

# BDIエージェントによる連続的な仮想世界における シミュレーションの実装

奈良女子大学大学院 人間文化研究科 情報科学専攻  
14840096 亀村美佳

2016年1月29日

## 概要

本論文では、連続的な仮想世界において、エージェントが外乱に対して安定した行動を獲得するためのシミュレーション環境の実装について述べる。近年、実世界で自律的に振る舞うロボットの実現に向けて、研究が進められている。連続な実世界では、離散な仮想世界とは異なる問題に直面する。ロボットないしエージェントの自律的な行動を可能とするには、外乱に対応できる行動を問題に応じたエージェントのプラン作成とその実行において用いる必要がある。この実現に向けて、我々はエージェントの適応性に着目し、カヌーレーシングをテストベッドとしてエージェントが学習を繰り返す中で環境の外乱を自ら分析し、事前に危険状態を認識、及び自律的に安定性の保持を可能とするシミュレーション環境の実現を目指した。

## 1 はじめに

我々は、実世界において自律的に振る舞うロボットに関する研究に取り組んでいる。連続な実世界では、仮想世界とは異なる様々な問題に直面する [7] ため、それらに対処できる能力が求められる。その実現に際し様々な実験が必要となるが、実世界で直ちに実行に移すのは難しい。そこで、連続的な仮想世界におけるシミュレーション環境の実装を目指した。

BDI モデルと呼ばれる、信念 (B)、願望 (D)、意図 (I) の 3 つの心的状態を用いて意思決定を行うモデルがある。この BDI モデルを用いた合理的かつ自律的なエージェントを BDI エージェントと呼ぶ。仮想世界における BDI モデルの有用性が [1] で示されていることから、我々はこの BDI エージェントを利用したシミュレータを提案する。

これまでに我々は、身体性のあるエージェントが動的な環境変化の影響を受ける事例としてカヌーレーシングをテストベッドとし、Jason と呼ばれる言語処理系と Java を用いたシミュレーションの実装に取り組んできた [10, 11, 12, 13]。カヌーレーシングでは、動的に変化する水流の影響を考慮する必要がある。また、離散的なグリッドワールドを用いた仮想世界とは異なり、連続な実世界ではエージェントの行動は正確に表現できず、誤差を伴う。シミュレーションの設計に当たってはそれらを考慮し、エージェントによる実世界での自律的な問題解決をシミュレートできることを目指した。

我々が目指す最終的な理想は、[6] が提示する完全エージェントを実現するシミュレーションである。[1] では、完全エージェントが身体性を有して実世界で長い期間生き延びて機能する際に生じる、仮想世界とは異なる問題の具体例を複数挙げ、仮想世界におけるシミュレーションの有用性と共に、実世界の完全なシミュレーションが困難であることが考察されている。しかしながら、シミュレーション環境を次第に実世界に近づけていくことによって、有用な知見が得られると我々は考えている。

これまでに我々は、異なる学習方式を併用可能な、エージェントの強化学習によるカヌーシミュレーションを実現し、また、カヌーレーシング以外の環境が異なる様々な問題設定にも対応できるものとした。これにより、BDI エージェントが状況に応じてプランを選んで行動する際に、獲得した行動をプラン中で用いることが可能となった [14, 15, 16]。

これらはエージェントの学習による行動決定や、カヌーレーシング以外の条件下のシミュレートに応用させることは可能にしたものの、ランダムな環境外乱による影響を考慮した問題解決までは至っていなかった。動的に変化する予測不可能な環境下で、ロボットないしエージェントが目標を達成するためには、エージェントが自律的に自己の状態を維持し続けることができなければならない。例えばカヌーレーシングでは、カヌーが転覆しないよう、船体の安定性 (後に 4 章で述べる) を保ちながら川下りをする必要がある。しかしながら、先行研究では、カヌーの転覆やそれに伴うカヌーの安定性の概念がなく、これをシミュレーションで表現することができなかった。

そこで本研究では、これらの概念を導入し、エージェントが学習を繰り返す中で自己の安定性を確立することを可能とするシミュレーション環境の作成を目指した。ここで、より自律的なエージェントの問題解決のための手段作成と選択を実現するために、完全エージェントの特徴的な基本概念である適応性 (後に 2 章で述べる) に着目し、カヌーレーシングのモデル化を行った。

安定性の概念の導入にあたっては、その定義が必要である。我々はこれに伴い、「アトラクタ」(後に 3 章で述べる) という概念を導入した。そして、安定性に関わるエージェントの状態を 3 つのクラスに区分し、目標達成のための基本行為を行える状態を保持するプランを実装し

た。具体的には、学習の繰り返しと並行して環境外乱による影響を分析し、自己の安定性に関わる状態がどの区分にあるか認知する。この時、カヌーが転覆しそうで何らかの対処が必要な状態にある場合は、それに対処するよう目標達成に向けた行動を選ぶ。これにより、エージェントが動的に変化する環境に適応し、自律的に安定性を確立して目標達成するシミュレートが可能となった。

## 2 完全エージェント

[6, p.81] が定義する完全エージェントの「完全」とは、実際の振る舞いを行う上で必要とされるすべてのものを備えていることを意味している。また、この完全エージェントの特徴として5つの基本的な概念として、「自律性」、「自己充足性」、「身体性」、「適応性」、「立脚性」を列挙している。完全エージェントは、我々が目指すシミュレーション、ひいては実世界で自律的に振る舞うロボットの実現において最も理想的なエージェントである。

### 2.1 エージェントの適応性

[6, p.93] では、絶え間なく変化する予測不可能な環境の中で、エージェントが長期間にわたって自己を維持するためには、エージェントは適応性を有していかなければならないとしている。ここでは主として生物学的エージェントに的を絞って、生命の生存という観点からその定義を述べているが、我々の研究においても、この概念を応用する、その理由は、問題解決のためのより良いプラン構築においては、エージェントが環境外乱から身を守り、あるいはその影響を最小限にとどめることで自己の安定した状態を維持できる行為の学習及び選択が必要と考えられるためである。そこで本研究では、自己の安定性の保持を考慮した、適応性のあるシミュレーション環境の実装を進めた。

### 2.2 カヌーレーシングに必要な適応性

カヌーレーシングで求められる適応性は、環境外乱下で自己の安定性を保持できるよう自律的に行動を変化させていくことである。

カヌーはレース中に、転覆、あるいは衝突による船体の破損の危険に晒される。これは、川の横波やストロークにより船体が傾くことや、障害物への衝突が影響して起こる危険である。特に川の横波は一定ではなく、それがカヌーに与える影響の度合いもまた、予測不可能である。この環境下で、カヌーは危険を回避することで安定した状態を保ちながら、目標を達成しなければならない。

実世界におけるこれらの危険回避の手段としては、まず、障害物への衝突については、進行先に障害物がある際、それを目視により認識し、進行方向の変更をすることで衝突を回避できると考える。次に転覆については、転覆する前に“転覆しそうな状態”を認識し、状態を回復する行動を取ることで、転覆を未然に防ぐことができる。

我々はエージェントの適応性を本研究に導入するにあたり、この手段に基づき、仮想世界においてエージェントが安定性を保つプラン構築へのモデル化を進めた。

### 3 安定状態とアトラクタ

2章では、エージェントの適応性とカヌーレーシングで求められる適応性について述べ、本研究のエージェントにも適応性の概念が必要であることを示した。ここで適応性の導入に際して、カヌーレーシングの場合適応性とは、安定性を保持しながら行動できるようになることであるため、エージェントの安定性に関わる状態をシミュレータ上でいかに表現するかについての定義が必要である。我々は、アトラクタという概念を用いて、安定性の定義を行う。アトラクタの定義については研究や文献により多少の差があるため、本章では各々の定義を示した上で、本研究での定義について検討する。

