

電子情報工学専攻

Advanced Electronic and Information Engineering Course

平成29年度 専攻科特別研究論文

プログラムの実装戦略推定時における脳波の時系列分析

Programmer's Electroencephalogram who Found
Implementation Strategy

指導教員名 上野秀剛 准教授

論文提出者名 山本愛子

独立行政法人 国立高等専門学校機構

奈良工業高等専門学校 専攻科

National Institute of Technology, Nara College
Faculty of Advanced Engineering

プログラムの実装戦略推定時における 脳波の時系列分析

Programmer's Electroencephalogram who Found Implementation Strategy

山本 愛子

Aiko Yamamoto

独立行政法人国立高等専門学校機構
奈良工業高等専門学校専攻科電子情報工学専攻
大和郡山市矢田町22番地 (〒639-1080)

National Institute of Technology, Nara College, Faculty of Advanced Engineering
22 Yata-cho, Yamatokoriyama, Nara 639-1080, Japan

Abstract: Electroencephalogram (EEG) used on a wide research area to measure human brain activities. In Software Engineering research area, several studies combined the program comprehension and EEG measurement. The EEG is one of the most easy and the most low cost method to measure the programmer's comprehension process. Therefore, the detection of programmer who need support to understand a program using the EEG measurement is a practical method. In this study, author analyze programmers' EEG who found implementation strategy to reveal the usefulness of the EEG measurement.

As the index of programmer's brain activity, author select α wave, β wave, and the ratio of α to β , which is the frequency component of EEG. In the experiment, each subject assigned programming tasks, and the EEG during/after the task is measured. In the first experiment, the frequency components at during task and after task are analyzed. As a result, α wave of subjects who found implementation strategy is a significantly high from subjects who cannot found implementation strategy at both during/after task. The result suggests that α wave involved in measured EEG is useful to understand the programmer's state.

In the second experiment, author analyze time series change of the EEG, to clarify the time period when the most influenced from the task condition. The average value of each 1, 5, and 10 seconds α wave power spectrum is used as an index. The result of the analysis shows the time period from the sixteen seconds after the task start to the end is most influenced from the task result.

Results of two experiments show α wave of programmers who found implementation strategy is a significant index. Also the result shows the EEG from early period of the task is a low contribution for analysis. These results suggested that EEG is the useful index to distinguish the programmer who cannot found implementation strategy. EEG as the index to distinguish the programmer's state able to help them without any interruption, hence useful to improve work/education efficiency.

目次

1	はじめに	2
2	関連研究	3
2.1	プログラム理解時の脳の働き	3
2.2	プログラム理解の計測	3
2.2.1	心理学的アプローチ	3
2.2.2	生体情報計測を用いたアプローチ	4
2.2.3	脳活動計測を用いたアプローチ	5
2.3	脳波と心理状態	7
3	脳波	8
3.1	計測方法	8
3.2	周波数が示す特徴	8
3.3	周波数分析	9
4	実験	11
4.1	実験環境	11
4.2	タスク	12
4.3	脳波計測	13
4.4	手順	15
4.5	時系列分析	15
5	結果と考察	17
5.1	実験1	17
5.1.1	タスク中の脳波	17
5.1.2	タスク後の脳波	19
5.1.3	各個人の成分の比較	21
5.2	実験2	22
6	おわりに	25
7	謝辞	27
	参考文献	28

1 はじめに

現代の高度情報化社会では，多くの教育機関や企業でプログラミング技法を学ぶ学習者が急増している．プログラミングを行う開発・教育現場で，作業の効率や教育効果を上げるためには指導者からの支援が必要である．支援対象は，文法がわからない初心者や文法はわかるが仕様を理解できない場合などいくつか考えられるが，本研究では文法，仕様ともに理解できているが，実装したい処理に対して必要なアルゴリズムやデータ構造の組み合わせ（以降，実装戦略）を思いつかないプログラマを対象とする．その対象となるプログラマの理解度や思考状態を外観から把握するのが難しい．

人の思考状態や心理状態をみるための方法として脳波計測がある [1]．脳波の周波数成分である α 波と β 波はリラックス状態や精神活動状態によって変動するとされており [2]，さまざまな作業における心理状態の計測に用いられている．これはプログラマの思考状態の計測に向いている．なぜなら，実装戦略を思いついているかどうかは心理に影響するからである．実装戦略を推定できていないプログラマは，ストレスを感じイライラしている心理状態にあると考えられる．一方，実装戦略を推定できているときには，比較的ストレスが少なく，快感を抱いている心理状態にあると考えられる．つまり，プログラマの実装戦略を推定できた状態とそうでない状態の差は，プログラマの心理状態に反映されると考えられる．また，他の脳活動測定装置と比べて測定が手軽で低コストであることも，脳波がプログラマの思考状態の計測に向いている理由の1つである [3]．そこで本研究では，プログラマがプログラマの実装戦略を推定できたか否かで脳波に差異が存在するか検証する．

実験では，プログラム問題のタスクを被験者に与えたときの脳波を計測し，プログラマの実装戦略を推定できたときと推定できなかったときに脳波に差が生じるか計測する．さらにタスク中の α 波に対し，時系列分析を行い，タスクの影響が α 波に強く現れる時間があるか検証する．

2 関連研究

2.1 プログラム理解時の脳の働き

従来，ソフトウェア開発に必須な作業であるプログラム理解を支援したり，理解過程や理解度を計測するために様々な研究が行われてきた[4][5]．近年では，認知心理学や神経外科の知見を活かし，人間の脳の働きや制約を考慮したプログラム理解のモデル化が行われている．

Parninらは，複数のプロジェクトで内容の異なる業務に携わる開発者について，プログラムに対する記憶が保持される様子を認知心理学の観点から理論的に分析した[6]．彼らはまた，実験とインタビューによって，プログラムに関する短期記憶や長期記憶の精度，ならびに保持の様子が時間経過によってどのように変化するかを分析した．

中村らは，プログラム理解時の人間の短期記憶，特に変数およびコントロールフローの記憶と想起に着目し，プログラム理解のモデル構築を行った．このモデルは実際にプログラムの読解にかかった時間をよく近似することが示されている[7]．石黒らは中村らのモデルを改良し，ある記憶を繰り返し参照することで内容を忘れにくくなるリハーサル効果がプログラムの読解速度に影響を与えた可能性について述べている[8]．

2.2 プログラム理解の計測

2.2.1 心理学的アプローチ

教育の場や企業で開発者・プログラムの評価を目的に，プログラム理解を計測することが一般的に行われている．

たとえば，プログラム理解に関する人的要因(理解度，開発者の行動など)を定性的に評価する方法として，レトロスペクティブ法やThink-aloud法など実験心理学の手法が用いられてきた[4][9]．両手法は，被験者が特定の作業(タスク)を実行する被験者実験において，実験中の被験者の感想や行動，思考内容について明らかにするための方法である．レトロスペクティブ法では，被験者実験の終了後，あるいは実験を定期的に中断し，インタビューやアンケートを実施し，過去にさかのぼってプログラム理解の過程を答えてもらう．Think-aloud法では，被験者実験の実行時，被験者自身の思考内容や行動について随時発話することで同様の情報を得ることを目的としている．この二手法は古典的な方法ではあるが，それぞれ以下のような弱点がある．

1. 結果が被験者の主観に依存するため，定量的に比較できない(共通)
2. 実験終了後の一時点での記憶となり，時間分解能が存在しない(レトロスペクティブ法)

3. 実験で取り組むタスクの本質とは関係のない発話という行動を行うため、被験者への負荷が高い (Think-aloud 法)

これらの問題のうち、2) および 3) を緩和するため、Karahasanovic らはレトロスペクティブ法や Think-aloud 法に代わる手法として、フィードバック・コレクション法を提案している [10]. この手法では、定期的に「いま何について考えていましたか?」と尋ねるダイアログを画面に表示し、被験者はこれに自由記述形式で解答する.

Karahasanovic らによると、この手法はレトロスペクティブ法および Think-aloud 法のいずれよりも学習コストが低く、レトロスペクティブ法に比べて理解の妨げになる要素を多く発見できると述べている. ただし、フィードバック・コレクション法は Think-aloud 法よりも時間分解能が低く、時間分解能と計測時に被計測者にかかる負荷の間にトレードオフの関係があることが示唆された [11]. すなわち、フィードバック・コレクション法を用いて開発者のプログラム理解の成否を即時的に知ろうとすればするほど、計測手法自体が開発者の作業を阻害してしまうと言える.

