獲得免疫系に基づいた強化学習による制御器設計に関する研究

細川　嵩

電気通信大学大学院電気通信学研究科

博士（工学）の学位申請論文

2015年 3月

獲得免疫系に基づいた強化学習による制御器設計に関する研究

博士論文審査委員会

主査　樋口 幸治　准教授

委員　中野 和司　理事・副学長

委員　桐本 哲郎　教授

委員　新　　誠一　教授

委員　内田 雅文　准教授

2015年 3月

A Controller Design Using Adaptive Immune System Based Reinforcement Learning

Shu HOSOKAWA

Abstract

In recent years, many autonomous mobile robots have been used for various purposes. The industrial robot controller has designed by expert engineers. Expert engineers can be adjusted to suit different situations and objects of the robot controller. In addition to the industrial robot, many home use robots have been produced. For example, these robots have been produced for home cleaning, nursing, security guard, etc. However, these cannot use the methods for controlling the industrial robots. Because, home robot users who are not expert engineers are not able to adjust the robot controller. As a result, a simplified method is required for designing the controller.

The machine learning methods have been focused on characteristics that robot's adaptive behavior can be gotten from action results. Reinforcement learning is a type of machine learning methods, which does not require detailed teaching signals by a human. This method is learned based on the result of trial and error. It is not necessary to give detailed prior information on the controller from this feature. But, the learning process needs a huge amount of time for the trial and error. If learning methods have been applied to an actual robot, that of fast learning convergence speed is more important than a property which is able to get the optimal policy. In addition, the reinforcement learning also has a problem in parameter selection and the curse of dimensionality. On the other hand, the mechanism of evolution and ecological mechanism possessed by the organisms, has been modeled in an engineering sense, and various based on the modeling approach attempts have been actively carried out to the areas such as learning and optimal solution search. Especially, among the modeling approaches, the immune system attracts much attentions. The human immunity-based reinforcement learning method is built on the basis of the adaptive immune system of a human. This learning method has

a faster learning speed than famous methods (such as Q-learning, ProfitSharing, etc), to model-free methods. However, there are also some disadvantages as well as other methods.

At first, since this approach needs the assumption that it works well in a discrete state space environment, it is apt to fail to learn, or to show a decrease in learning convergence speed, when applied to a continuous state space environment. Even if it learns successfully, it requires a lot of computer memory. For a continuous state space environment, there are some methods required probabilistic models and/or the number of divisions to be set in advance according to the environmental dimensions. However, it is difficult to set appropriate values before learning. This paper aims to improve our previous immunity-based reinforcement learning method in order to extend it to a continuous state space. Previous learning methods have been used to select an action only by using the information that has matched sensor observations and memorized states. We take the fitness of memorized states and sensor observations into account, and make use of the fitness and the reward gained from the environment for action selection. The validity of the proposed method is demonstrated through simulations. The improved method is able to perform learning even in a continuous state space environment

Secondly, when applying model-free methods to stabilizing control tasks, we cannot acquire a policy to achieve the goals. The model-based methods can acquire the policy of the stabilizing control by giving a negative reward at a change from a stable state to an unstable state. Since the model-free learning method cannot deal with negative reward values, the reward value has to take positive values. In this case, there is a great risk of learning an undesirable behavior of changing from a stable state to an unstable state according to reward values. We improve a reward allocation method for the stabilizing control tasks. In the stabilizing control tasks, we use the Semi-Markov decision process (SMDP) as an environment model. The validity of the method is demonstrated through simulation for stabilizing control of an inverted pendulum. We show the conditions of reward allocation for the stabilizing control tasks, and introduce an example of reward allocate function for it. Since the reward is allocated only from the duration time of action, we do not need to change the reward value according to each environment.

獲得免疫系に基づいた強化学習による制御器設計に関する研究

細川　嵩

概要

生産工程などあらかじめ作業内容や環境が固定された状況で用いられる産業用ロボットに対し，最近では人間の代わりに日常環境で用いられる家の中の掃除を行う家庭用ロボットや，介護用ロボット，警備を行うロボットなどが数多く登場している．産業用ロボットなどでは目標や動作環境が固定されているので，通常の最適制御などにより最適な行動を設定することができる．しかし，今後導入が見込まれる家庭用のロボットは運用先によって目標とする状態や目標達成に必要な行動セット（政策）が異なるため，それぞれの運用先に合わせた適切な政策を設定しなければならないが，われわれが多種多様なロボットに対して，また考えうる環境条件すべてを考慮して適切な政策を設定するのは大きな負荷となる．

本研究では，ロボットコントローラの容易な構築を実現するために強化学習による手法を取り扱う．強化学習はロボットの内部状態や詳細な環境情報を与えなくとも，ロボット自身による試行錯誤の結果より自動的に適切なコントローラを学習することが可能である．一般的に目標達成に最適な政策を得るためには膨大な学習時間を必要とするため，特にロボットのコントローラへの応用では最適な政策を得ることよりも学習時間の短縮が重要となる．しかし，強化学習では“ 次元の呪い ”と呼ばれる環境認識に関する問題や，報酬や内部パラメータの初期値によっては学習がなかなか進まない，といった問題がある．

一方，生物の持つ生態機構や進化の仕組みなどを工学モデル化し，最適解探索や学習などの分野に応用する試みが盛んに行われている．その一つに免疫機構の振る舞いに着目し，その働きをモデル化した免疫型強化学習がある．免疫型強化学習法は従来の強化学習法と比べ，特定環境において準最適解を高速な学習収束速度で得ることができる．しかし，免疫型強化学習は動作環境が連続値で表現される場合では従来の強化学習法と同じく次元の呪いによる影響を受けてしまう．これは免疫型強化学習法のアルゴリズムにおいて環境情報を離散値へ変換する必要があるためである．この変換方式として動作環境の連続値表現を一定の間隔で区

切ることによって離散値表現に置き換えを行うタイルコーディングが多く用いられている．この際，状態を区切る間隔によって学習の収束速度および得られる解の質のトレードオフが発生するが，多くの場合において事前に適切な間隔を知ることはできない上，学習途中で離散化の間隔を変更することもできない．このため，事前に適切な離散化間隔を設定する必要のあるタイルコーディングによらない状態表現方法が必要となる．さらに，制御工学で重要な安定状態を維持するといった課題においても十分な解を得ることができない．免疫型強化学習や Profit Sharing をはじめとした一部の強化学習法では，タスクの達成のための最適解を得るのではなく，実用的な解を短時間で得ることを目標に主眼をおいてアルゴリズムが構築されているからである．またその制約条件として，報酬は正の値を使用しなければならないこともあげられる．安定化制御問題では報酬を与える明確なタイミングとして安定状態から不安定状態へ遷移した時が考えられる．この場合においては望ましくない状態へ遷移したため罰報酬を与える必要があるが，これまでの手法では正しく罰を取り扱うことができない．このため，安定化制御を考慮した報酬の処理法が必要となる．

本研究ではこれらの問題を解決する手法を提案し，実ロボットへ適用できる学習によるコントローラの構築法を確立することが目的である．

連続値環境を前提とした免疫型強化学習法の拡張方法を提案する．拡張したアルゴリズムが従来の離散型免疫型強化学習法の更新方式と等価であることを示し，さらに連続値環境に用いる際に利点となる状態の取り扱い方法について述べる．この提案手法を倒立振子の振り上げ制御シミュレーション例などに適用し，従来の代表的な強化学習法と比較をおこない，その有効性を示す．

従来の報酬割り当て関数が安定化制御問題へ適用できないことを示し，安定化制御問題へ適用する際の条件の検討を行う．得られた条件から Profit Sharing 及び免疫型強化学習において有効な報酬割り当て関数の一例を提案する．提案する報酬関数を用いて倒立振子の安定化制御および RoboCup サッカーシミュレーションリーグのサブ問題である Keepaway のシミュレーションに適用し，その有効性を示す．

目 次

図 目 次

表 目 次

第1章　緒論

1.1　知能ロボットとロボカップ

図 1.1: RoboCup Japan Open 2012 サッカー小型リーグ

生産工程などあらかじめ作業内容や環境が固定された状況で用いられる産業用ロボットのみならず，人間の代わりに日常環境で用いられる家の中の掃除を行う家庭用ロボットや，介護用ロボット，警備を行うロボットなどが数多く発表，市販化されている [2][3][4][5][6]．産業用ロボットなどでは目標や動作環境が固定されているので，制御理論（たとえば，[7][8]）などにより目標達成のための最適な行動を設定することができる [9]．また，ロボットの使用者は専門家であり，十分な知識を持っているため環境に応じた調整などの作業をすることができる．しかし，家庭用ロボットは運用先によって目標とする状態や目標到達に必要な行動セット (政策) が異なるため，それぞれの運用先に合わせた適切な政策を設定しなければならない．また家庭用ロボットでは使用者が制御やロボティクス分野などの専門家ではない場合が圧倒的多数であり，多種多様なロボットに対して，また考えうる環境条件を考慮した適切な政策を設定するのは困難である．

このような日常環境で使用されるロボット技術の開発のために，国際ロボット競技大会のロボカップ (RoboCup)[10] が開催されている (図 1.1)．この競技大会は “西暦 2050 年サッカーの世界チャンピオンチームに勝てる自律型ロボットのチームを作る” という最終目標（ランドマーク）を掲げ，ロボット工学や人工知能などの基礎技術の研究促進を目的としている．また，RoboCup はサッカー競技だけではなく，災害現場で使用するレスキューロボットや家庭内ロボットなど他分野の競技も行われており，これらの技術を応用することを目指したランドマークプロジェクトでもある．これらの目標を実現するためにはロボットのハードウエアの設計技術や制御技術，センシング技術，環境識別技術，また複数台ロボットが協調して動作する場合などではフォーメーションの形成法など様々な課題が存在する [11][12][13][14][15][16]．また，生産現場のロボットと異なりサッカーゲームやレスキューロボットなどは時間とともに刻々と周囲の環境が変化している中で動作させる必要がある．このために，リアルタイムでの行動の意志決定や制御を行う必要がある．しかしこれらのロボットの意志決定方法は多くの場合，動作前に一意に設定することは困難である．たとえばサッカーロボットの例で考えると，最終的な目標は敵チームの得点をできるだけ抑え，自チームの得点をあげることである．この目標を達成するためには自分や味方，敵の位置など情報から適切な行動 (たとえばパスやドリブル，シュートなど) を選択する必要がある．しかし，これには敵ロボットを含めたシステム全体の情報を知っている必要があるが，事前に得られる情報はごくわずかである．このためロボット自身が環境からの情報を能動的に取得し，自ら判断して自律的に動作を行えるようにする知能化技術が重要となる．

1.2 知能化技術と学習

1.2.1 機械学習

ロボット知能化のための研究分野として機械学習法がある．機械学習法はシステムの物理モデルなどから制御器を設計する手法とは異なり，ロボットの観測情報やとった行動などを元に制御器を構築する手法である．この機械学習に重要となるのがセンサー入力に対してどのように出力を決定するかということである．学習機構についての研究は生物の持つ生態機構や進化の仕組みなどを工学モデル化し，最適解探索や学習などの分野に応用する試みが最近盛んに行われている．代表的

表 1.1: 代表的な機械学習法

手法名	特徴	欠点
遺伝的アルゴリズム	高速な最適解探索アルゴリズム	解の逐次的な評価が必要
ニューラルネットワーク	複雑な非線形要素を持つ問題においても適用可能	教師データが必要
強化学習	試行錯誤の結果より自律的に学習を行う	次元の呪いによる影響を受ける

なものに，遺伝的アルゴリズム（Genetic Algorithm: GA）[17] やニューラルネットワーク [18]（Neural Network: NN）がある (表 1.1)．遺伝的アルゴリズムは，生命の進化において重要な役割を持つ遺伝子の世代交代時の振る舞いに着目した手法であり，解析的な手法よって最適解を求めることが困難な問題において，高速に準最適解を獲得できる手法として知られている．しかし，最適解を探索していく際に現在得られた解に対する評価が必要なため，報酬や罰則といった曖昧な評価値しか得られない場合には使用が困難である．また，ニューラルネットワークは脳細胞の情報記憶や伝送手法をモデルとした記憶機構であり，ある入力とそれに対する出力の関係を記憶することができる．これは入出力関係が単純な数式で表せない強い非線形性を持つ場合などに，その入出力関係を同定するのに有効である．しかし，一般的にニューラルネットワークの学習には教師データが必要となるため，未知環境や複雑環境における学習には適さない．ロボットがとるべき行動を教師データとして利用して学習を行う手法は教師あり学習と呼ばれる．このため，あらかじめタスクを達成するための有効な行動がわかっている必要がある．しかし，環境情報が事前にわからなかったり，複雑な環境などの大局的な目標は立てられるもののそこに至るまでの具体的な行動例や時系列にそった実行すべき行動セットを事前に求めることが困難な場合では教師データを用意することができない．そのような場合では，ロボットが試行錯誤的に行動を実行し環境から得られた結果をもとに自己の方策を改善していく，という教師なし学習の方式が望まれる．教師なし学習の代表例としてはクラスタリングや主成分分析，強化学習などがある．強化学習は，環境から得られる報酬を元に学習を行う手法であり，多足歩行ロボットの歩様獲得や全方向移動ロボットの制御などロボット制御に関して多くの研究が行われている [19][20]．しかし，強化学習には次元の呪いとも呼ばれる環境認識に関する問題があるほか，報酬や罰則を得るまでに多くの行

動選択が必要な場合に学習がなかなか進まないという問題がある．

1.2.2 強化学習法

強化学習 [21][22] は環境から与えられる報酬を元に目標を達成する政策を学習する手法であり，単位時間当たりに得る報酬が最大化することが目標となる．強化学習の大きな特徴として遅延報酬を取り扱うことができることがあげられる．ニューラルネットやファジィ理論 [23][24] などを基にした学習方式 [25] では各行動に対しての評価（報酬）を逐次的に与える必要があるが，強化学習は行動を行った時点で報酬が与えられなくとも，後に報酬を得た時点からさかのぼって評価を行うことができる．強化学習ではモデルベース型とモデルフリー型の手法に大別 [21] することができ，その特徴には大きな差違がある．それぞれの学習型における強化学習法の代表例とその特徴を述べる．

a) モデルベース型 モデルベース型の手法では学習を行う全体の状態からタスク達成のための各状態における行動の評価を行う．この各状態における行動評価値のことを一般に Q 値と呼ぶ．Q 学習はモデルベース型の代表的な学習手法である [26]．この学習手法はマルコフ決定過程 (MDP) 環境において無限回の試行を行った際に最適解が得られることが知られている手法である．Q 値の更新式は次式で示される．

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(R + \gamma \sum_{a_i \in A} Q(s', a_i) - Q(s,a) \right) \tag{1.1}$$

ここで R は環境から受け取った報酬値，$\alpha (0 < \alpha < 1)$ は学習率，$\gamma (0 \leq \gamma < 1)$ は割引率，s' は行動を実行して遷移後の状態である．割引率は将来受け取る報酬値をどれくらい重視するかを調整するパラメータであり，1 に近い値を設定すると将来全体に渡って得る報酬の合計を重視し，0 に近づけることにより直近に得られる報酬を重視するように学習が行われる．モデルベース型の強化学習法は得られる解の質が高いことから多くの分野への適用検討がされているが [27][28]，学習解の収束性においては次に述べるモデルフリー型の手法に劣る．モデルベース型のほかの強化学習法には Sarsa [29] などがある．

b) モデルフリー型 モデルフリー型の手法では，報酬を得るまでのエピソード中で経験した状態ー行動のみの学習を行う．Profit Sharing は Q 値の更新時に他の

状態のQ値を使用ぜず，与えられた報酬のみによって各状態でのQ値の更新を行う，モデルフリー型の強化学習手法の1つである [30].

Q学習などのモデルベース型学習システムは，与えられた報酬と他の状態s'のQ値を基に状態sのQ値の更新を行う．この方式は最適もしくはそれに近い解を得ることができるが，学習に多くの時間を必要としてしまう．モデルフリー型の学習システムの特徴は値の更新に他状態のQ値を用いないので，選択された頻度の高い行動についての学習が高速に行われる．しかし，最適解を得られる保証はない.

Profit SharingによるQ値更新の基本方針は，各行動に対して割り当てられる報酬関数$r(t)$にQ値を収束させることである．これを満たしたときにタスクに対して有効な解を得ることができる．初期状態s_0からの行動実行回数(以後ステップ数と記述する)をt，そのときの状態をs_t，選択した行動をa_t，s_tに対するa_tのQ値を$Q(s_t, a_t)$とし，具体的なQ値の更新法を説明する．Profit Sharingでは選択した行動のQ値から行動を行うためのセリ値$C_{bid}Q(s_t, a_t)$を支払い，選択した行動を実行する(C_{bid}はセリ値を計算するための係数である)．このセリ値の支払いは，タスクから報酬を受け取った際に各行動の報酬の享受と同時に一括して行われる．タスクから得た報酬を，報酬関数$r(t)$に従い，選択した行動のQ値に加える．支払ったセリ値に対して大きな報酬を得た場合Q値が増加し，反対に支払ったセリ値よりも受け取る報酬値が少ない場合はQ値が減少する．これを繰り返すことにより，最終的には$Q(s_t, a_t)$を報酬関数$r(t)$に収束をさせることができる．Q値の更新式は((1.2)式)で表される．なお，報酬を受け取るまでに要したステップ数を$step$とする.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + C_{bid}[r(t) - Q(s_t, a_t)] \tag{1.2}$$

$$\text{where} \quad t = 0, \cdots, step - 1;$$

Profit Sharingが提案されたときには，報酬関数は受け取った報酬を行った行動すべてに均一に与える関数を用いられていたが，後に種々の問題を解決すべく合理性定理に基づいた報酬関数の設計法が宮崎らに [1] よって提案されている．合理性定理とは目標達成に無効な行動を抑制する条件をまとめたものである。詳細な合理性定理や報酬関数の設計条件などは付録Aを参照されたい．報酬をR_0，減少率を$D(< 1)$とすると，報酬関数は(1.3)式として表される.

$$r(t) = R_0 \times (D)^{step-t} \tag{1.3}$$

モデルフリー型の強化学習法は高速な学習収束性を有しており，実ロボット環境への適用が期待される．その他のモデルフリー型の強化学習手法としてモンテカルロ法などがある．

1.3 研究の目的

多くの機械学習手法はニューラルネットワークや遺伝的アルゴリズムなどのように生物が備えていると働きを工学モデル化している．一方で免疫機構の振る舞いに着目した免疫型システム [31][32] もいくつかの手法が提案されている．免疫系は，自己・非自己の認識，クローン選択，ネガティブ（ポジティブ）選択，学習・記憶などの機能を持つことが知られており，これらの機能を工学モデル化することにより，これまで解決の困難であった種々の問題に対する新しい解決策を与えることが期待されている．特に免疫系は，例え未知の病原菌であっても，多くの場合対処することが可能であるという特徴を持っている．そのため，この免疫系の特徴をうまく工学的にモデル化することにより，ロボットの未知環境や複雑環境への適応という課題に対し有効な解決策を与えることが期待される．しかし，免疫系の工学応用に関する研究では [33][34][35] などがあるが比較的新しい研究分野であり，遺伝的アルゴリズムやニューラルネットワークのように，確立された具体的な数式モデルやアルゴリズムはまだ存在しない．そこで本研究では，この免疫系を基にした免疫型強化学習法 [36] を中心として，自律ロボットための強化学習による制御器設計を行う．

強化学習を自律ロボット環境に適用する際の1つめの問題として自律自律ロボットの動作環境は多くの場合において連続値環境であるが，多くの強化学習に関する研究では離散環境についての研究が主であった．離散環境を前提とした強化学習法を連続値環境へ適用した場合，状態の離散化度合いが学習の収束や得られる政策などの性能に大きく影響を及ぼす．離散化度合いを細かくすることによりある程度連続値環境表現に近づけることが可能であるが，ノイズの影響を受けやすくまた学習の収束に多くの時間を必要とする．反対に離散化度合いを荒くすることで前述の問題に対する影響は低減されるが，環境を正しく認識することができなくなる恐れがあり学習が不可能となる．これらの問題を解決するために複数の

学習器を用いそれらの線形和を取ることで離散化の影響を低減する手法 [37] や行動と環境認識を分けた Actor-Critic を用いた手法 [38] 等がある．しかし，これらの手法は依然として離散化度合いの決定問題や確率モデルの事前設定が必要となる．また，これの離散化度合いなどのパラメータを誤って設定した場合は，今までの学習結果を初期化して再度学習を実行しなければならない．一方，人体の備える獲得免疫機構では病原体に対して特定の情報のみで認識を行うのではなく，さまざまな要素が複合的に作用して病原体に対する対処を行う．このため，獲得免疫系の抗原認識作用を再モデリングし，それを免疫型強化学習器へ適用することで連続値環境用の学習手法を構築する．

２つ目の問題は免疫型強化学習法を初めとするモデルフリー型の強化学習方式では報酬を得るまでの時間を最短化するような問題において準最適解を短時間で獲得することができる．モデルフリー型の強化学習法ではタスクを達成について報酬を与えることを前提に学習方式が最適化されてきたことによる．その一方で安定化制御などの一定状態内を維持する様な問題においては望まない結果を得ることがあった．安定化制御問題では報酬を与える場合が所望する状態からそうではない状態に遷移した場合であり，これは多くの場合において罰報酬として取り扱われる．一般的に罰報酬は負の値として与えられ，モデルベース型の強化学習手法によって安定化状態を維持する手法が提案されている [27][39][40]．一方，モデルフリー型の強化学習法では負の値を取り扱うことができない [41]．このため，正の報酬値によって罰報酬を表現しなければならないが，従来の報酬関数では安定化状態を崩すように学習が行われてしまう．以上から安定化制御のための報酬間数を設計することにより，モデルフリー型の強化学習方式の利点を生かした学習器の構築を行う．

1.4　本研究の構成

本研究の構成は以下の通りである．第 2 章では基礎礎礎となる生物が備えている免疫系とその働きをモデル化した免疫型強化学習器について述べる．免疫系は複雑な動作を，複数細胞の連携により実現することで生物の生体機能を保護している．本研究では免疫系のうち病原によって動作を変え，その働きを記憶する獲得免疫系についてを述べ，その働きをモデル化した強化学習法について説明をする．

第 3 章では，免疫型強化学習器の連続値環境への適用法について述べる．本研

究では獲得免疫系の細胞間の情報伝達法を見直すことによりこれらの問題点を解決した強化学習法を提案する．提案手法が従来の免疫型強化学習器と同等の更新作用を有し，かつ連続値環境に適用した場合の利点を説明する．学習器を倒立振子の振り上げ制御の例に適用し性能の評価を行う．

第 4 章では安定化制御問題における強化学習器への報酬関数の設計法について述べる．従来のモデルフリー型の学習器にて用いる報酬関数が安定化制御問題へ適用することができないことを示し，安定化制御問題に適する報酬関数の条件について述べる．求めた報酬関数の条件より具体的な報酬関数の一例を示し，Profit Sharing および免疫型強化学習器に適用をする．倒立振子の安定化・Keepaway タスクなどの例にその手法を適用し性能の評価を行う．

第 5 章は全体のまとめである．研究の総括と今後の課題について述べる．

第2章　獲得免疫系を参考にした強化学習法

2.1　はじめに

モデルフリー型の強化学習手法は与えられたタスクに対する最適解を得ることはできないが，短時間で解を得られる強化学習手法である．実際のロボットなどへ学習機構の実装を目指した場合では解を得るための試行が可能な限り短い事が求められる．人間に備わっている免疫機構では未知の病原体についても対処が可能であり，学習機能により 1 度罹患した病原体には短時間で対処することができる．この働きを参考にした強化学習法が免疫型強化学習であり，学習器のアルゴリズムとその特徴について説明する．

2.2　人工免疫系

本節では人体の免疫作用について述べる．はじめに，免疫系の全体像について概説する．免疫系は，クローン選択やネガティブ（ポジティブ）選択，免疫ネットワーク [32] など種々の興味深い特徴を有しており，その特徴に基づいた工学システムに関する研究 [42][31][43] も多く行われている．ここでは，免疫系のうち学習アルゴリズムに参考としている獲得免疫系の免疫作用について中心に説明する．次章以降の研究で使用している，T 細胞と B 細胞，抗体の連携を中心とする獲得免疫系の病原体駆除のメカニズムについて説明する．

2.2.1　免疫系の概要

人体では，循環器系や神経系など多くのシステムが働いており，生命を維持するために機能している．この中で免疫系は，体外から侵入した病原体や毒素，体内の細胞が変化したガン細胞など生体を脅かす存在を体内から排除し，恒常性を維持するために働いている [44].

表 2.1: 免疫系の分類

免疫タイプ	特徴	抗原への挙動
自然免疫	非特異防御	反応が素早い
獲得免疫	特異防御	学習・記憶能力がある

免疫系を大きく分けると非特異的に防御を行う自然免疫系と特異的に防御を行う獲得免疫系に大別することができる．まず自然免疫系が，体内に侵入してきた病原体や毒素などに対し防御する．自然免疫系は，人体の粘膜などによって病原の侵入を阻み，白血球などは侵入した異物を貪食することなどによって一律に排除もしくは中和しようとするものである．この反応は非特異的であるため，どのような病原に対しても一様に素早く機能し，人間が生まれてから備えられているため自然免疫とも呼ばれる．この自然免疫系を通り抜けて人体に侵入してきた病原体や毒素，人体内の細胞がガン細胞などに変異してしまったものについては獲得免疫系によって中和・排除される獲得免疫系では病原体や細胞についてその細胞の種類を区別や認識，対応する免疫細胞にそれぞれ役割が分かれている．獲得免疫系は特異的反応をとるため病原体に対する情報が必要なため初動は自然免疫系よりも遅いが，病原体のタイプに応じてその駆除に効果的な細胞・抗体を集中的に投入して対応するため，病原体の駆除能力は高い．また，記憶・学習機能があるため，同じ病原体が再び体内に侵入してきた際には，1 度目よりも素早く効果的に機能することができる．この 2 種類の免疫系の連携により生体は守られている．免疫型強化学習器では，侵入した病原体に対して特異的反応により効率的に対処ができる獲得免疫系に注目し，行動選択および学習・記憶機構の構築を行っている．次節において獲得免疫系の詳細な働きについて述べる．

2.2.2 獲得免疫系

獲得免疫系は複数の役割の異なる細胞が連携しながら種々の病原体に対処している．大まかな働きは病原体（抗原）認識，T 細胞の反応活性化，B 細胞の活性化と抗体の産生という流れで反応が起こる．また獲得免疫系の特徴として，一度体内に侵入してきた病原体について学習・記憶し，再び同じタイプの病原体が体内に侵入してきた場合にこれに対し素早く反応し，病気の進行を早くに食い止めることができる．獲得免疫系の構成は，図 2.1 に示すとおりである．

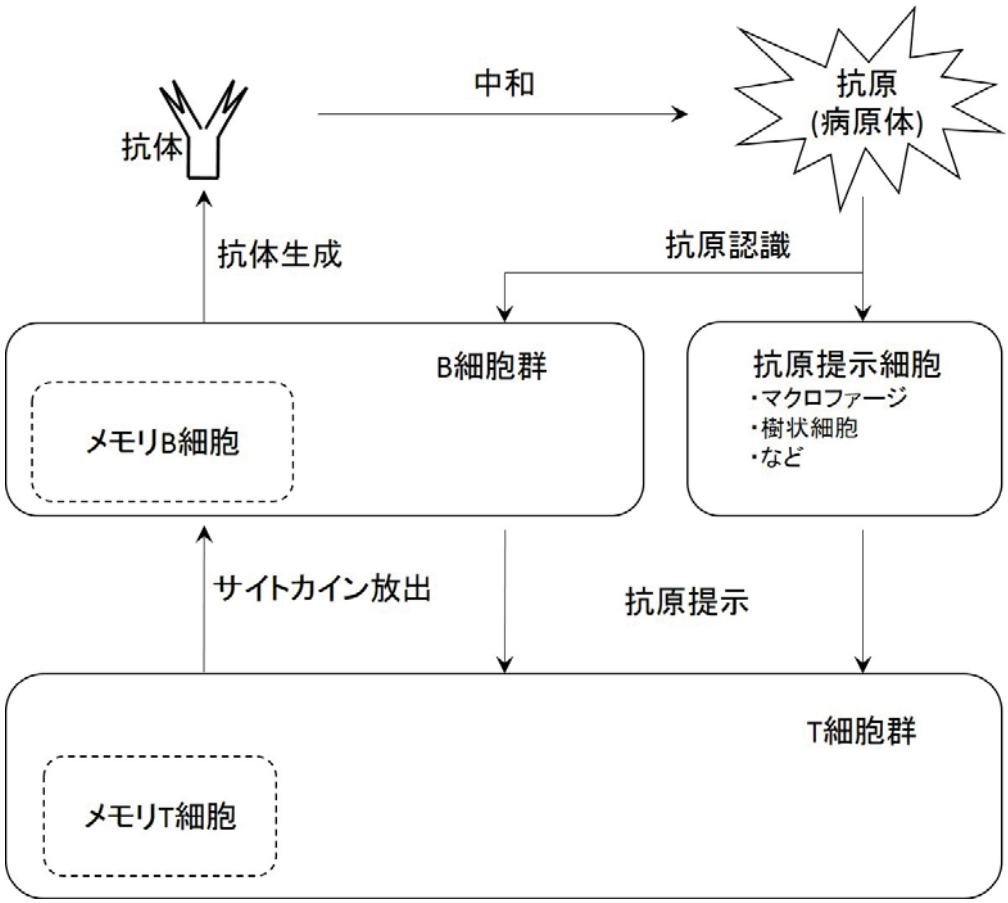

図 2.1: 獲得免疫系の構成

まず，抗原とは免疫反応を引き起こす物質全般を指す言葉である．これは例えば病原体のほか，場合によっては花粉や自己の細胞 (がん細胞) なども抗原となりうる．人体内に存在する抗原は，樹状細胞やマクロファージなどの食細胞により取り込まれ，タンパク質の破片であるペプチドと呼ばれる物質に分解される．このペプチドには取り込んだ抗原の特徴を示す情報が含まれており，この情報をヘルパー T 細胞と呼ばれる免疫細胞に提示する．ヘルパー T 細胞の表面にはこのペプチドを認識するための受容体があり，特定の病原体のみに合致する．上記ヘルパー T 細胞に抗原の情報を提示する細胞を，抗原提示細胞と呼ぶ．なお，マクロファージなどは自然免疫を担う細胞でもあり，直接病原体の駆除も行っている．