#### 3.1 アトラクタに関する先行研究と文献

##### 3.1.1 高田らによる定義

[1]では、実世界での行為の実行は、環境との連続的な相互作用を通じたアトラクター<sup>1</sup>状態を構築することで実現されているとし、それゆえシミュレーション環境にアトラクター状態の導入が必要であるとしている。そして、実世界と相互作用する基本行為を安定的に実行させるため、[2]の「アトラクター状態」の導入の提案をふまえ、基本行為のアトラクター状態を一般的に以下の通り定義している [3, p.1-2]。

##### 1. 維持状態

基本行為を構成する動作列が安定的に実行されている状態であり、そのまま基本行為を構成する動作列の実行を継続する。

##### 2. 引き込み状態

基本行為を構成する動作列が何らかの外乱で正常に実行できなくなりかけているが、元のアトラクター状態へ戻るための動作を実行することで元の維持状態に戻れる状態である。また、引き込み状態は、簡単な動作で維持状態に戻れる状態と、かなり懸命に回復を試みないと戻れない状態などに分類することもできる。

##### 3. 離脱状態

基本行為を構成する動作列が何らかの外乱で正常に実行できなくなり、もう元の維持状態に戻れなくなった状態であり、基本行為の実行を中断し、基本行為が失敗したことを返す。

##### 3.1.2 Pfeiferらによる定義

[7]は、非線形ダイナミカルシステム<sup>2</sup>におけるアトラクタについて述べている。時間経過に従って、システムが自発的にある状態（あるいは状態のクラス）に引き込まれるとき、これを

<sup>1</sup>[1]によると、力学系において、過渡状態を経た後で定常的に観察される状態を「アトラクター」と呼ぶとしている。

<sup>2</sup>ダイナミカルシステムに関してはいくつかの基本的概念が存在し、「ダイナミカルシステム」「カオス」「非線形ダイナミクス」「複雑システム」といった言葉がよく使われる。[7]では数学的な厳密性にはこだわらず、物理的身体を持つエージェント単体やその群について考察するための直感的なメタファーとして、これらの基本的概念を用いている。[7, p96]

「アトラクタ」と呼ぶ [7, p.97]。また、アトラクタに引き込まれるすべての軌道の集合、あるいは元のアトラクタ状態へ戻るようなこの領域のことを「引き込み領域」と呼び [7, p.98,103]、パピーの実験（付録 A）を例に、身体性を持つエージェントの振る舞いについて以下の特徴を 2 点挙げている。

- 安定な運動パターンの数には限りがある。
- 安定な運動パターンはモータ速度、ロボットの形態と環境によって決まるアトラクタ状態であり、そこへとロボットは「引き込まれる」ので、多少外乱が入っても再びそこへと落ち着く。

### 3.2 本研究のカヌーのアトラクタ状態

以上を踏まえ、本研究におけるアトラクタの導入の提案と、そのモデル化について述べる。[7] のパピーの実験では、パピーは 4 本の脚を動かしているうちに安定した行動として「歩行」のようなある特定の運動パターンを創発する。カヌーの例でこれに相当するものは、ストロークにより、転倒する状態に近づくことなく安定して漕ぎ続けている状態と考えられる。そこで我々はこれを「漕艇」というアトラクタ状態としてカヌーシミュレーションに導入する。また、これに伴う本研究におけるエージェントの状態の分類を示し、以下はそれに従って述べるものとする。

- アトラクタ領域

エージェントが安定な運動パターンを行える領域及びその状態であり、目標達成のための基本行為を正常に行える状態である。本研究では、「漕艇」という一つのアトラクタのみ導入している。

- 引き込み領域

あるアトラクタ領域からは外れているものの、何らかの行為あるいは影響によって、あるアトラクタ領域への復帰が可能な領域及びその状態。

- 逸脱領域

アトラクタ領域とその引き込み領域から外れ、アトラクタ領域へ復帰することができない領域及びその状態。

シミュレーションの課題設定については次章で述べているが、エージェントが最終的な目標であるゴールをするには、上記に示したアトラクタ領域もしくは引き込み領域をエージェントが自律的に維持することが必要条件である。

アトラクタという概念を考慮すると、安定性とは、アトラクタ領域にある状態を保とうとする性質であると捉えることができる。また、安定な状態とは、エージェントがアトラクタ領域にあることである。引き込み領域にある時は、エージェントは不安定な状態だといえるが、何らかの適切な対処を取ることで、アトラクタ領域に復帰し安定な状態に戻ることが可能である。しかし、適切な対処を取らなかった場合、あるいはその対処が効果的ではなかった場合、不安定な状態にあるエージェントは、そのまま引き込み領域から逸脱領域に遷移し、安定な状態に戻ることは不可能となり、目標達成のための基本行為を全く行えない状態に陥る。



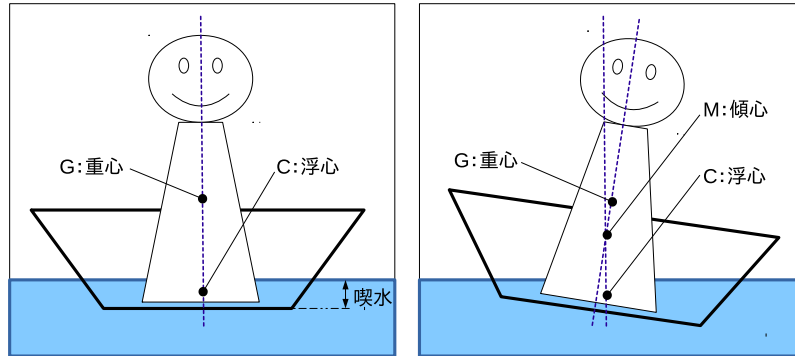


図 2: 実際にカヌーが傾いた状態 (不安定な場合)<sup>5</sup>

#### 4.2 シミュレーション上のカヌーの安定性

実世界のカヌーの安定性は、カヌーに働く複数の力によって 4.1 節の式 (1) で定まる。例えばカヌーが図 1 の状態にある時は、復原力が作用し元に戻る力が働くため、常に安定であるといえる。しかしながら実際には、現実のカヌーの状態は一定ではなく、環境外乱の影響や動作する過程で船体の傾きや傾心、浮心が変化する。このためカヌーが常に安定な状態にあるということではなく、何らかの原因で安定性を失い転覆することが充分にありうる。

船舶の復原性については、[9] が、船が復原力を有するか否かは、重心  $G$  と傾心  $M$  からなる  $GM$  の正負で左右されるとした上で、 $M$  が  $G$  より上にある時を  $GM > 0$  とすれば、

$$\begin{aligned}
 GM > 0 & \text{ 安定} \\
 GM < 0 & \text{ 不安定} \\
 GM = 0 & \text{ 中性}
 \end{aligned}
 \tag{2}$$

であると述べている [9, P.62]。また、傾斜角がごく小さい間では  $M$  はほぼ一定点となり、復原力は船体の傾き  $\theta$  に比例するが、傾斜がさらに大きくなると、 $M$  はもはや一定点ではなくなり、復原力は  $\theta$  に比例しなくなる。この復原力と船体の傾き  $\theta$  の関係を図 3 で示す。

以上を踏まえると、船体の傾きが小さい時、カヌーには復原力が働き安定性を保ちやすいといえるが、傾きがある一定を超えると復原力は次第に小さくなり、安定性が低くなる。そこで本研究では、仮想世界でカヌーの安定性をモデル化するにあたり、安定性を単純化して捉え、カヌーが転覆するか否か、すなわちエージェントが 3.2 節のどの領域にあるかを式 (3) で定めた。

$$|\theta| \leq I \tag{3}$$

$\theta$ : カヌーの傾き ( $-90 < \theta < 90$ )