また、被験者の主観に依存しない定量的な計測法として、プログラム理解終了時の試験を用いる方法や、プログラム理解にかかった時間を測定する手法が用いられてきた. 栗山らは、コードレビューにおけるバグ発見率とプログラム理解度の関係を調べるために、レビュー終了後にテストを課し、高得点を取った被験者ほどバグ発見効率が高くなるという結果を得た [12]. Dunsmore らは、プログラム理解度を測定するためのテスト項目について、どのような問題を出題することで実際の理解度を測定できるかについて比較実験を行った [13].

これら定量的計測法の課題は、定性的計測における問題点と同じく、時間分解能が存在しないことである. すなわち、プログラム理解中の特定の時点において、開発者がプログラムを順調に理解できているかどうかは、以上の方法では知ることができない.

2.2.2 生体情報計測を用いたアプローチ

心理学的アプローチの制約から、生体情報計測を用いた手法が提案されている. この手法は、プログラム理解に関する人的要因を定量的に観測可能にする、かつ時間分解能が高く、計測時の負荷が低いアプローチである.

村岸は、皮膚抵抗値 (Skin Resistance Level, SRL) の計測を用いて C 言語プログラミング中の作業負荷計測を試みた [14]. 被験者実験によって、プログラミング中の精神的負荷、たとえば成果の納期が近いかどうかといった要素と SRL の値に関係があることを報告している. SRL 測定によって計測できる作業負荷とは、簡単に言うとフラストレーションや精神的な緊張のことを指し、また逆に負荷のない状態とはリラックスしている状態を指す. このことは、村岸らの実験において残り時

間が短い状態で、仕様変更を命じられた開発者の負荷が上昇した点にも明らかである。しかし、SRL測定のもつ問題として、測定時に起こる皮膚抵抗反射や自発性皮膚抵抗反射といった一過性の皮膚抵抗変化が計測ノイズとなり、計測時の時間分解能が低下することが挙げられる。さらに極度に強い心理的ストレスがかかった場合に、リラックス状態の抵抗値に戻るまで20~30分かかることも報告されている。

これに対して、本研究のアプローチである脳活動の計測は、フラストレーションや精神的な緊張にも影響を受けるが、基本的には脳が実際に知的活動を行う際に必要としたエネルギーの量を測定していると考えられる。このことは、プログラム理解に必要なとされた知的活動の量と直接関係のない精神的負荷が計測値に現れることを軽減できることを意味する。

また、視線計測によって、画面中あるいは現実世界のどの部分を注視しているか、あるいはどこも注視していないかといった情報からプログラム理解の様子を観察する方法も多く研究されている。

たとえば上野らは、コードレビューを行っている人物の視線移動をリアルタイムに記録し、ある時点においてコードや仕様書のどの部分を読んでいるかを計測する、Dresrem2というツールを作成した。上野らは同ツールと視線計測装置を利用した被験者実験で、はじめに全体に目を通してから一部のコードを集中して読む「スキャンパターン」が多く見受けられるほど、コードレビュー時のバグ発見効率が高くなることを発見した[1]。

視線計測を用いることで、他の計測方法では知り得ない、人間が画面上のどこに注目しているかや、その順序に関する情報を客観的に知ることができるという利点がある。一方、特定部分を読んでいる開発者が何を考えているか、どのように感じているかを知ることができず、あくまで推測に頼ることになる。また原理的に、何かを見たり読んだりすることを伴わない知的活動を計測できないという課題もある。たとえば、作業を中断してリラックスしようとしている場合や、頭の中で思い悩んでいるような場合に適用できない。

2.2.3 脳活動計測を用いたアプローチ

脳血流や脳波から脳の活動を計測する機器が比較的手に入りやすくなったこともあり、近年はプログラム理解と脳計測を組み合わせた研究が注目を集めている。こうした研究の多くは、プログラム理解・作成に含まれる認知プロセスを同定したり、開発者のストレスあるいは集中の状態を推定するものである。

Siegmundらは、fMRIを用いてプログラム理解における脳の部位ごとの活性化を調査している[15]。最大18行の短いソースコードを理解するタスクを対象とした実験の結果、問題解決、記憶、および文章理解に関係する脳領域がプログラム理解時に活発になることを示している。同論文では、プログラム理解と脳活動の関

係を探ることの重要性が主張されており、長期的には「何が悪いプログラマや悪いプログラムを生む原因になるか?」のような課題に解答したいと述べている。fMRIを用いた計測は、空間分解能が高く、特定の脳部位や認知プロセス（記憶、言語など）とプログラム理解の関係性を調査するには最も信頼性の高い手法である。一方で時間分解能が低く、被験者の体動を強く制限することになるため、実際のプログラミング現場を模した環境での実験には向いていない。

そこで、安価で簡易な脳波計測機やポータブルNIRSのような機器を用い、実環境に近い状態でプログラム理解を計測する試みが提案されている。

中川らは、プログラム理解活動を定量的に評価することを目的に、NIRSを用いて前頭前野の脳血流を計測することでプログラム理解に困難が生じている状態の判別が可能か検証する実験を行った。実験により、課題の難易度によって脳活動に差があり、課題の序盤から中盤にかけて脳血流値の正の変化量が最大になると示している [16]。

Fritzらはプログラム理解時の視線、脳波、筋電を同時計測し、機械学習を用いて計測値からタスクの性質や開発者の状態の推定を試みた [17]。具体的には、意図的に結果を想像しにくくしたプログラム(高難度)とそうでないプログラム(低難度)を被験者に与え、その結果を予測させる問題を解いている際の計測値から、プログラムの読解難度を推定させる。この機械学習を用いた手法は、精度や再現率が二値分類で6割から7割程度とそう高くないため、以前さらなる調査が必要な状態であると言える。

これ以降、同様に多くのセンサを用い、機械学習を用いた測定手法に関する研究が活発に行われている。たとえば、Zugerらはプログラム理解を試みる開発者の作業効率を下げずに作業割り込みを発生させることを目的に、「作業割り込みを行っても問題ないか?」を二値で推定することを試みた [18]。Mullerらはプログラム理解中の開発者の感情状態（集中・発散や幸福・不幸）を心理学的なモデルと突き合わせて推定することを試みた [19]。これらの研究ではいずれも脳波計測機と視線計測機の計測値が機械学習の精度向上に寄与しており、視線・脳波記録がプログラム理解の測定において一定の効果を上げていることが確認できる。

上記の研究から、プログラム理解度と脳活動には関連があることが示されている。そして、脳波記録がプログラム理解の測定において一定の効果を上げていることから、本研究ではプログラムの実装戦略を推定できたか否かの識別が脳波計測によって可能か実験によって検証する。脳波は、測定が手軽な上に低コストであるため、プログラムの理解が不十分なプログラマへの迅速な支援に用いるツールとして実用性が高いと考えられる。

2.3 脳波と心理状態

人の脳活動を非侵襲で簡単に測定する手法として、脳波(Electroencephalogram:EEG)はさまざまな研究分野で用いられている。

満倉は、17年前から脳波からその意味を検出する研究をしており、小型の脳波計測器のみで人の感性を取得できる装置を構築し、脳波からストレスを検知するシステムを提案している[3]。上野らの研究では、ソフトウェアのユーザビリティ評価の際に人間の心理状態を定量的・客観的に評価する方法として脳波の有用性が確かめられている[20]。森田は、暗算課題および視覚野に特化した画像処理課題を用いて、安静時、課題遂行時の脳波と心電図を同時測定し、脳の2つの部位の間に成立しているネットワークを調べた。その結果、低域 θ 波の脳部位間のネットワークは課題に対する集中度の高さを反映すると推測された[21]。

本研究が対象とするプログラム理解においても、実装戦略を推定できた状態とそうでない状態ではプログラマの心理状態が異なり、脳波にその影響が反映されると考えられる。そこで本研究では、プログラミング中の作業者の心理状態の推定に脳波が有用だと推測し、脳波を用いて脳活動を測定する。

3 脳波

3.1 計測方法

脳波とは，脳から生じる電気活動を電位を縦軸，時間を横軸にとって記録したものである [22]. 脳波は，頭皮上に装着した電極から計測される．電極の配置は図 3.1 に示す国際式 10-20 電極法 [23] に則って行う．国際式 10-20 電極法では耳のアースを除き 19 箇所 of 装着位置が指定されており，検査や研究の目的によって使用する電極を決定する．

脳波の導出法には主に基準電極導出法と双極導出法の 2 種類の方法がある．基準電極導出法では，脳電位の電場内に装着した計測用電極と，電場外に装着した基準電極の 2 つの電極の電位差として脳電位を測定する．双極導出法は，基準電極を用いず，2 つの計測用電極を脳電位の電場内に置いて記録する方法である．脳電位は 2 つの電極の電位差として測定される．計測用電極の電極間隔が狭い場合に電位差を計測する際は基準電極導出法を，優勢な背景成分を除去して部位差を強調する目的で計測する際には双極導出法を選択する [24].

