2021 年度 博士研究論文

# ドライバ操作における手指動作の分析を基にした手先の器用の指導についての研究

# Research of the Hand Dexterity based on the analysis of Finger Motion of Screwdriver Operation

愛知教育大学大学院・静岡大学大学院
教育学研究科共同教科開発学専攻

3094-0001 青木 麟太郎

指導教員 紅林 秀治

2021 年 11 月 16 日

# 目次

序章　研究の目的と方法　9

第 1 章　研究の背景と課題の設定　13
1.1　研究の背景　13
1.2　先行研究の整理　14
1.3　研究の目的・方法と課題の設定　15

第 2 章　手先の器用について　17
2.1　器用の概念　17
2.1.1　一般的な辞典における器用の定義　17
2.1.2　専門分野の辞典・事典における器用の説明　18
2.1.3　論文や図書における器用の説明　20
2.2　巧緻性について　21
2.2.1　器用とは　21
2.2.2　巧緻性の概念　21
2.2.3　手先の器用と巧緻性　23
2.3　器用に関する先行研究の整理　23
2.3.1　ものづくりへの意識調査から見た器用の整理　24
2.3.2　器用・不器用を自認する理由と問題点および指導法　25
2.4　技術教育で習得させたい技能　26
2.4.1　普通教育としての技術教育に必要な手先の器用　27
2.5　本章のまとめ　28

第 3 章　道具の選定　29
3.1　技術教育で指導されてきた道具操作とその研究方法　29
3.2　道具の選定　31
3.2.1　技術・家庭（技術分野）の授業で使用する道具　32

3.2.2 日常生活で使用する道具 . . . . . 32
3.2.3 「速く」「細かく」「正確な」手指を使った動きが求められる道具 33
3.2.4 新しい道具に対応できるよう，操作原理が他の道具の操作でも転移しそうな道具 . . . . . 34
3.3 本章のまとめ . . . . . 34

第 4 章 簡易手指動作分析システムの開発 37
4.1 開発方針 . . . . . 37
4.2 システムの概要 . . . . . 38
4.3 モーションキャプチャのプログラム . . . . . 38
4.3.1 手指の検出 . . . . . 38
4.3.2 プログラムの概要 . . . . . 39
4.3.3 データの利用 . . . . . 40
4.4 動作分析のプログラム . . . . . 42
4.4.1 画面について . . . . . 42
4.4.2 アニメーションによるモーションの再現 . . . . . 44
4.4.3 時間の経過における変位の比較 . . . . . 45
4.4.4 空間や平面における変位の比較 . . . . . 46
4.4.5 アニメーションとグラフ . . . . . 46
4.5 評価試験 . . . . . 47
4.5.1 箸操作の試験 . . . . . 48
4.5.2 箸操作の試験結果 . . . . . 48
4.5.3 キーボード操作の試験 . . . . . 48
4.5.4 試験 1 の結果 . . . . . 50
4.5.5 試験 2 の結果 . . . . . 51
4.6 本章のまとめ . . . . . 52

第 5 章 簡易手指動作分析システムの活用と改良 53
5.1 評価試験の位置づけ . . . . . 53
5.2 授業 1 の概要 . . . . . 54
5.2.1 モーションキャプチャを活用した授業 . . . . . 54
5.2.2 授業 1 の結果 . . . . . 56
5.3 授業 2 の概要 . . . . . 58
5.3.1 先行研究での検証方法 . . . . . 59
5.3.2 生徒の実態 . . . . . 59

5.3.3 授業計画 . . . . . 60
5.3.4 モーションキャプチャを活用した授業 . . . . . 61
5.3.5 授業 2 の結果 . . . . . 62
5.4 簡易手指動作分析システムの改良 . . . . . 65
5.4.1 追加機能 . . . . . 66
5.4.2 評価試験 . . . . . 67
5.5 本章のまとめ . . . . . 69

第 6 章 手指の巧緻性によるドライバ操作時の手指動作の違い 71
6.1 実験の目的 . . . . . 71
6.1.1 大学生と中学生の実態 . . . . . 71
6.2 実験の方法 . . . . . 73
6.3 実験の結果 . . . . . 75
6.3.1 手指動作の分類 . . . . . 75
6.3.2 作業時間 . . . . . 76
6.3.3 関節各部位・手首の各角度 . . . . . 77
6.3.4 作業時間と手首・関節各部位との関係 . . . . . 78
6.3.5 生徒の感想 . . . . . 83
6.4 考察 . . . . . 83
6.4.1 作業時間の比較 . . . . . 83
6.4.2 3 群ごとの特徴 . . . . . 83
6.4.3 教育的考察 . . . . . 84
6.5 本章のまとめ . . . . . 85

第 7 章 押圧力測定装置の開発 93
7.1 装置の概要 . . . . . 93
7.1.1 装置の形状 . . . . . 94
7.1.2 計測保存プログラム . . . . . 94
7.1.3 押圧力表示プログラム . . . . . 96
7.2 評価試験 . . . . . 97
7.2.1 試験 1 の概要と結果 . . . . . 98
7.2.2 試験 2 の概要と結果 . . . . . 98
7.2.3 試験 3 の概要と結果 . . . . . 99
7.3 本章のまとめ . . . . . 101

第 8 章 作業しやすさによるドライバ操作時の手指動作の違い 103

8.1 実験の目的 . . . . . 103
8.2 中学生群の実態 . . . . . 104
8.2.1 アンケート調査 . . . . . 104
8.2.2 握力測定 . . . . . 105
8.2.3 各群の実態 . . . . . 106
8.3 実験の方法 . . . . . 106
8.4 実験の結果 . . . . . 106
8.4.1 締め残ったネジの結果と握力との関係 . . . . . 106
8.4.2 両群の共通点 . . . . . 107
8.4.3 各群でのドライバ操作の要素間における相関 . . . . . 107
8.4.4 生徒の感想 . . . . . 108
8.5 考察 . . . . . 110
8.6 本章のまとめ . . . . . 111

第 9 章 ドライバ操作の技能指導上の要点リストの作成 113
9.1 ドライバ操作の技能指導上の要点リスト . . . . . 113
9.1.1 ドライバ操作の要点整理 . . . . . 113
9.1.2 要点リスト作成の方針 . . . . . 115
9.1.3 作成した要点リストについて . . . . . 115
9.2 本章のまとめ . . . . . 116

第 10 章 ドライバ操作の技能指導の評価 117
10.1 実験の目的 . . . . . 117
10.2 中学生の実態 . . . . . 117
10.3 実験方法の概要 . . . . . 118
10.4 実験の結果 . . . . . 119
10.4.1 握力と締め残ったネジの長さの関係 . . . . . 119
10.4.2 各群でのドライバ操作の要素間における相関 . . . . . 120
10.4.3 残ったネジの長さが 1.5mm 以下の群の特徴 . . . . . 121
10.4.4 技能習熟度別による手先の器用の分類 . . . . . 122
10.4.5 事後アンケートの結果 . . . . . 126
10.5 理解度に関する評価試験 . . . . . 130
10.6 考察 . . . . . 133

終章 研究の成果と今後の課題 135

謝辞 139

参考文献 141

付録 A　簡易手指動作分析システムのプログラム 151

付録 B　押圧力測定装置のプログラム 223

# 序章　研究の目的と方法

本研究の目的は，手指のモーションキャプチャシステムを用い，ドライバ操作時の手指動作を分析した結果から，中学生を対象に，道具操作時の速く細かく正確な手指を使った動きを身につけさせるため，ドライバ操作時の手指動作に何が求められるかを明らかにすることである。

身体動作をモーションキャプチャできるシステムは，運動発達や作業環境，技能習得過程で利用されてきた。志民らは身体のモーションキャプチャにより再現したアニメーションやグラフを活用し，音楽教育を専攻する学生に，指揮する自身の動きを分析・改善させた [1]。薄井は身体のモーションキャプチャにより再現したアニメーションを活用し，舞踊を学ぶ学生に，自身の動きと指導者・熟練者の動きとの違いや動作イメージと実際の動作との差に気づかせ，動きを修正させた [2]。しかし技能習得過程におけるモーションキャプチャの教育的活用は，技能習得対象の動きに触れてきたり既に学習した経験があったりした，動きのイメージがある学生を対象としてきたが，技能習得対象の動きを初めて学ぶ学習者（以後，初学者）を対象としていなかった。また手作業における技能習得には，手指の動作が重要になる。しかし，手指動作が身体動作より狭い動作範囲のため，初学者には手指作業の目のつけどころがわかりにくいと考えられる。そこで，本研究ではモーションキャプチャシステムを用い，道具操作時の手指動作を分析した結果をもとに，道具操作上の要点を整理していくことを試みた。さらに，整理した道具操作上の要点を用いて技能指導した結果，道具操作における手指動作では何が重要であるかを明らかにする。筆者は，日常使用頻度が高いドライバ操作における手指動作について分析し，手指動作を基にした指導方法を考案した。

本論文は，12 章により構成されている。

第 1 章では，普通教育としての技術教育において，設計・計画・創造する活動が重要視されている背景がある一方，設計・計画・創造する活動の精度を高めるには，道具を自身の思った通りに操作できる手先の器用をある程度養う必要があると考えた。そこで，普通教育としての技術教育に必要な手先の器用を明らかにするための研究方法を整理した。

第 2 章では，器用や巧緻性の概念を確認し，器用に関する先行研究と技術教育で習得さ

せたい技能を整理した。整理した結果をもとに，本研究では，普通教育としての技術教育に必要な手先の器用を，「道具操作のための巧緻性（「速く」「細かく」「正確な」手指を使った動き）」，「新しい道具操作技能を習得しやすい資質・能力」と定義することにした。

第 3 章では，第 2 章で定義した「道具操作のための巧緻性」を学習の初期段階で習得すると考え，普通教育としての技術教育で使用される道具の中で，どの道具を選定すべきを検討することにした。そこで，技術・家庭（技術分野）の授業と日常生活で使用する，速く細かく正確な手指を使った動きが求められ，かつ，新しい道具操作の技能習得にも転移し易い道具として，ドライバを選定することにした。

第 4 章では，簡易手指動作分析システムの概要を整理した。簡易手指動作分析システムでは手指にセンサをつけたり，複数台のカメラで撮影したりしなくても手指動作を計測することができる。また，計測したデータをもとに，手指のアニメーションを再現したり，関節各部位の変化をグラフ化したりすることができる。

第 5 章では，簡易手指動作分析システムを用い，中学校技術・家庭（技術分野）の授業において，技能学習に活用した。その結果，現状のシステムではドライバ操作時の手指の動作を計測できないことがわかった。そこで，手指動作を計測する計測デバイスである Leap Motion を上に設置し，ドライバ操作時の手指動作を計測できるように改良した。

第 6 章では，手指の巧緻性が異なる大学生と中学生を対象に，ドライバ操作時の手指動作を計測・比較した。その結果，大学生・中学生によらず，手首の動きが速い学生は，ネジを締めるのが速い傾向にあるとわかった。また，大学生は中学生に比べ，親指の各関節が一緒に大きく動いているとわかった。

第 7 章では，ロードセルを用い，ドライバからネジに向けて押す力を測定できる装置を開発した。本装置と簡易手指動作分析システムを用い，ドライバ操作時の手指動作と押圧力が計測できることを確かめることができた。

第 8 章では，中学生を対象に，ドライバ作業しやすいか否かにより，ドライバ操作時の手指動作や押圧力を計測し，どういった共通点・相違点があるかを分析した。分析をした結果，作業のしやすさによらず，ドライバ操作時の押圧力が小さいほど，押圧力の合計が小さくなるとわかった。作業しやすい条件は作業しにくい条件に比べ，押圧力が大きければネジを締めることができる条件であった。また，握力が小さい学生ほど，柄を持つ手の人差し指が伸び，薬指・小指でおおうように握っていないとわかった。

第 9 章では，第 6,8 章での結果と有川らの先行研究をもとに，ドライバ操作の技能指導上の要点リストを作成した。

第 10 章では，中学生を対象に，第 9 章のリストを用いドライバ操作の技能指導をした。その結果，握力が小さい学生でもドライバの持ち方を変えることにより，ネジを締めることができるようになった。

終章では，ドライバ操作技能において，「握力が大きく，指先が長い」，「5 本の指すべて

で，道具をしっかりと固定する」，「手指の動きに無駄な動きがない」といった要点が求められると考察した。そこで，普通教育としての技術教育に必要な手先の器用を身につけるため，速く細かく正確な手指の動きとして，5 本の指すべてで，道具をしっかりと固定すること，および道具操作時に無駄な動きがなくなるよう最適化されていく学習過程を身につける必要があると考えた。

以上より，本論文では，研究目的「簡易手指動作分析システムを用いドライバ操作時の手指動作を分析した結果をもとに，中学生を対象に，道具操作時の速く細かく正確な手指を使った動きを身につけさせるには，何が求められるかを明らかにすること」を達成できたと考えているが，今後の課題として，技術・家庭 (技術分野) の授業において，学習の初期段階で普通教育としての技術教育に必要な手先の器用を養う学習の効果を確かめることであると考えている。

# 第 1 章

# 研究の背景と課題の設定

普通教育としての技術教育では時代の変化により，設計・計画を重視されるようになってきた。それに合わせ，設計・計画した製作活動に必要な手先の器用を検討する必要があると考えた。そこで本章では，普通教育としての技術教育に必要な手先の器用を明らかにするための研究方法を整理した。

## 1.1　研究の背景

1960 年代の半ば，「ナイフで鉛筆が削れない」「ヒモが結びない」など，児童の手先が不器用になってきたことが指摘された [3]。その後，1970 年代には手先が不器用であることが子どもの成長や発達に影響するとして，問題視されるようになった [3]。一方，日常生活に必要な手指動作は，幼児期の終わり頃までにほぼ発達し，巧緻な手指の動きはその後も発達を続けること [4][5] がわかっている。そのため，児童・生徒が日常生活で使用する道具や遊び，生活習慣などの変化に合わせ，手先の器用に質的な変化が起きたという主張もある [4]。以上，2 つの主張で共通するのは，「児童・生徒の手先の器用が時代とともに変化していること」と，「手先の器用と道具の操作が関連すると捉えられていること」だと考えられる。そのため，筆者は，児童・生徒の手先の器用が変化してきたことに合わせ，児童・生徒がこれまで通りに，道具を自身の意思通りに操作できているかに疑問を抱いた。

道具を自身の意思通りに操作する活動は，中学校技術・家庭（技術分野）の授業で必要な活動である。また，昨今，世界的に STEM や STEAM といったものづくりを含む教育活動が取り組まれるようになり，道具を自身の意思通りに操作して，ものづくりによる問題解決する力を養う必要もあると考えられる。そのため，学校教育では児童・生徒が道具を自身の意思通りに操作する活動が求められているが，学校教育でのものづくり活動では重要視される場面に変化が生じてきている。

## 1.2　先行研究の整理

技術教育は発達段階ごとに，学習内容が検討されてきた。「技術教育に関する研究を行い，その振興普及および会員相互の連絡を図り，もって技術教育の発展に寄与することを目的」とした，一般社団法人の学会として，日本産業技術教育学会 [6] がある。日本産業技術教育学会では製作活動を小学校，中学校，高等学校の段階ごとに学ぶ，プロセスの量・質を変化させるアプローチを提案している。大橋 [7] は，第 1 期を小学校高学年から中学校前半までの期間 (10〜13 歳 (小 4・小 5〜中 1))，第 2 期を中学校後半から高等学校までの期間，第 3 期を大学から一般社会人（たとえば，技術者）の期間と設定し，それぞれで学ぶ教育の目標を定義し，期間ごとに「設計の過程」「工程計画」「製作」の質が高まる方法を学習させた。紅林 [8][9] は普通教育としての技術教育として，設計の過程を中心にすえ，ものづくりをシステムづくりと捉え直す技術教育を提案している。また，国際バカロレア機構が認定した日本の学校では，技術・家庭に相当する教科「Design」がある。Design の授業では，「A　探究と分析」，「B　アイデアの発展」，「C　解決策の創造」，「D　評価」の順に問題解決のプロセスを何度も繰り返され，年齢ごとにプロセスで求められる質が変化する [10]。技術・家庭（技術分野）における現行の教科書では，4 領域でのものづくりにおいて，まず課題の設定を明確にして進めるようまとめられている [11][12][13]。以上より，普通教育としての技術教育では，設計・計画を重視するアプローチが実施されるようになったと言える。

昨今では小学校でのプログラミング教育が必修化され，論理的思考力や創造性，問題解決能力等の育成 [14] も求められるようになり，普通教育としてのプログラミング教育も検討・推進されている。紅林 [15] は，小中学校における普通教育としてのプログラミング教育について，PDCA サイクルの Do から始めて Plan に至る授業を提案している。大村ら [16] は，初学者を対象に，ソフトウェア設計の過程を普通教育に適用する方法を提案している。松ら [17] は，Google，Apple，Micro soft などの IT 企業や，イギリスやロシア，エストニア，ニュージーランド等も初等中等教育段階でのプログラミング教育に力を入れていることを報告している。福井ら [18] は，「普通教育としてのプログラミング教育では，生徒の資質・能力育成の観点から，生徒一人ひとりにとって新しく，生徒一人ひとりを最大限に尊重することが重要であり，教科や分野にとらわれず新しいことを生み出す人格や態度を育成し，創造的問題解決へ向かわせることが重要となる」と述べている。以上より，普通教育としての技術教育や普通教育としてのプログラミング教育が時代にあわせ，思考力や創造力の育成を重視するように変化してきたからこそ，設計・計画・創造する活動が重要になったと言える。つまり，設計・計画・創造する活動をものづくりとして実行したり，設計・計画に必要な情報を得る活動を実行したりする場面を授業の中に位置づけ

る必要がある。しかし，そのためには，道具・機械を操作する手先の器用を養う必要がある。なぜならば，設計・計画・創造には，創造したいものをイメージするだけでなく，どのような手順で完成させるのかという見通しを持つことなしに設計・計画できないからである。したがって，ものづくりをした経験，つまり，道具を適切に使用したり，加工したり，組み立てたりした経験が必要となる。また，土井は [19] 作業段取り (目標に向け，最適な方法や順序を思考する過程と意志決定する過程及び工具などを準備する行為の過程) を中学校技術科教育で行った結果，難易度の高い作業を予測したり，予期せぬ事態を解決できるよう工夫したりする力が身につきにくいことを明らかにしている。この結果について，児童・生徒や職人のものづくりへの意識調査による結果も踏まえ，「学習の初期段階は基礎的な知識の不足や当面の技の獲得に目が向けられ，工夫する，推論する，問題解決するなどには，目が向け難いという事実がある」とし，作業段取りの前段階 (学習の初期段階) において，実技的技能 (基礎的な技能) を獲得する必要があると考察している。そのため，普通教育としての技術教育・プログラミング教育において，設計・計画・創造する活動の精度を高めるには，道具・機械を操作する手先の器用をある程度は養う必要があると捉えた。

## 1.3 研究の目的・方法と課題の設定

本研究では，普通教育としての技術教育に必要な手先の器用を明らかにすることである。そのために，以下の 1〜4 で課題を整理していく。

1. 手先の器用がどう捉えられているか，どうあるべきかを明らかにし，普通教育としての技術教育において，生徒に獲得させたい手先の器用を定義する。（以後，課題 1）
2. 普通教育としての技術教育で生徒に獲得させたい手先の器用や，技術教育で扱われる道具をもとに，手先の器用における手指動作の違いが確認しやすい道具を選定する。（以後，課題 2）
3. 発達段階と作業条件の違いにより，選定した道具を操作した時，手指動作にどういった違いがあるかを確かめる。（以後，課題 3）
4. 道具操作時の手指動作の特徴をもとに，道具操作の熟達化を促し，中学生が獲得した手先の器用ごとに，その違いを分析する。（以後，課題 4）

上記の課題 1〜4 を設定した理由を述べる。1 を課題とした理由は，一般的な視点と専門的な視点において，手先の器用に対するイメージに乖離があったり，要求が違ったりする可能性があると考え，児童・生徒・保護者と社会といった 2 つの視点からのニーズに答える，普通教育としての技術教育に必要な手先の器用を整理すべきであるからである。

2 を課題とした理由は，普通教育としての技術教育に必要な手先の器用と，実際の手指動作との関係を明らかにするため，明らかにしたい手先の器用にあった道具を選ぶ必要があると考えたためである。

3 を課題とした理由は，技能習得の為の学習初期段階において，道具操作時の手指動作を運動発達に合わせた指導内容になっていたり，様々な作業条件で道具を操作することで手指動作を発達させたりするための，要点を指導したいと考えた。つまり，発達段階の違いにより，「どう手指動作が発達していけばいいか」，作業条件の違いにより，「どういった手指動作の特徴が共通・相違しているか」を知ることで，技能習得過程において，道具操作時の手指動作時の要点を得やすいと考えたためである。

4 を課題とした理由は，道具操作時の手指動作時の要点を伝えた時，現状の中学生ならば，どこまで手先の器用を得ることができるか，普通教育としての技術教育に必要な手先の器用に向け，何に留意する必要があるかを確認するためである。

# 第 2 章

# 手先の器用について

前章では，普通教育としての技術教育において設計・計画が重視されるようになり，その活動を実施するためにも，現在求められている手先の器用を明らかにする必要があると考えた。本章では，1.3 節の課題 1 を明らかにするため，日本の現代社会では，手先の器用がどう捉えられているか，どうあるべきかを整理し，整理した結果と技術教育で習得させたい技能とを見比べ，普通教育としての技術教育に必要な手先の器用を定義する。

## 2.1　器用の概念

一般的にイメージする「器用」と，実際の「器用」との間に齟齬があるのではないかと考え，手先の器用がどう捉えられているかを調べた。最初に，辞典や事典，文献にある器用の定義や説明を調べた。調べた器用の定義・説明は，「1. 一般的な辞典，2. 専門分野 (心理学分野) の辞典・事典，3. 雑誌や図書」の 3 種類から調べ，その結果をそれぞれ表にまとめた。それぞれの器用に対する捉えの共通点と相違点を明らかにした。

### 2.1.1　一般的な辞典における器用の定義

一般的には「器用」がどう捉えられているかを確認するため，「一般的な辞典」における器用の定義を整理した。現代の日本人が「器用」をどう捉えるかを抽出するため，一般的な辞典として，2010 年以降に出版された辞典を選定した。表 2.1 に一般的な辞典における「器用」の定義を示す。

表 2.1 より，器用について，「手先を使った細かい動きができること」や「要領よく物事を処理すること」がイメージにあるとわかる。また，表 2.1 の No.2,3,8 から，才能と捉えられていることもわかる。

表 2.1　一般的な辞典における器用の定義

| No | 第 1 著者 | 出版日 | 器用の定義 |
|---|---|---|---|
| 1 | 金田一京助 [90] | 2012 | 1．こまかい仕事をうまくやるようす。<br>2．うまくたちまわるようす。 |
| 2 | 村松明 [91] | 2012 | 1．からだを思うように動かして，芸事・工作などをうまくこなすこと。また，そのさま。<br>2．要領よく，いろいろな物事を処理すること。また，そのさま。<br>3．抜けめなく立ち回ること。またそのさま。<br>4．不平不満なく，受け入れること。いさぎよいこと。また，そのさま。<br>5．すぐれた才能のあること。また，その人。 |
| 3 | 新村出 [92] | 2018 | 1．物事の役に立つ才能のあること。賢いこと。<br>2．手先がよくきき，技芸に巧みなこと。細かい仕事をうまく処理すること。<br>3．いさぎよいこと。 4．容貌。器量。<br>5．抜け目がなく，要領がよく立ち回ること。 |
| 4 | 見坊豪紀 [93] | 2018 | 1．手先の技術がうまいようす。<br>2．（気がきいて）うまくものごとが処理できるようす。 |
| 5 | 小野正弘 [94] | 2019 | 1．こまかい仕事をうまくやるようす。<br>2．ものごとを要領よくやるようす |
| 6 | 西尾実 [95] | 2019 | 1．物事，特に細かい仕事をうまくやってのけるさま。<br>2．要領がよいさま |
| 7 | 中村明 [96] | 2019 | 技芸などで細かいことを巧みにこなすさま。 |
| 8 | 山田忠雄 [97] | 2020 | 1．手先を使う細かい仕事などを，たいして苦労もせずにこなす技量を備えていること（様子）。<br>2．判断力・応用力にすぐれ，専門以外のことでも無難にこなす才能を備えていること（様子）。 |

### 2.1.2　専門分野の辞典・事典における器用の説明

研究者が一般的には「器用」をどう捉えているか確認するため，「専門分野 (心理学分野) の辞典・事典」における器用の定義や説明を整理した。表 2.2 に専門分野の辞書における「器用」の説明を示す。

表 2.2 の No.3,4,6 より，器用について，「手先の動き」というイメージがあるとわかる。また，表 2.2 の No.1,2,4,5 より，「新しい技能を身につけやすい性質」といったイメージでも捉えられ，その技能習得の遅速が器用か否かであることを示している。しかし，表 2.2 の No.5 より，技能習得の遅速は個人差であり，最終的な到達度を決定するとは限ら

ないため，器用な人が器用でない人より，最終的に到達する技能が優れているわけではないことに留意する必要がある。また，表 2.2 の No.1 より，器用が「小範囲の動作内でのみ一般性を有するにすぎない」とあり，すべての「新しい技能を身につけやすい性質」ではないと捉えていることがわかる。

