2　強化学習……10
4.2.3　報酬の設定と各システムの状態の設定……11
4.3　実験結果……13
4.4　考察……15
4.5　まとめ……16

第５章　提案手法……17
5.1　概要……17
5.2　提案手法の流れ……17
5.3　提案手法の詳細……20
5.3.1　状態について……20
5.3.2　使用するパラメータの更新について……21
5.3.3　分割について……22
5.3.4　融合について……26

第６章　提案システムの検証……30
6.1　従来システムとの比較……...30
6.1.1　実験内容……30
6.1.2　実験設定……30
6.1.3　実験結果……33
6.1.4　考察……35
6.2　提案システムの検証：ダムの放水問題……...37
6.2.1　ダムの放水問題……37
6.2.2　実験内容……38
6.2.3　実験設定……38
6.2.4　実験結果……41
6.2.5　考察……43

第７章　結論……45
7.1　まとめ……...45
7.2　今後の課題……...46

謝辞……47
参考文献……48

# 第 1 章　序論

## 1.1 背景

近年，ロボットは様々な形で社会に普及してきた．例えば受付ロボットや掃除ロボット，エンタテイメントロボットなど様々な場所でロボットを目にする機会が増えてきた[1][2]．ロボットの発達に伴い，一般の家であったり，病院であったり工場であったりロボットは活躍する場が多種多様になっている [3][4][5]．しかしロボットを様々な場所で利用するために，それぞれの環境に合わせてロボットの行動を設計するのは非常に困難である．例えばどんな道でも歩ける歩行ロボットの動きを設計する場合，道や障害物はその場所によって全て異なっているため，実現するためにはすべての道や障害物の情報を設計しなければいけない．しかし実際にすべての道や障害物の情報をあらかじめ教えるのは不可能に近い．このため環境に合わせたロボットの動きを事前に生成するのは大変困難である[4][5]．よって環境に合わせて行動するロボットが必要とされている．環境に合わせて行動するロボットを実現するための方法の一つとして機械学習がある．現在，利用されている機械学習の一例ではニューラルネットワーク，遺伝的アルゴリズム，強化学習などがある．本研究では強化学習を研究対象とする．

強化学習とは，数値化された報酬をより多く得るためにどの行動を選択すればよいかを学習する学習手法である．強化学習では学習者はどの行動を取るべきかを教えられない．学習者は試行錯誤を繰り返しより多くの報酬に結びつく行動を見つけ出すように学習を行う．学習者は何も教えられず，試行錯誤を行い学習を行うという特徴のため強化学習では，未知の問題でも試行錯誤を繰り返して自身の経験から学ぶことが出来る学習だと言える[6][7].

強化学習は未知の問題でも自身の経験から学ぶことが出来る学習のため実機のロボットに使用される頻度が高い学習である．強化学習を使用する際，強化学習で使用する項目は，学習時間やメモリの問題等の面から，学習が出来るような必要最低限の数に絞って設定を行う．しかし実機のロボットはセンサで読み取った値を扱う．センサで読み取った値というのは強化学習を使用するために分類等がなされていない．このため実機に強化学習を適用する際に，センサからのデータを直接使うのは難しいといった問題がある．センサからのデータを強化学習で扱うという問題で，最も困難な問題の一つとして状態空間の構成が難しいといった問題がある．状態空間とは強化学習を使用する場合の重要な要素の一つである．従来の手法ではほとんどの場合，設計者が事前に状態空間を設計して強化学習を使用する．しかしこの設計がロボットにとって最適な設計になっているとは限らない．設計が適切ではないと強化学習でうまく学習が出来ない．よってこの問題を解決するために学

習者が学習を行いながら状態空間を構成する方法を考え，強化学習を実機のロボットに適用する研究が行われている[8][9].

## 1.2 本研究の目的

本研究では，実機のロボットに強化学習を適用する場合，最も困難な問題の一つである状態空間の構成について着目する．従来の手法では設計者が事前に状態空間を設計するため，ロボットにとって最適な設計になっているとは限らないという問題が生じる．この問題点を解決するために，設計者ではなく学習者が状態空間を設定するシステムを提案することを目的とする.

## 1.3 本論文の構成

第２章では，強化学習についての説明を行う.
第３章では，具体的に強化学習の問題点を考察し，本研究で問題点を解決するための概略を説明する.
第４章では，従来の手法では具体的にどのような問題が発生するかを確認するため，従来の手法を用いて実験を行う.
第５章では，強化学習の問題点を解決するため，本研究での具体的な手法の提案を行う．手法では状態を自動で設定するための方法について説明を行う．本研究では状態の自動設定に分割と融合を用いて状態の設定を行うため，分割と融合についての説明を行う.
第６章では，提案手法の検証を行う．提案手法で，状態がどのように作成されたかを確認する．また提案手法と従来手法を比較することで問題点が改善されたかどうか検証する．そして実際の問題に適用することを考え，具体的なタスクであるダムの放水問題に提案手法を適用し提案手法で学習が行えているかどうかを確認する.
第７章では，最後に本論文のまとめを行い，今後の課題を示す.

# 第２章　強化学習

## 2.1　強化学習の概要

強化学習(Reinforcement Learning)とは，試行錯誤を通じて環境への適応を試みる学習である．実際に学習を行う学習者のことをエージェントと呼ぶ.

強化学習は教師あり学習(Supervised learning)とは異なり，ある状態に対する正しい行動を明示的に示す教師が存在しない．かわりにエージェントは報酬を手がかりに学習を行う．強化学習を使用するときに報酬を設定する際には，望ましい状態には高い値，望ましくない状態には低い値を割り当てて学習を行わせる.

強化学習の最大の利点は学習目標を報酬与えるだけで，目標の状態に至るような行動が得られるという点である．教師あり学習だと設計者は環境の推測が難しい場合や，設計の檀家では分からない未知のパラメータが存在すると，問題の解決方法を推測することが出来ない．しかし強化学習では，達成するべき目標を報酬というパラメータで指示することで簡単に学習を行うことが出来る.

強化学習を用いる際にはエージェントは状態，行動，報酬という３つの項目を用いて学習を行う.

状態とは，エージェントが観測した周囲の環境の状態である.

行動とは，エージェントが行うことの出来る行動の事を指す．エージェントの行動によって状態が変化する.

報酬とは，エージェントが行動をした結果，得られるものである．エージェントは学習を行う時に報酬を利用して評価を行う．報酬はスカラー量で与えることが出来る．強化学習では報酬をより多く得られるように学習を行う.

強化学習はこれらの項目を用いて以下の流れで行う.

ステップｔの時.
・エージェントは環境の状態$S_t$を観測する.
・強化学習により，報酬を入手できると判断した行動$a_t$を行う.
・エージェントの行動によって環境の状態が遷移する$(S_t \rightarrow S_{t+1})$.
・エージェントはその遷移に応じた報酬$r_{t+1}$を得る.
・報酬を得たことにより，現在の状態の時の行動を評価する.
ステップｔ＋1 に遷移.

図 2.1 に強化学習の枠組みを示す.

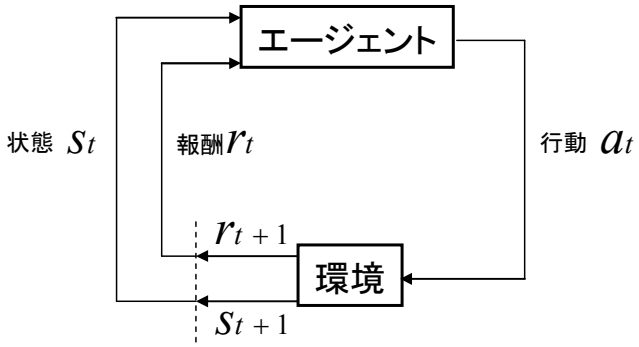



図 2.1　強化学習の枠組み

強化学習には主に次の３つの構成要素がある．

１，方策(policy)

方策とはエージェントの行動を決定する際の方針を定義する．エージェントは行動を選択するとき，方策に基づいて行動を選択する．

２，報酬関数(reward function)

報酬関数とは，エージェントが受け取る報酬を決定するものである．エージェントは行動を行った結果，報酬関数によって導き出された報酬を得る．

３，評価関数(value function)

エージェントがある状態から方策にしたがって行動した時，将来的にどのくらい報酬を得られるかを予測する関数である．評価関数の値が高くなるように方策を改善していくことで学習が行われる．

強化学習はこれらの要素を用いて学習を行う．
次節ではこれらの構成要素を用いて，強化学習を行う学習部について説明を行う．

## 2.2　行動評価手法

行動評価手法とは，どのくらい報酬が得られるかを予測するための手法である．行動評価手法では知識を更新することで，よりどの状態と行動が報酬に結びつくかを算出するものである．知識とはある状態の時にある行動を取ると得ることが出来る報酬の期待値を表し，ここでは Q 値と呼ぶ．

次に本研究で使用する手法について説明する．

・加重平均法

加重平均法とは，以下の(2.1)式で Q 値である $Q(s,a)$ を更新する行動選択手法の一つである．

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[r_{t+1} - Q(s_t,a_t)] \qquad (2.1)$$

$\alpha$ はステップサイズパラメータで，強化学習を適用する際には $0 \leq \alpha \leq 1$ の範囲で任意の値を設定する．

加重平均法の特徴は，$\alpha$ の値の設定によって最新の報酬を重視するか，それまでの経験を重視するかを大まかに設定できる点である．

## 2.3 行動選択手法

行動選択手法とは，方策にあたる手法である．ここでは本研究で使用する手法について説明する．

・ε -greedy 法

エージェントが行動を決定する際に，その時点で最も報酬が得られると判断した行動を選択する．ただし小さい確率 ε で，報酬が得られるかどうかの知識とは無関係にランダムな行動を取る手法である．

小さい確率 ε でランダムな行動を選択することで，価値が低いと判断される行動も選択する．価値が低い行動を選択することで，本当はさらに良い行動かどうか可能性を確かめることが出来る．

## 2.4 まとめ

本章では，本研究で取り上げる強化学習の概要について説明をした．強化学習でエージェントは状態，行動，報酬という項目を使用し学習を行う．次章では，強化学習の問題として状態の設定の方法を取り上げる．

# 第３章　問題提起と解決策

ここでは強化学習を使用する際に重要となる状態の設定方法の問題点について述べる．

## 3.1 強化学習の問題点

### 3.1.1　実機のロボットを扱うときの問題点

強化学習では，エージェントは認識することが出来る環境の状態，エージェントが取る事のできる行動，エージェントが行動した結果で得られる報酬によって，その状態で最大の報酬が得られるような行動を選択するように学習を行う．強化学習を用いる場合，状態や行動は学習時間やメモリの問題等の面から，抽象化を行い必要最低限の数に絞って設定を行うことが一般的である．

ここで強化学習を実機のロボットに適用を行う場合を考える．エージェントにあたるロボットはセンサを通して，現在の自身の状況を判断する．このため学習の際にエージェントが環境から与えられる値はセンサから読み取った値となる．センサから読み取った値というのは，スカラー値で表されたデータであり，強化学習で用いるために抽象化はされていない．このため実機に強化学習を適用する際，実データを使うのは難しいといった問題がある．本研究では，このようにセンサから得られる数値で表されるデータのことを実データと呼ぶ．

実データを扱う問題の中で，最も困難な問題の一つに状態の構成という問題がある．エージェントが強化学習を行う時には，状態一つ一つについてどのような行動を取ればより多くの報酬が得られるのか学習を行う．このため全ての実データを状態として学習を行うと状態の数が大量に出来てしまい，扱うデータ量が莫大になる．また，それぞれの値ごとに学習を行うため，学習を行う対象が多すぎて学習効率が落ちるといった問題がある．

このような実データの問題を解決するために，従来の手法では，設計者が実データから状態に決定するように設計を行っている．従来手法では，まず設計者はエージェントにセンサからどのような値が入力されかを考える．設計者はエージェントが扱う値の推測を行い，どのような実データがどの状態に当てはまるのかを，エージェントが取り組む問題やエージェントが学習を行う環境を想定して設計する．エージェントが学習を行う時には，環境の情報である実データをセンサから得る．エージェントはセンサから得た実データから状態を決定して学習を行う．以上の流れで実データを強化学習に適用している．

しかし従来の手法では，設計者が設計の段階で状態の設定を行うため，学習を行う環境をある程度知る必要がある．このため未知のパラメータなどが存在し，環境の推測が難しい問題では学習を行えるように状態の設定が行えない．また設計者が設定した状態は必ず

しも環境に適した状態の設定になっているとは限らない．これらが原因で状態に対して，最も報酬を得ることのできる行動が決定できずに，学習が正しく行えないことが考えられる．