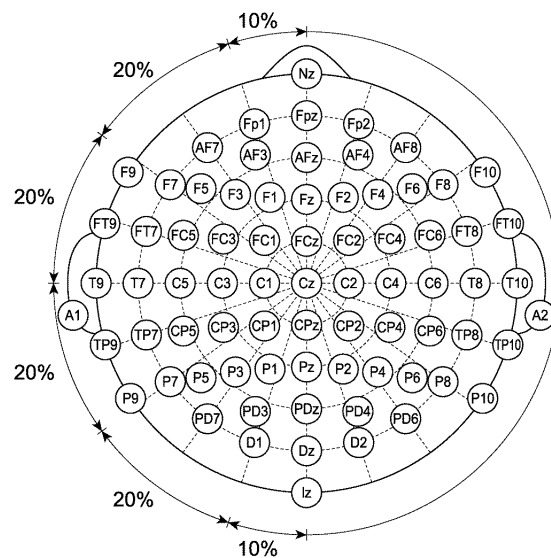


図 3.1 国際式 10-20 電極法

3.2 周波数が示す特徴

一般的に脳波は時間の経過にともなって複雑な電位変動を示す不規則な振動現象とみなされ，高速フーリエ変換 (FFT) を用いてパワースペクトルが求められる．脳波からパワースペクトルを求める際には，国際脳波学会によって周波数帯

域ごとに付けられた分類および名称を用いる [2]. 各帯域の名称と周波数帯域を以下に示す.

- δ 波: 0.5~4Hz
- θ 波: 4~8Hz
- α 波: 8~14Hz
- β 波: 14~30Hz
- γ 波: 30Hz 以上

δ 波や θ 波は睡眠状態にあるときに出現する. α 波は安静状態にあるときに強く表れる周波数帯域で, リラックスし, 何かに没頭しているときに出現する. 他の周波数帯域の波と比べて振幅も連続性も最も高い. 眠気を感じるなど覚醒が低下してくると, α 波の振幅が低下して不連続になる. また, α 波は開眼すると大幅に減少し, 閉眼すると再び出現する. 一般的にこれを α 波減衰と呼ぶ. また, 緊張や不快な感情を抱いているときや日常の思考状態では β 波が出現する. γ 波は, 不安で興奮しているときに出現する [24][2]. これらの周波数帯域の内, α 波と β 波はリラックス状態や精神活動状態によって変動するとされており [2], さまざまな作業における人間の心理状態の計測指標に用いられている [25]. また, α 波と β 波の比率は脳の活動を計測するための指標としてよく用いられている.

本研究において, プログラムの実装戦略を推定できなかった状態では, 推定できたときに比べて緊張や不快な感情が大きくなると考えられる. したがって, 実装戦略を推定できたか否かによって心理状態に差異が生じ, それに伴って脳波の α 波と β 波の周波数成分にも差異がみられると仮定し, 実験で検証する.

3.3 周波数分析

一般的に脳波の分析では, 計測したデータに含まれる周波数ごとの成分の大きさを調べる周波数分析を行う. 周波数分析をするには, 時間の経過にともなって複雑な電位変動を示す脳波を不規則な振動現象とみなし, 高速フーリエ変換 (FFT) を用いてパワースペクトルを求めることが多い. FFT は離散フーリエ変換 (DFT) における計算の無駄を省くため, 三角関数の周期性を利用した計算技法で, DFT に比べ計算時間は速いが, データ点数は 2 のべき乗でなければならないという制約がある.

FFT をする前に, エイリアシングの混入に注意してアナログデータを A/D 変換して離散データにする. エイリアシングの混入を防ぐにはアナログフィルターで高周波をカットするか, 求めたい周波数よりも 2 倍程度高周波成分までをサンプリングしておく. また, サンプリング長も低周波カットとして問題となり, これら

はサンプリング定理として知られている。サンプリング定理とは、「サンプリング長を T 秒，サンプリング間隔を S 秒とすると，高周波側は長さ S ごとの平滑化による $1/2 \cdot S$ 以上の周波数の高周波カットオフフィルターとして作用し，低周波数側では $1/2 \cdot T$ 以下の周波数の低周波カットオフフィルターとして作用する」とまとめられる。

FFTによってパワースペクトルを求める一般的な手順は以下の1から6である。なお，観察したい周波数よりもはるかに低い周波数の変動成分をトレンドという。

1. データの決定

サンプル数が 2^N (N は整数) のデータを用意する。

2. トレンドの除去

トレンドをフィルターで除去する。

3. データウィンドウ

ある区間のデータをそのまま用いると，データの最後と最初が不連続なデータになるのでスペクトルに大きなゆがみが生じる。そのため，データに窓関数をかけてことでデータの両端をなだらかに0に近づけ，データの最初と最後の段差を取り去る。

4. FFTの計算

t を時間， x を計測した脳波の生データ， $g(x)$ をすべての実数と定義したとき，パワースペクトル $f(t)$ は以下の式で求められる。

$$f(t) = \int_{-\infty}^{\infty} g(x)(\cos(2\pi xt) + jsin(2\pi xt))dx$$

5. スペクトルの平滑化

スペクトルは一般に振動が激しいので，周波数領域で一定のバンド幅をもつ窓関数を乗じて移動平均をとり，平滑化操作をおこなう。

6. スペクトルの表示

各スペクトルは，0からナイキスト周波数までの周波数軸に対して表示する。

4 実験

一部が欠けたJavaのソースコードと仕様からなる問題を提示し、仕様を満たすプログラムを考えてもらうタスクを被験者に与え、脳波を計測する実験を2回行う。実験1では、タスク中とタスク後それぞれにおける α , β , β/α の計測時間全体の傾向をみる。実験2では、タスク中における α の計測時間内での変化をみる。被験者は奈良工業高等専門学校情報工学科の学生17人で、年齢は16歳から20歳、全員がJavaによるプログラミングの基礎講義を受講済みである。

4.1 実験環境

実験は被験者1名と実験者2名のみが居る静かな部屋で実施する。脳活動の計測装置はナノテックイメージ社製NeXus-10 MARK IIを用いる。本装置は脳波だけでなく、脳血流、脈波、呼吸、発汗、心電、皮膚漏といった生体信号をリアルタイムに計測・解析できるシステムである。図4.1に装置の外観と装着時の様子を示す。本装置の計測周期は256/sec, 脳波用の電極は本研究では1チャンネルのみ使用するが、最大8チャンネルまで使用できる。計測されたデータはbluetooth経由でPCに転送され、csvファイル形式で出力される。

実験者は、1台のPCで装置の制御と、データの記録、および問題の提示をする。被験者にはPCに接続した別のディスプレイで問題を提示する。被験者は、問題を見て頭の中でプログラムの実装戦略を考える。脳波以外の体動によるノイズ（アーチファクト）を抑制するためにキーボードやマウスなどの操作は一切行わない。また、同様の理由から、被験者はヘッドレスト、肘掛け、足置きを備えたイスに座り、椅子の高さやディスプレイの高さを事前に調節する。



[a] 装置の外観



[b] 装着時の様子

図 4.1 計測装置

使用したPCは、CPUがIntel(R) Core(TM)i5-3380M 2.90GHzでメモリ搭載量は4GBで

EEG Measurement Experiment Tool

File Start Time 54 sec

入力された整数x,y,zの中央値をもとめるプログラムを作成したい。
プログラムが完成するように、不足している部分を考えてください。

```
import java.util.Scanner;
public class Center {

    //中央値を返すメソッド
    static int tyuo(int a,int b,int c)
    {

        //////////////////////////////////////

        ここを考えてください。

        //////////////////////////////////////

    }

    public static void main(String[] args)
    {
        Scanner stdIn=new Scanner(System.in);
        int x=0,y=0,z=0,tyuodata;
        System.out.print("xを入力:");
        x=stdIn.nextInt();
        System.out.print("yを入力:");
        y=stdIn.nextInt();
        System.out.print("zを入力:");
        z=stdIn.nextInt();
        tyuodata=tyuo(x,y,z);
        System.out.println("中央値は"+tyuodata+"です。");
    }
}
```

