

# プログラム理解時における脳波特徴の把握を目的とした 時系列分析の試み

石田 豊実<sup>1,a)</sup> 上野 秀剛<sup>2,b)</sup>

**概要:** ソフトウェア開発作業の多くを占めるデバッグを効率化するためには、デバッグ中のプログラマの状態を把握し、適切な支援を行うことが重要である。プログラマの内的状態を外部から把握するための方法として視線や脳活動といった生体活動に着目した手法がある。本研究では、ソースコードを読むプログラマを対象に脳波を計測し、プログラムの動作内容を理解したか、また、バグの有無を判断できたか識別する。特に、プログラマへの迅速で精度の高い支援を行うためには、リアルタイムでの状態識別が必要なため、脳波の時系列変化を理解し、リアルタイムでの状態識別に必要な特徴を捉えることが重要である。そこで本研究では脳波の時系列変化を分析する事を試みる。実験ではプログラム動作の理解とバグ判断の2つのフェーズについて、理解・判断の成否で脳波の特徴に違いがあるか検証する。

ISHIDA TOYOMI<sup>1,a)</sup> UWANO HIDETAKE<sup>2,b)</sup>

## 1. はじめに

ソフトウェア開発作業の多くを占めるデバッグを効率化するためには、デバッグ中のプログラマの状態を把握し、適切な支援を行うことが重要である。しかしながらデバッグの作業は思考作業が多く含まれるため、プログラマの状態を外観から把握するのは難しい。プログラマの内的状態を外部から把握するための方法として視線や脳活動といった生体活動に着目した手法がある。生体活動の1つである脳活動は思考状態や心理状態によって変化することが知られており、プログラム理解に関する研究においてもよく用いられている [1], [2]。脳活動を計測する手法の中でも脳波は時間分解能が高いことから作業中のプログラマの状態をリアルタイムに識別し、適切な支援を行う手法に適していると考えられる。

デバッグ作業でバグを見つけるためにはプログラムの動作と共に仕様を理解し、仕様通りに動作しているかを判断する必要がある。プログラムの動作を理解するためにはソースコードから処理の流れや各変数、メソッドの役割などを読み取り、プログラム全体の動作として解釈する必要

がある。したがって、ソースコードの個々の行に対する理解と、複数行からなるメソッドやクラス、プログラム全体といったより大きな処理への統合と、それに伴う抽象的なレベルでの処理内容の理解が必要と考えられる。一方で、バグを検出するためには理解したプログラム動作に処理上の誤り（たとえば、初期化無しでのデータ使用や配列範囲外へのアクセス）が含まれていないか確認すると共に、プログラムの動作が仕様を満たしているか判断する必要がある。したがって、自然言語で記述された仕様が意味している内容と、ソースコードとして記述されたプログラムの動作を比較し、一致しているか判断する必要があると考えられる。本研究ではデバッグを 1) プログラムの動作理解と、2) バグ判断の2つの作業からなる活動ととらえ、それぞれの作業を区別して分析する。

また、本稿では脳波の時系列変化について分析する。脳波は思考作業の状態によって変化する一方で、その状態に至るためにプログラムをどのような手順で理解しているのかを知ることができない。そこで本研究では、プログラム理解時の視線を同時に計測することで、プログラムをどのような順に読み、どのような思考状態になったのかを分析する事を目指す。視線計測はある作業の熟練者や初心者を対象に作業過程を分析するために良く用いられており [3]、プログラミングを対象とした研究も複数行われている [4], [5], [6]。デバッグ作業中の開発者の脳波を時系列に

<sup>1</sup> 奈良工業高等専門学校 システム創成工学専攻 情報システムコース

<sup>2</sup> 奈良工業高等専門学校情報工学科

<sup>a)</sup> ishida@info.nara-k.ac.jp

<sup>b)</sup> uwano@info.nara-k.ac.jp

分析する事で、脳波の特徴が変化した前後にプログラムのどこを見ていたかを視線計測の結果から知ることができ、プログラムの理解過程をより詳細に分析する事ができると考えられる。