## 3.2　問題点の解決策

ここで従来の手法での問題点を解決するために以下の解決策を提案する．

従来の手法では，設計者が環境を推測して状態を設計することによって問題が発生している．本研究では，この強化学習の問題点を解決するために，設計者が強化学習の状態を設計するのではなく，エージェントが自身で状態の設定(状態学習)を行い，タスクに対する行動学習を行う手法を提案する．エージェントが自身で状態を設定することで，設計者は設計の段階で学習を行う環境の推測を行わず，エージェントは学習を行う環境ごとに状態を生成して行動学習を行うことが出来る．これによって設計の段階で環境の推測を行わなくてよくなり，環境ごとに状態の設定を行わなくて良くなる．

実機のロボットに強化学習を用いて常態学習を行う場合，エージェントであるロボットは環境から得た情報である実データを使用する．入力された実データから状態を決定する．エージェントは状態の判断基準を決定するため過去の経験を用いて，実データを状態に振り分ける基準を学習する．ここで言う過去の経験とは，エージェントが学習を行う過程で得られた報酬などのデータのことである．

エージェントはセンサから読み取った値や行動の結果によって得られた報酬など，過去の経験を用いることで，センサから読み取った実データの値を適切な状態として扱えるような状態の設定を学習する．

以下にエージェントが状態を自動で設定し学習を行う流れを載せる．

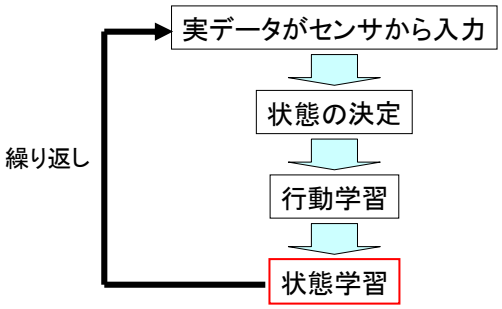



図 3.1　解決法の流れ

以上の工程を繰り返すことで，エージェントは効率よく学習が出来るように状態を設定しながら学習を行うことを実現する．

次章では具体的に従来の手法でどのように問題が発生するかを確認する．

# 第 4 章　従来手法の検証

本章では，第 3 章で紹介した強化学習の従来手法を検証する．従来手法の検証を行う目的は，実験を行うことで状態の設定が正しく行われていない場合に発生する問題を確かめ，状態の設定が重要であることを確認するためである．

## 4.1　実験内容

本章の実験では従来手法では状態の設定が学習に影響を及ぼすことを確認する．今回はエージェントに実データが入力された時に適切な行動を取ると報酬が得られるという簡単な問題に対して，状態の設定による学習効率の違いを確認する．

実験ではまずセンサから入力される実データを仮想で作成する．次に作成したデータを用いて，従来の手法で行われているように強化学習の設計の段階で状態の設計を予め行って学習を行う．検証ではシステムを二種類用意する．一つ目は状態によって行動が確定できるように状態を設定したシステムである．もう一つは状態によって行動が定まらないように設定を行ったシステムである．本研究では前者を適切な状態設定を行ったシステム，後者を不適切な状態設定を行ったシステムとする．これら二つのシステムを比較することで，状態の設定が正しく行われないと学習が正しく行われないといった問題が発生することを確認する．今回の実験では，実機を用いずにシミュレーションで行う．

## 4.2　実験設定

### 4.2.1　データの作成

エージェントに入力される実データ $D(k)$ を作成する．今回作成したデータは推移している値にノイズを乗せたものを想定して作成を行っている．データは以下の式で作成する．

$$D(k+1) = D(k) \pm c \qquad (0 \le D(k) \le 1) \qquad (4.1)$$

k はステップ数を表し，作成するデータ数 M の分だけ計算する．またその時々の k をデータ番号と呼ぶ．

例：データ番号 1000 の時のデータの値は $D(1000)$ .

c はランダムノイズを表している．

また式(4.1)の ± はデータ数 n 回ごとに確率 p で変化するものとする．

式(4.1)より，本実験では以下のようにパラメータの初期値を設定し，データを作成した．

表 4.1　データ作成に用いたパラメータの初期値設定

| パラメータ | 記号 | 初期値 |
|---|---|---|
| 実データの初期値 | D(1) | 0.5 |
| データ数 | M | 30000 |
| ランダムノイズ | c | 0<c<0.1 |
| 変化の間隔 | n | 10 |
| 変化の確率 | p | 0.1 |

初期値により作成したデータのデータ番号 1 から 10000 までを図 4.1 に示す．

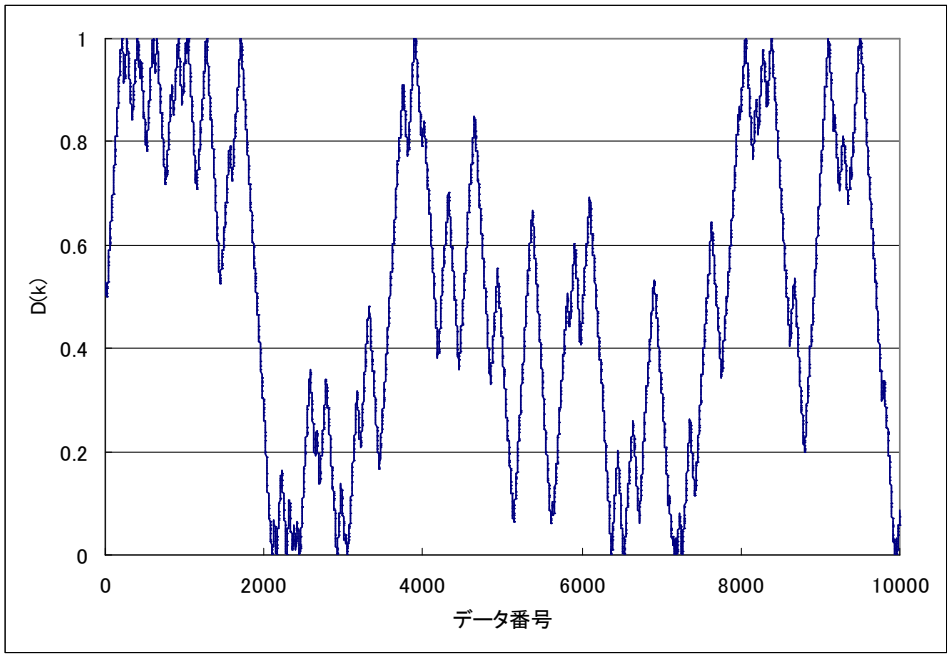



図 4.1　作成した実データ

### 4.2.2　強化学習

ここでは強化学習に関する設定を説明する．

・学習を行うタイミングと実データの扱い方

この実験では入力される実データで定義したデータ番号を１単位時間として扱う．強化学習を行う際にはデータ番号１つを１ステップとする．エージェントにはデータ番号ごとにデータ値 $D(k)$ が入力される．またエージェントの学習は１データごとに行うもい，学習の試行回数は作成したデータ数行う．以下に概要図を示す．

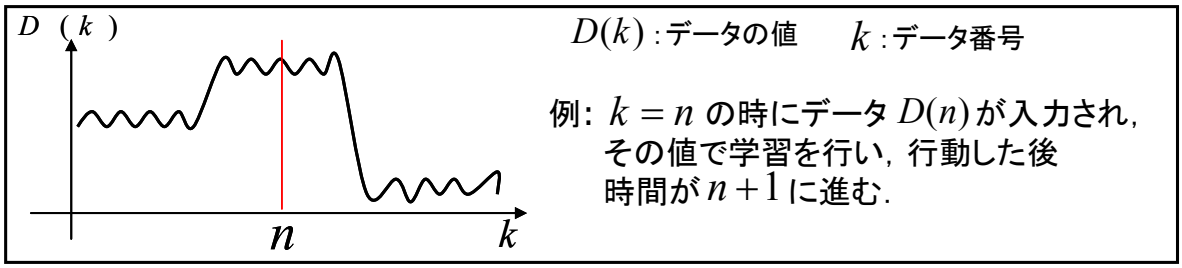



図 4.2　実データの扱い方

・エージェントの状態，行動について

状態：本実験ではを閾値を設定の上で３つの状態を作成する．3 段階に分けた状態をそれぞれ，$S_0$，$S_1$，$S_2$ とする．エージェントはその状態を使用して学習を行う．状態の具体的な設定方法については「4.2.3　状態の設定と報酬の設定」で述べる．

行動：それぞれの３つの状態の時に A，B，C いずれかの行動を行う．エージェントはどの行動が現在の状態で最も報酬が得られるであろう行動がどれかを学習する．

・行動学習手法

本実験では加重平均法を用いる．加重平均法は以下の式によって Q 値である $Q(s,a)$ を更新する．なお Q 値とは行動を決定するための知識で，ある状態の時にある行動を取ると得られるであろう報酬の期待値である．

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} - Q(s_t, a_t)] \qquad (4.1)$$

・行動選択手法

本実験では ε -greedy 法を用いる．ε -greedy 法とは最も高い Q 値を持つ行動を選択するが，ε の確率でそれとは無関係にランダムな行動を選ぶ手法である．

以下の表は行動学習手法と行動選択手法で用いるパラメータの初期値である．

表 4.2　強化学習で使用するパラメータの初期値設定

| パラメータ | 記号 | 初期値 |
|---|---|---|
| ステップサイズパラメータ | $\alpha$ | 0.01 |
| 発生確率 | $\varepsilon$ | 0.1 |

### 4.2.3　報酬の設定と各システムの状態の設定

・報酬の設定

本実験では，適切な状態設定を行ったシステムと不適切な状態設定を行ったシステムの二種類を構築する．そのためにまず環境に対する報酬の与え方を定義する．報酬を決定することでどのような設定を行うと適切な状態設定が出来ているかを決定することが出来る．

報酬の設定は実データの範囲を決定し，その範囲内で特定の行動を取ると報酬が得られるように設定した．報酬の設定を表 4.3 に示す．

表 4.3　データの値と行動に関する報酬の設定

| 行動 / データ：$D(k)$ | A | B | C |
|---|---|---|---|
| $0.65 \le D(k) \le 1$ | 1 | 0 | 0 |
| $0.35 < D(k) < 0.65$ | 0 | 1 | 0 |
| $0 \le D(k) \le 0.35$ | 0 | 0 | 1 |

表 4.3 より，例えば$0.65 \le D(k) \le 1$の時に行動 A を行うと報酬を 1 得るというようにそれぞれのデータの値の幅と行動によって報酬を設定した．

この報酬の設定を基に，状態によって行動が確定できるように状態を設定したシステムと状態によって行動が定まらないように設定を行ったシステムの二種類を比較する．

・各システムの状態の設定

次に比較する二種類の状態の設定を行う．実験では，適切な状態設定を行ったシステムと不適切な状態設定を行ったシステムの二種類を比較する．前者をシステム 1 とし，後者をシステム 2 とする．それぞれのシステムの設定を以下に示す．

表 4.4　システム１：適切な状態設定

| 実データの範囲 | 状態 |
|---|---|
| $0.65 \le D(k) \le 1$ | $S_0$ |
| $0.35 \le D(k) \le 0.65$ | $S_1$ |
| $0 \le D(k) \le 0.35$ | $S_2$ |

表 4.5　システム２：不適切な状態設定

| 実データの範囲 | 状態 |
|---|---|
| $0.8 \le D(k) \le 1$ | $S_0$ |
| $0.2 \le D(k) \le 0.8$ | $S_1$ |
| $0 \le D(k) \le 0.2$ | $S_2$ |

システム１では状態の設定で $D(k)$ の範囲が，報酬の設定と同じ範囲で設定を行っている．報酬の設定と同じ範囲で状態の設定を行っていることで，その状態の時にどの行動を取ると報酬が得られるかが確定するため，状態ごとに行動が決定するように学習を行える．

一方，システム２では状態を設定で $D(k)$ の範囲が，報酬の設定とは異なり，$S_1$ の範囲が広くなっている．この事により状態 $S_1$ の時，全ての行動で報酬を得る可能性がある．実データにより報酬を得ることが出来る行動が変わってしまい，状態によって行動が特定できないため，正しく学習が出来ないと推測できる．これらの関係の概要図を図 4.3 に表す．

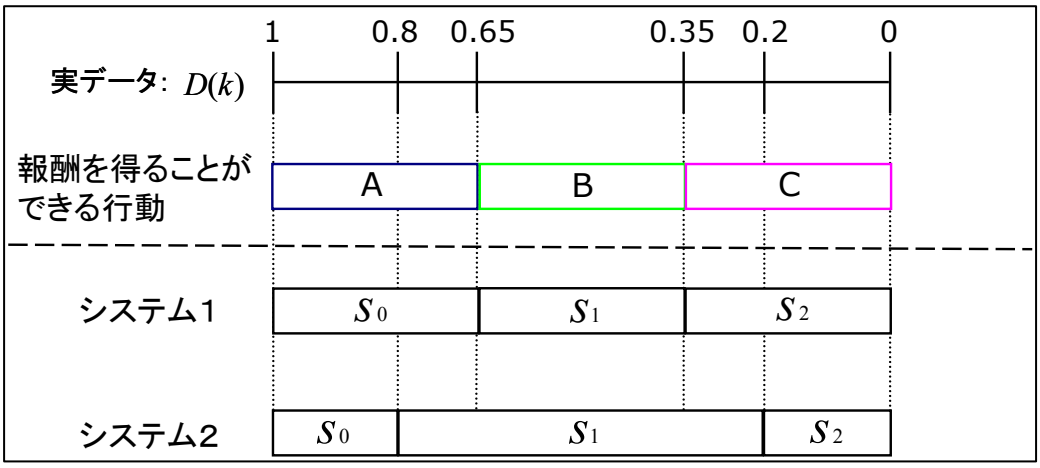



図 4.3　報酬の設定とそれぞれのシステムの状態設定の関係

図 4.3 を見るとシステム２では実データ 0.7 が入った時に状態 $S_1$ と判断する．報酬の設定では行動 A を選択すると報酬を得ることが出来るが，システム２の状態 $S_1$ では行動 B や行動 C でも報酬が得られる時があるため，0.7 が入力された時に行動 A を選択できない可能性が高い

## 4.3　実験結果

適切な状態設定を行ったシステム１と不適切な状態設定を行ったシステム２でそれぞれ学習を行った結果を示す.