図 4.2 タスク提示ツール

ある。被験者に問題を提示するために使用するディスプレイは21.3インチで、解像度は1920 x 1200[ppi]、向きは横向きである。問題提示のために、図4.2に示す実験用ツールをC#言語で作成した。本ツールはテキスト形式で保存された問題と制限時間を被験者に表示する。

4.2 タスク

実験ツール上に提示される問題文に従って、プログラムの実装戦略を口頭で解答してもらうタスクを設定する。プログラム問題は、自然言語で書かれたプログラムの仕様とJavaで書かれたソースコードからなる12問を用意する。ソースコードは不足部分が1問につき1箇所あり、被験者は不足部分にどのような処理手順でソースコードを書けば仕様を満たすか考える。問題文は1問ごとにディスプレイ上に表示する。

プログラム問題は、すべての問題で実装戦略を推定できる被験者や、すべての

問題で推定できない被験者が現れないように、以下の5段階の難易度を設定し、用意する。

- 難易度1: 条件分岐のみで構成
- 難易度2: 1重for文と条件分岐で構成
- 難易度3: 2重for文や3重for文と条件分岐で構成
- 難易度4: メソッドを2つ使用
- 難易度5: 再帰的アルゴリズムを含むメソッド3つを使用

タスクとして提示する問題の一覧を表4.1に示す。問題の順番は、順序効果を考慮しカウンターバランスを行った。

表 4.1 タスク

	ファイル名	概要
A	Center.java	3値の中央値をもとめる
B	Copy.java	配列の要素を逆順に他の配列にコピー
C	EightQueen.java	8王妃問題
D	FizzBazz.java	条件に従った出力の変更
E	Henkan.java	英字の大文字小文字変換
F	Max.java	3値の最大値をもとめる
G	Multi.java	配列の積をもとめる
H	Secret.java	シーザー暗号の生成
I	Sort.java	クイックソート
J	Tenti.java	転置行列をもとめる
K	Triangle.java	左上が直角の三角形の表示
L	Uruudoshi.java	閏年の判定

タスクには制限時間を設け、1問につき60秒間とする。その後、被験者は実験者に口頭で回答を述べる。被験者が述べた回答を受けて、プログラムの実装戦略を推定できたか否かを実験者が評価する。被験者には回答の正否は伝えない。なお、本稿では回答が誤っている場合であっても、なんらかの実装戦略を回答した場合には実装戦略を推定できたと評価する。

4.3 脳波計測

1つ目のタスク開始前に被験者に脳波計測装置を装着し、タスク中、およびタスク後の脳波を計測する。タスク中の脳波を60秒間、タスク後の脳波を120秒間計測する。問題の多さから被験者の疲労を考慮して、実験の途中で随時休憩が必要か被験者に尋ね、5分程度の休憩を取る。

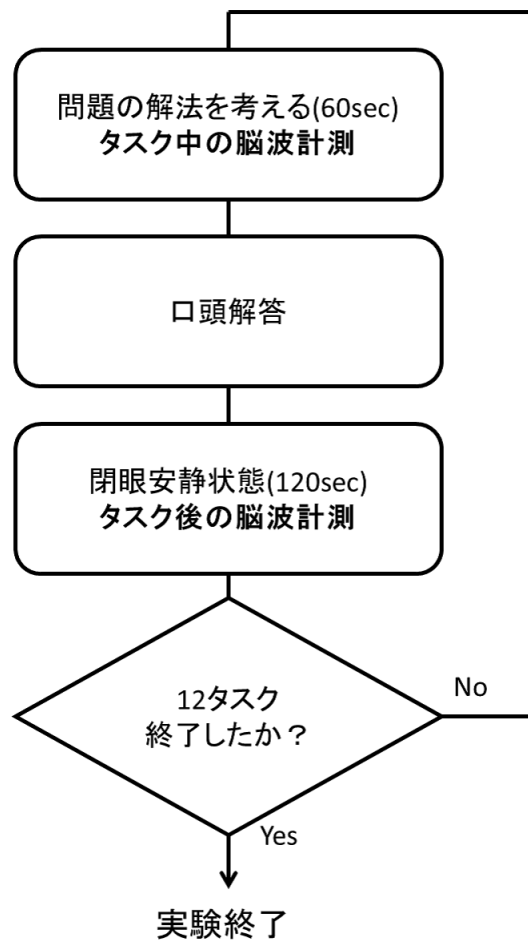


図 4.3 計測の流れ

脳波の計測に用いる電極は、グラウンド電極を右耳(A2)とし、導出法については基準電極導出法(片側耳朶法)を用いて、基準電極を左耳(A1)、計測用電極を後頭部(Pz)に配置した。計測用電極を後頭部(Pz)に配置した理由は、後頭部は筋電位など脳波以外の生体現象のアーチファクトが入りにくいためである。電極の装着不良によるアーチファクトを防ぐために被験者には頭部用ネット包帯を着用してもらう。また、脳波は筋電位によるアーチファクトにより影響を受けやすいため、実験中はできるだけ体を動かさないように被験者に指示する。

4.4 手順

実験の手順を以下に示す。

1. 実験説明・準備

実験の流れについての説明と、脳波計測時の注意を行う。例題として用意した問題を用いて、タスクの内容を説明する。

2. 装置の設定

4.3節で説明した3箇所電極をつけ、脳波計測装置の設定を行う。脳波が正常に取れているかを確認するために、閉眼安静状態の脳波を計測する。

3. タスクの実施

ディスプレイに問題を60秒間表示し、実装戦略を考えてもらう。

4. 口頭での解答

考えた実装戦略を述べてもらい、被験者がプログラムの実装戦略を推定できたか評価する。

5. タスク後脳波計測

閉眼安静状態を120秒間維持し、脳波を計測する。

6. 全タスクの実施

手順3, 4, 5を12回繰り返す。

4.5 時系列分析

計測時間の中でも特にタスクの影響が強く現れる時間帯を明らかにするために時系列分析を行う。タスクの影響が強く現れる時間帯が明らかになると、分析時間の短縮につながると考えられる。分析は、タスク中に計測された脳波の α 波について行う。計測開始時身体が大きく動いた被験者が多かったため、筋電位によるノイズの影響を考慮し、タスク開始直後の5秒を除いた55秒間を分析対象とする。分析の流れを以下に示す。

1. ノイズ除去

計測した脳波に2-50HzのInfinite Impulse Response(IIR:無限インパルス応答)バンドパスフィルタにより, 体動, 眼球運動などのアーチファクトの除去を行う.

2. フーリエ変換

被験者 s がタスク n を実施した際の t 秒におけるノイズ除去済みの脳波を $BW(s, n, t)$ とする. 計測周期 256/sec より $N=256$ で $BW(s, n, t)$ に対しFFTをかけ, 毎秒のパワースペクトルを得る. パワースペクトルを求める周波数範囲は α 波として定義されている周波数帯域である8Hz~14Hzとする.

3. 成分抽出

得られたパワースペクトルから, α 波の帯域である8Hz~14Hzの成分を抽出する.

$$\alpha(BW(s, n, t))$$

$$\beta(BW(s, n, t))$$

4. 正規化

脳波は個人差が大きいため, 抽出された成分データを被験者の平均値で正規化する. ここで, T_S はタスク開始時間を, T_E はタスク終了時間を, N はタスク数を表す.

$$Norm\alpha(s, n, t) = \frac{\alpha(BW(s, n, t))}{\frac{1}{(T_E - T_S)N} \sum_{t=T_S}^{T_E} \sum_{n=1}^N \alpha(BW(s, n, t))}$$

5. 全被験者の平均値を算出

平均値の算出は, 1秒毎, 5秒毎, 10秒毎で行う. 各被験者が各タスクにおいてプログラムの実装戦略を推定できたか否かを $Success(s, n)$ で表す.

$$Success(s, n) = \begin{cases} 1 & \text{推定できた (Success)} \\ 0 & \text{その他 (Fail)} \end{cases}$$

推定ができた場合とできなかった場合, それぞれにおける正規化後の α 波の全被験者の平均値を求め, それぞれ $S\alpha$ と $F\alpha$ とする. ここで, $SuccessTask$ は $Success$ のタスク総数, $FailTask$ は $Fail$ のタスク総数を表す.