表 2.2　専門分野の辞典・事典における器用の説明

| No | 第 1 著者 | 出版日 | 器用の説明 |
|---|---|---|---|
| 1 | 下中弥三郎 [98] | 1957 | ・主として運動技能に熟達しやすい性質<br>・技能熟達の過程が容易に発展するものもあれば，そうでないものもある。前者は器用な人であり，後者は不器用な人であると言われる。<br>・比較的特殊なものであり，少なくとも小範囲の動作内でのみ一般性を有するにすぎない。 |
| 2 | 園原太郎 [99] | 1971 | ・新しい動作技能を習得しやすい性質<br>・動作の初期の段階では，それぞれの動作が相互にまとまりのないものと認知されているが，習熟が進むにつれて，それぞれの動作の関係が認知され一群のまとまりのある動作になっていく。この過程の遅速が器用さのちがいを示すものであると考えられる。 |
| 3 | 宮城音弥 [100] | 1979 | 手さきの仕事がうまくできる性質であるが，とくに細かい仕事について言い，正確さと速さが検査される。 |
| 4 | 大中邦彦 [101] | 1981 | 体肢，とくに上肢を使っての運動，または作業の熟練技能の獲得にすぐれてる性質 |
| 5 | 日本スポーツ心理学会 [102] | 2008 | ・新しい運動技能の習得に取り組んだとき，その運動技能を短い期間で習得したりすぐに上達したりできる者とそうでない者とが存在する。前者が器用な人であり，後者が不器用な人である。<br>・ある人が器用である否かは，学習の初期段階でのみ上達の進み具合から判断されることが多い。しかしながら，学習の進み具合には多様な個人差があり，初期段階での学習速度が最終的な到達度を決定するとは限らない。運動技能の学習や指導に当たっては，初めは遅いけれども，最終的には十分なレベルに達する可能性を視野にいれておく必要がある。 |
| 6 | 谷田貝公昭 [103] | 2011 | 手先や身のこなしが巧みなこと。心理学の世界でいう感覚運動領域での巧緻性である。 |

### 2.1.3　論文や図書における器用の説明

研究者・専門家が個人でどう捉えているかを確認するため，「論文や図書」における器用の説明を整理した。表 2.3 に論文や図書における「器用」の説明を示す。

表 2.3 の No.1 より，道具を思ったように敏捷に操作できることが器用に繋がっていると考えられる。表 2.3 の No.2 は表 2.1 より，器用が才能であるという定義だけでなく，合理的な方法を身につけることが重要と捉えているとわかる。表 2.3 の No.3 は，表 2.2 にある「手先の動き」や「新しい技能を身につけやすい性質」を含む解釈である。表 2.3 の No.4 より，器用さには「動作の巧緻性」と「認知的な融通性」の 2 つが含まれていると考えられる。表 2.3 の No.5 にある「目的に向かって適切な時間に，適切な筋が，ちょうど適切な力を出して動くような運動」が表 2.3 の No.1 での説明と同義に捉えていると考えられる。

表 2.3　論文や図書における器用の説明

| No | 第 1 著者 | 種類 | 出版日 | 器用の説明 |
|---|---|---|---|---|
| 1 | 時実利彦 [104] | 図書 | 1977 | 器用ということは，自分でこうしようと思った運動動作が思った通りに，間違いなく，敏捷にできることをいう |
| 2 | 森下一期 [105] | 図書 | 1994 | 器用さは天性によるだけでもなく，闇雲に練習するのでもなく，合理的な方法を身につけること |
| 3 | 槇塚忠穂 [106] | 論文 | 2000 | 手先が器用な人は，手指動作であれば，何をやらせてもはじめから上手くやってしまう。器用さとは手指技能においては一つの素質である。 |
| 4 | 久保田競 [107] | 図書 | 2013 | 器用さは，単に動作の巧緻性ばかりでなく，認知的な融通性も含めて考える必要がある。そのような器用さを支えるのは，身体であり脳である。 |
| 5 | 久保田競 [108] | 図書 | 2015 | ・器用な運動とは，要素的な運動がつぎつぎと協調して行われ，余分な筋肉が使われず，目的に向かって適切な時間に，適切な筋が，ちょうど適切な力を出して動くような運動<br>・「1．受容器と筋肉（効果器）のフィードバックが時間的・空間的に高度に組織化されている　2．時間的・空間的な運動パターンが形成されている　3．時間関係の予測と順序だった段階的な反応が行われる」この三要素がバランスよく働いて，器用さが生まれるのである。 |

## 2.2　巧緻性について

2.1 では，器用の説明を分野ごとに整理した。器用と，器用に類似する巧緻性の特徴をまとめ，器用と巧緻性との関係を整理した。

### 2.2.1　器用とは

表 2.1〜2.3 から，一般的イメージと研究者が抱く「器用」に対するイメージの共通点は,「手先を使った動きができること」や才能 (新しい技能習得しやすい性質) だと言える。この共通点は，生徒や保護者が中学校技術・家庭（技術分野）の授業を通して，身につけたい器用だと考えられる。

相違点として，一般的なイメージでは要領よく物事を処理すると考えられる。一方，専門分野では，器用が合理的な方法を身につけるという解釈や，初期段階で新しい技能習得に時間がかかっても，最終的に到達する技能が低いとは限らなかったり，器用の通用する動作の範囲が限定的であったりすると考えられている。

また，表 2.2 の No.6 と表 2.3 の No.4 では「巧緻性」が器用に類似する意味をもつと考えられる。そこで，器用に類似する単語である「巧緻性」の意味を整理した。

### 2.2.2　巧緻性の概念

巧緻性の意味を確認するため，辞典や事典，図書および論文における巧緻性の定義を整理した。表 2.4 に，辞典・事典，図書および論文における「巧緻性」の説明を示す。

表 2.4 内の巧緻性の説明において，共通する特徴は以下の通りである。

- No.2〜6,8 にある巧緻性の説明から,「手指・指先の動きであること」
- No.1,3,8 にある巧緻性の説明から,「速く動かすこと」
- No.2〜4 にある巧緻性の説明から,「正確な動きであること」
- No.4〜6 にある巧緻性の説明から,「細かい精密な動きであること」
- No.1,7 にある巧緻性の説明から,「神経と筋との間で調整した動きであること」

また，一般的には，表 2.4 の No.5,6 より，身体の大きな動きが手指の細かい動きより，先に発達が進み，コントロール可能になると言える。そのため，手指の細かい動きを確認すれば，運動発達の進み具合がわかるとも捉えられる。

表 2.4　辞典・事典，図書および論文における巧緻性の説明

| No | 第 1 著者 | 種類 | 出版日 | 巧緻性の説明 |
|---|---|---|---|---|
| 1 | 人間工学用語研究会 [113] | 事典 | 1983 | 巧緻性 (skill)：動作の敏しょう性を表す言葉で，生理学的にいえば神経と筋の間の調整力がうまく働いている状態 |
| 2 | 枝光慶江 [114] | 論文 | 1994 | 「手指の巧緻性」を「目標を手指を使って正確に表現できる能力」とした。 |
| 3 | 中村隆一 [115] | 図書 | 2000 | ・手と指の巧緻性 (manual dexterity)：手と指による運動がすばやく，正確であること<br>・指の巧緻性 (finger dexterity)：指先だけで行う小物の操作が速く，正確であること |
| 4 | 和田攻 [116] | 事典 | 2002 | ・巧緻動作 (skill activity)：小児の運動発達において最もレベルの高い運動コントロールで，手指によるつかみ・つまみなどの細かな動作<br>・巧緻握り (precision grip)：手による細かい精密な握り動作をいう。肩・肘・手関節によって手という器官を目的の位置に向け，さらに手指を動かさなければならない。手自体の運動性に加えて，感覚および視覚によるフィードバックが必要 |
| 5 | 最新医学大辞典編集委員会 [117] | 辞典 | 2005 | ・巧緻性 (skillfullness,dexterity)：特に手指の動作における指先の器用さの程度をさす場合に使われ，細かい精密な作業を行う能力といってよい。<br>・高い巧緻性を発揮させるためには各指の分離した運動や複合運動のほかに，効果器である手指が効率よく動くために以下の点が必要である。まず，体幹に近い部分の肩関節や肘関節，手関節などの安定性 (支持性)，ついて上肢の運動性，そして手の繊細な感覚機能である。 |
| 6 | 持丸正明 [4] | 図書 | 2013 | ・巧緻運動は手先の操作のように微細な運動を，それと対をなす概念である粗大運動は頭や体幹を大きく動かす運動を指す。<br>・運動発達には一定の順序があり，頭部から尾部へ中心部から周辺部へ，そして粗大な運動から巧緻な運動へと発達が進む。手の運動発達においてもこれらの法則をみることができ，上腕から前腕，手掌，指という順序で，すなわち体幹に近い中心部から末端方向へとコントロールが可能になっていく。<br>・大きな筋力を使う上腕の動きは粗大だが，その後に発達する手掌や指の動きは細かな筋肉を使う巧緻なものである。 |
| 7 | 日本生理人類学会 [118] | 事典 | 2015 | 巧緻性 (スキル) とは，神経と筋との協同作業がうまくいくようにする働き |
| 8 | 見坊豪紀 [112] | 辞典 | 2018 | 巧緻性：運動するときのからだの動きがなめらかですばやいこと。 |

### 2.2.3　手先の器用と巧緻性

表 2.5 に手先の器用と巧緻性それぞれの説明を示す。表 2.5 より，巧緻性には，「手指・指先の動きであること」や「速く動かすこと」，「正確な動きであること」，「神経と筋との間で調整した動きであること」といった，器用での「手先を使った動きができること」をより具体的にした特徴が見られる。そのため，手先の器用とは，巧緻性と，「新しい技能習得しやすい性質」などのそれ以外の要素に分類できるという主張で，本論文も進めていく。

また，巧緻性が「身体の動きが発達した後に発達する動き」であるため，初期段階で新しい技能習得しにくい原因として，身体の動きが発達していない，または，神経と筋との間での動きが調整されていないことにあるとも考えられる。これは，現在の児童・生徒に見られる手先の器用に関する特徴と考えられる。

表 2.5　手先の器用と巧緻性の説明

| 手先の器用 | 巧緻性 |
|---|---|
| ・手先を使った動きができること | ・手指・指先の動きであること |
| ・才能 (新しい技能習得しやすい性質) | ・速く動かすこと |
| ・合理的な方法を身につけること | ・正確な動きであること |
| ・初期段階で新しい技能習得に時間がかかっても，最終的に到達する技能が低いとは限らないこと | ・細かい精密な動きであること |
| | ・神経と筋との間で調整した動きであること |
| ・器用の通用する動作の範囲が限定的であること | ・身体の動きが発達した後に発達する動き |

## 2.3　器用に関する先行研究の整理

1990 年代，中学校技術・家庭（技術分野）の授業において，ものをつくるとき自分が不器用であると意思表示し，製作に取り組まない生徒の問題が指摘された [79]。これは，1990 年代以降，児童・生徒の手先の器用が学校教育でのものづくり活動において，使用する道具を思った通りに操作できないという理由が一つに考えられる。そこで，1990 年代から 2010 年代までの間では，児童・生徒や保護者がものづくり教育に対し，どう考えているかを明らかにするため，ものづくりへの意識調査から整理した。また，器用・不器用を意識する理由や問題点，指導法も整理した。

### 2.3.1　ものづくりへの意識調査から見た器用の整理

土井らは 1994 年から 1997 年 7 月の間に，小学 5 年生から中学 3 年生までの 340 名 [80]，中学校の全学年 7819 名 [81]，小学 3 年生から高等学校 3 年生までの 4260 名 [82] へ，ものづくりへの意識調査を行っている。その結果，1990 年代における児童生徒のものづくりに関する特徴として，①小学校から中学校へ進級すると「つくることが好き，器用感」の割合が低下すること [80]，②ものづくりの経験量が多い児童生徒には器用であると自認していたり，家庭でもものづくりの環境が整っていたり，ものづくりの経験量が少ない児童生徒に比べ働くことへの関心が高かったりすること [81][82] を明らかにした。

土井らが示した児童生徒のものづくりに関する特徴①，②は，2010 年代前半に行われた意識調査の結果に類似する。2011 年，小林ら [83] は小学 1 年生から高等学校 3 年生までの 1306 名を対象にものづくりへの意識調査を行い，小学校高学年から中学校にかけて「図画工作，技術・家庭科」といったものづくり科目を嫌いと回答する割合が急激に高くなることを明らかにした。小林らが示した結果は土井らが示した特徴①と同じであると言える。2014 年，三宅ら [84],[85] は中学校の全学年 400 名を対象にものづくりへの意識調査を行い，「ものづくりを好き」と回答した生徒が，器用であると自認していたり，私生活等でおもちゃの分解や，工作や料理などで自分なりのアレンジをした活動経験が多かったり，機械いじりの好きな身内の存在がいたりする傾向にあることを明らかにした。三宅らが示した結果は土井らが示した特徴②と同じであると言える。

さらに，成人や保護者を対象にしたものづくりへの意識調査も確認した。2002 年，三山ら [86] は成人 59 名を対象に，技術・家庭（技術分野）への意識調査を行い，技術科の中心的目標として，もの作りや道具の使い方などの「テクニック」が 54.0 %で最も多く望まれていることを明らかにした。2005 年，小森ら [87] は中学生 3 年生 600 名と保護者 529 名を対象に，技術・家庭への意識調査を行い，技術・家庭科で身に付いた (と思われる) 資質・能力として，「手先が器用になった」「道具を使うことに自信が持てるようになった」が身に付いたと感じている生徒が回答全体の 30 %程度，保護者が回答全体の 20 %未満であることを明らかにした。2017 年，吉村ら [88] は中学校とものづくり教室に子どもを参加させる保護者 238 名を対象に，ものづくり活動への意識調査を行い，多くの保護者が学校教育におけるものづくり活動に十分満足しておらず，「器用さ」や「工具を使うことになれる」などのカテゴリを，ものづくり活動で生徒に最も身につけてほしい力であることを明らかにした。

したがって，1990 年代から 2010 年代までの学校教育におけるものづくり活動の問題として，中学校技術・家庭（技術分野）の授業が小学校図画工作の授業に比べ，家庭でものづくりを経験してこなかった生徒が，道具を操作する活動が思ったようにできないと感

じ，手先が器用でないと自認する傾向にあったと言える。また，保護者は中学校技術・家庭（技術分野）の授業における学習目標が，器用や道具操作の習熟だと意識し，生徒に器用や道具操作の習熟が身に付いていないと捉えているとわかった。

### 2.3.2 器用・不器用を自認する理由と問題点および指導法

器用・不器用意識の理由に関する調査を確認していく。土井 [74] は小学 3 年から中学 3 年までの計 444 名を対象に，自由記述式の調査を行った結果，器用だと自認する児童・生徒の理由が，「細かいことができる (22 %)」や「うまくできる (15 %)」など作業の結果と，「人から器用と言われた (16 %)」という他者評価において，高い割合を占めた。不器用だと自認する児童・生徒の理由では，「うまくできない (19 %)」，「細かいことができない (14 %)」，「仕上がりがきたない (8 %)」など作業の結果を高い割合で理由に選んでいた。また中学生が小学生に比べ，ものをつくる速さが器用・不器用を自認することに関与しているとわかった。これは，土井らが中学 1 年生を対象に行った別の調査 [78] でも，器用・不器用を自認するか否かにより，細かな作業や速く処理し，仕上げがきれいに出来るという作業処理や作業結果に差があるとわかった。また，川端ら [75] は高校 1 年生 496 名を対象に，選択式の調査を行った結果，自分を器用・不器用と見なす理由として，「人から言われた」と「技術・家庭科での作品の出来栄えの良し悪し」を挙げる傾向にあったと報告している。

北沢ら [24] は，日常生活での手工具使用経験が多い児童と少ない児童を対象に，両児童の経験が少ない「鉛筆をナイフで削る作業」の技能習得過程を比較した結果，手工具使用経験が多い児童が少ない児童に比べ，経験を重ねるごとに技能の成長が早いことを明らかにしている。これは，土井ら [89] が小学 3 年から中学 3 年までの 7810 名を対象にした調査において，器用・不器用を自認する要因として，自己評価や他者評価だけでなく，生活経験・環境も影響することを明らかにしたことに通じる結果と言える。

以上のことから，器用・不器用を自認するようになる理由として，①自己・他者評価，②生活経験・環境，③中学校技術・家庭の授業において，細かな作業や速く処理し，仕上げがきれいに出来るという作業処理・作業結果が挙げられる。

器用でないと自認することは，製作に取り組まない生徒を生み出す [79] だけでなく，未成年者の技能習得や性格形成に影響を及ぼすことがわかっている。鳴海ら [76] は高校生を対象に，手縫いの反復練習の効果を調査した結果，不器用と自認する生徒の上達度が低いことを明らかにしている。発達性強調運動障害児 (DCD 児) は知的な問題をもたないが，手指の動きに発達の遅れがあるために，小学校で手先の不器用から勉強の進度に追いつけなかったり，自分の不器用を自認して自信をなくしたりして，その後の学業，社会適応，情緒的発達に影響を与える [77]。土井ら [78] が中学生 1 年生を対象に，器用感・不器用感に

関する意識調査の結果とペーパークラフト飛行機の製作過程・完成度を比較したところ，器用・不器用を自認する各生徒群の間に作業速度や作業動作といった作業パフォーマンスに明白な違いがなかったが，器用と自認する生徒の自尊感情が高く，不器用と自認する生徒の自尊感情が低いことを明らかにしている。また，中学 3 年生以降では器用であるか否かがはっきりと分離することもわかっている [80]。そのため，ものづくり教育では，中学 3 年生になる前に，児童・生徒の手先の器用に配慮した指導を検討する必要がある。

土井ら [80] は手先の器用に配慮した指導法として，段取り（見通し）を立て，メタ認知を促すことを提案している。段取りの学習によって，生徒は実際の技能と理想の技能との落差が小さくなり，学習への不安感を低減させる効果があったとまとめている。しかし，そもそも児童・生徒へ段取りの学習が必要になっているのはなぜなのか。児童・生徒が実際よりも目標とする技能が高いと考え，落差が生まれているのではないか。つまり，児童・生徒は，1990 年代から 2010 年代までに，中学技術・家庭（技術分野）に対し，実際よりも高い目標の技能を求めているから落差が生まれる可能性を考える必要がある。

## 2.4　技術教育で習得させたい技能

中学校技術・家庭科技術分野では「生活と技術について基礎的な理解を図るとともに，それに係る技能を身に付ける」との目標 [109] が揚げられ，中学生に基礎的な技能を習得させることが求められている。古川らはこの目標を下記のように解説している。

- 「技能」の指導に当たっては，習得した一つの技能を他の領域の技能の向上につながるように工夫することが必要である。例えば，「材料と加工の技術」における木材と金属の切断加工の技能に関して，共通点と相違点を理解しながら習得することなどが挙げられる [110]。
- 「技能」については，加工，栽培，プログラミング等のいずれの内容に関しても，獲得した個別の技能を自分の経験や他の技能と関連付け，変化する状況や課題に応じて活用できる技能として習熟させることを意識した指導とすることが必要である [111]。

上記にある内容から，技術教育では習得した技能を自身の経験や他の技能と関連付け，変化する状況や課題に応じて活用できる技能を習熟させることを求めていることがわかる。また，学習の初期段階では技能を一つ習得させることが先行していることもうかがえる。これは，学習初期段階に実技的技能 (基礎的な技能) を育成し，工夫したり推論したり問題解決したりする活動の精度を高める必要があるとする，土井 [19] の主張に通じる。したがって，普通教育としての技術教育では，まず一つの技能を習得させることが重要であり，習得した技能をもとに，他の技能と関連付けて新たな技能を習熟させたり，設計・

計画・創作する活動の精度を高めたりすることが期待されていると捉える。

### 2.4.1　普通教育としての技術教育に必要な手先の器用

普通教育としての技術教育に必要な手先の器用を定義するため，2.2.3 節の内容，2.3 節の内容，および普通教育としての技術教育で習得させたい技能とを見比べていく。

2.2.3 節では，手先の器用とは，巧緻性と，「新しい技能習得しやすい性質」などのそれ以外の要素に分類できる。これは，巧緻性には，「手指・指先の動きであること」「速く動かすこと」，「正確な動きであること」，「細かい精密な動きであること」などの特徴があるためである。

2.3 節では，生徒が器用・不器用を自認する理由の一つとして，「中学校技術・家庭の授業において，細かな作業や速く処理し，仕上げがきれいに出来るという作業処理・作業結果」がある。また，中学校技術・家庭（技術分野）の授業において，家庭でものづくりを経験してこなかった生徒が，実際よりも高い目標の技能を求め，道具を操作する活動が思ったようにできないと感じ，保護者は中学校技術・家庭（技術分野）の授業を通して，生徒に器用や道具操作の習熟が身に付いていないと捉えているとわかった。これは中学校技術・家庭の授業において，生徒が巧緻性を身につけることができるか，道具を操作できるようになるかを意識していると捉えることができる。そのため，生徒・保護者は生徒が中学校技術・家庭の授業において，道具を操作できる巧緻性を身につけることを求めていると言える。また保護者の求める「器用」が「手先の器用」と同義であれば，中学校技術・家庭の授業において，「新しい技能習得しやすい性質」などを身につけることも求められていると言える。

普通教育としての技術教育では設計・計画・創作の活動で精度を高めるためにも，まず一つの技能を習得させること，次に習得した技能をもとに新たな技能を習熟させることが求められる。日本産業技術教育学会では，中学校学習指導要領が平成 29 年に告示されて以降の刊行物 [119] で，技能を「材料や道具の特徴，くせを修得し，それを使いこなして目的を解決する技。」と説明している。そのため，普通教育としての技術教育でも生徒が 1 つの道具を使いこなすことを目指している。そして，使いこなせるようになった道具の操作経験から他の道具を使いこなす活動につなげることも求められていると判断した。これは，巧緻性と，「新しい技能習得しやすい性質」といった特徴をもつ「手先の器用」に通じる。したがって，普通教育としての技術教育では手先の器用が求められていると捉えた。

以上により，普通教育としての技術教育に必要な手先の器用は 2 つあり，「道具操作のための巧緻性（「速く」「細かく」「正確な」手指を使った動き）」と，「新しい道具操作技能を習得しやすい資質・能力」と定義することにした。また，これらの 2 つには学習する順番があり，①学習の初期段階に道具操作のための巧緻性を習得，②習得した「巧緻性」

をもとに，新しい道具操作に関連させたり，技能習得しやすかったりする資質・能力を養う，といった形になる。

## 2.5　本章のまとめ

本章では，普通教育としての技術教育に必要な手先の器用を整理するため，手先の器用の概念整理，器用に関する先行研究の整理，技術教育で習得させたい技能の順で確認した。

手先の器用には「速く」「細かく」「正確な」手指を使った動きを意味する「巧緻性」と，「新しい技能習得しやすい性質・才能」といった特徴が代表的なイメージであるとわかった。また，一般的なイメージにない専門的見地からは手先の器用について，「合理的な方法を身につけること」，「初期段階で新しい技能習得に時間がかかっても，最終的に到達する技能が低いとは限らないこと」，「器用の通用する動作の範囲が限定的であること」といった特徴もあるとわかった。

児童・生徒が器用・不器用を自認するようになる理由として，「自己・他者評価」，「生活経験・環境」，「中学校技術・家庭の授業において，細かな作業や速く処理し，仕上げがきれいに出来るという作業処理・作業結果」が挙げられる。そして，1990 年代から 2010 年代までの中学校技術・家庭（技術分野）の授業において，家庭でものづくりを経験してこなかった生徒が，実際よりも高い目標の技能を求め，道具を操作する活動が思ったようにできないと感じ，保護者は中学校技術・家庭（技術分野）の授業を通して，生徒に器用や道具操作の習熟が身に付いていないと捉えているとわかった。

技術教育で習得させたい技能として，学習の初期段階では技能を一つ習得させ，習得した技能を自身の経験や他の技能と関連付け，変化する状況や課題に応じて活用できる技能を習熟させることを求めていることがわかった。

以上の調査結果を見比べ，普通教育としての技術教育に必要な手先の器用は 2 つあり，「①道具操作のための巧緻性（「速く」「細かく」「正確な」手指を使った動き）」と，「②新しい道具操作技能を習得しやすい資質・能力」と定義することにした。また，これらの 2 つには①，②の順に学習する形になる。しかし，「器用の通用する動作の範囲が限定的である」という特徴も考えられるため，学習の初期段階に扱う道具を慎重に選ぶ必要がある。また，道具を操作する手指の動きを分析することも求められる。

次章では，習得した「道具操作時の巧緻性」を他の道具操作時の技能習得に関連付けやすく，学習の初期段階で習得する「道具操作時の巧緻性」の選定，すなわち，普通教育としての技術教育で使用される道具の中で，どの道具を選定すべきか検討していく

# 第 3 章

# 道具の選定

前章では，児童・生徒・保護者と社会から求められる，普通教育としての技術教育に必要な手先の器用を「道具操作のための巧緻性（「速く」「細かく」「正確な」手指を使った動き）」と，「新しい道具操作技能を習得しやすい資質・能力」と定義し，新しい道具操作技能と関連させやすく，普通教育としての技術教育の初期段階で扱う道具を選定する必要がある。本章では，1.3 節の課題 2 を明らかにするため，技術教育で指導されてきた道具操作とその研究方法や普通教育としての技術教育の初期段階で扱う道具に必要な条件から，簡易手指動作分析システムで操作時の手指動作を計測できる道具を選定し，選定した道具操作の技能と器用の評価との関係を整理していく。

## 3.1　技術教育で指導されてきた道具操作とその研究方法

技術教育で指導されてきた道具操作とその研究方法を明らかにするため，普通教育としての技術教育で使用する道具・機械の操作について研究してきた学会として，日本産業技術教育学会での活動をもとに分析していく。日本産業技術教育学会の活動を分析するため，日本産業技術教育学会が発行した，第 1 巻・第 1 号 (1958) から第 63 巻・第 3 号 (2021) までの会誌「日本産業技術教育学会誌」[20] を確認した。

日本産業技術教育学会では様々な方法で児童・生徒を対象に，道具・機械を操作できるかを確かめたり，道具・機械を操作する力を養ったりしている。会誌「日本産業技術教育学会誌」から，日本産業技術教育学会では以下の 2 種類に大別した，道具・機械操作について研究している。

- 「かんな，のこぎり，げんのう，卓上ボール盤，旋盤，角のみ盤」といった，身体の位置や腕の動き，しっかりと固定する持ち方に重点をおく道具・機械の操作
- 「ドライバ，さしがね，スコヤ，キリ，ニッパ，はけ，キーボード」といった，手