本稿では、脳波と視線を組み合わせて分析することを目的として、デバッグ時の脳波を時系列分析する。デバッグ作業をプログラムの動作理解とバグ判断の2つの作業からなる活動ととらえ、それぞれの開始時と終了時に脳波に変化があるか実験で調べる。また、より詳細な分析としてタスク時間を5秒ごとに区切って時系列の変化を分析する。

## 2. 関連研究

プログラム理解の過程を定量的に評価する方法の1つとして、生体情報を用いた手法が提案されている。生体情報の計測は一般に、アンケートやインタビューと比べて時間分解能が高く、定量的な計測が可能であることから、プログラム理解の計測に適していると考えられる。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 [7]。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。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分かかるとも報告されている。

これに対して、本研究でも用いる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ものの、脳が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とされた知的活動の量と直接関係のない精神的負荷がノイズとして計測値に現れにくいことを意味する。近年では、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、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。

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 [8][1]。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、および文章理解に係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。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。

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 [9]。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、さらなる調査が必要な状態であると言える。また、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 [5]。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 [6]。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을上げていることが確認できる。

これらの研究結果はいずれも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。本研究ではプログラムのデバッグ作業を対象に脳波の時系列分析を行い、プログラム理解の有無やバグ判断の正否によって周波数成分に違いがみられるか実験で検証する。

## 3. 脳波と視線

### 3.1 脳波

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 [10]。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1に示す国際式10-20電極法に則って行う [11]。

国際式10-20電極法では耳のアースを除き19箇所の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

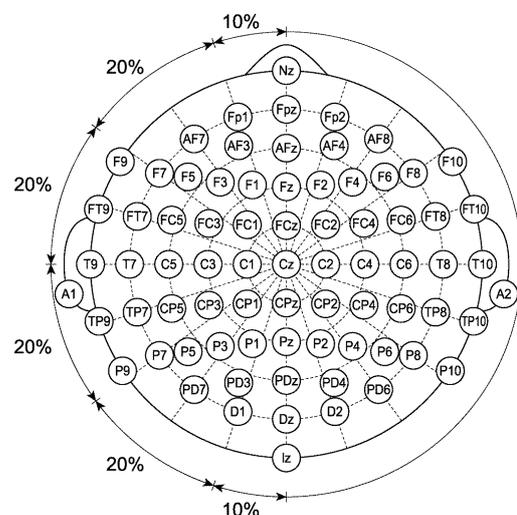


図1 国際式10-20電極法

する電極を決定する。また、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種類の方法があり、目的に応じて適切な導出法を選択する必要がある[12]。一般的に脳波は時間の経過とともに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[13]。各帯域の名称と周波数帯域を以下に示す。

- $\delta$  波: 0.5~4Hz 未満
- $\theta$  波: 4~8Hz 未満
- $\alpha$  波: 8~14Hz 未満
- $\beta$  波: 14~30Hz 未満
- $\gamma$  波: 30Hz 以上

$\delta$  波や  $\theta$  波は睡眠状態にあるときに出現する。 $\alpha$  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、 $\alpha$  波の振幅が低下して不連続になる。また、 $\alpha$  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  $\alpha$  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  $\beta$  波が出現する。 $\gamma$  波は、不安で興奮しているときに出現する[12][13]。これらの周波数帯域の内、 $\alpha$  波と  $\beta$  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[13]、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[14]。

本研究では知的活動の一種であるデバッグにおける作業者の状態を脳波から定量的に分析する。デバッグ作業中の作業者は作業の進捗によって脳活動が変化すると考えられ、周波数成分の違いから作業の成否が判断できる可能性がある。特に、本稿では視線を同時に計測することを目的に時系列での周波数成分の変化を分析する。