$$S\alpha(t) = \frac{\sum_{s, n, Success(s, n)=1} Norm\alpha(s, n, t)}{SuccessTask}$$

$$F\alpha(t) = \frac{\sum_{s, n, Success(s, n)=0} Norm\alpha(s, n, t)}{FailTask}$$

5 結果と考察

本章では，被験者17名に対して実験を行った結果を示す。

5.1 実験1

5.1.1 タスク中の脳波

計測した脳波データの例を図5.1に，脳波を解析して得られたパワースペクトルの例を図5.2に示す。プログラムの実装戦略を推定できたタスクを *Success* グループ，推定できなかったタスクを *Fail* グループに分類し，タスク中の α ， β ， β/α の値を図5.3の箱ひげ図に示す。

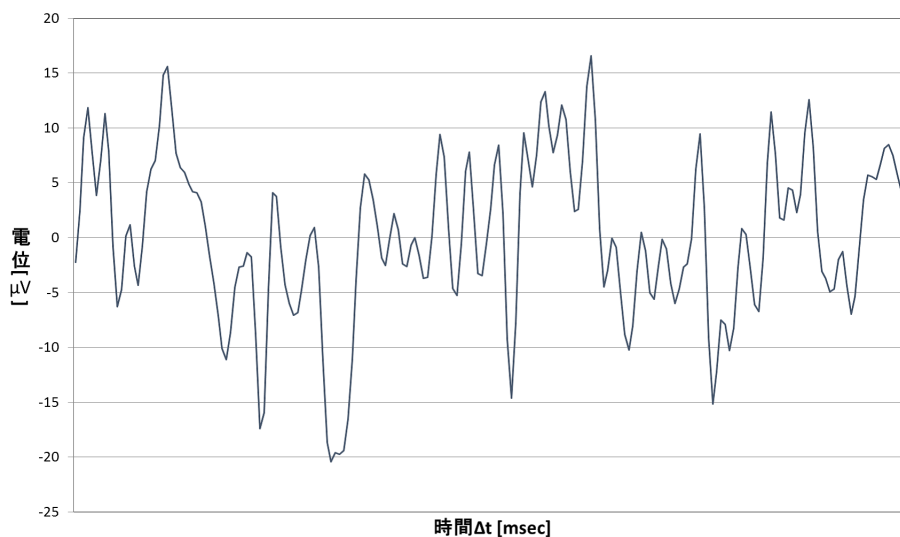


図 5.1 計測した脳波の例

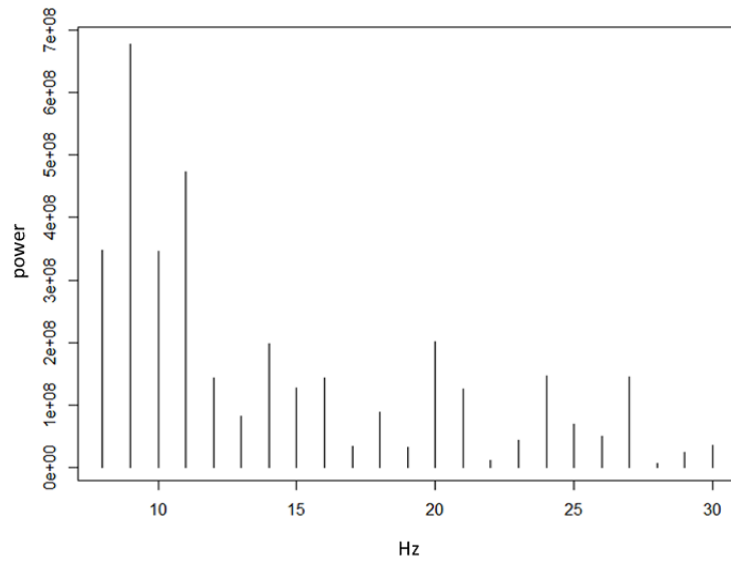


図 5.2 パワースペクトルの例

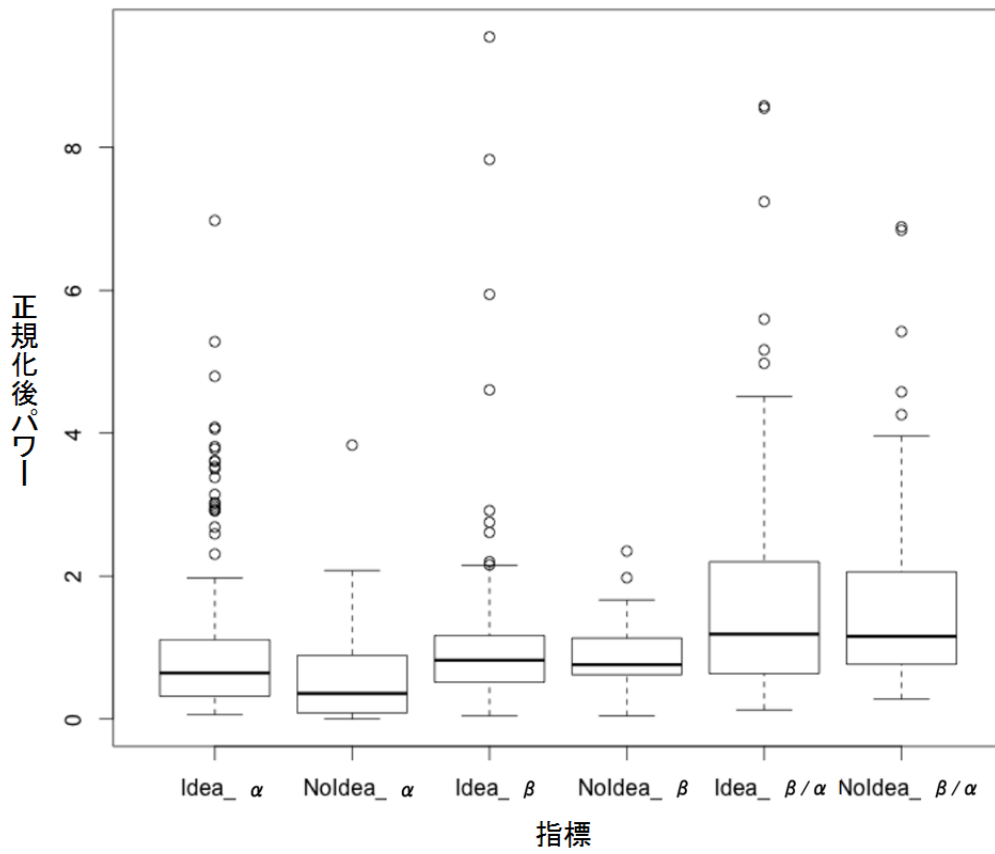


図 5.3 タスク中の各指標の比較

箱の中央付近の横線はデータの中央値を表し、箱の横線は下側がデータの第1四分位数、上側が第3四分位数を表す。箱の上下の短い横線は下側がデータの最小値、上側がデータの最大値を表す。箱の外側にある点は外れ値である。図5.3から、すべての指標で外れ値が多いものの、*Success*と*Fail*の α は中央値が0.661と0.647で*Success*の方が大きいことが読み取れる。また、平均値も1.061と0.759といずれも*Success*の方が大きい。同様に β についても中央値で0.823と0.760、平均値で1.032と0.874といずれも*Success*の方が大きかった。 β/α については中央値が1.187と1.157、平均値が1.616と2.366となり、中央値では*Success*の方が大きかったが平均値は*Fail*の方が大きかった。

3つの指標について、それぞれF検定で分散を調べ、その結果に基づいて各指標の*Success*と*Fail*との差に対してt検定を行った。その結果、 α は $p = 0.049$ (Welchのt検定)、 β/α は $p = 0.035$ (Studentのt検定)で*Success*と*Fail*に有意差が見られた。 β は $p = 0.406$ (Studentのt検定)で有意差が見られなかった。

3章で述べた α 波の特徴から、*Success*のタスク中において被験者は*Fail*のタスクに比べて、リラックスした状態で集中してタスクに取り組んでいたのではないかと考えられる。これは、実装戦略を推定できたかによって生じる被験者の心理的変容が脳波に反映されていることを示唆する。

また、実環境においてプログラマの状態の識別を60秒以内に行うことに重要な意味はない。本実験の結果からいえることは、タスクを開始して60秒程度で、プログラマ自身の中でプログラムの実装戦略を推定する際に、心理的変容が起こり、その影響が即時に脳波に現れていることが明らかになった点である。以上から、タスク中に脳波計測を行うと、プログラマの思考状態を即時に把握できるので、理解が不十分なプログラマへの迅速な支援につながると考えられる。