指の形や速く細かく正確な動きに重点をおく道具の操作

本研究では「速く」「細かく」「正確な」手指動作を分析したいため，前者の道具・機械操作は不向きと判断した。次に，後者の道具操作時の手指動作をさらに分類すると，「さしがね，スコヤ，はけ」といった，片方の手ではえんぴつを握る形を維持するのに時間をかける操作となる。また，設計・計画・創造の活動における精度を高めるために道具操作時の手指動作の特徴を明らかにしたいため，キーボードも本研究に不向きと判断する。以上により，「ドライバ，キリ，ニッパ」のいずれかが速く細かく正確な手指動作を求めるため，向いていると思われる。

次に，道具・機械を操作する手先の器用に関する研究としては，以下の 8 種類に大別できる。

No.1： 道具・機械操作の練習方法を工夫した指導を行ったり，操作の結果うまれた作品の完成度で手先の器用を分析したりする研究 [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [34] [35] [36] [37] [38] [39]

No.2： 動画で作業時の様子や要点を撮影・表示し，手先の器用を身につけさせたり分析したりする研究 [40] [41] [42] [43] [44] [45] [46] [47]

No.3： 道具にかかる力や速さを数値化したりグラフ化したりした様子を確認し，手先の器用を身につけさせたり分析したりする研究 [48] [49] [50] [51] [52] [53] [54] [55] [56] [57]

No.4： 改良した教具を用い，道具操作を練習し，手先の器用を身につけさせたり分析したりする研究 [58] [59] [60] [61] [62]

No.5： 筋電計を用い，道具を操作する筋肉の反応を分析し，手先の器用を分析する研究 [63] [64] [65]

No.6： 握力をもとに，道具を操作する手先の器用を分析する研究 [66] [67]

No.7： シミュレーションソフトを用い，道具・機械を操作する要点を示したり，結果を分析したりし，手先の器用を分析する研究 [68] [69]

No.8： モーションキャプチャシステムを用い，道具を操作する際の動作を表示したり分析したりし，手先の器用を身につけさせたり分析したりする研究 [70] [71] [72] [73]

No.1，3，4〜7 に分類した先行研究をみると，道具操作に関連する要素を数値で求め，求めた数値を「定量的に分析し，手先の器用を明らかにしようと試みた報告がある。数値化して道具操作時の動作を分析する方法として，モーションキャプチャシステムを用いた計測・分析の方法がある。しかし，「ドライバ，キリ，ニッパ」をモーションキャプチャによる計測・分析した研究が管見の限り見当たらない。身体の位置や腕の動かし方などが重要となる道具操作において，対象物を固定することは重要であるが，それは「ドライバ，キリ，ニッパ」でも変わらないだろうか。ドライバ操作では，片方の手指をよく動かし，

柄を回すため，対象物を固定しにくいと思われる。キリ操作では，手を広げた状態で操作するため，対象物を固定するために持つといった手の動かし方でない。ニッパ操作では先行研究において，被膜とりをしている。被膜とりをするニッパ操作時の手では，柄をどう握ったり動かしたりするより，ニッパの先にかかるほどよい力へと調整する活動が重要となるため，モーションキャプチャによる計測には不向きと判断した。以上により，モーションキャプチャシステムで計測・分析する対象として，「ドライバ」操作時の手指動作であれば，速く細かく正確な手指の動きの合間で，ドライバを固定する必要がどれだけあるかを確認しやすい。すなわち，普通教育としての技術教育では今後，操作時に重点をおくのが身体動作の道具だけでなく手指動作の道具操作の場合にも，固定することが重要となってくるのかを検討する必要があると考える。しかし，日本産業技術教育学会だけでなく，他の条件とあわせ，普通教育としての技術教育において重要と判断される道具を選定すべきである。そこで，次節で条件整理と選定する道具を検討していきたい。

## 3.2 道具の選定

道具操作時の手先の動きを抽出できる道具の選定を行う。分析対象の道具には次のような条件を求めたいと考えた。

1. 技術・家庭（技術分野）の授業で使用する道具
2. 日常生活で使用する道具
3. 「速く」「細かく」「正確な」手指を使った動きが求められる道具
4. 新しい道具に対応できるよう，操作原理が他の道具の操作でも転移しそうな道具

1 を条件にした理由は，普通教育としての技術教育で使用する道具を選ぶことで，設計・計画・創作だけでなく，製作といったものづくり活動全般で役立つ機会を設けることができるためである。また，技術・家庭（技術分野）の授業で道具操作時の巧緻性を確かめやすくするためでもある。

2 を条件とした理由は，日常生活で使用する道具を選ぶことで，学習内容・成果を日常生活につなげやすかったり確認しやすかったりするためである。

3,4 を条件とした理由は，第 2 章で定義した「普通教育としての技術教育に必要な手先の器用」の説明から，条件に加えたためである。また，ペグボードのように，明らかにしたい手指の巧緻性に合わせ，試験方法を考案する研究のアプローチも考えられる。しかし，開発した試験方法が目的に一致した内容になっているかを明らかにしたり，手先の器用別にどういった手指の特徴があるかという習熟度 (指標) を作成したり，習熟した手指の巧緻性が普通教育としての技術教育にどれだけつながるのかを分析したりするのは，研究に 3 年以上の年月を必要とすると考え，本研究では，その手法を扱わないことにしたた

めである。

### 3.2.1　技術・家庭（技術分野）の授業で使用する道具

技術・家庭（技術分野）の授業で使用する道具を明らかにしていく。技術・家庭（技術分野）の授業で使用する道具を調査するため，教授用に使われる生徒用図書である技術・家庭（技術分野）の教科書を選んだ。教科書では，技術・家庭（技術分野）で新たに学習する語句や，学習内容を扱う際に必要となる語句を「索引」にまとめている可能性がある。そこで，各教科書の索引にある道具・機械を調査した。

教科書会社は，開隆堂，東京書籍，教育図書の 3 社を選んだ。また，平成 29 年改訂前後の教科書を選定することにより，索引にある道具・機械が平成 29 年改訂以前から技術科で利用されやすいのかを確認した。そのため，平成 28 年と令和 3 年に発行された教科書の 6 冊と，教育図書が令和 3 年に発行した教科書の付録 1 冊の計 7 冊 [11][12][13][126][127][128][129] の索引を調査した。なお，開隆堂が平成 28 年に発行した教科書の索引には道具・機械が載っていないため，調査対象から外した。

調査の結果，令和 3 年に発行された 4 冊すべての索引にある道具・機械は「げんのう，さしがね，卓上ボール盤，ねじ回し（ドライバ)」であった。そのうち，平成 28 年に発行された 2 冊の索引にも教科書の索引に記載されている道具・機械は「さしがね，卓上ボール盤，ねじ回し（ドライバ)」であったため，2015 年以降，これら 3 つの道具・機械は技術科の学習内容を象徴する道具・機械だと言える。また，2010 年代にも，さしがねや卓上ボール盤，ねじ回し，げんのうは中学技術・家庭（技術分野）の授業で扱われている報告 [130]_[132] も確認できた。

### 3.2.2　日常生活で使用する道具

技術・家庭（技術分野）の授業で使用する道具の中で，日常生活でも使用されているかを確認していく。

長谷川ら [133] は 1970 年代～2000 年代にかけて児童の道具・機械使用について調査を行い，ドライバとかなづちの操作経験をきく質問が選ばれている。また，時代によらず，ドライバとかなづちを使用したことがあると回答した小学 6 年生の割合が 80 %以上で変化がないこともわかる。また，子どもの生活技術のあり方を科学的に調査することを研究活動の主なテーマにする「子どもの生活科学研究会」の刊行物 [134] では，子供とマスターする生活技術の代表例として，ドライバ操作とかなづち操作が紹介されている。これらの調査や文献には，さしがねや卓上ボール盤が登場してこない。そのため，ドライバやかなづちはさしがねや卓上ボール盤に比べ，日常生活でも使用する道具であると考えられる。

また，長谷川らの報告 [133] から，1970 年代～2000 年代で，ドライバを操作したことがあると回答した小学 6 年生の割合が 80 %以上で変化がなかった。しかし末富ら [138] が 1970 年代に調査した報告では，小学 6 年生男子の割合が 80 %以上，小学 6 年生女子の割合が 50 %程度であった。1970 年代～2000 年代の調査 [133]_[142] では，「①女子児童が男子児童よりドライバ使用経験が少ない，②女子児童は男子児童よりネジ締めができない」ことが調査の結果としてまとめられている。そのため，ドライバ使用経験が少ないことが，ドライバによるネジ締めができないことにつながる道具操作だと言える。これらの調査ではドライバ操作経験のアンケート調査や実際にネジ締めを行っているが，ネジ締めを行っている時に手先の動きに着目した研究はない。

有川と野方は，中学生を対象に，手動式ドライバ操作の技能指導 [37],[39] を行っている。また，有川らは，実験器具でドライバ操作時の押す力や回す力，ドライバ軸の傾きを計測し，操作者の巧緻性に繋がる手動式ドライバ操作の指導上の要点 [57] を明らかにしている。有川らは指導上の要点を明らかにするため，生徒がドライバ操作する様子をみて，ドライバ操作技能を以下の 4 段階に分類している。

1. 操作 (木ねじを締め込む・緩めて外す) を最後まで完了することができない。
2. 操作の際，軸ずれが大きく，あるいは軸ぶれが多く， かつ操作の手際が悪い (時間を要する)。
3. 操作の際，軸ずれが大きく，あるいは軸ぶれが多く，または操作にもたつきがある。
4. 操作の際，軸ずれは小さく，軸ぶれも少なく，かつ 操作にもたつきがなく手際がよい。

この評価を用い，ドライバ操作時の手指動作を割り当てていけば，速く細かく正確な手指動作とはどういった特徴をもつのかを得やすいと考えられる。しかし，ドライバ操作時の手指動作を定性的に分析しているが，定量的に分析できていない。また，中学生が，ドライバの先端とネジ頭部の溝の大きさが合わない条件下でドライバ操作を行っているが，条件の違いによる分析が行われていない。

### 3.2.3 「速く」「細かく」「正確な」手指を使った動きが求められる道具

「速く」「細かく」「正確な」手指を使った動きが求められる道具操作を確認するためには，手指の巧緻性が身体全体の動きを使った粗大運動に比べ，道具操作に必要となる道具を選定していく。

1970 年代からの日本産業技術教育学会誌において，児童・生徒における道具操作の実態を確認した。その結果，のこぎり引きやかんながけといった木工具の操作時の児童・生徒の実態 [48][49][21][70][23][28][73] が調査されている。これらの動きは技能習得がしにくく，習

得による違いが分かりやすいため分析されてきたと考えられる。しかし，これらの木工具は身体全体の動きや腕の動きが重要であり，手指の細かい動きというより握り方が求められる。そのため，のこぎりやかんながけは本研究での「普通教育としての技術教育に必要な手先の器用」を測る上での道具に不向きと判断した。しかし，その研究方法はこれからの研究には有効である。のこぎりやかんながけの研究では動きの違いを比較するため，熟練者と非熟練者の動きを分析し，その動作の特徴を抽出している。そのため，手先の器用を明らかにするためには，運動発達の段階が異なる群どおしの動きを分析するアプローチも有効と考えられる。

### 3.2.4　新しい道具に対応できるよう，操作原理が他の道具の操作でも転移しそうな道具

新しい道具に対応できるよう，操作原理が他の道具の操作でも転移しそうな道具として，ドライバがかなづちより適していると判断する。その理由について述べていく。

ドライバにはてこの原理が働き，小学校 6 年生の理科で「てこの原理」[135] を扱われている。下田ら [136] は，小学 6 年生を対象にてこの原理で，ドライバの操作から生活物の仕組みへと学習の転移できることを明らかにしている。稲田ら [137] は，中学 3 年生を対象にてこの原理で，看護分野での身体の動かし方を指導し，理科の有用性を感じるようになったことを明らかにしている。そのため，てこの原理を踏まえ道具操作の規則性を学習すれば，その後使用する道具操作の熟達も期待できる。

## 3.3　本章のまとめ

本章では，「普通教育としての技術教育に必要な手先の器用」を確認するため，「技術・家庭（技術分野）の授業で使用する道具」，「日常生活で使用する道具」，『「速く」「細かく」「正確な」手指を使った動きが求められる道具』，「新しい道具に対応できるよう，操作原理が他の道具の操作でも転移しそうな道具」といった条件にある道具を選定した。その結果，ドライバを選定することにした。

ドライバを選定する過程で，ドライバ操作時の手指動作を進めるための情報や研究アプローチをえることができた。

- ドライバ使用経験が少なければドライバ操作技能が小さくなる
- ドライバ操作技能を 4 段階に分類されている。
  1. 操作 (木ねじを締め込む・緩めて外す) を最後まで完了することができない。
  2. 操作の際，軸ずれが大きく，あるいは軸ぶれが多く， かつ操作の手際が悪い (時間を要する)。

3. 操作の際，軸ずれが大きく，あるいは軸ぶれが多く， または操作にもたつきがある。
4. 操作の際，軸ずれは小さく，軸ぶれも少なく，かつ 操作にもたつきがなく手際がよい。

- ドライバ操作時の手指動作を定量的に分析されていない
- ネジとドライバの条件があっていない場合での分析がされていない
- 先行研究では運動発達の段階が異なる 2 群間の動きを比較し，動きの特徴を分析する

そこで，手先の器用を明らかにするため，ドライバ操作時の手指動作を定量的に分析していくこととする。また，本研究では「普通教育としての技術教育に必要な手先の器用」の中でも，「「速く」「細かく」「正確な」手指を使った動き」をどれくらい習得すればよいのか，習得にはどういった要因があるのかを明らかにしていく。そのためには，ドライバ操作の技能指導を行い，習熟度別にどういった特徴があるかを得たい。ドライバ操作の技能指導を行うためには，ドライバ操作時の手指動作における要点を明らかにする必要がある。そこで，「運動発達の段階が異なる 2 群間での動きを比較し，動きの特徴を分析すること」，「ネジとドライバのサイズがあっていない条件下での動作の違いを分析すること」をする必要がある。

次章では，本研究で使用する，手指動作を定量的に調べることができるシステムについて述べていく。

# 第 4 章

# 簡易手指動作分析システムの開発

前章では,「普通教育としての技術教育に必要な手先の器用」を確認するため，ドライバ操作時の手指動作を定量的に調べることができるシステムが必要であると考えた。本章では，道具操作時の手指動作を計測できるモーションキャプチャシステムとして選定した，簡易手指動作分析システムについて解説する。

## 4.1　開発方針

手指動作の分析方法には，モーションキャプチャにより，手指の動作を計測したり，定量的に把握したりする方法がある。それらの方法は，複数台のカメラを配置したり [120]，手指にマーカを装着させたり [120]，センサ付グローブをはめたりして手指の動作を計測する [121]。しかし，これらの計測方法は，計測の度にマーカを指につけるため，マーカの位置がずれてしまったり，グローブをはめた状態で作業すると手の感覚が変わり正確な作業ができなくなったりする。また，モーションキャプチャできるよう複数台のカメラを配置する必要もある。筆者らは，教育現場で利用できるシステムの開発を考えている。そのため，既存のシステムでは，必要とされる装備が多く，小中学生が学習する教育現場で扱うには不向きであると判断した。そこで，マーカなどを装着したり，複数台のカメラを配置したりせずに，モーションキャプチャを可能にする方法を検討した。

検討の結果，開発するシステムには手指動作の計測からはじまり，手指動作を定量的に分析するまでの流れを実現するため，以下の機能が求められると判断した。

1. マーカなどを装着したり，複数台のカメラを配置したりせずに，手指を検出する。
2. 関節各部位の座標値を取得できる。
3. 取得した手指のデータを時系列に保存できる。
4. 保存したデータから，アニメーションで再現したり，指先や関節各部位の変位をグ

ラフ化したりすることができる。

以上 4 つの機能の実現を開発の方針とし，システムの開発を行った。また，開発したシステムを簡易手指動作分析システムと呼称した。

## 4.2　システムの概要

簡易手指動作分析システムは Leap Motion を用いてモーションキャプチャを行う。Leap Motion により，手指の動きを検出し，手指の指先と各関節部位の位置座標（単位は mm）を取得できるため，取得したデータを CSV 形式で保存する。そして，取得した位置座標を基に，関節部位で構成するアニメーションで動作を再現したり，指先と関節各部位の変位をグラフ化したりすることによって，動作を分析する。図 4.1 にシステムの概要を示す。また，図 4.2 に Leap Motion と PC を接続したシステムの概観を示す。

簡易手指動作分析システムは，モーションキャプチャのプログラムと動作分析のプログラムの二つから構成されている。以下では，各プログラムについて述べていく。

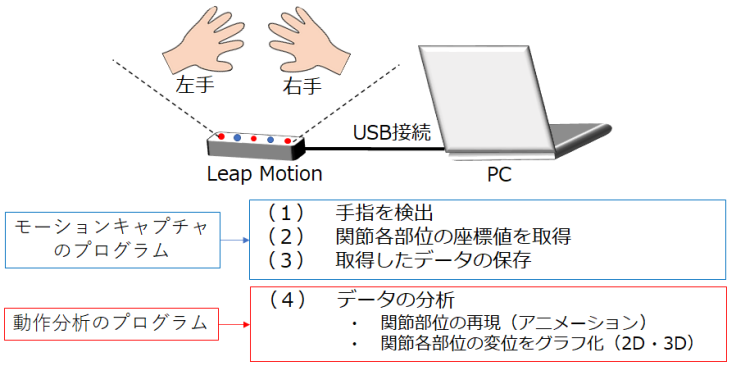


図 4.1　簡易手指動作分析システムの概要

## 4.3　モーションキャプチャのプログラム

手指動作を検出し，検出したデータの保存を可能にするプログラムについて述べる。

### 4.3.1　手指の検出

開発の方針 1,2 を可能とする，手指動作を検出する計測デバイスとして，Leap Motion を用いている。Leap Motion　とは Leap Motion 社から販売された，マウス操作や画面タッチをせず，手や指，ペンなどの先端がとがった道具のジェスチャーによってコン

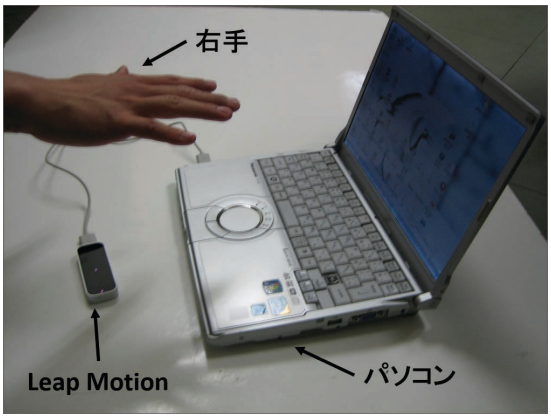


図 4.2　LeapMotion と PC を接続したシステムの概観

ピュータを操作するための入力デバイスである [143][144]。Leap Motion は 3 つの LED による赤外線の照射と，2 つのカメラによる赤外線画像のみの撮影によって，手指の動きを検出する。そのため，マーカなどを装着したり，複数台のカメラを配置したりせずに，パソコンとの USB 接続で手指のモーションキャプチャが可能となる。

### 4.3.2　プログラムの概要

モーションキャプチャのプログラムは，Processing[145][146] を用いて作成した。作成したプログラムは，開発の方針 1〜3 に相当する。プログラムでは，Leap SDK [147] を利用して手指を検出し，関節各部位の座標値を保存できる。図 4.3 に作成したプログラムのフローチャートを，図 4.4 にプログラムの実行画面を示す。

図 4.4 の (1) にある「はじめ」のボタンにより，モーションキャプチャを開始する。モーションキャプチャを開始すると，Leap Motion にかざした手指を検出し，関節各部位の座標 ( x ， y ， z ) を取得する。各座標 ( x ， y ， z ) は，Leap Motion を中心とした左右の向きが x ，上下の向きが y ，前後の向きが z の座標値 (単位は mm) で記録される。同時に PC 画面では，取得したデータを基に，画面上でアニメーションと，モーションキャプチャの経過時間がリアルタイムで表示される。このモーションキャプチャの経過時間は，図 4.4 の (3) のように画面左下に表示した。作成したプログラムでは計測した時間と取得した値を CSV 形式で保存できる。キャプチャは，図 4.4 の (1) にある「おわり」のボタンにより終了する。また図 4.4 の (2) にある「目印を付けた経過時刻」ボタンにより，保存データに目印を付けることができる。ボタンをクリックすることで，その時の経過時刻が同じデータファイル内に追加保存される。PC 画面では，クリックした後の時間が秒単位で表示される (図 4.4 の (4))。「目印を付けた経過時刻」ボタンを複数回クリックした場合，そのたびごと経過時間は保存される。図 4.4 の (4) に示す時間は最後にクリッ

クされてからの時間を示す。これによって，保存したデータの中から分析したい箇所を特定しやすくなる。

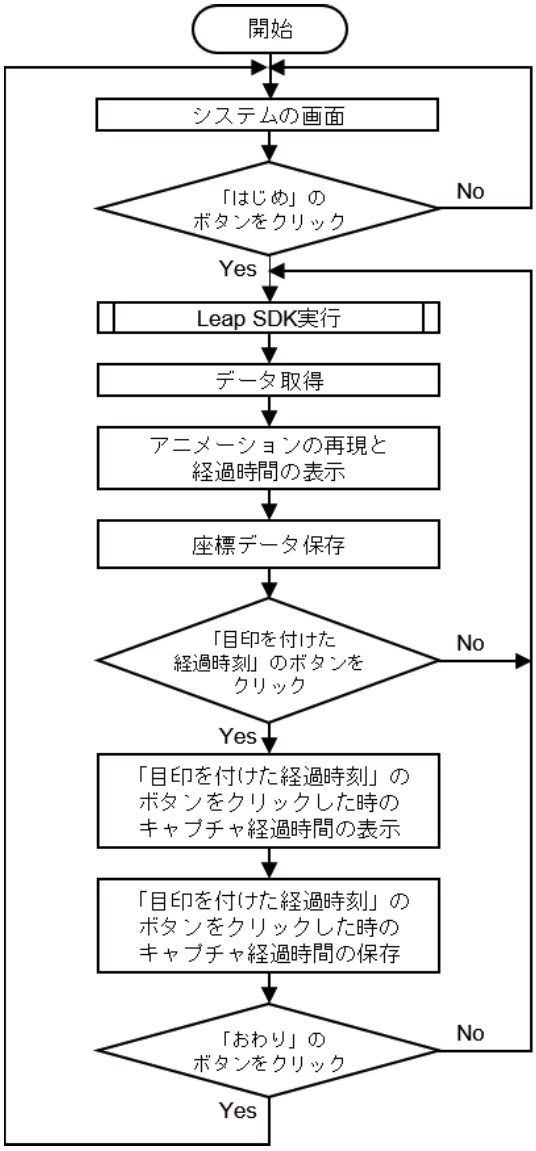


図 4.3　プログラムのフローチャート

### 4.3.3　データの利用

データは関節各部位と指先における x， y， z 座標の値 (mm) を計測しているため，CSV 形式のデータを利用できるアプリケーションソフトによって開発の方針 4 が可能となる。図 4.5 に保存したデータの一部を Microsoft 製表計算ソフト Excel で読み込んだ画像を示す。図 4.5 の (1) に示す Timestamp は，モーションキャプチャ開始後の経過時

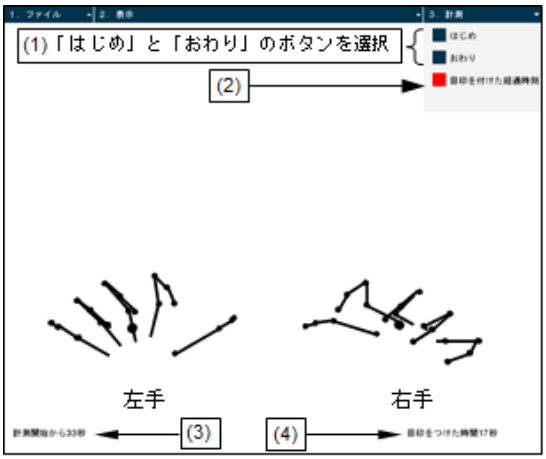


図 4.4　計測時に画面上で再現されるアニメーション

間である。図 4.5 の (2) が，親指の関節各部位と指先の x， y， z 座標の値である。親指の関節部位は手首の関節から指先へと順に，図 4.7 の (9) に示す片手で合計 25 点の指先と関節各部位の座標 ( x， y， z ) が格納される。図 4.5 にある (3) の部分が人差し指の手首にある関節の x， y， z 座標となる。

