

2023 年度 卒業論文

生体信号を用いた格闘ゲーム AI の 難易度調整

法政大学 理工学部 応用情報工学科

荻野晴大 (20X3030)

指導教員	提出確認日	印
平原誠		

目次

第1章 はじめに	1
第2章 格闘ゲーム	1
第3章 従来研究	2
第4章 本研究のアイデアと実験概要	2
4-1 本研究のアイデアと提案手法	2
4-2 使用する AI	3
4-3 AI の強さの調整	3
4-4 本研究の実験の概要	3
第5章 実験環境	4
5-1 測定機器	4
5-2 ゲームプレイ環境	4
第6章 脳波を用いた難易度調整 (1 パラメータ)	4
6-1 被験者 A での実験	5
6-1-1 推定実験及びその結果	5
6-1-2 検証実験及びその結果	6
6-1-3 考察	8
6-2 被験者 B での実験	9
6-2-1 推定実験及びその結果	9
6-2-2 検証実験及びその結果	10
6-2-3 考察	11
6-3 今後の展開	12
6-4 同日中での実験	13
6-4-1 推定実験及びその結果	13
6-4-2 検証実験及びその結果	13
6-4-3 考察	15
6-5 今後の展開	16
第7章 脳波を用いた難易度調整 (2 パラメータ①)	16
7-1 被験者 A での実験	16
7-1-1 推定実験及びその結果	17
7-1-2 検証実験及びその結果	18
7-1-3 考察	20
第8章 AI の仕様変更	21
第9章 脳波を用いた難易度調整 (2 パラメータ②)	21
9-1 被験者 A での実験	22

9-1-1	推定実験及びその結果	22
9-1-2	検証実験及びその結果	23
9-1-3	考察	25
9-2	被験者 C での実験	26
9-2-1	推定実験及びその結果	26
9-2-2	検証実験及びその結果	27
9-2-3	考察	29
9-3	被験者 D での実験	30
9-3-1	推定実験及びその結果	30
9-3-2	検証実験及びその結果	31
9-3-3	考察	33
第 10 章	独自の集中指標	34
第 11 章	独自の集中指標を用いた難易度調整 (2 パラメータ)	35
11-1	被験者 A での実験	35
11-1-1	推定実験及びその結果	36
11-1-2	検証実験及びその結果	37
11-1-3	考察	39
11-2	被験者 C での実験	40
11-2-1	推定実験及びその結果	40
11-2-2	検証実験及びその結果	41
11-2-3	考察	43
11-3	被験者 D での実験	44
11-3-1	推定実験及びその結果	44
11-3-2	検証実験及びその結果	46
11-3-3	考察	48
第 12 章	実験結果まとめ	49
第 13 章	比較実験	52
13-1	実験概要	52
13-2	接戦 AI の仕様	53
13-3	被験者 A での実験及びその結果	55
13-4	被験者 C での実験及びその結果	57
13-5	被験者 D での実験及びその結果	58
13-6	比較実験の全体結果と考察	60
第 14 章	総括	61
14-1	結論	61
14-2	今後の課題	62

謝辭	64
参考文献.....	64
付録	66

第1章 はじめに

より楽しいゲーム体験を提供するためには、プレイヤーに没入感やゾーンを感じさせることが重要であると Chen は提唱している[1]。この Chen の理論は Csikszentmihalyi が提唱したフロー理論[2]に基づいて考えられており、フロー理論における個人のスキルレベルに応じた難易度の課題が没入感やゾーンを感じさせるために重要であるという部分が、ゲームにおいても同様に重要であるとしている。これはゲームが難しすぎると挫折してしまい、簡単すぎると退屈してしまい、丁度いい難易度であると没入できるということを意味する。また没入感やゾーンの体験がゲームに対する評価に大きく関わっているとしていて、これは、よりのめり込んで遊んだゲームほど、より楽しかったと評価されるということである。

一般的に格闘ゲームを含む対戦ゲームにおいては、対戦相手の強さがゲームの難易度にあたる。

プレイヤーが自身のスキルレベルに応じた難易度のゲームを体験した際に、高い集中度合を発揮するならば、逆にプレイヤーの集中度合を測定することで、プレイヤーに最適な難易度が予測出来るのではないかと考えた。また集中度合はプレイヤーの体の状態や挙動に反映されるのではないかと考えた。

そこで本研究では格闘ゲームにおいてプレイヤー個人の生体信号を基にスキルレベルに適した難易度を予測し、難易度を最適に調整することでゲームを楽しんでもらうことを目的とする。

第2章 格闘ゲーム

本研究では格闘ゲームとして立命館大学制作の「FightingICE」を使用した[3]。図1に示すように、1対1で戦うゲームであり、1試合3ラウンドで2ラウンド取ったら勝利である。制限時間は1ラウンドに1分である。キャラクターの初期体力(HP)は互いに300であり、攻撃を相手に当てることで、相手のHPを減らしていく。先にHPが0になった方の敗北である。また制限時間の1分が経過してしまった場合は、HPの少なかった方の敗北となる。様々な行動で蓄積される「ENERGY」を消費して繰り出す技も存在する。

本研究ではプレイヤーの対戦相手としてAIを使用した。プレイヤーの操作するキャラクターは「GARNET」とし、キャラクター上部に「P1」と表示される。AIの操作するキャラクターは「ZEN」とし、キャラクター上部に「P2」と表示される。「FightingICE」の詳細は付録のセクションBに示す。



図1 ゲーム画面

第3章 従来研究

石原ら[4]はゲーム内のデータのみを用いて、プレイヤーとAIのキャラクターのHPの差を0に近づけ、接戦になるように難易度を調整した。この手法によって制御されるAIとの対戦は、純粋に勝利を目指すAIとの対戦よりも楽しいと評価された。

第4章 本研究のアイデアと実験概要

本章では格闘ゲーム、プレイヤー及び集中状態についての筆者の考え、本研究で使用するAIの仕様と強さの調整方法、実験の概要と提案するAIの難易度調整の手法を示す。

4-1 本研究のアイデアと提案手法

上記の従来研究[4]の方法では試合を接戦に近づけることしかできず、勝てる相手と戦って勝利することが楽しいプレイヤーや、強くなるために自身より強い相手が欲しいプレイヤーなど、様々なプレイヤーの需要には応えられないと考えられる。

本研究ではゲーム内データではなく、ゲームをプレイするプレイヤーの生体信号を基に難易度を調整する方法、すなわちAIの強さを当該プレイヤーに合わせて調整する方法を新たに提案する。

強い行動を頻繁に取るAIとの対戦を好むプレイヤーもいれば、強い行動も弱い行動もバランスよく取るAIとの対戦を好むプレイヤーもいる。プレイヤーそれぞれに最適なAIの強さがあるだろう。プレイヤーにとって最適な強さのAIと対戦しているときは、集中度合が高くなるということが期待できる。人の集中度合を直接把握することは不可能であるが、人の集中度合が変化したときは身体に、その影響が現れることがある。このことから本研究

では、集中度合が反映されていると考えられる生体信号を測定対象とする。

AI との対戦時に測定されたプレイヤーの生体信号から集中度合を評価し、関数近似によって当該プレイヤーにとって最適と予想される AI の強さを導き出すという手法が、本研究において提案する生体信号を用いた格闘ゲーム AI の難易度調整の手法である。

以降は簡略のためにプレイヤーの集中度合を測定すると表現する。

4-2 使用する AI

本研究で使用する AI はキャラクター間の距離を 6 種類、プレイヤーのキャラクターの状態を 56 種類、AI のキャラクターの位置（空中または地上）を 2 種類で識別する。よって空中、地上のそれぞれで、計 $6 \times 56 = 336$ 種類をゲーム状況として識別する。また、AI のキャラクターの ENERGY がどのくらいあるかも把握しており、残量に応じて選択可能な行動は変わる。但し、公式の AI コンペティションのレギュレーションに則り、AI が識別するゲーム状況は 15 フレーム（0.25 秒）前のものとした。それぞれの状況において取り得る AI の行動には、空中行動 15 種類もしくは地上行動 25 種類がある。AI の行動決定には計 $336 \times (15 + 25) = 13,440$ 個の評価値を用いた。この評価値は事前に自己対戦方式にて学習したものである。評価値は 0~10 の範囲で変動し、評価値が高いほど強い行動となっている。AI の学習や仕様の詳細は付録のセクション C-1 に示す。

4-3 AI の強さの調整

AI の強さの調整は、正規乱数における平均と標準偏差のパラメータを調整することで実現する。AI が行動を決定する際に、正規乱数を生成し、現在のゲーム状況において選択可能な行動の中から、乱数値に最も近い評価値を持つ行動を選択するように設定した。これによって正規乱数の平均と標準偏差という 2 つのパラメータを変えることで、AI の行動の平均的な強さやランダム性が変化する仕組みとなっている。

4-4 本研究の実験の概要

個々のプレイヤーに対して AI の強さを調整するが、調整のために一定数の試合を行ってもらう必要があり、また調整が上手くできているかを検証するためにも一定数の試合を行ってもらう必要がある。実験の途中でプレイヤーが著しく上達してしまうと実験結果の評価が正しく行えないので、プレイヤーには事前練習として 30 分以上ゲームをプレイしてもらい、ゲームの実力をある程度高めてもらった。

正規乱数の平均と標準偏差を変えることで AI の強さを変化させて、AI と対戦する際のプレイヤーの集中度合を測定すれば、プレイヤーの集中度合がピークになる条件が見つめられると想定した。よって、本研究では集中度合をガウス関数で近似する。ガウス関数近似の詳細は付録のセクション E に示す。

本実験は推定実験と検証実験の 2 つから構成される。

推定実験ではプレイヤーに複数の条件の AI と対戦してもらう。各条件の AI との対戦で測定された集中度合をガウス関数で近似する。近似式からプレイヤーの集中度合が最大になると考えられる条件（最適条件）を推定する。

検証実験では推定実験により推定された最適条件の AI と試合をしたときに、プレイヤーの集中度合が最大になるのか、加えて楽しい試合をプレイヤーに体験させることができるのかを確かめる。そのため、プレイヤーに最適条件を含む複数の条件の AI と対戦してもらい、集中度合を測定する。また試合に関するアンケートに 1 試合ごとに答えてもらう。

第 5 章 実験環境

本章では実験を行うために必要な集中度合を測定する機器やゲームプレイの環境について具体的に示す。

5-1 測定機器

集中度合が反映されていると考えられる生体信号の 1 つとして脳波があり、これを本研究での測定対象とする。脳波からの集中度合はニューロスカイ社の脳波計 MindWaveMobile2 と専用ソフトウェア NeuroView で測定した。測定された集中度合を表す数値（Attention）を指標とした。なお Attention は 1 秒ごとに数値が記録されるので、試合中に記録された数値の平均（平均 Attention）を指標とした。測定機器の詳細は付録のセクション A に示す。

5-2 ゲームプレイ環境

モニターは JAPANNEXT 社の JN-27VG165FHDR、音声出力は上記モニターの内蔵スピーカーを使用し、音量は 50 とした。ウィンドウサイズはデフォルト設定の 960×640 とした。

本研究で使用している格闘ゲーム「FightingICE」は PC で動作し、キャラクター操作はキーボード入力でのみ可能である。しかし、キーボード入力でのゲームプレイは一般的ではなく、慣れるまでの事前練習が更に長く必要になると判断し、ゲームコントローラーでの入力でのキャラクター操作を行うこととした。ソニーインタラクティブエンタテインメント社のゲームコントローラーである DualShock4 と free Colors 社のソフトウェアである JoyToKey によって、ゲームコントローラーの入力にキーボード入力を対応させることが可能になる。ゲームコントローラーの詳細は付録のセクション A に示す。

第 6 章 脳波を用いた難易度調整（1 パラメータ）

AI の強さを変化させるための調整項目として正規乱数の平均と標準偏差があるが、AI の強さにより大きく影響しているのは平均の方だと考えられる。また実験を簡易にして実施しやすくするために、まずは正規乱数の標準偏差は固定値として平均のみの 1 パラメータ

の最適化を試みる。本章では Attention に基づいて、正規乱数の平均のみを個々のプレイヤーに最適な値に調整する実験について示す。被験者 2 名 (A, B) に実験を行った。なお、被験者 A は筆者である。

本章の推定実験では、以下の 4 つの正規乱数の条件下で行われた各試合の平均 Attention をガウス関数近似して、平均 Attention が最大になる正規乱数の条件 (Attention 最適条件) を推定する：①平均値 3 標準偏差 1，②平均値 5 標準偏差 1，③平均値 7 標準偏差 1，④平均値 9 標準偏差 1。

実験当日、プレイヤーには操作練習を 3 分間行ってもらった。その後、上記の 4 条件の AI とランダムな順番で 1 試合ずつ対戦してもらった。1 試合ごとに休憩を 3 分取ってもらった。4 条件下で行われた各試合の平均 Attention をガウス関数で近似し、Attention 最適条件を推定した。ガウス関数近似の詳細は付録のセクション E に示す。

本章の検証実験では、Attention 最適条件を含む 5 条件で 2 試合ずつ行ってもらい、平均 Attention が最大になるか、加えて楽しい試合を体験させられたかを確認した。なお検証実験の 5 条件は個々のプレイヤーによって異なる。

操作練習を 3 分間行ってもらった後、5 条件の AI とランダムな順番で 2 試合ずつの計 10 試合を行ってもらった。1 試合ごとにアンケートに回答してもらい、その後に休憩を取ってもらった。アンケートと休憩を合わせて 3 分とした。

アンケートの内容は以下の 8 項目で、1~6 の 6 段階で 6 を最高評価として回答してもらった：①試合が楽しかったか？②相手が強かったか？③相手の行動が予測できたか？④キャラクターを思い通りに動かせたか？⑤相手の行動が多様だと感じたか？⑥ゲームに集中していたか？⑦新しいコマンドに積極的に挑戦したか？⑧勝ちたいと思ったか？

6-1 被験者 A での実験

本セクションでは被験者 A をプレイヤーとしたときの実験結果とその考察について示す。

6-1-1 推定実験及びその結果

推定実験として、上記の 4 条件下で行ってもらった各試合の平均 Attention をガウス関数で近似した。

推定実験の結果として各試合の平均 Attention，及びそれをガウス関数で近似したグラフを図 2 に示す。横軸は正規乱数の平均であり、橙色でプロットされた各点は各試合の平均 Attention (± 1 SE)を、破線は平均 Attention のガウス関数近似を、緑色でプロットされた点はガウス関数近似において平均 Attention が最大になる点を示す。

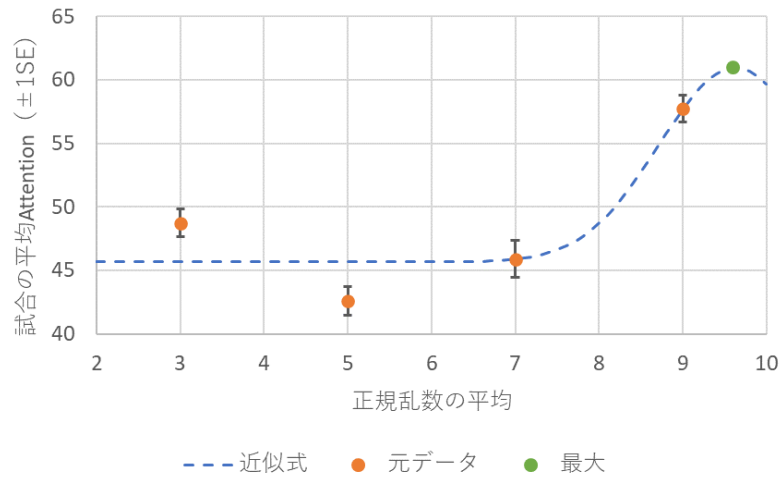


図2 推定実験結果とガウス関数近似

図2より、平均9.6標準偏差1の正規乱数がAttention最適条件となる。

6-1-2 検証実験及びその結果

推定実験より、Attention最適条件は平均9.6標準偏差1の正規乱数である。よって平均9.6標準偏差1を含む以下の5条件下で試合を行ってもらい、検証した：①平均値5標準偏差1、②平均値7標準偏差1、③平均値9標準偏差1、④平均値9.6標準偏差1、⑤平均値10標準偏差1。

検証実験の結果として推定実験で近似したガウス関数、及び検証実験の各試合の平均Attentionを図3に示す。横軸は正規乱数の平均であり、破線は推定実験でのガウス関数近似を、緑色でプロットされた点はガウス関数近似において平均Attentionが最大になる点を、赤色でプロットされた各点は検証実験での各試合の平均Attention (±1SE)を示す。

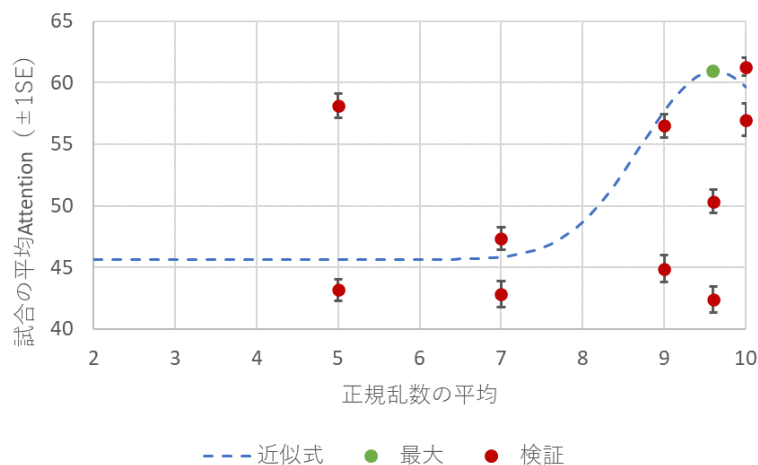


図3 検証実験結果

図3より、Attention 最適条件である平均9.6 標準偏差1の正規乱数条件のAIと対戦してもらったときに、平均Attentionが最大とならなかったことが確認できる。また、ガウス関数近似と各試合の平均Attentionとの差の絶対値は平均8.97であった。すなわち、検証実験の各試合の平均Attentionが平均8.97だけガウス関数近似から離れた値であった。

検証実験の結果としてアンケートの回答結果を表1に示す。赤字での表記はAttention最適条件での回答結果と平均Attentionを示す。回答結果の各数値は各条件での2試合分の平均である。また、平均Attentionについても各条件での2試合分の平均である。

表1 検証実験アンケート結果

正規乱数の平均		5	7	9	9.6	10
アンケート項目	・試合が楽しかったか?	4.5	5	5.5	5.5	4.5
	・相手が強いと感じたか?	3.5	4	4.5	5	5.5
	・相手の行動が予測できたか?	3.5	3.5	4.5	4	6
	・キャラクターを思い通りに動かせたか (操作ミスしなかったか)?	5	5.5	4.5	5.5	5.5
	・相手の行動が複雑(多様)だと感じたか?	5	5	4	4	3
	・試合に集中していたと思うか?	4.5	4	5	5	4
	・新しいコマンド(技)に積極的に挑戦したか?	1.5	1.5	3	1	1
	・勝ちたい(負けたくない)と思ったか?	3.5	4	4.5	5	4.5
試合の平均Attention		50.7	45.1	50.7	46.4	59.2

図3よりガウス関数近似による予測の通りにはならなかったが、表1よりアンケートの試合の楽しさについて、Attention最適条件である平均9.6 標準偏差1の正規乱数条件が5

条件のうちで最高評価となったことがわかる。また、回答結果と各試合の平均 Attention の間で相関性が見られたのは 2 項目であり、「相手の行動が予測できたか?」と「相手の行動が複雑（多様）だと感じたか?」であった。

各試合の各ラウンド終了時のプレイヤーと AI の HP の差の平均（平均 HP 差）を表 2 に示す。各数値が正のときは、平均的にラウンド終了時にプレイヤー側が HP をより多く残していたことを表す。

表 2 ラウンド終了時の平均 HP 差

正規乱数の平均	5	7	9	9.6	10
第 1 試合の平均	141.7	162.0	161.7	141.0	141.7
第 2 試合の平均	202.0	117.3	129.0	145.3	145.3
全体平均	171.8	139.7	145.3	143.2	143.5

表 2 より、キャラクターの初期 HP が 300 に設定されているので、平均的に初期 HP の半分程度をプレイヤー側がより多く残し AI に圧勝していたことがわかる。

6-1-3 考察

図 3 より、ガウス関数近似による予測の通りにはならなかった。この理由について、以下の 4 点が考えられる：

- ① 図 3 では青い破線で示されたガウス関数近似に沿うように各試合での平均 Attention が赤い点でプロットされるような結果を想定していた。しかし、そのような結果のグラフにはならず、ガウス関数近似と各試合での平均 Attention とが平均 8.97 離れていた。このことからガウス関数による近似が平均 Attention の変化に適していなかったことが理由の 1 つとして考えられる。
- ② 表 2 では正規乱数の平均を大きい値に変えたときに AI が強くなり、正規乱数の平均と HP 差の全体平均が負の相関関係になることを期待していた。しかし、そのような結果にはならず、正規乱数の平均に関わらず被験者 A が AI に圧勝していた。このことから被験者の実力に対して、AI の根本的な強さが不十分であったことが理由の 1 つとして考えられる。
- ③ 生体信号を用いてプレイヤーの集中度合を評価している以上、プレイヤーの身体の状態を一定に保ち、実験の条件を揃えることが望ましいと考えられる。しかし、時間的都合により推定実験と検証実験が異なる日に行われた。このことが理由の 1 つとして考えられる。
- ④ 正規乱数が同条件である 2 試合の間で平均 Attention の差の絶対値を求め、5 条件での平均を計算したところ、同条件の 2 試合間で平均 8.66 の差があった。このことから正規乱数が同条件であっても平均 Attention のバラつきが大きく、推定や検証のた

めの試合数が足りていないことが理由の1つとして考えられる。

表1より、アンケートの回答結果と各試合の平均 Attention の間で、相関性が見られる項目は2つあった。

1つ目は「相手の行動が予測できたか?」という項目であり、相関係数が0.884であった。この理由について、以下の点が考えられる：プレイヤーが集中して試合に取り組めば、直前のゲーム状況からAIが次にとる行動が予測できるということ。

2つ目は「相手の行動が複雑(多様)だと感じたか?」という項目であり、相関係数が-0.749であった。この理由について、以下の点が考えられる：上記の行動の予測と同様に、プレイヤーが集中して試合に取り組めば、AIが次にとる行動が予測でき、ひいてはAIの行動に複雑さ(多様さ)を感じないということ。

6-2 被験者Bでの実験

本セクションでは被験者Bをプレイヤーとしたときの実験結果とその考察について示す。

6-2-1 推定実験及びその結果

推定実験として、本章冒頭にて記した4条件下で行ってもらった各試合の平均 Attention をガウス関数で近似した。

推定実験の結果として各試合の平均 Attention, 及びそれをガウス関数で近似したグラフを図4に示す。図4の見方は図2と同じである。

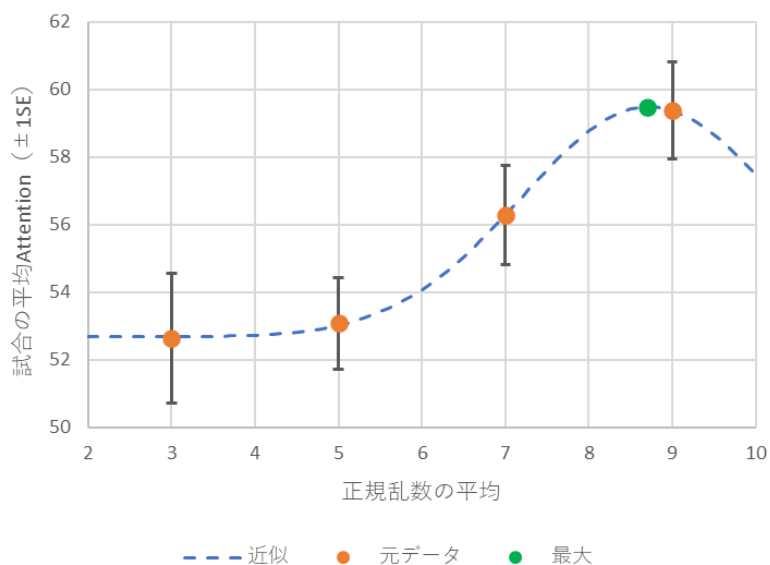


図4 推定実験結果とガウス関数近似

図4より、平均8.7標準偏差1の正規乱数がAttention最適条件となる。

6-2-2 検証実験及びその結果

推定実験より、Attention 最適条件は、平均 8.7 標準偏差 1 の正規乱数である。よって平均 8.7 標準偏差 1 を含む以下の 5 条件下で試合を行ってもらい、検証した：①平均値 5 標準偏差 1，②平均値 7 標準偏差 1，③平均値 8 標準偏差 1，④平均値 8.7 標準偏差 1，⑤平均値 10 標準偏差 1。

検証実験の結果として推定実験で近似したガウス関数、及び検証実験の各試合の平均 Attention を図 5 に示す。図 5 の見方は図 3 と同じである。

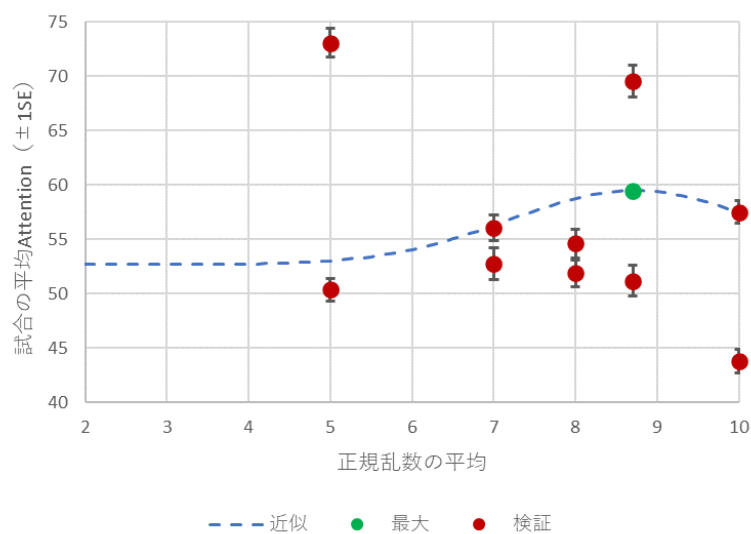


図 5 検証実験結果

図 5 より、Attention 最適条件である平均 8.7 標準偏差 1 の正規乱数条件の AI と対戦してもらったときに、平均 Attention が最大とならなかったことが確認できる。また、ガウス関数近似と各試合の平均 Attention との差の絶対値は平均 6.96 であった。すなわち、検証実験の各試合の平均 Attention が平均 6.96 だけガウス関数近似から離れた値であった。

検証実験の結果としてアンケートの回答結果を表 3 に示す。表 3 の見方は表 1 と同じである。

表 3 検証実験アンケート結果

正規乱数の平均		5	7	8	8.7	10
アンケート項目	・試合が楽しかったか？	5	5.5	4	5	4.5
	・相手が強いと感じたか？	2.5	4.5	4.5	4	4.5
	・相手の行動が予測できたか？	4	4	3.5	3.5	3.5
	・キャラクターを思い通りに動かせたか (操作ミスしなかったか)？	4.5	5	4	5	4
	・相手の行動が複雑(多様)だと感じたか？	3	4.5	4	4	4
	・ゲームに集中していたと思うか？	4.5	6	4.5	5.5	4
	・新しいコマンドに積極的に挑戦したか？	3.5	2.5	2	3.5	2.5
	・勝ちたい(負けたくない)と思ったか？	5.5	6	5.5	6	6
試合の平均Attention		61.7	54.4	53.2	60.3	50.6