実験結果ではまず，それぞれのシステムで報酬の設定範囲での行動の選択率を見て，報酬が得られるように行動を選択しているかどうか確認する．次にシステムごとの最終的な報酬の入手率を比較し，学習効率にどの程度影響を及ぼしているかを確認する．最後にシステム２で行動が特定できないと推測される状態 $S_1$ の報酬の期待値の推移を確認し，エージェントがどのように学習しているかを考察する.

報酬の設定で用いたデータ値 $D(k)$ の範囲ごとの行動の選択率を以下に示す．行動の選択率は，それぞれの行動の回数から，データ値 $D(k)$ の範囲ごとに選択された行動の総数を割ったものである.

・システム１の行動選択率

表 4.6　システム１：データ値 $D(k)$ ごとの行動の選択率

| データ値＼行動 | A | B | C |
|---|---|---|---|
| $0.65 \le D(k) \le 1$ | 93.29 | 3.57 | 3.14 |
| $0.35 < D(k) < 0.65$ | 3.10 | 93.31 | 3.58 |
| $0 \le D(k) \le 0.35$ | 3.12 | 3.47 | 93.42 |

単位：%

・システム２の行動選択率

表 4.7　システム２：データ値 $D(k)$ ごとの行動の選択率

| データ値＼行動 | A | B | C |
|---|---|---|---|
| $0.65 \le D(k) \le 1$ | 77.53 | 18.33 | 4.15 |
| $0.35 < D(k) < 0.65$ | 13.85 | 72.40 | 13.75 |
| $0 \le D(k) \le 0.35$ | 4.76 | 21.13 | 74.10 |

単位：%

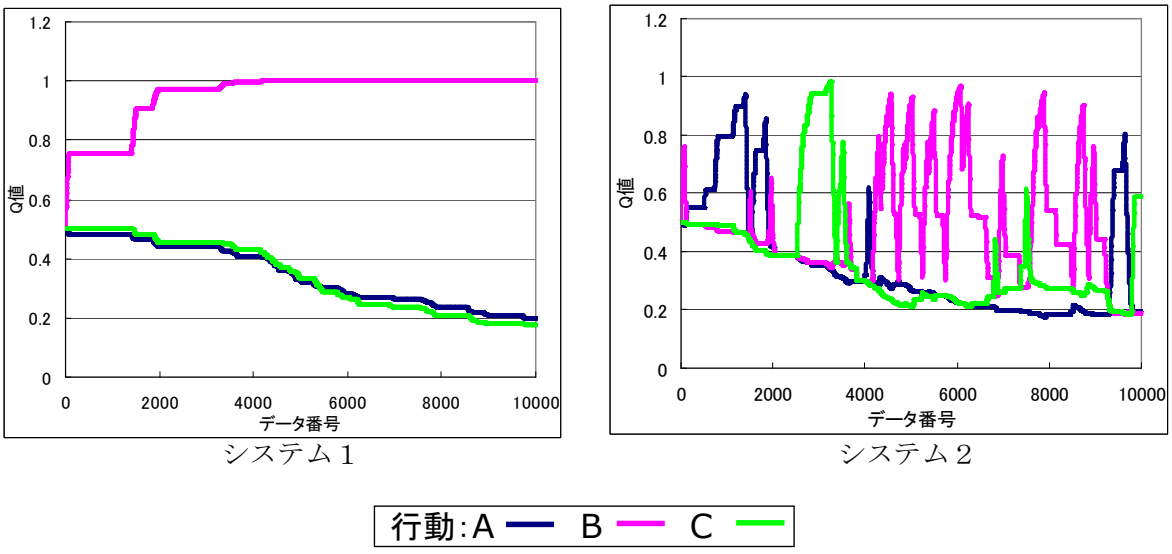
・報酬入手率の比較

今回学習を行ったシステムと学習を行わずにランダムに行動をしたものの報酬の獲得率を示す．報酬の入手率は，入手した報酬の総和を試行回数で割ったものである．

表 4.8　それぞれのエージェントとランダム行動の報酬の獲得率

| | 報酬入手率(%) |
|---|---|
| システム１ | 93.34 |
| システム２ | 74.72 |
| ランダム選択 | 33.65 |

・システム１，システム２の状態 $S_1$ の報酬の期待値の推移




システム１　　システム２

図 4.4　それぞれのシステムの状態 $S_1$ での Q 値の推移

## 4.4 考察

表 4.6 を見ると, それぞれのデータ値 $D(k)$ の範囲で報酬が得られる行動が 90%以上選択されていることがわかる．システム 1 では学習の結果，報酬が得られる行動を学習して選択していることが分かる．

次に表 4.7 を見ると，こちらもそれぞれのデータ値 $D(k)$ の範囲で報酬が得られる行動が最も選択されている．ただし報酬が得られる行動の選択が 70%程度になっている．また報酬が得られない行動の選択率がシステム 1 と比べ上昇しているものがある．これは状態の設定が報酬の設定と異なるように設定しているため，実データの値によって報酬を得ることが出来ない行動を選択してしまうためである．

これらの結果から，それぞれのシステムで言えることは，報酬を得ることが出来る行動の選択率が最も高いことである．これは学習を行って行動を選択していることがわかる．しかしシステム 2 ではシステム 1 に比べて報酬を得ることが出来る行動の選択率が低くなっている．これは状態の設定が正しく行えていないためであると考えられる．例えばデータ値が $0.35 \leq D(k) \leq 0.65$ の時では，行動 B を行うと報酬を得ることが出来るが，表 4.5 の結果を見るとシステム 2 では，ほかの二つの行動もシステム 1 に比べ，ある程度選択されている．これはシステム 2 では $0.2 \leq D(k) \leq 0.8$ を状態 $S_1$ として扱っているため，図 4.3 よりこの状態ではどの行動でも報酬が得られる可能性があるためである．よってシステム 2 では状態ごとに報酬を得られる行動が特定されないため，選択率にばらつきが生じているものと考えられる．

次に表 4.8 より報酬の入手率について考察を行う．この結果では，ランダムに行動を選択しているものと比較するとそれぞれのシステムではランダムより多く報酬を獲得していることがわかる．ただ状態の設定が間違えているシステム 2 はシステム 1 に比べて報酬の取得率が低下していることがわかる．

行動の選択率と報酬の入手率の検討を行った結果，システム 2 では学習効率が低下していることが分かった．ここでシステム 2 では具体的にどのように学習を行っていたかを図 4.4 のそれぞれのシステムの状態 $S_1$ での Q 値の推移によって考察を行う．

Q 値とは報酬の期待値なので，今回の学習手法では $\varepsilon$ の確率ではランダムな行動を選択されるが，それ以外の時には最も Q 値が高い行動が選択される．システム 1 は行動 B が最も Q 値が高く，また値が収束している．このため状態 $S_1$ では行動 B を取ればよいと学習している．一方システム 2 ではその時々により一番 Q 値が高い行動が変わっている．またどの行動も収束していないので，この状態ではどの行動を取ればよいかが確定していない．よって学習が正しく行われていないことがわかる．

## 4.5 まとめ

本実験で言えることは，状態の設定を間違えて行うと，報酬の獲得率の低下や，ある状態に対して報酬が得られる行動が特定できないなど，学習に支障が出ることがわかった．このため強化学習を用いる際には，環境や報酬を踏まえて状態を設定する必要があることが確認できた．

# 第 5 章　提案手法

本章では，3 章で紹介した実データから状態の自動設定を行うための具体的な手法について提案する．状態の自動設定を行うことにより，従来の強化学習の問題点である状態の設定についての問題の解決を行う．

はじめに提案手法の概要を説明し，提案手法の全体像を示す．次に提案手法の流れを説明した後，詳細な説明する．

## 5.1　概要

本研究では，センサ等から読み取った値である実データを強化学習の状態として使用するために，状態の自動設定を行う手法を提案する．状態の自動設定を行うことによって，エージェントが状態を自動で設定し，状態の設定を変えながら学習を行い，より学習がしやすいように状態を設定することができる．

提案手法では，エージェントは従来の強化学習と同じように，あるタスクに対する学習を行わせ，その学習で得られた報酬等のデータを用いて状態の設定を自動で行う．状態を作成することで，その都度作成した状態を利用して学習を行う．

## 5.2　提案手法の流れ

本研究では状態の自動設定を実現するために，センサのレンジに上限と下限の範囲があることに注目した．実機のロボット等を使用する場合，ハードウェア的な理由でセンサに入力される値の最大値や最小値が存在する．この最大値や最小値を振り切ってしまうとセンサでは値を読み取ることが出来ない．つまり実機のロボットがセンサを使って学習を行う際には，センサから読み取れる範囲内のデータで学習を行うことになる．本研究では，このセンサで読み取ることの出来る範囲を分割していくことで状態の自動設定を実現する．

センサのレンジの最大値を $D_{\max}$，最小値 $D_{\min}$ とする．概要図を以下に示す．

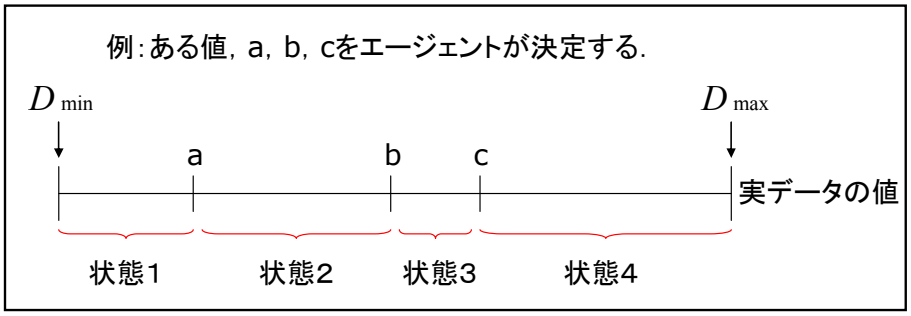



図 5.1：状態の作成の概要図

次に強化学習で行う行動学習，及び状態の自動設定を行う状態学習の流れについて説明を行う．提案手法の流れは図 5.4 のようになる．

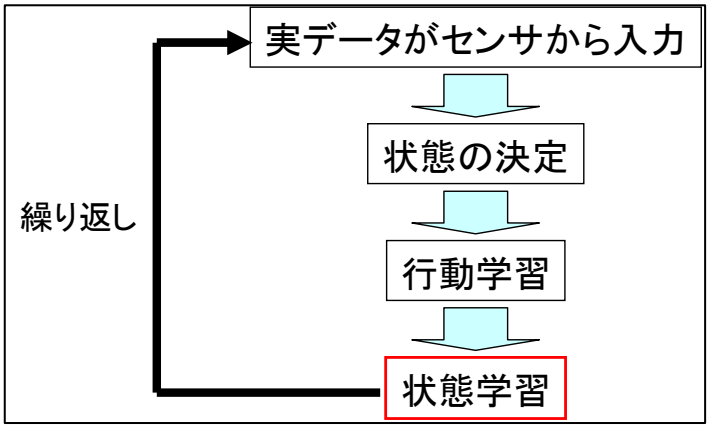



図 5.2　提案手法の流れ

提案手法ではまずセンサから読み取った実データの値がどの状態に当てはまるかを決定する．決定した状態を用いてエージェントはタスクに対して行動学習を行う．エージェントは学習を行った際に得られたデータを用いて状態を設定するための状態学習を行う．状態学習を行うことで，環境ごとの学習に適した状態を設定して学習を行うことが出来るシステムを実現する．