(1) (2) (3)

| Timestamp | Thumb(1) | | | Thumb(2) | | | Thumb(3) | | | Thumb(4) | | | Index(1) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | x | y | z | x | y | z | x | y | z | x | y | z | x | y | z |
| 0 | -69.1603 | 100.4143 | 104.5709 | -86.2389 | 106.0444 | 61.57177 | -88.6281 | 102.7782 | 29.99508 | -84.5081 | 96.51937 | 9.468101 | -55.107 | 117.0782 | 100.7416 |
| 0.104087 | -34.7605 | 92.30145 | 104.8602 | -48.9858 | 105.5107 | 62.48741 | -50.5769 | 110.4319 | 31.0754 | -53.0097 | 114.5093 | 9.745586 | -23.235 | 111.1939 | 105.1515 |
| 0.138718 | -24.4924 | 92.70902 | 103.2004 | -38.9522 | 106.3396 | 61.04097 | -41.7284 | 112.3368 | 29.89951 | -46.474 | 117.8048 | 9.281949 | -13.4375 | 111.8759 | 103.7159 |
| 0.173364 | -14.3024 | 92.58759 | 102.5645 | -29.7739 | 106.2962 | 60.79126 | -34.1957 | 112.9273 | 29.97016 | -41.2953 | 119.2889 | 10.30726 | -3.54727 | 111.9263 | 102.9944 |
| 0.208075 | -6.8497 | 92.29932 | 101.3767 | -23.0491 | 105.8647 | 59.83328 | -28.7617 | 112.8344 | 29.30044 | -37.6651 | 119.7252 | 10.57211 | 3.653487 | 111.7793 | 101.6172 |
| 0.24376 | -0.95105 | 91.64794 | 100.4359 | -17.6828 | 105.5921 | 59.23018 | -24.4033 | 113.1294 | 29.03934 | -34.6741 | 120.6214 | 11.26611 | 9.487687 | 111.1629 | 100.6443 |
| 0.278151 | 3.740233 | 90.78629 | 99.50557 | -13.6696 | 105.2771 | 58.77214 | -21.3622 | 113.4545 | 28.98257 | -32.5927 | 121.4026 | 12.00585 | 14.04534 | 110.3719 | 99.73369 |
| 0.294773 | 5.184002 | 90.44222 | 98.89446 | -12.4673 | 105.2889 | 58.39378 | -20.5883 | 113.844 | 28.82481 | -32.2576 | 122.0473 | 12.27072 | 15.43069 | 110.0579 | 99.16718 |
| 0.329842 | 7.959727 | 90.14727 | 98.08575 | -10.1875 | 105.5884 | 58.02893 | -19.0324 | 114.7392 | 28.84853 | -31.2506 | 123.2853 | 12.8741 | 18.1116 | 109.8124 | 98.34136 |
| 0.381473 | 10.78065 | 89.32322 | 97.10824 | -7.91112 | 106.1743 | 57.8787 | -17.4488 | 116.4985 | 29.31461 | -30.154 | 125.6465 | 14.07001 | 20.73396 | 109.084 | 97.6451 |
| 0.416178 | 12.21239 | 88.81097 | 96.38148 | -6.87359 | 106.7722 | 57.84039 | -16.9916 | 118.0264 | 29.83263 | -30.1011 | 127.6478 | 15.23558 | 22.04686 | 108.624 | 97.13866 |
| 0.450819 | 13.24607 | 88.89005 | 95.99246 | -6.21239 | 107.6927 | 58.04345 | -16.8121 | 119.6393 | 30.5046 | -30.2896 | 129.6166 | 16.49337 | 22.99495 | 108.7412 | 96.85127 |
| 0.485579 | 13.5501 | 89.41195 | 95.69452 | -6.17397 | 108.7219 | 58.13942 | -17.1487 | 121.1023 | 30.94115 | -30.8264 | 131.2804 | 17.27255 | 23.24484 | 109.2884 | 96.57943 |
| 0.52032 | 13.6628 | 89.33987 | 96.49378 | -6.29068 | 108.9623 | 58.22295 | -17.5969 | 121.6308 | 31.29447 | -31.4394 | 131.9245 | 17.88069 | 23.30695 | 109.239 | 96.41969 |
| 0.555031 | 13.42699 | 88.81903 | 95.17197 | -6.86823 | 108.6693 | 58.20752 | -18.5305 | 121.5502 | 31.53285 | -32.4813 | 131.8983 | 18.27411 | 23.0047 | 108.748 | 96.14671 |
| 0.589673 | 11.42117 | 87.40284 | 93.93726 | -9.21782 | 107.6996 | 57.4082 | -21.2694 | 120.9498 | 31.08986 | -35.3361 | 131.4515 | 18.07653 | 20.99003 | 107.3295 | 95.03667 |
| 0.624405 | 7.879357 | 85.87441 | 91.75 | -12.6971 | 106.6112 | 55.43351 | -24.9438 | 120.2346 | 29.39726 | -39.0543 | 130.9456 | 16.60387 | 17.57511 | 105.7305 | 93.00191 |
| 0.659218 | 5.51428 | 84.60803 | 90.81512 | -15.3538 | 105.5937 | 54.80928 | -27.9828 | 119.445 | 29.07764 | -42.1798 | 130.2525 | 16.46235 | 15.26752 | 104.4329 | 92.1168 |
| 0.693756 | 2.705395 | 82.86668 | 90.17291 | -18.5834 | 104.2904 | 54.67488 | -31.6877 | 118.4939 | 29.37662 | -46.0151 | 129.495 | 17.0801 | 12.62099 | 102.606 | 91.54627 |
| 0.711056 | 1.107993 | 81.52591 | 90.22122 | -20.3096 | 103.276 | 55.00024 | -33.5478 | 117.7114 | 29.90378 | -47.8804 | 128.8629 | 17.74952 | 11.15728 | 101.193 | 91.65694 |
| 0.74578 | -2.90887 | 76.74758 | 90.21637 | -24.2021 | 99.73219 | 55.71126 | -37.3733 | 115.0142 | 31.08515 | -51.4636 | 126.6916 | 19.14336 | 7.506706 | 96.1853 | 92.0965 |
| 0.780497 | -6.17422 | 69.30174 | 90.32102 | -26.9174 | 94.21468 | 56.83427 | -39.6674 | 110.7993 | 32.83732 | -53.2095 | 123.2955 | 21.09173 | 4.79846 | 88.33717 | 92.98496 |
| 0.815165 | -8.54952 | 61.44109 | 90.55188 | -27.82 | 87.90842 | 57.37922 | -39.4065 | 105.4767 | 33.49251 | -51.8559 | 118.6863 | 21.32613 | 3.110447 | 79.93331 | 94.00545 |
| 0.849876 | -9.30308 | 49.84305 | 91.22181 | -26.785 | 78.07864 | 58.51965 | -36.8053 | 96.73189 | 34.74749 | -47.9906 | 110.6878 | 22.19266 | 2.883046 | 67.71901 | 95.89006 |
| 0.884443 | -9.23857 | 39.16085 | 91.60234 | -25.0611 | 68.85781 | 59.35275 | -33.5814 | 88.36497 | 35.68128 | -43.7471 | 102.9662 | 22.99403 | 3.386672 | 56.40352 | 97.35889 |
| 0.919206 | -9.42666 | 31.33398 | 91.56009 | -23.6872 | 61.69702 | 59.20239 | -30.9452 | 81.53304 | 35.38396 | -40.2607 | 96.47792 | 22.44657 | 3.600646 | 48.01157 | 98.0402 |
| 0.953941 | -9.96931 | 26.86194 | 91.69518 | -22.8933 | 57.27981 | 58.8315 | -29.2359 | 77.07921 | 34.72301 | -38.0628 | 92.16131 | 21.60353 | 3.39365 | 43.11357 | 98.56249 |
| 0.988579 | -11.0563 | 23.69996 | 92.41597 | -22.6089 | 53.98549 | 58.92537 | -28.1548 | 73.68143 | 34.5373 | -36.5918 | 88.85561 | 21.26836 | 2.591551 | 39.56954 | 99.60887 |
| 1.02326 | -11.3209 | 20.95372 | 93.67829 | -21.9325 | 50.80552 | 59.49496 | -26.9745 | 70.22558 | 34.776 | -35.3113 | 85.35376 | 21.39326 | 2.412481 | 36.65241 | 101.0803 |

図 4.5　保存されたデータ

## 4.4　動作分析のプログラム

保存されたデータを読み込み，各関節と指先の変位の様子を「アニメーションによるモーションの再現」,「時間経過における変位」及び「空間における変位」の 3 つを可能にするプログラムについて述べる。

### 4.4.1　画面について

作成したプログラムは動作画面と調整画面の 2 つで構成されている。図 4.6 に動作画面を示す。図 4.7 に調整画面を示す。

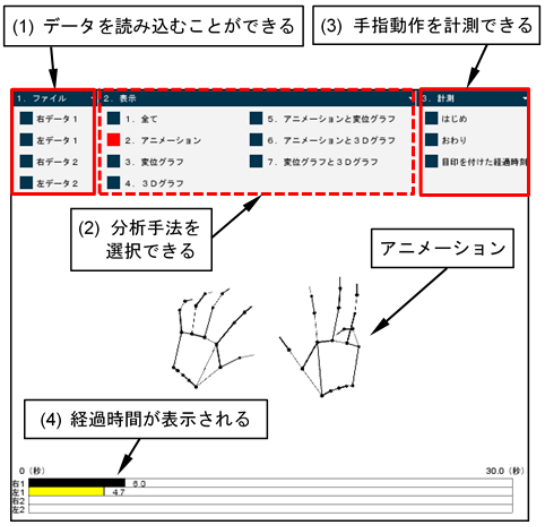


図 4.6　動作画面

**動作画面**

図 4.6 の動作画面では，手指動作を計測したり，データを読み込んだり，読み込んだデータを基に分析手法を表示したりすることができる。図 4.6 の (1) に示すボタンでは，モーションキャプチャされたデータを右手左手それぞれ 2 人分読み込むことができる。図 4.6 の (2) に示すボタンでは，「アニメーションによるモーションの再現」,「時間の経過における変位」,「空間における変位」及び「空間におけるアニメーションの再現」のいずれかを表示する。図 4.6 の (3) では，手指動作を計測することができる。図 4.6 の (4) では経過時間が表示される。またマウスクリックによって経過時間を選択することもできる。

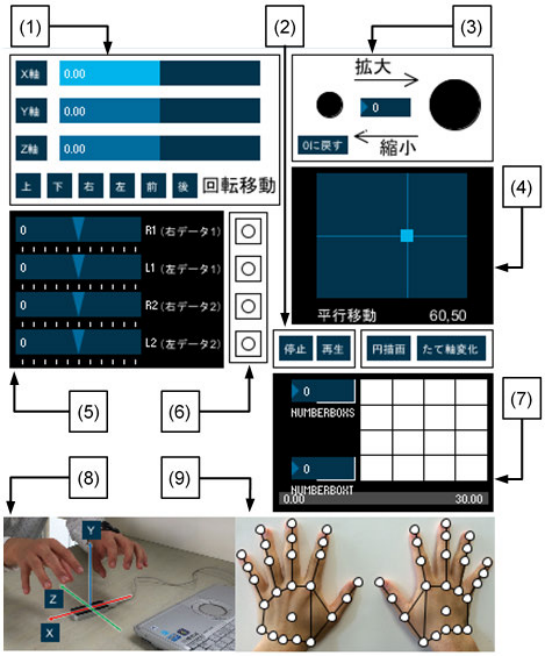


図 4.7　調整画面

### 調整画面

図 4.7 は，表示された分析手法を微調整する機能を集めた画面である。図 4.7 には 9 つの機能が備わっている。図 4.7 の (1) では，アニメーションや 3D グラフを回転させることができる。図 4.7 の (2) にある 2 つのボタンでは，両手のアニメーションを再生と停止する。図 4.7 の (3) では，アニメーションや 3D グラフを拡大・縮小させることができる。図 4.7 の (4) では，動作画面に表示されたアニメーションや 3D グラフの位置を上下・左右に動かすことができる。図 4.7 の (5) では，左右のモーションをそれぞれ 5 倍速までの早送りと巻き戻しをする。図 4.7 の (6) において，動作画面に表示させたい手を選択することができる。図 4.7 の (7) では，時間における変位グラフの縦軸と横軸の最大値・最小値を，それぞれドラック＆ドロップすることで調整する。図 4.7 の (8) において x ， y ， z ボタンをクリックすることで，表示したい変位グラフの座標値を選択することができる。図 4.7 の (9) では，手指の先端，各関節及び重心の中から 1 つを選択する。これによって，図 4.7 に表示されるアニメーションや 3D グラフを様々な角度から見たり，時間における変位グラフを細分化したりすることができる。表 4.1 に，図 4.7 にある機能のいずれが各分析手法で行えるのかを示した。

表 4.1　各分析手法と機能との関係

| 分析手法 | アニメーション | 変位グラフ | 3D グラフ | アニメーションと 3D グラフ |
|---|---|---|---|---|
| ①回転 | ○ | × | ○ | ○ |
| ②再生・停止 | ○ | × | × | ○ |
| ③拡大・縮小 | ○ | × | ○ | ○ |
| ④実行画面の調整 | ○ | × | ○ | ○ |
| ⑤速度調整 | ○ | × | × | ○ |
| ⑥実行画面に表示 | ○ | ○ | ○ | ○ |
| ⑦縦軸と横軸の調整 | × | ○ | × | × |
| ⑧関節の選択 | ○ | ○ | ○ | ○ |
| ⑨縦軸の選択 | × | ○ | × | × |

○：分析手法に対してその機能が扱えることを意味する。
×：分析手法に対してその機能が扱えないことを意味する。

### 4.4.2　アニメーションによるモーションの再現

図 4.6 の (2) にある「2. アニメーション」ボタンにより，読み込んだデータから手指のモーションをアニメーションで再現することができる。これによって，指先と各関節の動きを視覚的に捉えることができる。図 4.6 には動かしたいデータのアニメーションをマウスの動きに合わせて移動させる機能が備わっている。これによって，2 つのアニメーションが重なっても分析しやすくなる。図 4.8 の (a) と (b) にそれぞれ 2 つのアニメーションを重ねて表示した様子と分離した様子を示す。

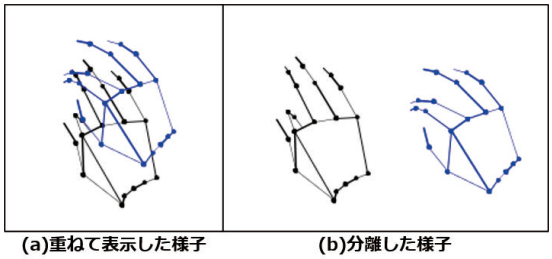


図 4.8　2 つのアニメーション画像

### 4.4.3　時間の経過における変位の比較

図 4.6 の (2) で「3. 変位グラフ」ボタンをクリックすると，指定した関節や指先の動きを指定した座標軸での時間変位でグラフ表示する。図 4.9 に実行画面に表示されたグラフを示す。実行画面に表示したい時間変位の方向は図 4.7 の (8) の x，y，z ボタンのクリックで指定し，表示する関節や指先は図 4.7 の (9) の手指関節モデル内の○印をクリックすることで選択する。また図 4.9 の右上には縦軸と横軸の尺度が表示される。図 4.10 に 2 人の学生がキーボード操作を行っている時の，右手人差し指の指先における x 成分の時間による変位を示したものを示す。

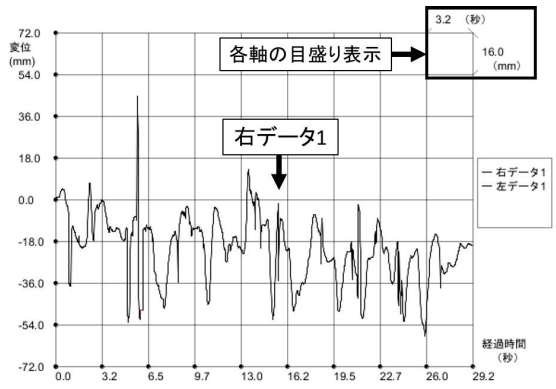


図 4.9　実行画面での時間と変位 (1 人)

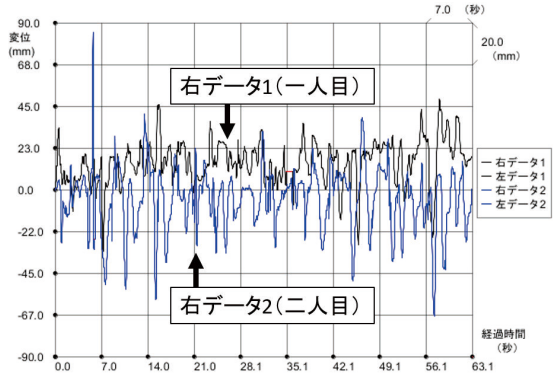


図 4.10　実行画面での時間と変位 (2 人)

### 4.4.4　空間や平面における変位の比較

図 4.6 の (2) にある「4. 3D グラフ」ボタンにより，読み込んだデータから各関節と指先における空間内の軌跡をグラフ化することができる。図 4.11 は 2 人の学生がラジオペンチを 10 回開閉した時の，人差し指指先の動きの軌跡を実行画面に表示した 3 次元グラフである。Leap Motion から見て水平方向 (左右の方向) がグラフの x 軸，垂直方向 (上下方向) が y 軸，距離の手指関節モデルの丸印を任意に選択できる。図 4.11 の右 (奥行き) が z 軸になる。関節の指定は，図 4.7 にある (9) で行い，図 4.11 上には各平面の目盛りが縦横ともに 20mm であることが表示されている。図 4.11 に示した 3 次元グラフを xy，yz，zx の各平面について表示した 2 次元グラフを図 4.12 に示す。図 4.12 にある各平面で見たとき，図 4.11 のように 2 つの変位の軌跡が重なっている場合の比較の際に，zx 平面における動かし方に周期性の差を確認できる。このように作成したプログラムでは，指定した関節の動きを空間 (3 次元) 的に捉えることも，平面 (2 次元) 的に捉えることもできる。

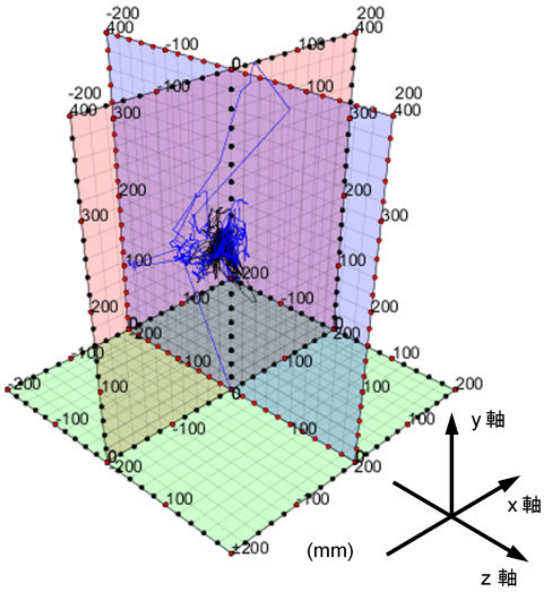


図 4.11　実行画面に表示される 3 次元グラフ

### 4.4.5　アニメーションとグラフ

図 4.6 の (2) にある「6. アニメーションと 3D グラフ」ボタンにより，アニメーション画像の動きの軌跡から 3D グラフや 2D グラフを描く。アニメーションの動きから軌跡を描くため，グラフの複雑な線が関節の変位であることを簡単に認識できる。図 4.13 にそ

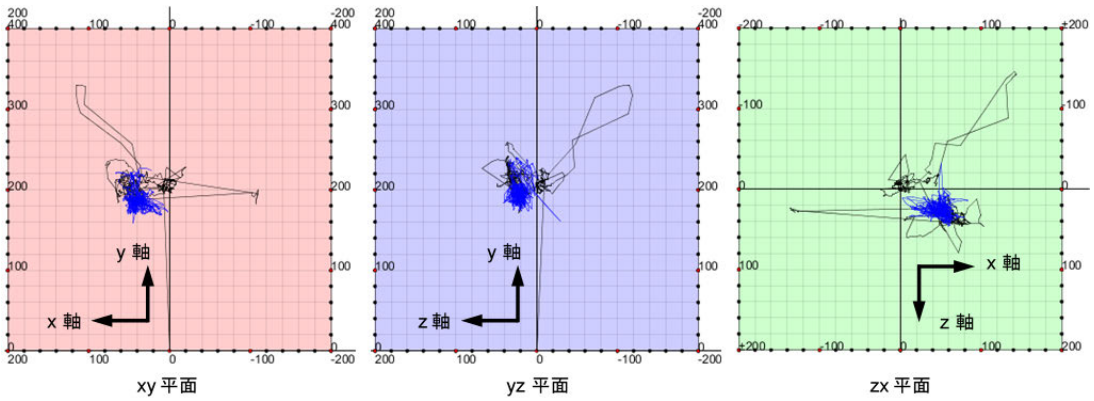


図 4.12　実行画面に表示される 2 次元グラフ

の実行画面を示す。図 4.13 では，2 つのデータの人差し指の指先における軌跡を 2D グラフ (zx 平面) 上に描いている。

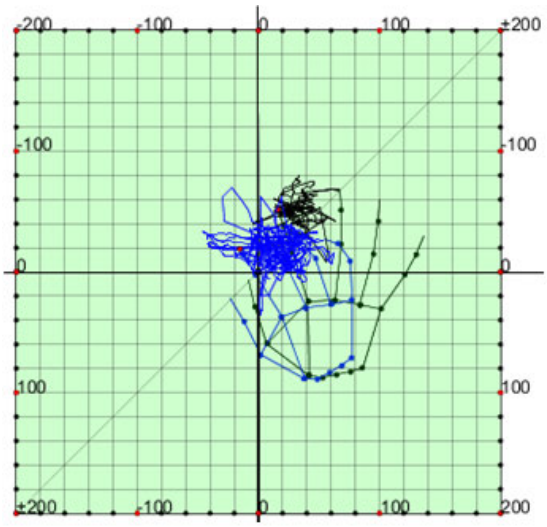

図 4.13　軌跡を描くアニメーション画像 (zx 平面)

## 4.5　評価試験

開発したシステムを用い，箸操作とキーボード操作時の手指動作を計測し，分析した。箸操作では，泉田らの研究 [150] の結果と比較した。キーボード操作では，キーボード操作技能による動作の違いを比較した。

### 4.5.1　箸操作の試験

試験は泉田らの研究 [150] から，箸の先端間隔 50mm 幅に開いた状態から 7mm 立方のスチンレンボードをつまむ動作を行った。箸には宮崎食器株式会社のチョップスティックジャポネ クリア／ブラック [151] を使用した。また先端間隔を 50mm 幅で開いた状態にするため，さしがねを用いた。なお泉田らの研究では，箸操作時の各関節にマーカをつけ 3 次元動作解析装置でモーションキャプチャを行っている。図 4.14 に先行研究にある箸操作の試験 [150] を，開発したシステムを用いて行った実験風景を示す。

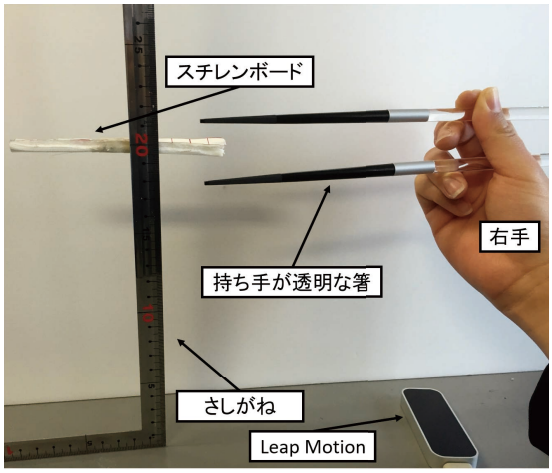


図 4.14　開発したシステムを用いた箸操作試験の様子

### 4.5.2　箸操作の試験結果

図 4.15 に，右手人差し指の第 2 関節の，箸使用におけるつまみ動作と関節角度の推移を示す。図 4.15 ではつまむ動作開始・終了時の関節角度がそれぞれ最小値・最大値となり，泉田らの研究と同じ特徴が確認できた。

### 4.5.3　キーボード操作の試験

Leap Motion から上に 150mm の位置にガラス製のキーボードを置き，タイピングを行った。ガラス製のキーボードには Bastron 社の Ultra Slim Touch Keyboard(キーレイアウト 84 キー英語配列，幅 340mm，奥行き 160mm，高さ 5.2mm)[152] を使用した。図 4.16 にガラス製のキーボードを示す。

試験は 2 種類行った。1 つ目の試験 (以下，試験 1 と表記) は，ミスタッチに関する試験

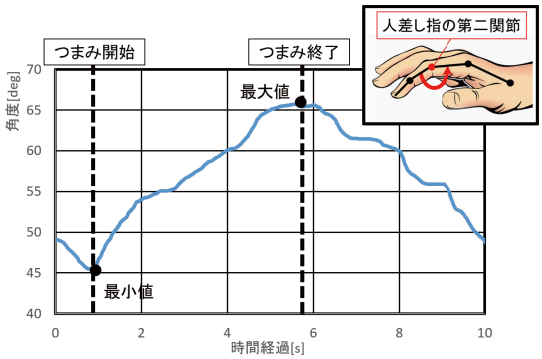


図 4.15　人差し指の第二関節における角度の推移

図 4.16　ガラス製のキーボード

である。2 つ目の試験 (以下，試験 2 と表記) は，タイピング試験を行った際の指の動きを確認するための試験である。

試験 1 を行う目的は，被験者がガラス製キーボードであることが原因でミスタッチが起こりやすくなるのかを調査することである。ガラス製キーボードは，通常のキーボードと異なり各キーがガラス板上に描かれているだけである。そのため，凹凸のある通常のキーボードキーに触れている被験者にとって，キー位置の把握と入力操作に影響を与えるのではないかと考えたからである。

試験 1 では，キーボード上の「Q」を押した状態から「P」を押した状態へ遷移した際の指先における座標値を 10 回計測し，「Q」と「P」の座標値を比較した。試験した指先は右手と左手の人差し指と中指の計 4 本である。「Q」と「P」を打鍵するのは本来の割り当て指からすると小指であるが，これによらず多くの人がキーボード操作でよく使用する人差し指と中指を評価の対象に選んだ。

試験 2 を行う目的は，通常のキーボード操作における指の動きを確認するためである。キーボード操作は，操作条件を同じにするため，Web 上のタイピング試験 (タイピング技能検定イータイピング・マスターの 6 級 [153]) を利用した。その試験では，被験者は PC 画面に表示される文字 (単語) をキーボードから入力していく。時間内に正確に入力でき

表 4.2　キー「Q」と「P」を押した各指の座標値

| 分類 | | | Q | | P | |
|---|---|---|---|---|---|---|
| | | | X 座標 | Z 座標 | X 座標 | Z 座標 |
| 右手 | 人差し指 | 平均 | -98.2mm | -47.0mm | 64.1mm | -46.1mm |
| | | 標準偏差 | 3.0mm | 5.8mm | 3.3mm | 2.9mm |
| | 中指 | 平均 | -97.8mm | -46.4mm | 67.0mm | -44.1mm |
| | | 標準偏差 | 4.1mm | 6.7mm | 2.7mm | 8.9mm |
| 左手 | 人差し指 | 平均 | -95.0mm | -41.8mm | 66.7mm | -43.0mm |
| | | 標準偏差 | 3.4mm | 8.0mm | 3.0mm | 5.5mm |
| | 中指 | 平均 | -95.0mm | -42.8mm | 67.9mm | -47.6mm |
| | | 標準偏差 | 3.2mm | 5.6mm | 4.6mm | 5.4mm |