そして、特にタスク中の脳波に含まれる α の大きさがプログラムの実装戦略を推定できたかに関係があったと考えられ、 α や β/α が指標として有用である可能性を示唆する。以上より、プログラミングのタスク中の脳波を計測すると、計測した脳波に含まれる α によってプログラマの状態を把握し識別できる可能性が示された。

5.1.2 タスク後の脳波

*Success*と*Fail*の2グループについて、タスク後の α 、 β 、 β/α の値を図5.4の箱ひげ図に示す。*Success*と*Fail*の α は中央値が0.845と0.047、平均値が1.157と0.377といずれも*Success*の方が大きかった。 β も同様に、中央値で0.559と0.453、平均値で1.083と0.668といずれも*Success*の方が大きかった。 β/α については中央値が1.007と4.803、平均値が3.344と14.960といずれも*Fail*の方が大きかった。

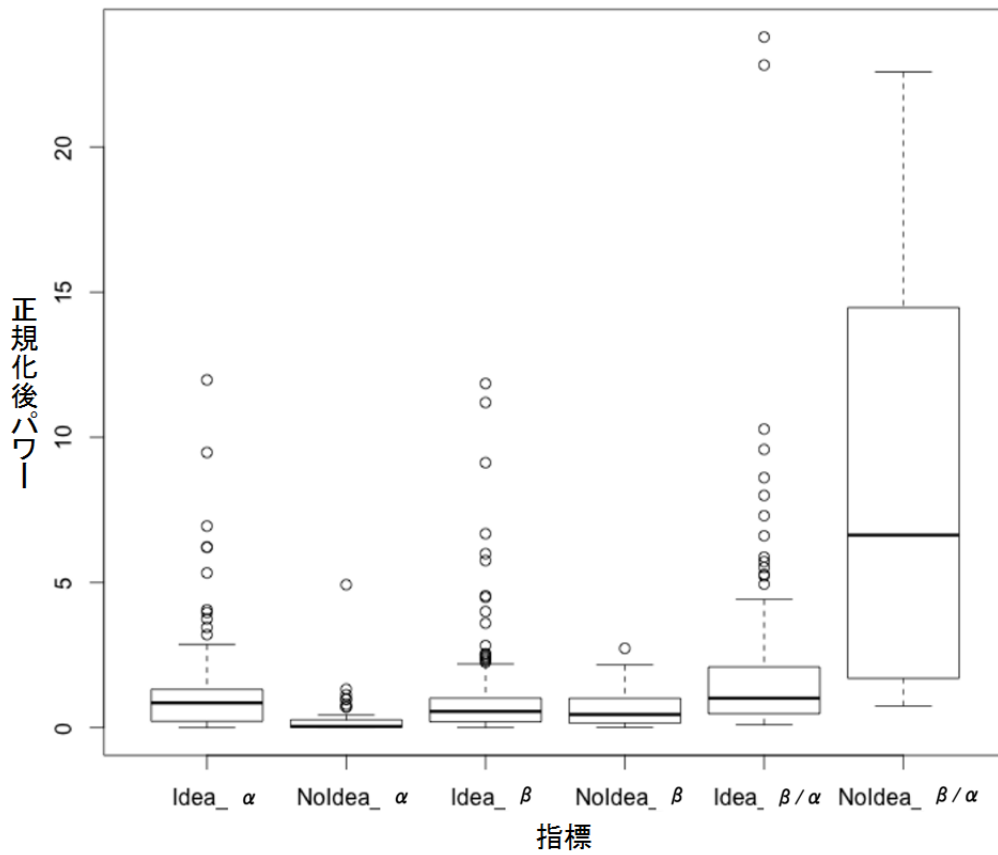


図 5.4 タスク後の各指標の比較

3つの指標について、それぞれF検定で分散を調べ、その結果に基づいて各指標の *Success* と *Fail* との差に対してt検定を行った。その結果、 α は $p = 0.003$ (Studente のt検定)、 β/α は $p = 0.002$ (Student のt検定) で *Success* と *Fail* に有意差が見られた。ただし β/α では、17件中1件のデータで含まれていた他の10倍程度の大きな値が、結果に影響を及ぼしていると考えられる。この値を除外して再度検定した結果、 $p = 0.343$ (Welch のt検定) と有意差は見られなかった。 β は $p = 0.147$ (Student のt検定) で有意差が見られなかった。

5.1.1節と同様に、3章で述べた α 波の特徴から *Success* のタスク後は *Fail* のタスク後に比べて、被験者はリラックスした状態だったと考えられる。これは、タスクの影響によって生じたタスク中の心理的変容がタスク後まで継続して脳波に反映されていることを示唆する。

そして、特にタスク後の脳波に含まれる α の大きさがプログラムの実装戦略を推定できたかとの関係があったと考えられ、 α が指標として有用である可能性が示された。以上より、プログラミングのタスク後の脳波を計測すると、計測した脳波に含まれる α によってプログラムの状態を把握し識別できる可能性が示された。また、タスク後の脳波にはアーチファクトが少なく α 波減衰も起こらないこ

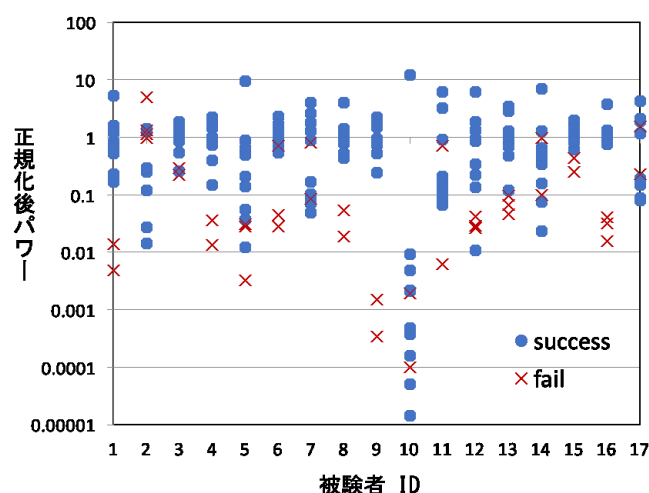


図 5.5 各被験者の α

とから、タスク中の脳波と比べてプログラムの実装戦略を推定できたかどうかを識別するための指標としてより適切であると考えられる。一方で、タスク後の脳波を用いた分析を行う場合、本実験の実験設定のように安静状態が必要になり、計測に必要な時間も長くなる。今後、教育現場の学生や企業の新人研修などでの利用が簡易になるよう、タスク後とタスク中の脳波それぞれを用いた分析が可能になることが望ましい。また、タスク中、タスク後いずれの結果も3つの指標全てにおいて外れ値の多さが目立った。原因の1つとして被験者の個人差が考えられるが、これについては5.1.3節で考察する。

5.1.3 各個人の成分の比較

5.1.2節で述べたとおり、本実験の結果は個人差による影響を受けた。図5.5は *Success* のタスク後における各被験者の α の値を示す。x軸は被験者17人のID(1~17)を示し、y軸は正規化後の α の大きさを表す対数軸である。

図5.5から計測した脳波にはかなり大きな個人差が含まれていたと読み取れる。例えば、図5.5で被験者10の多くの値が0.01未満になっており、他の被験者と比べて極端に値が小さい。脳計測を行っている研究の多くで、一部の被験者で極端に値が異なる場合や、傾向が反対になる場合が報告されている[16]。本研究の結果も個人差によって受けた影響を考慮し、タスク後の各被験者について *Success* と *Fail* の差を分析する。

表5.1に各被験者ごとの *Success* と *Fail* における各成分の値を示す。各成分の値はいずれも中央値で、p値はt検定の結果を示す。また、有意差を確認したp値が0.05未満のものにはアスタリスクを1つ、p値が0.01未満のものにはアスタリスクを2つ付す。被験者17人中12人で *Success* の α が大きく、9人で有意差が見られた。同様