表 3 より、アンケートの試合の楽しさにおいて Attention 最適条件である平均 8.7 標準偏差 1 の正規乱数条件が、5 条件のうちで最高評価とならなかったことがわかる。また、回答結果と各試合の平均 Attention の間で相関性が見られたのは 2 項目であり、「新しいコマンドに積極的に挑戦したか？」と「相手が強いと感じたか？」であった。

各試合の各ラウンド終了時のプレイヤーと AI の平均 HP 差を表 4 に示す。表 4 の見方は表 2 と同じである。

表 4 ラウンド終了時の平均 HP 差

正規乱数の平均	5	7	8	8.7	10
第 1 試合の平均	92.0	60.3	-17.0	-55.0	-25.7
第 2 試合の平均	160.0	80.0	-49.0	97.7	63.3
全体平均	126.0	70.2	-33.0	21.3	18.8

表 4 より、正規乱数の平均を高く設定すると AI が強くなり、試合が接戦になる傾向にあったことがわかる。

6-2-3 考察

図 5 より、ガウス関数近似による予測の通りにはならなかった。この理由について、以下の 3 点が考えられる：

- ① 被験者 A と同様に、ガウス関数近似と各試合での平均 Attention とが平均 6.96 離れていた。このことからガウス関数による近似が平均 Attention の変化に適していなかったことが理由の 1 つとして考えられる。
- ② 被験者 A と同様に、推定実験と検証実験が異なる日に行われたことが理由の 1 つと

して考えられる。

- ③ 被験者 A と同様に、正規乱数が同条件である 2 試合の間で平均 Attention の差の絶対値を求め、5 条件での平均を計算したところ、同条件の 2 試合間で平均 12.17 の差があった。このことから正規乱数が同条件であっても平均 Attention のバラつきが大きく、推定や検証のための試合数が足りていないことが理由の 1 つとして考えられる。

表 3 より、アンケートの結果と試合の平均 Attention の間で、相関性が見られる項目は 2 つあった。

1 つ目は「新しいコマンドに積極的に挑戦したか？」という項目であり、相関係数が 0.889 であった。この理由について、以下の点が考えられる：今まで使用してこなかった新しい技を出すためのコマンドに挑戦するということは、そのコマンド入力の動作に不慣れなため、高い集中状態を必要とすること。但しコマンド入力に成立するか否かは、プレイヤーの手元の操作精度にしか依存しないと考えられるので、こういった点から生じる高い集中状態は相手の強さの調整の良し悪しを反映しているとは言えないと考えられる。

2 つ目は「相手が強いと感じたか？」という項目であり、相関係数が -0.816 であった。この理由について、以下の点が考えられる：プレイヤーが高い集中状態で試合に取り組むとき、1 つ目の項目「新しいコマンドに積極的に挑戦したか？」についての考察より、プレイヤーは新しいコマンドに挑戦していると考えられる。よって、プレイヤーが自身のキャラクターの行動に意識を割いているので、AI の行動に対する意識が薄くなると考えられる。つまり、プレイヤーが AI の行動を気にしていないために、結果として AI が弱いと感じるということ。

図 5、表 3 及び表 4 より、検証実験において正規乱数の平均を高く設定することで、接戦となった試合はあったが、正規乱数の平均を高く設定した条件の AI との試合が楽しいとは言えない結果であったし、同様に試合の平均 Attention が高かったとも言えない結果であった。このことから被験者 B は、試合が接戦であれば楽しいとする従来研究[4]の主張には当てはまらないプレイヤーであると考えられる。また被験者 A での実験の考察で述べた、「正規乱数の平均に関わらず被験者 A が AI に圧勝していたため被験者 A の実力に対して、AI の根本的な強さが不十分であったこと」が被験者 B においては認められない。よって、被験者 B の結果については、ガウス関数近似による予測精度の低さの理由として、プレイヤーと AI との実力差を挙げるのは不適切であると考えられる。

6-3 今後の展開

本研究の今後の展開としては、推定実験及び検証実験を同じに日に行い、正規乱数の調整の精度が向上するのかを試すこと、正規乱数の平均値と標準偏差の 2 つのパラメータを同時に調整すること、正規乱数を調整するための集中度合を測る指標に新しく試合中の心拍数や瞬きの回数（瞬目回数）を追加して、調整の精度向上を目指すことである。

6-4 同日中での実験

本実験では推定実験と検証実験を同日中に行った場合に、検証実験において平均 Attention がガウス関数近似に沿うような結果が得られるかを確かめる。

被験者 A をプレイヤーとして実験を行い、事前準備や測定機器、実験内容は 6-1 と同じとした。6-1 と異なる点は推定実験と検証実験を同日中に行うこと、検証実験の正規乱数の条件設定、検証実験を各条件につき 1 試合ずつで行うことである。

6-4-1 推定実験及びその結果

推定実験として、本章冒頭にて記した 4 条件下で行ってもらった各試合の平均 Attention をガウス関数で近似した。

推定実験の結果として試合の平均 Attention、及びそれをガウス関数で近似したグラフを図 6 に示す。図 6 の見方は図 2 と同じである。

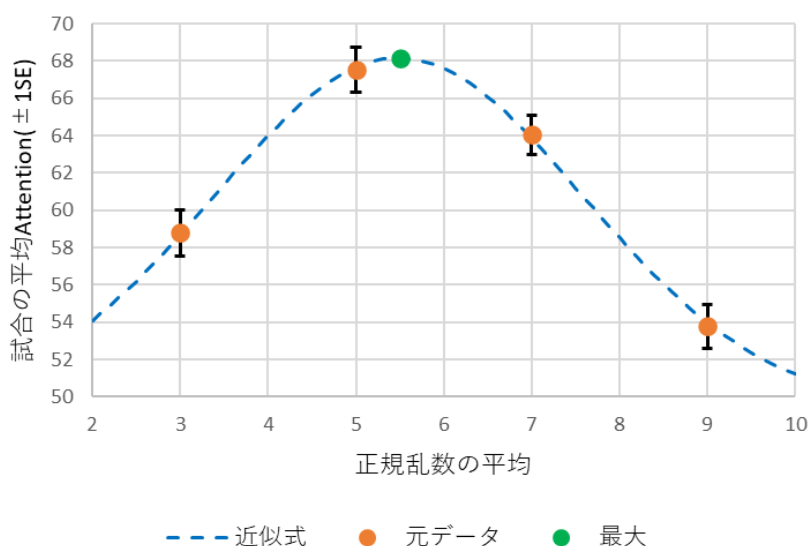


図 6 推定実験結果とガウス関数近似

図 6 より、平均 5.5 標準偏差 1 の正規乱数が Attention 最適条件となる。

6-4-2 検証実験及びその結果

上記の通り、推定実験と同じ日に検証実験を行った。推定実験より、Attention 最適条件は、平均 5.5 標準偏差 1 の正規乱数である。よって平均 5.5 標準偏差 1 を含む以下の 5 条件下で試合を行ってもらい、検証した：①平均値 2 標準偏差 1、②平均値 4 標準偏差 1、③平均値 5.5 標準偏差 1、④平均値 7 標準偏差 1、⑤平均値 9 標準偏差 1。

但し、本検証実験では時間的都合により各条件につき 1 試合ずつ行ってもらった。

検証実験の結果として推定実験で近似したガウス関数、及び検証実験の各試合の平均

Attention を図 7 に示す. 図 7 の見方は図 3 と同じである.

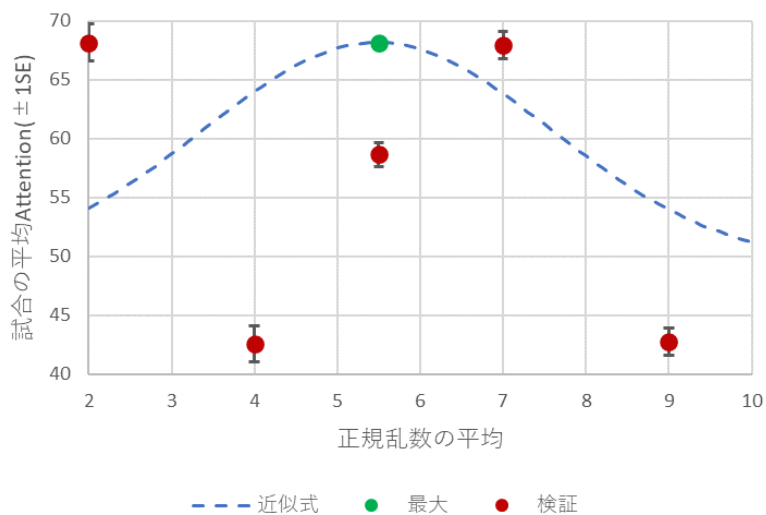


図 7 検証実験結果

図 7 より, Attention 最適条件である平均 5.5 標準偏差 1 の正規乱数条件の AI と対戦してもらったときに, 平均 Attention が最大とならなかったことが確認できる. また, ガウス関数近似と各試合の平均 Attention との差の絶対値は平均 12.08 であった. すなわち, 検証実験の各試合の平均 Attention が平均 12.08 だけガウス関数近似から離れた値であった. また, 推定実験及び検証実験を同日中に行っても, 各試合の平均 Attention がガウス関数近似に沿うような検証結果が得られなかった.

検証実験の結果としてアンケートの回答結果を表 5 に示す. 赤字での表記は Attention 最適条件での回答結果と平均 Attention を示す. 回答結果及び平均 Attention は各条件での 1 試合の結果である.

表 5 検証実験アンケート結果

正規乱数の平均		2	4	5.5	7	9
アンケート項目	・試合が楽しかったか？	1	5	5	6	3
	・相手が強いと感じたか？	1	4	5	5	3
	・相手の行動が予測できたか？	5	4	5	3	6
	・キャラクターを思い通りに動かせたか (操作ミスしなかったか)？	6	5	5	6	6
	・相手の行動が複雑(多様)だと感じたか？	1	4	4	5	3
	・ゲームに集中していたと思うか？	2	5	5	6	3
	・新しいコマンドに積極的に挑戦したか？	1	2	2	2	2
	・勝ちたい(負けたくない)と思ったか？	1	3	5	6	2
試合の平均Attention		68.2	42.6	58.6	68.0	42.7

表 5 より、アンケートの試合の楽しさについて、Attention 最適条件である平均 5.5 標準偏差 1 の正規乱数条件が、5 条件のうちで最高評価とならなかったことがわかる。

各試合の各ラウンド終了時のプレイヤーと AI の平均 HP 差を表 6 に示す。表 6 の見方は表 2 と同じである。

表 6 ラウンド終了時の平均 HP 差

正規乱数の平均	2	4	5.5	7	9
第 1 試合の平均	239.3	232.3	168.7	158.7	168.3

表 6 より、キャラクターの初期 HP が 300 に設定されているので、平均的に初期 HP の半分以上をプレイヤー側がより多く残し AI に圧勝する傾向にあったことがわかる。

6-4-3 考察

図 7 より、ガウス関数近似による予測の通りにはならなかった。この理由について、以下の 2 点が考えられる：

- ① 6-1-3 と同様に、ガウス関数近似と各試合での平均 Attention とが平均 12.08 と広く離れていた。このことからガウス関数による近似が平均 Attention の変化に適していなかったことが理由の 1 つとして考えられる。
- ② 表 6 より 6-1-3 と同様に、正規乱数の平均に関わらず被験者 A が AI に圧勝する傾向にあった。このことから被験者の実力に対して、AI の根本的な強さが不十分であったことが理由の 1 つとして考えられる。

図 3、図 5 及び図 7 より、推定実験と検証実験を同じ日に行っても、検証実験の試合の平均 Attention がガウス関数近似に特段近づくことはなかった。この理由について、以下の点

が考えられる：脳波から算出される Attention のバラつきは大きく、これは生体信号である脳波を利用することに起因する Attention の再現性の低さや、本研究で使用している簡易脳波計（ドライタイプの 2 つの電極からなる単極誘導を用いた測定機器）の正確性の低さに原因があるということ。

6-5 今後の展開

6-4 検証実験の結果より、推定実験と検証実験を同じ日に行っても、特段良い結果が得られるわけではないと考えられる。よって今後は被験者の負担を分散させるため、及び被験者の集中力が長時間保てず、条件に不公平性が生まれてしまうのを防ぐために、推定実験と検証実験を別日に行うこととする。

第 7 章 脳波を用いた難易度調整 (2 パラメータ①)

これまでの実験では正規乱数の平均のみの推定であったが、本章では正規乱数の平均と標準偏差の 2 パラメータを同時に推定し、難易度調整を行う実験について示す。被験者 1 名 (A) に実験を行った。なお、被験者 A は筆者である。

本章の推定実験では、以下の 9 つの正規乱数の条件下で行われた各試合の平均 Attention をガウス関数近似して、Attention 最適条件を推定する：①平均値 2 標準偏差 0.5, ②平均値 2 標準偏差 1, ③平均値 2 標準偏差 2, ④平均値 5 標準偏差 0.5, ⑤平均値 5 標準偏差 1, ⑥平均値 5 標準偏差 2, ⑦平均値 8 標準偏差 0.5, ⑧平均値 8 標準偏差 1, ⑨平均値 8 標準偏差 2。

実験当日に操作練習を 3 分間行ってもらった。その後、上記の各条件の AI とランダムな順番で 1 試合ずつ行ってもらい、1 試合ごとに休憩を 3 分取ってもらった。9 条件下で行ってもらった各試合の平均 Attention をガウス関数で近似し、Attention 最適条件を推定した。ガウス関数近似の詳細は付録のセクション E に示す。

本章の検証実験では、Attention 最適条件を含む 9 条件で 1 試合ずつ行ってもらい、Attention 最適条件の AI と対戦してもらったときに平均 Attention が最大になるか、加えて楽しい試合を体験させられるかを確かめた。

操作練習を 3 分間行ってもらった後、9 条件の AI とランダムな順番で 1 試合ずつ行ってもらった。1 試合ごとにアンケートに回答してもらい、その後に休憩を取ってもらった。アンケートと休憩を合わせて 3 分とした。

第 6 章と同様にアンケートの内容は 8 項目で、1~6 の 6 段階で 6 を最高評価として回答してもらった。

7-1 被験者 A での実験

本セクションでは被験者 A をプレイヤーとしたときの実験結果とその考察について示す。

7-1-1 推定実験及びその結果

推定実験として上記の 9 条件下で行ってもらった各試合の平均 Attention をガウス関数で近似した。

推定実験の結果として各試合の平均 Attention を図 8 及び表 7 に示す。図 8 において、横軸は正規乱数の標準偏差, 奥行きは正規乱数の平均であり, 各頂点は各試合の平均 Attention を示す。表 7 において縦軸は正規乱数の平均であり, 横軸は正規乱数の標準偏差である。太字での表記は各試合の平均 Attention を示し, 赤字での表記は測定された平均 Attention の最大値を示す。

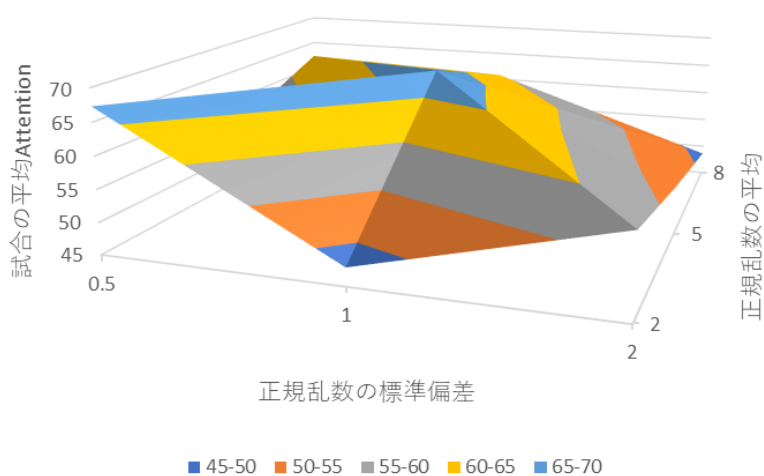


図 8 推定実験結果のグラフ

表 7 推定実験結果の表

試合の平均Attention		正規乱数の標準偏差		
		0.5	1	2
正規乱数の平均	2	67.29	47.78	57.43
	5	54.06	68.28	52.69
	8	62.28	60.81	48.61

推定実験の結果として, 平均 Attention をガウス関数で近似したグラフを図 9 に示す。横軸は正規乱数の標準偏差, 奥行きは正規乱数の平均であり, 縦軸は平均 Attention を示す。

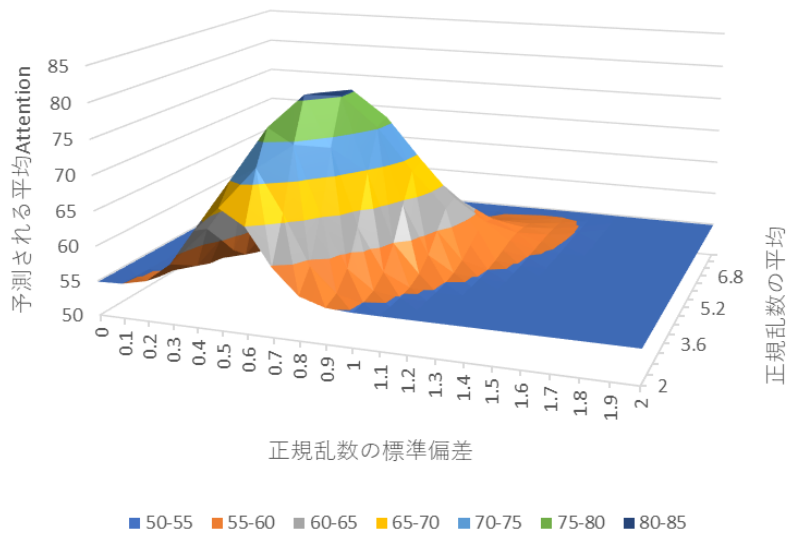


図9 ガウス関数近似のグラフ

図9より、Attention 最適条件は平均 3.6 標準偏差 0.74 である。

7-1-2 検証実験及びその結果

推定実験より、Attention 最適条件は、平均 3.6 標準偏差 0.74 である。よって平均 3.6 標準偏差 0.74 を含む以下の 9 条件下で試合を行ってもらい、検証した：①平均 2 標準偏差 0.5, ②平均 2 標準偏差 0.74, ③平均 2 標準偏差 1, ④平均 3.6 標準偏差 0.5, ⑤平均 3.6 標準偏差 0.74, ⑥平均 3.6 標準偏差 1, ⑦平均 5 標準偏差 0.5, ⑧平均 5 標準偏差 0.74, ⑨平均 5 標準偏差 1。

検証実験の結果として各試合の平均 Attention を図 10 及び表 8 に示す。図 10 の見方は図 8 と同じである。表 8 において縦軸は正規乱数の平均であり、横軸は正規乱数の標準偏差である。太字での表記は各試合の平均 Attention を示し、赤字での表記は Attention 最適条件で測定された平均 Attention を示す。

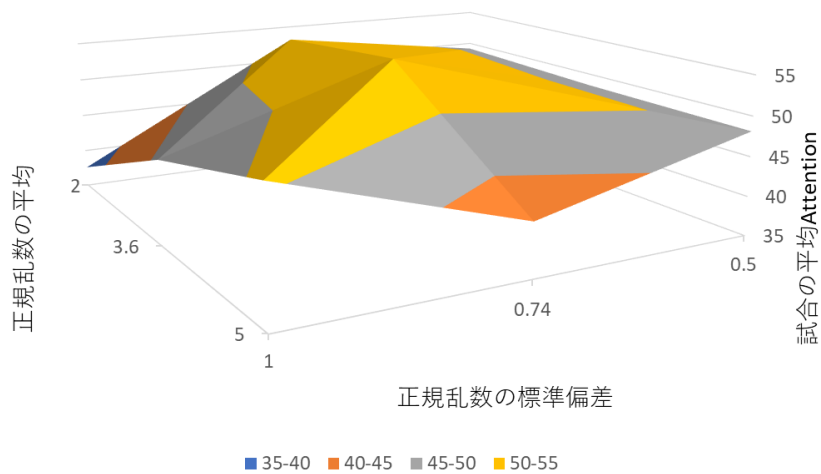


図 10 検証実験結果のグラフ

表 8 検証実験結果の表

試合の平均Attention		正規乱数の標準偏差		
		0.5	0.74	1
正規乱数の平均	2	49.15	53.10	37.59
	3.6	48.42	54.92	45.68
	5	48.11	41.69	50.67

図 10 及び表 8 より、ガウス関数近似による予測の通り、Attention 最適条件である平均 3.6 標準偏差 0.74 の正規乱数条件が平均 Attention において最大となったことがわかる。図 9 におけるガウス関数近似と図 10 における検証実験の各試合の平均 Attention との差の絶対値は平均 14.95 であった。すなわち、検証実験の各試合の平均 Attention が平均 14.95 だけガウス関数近似から離れた値であった。

検証実験の結果としてアンケートの回答結果を表 9 に示す。赤字での表記は Attention 最適条件での回答結果と平均 Attention を示す。回答結果及び平均 Attention は各条件での 1 試合の結果である。

表9 検証実験アンケート結果

正規乱数の平均, 標準偏差		2, 0.5	2, 0.74	2, 1	3.6, 0.5	3.6, 0.74	3.6, 1	5, 0.5	5, 0.74	5, 1
アンケート項目	・試合が楽しかったか?	1	1	1	2	4	5	4	5	4
	・相手が強いと感じたか?	1	1	1	2	2	3	4	5	3
	・相手の行動が予測できたか?	5	5	5	5	3	3	4	3	4
	・キャラクターを思い通りに動かせたか (操作ミスしなかったか)?	6	5	5	6	5	5	5	6	6
	・相手の行動が複雑(多様)だと感じたか?	1	1	2	2	4	5	4	5	4
	・ゲームに集中していたと思うか?	1	1	1	2	3	4	4	5	4
	・新しいコマンドに積極的に挑戦したか?	1	2	1	2	3	3	1	1	2
	・勝ちたい(負けたくない)と思ったか?	1	1	1	1	1	2	3	4	1
試合の平均Attention		49.2	53.1	37.6	48.4	54.9	45.7	48.1	41.7	50.7

表9より、アンケートの試合の楽しさについて Attention 最適条件である平均 3.6 標準偏差 0.74 の正規乱数条件が、9 条件のうちで最高評価とならなかったことがわかる。また、アンケートの結果と試合の平均 Attention の間には、特に相関性が見られる項目はなかった。

各試合の各ラウンド終了時のプレイヤーと AI の平均 HP 差を表 10 に示す。各数値が正のときは、平均的にラウンド終了時にプレイヤー側が HP をより多く残していたことを表す。

表 10 ラウンド終了時の平均 HP 差

平均 HP 差		正規乱数の標準偏差		
		0.5	0.74	1
正規乱数の平均	2	233.3	217.7	248.7
	3.6	212.7	204.7	178.3
	5	189.3	215.3	194.3

表 10 より、キャラクターの初期 HP が 300 に設定されているので、平均的に初期 HP の半分以上をプレイヤー側がより多く残し、AI に圧勝する傾向にあったことがわかる。

7-1-3 考察

図 10 及び表 8 より、ガウス関数近似による予測の通り、Attention 最適条件である平均 3.6 標準偏差 0.74 の正規乱数条件が平均 Attention において最大値を記録した。この理由として、以下の点が考えられる：AI の強さを変える上で今までの実験では固定値としていた正規乱数の標準偏差が、単に行動のランダム性を左右するだけではない重要な項目であり、それが今回の実験ではパラメータとして調整されたことが理由として考えられる。

第8章 AIの仕様変更

今までの実験で使用していたAIの仕様は先述の通りAIの行動の選択肢が地上において25種類、空中において15種類であったが、この中に「何もしない」という選択肢が無い仕様であった。そのために正規乱数の平均を変化させてもAIは攻撃、移動、防御のうち何かしらの行動を常に選択していた。これによって弱いつもりで設定した条件のAIでも試合においてはある程度の強さを持つようになっていた。また強いつもりで設定した条件のAIでの試合においては、AIは行動し続けるためプレイヤー側は待っていれば、そのうちAIは隙をさらすようになっていた。

真に弱い相手というのは何もしない相手、一切行動しない相手であると考えられる。また強い相手というのは行動に緩急があると考えられる。以上より今までの実験で使用していたAIの選択肢に「何もしない」という選択肢を追加する。追加する選択肢は以下の3つである：①STAND（地上行動，立ったまま何もしない），②CROUCH（地上行動，しゃがんだまま何もしない），③AIR（空中行動，空中で何もしない）。

上記の3つの選択肢を加えたことにより、AIの取りうる行動は空中行動で16種類、地上行動で27種類となった。またAIの持つゲーム状況と行動の組み合わせの評価値が計 $336 \times (16 + 27) = 14,448$ 個になった。AIが識別するゲーム状況が336種類であることと15フレーム（0.25秒）前のものであることに変わりはない。また評価値が0~10の範囲であることにも変わりはない。但し、評価値は自己対戦学習により1から学習し直したものをを用いた。仕様変更後のAIの詳細は付録のセクションC-2に示す。

第9章 脳波を用いた難易度調整（2パラメータ②）

本章では仕様変更をしたAIを用いて実験を行う。第7章と同様に正規乱数の平均と標準偏差の2パラメータを同時に推定し、難易度調整を行う実験について示す。被験者3名（A, C, D）に実験を行った。なお、被験者Aは筆者である。

本章の推定実験では、操作練習を3分間行ってもらった後、第7章と同様に以下の9つの正規乱数の条件のAIとランダムな順番で1試合ずつ行ってもらう：①平均値2標準偏差0.5，②平均値2標準偏差1，③平均値2標準偏差2，④平均値5標準偏差0.5，⑤平均値5標準偏差1，⑥平均値5標準偏差2，⑦平均値8標準偏差0.5，⑧平均値8標準偏差1，⑨平均値8標準偏差2。

各試合の平均Attentionをガウス関数近似して、Attention最適条件を推定する。なお試合間の休憩時間は3分である。ガウス関数近似の詳細は付録のセクションEに示す。

本章の検証実験では、操作練習を3分間行ってもらった後、Attention最適条件を含む9条件のAIと1試合ずつ行ってもらい、Attention最適条件のAIと対戦してもらったときに平均Attentionが最大になるか、加えて楽しい試合を体験させられたかを確認した。なお検証実験の9条件は個々のプレイヤーによって異なる。1試合ごとにアンケートに回答してもらい、その後に休憩を取ってもらった。アンケートと休憩を合わせて3分とした。第6章と

同様にアンケートの内容は 8 項目で、1~6 の 6 段階で 6 を最高評価として回答してもらった。

9-1 被験者 A での実験

本セクションでは被験者 A をプレイヤーとしたときの実験結果とその考察について示す。

9-1-1 推定実験及びその結果

推定実験として、本章冒頭に記した 9 条件下で行ってもらった各試合の平均 Attention をガウス関数で近似した。

推定実験の結果として各試合の平均 Attention を図 11 及び表 11 に示す。図 11 において、横軸は正規乱数の標準偏差、奥行きは正規乱数の平均であり、各頂点は各試合の平均 Attention を示す。表 11 において縦軸は正規乱数の平均であり、横軸は正規乱数の標準偏差である。太字での表記は各試合の平均 Attention を示し、赤字での表記は測定された平均 Attention の最大値を示す。