状態の設定は，各行動ごとに行う．学習の初めの段階ではセンサから読み取ることが可能な全ての範囲を 1 つの状態として扱う．行動学習を行うに従い，エージェントはセンサから読み取った値や報酬などのデータを増やしていく．エージェントはこれらのデータを用いて，この状態が本当に正しいのかそれぞれの行動ごとの状態について学習を行う．エージェントは状態が間違えていると判断した場合は規則に従い，状態を設定しなおす．状態の設定の変更を繰り返すことで，より効率よくタスクに対する行動学習が行えるように状態設定の学習を行う．

今回の手法では，この状態学習を分割と融合を繰り返すことで実現する．分割というのはある状態の設定が適切ではないと判断された時に，一つの状態を複数の状態に分けることで，状態の設定を変えることである．(図 5.3)

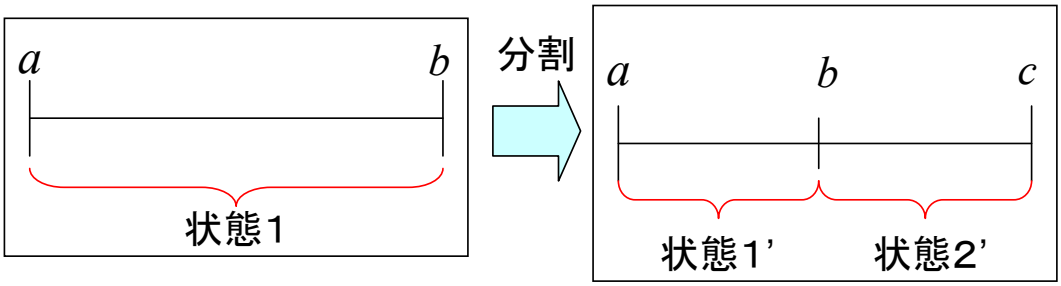



図 5.3　分割の例

融合というのは 2 つの状態が似ている場合に同一と見なし，一つの状態に統合することである．(図 5.4)

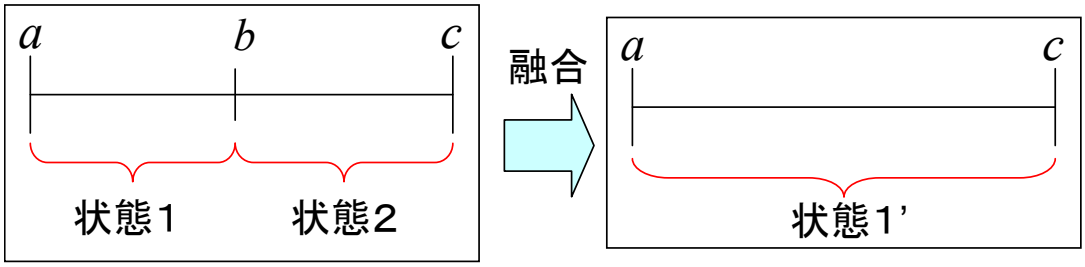



図 5.4　融合の例

エージェントは状態の分割と融合を繰り返すことで学習を行っている環境に適したグループの分け方を見つけようとする．これによって状態の数が過剰に多くならないように状態を設定して学習を行うことが出来る．

次節では，状態学習を行うための具体的な方法について説明をする．

## 5.3 提案手法の詳細

本節では，提案手法での状態の定義，学習で使用する各項目及び，分割，融合ついて説明する．それぞれの算出方法を以下に記述する．

なお本節ではエージェントは以下の動作をして，以下の情報を得たものとする．

ステップ $t$ の時.
ある入力された実データの値 $D(t)$ を判断する.
その実データ $D(t)$ に対応するグループを決定する.
強化学習の行動選択手法を用いて，状態 A の時，行動 $a_t$ を選択する.
状態 A の時に行動 $a_t$ を行った結果，得られた報酬を $r_t$ とする.

### 5.3.1 状態について

今回の手法では，センサでレンジを利用して状態の設定を行う．ここで状態は次のように定義する.

状態は始点，終点の値を持つ．実データがどの状態に属するかを検索する際には始点，終点を使用する．また状態ごとに報酬の期待値 E を持つ.

次に学習で状態を使用するときの説明をする．学習を行う時は実データがどの状態に属するかを決定しなければいけない．今回の手法では状態の始点と終点を利用する．実データ $D(t)$ がどのグループに属するかを検索する時は,

$$(\text{始点}) < D(t) < (\text{終点})$$

が成り立つ全てのグループに属する.

グループ A の始点を $a_1$，終点を $a_2$ とすると，実データ $D(t)$ が始点 $a_1$ より大きく，終点 $a_2$ より小さいならば，実データ $D(t)$ はグループ A に属することになる．この手順で当てはまるすべての状態を決定する.

センサで読み取ることの出来る最大値を $D_{\max}$ ，最小値を $D_{\min}$ とするとグループの概略図は図 5.5 のようになる.

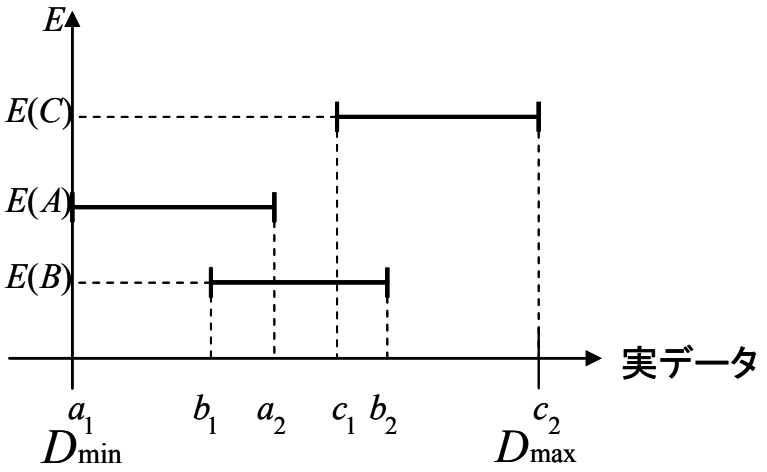



図 5.5　作成されたグループの一例

### 5.3.2　使用するパラメータの更新について

分割と融合をどのような時に行えばよいか考える．今回の提案手法では以下のような時に分割と融合が起きるようにする．

・分割
ある状態が充分に学習されているにもかかわらず，報酬が確定的ではない時．
・融合
それぞれの隣り合った状態が充分に学習が行われている．またそれぞれの状態が同じくらいの報酬の期待値を持っていて，得られる報酬が確定的である時．

これらの条件を満たしているほど高い確率で分割・融合が発生するようにした．提案手法ではこれらの条件を判定するために，それぞれの状態ごとに３つの項目を設定する．ここで紹介する項目は，エージェントが学習を通して得たデータを使用し，それぞれの項目を更新する．

１、要素数（N）
エージェントは実データが入ってきた時に，その状態が何回参照されているかをカウントする．要素数が多いほど，その状態の情報量が多く，充分に学習を行ったと判断する．要素数は以下の式で更新する．

$$N(A, t+1) = 1 + (1 - \frac{1}{x}) \cdot N(A, t) \qquad (5.1)$$

x は収束させたい値を設定する．学習中では要素数が x に近づくほどその状態の学習が充分になされたと判断する．

また式(5.1)でカウント数を更新することで N の値が無限に増え続けることを防止することも出来る．なお他の計算で要素数 N を使用する場合は，N の値は求めた値以上の最小の整数とする．

２、期待値（$E$ ）

期待値はエージェントが強化学習を用いて行動を決定する際に使用する．ここで求める期待値とは強化学習の行動学習手法に当たるもので，行動学習の際にも使用する．今回の手法では，ある状態の報酬の期待値を以下の式により更新する．

$$E(A, t+1) = \frac{N(A, t) \cdot E(A, t) + r_t}{N(A, t) + 1} \qquad (5.2)$$

３、分散（σ）

その状態の時に得られた報酬の分散を表す．分散が小さいほどその状態で得られる報酬は確定的であることが表せる．そのグループの報酬の分散を以下の式により更新する．

$$\sigma^2(A, t+1) = \frac{N(A, t) \cdot \sigma^2(A, t) + (E(A, t+1) - r_t)^2}{N(A, t) + 1} \qquad (5.3)$$

また分散の値が大きいほど，様々な報酬が得られる実データが入ってくると判断できる．よって分散が大きいほどその状態の設定は適切ではないと判断できる．

それぞれの項目を更新し，以降に記述する分割の発生条件及び，融合の発生条件を求める．

### 5.3.3　分割について

分割については，分割を行う方法と分割が発生する条件について説明する．状態学習で実際に分割が発生する場合，分割が発生する条件を満たした場合，分割を行う方法で説明する方法で分割を行う．

・分割の方法について

本研究では，センサのレンジの最大値と最小値の間を分割していくことで状態の自動設

定を実現する．分割が発生したとき，今回の手法では１つの状態を２つの状態に分割する方法と３つに分割する方法を紹介する．

（１）２つの状態に分割する方法

分割を行う対象の状態の始点を a,終点を b とする．
２つの状態に分割を行う．それぞれの状態の始点終点は，

分割後状態１…始点 $a$ ，終点 $\frac{a+b}{2}$ ．

分割後状態２…始点 $\frac{a+b}{2}$ ，終点 $b$ ．

のように定義する．また各々の要素数，期待値，分散は以下のように設定しなおす．

分割する前の状態を A，分割した後の状態を A‘，B’とすると分割が発生した後は以下のように更新する。

・要素数

$$N(A')=1, N(B')=1 \qquad (5.4)$$

・期待値

$$E(A')=E(A), E(B')=(A) \qquad (5.5)$$

・分散

$$\sigma^2(A')=0, \sigma^2(B')=0 \qquad (5.6)$$

２つの状態に分割を行った時の概念図を以下に示す．

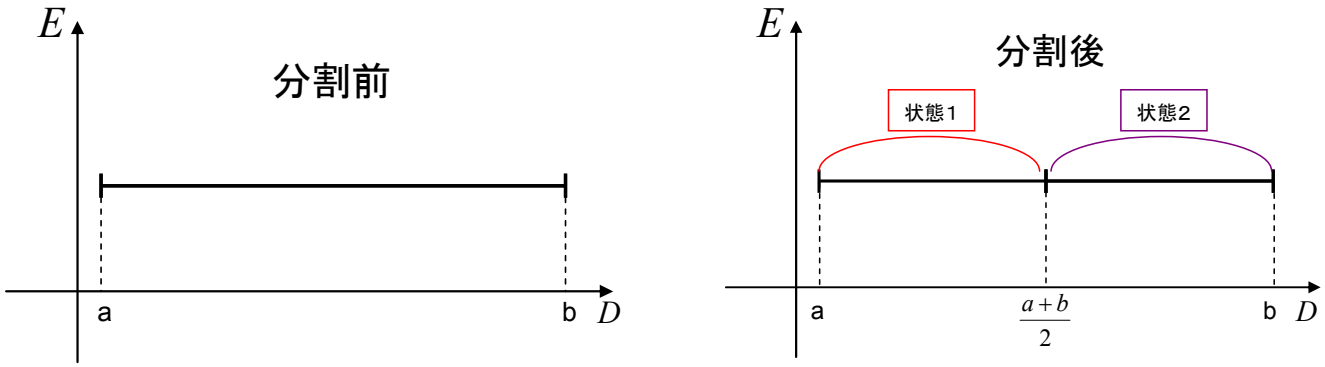



図 5.6　状態を２つの状態に分割を行った時

（２） ３つの状態に分割する方法

分割を行う対象の状態の始点を a,終点を b とする.
３つの状態に分割を行う．それぞれの状態の始点終点は,

分割後状態１…始点 $a$ ，終点 $\frac{a+b}{2}$ .

分割後状態２…始点 $\frac{a+b}{2}$ ，終点 $b$ .