$I$ : カヌーが 3.2 節で定めたアトラクタ領域、もしくは引き込み領域にいられる  $\theta$  の限界角度

カヌーの傾き  $\theta$  は、水面に対する垂線と、カヌーの底面に対する垂線がなす角度である（図 4）。 $\theta$  が式 (3) を満たすとき、カヌーは 3.2 節で定めたアトラクタ領域、あるいは引き込み領域にあり、転覆していない状態である。

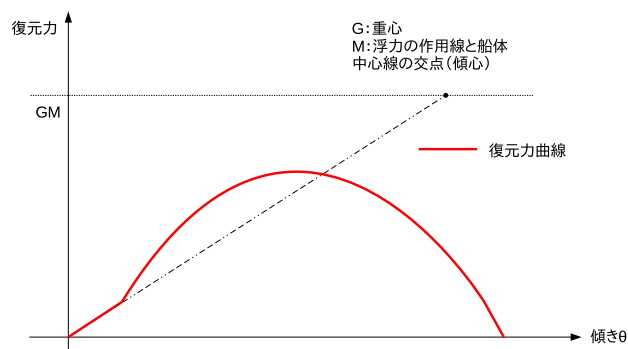


図 3: カヌーの傾きと復元力の関係<sup>6</sup>

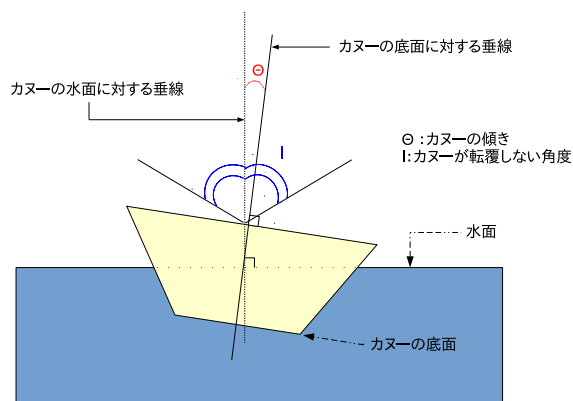


図 4: 本研究でのカヌーの傾きのモデル化

## 5 実装方針

本節ではシミュレーションにおけるカヌーレーシングの概要と問題設定を述べる。

<sup>6</sup>[9, P62 図 3-5] を元に作成。

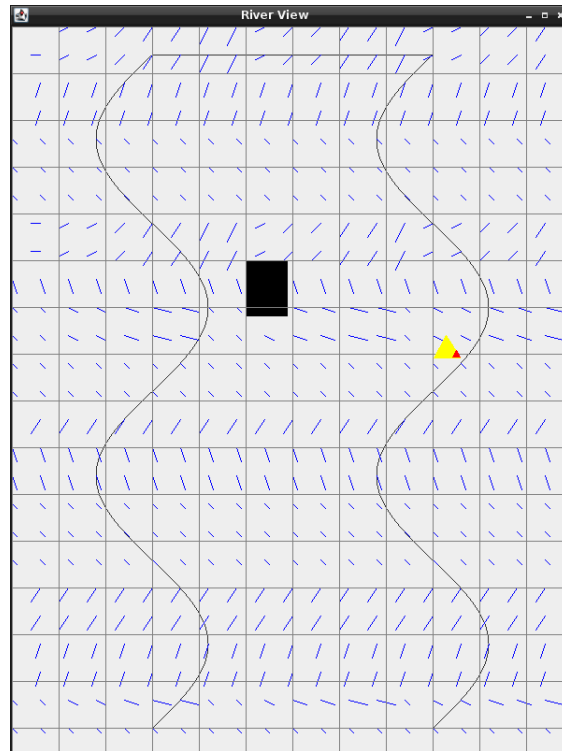


図 5: 川の図



図 6: カヌーの図

## 5.1 課題設定

本研究におけるカヌーレーシングにおけるエージェントの最終目標は、川をより早くゴールすることである。ゴールするとは、ここでは連続空間の中にある川の上端からスタートして下端に到達し、川を下り終えること(以下、これを一つのエピソードと呼ぶ)である。エージェントは、連続空間内で基本行為を繰り返すことで、この目標の達成を目指す。

川を一つのコースとみなし、コースアウト、もしくは途中で転覆した場合はそこで行為を終了し、目標不達成とする。エージェントの行為の結果は、漕ぎ方と川の流れに影響を受け、また、同じ流れて同じ漕ぎ方をしたつもりでも、移動先には誤差を生じる。

## 5.2 川とエージェントの定義

図 5 は、実際に表示するシミュレート画像の一部である。エピソードを任意の回数繰り返して強化学習を行った後に、カヌーに見立てた図 6 のエージェントが図 5 の川を下る様子をシミュレートする。図 5 の黒く塗りつぶされている範囲は、障害物があるものとする。斜線は川の流れを示し、斜線が長いほど、川の流れが急であることを示す。なお、図 6 の赤い先端部分がカヌーの先頭である。

### 5.3 基本行為

カヌーカヤックは両側にブレードがあるパドルを使った、「フォワード・ストローク（前進）」、「リバース・ストローク（後進）」、「ストップング（停止）スイープ・ストローク（回転）」、「ドロロー・ストローク（手前横方向移動）」、「スカーリング・ストローク（横方向移動）」と呼ばれる一般的に使用されているストロークがある [2][4]。本研究ではこれを踏まえ、左右のストロークによる、5つの目標達成のための基本行為を定義した。それらが、「前進」、「後進」、「左進」、「右進」、「何もしない」である。左進及び右進は、片側のみのストロークによる動作を想定したものである。基本行為によるカヌーの動作の定義は、[17]で示している。

さらに我々は、カヌーが転覆せずに安定な状態を保つことができるようにするため、カヌーの状態立て直し行為を基本行為として2つ定義した。これらは、左右どちらかに傾いているカヌーが引き込み領域にある時、もとの「漕艇」のアトラクター領域に戻るためにとるべき基本行為である。以上、7つの基本行為によるカヌーの動作、及び基本行為がカヌーの傾きに与える影響を以下に示す。

いずれの場合も、川の流れの影響として、正規分布に基づく乱数と川の流れによって決まるランダムな値  $x$  が、カヌーの傾き  $\theta$  にさらに加算される。

- 何もしない

川の流れに身をまかせ、ストロークを使用しない。

- 前進

両側のストロークによる進行。川の流れを受けながら、進行方向に直進するよう漕ぐ。

- 後退

両側のストロークによる後退。川の流れを受けながら、進行方向と逆に直進するよう漕ぐ。

- 右進

左側のストロークにより進行する。川の流れを受けながら、進行方向に向かって右前方に進行するよう漕ぐ。

また、右進では、カヌーの進行方向の変化に伴い、カヌーが右に傾く。本研究では、基本行為によりカヌーが傾く角度として  $\delta(0 < \delta < 90$  かつ  $0 < \delta < I)$  を与え、これを  $\gamma(0 < \gamma < 1)$  で重み付けした値をカヌーの傾きに加算する。本研究では  $\delta$  および  $\gamma$  は、定数 15.0 および 0.8 として与えた

- 左進

右側のストロークにより進行する。川の流れを受けながら、進行方向に向かって左前方に進行するよう漕ぐ。

向きが左右逆であることと、 $\delta$  が  $-90 < \delta < 0$  を満たすこと以外、「右進」と同様である。本研究では、 $\delta$  は定数  $-15.0$  として与えた。

- 右へ体勢を立て直す

左に傾いているカヌーが引き込み領域にある時に、もとのアトラクター領域に復帰するため、カヌーの傾きを任意の角度変更する基本行為。ストロークは行わず川の流れに身をまかせて下り、カヌーは進行よりも体勢の立て直しに専念する。この基本行為による