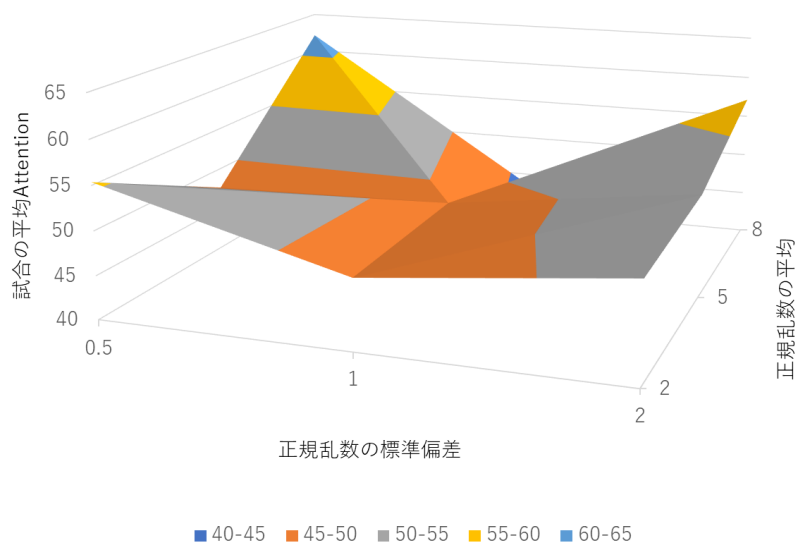


図 11 推定実験結果のグラフ

表 11 推定実験結果の表

試合の平均 Attention		正規乱数の標準偏差		
		0.5	1	2
正規乱数 の 平均	2	55.35	47.96	51.11
	5	47.60	48.34	52.05
	8	62.13	44.23	57.13

推定実験の結果をガウス関数で近似したグラフを図 12 に示す。横軸は正規乱数の標準偏差，奥行きは正規乱数の平均であり，縦軸は平均 Attention を示す。

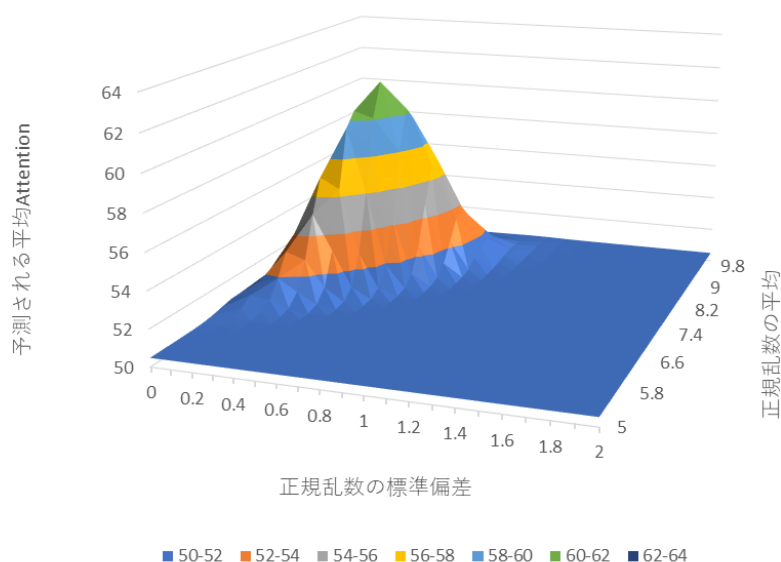


図 12 ガウス関数近似のグラフ

図 12 より，ガウス関数近似から Attention 最適条件は平均 8.0 標準偏差 0.50 である。

9-1-2 検証実験及びその結果

推定実験より，Attention 最適条件は，平均 8.0 標準偏差 0.50 である。よって平均 8.0 標準偏差 0.50 を含む以下の 9 条件下で試合を行ってもらい，検証した：①平均 5 標準偏差 0.1，②平均 5 標準偏差 0.5，③平均 5 標準偏差 1，④平均 8 標準偏差 0.1，⑤平均 8 標準偏差 0.5，⑥平均 8 標準偏差 1，⑦平均 10 標準偏差 0.1，⑧平均 10 標準偏差 0.5，⑨平均 10 標準偏差 1。

検証実験の結果として各試合の平均 Attention を図 13 及び表 12 に示す。図 13 の見方は図 11 と同じである。表 12 において縦軸は正規乱数の平均であり、横軸は正規乱数の標準偏差である。太字での表記は各試合の平均 Attention を示し、赤字の表記は Attention 最適条件の結果を示し、緑字の表記は実際に測定された平均 Attention の内の最大値を示す。

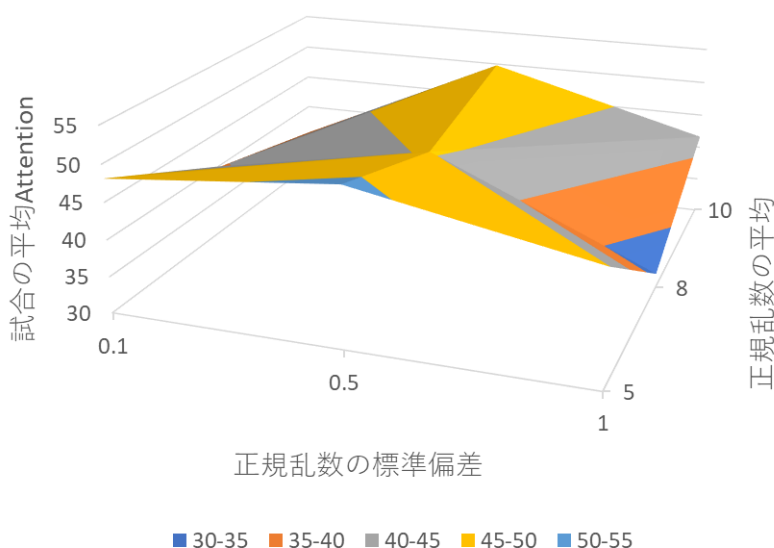


図 13 検証実験結果のグラフ

表 12 検証実験結果の表

試合の平均Attention		正規乱数の標準偏差		
		0.1	0.5	1
正規乱数の平均	5	48.11	51.15	45.35
	8	39.73	45.46	31.97
	10	35.63	49.79	41.64

図 13 及び表 12 より、ガウス関数近似による予測の通りにはならず、Attention 最適条件である平均 8.0 標準偏差 0.50 の正規乱数条件が平均 Attention において最大とならなかったことがわかる。図 12 におけるガウス関数近似と図 13 における検証実験の各試合の平均 Attention との差の絶対値は平均 8.72 であった。すなわち、検証実験の各試合の平均 Attention が平均 8.72 だけガウス関数近似から離れた値であった。

検証実験の結果としてアンケートの回答結果を表 13 に示す。赤字での表記は Attention

最適条件での回答結果と平均 Attention を示す。緑字の表記は実際に測定された平均 Attention の内の最大値を示す。回答結果及び平均 Attention は各条件での 1 試合の結果である。

表 13 検証実験のアンケート結果

正規乱数の平均, 標準偏差		5, 0.1	5, 0.5	5, 1	8, 0.1	8, 0.5	8, 1	10, 0.1	10, 0.5	10, 1
アンケート項目	・試合が楽しかったか?	5	4	5	3	4	5	2	3	4
	・相手が強いと感じたか?	2	3	2	4	5	5	3	4	5
	・相手の行動が予測できたか?	3	3	2	5	4	3	6	5	5
	・キャラクターを思い通りに動かせたか (操作ミスしなかったか)?	5	6	6	5	6	5	6	6	6
	・相手の行動が複雑(多様)だと感じたか?	5	4	6	3	3	5	2	3	3
	・ゲームに集中していたと思うか?	4	4	5	4	5	5	3	3	4
	・新しいコマンドに積極的に挑戦したか?	2	2	1	1	1	1	1	3	1
	・勝ちたい(負けたくない)と思ったか?	2	3	1	2	4	5	2	3	3
試合の平均Attention		48.1	51.2	45.4	39.7	45.5	32.0	35.6	49.8	41.6

表 13 より、アンケートの試合の楽しさについて、Attention 最適条件である平均 8.0 標準偏差 0.50 の正規乱数条件が 9 条件のうちで最高評価とならなかった。また、アンケートの結果と試合の平均 Attention の間には、特に相関性が見られる項目はなかった。

各試合の各ラウンド終了時のプレイヤーと AI の平均 HP 差を表 14 に示す。各数値が正のときは、平均的にラウンド終了時にプレイヤー側が HP をより多く残していたことを表す。

表 14 ラウンド終了時の平均 HP 差

平均 HP 差		正規乱数の標準偏差		
		0.1	0.5	1
正規乱数の平均	5	168.0	174.0	220.3
	8	180.0	173.3	122.0
	10	214.7	159.7	171.3

表 14 より、キャラクターの初期 HP が 300 に設定されているので、平均的に初期 HP の半分程度をプレイヤー側がより多く残し、AI に圧勝する傾向にあったことがわかる。

9-1-3 考察

図 13 及び表 12 より、Attention 最適条件である平均 8.0 標準偏差 0.50 の正規乱数条件が平均 Attention において最大値にならず、ガウス関数近似による予測の通りにはならな

った。この理由として、以下の点が考えられる：

- ① 図 13 では中央が高くなり、平均 5 標準偏差 0.1 から平均 10 標準偏差 1 に尾根が伸びるような山型の結果の等高線グラフを想定していた。しかし、そのような結果のグラフにはならず、ガウス関数近似と各試合での平均 Attention とが平均 8.72 離れていた。このことからガウス関数による近似が平均 Attention の変化に適していなかったことが理由の 1 つとして考えられる。
- ② 表 14 では正規乱数の平均を大きい値に変えたときに AI が強くなり、正規乱数の平均と HP 差の全体平均が負の相関関係になることを期待していた。しかし、そのような結果にはならず、正規乱数の平均に関わらず被験者 A が AI に圧勝していた。このことから改良を加えたが依然として被験者の実力に対して、AI の根本的な強さが不十分であったことが理由の 1 つとして考えられる。

9-2 被験者 C での実験

本セクションでは被験者 C をプレイヤーとしたときの実験結果とその考察について示す。

9-2-1 推定実験及びその結果

推定実験として、本章冒頭に記した 9 条件下で行ってもらった各試合の平均 Attention をガウス関数で近似した。

推定実験の結果として各試合の平均 Attention を図 14 及び表 15 に示す。図 14 の見方は図 11 と同じである。表 15 の見方は表 11 と同じである。

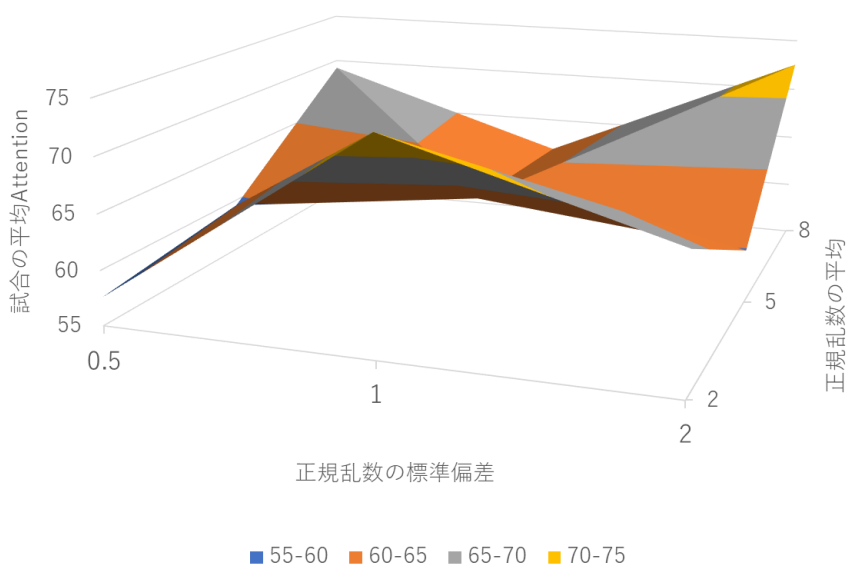


図 14 推定実験結果のグラフ

表 15 推定実験結果の表

試合の平均 Attention		正規乱数の標準偏差		
		0.5	1	2
正規乱数の 平均	2	57.56	73.99	67.01
	5	59.48	62.43	59.87
	8	69.18	61.79	72.54

推定実験の結果をガウス関数で近似したグラフを図 15 に示す。図 15 の見方は図 12 と同じである。

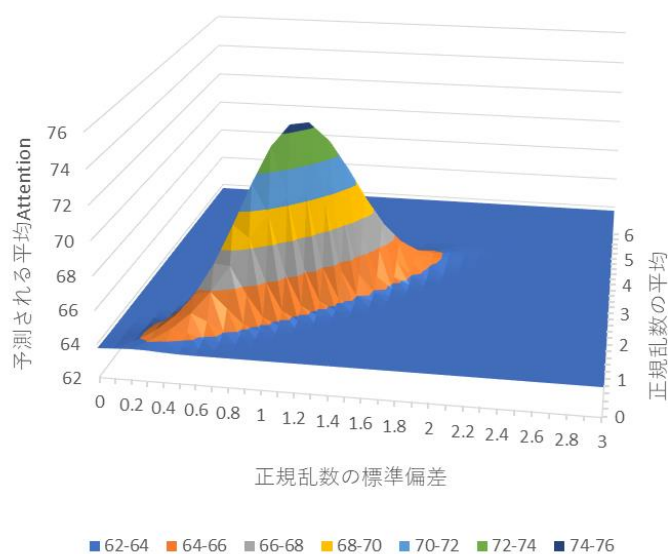


図 15 ガウス関数近似のグラフ

図 15 より、ガウス関数近似から Attention 最適条件は平均 2.0 標準偏差 1.04 である。

9-2-2 検証実験及びその結果

推定実験より、Attention 最適条件は、平均 2.0 標準偏差 1.04 である。よって平均 2.0 標準偏差 1.04 を含む以下の 9 条件下で試合を行ってもらい、検証した：①平均 0 標準偏差 0.5, ②平均 0 標準偏差 1.04, ③平均 0 標準偏差 1.5, ④平均 2 標準偏差 0.5, ⑤平均 2 標準偏差 1.04, ⑥平均 2 標準偏差 1.5, ⑦平均 4 標準偏差 0.5, ⑧平均 4 標準偏差 1.04, ⑨平均

4 標準偏差 1.5.

検証実験の結果として各試合の平均 Attention を図 16 及び表 16 に示す。図 16 の見方は図 11 と同じである。表 16 の見方は表 12 と同じである。

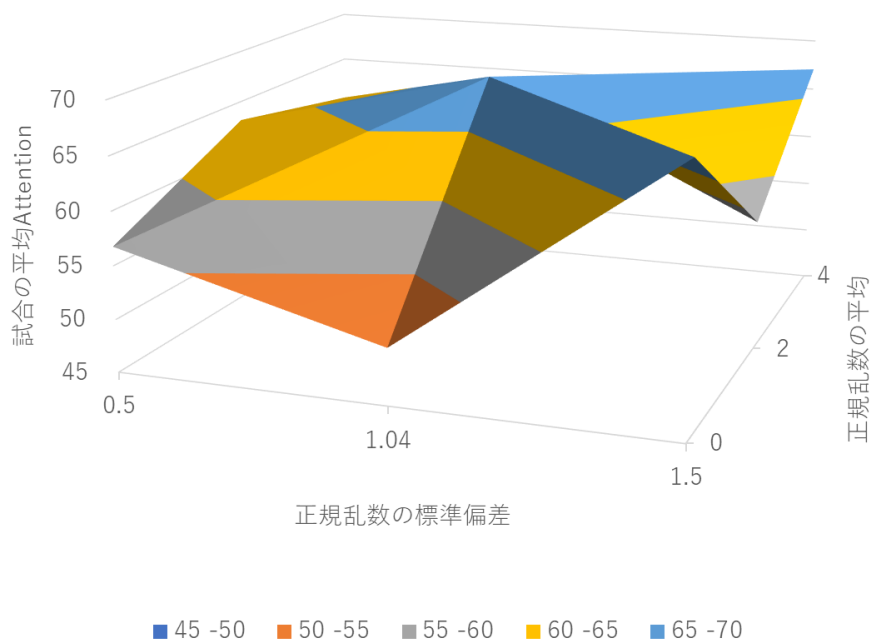


図 16 検証実験結果のグラフ

表 16 検証実験結果の表

試合の平均Attention		正規乱数の標準偏差		
		0.5	1.04	1.5
正規乱数の平均	0	56.75	50.26	68.93
	2	63.14	69.17	57.25
	4	60.64	47.55	67.06

図 16 及び表 16 より、Attention 最適条件である平均 2.0 標準偏差 1.04 の正規乱数条件が試合の平均 Attention において最大となり、ガウス関数近似による予測の通りになったことがわかる。図 15 におけるガウス関数近似と図 16 における検証実験の各試合の平均 Attention との差の絶対値は平均 6.79 であった。すなわち、検証実験の各試合の平均 Attention が平均 6.79 だけガウス関数近似から離れた値であった。

検証実験の結果としてアンケートの回答結果を表 17 に示す。表 17 の見方は表 13 と同じである。

表 17 検証実験のアンケート結果

正規乱数の平均, 標準偏差		0, 0.5	0, 1.04	0, 1.5	2, 0.5	2, 1.04	2, 1.5	4, 0.5	4, 1.04	4, 1.5
アンケート項目	・試合が楽しかったか?	5	5	5	6	4	6	5	5	5
	・相手が強いと感じたか?	4	5	4	4	3	5	4	6	4
	・相手の行動が予測できたか?	5	6	4	5	4	5	5	5	5
	・キャラクターを思い通りに動かせたか (操作ミスしなかったか)?	6	5	5	5	5	5	5	5	5
	・相手の行動が複雑 (多様) だと感じたか?	5	5	5	4	3	5	4	6	5
	・ゲームに集中していたと思うか?	5	5	5	5	4	6	5	5	5
	・新しいコマンドに積極的に挑戦したか?	6	6	5	6	5	6	6	5	5
	・勝ちたい (負けたくない) と思ったか?	4	4	4	5	4	5	4	5	5
試合の平均Attention		56.8	50.3	68.9	63.1	69.2	57.3	60.6	47.6	67.1

表 17 より、アンケートの試合の楽しさにおいて Attention 最適条件である平均 2.0 標準偏差 1.04 の正規乱数条件が 9 条件のうちで最高評価とならなかったことがわかる。また、アンケートの回答結果と各試合の平均 Attention の間で、特に相関性が見られる項目は 2 つあり、「相手が強いと感じたか?」と「相手の行動が予測できたか?」であった。

各試合の各ラウンド終了時のプレイヤーと AI の平均 HP 差を表 18 に示す。表 18 の見方は表 14 と同じである。

表 18 ラウンド終了時の平均 HP 差

平均 HP 差		正規乱数の標準偏差		
		0.5	1.04	1.5
正規乱数の平均	0	260.0	250.0	238.3
	2	220.0	150.0	104.3
	4	176.7	96.0	163.3

表 18 より、キャラクターの初期 HP が 300 に設定されているので、平均的に HP をプレイヤー側がより多く残し、AI に勝つ傾向にあったことがわかる。また正規乱数の平均と標準偏差を高く設定したときに、HP 差が小さくなる傾向にあったことがわかる。

9-2-3 考察

図 16 及び表 16 より、Attention 最適条件である平均 2.0 標準偏差 1.04 の正規乱数条件が試合の平均 Attention において最大となり、ガウス関数近似による予測の通りになった。

この理由として、以下の点が考えられる：被験者 C は格闘ゲームの経験が少ない初心者であったことから、本実験で用いている AI が実力的に十分対応できたということ。

表 17 より、アンケートの回答結果と各試合の平均 Attention の間で、相関性が見られる項目は 2 つあった。

1 つ目は「相手が強いと感じたか？」という項目であり、相関係数が -0.862 であった。この理由について、以下の点が考えられる：プレイヤーが高い集中状態にあると強くなり、相手である AI が弱く感じたということ。

2 つ目は「相手の行動が予測できたか？」という項目であり、相関係数が -0.736 であった。この理由について、以下の点が考えられる：AI の行動のランダム性が高くなり予測ができなくなると、プレイヤーは AI の行動を注意深く観察し対応しようとするために高い集中状態になると考えられる。

9-3 被験者 D での実験

本セクションでは被験者 D をプレイヤーとしたときの実験結果とその考察について示す。

9-3-1 推定実験及びその結果

推定実験として、本章冒頭に記した 9 条件下で行ってもらった各試合の平均 Attention をガウス関数で近似した。

推定実験の結果として各試合の平均 Attention を図 17 及び表 19 に示す。図 17 の見方は図 11 と同じである。表 19 の見方は表 11 と同じである。

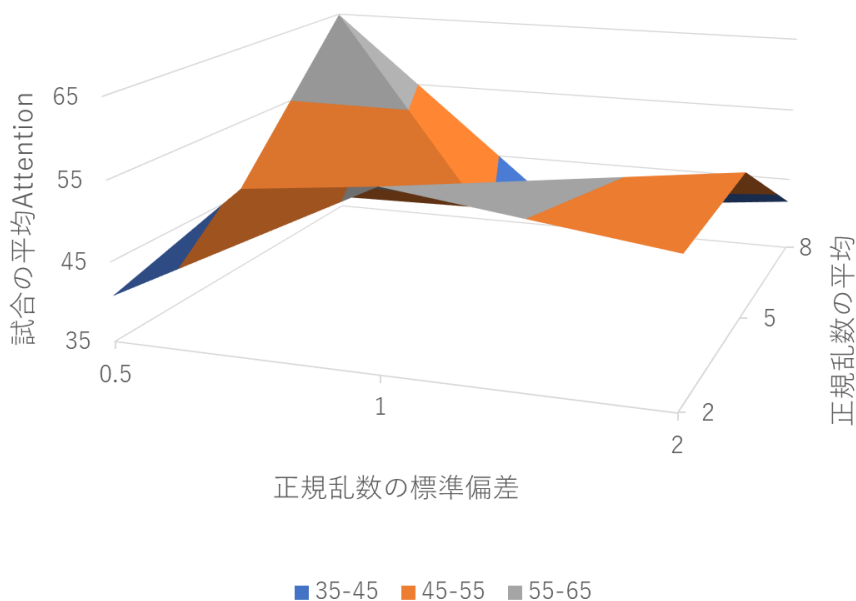


図 17 推定実験結果のグラフ

表 19 推定実験結果の表

試合の平均 Attention		正規乱数の標準偏差		
		0.5	1	2
正規乱数 の 平均	2	40.71	57.15	52.82
	5	45.82	46.29	53.83
	8	64.93	38.19	41.75

推定実験の結果をガウス関数で近似したグラフを図 18 に示す。図 18 の見方は図 12 と同じである。

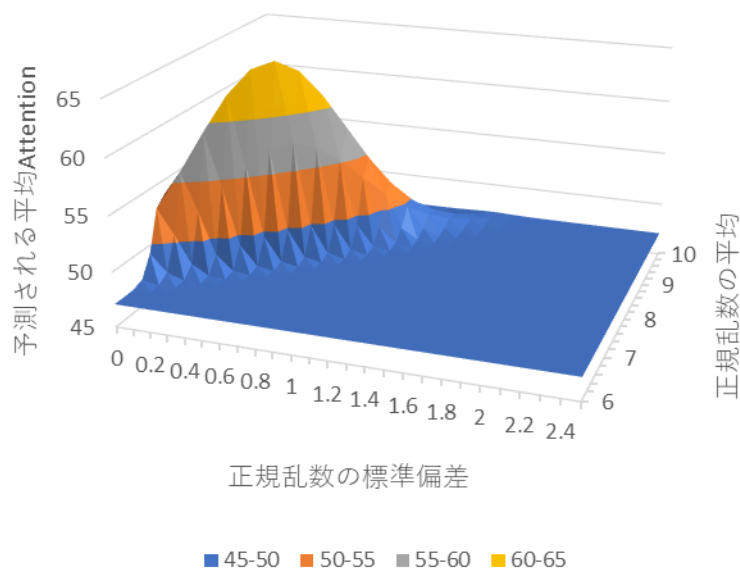


図 18 ガウス関数近似のグラフ

図 18 より、ガウス関数近似から Attention 最適条件は平均 8.0 標準偏差 0.46 である。

9-3-2 検証実験及びその結果

推定実験より、Attention 最適条件は、平均 8.0 標準偏差 0.46 である。よって平均 8.0 標準偏差 0.46 を含む以下の 9 条件下で試合を行ってもらい、検証した：①平均 6 標準偏差 0.1, ②平均 6 標準偏差 0.46, ③平均 6 標準偏差 1, ④平均 8 標準偏差 0.1, ⑤平均 8 標準偏差 0.46, ⑥平均 8 標準偏差 1, ⑦平均 10 標準偏差 0.1, ⑧平均 10 標準偏差 0.46, ⑨平均

10 標準偏差 1.

検証実験の結果として各試合の平均 Attention を図 19 及び表 20 に示す. 図 19 の見方は図 11 と同じである. 表 20 の見方は表 12 と同じである.