### 3.2 視線

視線とは、眼球の動きから画面上の点を見ている点を算出し、それを座標として記録したものである[15]。計測領域上で注視した座標が時系列順に記録され、その座標の移動順序や、停留時間などから被験者が何を見ているか分析できる。視線移動の計測は初心者と熟練者の違いを分析することを目的に、特に認知工学の分野において用いられている[16][3]。Lawらは腹腔鏡手術の訓練装置を使用している際の初心者と熟練者の視線の動きを分析し、熟練者は初心者に比べて視線が患部に集中することを明らかにしている[16]。村田らは自動車運転時の危険予知における熟練者と初心者の視線の動きを比較している[3]。実験の結果、危険度の高いカーブで熟練者と初心者の視線移動の違いが顕著になる一方で、直線道路や交通量の少ない道路では熟

練者と初心者の視線移動の違いが少なくなった。熟練者は初心者に比べて知識や経験が豊富であるため、作業を効率化する戦略を立てることに長けており、視線移動には戦略を反映した作業過程が反映されていると考えられる。したがって初心者と熟練者の視線移動を比較することで、効率の良い作業過程を明らかにし、教育や支援に有用な知見を得ることができる。

ソフトウェア工学の分野においても、デバッグやプログラム理解を対象とした視線移動の分析が行われている。Behrooziiらはホワイトボードを使ったプログラム理解の確認に視線計測を行う事でプログラマの状態を把握しようとしている[17]。また、Busjahnらは、初心者と熟練者には、ソースコードの読み方に違いがある事を視線計測を用いる事で明らかにしている[18]。自然言語で記述された仕様や設計、ソースコードからなるプログラムは行や段落、メソッドを単位とした視線移動の計測によって開発者の理解過程を高い時間分解能で定量的に分析することができる。本研究では今後、脳活動と視線移動を組み合わせた分析を行うために脳活動の時系列分析を行う。

### 3.3 脳波と視線の同時計測

デバッグには、変数定義の理解や、ソースコードのある行と他の行との関連の理解、複数行からなる処理の動作理解や仕様の理解など様々な過程があると考えられる。それぞれの過程は記憶力や計算能力、統合判断など異なる脳機能を必要とする作業であり、1つのデバッグタスクを行う中で時間の経過とともに実施する作業も変化すると考えられる。そのため、デバッグ中の脳活動を計測するにあたっては時系列の変化に着目することが有用である。また、各過程において作業者は各自の戦略に基づいた読み方を実施しており、その様子が視線移動に反映されていると考えられる。そのため、デバッグ時の作業者の脳波と視線を組み合わせることで、各過程に対する読み方と、その結果として現れる脳活動の変化を時系列にとらえることができる。

プログラム理解における脳波と視線を同時に計測した研究もこれまでに報告されている。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プログラマの状態の推定を試みている[9]。本研究では、デバッグ作業を構成する過程の変化やその成否を脳活動と視線移動から分析する。

## 4. 実験

日本語とJavaで記述されたプログラム課題を被験者に提示し、処理内容を理解する間の脳波を計測する。被験者は奈良工業高等専門学校(NTT)の学生5人で、年齢は19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。



図 2 装着時の様子

#### 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。図 2 に計測時の被験者の様子を示す。体動によるアーチファクトを抑えるために、ひじ掛け・足置きを備えた椅子に座り、頭部と体をできるだけ動かさないように指示する。

実験に使用するのは、脳波計測装置、タスク提示用 PC、脳波計測用 PC、記録用 PC である。脳波計測装置にはナノテックイメージ社製 NeXus-10 MARK II を用いる。本計測装置は計測周期は 256Hz であり、計測された脳波は Bluetooth 経由で脳波計測用 PC に転送され、CSV ファイル形式で出力される。

#### 4.2 タスク

タスクは被験者 1 人につき 16 問を与える。実験用に作成したタスク提示ツールを用いて被験者に日本語で記述されたプログラムの仕様と、対応する Java のソースコード 1 つを提示する。各タスクは動作理解ステップとバグ判断ステップで構成される。動作理解ステップでは被験者にプログラムの仕様とソースコードに加え、「6 行目が 2 回目に実行された時の a の値を教えてください」というように、ソースコードの動作を理解できているか確認するための質問を提示する。バグ判断ステップでは仕様とソースコードの動作が異なっているか（バグを含むか）判断してもらう。