たかどうかを PC 側で点数化し，基準点を超えれば 6 級合格となる。試験 1 と 2 は，大学生 20 名 (1 年生 8 名，4 年生 12 名) に対し実施した。

### 4.5.4　試験 1 の結果

「Q」の座標値は (-105，155，-60)〜(-90，155，-45)，「P」の座標値は (65，155，-60)〜(80，155，-45)(単位は mm) である。

表 4.2 に，キーボード上の「Q」と「P」を押した各指の X 座標 (左右方向) と Z 座標 (前後方向) の平均値と標準偏差を示す。図 4.17 に，キーボード上の「Q」と「P」と，各指の座標の平均値を示す。図 4.17 にある「十字」の中心が表 2 の平均値，「十字」の中心から上下左右にのびる直線の長さが表 2 の標準偏差である。図 4.17 より，各指の先端位置が「P」と「Q」の手前に集中していることがわかる。被験者は「P」と「Q」の各キーをほぼ確実にタッチしてはいたが，計測結果は表 4.2 と図 4.17 が示す結果となった。これは，指の先端として認識した位置が指の爪先よりも第一関節側へ 10mm ほどずれた位置を認識していると考えられる。これは，本システムで利用している Leap SDK(図 4.3) による指認識の際に表れる傾向である。

また，「P」の位置では各指の位置が左下に集中しているのは，「Q」から「P」へ指を遷移させる際に指を左から右へ遷移した結果，「Q」の位置よりも右にある「P」のキーをタッチする際の傾向と考えられる。これは，「Q」から「P」へ指を遷移させた時，「Q」の右側に指の座標値が集まる傾向にあったことが試験 1 の後，数名の学生の試行で確認された。

ガラス製キーボードを用いることで，ミスタッチが増えることが懸念されたが，通常キーボード操作と同じように扱えることがわかった。また，キーボード操作時の手指の動作が本システムで認識できることがわかった。さらに，指先座標が爪先の位置よりも第 1 関節側に 10mm ほどずれた位置を認識することから，指の先端よりも，指の各関節位置

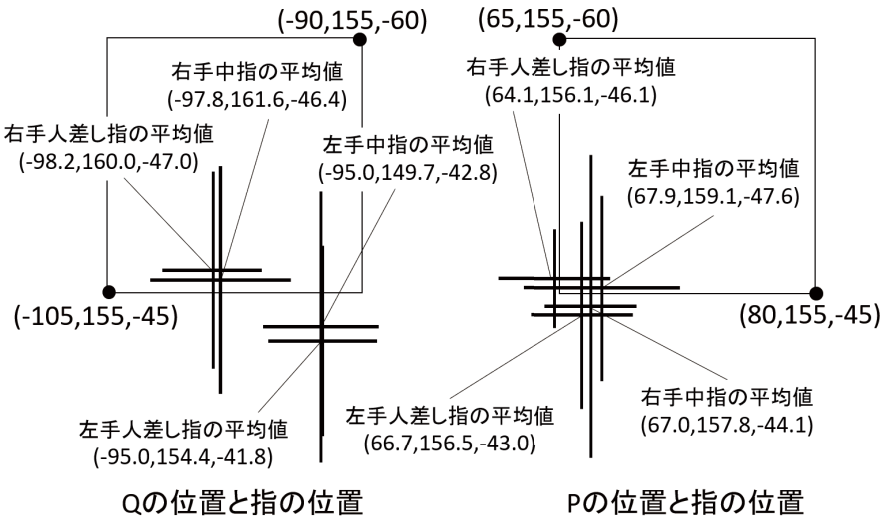


図 4.17　人差し指と中指の先端の位置

で計測する方がより正確な計測ができることがわかった。

### 4.5.5　試験 2 の結果

6 級試験の合格者は 17 名，不合者は 3 名であった。合格者と不合格者中から抽出した各 1 名の右手人差し指の第 2 関節の軌跡をグラフ化した。図 4.18 に軌跡をグラフ化したものを示す。

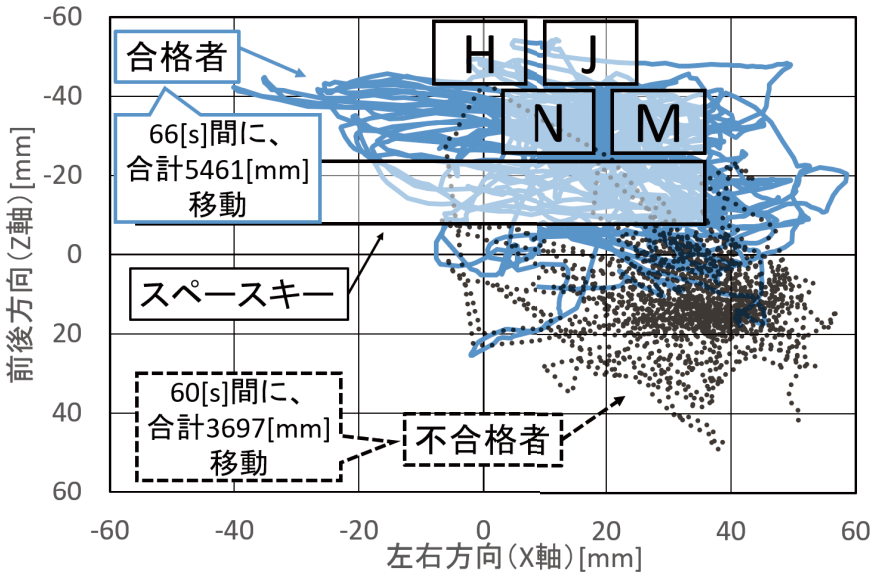


図 4.18　前後左右における右手人差し指の軌跡

表 4.3　合格者と不合格者における各指の移動距離 (mm)

| 分類 | 合格者<br>右手 | 合格者<br>左手 | 不合格者<br>右手 | 不合格者<br>左手 |
|---|---|---|---|---|
| 人差し指 | 5461mm | 4758mm | 3697mm | 3040mm |
| 中指 | 5174mm | 4481mm | 3687mm | 3063mm |
| 人差し指と中指との移動距離の差 | 287mm | 277mm | 10mm | 23mm |

図 4.18 に示すグラフ中の実線が合格者，点線が不合格者の軌跡となる。キーボードと右手人差し指との位置関係がわかるように，図 4.18 にキーボード上の「H」，「J」，「N」，「M」，「スペースキー」の位置を示す。表 4.3 に合格者と不合格者における，右手の人差し指第 2 関節と中指第 2 関節，左手の人差し指第 2 関節と中指第 2 関節の移動距離を示す。

図 4.18 より，合格者が不合格者より，右手人差し指による入力範囲が広かったり，試験に要した時間とその移動距離が長かったりすることがわかる。表 4.3 より，合格者が不合格者より，右手人差し指と中指との移動距離の差が 277mm 大きいことがわかる。これは，合格者は指を開いて打鍵する圧排動作をしているためである。この結果から，合格者は，手元を見ないで操作しているため，指の移動距離が多くなることがわかる。その反対に不合格者は，手元を見ながら作業をしたため，移動距離が少ないことがわかる。このように，本システムでは，ガラス製のキーボードを用いることにより，キーボード操作時の手指の関節の動きを分析できることがわかった。

## 4.6　本章のまとめ

本章では，手指の動作を計測し，計測したデータをもとに，アニメーションやグラフを表示する簡易手指動作システムについて整理した。簡易手指動作分析システムにより，箸操作とキーボード操作時の手指動作を計測し，動作の違いを定量的に分析できることを確認した。しかし，簡易手指動作分析システムを教育的に活用することができるかは明らかになっていない。そこで，次章では，簡易手指動作分析システムを教育現場で活用し，活用した結果をもとに簡易手指動作分析システムを改良していく。

# 第 5 章

# 簡易手指動作分析システムの活用と改良

前章では，簡易手指動作分析システムの概要を整理し，システムの教育的活用について評価を行っていない課題があるとまとめた。本章では，簡易手指動作分析システムを教育現場で活用し，システムの問題点を明らかにすることとした。システムの問題点をもとに，簡易手指動作分析システムを改良し，簡易手指動作分析システムで計測できる手指動作について述べていく。

## 5.1　評価試験の位置づけ

中学生が手指の技能測定できる簡易手指動作分析システム開発に向け，以下の条件を満たす必要があると考えた。

① 計測した手指動作の結果から，技能習得の方法やその過程をイメージできる。
② 計測した手指動作の結果から，手指の動作を比較し，動作の違いを捉えることができる。
③ 計測した手指動作の結果から，うまく作業している動作の特徴を捉えることができる。

計測した手指動作の結果は，4.4 節で示したアニメーション， 時間変位グラフ，3D グラフ，およびアニメーションの軌跡を意味する。

中学生がモーションキャプチャ・システムを理解できないために，動作の違いやポイントがわからない可能性がある。そこで中学生（初学者）が技能習得対象の手指作業によらず，計測した手指動作の結果から，技能習得のためにモーションキャプチャを活用することをイメージできるか確認するため，条件①を設定した。先行研究での身体動作における

モーションキャプチャ・システム [1][2] に比べ，開発したシステムは関節各部位の密集度が高い。そこで，初学者には開発したシステムで再現されたアニメーションによる動作の違いを視覚的に捉えにくい可能性があるため，条件②を設定した。身体動作に関するモーションキャプチャの教育的活用により，学習者は実際の動作に比べ削られた動きの情報（アニメーション）から，動きの特徴に気づくことができる [2]。そこで，開発したシステムにより実際の動作より削られた動きの情報（計測した手指動作の結果）から，初学者は動きの特徴に気づくことができるかを確認するため，条件③を設定した。条件①〜③を満たすために，授業によるシステムの評価試験を行った。これはモーションキャプチャによる技能習得の授業計画を立案する前提として，初学者が手指の技能測定できるシステムを開発できている必要があると考えたためである。システムの評価試験は中学校技術・家庭（技術分野）の授業で，東京都の私立 T 中学校第 3 学年を対象に 2 種類（以後，1 度目の授業による評価試験を授業 1，2 度目の授業による評価試験を授業 2）を行った。

## 5.2　授業 1 の概要

授業 1 では中学校技術・家庭（技術分野）の学習内容「D　情報の技術」において，中学校第 3 学年 4 クラス 53 人（男子 23 人，女子 30 人）を対象にし，「活用場面を考える学習」（50 分）を行った。授業 1 は計測した手指動作の結果から，技能習得の方法やその過程をイメージできるかを調べるために行った。

### 5.2.1　モーションキャプチャを活用した授業

授業ではノート型 PC（OS : windows8.1）を 1 台と Leap Motion を 1 台，コンピュータ室内で接続した。図 5.1 にコンピュータ室の様子を示す。表 5.1 に授業の流れを示す。

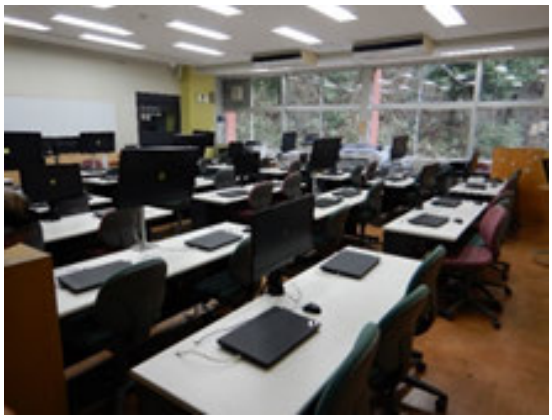

図 5.1　モニタと机の配置

表 5.1　授業の流れ (授業 1)

| 学習活動 | 時間配分 |
|---|---|
| ①　事前アンケートの実施 | 5 分 |
| ②　モーションキャプチャの実演 | 5 分 |
| ③　活用場面の検討と体験 | 20 分 |
| ④　発表 | 10 分 |
| ⑤　事前アンケート | 10 分 |

授業ではまず事前アンケートを実施し，これから学ぶ内容の見通しを持たせた（表 1 の①）。事前アンケートは記述式で，以下の 3 つの質問で実施した。

質問 (1)　器用とはなにか，教えてください。
質問 (2)　不器用とはなにか，教えてください。
質問 (3)　器用になるためにはどうすればいいか，教えてください。

次に手先が器用になる取り組みの一つとして，簡易手指動作分析システムにより教員の手指動作を計測したり，アニメーションの動作を示したり，時間変位グラフや 3D グラフで関節各部位の変位を示したりした（表 5.1 の②）。計測した手指動作の結果を，図 5.1 にある 2 人 1 台のモニタで確認させた。数人の班でテーマとモーションキャプチャの活用場面や指導方法を検討させ，班ごとに好きな手指動作のモーションキャプチャを行わせた（表 5.1 の③）。モーションキャプチャの様子や再現されたアニメーション，時間変位グラフや 3D グラフの様子を，手元のモニタでも確認させた。この学習活動の目的は中学生が，簡易手指動作分析システムによる技能測定の具体的なイメージや，動作の違いとコツを他人へ伝える工夫した表現の仕方を発想しやすくすることである。図 5.2 に中学生が手元のモニタで手指動作を確認する様子を示す。

班ごとに活用場面や指導方法を発表させ，事後アンケートを実施した（表 5.1 の④，⑤）。事後アンケートの質問項目は以下に示す。

質問 (4)　モーションキャプチャとは，どんなものかわかりますか。
質問 (5)　器用になるためにはどうすればいいか，教えてください。
質問 (6)　授業の感想を教えてください。

質問 (4) は 5 段階尺度（5,4 : 肯定的，2,1 : 否定的）による選択式で行った。質問 (5)，(6) は自由記述式で行った。

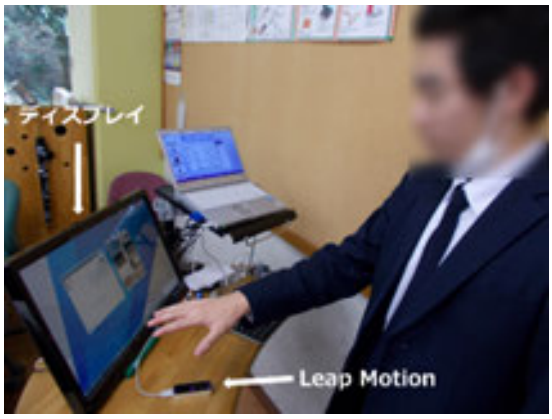


図 5.2　モニタで確認する中学生の様子

表 5.2　授業 1 の事前アンケート結果：質問 (1) の場合

| 頻出単語 | 件数 | 頻出単語 | 件数 |
|---|---|---|---|
| 細かい，丁寧，繊細，複雑 | 34 | 物事 | 7 |
| できる | 30 | きれい，滑らか | 7 |
| こなす，こなせる | 20 | 手先 | 5 |
| 作業，作る | 23 | 動き | 5 |
| 上手，上手い | 16 | 糸 | 5 |
| 要領，効率，センス，完璧，色々 | 12 | 針 | 5 |
| 人 | 9 | 得意 | 4 |
| すばやい，速い | 8 | 折り紙 | 2 |

### 5.2.2　授業 1 の結果

質問 (1)，(2)，(4) の結果，質問 (3)，(5) を比較した結果，および各班が考えた活用場面や指導方法について述べる。

表 5.2 に質問 (1) の頻出単語とその頻出数を示す。表 5.2 より，器用に対し，細かく，素早い作業や物事をこなしたり，滑らかな手先の動きであったり要領やセンスがよく，色々できたりするイメージがあるとわかる。これは，2.1 節で調べた器用の特徴に通じると思われる。また，針に糸を通す作業や折り紙を折るといった具体的な作業を回答する生徒もいるとわかった。

質問 (2) では質問 (1) の記述に「ない，不，反対，逆，苦手，下手，悪い」の語句を追加・変換した回答が 29 人（54.7 %）になった。それ以外の生徒の中で，不器用の例として，「5 教科の他にある，美術，技術，体育などができない」といった授業が不器用さを自認させる回答や，「ハサミを扱うのが下手」といった具体的な場面を挙げる生徒もいた。

表 5.3 に質問 (4) の結果を示す。表 5.3 より全体，男子，女子それぞれでの肯定的な回答の割合が 82.8 %以上であることがわかる。平均値では男子が女子より 0.29 大きく，標準偏差が変わらない。そのため質問 (4) の回答結果を t 検定したところ，男女の有意差がなかった (p=0.055)。

表 5.3　授業 1 の事後アンケート結果：質問 (4) の場合

| 回答結果 | 5 | 4 | 3 | 2 | 1 | 平均 | 標準偏差 |
|---|---|---|---|---|---|---|---|
| 全体 | 13 人 (25.0 %) | 32 人 (61.5 %) | 6 人 (11.5 %) | 1 人 (1.9 %) | 0 人 (0.0 %) | 4.10 | 0.66 |
| 男子 | 8 人 (34.8 %) | 13 人 (56.5 %) | 2 人 (8.7 %) | 0 人 (0.0 %) | 0 人 (0.0 %) | 4.26 | 0.62 |
| 女子 | 5 人 (17.2 %) | 19 人 (65.5 %) | 4 人 (13.8 %) | 1 人 (3.4 %) | 0 人 (0.0 %) | 3.97 | 0.68 |

表 5.4 に質問 (3)，(5) の頻出単語とその頻出数を比較した結果を示す。表 5.4 より，器用になるための取り組みとして，学習の前後で，練習や細かい作業を繰り返す学習方法から，モーションキャプチャにより比較したり，問題点を分析したりする学習方法へと意識に変化が生じたとわかる。そのため，技能習得の方法にモーションキャプチャ・システムが役立つと感じたことが考えられる。

表 5.4　質問 (3),(5) の頻出単語

| 頻出単語 | 質問 (3) | 質問 (5) |
|---|---|---|
| 練習 | 22 | 6 |
| 細かい作業，ちまちました作業 | 12 | 2 |
| 繰り返し | 5 | 1 |
| まね，比較，比べる | 1 | 19 |
| モーションキャプチャ | 0 | 16 |
| 分析，解析，研究 | 0 | 12 |
| 問題点，改善点，欠点 | 0 | 10 |
| 数値化 | 0 | 2 |

各班が考えた活用場面や指導方法は 24 グループ中，折り紙の折り方やタイピングなどの「手指動作」のみに着目した班の数が 10 グループであった。10 グループが考えた，簡易手指動作分析システムによる技能測定するテーマは「折り紙の折り方，タイピング，ピアノ，鉛筆の持ち方，手先が器用になる体操」であった。そのため，中学生が簡易手指動作分析システムによる技能測定を行う際，これらの動作が手指動作の違いや特徴を捉えやすいのではないかと考えた。また質問 (1), (2) の回答結果でも器用・不器用の説明とし

て，折り紙の折り方が挙げられていたが，針に糸を通す作業や，ハサミの操作を活用場面として挙げる班はなく，簡易手指動作分析システムで分析しにくい，または不向きであると判断されたと思われる。その中で，折り紙の折り方と鉛筆の持ち方を発想した各班の記述を表 5.5 に示す。

表 5.5　鉛筆の持ち方と折り紙の折り方に関する記述

| テーマ | 活用方法や指導方法 |
|---|---|
| 折り紙の折り方 | 器用な人は折り紙がきれいに折れる。<br>角と角がきれいに重なる |
| 鉛筆の持ち方 | ペン等が正しく持てている人ほど字は綺麗なのか？→しかし<br>モーションキャプチャでは LED を使用しており，色のついて<br>いるものだと誤って認識してしまうため，透明なペン等を使う。 |

表 5.5 にある折り紙の折り方に関する技能の測定方法では，折り紙の完成度から技能の分類方法を考えだしている。折り紙を折る手指動作を現状のシステムで計測しようとした時，Leap Motion　と手指動作の間に折り紙が重なり，手指が撮影できない。そのため，中学生が技能習得過程で製作品の完成度と合わせ，簡易手指動作分析システムによる手指の技能測定をしやすくすることで，多角的な視点で動きを捉えることができる取り組みにつながると考えた。

表 5.5 にある鉛筆の持ち方に関する技能の測定方法では，手指の誤認識を防ぐため，透明なペンを使うといった工夫を考えだしている。これは簡易手指動作分析システムの評価試験で行った，透明な箸やキーボードを活用し，手指動作を計測しやすくした取り組みであり，Leap Motion から手指の形を撮影できるので，現状のシステムでも計測しやすい。そこで，鉛筆を持つ手指動作と同じ手の形である，はんだづけ作業時の手指動作で，簡易手指動作分析システムの評価試験を分析した。

## 5.3　授業 2 の概要

授業 2 では中学校技術・家庭（技術分野）の学習内容「C　エネルギー変換の技術」において，中学校第 3 学年 4 クラス 59 人（男子 26 人，女子 33 人）を対象に，「はんだづけの学習」（50 分）を行った。授業 2 は，計測した手指動作の結果から，はんだづけ作業における手指の動作を比較し，動作の違いや特徴を捉えることができるかを調べるために行った。

### 5.3.1 先行研究での検証方法

授業 2 の検証方法には身体の動作をモーションキャプチャできるシステム（以後，簡易身体動作分析システム）を用いた，鉋がけの学習方法と調査方法 [71] を取り入れた。簡易身体動作分析システムを用いた検証方法 [71] は以下のようであった。

- 鉋がけを行う生徒と教員 1 名のモーションキャプチャを行い，モーションキャプチャの様子を大型モニタで確認した。
- キャプチャしたデータを基に，アニメーションで再現したり，各関節部位の 2 次元グラフや 3 次元グラフの変位を示したりし，生徒と教員の動きの違いを考えさせた。
- 授業後に 5 件法の選択式となる「モーションキャプチャとは，どんなものかわかりましたか。」「アニメーションでは，動きの違いがよくわかりましたか。」「グラフでは，動きの違いがよくわかりましたか。」と，記述式の「今日の授業を通しての感想を教えてください。」のアンケート調査を行った。

アンケートの調査結果から，先行研究では，「中学生は簡易身体動作分析システムで提示された，鉋がけ作業における身体動作の違いやコツを捉えることができること」を確かめることができた。そこで簡易手指動作分析システムによるはんだづけの学習では，簡易身体動作分析システムによる鉋がけの学習方法や調査方法を取り入れた。

### 5.3.2 生徒の実態

学校ではんだごてを使用した経験がある中学生の数を調査した。質問は授業を行う前に，アンケート調査とインタビュー調査を行った。アンケート調査では「はんだごてを使ったことがありますか」という質問に対して，回答を「1. はい」「2. いいえ」の 2 項目で行った。インタビュー調査ではアンケート調査後，全体講義形式で，はんだごての使用について確認した。表 5.6 にアンケート調査の回答結果をはんだごての使用経験別に示す。「はんだごてを使ったことがない」と回答した中学生（以後，未経験者）が 26 人（44.1 %）となり，使用したことのない中学生が半数弱いることが確認できた。インタビュー調査により，「はんだごてを使ったことがある」と回答した中学生 33 人 (55.9 %) は，はんだごてを作品へのウッドバーニングで使用した経験であることを確認した。また，部活動などの課外活動によって，はんだづけ経験のある中学生（以後，経験者）は 3 人 (5.1 %) の男子学生となるので，はんだごての使用経験のある中学生（以後，使用経験者）が 30 人（50.8 %）であり，対象とする中学生の 56 人（94.9 %）がはんだづけの初学者であるこ

表 5.6　アンケート調査の回答（n=59）

| 生徒の分類 | 人数 (割合) |
|---|---|
| はんだごて使用経験のない生徒：未経験者 | 26 人 (44.1 %) |
| はんだごて使用経験のある生徒：使用経験者 | 30 人 (50.8 %) |
| はんだづけ経験のある生徒：経験者 | 3 人 (5.1 %) |

とがわかった。

### 5.3.3　授業計画

授業の目標では，「振動モータ回路の作製を通して，はんだごてを安全に扱うようになる」とした。授業では「振動モータ回路の振動を利用した日常生活に役立つ作品の構想を課題とした。振動モータ回路はコイン電池ホルダ，小型ＤＣモータ，およびスライドスイッチを直列接続し，スイッチでモータの振動を切り替わる仕組みである。表 5.7 に振動モータ回路の部品表を示す。表 5.8 に授業計画を示す。表 5.8 の No.2 の授業で簡易手指動作分析システムを活用した。

表 5.7　振動モータ回路の部品表

| No | 部品名 | 必要数 |
|---|---|---|
| 1 | CH74 型コイン電池ホルダ (CH74 − 2032LF) | 1 個 |
| 2 | 小型 DC モータ (0716 − 1.3V) | 1 個 |
| 3 | スライドスイッチ (SS − 12D00 − G5) | 1 個 |
| 4 | リード線 | 1cm |
| 5 | プラスチック容器 | 1 個 |
| 6 | 風呂用ブラシ | 1 個 |

表 5.8　授業計画

| No | 学習内容 | システム使用 | 時数 |
|---|---|---|---|
| 1 | 材料や道具の特徴を学習する。 | | 1h |
| 2 | はんだごての使い方を学習する。 | ○ | 1h |
| 3 | はんだごてを使用し，作品製作を行う。 | | 2h |

### 5.3.4　モーションキャプチャを活用した授業

簡易手指動作分析システムを用い「はんだづけの技能学習」を行い，振動モータ回路の作製を始めた。授業ではノート型 PC(OS : windows8.1) を 1 台，Leap Motion を 1 台とプロジェクタ 1 台を技術室で接続し，大型スクリーンにモーションキャプチャの様子を投影した。表 5.9 に表 5.8 の No.2 の授業の流れ，図 5.3 に表 5.8 の No.2 の授業の様子を示す。