ものとして、カヌーの傾き  $\theta$  は、 $\delta(0 < \delta < I)$  に  $\gamma(0 < \gamma < 1)$  で重み付けした値を加算され、変化する。本研究では  $\delta$  および  $\gamma$  は、定数 5.0 および 0.8 として与えた。

- 左へ体勢を立て直す

右に傾いているカヌーが引き込み領域にある時に、もとのアトラクター領域に復帰するため、カヌーの傾きを任意の角度変更する基本行為。向きが左右逆であることと、 $\delta$  が  $-I < \delta < 0$  を満たすこと以外は「右へ体勢を立て直す」と同様である。本研究では、 $\delta$  は定数  $-5.0$  として与えた。

## 5.4 想定される障害

エージェントは、目標達成のために以下の障害を未然に防ぐ必要がある。また、本研究では、これらの障害に遭遇した場合、カヌーレーシングへの復帰が不可能とみなし、目標不達成としてエピソードを終了する。

1. コースアウト

カヌーがスタートしてからゴールする前に、川の外にカヌーが出てしまった場合を指す。

2. 障害物

実世界において、進行を邪魔する岩などを想定したものである。シミュレーション上では、黒く塗りつぶされたある一部分に障害物があるものとし、カヌーがその領域に入った場合は、障害物に衝突したものとする。

3. 転覆

本研究では、カヌーに船体の傾きの概念を導入している。シミュレーション上ではカヌーが転覆する原理を簡略化して捉え、4.2 に基づき船体の傾き  $\theta$  がある一定を超えるとカヌーが転覆すると定めた。

## 5.5 カヌーに与える環境外乱

カヌーレーシングでは、常に川の横波を受けながら進行する。そして、横波はカヌーの船体の傾きに影響を与える。流れの緩急は川の傾斜や形状に基づき決まるものの、川の水量によって変化し一定ではない。そこでシミュレーション上では、各々の地点の川の横波が船体の傾きに与える影響として、あらかじめ地点ごとに定めた川の流速を、ランダム関数との積として換算することで、動的に変化する川の流れがエージェントに与える影響を仮想世界で実現した。

## 5.6 安定性の確立と引き込み領域の定義

カヌーが安定な状態を保持するためには、エージェントが3.2節の逸脱領域に入る前に何らかの対処を取らなければならない。そこで本研究では、エージェントが3.2節の引き込み領域にある時、5.3節で示す目標達成のための基本行為をやめ、状態立て直し行為を取ることで、アトラクタ領域に復帰し、安定な状態を保持する手法を提案する。

これにあたり、引き込み領域にあるエージェントの状態について、明確な定義が必要となる。予めこの領域を定義しておくことも可能であるが、我々が目指すのは、環境に対し自律的に適

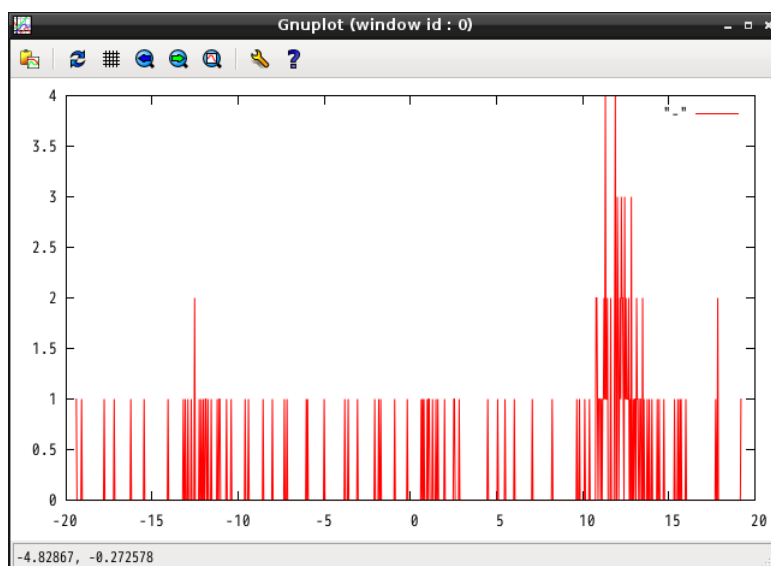


図 7: エピソード 10000 回中 154 回転覆時の、傾きの差  $diff$  の分布

応するエージェントである。そこで、安定状態や引き込み領域の定義をあらかじめエージェントには与えず、行為を繰り返す中でエージェントが環境からの影響を分析し、シミュレートの過程で引き込み領域をエージェントが獲得するという手段を選択した。

### 5.6.1 引き込み領域の定義の具体的な手法

本研究では引き込み領域を定義するにあたり、カヌーが転覆した時の、転覆時の傾きと、転覆直前の傾きの差（以下、 $diff$  とする）に着目した。これは、実世界のカヌーでも、転覆する時は直前に危険を知って転覆することが多く、人間は傾きの急変を認識することで危機を察して状態の持ち直しを図ると考えられるので、それをシミュレートするためである。

10000 回エピソードを繰り返す実験を行い、この差の分布をグラフ化したものが図 7 である。なお、この時転覆は計 154 回起きている。x 軸は  $diff$ 、y 軸はその  $diff$  を示した回数である。このグラフから、 $diff$  がおおよそ 12 あるいは  $-13$  の時に多く転覆したことが読み取れる。なお、正の値では右に傾き転覆し、負の値では左に傾き転覆したことを示す。このことから、 $diff$  が 12 あるいは  $-13$  を示すとき、その後転覆する可能性が高く、何らかの対処が必要であると言える。

ここで、転覆した回数が最も多い時の正負の  $diff$  をそれぞれ  $DIFFMAX$ 、 $DIFFMIN$  とし、シミュレーション上では、現在の傾きと直前の傾きの差  $D$  が以下の式 (4)、式 (5) を満たす時、エージェントは引き込み領域にあると定め、状態立て直し行為を取るものとした。

$$D < DIFFMIN \quad (4)$$

$$D > DIFFMAX \quad (5)$$

また、シミュレーション開始時は  $DIFFMAX = I$ 、 $DIFFMIN = -I$  としておき、エピソードを繰り返しながら、得られたデータの分析を行い、 $DIFFMAX$  及び  $DIFFMIN$  を更新する。これにより  $DIFFMAX$  と  $DIFFMIN$  の精度を高め、引き込み領域を適切な範囲に狭めていく。

### 5.6.2 DIFFMIN と DIFFMAX の決定方法

本節では、具体的な DIFFMIN と DIFFMAX の決定方法を述べる。DIFFMIN と DIFFMAX は、データ分析時に以下の手順で決定し、更新するものとする。

1. 分析する全データの中から、 $diff$  の最小値  $p1$  と最大値  $p2$  を求める。この時、

- $p2 < 0.0$  の時は、 $min = p1, max = I$
- $p1 \leq 0.0$  かつ  $p2 \geq 0.0$  の時は、 $min = p1, max = p2$
- $p1 > 0.0$  の時は、 $min = I, max = p2$

とする (ただし  $I$  は 4.2 節、式 (3) の  $I$  を示す)。

2.  $min$  から  $max$  までの範囲で、全データ中、負の領域で最も観測した回数の多い  $diff$  を DIFFMIN、正の領域で最も観測した回数の多い  $diff$  を DIFFMAX とする。

## 6 実験結果

### 6.1 実験の設定

本研究では、図 5 の川の形状及び川の流れにおいて実験する。また、強化学習では、 $\epsilon$ -greedy による行動選択と Q 学習による学習方式を使用した。報酬値の設定は表 1 で示す。