ヘルパー T 細胞は，免疫系の司令塔ともいうべき存在であり，抗原提示細胞よりもたらされた情報を受容体を介して読み取る．抗原情報と受容体が合致した場合にヘルパー T 細胞は活性化し，分裂をして増殖する．また，サイトカインと呼ばれる物質を外部放出し，提示された抗原に対して有効な攻撃手段を持つ免疫細

胞を活性化させ，抗原の駆除を促進させる．具体的にはＢ細胞やキラー細胞に対してサイトカインが伝達されそれぞれの細胞が活性化する．このうち，キラーＴ細胞は病原体に冒されるなどして変異した人体を構成する細胞を排除することを担当しており，Ｂ細胞の方は，外部より侵入した病原体などを担当している．実際にはＢ細胞が直接抗原に対して作用するのではなく，抗体と呼ばれる抗原を中和しその活性を抑える物質を産生する．これにより，抗原は無力化され，最終的に食細胞により貪食され駆除される．このようにして，獲得免疫系は体内の恒常性を維持している．なお，Ｂ細胞も抗原提示細胞として機能することができる．ただし，Ｂ細胞単体では活性化することはできず，ヘルパーＴ細胞からの指示を要する．

ところで，前述のヘルパーＴ細胞やＢ細胞などは特定の抗原に対して特異的に反応する．つまり，ある抗原に対しては特定のヘルパーＴ細胞やＢ細胞（またはキラーＴ細胞）しか反応しない．そのため，前述のとおり獲得免疫系は特異的防御といわれる．Ｔ細胞及びＢ細胞は多種多様な抗原に対して機能できるようにさまざまなタイプの細胞が常に生成されている．獲得免疫系の細胞及び生成された抗体は一定の寿命により死滅していくが，抗原の駆除に特に貢献したＢ細胞やヘルパーＴ細胞の一部は他の細胞と比べ特に長い寿命を得て体内を循環するようになる．このため，同じ抗原が再び体内に侵入してきた場合，その抗原に対し素早く反応し速やかに駆除する．これを免疫学的記憶という．この記憶作用を利用したものが，インフルエンザやはしかなどの予防接種である．

以上，獲得免疫系の反応について簡単に説明したが，免疫系は実際には各要素が複雑に絡み合い，それがちょうど平衡状態を保つことによって結果的に人体の恒常性を維持している．例えば，ヘルパーＴ細胞より放出されるサイトカインには非常に多くの種類があり，かつ１種類のサイトカインが複数の効果をもたらすようになっている．このサイトカインが複数種類放出されることにより，あるサイトカインが別のサイトカインの産生を促したり，協調・競合することにより，免疫系は全体として機能している．これは，サイトカインネットワークと呼ばれる．また，Ｂ細胞によって生成される抗体も見方を変えると抗原として作用するため，これを認識して別の抗体が生成されることによって生成されるイデオタイプネットワーク説などもある．

2.3 免疫型強化学習器

前節にて説明した獲得免疫系の働きを参考に構築した強化学習器が免疫型強化学習器である．本節では免疫型強化学習器のアルゴリズムを説明したのちに学習パラメータの設定基準やモデルフリー型の強化学習手法として有名な Profit Sharing[30] との比較を行う．

2.3.1 学習アルゴリズム

まず，多くの強化学習手法を構築する上で前提となるマルコフ決定過程 (MDP) を用いてロボットが動作する環境及び実行できる行動について獲得免疫系の働きにそれぞれ当てはめていく．エージェントが動作する全体の行動空間 S 内での状態を $s_i \in S$，エージェントが実行のできる行動 a_k とする．生物が備えている獲得免疫系では対処 (中和) すべき対象は抗原であり，この抗原について中和を行うのが抗体であるので，抗原をエージェントの状態 s_i，抗体エージェントが実行する行動 a_k と当てはめて学習器のモデル化を行う．ここで i は行動空間内の状態のインデックス，k は行動のインデックスである．免疫型強化学習器の概略を図 2.2 に示す．免疫型強化学習器では抗体 $Ab(s_i, a_k)$ の選択・生成することによってエージェントの行動が実行される．この抗体は濃度パラメータを持っており，生成時に最大値となるが時間が経つにつれ減少する．エージェントの行動にあたる抗体を生成するのは B 細胞であるが，B 細胞自体は直接抗原 (環境) を認識して抗体を生成することができない．抗体の生成には Th 細胞からのサイトサイトカインシグナルおよび B 細胞の活性度 (または B 細胞の数) が関係するため，抗体生成の評価値として次式を定義する．

$$v_k = m_k \times w_k(s_i) \tag{2.1}$$

ここで，v_k は抗体生成の評価値，m_k は B 細胞の活性度，$w_k(s_i)$ は状態 s_i に放出されているサイトカインシグナルである．実際の獲得免疫系における B 細胞の活性度 m_k はさまざまな要因によって変化するが，免疫型強化学習器では状態 s_i に対してあらかじめ実行することができない行動が判明している場合では $m_k = 0$，それ以外の状態では $m_k = 1$ をとるとする．これにより，あらかじめ実行できない行動がわかっている場合ではその行動の選択を抑制することができ，学習時間の短縮化が望める．サイトカインシグナルは抗原情報との適合度やメモリ T 細胞などによって放出されるが，状態が離散値で表現される環境では適合度は全て等しく

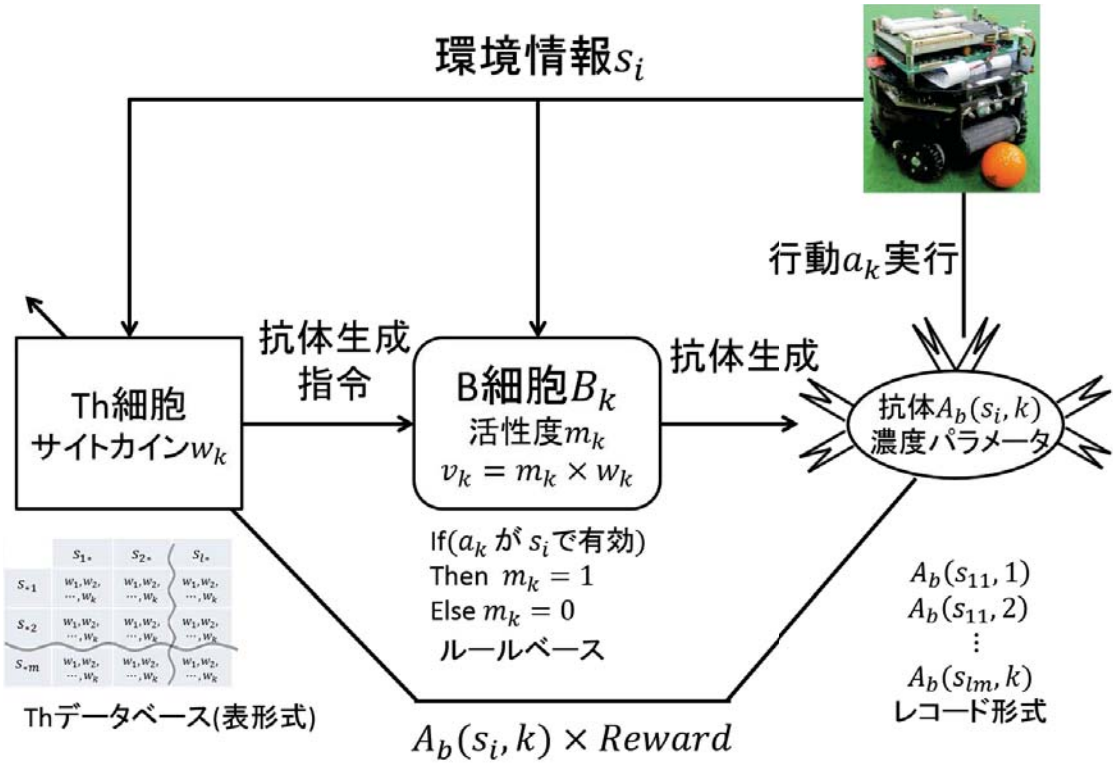

図 2.2: 免疫型強化学習器概略図

発生する．また，学習が行われていない状態ではメモリ T 細胞は存在しないため放出されるサイトカインシグナルに差違はない．実際の獲得免疫系でも同様の作用であるが，抗原中和に功績した Th 細胞はメモリ細胞として体内にとどまっている．この作用を模擬し，1 回の学習試行が行われるごと得られる報酬を元にメモリ T 細胞の情報は更新し，このメモリ T 細胞の情報をサイトカインシグナル $w_k(s_i)$ として使用する．このメモリ T 細胞は複数の細胞の情報によって構築されるため以降 Th データベースとして表現する．

以上が獲得免疫系を強化学習法へのモデル化についてであるが，実際の評価値更新 (Th データベースの更新法) 及び行動選択について次にまとめる．

2.3.1.1 Th データベースの更新

エージェントが目標状態に到達し環境から報酬を得た場合，Th 細胞群のサイトカインシグナル w_k を更新する．サイトカインシグナルの更新は次式を用いて行う．

$$w_k(s_i) \leftarrow w_k(s_i) + \alpha(r_k(s_i) - w_k(s_i)) \tag{2.2}$$

$$r_k(s_i) = \begin{cases} A_b(s_i,k) \times R & : A(s_i,k) \text{が存在する場合} \\ 0 & : \text{そのほか} \end{cases} \tag{2.3}$$

ここで，R は環境から得た報酬値を，$\alpha(0 < \alpha < 1)$ は学習率を表している．更新は全ての w_k について行われ，更新に使用された抗体を消滅させる．

次に，環境に対して最適なルールを獲得するためのパラメータ α,β 設定基準について説明する．

2.3.1.2 行動選択

1 エージェントの状態が s_i の場合，Th データベースから各 B 細胞へのサイトカインシグナル $w_k(s_i)$ を放出する

2 状態 s_i における B 細胞の活性度 m_k を取得する

3 B_k の評価値を $v(k) = m_k \times w_k$ とし，ルーレット選択などの行動選択手法を用いて B 細胞を決定する

4 選択された B 細胞に設定されている行動を実行する

5 選択された k 番目の B 細胞によって抗体 $Ab(s_i,k)$ を生成し，行動の濃度パラメータを $Ab(s_i,k) = 1$ に設定する．なお，同一抗体を生成する場合は抗体の濃度パラメータのみを $Ab(s_i,k) = 1$ に再設定する

6 過去に生成された他の抗体は (2.4) 式を用いて濃度の更新を行う．

$$A_b \leftarrow \beta \times A_b \tag{2.4}$$

なお，$\beta(0 < \beta < 1)$ は抗体濃度の減衰係数を表す．

以上の処理を 1 ステップとして繰り返して B 細胞の選択，抗体の生成を行い状態遷移をする．状態遷移の結果，目標に到達した場合に報酬を受け取り Th 細胞群の更新を行う．

評価値 v_k をもちいた行動選択において使用する行動選択手法は前に述べた初期値及び正の報酬値が与えられる場合はルーレット選択や局所解脱出を考慮した手法 [45] などを用いることができる．行動選択手法の詳細な検討は 2.3.3 小節にて述べる．

2.3.2 Profit Sharing との比較

免疫型強化学習器は学習の速度を優先させ，パラメータに依存しない学習方式である．本小節では，免疫型強化学習器と同じくモデルフリー型の学習方式である Profit Sharing との更新方式の比較を行う．モデルフリー型の学習方式で重要となるのはどのように経験した行動に報酬を割り当てるかということである．このことについて宮崎らが提案した等比減少関数を使用することによって合理的な学習を行えることを示している [1].

$$r(t) = R \times \frac{1}{S} \tag{2.5}$$

ここで，$r(t)$ は分配する報酬値，R は環境から与えられた報酬値，S は有効行動数+1 である．免疫型強化学習法は抗体の減衰係数を $\beta = \frac{1}{S}$ と設定することにより，この報酬分配則と等価な報酬を割り当てることができる．

次に，各更新プロセスにおける更新式の働きを解析する．免疫型強化学習器と Profit Sharing 更新式の大きな違いは，報酬を受け取ったエピソード中に経験しなかった状態-行動についても評価値を更新することである．ここでは，行動選択に b) 節にて述べるルーレット選択 ((2.20) 式) によって求められる確率を元に行動を選択するルーレット選択を使用する場合について各状態において有効・無効行動選択確率の増減について議論する．有効行動とはタスクを達成するために有効な行動，無効行動とはタスクの達成に寄与しない行動のことである．以後の解析において，環境から与えられる報酬および Q 値，サイトカインシグナルの初期値はともに正の値であることを仮定する．

a) 経験しなかった状態の更新 報酬を受け取ったエピソード中で経験しなかった状態において Profit Sharing では更新を行わない．

$$Q'(s, a_i) \leftarrow Q(s, a_i),\ \forall a_i \in A \tag{2.6}$$

よって，行動の選択確率の変化はおこならない．一方，免疫型強化学習器では Q 値の更新が行われる．報酬を受け取ったエピソード中で経験していない状態については，抗体情報が生成されていない．よって全ての行動に対して一律の報酬が割り当てられる ($r(t) = 0$)．この報酬値を使用して更新を行うと状態内の全ての行動の評価値が $(1 - \alpha)$ 倍に値が更新されるが，行動選択時において特定の行動の評価値が強化されることはない．

$$w'(s, a_i) \leftarrow (1 - \alpha) w(s, a_i),\ \forall a_i \in A \tag{2.7}$$

よって，Profit Sharing と免疫型強化学習器の更新内容は同等である．

b)　有効行動と無効行動の同時更新　ProfitSharing および免疫型強化学習の報酬関数がともに宮崎らの合理性定理に従っている場合は有効行動が強化されるため，有効な政策が得られる方向に得られるように学習が収束する．

よって，経験した行動が有効行動および無効行動のみの場合について考える．

c)　有効行動のみの更新　報酬を受け取ったエピソード中の経験したある状態 s において有効行動のみ選択した場合について考える．ここでは簡単のため，選択できる行動が有効行動 a_1 と無効行動 a_2 の 2 種類のみのについて取り扱う．それぞれの行動に対しての Q 値の更新は以下のように行われる．

$$Q'(s,a_1) \leftarrow Q(s,a_1) + \alpha(r - Q(s,a_1)) \tag{2.8}$$

$$Q'(s,a_2) \leftarrow Q(s,a_2) \tag{2.9}$$

Q 値の更新後と更新前のルーレット選択における有効行動の選択確率の変化 $\Delta P_q(s,a_1)$ は次式となる．

$$\begin{aligned}\Delta P_q(s,a_1) &= \frac{Q'(s,a_1)}{Q'(s,a_1)+Q'(s,a_2)} - \frac{Q(s,a_1)}{Q(s,a_1)+Q(s,a_2)} \\ &= \frac{\alpha\left(r - Q(s,a_1)\right)Q(s,a_2)}{\left((1-\alpha)Q(s,a_1)+\alpha r + Q(s,a_2)\right)\left(Q(s,a_1)+Q(s,a_2)\right)}\end{aligned} \tag{2.10}$$

有効行動に関する Q 値の更新であるため，$\Delta P_q(s,a_1) > 0$ となることが望まれる．仮定した条件から分母は常に正の値であるが，受け取った報酬より Q 値の値が高い $(r < Q(s,a_1))$ 場合において $\Delta P_q(s,a_1)$ の値が負の値となり有効行動の選択確率が抑制される．この有効行動の抑制は Q 値の初期値を大きく設定した学習セットの場合，学習初期においてたとえ有効行動を選択しても無効行動が強化されてしまうため学習収束速度に影響を与えることとなる．

一方，免疫型強化学習のサイトカインシグナルの更新は次式となる．

$$w'(s,a_1) \leftarrow w(s,a_1) + \alpha(r - w(s,a_1)) \tag{2.11}$$

$$w'(s,a_2) \leftarrow (1-\alpha)w(s,a_2) \tag{2.12}$$

更新後と更新前のルーレット選択における有効行動の選択確率の変化 $\Delta P_w(s,a_1)$

は次式となる．

$$\begin{aligned}\Delta P_w(s,a_1) &= \frac{w'(s,a_1)}{w'(s,a_1)+w'(s,a_2)} - \frac{w(s,a_1)}{w(s,a_1)+w(s,a_2)} \\ &= \frac{\alpha r w(s,a_2)}{(w(s,a_1)+\alpha(r-w(s,a_1))+(1-\alpha)w(s,a_2))\,(w(s,a_1)+w(s,a_2))}\end{aligned} \tag{2.13}$$

Profit Sharing の場合と同様に仮定した条件から分母は正の値となり，分子の部分も正の値となる．このため，免疫型強化学習器では有効行動のサイトカインシグナルの更新では常に有効行動を強化するように更新が行われる．このため，学習収束速度を速めることが可能となっている．

d)　無効行動のみの更新　有効行動の例と同じく無効行動のみが選択された場合について考える．無効行動について報酬が与えられた場合に Profit Sharing での Q 値の更新は次式となる．

$$Q'(s,a_1) \leftarrow Q(s,a_1) \tag{2.14}$$

$$Q'(s,a_2) \leftarrow Q(s,a_2)+\alpha(r-Q(s,a_2)) \tag{2.15}$$

Q 値の更新後と更新前のルーレット選択における有効行動の選択確率の変化 $\Delta P_q(s,a_1)$ は次式となる．

$$\Delta P_q(s,a_1) = \frac{-Q(s,a_1)\alpha(r-Q(s,a_2))}{(\alpha(r-Q(s,a_2))+Q(s,a_1)+(Q(s,a_2)\,(Q(s,a_1)+Q(s,a_2))} \tag{2.16}$$

有効行動の更新時と同じく有効行動が強化されるかどうかは報酬と現在の Q 値 ($r < Q(s,a_2)$ の場合は強化) によって定まる．Q 値の初期値が非常に小さい値の場合では無効行動に与えられる報酬値が大きくなってしまい，無効行動を強化するように学習が行われる．この影響を打ち消すためには有効行動を複数回選択・学習して有効行動の評価値を上昇させる必要があるが，行動の選択は重み付きの確率でおこなわれるため有効行動が選択されるまで多くの試行を必要とし，学習の収束速度に影響を与える．

一方，免疫型強化学習のサイトカインシグナルの更新式は次式となる．

$$w'(s,a_1) \leftarrow (1-\alpha)w(s,a_1) \tag{2.17}$$

$$w'(s,a_2) \leftarrow w(s,a_2)+\alpha(r-w(s,a_2)) \tag{2.18}$$

表 2.2: 任意の状態において強化される行動パターン

更新条件	免疫型強化学習	Profit Sharing
行動なし	変化なし	変化なし
有効行動と無効行動の更新	有効行動を強化	有効行動を強化
有効行動のみの更新	有効行動を強化	Q 値によって変化
無効行動のみの更新	有効行動を抑制	Q 値によって変化

更新後と更新前のルーレット選択における有効行動の選択確率の変化 $\Delta P_w(s, a_1)$ は次式となる．

$$\Delta P_w(s, a_1) = \frac{-\alpha r Q(s, a_1)}{((1-\alpha)w(s, a_1) + w(s, a_2) + \alpha(r - w(s, a_2)))\,(w(s, a_1) + w(s, a_2))} \tag{2.19}$$

免疫型強化学習器においても無効行動のみのサイトカインシグナル更新では有効行動を強化する更新は行われない．しかし，無効行動の強化は Profit Sharing よりも少ないため (報酬値が小さいため) 学習収束速度への影響が少ないといえる．

以上から Profit Sharing は学習初期の報酬値と Q 値の差がある場合，学習速度を阻害する可能性があることを示した．この影響をできるだけ抑えるには Q 値の初期値を適切に設定する必要がある．一方で，免疫型強化学習器はサイトカインシグナルの更新時に初期値の影響はほとんど受けることなく，有効行動の強化が可能であることが示された．

2.3.3 行動選択手法についての一考察

強化学習は試行錯誤の結果よりある時点において選択すべき行動を学習していくが，学習機能をうまく動かすためのに重要となるのが探索と搾取のバランスになる [46][47]．ここでの探索は任意の行動を実行してそれに対して得られる報酬値を調査することであり，搾取は探索によって得た学習結果を利用した適切と思われる行動の選択である．一般的に学習初期であればさまざまな行動を経験した方が探索の効率が高い傾向となる．搾取による適切な行動選択の確度を高くするためには十分な探索が不可欠であるが，どの時点で探索と搾取を切り替えればよいか，といった明確な指針を任意のタスクに対して設定することは困難である．強化学習の研究において重要なテーマとなっており，学習アルゴリズムからの検討や行動選択時での検討 [45] などさまざまな研究が行われている．免疫型強化学習

[36] においても探索と搾取のバランスをとるための学習パラメータ選定方法についても議論されている．詳細は付録 B を参照されたい．

ここでは行動選択手法について提案されているさまざまな手法において，免疫型強化学習器に最も適した手法について検討を行う．強化学習手法の行動選択手法として代表的な手法として以下ようなものがあり，その概略を説明する．

a) グリーディ手法，ε-グリーディ手法 学習結果の搾取を積極的に利用した手法がグリーディ手法である．グリーディ手法は得られた学習結果のうち，最も評価値が高い行動を選択する手法である．この手法において探索行為は学習初期の限られた時間内でしか行われない．強化学習手法では学習を始める際に各行動の評価値の初期値を任意の値として設定する．実行した行動について報酬が与えられた時，評価値の更新アルゴリズムによって評価値の初期値より低い値もしくは高い値に更新がおこなれる．仮に実行した行動の評価値が初期値より低い値に更新された場合，次回の行動選択ではその他の行動が選択され，より適する行動の探索が行われる．一方，初期値よりもより高い値に評価値が更新された場合では，次回の行動選択において同じ行動のみ選択される．これは，もし他の行動の方が評価値が高いものがあっても，行動実行時に確定的な状態遷移が起こる環境においては初回に選択 (探索) された行動が以後選択され続けるといった懸念がある．この場合，想定される報酬値よりも初期値を大きく設定する事によりある程度の探索が促進されるが，トレードオフの根本的な解決にはならない．

この探索と搾取のバランスをとる方法としてランダム選択とグリーディ選択を組み合わせた ε-グリーディ手法がある．ε-greedy 手法では行動選択を行う前に事前に定義した確率 $\varepsilon(0 \leq \varepsilon \leq 1)$ を用いてランダム選択を行うか，グリーディ選択を行うか決定する．$\varepsilon = 0$ のときグリーディ手法，$\varepsilon = 1$ のときにランダム選択と同一になる．この手法においては定期的にランダム選択が実行されるため探索の機会はある程度確保されるため，グリーディ手法と比べてより探索範囲が広くなり，適切な解を学習できる可能性がある．しかし，探索と搾取のバランスを確率 ε によって適切に設定する必要がある．

b) ルーレット選択，ボルツマン選択 上記の手法は行動の評価値から直接的に行動を選択する (最も高い評価値の行動を選択) 手法であったが，行動の評価値を確率分布に変換してから行動を選択する手法がある．よく使用される確率分布へ

表 2.3: 行動選択手法の比較

行動選択手法	探索能力	搾取	特徴
グリーディ手法	×	◎	搾取を積極的に用いる 探索をほぼ行わない
ε-グリーディ手法	△	○	一定の確率で探索を常に行う 設定パラメータに性能が依存
ルーレット選択	○	△	負の評価値を扱えない 搾取には評価値に大きな差違が必要
ボルツマン選択	○	○	設定パラメータに性能が大きく依存

の変換式として次式のルーレット選択とボルツマン選択がある．

$$p(a|s) = \frac{Q(s,a)}{\sum_{a_i \in A} Q(s,a_i)} \tag{2.20}$$

$$p(a|s) = \frac{\exp(Q(s,a)/T)}{\sum_{a_i \in A} \exp(Q(s,a_i)/T)} \tag{2.21}$$

ここで，ボルツマン選択 ((2.21) 式) における $T(0 \leq T)$ は温度係数と呼ばれており，行動選択のランダム度合いを設定するパラメータである．なお，この温度係数を $T = 0$ とした場合にはグリーディ手法と等しくなり，$T = \infty$ としたときはランダム選択となる．これらの手法の共通点は学習初期における各行動の評価値が収束していない場合は均等に行動選択され，学習が進んで評価値が収束すると評価値の高い行動が確定的に選択されるようになる．ルーレット選択の利点としてはパラメータ設定が必要ない点があげられるが，無効行動と有効行動の評価値に差が小さい場合にはランダムに行動選択が行われることと負の評価値がある場合正しく確率を計算できない．反対にボルツマン行動選択は負の報酬値や無効行動と有効行動の差が小さい場合での温度係数 T を適切に設定することにより対応可能であるが，探索と搾取の関係を決める温度係数の設定問題がそのまま残されている．

以上の行動選択手法の特徴を表 2.3 にまとめる．これらの中から免疫型強化学習器に適する行動選択手法について考察する．まず，積極的な探索を行わないグリーディ手法は今回の考察から除外する．強化学習の行動選択手法に求められるのは学習初期には十分な探索が行われ，学習後期には結果の搾取が行われることである．この点を考慮すると ε-greedy 手法は探索行為が学習結果と独立して設定される確率 ε によってのみ調整される．この確率を動的に変更することにより上記の条件を満足することができるが，パラメ差違ータ変更則を別途導く必要がある．

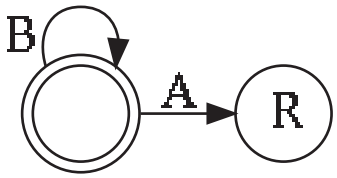

図 2.3: 回帰ルールを含む行動選択

ルーレット選択とボルツマン選択は本質的には同じアルゴリズムであるが，確率分布の導出方法が異なる．両手法の違いが顕著に表れるのは無効行動と有効行動の評価値の差違が小さい場合である．ルーレット選択では差違がない場合は平等に確率を計算してしまうため，互いを区別することができない．一方，ボルツマン選択は温度係数 T を小さく設定することによりグリーディ手法に近づけることができるため，無効行動と有効行動の区別をすることができる．このため，免疫型強化学習器によって更新される報酬値の傾向からルーレット選択もしくはボルツマン選択のどちらが適するかを検討する．

まず，ある状態において無効行動と有効行動の分離が最も困難な場合においての報酬について考える．最も分離が困難な例は宮崎らの合理性定理より唯一の回帰的無効ルールを含む場合である．詳細は付録 A.2 の補題 1 を参照されたい．図 2.3 の場合において回帰的無効行動 B および有効行動 A が 1 回ずつ実行され，有効行動に対して報酬 R が割り当てられた場合を考える．2.3.2 小々節と同様に抗体濃度更新式 ((2.4) 式) の定数 $\beta = 1/S$(ここで付録 A.2.1 にて述べられている強化減少比を $S = L + 1$，有効ルール数 $L = 1$) とすると，疫型強化学習で回帰的無効行動及び有効行動にに割り当てられる報酬値 r_B, r_A は以下のようになる．

$$r_A = R \tag{2.22}$$

$$r_B = \frac{R}{S} = \frac{R}{L+1} = \frac{R}{2} \tag{2.23}$$

免疫型強化学習において行動の評価値の更新は (2.2) 式に表されるように与えられる報酬値に近づくように更新がなされる．つまり与えられる報酬値 (2.22)，(2.23) 式の通りに評価値が更新された場合のルーレット選択による回帰的無効行動の選択確率は $p(B|s) = \frac{1}{3}$ となる．実際には，有効行動数は $1 \leq L$ であるため，有効行動数が L かつ評価値が報酬関数に収束した場合における無効行動の選択確率は以