本実験ではプログラムに対する理解の有無が同程度の件数計測できるように、タスクの難易度と制限時間を調整する。タスクの難易度は低難易度と高難易度をそれぞれ 8 件用意する。低難易度は main メソッドのみからなり、1 重の繰り返し文や条件分岐で構成された理解が容易と思われるソースコードを使用する。高難易度のソースコードは複数メソッドの使用や再帰構造を持ち、制限時間以内での理解が難しいと期待できる複雑なアルゴリズムを使用する。制限時間はそれぞれのステップごとに設定し、低難易度の理解には十分で、かつ高難易度の理解に不十分であることを予備実験によって事前に確認した。また、各タスクを提示する順番は、順序効果を考慮しカウンターバランスを行

表 1 実験に使用したタスクの一覧

	難易度	仕様
1	easy	階乗の計算
2		最大値検索
3		素数判定
4		中央値検索
5		累乗計算
6		2 つの数値の入れ替え
7		指定した文字列が含まれているか判定
8		文字列を反転させる
9	difficult	ハノイの塔
10		経路数を求める
11		順列を全列挙する
12		組み合わせを漸化式から求める
13		支払う硬貨の組み合わせを求める
14		文字列の組み合わせを求める
15		雲の軌道予測
16		最小公倍数と最大公約数を求める

う。表 1 にタスクの一覧を示す。

##### 4.2.1 プログラム動作理解ステップ

このステップでは、被験者にタスクを見せ、ソースコードの動作を理解しているか確認するための問いに口頭で回答してもらう。回答内容が事前に用意した解答と一致していればソースコードの動作を理解しているとみなす。回答が一致しないか、制限時間を超過した場合、動作を理解していないとみなす。回答の正誤は被験者に伝えず、制限時間は 2 分 30 秒とした。

##### 4.2.2 バグ判断ステップ

バグ判断ステップはプログラム動作理解ステップの完了直後に行う。バグ判断ステップでは、被験者にプログラム動作理解ステップで見せたソースコードの一部を改変したソースコードを見せる。被験者には改変後のソースコードが仕様を満たすかどうか判断し口頭で回答してもらう。回答の正誤は被験者に伝えず、制限時間は 1 分とした。改変部分を探す作業に時間がかからないよう、改変部分には色付けをして表示する。図 3 に改変前後のプログラムの例を示す。

#### 4.3 分析

プログラム動作理解ステップとバグ判断ステップそれぞれについて、タスク中の脳波を分析する。個々の脳波データを 0.2 秒ごとに分割し、短時間フーリエ変換 (STFT) を用いて  $\alpha$  波と  $\beta$  波の強さ (パワースペクトル) を求める。脳波の周波数成分は個人差が大きく、また、被験者によっては少数の極端に大きな値が計測される場合がある。本研究では抽出されたパワースペクトルについて、各被験者の各タスクにおける中央値で正規化し、タスク中の各時間帯においてパワースペクトルがどのように変化するか分析する。

仕様: 変数の階乗を計算し表示する.

<pre>public class Main {     public static void main(String arg[]) {         int a = 1, b = 4;          while (b &gt; 1) {             a = a * b;             b--;         }          System.out.println(a);     } }</pre>	<pre>public class Main {     public static void main(String arg[]) {         int a = 1, b = 4;          while (b &gt;= 1) {             a = a * b;             b--;         }          System.out.println(a);     } }</pre>
変更前	変更前

図 3 ソースコードの改変

正規化後のパワースペクトルについて 5 秒ごとに平均をとり、タスク中の時系列変化を求める。その後、各ステップについての正誤に基づいて 2 つのグループ (*success*, *failure*) に分け、各ステップのグループ間で比較を行う。プログラム動作の理解やバグ有無の判断を正しく行っている被験者は脳活動が活発な状態にあると考えられる。そのため、 $\alpha$  波や  $\beta$  波、またはその両方が大きくなると考えられる。