分割後状態３…始点 $\frac{3a+b}{4}$ ，終点 $\frac{a+3b}{4}$ .

のように定義する．また各々の状態の要素数，期待値，分散は以下のように設定しなおす.
分割する前の状態を A，分割した後の状態を A ‘，B’，C ‘とすると分割が発生した後は以下のように更新する。

・要素数

$$N(A')=1, N(B')=1, N(C')=1 \qquad (5.7)$$

・期待値

$$E(A')=E(A), E(B')=(A), E(C')=E(A) \qquad (5.8)$$

・分散

$$\sigma^2(A')=0, \sigma^2(B')=0, \sigma^2(C')=0 \qquad (5.9)$$

３つの分割の概念図を以下に示す.

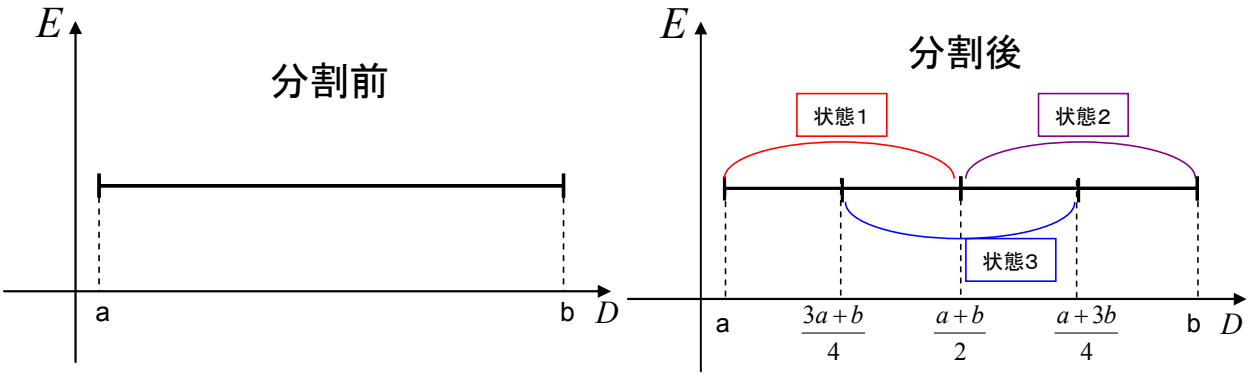



図 5.7　状態を３つの状態に分割を行った時

・分割が発生する条件

分割のタイミングは，要素数などそれぞれのパラメータが更新された後に求める分割の発生確率により発生する．また分割の対象になる状態はその試行のときに決定した全ての状態が対象なる．

分割が発生する時は，充分な情報量があるにもかかわらず，得られる報酬が確定的ではないときに発生する．

よって提案手法では以下の時に発生確率が高くなるように設定を行う．

１，その状態の要素数が多い時

２，その状態の報酬の分散が大きい時

これらの条件を満たすほど分割が発生する確率を上げるようにする．

分割を行う状態を状態 A とすると，分割が発生する確率は以下の式で求める．

$$P_{division}(A) = u(A) \cdot v(A) \qquad (0 \le P_{division}(A) \le 1) \quad (5.10)$$

式(5.10)の右辺の要素は式(5,11)及び，式(5.12)で求める．

$$u(A) = \frac{1}{1 + e^{\alpha(-N(A)+\beta)}} \qquad (5.11)$$

$u(A)$ はシグモイド関数を用いた要素数 N に関する式である．要素数 N は式(5.1)で求めたものを使用する．

$\alpha$ はシグモイド関数の収束する速度を表し，$\beta$ はシグモイド関数の編曲点を表す．$\alpha$，$\beta$ はそれぞれ(5.2)式で収束させる値によって変動する．

$\alpha$ は任意で設定する．$\beta$ は式(5.2)で設定する x の値を用いて $\beta = \frac{x}{2}$ とする．

式(5.5)では要素数 N が大きくなればなるほど $u(A)$ が 1 に近づく．このため要素数が多くなるほど分割の確率が上がる．

$$v(A) = \frac{\sigma^2(A)}{\sigma^2_{\max}} \qquad (5.12)$$

式(5.12)は分散に関する式である．分散は式(5.3)で求めたものを使用する．また $\sigma^2_{\max}$ は以下の式で求める．

$$\sigma_{\max}^{2} = \left( \frac{r_{\max} - r_{\min}}{2} \right)^{2} \qquad (5.13)$$

$r_{\max}$ はエージェントが今まで得た報酬の最大値で，$r_{\min}$ はエージェントが今まで得た報酬の最小値である．式(5.13)より，想定される分散の最大値が求まる．

式(5.12)では分散が大きくなるほど値が大きくなる．このため分散が大きいほど分割の確率が上がる．

これらを計算することで $P_{division}$ を求める．$P_{division}$ はその状態の要素数と分散が大きくなるほど値が大きくなり分割する確率が大きくなる．よって要素数が集まっているにもかかわらず確定的ではない状態ほど分割が発生することになる．

### 5.3.4 融合について

融合については，融合を行う方法と融合が発生する条件について説明する．状態学習で実際に融合が発生する場合，融合が発生する条件を満たした場合，融合を行う方法で説明する方法で分割を行う．

・融合の方法について

融合については状態の始点と終点が交わっている状態同士を対象とする．融合を行う場合は，融合の元になる状態と融合される状態の二つが存在する．

状態 A の始点を min_A，終点を max_A，状態 B の始点を min_B，終点を max_B とすると，以下の条件の内一つが当てはまる時に状態 A と状態 B が融合の対象となる．

条件１：$\min\_A \le \max\_B$ かつ $\max\_A \ge \max\_B$
条件２：$\max\_A \ge \min\_B$ かつ $\min\_A \le \min\_B$
条件３：$\min\_A \le \min\_B$ かつ $\max\_A \ge \max\_B$

融合する条件の一例を概略図で以下に示す．

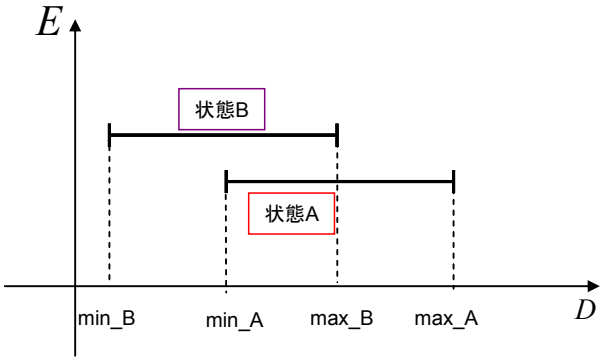



図 5.8　融合条件 1 の例

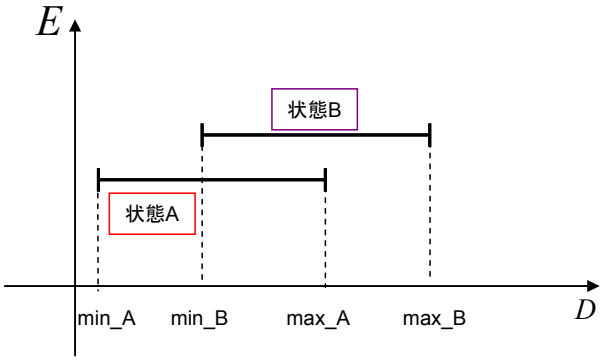



図 5.9　融合条件 2 の例

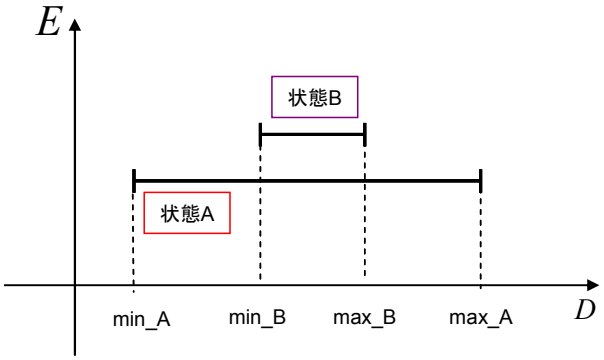



図 5.10　融合条件 2 の例

融合が発生した場合，融合後の状態の始点はそれぞれの始点を比較して小さい値，終点はそれぞれの終点を比較して大きい値を採用する．要素数，期待値，分散は以下のように引き継ぐ．

融合する前の状態を A，B，融合した後の状態を A‘とすると融合が発生した後は以下のように更新する。

・要素数

$$N(A') = 1 \quad (5.14)$$

・期待値

$$E(A') = \frac{E(A) \cdot N(A) + E(B) \cdot N(B)}{N(A) + N(B)} \quad (5.15)$$

・分散

$$\sigma^2(A') = 0 \quad (5.16)$$

・融合が発生する条件

確率によって融合が発生するタイミングは，要素数などそれぞれのパラメータが更新された後に求める融合の発生確率により発生する．また融合の元になる状態はその試行のときに参照された全ての状態が対象なる．

融合が発生する時は，それぞれの隣り合った状態が充分な情報量がある．また同じくらいの報酬の期待値を持っていて，得られる報酬が確定的である時に発生する．
よって提案手法では以下の時に発生確率が高くなるように設定を行う．

１，それぞれの状態の要素数が多い時
２，それぞれの状態の期待値が似たような値の時
３，それぞれの状態の分散が小さいとき

これらの条件を満たすほど融合が発生する確率を上げるようにする．

融合を行う状態を状態 A，状態 B とすると，融合が発生する確率は以下の式で求める．

$$P_{fusion}(AB) = c \cdot f(AB) \times d \cdot g(AB) \times u(A) \cdot u(B)$$

$$(0 \le P_{fusion}(AB) \le 1) \quad (5.17)$$

式(5.17)の右辺は式(5.11)と次に説明する式(5.18)，式(5.19)で求める．

$$f(AB) = \frac{1}{e^{\frac{\sigma^2(A)}{\tau_1}} \cdot e^{\frac{\sigma^2(B)}{\tau_1}}} \quad (5.18)$$

式(5.18)は分散に関する式である．右辺で使用する分散の値は式(5.3)で求めたものを使用する．この式では状態 A と状態 B の分散がそれぞれ小さい場合に得られる値が大きくなる．このためそれぞれの状態で得られる報酬が確定的な場合，融合する確率が上がる．

$$g(AB) = \frac{1}{(e^{\frac{h(A)}{\tau_2}} - e^{\frac{h(B)}{\tau_2}})^2 + 1} \qquad (5.19)$$

式(5.19)は期待値に関する式である．$h(A)$ は以下の式で求める．

$$h(A) = \frac{E(A) - E_{\min}}{E_{\max} - E_{\min}} \qquad (5.20)$$

$E_{\max}$ はエージェントが今まで算出した報酬の期待値の最大値である．$E_{\min}$ はエージェントが今まで算出した報酬の期待値の最小値である．右辺で使用する期待値の値は式(5.2)で求めたものを使用する．

式(5.19)は状態 A と状態 B の期待値の差が小さい場合に得られる値が大きくなる．このためそれぞれの状態の期待値が似ている値な場合，融合する確率が上がる．

これらを計算することで $P_{fusion}$ を求める．$P_{fusion}$ は要素数が大きくなるほど大きくなる．

またそれぞれのグループの分散が小さい場合に大きくなる．さらにそれぞれのグループの期待値の差が小さい場合に大きくなる．よって要素数が集まっていて，確定的で似た価値を持つグループほど融合が発生することになる．

次章では，本章で紹介した提案手法の検証を行い，提案手法を用いて学習がどのように行われるのかを検証する

# 第６章　提案システムの検証

本章では，4 章で取り扱った問題とダムの放水問題を用いて提案手法の検証を行う．まず 4 章の従来の手法で行った実験と同じものを提案システムで行い，状態学習が出来ているかを確認し，提案システムが従来の手法での問題点を解決しているかどうか検証を行う．次に提案システムを状態の遷移が加わりより複雑な問題例としてダムの放水問題に適用し，提案システムの有用性を検証する．

## 6.1　従来システムとの比較

本節では従来手法との比較を行う．４章で行った実験を提案システムに適用し，行動学習が適切に行えるように状態学習が行われているか，また従来の手法とは学習効率に差が生まれるかを検証し，従来の手法での問題点を解決しているか確認をする．

### 6.1.1　実験内容

本実験では，４章で行った従来の手法の検証で用いたデータや報酬等，同じ環境を使用して実験を行う．実験ではエージェントは，ある実データに対して，適切な行動を行うと報酬を得ることが出来るという問題である．学習部分では提案システムを使用する．本実験では，実機を用いずにシミュレーションで行う．センサ等から入力される実データを仮想で作成し，提案手法に適用し，学習を行う．検証では 4 章で検証を行った 2 種類のシステムと本実験で実験を行う提案手法の結果を比較する．適切に状態の設定が行われているシステムと，不適切に状態の設定が行われているシステム，提案手法のそれぞれを比較し，提案手法を用いることで状態の設定についての問題点が解消されるかどうかを検証する．

### 6.1.2　実験設定

・実データについて

入力される実データ $D(k)$ を作成する．本章の実験では，４章で作成したデータと同じものを使用し，結果を考察する．このデータをエージェントがセンサから読み取った値として使用する．今回作成したデータは推移している値にノイズを乗せたものを想定して作成を行っている．４章で作成したデータは式(6.1)で作成している．

$$D(k+1) = D(k) \pm c \qquad (0 \le D(k) \le 1) \qquad (6.1)$$

k はステップ数を表し，作成するデータ数 M の分だけ計算する．またその時々の k をデータ番号と呼ぶ.
例：データ番号 1000 の時のデータの値は $D(1000)$ .

c はランダムノイズでノイズを表している.
± はデータ数 n 回ごとに確率 p で変化するものとする.
式(6.1)より，本実験では以下のようにパラメータの初期値を設定し，データを作成した.