下となる．

$$p(B|s) = \frac{1}{L+2} \leq \frac{1}{3} \tag{2.24}$$

これは最も分離が困難な回帰ルールを含む場合であって，その他の場合では無効行動の選択確率はより低くなる．このことによりルーレット選択を使用した場合でも無効行動の選択確率は抑制される．さらに免疫型強化学習では選択されなかった，つまり報酬を受け取らなかった行動については (2.2) 式によって一様に評価値を $1-\alpha(0<\alpha<1)$ 倍に減少させる働きもある．これにより，無効行動が選択される確率は更に少なくなる．ボルツマン選択においても温度係数 T を適切に設定することにより同様の議論となる．しかし，探索能力と搾取のバランスを制御するパラメータがボルツマン選択の温度係数 T と免疫型強化学習器の学習率[1] α の 2 種類が存在することになりパラメータ選択に注意が必要となる．またボルツマン選択は行動選択時に都度指数計算を行う必要があり計算負荷があがる懸念があるため，免疫型強化学習器の行動選択にはルーレット選択を用いることが望ましい．

2.4 おわりに

本章では，本研究で提案している手法の考え方の基礎となる，免疫系の基本的な仕組みについて説明した．免疫系は，自然界において人体を生存させ続けるためにきわめて重要な働きをしているものであり，この仕組みをうまく活用することにより，様々な環境において自律的に動作する知能エージェントを実現する大きな手助けとなることが期待される．この働きを利用したモデルフリー型の学習方式である免疫型強化学習について説明をした．免疫型強化学習法はモデルフリー型で代表的な強化学習手法の 1 つである Profit Sharing より素早い収束速度を持つことを説明した．次章からはこの強化学習法の適用範囲を拡張するための方法について述べる．

[1] 付録 B を参照

第3章　状態の連続値表現を考慮した免疫型強化学習法

3.1　はじめに

本章では免疫型強化学習器を環境情報の連続値表現で使用できるように改良を行う．そのために獲得免疫系の構造を再確認し，学習器の再モデル化を行う．獲得免疫系では予防接種の例のように病原体に類似している別の病原体 (抗原) をあらかじめ摂取しておくことにより免疫力を獲得することができる．従来の免疫型強化学習器ではセンサ観測値と行動を記憶している状態が完全一致した情報のみを利用して行動選択を行っていた．本章では実際の抗原検知メカニズムを基にした環境の連続値表現を直接取り扱うことのできる方法を提案する．提案手法の有効性をさまざまなシチュエーションのシミュレーション適用することによって検証を行う．

3.2　連続状態表現への拡張

前章にて説明をした免疫型強化学習器では行動選択使用する行動の評価値 $v(k)$ を (2.1) 式を使用して求めていた．免疫型強化学習では環境はMDP の特性を満たし，環境状態が離散値で表現されることを前提としてアルゴリズムが構築されていた．この場合，センサ観測値と記憶している状態が完全に一致するため，過去に学習をしたTh 細胞のサイトカインシグナル w_k をそのまま使用することができる．人間の獲得免疫ではTh 細胞で提示された抗原を認識するTCR は 10^{18} をこえるバリエーションを表現することができるため，迷路探索問題などの限られた次元数の離散表現が可能な環境では問題なく使用することができる．

しかし，実ロボットが動作する連続値環境への応用を考えた場合では次元数がより高次元になったり，離散化度合いの問題が無視できなくなる．一般に次元数を多くかつ離散化度合いを細かくすれば詳細な環境表現が可能であるが，取り扱わな

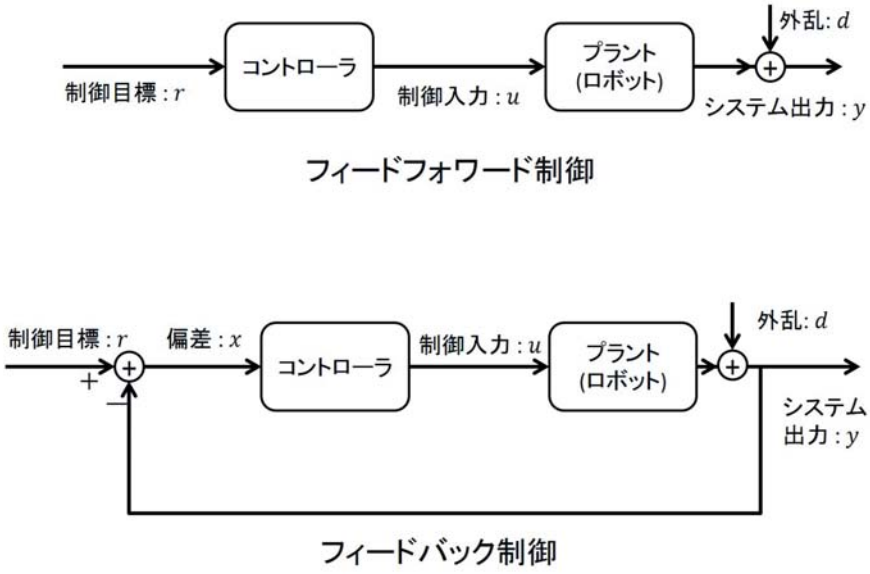

図 3.1: 代表的なコントローラ構造

くてはならない状態数の爆発が発生する．場合によっては Th 細胞が備える TCR の表現数を越える可能性も否定できない．また，上記はセンサ情報などが現状態を完全に観測できることを前提としているが，実ロボットの場合では図 3.1 に示すように観測ノイズ (外乱) などの影響により観測した情報に不確かさが含まれる場合がある．この不確かさを含んだまま細かい離散化を行うと状態認識の不一致が発生し，MDP すら満たすことができなくなる恐れがある．このような問題を解決する手法として図 3.2 のように粗な状態分割まま起点をずらした複数の Q テーブルを使用する手法 [37] などがあるが，次元の呪いによる影響が完全に解決されたわけではない．

ここで実際の獲得免疫系においての振る舞いを再確認すると，2.2.2 小節で述べた通り，Th 細胞は抗原の認識は抗原提示細胞によって分解されたペプチドの他に提示をした細胞についても同時に認識している．このペプチド単体情報では Th 細胞の一部の受容体 (TCR) のみとしか合致しないため Th 細胞の活性度はあまり増加しない．しかし複数の受容体に刺激がもたらされた場合，その度合いに応じて B 細胞へとサイトカインシグナルを放出する [44]．実際の抗原認識の働きに着目し，Th 細胞の活性度現状態と記憶されている状態との距離を利用する．以上のように獲得免疫作用を再モデリングすることにより行動選択にサイトカインシグナルと活性度を利用した強化学習器修正を行う．

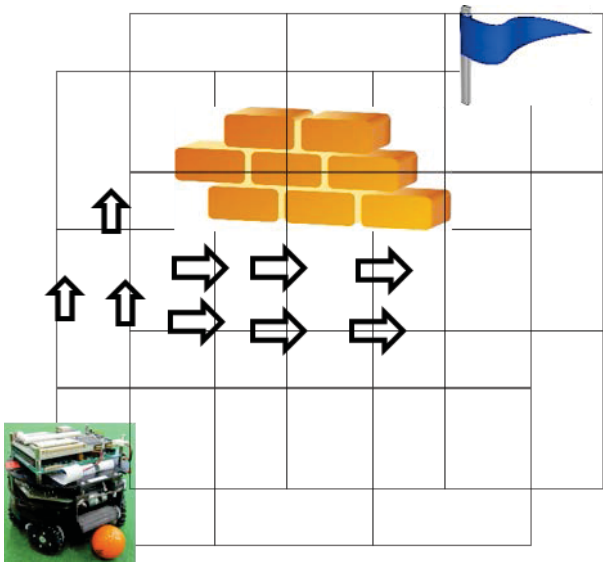

図 3.2: 状態分割をずらす手法

連続値表現の免疫型強化学習器アルゴリズムを以下のように構築する．Th 細胞を状態 $\boldsymbol{\xi} = [\xi_1, \xi_2 \cdots, \xi_n]$，行動 a_k，およびサイトカインシグナルを記録した細胞として生成をする．ただし，すべての状態およびすべての行動について同一のサイトカインシグナルを出力する特別な細胞 Th_0 を 1 つ生成する．この細胞にサイトカインシグナルの初期値 w_{ini} を設定する．j 番目の Th 細胞に記憶されている状態 $\boldsymbol{\xi}^j$ と現状態 $\boldsymbol{\xi}'$ との活性度の計算に次式を用いる．

$$L(Th_j, a_k) =$$

$$\begin{cases} \kappa \sum_{p=1}^{n} |\xi'_p - \xi^j_p| & a_k \text{ memorized} \\ \infty & \text{otherwise} \end{cases} \tag{3.1}$$

(3.1) 式の距離計算方式はマンハッタン距離 (L_1-距離) であり，各次元ごとの距離の総和を距離としたものである．

κ はゲインパラメータで正の値を設定する．κ の値を大きくすると細胞の数を制限できる．これは，離散化度合いを細かくした場合と等価になる．活性度と評価値を用いて Th 細胞が出力するサイトカインシグナルを求める．

$$w_k = \sum_{j=0}^{N} \frac{W_j}{\exp(L(Th_j, a_k))} \tag{3.2}$$

ここで N は Th 細胞の総数，W_j は j 番目の Th 細胞に記憶されている評価値である．

連続値環境を考慮した行動選択アルゴリズムは以下となる．

1 エージェントの状態が $\boldsymbol{\xi}'$ の場合，状態 $\boldsymbol{\xi}'$ における B 細胞の活性度 m_k を取得する

2 Th 細胞が出力するサイトカインシグナルを (3.2) 式を用いて計算する

3 $v(k) = m_k \times w_k$ として，行動選択における B_k の評価値を $v(k)$ としてルーレット選択を行う．

$$P(\xi', a_k) = \frac{v(k)}{\sum_i v(i)} \tag{3.3}$$

4 選択された k 番目の B 細胞によって抗体 $Ab(\boldsymbol{\xi}, k)$ を生成し，行動の濃度パラメータを 1 に設定する．なお，同一抗体を生成する場合は抗体の濃度パラメータのみを 1 に再設定する

5 過去に生成された他の抗体は次式を用いて濃度の更新を行う．

$$A_b \leftarrow \beta \times A_b \tag{3.4}$$

なお，$\beta(0 < \beta < 1)$ は抗体濃度の減衰係数を表す．

1 行動選択によって生成された抗体情報を元に Th 細胞を生成し，評価値を以下の値に設定して抗体情報を削除する．

$$W_j = \alpha \times A_b(\boldsymbol{\xi}, k) \times R \tag{3.5}$$

2 次式ですべての Th 細胞の評価値を更新する．

$$W_j \leftarrow W_j(1 - \alpha) \tag{3.6}$$

提案アルゴリズムを図 3.3 に示す．図上の赤い部分が本研究において改良を行った部分となる．離散型の強化学習器では作業空間の大きさをあらかじめ求める必要があったが，以上の様にアルゴリズムを構築することによって作業空間の大きさを事前に決める必要が無くなる．また，κ の値を変化させることで行動選択に用いられる細胞の数 (離散化度合いに相当) を変化させることができるため，学習前に厳密に状態分割数を設定する必要がなくなる．

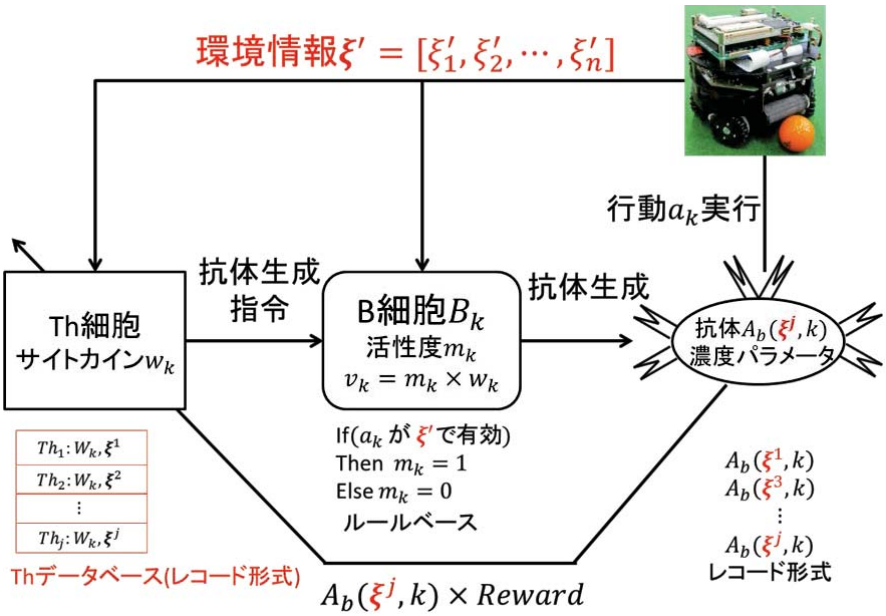

図 3.3: 連続値環境向け免疫型強化学習器概略

3.3 離散型強化学習法との比較

本節では離散型強化学習器と提案する連続型強化学習器の比較を行う．免疫型強化学習法の特徴はサイトカインシグナルの更新時に報酬を受け取る Th 細胞以外の細胞も更新が行われるという点であり，アルゴリズムの修正では行動選択時に適合度を導入し，経験した状態を個別に管理している．これらの修正を行っても離散型強化学習器の働きと同等になることを示す．

まず，はじめに更新プロセスの等価性について述べる．ここでは，連続型・離散型ともに同じ行動を選択し，環境から同様の報酬が与えられた場合を考える．図 3.4 は連続値による状態表現 (ξ_j) と離散値による状態表現 (s_i) においての Th データベースを示したものである．連続値状態表現では行動を選択した状態を点として，離散値状態表現ではあらかじめ定めた一定範囲内にある状態をまとめて表現している．報酬が与えられるのは行動を選択（実行）した時点への状態表現を持つ Th 細胞である．この時点で，生成された抗体情報から離散型の強化学習法では状態の連続値状態表現 (ξ_j) から離散状態表現 (s_i) に変換される．仮定した条件より連続型・離散型ともに抗体情報の更新は同等である．ここでエピソード ep 回学習した後の状態 s_i，行動 a_i のサイトカインシグナル値は以下のように表すことが

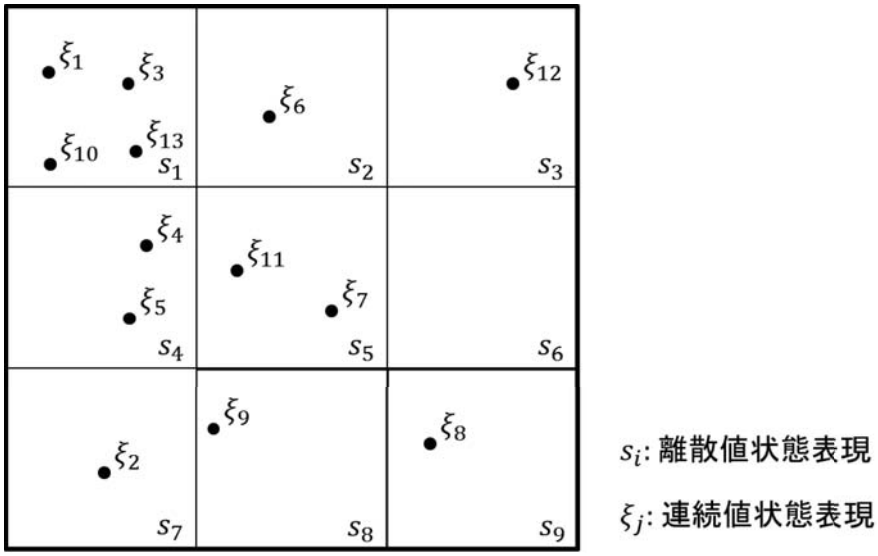

図 3.4: 連続値状態表現と離散値状態表現

できる．

$$w_{ep}(s,a) = \sum_{p=1}^{ep} \alpha^p (1-\alpha)^{p-1} r_p + (1-\alpha)^p w_{ini} \tag{3.7}$$

ここで，r_{ep} はエピソード ep で与えられた報酬，w_{ini} はサイトカインシグナルの初期値である．同様に連続型の強化学習法でのサイトカインシグナルの更新値は (3.5),(3.6) 式から以下となる．

$$w_{ep}(\xi,a) = \alpha^{ep}(1-\alpha)^{ep-1} r_p \tag{3.8}$$

これを離散型強化学習器と同じ範囲をまとめ，初期値となる Th 細胞 Th_0 の出力するサイトカインシグナルを加えると離散型の強化学習器と同じ更新式であることがわかる．

離散値状態表現を用いた従来手法と提案した連続値状態表現の免疫型強化学習の最大の違いは行動選択時の状態情報の取り扱い法である．離散値状態表現はサイトカインシグナルの更新法と同じく，図 3.5 に示すようなタイルコーディングなどを用いて現在の状態を離散化した状態に置き換えて処理をする．一方，提案手法は図 3.6 に示すように現在の状態を中心として周囲の情報を収集する．収集された Th 細胞情報は現状態と距離が近いほど高い適合度を表し，距離が遠くなるにつれ適合度が低く割り当てられる．もし，状態分割の境界に非常に近い場所に状態にいた場合であっても離散型の方式であればほかの状態に割り当てられた報酬

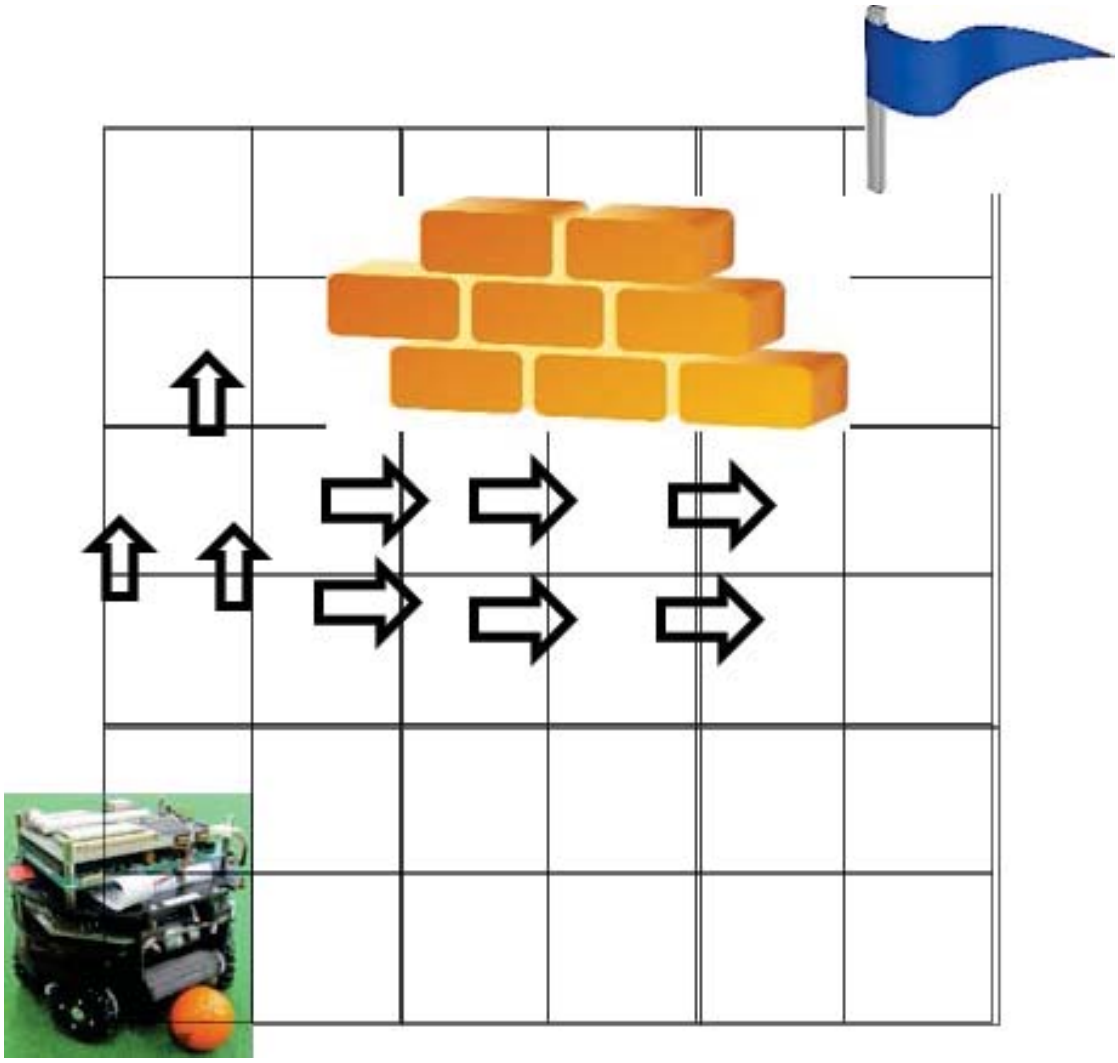

図 3.5: 離散値状態表現における行動選択

を利用することができない．これは提案手法の距離計算法に当てはめて考えると，同一状態にあるものは距離を０，それ以外を距離無限大として取り扱っていることになる．このため，離散化度合いが荒い場合では正しい行動を選択することができない．

提案手法では，保存されたサイトカインシグナルごとに距離を用いた適応度をもとめているため個々の経験を十分に生かすことができる．

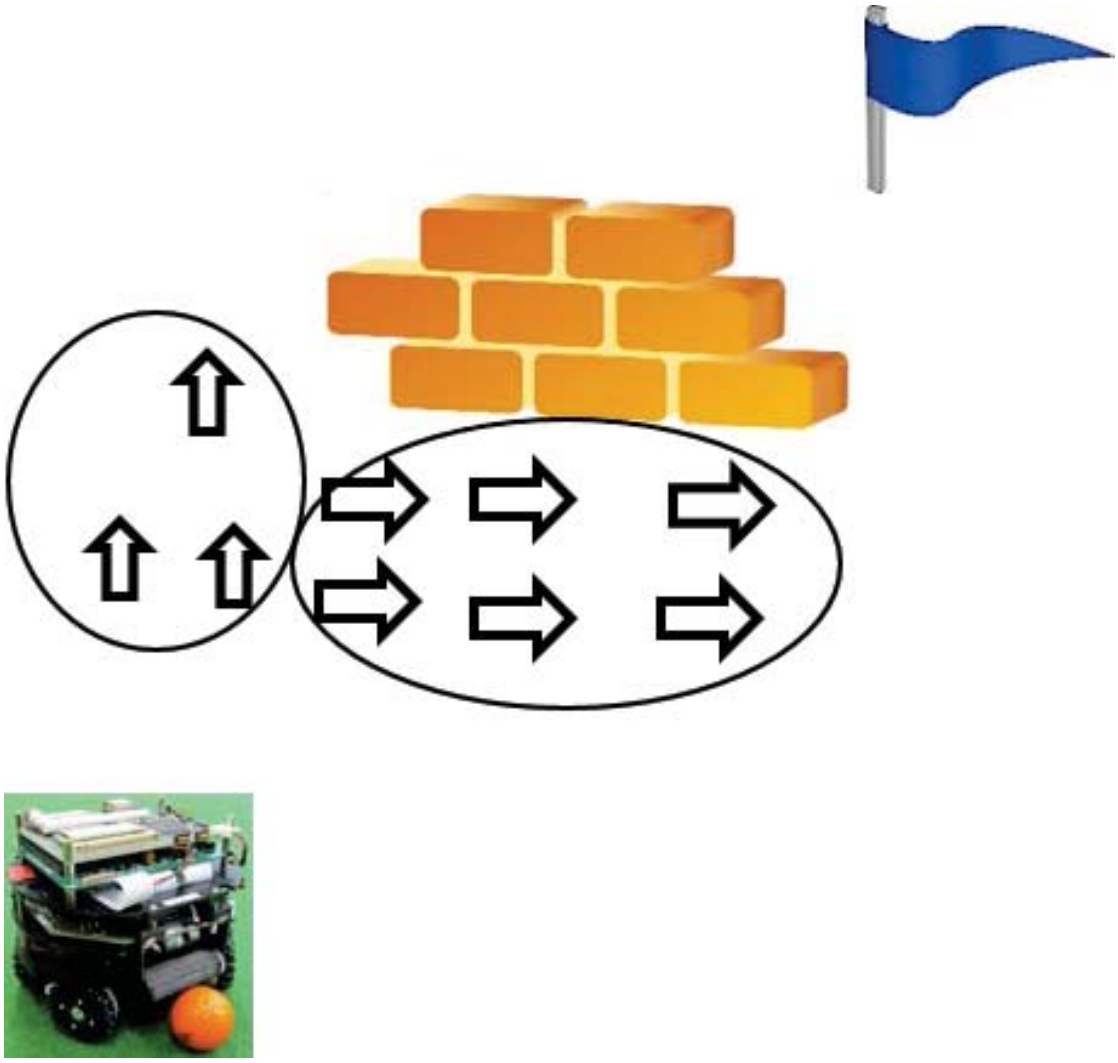

図 3.6: 連続値状態表現における行動選択

3.4 連続値環境への適用シミュレーション結果

3.4.1 マウンテンカーへの適用

a) 問題設定 マウンテンカー問題 [42] は強化学習のベンチマーク問題の一つの例であり，急坂を台車が上るための政策を得ることが目的となる (図 3.7).

$$y = 1 - \cos(\pi x/10) \tag{3.9}$$

この問題は迷路探索などと同様に一定の行動を選択し続けるだけではタスクを解くことができないうえ，環境の状態表現が連続値で表される．本検証において台

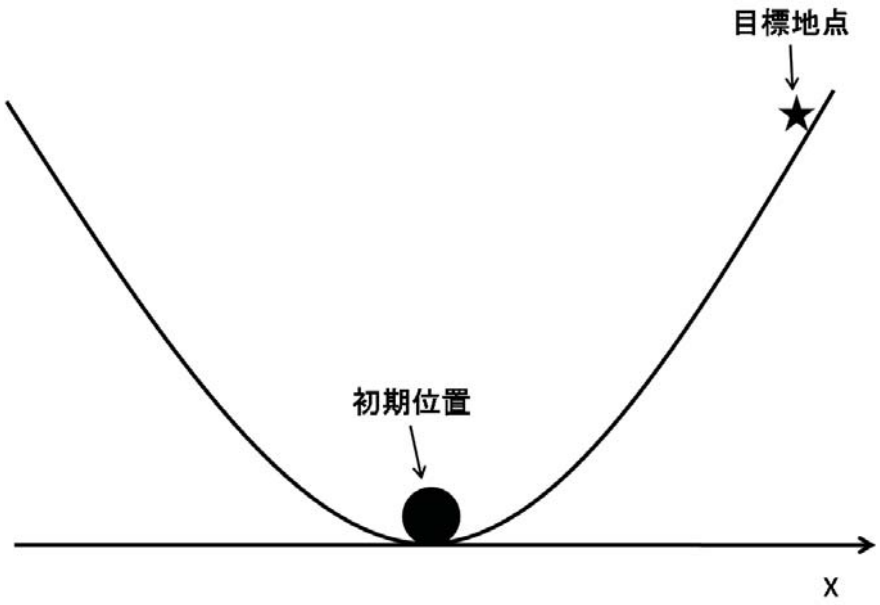

図 3.7: 坂道を登るシミュレーション

車モデルのシミュレーションを下記のオイラー法によって行う．

$$x_t = x_{t-1} + v_{t-1}\Delta t + \frac{1}{2}u\Delta t^2 \tag{3.10}$$

$$v_t = v_{t-1} + u\Delta t \tag{3.11}$$

ここで台車の位置を x，速度 v，加速度 u である．今回のシミュレーションにおいてシミュレーション周期 $\Delta t = 0.02$[sec] とした．台車の加速度 u は，台車に入力するトルク a と路面の勾配から受ける力を考慮して次式で設定した．

$$u = a - 1.2\sin(\pi x/10) \tag{3.12}$$

制御目標は初期状態を $x = 0, v = 0$ の谷の部分に静止している状態から，台車への入力トルク a を行動リスト $A = [-0.5, -0.1, 0, 0.1, 0.5]$ の中から 1 つ選択をして目標位置 $x = 6$ へ到着させることを目標とする．台車の加速度は ((3.12) 式) より，入力トルクと比べ路面の勾配から受ける重力加速度の方が入力する入力トルクより大きいため，最大トルクを入力し続けても目標に到達することはできない．このため，入力トルクに現在位置との差分を用いる単純な比例制御などでは目標を達成することができない．目標達成するためにはいったん逆方向へ移動して台車に勢いをつけるなどの政策を学習する必要がある．目標位置との距離差が $|\Delta x| < 0.4$ 以下の場合に報酬値 $R = 10$ を与える．上記の問題設定において，最適な政策は

表 3.1: マウンテンカーシミュレーションにおける学習パラメータ

パラメータ名	Q 学習	免疫型強化学習器 (離散型)	提案手法
学習率 α	0.1	0.1	0.1
割引率 γ	0.9	-	-
減少率 β	-	0.2	0.2
行動選択法	ε-greedy ($\varepsilon = 0.01$)	ルーレット選択	ルーレット選択

表 3.2: マウンテンカーシミュレーション状態分割パターン

分割パターン	x 方向分割数	$\dot{x}$ 方向分割数	総状態数
P_1	30	40	1200
P_2	60	80	4800
P_3	120	160	19200