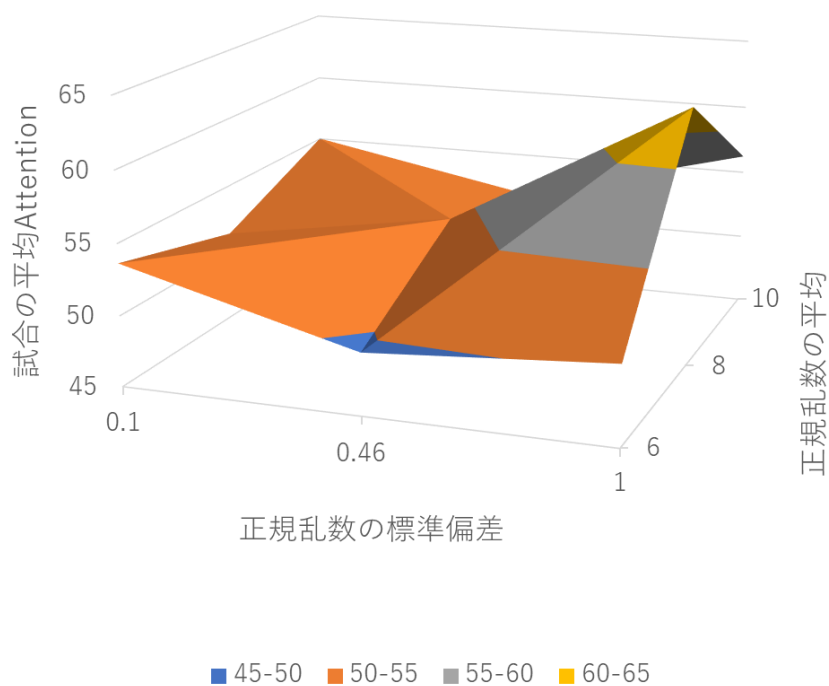


図 19 検証実験結果のグラフ

表 20 検証実験結果の表

試合の平均Attention		正規乱数の標準偏差		
		0.1	0.46	1
正規乱数の平均	6	53.63	49.33	50.53
	8	51.38	54.03	63.32
	10	55.02	51.90	56.23

図 19 及び表 20 より, Attention 最適条件である平均 8.0 標準偏差 0.46 の正規乱数条件が試合の平均 Attention において最大とならず, ガウス関数近似による予測の通りにならなかったことがわかる. 図 18 におけるガウス関数近似と図 19 における検証実験の各試合の

平均 Attention との差の絶対値は平均 7.27 であった。すなわち、検証実験の各試合の平均 Attention が平均 7.27 だけガウス関数近似から離れた値であった。

検証実験の結果としてアンケートの回答結果を表 21 に示す。表 21 の見方は表 13 と同じである。

表 21 検証実験のアンケート結果

正規乱数の平均, 標準偏差		6, 0.1	6, 0.46	6, 1	8, 0.1	8, 0.46	8, 1	10, 0.1	10, 0.46	10, 1
アンケート項目	・試合が楽しかったか?	4	6	5	4	4	5	4	5	5
	・相手が強いと感じたか?	3	6	2	4	3	5	3	5	3
	・相手の行動が予測できたか?	3	3	3	5	4	3	4	4	5
	・キャラクターを思い通りに動かせたか (操作ミスしなかったか)?	4	5	5	4	4	5	3	4	3
	・相手の行動が複雑 (多様) だと感じたか?	4	6	2	3	3	5	3	3	2
	・ゲームに集中していたと思うか?	3	6	5	5	4	5	5	6	5
	・新しいコマンドに積極的に挑戦したか?	5	3	5	3	5	2	5	4	6
	・勝ちたい (負けたくない) と思ったか?	4	6	5	4	4	5	4	5	5
試合の平均Attention		53.6	49.3	50.5	51.4	54.0	63.3	55.0	51.9	56.2

表 21 より、アンケートの試合の楽しさにおいて Attention 最適条件である平均 8.0 標準偏差 0.46 の正規乱数条件が 9 条件のうちで最高評価とならなかったことがわかる。また、アンケートの回答結果と各試合の平均 Attention の間で、特に相関性が見られる項目はなかった。

各試合の各ラウンド終了時のプレイヤーと AI の平均 HP 差を表 22 に示す。表 22 の見方は表 14 と同じである。

表 22 ラウンド終了時の平均 HP 差

平均 HP 差		正規乱数の標準偏差		
		0.1	0.46	1
正規乱数の平均	6	18.3	49.7	105.0
	8	50.3	52.3	64.0
	10	50.3	-70.7	27.3

表 22 より、平均的に HP をプレイヤー側がより多く残し、AI に勝つ傾向にあったが、HP 差はあまり大きくなく、平均 10 標準偏差 0.46 の条件では負けていたことがわかる。

9-3-3 考察

図 19 及び表 20 より、Attention 最適条件である平均 8.0 標準偏差 0.46 の正規乱数条件

が平均 Attention において最大値にならず、ガウス関数近似による予測の通りにはならなかった。この理由として、以下の点が考えられる：図 19 では中央が高くなり、平均 6 標準偏差 0.1 から平均 10 標準偏差 1 に尾根が伸びるような山型の結果の等高線グラフを想定していた。しかし、そのような結果のグラフにはならず、ガウス関数近似と各試合での平均 Attention とが平均 7.27 離れていた。このことからガウス関数による近似が平均 Attention の変化に適していなかったことが理由の 1 つとして考えられる。

第 10 章 独自の集中指標

ここまでの実験ではプレイヤー（被験者）の集中度合を把握するための指標として、Attention（脳波）を用いてきたが、安定性に欠けるため試合数を増やす、もしくは測定項目を増やす必要性が考えられた。しかし、試合数を増やすことはプレイヤー（被験者）の負担が増えることに繋がる懸念される。よって、集中状態を測るための生体信号を新たに追加する。

高津らの研究[5]では、講義中の生徒の状態について RRI の増大と集中の弛緩が関係しているとされている。すなわち、R 波（心電図波形におけるピークの部分）の間隔が長くなり心拍数が低くなる時集中度合が下がるということである。よって本研究の集中状態を測るための生体信号として心拍数を追加し、心拍数が高い方がより集中しているとみなす。心拍数は Fitbit 社の fitbit inspire2 を用いて、試合中に測定された数値データの平均を測定値とする。fitbit inspire2 の詳細は付録のセクション A に示す。

兜森らの研究[6]では、高度な計算作業のタスクにおいては瞬目回数が少なくなるとされている。すなわち、集中力が必要とされるタスクでは瞬目回数が少なくなりやすいということである。よって本研究の集中状態を測るための生体信号として瞬目回数を追加し、瞬目回数が少ない方がより集中しているとみなす。瞬目回数についてはジズ社の JINS MEME を用いて、試合中に瞬きの速度において 0 より大きい値が測定された回数から 1 分間当たりの瞬目回数を算出し、それを測定値とする。JINS MEME を用いた瞬目回数の測定では、メガネ本体がズレてしまうなど装着状態の原因から瞬きが正しく記録されない場合があった。そこで本研究では瞬目回数には下限を設けて、それより小さい値は測定ミスとして扱うこととする。Tubota らの研究[7]では、ヒトはリラックス時に平均して約 21 回の瞬きをしていたが、デジタルテキストを読むときには平均して約 7 回にまで減少したとされている。よって本研究では少なめに見積もって 1 分間当たりの回数が 5 回未満のデータは測定ミスとする。JINS MEME の詳細は付録のセクション A に示す。

Attention, 心拍数, 瞬目回数をそれぞれ標準化し、式(1)で計算される値を独自の集中指標（独自指標）として用いる。

$$O_i = \frac{A_i + H_i - B_i}{3} \quad (1)$$

ここでの各種変数は表 23 の通りである。

表 23 式(1)及び式(2)の各種変数

O_i	条件 <i>i</i> の独自指標
A_i	条件 <i>i</i> の Attention (標準化)
H_i	条件 <i>i</i> の心拍数 (標準化)
B_i	条件 <i>i</i> の瞬目回数 (標準化)

但し、瞬目回数が測定ミスとして扱われた条件については式(2)で計算される値を独自指標として用いる。

$$O_i = \frac{A_i + H_i}{2} \quad (2)$$

第 1 1 章 独自の集中指標を用いた難易度調整 (2 パラメータ)

本章では独自指標に基づいて、正規乱数の平均と標準偏差の 2 パラメータを同時に推定し、難易度調整を行う実験について示す。被験者 3 名 (A, C, D) に実験を行った。なお、被験者 A は筆者である。

本章の推定実験では、操作練習を 3 分間行ってもらった後、第 7 章と同様に以下の 9 つの正規乱数の条件の AI とランダムな順番で 1 試合ずつ行ってもらう：①平均値 2 標準偏差 0.5, ②平均値 2 標準偏差 1, ③平均値 2 標準偏差 2, ④平均値 5 標準偏差 0.5, ⑤平均値 5 標準偏差 1, ⑥平均値 5 標準偏差 2, ⑦平均値 8 標準偏差 0.5, ⑧平均値 8 標準偏差 1, ⑨平均値 8 標準偏差 2。

各試合の独自指標をガウス関数近似して、独自指標が最大になる正規乱数の条件 (独自指標最適条件) を推定する。なお試合間で休憩を 3 分取ってもらおう。ガウス関数近似の詳細は付録のセクション E に示す。

本章の検証実験では、操作練習を 3 分間行ってもらった後、独自指標最適条件を含む 9 条件の AI と 1 試合ずつ行ってもらおう。独自指標最適条件の AI と対戦してもらったときに、独自指標が最大になるか、加えて楽しい試合を体験させられたかを確認した。なお検証実験の 9 条件は個々のプレイヤーによって異なる。1 試合ごとにアンケートに回答してもらい、その後に休憩を取ってもらった。アンケートと休憩を合わせて 3 分とした。第 6 章と同様にアンケートの内容は 8 項目で、1~6 の 6 段階で 6 を最高評価として回答してもらった。

11-1 被験者 A での実験

本セクションでは被験者 A をプレイヤーとしたときの実験結果とその考察について示す。

11-1-1 推定実験及びその結果

推定実験として、本章冒頭に記した 9 条件で行ってもらった各試合の独自指標をガウス関数で近似した。

推定実験の結果として各試合の独自指標を図 20 及び表 24 に示す。図 20 において、横軸は正規乱数の標準偏差、奥行きは正規乱数の平均であり、各頂点は各試合の独自指標を示す。表 20 において縦軸は正規乱数の平均であり、横軸は正規乱数の標準偏差である。太字での表記は各試合の独自指標を示し、赤字での表記は独自指標の最大を示す。

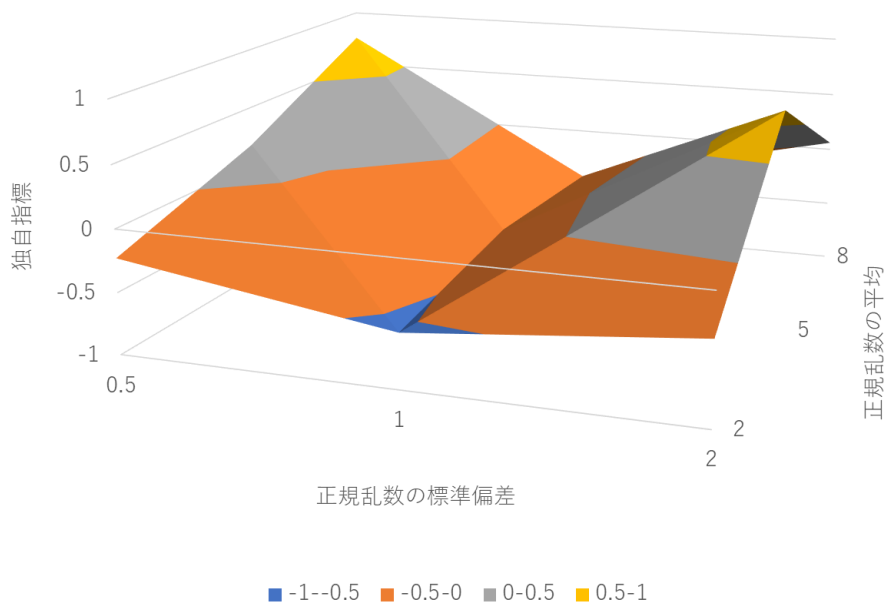


図 20 推定実験結果のグラフ

表 24 推定実験結果の表

独自指標		正規乱数の標準偏差		
		0.5	1	2
正規乱数の平均	2	-0.229	-0.562	-0.344
	5	0.170	-0.377	0.795
	8	0.757	-0.437	0.060

推定実験の結果をガウス関数で近似したグラフを図 21 に示す。横軸は正規乱数の標準偏差，奥行きは正規乱数の平均であり，縦軸は独自指標を示す。

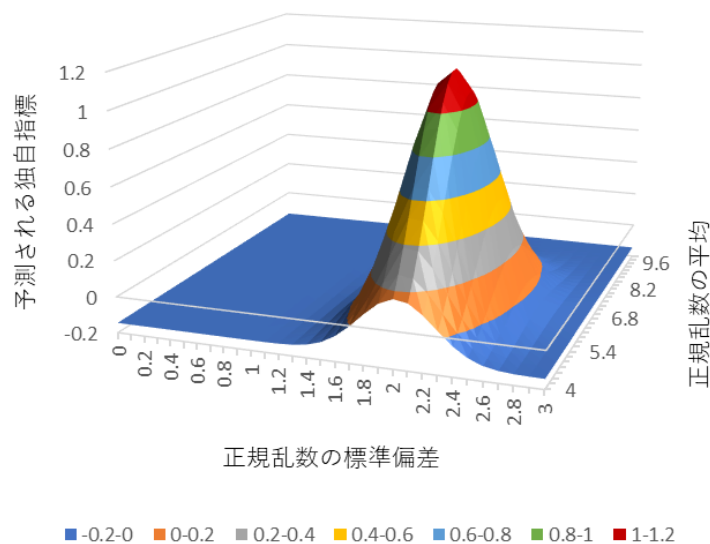


図 21 ガウス関数近似のグラフ

図 21 より，ガウス関数近似から独自指標最適条件は平均 5.9 標準偏差 2.09 である。

11-1-2 検証実験及びその結果

推定実験より，独自指標最適条件は，平均 5.9 標準偏差 2.09 である。よって平均 5.9 標準偏差 2.09 を含む以下の 9 条件下で試合を行ってもらい，検証した：①平均 4 標準偏差 1.5，②平均 4 標準偏差 2.09，③平均 4 標準偏差 2.5，④平均 5.9 標準偏差 1.5，⑤平均 5.9 標準偏差 2.09，⑥平均 5.9 標準偏差 2.5，⑦平均 8 標準偏差 1.5，⑧平均 8 標準偏差 2.09，⑨平均 8 標準偏差 2.5。

検証実験の結果として各試合での独自指標を図 22 及び表 25 に示す。図 22 の見方は図 20 と同じである。表 25 において縦軸は正規乱数の平均であり，横軸は正規乱数の標準偏差である。太字での表記は各試合での独自指標を示し，赤字での表記は独自指標最適条件の結果を示す。緑字での表記は 9 条件の内の独自指標の最大値を示す。

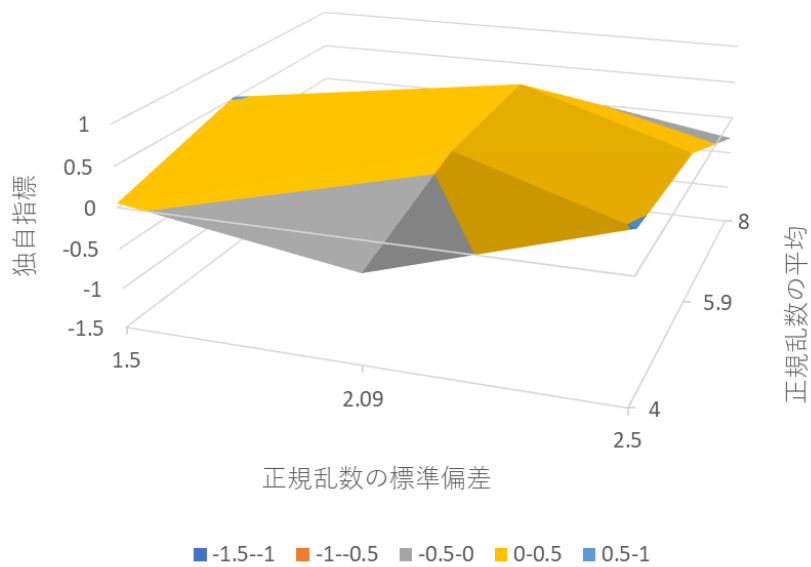


図 22 検証実験結果のグラフ

表 25 検証実験結果の表

独自指標		正規乱数の標準偏差		
		1.5	2.09	2.5
正規乱数の平均	4	0.053	-0.390	0.519
	5.9	0.516	0.105	0.405
	8	-1.036	0.193	-0.290

図 22 及び表 25 より、独自指標最適条件である平均 5.9 標準偏差 2.09 の正規乱数条件が独自指標において最大とならず、ガウス関数近似による予測の通りにならなかったことがわかる。図 21 におけるガウス関数近似と図 22 における検証実験の各試合の独自指標との差の絶対値は平均 0.488 であった。すなわち、検証実験の各試合の独自指標が平均 0.488 だけガウス関数近似から離れた値であった。

検証実験の結果としてアンケートの回答結果を表 26 に示す。赤字での表記は独自指標最適条件での回答結果と独自指標を、緑字での表記は 9 条件の内の独自指標の最大値を示す。回答結果及び独自指標は各条件での 1 試合の結果である。

表 26 検証実験のアンケート結果

正規乱数の平均, 標準偏差		4, 1.5	4, 2.09	4, 2.5	5.9, 1.5	5.9, 2.09	5.9, 2.5	8, 1.5	8, 2.09	8, 2.5
アンケート項目	・試合が楽しかったか?	3	4	3	5	4	4	5	5	4
	・相手が強いと感じたか?	3	4	4	4	4	4	5	5	5
	・相手の行動が予測できたか?	3	2	2	4	3	3	5	3	3
	・キャラクターを思い通りに動かせたか (操作ミスしなかったか)?	5	4	5	6	5	6	5	6	5
	・相手の行動が複雑(多様)だと感じたか?	4	4	6	3	4	5	3	4	5
	・ゲームに集中していたと思うか?	3	4	3	5	4	4	5	5	4
	・新しいコマンドに積極的に挑戦したか?	1	2	2	1	2	2	2	1	2
	・勝ちたい(負けたくない)と思ったか?	2	4	2	3	3	2	4	5	4
独自指標		0.05	-0.39	0.52	0.52	0.11	0.41	-1.04	0.19	-0.29

表 26 より, アンケートの試合の楽しさにおいて独自指標最適条件である平均 5.9 標準偏差 2.09 の正規乱数条件が 9 条件のうちで最高評価とならなかったことがわかる. また, アンケートの回答結果と各試合での独自指標の間で, 特に相関性が見られる項目はなかった.

各試合の各ラウンド終了時のプレイヤーと AI の平均 HP 差を表 27 に示す. 各数値が正のときは, 平均的にラウンド終了時にプレイヤー側が HP をより多く残していたことを表す.

表 27 ラウンド終了時の平均 HP 差

平均 HP 差		正規乱数の標準偏差		
		1.5	2.09	2.5
正規乱数の平均	4	186.3	98.7	154.0
	5.9	130.0	143.3	188.3
	8	120.3	124.7	129.7

表 27 より, 平均的に HP をプレイヤー側がより多く残し, AI に勝つ傾向にあったことがわかる.

11-1-3 考察

図 22 及び表 25 より, 独自指標最適条件である平均 5.9 標準偏差 2.09 の正規乱数条件が独自指標において最大値にならず, ガウス関数近似による予測の通りにはならなかった. この理由として, 以下の点が考えられる:

- ① 図 22 では中央が高くなり, 平均 4 標準偏差 2.09 から平均 8 標準偏差 2.09 に尾根が伸びるような山型の結果の等高線グラフを想定していた. しかし, そのような結果のグラフにはならず, ガウス関数近似と各試合での独自指標とが平均 0.488 離れてい

た。このことからガウス関数による近似が独自指標の変化に適していなかったことが理由の1つとして考えられる。

- ② 表 27 では正規乱数の平均を大きい値に変えたときに AI が強くなり、正規乱数の平均と HP 差の全体平均が負の相関関係になることを期待していた。しかし、そのような結果にはならず、正規乱数の平均に関わらず被験者 A が AI に勝利していた。このことから改良を加えたが依然として被験者の実力に対して、AI の根本的な強さが不十分であったことが理由の1つとして考えられる。

11-2 被験者 C での実験

本セクションでは被験者 C をプレイヤーとしたときの実験結果とその考察について示す。

11-2-1 推定実験及びその結果

推定実験として、本章冒頭に記した 9 条件で行ってもらった各試合の独自指標をガウス関数で近似した。

推定実験の結果として各試合の独自指標を図 23 及び表 28 に示す。図 23 の見方は図 20 と同じである。表 28 の見方は表 24 と同じである。

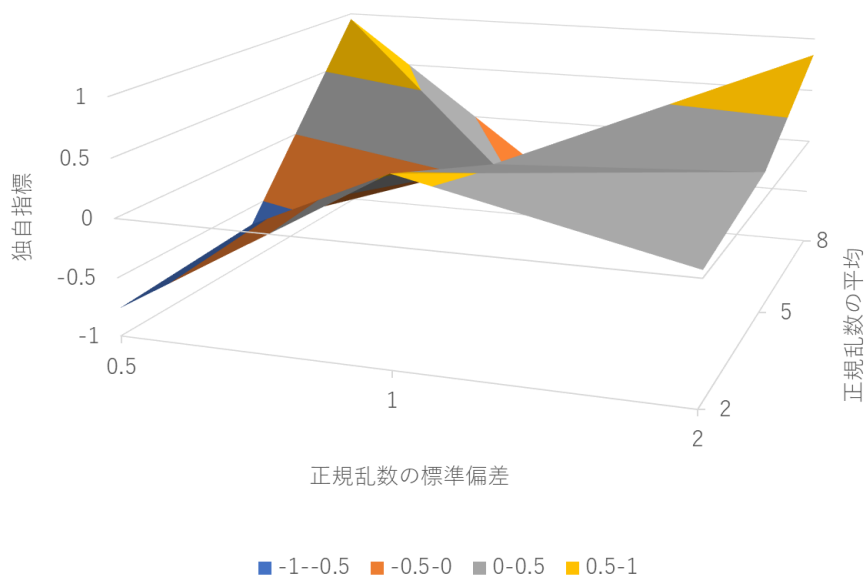


図 23 推定実験結果のグラフ

表 28 推定実験結果の表

独自指標		正規乱数の標準偏差		
		0.5	1	2
正規乱数の平均	2	-0.763	0.571	0.062
	5	-0.667	0.111	0.238
	8	0.945	-0.700	0.845

推定実験の結果をガウス関数で近似したグラフを図 24 に示す。図 24 の見方は図 21 と同じである。

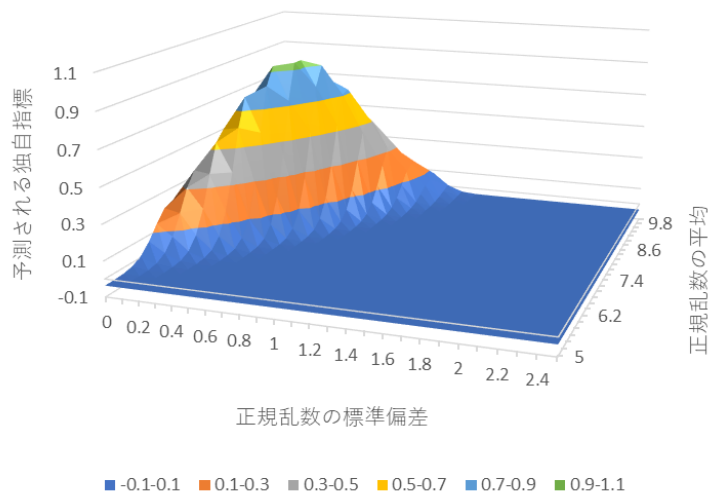


図 24 ガウス関数近似のグラフ

図 24 より、ガウス関数近似から独自指標最適条件は平均 8.1 標準偏差 0.49 である。

11-2-2 検証実験及びその結果

推定実験より、独自指標最適条件は、平均 8.1 標準偏差 0.49 である。よって平均 8.1 標準偏差 0.49 を含む以下の 9 条件下で試合を行ってもらい、検証した：①平均 6 標準偏差 0.1, ②平均 6 標準偏差 0.49, ③平均 6 標準偏差 1, ④平均 8.1 標準偏差 0.1, ⑤平均 8.1 標準偏差 0.49, ⑥平均 8.1 標準偏差 1, ⑦平均 10 標準偏差 0.1, ⑧平均 10 標準偏差 0.49, ⑨平均 10 標準偏差 1。

検証実験の結果として各試合での独自指標を図 25 及び表 29 に示す。図 25 の見方は図 20 と同じである。表 29 の見方は表 25 と同じである。

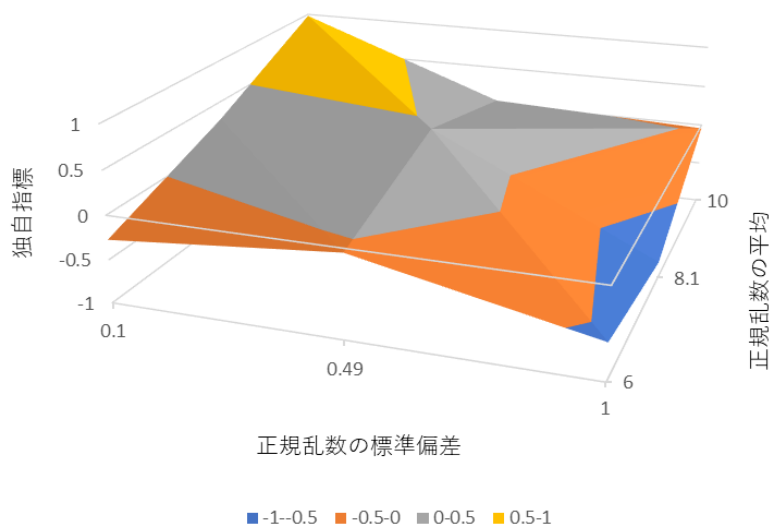


図 25 検証実験結果のグラフ

表 29 検証実験結果の表

独自指標		正規乱数の標準偏差		
		0.1	0.49	1
正規乱数の平均	6	-0.276	-0.048	-0.583
	8.1	0.285	0.448	-0.818
	10	0.979	0.055	-0.043

図 25 及び表 29 より、独自指標最適条件である平均 8.1 標準偏差 0.49 の正規乱数条件が独自指標において最大とならず、ガウス関数近似による予測の通りにならなかったことがわかる。図 24 におけるガウス関数近似と図 25 における検証実験の各試合の独自指標との差の絶対値は平均 0.414 であった。すなわち、検証実験の各試合の独自指標が平均 0.414 だけガウス関数近似から離れた値であった。

検証実験の結果としてアンケートの回答結果を表 30 に示す。表 30 の見方は表 26 と同じである。

表 30 検証実験のアンケート結果

正規乱数の平均, 標準偏差		6, 0.1	6, 0.49	6, 1	8.1, 0.1	8.1, 0.49	8.1, 1	10, 0.1	10, 0.49	10, 1
アンケート項目	・試合が楽しかったか?	6	6	5	6	6	6	4	5	6
	・相手が強いと感じたか?	6	5	5	6	5	6	6	6	6
	・相手の行動が予測できたか?	6	6	6	6	6	6	5	5	6
	・キャラクターを思い通りに動かせたか (操作ミスしなかったか)?	5	6	6	6	5	6	4	5	5
	・相手の行動が複雑(多様)だと感じたか?	5	5	5	6	6	5	5	5	4
	・ゲームに集中していたと思うか?	5	6	5	6	6	6	5	5	5
	・新しいコマンドに積極的に挑戦したか?	6	6	6	6	6	6	5	6	5
	・勝ちたい(負けたくない)と思ったか?	5	5	5	6	5	6	5	5	5
独自指標	-0.28	-0.05	-0.58	0.28	0.45	-0.82	0.98	0.05	-0.04	

表 30 より, アンケートの試合の楽しさにおいて独自指標最適条件である平均 8.1 標準偏差 0.49 の正規乱数条件が 9 条件のうちで最高評価となったことがわかる. また, アンケートの回答結果と各試合での独自指標の間で, 特に相関性が見られる項目は「キャラクターを思い通りに動かせたか(操作ミスしなかったか)?」であった.

各試合の各ラウンド終了時のプレイヤーと AI の平均 HP 差を表 31 に示す. 表 31 の見方は表 27 と同じである.

表 31 ラウンド終了時の平均 HP 差

平均 HP 差		正規乱数の標準偏差		
		0.1	0.49	1
正規乱数の平均	6	21.3	76.7	100.7
	8.1	19.3	51.0	49.0
	10	-127.0	-50.0	-29.7

表 31 より, 正規乱数の平均を高くするにつれて, 平均 HP 差が小さくなっていく傾向にあったことがわかる.