表 6.1　データ作成に用いたパラメータの設定

| パラメータ | 記号 | 設定 |
|---|---|---|
| 実データの初期値 | D(1) | 0.5 |
| データ数 | M | 30000 |
| ランダムノイズ | c | 0<c<0.1 |
| 変化の間隔 | n | 10 |
| 変化の確率 | p | 0.1 |

初期値により作成したデータのデータ番号 1 から 10000 までを図 6.1 に示す.

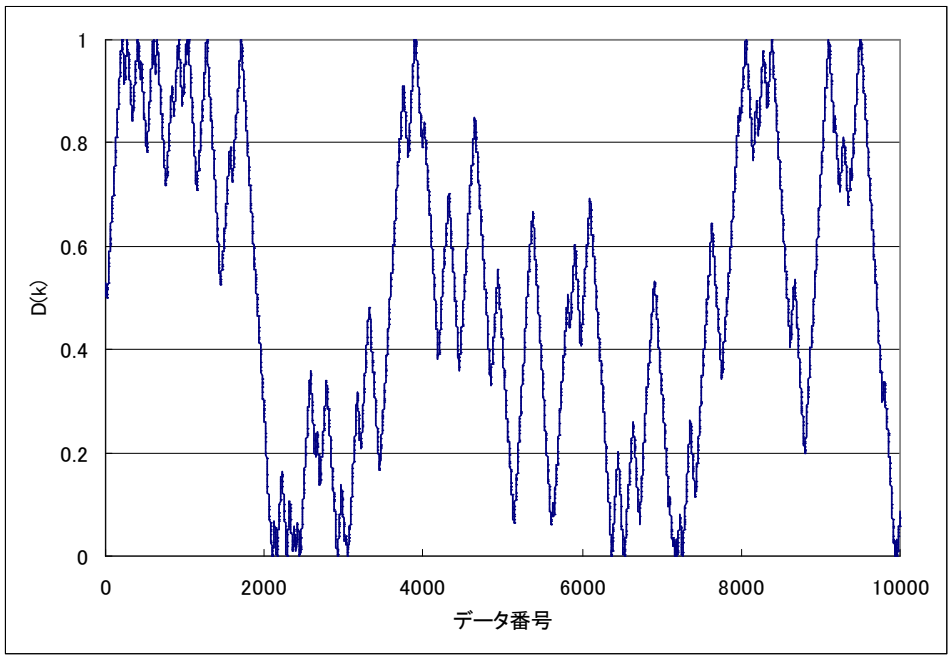



図 6.1　使用する実データ

・強化学習

（1）学習を行うタイミングと実データの扱い方

この実験では入力される実データで定義したデータ番号を 1 単位時間として扱う．強化学習を行う際にはデータ番号 1 つを 1 ステップとする．エージェントにはデータ番号ごとにデータ値 $D(k)$ が入力される．またエージェントの学習は1データごとに行うもい，学習の試行回数は作成したデータ数行う．以下に概要図を示す．

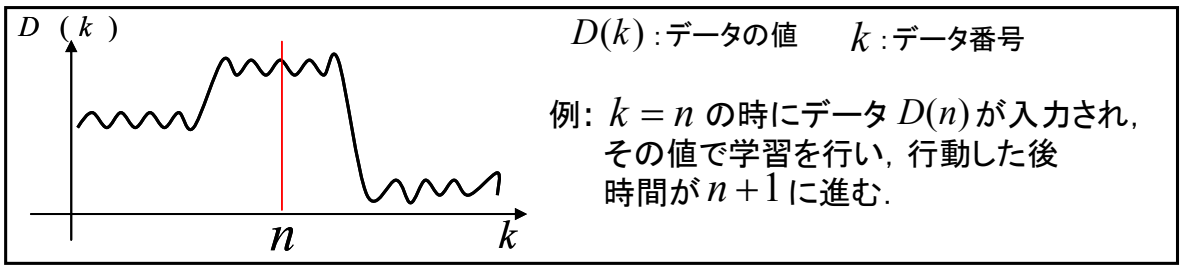



図 6.2　実データの扱い方の例

（2）エージェントの状態，行動について

状態：与えられたデータを提案手法に基づいて，状態を決定して学習を行う．

行動：エージェントには A，B，C の行動を行うことが出来る．いずれかの行動を行い，どの行動が入力された実データの時に最も報酬が得られるであろう行動か学習する．

（3）行動選択手法

本実験では ε -greedy 法を用いる．

今回の実験では以下のようにパラメータを設定した．

表 6.2　強化学習で使用するパラメータの設定

| パラメータ | 記号 | 設定 |
|---|---|---|
| 発生確率 | ε | 0.1 |
| 要素数の収束値 | x | 100 |
| シグモイド関数の収束速度 | α | 0.08 |
| シグモイド関数の編曲点 | β | 50 |
| 融合発生確率の期待値の重み | c | 1 |
| 融合発生確率の分散の重み | d | 1 |

３，報酬の設定

報酬の設定も使用する実データと同様に 4 章と同じ設定で実験を行う．報酬の設定は以下のようになっている．報酬の設定は実データの範囲を決定し，その範囲内で特定の行動を取ると報酬が得られるように設定している．

表 6.3　データの値と行動に関する報酬の設定

| データ:$D(k)$ ＼ 行動 | A | B | C |
|---|---|---|---|
| $0.65 \le D(k) \le 1$ | 1 | 0 | 0 |
| $0.35 < D(k) < 0.65$ | 0 | 1 | 0 |
| $0 \le D(k) \le 0.35$ | 0 | 0 | 1 |

・提案手法の分割の方法について

今回の実験では状態を２分割する方法で状態学習を行う．

### 6.1.3　実験結果

以下に実験結果を示す．まず図 6.3 から 6.5 までは，それぞれの行動に対する作成された状態を示す．この図は状態の分布を表しており，ある状態が実データのどこからどこまでを占めているかと，その状態を選択した時の報酬の期待値を表している．

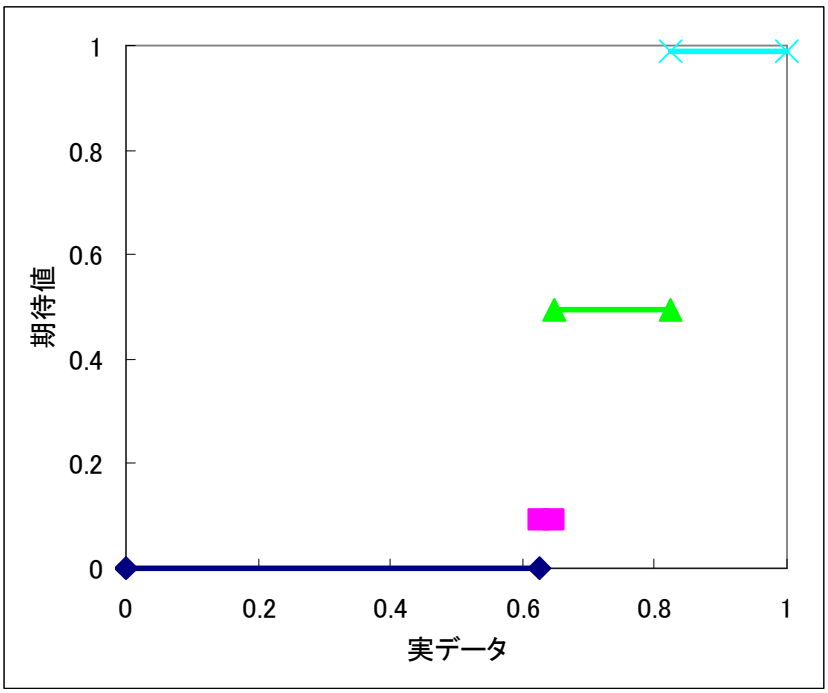



図 6.3：行動 A に対する作成された状態

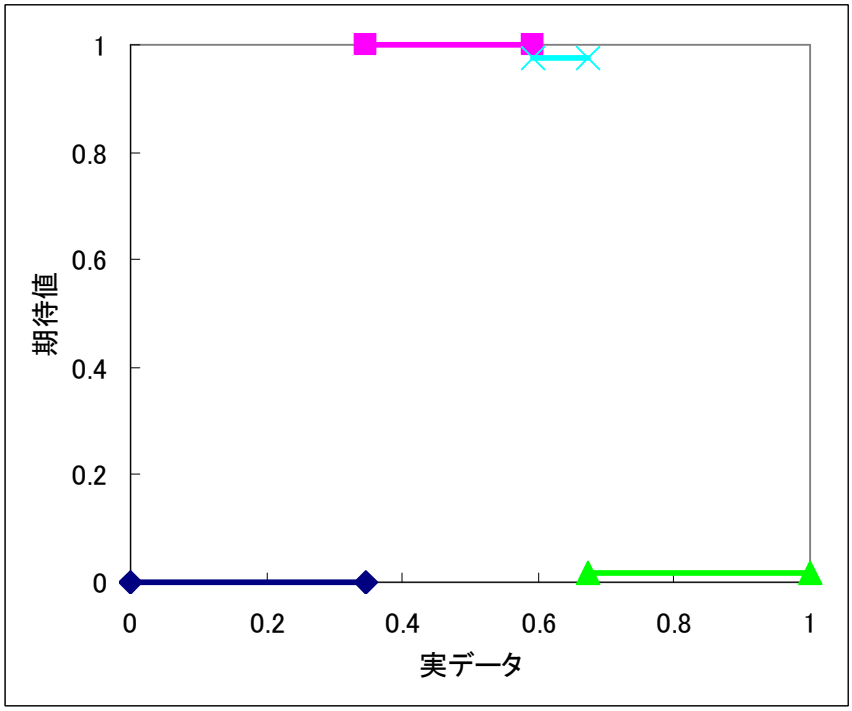



図 6.4：行動 B に対する作成された状態

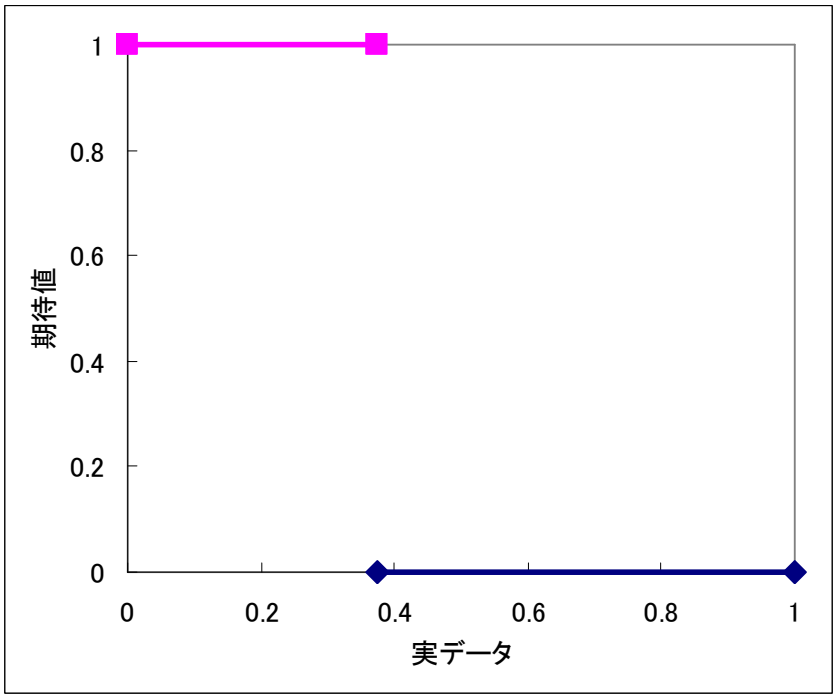



図 6.5：行動 C に対する作成された状態

次に 4 章で紹介した従来の手法と提案手法を報酬の入手率によって比較を行う．従来の手法では，適切な状態の設定を行って学習を行ったエージェント 1 と適切ではない状態の設定で学習を行ったエージェント 2 を用いて検証を行った．これらの報酬の入手率と本実験で行った提案手法の報酬入手率を比較する．なおランダム選択とはランダムに行動を行って報酬のどのくらい得たかを検証したものである．結果は以下のようになった．