ゴールする	150
コースアウト	-100
障害物に衝突	-100
転覆	-100
基本行為を一回行う	-1

表 1: 報酬値の設定

4.2 節では、シミュレーション上の安定性を示した。本実験では式 (3) の  $I$  を 60 とし、カヌーの傾き  $\theta$  が以下を満たすとき、カヌーは転覆しないものとする。

$$|\theta| \leq 60$$

実験では 30000 回エピソードを繰り返し、強化学習を行った。強化学習後に、その結果を描画する。また、DIFFMIN と DIFFMAX の更新は、強化学習中に 150 回転覆することに行う。これは、環境変化がエージェントに与える影響を分析するにあたり、ある程度のデータ数が必要であるためである。実世界においても、一度や二度の経験から学習し、行為に反映、及び改善することは非常に困難であり、学習結果を反映させることができるだけの経験数、あるいは経験値が必要である。従って、仮想世界においても、一定数のデータを集約した後に行為選択に反映させることは、妥当であると考えられる。

## 6.2 実行結果

30000回の強化学習後のシミュレート画像を図8で示す。表2は、強化学習中に得られたデータを分析して更新したことによる  $DIFFMIN$  と  $DIFFMAX$  の変化を表す。表2の結果から、本実験での最終的な引き込み領域が定まる(式(6))。

$$\begin{aligned} D &< -11.777095536482065 \\ D &> 20.6674360021526 \end{aligned} \tag{6}$$

15-19ページに、描画中のエージェントの行為選択の結果を表す Jason のコンソール出力を示す。出力中、「aaa」はエージェントの名前であり、その右の出力はそれぞれ以下のことを表す。

- `cpos(loc(X,Y))`  
エージェントの位置を  $xy$  座標で表す。 $X$ 、 $Y$  がそれぞれエージェントの  $x$  座標と  $y$  座標を示す。
- `attract_mn(引数)`  
引数が  $DIFFMIN$  を表す。
- `attract_mx(引数)`  
引数が  $DIFFMAX$  を表す。
- `list_diff(引数)`  
引数が、その時の  $diff$  を表す。
- `select is N`  
 $N$  は選択した基本行為を表す。0、1、2、3、4、はそれぞれ、5.3節の「何もしない」、「前進」、「後退」、「右進」、「左進」を示す。

15-19ページの赤字の部分が、エージェントが引き込み領域にあり、状態立て直し行為を選択していることを示している。

転覆データ数	$DIFFMIN$	$DIFFMAX$
0	-60.0	60.0
150	-12.342050065761775	20.6674360021526
300	-11.777095536482065	20.6674360021526

表 2:  $DIFFMIN$  と  $DIFFMAX$  の変化

図8中のエージェントの行動選択の結果

```
[aaa] cpos(loc(330,40))
[aaa] attract_mn(-11.777095536482065)
[aaa] attract_mx(20.6674360021526)
[aaa] list_diff(12.927896817297396)
```

```

[aaa] select is 4
左へ曲がる true
X=324.14213562373095   Y=94.14213562373095

[aaa] cpos(loc(324.14213562373095,94.14213562373095))
[aaa] attract_mn(-11.777095536482065)
[aaa] attract_mx(20.6674360021526)
[aaa] list_diff(-19.109939174458958)
[aaa] you're danger
[aaa] attractLeft
右へ立て直す true
X=329.14213562373095   Y=119.14213562373095

[aaa] cpos(loc(329.14213562373095,119.14213562373095))
[aaa] attract_mn(-11.777095536482065)
[aaa] attract_mx(20.6674360021526)
[aaa] list_diff(4.087815458026428)
[aaa] select is 3
右へ曲がる true
X=330.0   Y=148.2842712474619

[aaa] cpos(loc(330,148.2842712474619))
[aaa] attract_mn(-11.777095536482065)
[aaa] attract_mx(20.6674360021526)
[aaa] list_diff(11.96044578501432)
[aaa] select is 3
右へ曲がる true
X=330.85786437626905   Y=177.42640687119285

[aaa] cpos(loc(330.85786437626905,177.42640687119285))
[aaa] attract_mn(-11.777095536482065)
[aaa] attract_mx(20.6674360021526)
[aaa] list_diff(11.661581203903433)
[aaa] select is 1
前進する true
X=360.0   Y=206.5685424949238

[aaa] cpos(loc(360,206.5685424949238))
[aaa] attract_mn(-11.777095536482065)
[aaa] attract_mx(20.6674360021526)
[aaa] list_diff(-0.4376925671889893)
[aaa] select is 4
左へ曲がる true
X=344.14213562373095   Y=265.71067811865476

[aaa] cpos(loc(344.14213562373095,265.71067811865476))
[aaa] attract_mn(-11.777095536482065)
[aaa] attract_mx(20.6674360021526)

```

```
[aaa] list_diff(-16.00145883669216)
[aaa] you're danger
[aaa] atractLeft
右へ立て直す true
X=359.14213562373095   Y=310.71067811865476

[aaa] cpos(loc(359.14213562373095,310.71067811865476))
[aaa] atract_mn(-11.777095536482065)
[aaa] atract_mx(20.6674360021526)
[aaa] list_diff(9.579269823159812)
[aaa] select is 4
左へ曲がる true
X=353.2842712474619   Y=339.8528137423857
[aaa] cpos(loc(353.2842712474619,339.8528137423857))
[aaa] atract_mn(-11.777095536482065)
[aaa] atract_mx(20.6674360021526)
[aaa] list_diff(-11.26474305075878)
[aaa] select is 4
左へ曲がる true
X=347.42640687119285   Y=368.99494936611666

[aaa] cpos(loc(347.42640687119285,368.99494936611666))
[aaa] atract_mn(-11.777095536482065)
[aaa] atract_mx(20.6674360021526)
[aaa] list_diff(-13.368052880247758)
[aaa] you're danger
[aaa] atractLeft
右へ立て直す true
X=332.42640687119285   Y=383.99494936611666

[aaa] cpos(loc(332.42640687119285,383.99494936611666))
[aaa] atract_mn(-11.777095536482065)
[aaa] atract_mx(20.6674360021526)
[aaa] list_diff(2.6052724537566654)
[aaa] select is 2
後進する true
X=331.5685424949238   Y=413.1370849898476
[aaa] cpos(loc(331.5685424949238,413.1370849898476))
[aaa] atract_mn(-11.777095536482065)
[aaa] atract_mx(20.6674360021526)
[aaa] list_diff(-1.5366518531798903)
[aaa] select is 4
左へ曲がる true
X=315.71067811865476   Y=472.27922061357856
```

```

[aaa] cpos(loc(315.71067811865476,472.27922061357856))
[aaa] attract_mn(-11.777095536482065)
[aaa] attract_mx(20.6674360021526)
[aaa] list_diff(-12.603063132468495)
[aaa] you're danger
[aaa] attractLeft
右へ立て直す true
X=330.71067811865476   Y=517.2792206135786

[aaa] cpos(loc(330.71067811865476,517.2792206135786))
[aaa] attract_mn(-11.777095536482065)
[aaa] attract_mx(20.6674360021526)
[aaa] list_diff(0.599047555035277)
[aaa] select is 1
前進する true
X=364.68434407966504   Y=538.6037759339154
[aaa] cpos(loc(364.68434407966504,538.6037759339154))
[aaa] attract_mn(-11.777095536482065)
[aaa] attract_mx(20.6674360021526)
[aaa] list_diff(-0.5614580102374589)
[aaa] select is 3
右へ曲がる true
X=365.5422084559341   Y=567.7459115576463
[aaa] cpos(loc(365.5422084559341,567.7459115576463))
[aaa] attract_mn(-11.777095536482065)
[aaa] attract_mx(20.6674360021526)
[aaa] list_diff(11.459307032586786)
[aaa] select is 4
左へ曲がる true
X=394.68434407966504   Y=596.8880471813773
[aaa] cpos(loc(394.68434407966504,596.8880471813773))
[aaa] attract_mn(-11.777095536482065)
[aaa] attract_mx(20.6674360021526)
[aaa] list_diff(-11.646242681868529)
[aaa] select is 4
左へ曲がる true
X=393.826479703396   Y=646.0301828051082

[aaa] cpos(loc(393.826479703396,646.0301828051082))
[aaa] attract_mn(-11.777095536482065)
[aaa] attract_mx(20.6674360021526)
[aaa] list_diff(-17.124857128395185)
[aaa] you're danger
[aaa] attractLeft