19 回の行動で報酬を得ることができる．エピソード中で台車位置が $-10 \leq x \leq 10$ の範囲外に出た場合は，罰報酬などを与えず $x = -10$ もしくは $x = 10$ に移動して速度 $v = 0$ としてシミュレーションを続行した．また，台車の速度の最大値を $|\dot{x}| = 5$ を上限として設定した．これらの状態における制限は離散型の強化学習器を使用するうえで，状態分割を行う範囲を規定するためである．実機に適用する際ではこの範囲を物理制約などから設定をする．各エピソードの打ち切りステップを 5000 として 50000 エピソードの試行をそれぞれ 100 セット行う．今回の提案手法および離散型の Q 学習，離散型の免疫型強化学習の比較シミュレーションについて行った．提案手法のパラメータ κ の働きの検証，また離散型の強化学習手法においての状態分割について比較を行うため状態分割数を変えて検証を行った．学習器へ入力台車の位置および速度を状態 $\mathbf{s} = [x, \dot{x}]$ として学習器に与え，表 3.2 に示すように状態分割を等間隔に行った．また，表 3.1 に各手法で使用した学習パラメータをまとめた．2.3.3 小節の考察をもとに，提案手法及び改良前の免疫型強化学習手法の行動選択にはルーレット選択を使用した．Q 学習では行動選択手法に用いられることの多い ε-グリーディ手法を用い，探索度合いを一般的に用いられている $\varepsilon = 0.01$ と設定した．

b) 学習結果と考察 マウンテンカー問題においての学習器の性能比較指標として学習の収束速度，および学習解の質を使用して評価する．ここでいう学習解の質は目標に到達するまでに行った行動選択回数のことを示す．今回の場合，最適

政策を得た場合の最小行動選択数がわかっているため，次式を学習解の質とする．

$$Q_{al}(\pi) = \frac{\mathrm{STEP}_{opt}}{\mathrm{STEP}_{\pi}} \tag{3.13}$$

学習の収束速度は学習器が探索による行動の評価値がある程度安定し，一定の解が得られるまでのステップ数とする．各手法において学習結果を表 3.3 に示す．学習の収束速度の結果としてQ学習の結果を図3.8，離散型強化学習器の結果を図3.9，提案手法の結果を図 3.10 に示す．これらの図は各エピソードでの目標到達までのステップ数を学習セット毎に平均したものである．比較を行ったどの手法においても，エピソードを継続することによりランダム行動選択よりも効率的に目標を達成することのできる政策を獲得できていることがわかる．しかし学習収束速度および，最終的に得られた解についてはそれぞれ異なる結果を示している．まず，それぞれの手法において状態分割数を大きくすることにより，学習結果の質(学習後の平均ステップ数)がよくなる傾向を示している．一方で，学習の収束速度については状態分割数を大きくすることにより，長い時間が必要になっている傾向が読み取れる．これは次元の呪いの影響が大きくトレードオフが発生することを示している．

次に個別の手法の結果について検討を行う．まずQ学習の結果を見ると学習収束までの時間が必要とするが学習の質は提案手法と同等の高い値を示している．Q学習は MDP 環境下で無限回の学習を行った際に最適解が得られることが理論的に証明されており，これに近い結果となっている．しかし，最終的に実行している学習後の平均ステップは 21 と最適解である 19 ステップではない．これは，ε-グリーディ手法では一定確率でランダム探索となるためであり，言い換えるとこの行動選択手法を使用している限り平均して最適解を選択することはない．常に最適解を得るためには学習結果の搾取に重点を置かなければならないが，このためには ε の値を動的に調整，もしくはボルツマン選択を使用して温度係数を適正に設定する必要がある．離散値型の免疫型強化学習では，Q学習と比べ短い時間で学習が収束していることが確認できる．しかしながら，Q学習と同等のエピソードを実行しても平均ステップがよい結果を示すことはない．これは，Q学習が環境を同定しながら学習を行うが，免疫型強化学習は経験した行動を元に学習を行うためである．免疫型強化学習器が最適な行動を学習するためには，ランダム選択において最適解となる行動を経験しなくてはならない．今回のシミュレーション条件では最適解の場合19ステップ必要となるが選択可能な行動が5種類あるため，行動の選択パターンは $5^{19} \simeq 19 \times 10^{12}$ 存在する．このため，今回の場合では

表 3.3: 学習結果の比較

学習手法	学習後の平均ステップ数	学習結果の質	学習収束エピソード
提案手法 P_1	21	0.90	200
提案手法 P_2	24	0.79	200
提案手法 P_3	26.1	0.73	200
Q 学習 P_1	33	0.58	1000
Q 学習 P_2	24	0.79	2000
Q 学習 P_3	21	0.90	3000
免疫型強化学習 P_1	51	0.37	300
免疫型強化学習 P_2	42	0.45	1500
免疫型強化学習 P_3	33	0.58	40000

最適解が学習される可能性はほぼ無い．最後に提案手法についてみると Q 学習と同等の学習結果を有しながら，学習の収束スピードは 1 番早い事が確認できる．Q 学習及び，離散型の強化学習器では離散化度合いによって学習結果が左右されているが，提案手法では離散化によって無視される近傍状態の評価値も利用できるため効率的に動作している事がわかる．

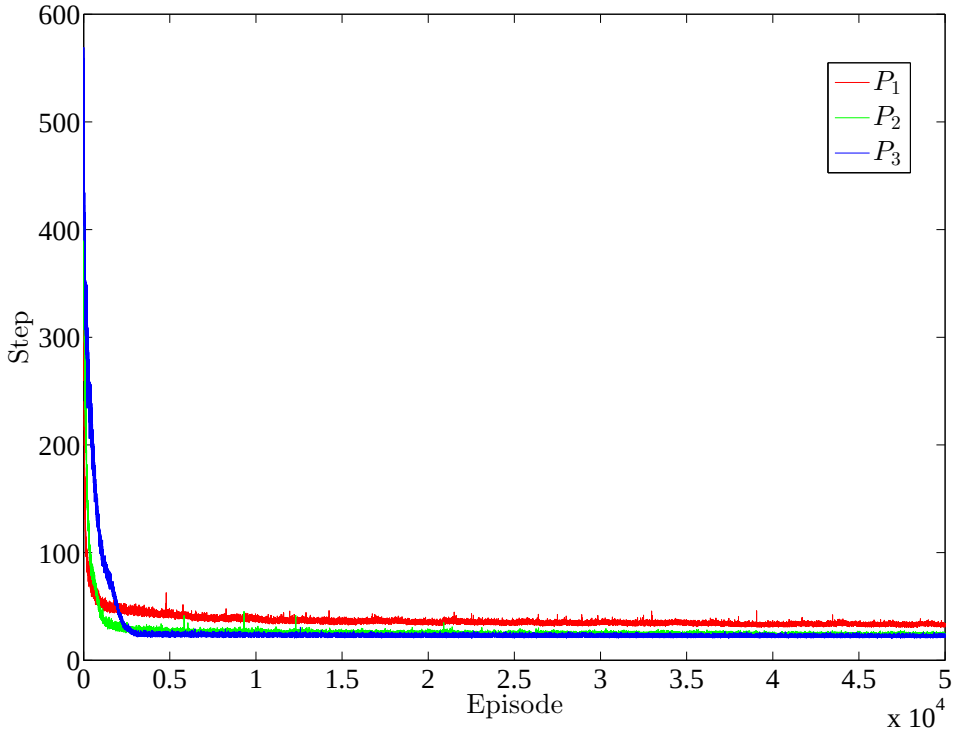

図 3.8: Q 学習での学習結果

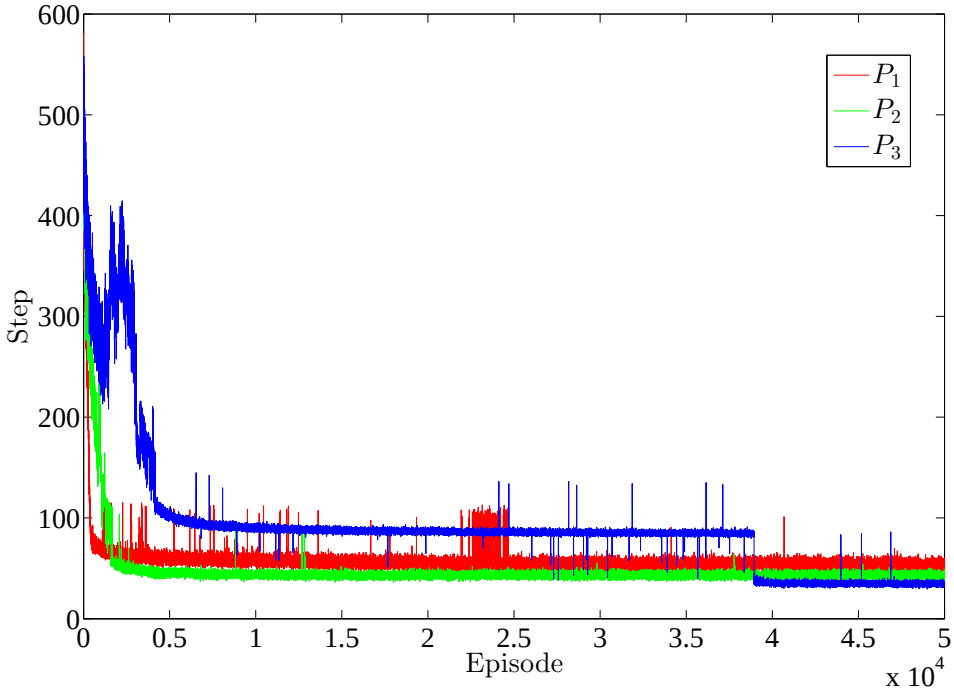

図 3.9: 離散型免疫型強化学習器での学習結果

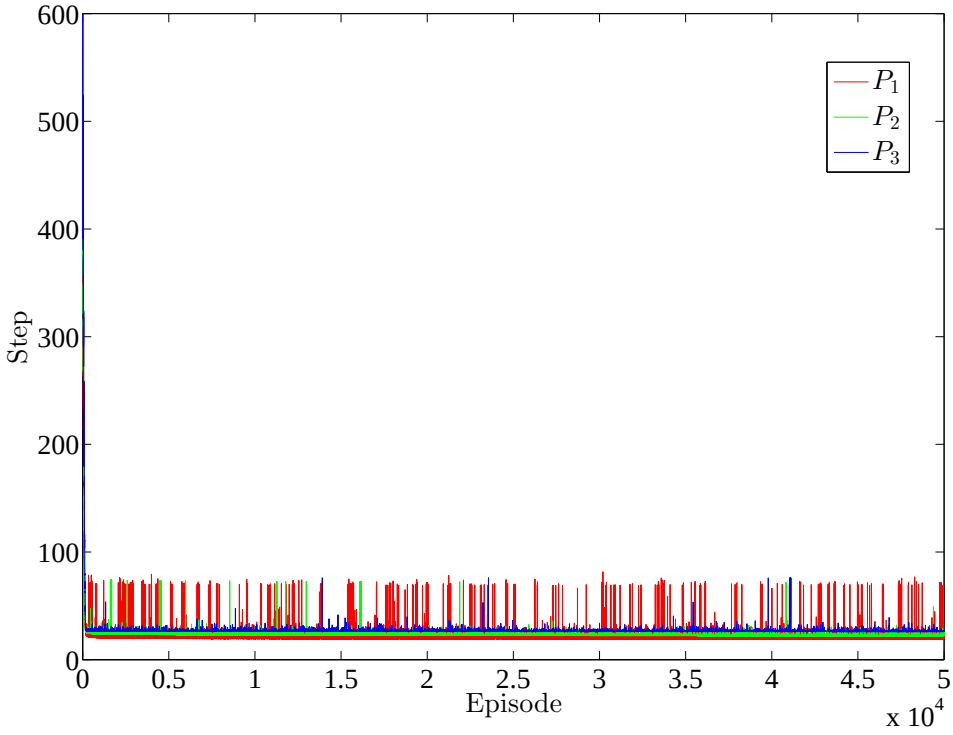

図 3.10: 提案手法での学習結果

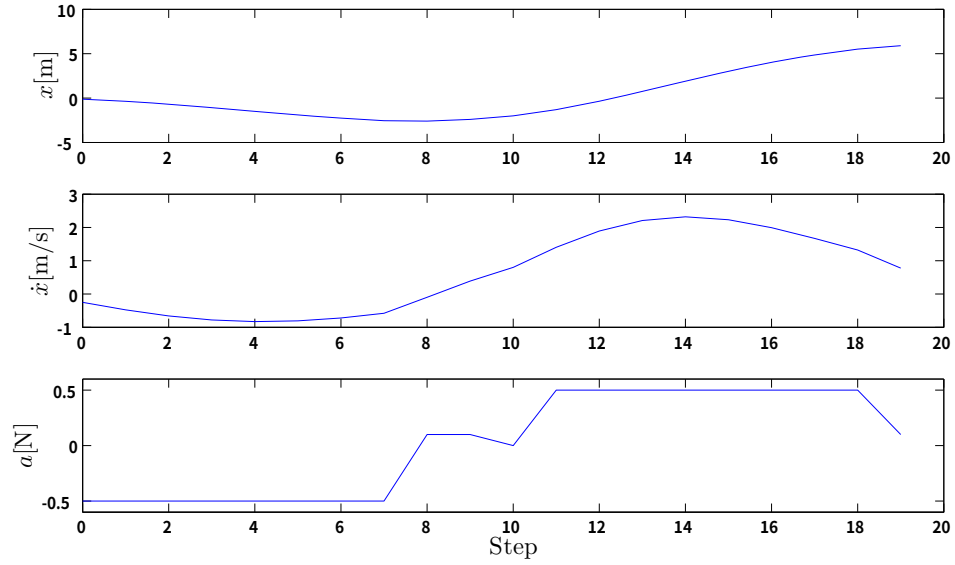

図 3.11: 提案手法のログ

提案手法の学習プロセスによって選択される行動の変化ついて，行動 (入力値) を行動を選択される確率で重み付けを行った値をプロットすることにより検討を行う (図 3.12，3.13，3.14)．学習開始直後では全域にわたって選択される行動が 0 に近い値となっている．これは，0 が選択される確率が高いのではなく，どの行動も等しい確率で選択される可能性があるためである．図 3.13 は学習途中である 100 エピソードのときのプロットである．データが追加されることにより，選択される行動が決定されつつある事が確認できる．図 3.14 は学習終了後の 450 エピソードのときのプロットである．大まかに $-4 < x < 1, -3 < \dot{x} < 0$ の範囲では反動をつけるために目標から遠ざかる負方向へのトルク入力，$-3 < x < 6, 0 < \dot{x} < 5$ の範囲では目標へ近づくための正方向のトルク入力，$6 < x < 8, 2 < \dot{x} < 4$ の範囲では目標で停止するための負方向のトルク入力となっている．以上から提案手法はクラスタリングのように自動的に範囲を区切り，その範囲での適した行動が学習できていることを確認できる．

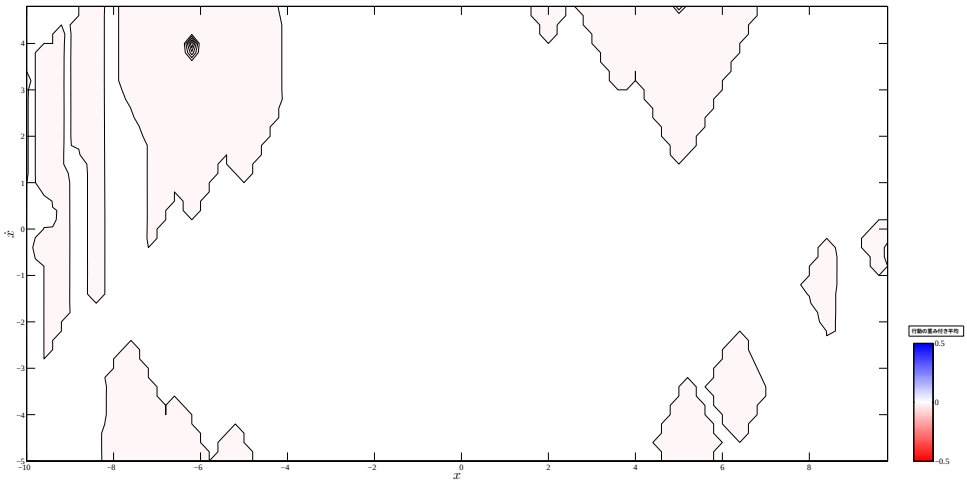

図 3.12: 学習直後 (1 エピソード) での行動の重み付き平均

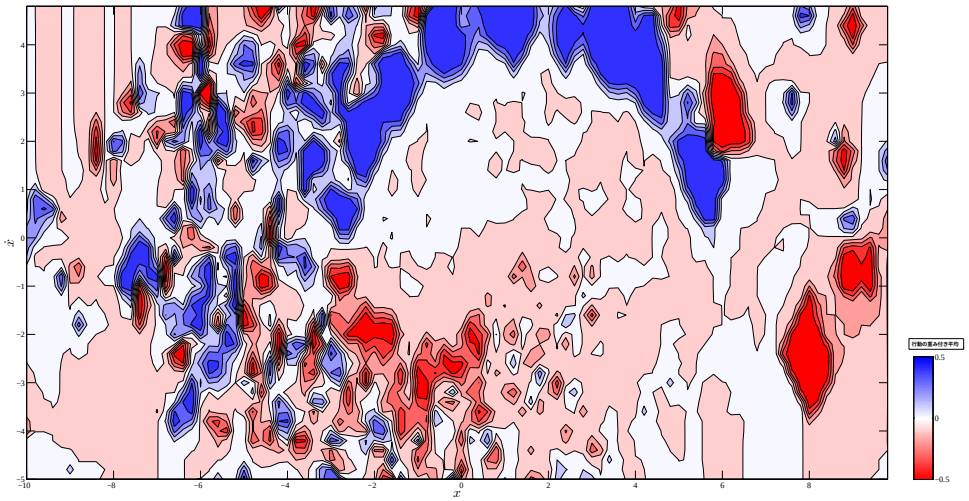

図 3.13: 学習中盤 (100 エピソード) での行動の重み付き平均

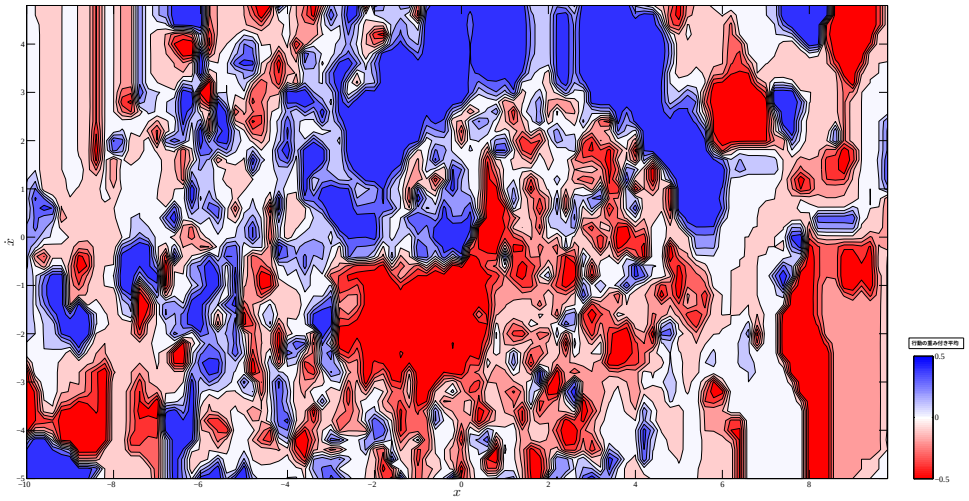

図 3.14: 学習終了後 (450 エピソード) での行動の重み付き平均

3.4.2 倒立振子の振り上げへの適用

本小節では，倒立振子の振り上げ制御に提案手法を適用して有効性の評価を行う．

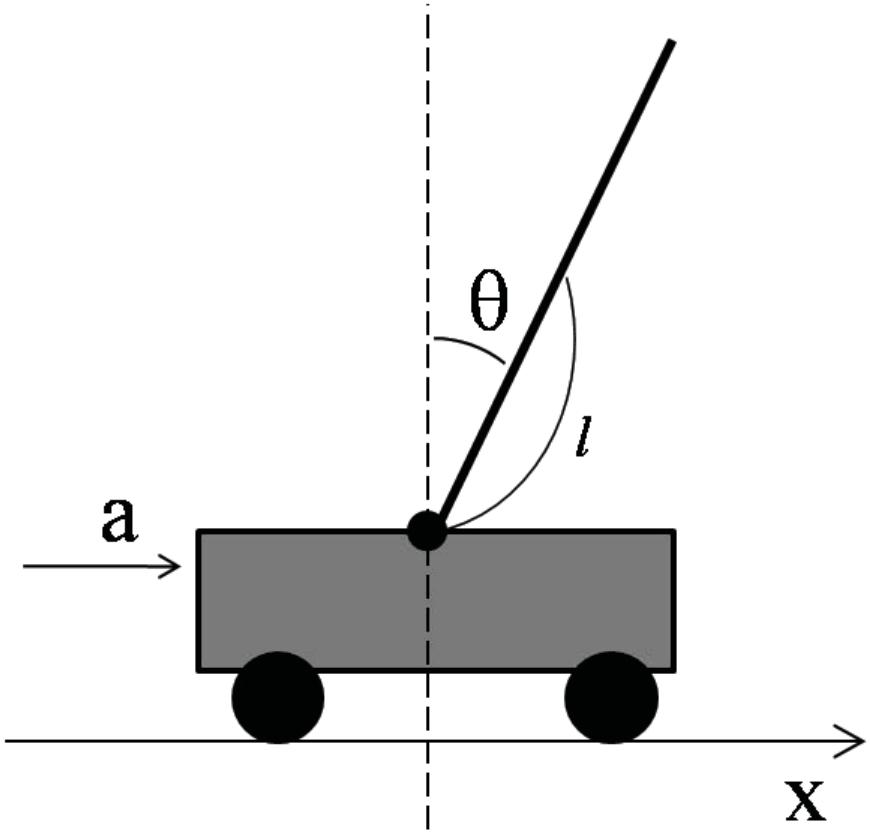

図 3.15: 倒立振子

a)　問題設定　倒立振子の構造を図3.15 に示す．倒立振子は平面上を移動する台車に振り子を取り付けた構造をしている．振り子は台車への取り付け点を中心として自由に回転運動をすることができる．しかし，倒立振子の角度を直接制御することはできない．このため，このシステムは自由度よりアクチュエータの数が少ない劣駆動システムである．次式に倒立振子の運動方程式を示す．

$$(M+m)\ddot{x} + ml\cos\theta\ddot{\theta} + D_x\dot{x} + ml\sin\theta\dot{\theta} \;=\; a \tag{3.14}$$

$$ml\cos\theta\ddot{x} + (ml^2+I)\ddot{\theta} + D_\theta\dot{\theta} - mgl\sin\theta \;=\; 0 \tag{3.15}$$

ここで，M は台車の質量，m は振子の重さ，l は振子の重心までの長さ，D_x は台車の摩擦，D_θ は振子の回転方向の摩擦，I は振子の回転モーメントである．

今回のシミュレーションでは振り子が真下を向いている状態 $(\theta = \pi)$ から行動リスト $A = [-10, 0, 10]$ の中から台車へのトルク入力値を 1 つ選択して，振り上げ動作 $(\theta = 0, \dot{\theta} = 0)$ の学習を行った．学習器の設計時には運動方程式や物理パラメー

表 3.4: 倒立振子シミュレーションの物理パラメータ

パラメータ名	値
M	1
m	0.1
l	0.5
D_x	0.0005
D_θ	0.000002
I	0.00002
Δt	0.01

表 3.5: 初期状態と目標状態

パラメータ	初期状態	目標状態
x	0	どこでもよい
$\dot{x}$	0	0 ± 0.5
θ	π	0 ± 0.5
$\dot{\theta}$	0	0 ± 0.2

タ (表 3.4) はわからないものとして，学習器には観測情報として状態 $[x, \dot{x}, \theta, \dot{\theta}]$ が与えられるが，それぞれの値には ± 0.1 の観測ノイズが加えられている．強化学習手法ではマルコフ性を満たすことを前提として構築されている学習手法である．このため，学習制御器に入力される情報は台車の位置や速度，振子の角度や角速度全てを入力する必要がある．シミュレーションに使用する倒立振子の可制御性，可観測性についての検討を付録 D.1 にまとめた．付録での検討はシステムを線形近似した場合についてであり，振り上げ制御の例では線形近似ができないため解析結果をそのまま当てはめられるものではない．非線形システムとしての解析は [48] などを参照されたい．学習試行においては各状態が目標状態に到達した場合に報酬値 10 を与え，1 エピソードの終了とした．台車の可動範囲は $-10 \leq x \leq 10$ とし，この範囲外になる場合は $\dot{x} = 0$ として台車を停車させた．ランダム選択における平均的な振り上げ所要ステップ数は 4000 ステップであったため，5000 ステップを経過しても目標状態に到達できなかった場合は報酬値を与えずに試行を終了してつぎのエピソードの学習を開始した．比較手法として離散型の免疫型強化学習器を用いた．サイトカインシグナルの初期値を $w_{ini} = 0.01$，学習パラメータとして $\alpha = 0.1$，$\beta = 0.2$，行動選択手法としてルーレット選択を用いた．離散状態の強化学習器と提案手法の結果は図 3.16 となった．

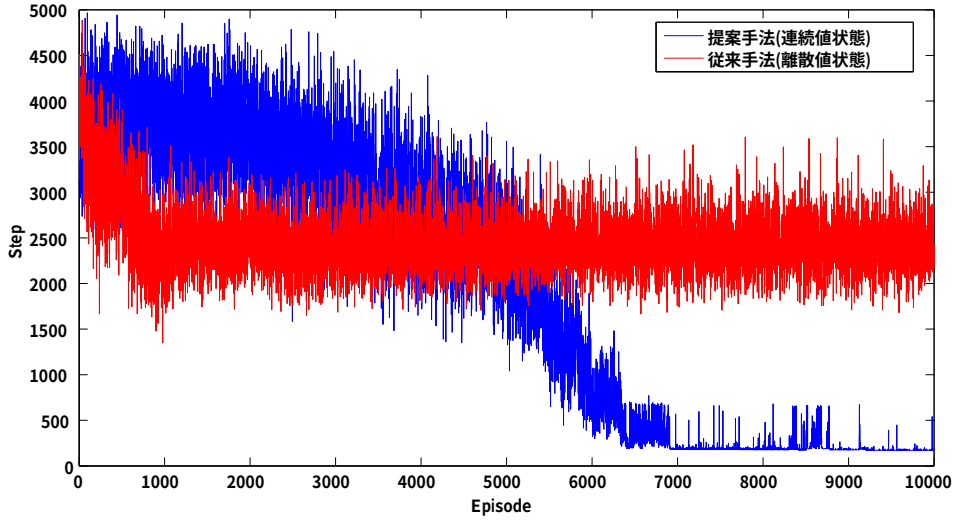

図 3.16: 振り上げ制御行動の獲得時間

両手法において学習初期のエピソードでは振り上げ動作に 4000 ステップ前後必要としている．離散型の免疫型強化学習器は 1000 エピソード付近まで学習効果が出ており，約 2500 ステップまで所要ステップを短縮しているがそれ以上の政策を学習することができていない．これは，離散化度合いが荒いため状態の切り分けができないためである．一方，提案手法では観測ノイズがのった場合でもエピソードを重ねることにより振り上げ動作を獲得することができた.

3.5 おわりに

離散値環境を元に構築された，免疫型強化学習器の抗原認識部分を再モデリングすることにより連続値環境に適用できるように改良を行った．改良を行った手法は行動の評価値の更新という意味では従来の離散値表現の強化学習手法と等価な更新が行われ，学習をやり直さなくとも行動選択を行う際に離散化度合いを動的に調整することも可能である．改良した手法を連続値で環境が表現される山登り問題および倒立振子の振り上げ問題に適用しシミュレーションによって検証を行った．提案手法は従来の強化学習器では離散化パラメータの設定によっては政策の学習に失敗する可能性のある連続値環境においても学習を行うことができた.