表 5.9　表 5.8 の No.2 の授業の流れ（授業 2）

| 学習活動 | 時間配分 |
|---|---|
| ①　はんだづけの説明 | 5 分 |
| ②　教員と生徒の動きを計測 | 5 分 |
| ③　動作の違いを比較 | 5 分 |
| ④　実際にはんだ付け作業を行う | 25 分 |
| ⑤　アンケートの実施・回収 | 10 分 |

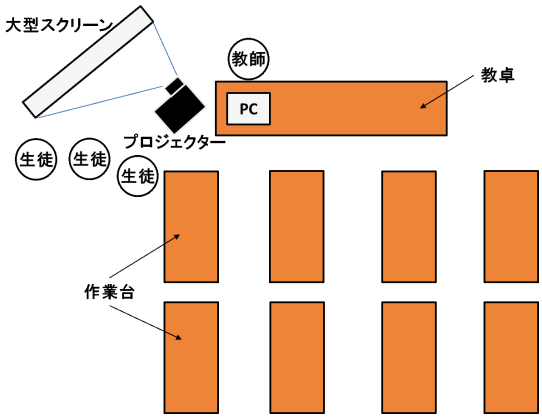


図 5.3　プロジェクタと机の配置

授業ではまず，ハンダやはんだごてに関する基本的な知識・使い方を復習し，はんだづけを説明した（表 5.9 の①）。次に，教員と中学生 1 名ずつが実際にはんだづけ作業を行い，作業時の手指動作を計測した（表 5.9 の②）。その動作は，廃材に置いた部品をセロハンテープで固定し，部品の端子にハンダとはんだごてをあてた時の手指動作となる。はんだづけ作業の手指動作を計測するにあたって，表 5.7 の No.1 にある部品（CH74 型コイン電池ホルダ (CH74 − 2032LF)），セロハンテープ，廃材，はんだごて（20W）[148]），およびハンダを用意した。簡易手指動作分析システムに読み込んだ教員と中学生の動作データを基に，手指動作をアニメーションで再現したり，各関節部位の 2 次元グラフや 3 次元

グラフで確認したりして，動作の違いを比較し，動作の特徴を中学生に示した（表 5.9 の③）。中学生に示した動作の特徴は，「右手で持ったはんだごての先端と部品の端子を見やすくするために，はんだごてをできるだけ斜めにしてあてること」である。図 5.4 に中学生に示した，部品の位置に対する，はんだづけ作業時のはんだごての角度を示す。動作の違いを比較した後，振動モータ回路の部品を廃材にセロハンテープでとめ，実際にはんだづけ作業を行わせた（表 5.9 の④）。授業終了 10 分前でアンケート調査を行い，回収した（表 5.9 の⑤）。

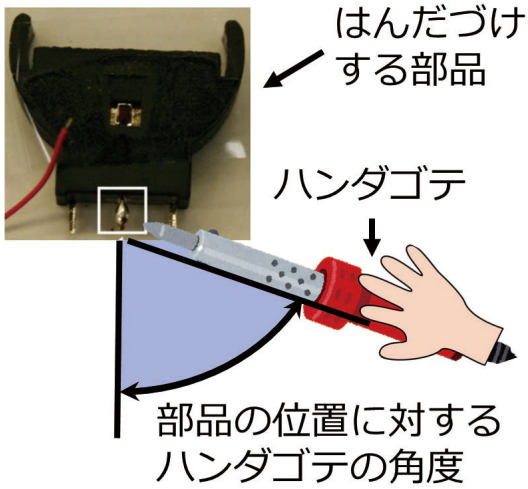


図 5.4　中学生に示した，はんだづけ作業時の手指の様子

### 5.3.5　授業 2 の結果

表 5.9 ので比較した教員と中学生における手指動作の違いと，表 5.8 にある No.2 の授業で行ったアンケートの調査の分析結果について述べる。

図 5.5 に，教員と中学生における右手人差し指の第 3 関節の違いを示す。図 5.5 より教員と中学生の手指動作を比較した際，部品のある 0 地点に対し，教員が中学生より広い視野ではんだづけ作業を行っていることがわかる。そのため，中学生は，教員と中学生における手指動作の違いを定量的に捉えることができたと考えられる。

アンケートは 6 つの質問の構成からなる。アンケートの質問項目は次の通りである。

質問 (1)　モーションキャプチャとは，どんなものかわかりましたか。
質問 (2)　アニメーションでは，動きの違いがわかりましたか。
質問 (3)　グラフでは，動きの違いがわかりましたか。
質問 (4)　アニメーションは，はんだごてを使うときの参考になりましたか。

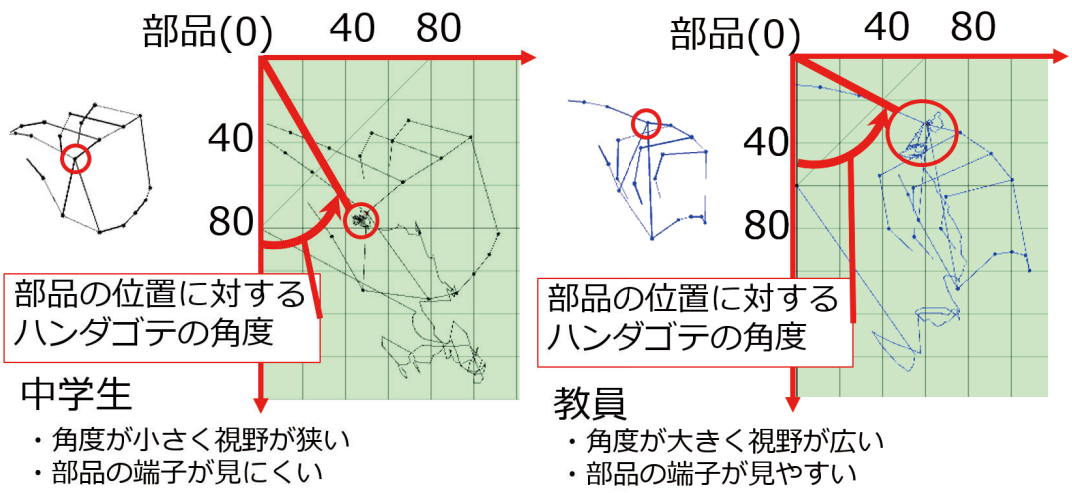


図 5.5　教員と中学生における人差し指の第 3 関節の違い

質問 (5)　グラフは，はんだごてを使うときの参考になりましたか。

質問 (6)　今日の授業を通しての感想を書いてください。

質問 (1)〜(5) は 5 段階尺度（5,4：肯定的，2,1：否定的）による選択式で行った。質問 (6) は自由記述式で行った。

表 5.10 に，はんだごて使用経験別に質問 (1)〜(5) の結果の平均と，() 内に標準偏差をまとめたものを示す。表 5.10 にある質問 (1)〜(4) より，平均値の高い生徒群が経験者，平均値の低い生徒群が未経験者であることがわかる。そのため，モーションキャプチャの教育的活用には，指導対象の動きを事前に経験しているかに影響があることが考えられる。表 5.10 の質問 (5) より，各生徒群の平均値と標準偏差には差が出ていないことがわかる。これは，はんだごての使用経験によらず，グラフで動きの違いやポイントを捉えても，実際の動作より削られた情報量が多くわかりにくかったと考えられる。そのため，グラフで動きの違いやポイントを捉えるには学習者が動きのイメージをもつとより有効になっていくと考えられる。

表 5.11，5.12 に，質問 (1)〜(5) の結果を未経験者と使用経験者ごとに示す。表 5.11, 5.12 より，質問 (1)〜(5) における肯定的な回答の割合が 61.5 %以上であることがわかる。そのため，初学者でもはんだごて使用経験によらず，簡易手指動作分析システムを用い，モーションキャプチャによる動作の違いを視覚的に捉え，モーションキャプチャで得た情報を活用することが十分可能であるとわかった。また質問 (1)〜(5) の結果から，使用経験者は未経験者より肯定的な回答の割合が高いことがわかった。使用経験者の方が動作のポイントを捉えることができたと考えられるため，質問 (1)〜(5) の回答結果を t 検定した結果，はんだごて使用経験による有意差がなかった（質問 (1) p=0.22，質問 (2)

表 5.10　授業 2 のアンケート結果：質問 (1)〜(5) の場合

| 分類 | | 質問 | | | | |
|---|---|---|---|---|---|---|
| | | (1) | (2) | (3) | (4) | (5) |
| 初学者 | 未経験者 | 3.73 (0.96) | 3.69 (0.74) | 3.73 (0.87) | 3.65 (0.98) | 3.58 (0.95) |
| | 使用経験者 | 4.03 (0.85) | 3.83 (1.02) | 3.90 (0.84) | 3.83 (0.91) | 3.70 (0.99) |
| 経験者 | | 4.67 (0.58) | 4.67 (0.58) | 4.67 (0.58) | 4.00 (1.00) | 3.67 (1.15) |

p=0.55，質問 (3) p=0.47，質問 (4) p=0.48，質問 (5)p=0.64）。

表 5.11　未経験者群における授業 2 のアンケート結果：質問 (1)〜(5) の場合

| 回答 | 質問 (1) | 質問 (2) | 質問 (3) | 質問 (4) | 質問 (5) |
|---|---|---|---|---|---|
| 5 | 6 人 (23.1 %) | 3 人 (11.5 %) | 4 人 (15.4 %) | 4 人 (15.4 %) | 4 人 (15.4 %) |
| 4 | 10 人 (38.5 %) | 13 人 (50.0 %) | 13 人 (50.0 %) | 13 人 (50.0 %) | 10 人 (38.5 %) |
| 3 | 7 人 (26.9 %) | 9 人 (34.6 %) | 8 人 (30.8 %) | 6 人 (23.1 %) | 10 人 (38.5 %) |
| 2 | 3 人 (11.5 %) | 1 人 (3.8 %) | 0 人 (0.0 %) | 2 人 (7.7 %) | 1 人 (3.8 %) |
| 1 | 0 人 (0.0 %) | 0 人 (0.0 %) | 1 人 (3.8 %) | 1 人 (3.8 %) | 1 人 (3.8 %) |

表 5.13 にはんだづけ操作の初学者における質問 (6) の結果をカテゴリー別に分類したものを示す。カテゴリーはモーションキャプチャ・システムに対する肯定的な回答と否定的な回答の 2 つに分類した。表 5.13 の「肯定的な回答」と「否定的な回答」について書いた生徒がそれぞれ 56 人中 13 人（使用経験者が 7 人，未経験者が 6 人），9 人（使用経験者が 4 人，未経験者が 5 人）となり，手指のモーションキャプチャが役立ったと感じる中学生がいた一方，複雑で難しいと感じる中学生もはんだごて使用経験によらずいた。そのため，現状のシステムでは初学者が手指のモーションキャプチャを視覚的に捉えにくかったと考えられる。また，経験者の 1 人は質問 (6) について「部活で使っているうちに我流になっていたのがわかった」と回答した。そのため，計測した手指動作の結果から，既習者が自身のはんだづけする動きを修正するのに役立ったと考えられる。

表 5.12　使用経験者群における授業 2 のアンケート結果：質問 (1)〜(5) の場合

| 回答 | 質問 (1) | 質問 (2) | 質問 (3) | 質問 (4) | 質問 (5) |
|---|---|---|---|---|---|
| 5 | 10 人 (33.3 %) | 8 人 (26.7 %) | 8 人 (26.7 %) | 7 人 (23.3 %) | 6 人 (20.0 %) |
| 4 | 12 人 (40.0 %) | 13 人 (43.3 %) | 12 人 (40.0 %) | 6 人 (20.0 %) | 13 人 (43.3 %) |
| 3 | 7 人 (23.3 %) | 6 人 (20.0 %) | 9 人 (30.0 %) | 6 人 (20.0 %) | 8 人 (26.7 %) |
| 2 | 1 人 (3.3 %) | 2 人 (6.7 %) | 1 人 (3.3 %) | 3 人 (10.0 %) | 2 人 (6.7 %) |
| 1 | 0 人 (0.0 %) | 1 人 (3.3 %) | 0 人 (0.0 %) | 0 人 (0.0 %) | 1 人 (3.3 %) |

表 5.13　授業 2 の「はんだづけの学習」を行った中学生の感想

| No | 肯定的な回答：13 人 | 否定的な回答：9 人 |
|---|---|---|
| 1 | プリントや絵だけでは，わからないものも実際に見たり，モーションキャプチャなどで生徒と先生の違いがよくわかってできやすかったです。 | くらべるときには同時にとって，どちらかをうすくしてかさねたらもっとわかると思った。 |
| 2 | アニメーションやグラフを通してはんだごてを使う際のコツがわかり，実際に自分がやった時にもちゃんとはんだをつけることができました。 | はんだごてを使うことができた。モーションキャプチャがあまり良くわからなかった。 |
| 3 | グラフやアニメーションが示していたように手の幅を広げて視界を広げるとやりやすいことがわかったので，次回にも生かしていきたい。 | モーションキャプチャよりビデオで手元を映した方が良かった。 |
| 4 | グラフやアニメーションを使うことによって，分かりやすかったです。 | アニメーションのモーションキャプチャが難しかったです。 |
| 5 | モーションキャプチャを通して，はんだごての動きが分かった。 | モーションキャプチャが難しかった。 |

## 5.4　簡易手指動作分析システムの改良

表 5.14 に簡易手指動作分析システムを用いた 2 種類の試験結果を示す。表 5.14 にある授業 1, 2 の各問題点を問題点 1，問題点 2 とし，各問題点の解決策について述べる。

問題点 1 を解決するためには，Leap Motion を上に設置し，逆向きにした状態で手指動

作を計測し，手指と Leap Motion との間に実物が入らないようにする必要がある。Leap SDK では，逆向きに設置した Leap Motion で手指動作を計測できる [149]。図 5.6 に，Leap Motion を上に設置し，逆向きにした状態で手指動作を計測できるよう改良した簡易手指動作分析システムの概念を示す。

問題点 2 を解決するためには，手指を重ねた時に色が重なって見えにくくならないよう，アニメーションの色を変える必要がある。また手指動作は身体動作に比べ動作範囲が狭く複雑な動きであるため，視覚的に捉えにくい。そのため，簡易手指動作分析システムの改良案として，動作分析のプログラムではアニメーションの色を変え，指定した指や関節各部位だけをアニメーションで表示できるよう改良すれば，初学者に対しての技能指導にも役立つと考えられる。

表 5.14　2 種類の試験について

| 種類 | 授業 1 | 授業 2 |
|---|---|---|
| 仮説 | 計測した手指動作の結果から，技能習得の方法やその過程をイメージできるかを調べること | 計測した手指動作の結果から，はんだづけ作業における手指の動作を比較し，動作の違いを捉えることができるかを調べる |
| 意義 | 条件を満たす手指作業を確認すること | 条件，を満たすのかを確認すること |
| 検証方法 | ・2 人並びの机 1 台に配置された，1 台のモニタでモーションキャプチャを確認<br>・班ごとにモーションキャプチャを実際に体験し，手元のモニターでモーションキャプチャの様子を確認<br>・班ごとに考えたモーションキャプチャの活用場面や指導方法を発表する<br>・授業後に行ったモーションキャプチャの理解度を確認する 5 件法の選択式となる質問と，班ごとのプリントにある記述を整理した。 | ・はんだづけを行う生徒と教員 1 名のモーションキャプチャを行い，モーションキャプチャの様子を大型モニタで確認<br>・キャプチャしたデータはアニメーションで再現したり，各関節部位の 2 次元グラフや 3 次元グラフの変位を示したりし，生徒と教員の動きの違いを比較<br>・簡易手指動作分析システムで得た情報を踏まえ，生徒ははんだづけを実際に行う。<br>・授業後に行った，5 件法の選択式となる質問と，感想を記述する質問からなるアンケートを整理した。 |
| 問題点 | Leap Motion と手指動作の間にものがあると手指動作を計測できない。<br>(以後，問題点 1) | 手指のモーションキャプチャが視覚的に捉えにくい。（以後，問題点 2） |

### 5.4.1　追加機能

図 5.7 にパソコンと接続した Leap Motion を示す。図 5.8 にプログラムの実行画面を示す。図 5.8 の (1) にある「下から」,「上から」のボタンにより，Leap Motion を手指の

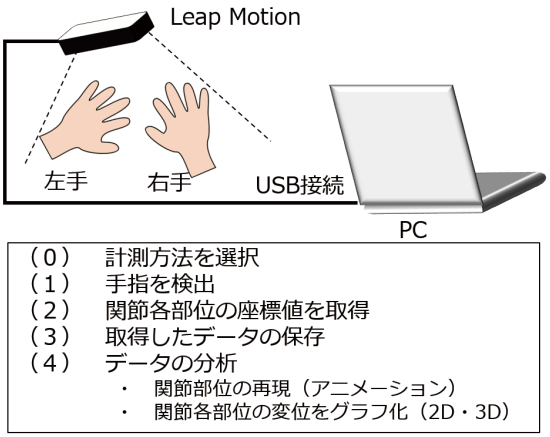


図 5.6　改良したシステムの概要

下に設置した状態 (「下から」のボタン)，手指の上に設置した状態 (「上から」のボタン) の計測環境いずれかを選び，モーションキャプチャを開始する。なお，問題点 2 に関しては，中学生が簡易手指動作分析システムを用い，自分の動きを分析するのは難しいと思われるため，改良を行わなかった。

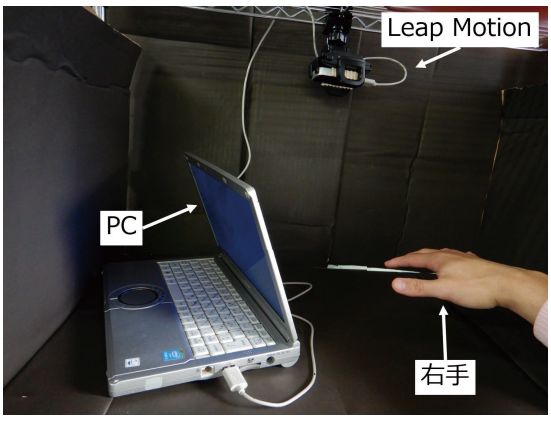


図 5.7　LeapMotion と接続した PC

### 5.4.2　評価試験

改良したシステムで手指の上に Leap Motion を設置した状態で計測しやすい手指動作の特徴を明らかにするため，試験 1 で生徒から活用場面として挙がった「折り紙を折る，タイピング，ピアノを弾く，針に糸を通す，ハサミで紙を切る」時の手指動作を計測できるか確認した。なお，生徒から活用場面として挙がった「手先が器用になる体操」の手指動作を空中での手指動作，「鉛筆で書く」手指動作をはんだごてを使う手の形と捉え，手

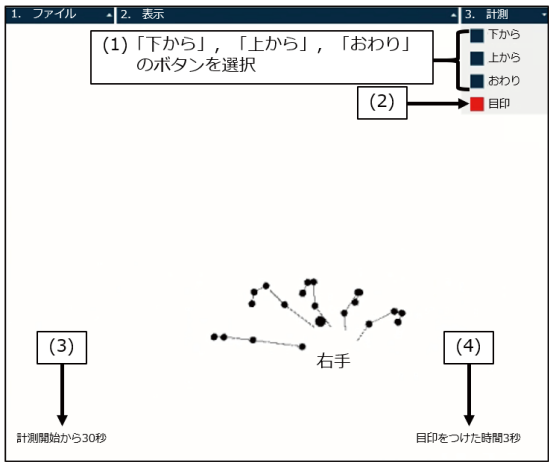


図 5.8　計測時に画面上で再現されるアニメーション

表 5.15　改良したシステムで確かめた手指の動き

| 手指動作の種類 | 計測できるか | 計測できた環境 |
|---|---|---|
| 折り紙を折る | ○ | 黒い板を置き，黒い紙を折る<br>手首側のある位置に Leap Motion を設置 |
| タイピング | ○ | 黒いキーボードに反射防止シートを貼る<br>手首側のある位置に Leap Motion を設置 |
| ピアノを弾く | ○ | 指先のある位置に Leap Motion を設置 |
| 針に糸を通す | ○ | 針や糸は黒色でなくても計測できる<br>手首側のある位置に Leap Motion を設置 |
| ハサミで紙を切る | ○ | ハサミや紙は黒色でなくても計測できる<br>ハサミを持つ手の平がわかる位置に Leap Motion を設置 |

指の下に Leap Motion を設置した計測環境で計測できることを確認した手指の形・動作であるため，確認しなかった。確認した結果を表 5.15 に示す。

表 5.15 より，「折り紙を折る，タイピング，ピアノを弾く，針に糸を通す，ハサミで紙を切る」手指動作は，計測時の環境を工夫することで誤認識せず計測できるとわかった。また，先行研究において，Leap Motion を用いたシステムでは，手話 [122] や空書き [123]，手遊び・影絵 [124]，何も持たずぞうきんを絞る [125]，といった手指動作を計測する先行研究が報告され，千吉良ら [72] が開発したタイピング練習ソフトでも黒いキーボードの上に，逆向きにした Leap Motion を指先のある位置に設置し，タイピングする手指動作を計測している。そのため，改良したシステムでは，折り紙を折ったり，タイピングしたり，ピアノを弾いたりする手指動作などといった，手を広げて行う動作（圧排動作 [115]）だけでなく，針に糸を通したりハサミで紙を切ったりする手指動作などといった，何らかの対象物をつかむ動作（把握動作 [115]）にもより対応できるようになったと判断した。し

かし，ハサミで紙を切る手指動作を計測するには，ハサミを持つ手の平がわかる位置に Leap Motion を設置しなければ，正常に計測し続けることができなかった。また，タイピングする手指動作を計測する際の Leap Motion を設置する場合でも，千吉良ら [72] の研究と比較し，指先か手首側で違っている。そのため，何らかの対象物をつかむ手指動作を計測するためには，手指の形が捉えやすくなるような位置に Leap Motion を設置する必要があると考えられる。

## 5.5　本章のまとめ

本章では，中学校技術・家庭（技術分野）において，簡易手指動作分析システムを活用し，活用した結果を基にシステムの教育的な活用方法を考察した。システムの教育的活用を考察するにあたり，2 つの授業による評価試験を行った。1 つ目の授業では，計測してみたい手指の動作を生徒に考えさせた。その結果，現状のシステムでは計測できない折り紙を折る手指動作の計測が挙がった。2 つ目の授業では，はんだづけの学習時に中学生と教員の手指動作の比較を行った。その結果，はんだづけ操作の初学者には動作の違いやコツを視覚的に捉えにくいことが分かった。そこで，Leap Motion を上に設置し逆向きにした状態で計測できる機能を，簡易手指動作分析システムに追加した。これにより，生徒が簡易手指動作分析システムの活用場面で挙げた，折り紙を折る手指動作などを計測できるようになった。また，Leap Motion を上に設置し逆向きにした状態で計測する時，操作時の手指の形がわかる位置に Leap Motion を設置する必要があると分かった。

次章では，簡易手指動作分析システムで計測できる道具操作時の手指動作の選定，すなわち，普通教育としての技術教育で使用される道具の中で，どの道具を選定すべきか検討していく。

# 第 6 章

# 手指の巧緻性によるドライバ操作時の手指動作の違い

前章では簡易手指動作分析システムを教育現場で活用し，Leap Motion を上に設置し，手指動作を計測できるように改良した。本章では，1.3 節の課題 3 を明らかにするため，改良したシステムを用い，手指の巧緻性が異なる大学生と中学生を比較し，ドライバ操作における手指動作の違いを分析していく。

## 6.1　実験の目的

本実験の目的は，手指の巧緻性が異なる大学生と中学生を比較し，ドライバ操作における手指動作の違いを分析することである。手指の巧緻性が異なる大学生群と中学生群を選定できているかを確かめるため，事前にアンケート調査を行った。

### 6.1.1　大学生と中学生の実態

大学生と中学生にはモノづくりに関するアンケート調査を行い，そのうち，利き手と，ドライバ使用時期の回答結果を述べる。手指の巧緻な動きが発達した段階の違いによる，ドライバ操作における手指動作の違いを分析しやすくするため，大学生は国立大学 S 大学の教員養成課程で，技術科教育か家庭科教育を専修する第 3,4 学年と大学院生 27[名](21〜28[歳]) を選んだ。中学生は先行研究 [5] から年齢的に手指が巧緻な動きに発達していない，私立 S 中学校の第 1 学年 25[名] を対象とした。アンケートは大学生を対象とした 5 つの質問，中学生を対象とした 6 つの質問となり，利き手と，ドライバ使用時期の質問項目は次の通りである。

【大学生を対象としたアンケート】

質問①　あなたは右利き，左利きどちらですか？
質問②　あなた自身は手先が器用だと思いますか？
質問③　質問②で「4,5」の回答を選んだ人は，いつからそう考えていますか？
質問④　本実験以外で，ドライバを使ったことがありますか？
質問⑤　質問④で「1」の回答を選んだ人は，いつ使ったことがありますか？

【中学生を対象としたアンケート】

質問 1　あなたは右利き，左利きどちらですか？
質問 2　あなた自身は手先が器用だと思いますか？
質問 3　質問 2 の回答を思うようになったのは，いつからだと思いますか？
質問 4　銅鏡製作でどの作業が一番簡単でしたか？
質問 5　銅鏡製作でどの作業が一番難しかったですか？
質問 6　ドライバをいつ使ったことがありますか？

アンケートは回答が選択式になっている。利き手の回答は質問①，1 が 2 択 (1：右利き，2：左利き) になる。器用意識の回答は質問②，2 が 5 択 (1:とても思う，2:思う，3:どちらでもない，4:思わない，5:全く思わない)，質問③が 5 択 (1:小学校入学前，2:小学生，3:中学生，4:高校生，5:大学生) と質問 3 が 5 択 (1:小学校入学前，2:小学 1,2 年，3:小学 3,4 年，4:小学 5,6 年，5:中学 1 年) になる。ドライバ使用時期の回答は質問④が 2 択 (1：はい，2：いいえ)，質問⑤，6 が 10 択となる。中学生の作業難易度を確認した回答は質問 4, 5 が 4 択 (1:絵のデザインを書く，2:カッターで紙を切る，3:木にネジを入れる，4:防水ペーパーで磨く) になる。