## 5. 結果と考察

### 5.1 タスク開始時と終了時の比較

図 4, 5 に各ステップにおけるタスク開始時と終了時の脳波を示す。図の左側は  $\alpha$  波、右側は  $\beta$  波を表しており、縦軸は正規化後のパワースペクトル、横軸は start がタスク開始直後の 5 秒間、end がタスク終了直前の 5 秒間を表す。値は各群に分類されたタスクの平均値を表す。

動作理解ステップ (図 4) の開始時と終了時を比べると *success*, *failure* の両方で  $\alpha$  波と  $\beta$  波ともに上昇しているが、*success* のほうがより大きく上昇している。開始時と

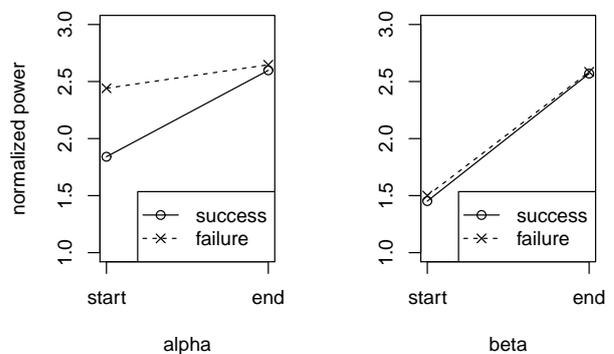


図 5 バグ判断ステップの脳波

終了時のパワースペクトルについて Welch の t 検定を行った結果、*success* の  $\alpha$  波 ( $p = 0.002$ ) と  $\beta$  波 ( $p < 0.001$ ) に有意な差が見られ、*failure* の  $\alpha$  波 ( $p = 0.385$ ) と  $\beta$  波 ( $p = 0.420$ ) には有意な差は見られなかった。これは、プログラムの動作理解が成功した時には脳活動が活発な状態になっているためと考えられる。理解の成功と脳活動の活発化の因果関係については明らかではないものの、脳波を計測することで動作理解の正否が判別できる可能性がある。

バグ判断ステップ (図 5) についても開始時と終了時を比べると *success*, *failure* の両方で  $\alpha$  波と  $\beta$  波ともに上昇しているが、*success* の  $\alpha$  がより大きく上昇しているのに対して、 $\beta$  波は *success*, *failure* の両方で大きく上昇している。検定の結果、*success* の  $\alpha$  波 ( $p = 0.005$ ) と  $\beta$  波 ( $p < 0.001$ )、*failure* の  $\beta$  波 ( $p = 0.003$ ) に有意な差が見られ、*failure* の  $\alpha$  波 ( $p = 0.617$ ) には有意な差は見られなかった。この結果は動作理解ステップと同様に脳波を計測することでバグ判断の正否が判別できる可能性を示している。一方で、動作理解ステップとは異なり、バグを正しく判断できなかった時に  $\beta$  波が有意に大きくなっている。