表 5.1 各個人の成分の比較

被験者	α			β			β/α		
	Success	Fail	p 値	Success	Fail	p 値	Success	Fail	p 値
1	0.754	0.009	0.036*	0.579	0.027	0.756	1.104	3.722	0.001**
2	0.270	1.230	0.182	0.686	1.499	0.048*	2.496	1.096	0.244
3	1.186	0.256	0.000**	0.464	3.563	0.299	0.349	13.30	0.003**
4	1.212	0.025	0.000**	0.330	0.206	0.647	0.613	7.156	0.000**
5	0.206	0.028	0.492	0.195	0.046	0.117	2.004	1.656	0.670
6	1.071	0.044	0.049*	0.591	0.077	0.091	0.944	1.488	0.903
7	0.529	0.442	0.520	0.781	0.618	0.338	2.860	4.129	0.991
8	0.907	0.036	0.006**	0.803	0.370	0.076	0.844	13.49	0.001**
9	1.219	0.001	0.029*	0.066	0.033	0.361	0.079	40.94	0.001**
10	0.001	0.001	0.676	0.004	0.005	0.339	7.306	19.66	0.910
11	0.176	0.357	0.617	0.709	0.987	0.979	3.501	24.32	0.025*
12	0.856	0.029	0.290	0.296	0.184	0.546	0.632	5.974	0.546
13	0.941	0.067	0.010*	0.418	1.164	0.987	0.755	12.14	0.000**
14	0.442	0.541	0.511	0.732	0.811	0.809	1.469	3.992	0.807
15	1.101	0.344	0.029*	0.800	1.231	0.783	0.913	3.206	0.003**
16	1.073	0.031	0.041*	1.024	0.508	0.050	0.633	16.22	0.024*
17	0.187	0.891	0.812	0.207	0.786	0.812	1.026	1.170	0.812

に、 β は17人中6人で*Fail*の値が大きく、1人で有意差が見られた。 β/α は17人中15人で*Fail*の値が大きく、9人で有意差が見られた。

以上の結果は、従来研究と同様に、脳波は個人差が大きいものの、個人内では安定しており、プログラムの実装戦略を推定できたか否かをみる手法として有用である可能性を示してる。個人差を考慮した、分析方法を確立することは本研究の興味深い発展のひとつである。

5.2 実験2

正規化後の α 波の全被験者の平均値を1秒毎に求め、それぞれ $S\alpha$ と $F\alpha$ とした結果を図5.6に示す。t検定によって有意差($p < 0.05$)を確認できた所に緑のマーカーを付す。

図5.6より、タスク後半にかけて $S\alpha$ が $F\alpha$ に比べて高いことがわかった。そして有意差が得られた点は11点あった。しかし、それらの点は離散しており、タスクの影響が強く現れる時間帯を示すことはできていない。その要因の1つは、脳波計測に多く含まれている筋電位から生じるノイズの影響が考えられる。平均値の算出時間幅を伸ばすことで、このノイズの影響を抑えた脳波から $S\alpha$ と $F\alpha$ の傾向をよりの確に得ることが望ましい。

そこで、正規化後の α の全被験者の平均値を移動平均を用いて5秒毎と10秒毎

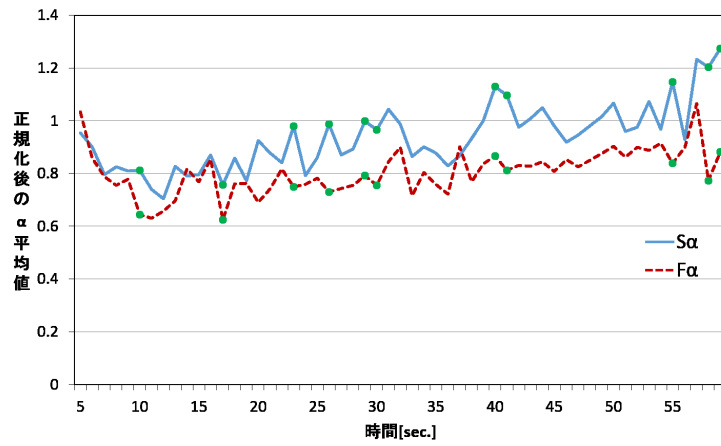


図 5.6 1 秒毎の α 平均値

で求め、それぞれ $S\alpha$ と $F\alpha$ とした結果を図 5.7 と図 5.8 に示す。t 検定によって有意差 ($p < 0.05$) を確認できた所に緑のマーカーを付す。

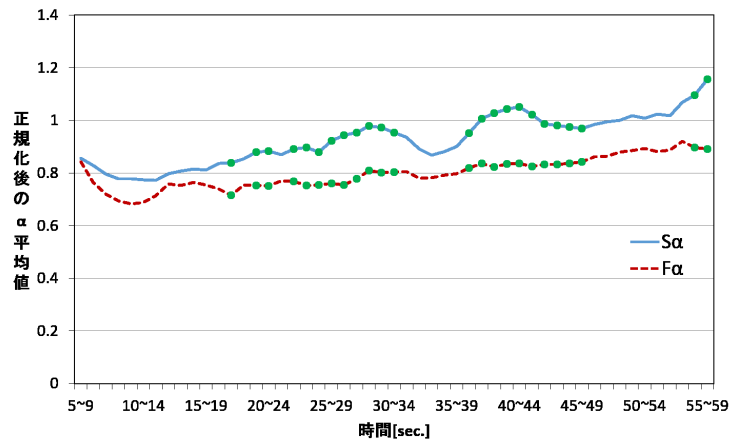


図 5.7 5 秒毎の α 平均値

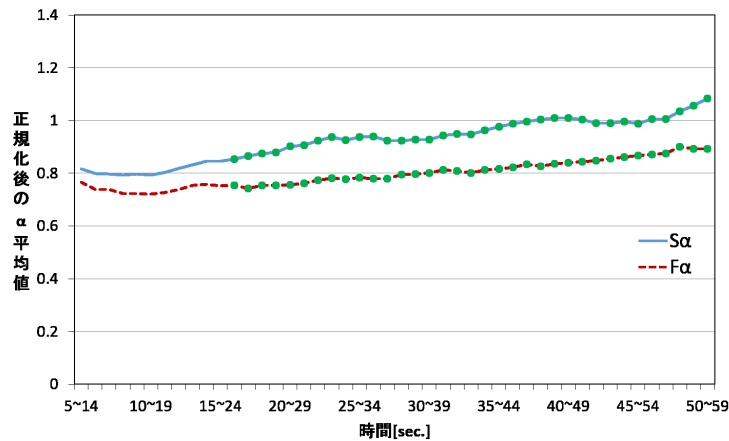


図 5.8 10 秒毎の α 平均値

図 5.7, 図 5.8 とともに, 図 5.6 と同様にタスク後半にかけて $S\alpha$ が $F\alpha$ に比べて高い結果を示している. そして 5 秒毎に平均値を算出した場合, 有意差が得られた点は 24 点あった. 有意差が確認できた所は, 17-21~30-34, 36-40~45-49, 54-58~55-59 の時間帯だった. 1 秒毎にみたときに比べて複数の連続した時間帯で有意差が得られた.

そして 10 秒毎に平均値を算出した場合は, 有意差が得られた点は 35 点あった. 有意差が確認できた所は, 16-25~50-59 の時間帯だった. 計測開始 16 秒以降から計測終了まで連続した時間帯で有意差が得られた. この結果は, 計測開始 16 秒から計測終了 59 秒の間, タスクの影響が $S\alpha$ や $F\alpha$ に表れることを示唆している.

1 秒毎, 5 秒毎, 10 秒毎に平均値を算出した結果, タスクの影響が強く現れる時間帯は, 計測開始 16 秒から計測終了の時間帯に現れると考えられる. したがって, タスク開始直後から約 16 秒を分析対象から外すことで分析時間の短縮になるといえる. また以上の結果より, 平均値の算出時間幅を伸ばすことで, ノイズの影響を抑えた $S\alpha$ と $F\alpha$ の傾向を得られるが, ノイズと同時に各被験者の特徴も失っていることが危惧される. そのため, 平均値を算出する時間は 5~10 秒前後で求めることが望ましいといえる.