利き手の回答結果を述べる。質問①，1 の回答結果，大学生 27[名] の中で右利きが 26[名]，左利きが 1[名]，中学生 25[名] 全員が右利きであることがわかった。なお，左利きの大学生への口頭の質問により両利きとわかったため，実験では右手でドライバ操作を行わせた。

ドライバ使用時期の回答結果を述べる。表 6.1 に質問⑤，6 の回答結果を示す。表 6.1 より，対象の中学生群ではドライバを使用した割合が，中学校での生活以外で 50 %未満となることがわかった。これは対象の中学生群が先行研究の小学生群 [154] よりドライバ使用時期の割合が小さい。また対象の大学生群ではドライバを使用した割合が，小学校入学後の日常生活で 60 %以上，中学校・大学の生活で 70 %以上になるとわかった。そのため，先行研究 [5] と同じ，手指の巧緻な動きが発達した段階の異なる 2 つの対象群を選ぶことができたと考えられる。なお，対象の中学 1 年生は中学校の技術科授業で予めネジを締めた後，緩めた状態の穴に，ドライバでネジを締める作業を 1 度行っている。

表 6.1　アンケートの質問⑤，6 のの回答結果

| 被験者 | 回答時期 | 小学校入学前 | 小学生 | | | 中学生 | 高校生 | 大学生 |
|---|---|---|---|---|---|---|---|---|
| | | | 1,2 年 | 3,4 年 | 5,6 年 | | | |
| 中学生 N=25 | 日常生活 | 6 人 (24%) | 6 人 (24%) | 8 人 (32%) | 11 人 (44%) | 6 人 (24%) | | |
| | 学校生活 | 0 人 (0%) | 2 人 (8%) | 11 人 (44%) | 5 人 (20%) | 25 人 (100%) | | |
| 大学生 N=27 | 日常生活 | 3 人 (11%) | 18 人 (67%) | | | 21 人 (78%) | 18 人 (67%) | 23 人 (85%) |
| | 学校生活 | 0 人 (0%) | 5 人 (19%) | | | 24 人 (89%) | 8 人 (30%) | 21 人 (78%) |

## 6.2　実験の方法

実験では有川らの実験方法 [57] を参考に，角材に予め挿入した木ネジを締めて緩めるドライバ操作を行わせ，モーションキャプチャシステムを用い，ドライバ操作時の手指動作を計測した。表 6.2 に，実験に用いた材料と道具を示す。図 6.1 に，表 6.2 の材料と道具を示す。

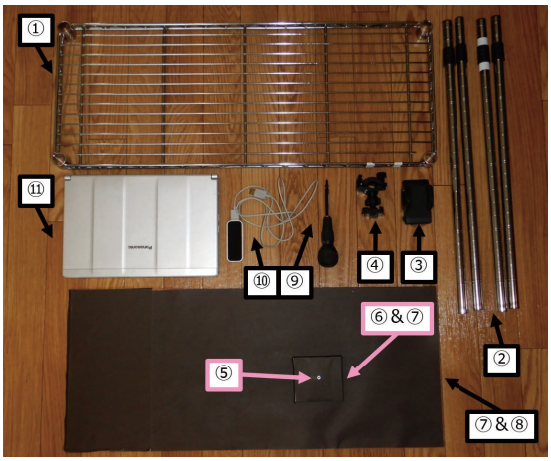


図 6.1　表 6.2 の材料と道具

表 6.2 の⑥にある角材には予め穴を開け，事前にネジを締めて緩めた状態で行った。これは大学生と中学生の筋力差 (押す力) により，手指動作の特徴を比較できないことがないようにするとともに，大学生と中学生が締める作業と緩める作業時に同じようにネジへ向けて力を加えながらドライバ操作ができる手指動作に最適化されているかを比較しや

表 6.2　材料・道具表

| 番号 | 材料・道具 | 数 |
|---|---|---|
| ① | 棚 型番：ST7530 サイズ：745[mm] × 295[mm] × 23[mm] | 1 |
| ② | ポール　型番：PHT-0060SL 直径：19[mm]　長さ：580[mm] | 4 |
| ③ | ガラスホルダ　型番：KS-213A | 1 |
| ④ | 専用取付 ステクランクタイプ 型番：KS-21TC | 1 |
| ⑤ | 木ネジ　直径：3[mm]　長さ：20[mm] | 1 |
| ⑥ | 角材　サイズ：70[mm] × 80[mm] × 40[mm] | 1 |
| ⑦ | 黒い紙 | 1 |
| ⑧ | ベニヤ板 サイズ：745[mm] × 295[mm] × 3 [mm] | 1 |
| ⑨ | ドライバ　型番：220W-3(ビット 3 本組) | 1 |
| ⑩ | Leap Motion | 1 |
| ⑪ | PC OS:windows10　型番：CF-SZ6 | 1 |

すくするためである。手指動作を計測しやすくするため，角材とベニヤ板 (表 6.2 の⑥，⑧) に黒い紙 (表 6.2 の⑦) を貼り，黒いネジとベッセル社の黒いドライバ [155](表 6.2 の⑤，⑨) を使用した。ドライバの先端は (+)1，ネジには (+)2 を使い，緩い状態でもドライバ操作をできるか確かめられるようにした。角材とベニヤ板は接着剤で固定し，大学生と中学生がドライバ操作時の手指動作に集中できるようにした。実験では角材にネジを 5[mm] ほど挿入した状態からネジを締めさせた。作業手順は以下の通りで行った。

1. 大学生と中学生は Leap Motion に右手だけを出し，画面上にアニメーションが表示される。
2. 大学生と中学生はドライバを持ち，角材に挿入されたネジのネジ頭にドライバの先端を固定する。
3. 筆者の合図とともに，ネジをドライバで締め始める。
4. 木ネジのネジ頭しか見えない状態まで締め終えたら，筆者が合図を出し，大学生と中学生が木ネジを緩め始める。
5. 大学生と中学生は木ネジを角材から取り出した時，筆者からの合図で，ドライバ操作をやめる。

筆者が合図した時，筆者が図 5.8 の (2) にある「目印」ボタンを押し，データ内の締める作業と緩める作業を分類した。また中学生にはドライバ操作後，ドライバ操作の感想をまとめさせた。図 6.2 に，中学生がドライバ操作している様子を示す。

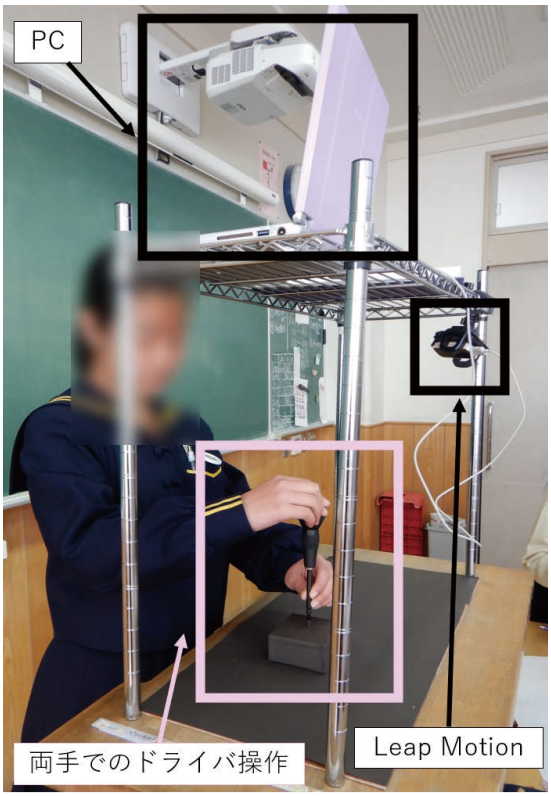


図 6.2　中学生のドライバ操作を計測する様子

## 6.3　実験の結果

計測データを基に，大学生と中学生がドライバ操作した手指動作を分析した。分析では独立した大学生群と中学生群を対象に，正規性を想定しないデータ間での客観的評価をするため，ノンパラメトリック検定を活用した。手指動作の分類，作業時間，関節各部位・手首の各角度，作業時間と関節各部位・手首との関係，関節各部位間の相関関係について順に述べる。

### 6.3.1　手指動作の分類

実験の結果，ドライバ操作時の手指動作を大別すると 3 種類になった。表 6.3 に，大学生と中学生を操作の型別に人数と，( ) 内に誤認識者を除いた数をまとめた。1 種類目は，ドライバの柄を上から右手で覆うようにし操作する持ち方 (以後，片手操作) となる。2 種類目は，ドライバの柄を上から右手で覆うようにし，左手でドライバの軸を支え操作する持ち方 (以後，両手操作) となる。3 種類目はドライバの柄を上から左手で覆うように操作

し，右手でもドライバの軸を回す持ち方となる。大学生 1 名の締める作業と中学生 3 名の締める・緩める作業では手の左右誤認識があった。

表 6.3　ドライバ操作の分類表

| 分類 | | 片手操作 | 両手操作 | その他 |
|---|---|---|---|---|
| 中学 | 締める作業 | 0 人 | 20 人 (17 人) | 5 人 |
| | 緩める作業 | 0 人 | 20 人 (17 人) | 5 人 |
| 大学 | 締める作業 | 19 人 | 8 人 (7 人) | 0 人 |
| | 緩める作業 | 17 人 | 10 人 | 0 人 |
| 全体 | 締める作業 | 19 人 | 28 人 (24 人) | 5 人 |
| | 緩める作業 | 17 人 | 30 人 (27 人) | 5 人 |

表 6.3 より，締める・緩める作業によらず，大学生群では片手操作の大学生が，中学生群では両手操作の中学生における人数が最も多いとわかった。以降の実験結果では，ドライバの柄を上から覆うようにした持ち方の，片手操作・両手操作の大学生 27[名] と両手操作の中学生 17[名] における計測データの結果について述べていく。計測データは，両手操作の大学生と片手操作の大学生を合わせた群，両手操作の大学生と両手操作の中学生を合わせた群，および，大学生と中学生を合わせた群で整理した。

### 6.3.2　作業時間

表 6.4 に，片手操作・両手操作の大学生群と両手操作の中学生群ごとに，ネジを締める・緩めるのにかかった時間の平均，( ) 内にその標準偏差を示す。

表 6.4　両群におけるドライバ操作の作業時間

| 分類 | 中学生群　両手 | | 大学生群　片手 | | 大学生群　両手 | |
|---|---|---|---|---|---|---|
| | 母数 [人] | 作業時間 [s] | 母数 [人] | 作業時間 [s] | 母数 [人] | 作業時間 [s] |
| 締める作業 | 17 | 20.05 (5.70) | 19 | 26.34 (5.45) | 7 | 22.61 (5.53) |
| 緩める作業 | 17 | 36.88 (11.69) | 17 | 26.92 (5.44) | 10 | 22.95 (4.49) |

表 6.4 より，平均の作業時間が短い順に並べると，締める作業では両手操作の中学生群，両手操作の大学生群，片手操作の大学生群，緩める作業では両手操作の大学生群，片手操作の大学生群，両手操作の中学生群となることがわかった。

片手操作・両手操作の大学生群と両手操作の中学生群の中から 2 群ごとに，作業時間の

分布差があるかをマン・ホイットニーの U 検定で確認した。その結果，締める作業では両手操作の中学生群と片手操作の大学生群の間，緩める作業では両手操作の中学生群と両手操作の大学生群の間，および両手操作の中学生群と片手操作の大学生群の間で $p < 0.01$ となり，作業時間の分布に有意差があることがわかった。それ以外の 2 群間では $p > 0.05$ となり，作業時間の分布に有意差がないとわかった。また両手操作の中学生群と片手操作・両手操作の大学生群の間に，作業時間の分布差があるかをマン・ホイットニーの U 検定で確認した。その結果，両手操作の中学生群と，片手操作・両手操作の大学生群との間では締める・緩める作業で $p < 0.01$ となり，作業時間の分布に有意差があることがわかった。

### 6.3.3　関節各部位・手首の各角度

関節各部位の呼称を，親指が指先から『第 1，2 関節』，その他の指が指先から『第 1，2，3 関節』，手首が縦方向に動く角度を手首『第 1 角度』，手首が横方向に動く角度を手首『第 2 角度』とした。これらの呼称を図 6.3 に示す。図 6.3 にある，手首の角度も分析の対象としたのは，リハビリテーションの領域において，日常生活の動作をできるか確認する指標に手首の動く角度範囲があり，立ってドライバを使う時の手首が動く角度範囲も調査されているためである [156]。そこで，ドライバ操作時に動いた手首の角度範囲も重要と考え，本研究に取り入れた。

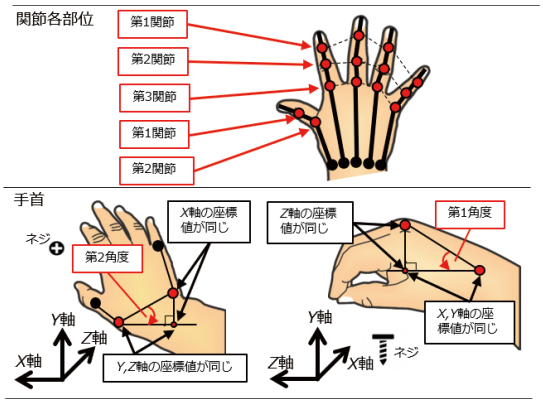


図 6.3　関節各部位と手首の各角度の呼称

表 6.5 に，片手操作・両手操作の大学生群と両手操作の中学生群ごとに，ネジを締める・緩める作業時の角度の平均，( ) 内にその標準偏差を示す。この表より，締める作業において中学生群が他の 2 群に比べ，親指の関節各部位と手首の第 1・2 角度における平均の角度が大きく，親指・中指第 1 関節以外の関節各部位における平均の角度が小さいことがわかった。また，緩める作業において中学生群が他の 2 群に比べ，親指の第 1 関節と手首

の第 1・2 角度における平均の角度が大きく，薬指・小指の関節各部位における平均の角度が小さいことがわかった。

表 6.5 より，締める・緩める作業において，片手操作の大学生群が他の 2 群に比べ，親指の関節各部位と手首第 1 角度における平均の角度が小さく，薬指第 3 関節と小指第 1・2 関節における平均の角度が大きいことがわかった。また両手操作の大学生群では締める・緩める作業において，手首の第 2 角度における平均の角度が小さく，人差し指の関節各部位と中指第 2 関節における平均の角度が大きいこともわかった。

各群の間に，角度の分布差があるかをマン・ホイットニーの U 検定で確認した。その結果，締める作業では，$p > 0.05$ となる組み合わせが，どの群の間の中指第 1 関節，片手操作の大学生群と両手操作の大学生群の間，および両手操作の中学生群・大学生群と片手操作の大学生群の間の人差し指第 1 関節となった。緩める作業では，$p > 0.05$ となる組み合わせが，両手操作の大学生群と両手操作の中学生群の間の親指第 2 関節，片手・両手操作の大学生群と両手操作の中学生群の間の中指第 1 関節，両手操作の大学生群と片手操作の大学生群の間の手首第 1 角度となった。それ以外の群の間では $p < 0.05$ となり，角度の分布に有意差があった。

### 6.3.4　作業時間と手首・関節各部位との関係

大学生と中学生の作業時間が短くなるにつれ，手首・関節各部位の角度・角加速度がどういう傾向になるかを確かめた。ケンドールの検定 (以後，Ken 検定) とスピアマンの検定 (以後，Spe 検定) を用い，傾向を確かめた。角度・角加速度には，第 1,3 四分位数・四分位範囲を用いた。

#### 作業時間と手首との関係

各群の作業別に，手首の角度・角加速度と作業時間との間で傾向を確かめた。表 6.6 に，各群の作業別に，Spe 検定した相関係数をまとめたものを示す。

表 6.6 より，中学生群と大学生群を合計した全群において，締める・緩める作業時ともに，手首第 1・2 角度の角度における第 1, 3 四分位数では Ken 検定と Spe 検定で $p < 0.05$ となり，相関があるとわかった。また締める作業時，手首第 2 角度の角加速度における第 1, 3 四分位数と四分位範囲では Ken 検定と Spe 検定で $p < 0.05$，緩める作業時，手首第 1・2 角度の角加速度における第 1, 3 四分位数と四分位範囲では Ken 検定と Spe 検定で $p < 0.05$ となり，相関があるとわかった。また，締める作業時における片手・両手操作の大学生群では手首第 1・2 角度，緩める作業時における両手操作の中学生群では手首第 2 角度と，作業時間との間で相関関係があるとわかった。なお，緩める作業時の大学生群と締める作業時の中学生群において，手首の角加速度では $p > 0.05$ となり相関がなかった。

緩める作業時における両手操作の中学生群・大学生群において，Ken 検定と Spe 検定で $p < 0.05$ となるのは，角度と作業時間との間で手首第 1 角度の第 1, 3 四分位数，角加速度と作業時間との間で手首第 1・2 角度の第 1,3 四分位数と四分位範囲であった。締める作業時における各大学生群において Ken 検定と Spe 検定で $p < 0.05$ となるのは，片手操作の大学生群で手首第 1 角度の第 1 四分位数，両手操作の大学生群で手首第 1 角度の第 3 四分位数と四分位範囲であった。

### 作業時間と関節各部位との関係

各群の作業別に，関節各部位の角度・角加速度と作業時間との間で傾向を確かめた。表 6.7, 6.8 に，各群の作業別に，Spe 検定した相関係数をまとめたものを示す。

表 6.7, 6.8 より，片手操作の大学生群において，Ken 検定と Spe 検定で $p < 0.05$ となったのは，締める作業が親指第 1 関節の第 3 四分位数と四分位範囲，および親指第 2 関節の第 3 四分位数，緩める作業が親指第 1・2 関節の第 1, 3 四分位数であり，相関があるとわかった。表 6.7, 6.8 を基に，作業時間ごとに親指第 1 関節の角度 (第 1, 3 四分位数) と近似直線を，締める・緩める作業ごとにプロットしたものを図 6.4 に示す。この図より，作業時間が短くなるほど，締める・緩める作業ともに角度が小さくなる傾向にあるとわかった。

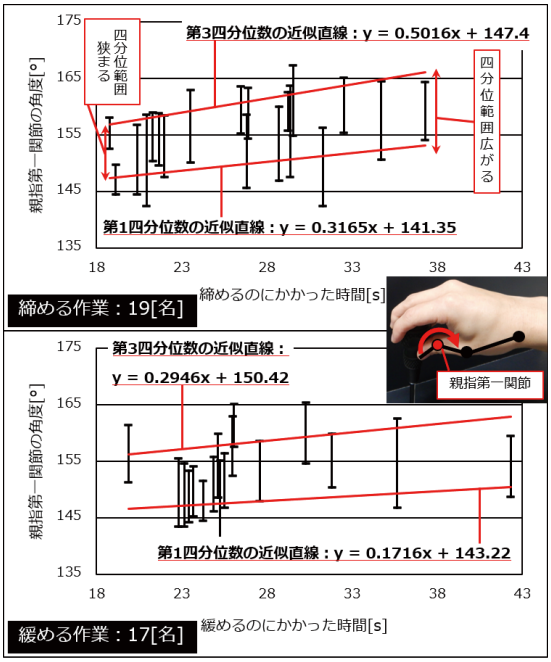


図 6.4　片手操作の大学生群における親指第 1 関節の角度

表 6.7, 6.8 より，両手操作の中学生群と片手操作・両手操作の大学生群の両群ともに，

中指第 3 関節の第 1 四分位数と四分位範囲では Ken 検定と Spe 検定で $p < 0.05$ となり，相関があるとわかった。表 6.7, 6.8 を基に，作業時間ごとに中指第 3 関節の角度 (第 1, 3 四分位数) と近似直線を，群ごとにプロットしたものを図 6.5 に示す。この図より，締めるのにかかった時間が短くなるほど，角度変化が大学生群では大きく，中学生群では小さくなる傾向にあるとわかった。

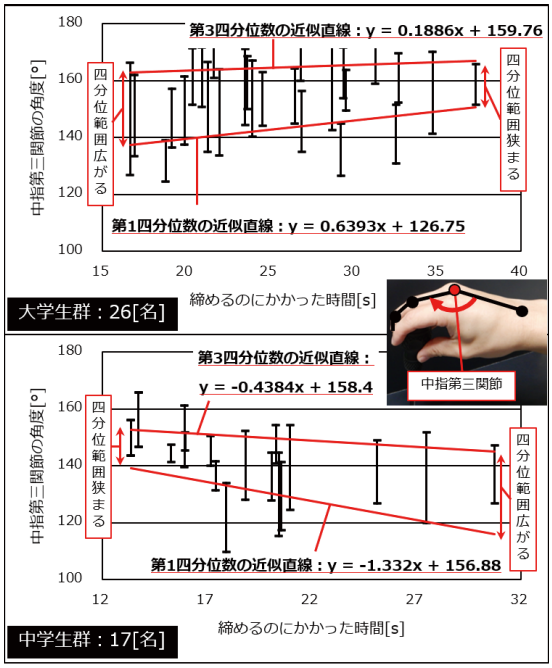


図 6.5　両群における締める作業時の中指第 3 関節の角度

表 6.7, 6.8 より，Ken 検定と Spe 検定で $p < 0.05$ の中指第 2 関節は，両手操作の中学生群が第 1, 3 四分位数，片手・両手操作の大学生群が第 1 四分位数と四分位範囲だとわかった。また両手操作の大学生群では，締める作業時における中指第 3 関節の第 1 四分位数と四分位範囲では Ken 検定と Spe 検定の p 値で $p < 0.05$ となり，図 6.5 の大学生群と同様に，締めるのにかかった時間が短くなるほど，角度変化が大きくなる傾向にあるとわかった。なお，両手操作の中学生群では，締める作業時の薬指第 2・3 関節と小指第 2 関節の第 1, 3 四分位数が Ken 検定と Spe 検定で $p < 0.05$ となり，Spe 検定の相関係数が-0.53 以下で負の相関があるとわかった。

また，片手・両手操作の大学生群において，締める作業では親指第 2 関節・中指第 3 関節・小指第 3 関節の角加速度と作業時間との間で相関関係があるとわかった。また緩める作業では，親指第 2 関節における角加速度の第 3 四分位数と四分位範囲で負の相関，小指第 3 関節における角加速度で相関があるとわかった。なお，締める作業における片手・両手操作の大学生群では，作業時間と角加速度との間で，Ken 検定と Spe 検定し $p < 0.05$

となるのは，親指第 2 関節，人差し指第 1・2・3 関節，中指第 2・3 関節，および薬指第 3 関節の第 1,3 四分位数と四分位範囲と，薬指・小指の第 2 関節の第 3 四分位数と四分位範囲であった。

表 6.7, 6.8 以外で，関節各部位の角加速度と作業時間との間で Ken 検定と Spe 検定で $p < 0.05$ となる群について述べる。緩める作業における両手操作の中学生群・大学生群では，中指第 1 関節の第 1 四分位数と小指第 1・2・3 関節の第 3 四分位数を含まない，関節各部位の第 1,3 四分位数と四分位範囲であった。片手操作の大学生群では，締める作業が親指第 2 関節の第 1 四分位数，人差し指第 3 関節の第 1, 3 四分位数と四分位範囲，中指第 3 関節の第 3 四分位数，中指第 2 関節の第 1 四分位数と四分位範囲，および小指第 3 関節の第 3 四分位数と四分位範囲，緩める作業が親指第 2 関節の第 3 四分位数と四分位範囲，および親指第 1 関節の四分位範囲であった。両手操作の大学生群では締める作業時における中指第 3 関節の第 1 四分位数と四分位範囲であった。両手操作の中学生群では，締める作業が薬指第 2 関節の第 1 四分位数と四分位範囲，緩める作業が親指の第 1・2 関節の第 3 四分位数と四分位範囲，および小指第三関節の四分位範囲であり，Spe 検定の相関係数がすべて |0.31| 以上 |0.44| 以下の範囲内となった。

#### 関節各部位間の相関関係

大学生群と中学生群ごとに，関節各部位の間で相関関係があるかを確かめた。学生ごとにデータ数が 200 以上あるため，相関を確かめるには Spe 検定を用いた。

表 6.9 に，両手操作の中学生群，片手・両手操作の大学生群ごとに，各指の関節角度間で，Spe 検定した相関係数の平均，( ) 内に標準偏差，および Spe 検定し $p \geqq 0.05$ だった人数をまとめ，締める・緩める作業ごとに示す。なお，表 6.9 にない中指，薬指，および小指の第 1・2 関節間における相関係数の平均が 0.27 未満となり，他の関節間に比べ弱い相関関係であった。

表 6.9 より，親指第 1・2 関節間では，片手・両手操作の大学生群が両手操作の中学生群より正の相関関係になり，人差し指第 1・2 関節間，および中指，薬指と小指の第 2・3 関節間では，両手操作の中学生群が片手・両手操作の大学生群より正の相関関係になるとわかった。図 6.6 に，片手・両手操作の大学生群と両手操作の中学生群の中で作業時間が最も短い大学生と中学生における，親指第 1・2 関節の角度をプロットしたものを示す。図 6.7 に，図 6.6 の大学生と中学生における，作業時間の経過による親指第 1・2 関節の角度変化をグラフ化したものを示す。