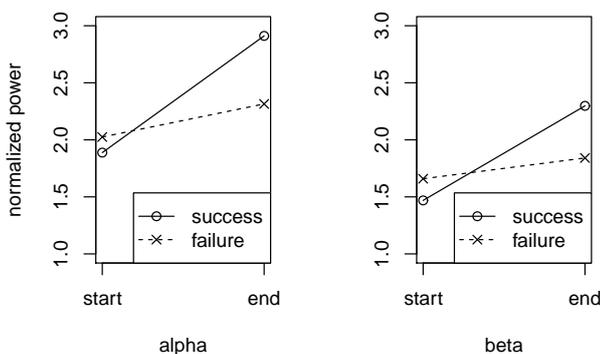


図 4 動作理解ステップの脳波

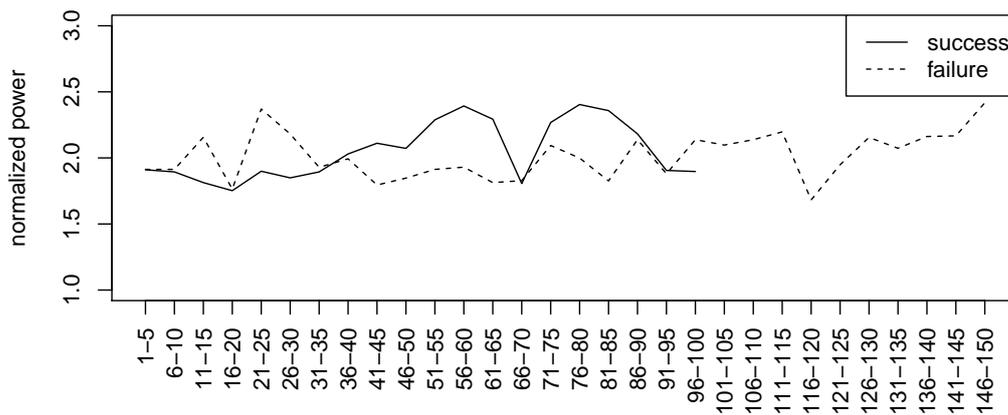


図 6 動作理解ステップの時系列変化

表 2 タスク終了時の脳波

ステップ	脳波帯域	success	fail	p 値
動作理解	alpha	2.91	2.31	0.060
	beta	2.30	1.84	0.089
バグ判断	alpha	2.60	2.65	0.883
	beta	2.57	2.58	0.970

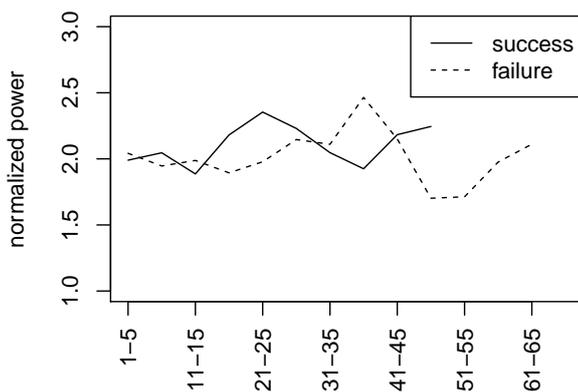


図 7 バグ判断ステップの時系列変化

プログラム動作理解はソースコードを読んで理解する作業であるのに対し、バグ判断はソースコードと仕様を比較してバグかどうか判断する作業である。そのため、脳活動もステップ間で異なると考えられ、その違いが  $\beta$  波に現れたものと考えられる。デバッグを構成する複数の作業ごとの脳活動の違いを明らかにすることは今後の課題である。

各ステップの成否によってステップ開始時と終了時の脳波に差が見られたが、ステップ終了時の脳波を比較することでも *success* と *failure* を区別できる可能性がある。表 2 にタスク終了時における *success* と *failure* の脳波を示す。動作理解ステップとバグ判断ステップのそれぞれにおいて、*success* と *failure* のタスク終了時を比較した結果、いずれのステップにおいても  $\alpha$  波と  $\beta$  波ともに有意差は見られなかった。図 4, 5 はいずれのステップにおいても成否にかかわらずパワースペクトルが大きくなっていることを示している。これは、動作理解やバグ有無の判断を行う事

によって成否に関係なく脳活動が活発化しており、ステップ終了時の脳活動の比較のみでは区別が付かない可能性を示唆している。タスク開始時と終了時の比較を行う事でタスクによる変化を捉えることができ、成否による差が現れていると考えられる。

## 5.2 時系列分析

本研究では脳波と同時に視線移動を計測し、組み合わせた分析を行う事を目的としている。そのため、デバッグ作業中の脳活動の変化をより詳細に分析する必要がある。本節では各ステップの *success* と *failure* それぞれについて、脳波の時系列変化を分析する。タスク開始時と終了時で脳波に差がある事が分かったため、その差がタスク開始から何秒で現れるのか調べるためにタスク開始から 5 秒の区間とそれ以降の区間で差の検定を行う。