第4章　安定化制御における強化学習の報酬関数

4.1　はじめに

本章では安定化制御問題におけるモデルフリー型の報酬割り当て関数について議論する．はじめに，合理性定理を満たす報酬関数を安定化制御問題に適用した場合に安定化状態が維持できない報酬が割り当てられることを示す．その後に安定化制御問題において報酬割り当て時に満たすべき条件をあげ，報酬関数の設計を行う．本研究で提案した報酬関数は免疫型強化学習器のみではなく，同じくモデルフリー型強化学習の代表例である Profit Sharing においても有効である．このため，報酬関数の設計を Profit Sharing の枠組みにおいて行い，最後に免疫型強化学習器への適用を行う．

4.2　合理性定理を満たした報酬関数の問題点

本章では宮崎らによって提案された合理正定理を満たした報酬関数 (1.3) 式が安定化制御問題に適用できないことを示す．合理性定理の内容においては付録 A を参照されたい．

はじめに前節で示した補題 1 の検討を行う．この補題では，無効ルールを抑制するのが最も困難な構造として唯一の回帰的無効ルールが存在する場合としている．唯一の自己回帰ルールを含む構造では無効ルールが連続して選択され続ける可能性が最も高く，有効ルールの強化が最も行われにくくなる．このことは，意志決定が一定間隔で行われる MDP と任意時間で行われる SMDP では違いは発生しないため，この補題は問題ない．次に，補題 A.1 の検討を行う．この補題では回帰的無効ルールを抑制するための抑制条件について (A.1) 式としている．この補題でも補題 1 と同様のため，問題は無い．

これらの補題により求められた報酬関数例 ((A.4) 式) では 1 ステップで選択し

た行動が終了することを前提としている．このため，行動が終了するまでに数ステップを要するSMDP 環境では状態遷移の回数を最小とする様に学習が行われるため，目標到達まで必要とする時間が最小となるとは限らない．これは報酬の減衰がステップ数で行われているためであり，割り当てる報酬の減衰を行動を実行した時点で行うことにより解決が可能となる．

$$r(t) = R_0 \left(\frac{1}{S}\right)^{(end\_time-t)} \tag{4.1}$$

ここで，end_time は目標到達までに必要とした時間，R_0 は環境から受け取った報酬値を示す

4.3 報酬関数の設計

本節では，モデルフリー型強化学習器の安定化制御問題への適用法および報酬割り当て関数の検討を行う．そして，安定化制御問題における政策を獲得するための報酬の割り当て条件を検討する．その条件を使用して報酬関数の設計を行う．

4.3.1 セミマルコフ決定過程 (SMDP)

目標到達までの時間を最短化する問題では単一の状態にとどまる行動をとり扱う必要がなかった．一方で，目標状態を維持するタスクにおいては安定化状態内であれば同じ状態をとり続けても問題がない．しかし，マルコフ決定過程においては状態遷移しなくとも単位時間ごとに行動を必ず選択する必要がある．このため，状態分割が荒い場合ではその状態を維持するために有益ではない行動であっても直ちに他の状態に遷移するとは限らないため正しく行動の評価を行うことができない．すなわち，この場合では有効な行動と無効な行動が等しく自己回帰ルールとして見なされることになる．また，他状態へ遷移することがどの行動を実行しても変わらない場合では安定化に有効な行動を最後に選択してしまった場合はこの行動に対しての評価値は不当に下げられてしまう．改善策として図4.1 のように状態分割を細かくすることによりある程度行動の評価を区別しやすくすることができるがこれも完璧ではないうえ学習に多くの時間を必要とする．

本手法では行動選択を一定時間ごとに行わないSMDP で環境をモデル化する[49]．この方法も状態分割についてのトレードオフが完全に解決されるわけではな

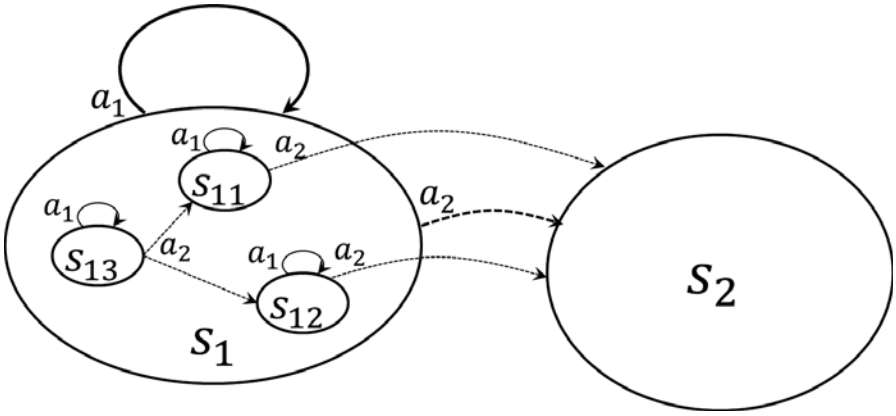

図 4.1: 状態分割の例

いが，自己回帰ルールを取り扱わなくてもよくなる．このため状態分割を比較的荒く設定することが可能となる．

4.3.2 報酬分配

モデルフリー型の強化学習法では状態価値推定値の更新を報酬を受け取った際に一括して行う手法のため，Q 学習などの逐次更新を行う手法では用いることのできないエピソードの継続時間を使用することができる．しかし報酬値としてエピソードの継続時間をそのまま採用した場合では非常に大きな報酬値が与えられることが多々ある．これは実際に計算を行う際に値のオーバーフローを引き起こしたり，状態価値推定値の初期値が非常に小さい場合は十分な解探索が行われない危険性がある．このため，本研究では与えられる報酬値は一定値とし，報酬を行動の時間から割り当てる．また，各エピソードは安定条件内から始まることを仮定し，図を元に報酬関数の検討を行う．

まずはじめに，以下の 2 つの状態遷移をして報酬を受け取ったときを考える．

State transition example(1)

$$x_a \to y_b \to z$$

State transition example(2)

$$x_a \to y_a \to x_a \to y_b \to z$$

ここで，x, y は安定状態，z は不安定状態，下付文字は選択した行動を示す．上記の例では y から z に遷移する行動 b を抑制することが目的となる．安定化制御問題

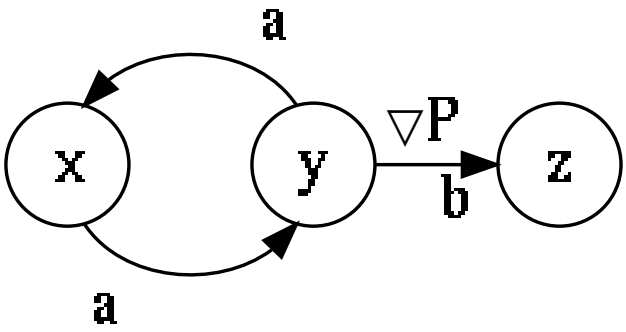

図 4.2: 状態遷移例

において報酬を受け取る直前にとった行動は選択すべきでない行動であるから報酬値の割り当ては 0 もしくはそれに近い値が適当である．一方，報酬を受け取る以前の行動は安定化状態を維持するのに貢献した行動であった可能性が高いので，エピソード終了に近い時間 t_b に選択した行動よりも初期状態に近い時間 $t_a(< t_b)$ で選択した行動に報酬値を多めに割り当てるのが妥当である．

$$r(t_a) > r(t_b)$$

これらの条件を満足する報酬割り当て関数 $(r(t))$ として報酬を受け取った時点から初期状態方向にみて割り当て報酬値が増加する関数形があげられる．

その例として図 4.3 の 3 種類があげられる．図 3 の横軸がエピソード継続時間で正規化した時間，縦軸が割り当て報酬値を示している．エピソードの継続時間が短い学習初期では，不安定状態に遷移する行動をできるだけ早く抑制したい．報酬を受け取る直前の行動は不安定状態へ遷移する行動であるので，これに割り当てる報酬を低くする．一方エピソード開始直後の行動は不安定状態へ直接遷移する行動ではないので，できるだけ報酬を割り引きたくない．この条件において報酬割り当て関数は type C は適当ではない．また，学習中盤においては学習が進むことによりエピソード継続時間が長くなる．この場合ではエピソード中盤での行動も安定化に寄与している可能性があるので，エピソード初期と同等の報酬を分

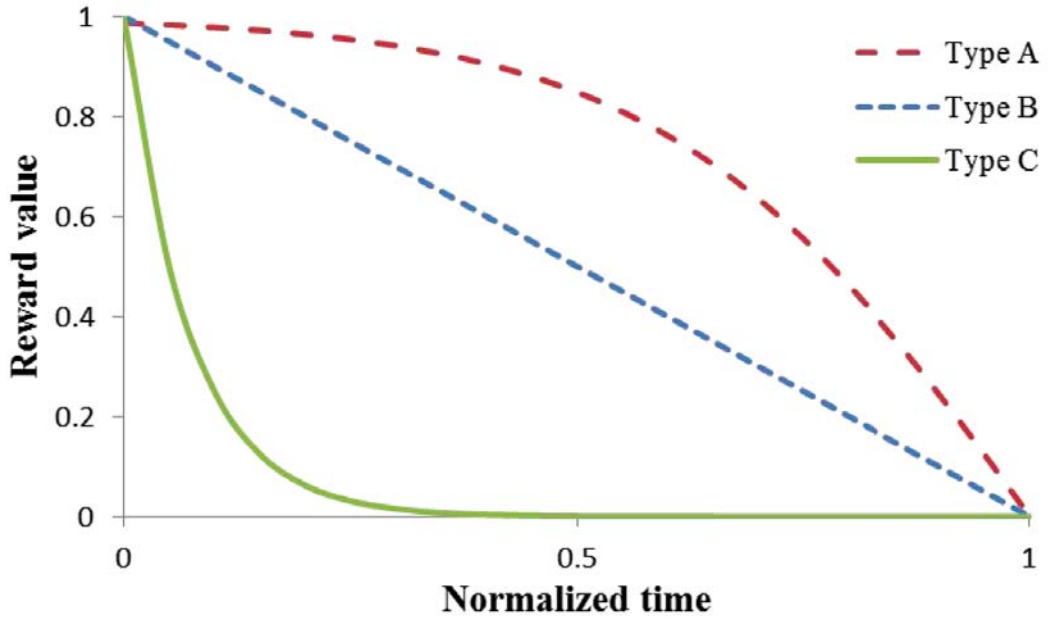

図 4.3: 報酬関数例

配したい．type A の関数では安定化状態が終了する直前の行動には報酬を分配せず，エピソード初期〜中盤にかけて報酬を多く分配できる．図 4.3 の type A 型報酬関数例として次式があげられる．

$$r(t) = 2\left(0.5 - \frac{1}{1 + \exp(-(t-1)/T_a)}\right) \tag{4.2}$$

ここで，t はシミュレーション時間で正規化した時間，T_a は関数の傾きを決定する定数である．

さらに学習が進むにつれエピソードの継続時間が長くなると同一の状態で複数回の行動選択が行われるようになる．しかし typeA 型の報酬関数では正規化した時間を用いて報酬割り当てをしているため安定状態を抜ける行動を区別することができなくなる．ここで任意の状態間を循環して遷移し続ける場合の行動選択回数を考える．循環する行動を a_{loop}，循環状態から外れる行動を a_{open} とすると，明らかに状態を循環する行動を選択する回数の方が多くなる $(Num(a_{loop}) > Num(a_{open}))$．また，$a_{loop}$ は a_{open} に対して必ず先に選択されるので，受け取る報酬値は必ず a_{loop} の方が大きくなる．これらのことから，行動評価値の和をとることによりこれらの行動の報酬値の差違を作ることができる．以上から，安定化を考えた場合にお

いて Profit sharing の Q 値の更新式は以下にかける．

$$Q(s_i, a_i) \leftarrow Q(s_i, a_i) + \alpha \left[\sum_t r(t|s_i, a_i) - Q(s_i, a_i) \right] \tag{4.3}$$

上記の報酬関数は $[0, 1)$ の範囲をとるので Q 値の初期値は $1 + \alpha$ に設定すればよい．これによってボルツマン選択やルーレット選択においても十分に解探索が行われる．

4.4 シミュレーションによる検証

本節では提案手法の有効性を以下の制御課題に適用し，検証を行う．

1 倒立振子の安定化制御問題

2 T 字型の倒立振子の安定化制御問題

3 Keepaway タスク

4.4.1 倒立振子の安定化制御問題

b) 問題設定 3.4.2 項にて用いた倒立振子系 (図 3.15) を用いた安定化制御においての検証を行う．倒立振子の運動方程式・物理パラメータは 3.4.2 項にて使用したものと同様である．今回のシミュレーションでは状態を $x = 10, \dot{x} = 10, \theta = 50, \dot{\theta} = 50$ で分割した．今回のシミュレーションでは振り子は倒立状態 $(\theta = 0)$ から，行動リスト $A = [-10, -1, -0.1, 0, 0.1, 1, 10]$ の中から行動を選択して，安定化制御の学習を行った．倒立振子が倒れるか台車が指定範囲から出た場合 (表 4.1) にエピソードを終了して初期状態から次エピソードを開始する．また，4000 ステップ以上倒立状態が続いた場合は安定化制御成功としてエピソードを打ち切って，次エピソードを開始する．

シミュレーションは提案手法の他に比較検討用に Q 学習および宮崎らの報酬関数を用いた Profit Sharing について行った．提案手法および Profit Sharing 法については安定化状態から不安定状態に遷移した場合に 1 の報酬，Q 学習では-10 の報酬を与える

表 4.1: 倒立振子の安定化制御における初期状態と目標状態

状態	初期状態	目標状態
[pm]x	0	0 ± 3 [m]
$\dot{x}$	0	-
θ	0	0 ± 0.1[rad]
$\dot{\theta}$	0	-

表 4.2: 学習結果の比較

学習手法	学習後の平均ステップ数	学習収束エピソード
提案手法	3500	60000
Q 学習	550	-
ProfitSharing	85	300

c)　学習結果と考察　提案手法と Q 学習，宮崎らの報酬関数を使用した Profit Sharing においてそれぞれ学習した結果を図 4.4，数値比較を表 4.1 に示す．また，それぞれ 100000 エピソードの学習後に得られた政策の例を図 4.5〜4.7 に示す．
宮崎らの報酬分配関数では図 4.4 の結果から安定化行動を獲得でできず，どちらかというと振子をできるだけ早く倒す政策が学習された．学習後の政策例の図 4.6 からもその傾向を読み取ることができる．これは環境から与えられる報酬が不安定状態に遷移した場合に正の報酬値が割り当てられ，その報酬を受けるための合理的な政策を学習したためである．このため，安定化制御問題において合理性定理を満たした関数を Profit Sharing の報酬関数に使用することは不適当である．Q 学習では図 4.4 の結果から徐々に安定化時間を延ばすことに成功しているが，シミュレーション試行中に安定化状態を長時間維持するだけの政策を学習することはできなかった．提案する報酬関数を用いた Profit Sharing 法ではほかの手法と比べより少ないエピソード数で安定化行動を獲得できた．図 4.8 は提案手法を使用して学習を行ったときに遷移をした状態遷移の一部抜粋である．特定の状態間を遷移しており，安定化状態を維持する政策を学習できていることが確認できる．

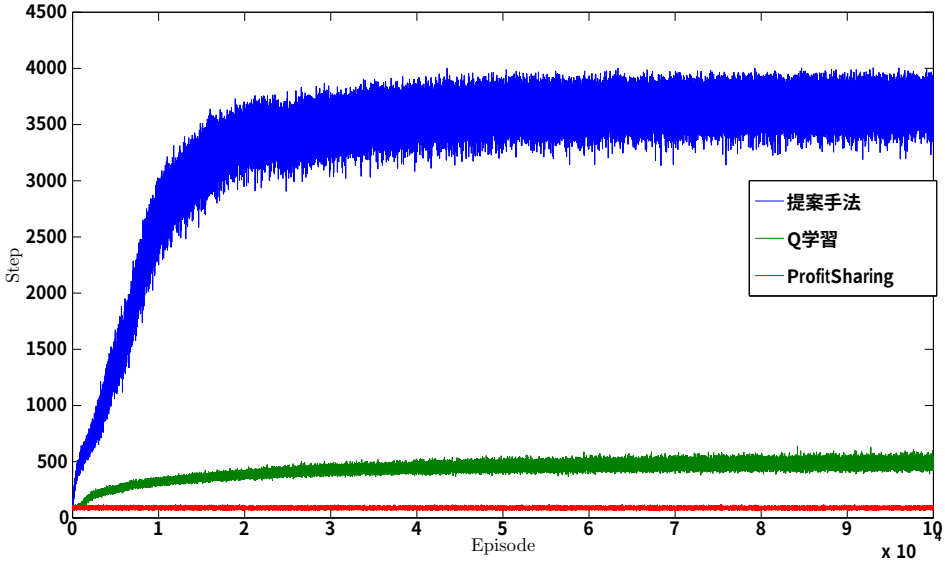

図 4.4: 倒立振子の安定化問題の学習時間比較

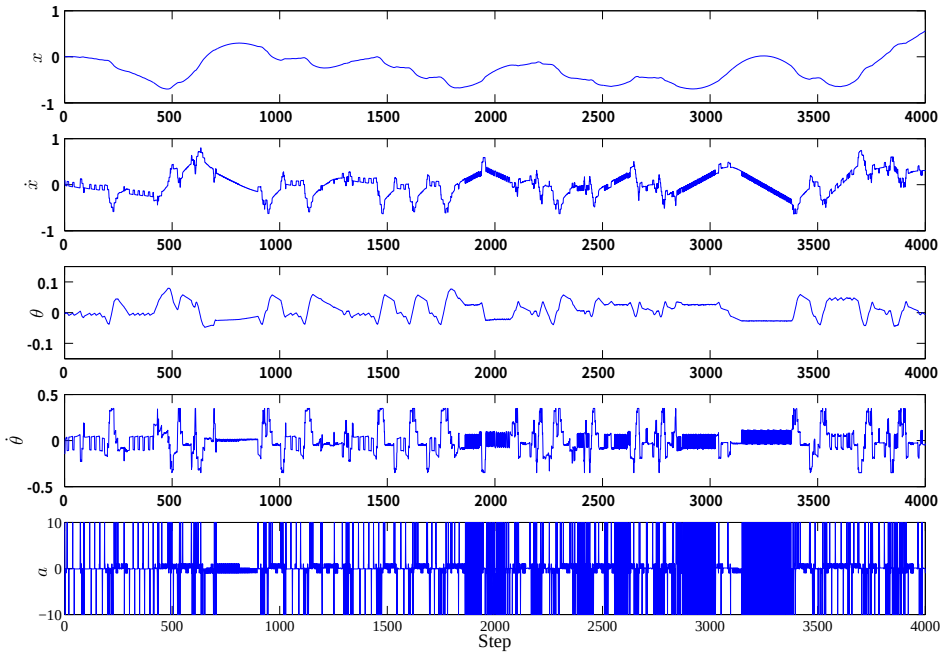

図 4.5: 提案報酬関数を使用した Profit Sharing の学習結果

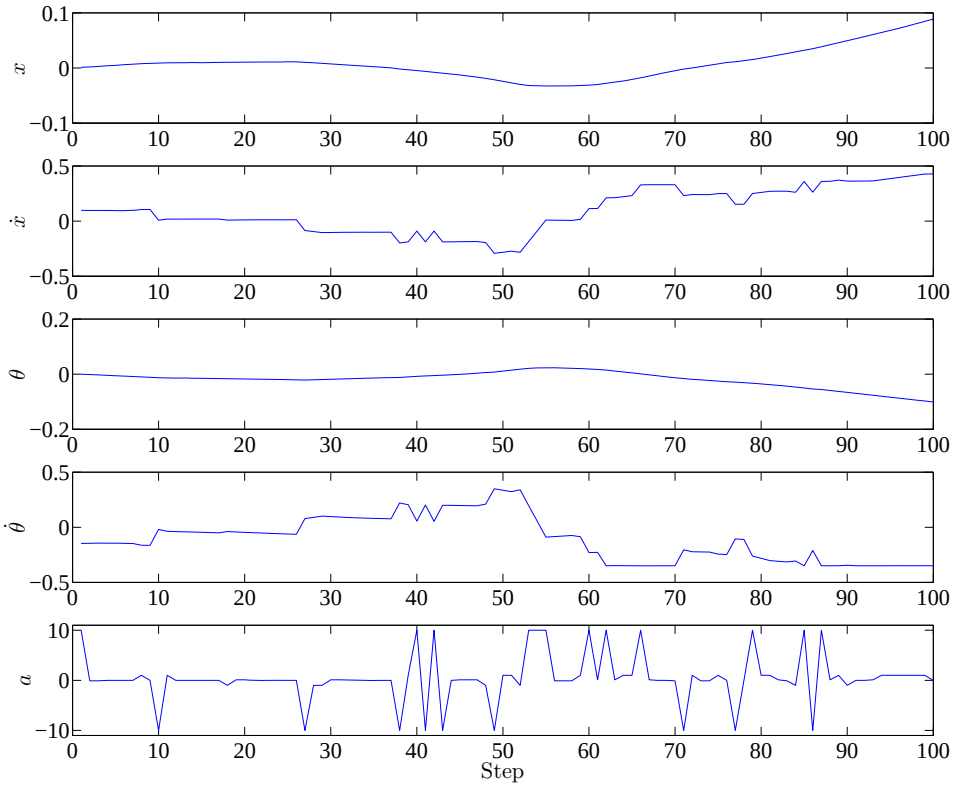

図 4.6: 宮崎らの報酬関数を使用した Profit Sharing の学習結果

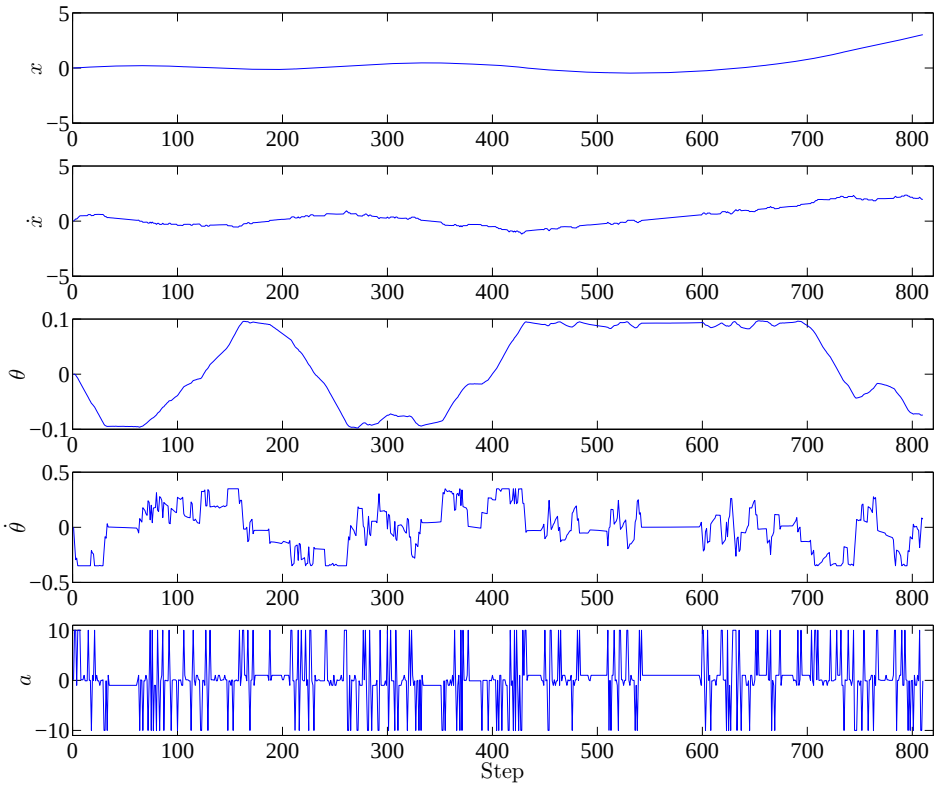

図 4.7: Q 学習での学習結果

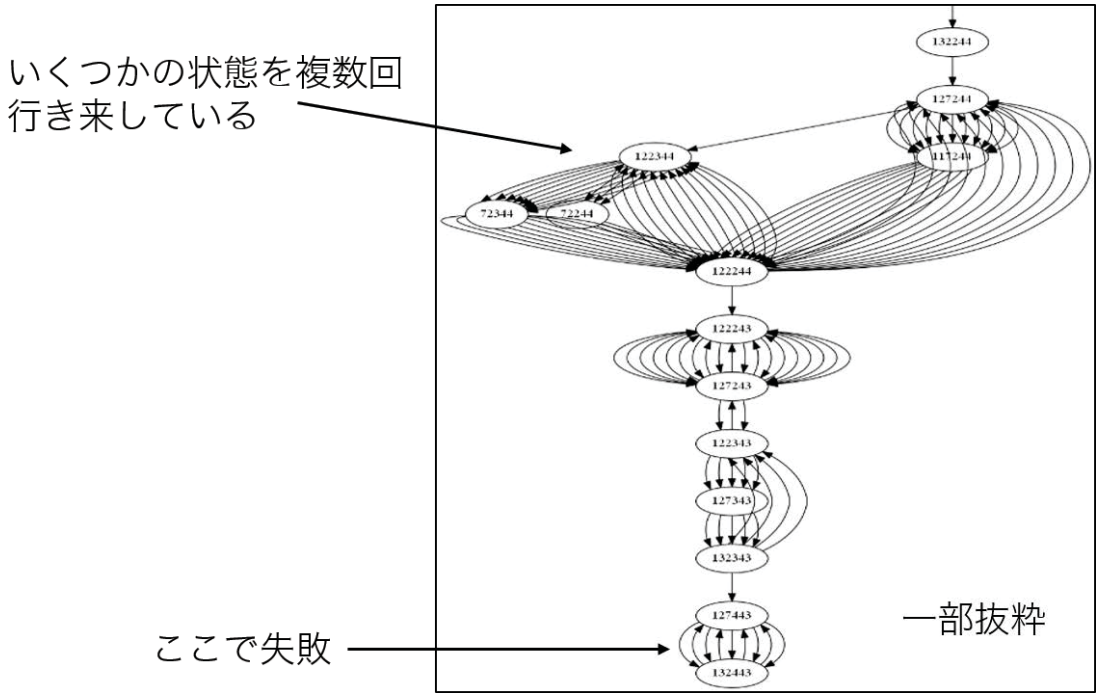
いくつかの状態を複数回
行き来している
ここで失敗
一部抜粋
132244
127241
122344
117240
72344
72244
122244
122243
127243
122343
127343
132343
127443
132443

図 4.8: 状態遷移例

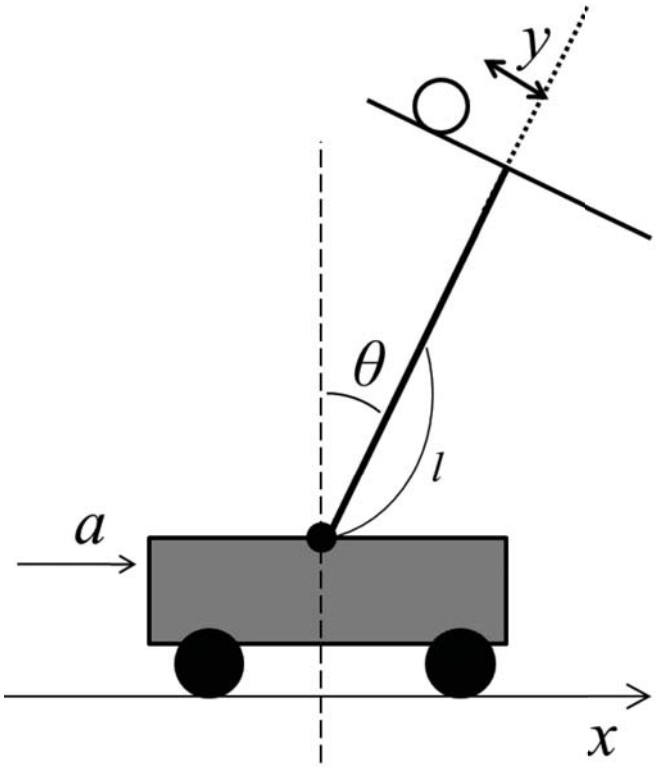

図 4.9: T 字型倒立振子

4.4.2 T 字型の倒立振子の安定化制御

本小節ではタスクの難易度を上げた場合においての学習結果の比較を行う.

a) 問題設定 図 4.9 に示すより複雑な系を持つ T 字型振子の上部にボールをのせた倒立振子 (以下，T 字型の倒立振子) の制御問題について適用して検証を行った．T 字型の倒立振子の制御目的は振子の上に配置されたボールを落とさないように制御することである．線形近似を行った T 字型の倒立振子の運動方程式を示す.

$$(M+m+m_b)\ddot{x}+(ml/2+m_b(r+l))\ddot{\theta}+m_b\ddot{y}+D_x\dot{x}=a \quad (4.4)$$

$$(ml/2+m_b(r+l))\ddot{x}+(ml/2+m_b(r+l)y^2+I)\ddot{\theta}+m_b(r+l)\ddot{y}+D_\theta\dot{\theta}$$
$$-g(ml/2+m_b(r+l)\theta-m_bgy=0 \quad (4.5)$$

$$m\ddot{x}+m_b(r+l)\ddot{\theta}+(m_2+I_b/r^2)\ddot{l}+D_b\dot{l}-mg\theta=0 \quad (4.6)$$

表 4.3: T 字型の倒立振子シミュレーションの物理パラメータ

パラメータ名	値
M	1
m	0.1
m_b	0.01
r	0.005
l	0.5
D_x	0.0005
D_θ	0.000002
D_b	0.0005
I	0.00002
I_b	0.0000001
Δt	0.01