```

```

右へ立て直す true
X=373.826479703396   Y=691.0301828051082

[aaa] cpos(loc(373.826479703396,691.0301828051082))
[aaa] attract_mn(-11.777095536482065)
[aaa] attract_mx(20.6674360021526)
[aaa] list_diff(3.456104677081157)
[aaa] select is 1
前進する true
X=427.8001456644063   Y=709.7056274847714
[aaa] cpos(loc(427.8001456644063,709.7056274847714))
[aaa] attract_mn(-11.777095536482065)
[aaa] attract_mx(20.6674360021526)
[aaa] list_diff(-0.13511724706889794)
[aaa] select is 3
右へ曲がる true
X=428.6580100406753   Y=738.8477631085024
[aaa] cpos(loc(428.6580100406753,738.8477631085024))
[aaa] attract_mn(-11.777095536482065)
[aaa] attract_mx(20.6674360021526)
[aaa] list_diff(12.459841015020423)
[aaa] select is 3
右へ曲がる true
X=429.51587441694437   Y=767.9898987322333
ゴールしました。
[aaa] test

```

図 8 中のエージェントの行動選択の結果

このように、エージェントは引き込み領域の範囲を自ら推定し、その領域にあれば、安定する方向に行為を選ぶことで、安定した行動を取ることができている。

## 7 考察

本章では、6章の結果についての考察と今後の課題について述べる。

図 9 は、30000 回強化学習を繰り返した時の、最終的な *diff* の分布を表したグラフである。なお、本実験では 30000 回のエピソードのうち計 403 回転覆した。図 7 と同様に、*x* 軸は転覆時の *diff*、*y* 軸はその *diff* を示した回数である。*y* 軸の値が大きいほど、その時転覆した回数が多いことを示す。すなわち、*diff* がいかなる時に転覆しやすいかを読み取ることができる。

表 2 から、150 回転覆するごとに、転覆時の *diff* の分布を分析し、*DIFFMIN* と *DIFFMAX* を更新できていることが分かる。エピソード開始時は *DIFFMIN* = -60.0、*DIFFMAX* = 60.0 とした

$$D < -60.0$$

$$D > 60.0$$

上記の範囲が引き込み領域であったが、*DIFFMIN* と *DIFFMAX* の更新により、引き込み領域の範囲を狭め、シミュレートの過程でより望ましい範囲に再定義することができた。これにより、実際の行為選択の過程において、エージェントが引き込み領域に遷移した際に状態立て直し行為を選択しており (p.18)、安定性の概念の導入と、身体性エージェントが安定性を保持しながら問題解決を目指すシミュレートが実現できた。

## 7.1 強化学習のみの実行結果との比較

本節では、状態の立て直し行為を行わず、安定性の保持を考慮せずに強化学習の結果のみに基づいてシミュレートした結果と、安定性を保持しながら問題解決を目指す 6 章の結果を比較、考察する。

6 章の比較実験として、30000 回エピソードを繰り返し、状態の立て直し行為を行わずに問題解決を目指すシミュレーションを行った。この比較実験の結果を描画したものが図 10、また、最終的な *diff* の分布が図 11 である。なお、30000 回のエピソードのうち、転覆した回数は計 589 回である。

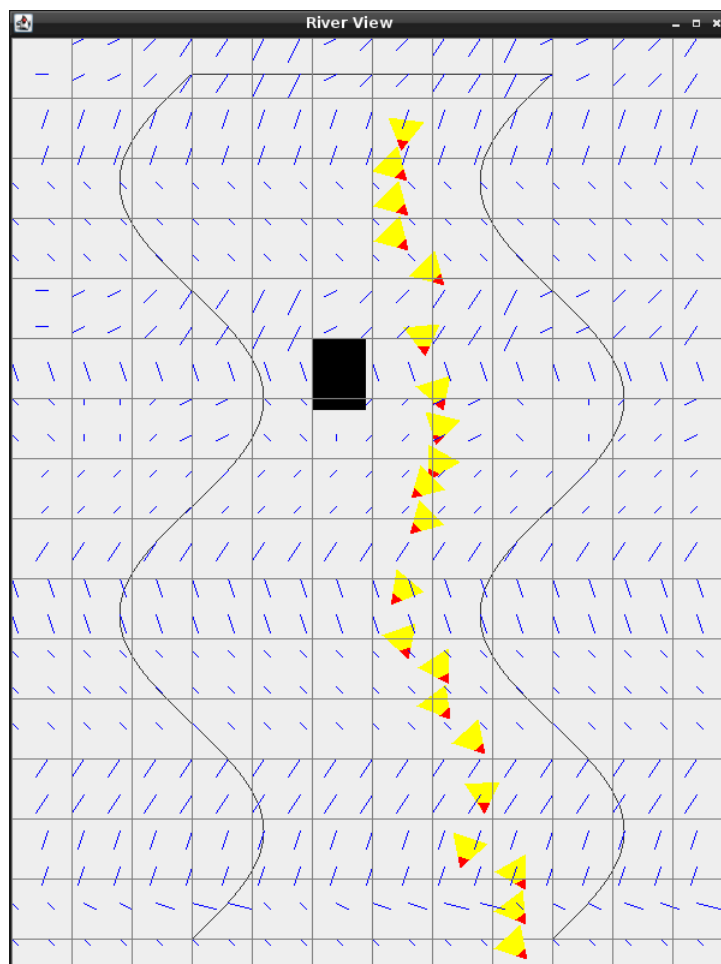


図 8: 30000 回強化学習後のシミュレート画像

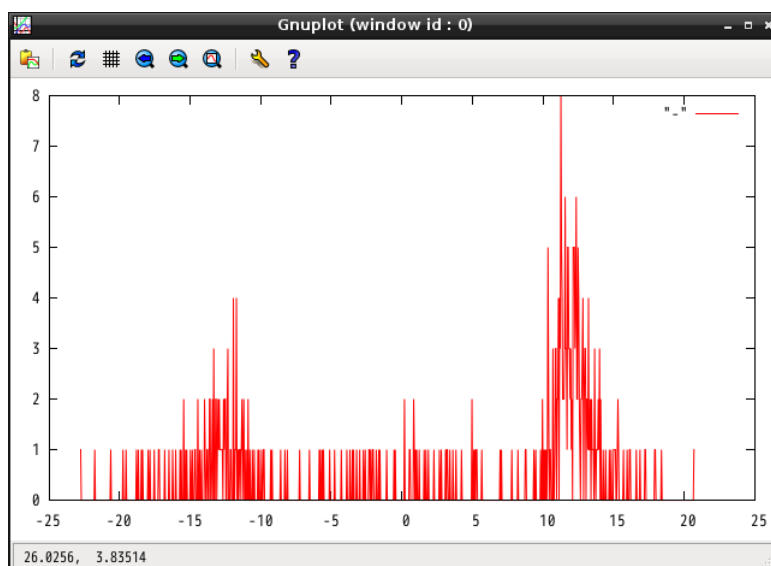


図 9: エピソード 30000 回中 403 回転覆時の *diff* の分布

図 8,10 を見ると、どちらも最終的にゴールできていることが分かる。また、図 9,11 を比較すると、最終的な *DIFFMIN* と *DIFFMAX* の最適値は、それほど違いがないことが読み取れる。しかしながら、エピソード 30000 回中の転覆回数は、安定性の保持を考慮したシミュレーション（6 章）では計 403 回、安定性の保持を考慮しないシミュレーション（比較実験）では計 589 回となり、結果が大きく異なった。従って、安定性の保持を考慮して行為選択をすることで、シミュレートの結果を改善することができたといえる。