図 6, 7 に各ステップにおける脳波の時系列変化を示す。縦軸はパワースペクトル、横軸はタスク開始からの時間 (5 秒間隔) を示す。なお、各被験者のタスクごとにタスクの終了時間が異なっているため、時間区間が後半になるほどデータ数が少なくなる。本稿では、各時間区間についてデータ件数が 5 件以下のものを分析から除外する。動作理解ステップ (図 6) の *success* は後半になるほど値が大きくなる傾向が見られた。Welch の *t* 検定の結果、41 秒以降のすべての区間において 1-5 秒の区間と有意差 ( $p < 0.05$ ) が見られた。*failure* も同様に後半になるほど値が大きくなる傾向が見られたが、*success* と比較すると値が低かった。

*failure*においても106秒以降のすべての区間において1-5秒の区間と有意差 ( $p < 0.05$ ) が見られた。この結果はプログラムの動作理解に成功した時には、タスク開始から早い段階で脳活動が活発になっていることを示しており、作業途中であっても判別が可能であることを示唆している。

バグ判断ステップ (図7) においても動作理解ステップと同様に *success* が後半になるほど値が大きくなる傾向が見られた。Welch の t 検定の結果、11秒以降のすべての区間において1-5秒の区間と有意差 ( $p < 0.05$ ) が見られた。*failure* は一度値が大きくなった後に、開始時点よりも値が低くなっており、41秒以降のすべての区間において1-5秒の区間と有意差 ( $p < 0.05$ ) が見られた。この結果は動作理解と同様にバグ有無の判断に成功した事をタスク開始から早い段階で判別できる可能性を示している。また、バグを正しく判断できていない時には、脳活動が不活発になっている可能性がある。時系列における区間ごとの特徴の分析は今後の課題である。

なお、図4に示した分析においては *failure* の開始時と終了時で  $\alpha$  波には有意差は見られず、図6の結果と異なっている。図4では各被験者のタスクごとの開始直後の5秒と、終了直前の5秒のパワースペクトルを比較している。そのため、*success* や *failure* それぞれについてタスクに対する脳活動の状態が均一であったと考えられる。一方で、本節の時系列分析では時間ごとに比較しているため、各時間区間にはタスクを終える直前の脳活動やタスク中といった異なる状態のデータが含まれていると考えられる。今後、時系列の分析を進めるためには、被験者やタスクの時間区間ごとに異なる活動を考慮した分析を行う必要がある。

## 6. おわりに

本稿ではデバッグを行うプログラムの理解過程を時系列に計測することを目的に、脳波特徴の時系列分析を試みた。仕様とソースコードを対象にした動作理解ステップとバグ判断ステップからなる被験者実験を行い、各ステップの正否ごとに周波数成分の変化を分析した。実験の結果、動作理解ステップを正しく終えた被験者は開始時と終了時の  $\alpha$  波と  $\beta$  波が有意に高く、バグ判断ステップを正しく終えた被験者は開始時と終了時の  $\alpha$  波が有意に高かった。各ステップ開始からの5秒ごとの脳波を見るとステップの正否にかかわらず後半の  $\alpha$  波が有意に高くなり、ステップを正しく終えた被験者群はより早い時点で有意に高くなる傾向が見られた。この結果はプログラムの動作理解やバグ判断の正否が脳波から求めた周波数成分に反映されており、時系列の変化を見ることで成否を判別できる可能性を示唆している。

本研究の今後の課題として、脳波データごとの時間の扱いについて考慮する必要がある。各タスクのステップは被験者によって異なる時間で終了するため、ステップの正否

ごとに時系列のグラフを作成する際に後半のデータ数が減少する。加えて、短時間でステップを終えた被験者と長時間を要した被験者で各時間区間で行っている作業が異なっている恐れがある。たとえば、30秒で動作理解を終えた被験者がある時間区間 (たとえば10-15秒) で変数の役割を理解しているのに対して、60秒で動作理解を終えた被験者は同じ時間区間に (10-15秒) メソッドの呼び出し関係を確認している可能性がある。今後、タスク開始から終了までの時間を割合で区切るなど、できるだけ同じ作業を行っている区間を抽出する事でより精度の高い分析方法を検討する。

また、本稿の実験では動作理解ステップの後にバグ判断ステップを行っているため、結果について順序効果が影響している可能性がある。今後の分析では動作理解ステップの影響を考慮した正規化をバグ判断ステップに行うことで、順序効果の影響を排除した分析が可能になると考えられる。