使用した物理パラメータを表 4.3 に示す．T 字型の倒立振子の可制御性，可観測性についての検討を付録 D.1 にまとめている．学習器には状態として $x, \dot{x}, \theta, \dot{\theta}, y, \dot{y}$ を取り扱い，それぞれの値に観測ノイズとして $\sigma = 0.0001$ の正規乱数を付加している．それぞれの状態を $x = 10, \dot{x} = 10, \theta = 29, \dot{\theta} = 40, y = 5, \dot{y} = 5$ で分割した．今回のシミュレーションでは振り子は倒立状態 ($\theta = 0$) から，行動リスト $A = [-10, -1, -0.1, 0, 0.1, 1, 10]$ の中から行動を選択して，安定化制御の学習を行った．倒立振子が倒れる ($\mathrm{Abs}(\theta) > 0.1[\mathrm{rad}]$) か台車が指定範囲から出た場合 ($\mathrm{Abs}(x) > 3[\mathrm{m}]$) はエピソードを終了して初期状態から次エピソードを開始する．

b) 学習結果と考察 提案手法と Q 学習においてそれぞれ学習した結果を図 4.10 に示す．従来の報酬分配関数では安定化行動を獲得できず，振子をできるだけ早く倒す政策が学習された．一方で Q 学習ではある程度安定化時間を延ばすことに成功しているが，それもわずかである．提案手法はほかの手法と比べより少ないエピソード数で安定化行動を獲得できた．

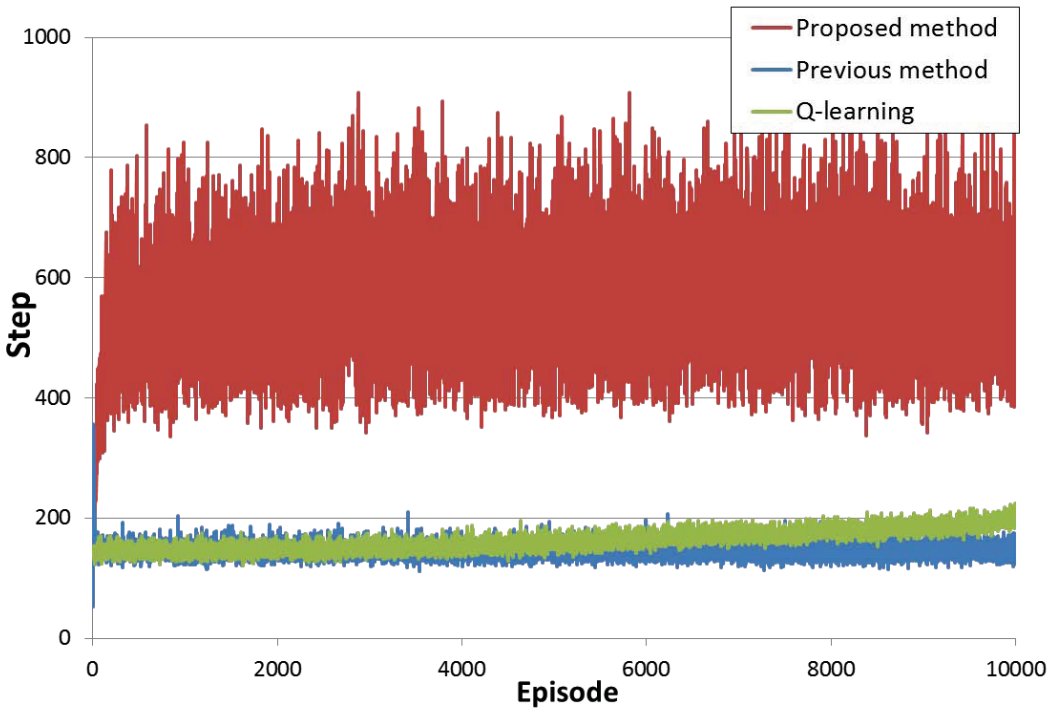

図 4.10: 学習収束速度の比較

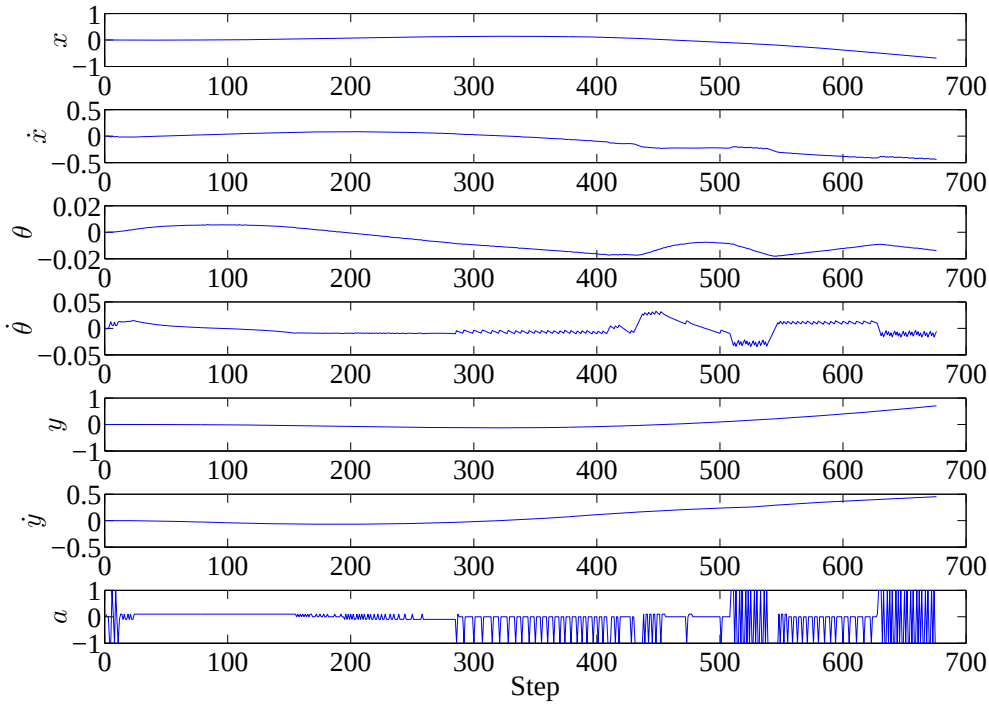

図 4.11: 提案手法での T 型倒立振子制御の学習結果

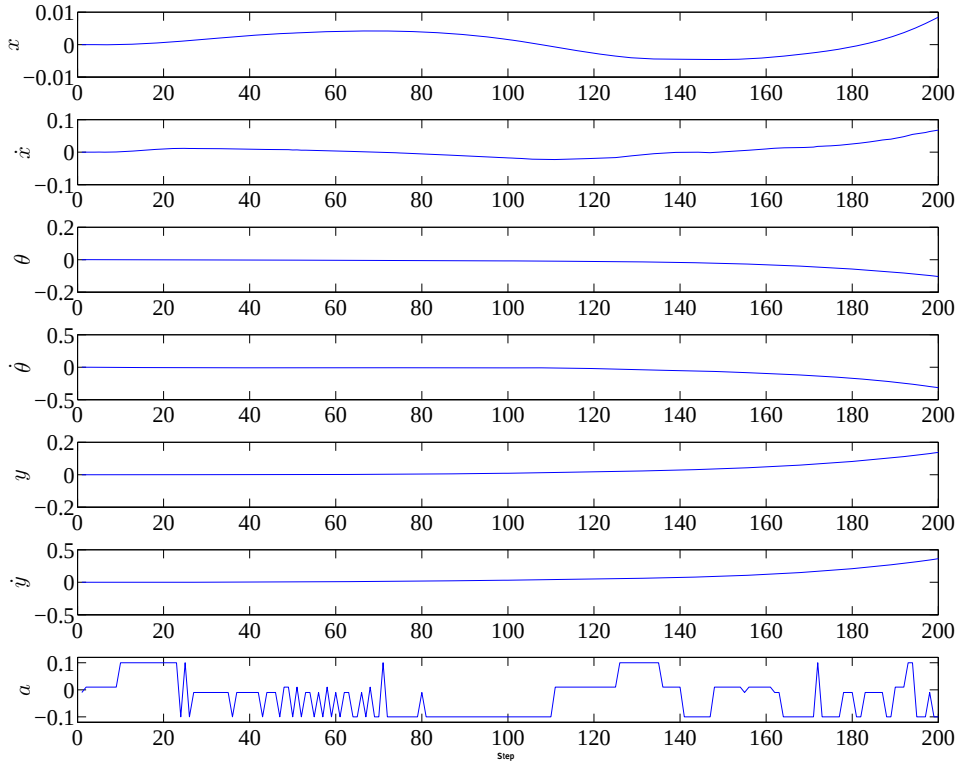

図 4.12: Q 学習での T 型倒立振子制御の学習結果

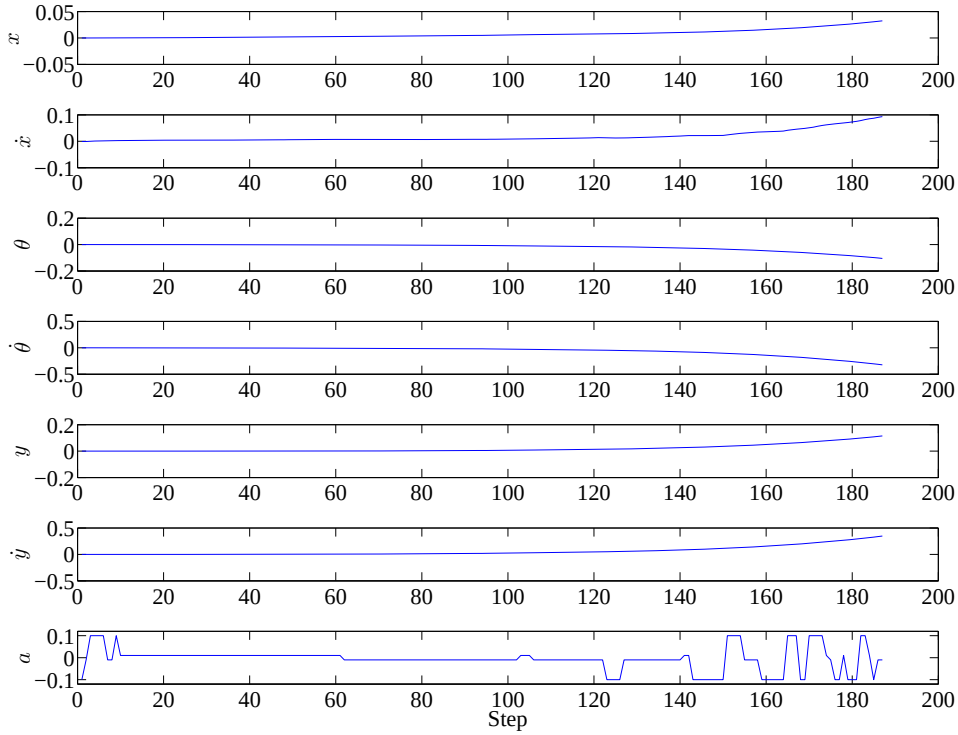

図 4.13: 宮崎らの報酬間数での T 型倒立振子制御の学習結果

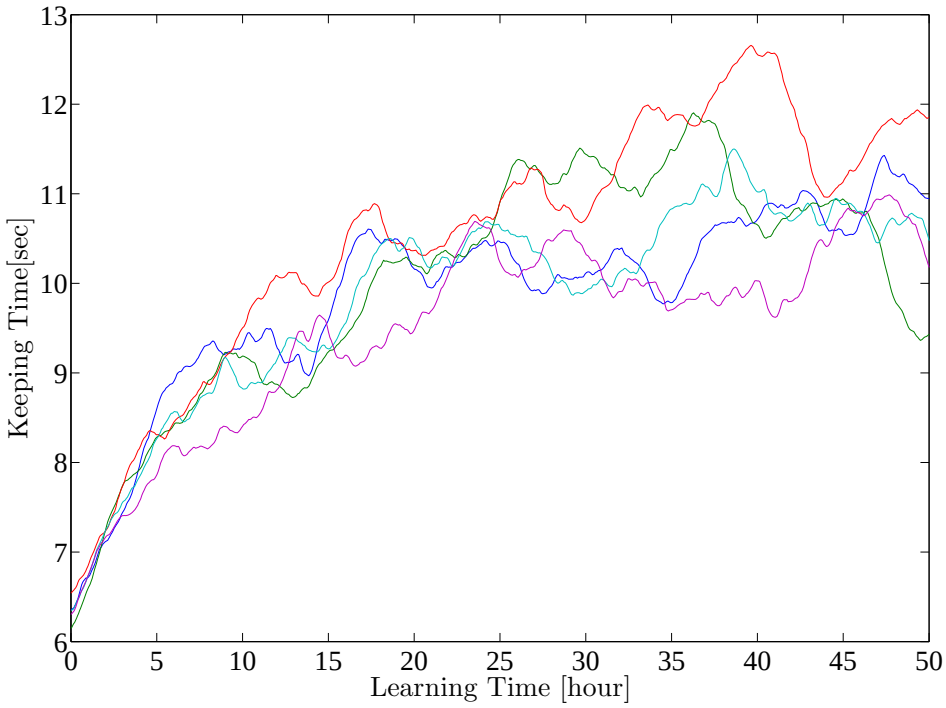

図 4.14: 3 対 2 の Keepaway タスクでの学習結果

4.4.3 Keepaway タスクへの適用

a) 問題設定 Keepaway は robocup 2D シミュレーションリーグのシステムを利用した強化学習のベンチマーク問題の一つである．このタスクは敵チームにボールをとられないように自チーム内でパスをつなぐことである．このタスクについての詳細な説明は付録 C を参照されたい．今回の検証では味方ロボットが 3 台，ボールを奪いにくる敵ロボットは 2 台とした．ロボットの動作範囲は 20m × 20m 内である．エピソードはボールを敵に奪われるか，動作範囲外にボールがでた場合終了となる．学習の各試行ごとにスタート時にボールを保持しているロボットが変更される．

b) 学習結果と考察 それぞれ 10 回ごとの試行を行い，学習時間（実時間）あたりの継続時間の結果を図 4.14 に示す．安定化を考慮した報酬関数を用いることによってボールの保持時間が延びていることが確認できる．

4.5 免疫型強化学習器への適用

4.5.1 アルゴリズムの修正

前章にてあげた報酬割り当て条件を使用して以下の様に免疫型強化学習器のアルゴリズムを構築する.

1 エージェントの状態が s_i の場合，Th 細胞群から各 B 細胞へのサイトカインシグナル w_k，状態 s_i における B 細胞の活性度 m_k を取得する

2 行動選択における B_k の評価値を $v(k)(= m_i \times w_k)$ として行動選択を行い, 抗体を生成する B 細胞を決定する

3 選択された k 番目の B 細胞によって抗体 $Ab(s_i, k)$ を生成し，その時間を抗体情報として記憶する.

以上の処理を繰り返して B 細胞の選択，抗体の生成を行い状態遷移をする.

エージェントが目標状態から不安定状態に遷移した場合，Th 細胞群のサイトカインシグナル w_k を更新する．サイトカインシグナルの更新は次式を用いて行う.

$$w_k(s_i) \leftarrow w_k(s_i) + \alpha(r_k(s_i) - w_k(s_i)) \tag{4.7}$$

$$r_k(s_i) = \begin{cases} r(s_i, a_k) & : \text{If } A_b(s_i, k) \text{ exists} \\ \min_{a_i \in A} r(s_i, a_i) & : \text{If } A_b(s_i) \text{ exists} \end{cases} \tag{4.8}$$

ここで，$\alpha(0 < \alpha < 1)$ は学習率を表している．更新は遷移した状態全ての w について行われ，更新に使用された抗体は消滅する.

4.5.2 倒立振子の安定化制御問題での検証

提案をした安定化制御における免疫型強化学習器の有効性を倒立振子系 (図 3.15) のシミュレーションによって検証を行った.

a) 収束速度の比較 提案した免疫型強化学習器と強化学習器において有名な ProfitSharing および Q-learning と学習結果の比較を行う．Q-learning は安定化状態から不安定状態に遷移した場合 $R = -1$ の罰報酬を与え，その他の手法は $R = 1$ の報酬を与えた．それぞれの手法ついて 10 セットの試行を行い，エピソードごと

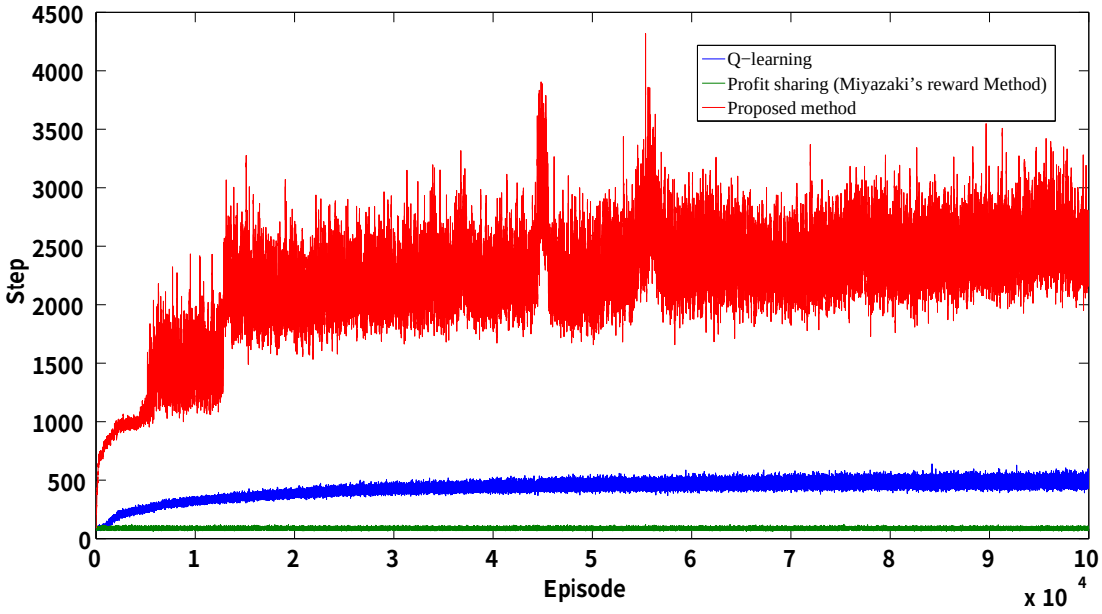

図 4.15: 倒立振子の学習収束時間比較

の平均ステップ数を図 4.15 に示す．この図の縦軸は安定化状態を維持することのできたステップ数，横軸はエピソード数を示している．

提案手法はエピソードを重ねる毎に安定化状態を維持することができているが，ProfitSharing では安定化時間を長くすることができていない．Q-learning では安定化状態を短時間だけのばすことができているが，さらなる学習時間を必要としている．

b) **外乱による学習性能への影響** 実環境に学習手法を適用した場合に以下にあげる外乱による影響が懸念される．

- 観測ノイズによる影響
- 初期偏差による影響

これらは制御対象が実際の適用環境に設置されるまで知ることができない．提案する強化学習器を用いることによって環境に左右されずに一定のパフォーマンスを得ることができることを検証する．

はじめに，観測ノイズの有無によるシミュレーション結果を図 4.16 に示す．この図の結果より観測ノイズの有無にかかわらず同等の学習結果を得ることができている．これは，観測ノイズの影響により若干の観測状態の混同が起きているが，

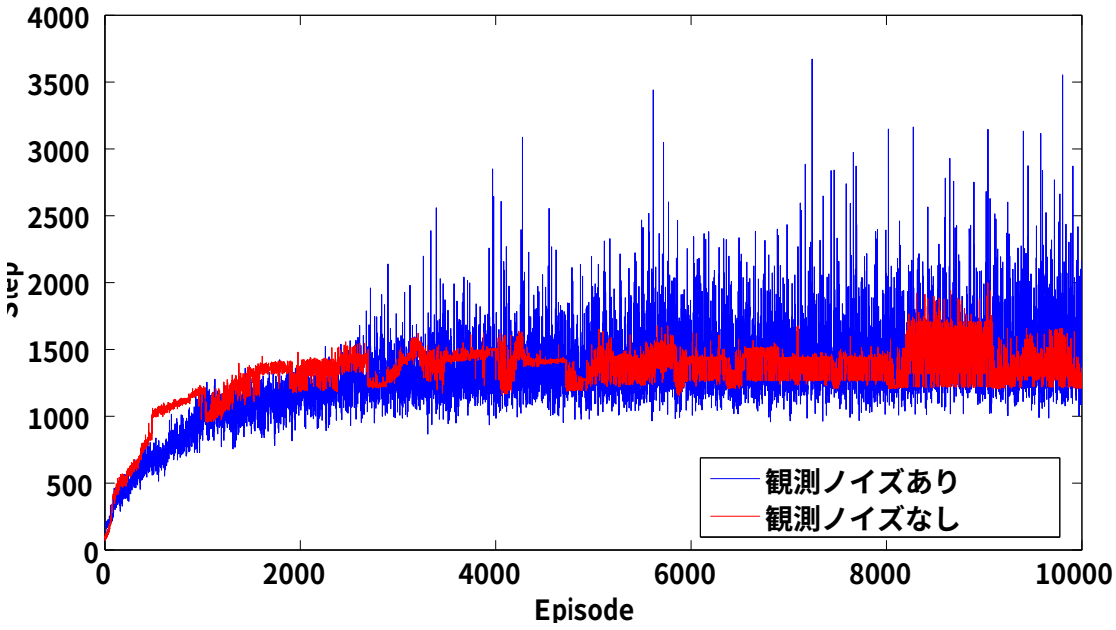

図 4.16: 観測ノイズを含んだ環境における学習収束速度の比較

SMDP によるモデル化を行っているため状態分割を荒くすることができたため大きな影響を受けなかったためである．

次に，初期偏差として台車の路面が $\varphi = [0.2, 0.1, 0.01][rad]$ だけ傾いている環境においてのシミュレーション結果を図 4.18 に示す．この図の結果より初期偏差があることにより学習の収束ステップ数の違いが見られるが，どの傾きにおいても安定化状態を長くする方策を学習することができている．このような環境では路面の傾きに応じて台車に適切なバイアスを加えなければならない．今回のシミュレーションにおいて選択できる行動が離散化度合いが荒かったため適切な入力トルクを選択できなかったためである．

4.6 おわりに

安定化制御問題におけるモデルフリー型の強化学習法の報酬の割り当て方について考察を行い，報酬関数の一例を示した．提案した報酬関数はシミュレーション時間のみによって報酬を分配するので，与える報酬値を変える必要が無い．

今後の課題としては倒立振子の振り上げ制御器 [50] と組み合わせることにより相反する一連のタスクを達成することができる学習器を構築することである．

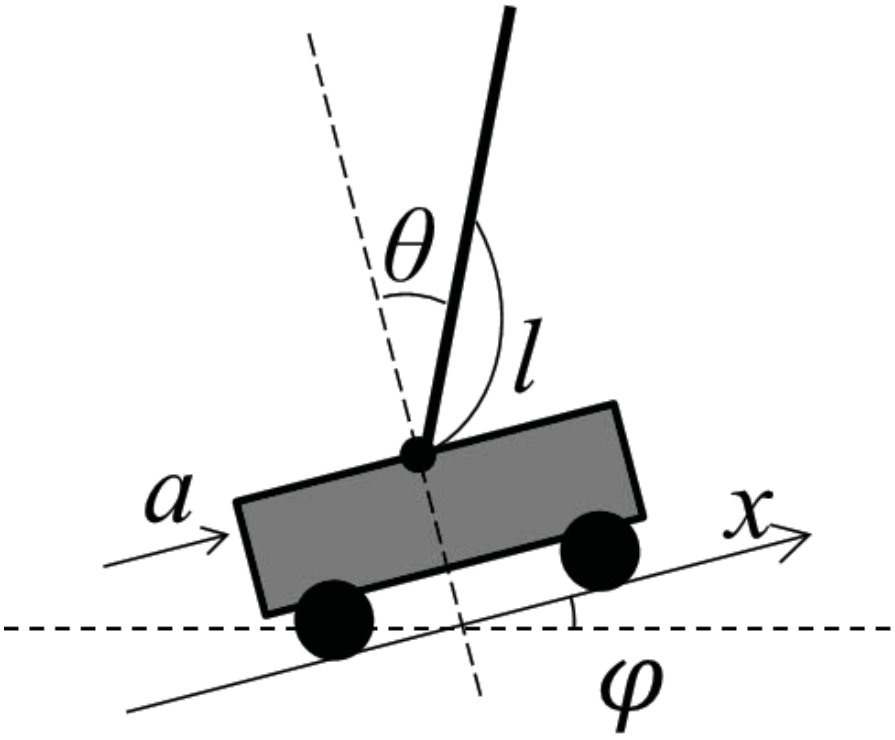

図 4.17: 初期偏差 (路面の傾き) がある倒立振子環境

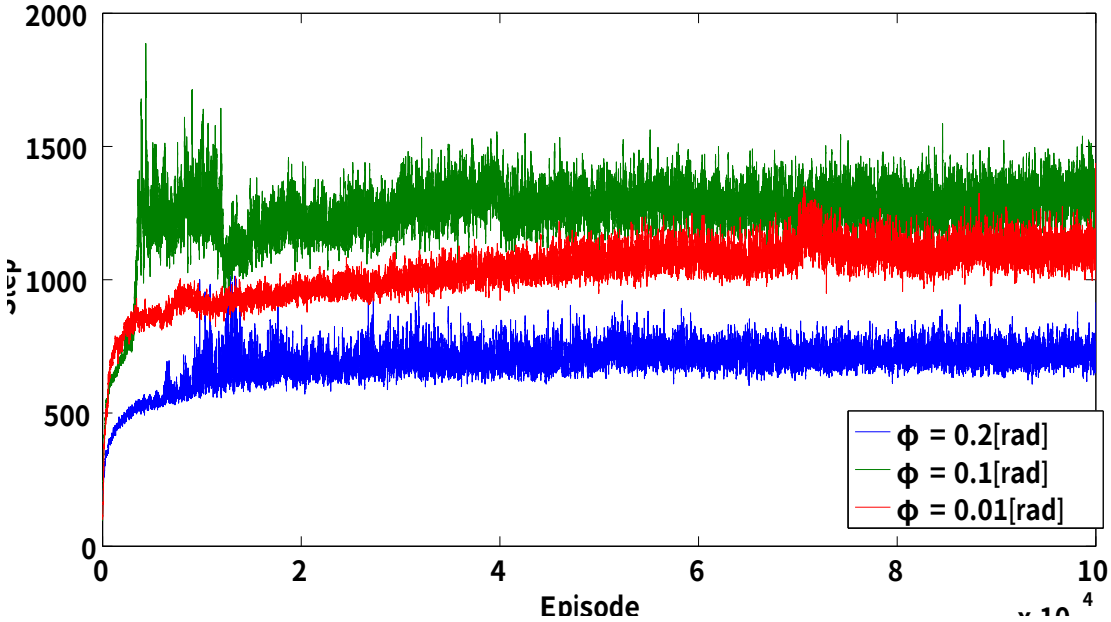

図 4.18: 初期偏差がある環境における学習収束速度の比較

第5章　結論

5.1　研究成果のまとめ

本研究では，自律ロボットの幅広い応用を実現する上で重要となる学習・記憶機構の効率的な構築を目的として主に免疫型強化学習法の改良を行い，シミュレーションによってその有効性を評価した．

自律ロボットにはハードウエアの設計技術やセンシング技術，環境認識技術などのさまざまな課題があり，これらの技術開発のために国際ロボット競技大会のRoboCupが開催されている．この目標を達成するには多くの研究課題が存在するが，特に行動選択においては，対戦相手の振る舞いは事前にわからないなか，試合に勝利するといった最終目標を達成する必要があり，これにはリアルタイムに行動を決定・調整し続けることが必要であり，興味深い研究テーマである．

一方，システムに学習・記憶記憶機構を持たせることにより，これまで困難とされていた抽象的な目標への追従，未知環境や複雑な環境へのシステムの適応などを可能とする，知能システムに関する研究が盛んに行われている．しかし，一般的にシステムや環境が複雑であればあるほど学習に長い時間を必要とし，自律ロボットへの搭載が困難となっていた．これに対し，生物の働きや進化の仕組みなどを工学的にモデル化し，学習機構として応用に関する研究も多く行われている．本研究ではその中で獲得免疫系の働きを中心に工学モデル化した免疫型強化学習器に着目し，この学習器の応用範囲の拡張を行った．

以下に各章で得られた研究結果を要約する．

第2章では本研究で取り扱う生物が備えている免疫系の説明を行った．免疫系には複雑な機能を，個別の免疫細胞が役割分担をしながら連携動作することにより人体の健康を維持している．本研究では免疫機構のうち，獲得免疫と呼ばれるシステムに着目し，その働きを工学モデル化した免疫型強化学習法について説明をした．免疫型強化学習器は従来の強化学習法と比べ，内部パラメータの初期値によらず，準最適解を短時間で学習することが可能である．このことを，強化学