11-2-3 考察

図 25 及び表 29 より, 独自指標最適条件である平均 8.1 標準偏差 0.49 の正規乱数条件が独自指標において最大値にならず, ガウス関数近似による予測の通りにはならなかった. この理由として, 以下の点が考えられる: 図 25 では中央が高くなり, 平均 6 標準偏差 0.1 から平均 10 標準偏差 1 に尾根が伸びるような山型の結果の等高線グラフを想定していた. しかし, そのような結果のグラフにはならず, ガウス関数近似と各試合での独自指標とが平均 0.414 離れていた. このことからガウス関数による近似が独自指標の変化に適していなかつ

たことが理由の1つとして考えられる。

表 30 より、アンケートの回答結果と各試合での独自指標の間で、相関性が見られる項目は「キャラクターを思い通りに動かせたか(操作ミスしなかったか)?」という項目であり、相関係数が -0.702 であった。この理由について、以下の点が考えられる：プレイヤーが高い集中状態にあるとゲーム状況の変化やAIの行動に素早く反応できるようになる。しかし、それに合わせてゲームコントローラーを素早く入力しようとして操作ミスを多く引き起こしたということ。

表 30 及び表 31 より、アンケートの試合の楽しさについての回答が最高値である 6 とされている条件では、平均 HP 差が $-29.7\sim 51.0$ の間であった。このことから被験者 C は、より接戦である試合に対して楽しさを感じていると考えられる。これは試合がより接戦であればより楽しいとした従来研究[4]の主張に沿った結果である。

11-3 被験者 D での実験

本セクションでは被験者 D をプレイヤーとしたときの実験結果とその考察について示す。

11-3-1 推定実験及びその結果

推定実験として、本章冒頭に記した 9 条件で行ってもらった各試合の独自指標をガウス関数で近似した。

推定実験の結果として各試合の独自指標を図 26 及び表 32 に示す。図 26 の見方は図 20 と同じである。表 32 の見方は表 24 と同じである。

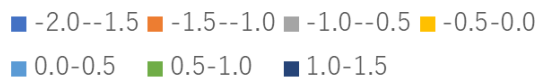
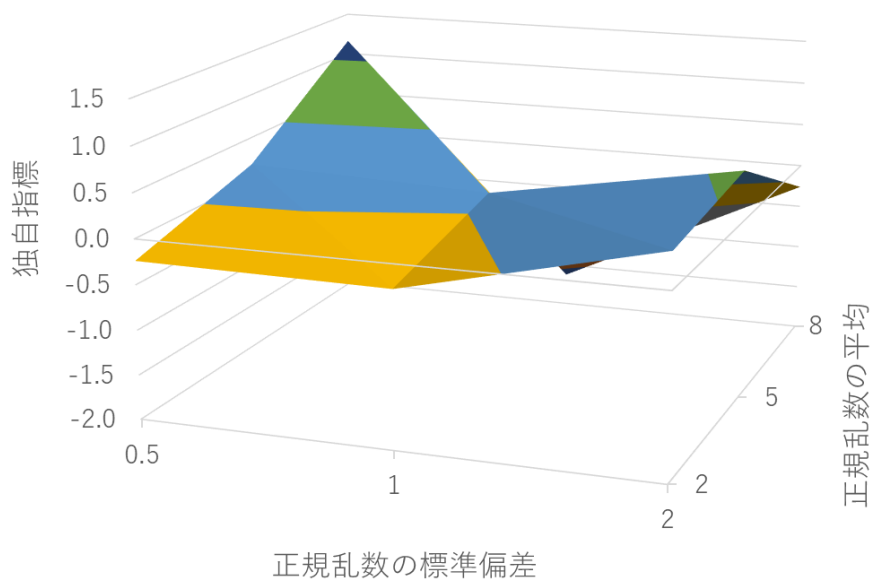


図 26 推定実験結果のグラフ

表 32 推定実験結果の表

独自指標		正規乱数の標準偏差		
		0.5	1	2
正規乱数の平均	2	-0.239	-0.266	0.400
	5	0.184	0.081	0.567
	8	1.160	-1.625	-0.262

推定実験の結果をガウス関数で近似したグラフを図 27 に示す. 図 27 の見方は図 21 と同じである.

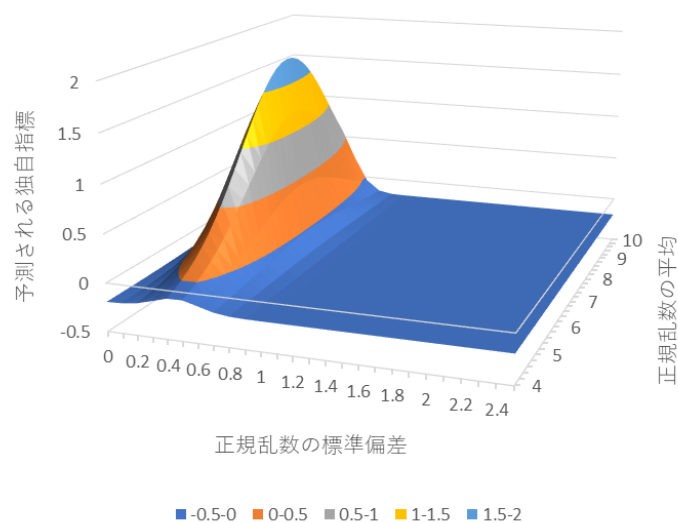


図 27 ガウス関数近似のグラフ

図 27 より，ガウス関数近似から独自指標最適条件は平均 7.9 標準偏差 0.38 である．

11-3-2 検証実験及びその結果

推定実験より，独自指標最適条件は，平均 7.9 標準偏差 0.38 である．よって平均 7.9 標準偏差 0.38 を含む以下の 9 条件下で試合を行ってもらい，検証した：①平均 6 標準偏差 0.1，②平均 6 標準偏差 0.38，③平均 6 標準偏差 1，④平均 7.9 標準偏差 0.1，⑤平均 7.9 標準偏差 0.38，⑥平均 7.9 標準偏差 1，⑦平均 10 標準偏差 0.1，⑧平均 10 標準偏差 0.38，⑨平均 10 標準偏差 1．

検証実験の結果として各試合での独自指標を図 28 及び表 33 に示す．図 28 の見方は図 20 と同じである．表 33 の見方は表 25 と同じである．

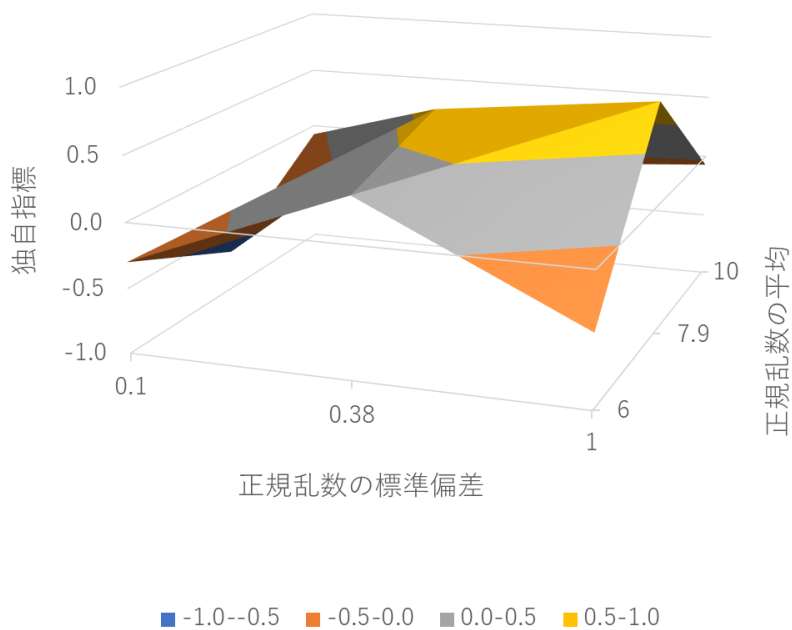


図 28 検証実験結果のグラフ

表 33 検証実験結果の表

独自指標		正規乱数の標準偏差		
		0.1	0.38	1
正規乱数の平均	6	-0.301	0.356	-0.444
	7.9	-0.692	0.620	0.805
	10	-0.077	-0.198	-0.069

図 28 及び表 33 より、独自指標最適条件である平均 7.9 標準偏差 0.38 の正規乱数条件が独自指標において最大とならず、ガウス関数近似による予測の通りにならなかったことがわかる。図 27 におけるガウス関数近似と図 28 における検証実験の各試合の独自指標との差の絶対値は平均 0.549 であった。すなわち、検証実験の各試合の独自指標が平均 0.549 だけガウス関数近似から離れた値であった。

検証実験の結果としてアンケートの回答結果を表 34 に示す。表 34 の見方は表 26 と同じである。

表 34 検証実験のアンケート結果

正規乱数の平均, 標準偏差		6, 0.1	6, 0.38	6, 1	7.9, 0.1	7.9, 0.38	7.9, 1	10, 0.1	10, 0.38	10, 1
アンケート項目	・試合が楽しかったか?	5	4	4	5	4	5	2	4	4
	・相手が強いと感じたか?	4	2	2	3	3	3	1	3	5
	・相手の行動が予測できたか?	3	2	2	6	5	3	5	4	2
	・キャラクターを思い通りに動かせたか (操作ミスしなかったか)?	2	4	4	4	4	4	4	5	4
	・相手の行動が複雑(多様)だと感じたか?	4	5	4	2	3	4	2	3	5
	・ゲームに集中していたと思うか?	5	4	4	6	3	5	4	4	6
	・新しいコマンドに積極的に挑戦したか?	2	3	4	2	2	4	4	3	2
	・勝ちたい(負けたくない)と思ったか?	5	4	4	5	4	5	2	4	4
独自の集中指標		-0.30	0.36	-0.44	-0.69	0.62	0.80	-0.08	-0.20	-0.07

表 34 より、アンケートの試合の楽しさにおいて独自指標最適条件である平均 7.9 標準偏差 0.38 の正規乱数条件が 9 条件のうちで最高評価とならなかったことがわかる。表 34 より、アンケートの回答結果と各試合での独自指標の間で、特に相関性が見られる項目はなかった。

各試合の各ラウンド終了時のプレイヤーと AI の平均 HP 差を表 35 に示す。表 35 の見方は表 27 と同じである。

表 35 ラウンド終了時の平均 HP 差

平均 HP 差		正規乱数の標準偏差		
		0.1	0.38	1
正規乱数の平均	6	-84.0	25.7	78.0
	7.9	-62.0	94.7	-45.0
	10	46.3	83.7	1.0

表 35 より、正規乱数の平均と標準偏差の設定に関わらず、平均 HP 差が 0 に近く接戦になる傾向にあったことがわかる。

11-3-3 考察

図 28 及び表 33 より、独自指標最適条件である平均 7.9 標準偏差 0.38 の正規乱数条件が独自指標において最大値にならず、ガウス関数近似による予測の通りにはならなかった。この理由として、以下の点が考えられる：図 28 では中央が高くなり、平均 6 標準偏差 0.38 から平均 10 標準偏差 0.38 に尾根が伸びるような山型の結果の等高線グラフを想定していた。しかし、そのような結果のグラフにはならず、ガウス関数近似と各試合での独自指標とが平均 0.549 離れていた。このことからガウス関数による近似が独自指標の変化に適していな

かったことが理由の1つとして考えられる。

第12章 実験結果まとめ

本章では、ここまでの実験結果をまとめ、その考察について示す。

10回の実験結果から推定実験によって、プレイヤーの集中度合（平均 Attention または独自指標）を最大化できたか及び試合の楽しさを最大化できたか、という2点をまとめた表を表36に示す。各種実験において何回成功したかを示し、斜線の右側が試行回数で左側が成功回数である。

表36 全体の結果（集中度合と楽しさ）

		集中度合の最大化	楽しさの最大化
Attentionでの実験	1パラメータ	0/3	1/3
	2パラメータ	2/4	0/4
独自指標での実験	2パラメータ	0/3	1/3
合計		2/10	2/10

表36より、全体では集中度合の最大化と楽しさの最大化がそれぞれ10回中2回成功したことがわかる。一方で、全被験者の各検証実験の全条件（5条件または9条件）から1条件を等確率で無作為に選ぶと考えたとき、特定の条件が選ばれる確率は1回の検証実験あたり平均で13.8%である。集中度合の最大化と楽しさの最大化において、ランダムと考えた平均確率よりも実際の成功率のほうがわずかに高い。このことから生体信号を用いて集中度合や楽しさを把握しようとする本研究の試みは、精度は低いものの、ある程度達成できているのではないかと考えられる。

10回の実験結果において、集中度合とアンケートの試合の楽しさの回答結果の間で強い正の相関が確認された回数は0であった。この結果は、第1章で紹介した「没入感やゾーンといった高い集中状態を体験できるゲームほど、より楽しいと評価される」というChenの理論[1]には沿わない結果である。この結果の理由として、以下の点が考えられる：プレイヤーの集中度合を測定する部分で生体信号を用いているために、集中度合の測定精度が低くなってしまふということが考えられる。

第6章のAttentionでの実験（セクション6-1及びセクション6-2）の結果において、同条件での平均Attentionのバラつきが大きいということから、平均Attentionは精度が低く安定性に欠けると考察してきた。このことから推定実験及び検証実験の試合数を増やすか、生体信号の測定項目を増やすべきではないかと考えた。本研究ではプレイヤーの負担も考慮して測定項目を増やす方向性で研究を進め、心拍数と瞬目回数を測定項目に追加するこ

ととした。これらの項目を追加したことによって、当初の問題であった集中度合の測定精度や安定性が改善されたのかについてまとめる。但し、独自指標での実験では、同プレイヤーによる同条件での試合を複数回で行っていないため、同条件での集中度合のバラつきという点では比較ができない。

よって、実験 k における非安定性 (NS_k) として、集中度合のガウス関数近似の山の高さ (HE_k) に対する、ガウス関数近似と検証実験結果の集中度合の差の絶対値の平均 (DI_k) の割合を比較する。ガウス関数近似は AI の強さを変えたときに起こる集中度合の変化を推定したものであるから、その山の高さ (最大値と最小値の差) (HE_k) は AI の強さを変えたときに起こる集中度合の変化量にあたる。これに対して、ガウス関数近似と検証実験結果の集中度合の差の絶対値の平均 (DI_k) は、推定に対して検証結果がどれだけ離れているかを示す値である。よって非安定性 (NS_k) は、起こり得る集中度合の変化量に対して、推定と検証での集中度合の乖離の割合を計算したものであり、ガウス関数近似に対してどれだけ沿った検証実験結果が得られたかを示す。すなわち、測定される集中度合の再現性の無さを示す。

非安定性は式(3)で計算される。

$$NS_k = \frac{DI_k}{HE_k} = \frac{\frac{1}{N_k} \sum_{l=1}^{N_k} |ES_{kl} - VE_{kl}|}{(max_k - min_k)} \quad (3)$$

ここでの各種変数は表 37 の通りである。

表 37 数式(3)の各種変数

NS_k	実験 k の非安定性
DI_k	実験 k のガウス関数近似と検証実験結果の集中度合の差の絶対値の全条件平均
HE_k	実験 k の集中度合のガウス関数近似の山の高さ
ES_{kl}	実験 k の条件 l におけるガウス関数近似による推定値
VE_{kl}	実験 k の条件 l における検証実験での測定値
max_k	実験 k のガウス関数近似の最大値
min_k	実験 k のガウス関数近似の最小値 (近似したガウス関数の定数項 b)
N_k	実験 k の条件数 (第 9 章と第 11 章では $N_k = 9$)

第 9 章 (脳波を用いた難易度調整 (2 パラメータ②)) 及び第 11 章 (独自の集中指標を用いた難易度調整 (2 パラメータ)) での被験者 3 名 (A, C, D) の実験結果における非安定性を表 38 示す. 最も右側の列は 3 名の平均を表す. また最下段は Attention での実験 (第 9 章) の値を基準とし, 独自指標での実験 (第 11 章) との差を示す. 被験者 3 名の非安定性の平均 ($\pm 1SD$) を図 29 に示す. 青のグラフが Attention での実験 (第 9 章) での平均で橙色のグラフが独自指標での実験 (第 11 章) での平均である.

表 38 被験者 3 名の非安定性

被験者	A	C	D	平均
Attention (第9章)	0.757	0.607	0.410	0.591
独自指標 (第11章)	0.366	0.411	0.277	0.351
差	-0.391	-0.196	-0.133	-0.240

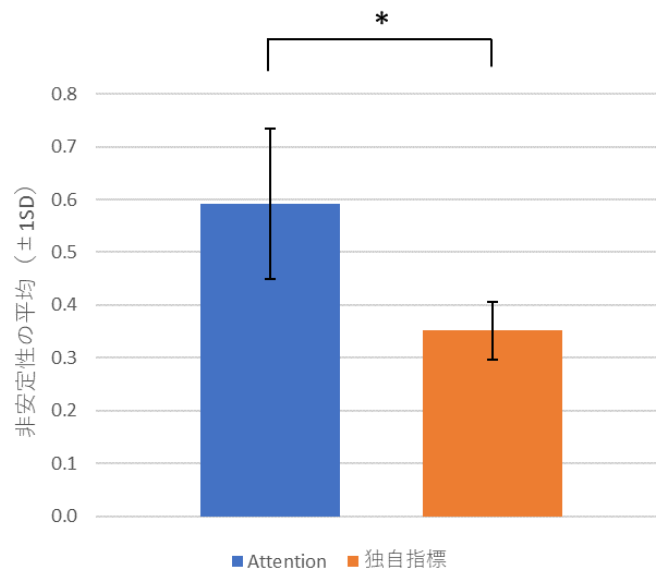


図 29 被験者 3 名の非安定性の平均

表 38 より，被験者 3 名とも Attention での実験（第 9 章）のほうが独自指標での実験（第 11 章）よりも非安定性が大きいことがわかる．図 29 に示す各実験での平均の差に対して，対応ありの t 検定を行った．帰無仮説は「非安定性において Attention での実験（第 9 章）の平均と独自指標での実験（第 11 章）の平均に差は無い」であり，対立仮説は「非安定性において Attention での実験（第 9 章）の平均と独自指標での実験（第 11 章）の平均に差は有る」である．被験者が 3 名なので自由度は 2 であり，自由度 2 の有意水準 5% の片側検定の t 値は -2.92 である．一方で，非安定性において Attention での実験（第 9 章）の平均と独自指標での実験（第 11 章）の平均の差の検定統計量は -3.09 であった．このことから，片側有意水準 5% のもとで「非安定性は Attention での実験（第 9 章）の平均よりも独自指標での実験（第 11 章）の平均のほうが有意に小さい」といえる．このことから，生体信号の測定項目を追加したことにより，集中度合の測定の安定性が改善されたと考えられる．

第 13 章 比較実験

本章では本研究において提案する難易度調整の手法の有用性を従来手法と比較する．比較実験として，本研究において提案する 2 種類の難易度調整の手法（Attention に基づいた難易度調整と独自指標に基づいた難易度調整）と従来研究[4]のような試合をより接戦にしようとする難易度調整の手法のうち，どの手法がプレイヤーにより楽しい試合を体験させられるのかを確かめる実験を行った．

13-1 実験概要

本実験では、第9章（脳波を用いた難易度調整（2パラメータ②））における Attention 最適条件の AI（Attention 最適 AI）と第11章（独自の集中指標を用いた難易度調整（2パラメータ））における独自指標最適条件の AI（独自指標最適 AI）及び従来研究[4]のように試合を接戦にしようとする AI（接戦 AI）の3条件の AI と対戦してもらう。そして、どの試合が一番楽しかったのかをプレイヤーに確認する。

本実験は被験者3名（A, C, D）に実験を行った。なお、被験者 A は筆者である。

本実験では、操作練習を3分間行ってもらった後、以下の3つの条件の AI とランダムな順番で1試合ずつ行ってもらう：①Attention 最適 AI, ②独自指標最適 AI, ③接戦 AI。

Attention, 心拍数及び瞬目回数を測定しながら各条件につき1試合ずつ対戦してもらう。1試合ごとにアンケートに回答してもらい、その後に休憩を取ってもらった。アンケートと休憩を合わせて3分とした。第6章と同様にアンケートの内容は8項目で、1~6の6段階で6を最高評価として回答してもらった。最後にプレイヤーに3試合のうちで、楽しかった順番をつけてもらった。

13-2 接戦 AI の仕様

接戦 AI は従来研究[4]で用いられた AI ではないが、従来研究[4]で提唱された互角の試合が一番楽しいとする主張に基づいた AI である。すなわち、接戦 AI はプレイヤーと AI の HP 差が0になるように行動する AI である。接戦 AI はモンテカルロ木探索を利用した複数回のシミュレーションを基に行動を決定する AI である。現在のゲーム状況を基に次の行動をシミュレーションし、その結果を評価することを規定の時間が経過するまで繰り返す。但し、公式の AI コンペティションのレギュレーションに則り、接戦 AI がシミュレーションするゲーム状況は15フレーム（0.25秒）前のものとした。なお、シミュレーションは接戦 AI が行動を1度行うと同時に、シミュレーション上のプレイヤー（仮想プレイヤー）が行動を1度行うというものである。仮想プレイヤーの行動はシミュレーションごとにランダムで決定される。接戦 AI は行動の評価値が高い行動ほど重点的にシミュレーションし、一定時間が経過したタイミングで最もシミュレーション回数が多い行動を採用し、実際のゲームにて実行する。

シミュレーションに関わる数値として、接戦 AI の行動 j に対する評価値 (UCB_j) と接戦 AI の行動 j に対する i 回目のシミュレーション ij に対する得点 ($score_{ij}$) が存在する。接戦 AI はシミュレーションを行う際に行動の評価値 (UCB_j) を参照して、評価値 (UCB_j) が最大の行動をシミュレーション上での接戦 AI の行動とする。なお、各種行動について最低でも1回はシミュレーションされるように、評価値 (UCB_j) の初期値はかなり大きい数値がセットされる。

接戦 AI のシミュレーション ij の得点 ($score_{ij}$) は式(4)で計算される。接戦 AI と仮想プレイヤー両方の行動が終わった後の HP 差が小さくなる行動がより高く評価され、HP 差が大きくなる行動がより低く評価される式となっている。

$$score_{ij} = -|AA_{ij} - AP_{ij}| \quad (4)$$

ここでの各種変数は表 39 の通りである.

表 39 数式(4)の各種変数

$score_{ij}$	シミュレーション ij の得点
AA_{ij}	シミュレーション ij において 接戦 AI の行動 j と仮想プレイヤーの行動 (ランダム) が 終わった後の接戦 AI の HP
AP_{ij}	シミュレーション ij において 接戦 AI の行動 j と仮想プレイヤーの行動 (ランダム) が 終わった後の仮想プレイヤーの HP

接戦 AI の行動の評価値 (UCB_j) は式(5)で計算される. 第 1 項は接戦 AI の行動 j をシミュレーションしたときの得点 ($score_{ij}$) の平均であり, n_j は行動 j のシミュレーション回数である. すなわち, 第 1 項は接戦 AI の行動 j を採用したときの期待値である. 第 2 項は接戦 AI の行動 j のシミュレーション回数 (n_j) が全体のシミュレーション回数 (N) に対して少ないときに大きな値をとる項である. これによって, 期待値の小さい行動 j であっても, シミュレーション回数 (n_j) が少ないのであれば, その期待値に信憑性が薄いとして, さらに行動 j にシミュレーションを行うようになる. なお, c は正の定数であり, 評価値 (UCB_j) への第 1 項と第 2 項の影響度のバランスを決めている.

$$UCB_j = \frac{1}{n_j} \sum_{i=1}^{n_j} score_{ij} + c \sqrt{\frac{2 \ln(N)}{n_j}} \quad (5)$$

ここでの各種変数及び式は表 40 の通りである.