表 6.4：提案手法と従来の手法の報酬の獲得率

| | 報酬入手率 |
|---|---|
| 提案手法 | 91.92 |
| エージェント 1 | 93.34 |
| エージェント 2 | 74.72 |
| ランダム選択 | 33.65 |

### 6.1.4　考察

行動ごとに状態の作成結果を考察する．行動 A では表 6.2 の報酬の設定を見ると，$0.65 \le D(k) \le 1$ の時に報酬を入手することが出来る．図 6.3 を見ると 0 から 0.6 付近まで報酬の期待値が 0 な状態が作成されている．そして 0.6 から 1 までは段階的に期待値が上昇しながらいくつかの状態が作成されている．

行動 B では表 6.2 の報酬の設定を見ると，$0.35 \le D(k) \le 0.65$ の時に報酬を入手することが出来る．図 6.4 を見ると報酬が入手できる範囲では期待値が限りなく 1 に近い状態が作成されている．また報酬が入手出来ない範囲では期待値が 0 の状態が作成されている．報酬を得ることが出来る状態と報酬を入手出来ない状態がはっきりと区別されて作成されている．

行動 C では表 6.2 の報酬の設定を見ると，$0 \le D(k) \le 0.35$ の時に報酬を入手することが出来る．図 6.5 を見ると報酬が入手できる範囲である 0 から 0.35 では期待値が 1 の状態が作成されている．またそれ以外の範囲は報酬の期待値が 0 の状態が作成されている．行動 C では報酬を得られる範囲と報酬を得られない範囲できれいに二分割されている．

これらのそれぞれの行動に対する作成された状態の結果を見ると，行動 A では状態の期待値が段階的に作成されており，明確に分類はなされていないがそれぞれの行動ごとに報酬が得られる範囲がほぼ確定している．行動 A に関してはより学習を行えば明確になっていくと推測できる．この結果から，報酬を与える基準が明確である問題に対しては行動ごとに適切に状態を設定できていることが示せた．

次に従来の手法と提案手法の報酬の入手率を比較する．表 6.3 の結果を比較すると提案手法の報酬の入手率は 91.92%となっている．これは適切な状態設定を行っているエージェント 1 とほぼ同じように報酬を入手している．今回の検証では行動選択手法で ε -greedy 法を使用している． ε -greedy 法では ε の確率でランダムな行動を選択する．今回の ε の設定は 0.1 と設定しており，10%の確率でランダムな行動を選択する．よって最も報酬を得られる行動がほぼ確定した場合でも，一定の確率でランダムな行動を選択してしまう．このことから報酬の入手率は最大でも約 90%になり，それ以上はランダム選択による確率的なものになる．今回の結果では提案手法とエージェント 1 の報酬の入手率はそれぞれ 90%を超えている．よって ε -greedy 法の性質上，結果には若干の差があるが，この 2 つの報酬の入手率にはほぼ同じだといえる．

また提案手法と適切ではない状態の設定で学習を行っているエージェント 2 の報酬の入手率を比較すると，報酬の入手率に差が出ていることがわかる．これはエージェント 1 との比較でも言えることだが，提案手法では状態の設定が限りなく適切に近く行われて学習しているためだと考えることが出来る．

以上のことから，提案手法では状態の設定をより適切に近い設定で行い，学習を行っていることが示せた．

## 6.2　提案システムの検証：ダムの放水問題

ここでは提案システムを状態の遷移が加わりより複雑な問題例に適用し，システムの有用性を検証する．この実験では状態の遷移がある問題に対して，提案システムを用いて状態学習が行われているのかを確認する．

### 6.2.1　ダムの放水問題

本実験ではダムの放水問題を取り上げる．水門の制御問題は古くから人の生活に関わっている問題である．ダムの放水問題ではエージェントの行動によって環境が変化する問題の一つである．

ダムの放水問題とは以下のような問題である．

・ダムには川や雨など外的要因から，水が流れてくる．
・ダムには水を蓄えられる量(最大貯蓄量)が決まっていて，水が最大貯蓄量を超えて入ってきた場合はダムが決壊する．
・ダムは一定量の放水を行う．適量の放水量だと良いがダムの水の貯蓄量によっては水を止めたり，大量に流さなければいけない．

図 6.6 にダムの放水問題の概要図を示す．

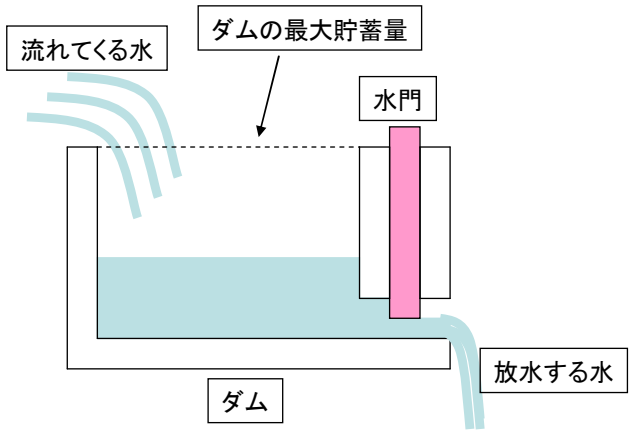



図 6.6　ダムの放水問題の概要図

本実験ではこの問題を提案手法の強化学習で適用し，学習を行う．

### 6.2.2　実験内容

ダムの放水問題に対して提案システムを適用する．本実験では，シミュレーションで行う．流れてくる水の量の推移を仮想で作成し，ダムに水を流す．エージェントはダムが決壊しないようにしながら，報酬が得られる適切な量の水を放水するように水門の開け方を制御する．エージェントが提案手法を用いてダムの貯水量に対して適切な行動が選択して報酬が得られるように状態の設定が出来るかどうかを検証する．

実験でのエージェントの動作の流れは以下のようになる．

①エージェントはダムの貯水量で水門の開け方を判断し，水門を調整して放水を行う．
②水門の開け方によってダムの貯水量が変化する．
③ダムに流れてくる水が入ってきて貯水量が変化する．
④この時のダムの貯水量と②で流した水の量によってエージェントは報酬を得る．

エージェントが学習を行うタイミングと学習対象は，④の時に学習を行う．また学習の対象となる実データの貯水量は①の時の貯水量を対象に学習を行う．

このようなタイミングで学習を行う理由はダムが決壊したときにマイナスの報酬を与える必要があるためである．

### 6.2.3　実験設定

・データの作成

流れてくる水の量として扱う実データ $D(k)$ を作成する．今回作成したデータはある程度規則性があり流れを表すデータにノイズを乗せたものを想定している．本実験のデータは以下の式で作成する．

$$D(k) = 0.5 \cdot a \cdot \sin(b \cdot k) + 0.5 \pm c \qquad (0 \le D(k) \le 1) \qquad (6.2)$$

k はステップ数を表し，作成するデータ数 M の分だけ計算する．またその時々の k をデータ番号と呼ぶ．
例：データ番号 1000 の時のデータの値は $D(1000)$ .

$a$ は式(6.2)の振幅を表している． $0 \le a \le 1$ の範囲で任意に設定する． $b$ は式(6.2)の周期を表している．また c はランダムノイズを表している．
また±は 1 ステップごとにランダムで変化するものとする．

式(6.1)より，本実験では以下のようにパラメータの初期値を設定し，データを作成した．

表 6.5　データ作成に用いたパラメータの設定

| パラメータ | 記号 | 設定 |
|---|---|---|
| 振幅 | $a$ | 1 |
| 周期 | $b$ | 0.06 |
| ランダムノイズ | c | 0<c<0.05 |
| データ数 | M | 30000 |

初期値によりデータ番号 0~1000 までの作成したデータを図 6.7 に示す．

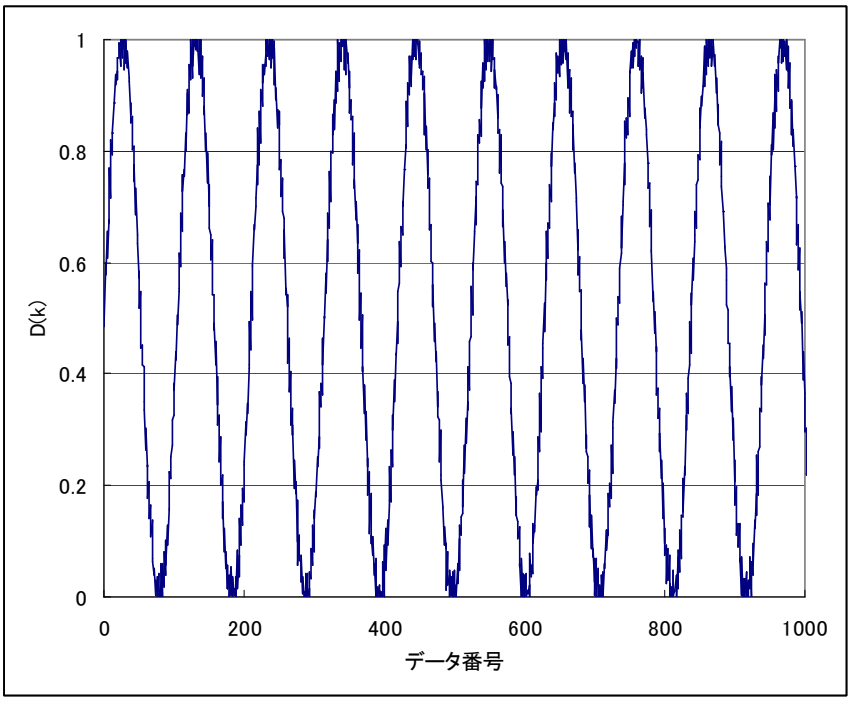



図 6.7　作成した実データ

・強化学習

（1）学習を行うタイミングと実データの扱い方

6.1.2 節で述べたのと同じように，実データを扱い，学習を行う．

（2）エージェントの状態，行動について

状態：ダムの貯水量を得ることが出来る実データとして，提案手法に基づいて，状態を作成して学習を行う．

行動：エージェントは水門の制御を行動とする．本実験では，「門を閉じる」「門を少し開く」「門を全開にする」の３つの行動を行えるものとする．

それぞれの行動によるダムの貯水量の変化を以下の表に示す.

表 6.6　行動による貯水量の変化

| 行動 | 制御 | 放水する水の量 |
|---|---|---|
| 行動０ | 門を閉じる | 0 |
| 行動１ | 門を少し開く | 0.5 |
| 行動２ | 門を全開にする | 1.0 |

また，その時のダムの貯水量が行動による減少量未満だった場合，放水する水の量はその時の貯水量になる．行動を行うとその行動によって放水された量だけダムの貯水量が減少する.

（３）行動選択手法

本実験では $\varepsilon$ -greedy 法を用いる.

以下の表は強化学習で用いるパラメータの初期値である.

表 6.7　強化学習で使用するパラメータの初期値設定

| パラメータ | 記号 | 初期値 |
|---|---|---|
| 発生確率 | $\varepsilon$ | 0.1 |
| 要素数の収束値 | x | 100 |
| シグモイド関数の収束速度 | $\alpha$ | 0.08 |
| シグモイド関数の編曲点 | $\beta$ | 50 |
| 融合発生確率の期待値の重み | c | 1 |
| 融合発生確率の分散の重み | d | 1 |
| 融合発生確率の期待値に関するパラメータ | $\tau_2$ | 0.04 |
| 融合発生確率の分散に関するパラメータ | $\tau_1$ | 0.04 |

３，ダムの放水問題の設定と報酬の設定

ダムには最大貯水量が設定されている．流れてくる水がダムに入ってくることでダムの貯水量が最大貯水量を上回った場合，ダムは決壊を起こす．決壊した場合はダムの貯水量は最大貯水量となり，決壊したという結果をエージェントに返す.

以下の表にダムの放水問題の設定を示す.