## 7.2 引き込み領域の獲得に関する分析

安定性の概念として重要な引き込み領域の定義について、図 9 の負の領域を見ると、*diff* が  $-12.0$  付近を示した回数が多い。本実験の最終的な *DIFFMIN* は  $-11.777095536482065$ （表 2）であることから、*DIFFMIN* を適切な値に更新できているといえる。一方で、図 9 の正の領域では、*diff* が  $11.5$  付近を示した回数が多く、本実験で得られた最終的な *DIFFMAX* である  $20.6674360021526$ （表 2）とは差が見られた。

30000 回のエピソードの結果から、*D* が  $11.5$  付近の時にカヌーが転覆する可能性が高いことを図 9 は示しており、本来は *DIFFMAX* もこの結果に準じた  $11.5$  付近の値を取ることが望ましい。すなわち、実験では 300 回転覆した時点では *DIFFMAX* の最も望ましい値として  $20.6674360021526$  が得られ、引き込み領域を式 (6) としたものの、その後も転覆を繰り返して得られたデータも考慮すると、実際には *DIFFMAX* のより望ましい値は約  $11.5$  であったといえる。また、これに伴い引き込み領域は

$$D < -11.777095536482065$$

$$D > \text{約 } 11.5$$

が、より適切な範囲となる。

従って、本実験で得られた引き込み領域の定義（式 (6)）は、その精度としては十分なものであるとはいえず、今後この精度を高めるため検討が必要であると考えられる。

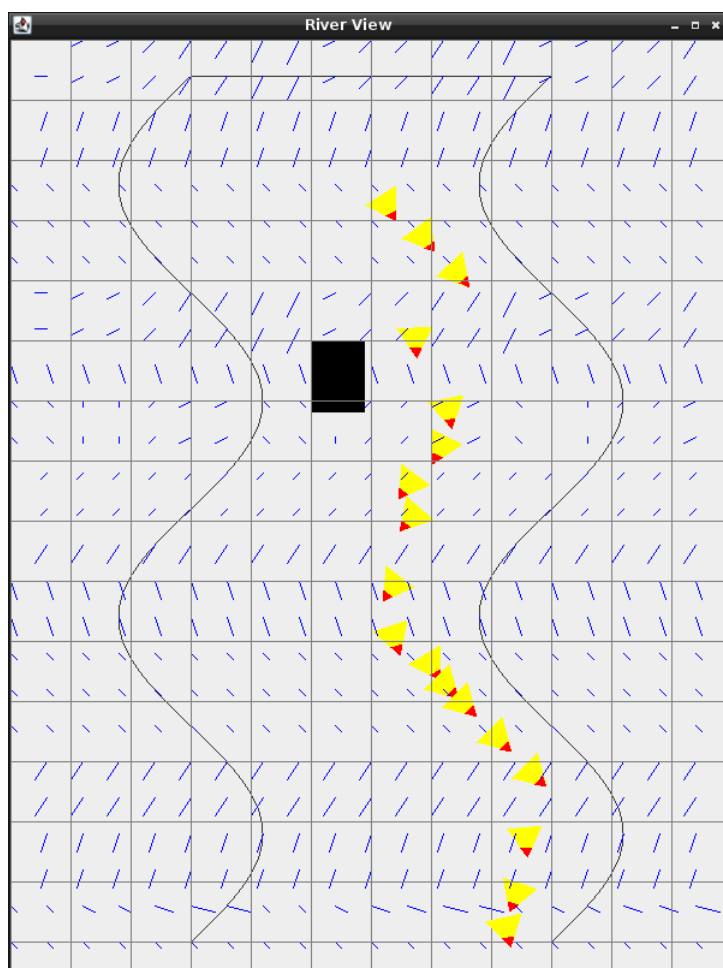


図 10: 30000 回強化学習後のシミュレート画像 (状態の立て直し行為は行わない)

上述の問題についての解決案を、以下に述べる。

- データの分析を行う頻度、及び引き込み領域の再定義を行う頻度を上げる。

本実験では、環境外乱による影響に関するデータ (転覆データ) の分析を、150 のデータ数ごとに行った。実世界においても、一定量の経験数や経験値がなければそれを行為に反映することは難しく、データを分析するためにはある程度のデータ数が必要である。しかしながら、分析を行う頻度が低いと、その分、少数のデータを行為に反映することができないという問題がある。そこで、データの分析頻度を上げ、少数のデータを反映することができれば、引き込み領域の精度も上げることができると考える。

一方で、過度にデータの分析を行っても、データ数が少なければ分析結果、例えば *diff* の分布に有意差がみられず、*DIFFMIN* や *DIFFMAX* を推定することができない。また、分析するデータが偏っていれば、分析結果も偏ったものとなり、望ましい結果を得られないことも考えられる。

従って、データの分析を行い引き込み領域を再定義する頻度については、上記の問題を考慮した上で、最適な頻度を導き出すことが重要である。これについては、まず、分析を行うタイミングを 50 回ごと、150 回ごと、300 回ごとなど頻度を変えて実験を行い、

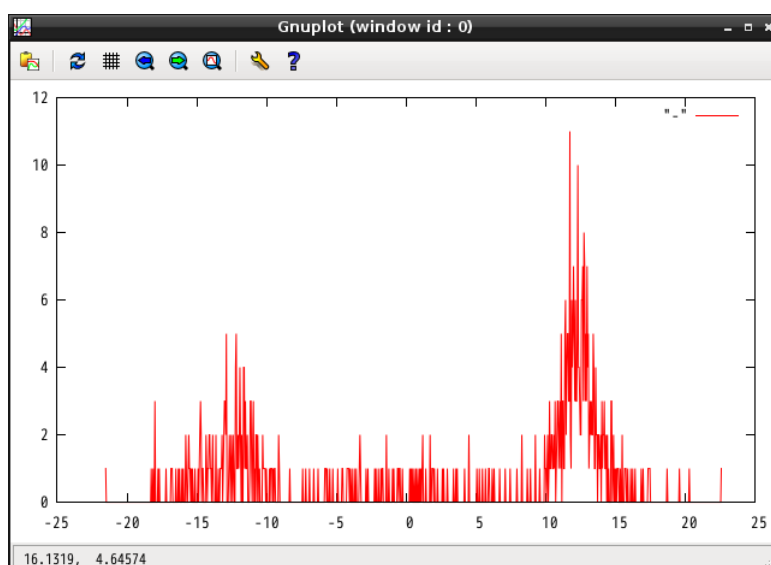


図 11: エピソード 30000 回中 589 回転覆時の  $diff$  の分布 (状態の立て直し行為は行わない)

それが引き込み領域の定義にどう変化を与えるかについて観察する必要がある。

- $diff$  の集計上の刻みを再検討する。

本実験では、 $diff$  は非常に細かい刻みで集計している。これは、そうすることにより、例えば  $diff$  を整数のみからなる値とする時よりも、 $diff$  の値、そして引き込み領域をより細かく定めることができると考えたためである。 $diff$  の取りうる値をより詳細に表現することができれば、データ分析時にそれぞれの  $diff$  の小さな差を考慮することができるという長所もあり、値を細かく取ることは仮想世界でシミュレートする上で意義があると考えていた。

しかしながら、本実験の結果を考慮すると、 $diff$  はあまり細かい刻みで集計する必要性はなく、むしろ現状よりも  $diff$  の値を大きめの刻みで集計した方が、データ分析時においても都合が良いという結論に至った。表 3 は、図 9 のグラフ作成時に使用したデータの一部抜粋である。表 3 を見ると、図 9 のグラフプロット時に  $x$  軸の値を非常に細かく取っていることが分かる。