表 40 数式(5)の各種変数及び式

UCB_j	行動 j の評価値
$\frac{1}{n_j} \sum_{i=1}^{n_j} score_{ij}$	行動 j の得点の平均値
c	正の定数（実験では $c = 3$ ）
N	全体のシミュレーション回数
n_j	行動 j のシミュレーション回数

モンテカルロ木探索の仕組みとして、行動 j のシミュレーション回数 n_j が閾値を超えた場合に、子ノードとして、その行動 j の後のゲーム状況に対してシミュレーションを行う。すなわち、現在のゲーム状況から接戦 AI が行動 j を1度行い、仮想プレイヤーがランダムな行動を1度行った後の状況から、接戦 AI が行動 k を1度行い、仮想プレイヤーがランダムな行動を1度行ったとするシミュレーションである。そして、そのシミュレーション結果は親ノードの行動 j の評価に影響する。行動 j のシミュレーション回数が閾値を超え、子ノードが展開されている場合、子ノードのシミュレーションの得点は、そのまま親ノードのシミュレーションの得点にもされる。またシミュレーション回数は子ノードと親ノードのそれぞれで1回ずつ増やされる。接戦 AI の詳細は付録のセクション D に示す。

13-3 被験者 A での実験及びその結果

本セクションでは被験者 A をプレイヤーとしたときの実験結果について示す。

比較実験として、13-1 実験概要に記した 3 条件で 1 試合ずつ行ってもらう。9-1-1 Attention に基づく難易度調整の被験者 A における推定実験及び 11-1-1 独自指標に基づく難易度調整の被験者 A における推定実験より、被験者 A の比較実験の 3 条件は以下のようになる：①平均 8 標準偏差 0.5 (Attention 最適 AI)、②平均 5.9 標準偏差 2.09 (独自指標最適 AI)、③接戦 AI。

比較実験の結果として各試合の平均 Attention、心拍数、瞬目回数、独自指標、ラウンド終了時の平均 HP 差及び楽しかった順番を表 41 に示す。ここで平均 HP 差はラウンド終了時のプレイヤーの HP から AI の HP を引いた値の 3 ラウンド平均であり、正の値であるときはプレイヤーが平均的に HP をより多く残していたことを表す。赤字での表記は瞬目回数が測定ミスとして扱われていることを示す。被験者 A の実験結果では、瞬目回数が 1 回しか正しく測定できなかったため、標準化が行えなかった。よって独自指標は全て平均 Attention と心拍数からの算出になっている。

表 41 比較実験の結果

条件	Attention 最適AI	独自指標 最適AI	接戦AI
平均 Attention	55.32	43.15	50.78
心拍数	68.23	67.38	67.07
瞬目回数 (毎分)	1.49	21.00	0.82
独自指標	1.012	-0.690	-0.322
平均HP差	+138.3	+100.7	+97.3
楽しかった 順番	3	2	1

表 41 より、接戦 AI が最も楽しいと評価され、平均 HP 差が最も 0 に近かったことがわかる。

比較実験の結果としてアンケートの回答結果を表 42 に示す。回答結果は各条件での 1 試合の結果である。

表 42 比較実験のアンケート回答結果

条件	Attention 最適AI	独自指標 最適AI	接戦AI	
アンケート 項目	・試合が楽しかったか？	4	5	6
	・相手が強いと感じたか？	5	4	6
	・相手の行動が予測できたか？	5	3	2
	・キャラクターを思い通りに動かせたか (操作ミスしなかったか)？	6	6	5
	・相手の行動が複雑(多様)だと感じたか？	3	5	6
	・ゲームに集中していたと思うか？	5	5	6
	・新しいコマンドに積極的に挑戦したか？	1	1	1
	・勝ちたい(負けたくない)と思ったか？	4	5	6
楽しかった順番	3	2	1	

表 42 より、被験者 A の感じる楽しい試合は、相手が強く、相手の行動が予測できず、相手の行動が複雑で、勝ちたいという思いが強くなる試合であると考えられる。

13-4 被験者 C での実験及びその結果

本セクションでは被験者 C をプレイヤーとしたときの実験結果について示す。

比較実験として、13-1 実験概要に記した 3 条件で試合を行ってもらう。9-2-1 Attention に基づく難易度調整の被験者 C における推定実験及び 11-2-1 独自指標に基づく難易度調整の被験者 C における推定実験より、被験者 C の比較実験の 3 条件は以下のようになる：① 平均 2 標準偏差 1.04 (Attention 最適 AI)、② 平均 8.1 標準偏差 0.49 (独自指標最適 AI)、③ 接戦 AI。

比較実験の結果として各試合の平均 Attention、心拍数、瞬目回数、独自指標、ラウンド終了時の平均 HP 差及び楽しかった順番を表 43 に示す。表 43 の見方は表 41 と同じである。

表 43 比較実験の結果

条件	Attention 最適AI	独自指標 最適AI	接戦AI
平均 Attention	50.80	61.13	70.68
心拍数	73.29	70.31	71.59
瞬目回数 (毎分)	6.21	4.22	6.23
独自指標	0.246	-0.462	0.062
平均HP差	+201	-11	+29.7
楽しかった 順番	2	3	1

表 43 より、接戦 AI が最も楽しいと評価され、平均 Attention が最も高かったことがわかる。

比較実験の結果としてアンケートの回答結果を表 44 に示す。回答結果は各条件での 1 試合の結果である。

表 44 比較実験のアンケート回答結果

条件		Attention 最適AI	独自指標 最適AI	接戦AI
アンケート 項目	・試合が楽しかったか？	4	5	6
	・相手が強いと感じたか？	3	6	5
	・相手の行動が予測できたか？	5	6	5
	・キャラクターを思い通りに動かせたか (操作ミスしなかったか)？	5	5	5
	・相手の行動が複雑(多様)だと感じたか？	4	4	6
	・ゲームに集中していたと思うか？	4	5	5
	・新しいコマンドに積極的に挑戦したか？	5	5	5
	・勝ちたい(負けたくない)と思ったか？	4	4	6
楽しかった順番		2	3	1

表 44 より、被験者 C の感じる楽しい試合は、相手の行動が予測できず、相手の行動が複雑で、勝ちたいという思いが強くなる試合であると考えられる。

13-5 被験者 D での実験及びその結果

本セクションでは被験者 D をプレイヤーとしたときの実験結果について示す。

比較実験として、13-1 実験概要に記した 3 条件で試合を行ってもらう。9-3-1 Attention に基づく難易度調整の被験者 D における推定実験及び 11-3-1 独自指標に基づく難易度調整の被験者 D における推定実験より、被験者 D の比較実験の 3 条件は以下のようになる：
 ①平均 8 標準偏差 0.46 (Attention 最適 AI)，②平均 7.9 標準偏差 0.38 (独自指標最適 AI)，
 ③接戦 AI。

比較実験の結果として各試合の平均 Attention，心拍数，瞬目回数，独自指標，ラウンド終了時の平均 HP 差及び楽しかった順番を表 45 に示す。表 45 の見方は表 41 と同じである。被験者 D の実験結果では、瞬目回数が全て正しく測定できなかった。よって独自指標は全て平均 Attention と心拍数からの算出になっている。

表 45 比較実験の結果

条件	Attention 最適AI	独自指標 最適AI	接戦AI
平均 Attention	52.79	45.90	51.11
心拍数	72.43	87.00	89.63
瞬目回数 (毎分)	0.000	0.435	0.000
独自指標	-0.174	-0.347	0.521
平均HP差	+54.3	+100.0	+21.0
楽しかった 順番	3	2	1

表 45 より、接戦 AI が最も楽しいと評価されたこと、心拍数が最も高かったこと、独自指標が最も高かったこと及び平均 HP 差が最も 0 に近く接戦であったことがわかる。

比較実験の結果としてアンケートの回答結果を表 46 に示す。回答結果は各条件での 1 試合の結果である。

表 46 比較実験のアンケート回答結果

条件	Attention 最適AI	独自指標 最適AI	接戦AI	
アンケート 項目	・試合が楽しかったか？	2	4	6
	・相手が強いと感じたか？	2	4	5
	・相手の行動が予測できたか？	5	4	3
	・キャラクターを思い通りに動かせたか (操作ミスしなかったか)？	5	6	5
	・相手の行動が複雑(多様)だと感じたか？	2	3	4
	・ゲームに集中していたと思うか？	4	5	6
	・新しいコマンドに積極的に挑戦したか？	1	1	5
	・勝ちたい(負けたくない)と思ったか？	2	4	6
楽しかった順番	3	2	1	

表 46 より、被験者 D の感じる楽しい試合は、相手が強く、相手の行動が予測できず、相手の行動が複雑で、自分自身の集中度合の高まりを実感でき、勝ちたいという思いが強くなる試合であると考えられる。

13-6 比較実験の全体結果と考察

本セクションでは被験者 3 名 (A, C, D) の結果をまとめ、その考察について示す。

被験者 3 名の結果をまとめた全体結果を表 47 に示す。平均 Attention、心拍数及び独自指標は表 41、表 43 及び表 45 より 3 名それぞれで Attention 最適 AI の結果を基準にした変化量を計算し、平均した値である。平均 HP 差及び楽しかった順番は表 41、表 43 及び表 45 より 3 名それぞれの各条件の値をそのまま平均した値である。なお、瞬目回数は測定ミスとされた試合が多かったので省略する。

表 47 比較実験の全体の結果

条件	Attention 最適AI	独自指標 最適AI	接戦AI
平均 Attention	0.00	-2.91	4.55
心拍数	0.00	3.58	4.78
独自指標	0.00	-0.86	-0.27
平均HP差	+131.2	+63.2	+49.3
楽しかった 順番	2.7	2.3	1.0

表 47 より、接戦 AI が最も楽しいと評価されたこと、心拍数が最も高いこと及び平均 HP 差が最も 0 に近いことがわかる。表 47 について以下の点が考えられる：

- ① 接戦 AI が最も楽しいと評価され、尚且つ平均 HP 差が最も 0 に近かった。このことから比較実験の全体結果は、より接戦である方がより楽しいとする従来研究[4]の主張に沿った結果であると考えられる。
- ② 接戦 AI が最も楽しいと評価され、本研究の提案 AI (Attention 最適 AI 及び独自指標最適 AI) は、接戦 AI よりも楽しい試合をプレイヤーに体験させることができな

った。この理由として、接戦 AI と提案 AI では構造が異なるということが考えられる。具体的には行動決定のために、接戦 AI はシミュレーションを行うためゲーム状況を高い精度で把握しているが、提案 AI はゲーム状況と行動の組み合わせに対する評価値を参照するためゲーム状況を 336 種類という非常に粗い精度でしか把握できないということである。例えば、キャラクター間の距離の把握については、接戦 AI は距離が 1 ピクセルでも異なれば、その差がシミュレーションに反映されるのに対して、提案 AI は距離が 0~100 ピクセルまでは同じ距離とみなす仕様となっている。このような AI の構造の差異がプレイヤーの感じる楽しさに違いをもたらしているのではないかと考えられる。

- ③ 独自指標最適 AI が Attention 最適 AI よりも楽しいと評価された。この理由として、脳波 (Attention) だけではなく心拍数及び瞬目回数といった複数の生体信号を用いることで、より多角的により正確にプレイヤーの集中度合及び楽しさを把握できるようになったということが考えられる。

第 14 章 総括

本章では実験全体に対する本研究の結論について示す。また本研究を発展させるために解決するべきであろう課題や、その為のアイデアについて示す。

14-1 結論

本研究の実験結果全体に対する結論を以下に示す。

- ① 従来研究[4]の「試合が接戦であるほど楽しい」という主張に当てはまらないプレイヤーもいるだろう、ということが筆者の考えた仮説であった。そして被験者 B の実験結果 (セクション 6-2) だけは、従来研究[4]の主張に当てはまらない結果と考えられる。よって筆者の主張も間違いではないと考えられる。
- ② プレイヤーの集中度合とゲームの楽しさが比例のような関係性にあるという Chen の理論[1]に沿った実験結果が得られなかった。
- ③ プレイヤーの集中度合の測定に生体信号を用いるという試みについては、本研究の実験においては低い水準であるものの、ある程度集中度合を把握できていていると考えられる。しかし、測定の精度の低さや再現性の低さは課題である。
- ④ 脳波 (Attention) だけでの実験よりも、測定項目に心拍数と瞬目回数を追加した独自指標での実験のほうが、僅かではあるが試合が楽しいという傾向にあった。また非安定性という本研究独自の評価方法であるが、脳波 (Attention) だけでの実験よりも独自指標での実験のほうが集中度合の測定の安定性が高かった。
- ⑤ プレイヤーから測定された生体信号より集中度合を評価し、関数近似によって当該プレイヤーにとって最適と予想される AI の強さを導き出すという手法が、本研究において提案する難易度調整の手法であった。しかし、この手法では従来研究[4]の「試

合が接戦であるほど楽しい」という主張に則った難易度調整の手法（接戦 AI）よりも楽しい試合を体験させることができなかった。

14-2 今後の課題

本セクションでは本研究を発展させるための課題やアイデアについて示す。

本研究の実験ではプレイヤーの集中度合の把握のために生体信号を測定してきた。しかし、生体信号の測定の精度の低さや再現性の低さは課題である。この課題を解決する方法の1つとして、各種測定機器及び測定方法をより高性能なものに変更することが考えられる。具体例としては以下の通りである：

- ① 脳波計は簡易脳波計（2つの電極を額と耳朶に装着する単極誘導を用いた測定機器）から、国際脳波学会が推奨する「10-20法」に基づいて、前頭部から後頭部まで等間隔で19個の電極を貼り、さらに両耳朶それぞれに電極貼り、合わせて21個の電極で測定を行うような脳波計に変えること。
- ② 心拍計はスマートウォッチ（手首の部分に緑の光を照射し、ヘモグロビンの動きで心拍を測定する機器）から、胸部に直接装着し、電気信号を電極で取得できる胸ベルト型の機器に変えること。
- ③ 瞬目回数の測定は JINS MEME（眉間の電極と鼻パッドの電極の電位差）から、カメラと映像解析を合わせた測定方法に変えること。

脳波（Attention）だけでの実験よりも、測定項目に心拍数と瞬目回数を追加した独自指標での実験のほうが、僅かではあるが試合が楽しいという傾向にあった。また非安定性という本研究独自の評価方法であるが、脳波（Attention）だけでの実験よりも集中度合の測定の安定性が向上した。このことから更に生体信号の測定項目を増やすことで、より多角的に集中度合及び楽しさを捉えられるようになり、より楽しい試合を体験させることができるのではないかと考えられる。また集中度合の測定の安定性がより向上することも期待できる。追加を検討する生体信号の具体例としては以下の通りである：

- ① 皮膚導電率反応（Skin Conductance Response : SCR）という生体信号がある。志水らの研究[8]では SCR を被験者の集中を検出する生体信号として使用した。SCR は人が動揺などを感じたときに遅れて手掌や足底に精神性の発汗が生じ、その結果として皮膚表面に起こる電気抵抗の変化を検出するものである。SCR 値が小さいときは集中状態にあり、大きいときは動揺、緊張及び非集中状態にあるとした。その上で数独を被験者に解いてもらったところ難しい問題ほど、解答に時間がかかり、無記入の時間が長くなり、SCR 値が小さい時間が減少し、無記入時の SCR 値が大きくなった。このことから、SCR を生体信号の測定項目に追加することで、格闘ゲームにおいてもプレイヤーの集中度合、動揺及び緊張などが把握できるのではないかと考えられる。

- ② 視線移動という生体信号がある。章らの研究[9]では被験者が「上の空」（非集中状態）にあるときの視線移動のパターンを抽出した。更にそのパターンが確認された被験者を集中状態に戻すためには、どのようなロボットの行動で被験者に働きかけるのが効果的かを調べた。被験者の注視点位置を測定しながら、講義映像を見せたところ、非集中状態特有の視線移動パターンとして3パターンが抽出された。このことから、格闘ゲームにおいても非集中状態のプレイヤー特有の視線移動パターンが存在することが期待できる。また、それを抽出できれば視線移動という生体信号からプレイヤーの集中度合を把握できるのではないかと考えられる。
- ③ 体動という生体信号がある。内田らの研究[10]では、被験者が勉強しているときの手首の移動（角速度、方位及び加速度）と脈波を測定した。測定データに対して集中しているかどうかを示すラベリングとして、勉強しているかどうか、勉強に関連する会話をしているかどうか、またはそれらが短時間で遷移した状態にあるという5種類のラベルを付けて、機械学習を行った。結果として機械学習による推定精度が80%を超え、脈波のみで学習したときよりも高い推定精度であった。このことから、格闘ゲームにおいてもプレイヤーの体動や脈波を測定し、機械学習することによって、プレイヤーの集中度合を体動や脈波から把握できるのではないかと考えられる。
- ④ 皮膚血流量という生体信号がある。麓の研究[11]では、被験者の脈拍数と皮膚血流量を測定しながら、作業課題や計算課題を行ってもらった。その結果として皮膚血流量は課題試行中が一番少なく、課題試行間の休憩中が次に少なく、安静中が一番多いという結果が得られた。また初回の課題試行中の皮膚血流量が一番少なく、試行回数が増えるにつれて課題試行中の皮膚血流量が多くなったという結果も得られた。さらに脈拍数の結果と皮膚血流量の結果を比較したときに、課題試行中、課題試行間の休憩中及び安静中それぞれの測定値の大小関係において皮膚血流量のほうが脈拍数よりも大小関係がハッキリと表れた。よって課題への注意集中は脈拍数よりも皮膚血流量のほうが、より忠実に反映されているとした。このことから、皮膚血流量を生体信号の測定項目に追加することで、格闘ゲームにおいてもプレイヤーの集中度合が把握できるのではないかと考えられる。

測定された生体信号より集中度合を評価し、関数近似によって最適と予想されるAIの強さを導き出すという手法が、本研究において提案する難易度調整の手法であった。しかし、この手法では従来研究[4]の「試合が接戦であるほど楽しい」という主張に則った難易度調整の手法（接戦AI）よりも楽しい試合を体験させることができなかった。この課題を解決する方法として、本研究で用いてきたAIよりも強いAIを用意することが考えられる。具体例としてはモンテカルロ木探索（MCTS）やDeep-Q-Network（DQN）を用いたAIが挙げられる。これらAIを難易度調整可能な形で用意できれば、本研究で用いたAIよりも強いAIで実験を行えると考えられる。

検証実験での集中度合の測定結果がガウス関数近似に沿うような結果を期待していたが、そのような結果は殆ど得られなかった。このことから AI の強さを変えたときに測定される集中度合の変化に対して、ガウス関数近似が適していなかった可能性が考えられた。この課題を解決する方法として、ガウス関数ではなく多項式で近似することや機械学習で予測をすることが考えられる。

謝辞

本研究を進めるにおいて、丁寧なご指導と助言をいただいた平原誠准教授と沢山の貴重なご意見をいただいた脳情報処理研究室の皆様、並びに被験者役を務めていただいた同期の方々に心より感謝いたします。

参考文献

- [1] Jenova Chen : “Flow in Games (and everything else)” COMMUNICATION OF THE ACM, Vol.50, No.4, pp.31-34, 2007.
- [2] Mihaly Csikszentmihalyi : “Flow : The Psychology of Optimal Experience” Harper Perennial, 1990.
- [3] 立命館大学 Intelligent Computer Entertainment Lab : “FightingGameAI Competition”, <https://www.ice.ci.ritsumeai.ac.jp/~ftgaic/index.htm>
- [4] 石原誠, 宮崎泰地, 原田智広, ターウォンマットラック : “対戦格闘ゲームにおけるゲーム AI や操作法の違いがプレイヤーの感じる面白さに与える影響の分析” 情報処理学会論文誌, Vol.57, No.11, pp.2414-2425, 2016.
- [5] 高津浩彰, 小関修 : “心拍変動を用いた講義の集中度の評価の試み” 豊田工業高等専門学校研究紀要, 第 39 号, pp.149-152, 2006.
- [6] 兜森仁志, 安彦智史, 長谷川大, 佐久田博司 : “web カメラを用いた瞬き検出による集中度評価” 情報処理学会第 77 回全国大会講演論文集, pp.931-932, 2015.
- [7] Kazuo Tubota, Katsu Nakamori : “Dry Eyes and Video Display Terminals” The New England Journal of Medicine, Vol.328, No.8, p.584, 1993.
- [8] 志水雅俊, 棟方渚, 松原仁 : “皮膚導電率反応による集中の持続とパフォーマンスの分析” 情報処理学会第 74 回全国大会講演論文集, pp.145-146, 2012.
- [9] 章毓焯, 田和辻可昌, 松居辰則 : “視線計測を用いた学習者の集中力の変化の検出と Robot による集中力回復の試み” 関東地区教育システム情報学会 2018 年度学生研究発表会論文集, pp.43-44, 2018.
- [10] 内田昂, 磯山直也, ロベズ・ギヨーム : “脈波及び体動情報を用いた集中状態推定精度の検証” マルチメディア, 分散強調とモバイルシンポジウム 2017 論文集, pp876-879, 2017
- [11] 麓 信義 : “課題遂行時の注意集中と脈拍数, 指尖皮膚血流量の変化” 心理学研究, Vol.48, No.5, pp.289-295, 1977.

付録

付録

A) 測定機器とコントローラー.....	67
B) 実験環境.....	71
C) 実験で使⽤した AI.....	72
C-1 第4章での仕様.....	72
C-2 第8章での仕様.....	74
D) 接戦 AI.....	74
E) ガウス関数近似.....	75
E-1 1次元に対する近似.....	75
E-2 2次元に対する近似.....	82
F) 法政⼤学⽣に向けて.....	98
付録参考文献.....	101

A) 測定機器とコントローラー

本セクションでは実験に際して使用した測定機器及びゲームコントローラーについて示す。

NeuroSky 社の脳波計 MindWaveMobile2 を図 A-1 に示す。この脳波計は頭部に装着し、電極を左耳朶と額に当てて使用する。使用時は特に額のほうの電極がしっかりと皮膚に接触していることに注意する。また使用時は専用ソフトウェア NeuroView と合わせて使用するが、接続がうまくいかない場合は PC と脳波計の Bluetooth ペアリングからやり直すことを推奨する。



図 A - 1 MindWaveMobile2

心拍数を測定するために使用した Fitbit 社のスマートウォッチ fitbit inspire2 を図 A-2 に示す。このスマートウォッチは左手首に装着して使用する。使用時は本体背面が皮膚に接触するように装着し、緩くなりすぎないように注意してバンドを締める。使用前にスマートフォンに専用アプリ Fitbit をインストールし、連携しておくこと。測定したデータは API を利用して取得できるので、本研究では CSV 形式で保存するプログラム A-1 を作成した。プログラム作成にあたって、Python で心拍データを取得する方法を説明しているサイト[A1]を参考にした。プログラム内の”ID”, ”SECRET”及び”token.txt”の内容は個人で異なるので注意する。これらの取得も同じサイト[A1]を参考に行うこと。データを取得する対象の日付を実験を行った日付に変更することに注意する。



図 A - 2 fitbit inspire2

```
#FitbitAPI

from ast import literal_eval
import fitbit
import pandas as pd
import csv

CLIENT_ID      = "XXXXXX"      #自身の ID
CLIENT_SECRET  = "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" #自身の SECRET
TOKEN_FILE     = "token.txt" #トークンを書き込んだテキストファイルを用意する

tokens = open(TOKEN_FILE).read()
token_dict = literal_eval(tokens)
access_token = token_dict['access_token']
refresh_token = token_dict['refresh_token']

def updateToken(token):      #トークン更新用関数
    f = open(TOKEN_FILE, 'w')
    f.write(str(token))
    f.close()
    return

#認証
```



```

client = fitbit.Fitbit(CLIENT_ID, CLIENT_SECRET,access_token =
access_token, refresh_token = refresh_token, refresh_cb = updateToken)

# 日付
DATE = "2023-11-24"
# 心拍数を取得(1 秒単位)
data_sec = client.intraday_time_series('activities/heart', DATE,
detail_level='1sec')
heart_sec = data_sec["activities-heart-intraday"]["dataset"]
heart_rat = pd.DataFrame.from_dict(heart_sec)
heart_row=heart_rat.values.tolist()

# CSV ファイルに書き込み、保存
with open (DATE+'_HeartBeat.csv', 'w', newline='') as csv_file:
    writer=csv.writer(csv_file)
    writer.writerows(heart_row)

print("complete")

```

プログラム A - 1 心拍数データを取得するプログラム

瞬目回数の測定機器として使用したジンス社の JINS MEME を図 A-3 に示す。この機器は一般的な眼鏡と同様に装着して使用する。使用時は鼻あてが皮膚に接触するように装着する。使用前にスマートフォンに専用アプリ JINS MEME をインストールし、連携しておくこと。また測定には専用アプリ JINS MEME Logger (有料) を購入してスマートフォンにインストールしておく必要がある。JINS MEME を装着した状態で、2つのスマートフォンアプリと接続して、JINS MEME Logger で測定する。実験では測定終了後に生成されるデータファイルの内、ファイル名に”currentData.csv”と付いているファイルを利用した。ファイル内の”blinkSpeed”という項目のデータを参照し、0 よりも大きい値が記録されていれば、そのタイミングで瞬きをしていたということである。よって、0 よりも大きい値の個数をカウントし、試合時間から 1 分あたりの瞬目回数を算出し、これを瞬目回数の記録とした。



図 A - 3 JINS MEME

ゲームの入力機器として使用したソニーインタラクティブエンタテインメント社の DualShock4 を図 A-4 に示す。このゲームコントローラーは PC と USB で有線接続して使用する。ゲームプレイ前に PC に free Colors 社の JoyToKey をインストールし、キーボード入力との対応を設定しておくこと。実験では、キーボードの十字キーをコントローラーの十字ボタン及び左スティックに対応させ、キーボードの Z キーはコントローラーの □ ボタンに対応させ、キーボードの X キーはコントローラーの × ボタンに対応させ、キーボードの C キーはコントローラーの R1 ボタンに対応させた。但し、コマンド入力を行いやすくするために、キーボードの上キーとコントローラーの左スティックの上方向は対応させず、キーボードの上キーはコントローラーの左スティックの上方向と R2 ボタンの同時入力に対応させた。コマンドとは、特定の行動をするために必要な複雑な入力のことであり、キャラクターを動かす移動キー入力（十字ボタンまたは左スティック）と攻撃キー入力（□ボタン、×ボタンまたは R1 ボタン）を組み合わせた入力である。具体例としては、自身のキャラクターが画面左側の地上にいて、相手のキャラクターが画面右側にいるときに、(↓↘→) という移動キー入力と □ ボタン入力の組み合わせによって「STAND_D_DF_FA」という行動を行える。



図 A - 4 DualShock4

B) 実験環境

本研究では格闘ゲームとして立命館大学知能エンターテインメント研究室製作の FightingICE を使用した。FightingICE のバージョンは 5.2 を利用して実験を行った。FightingICE の利用には、ゲームファイル(DareFightingICE)と JavaDevelopmentKit(JDK)が必要である。JDK はバージョン 17 以降をインストールして、パスを通しておくこと。ゲームファイル (DareFightingICE) のバージョン 5.2 を利用する場合は zip ファイルをダウンロード、展開し、"FightingICE.bat"という名前の bat ファイルを実行すればゲームが起動する。但し、本研究の実験では初期 HP を 300 に設定しているため、同じ条件で実験を行うには bat ファイル内の"--limithp 400 400"を"--limithp 300 300"に書き換える必要がある。また、ゲームの背景も同じ条件で実験を行うには、bat ファイル内の"--grey-bg"を削除する必要がある。最新バージョンの 6.1 でも問題なく同様に実験が行える (2023 年 12 月 30 日時点)。但し、同じ条件で実験を行うには"run-windows-amd64.bat"という bat ファイル内を上記のように書き換える必要がある。また、第 4 章で紹介した AI を利用するには、"javax.json-(version).jar"という jar ファイルが必要になるが、バージョン 6.1 の zip ファイルに同梱されていないので別バージョンなどからコピーする必要がある。このファイルは、JSON 形式でのデータの読み書きを行うために必要なライブラリである。図 B-1 のように、"javax.json-(version).jar"という jar ファイルを"lib"フォルダ内に置いておく必要がある。

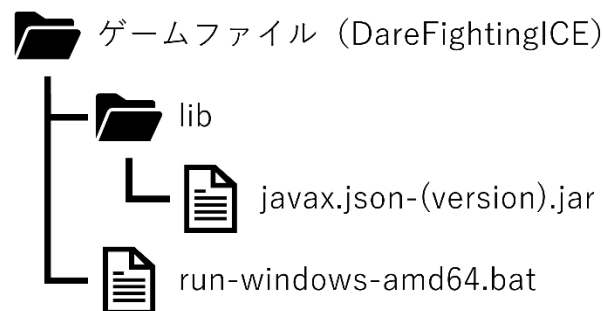


図 B-1 フォルダ構成図

第 13 章にて使用した接戦 AI は一定時間内で可能な限りシミュレーションを行うという仕様のため、PC の演算能力に依存してシミュレーション回数が増減すると考えられる。すなわち、PC の演算能力が弱いと接戦 AI は弱くなると考えられる。本研究で使用した PC のスペックを表 B-1 に示す。

表 B-1 PC のスペック

OS	Windows10
CPU	Intel Core i9-7900X
GPU	NVIDIA GeForce RTX 3060
メモリ	32GB

C) 実験で使用した AI

本セクションでは第 4 章で紹介した AI の仕様と第 8 章にて行われた仕様変更について詳細を示す。

C-1 第 4 章での仕様

本研究の実験に使用した AI は、FightingICE バージョン 5.2 ダウンロード時に付属している”SimpleAI”という AI がベースになっている。同じモデルの AI を 2 つ用意し、AI 同士を対戦させて、それぞれの AI で学習させる自己対戦形式で 500 試合 (1500 ラウンド) を戦わせ、AI にゲーム状況と行動の組み合わせに対する評価値 (R 値) を学習させた。また、R 値は学習中も学習後も 2 つの AI 間で一度も同期しない。行動をランダムで選択する確率を 100% から学習を始め、1 ラウンド終了につき 0.083% 下げながら 400 試合 (1200 ラウンド) を行った後、ランダム性がない状態で 100 試合 (300 ラウンド) を行って、学習終了とした。学習時に AI に操作させたキャラクターは”ZEN”と”GARNET”であり、”ZEN”を操作して学習した R 値を実験に使用した。

R 値は初期値 5 から 0~10 の範囲で変動し、R 値の更新はラウンド終了時に行われる。ラウンド m でのゲーム状況と行動の組み合わせを全て記録しておき、ラウンド終了時の HP 差から式(C-1-1)で更新値($value_m$) を計算する。そのラウンド m で記録されたゲーム状況と行動の組み合わせ全ての現在の R 値に対して更新値($value_m$) を加算する。ラウンド終了時に自身の HP が多く残っていて相手の HP が少なければ、更新値($value_m$) が大きくなるような式になっている。なお、分母の 120 はユーザが任意に決めるパラメータである。また、更新値($value_m$) は -1 を下限、1 を上限とした。

$$value_m = \frac{ApHP_m - AeHP_m}{120} \quad (C-1-1)$$

ここでの各種変数については表 C-1-1 の通りである。

表 C-1-1 式(C-1-1)の各種変数

$value_m$	ラウンド m での更新値
$ApHP_m$	ラウンド m 終了時の自身の HP
$AeHP_m$	ラウンド m 終了時の相手の HP

さらに各ラウンドの勝敗によっても R 値を増減させる。ラウンドに勝利した場合は 0.2、敗北した場合は -0.1 を、そのラウンド m で記録されたゲーム状況と行動の組み合わせ全ての現在の R 値に対して加算する。

AI はゲーム状況としてキャラクター間の距離を 6 種類で識別する。キャラクター間の距離はキャラクター間を直線で結んだときの線分の長さであり、y 軸も影響する。キャラクター間の距離 D (ピクセル) に対する 6 種類の分類は以下の通りである：① $D \leq 100$, ② $100 < D \leq 200$, ③ $200 < D \leq 400$, ④ $400 < D \leq 600$, ⑤ $600 < D \leq 800$, ⑥ $800 < D$ 。

AI はゲーム状況として相手のキャラクターの状態及び行動を以下の 56 種類で識別する：
 1.NEUTRAL, 2.STAND, 3.FORWARD_WALK, 4.DASH, 5.BACK_STEP, 6.CROUCH,
 7.JUMP, 8.FOR_JUMP, 9.BACK_JUMP, 10.AIR, 11.STAND_GUARD,
 12.CROUCH_GUARD, 13.AIR_GUARD, 14.STAND_GUARD_RECOV,
 15.CROUCH_GUARD_RECOV, 16.AIR_GUARD_RECOV, 17.STAND_RECOV,
 18.CROUCH_RECOV, 19.AIR_RECOV, 20.CHANGE_DOWN, 21.DOWN, 22.RISE,
 23.LANDING, 24.THROW_A, 25.THROW_B, 26.THROW_HIT, 27.THROW_SUFFER,
 28.STAND_A, 29.STAND_B, 30.CROUCH_A, 31.CROUCH_B, 32.AIR_A, 33.AIR_B,
 34.AIR_DA, 35.AIR_DB, 36.STAND_FA, 37.STAND_FB, 38.CROUCH_FA,
 39.CROUCH_FB, 40.AIR_FA, 41.AIR_FB, 42.AIR_UA, 43.AIR_UB, 44.STAND_D_DF_FA,
 45.STAND_D_DF_FB, 46.STAND_F_D_DFA, 47.STAND_F_D_DFB,

48.STAND_D_DB_BA, 49.STAND_D_DB_BB, 50.AIR_D_DF_FA, 51.AIR_D_DF_FB,
52.AIR_F_D_DFA, 53.AIR_F_D_DFB, 54.AIR_D_DB_BA, 55.AIR_D_DB_BB,
56.STAND_D_DF_FC.