表 6.8　ダムの放水問題の設定

| パラメータ | 設定 |
|---|---|
| ダムの最大貯蓄量 | 2 |

エージェントが報酬を受け取るタイミングはエージェントが行動した後，ダムに水が流れてきてダムの貯水量が決まった時である.

報酬の設定を以下の表に示す．エージェントに報酬を与える時に条件が重複した場合，優先順位が高い報酬をエージェントに与える.

表 6.9　報酬の設定

| 優先順位 | 環境の変化 | 報酬 |
|---|---|---|
| 1 | ダムが決壊した時. | -1 |
| 2 | 放水する水の量が 0.4～0.5 の時. | 1 |
| 3 | その他の時. | 0 |

・提案手法の分割の方法について

今回の実験では状態を２分割する方法で状態学習を行う.

### 6.2.4　実験結果

まずどの程度学習成果が出ているか，30000 回試行を行った結果の獲得報酬の総和を表 6.10 に示す．また比較基準としてランダムに行動を選択したものと比較する.

表 6.10　獲得報酬の総和

| システム | 報酬の総和 |
|---|---|
| 提案手法 | 7105 |
| ランダム選択 | -5404 |

次に実データの値と期待値によるグループの構成の結果を行動ごとに示す.

行動 0 の時のグループの構成を図 6.8 に示す.

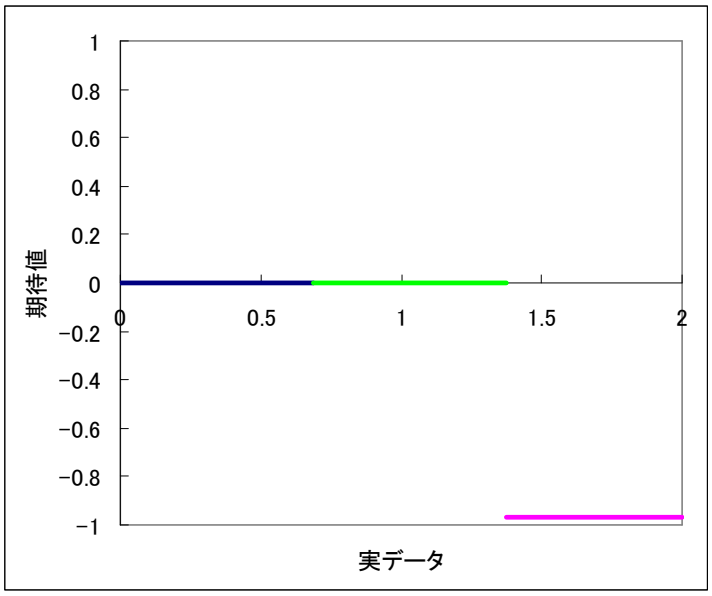



図 6.8：行動 0 の状態の構成

行動 1 の時のグループの構成を図 6.9 に示す.

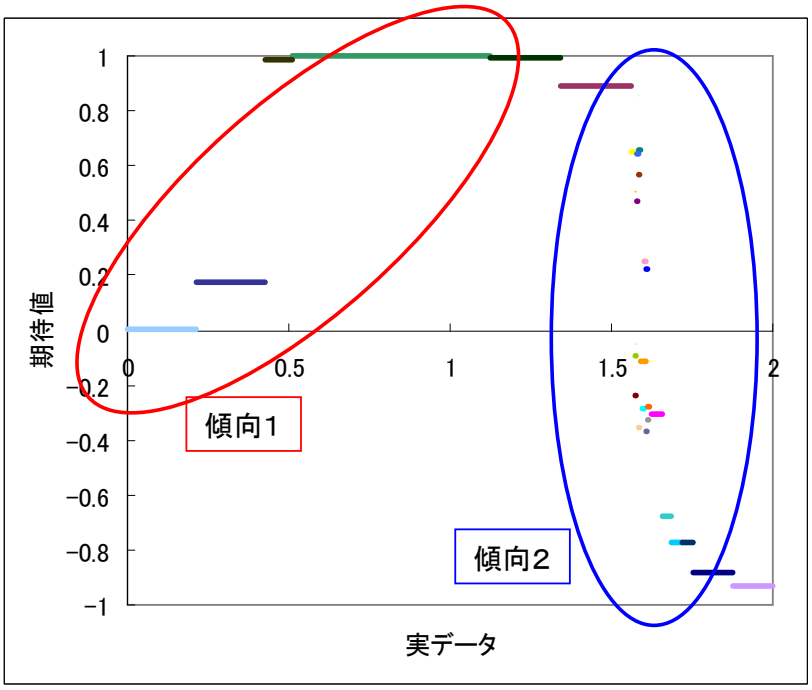



図 6.9：行動 1 の状態の構成

行動 2 の時のグループの構成を図 6.10 に示す.

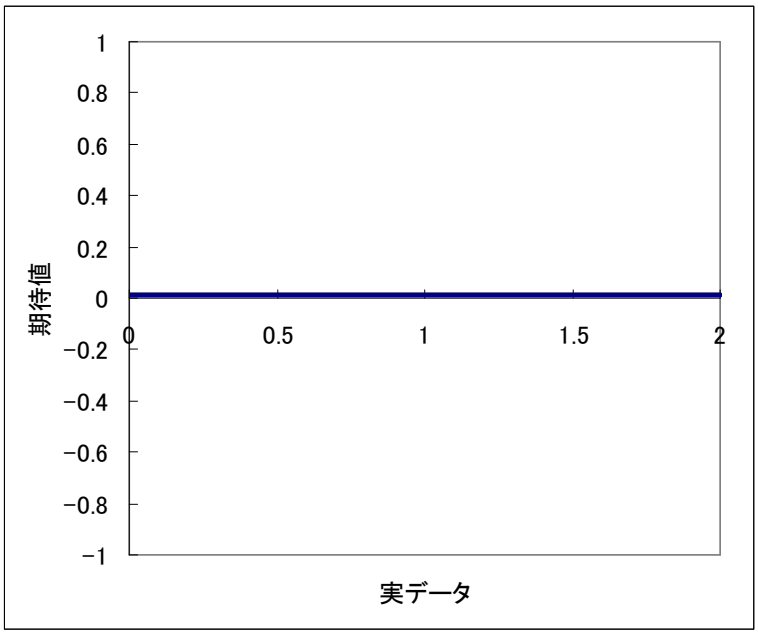



図 6.10：行動２の状態の構成

### 6.2.5 考察

表 6.10 を見ると，ランダム選択より報酬を入手している．これは報酬を得られる行動である行動１を選択しつつ，ダムの決壊を発生させないように貯水量が多くなれば行動２を選択するというように，実データに合わせて学習が行えているからであると考えられる．

図 6.8 では行動０の結果を示している．行動０は門を閉じて水を流さない行動である．行動０では，状態として使用するグループの数が３個作成された．行動０では報酬を得ることが出来ないため，決壊を起こす可能性が高い貯水量のときには期待値の低い状態が作成されている．決壊が起きない貯水量のときは報酬が得られないため０付近で期待値が収束していると考えられる．

図 6.9 では行動１の結果を示している．行動１では水の量を 0.5 流す．ダムに水が貯蓄されているのであれば必ず報酬が得られる行動である．行動 1 では，状態として使用するグループの数が３１個になった．図 6.9 を見ると大きく分けて二つの傾向に分類されていると推測できる．

図 6.9 の傾向１では，実データの値が小さい時は報酬が得られないため低い値になっており，報酬が得られるようになる水の量をダムが満たすようになると，報酬を得ることが出来るため期待値が高くなっている．

図 6.9 の傾向２では似たような実データの値の範囲でも期待値がバラバラになっている．

この理由は実データの時系列が関係していると考えられる．実データの値が大きい時は，まもなくダムが決壊する寸前で，この後に流れてくる水の量が増えると決壊する．しかし流れてくる水の量が減ってくるとある程度は安全である．よって実データの値が大きな時は水の流れに２つの可能性があることになる．このため次にどの程度流れてくる水の量が入ってくるかを予測できないと学習は難しい．今回は時系列を想定した学習法を採用していないので，流れてくる水の量が推測できずに，その時々で得られる報酬が変わってしまい，結果分割が大量に発生し，状態が安定しないものだと考えられる．

図 6.10 では行動２の結果を示している．行動２では水の量を 1.0 流す．流れてくる水の量の最大値が 1 なので行動２を取り続ければ必ず決壊はしない行動である．行動２では状態の数は１つになった．行動２で報酬を得るためには，報酬が得られる放水量と貯水量が同じときだけである．よって状態が一つになった理由は，どのような貯水量でも報酬が得られないと判断された結果であると考えられる．行動２は学習の序盤では最も選択される行動であるが，報酬が０の時がほとんどであるため期待値が０付近で収束する．最終的には実データの値によっては行動１の方が優先されるようになる．

よってこれらの結果から状態の設定を行って学習を行うことは出来るが，時系列が関係する問題では状態が確定されずに大量に発生してしまうといった問題が発生することがわかった．

# 第 7 章　結論

## 7.1　まとめ

本研究では，強化学習においてエージェントが自動で状態を獲得し，獲得した状態を用いて学習を行うシステムを提案した．

従来のシステムの検証において，状態の設定を適切に行って学習を行った場合と，適切ではない状態の設定で学習を行った場合を比較した．その結果，状態の設定が適切に行われないと学習効果に影響を及ぼすことを確認した．

このため提案手法では，エージェントが状態を自動的に獲得するように，センサのレンジを利用し，状態を設定する方法を用いた．センサのレンジの範囲内に状態となる領域を確保し，作成した状態を分割と融合を用いて，設定を変えることで状態設定をより適切に近づけるようにした．

実験では，従来のシステムで検証した問題と同じ環境で実験を行い，比較を行った．この結果，従来のシステムの状態の設定を適切に行った場合と同等の学習効果が得られることを確認した．

次により現実問題に近いダムの放水問題を学習対象とし，提案手法を適用して学習を行った．この結果，提案手法ではその瞬間ごとでは学習が行えるが，未来の予測など環境に時間が関わってくると学習を収束させることが難しいことが分かった．

よって提案手法は従来のシステムとの検証で行ったような時系列を用いない簡単な問題では，状態の設定を適切に行いながら学習をすることが出来るが，時系列が関係する環境においては学習が出来ないという結果になった．

## 7.2　今後の課題

まとめで述べたように時系列が関係するタスクでは，状態の決定がはっきりなされない．これは今回の提案手法で用いた強化学習が，時系列を考慮していないものを利用したためと考えられる．

また状態の決定が出来ないタスクには，報酬が確率的に与えられるタスクの場合も同様の理由で考えられる．報酬の入手する確率が 50%の状態があるとすると，提案手法では無限に分割を続けてしまうという問題がある．状態によって報酬が確定しない環境に対してどのように対応するかを考える必要がある．

今回の提案手法では行動ごとに状態の作成を行っている．本研究では状態の構成に着目

し手法を提案したが，実機に適用する場合行動が大量にある場合が考えられる．この場合行動の数だけ状態の数も増えてしまうので，状態を作成する際には行動についても間上げる必要がある．

# 謝辞

本論分を結ぶにあたり，日頃より懇切なるご指導を賜りました倉重健太郎先生に深く感謝の意を表します．また，ご指導，ご助言を頂いた畑中雅彦先生，本田泰先生，須藤秀紹先生，渡辺修先生に感謝の意を表します．そして，論文の査読や助言をして頂いた院生の尾上由希子さん，池田憲弘さん，木島康隆さん，発表スライドで助言を頂いた同輩の黒滝麗子さんに感謝いたします．

参考文献

[1] 小林 宏，表情豊かな顔ロボットの開発と受付システムの実現，日本ロボット学会誌，Vol.24 No.6，pp708~711，2006

[2] 柴田 崇徳，メンタルロボット・パロとロボット・セラピーの展開，日本ロボット学会誌，Vol.24 No.3，pp319~322，2006

[3] 橋爪 誠，手術ロボットの現状と将来，日本ロボット学会誌，Vol.22 No.4，pp423~425，2004

[4] 長谷川 勉，環境プラットホーム「ロボットタウン」，日本ロボット学会誌，Vol.26 No.5，pp411~414，2008

[5] 大場 光太郎，大原 賢一，ユビキタス・ロボティクス，日本ロボット学会誌，Vol.25 No.4，pp505~508，2007

[6] Richard S. Sutton and Andrew G.Barto，強化学習，森北出版株式会社

[7] 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，October 1999

[8] 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，pp118~124，1999

[9] 二本 真，高次元連続状態空間での強化学習におけるクリティカル状態を利用した適応的関数近時，北陸先端科学技術大学院大学，修士論文，2007