6 おわりに

本研究では、プログラマがプログラムの実装戦略を推定できたか否かを識別する方法として、脳波に着目し、2つの実験を行った。

実験1では、タスク中とタスク後に脳波を計測し、実装戦略を推定できたか否かで生じる周波数成分の差異を分析し、プログラマの状態の識別方法として脳波の有用性を検証した。識別指標として8~13Hzの帯域である α 波と14~30Hzの帯域である β 波、 α 波と β 波の大きさの比である β/α を用いた。実験1の結果、タスク中の計測結果では、実装戦略を推定できたタスクで α と β/α が有意に大きかった。タスク後の計測結果では、実装戦略を推定できたタスクで α と β/α が有意に大きかったが、 β/α は他のデータと極端に値の差が大きい1件のデータを除くと有意差が見られなかった。3つの指標のうち、タスク中、タスク後いずれの計測結果でも実装戦略を推定できたタスクで比較的大きく有意差が見られた α 波がプログラマの状態の識別に最も適しているといえる。計測した脳波に含まれる α 波の大きさによって、プログラマの状態の識別が可能になると検証された。

また、脳波の波形には個人差が見られた。各被験者ごとに実装戦略を推定できたときと、そうでないときとの脳波を比較した結果、被験者17人中12人で*Success*の α が大きく、9人で有意差が見られた。従来研究と同様に、脳波は個人差が大きいものの、個人内では安定しており、プログラムの実装戦略を推定できたか否かをみる手法として有用である可能性を示すことができた。すなわち、個人差の影響はあるものの、 α 波の大きさはプログラマの状態の識別指標として有用である。実験1の結果から、プログラマの脳波に含まれる α 波の大きさを指標に用いると、プログラマの状態を識別できるといえる。

実験2では、タスク中の脳波を計測し、計測時間の中でも特にタスクの影響が強く現れる時間帯を明らかにするために時系列分析を行った。識別指標として α 波を用い、1秒毎、5秒毎、10秒毎で平均値の算出を行った。実験2の結果、タスクの影響は計測開始約16秒から計測終了の時間帯に強く現れると考えられる。よって、計測時間の中でも特に実装戦略を推定できたか否かの判断に適しているのは、計測開始16秒から計測終了までと明らかになった。そこで今後は計測開始16秒を分析対象から外し、計測開始16秒から計測終了までの分析をより詳細に行うことが望ましいと考えている。

本研究の結果からプログラマの状態を識別する方法として脳波計測の有用性が確かめられた。プログラマの状態を識別する指標として脳波を用いることは、迅速なプログラマの状態の把握が可能になり、適切な支援によって作業効率や教育効果を高められると考えられる。例えば本研究の結果を実際の教育現場で用いる場合では、学習者ごとに課題を開始するタイミングが異なることに加え、課題ごとに実装戦略の推定に必要な時間が異なるため、特定の時間帯を分析に適

した領域として示すのは難しい。そのため、実際の教育現場では、IDE等を用いてソースコードの入力を開始した時点を実験におけるタスクの完了時点と見なし、その前後の脳波を分析することで、学習者が実装戦略を推定した上でソースコードを書き始めたのか識別できると考えられる。脳波計測から実装戦略を推定できたかどうかを区別することで、支援が不要な学習者の作業を妨げることなく、実装戦略を思いついていないまま実装を始めた学習者に対して優先的に支援が可能になる。

本研究ではプログラマの状態の識別に脳波計測が有用であると示すことができた。本研究で得た結果をさらに発展させると、機械学習を用いて脳波に含まれる α 波の大きさによってプログラマの状態の識別が可能になるだろう。

7 謝辞

本論文の執筆および研究をすすめるに当たり、多くの方々に協力して頂きました。この研究を卒業論文として形にすることが出来たのは、指導教員である上野秀剛准教授の熱心なご指導のおかげです。ありがとうございました。また、日常の議論を通じて多くの知識や示唆を頂いた上野研究室の皆様や、貴重な時間を割いて被験者実験に参加して下さった23名の学生の皆様へも心から感謝の気持ちと御礼を申し上げます。

参考文献

- [1] 上野秀剛, 中村匡秀, 門田暁人, 松本健一:”プログラムの視線を用いたコードレビュー性能の要因分析”; ソフトウェア工学の基礎 XIII 日本ソフトウェア科学会 FOSE2006, pp.103-112 (2006).
- [2] 宮田洋, 藤澤清, 柿木昇治, 山崎勝男:”新生理心理学生理心理学の基礎 ”; 北大路書房 (1998).
- [3] 満倉靖恵:”脳はウソをつかない—脳波で判るあなたの真実—”; 日本耳鼻咽喉科学会会報, Vol.118, No.4, pp.461-465 (2015).
- [4] J.Siegmund: Program comprehension: Past, present, and future; IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER), Vol.5, pp.13-20 (2016).
- [5] M. A. Storey. Theories, methods and tools in program comprehension: Past, present, and future. In 13th International Workshop on Program Comprehension (IWPC'05), pp.181-191 (2005).
- [6] C.Parnin. A cognitive neuroscience perspective on memory for programming tasks. In Proceedings of 22nd Annual Meeting of the Psychology of Programming Interest Group (PPIG) (2010).
- [7] M.Nakamura, A.Monden, T.Ito, K.Matsumoto, Y.Kanzaki, and H.Satoh: Queue-based cost evaluation of mental simulation process in program comprehension; In Proceeding of 9th IEEE International Software Metrics Symposium (METRICS 2003), pp.351-360 (2003).
- [8] 石黒誉久, 井垣宏, 中村匡秀, 門田暁人, 松本健一:”変数更新の回数と分散に基づくプログラムのメンタルシミュレーションコスト評価”; 電子情報通信学会技術報告, ソフトウェアサイエンス研究会, 第SS2004-32巻, pp.37-42 (2004).
- [9] Ericsson, K. Anders, Simon, Herbert A.: Verbal reports as data; Psychological review, Vol.87, No.3 (1980).
- [10] A.Karahasanovic, U. N. Hinkel, D. I. K. Sjoberg, and R. Thomas: Comparing of feedback-collection and think-aloud methods in program comprehension studies; Behaviour Information Technology, Vol.28, No.2, pp.139-164 (2009).
- [11] A.Karahasanovic, A.K. Levine, and R.Thomas: Comprehension strategies and difficulties in maintaining object-oriented systems: An explorative study; Journal of Systems and Software, Vol.80, No.9, pp.1541-1559 (2007).
- [12] 栗山進, 大平雅雄, 門田暁人, 松本健一:”プログラム理解度がコードレビュー達成度に及ぼす影響の分析”; 電子情報通信学会技術研究報告.SS, ソフトウェアサイエンス, Vol.104, No.571, pp.17-22 (2005).

- [13] A.Dunsmore:Expert programming knowledge:a schema-based approach;Psychology of programming,pp.205-222 (1990).
- [14] 村岸 巖:"皮膚抵抗値によるソフトウェア開発者の負荷評価に関する研究";Master's thesis, 奈良先端科学技術大学院大学 (1998).
- [15] J.Siegmund,A.Brechmann,J.Apel,C.Kastner,J.Liebig,T.Leich and G.Saake:Toward Measuring Program Comprehension with Functional Magnetic Resonance Imaging;Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering (FSE'12),No.24 (2012).
- [16] 中川 尊雄, 亀井 靖高, 上野 秀剛, 門田 暁人, 松本 健一:"脳血流計測に基づくプログラム理解行動の定量化";ソフトウェア工学の基礎XIV(ソフトウェア工学の基礎ワークショップFOSE2013),pp.191-196 (2013).
- [17] T.Fritz,A.Begel,Sebastian C Muller,S.Yigit-Elliott and M.Zuger:Using psycho-physiological measures to assess task difficulty in software development;In Proceedings of the 36th International Conference on Software Engineering (ICSE 2014),pp.402-413 (2014).
- [18] M.Zuger and T.Fritz:Interruptibility of software developers and its prediction using psychophysiological sensors;In Proceedings of 33rd Annual ACM Conference on Human Factors in Computing Systems,pp.2981-2990 (2015).
- [19] S.C.Muller and T.Fritz:Stuck and frustrated or in flow and happy:Sensing developers' emotions and progress;In 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering,Vol.1,pp.688-699 (2015).
- [20] 上野 秀剛, 石田 響子, 松田 侑子, 福嶋 祥太, 中道上, 大平 雅雄, 松本 健一, 岡田 保紀:"脳波を利用したソフトウェアユーザビリティの評価ー異なるバージョン間における周波数成分の比較";ヒューマンインタフェース学会論文誌,Vol.10,No.2,pp.233-242 (2008).
- [21] 森田 尚亨:"脳波コヒーレンスを用いた脳活動の推定について-ワーキングメモリ課題遂行中の脳活動の推定-";高知学園短期大学紀要, Vol.42, pp.13-26 (2012)
- [22] 音茂 龍司, 辻 貞敏:"よくわかる脳波判読第3版";金原出版 (2015).
- [23] 村上 郁也:"イラストレクチャー認知神経科学";オーム社 (2013).
- [24] 堀 忠雄:"生理心理学-人間の行動を生理指標で測る";培風館 (2008).
- [25] 大橋 力:"マルチメディアと脳";電子情報通信学会論文誌.D-II, 情報・システム,II-情報処理,Vol.79,No.4,pp.468-475 (1996).