習の中でも高速な学習収束速度を持つProfit Sharingのアルゴリズムとの比較から免疫型強化学習法がこれらの特徴を持つことを示した．

第3章では，連続値環境を前提とした免疫型強化学習法の拡張方法を提案した．連続値環境に拡張する上で，獲得免疫系の働きを見直し，Th細胞の抗原認識作用および記憶機構を再モデル化することにより連続値環境への拡張を行った．拡張したアルゴリズムが従来の離散型免疫型強化学習法の更新方式と等価であることを示すことによって離散値型の強化学習と同等の学習収束速度得ることができることを示した．さらに状態表現を記憶した情報との距離に用いることにより，離散型強化学習法で問題となっていた状態分割問題を解決し，学習途中でも状態分割間隔を変化させることができることを示した．この提案手法を山登り問題と倒立振子の振り上げ制御問題のシミュレーションに適用し，従来の代表的な強化学習法と比較を行った．その結果，従来の強化学習方式よりも妥当な解を高速な学習速度を得ることが確認された．

第4章では，モデルフリー型の強化学習法において多く持ち入れられてきた報酬割り当て関数が安定化制御問題へ適用できないことを示した．従来の報酬関数は環境から成功報酬を与えられることを前提として理論的な考察からより短時間で報酬が獲得できるように報酬関数の設計が行われていた．このため環境から罰則報酬が与えられる場合においても，罰報酬を可能な限り短時間で得るようなタスク達成のための最悪の政策を学習する可能性があった．このため，安定化制御問題においての環境情報の取り扱い方法や求められる報酬関数の条件の検討を行った．得られた条件から安定化制御問題では初期状態から報酬を得た状態に対して報酬が減少する関数が妥当となることを示し，Profit Sharingおよび免疫型強化学習において有効な報酬割り当て関数の一例を提案した．提案する報酬関数を用いて倒立振子の安定化制御・RoboCupサッカーシミュレーションリーグのサブ問題であるKeepawayのシミュレーションに適用した．その結果，提案した報酬関数を用いることにより，モデルフリー型の学習方式の利点である高速な学習収束速度を達成することが確認された．

5.2 今後の課題

本論文では免疫型強化学習器の応用範囲の拡大について取り扱ってきた．この手法を実際の自律ロボットに適用する際にさらに解決すべき問題について述べる．

第 1 にマルチエージェントシステムにおいての学習手法を確立することがあげられる．RoboCup サッカーやレスキューロボット，荷物運搬タスクなどにおいては複数台のロボットが協調して 1 つの問題解決に取り組んでいる．マルチエージェントの場合は学習主体以外も同時に学習が行われるため，ある時点で有効であった行動がすぐに無効な行動となることがある．これは一種の動的環境問題とも考えることができ，併せて検討すべき問題である．

付 録A　合理性定理[1]

本章では，宮崎らによって提案された学習結果に無効ルール (経路) が含まれない条件としての合理性定理 [1] を説明し，これを満たす報酬関数例について述べる．

A.1　準備

本項では用語の説明を図 A.1 に示した x, y, z の 3 状態が存在する環境を用いて説明する．この環境ではそれぞれの状態につき a, b の行動をとることができ，y から x に直接遷移した場合に報酬が得られる (▽マーク)．図 A.1 の環境でエージェ

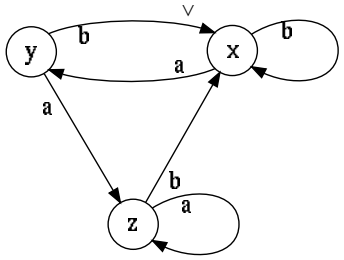

図 A.1: サンプル環境

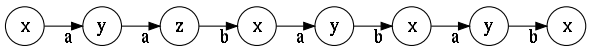

図 A.2: 状態遷移例

ントが $x_a, y_a, z_b, x_a, y_b, x_a, y_b$ と行動した場合を考える (図 A.2)．ここで状態の下付文字は各々の状態で選択した行動を示す．この行動セットでは報酬を 2 回受け取っているので，x_a, y_a, z_b, x_a, y_b と x_a, y_b の二つのエピソードに分けることができる．

Profit Sharing[30] や免疫型強化学習器 [51] などの非ブートストラップ型の強化学習法ではエピソード単位で政策の強化を行う．これらの手法では Q 値の更新に環境から受け取った報酬値とそれを受け取るまでの時間ステップを元にした報酬関数を利用する．ここで MDP 環境を仮定すると，時間は離散値となるので報酬関数 f_i によって報酬から i ステップ前の報酬値を参照する．長さ l のエピソード

$(r_l \ldots r_i \ldots r_2 \bullet r_1)$ に対して，ルールの重みを $w_{r_l} = w_{r_i} + f_i$ によって強化する場合について考える．

あるエピソードにおいて，同一感覚入力に対して異なるルールを選択している場合，その間のルール系列を迂回系列という．たとえば図 A.1 の環境におけるエピソード x_a, y_a, z_b, x_a, y_b では，迂回経路 y_a, z_b, x_a が存在する．迂回経路上のルールでは報酬の獲得に寄与しない可能性が有る．現在までの全てのエピソードで，常に迂回系列上にあるルールを無効ルールと呼び，それ以外を有効ルールと呼ぶ．無効ルールと有効ルールが競合して存在する場合，明らかに無効ルールを強化すべきではない．

A.2 無効ルールの抑制定理

本項では無効ルールの抑制を保証する定理を導出する．無効ルールの抑制とは，無効ルールがそれと競合する有効ルールを抑えて強化されることを防ぐことである．まず，無効ルールを抑制するのが一番困難なルールの競合状態を選ぶ．ここで，二つの競合構造 A と B について，A において無効ルールを抑制できる報酬関数のクラスが B のそれに包含されるとき，A は B よりも困難であるという．次に，もっとも困難な構造に対して，無効ルールを抑制するための報酬関数の必要十分条件を求める．最後に，任意のルールの競合状態に拡張をする．

補題 1 (最も困難な構造). 唯一の回帰的無効ルールの抑制が最も困難である．

証明 1 (補題 1 の証明). 証明のために次の言葉を定義する．一つの感覚入力に対して，適応可能なルールの数を**競合数**，可能な状態遷移の総数を**枝分かれ数**とする．

簡単のため有効ルール数 $L = 1$ とする．それ以外も全く同様である．明らかに，無効ルールが強化される回数が多いほど，無効ルールを抑制できる報酬関数の集合は小さくなり，無効ルールの抑制はより困難となる．よって，強化される回数の大小のみを考えれば十分である．枝分かれ数の小さい順に可能な競合構造を数え上げる．

1 枝分かれ数が 1 の場合
競合数が 1 なので，明らかに困難では無い．

2 枝分かれ数が 2 の場合
競合数が 1 の場合は先ほどと同様．競合数が 2 の場合を考える．回帰ルール

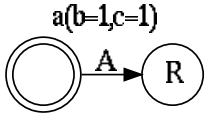

図 A.3: 枝分かれ数 1 の場合

を含む場合とそうで無い場合とに分けられる．ここで A を有効ルール，B を無効ルールとする．任意のエピソードで，1 回の A に付き B は繰り返し実行される可能性がある．従って，回帰ルールを含む方が困難である．

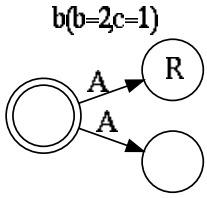

図 A.4: 枝分かれ数 2, 競合 1

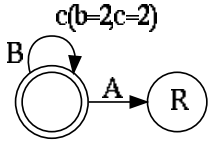

図 A.5: 枝分かれ数 2, 競合 2, 回帰ルール

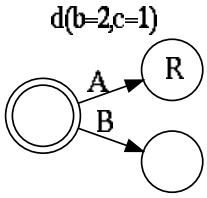

図 A.6: 枝分かれ数 2, 競合 2

3 枝分かれが 3 の場合
競合数が 1 の場合は (1) と同様．競合数が 2 の場合は (2) と同様に，唯一の回帰ルールと競合する場合困難になる．競合数が 3 の場合，無効ルールが 2 個となるので無効ルール 1 個あたりの強化回数は先ほどの場合よりも減少する．よって競合数 2 のときが最も困難となる．

同様に，枝分かれ数 n のときも，唯一の回帰的無効ルールと競合する場合が最も困難となる．ゆえに，無効ルールを抑制することが最も困難になるのは，有効ルールが唯一の回帰的無効ルールと競合する構造である．

□

証明の中では有効ルール数 $L = 1$ としてきたが，任意の有効ルール数において

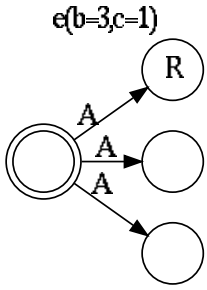

図 A.7: 枝分かれ数 3, 競合 1

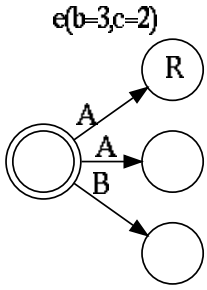

図 A.8: 枝分かれ数 3, 競合 2

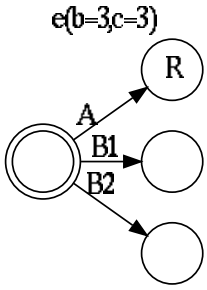

図 A.9: 枝分かれ数 3, 競合 3

も回帰的無効ルールと競合している場合がもっとも困難な構造となる．ここで，行動をとった結果感覚入力の変化が生じていないルールを回帰的であるという．

補題 2 (唯一の回帰的無効ルールの抑制)**.** 唯一の回帰的無効ルールが抑制される必要十分条件は

$$L\sum_{j=i}^{W} f_i < f_{i-1}, [\forall i = 1, 2, \ldots, W.] \tag{A.1}$$

ここで，W はエピソードの最大長，L は同一感覚入力下における有効ルールの最大個数である．

証明 2. 簡単化のために有効ルール数 $L = 1$ の場合について述べるが，そのほかの場合でも全く同様となる．

エピソード長を W とし，唯一の回帰的無効ルールと競合する有効ルールが，報酬から N ステップ前に選ばれていたとする．回帰的無効ルールの強化値が最大となるのは，有効ルールより以前のステップ $(W - N)$ において常に無効ルールが選択された場合である．この際，有効ルールの強化値が無効ルールの強化値よりも大きくなるためには，以下の不等式を満たす必要がある．

$$\sum_{j=N+1}^{W} f_j < f_N, [\forall N = 0, 1, \ldots, W - 1] \tag{A.2}$$

□

定理 1 (無効ルールの抑制). 任意の無効ルールが抑制される必要十分条件は

$$L\sum_{j=i}^{W} f_j < f_{i-1}, [\forall i = 1, 2, \ldots, W] \tag{A.3}$$

ここで，W はエピソードの最大長，L は同一感覚入力下に存在する有効ルールの最大個数である．(A.3) 式は**無効ルール抑制条件**と呼ばれる．

A.2.1 定理の意味

前項で述べた定理は有効ルールを差し置いて無効ルールが強化されることがおこならないという局所的合理性を保証している．従って，各感覚入力において最も大きな重みを持つルールを選択すれば良い．特に有効ルール数 $L = 1$ の場合，常に最適なルールを選択されることが保証される．一般的に，この L の値は学習以前に知ることはできないが，実装においてはとることのできる “行動数 - *1*” とすればよい．

従来の ProfitSharing[30] で用いられてきた定数関数および等差減少関数では定理を満たさず，非合理的な学習をする場合がある．定理を満たす最も簡単な強化関数としては，等比減少関数が考えられる．

$$f_n = \frac{1}{S} f_{n-1}, n = 1, 2, \ldots, W - 1 \tag{A.4}$$

ただし，$S \geq L + 1$ とする．

ここで，S を**強化減少比と呼ぶ**．この関数が定理を満たすことは次の様に確認できる．

$$\begin{aligned} L\sum_{j=i}^{W} f_j &= \frac{L}{S}\sum_{j=i-1}^{W-1} f_j \\ &= \frac{L}{S} f_{i-1} + \frac{L}{S}\sum_{j=i}^{W} f_j - \frac{L}{S} f_W \end{aligned} \tag{A.5}$$

従って，

$$\begin{aligned} L\sum_{j=i}^{W} f_j &= \frac{L}{S-1}(f_{i-1} - f_w) \\ &\leq f_{i-1} - f_w \\ &< f_{i-1} \end{aligned} \tag{A.6}$$

付 録B　免疫型強化学習器のパラメータ設定基準[36]

本章では免疫型強化学習器が最適ルールを獲得できるパラメータについて述べる．ここでいう最適ルールの獲得とは，ある状態 $\boldsymbol{s}_i$ において最も効率よく（より少ないルール選択回数で）報酬を得ることのできるルール $\overrightarrow{\boldsymbol{s}_i k}$ の評価値 $w_k(\boldsymbol{s}_i)$ が最大となることを指す．ここで最適ルール獲得能力について検討するため，図 B.1 の環境を考える．この環境において，ルール $\overrightarrow{\boldsymbol{s}_1 i}$ はその後 p 回のルール選択後に報酬を受け取ることができ，ルール $\overrightarrow{\boldsymbol{s}_1 j}$ はその後 q 回のルール選択後に報酬を受け取ることができる．このとき，$w_i(\boldsymbol{s}_1)$, $w_j(\boldsymbol{s}_1)$ が受け取ることのできる報酬はそれぞれ

$$r_i(\boldsymbol{s}_1) = \beta^p \times R \tag{B.1}$$

$$r_j(\boldsymbol{s}_1) = \beta^q \times R \tag{B.2}$$

である．ここで $p < q$ とする．つまり $\overrightarrow{\boldsymbol{s}_1 i}$ の方がより効率がよいルールとする．このとき，最適ルールを獲得するための必要条件は $w_i(\boldsymbol{s}_1) > w_j(\boldsymbol{s}_1)$ となる．そこで，(i)$\overrightarrow{\boldsymbol{s}_1 i}$ が選択された場合，(ii)$\overrightarrow{\boldsymbol{s}_1 j}$ が選択された場合，それぞれについて $w_i(\boldsymbol{s}_1) > w_j(\boldsymbol{s}_1)$ が成立するか検討する．なお，以下では更新前の評価値を $w_k(\boldsymbol{s}_1, t)$，更新後の評価値を $w_k(\boldsymbol{s}_1, t+1)$ として記述している．

(i) $\overrightarrow{\boldsymbol{s}_1 i}$ が選択された場合

$\overrightarrow{\boldsymbol{s}_1 i}$ が選択された場合，(2.2), (2.3) 式より

$$w_i(\boldsymbol{s}_1, t+1) = (1-\alpha) w_i(\boldsymbol{s}_1, t) + \alpha r_i(\boldsymbol{s}_1) \tag{B.3}$$

$$w_j(\boldsymbol{s}_1, t+1) = (1-\alpha) w_j(\boldsymbol{s}_1, t) \tag{B.4}$$

となるので，$w_i(\boldsymbol{s}_1, t+1) > w_j(\boldsymbol{s}_1, t+1)$ となるための条件は

$$\frac{w_i(\boldsymbol{s}_1, t) - w_j(\boldsymbol{s}_1, t)}{R} > \frac{-\alpha\beta^p}{1-\alpha} \tag{B.5}$$

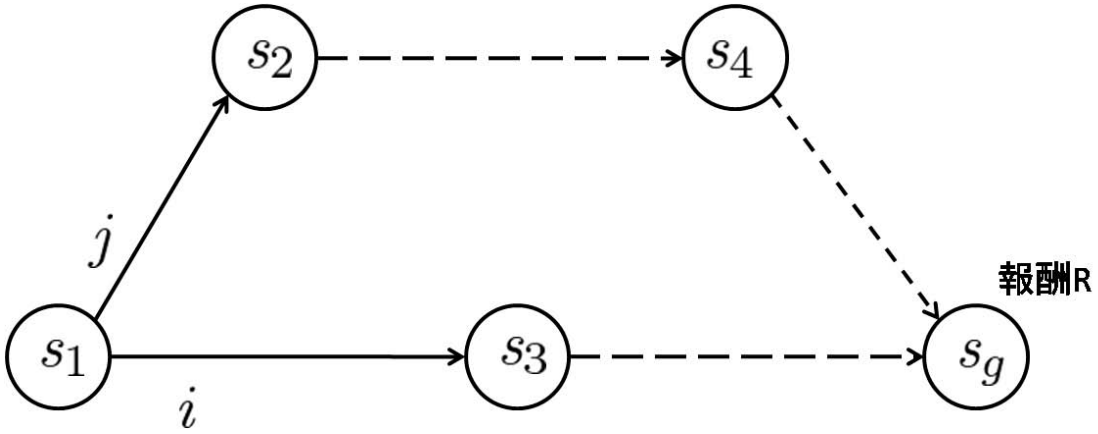

図 B.1: 報酬獲得が可能なルールが 2 種類存在する環境

となる．ここで，最も α の取りうる範囲に制約がかかるのは $w_i(\boldsymbol{s}_1, t) = 0$（$i$ の評価が最低）かつ $w_j(\boldsymbol{s}_1, t) = \beta^q \times R$（$j$ の評価が最高）の場合であるので，

$$-\beta^q > \frac{-\alpha\beta^p}{1-\alpha} \tag{B.6}$$

$$\beta^{q-p} < \frac{\alpha}{1-\alpha} \tag{B.7}$$

となるが，$0 < \beta < 1$ より β^{q-p} の中で最大のものは β となるので，最終的に

$$\beta < \frac{\alpha}{1-\alpha} \tag{B.8}$$

が得られる．

(ii) $\overrightarrow{s_1 j}$ が選択された場合

$\overrightarrow{s_1 j}$ が選択された場合，(2.2), (2.3) 式より

$$w_i(\boldsymbol{s}_1, t+1) = (1-\alpha) w_i(\boldsymbol{s}_1, t) \tag{B.9}$$

$$w_j(\boldsymbol{s}_1, t+1) = (1-\alpha) w_j(\boldsymbol{s}_1, t) + \alpha r_j(\boldsymbol{s}_1) \tag{B.10}$$

となるので，$w_i(\boldsymbol{s}_1, t+1) > w_j(\boldsymbol{s}_1, t+1)$ となるための条件は

$$\frac{w_i(\boldsymbol{s}_1, t) - w_j(\boldsymbol{s}_1, t)}{R} > \frac{\alpha\beta^q}{1-\alpha} \tag{B.11}$$

である．ここで左辺を Δw とおくと，(B.11) 式は

$$\Delta w > \frac{\alpha\beta^q}{1-\alpha} \tag{B.12}$$

となる．このとき，まず $\Delta w \leq 0$ つまり $w_i(\boldsymbol{s}_1, t) \leq w_j(\boldsymbol{s}_1, t)$ の場合は，$w_i(\boldsymbol{s}_1, t+1) > w_j(\boldsymbol{s}_1, t+1)$ とはなり得ない．一方，$\alpha\beta^q < \alpha$ なので，

$$\Delta w > \frac{\alpha}{1-\alpha} \tag{B.13}$$

である．これより $\alpha \to 0$ とすれば $\Delta w \to 0$ で成立する．ただし，(2.2) 式より α は学習率なので，$\alpha = 0_+$ とすると学習が進まなくなる．そのため，提案手法は最適ルールの獲得を保証することはできない．しかし，α を十分小さく取ることにより最適解探索能力を高めることはできる．よって学習速度と最適解探索能力はトレードオフの関係となる．

付 録C　Keepaway

C.1　Keepawayの概要

ロボカップサッカーのサブタスクとして，片方のチームが制限された領域内でボールを保持するkeeperとボールを奪うtakerが存在するKeepawayを考える．takerがボールを奪うかボールが領域外に出た場合にエピソードを終了し，プレイヤーをリセットする(keeperがボールを持った状態にもどす)．

タスクのパラメータとして領域のサイズとkeeperの台数，takerの台数がある．図C.1に3台のkeeperと2台のtaker(3v2)が20×20mの領域にいる場合と4v3で30×30の領域に居る場合を示す．

本研究では標準ロボカップサッカーシミュレータを使用している．ロボカップシミュレータ内のエージェントは150ms周期でボールやほかのエージェント等の距離と角度情報を取得している．エージェントはターンやダッシュ，キックなどのパラメータ化された行動を100ms周期ごとに実行する．従って，センシングと行動は非同期となる．ランダムノイズがすべてのセンシングと行動に付加される．個々のエージェントは別々のプロセスで制御される必要があり，エージェント間の通信は通信帯域が制限されたシミュレータを通してのみ許可されている．学習の観点から，これらの制約はチームメートが同時に独立して制御方策を学ぶ必要があり，学習結果を共有することも，チーム全体の意志決定を行うこともできない．従って，チーム内で方策を共有するマルチエージェント学習法は適用できない．

Keepawayタスクでは，全知のコーチがプレーを管理し，設定時間が経過するかボールが領域から出た場合にエピソードを終了する．エピソード開始時にコーチはボールの位置とプレイヤーの位置を領域内に半ランダムに設定する．takerは常に左下のコーナーからスタートする．3台のランダムで選ばれたkeeperがそのほかのコーナーに配置され，それ以外は中央に配置される．ボールは初期に左上のkeeperのそばに置かれる．初期配置の例を図C.1に示す．

Keepawayの利点は完全なロボットサッカーのタスクよりもほかの学習方法と比

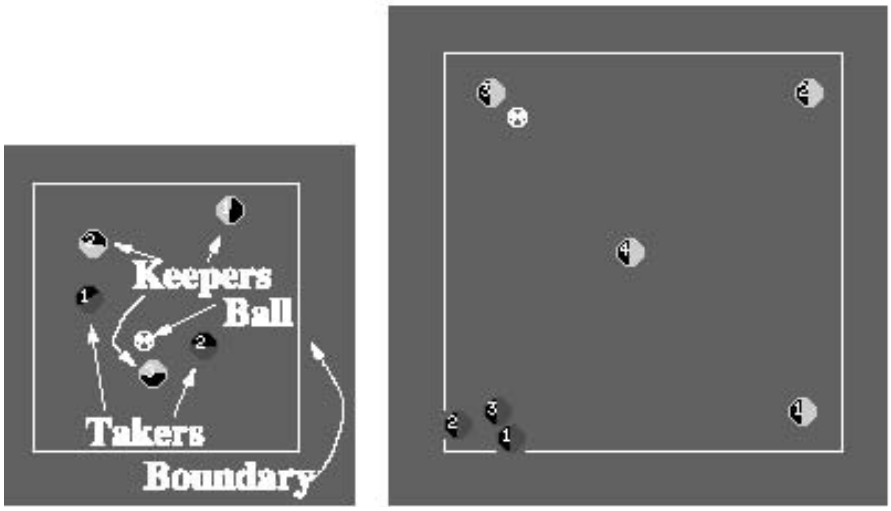

図 C.1: プレーヤの動作領域

較に適していることである．強化学習に加え，ロボカップチームでは遺伝的プログラミングやニューラルネットワーク，決定木などの機械学習法が組み合わされて構築されている．これらやその他の機械学習法のロボカップへのアプローチの不満点はそのほかのシステムに組み込まれていたり，フルサッカーのサブタスクとは別に設定されることである．従って，彼らは他手法との比較を行うことが困難になっている．Keepaway は全体の学習ができるほどシンプルであり，かつ単純な解決策では解けないくらい複雑性がある．従って，機械学習のベンチマーク問題としては有力な候補となる．

C.2 強化学習への Keepaway の割り当て

我々の Keepaway 法は離散時間のエピソードベースの強化学習法に適用することができる．ロボカップサッカーシミュレータは離散時間ステップ $t = 0, 1, 2 \ldots$, をそれぞれ 100ms ごとにシミュレートしている．各プレーヤは個別に学習し，異なる環境を認識する可能性がある．各プレーヤは，最初に行動を決定してからエピソードをスタートし，ボールをロストしたときにエピソードを終了する．

ドメインレベルの知識を組み込む方法として，我々はシミュレータレベルの基礎的行動ではなく， CMUnited-99 team が使用しているスキルをベースにしたハイレベルマクロを使用した．スキルは以下のものが存在する．

- HoldBall()

- PassBall()
- GetOpen()
- GotoBall()
- BlockPass()

PassBall() を除いたすべてのスキルは対応する基礎的な行動に対応した単純な関数となり，通常単一の時間ステップで実行できる．しかしながら PassBall() はボールを蹴る場所への移動や所望の方向へのキック等の基礎的な行動を拡張した逐次処理が必要となり，いくらかの時間ステップに影響を与える．さらにプレーヤのミスにより単純なスキルでさえ時間ステップに影響を与えることがある．これらの場合では次の行動選択がスキルを実行した 2 ステップ以上先になることがある．そのような可能性を扱うために，SMDP として取り扱うのが簡単である．SMDP は SMDP マクロが終了後した後に次のステップが開始される．SMDP マクロはサブポリシーとオプションと呼ばれる終了条件を含んでいる．

チームの視点から，それぞれのチームメートが全体の決定過程を分担していることから Keepaway は分散 SMDP として見なすことができる．プレーヤは共有知識なしに同時に学習するため，個々の知覚からタスクが提示される．それぞれの選択は基礎的な行動ではなくマクロによって行われる．選択された i 番目のマクロを $a_i \in A$ と表す．従って，いくらかのタイムステップが a_i と a_{i+1} 間で経過している．同様に i 番目のマクロ状態を $s_i \in S$，報酬を $r_i \in R$ と表す．keeper の各ステップでの目標はエピソードがより長く継続し，報酬を最大化することである．

C.2.1 Keepers

ここでは keeper がとることのできるマクロを示す．

keeper がボールのを保持するための予備実験において，keeper がボールのポジションにいないときに Receive 行動が必要であった．

- Receive

一方，ボールを保持している場合は本来の選択肢となる．ボールを保持できるか，チームメートにパスできる場合，マクロ {HoldBall, PassK_2ThenRecive, PassK_3ThenRecive, . . . , PassK_nThenRecive} から選択され，HoldBall の場合 1 ステップ実行され，Pass$k$ThenRecive

アクションはほかの keeper にパスをする．keeper はボールに近い順番からナンバーリングがされる．

ベンチマークポリシーの例として以下を上げる．

- Random： n 個のマクロをランダムに選択する
- Hold: 常に HoldBall を選択する
- Hand-coded: n 個のマクロの中から学習に使用している状態を条件として設定されているものを選択する

エージェントにつき 1 つの行動しか選択できないため，チーム全体の行動の一部のみしか制御することができないことに注意が必要である．一度ボールをパスするとボールが戻ってくるまで次の行動選択はチームメートの行動によってのみ左右される．また，それぞれのプレーヤが環境から異なる視点と別々の制御政策を学習する必要がある．

次章で説明をする価値関数近似法に使用する keeper の状態表現法について説明をする．それらの値は SMDP ステップのみで必要であり，それはボールを保持している keeper のみである．次の手順でほかの keeper($K_1 - K_n$) や taker($T_1 - T_m$)，環境の中心位置 (C) を用いて keeper の状態変数を定義する (図 C.2)．a, b 間の距離を dist(a, b)，b を頂点とした a, c の角度を ang(a, b, c) として以下の 13 状態変数を使用する．

- dist(K_1, C);dist(K_2, C);dist(K_3, C);
- dist(T_1, C);dist(T_2, C);dist(K_1, K_2);
- dist(K_1, K_3);dist(K_1, T_1);dist(K_1, T_2);
- Min(dist(K_2, T_1), dist(K_2, T_2));
- Min(dist(K_3, T_1), dist(K_3, T_2));
- Min(ang(K_2, K_1, T_1), ang(K_2, K_1, T_2));
- Min(ang(K_3, K_1, T_1), ang(K_3, K_1, T_2));

このリストは keeper と taker を増やすごとに線形に状態変数が増加していく．

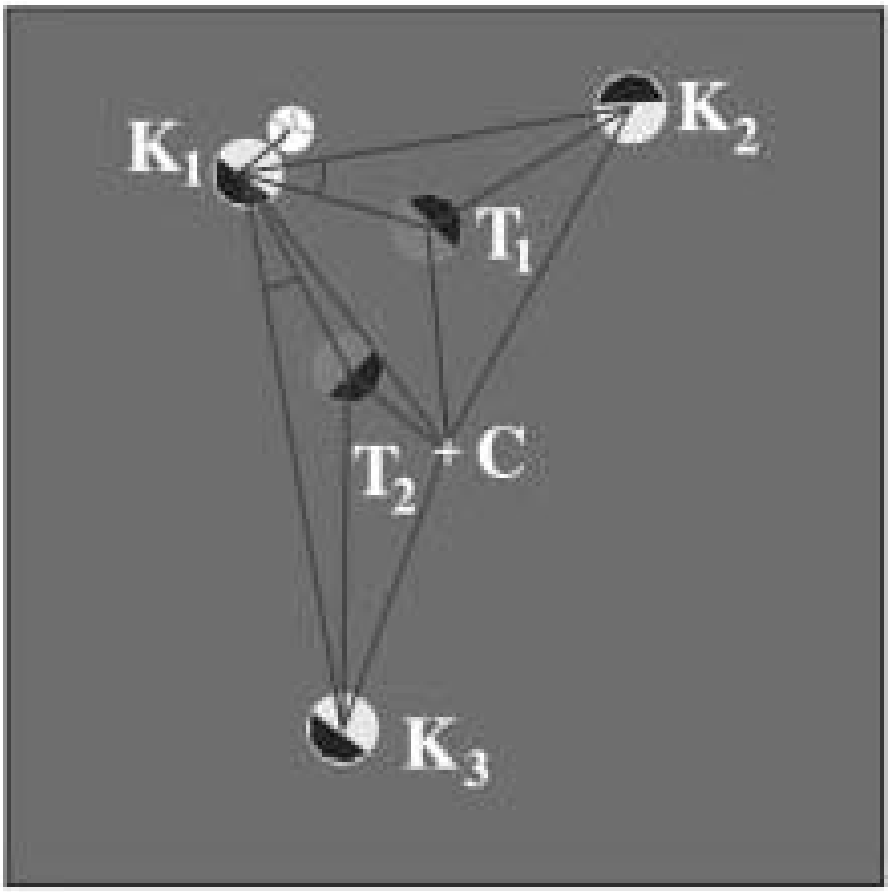

図 C.2: プレーヤの配置と状態変数

C.2.2 Taker

本研究では事前に定義された taker の動作に対する keeper の学習に焦点を置いているが，公平のために同じ枠組みを利用して taker の動作を指定する．

taker は基本的な行動を置き換えているマクロを選べば比較的単純である．taker がボールを保持しているとき HoldBall() を呼び出してボールを保持し続けようとする．それ以外では，リスト {GoToBall(), BlockPass(K_2),. . . , BlockPass(K_n)} の中から行動を選択する．keeper がボールを保持していないとき (パスの途中など) は K_1 はボールにもっとも近い keeper に割り当てられる．我々はボールを保持していないときの taker のベンチマークとして次の 3 つの政策を定義する．