$diff$  の取りうる値が細かいほど、データ分析時における、 $diff$  の刻みごとの回数 (図 9 の  $y$  軸) の差が小さく、これを解決し差をつけるため、分析する際に多くのデータ数を要することになる。そこで、 $diff$  の集計の刻みをもう少し大きくすれば、150 回ごとのデータ分析だとしても、一回の分析によって得られる結果がより正確なものになり、最終的な引き込み領域の定義にも変化が有るのではと考える。

## 8 終わりに

本論文では、シミュレーションに安定性の概念を導入し、エージェントが行為を繰り返しながら環境から受ける影響を分析することで、自己の安定性を確立し、環境に適応しながら目標

$diff$ (図9の $x$ 軸)	$diff$ を観測した直後に転覆した回数 (図9の $y$ 軸)
-22.74643508396829	1
-22.703002575241584	0
-22.659570066514878	0
-22.61613755778817	0
-22.572705049061465	0
-22.52927254033476	0
⋮	⋮
-13.755905777540127	1
-13.71247326881342	0
-13.669040760086714	0
-13.625608251360008	2
-13.582175742633302	2
-13.538743233906596	0
-13.49531072517989	0
-13.451878216453183	1
-13.408445707726477	2
-13.36501319899977	0
-13.321580690273064	3
-13.278148181546358	0
-13.234715672819652	0
-13.191283164092946	2
-13.147850655366241	2
-13.104418146639535	0
-13.060985637912829	1
-13.017553129186123	2
-12.974120620459416	1
-12.93068811173271	2
-12.887255603006004	1
-12.843823094279298	1
-12.800390585552591	1
-12.756958076825885	1
-12.713525568099179	1
-12.670093059372473	0
⋮	⋮

表 3: 図 9 プロット時に使用したデータの一部抜粋

達成するシミュレーションの実装について述べた。本研究において、身体性を有するエージェントが、動的に変化する環境下で自律的に安定性を保持することを、シミュレート上で表現することが可能となった。

今後は 7 章で述べた問題について取り組むことが課題であり、最終的な理想である完全エー

エージェントを実現したシミュレーションを目指し、自律型ロボットの研究の発展に貢献したい。

## 謝辞

本論文の執筆及び研究にあたり、指導教官の新出尚之准教授には丁寧なご指導と助言を賜りますと共に、多くの知識や示唆を頂きました。また、諏佐歩さんを始め、共同で研究を進めた同輩と後輩の方々の力添えを頂きました。皆様方に心から感謝の気持ちとお礼を申し上げたく、謝辞にかえさせていただきます。

## 参考文献

- [1] 高田司郎, 新出尚之. 行為のアトラクター状態を考慮した知能ロボットについて. In Proc. of JAWS2012, 2012.
- [2] 高田司郎, 新出尚之, 濱砂幸裕, 波部斉, 藤田恵. アトラクター状態を用いた実世界における基本行為の学習について. 情報処理学会研究報告 2013-MPS-92, No. 24, pp. 1-6 (2013), 2013.
- [3] 高田司郎, 新出尚之. アトラクター状態を用いた基本行為の頑健性について. In Proc. of JAWS2014, 2014.
- [4] 辰野勇. カヌー&カヤック入門. 山と溪谷社, 2005.
- [5] Richard S.Sutton, Andrew G.Barto (著), 三上貞芳, 皆川雅章 (訳). 「強化学習」. 森北出版株式会社, 1998.
- [6] Rolf Pfeifer, Christian Scheier (著), 石黒章夫, 小林宏, 細田耕 (訳). 「知の創成」. 共立出版株式会社, 2001.
- [7] R.Pfeifer, J.Bongard (著), 細田耕, 石黒章夫. 「知能の原理」. 共立出版株式会社, 2010.
- [8] 土木学会関東支部. カヌーの浮遊時の安定について. [https://www.jsce.or.jp/branch/kanto/01\\_07\\_koho/kanu/21/investigation\\_antei.pdf](https://www.jsce.or.jp/branch/kanto/01_07_koho/kanu/21/investigation_antei.pdf). (2015/11/29)
- [9] 社団法人 日本船舶海洋工学会. 船体運動力学. <http://www.jasnaoe.or.jp/publish/others/dl/sentaiundourikigaku.pdf>. (2015/1/25)
- [10] 濱田百合. 連続世界におけるエージェントの学習と意思決定に向けて. 奈良女子大学理学部情報科学科 2012 年度卒業論文, 2013.
- [11] 久妻さゆり. 実世界におけるエージェント構築に向けたシミュレーション環境の作成. 奈良女子大学理学部情報科学科 2012 年度卒業論文, 2013.
- [12] 亀村美佳. エージェントの知覚と学習による行動決定の実現. 奈良女子大学理学部情報科学科 2013 年度卒業論文, 2014.
- [13] 林果歩. 連続的なシミュレーション環境でのエージェントの学習と意思決定の実装について. 奈良女子大学理学部情報科学科 2013 年度卒業論文, 2014.

- [14] 宮田玲奈. カヌーレーシングのシミュレーション環境における強化学習について. 奈良女子大学理学部情報科学科 2014 年度卒業論文, 2015.
- [15] 柚木静香. 実世界のエージェント構築のためのシミュレーション環境の実現. 奈良女子大学理学部情報科学科 2014 年度卒業論文, 2015.
- [16] 亀村美佳. BDI エージェントによる連続的な仮想世界におけるシミュレーションの実装. In Proc. of JSAI2015, 2015.
- [17] 諏佐歩. 行為のアトラクター状態を考慮したカヌーシミュレータ環境の構築. 奈良女子大学理学部情報科学科 2015 年度卒業論文, 2016.

## A パピーの実験

ここでは、3.1.2 章で記述したパピーの実験 [7, p77,102-103] について述べる。パピーの実験では、以下に述べるパピーの走行時の運動パターンを観測する実験である。

### パピーの形態

「パピー」は、チューリッヒ大学の人工知能研究室で研究していた飯田史也によって開発された四脚ロボットである。非常に単純な設計であるにも関わらず、その振る舞いは驚くほど生物らしく安定である。パピーは、それぞれの脚の付け根と膝、足首に関節が一つずつ、体全体で 12 個の関節を持つ。四脚動物の筋腱システムを参考にし、各脚には、すねと腿の部分を繋ぐばねがある。背骨には弾性を持ち上下に変形することができる金属板が用いられている。また、足裏には圧力センサがあり、足が地面に着いたことがわかる。この圧力センサによって、脚先で観測される圧力パターンから、パピーの移動パターンを判別することができる。

パピーの胴体と脚、脚先は全てアルミニウムでできているため、どんな地面でも多少滑り、走行の際に、この滑りが安定性に貢献している。そのため、脚先にゴムパッドをつけ摩擦を増やすと、かえって転倒しやすくなる。

### パピーの制御と動作

制御は非常に単純であり、足の付け根の関節は、前後に繰り返し運動をする。パピーを地面に置くと、最初は床をひっかいているだけだが、何度か繰り返しているうちに比較的スムーズな走行パターンへと移行する。これは、周期的運動とロボットの形や取り付けられたばね、足裏の摩擦、そして重力の相互作用によるものである。自然な走行リズムに落ち着くと、ロボットと環境の相互作用により、ある特定の運動パターンが創発する。そして、4 本の脚が地面から同時に離れる浮遊層と接地相を繰り返す。

この実験では、モータを好きな速度で回すことはできても、任意の速度で走らせることはできない。これは、ある速度領域ではロボットは安定に移動することができるが、ある領域では移動が不規則になったり、転倒したりするためである。