謝辞 本研究はJSPS 科研費 JP16K00114 の助成を受けたものです。

## 参考文献

- [1] J. Siegmund, N. Peitek, C. Parnin, S. Apel, J. Hofmeister, C. Kastner, A. Begel, A. Bethmann, and A. Brechmann: "Measuring Neural Efficiency of Program Comprehension", In Proceedings of the 11th Joint Meeting on Foundations of Software Engineering (ESEC/FSE), pp.140-150 (2017).
- [2] 中川 尊雄, 亀井 靖高, 上野 秀剛, 門田 暁人, 鶴林 尚靖, 松本 健一: "脳活動に基づくプログラム理解の困難さ測定", コンピュータソフトウェア, Vol.33, No.2, pp.78-89, (2016).
- [3] 村田 厚生, 森若 誠: "危険予知課題における運転者の視覚情報処理特性: 運転初心者と運転熟練者の比較", 人間工学, Vol.46, No.6, pp.393-397 (2010).
- [4] P. Rodeghero, C. Liu, P. W. McBurney and C. McMillan: "An Eye-Tracking Study of Java Programmers and Application to Source Code Summarization," in IEEE Transactions on Software Engineering, Vol.41, No.11, pp.1038-1054 (2015).
- [5] Zuger.M and Fritz.T: "Interruptibility of software developers and its prediction using psycho-physiological sensors", In Proceedings of 33rd Annual ACM Conference on Human Factors in Computing Systems, pp.2981-2990 (2015).
- [6] Muller.S.C and Fritz.T: "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).
- [7] 村岸 徹: "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).
- [8] J. Siegmund, A. Brechmann, S. Apel, C. Kastner, J. Liebig, T. Leich, and G. Saake: "Toward Measuring Program Comprehension with Functional Magnetic Resonance Imaging", In Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering (FSE'12), No.24 (2012).

- [9] T. Fritz, A. Begel, S. C. Mller, S. Yigit-Elliott, M. Zger: "Using Psycho-Physiological Measures to Assess Task Difficulty in Software Development", In Proceedings of the International Conference on Software Engineering (ICSE), pp.402-413 (2014).
- [10] 音茂 龍司, 辻 貞敏: "よくわかる脳波判読 第3版", 金原出版株式会社 (2015).
- [11] 村上 郁也: "イラストレクチャー認知神経科学", 株式会社オーム社 (2013).
- [12] 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).
- [13] 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).
- [14] T. Oohashi, E. Nishina, M. Honda, Y. Yonekura, Y. Fuwamoto, N. Kawai, T. Maekawa, S. Nakamura, H. Fukuyama, H. Shibasaki: "Inaudiblehigh-frequency Sounds Affect Brain Activity: Hypersonic-effect", Journal of Neurophysiology, Vol.83, No.6, pp.3548-3558 (2000).
- [15] A. T. Duchowski: "Eye Tracking Methodology", Springer (2006).
- [16] B. Law, M. S. Atkins, A. E. Kirkpatrick, A. J. Lomax and C. L. Mackenzie: "Eye Gaze Patterns Differentiate Novice and Expert in a Virtual Laparoscopic Surgery Training Environment", In Proceedings of ACM Symposium of Eye Tracking Research and Applications (ETRA), pp.41-48 (2004).
- [17] Mahnaz Behroozi, Alison Lui, Ian Moore, Denae Ford, and Chris Parnin: "Dazed: Measuring the Cognitive Load of Solving Technical Interview Problems at the Whiteboard", In Proceedings of the International Conference on Software Engineering (ICSE), pp.93-96 (2018).
- [18] T. Busjahn, R. Bednarik, A. Begel, M. Crosby, J. H. Paterson, C. Schulte, B. Sharif, S. Tamm: "Eye Movements in Code Reading: Relaxing the Linear Order", In Proceedings of the 23rd International Conference on Program Comprehension (ICPC), pp.255-265 (2015).