- Random-T: n 個のマクロからランダムに行動を選択する
- All-to-ball: 常に GoToBall() を選択する

- Hand-coded-T: 短時間でボールに到達できる2台のtakerはGoToBall()を選択し，そうでない場合はもっともtakerから離れているk番目のkeeperに対してBlockPass(k)を選択する

ここで，takerが2台しか存在しない場合，All-to-ball政策とHand-Coded-T政策は等価となることに注意する．

takerの状態変数はkeeperのものと似たものとなっており，領域の中心，ほかのtakerの位置を使用する．$k_i$midはボールを保持しているエージェントからi番目とのエージェントの中間距離である．3台のkeeperと3台のtakerによる状態変数は以下の18個になる．

- dist(K_1, C);dist(K_2, C);dist(K_3, C);
- dist(T_1, C);dist(T_2, C);dist(T_3, C);
- dist(K_1, K_2);dist(K_1, K_3);dist(K_1, T_1);
- dist(K_1, T_2);dist(K_1, T_3);dist(T_1, K_2mid); dist(T_1, K_3mid);
- Min(dist(K_2mid, T_2), dist(K_2mid, T_3));
- Min(dist(K_3mid, T_2), dist(K_3mid, T_3));
- Min(ang(K_2, K_1, T_2), ang(K_2, K_1, T_3));
- Min(ang(K_3, K_1, T_2), ang(K_3, K_1, T_3));
- T_1よりもボールのそばにいるkeeperの台数

付 録D　倒立振子の制御特性の検討

本章ではシミュレーションに使用している倒立振子の可安定性，可制御性および可観測性について検討を行う．

D.1　一般的な倒立振子の場合

(3.14)(3.15) 式で示された倒立振子の運動方程式から線形の状態方程式への変換を行う．この線形化にともない振子の角度θと角速度$\dot{\theta}$が十分に小さいことを仮定する．

$$\sin\theta \simeq \theta, \cos\theta \simeq 1, \dot{\theta^2} = 0 \tag{D.1}$$

これを用いると (3.15) 式の運動方程式は下記となる．

$$(M+m)\ddot{x} + ml\theta\ddot{\theta} + D_x\dot{x} + ml\theta\dot{\theta} = a \tag{D.2}$$

$$ml\theta\ddot{x} + (ml^2+I)\ddot{\theta} + D_\theta\dot{\theta} - mgl\theta = 0 \tag{D.3}$$

倒立振子の状態を$[x, \theta, \dot{x}, \dot{\theta}]^{\mathrm{T}}$とし，シミュレーションに使用したパラメータでの状態方程式表現は以下となる．

$$\frac{d}{dt}\begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -0.9763 & -4.998\times10^{-4} & 7.969\times10^{-6} \\ 0 & 42.95 & 1.992\times10^{-3} & -3.506\times10^{-4} \end{bmatrix}\begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0.9996 \\ -3.984 \end{bmatrix} a \tag{D.4}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} \tag{D.5}$$

はじめにシステムの安定性について検討する．システムの安定性は任意の状態か x から，時間 t が無限大経過したのときに

$$\lim_{t\to\infty} x(t) \to 0$$

となる性質である．システムの安定性の判別方法として状態行列の固有値を調べる方法がある．倒立振子の状態行列の固有値を求めたところ $0, -6.554, 6.554, 0.0005$ であった．システムが安定性を有する場合，全ての実部の符号が負である必要があるが正の値を含んでいるためこのシステムは不安定なシステムである．

次に，制御器を設計する上でシステムが制御可能かどうかを判定する可制御性について検討する．可制御性は状態方程式の状態行列および出力行列から導出される可制御性行列 M_c のランクを調べる方法を用いる．以下が計算によって求めた倒立振子の可制御性行列である．

$$M_c = \begin{bmatrix} 0 & 0.999 & 0.0005 & 3.889 \\ 0 & -3.984 & 0.0034 & -171.1 \\ 0.999 & 0.0005 & 3.889 & -0.0027 \\ -3.984 & 0.0034 & -171.1 & 0.2133 \end{bmatrix}$$

この行列のランクを計算すると $rank(M_c) = 4$ であり，可制御性行列の行数とランクが一致しているためシステムは可制御性を有しているといえる．同時にシステムが可安定性を有していることも確認できる．

最後にシステムの初期状態を観測値から推定することができるかどうかを判定する可観測性について検討する．可制御性と同様に状態方程式の状態行列および出力行列から導出される可観測性行列 M_o のランクを調べる方法を用いる．以下が計算によって求めた倒立振子の可観測性行列である．

$$
M_o = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & -9.76 \times 10^{-1} & 5.00 \times 10^{-4} & 7.97 \times 10^{-6} \\
0 & 4.30 \times 10^{1} & 1.99 \times 10^{-3} & -3.51 \times 10^{-4} \\
0 & -9.76 \times 10^{-1} & 5.00 \times 10^{-4} & 7.97 \times 10^{-6} \\
0 & 4.30 \times 10^{1} & 1.99 \times 10^{-3} & -3.51 \times 10^{-4} \\
0 & -1.46 \times 10^{-4} & 2.66 \times 10^{-7} & -9.76 \times 10^{-1} \\
0 & -1.70 \times 10^{-2} & 2.97 \times 10^{-7} & 4.30 \times 10^{1} \\
0 & -1.46 \times 10^{-4} & 2.66 \times 10^{-7} & -9.76 \times 10^{-1} \\
0 & -1.70 \times 10^{-2} & 2.97 \times 10^{-7} & 4.30 \times 10^{1} \\
0 & -4.19 \times 10^{1} & -1.94 \times 10^{-3} & 1.97 \times 10^{-4} \\
0 & 1.84 \times 10^{3} & 8.56 \times 10^{-2} & -3.21 \times 10^{-2}
\end{bmatrix}
$$

この行列のランクを計算すると $rank(M_o) = 4$ であり，可観測性行列の列数と一致しているためシステムは可観測性を有しているといえる．

D.2 T 字型の倒立振子の場合

(4.4)(4.5)(4.6) 式で示された T 字型の倒立振子の運動方程式から線形の状態方程式への変換を行う．倒立振子の状態を $[x, \theta, y, \dot{x}, \dot{\theta}, \dot{y}]^{\mathrm{T}}$ とし，シミュレーションに使用したパラメータでの状態方程式表現は次式となる．

$$\frac{d}{dt}\begin{bmatrix} x \\ \theta \\ y \\ \dot{x} \\ \dot{\theta} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 00 & \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 5.45 & 9.96\times10^{-2} & 6.9\times10^{-4} & -1.61\times10^{-5} & 1.32\times10^{-2} \\ 0 & 117 & 2.14 & 5.08\times10^{-3} & -4.36\times10^{-4} & 0.353 \\ 0 & 37.1 & 0.692 & 1.32\times10^{-3} & -1.41\times10^{-4} & 0.474 \end{bmatrix} \begin{bmatrix} x \\ \theta \\ y \\ \dot{x} \\ \dot{\theta} \\ \dot{y} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1.38 \\ -10.2 \\ -2.65 \end{bmatrix} a \quad \text{(D.6)}$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \theta \\ y \\ \dot{x} \\ \dot{\theta} \\ \dot{y} \end{bmatrix} \quad \text{(D.7)}$$

通常の倒立振子の例と同様にシステムの固有値を求めたところ

$$\lambda = 0, -10.7855, 10.8997, 0.3920, -0.0327, 0.0005$$

であり実部に正の値を含んでいるので不安定システムである．た，可制御性行列のランクは $rank(M_c) = 6$，可観測性行列のランクは $rank(M_o) = 6$ であり，それぞれの行列の行・列数と等しいため可制御性と可観測性を備えたシステムであることがわかる．

$$M_c = \begin{bmatrix} 0 & -1.38 & -3.58\times10^{-2} & -5.59\times10^{1} & -1.03\times10^{1} & -6.57\times10^{3} \\ 0 & -1.02\times10^{1} & -9.38\times10^{-1} & -1.20\times10^{3} & -2.47\times10^{2} & -1.41\times10^{5} \\ 0 & -2.65 & -1.26 & -3.81\times10^{2} & -2.16\times10^{2} & -4.48\times10^{4} \\ -1.38 & -3.58\times10^{-2} & -5.59\times10^{1} & -1.03\times10^{1} & -6.57\times10^{3} & -1.96\times10^{3} \\ -1.02\times10^{1} & -9.38\times10^{-1} & -1.20\times10^{3} & -2.47\times10^{2} & -1.41\times10^{5} & -4.51\times10^{4} \\ -2.65 & -1.26 & -3.81\times10^{2} & -2.16\times10^{2} & -4.48\times10^{4} & -3.05\times10^{4} \end{bmatrix}$$

$$
M_O = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 5.45 & 9.96 \times 10^{-2} & 6.90 \times 10^{-4} & -1.61 \times 10^{-5} & 1.32 \times 10^{-2} \\
0 & 1.17 \times 10^{2} & 2.14 & 5.08 \times 10^{-3} & -4.36 \times 10^{-4} & 3.53 \times 10^{-1} \\
0 & 3.71 \times 10^{1} & 6.92 \times 10^{-1} & 1.32 \times 10^{-3} & -1.41 \times 10^{-4} & 4.74 \times 10^{-1} \\
0 & 5.45 & 9.96 \times 10^{-2} & 6.90 \times 10^{-4} & -1.61 \times 10^{-5} & 1.32 \times 10^{-2} \\
0 & 1.17 \times 10^{2} & 2.14 & 5.08 \times 10^{-3} & -4.36 \times 10^{-4} & 3.53 \times 10^{-1} \\
0 & 3.71 \times 10^{1} & 6.92 \times 10^{-1} & 1.32 \times 10^{-3} & -1.41 \times 10^{-4} & 4.74 \times 10^{-1} \\
0 & 4.93 \times 10^{-1} & 9.19 \times 10^{-3} & 1.79 \times 10^{-5} & 5.45 & 1.06 \times 10^{-1} \\
0 & 1.31 \times 10^{1} & 2.44 \times 10^{-1} & 4.68 \times 10^{-4} & 1.17 \times 10^{2} & 2.30 \\
0 & 1.76 \times 10^{1} & 3.28 \times 10^{-1} & 6.27 \times 10^{-4} & 3.71 \times 10^{1} & 9.16 \times 10^{-1} \\
0 & 4.93 \times 10^{-1} & 9.19 \times 10^{-3} & 1.79 \times 10^{-5} & 5.45 & 1.06 \times 10^{-1} \\
0 & 1.31 \times 10^{1} & 2.44 \times 10^{-1} & 4.68 \times 10^{-4} & 1.17 \times 10^{2} & 2.30 \\
0 & 1.76 \times 10^{1} & 3.28 \times 10^{-1} & 6.27 \times 10^{-4} & 3.71 \times 10^{1} & 9.16 \times 10^{-1} \\
0 & 6.41 \times 10^{2} & 1.17 \times 10^{1} & 2.79 \times 10^{-2} & 4.91 \times 10^{-1} & 1.99 \\
0 & 1.37 \times 10^{4} & 2.51 \times 10^{2} & 5.97 \times 10^{-1} & 1.30 \times 10^{1} & 4.26 \times 10^{1} \\
0 & 4.37 \times 10^{3} & 8.00 \times 10^{1} & 1.90 \times 10^{-1} & 1.76 \times 10^{1} & 1.39 \times 10^{1} \\
0 & 6.41 \times 10^{2} & 1.17 \times 10^{1} & 2.79 \times 10^{-2} & 4.91 \times 10^{-1} & 1.99 \\
0 & 1.37 \times 10^{4} & 2.51 \times 10^{2} & 5.97 \times 10^{-1} & 1.30 \times 10^{1} & 4.26 \times 10^{1} \\
0 & 4.37 \times 10^{3} & 8.00 \times 10^{1} & 1.90 \times 10^{-1} & 1.76 \times 10^{1} & 1.39 \times 10^{1} \\
0 & 1.31 \times 10^{2} & 2.42 & 5.14 \times 10^{-3} & 6.41 \times 10^{2} & 1.28 \times 10^{1} \\
0 & 3.11 \times 10^{3} & 5.74 \times 10^{1} & 1.23 \times 10^{-1} & 1.37 \times 10^{4} & 2.76 \times 10^{2} \\
0 & 2.57 \times 10^{3} & 4.72 \times 10^{1} & 1.08 \times 10^{-1} & 4.37 \times 10^{3} & 9.28 \times 10^{1} \\
0 & 1.31 \times 10^{2} & 2.42 & 5.14 \times 10^{-3} & 6.41 \times 10^{2} & 1.28 \times 10^{1} \\
0 & 3.11 \times 10^{3} & 5.74 \times 10^{1} & 1.23 \times 10^{-1} & 1.37 \times 10^{4} & 2.76 \times 10^{2} \\
0 & 2.57 \times 10^{3} & 4.72 \times 10^{1} & 1.08 \times 10^{-1} & 4.37 \times 10^{3} & 9.28 \times 10^{1} \\
0 & 7.54 \times 10^{4} & 1.38 \times 10^{3} & 3.28 & 1.31 \times 10^{2} & 2.35 \times 10^{2} \\
0 & 1.62 \times 10^{6} & 2.95 \times 10^{4} & 7.02 \times 10^{1} & 3.10 \times 10^{3} & 5.04 \times 10^{3} \\
0 & 5.14 \times 10^{5} & 9.41 \times 10^{3} & 2.24 \times 10^{1} & 2.57 \times 10^{3} & 1.63 \times 10^{3}
\end{bmatrix}
$$

謝辞

本論文を作成するにあたり，御多忙の中，最後まで熱心な御支援，御指導を賜りました樋口幸治准教授，中野和司前教授，桐本哲郎教授，新誠一教授，内田雅文准教授に深く感謝するとともに，ここに厚く御礼申し上げます．また，鳥取大学工学研究科機械宇宙工学専攻桜間一徳准教授には研究内容について様々なアドバイスを頂きました．深く感謝し厚く御礼申し上げます．また，手法に対して様々なアドバイスをして頂いた伊藤順吾博士，加藤祥治氏をはじめとするロボット班の皆様，研究室の皆様に深く感謝するとともに御礼申し上げます．

参考文献

[1] 宮崎和光, 山村雅幸, 小林重信. 強化学習における報酬割当の理論的考察. 人工知能学会誌, Vol. 9, No. 4, pp. 580–587, 1994.

[2] George A Bekey. 自律ロボット概論. 毎日コミュニケーションズ, 2007.

[3] アイロボット社. ルンバについて. http://www.irobot-jp.com/roomba/index.html.

[4] 産業技術総合研究所. パロのページ. http://paro.jp.

[5] SoftBank. Pepper とは. http://www.softbank.jp/robot/products/.

[6] 下笹洋一, 若林潔, 森口拓雄, 杉浦正則, 藤瀬弘樹, 小谷健太郎. 屋外警備ロボット alsok ガードロボ i (アイ) の開発と安全方針. 日本ロボット学会誌, Vol. 24, No. 2, pp. 156–158, 2006.

[7] 斉藤制海, 徐粒. 制御工学 -フィードバック制御の考え方-. 森北出版株式会社, 2003.

[8] 吉川恒夫, 井村純一. 現代制御理論. 株式会社　昭晃堂, 1994.

[9] E. Rimon and D.E. Koditschek. Exact robot navigation using artificial potential functions. *Robotics and Automation, IEEE Transactions on*, Vol. 8, No. 5, pp. 501 –518, Oct 1992.

[10] ロボカップオフィシャルサイト. URL: http://www.robocup.org.

[11] 早川朋久, 藤田政之. マルチエージェントシステムとビークルフォーメーション. 計測と制御, Vol. 46, No. 11, pp. 823–828, 2007.

[12] 桜間一徳, 宮崎裕史, 中野和司, 細川嵩. マルチエージェントシステムによる逃避ターゲットの包囲と誘導. 計測自動制御学会論文集, Vol. 48, No. 4, pp. 224–231, 2012.

[13] 鈴木学. 実環境を考慮したリーダ追従型隊列誘導におけるロボット群の移動. PhD thesis, 電気通信大学大学院, 2013.

[14] Wataru Inujima, Kazushi Nakano, and Shu Hosokawa. Multi-robot coordination using switching of methods for deriving equilibrium in game theory. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2013 10th International Conference on*, 2013.

[15] Shu Hosokawa, Joji Kato, Kazushi Nakano, and Kazunori Sakurama. Angle-based neuro-fuzzy navigation for autonomous mobile robots. *OS1-5 Int.Symp on Artificial Life and Robotics (AROB'11)*, 2011.1.

[16] Katsumichi Sameshima, Kazushi Nakano, Tetsuro Funato, and Shu Hosokawa. Strrt-based path planning with pso-tuned parameters for robocup soccer. *Artifical Life and Robotics*, Vol. 19, , 2014. to appear.

[17] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1989.

[18] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, Vol. 65, No. 6, pp. 386–408, 1958.

[19] K. Ito, Y. Fukumori, and A Takayama. Autonomous control of real snake-like robot using reinforcement learning; abstraction of state-action space using properties of real world. In *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on*, pp. 389–394, Dec 2007.

[20] 伊藤一之, 松野文俊. Qdsega による多足ロボットの歩行運動の獲得. 人工知能学会論文誌, Vol. 17, No. 4, pp. 363–372, 2002.

[21] Richard S.Sutton and Andrew G.Barto. *Reinforcement Learning an Introduction*. MIT Press, 1998.

[22] 高玉圭樹. マルチエージェント学習 -相互作用の謎に迫る-. コロナ社, 2003.

[23] L.A. Zadeh. Fuzzy algorithms. *Information and Control*, Vol. 12, No. 2, pp. 94 – 102, 1968.

[24] L.A. Zadeh. Fuzzy sets. *Information and Control*, Vol. 8, No. 3, pp. 338 – 353, 1965.

[25] Shin-ichi Horikawa, Takeshi Furuhashi, and Yoshiki Uchikawa. On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm. *IEEE transactions on Neural Networks*, Vol. 3, No. 5, pp. 801–806, 1992.

[26] Christopher J. C. H. Watkins and Peter Dayan. Technical note: q-learning. *Mach. Learn.*, Vol. 8, No. 3-4, pp. 279–292, 1992.

[27] J. Peng. *Efficient Dynamic Programming-based Learning for Control*. Northeastern University, 1993.

[28] Tyler Streeter, James Oliver, and Adrian Sannier. Verve: A general purpose open source reinforcement learning toolkit. *ASME Conference Proceedings*, Vol. 2006, No. 4255X, pp. 359–369, 2006.

[29] Rummery G. A. and M Niranjan. On line q-learning using connectionist systems. *Technical Report CUED/F-INFENG /TR 166, Engi neering Department, Cambridge University*, 1994.

[30] J. J. Grefenstette. Credit assignment in rule discovery systems based on genetic algorithms. In J. W. Shavlik and T. G. Dietterich, editors, *Readings in Machine Learning*, pp. 524–534. Kaufmann, San Mateo, CA, 1988.

[31] J. D. Farmer and N. H. Packard. The immune system, adaptation, and machine learning. *Physica D*, Vol. 22, pp. 187–204, 1986.

[32] N. K. Jarne. Idiotypic networks and other preconceived ideas. *Immunological Reviews*, No. 79, pp. 5–24, 1984.

[33] 近藤敏之, 黒石章夫, 内川嘉樹. 生体内免疫系を参考にした自律移動ロボットの行動調停機構の創発的生成に関する一手法. 計測自動制御学会論文集, Vol. 35, No. 2, pp. 262–270, 1999.

[34] Guan-Chun Luh, Wei-Wen Liu. An immunological approach to mobile robot reactive navigation. *Applied Soft Computing*, Vol. 8, No. 1, 2008.

[35] 伊藤順吾, 中野和司, 桜間一徳. 局所解脱出のための免疫型システムを用いた自律移動ロボットナビゲーション手法. 電子情報通信学会論文誌, Vol. J91-D, No. 2, pp. 504–508, 2008.

[36] Jungo Ito, Kazushi Nakano, Kazunori Sakurama, and Shu Hosokawa. Adaptive immunity based reinforcement learning. *Artificial Life and Robotics*, Vol. 13, No. 1, pp. 188–193, 2008.

[37] 松井藤五郎, 犬塚信博, 世木博久. 線形関数近似を用いた profit sharing 強化学習法. 第 16 回 人工知能学会全国大会, pp. 2D3–03, 2002.

[38] A.G.Barto, R.S. Sutton, and C. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on System, Man, and Cybernetics*, Vol. SMC-13, No. 5, pp. 834–846, 1983.

[39] Zheng Yu, Luo Siwei, Lv Ziang, and Wu Lina. Control parallel double inverted pendulum by hierarchical reinforcement learning. In *Signal Processing, 2004. Proceedings. ICSP '04. 2004 7th International Conference on*, Vol. 2, pp. 1614 – 1617, 2004.

[40] Shu Hosokawa, Joji Kato, and Kazushi Nakano. A reward allocation method for reinforcement learning in stabilizing conntrol tasks. *International Symposium on Artificial Life and Robotics*, pp. OS27–2, 2012.

[41] Atsushi Suzuki, Tohgoroh Matui, and Hirohisa Seki. Profit sharing considering penalty. *The 17th Annual Conference of the Japanese Society for Artificial Intelligence*, pp. 3F4–02, 2003.

[42] S. M. Garrett. *How Do We Evaluate Artificial Immune Systems?*, Vol. 13. MIT Press, 2005.

[43] 伊藤順吾, 新井香奈子, 桜間一徳, 中野和司. 免疫型システムを用いたサッカーロボットコントロールシステムの設計. 日本ロボット学会誌 = Journal of Robotics Society of Japan, Vol. 23, No. 5, pp. 637–640, jul 2005.

[44] 免疫学ハンドブック編集委員会 (編). 免疫学ハンドブック. オーム社, 2005.

[45] 細川嵩, 中野和司, 桜間一徳, 伊藤順吾. 局所解脱出を考慮した免疫型強化学習器について. 電子情報通信学会 2009 総合大会, 3 2009.

[46] 吉田和子, 石井信. 強化学習における exploration と exploitation の制御. 電子情報通信学会技術研究報告. NC, ニューロコンピューティング, Vol. 101, No. 154, pp. 41–48, 20010622.

[47] 今井遼太郎, 吉川毅, 野中秀俊, 杉本政則. 搾取と探索のトレードオフを解決する適応型強化学習の提案. *The 27th annual conference of japanese socitey of Artificial intelligence*, pp. 1E4–4, 2013.

[48] 桜間一徳, 原聡司, 中野和司. エネルギー制御法と制御ラグラジアン法による倒立振子の振上げ・安定化制御. 電気学会論文誌. C, 電子・情報・システム部門誌 = The transactions of the Institute of Electrical Engineers of Japan. C, A publication of Electronics, Information and System Society, Vol. 126, No. 5, pp. 617–623, may 2006.

[49] Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, Vol. 112, pp. 181–211, 1999.

[50] Shu Hosokawa, Kazushi Nakano, and Kazunori Sakurama. A consideration of human immunity-based reinforcement learning with continuous states. *Artifical Life and Robotics*, Vol. 15, No. 4, pp. 560–564, 2010.

[51] 伊藤順吾, 中野和司, 桜間一徳. 獲得免疫系の免疫反応を基にした強化学習機構の構築. 電子情報通信学会論文誌, Vol. J91-D, No. 10, pp. 2487–2496, 2008.

関連論文の印刷公表の方法および時期

1 全著者名：Shu Hosokawa, Kazushi Nakano, Kazunori Sakurama
論文題名：A consideration of human immunity-based reinforcement learning with continuous states
印刷公表の方法および時期: Artificial Life and Robotics, Vol.15, 2010 年
(3 章の内容)

2 全著者名：Shu Hosokawa, Joji Kato, Kazushi Nakano
論文題名： A Reward Allocation Method for Reinforcement Learning in Stabilizing Control Tasks
印刷公表の方法および時期: Artifical Life and Robotics,
Vol.19, No 2, pp 109-114, 2014.
(4 章の内容)

参考論文の印刷公表の方法および時期

1 全著者名：細川嵩, 中野和司, 桜間一徳, 伊藤順吾
論文題名 :局所解脱出を考慮した免疫型強化学習器について
印刷公表の方法および時期：電子情報通信学会 2009 総合大会, 2009.

2 全著者名：Shu Hosokawa, Kazushi Nakano
論文題名：A Consideration on Immunity-based Reinforcement Learning in a Continuous State Space Environment
印刷公表の方法および時期：Int. Symp. on Artifical Life and Robotics (AROB'10), OS1-2, 2010.

3 全著者名：細川 嵩, 中野和司
論文題名：免疫型強化学習器の連続状態環境への適用
印刷公表の方法および時期：電子情報通信学会総合大会, D-8-12, 2010.

4 全著者名：Shu Hosokawa, Joji Kato, Kazushi Nakano Kazunori Sakurama
論文題名：Angle-based neuro-fuzzy navigation for autonomous mobile robots
印刷公表の方法および時期：Int. Symp. on Artifical Life and Robotics (AROB'11), OS1-5, 2011.

5 全著者名：Shu Hosokawa, Joji Kato, Kazushi Nakano
論文題名：A Reward Allocation Method for Reinforcement Learning in Stabilizing Control Tasks
印刷公表の方法および時期：Int. Symp. on Artifical Life and Robotics (AROB'12), OS27-2, 2012.

6 全著者名：Shu Hosokawa, Kazushi Nakano
論文題名：A Reward Allocation Method for Reinforcement Learning in Stabilizing Control of T-inverted Pendulum
印刷公表の方法および時期：ECTI-CON 2012, 1329, 2012.

7 全著者名：Shu Hosokawa, Kazushi Nakano
論文題名：A Reward Allocation Method for Human Immunity - based Reinforcement Learning in a Stabilizing Control Problem
印刷公表の方法および時期：IWMST 2012, 95, 2012.

8 全著者名：Wataru Inujima, Kazushi Nakano, Shu Hosokawa
論文題目：Multi-robot coordination using switching of methods for deriving equilibrium in game theory
印刷公表の方法および時期:ECTI TRANSACTIONS ON COMPUTER AND INFORMATION TECHNOLOGY, Vol.8, No.2, pp.167-174, 2014.

9 全著者名：Jungo Ito, Kazushi Nakano, Kazunori Sakurama, Shu Hosokawa
論文題目：Adaptive Immunity Based Reimforcement Learning
印刷公表の方法および時期：Artificial Life and Robotics, Vol.13, No.1, pp. 188-193, 2008

10 全著者名：桜間一徳, 宮崎裕之, 中野和司, 細川　嵩 論文題目：マルチエージェントシステムによる逃避ターゲットの包囲と誘導
印刷公表の方法および時期：計測自動制御学会論文集, Vol.48, No.4, pp. 224-231, 2012.

11 全著者名：Katsumichi Sameshima, Kazushi Nakano, Tetsuro Funato and Shu Hosokawa
論文題目:StRRT-based Path Planning with PSO-tuned Parameters for RoboCup Soccer
印刷公表の方法および時期: Artifical Life and Robotics, Vol.19, 2014.
採録決定済み

著者略歴

2007年3月　電気通信大学電気通信学部電子工学科卒業
2007年4月　電気通信大学大学院電気通信学研究科電子工学専攻
博士前期課程入学
2009年3月　電気通信大学大学院電気通信学研究科電子工学専攻
博士前期課程修了
2009年4月　電気通信大学大学院電気通信学研究科電子工学専攻
博士後期課程入学
2013年3月　電気通信大学大学院電気通信学研究科電子工学専攻
博士後期課程単位取得退学
2013年4月　技術研究組合制御システムセキュリティセンター
就職