AIの行動において、プログラム上では各行動の間に”key.empty()”という関数が実行されており、この関数は現在の行動命令をキャンセルする働きをし、キャラクターの状態が一度”NEUTRAL”になる。連続で行うことが重要な行動に対して”key.empty()”を実行してしまうと、意図しない別の行動や不自然な挙動になってしまう。例えば、AIが”STAND_GUARD”という行動を続けようとしたときに、行動間に”key.empty()”が実行されると、”BACK_STEP”という別の行動になってしまう。またAIが”DASH”という行動を続けようとしたときに、行動間に”key.empty()”が実行されると、一瞬走っては、一瞬立ち止まるということを繰り返す挙動になってしまう。よって、以下の5つの行動のうち、同じ行動が連続するときには、プログラム上で行動間に”key.empty()”という関数を実行しないようにした：1.STAND_GUARD, 2.CROUCH_GUARD, 3.FORWARD_WALK, 4.DASH, 5.AIR_GUARD.

C-2 第8章での仕様

セクション C-1 及び第4章での仕様からAIの行動を3種類(①STAND, ②CROUCH, ③AIR)追加したので、R値を学習しなおした。自己対戦形式で3000試合(9000ラウンド)を行ってR値の学習を行った。行動をランダムで選択する確率を100%から学習を始め、1ラウンド終了につき0.014%下げながら2400試合(7200ラウンド)を行った後、ランダム性がない状態で600試合(1800ラウンド)を行って、学習終了とした。AIが操作するキャラクターやR値の更新はセクション C-1と同様である。

AIの行動で同じ行動が連続するときには、プログラム上で上述の”key.empty()”関数を実行しないようにした。対象となる行動は1つ追加されて以下の6つである：1.STAND_GUARD, 2.CROUCH_GUARD, 3.FORWARD_WALK, 4.DASH, 5.AIR_GUARD, 6.CROUCH.

D) 接戦 AI

第13章の実験で使用した接戦AIは”MctsAI”というAIがベースになっている。”MctsAI”は”FightingGameAICompetition”というサイトの”Get started-----First Step” [D1]のページ下部からダウンロードできる。

接戦AIを用意する際に”MctsAI”から変更した点は以下の3点である：①1回ごとのシミュレーションの得点(score)を計算する関数”getScore()”を式(4)のように変更した、②AIの地上行動に”STAND”及び”CROUCH”を、空中行動に”AIR”を追加した、③AIの行動で同じ行動が連続するときには、プログラム上で上述の”key.empty()”関数を実行しないようにした。対象となる行動は以下の6つである：1.STAND_GUARD, 2.CROUCH_GUARD,

3.FORWARD_WALK, 4.DASH, 5.AIR_GUARD, 6.CROUCH.

接戦 AI を用意する際に“MctsAI”から変更しなかったが、ユーザが変更可能なパラメータを表 D-1 に示す.

表 D-1 接戦 AI のパラメータ

UCB_C	式(5)の正の定数 c 実験では $c = 3$ とした
UCT_TIME	AI が 1 回の行動決定するために シミュレーションを行える制限時間 実験では $165 \times 10^5 [ns]$ とした
UCT_TREE_DEPTH	AI がシミュレーションを行える ノードの深さ制限 実験では 2 とした
UCT_CREATE_NODE_THRESHOLD	AI がシミュレーションを行う際に 子ノードを生成するかどうかの 親ノードのシミュレーション回数の閾値 実験では 10 回とした
SIMULATION_TIME	シミュレーションを 1 回行う際の シミュレートする時間 実験では 60 フレームとした

E) ガウス関数近似

本セクションでは、実験の際に行ったガウス関数近似の具体的な方法やプログラムについて示す.

本研究の実験では、測定した集中度合（平均 Attention または独自指標）をガウス関数近似することで、集中度合が最大なるであろう条件（最適条件）を予測してきた.

ガウス関数近似は最急降下法とランダム多スタート探索法を用いて行った. ある程度ランダムな初期パラメータから、誤差の二乗和の更新時の変化量が既定値以下になるまで最急降下法で近似するというのを複数回行い、その内で誤差の二乗和が最も小さいガウス関数を近似式として採択するという方法を取った.

E-1 1次元に対する近似

第 6 章での実験では、正規乱数の標準偏差を固定値として正規乱数の平均のみを個々のプレイヤーに対して、最適な値に調整しようとしてきた. よって、第 6 章の実験では、集中度合を目的変数とし、正規乱数の平均を説明変数としてガウス関数近似を行った. 説明変数が 1 つであるときガウス関数は式(E-1-1)になる. 式(E-1-1)は条件 n の正規乱数の平均 x_n か

ら集中度合 $f(x_n)$ を求める式である。実際のプログラムでは、各種パラメータの収束する値におおよそのあたりをつけるために、集中度合を標準化した値に対して近似を行っている。

$$f(x_n) = a \exp\left(\frac{-(x_n - \mu)^2}{2\sigma^2}\right) + b \quad (E-1-1)$$

近似に際して、可変パラメータとそれぞれの更新式を表 E-1-1 に示す。条件 n における実際に測定された集中度合（教師データ）を t_n とし、学習率を $\varepsilon(> 0)$ とした。

表 E-1-1 ガウス関数のパラメータと更新式

パラメータ	更新式
a	$a - \varepsilon \sum_{n=1}^N (f(x_n) - t_n) \exp\left(\frac{-(x_n - \mu)^2}{2\sigma^2}\right)$
b	$b - \varepsilon \sum_{n=1}^N (f(x_n) - t_n)$
μ	$\mu - \varepsilon a \sum_{n=1}^N (f(x_n) - t_n) \exp\left(\frac{-(x_n - \mu)^2}{2\sigma^2}\right) \frac{x_n - \mu}{\sigma^2}$
σ	$\sigma - \varepsilon a \sum_{n=1}^N (f(x_n) - t_n) \exp\left(\frac{-(x_n - \mu)^2}{2\sigma^2}\right) \frac{(x_n - \mu)^2}{\sigma^3}$ <p>($\sigma > 0$という制限を設けなければいけないが、プログラム E-1-1 には、そういった制限を設けられていない。)</p>

各パラメータの値を更新し、教師データ t_n と集中度合 $f(x_n)$ の誤差二乗和の計算に移る。誤差二乗和が各パラメータ更新前よりも大きくなってしまった場合、学習率 ε を 0.1 倍して各パラメータ更新からやり直す。誤差二乗和の減少量が 10 万分の 1 以下であった場合、そこで近似を終了とする。誤差二乗和の減少量が 10 万分の 1 より大きい場合、各パラメータの更新に移る。このようにパラメータ更新と誤差二乗和の評価を繰り返す。各パラメータの初期値について表 E-1-2 に示す。パラメータの初期値の決定に一様乱数を用いることで、ガウス関数近似のパラメータが異なる値に収束する可能性を生み出している。また複数回の試行を行うことで探索をある程度の範囲から行える。実際のプログラムでは 50 回の試行から誤差二乗和が最も小さいガウス関数を採択した。実際のプログラムをプログラム E-1-1 に示す。

表 E-1-2 ガウス関数のパラメータと初期値

パラメータ	初期値
a	+1~+3 の一様乱数の値
b	-1~0 の一様乱数の値
μ	教師データの最大値を記録した正規乱数の平均に -1~+1 の一様乱数を足した値
σ	+0.5~+1.5 の一様乱数の値

#1 次近似

```

import csv
import datetime
import math
import random
import matplotlib.pyplot as plt
import numpy as np

diff_mean=[3,5,7,9]      #正規乱数の平均

Attention_ave=[58.8,67.546875,64.05228758,53.76978417]  #集中度合
ave_ave=0      #集中度合の平均
ave_std=0      #集中度合の標準偏差
seido=0.00001  #誤差二乗和更新料 10 万分の 1 以下
g_min=0      #プロット領域最小
g_max=10     #プロット領域最大
g_m=(g_max-g_min)*10+1      #プロット量

for i in Attention_ave:
    ave_ave+=i
ave_ave/=len(Attention_ave)      #集中度合平均計算

for i in Attention_ave:
    ave_std+=(i-ave_ave)**2
ave_std=math.sqrt(ave_std/(len(Attention_ave)-1))      #集中度合標準偏差計算

```

```

Attention_ave_standarded=[]
for i in Attention_ave:
    Attention_ave_standarded.append((i-ave_ave)/ave_std)    #集中度合標準化

class Variables:          #ガウス関数のパラメータクラス
    def __init__(self):
        self.u=0.0
        self.o=0.0
        self.a=0.0
        self.b=0.0
        self.E=100000.0
        self.loop=0
        self.epsilon=0.01
        self.maximum=0.0

def kinji():
    u=Attention_ave_standarded.index(max(Attention_ave_standarded))
    u=diff_mean[u] + (random.random()*2.0-1)          #ガウス関数の中央,
max+(-1~+1)
    b=random.random()*-1.0          #ガウス関数の定数項, (-1.0~0)

    a=2.0 + (random.random()*2.0-1)          #係数, (1~3)
    o=1.0 + (random.random()-0.5)          #裾の広さ, (0.5~1.5)

    epsilon=0.01          #学習率
    E=0          #誤差二乗和
    dEdu=0          #微分
    dEdo=0
    dEda=0
    dEdb=0

    list=diff_mean #正規乱数の平均

    for i in list:
        _y=a*math.exp(-((i-u)**2)/(2*o**2))+b

```

```

t=Attention_ave_standarded[list.index(i)]
E+=(_y-t)**2
dEdu+=a*( _y-t)*math.exp(-((i-u)**2)/(2*o**2))*(i-u)/(o**2)
dEdo+=a*( _y-t)*math.exp(-((i-u)**2)/(2*o**2))*((i-u)**2)/(o**3)
dEda+=(_y-t)*math.exp(-((i-u)**2)/(2*o**2))
dEdb+=_y-t

u_new=u-dEdu*epsilon          #パラメータ更新
o_new=o-dEdo*epsilon          #0 以下にならない制限を設けるべき箇所
a_new=a-dEda*epsilon
b_new=b-dEdb*epsilon
E_new=0
dEdu=0
dEdo=0
dEda=0
dEdb=0

bl=True
loop=1
while bl :
    loop+=1
    for i in list:
        _y=a_new*math.exp(-((i-u_new)**2)/(2*o_new**2))+b_new
        t=Attention_ave_standarded[list.index(i)]
        E_new+=(_y-t)**2
        dEdu+=a_new*( _y-t)*math.exp(-((i-u_new)**2)/(2*o_new**2))*(i-
u_new)/(o_new**2)
        dEdo+=a_new*( _y-t)*math.exp(-((i-u_new)**2)/(2*o_new**2))*((i-
u_new)**2)/(o_new**3)
        dEda+=(_y-t)*math.exp(-((i-u_new)**2)/(2*o_new**2))
        dEdb+=_y-t

    if(E<E_new):                #誤差の二乗和が大きくなったら学習率 0.1 倍
        epsilon=epsilon*0.1
        u_new=u
        o_new=o

```

```

        a_new=a
        b_new=b
    else:
        if(E-E_new<=seido):      #誤差二乗和更新料 10 万分の 1 以下で停止
            bl=False
        else:
            u=u_new
            o=o_new
            a=a_new
            b=b_new
            E=E_new
            u_new=u-dEdu*epsilon
            o_new=o-dEdo*epsilon #0 以下にならない制限を設けるべき箇所
            a_new=a-dEda*epsilon
            b_new=b-dEdb*epsilon
            E_new=0
            dEdu=0
            dEdo=0
            dEda=0
            dEdb=0

    var=Variables()      #クラスでリターン
    var.u=u_new
    var.o=o_new
    var.a=a_new
    var.b=b_new
    var.E=E_new
    var.loop=loop
    var.epsilon=epsilon
    var.maximum=(a_new*math.exp(-
((0)**2)/(2*o_new**2))+b_new)*ave_std+ave_ave
    return var

best=Variables()
for j in range(50):      #50 回試行
    Avar=kinji()

```

```

    if (best.E>Avar.E) and (Avar.a>0):      #誤差の二乗和がより小さい方を採択,
and, 谷型は取らない
        best=Avar

print("u=",best.u,",  o=",best.o,",
a=",best.a,",  b=",best.b,",  E=",best.E)    #結果とプロット
print("loop=",best.loop,",  epsilon=",best.epsilon)
x = np.linspace(g_min,g_max,g_m)
y = (best.a*np.exp(-(x-
best.u)**2)/(2*best.o**2))+best.b)*ave_std+ave_ave
plt.plot(x,y,linestyle = "dashed")          #ガウス関数のプロット

#csv
date=datetime.datetime.now()
row=[best.u,best.o,best.a,best.b,best.E,best.maximum]
with open("ResultCSV_"+date.strftime("%m_%d_%H_%M")+".csv","w") as f:
    writer=csv.writer(f)
    writer.writerow(["u","o","a","b","E","max"])
    writer.writerow(row)
    writer.writerow([])
    writer.writerow(["正規乱数の平均","集中度合"])
    row=[]
    for j in range(g_m):
        row.append([x[j],y[j]])
    writer.writerows(row)

y = np.array(Attention_ave)
x = np.array(diff_mean)
plt.plot(x,y,color="orange")                #教師データ(実測値)
x=best.u
y=(best.a*np.exp(-(x-best.u)**2)/(2*best.o**2))+best.b)*ave_std+ave_ave
print("max=",y)
plt.plot(x,y,color="red",marker="o")        #最大プロット
plt.show()

```

プログラム E-1-1 1次元でのガウス関数近似プログラム

このプログラムを使用する際は、正規乱数の平均”diff_mean”とそれに対応する集中度合”Attention_ave”を書き換えて実行すること。実行後は図 E-1-1 のようなグラフが出力される。青い点線が近似したガウス関数であり、橙色の折れ線グラフが教師データであり、赤い点が近似したガウス関数における最大である。縦軸が集中度合であり、横軸が正規乱数の平均である。

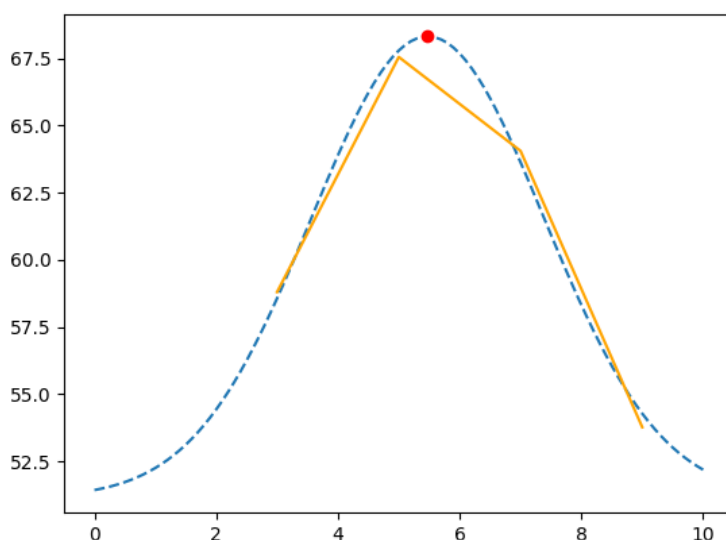


図 E-1-1 ガウス関数近似のグラフ

またプログラム実行後はガウス関数の各種パラメータと、正規乱数の平均を 0~10 まで 0.1 刻みで変化させたときの集中度合の予測値をまとめた csv ファイルが作成される。

E-2 2次元に対する近似

本セクションでは2次元に対するガウス関数近似の詳細を示すが、実験(第7章, 第9章及び第11章)を行った段階ではガウス関数の式を間違えていたことが判明した。この誤りに付随して各種パラメータの更新式, 及びプログラム内の計算も間違ったものであった。またプログラムでの各種パラメータの変化において、パラメータが取りうる値の範囲に制限を設けていなかった。下記に示す数式, 各種パラメータの更新式及びプログラムは修正済みのものである。誤りの具体的な内容, 誤りを修正したことによる集中度合が最大になると考えられる条件(最適条件)の変化及び近似式と教師データの誤差二乗和の変化を本セクションの最後に示す。

第7章, 第9章及び第11章での実験では、正規乱数の平均と標準偏差を個々のプレイヤーに対して、最適な値に調整しようとしてきた。よって、第7章, 第9章及び第11章の実

験では、集中度合を目的変数とし、正規乱数の平均と標準偏差を説明変数としてガウス関数近似を行った。説明変数が2つであるときガウス関数は式(E-2-1)になる。式(E-2-1)は条件 n の正規乱数の平均 x_n 及び標準偏差 y_n から集中度合 $g(x_n, y_n)$ を求める式である。実際のプログラムでは、各種パラメータの収束する値におおよそのあたりをつけるために、集中度合を標準化した値に対して近似を行っている。

$$g(x_n, y_n) = a \exp\left(-\frac{\sigma_y^2(x_n - \mu_x)^2 + 2(-\rho\sigma_x\sigma_y)(x_n - \mu_x)(y_n - \mu_y) + \sigma_x^2(y_n - \mu_y)^2}{2\sigma_x^2\sigma_y^2(1 - \rho^2)}\right) + b \quad (E-2-1)$$

近似に際して、可変パラメータとそれぞれの更新式を表 E-2-1 に示す。条件 n における実際に測定された集中度合（教師データ）を t_n とし、学習率を $\varepsilon(> 0)$ とした。

表 E-2-1 ガウス関数のパラメータと更新式

パラメータ	更新式
a	$a - \varepsilon \sum_{n=1}^N (g(x_n, y_n) - t_n) \times \exp\left(-\frac{\sigma_y^2(x_n - \mu_x)^2 + 2(-\rho\sigma_x\sigma_y)(x_n - \mu_x)(y_n - \mu_y) + \sigma_x^2(y_n - \mu_y)^2}{2\sigma_x^2\sigma_y^2(1 - \rho^2)}\right)$
b	$b - \varepsilon \sum_{n=1}^N (g(x_n, y_n) - t_n)$

μ_x	$\mu_x - \varepsilon a \sum_{n=1}^N (g(x_n, y_n) - t_n)$ $\times \exp\left(-\frac{\sigma_y^2(x_n - \mu_x)^2 + 2(-\rho\sigma_x\sigma_y)(x_n - \mu_x)(y_n - \mu_y) + \sigma_x^2(y_n - \mu_y)^2}{2\sigma_x^2\sigma_y^2(1 - \rho^2)}\right)$ $\times \frac{2\sigma_y^2(x_n - \mu_x) + 2(-\rho\sigma_x\sigma_y)(y_n - \mu_y)}{2\sigma_x^2\sigma_y^2(1 - \rho^2)}$
μ_y	$\mu_y - \varepsilon a \sum_{n=1}^N (g(x_n, y_n) - t_n)$ $\times \exp\left(-\frac{\sigma_y^2(x_n - \mu_x)^2 + 2(-\rho\sigma_x\sigma_y)(x_n - \mu_x)(y_n - \mu_y) + \sigma_x^2(y_n - \mu_y)^2}{2\sigma_x^2\sigma_y^2(1 - \rho^2)}\right)$ $\times \frac{2\sigma_x^2(y_n - \mu_y) + 2(-\rho\sigma_x\sigma_y)(x_n - \mu_x)}{2\sigma_x^2\sigma_y^2(1 - \rho^2)}$
σ_x	$\sigma_x - \varepsilon a \sum_{n=1}^N (g(x_n, y_n) - t_n)$ $\times \exp\left(-\frac{\sigma_y^2(x_n - \mu_x)^2 + 2(-\rho\sigma_x\sigma_y)(x_n - \mu_x)(y_n - \mu_y) + \sigma_x^2(y_n - \mu_y)^2}{2\sigma_x^2\sigma_y^2(1 - \rho^2)}\right)$ $\times (-1) \left(\frac{\left(2(-\rho\sigma_y)(x_n - \mu_x)(y_n - \mu_y) + 2\sigma_x(y_n - \mu_y)^2\right) \times 2\sigma_x^2\sigma_y^2(1 - \rho^2)}{(2\sigma_x^2\sigma_y^2(1 - \rho^2))^2} \right.$ $\left. - \frac{\sigma_y^2(x_n - \mu_x)^2 + 2(-\rho\sigma_x\sigma_y)(x_n - \mu_x)(y_n - \mu_y) + \sigma_x^2(y_n - \mu_y)^2}{(2\sigma_x^2\sigma_y^2(1 - \rho^2))^2} \right)$ $\times 4\sigma_x\sigma_y^2(1 - \rho^2)$ <p style="text-align: center;">$(\sigma_x > 0)$</p>

σ_y	$\sigma_y - \varepsilon a \sum_{n=1}^N (g(x_n, y_n) - t_n)$ $\times \exp\left(-\frac{\sigma_y^2(x_n - \mu_x)^2 + 2(-\rho\sigma_x\sigma_y)(x_n - \mu_x)(y_n - \mu_y) + \sigma_x^2(y_n - \mu_y)^2}{2\sigma_x^2\sigma_y^2(1 - \rho^2)}\right)$ $\times (-1) \left(\frac{(2(-\rho\sigma_x)(x_n - \mu_x)(y_n - \mu_y) + 2\sigma_y(x_n - \mu_x)^2) \times 2\sigma_x^2\sigma_y^2(1 - \rho^2)}{(2\sigma_x^2\sigma_y^2(1 - \rho^2))^2} \right.$ $\left. - \frac{\sigma_y^2(x_n - \mu_x)^2 + 2(-\rho\sigma_x\sigma_y)(x_n - \mu_x)(y_n - \mu_y) + \sigma_x^2(y_n - \mu_y)^2}{(2\sigma_x^2\sigma_y^2(1 - \rho^2))^2} \right)$ $\times 4\sigma_y\sigma_x^2(1 - \rho^2)$ <p style="text-align: center;">$(\sigma_y > 0)$</p>
ρ	$\rho - \varepsilon a \sum_{n=1}^N (g(x_n, y_n) - t_n)$ $\times \exp\left(-\frac{\sigma_y^2(x_n - \mu_x)^2 + 2(-\rho\sigma_x\sigma_y)(x_n - \mu_x)(y_n - \mu_y) + \sigma_x^2(y_n - \mu_y)^2}{2\sigma_x^2\sigma_y^2(1 - \rho^2)}\right)$ $\times (-1) \left(\frac{2(-\sigma_x\sigma_y)(x_n - \mu_x)(y_n - \mu_y) \times 2\sigma_x^2\sigma_y^2(1 - \rho^2)}{(2\sigma_x^2\sigma_y^2(1 - \rho^2))^2} \right.$ $\left. - \frac{\sigma_y^2(x_n - \mu_x)^2 + 2(-\rho\sigma_x\sigma_y)(x_n - \mu_x)(y_n - \mu_y) + \sigma_x^2(y_n - \mu_y)^2}{(2\sigma_x^2\sigma_y^2(1 - \rho^2))^2} \right)$ $\times (-4\sigma_x^2\sigma_y^2\rho)$ <p style="text-align: center;">$(-1 < \rho < 1)$</p>

各パラメータの値を更新して、教師データ t_n と集中度合 $f(x_n)$ の誤差二乗和の計算に移る。誤差二乗和が各パラメータ更新前よりも大きくなってしまった場合、学習率 ε を0.1倍して各パラメータ更新からやり直す。誤差二乗和の減少量が10万分の1以下であった場合、そ

ここで近似を終了とする。誤差二乗和の減少量が 10 万分の 1 より大きい場合、各パラメータの更新に移る。このようにパラメータ更新と誤差二乗和の評価を繰り返す。各パラメータの初期値について表 E-2-2 に示す。パラメータの初期値の決定に一様乱数を用いることで、ガウス関数近似のパラメータが異なる値に収束する可能性を生み出している。また複数回の試行を行うことで探索をある程度の範囲から行える。実際のプログラムでは 300 回の試行から誤差二乗和が最も小さいガウス関数を採択した。実際のプログラムをプログラム E-2-1 に示す。

表 E-2-2 ガウス関数のパラメータと初期値

パラメータ	初期値
a	+1~+3 の一様乱数の値
b	-1~0 の一様乱数の値
μ_x	教師データの最大値を記録した正規乱数の平均に -1~+1 の一様乱数を足した値
μ_y	教師データの最大値を記録した正規乱数の標準偏差に -0.5~+0.5 の一様乱数を足した値
σ_x	+1~+5 の一様乱数の値
σ_y	+0.5~+2.5 の一様乱数の値
ρ	-0.25~+0.25 の一様乱数の値

#2 次近似

```
import csv
import math
import random
import matplotlib.pyplot as plt
import numpy as np
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.animation as animation
import datetime

class Variables:          #ガウス関数のパラメータクラス
    def __init__(self):
        self.ux=0.0
        self.uy=0.0
        self.ox=0.0
```

```

self.oy=0.0
self.a=0.0
self.b=0.0
self.p=0.0
self.E=100000.0
self.loop=0
self.epsilon=0.01
self.maximum=0.0

Zlabel = "concentration indicator"    #"concentration
indicator"    or #"Attention"
diff_mean=[2, 5, 8]                #正規乱数の平均
diff_stddev=[0.5, 1, 2]            #正規乱数の標準偏差
Attention_ave=[[-0.239, -0.266, 0.400],
               [0.184, 0.081, 0.567],
               [1.160, -1.625, -0.262]] #集中度合、配置 ave[mean][stdev]
ave_ave=0                #集中度合の平均
ave_std=0                #集中度合の標準偏差
seido=0.00001           #誤差二乗和更新料 10 万分の 1 以下

element_count=0         #要素数
for i in Attention_ave:
    for j in i:
        ave_ave+=j
        element_count+=1
ave_ave/=element_count   #平均計算

for i in Attention_ave:
    for j in i:
        ave_std+=(j-ave_ave)**2
ave_std=math.sqrt(ave_std/(element_count-1))    #標準偏差計算

Attention_ave_standarded=[]
for i in Attention_ave:
    liii=[]
    for j in i:

```

```

        liii.append((j-ave_ave)/ave_std)
    Attention_ave_standarded.append(liii)          #集中度合の標準化,
Attention_ave_standarded[mean][stdev]

flat=np.array(Attention_ave_standarded)
flat=flat.flatten()
qr=np.argmax(flat)

def kinji():
    ux,uy=divmod(qr,len(diff_stdev))
    ux=diff_mean[ux] + (random.random()*2.0-1.0)    #ガウス関数の x 軸中
    央, max+(-1~+1)
    uy=diff_stdev[uy] + (random.random()-0.5)     #ガウス関数の y 軸中
    央, max+(-0.5~+0.5)
    b=random.random()*(-1.0)                      #定数項、(-1~0)
    a=random.random()*2.0+1.0                    #係数、(1~3)
    ox=random.random()*4.0+1.0                  #x 軸裾の広さ、(1~5)
    oy=random.random()*2.0+0.5                  #y 軸裾の広さ、(0.5~2.5)
    p=(random.random()-0.5)/2.0                 #x と y の相関係数、(-0.25~0.25)

    epsilon=0.01                                #学習率
    E=0
    E_old=0
    dEdux=0
    dEduy=0
    dEdox=0
    dEdoy=0
    dEda=0
    dEdb=0
    dEdp=0
    dEdux_old=0
    dEduy_old=0
    dEdox_old=0
    dEdoy_old=0
    dEdp_old=0
    dEda_old=0

```

```

dEdb_old=0
ux_old=0
uy_old=0
ox_old=0
oy_old=0
p_old=0
a_old=0
b_old=0

bl=True
loop=0
while bl:
    loop+=1
    for x in diff_mean:      #x は正規乱数の平均
        x_index=diff_mean.index(x)
        for y in diff_stdev:      #y は正規乱数の標準偏差
            y_index=diff_stdev.index(y)
            k=(-p*ox*oy)
            l=(x-ux)*(y-uy)
            si=(oy**2)*((x-ux)**2)+2*k*l+(ox**2)*((y-uy)**2)
            bo=2*(ox**2)*(oy**2)*(1-(p**2))
            EXP=math.exp(-si/bo)

            z=a*EXP+b          #z は集中度合の計算
            t=Attention_ave_standarded[x_index][y_index]      #t は教
師データ

            m=(z-t)*a*EXP
            E+=(z-t)**2
            dEdux+=m*(-2*(oy**2)*(x-ux)-2*k*(y-uy))/-bo
            dEduy+=m*(-2*(ox**2)*(y-uy)-2*k*(x-ux))/-bo
            dEdox+=(-m)*(((2*(-p*oy)*1+2*ox*((y-uy)**2))*bo)-
(si*4*ox*(oy**2)*(1-(p**2))))/(bo**2)
            dEdoy+=(-m)*(((2*(-p*ox)*1+2*oy*((x-ux)**2))*bo)-
(si*4*(ox**2)*oy*(1-(p**2))))/(bo**2)

```

```

        dEdp+=(-m)*((2*(-ox*oy)*1*bo)-(si*(-
4)*(ox**2)*(oy**2)*p))/(bo**2)
        dEda+=(z-t)*EXP
        dEdb+=z-t

    if(E_old<E) and (not loop==1):      #2 巡目以降で誤差の二乗和が大きくな
    ったとき
        epsilon=epsilon*0.1
        ux=ux_old-dEdux_old*epsilon
        uy=uy_old-dEduy_old*epsilon
        ox=ox_old-dEdox_old*epsilon
        if(ox<=0):
            ox=0+(0.1**10)      #100 億分の 1
        oy=oy_old-dEdoy_old*epsilon
        if(oy<=0):
            oy=0+(0.1**10)
        p=p_old-dEdp_old*epsilon
        if(p>=1):
            p=1-(0.1**10)
        elif(p<=-1):
            p=-1+(0.1**10)
        a=a_old-dEda_old*epsilon
        b=b_old-dEdb_old*epsilon

        dEdux=0
        dEduy=0
        dEdox=0
        dEdoy=0
        dEdp=0
        dEda=0
        dEdb=0
        E=0
    else:
        if(E_old-E<=seido) and (not loop==1):      #誤差二乗和更新料 10 万
    分の 1 以下で停止
            bl=False

```

```

else:
    ux_old=ux
    uy_old=uy
    ox_old=ox
    oy_old=oy
    p_old=p
    a_old=a
    b_old=b
    dEdux_old=dEdux
    dEduy_old=dEduy
    dEdox_old=dEdox
    dEdoy_old=dEdoy
    dEdp_old=dEdp
    dEda_old=dEda
    dEdb_old=dEdb

    ux=ux-dEdux*epsilon
    uy=uy-dEduy*epsilon
    ox=ox-dEdox*epsilon
    if(ox<=0):
        ox=0+(0.1**10)
    oy=oy-dEdoy*epsilon
    if(oy<=0):
        oy=0+(0.1**10)
    p=p-dEdp*epsilon
    if(p>=1):
        p=1-(0.1**10)
    elif(p<=-1):
        p=-1+(0.1**10)
    a=a-dEda*epsilon
    b=b-dEdb*epsilon
    dEdux=0
    dEduy=0
    dEdox=0
    dEdoy=0
    dEdp=0

```

```

        dEda=0
        dEdb=0
        E_old=E
        E=0

    Vari=Variables()
    Vari.ux=ux
    Vari.uy=uy
    Vari.ox=ox
    Vari.oy=oy
    Vari.a=a
    Vari.b=b
    Vari.p=p
    Vari.E=E
    Vari.loop=loop
    Vari.epsilon=epsilon
    Vari.maximum=(a*np.exp(-((oy**2)*((ux-ux)**2)+2*(-p*ox*oy)*(ux-
ux)*(uy-uy)+(ox**2)*((uy-uy)**2)))/(2*(ox**2)*(oy**2)*(1-
(p**2))))+b)*ave_std+ave_ave

    return Vari

g_xmin=0
g_xmax=10
g_x_sample=(g_xmax-g_xmin)*4+1      #for plot
g_ymin=0
g_ymax=3
g_y_sample=(g_ymax-g_ymin)*10+1     #for plot

best=Variables()
for i in range(300):                #300 試行
    aVari=kinji()
    if (best.E>aVari.E) and (aVari.a>0) and (1>aVari.p) and (aVari.p>-
1):
        #誤差の二乗和がより小さい方へ、係数がマイナスで谷型は取らない、相関係数は-
1~+1 でなければならない
        best=aVari

```



```

print("ux=",best.ux,",   uy=",best.uy,",   ox=",best.ox,",   oy=",best.oy,",
      a=",best.a,",   b=",best.b,",   p=",best.p,",   E=",best.E)
print("loop=",best.loop,",   epsilon=",best.epsilon,",   max=",best.maximu
m)

#csv
date=datetime.datetime.now()
row=[best.ux,best.uy,best.ox,best.oy,best.a,best.b,best.p,best.E,best.max
imum]
with open("ResultCSV_"+date.strftime("%m_%d_%H_%M")+".csv","w") as f:
    writer=csv.writer(f)
    writer.writerow(["ux","uy","ox","oy","a","b","p","E","max"])
    writer.writerow(row)
    writer.writerow([])
    mean=0
    rows=[]
    row=[0]
    for i in range(31):           #平均は 0~10 まで 0.2 刻み、標準偏差は 0~3 まで
0.1 刻み
        row.append(0+i*0.1)
        rows.append(row)
        for i in range(51):
            std=0
            row=[mean]
            for j in range(31):
                exp=math.exp(-((best.oy**2)*((mean-best.ux)**2)+2*(-
best.p*best.ox*best.oy)*(mean-best.ux)*(std-best.uy)+(best.ox**2)*((std-
best.uy)**2))/(2*(best.ox**2)*(best.oy**2)*(1-(best.p**2))))
                ZZ=(best.a*exp+best.b)*ave_std+ave_ave
                row.append(ZZ)
                std+=0.1
            rows.append(row)
            mean+=0.2
    writer.writerows(rows)
    f.close()

```

```

x = np.linspace(g_xmin,g_xmax,g_x_sample)          #プロット
y = np.linspace(g_ymin,g_ymax,g_y_sample)

for i in range(len(x)):
    if i==0:
        z = np.array([(best.a*np.exp(-((best.oy**2)*((x[i]-
best.ux)**2))+2*(-best.p*best.ox*best.oy)*(x[i]-best.ux)*(y-
best.uy)+(best.ox**2)*((y-best.uy)**2)))/(2*(best.ox**2)*(best.oy**2)*(1-
(best.p**2)))))+best.b)*ave_std+ave_ave])
    else:
        z_ = np.array([(best.a*np.exp(-((best.oy**2)*((x[i]-
best.ux)**2))+2*(-best.p*best.ox*best.oy)*(x[i]-best.ux)*(y-
best.uy)+(best.ox**2)*((y-best.uy)**2)))/(2*(best.ox**2)*(best.oy**2)*(1-
(best.p**2)))))+best.b)*ave_std+ave_ave])
        z=np.append(z,z_,axis=0)

yy_,xx_=np.meshgrid(y,x)

fig=plt.figure(figsize=(8,8))
ax=fig.add_subplot(111, projection="3d")
ax.set_xlabel("mean", size=16)
ax.set_ylabel("stdev", size=16)
ax.set_zlabel("Zlabel", size=16)

if z.min(<np.min(Attention_ave):          #等高線をつける底面をきめる
    mini=z.min()-z.min()/50
else:
    mini=np.min(Attention_ave)-np.min(Attention_ave)/50

t_ = np.array(Attention_ave)
x = np.array(diff_mean)
y = np.array(diff_stdev)
yy,xx=np.meshgrid(y,x)          #教師データのプロット

x=best.ux

```

```

y=best.uy
z_max = best.maximum          #最大値

def init():                  #アニメーション用関数、プロットする
    ax.set_xlabel("mean", size=16)
    ax.set_ylabel("stdev", size=16)
    ax.set_zlabel("Zlabel", size=16)
    ax.plot_surface(xx_,yy_,z,cmap="summer")
    ax.contour(xx_,yy_,z,colors="black", offset=mini)
    ax.scatter(xx,yy,t_,s=40,c="blue")
    ax.scatter(x,y,z_max,s=40,c="red")
    return fig,

def animate(i):             #画角をまわす
    ax.view_init(elev=30, azim=3.6*i)
    return fig,

ani=animation.FuncAnimation(fig, animate, init_func=init,
frames=range(100), interval=100, blit=True, repeat=False)    #アニメーション
作成

name="rotate_3d_"+date.strftime("%m_%d_%H_%M")+".mp4"
ani.save(name, writer="ffmpeg",fps=8, dpi=100)                #アニメーション保存

```

プログラム E-2-1 2次元でのガウス関数近似プログラム

このプログラムを使用する際は、正規乱数の平均"diff_mean", 正規乱数の標準偏差"diff_std"及びそれに対応する集中度合"Attention_ave"を書き換えて実行すること。プログラム実行後はガウス関数の各種パラメータと、正規乱数の平均を0~10まで0.2刻み、正規乱数の標準偏差を0~3まで0.1刻みで変化させたときの集中度合の予測値をまとめたcsvファイルが作成される。また図 E-2-1 のように近似したガウス関数のグラフ、近似したガウス関数における最大値のプロット及び教師データのプロットの3Dグラフの回転アニメーションが作成される。緑の面が近似したガウス関数のグラフであり、赤いプロットが近似したガウス関数における最大値であり、青いプロットが教師データである。アニメーションの保存に"ffmpeg"というライブラリが必要になる。

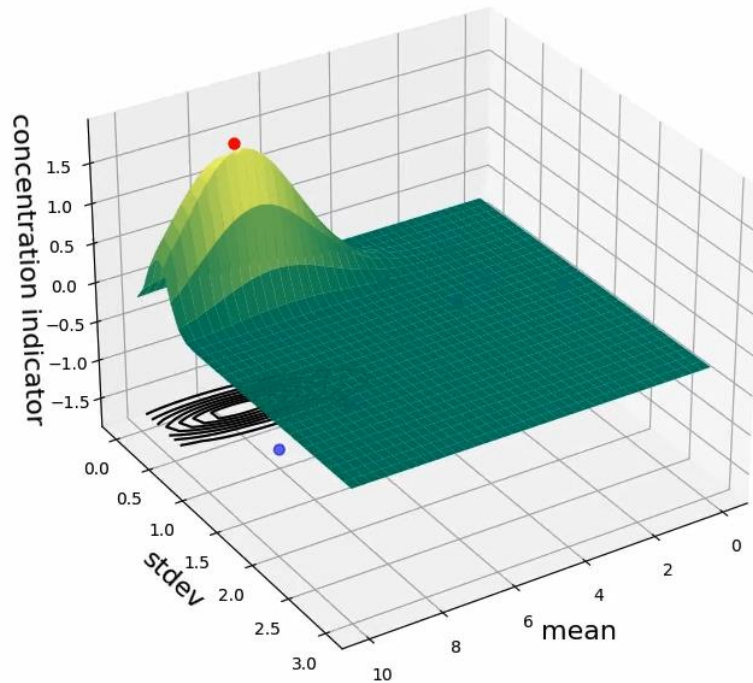


図 E-2- 1 3D 回転アニメーションの静止画

”ffmpeg”を導入するためには, ”ImageMagick”というソフトウェアをインストールするという方法が簡単である. ”ImageMagick”の公式サイト^[E1]のダウンロードページから各種OSに対応するインストーラーをダウンロードする. 恐らく, どのバージョンでも問題ないと考えられるが, 一番上のバージョンをダウンロードしておくのが無難である. インストールする際に”Install FFmpeg”という項目があるのでチェックボックスを確認してからインストールして, パスを通しておくこと.

ここからは実験を行った段階で間違えていた部分について示す. 実験を行った段階では, ガウス関数を式(E-2-2)のように間違えていた.

$$g(x_n, y_n) = a \exp \left(- \frac{\sigma_y^2 (x_n - \mu_x)^2 + 2(-\rho \sigma_x \sigma_y) (x_n - \mu_x) (y_n - \mu_y) + \sigma_x^2 (y_n - \mu_y)^2}{2\sigma_x^2 \sigma_y^2 (1 - \rho)^2} \right) + b \quad (E-2-2)$$

式(E-2-1)と式(E-2-2)の違いはexp ()内部の分母で、 $(1 - \rho^2)$ が正しいところを $(1 - \rho)^2$ としていた点である。各種パラメータの更新式においても同様に $(1 - \rho^2)$ が正しいところを $(1 - \rho)^2$ と間違えていた。但し、 ρ の更新式においては先ほどの誤りに加えて更新式の最後の部分で $(-4\sigma_x^2 \sigma_y^2 \rho)$ が正しいところを $(-4\sigma_x^2 \sigma_y^2 (1 - \rho))$ と間違えていた。プログラム内の計算も同様に間違えていた。またプログラムでの各種パラメータについて、取りうる値の制限として $(\sigma_x > 0, \sigma_y > 0, -1 < \rho < 1)$ という制限を設けていなかった。これらを修正したことによる最適条件の変化を表 E-2-3 に示す。なお時間的都合により検証実験のやり直しは行っていない。また、これらを修正したことによる近似式と教師データの誤差二乗和の変化を表 E-2-4 に示す。

表 E-2-3 最適条件の変化

プレイヤー (章, 集中度合の指標)	誤った近似での 最適条件 (正規乱数)	正しい近似での 最適条件 (正規乱数)
被験者 A (第 7 章, Attention)	平均 : 3.6 標準偏差 : 0.74	平均 : 5.4 標準偏差 : 0.91
被験者 A (第 9 章, Attention)	平均 : 8.0 標準偏差 : 0.50	平均 : 7.7 標準偏差 : 0.42
被験者 C (第 9 章, Attention)	平均 : 2.0 標準偏差 : 1.04	平均 : 3.2 標準偏差 : 1.25
被験者 D (第 9 章, Attention)	平均 : 8.0 標準偏差 : 0.46	平均 : 6.4 標準偏差 : 0.67
被験者 A (第 11 章, 独自指標)	平均 : 5.9 標準偏差 : 2.09	平均 : 6.0 標準偏差 : 1.54
被験者 C (第 11 章, 独自指標)	平均 : 8.1 標準偏差 : 0.49	平均 : 7.6 標準偏差 : 0.62
被験者 D (第 11 章, 独自指標)	平均 : 7.9 標準偏差 : 0.38	平均 : 7.0 標準偏差 : 0.89

表 E-2-4 近似式と教師データの誤差二乗和の変化

プレイヤー (章, 集中度合の指標)	誤った近似での誤差二乗和	正しい近似での誤差二乗和
被験者 A (第 7 章, Attention)	6.175	2.664
被験者 A (第 9 章, Attention)	4.135	4.134
被験者 C (第 9 章, Attention)	5.402	3.012
被験者 D (第 9 章, Attention)	4.304	2.789
被験者 A (第 11 章, 独自指標)	4.963	1.687
被験者 C (第 11 章, 独自指標)	6.025	4.937
被験者 D (第 11 章, 独自指標)	5.217	4.442

表 E-2-3 より, 誤りがあったことで検証実験において検証すべき最適条件が正しくなかったことがわかる。

表 E-2-4 より, 誤りがあったことで近似式と教師データの誤差二乗和が下がりきっていなかったことがわかる。誤りの修正による誤差二乗和の変化について, 表 E-2-4 の 7 回の実験の平均では誤差二乗和が 1.794 減少した。

F) 法政大学生に向けて

平原研管理の google ドライブの「2023 年度卒論修論」内の「20X3030_荻野晴大」というフォルダに本研究に関する資料及びプログラムをまとめた。フォルダの中身は以下の通りである：

- ・ 20X3030 荻野晴大中間発表レジュメ.pdf
- ・ 20X3030 荻野晴大中間発表レジュメ.docx
- ・ 20X3030 荻野晴大中間発表プレゼン資料.pdf
- ・ 20X3030 荻野晴大中間発表プレゼン資料.pptx
- ・ 20X3030 荻野晴大卒論 1 枚スライド.pptx
- ・ 20X3030 荻野晴大卒業論文レジュメ.pdf
- ・ 20X3030 荻野晴大卒業論文レジュメ.docx
- ・ 20X3030 荻野晴大卒業論文本論.pdf

- ・ 20X3030 荻野晴大卒業論文本論.docx
- ・ 20X3030 荻野晴大最終発表プレゼン資料.pdf
- ・ 20X3030 荻野晴大最終発表プレゼン資料.pptx
- ・ AI フォルダ
 - ・ Simple_ori2_noREC_scoreBase_rand_noLOG.jar：第 4 章での AI
 - ・ Simple_ori2_43_noLOG.jar：第 8 章での AI
 - ・ original2_u_01.data：第 4 章での AI の行動評価値
 - ・ original2_43_01.data：第 8 章での AI の行動評価値
 - ・ Mcts_DDR0.jar：接戦 AI
 - ・ javax.json-1.0.4.jar：json のライブラリ
- ・ プログラムフォルダ
 - ・ FitbitAPI-copy.py：プログラム A-1
 - ・ 1 次近似.py：プログラム E-1-1
 - ・ 2 次近似.py：プログラム E-2-1

セクション C-1 及び第 4 章での仕様の AI を使用するためには、図 F-1 のように AI 本体である“Simple_ori2_noREC_scoreBase_rand_noLOG.jar”という jar ファイルを“data”フォルダ内の“ai”フォルダ内に置いておく必要がある。また学習済み評価値をまとめた“original2_u_01.data”という DATA ファイルを“data”フォルダ内の“aiData”フォルダ内の“SimpleAI”フォルダ内に置いておく必要がある。

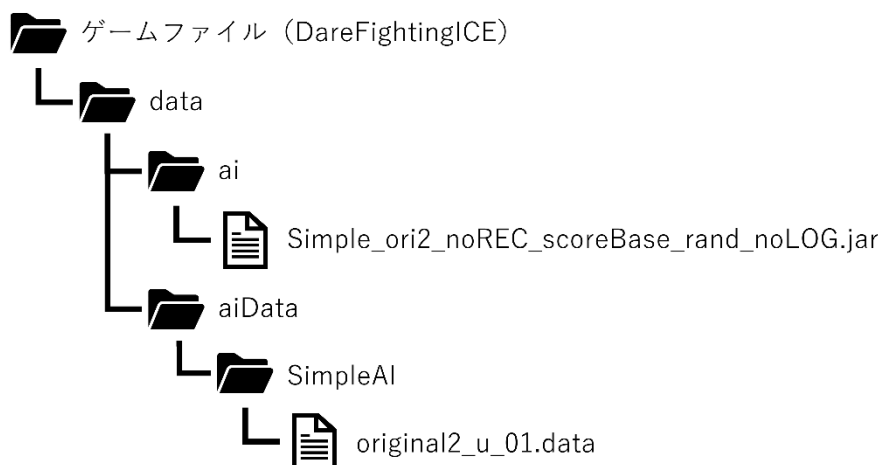


図 F-1 フォルダ構成図

この AI を使用する際にはコンソールやコマンドプロンプトに図 F-2 のように表示が出るので、AI の行動決定に用いる正規乱数の平均及び標準偏差を入力する必要がある。試合を行った後はコンソールやコマンドプロンプトに図 F-3 のように試合開始時刻、各ラウンド

終了時刻及び各ラウンド終了時の HP の表示が出る。また、その試合での時刻、プレイヤーの行動、AI の行動、キャラクター間の距離の分類及びキャラクター間の距離が csv ファイルにまとめられ、"data"フォルダ内に作成される。サンプリングレートは約 25Hz である。

```
1月 04, 2024 6:57:25 午後 util.LogWriter <init>
情報: Create instance: util.LogWriter
1月 04, 2024 6:57:25 午後 manager.InputManager startAI
情報: Initialize AI controller for P2
read from file\data\aiData\SimpleAI\original2_u_01.data
難易度(基準評価値)を設定してください(半角数字double)(0~10): 5
難易度の振れ幅(評価値の標準偏差)を設定してください(半角数字double)(0~10): 1.5
round start
1月 04, 2024 6:59:27 午後 gameScene.Play initialize
情報: AI controller is ready
1月 04, 2024 6:59:28 午後 util.ResourceDrawer <init>
情報: Create instance: util.ResourceDrawer
```

図 F-2 正規乱数の平均及び標準偏差の入力画面

```
FINISHING...
iswin: false || HP1:161 || HP2:0
start: Thu Jan 04 18:59:26 JST 2024
ROUND1
P1HP: 58
P2HP: 0
end : Thu Jan 04 19:00:22 JST 2024
ROUND2
P1HP: 98
P2HP: 0
end : Thu Jan 04 19:00:46 JST 2024
ROUND3
P1HP: 161
P2HP: 0
end : Thu Jan 04 19:01:09 JST 2024
difficulty_score : 5.0, stdev1.5
CSV書き込みが終了しました
1月 04, 2024 7:01:09 午後 gameScene.Play update
情報: Game over
```

図 F-3 試合結果の表示

セクション C-2 及び第 8 章での仕様の AI を使用するためには、図 F-4 のように AI 本体である "Simple_ori2_43_noLOG.jar" という jar ファイルを "data" フォルダ内の "ai" フォルダ内に置いておく必要がある。また学習済み評価値をまとめた "original2_43_01.data" という DATA ファイルを "data" フォルダ内の "aiData" フォルダ内の "SimpleAI" フォルダ内に置いておく必要がある。この AI も使用に際して、同様に図 F-2 のように表示されるので、AI の行動決定に用いる正規乱数の平均及び標準偏差を入力する必要がある。また試合の時刻や

HP も同様に図 F-3 のように表示され、試合中の情報をまとめた csv ファイルも作成される。

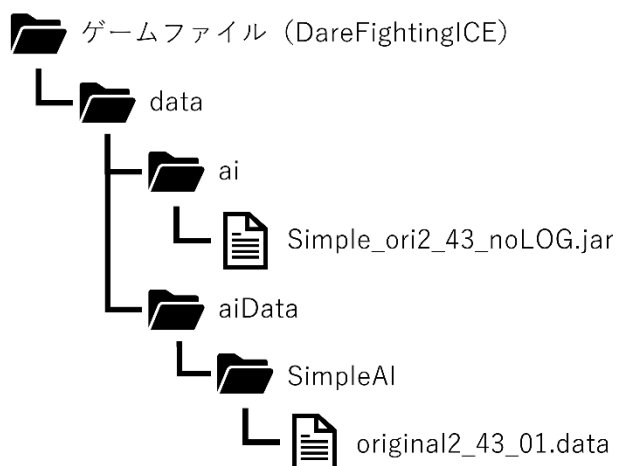


図 F-4 フォルダ構成図

第 13 章の接戦 AI を使用するためには、図 F-5 のように AI 本体である“Mcts_DDR0.jar”という jar ファイルを“data”フォルダ内の“ai”フォルダ内に置いておく必要がある。この AI を使用して試合を行った後は、コンソールやコマンドプロンプトに図 F-3 のように試合開始時刻、各ラウンド終了時刻及び各ラウンド終了時の HP の表示が出る。

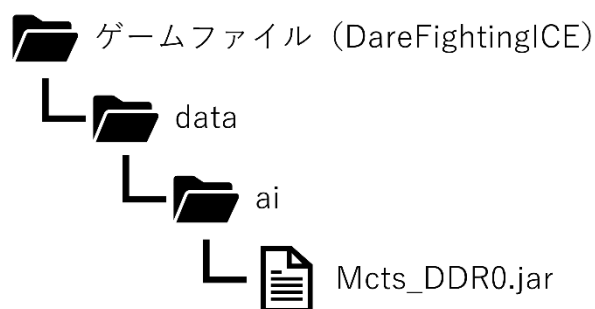


図 F-5 フォルダ構成図

付録参考文献

[図 A-1] 株式会社ニューロスカイ : “NeuroSky(脳波)”,
<https://www.neurosky.jp/brainwaves/>

[A1] サボテンパイソン：“[fitbit] 3. Python で Fitbit API を使って心拍数データを取得して matplotlib で表示”，<https://sabopy.com/py/fitbit-3/>

[D1] 立命館大学 Intelligent Computer Entertainment Lab：“FightingGameAI Competition(Get started ----- First Step)”，<https://www.ice.ci.ritsumei.ac.jp/~ftgaic/index-2h.html>

[E1] ImageMagick Studio LLC：“ImageMagick(Download)”，<https://imagemagick.org/script/download.php>