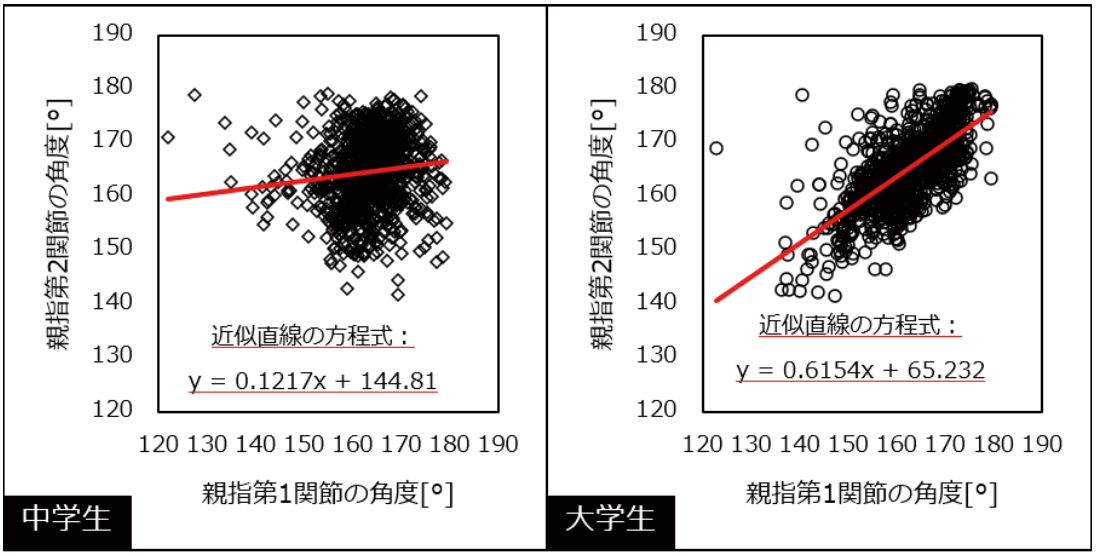


図 6.6　大学生と中学生における親指第 1・2 関節の角度分布

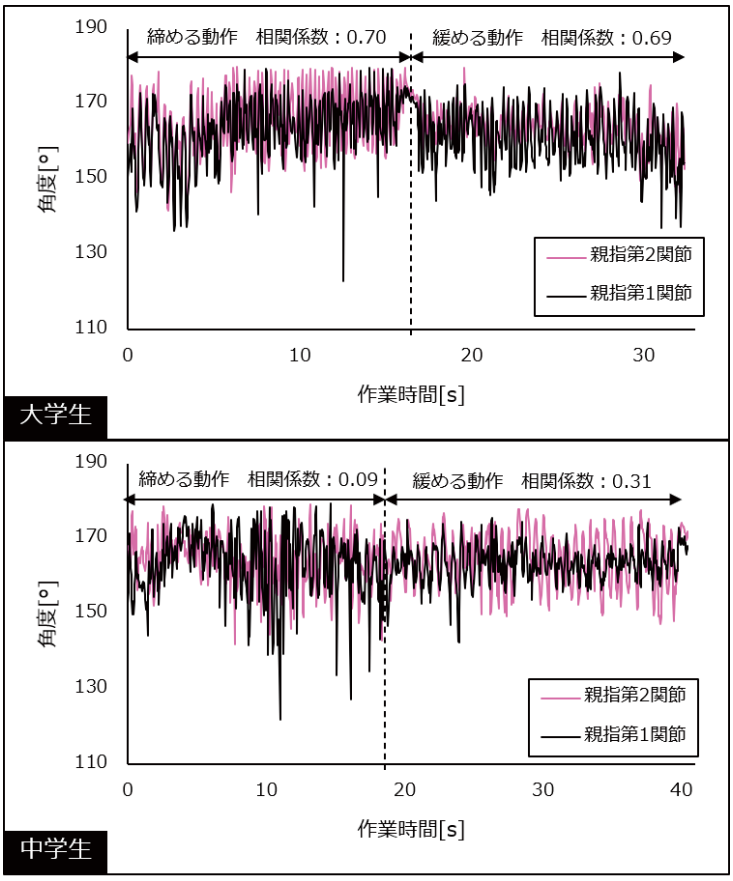


図 6.7　大学生と中学生における親指第 1・2 関節の角度変化

### 6.3.5 生徒の感想

表 6.10 に，生徒 25 人がドライバを操作した感想を，2 つに分類した結果を示す。表 6.10 より，「緩めるのが難しい」と記述した生徒が 10 人 (40%) となり，緩める際のポイントして，押す力を加えることに言及する生徒も複数人いることがわかった。また，「システムへの感想」を記述した生徒が 10 人 (40%) となり，生徒がモーションキャプチャシステムへの興味・関心をひくことができるとわかった。

## 6.4 考察

作業時間の比較，3 群ごとの特徴，教育的考察の順に述べていく。

### 6.4.1 作業時間の比較

締める作業時間の平均では両手操作の中学生群が他の大学生群に比べ小さく，緩める作業時間の平均では 2 つの大学生群が両手操作の中学生群に比べ小さくなった。また群ごとの作業時間の分布を分析した結果，緩める作業が締める作業より，群の間における作業時間の分布に有意差があるとわかった。そのため，本実験の条件における筋力の大きさは，緩める作業に対する影響が締める作業に比べ大きかったと考えられる。

### 6.4.2 3 群ごとの特徴

締める作業では両手操作の中学生，両手操作の大学生群，片手操作の大学生群の順に，平均の作業時間が大きくなった。これは角材の穴が既にネジを締めて緩めた状態であったため，両手操作の中学生群・大学生群が右手でドライバの柄を回し，左手でドライバの軸を支える役割分担で締めることができたためと考えられる。そのため，作業時間が短い中学生には中指第 2・3 関節，薬指第 2・3 関節および小指第 2 関節の角度が小さくなる傾向となり，両手操作の中学生群と両手操作の大学生群の間で，作業時間の分布差がなかったと考えられる。一方，片手操作の大学生群は右手でドライバの柄を回しながら，ドライバの軸を支える必要があったため，各指が役割分担した複雑な動きになると考えられる。そのため，作業時間が短い片手操作の大学生群には，親指第 1 関節の角度が小さいが，親指，人差し指，中指，および小指の関節各部位における角加速度が大きくなる傾向となり，両手操作の中学生群と片手操作の大学生群との間で，作業時間の分布差があったと考えられる。また中指第 1 関節は中学生群と大学生群の間に角度の分布差がないため，中指第 1 関節の角度を指導する重要性が他の関節各部位に比べ低いと考えられる。

緩める作業では両手操作の大学生群，片手操作の大学生群，両手操作の中学生群の順に，平均の作業時間が大きくなった。これは中学生群と大学生群との間で，親指の運動発達や筋力に違いがあったため，親指第 1・2 関節間の相関係数で大学生群が中学生群より大きかったと考えられる。一方，中学生群が大学生群に比べ，親指以外の指による動きをあてにしていたため，人差し指第 1・2 関節と，中指，薬指および小指の第 2・3 関節における相関係数が大学生群より大きかったと考えられる。中学生群において，大学生群に比べ，親指の関節間における相関係数が小さく，親指以外の関節間の相関係数が大きくなる傾向は，締める作業でも同様の結果であった。また，本実験を行った中学生 25 人中 10 人がドライバを操作した感想として，「締めるより緩めるのが難しかった」と記述し，緩める際にネジへ向けて押す力を加える重要性に気付いた生徒もいた。そのため，両手操作の中学生群は締める・緩める作業ともに，親指以外の指の関節各部位による角度の変化を規則的に行うが，この関節各部位の動きが大学生に比べ，ネジへ向けて押す力を加えにくい手指の形であったたため，緩める時間がかかったと言える。これは中学生が指の巧緻性 (指先だけで行う小物の操作が速く，正確であること)[115] によって，ドライバ操作していたと考えられる。

全被験者の群では，締める・緩める作業でともに，作業時間が短いと，手首第 2 角度の角加速度が大きいとわかった。これは手首と指の運動の速さ (指タッピングが敏速であること)[115] によって，ドライバ操作することが重要と考えられる。両手操作の中学生・大学生群では，緩める作業時間が短いと，中指と小指第 1 関節以外の関節各部位と手首第 1・2 角度の角加速度が大きくなるとわかった。また両手操作の大学生群は両手操作の中学生群に比べ，親指第 1・2 関節間の相関係数が大きい。そのため，両手操作の大学生は両手操作の中学生に比べ，親指の動きを大きく規則的にするとともに，各指の関節各部位と手首の角度変化を大きくできるよう，ドライバ操作していたと考えられる。これは緩める作業において，両手操作の中学生が指の巧緻性 [115] によってドライバ操作する一方，両手操作の大学生が手と指の巧緻性 (手と指による運動がすばやく，正確であること)[115] と手首と指の運動の速さ [115] によってドライバ操作していたと考えられる。

### 6.4.3　教育的考察

手塚は初心者に対し，旋盤技能やドリル技能の学習実験を通した技能習得の過程を記録した [157]。その結果，学習初期段階は作業時間を短くすることだけに意識させ，練習を重ねた後，製作品の正確さ (精度) も意識させる方法が最終的に，作業時間が短くなり製作品の精度も上がる有効な手段であることを明らかにした。そのため，中学生を対象とした，ドライバの技能指導においてもまず作業時間が短くなるよう手指の動かし方を指導する方法が有効であると考えられる。そこで，操作者の巧緻性に繋がるドライバ操作の要点とし

て，両手操作・片手操作に共通して「指先だけでなく手首も使い，手全体でドライバを回すこと」，「手首，親指と小指で回しやすいドライバの持ち方をすること」を伝えるのが有効ではないかと考えた。

有川らが中学生を対象に行った，アンケート調査とドライバ操作の技能指導に関する研究 [37]，[57] をもとに，中学生を対象にしたドライバ操作の技能指導初期段階に伝える要点を考察していく。中学生 384 名を対象にしたアンケート調査とドライバ操作の技能指導 [37] では，Ⓐ「ドライバーでボルト（ネジ）を締めたり緩めたりするには，ドライバーに手でどのような力を加えたらよいか」を，ドライバーの略図と加える力を矢印で書き込ませた結果，ネジを締める・緩める際のどちらでも「回す力」のみを書いた生徒が 280 名 (72.9%) となったこと，Ⓑドライバとネジのサイズが一致するドライバ操作の技能指導では，締める際に押す力，緩める際に引く力を加える意識に傾きやすいとわかる。Ⓐ，Ⓑに関しては，本実験を行った中学生が大学生に比べ，締めるのに時間がかからなかった一方，緩めるのに時間がかかった結果となり，その原因がネジに向けて押す力を加えにくい手指の形にあるのではないかという考察に通じる。

次に，ドライバの技能指導に関する研究 [57] には，①巧緻性が低い中学生の一人は両手指先で柄の先端をつまみ連続的に回していたこと，②片手・両手操作に共通して操作改善が期待できる手指動作への示唆として，柄を利き手で上から被せるように握ること，利き手はネジの硬さに応じて適宜持ちかえること，③操作難易度の観点から，まず両手操作時における力の加え方を掴ませることを提案している。①に関しては本実験の中学生が指の巧緻性 [115] によってドライバ操作していたという考察に通じる。②に関しては，握り方が分析対象の共通する特徴であり，本実験でも大学生群の中から，締める作業から緩める作業になり片手操作を両手操作に変える人も出た。③に関しては，本実験の中学生も全員，両手操作でドライバ操作を行っていた。また大学生群では両手操作が片手操作に比べ，締める・緩める作業の平均時間がともに短いが，2 つの大学生群の間で作業時間の分布差がない。そのため，本実験の条件下において，大学生群では両手操作が片手操作に比べ，操作難易度に大きな差があるわけではないと考えると，中学生を対象にした，ドライバ技能指導の初期段階には「ネジに向けて力を加える」，「柄を利き手で上から被せるように握る」，「両手操作で行う」，「親指の動きを大きく規則的にする」，および，「各指の関節各部位と手首における角度変化を大きく動かす」ことを伝えるのが有効ではないかと考えた。

## 6.5　本章のまとめ

簡易手指動作分析システムを用い，大学生と中学生がドライバ操作する手指動作を計測することができた。計測した結果から，ドライバ操作の作業時間が短くなるには年齢・操作によらず，手首の角加速度が大きくなる持ち方をすることが重要であるとわかった。両

手操作の中学生を対象に，作業時間が短くなることを目標とした，ドライバの技能指導を行う初期段階では，親指でドライバの柄を回すことを意識させる必要があるとわかった。しかし，中学生がドライバ操作時にネジに向けて押す力を意識しにくく，手首の回転が速くても，ドライバからネジに向けた力が適切に加わっていなければ，ネジの締め過ぎや締め不足になる恐れがある。すなわち，中学生を対象にドライバ操作の技能指導を行った時，ネジに向けて押す力に違いが出やすいのではないかと考えた。そこで，次章では，ネジに向けて押す力を計測できる装置を開発することにした。

表 6.5　各群における作業時の各関節の角度と手首の各角度

| 分類 | | 中学生群　両手 | | 大学生群　片手 | | 大学生群　両手 | |
|---|---|---|---|---|---|---|---|
| 作業別 | | 締める | 緩める | 締める | 緩める | 締める | 緩める |
| 人数 [人] | | 17 | 17 | 19 | 17 | 7 | 10 |
| 親指 | 第 2 [°] | 164.67 (7.18) | 162.21 (7.51) | 160.66 (7.65) | 159.91 (7.06) | 163.55 (7.97) | 162.57 (6.77) |
| | 第 1 [°] | 161.84 (7.89) | 158.16 (8.03) | 155.62 (8.92) | 153.38 (8.84) | 159.06 (13.41) | 157.48 (8.11) |
| 人差し指 | 第 3 [°] | 147.70 (15.36) | 151.97 (13.20) | 154.12 (16.34) | 152.11 (17.60) | 156.06 (13.41) | 152.60 (13.90) |
| | 第 2 [°] | 139.28 (19.10) | 145.18 (14.84) | 140.96 (20.16) | 138.53 (22.25) | 142.63 (19.13) | 146.05 (17.06) |
| | 第 1 [°] | 152.58 (16.90) | 156.62 (13.63) | 153.11 (16.84) | 153.23 (16.11) | 153.95 (16.14) | 157.17 (14.08) |
| 中指 | 第 3 [°] | 143.01 (18.24) | 150.05 (15.41) | 153.90 (17.64) | 154.68 (18.02) | 153.23 (16.41) | 157.28 (14.27) |
| | 第 2 [°] | 127.73 (25.75) | 138.79 (23.22) | 136.10 (25.99) | 137.05 (27.70) | 137.84 (27.30) | 147.17 (22.32) |
| | 第 1 [°] | 153.03 (20.30) | 155.62 (18.81) | 153.05 (19.47) | 154.18 (19.50) | 152.41 (21.17) | 159.17 (17.41) |
| 薬指 | 第 3 [°] | 137.96 (20.10) | 142.99 (19.48) | 150.41 (18.69) | 154.99 (18.20) | 146.33 (19.39) | 154.07 (16.98) |
| | 第 2 [°] | 122.35 (29.31) | 127.95 (30.23) | 135.35 (29.06) | 140.55 (29.13) | 127.12 (31.90) | 142.92 (27.31) |
| | 第 1 [°] | 147.34 (21.90) | 150.83 (20.15) | 153.23 (19.50) | 155.14 (19.81) | 151.37 (21.52) | 158.42 (19.20) |
| 小指 | 第 3 [°] | 136.81 (18.59) | 140.65 (17.65) | 149.52 (17.95) | 153.81 (17.71) | 146.54 (18.55) | 152.73 (16.38) |
| | 第 2 [°] | 122.33 (23.78) | 127.34 (25.69) | 135.73 (25.58) | 140.33 (26.81) | 129.59 (27.28) | 142.30 (24.57) |
| | 第 1 [°] | 146.97 (22.94) | 148.23 (21.52) | 155.47 (19.59) | 157.09 (20.22) | 152.98 (21.41) | 159.92 (20.08) |
| 手首 | 第 2 [°] | 33.65 (11.98) | 33.46 (12.15) | 18.74 (11.49) | 19.45 (12.45) | 17.94 (13.64) | 18.20 (10.77) |
| | 第 1 [°] | 45.04 (11.35) | 39.07 (12.87) | 19.77 (8.46) | 25.15 (12.26) | 27.19 (16.01) | 27.04 (16.10) |
| データ数 | | 8,123 | 13,926 | 12266 | 12039 | 4505 | 6025 |

表 6.6　各群における手首の角度・角加速度と作業時間との相関係数

| 作業 | | | 締める作業 | | | 緩める作業 | | |
|---|---|---|---|---|---|---|---|---|
| 群 | | | 全群 43 人 | | 全大学生群 26 人 | 全群 44 人 | | 中学生群 両手　17 人 |
| 種類 | | | 角度 | 角加速度 | 角加速度 | 角度 | 角加速度 | 角加速度 |
| 四分位数 | 第 1 | 手首第 2 角度 | -0.35* | 0.39** | 0.59** | 0.37** | 0.49** | 0.49* |
| | | 手首第 1 角度 | -0.35* | | 0.44* | 0.33* | 0.52** | |
| | 第 3 | 手首第 2 角度 | -0.30* | -0.36* | -0.48** | 0.30* | - 0.53** | -0.56* |
| | | 手首第 1 角度 | -0.29* | | -0.43* | 0.37** | -0.54** | -0.43* |
| 四分位範囲 | | 手首第 2 角度 | | -0.39** | -0.57** | | -0.52** | 0.57* |
| | | 手首第 1 角度 | | | -0.45* | | -0.53** | |

なお，* : Ken 検定と Spe 検定ともに $p<0.05$　** : Ken 検定と Spe 検定ともに $p<0.01$ となる

表 6.7　各群における関節各部位の角度・角加速度と作業時間との相関係数：締める作業

| 群 | | 大学生群 片手　19 人 | | 中学生群 両手　17 人 | | 全大学生群 26 人 | | 全大学生群 26 人 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 種類 | | 角度 | | | | | | 角加速度 | | |
| 指 | | 親指 | | 中指 | | 中指 | | 親指 | 中指 | 小指 |
| 関節 | | 第 2 | 第 1 | 第 3 | 第 2 | 第 3 | 第 2 | 第 2 | 第 3 | 第 3 |
| 四分位数 | 第 1 | | | -0.74** | -0.70** | 0.37* | 0.51** | 0.55** | 0.54** | 0.47* |
| | 第 3 | 0.60** | 0.72** | | -0.60** | | | -0.41* | -0.53** | -0.54** |
| 四分位範囲 | | | 0.39* | 0.63 | | -0.38* | -0.38* | -0.46* | -0.51** | -0.52** |

なお，* : Ken 検定と Spe 検定ともに $p<0.05$

** : Ken 検定と Spe 検定ともに $p<0.01$ となる

表 6.8　各群における関節各部位の角度・角加速度と作業時間との相関係数：緩める作業

| 群 | | 大学生群<br>片手　19 人 | | 全大学生群<br>26 人 | | |
|---|---|---|---|---|---|---|
| 種類 | | 角度 | | 角加速度 | | |
| 指 | | 親指 | | 親指 | 中指 | 小指 |
| 関節 | | 第 2 | 第 1 | 第 2 | 第 3 | 第 3 |
| 四分位数 | 第 1 | 0.59* | 0.52* | | 0.34* | 0.37* |
| | 第 3 | 0.59** | 0.58* | -0.39* | | -0.38* |
| 四分位範囲 | | | | -0.35* | | -0.37* |

なお，*：Ken 検定と Spe 検定ともに $p<0.05$

**：Ken 検定と Spe 検定ともに $p<0.01$ となる

表 6.9　各群における作業別の関節角度間の相関係数

| 群 | | 中学生群　両手 | | 大学生群　片手 | | 大学生群　両手 | |
|---|---|---|---|---|---|---|---|
| 作業 | | 締める | 緩める | 締める | 緩める | 締める | 緩める |
| 人数 | | 17 人 | 17 人 | 19 人 | 17 人 | 7 人 | 10 人 |
| 親指<br>第 1・2<br>関節間 | 相関<br>係数 | 0.39<br>(0.17) | 0.36<br>(0.28) | 0.73<br>(0.17) | 0.63<br>(0.22) | 0.64<br>(0.20) | 0.67<br>(0.15) |
| | $p \geqq 0.05$<br>の人数 | 0 人 | 4 人 | 0 人 | 0 人 | 0 人 | 0 人 |
| 人差し指<br>第 2・3<br>関節間 | 相関<br>係数 | 0.66<br>(0.15) | 0.51<br>(0.28) | 0.54<br>(0.27) | 0.58<br>(0.13) | 0.64<br>(0.20) | 0.57<br>(0.10) |
| | $p \geqq 0.05$<br>の人数 | 0 人 | 0 人 | 0 人 | 1 人 | 0 人 | 0 人 |
| 人差し指<br>第 1・2<br>関節間 | 相関<br>係数 | 0.55<br>(0.15) | 0.52<br>(0.14) | 0.26<br>(0.25) | 0.22<br>(0.24) | 0.26<br>(0.09) | 0.33<br>(0.17) |
| | $p \geqq 0.05$<br>の人数 | 0 人 | 0 人 | 4 人 | 4 人 | 0 人 | 1 人 |
| 中指<br>第 2・3<br>関節間 | 相関<br>係数 | 0.64<br>(0.12) | 0.57<br>(0.15) | 0.49<br>(0.23) | 0.51<br>(0.17) | 0.57<br>(0.17) | 0.42<br>(0.16) |
| | $p \geqq 0.05$<br>の人数 | 0 人 | 0 人 | 0 人 | 0 人 | 1 人 | 0 人 |
| 薬指<br>第 2・3<br>関節間 | 相関<br>係数 | 0.53<br>(0.12) | 0.50<br>(0.22) | 0.44<br>(0.21) | 0.40<br>(0.15) | 0.49<br>(0.16) | 0.38<br>(0.21) |
| | $p \geqq 0.05$<br>の人数 | 0 人 | 1 人 | 1 人 | 0 人 | 0 人 | 0 人 |
| 小指<br>第 2・3<br>関節間 | 相関<br>係数 | 0.53<br>(0.19) | 0.52<br>(0.16) | 0.43<br>(0.16) | 0.33<br>(0.18) | 0.50<br>(0.14) | 0.40<br>(0.25) |
| | $p \geqq 0.05$<br>の人数 | 0 人 | 1 人 | 2 人 | 0 人 | 0 人 | 3 人 |

表 6.10　本実験後の感想

| 分類 | テキストコード及び記述内容 | 件数 |
|---|---|---|
| 緩めるのが難しい | 難しい・簡単・苦手・押す：<br>「中に入れるまでは少し簡単だったけれど，外す時に最初うまく回らなくなってしまったので難しかったです」,「ドライバー試験をやって、ゆるく戻す方が難しいと思った」,「自分では思ったよりはやくできたと思います。抜く時が難しかったです」,「ネジを入れるよりも，抜く方が難しいと感じた」,「ネジを入れる時は簡単だったけれど，はずす時の力の入れ具合が難しかった」,「私が針をはめる時簡単だなと思いましたが，抜く時に，うまく回すことができなくて少し焦ってしまいました。→コツとしてドライバーを直角にして強く押すと，うまく回すことができる。」,「締める時はすごく簡単であっという間だったけど，緩める時は少し押さないとドライバーが動いちゃってすごく緩めにくかった」,「締めるのはできたけど，緩めてとるのが苦手でした。片手よりも両手でドライバーを握った方が締めやすかったです。」,「ドライバーを緩める時，押しながら回すのが大変で難しかったです。」,「中に入れる方が簡単だった。上に出す時は力の強さ（？）が分からなかった。回している時，ドライバーがネジからはずれて難しかった」 | 10 |
| システムへの感想 | キャプチャ：<br>「初めてモーションキャプチャを使い驚きました」,「キャプチャーを初めて見てすごいと思いました」,「パソコン（モーションキャプチャ）で手を映してやるのはすごいと思った」,「モーションキャプチャに驚きました」,「モーションキャプチャの動きが精密でびっくりした」,「今日，ドライバー試験をやってモーションキャプチャがすごく興味深かったです」,「初めて『モーションキャプチャ』を体験しましたが，少し不思議に思い興味が湧きました」,「モーションキャプチャは初めて見てすごいなと思った」,「『モーションキャプチャ』という技術を取り入れて客観的に細部まで詳しく見ることができ面白かったです。そういった研究も楽しそうだなと新しい進路として学ぶことも多くありました」,「新しい PC（パソコン)のでみた，手のやつ（モーションキャプチャ）がすごいな。と思った。センサーの仕組みなど，知りたいと思った」 | 10 |

# 第 7 章

# 押圧力測定装置の開発

前章では，手指の巧緻性により，ドライバ操作時の手首が速いという違いがあると分析したが，ネジに向けて押す力が適切でなければ，ネジの締め過ぎや締め不足になる恐れがある。そこで，本章では，ネジに向けて押す力を計測できるための装置と，簡易手指動作分析システムと併用し手指動作を計測できるかについて述べていく。

## 7.1　装置の概要

押圧力測定装置では，押圧力の計測からはじまり，保存したデータを基に定量的に分析するまでの流れを実現できる。実現するためには以下の機能が求められると判断した。

1. ドライバで正面から力を加えることができる。
2. 手動式ドライバと動力式ドライバそれぞれで押圧力を計測できる。
3. 押圧力を検出し，押圧力の大きさを取得できる。
4. 取得したデータを時系列に保存できる。
5. 取得したデータを基に，リアルタイムでグラフ表示できる。
6. 簡易手指動作分析システムと開発した装置をそれぞれ用い，手指動作と押圧力を計測できる。

以上 6 つの機能の実現を開発の方針とした。

開発方針 1, 6 の選定理由を説明する。押圧力の大きさは作品に取り付けるネジの位置・向きによって異なる。ドライバからネジに向けて力を加える時，ネジが下向きにある場面では身体の体重をかけやすいため，押圧力がネジへと働きやすい。一方，ネジが正面にある場面では身体の体重を意識してかけなければ，押圧力がネジへと働かない。そのため，ネジの向きは正面が下向きより押圧力の必要性が高く，ドライバ操作時，押圧力に気づきやすいと思われる。また，前章で行った中学生と大学生を対象にしたドライバ操作の分析

結果により，年齢・片手両手によらず，ドライバの柄を持つ手の手首を素早く回転させるほど，作業時間が短くなると分析した。しかし，ネジの向きが下向きから正面へと変化した時，押圧力をより意識した手指の動作でも同様の特徴となるか明らかにできていない。さらに，有川ら [?] は，中学生を対象に，ネジの向きが下向きでの手動式ドライバ操作を分析した結果，「両手での作業ではドライバの柄を回さない方の手で，柄の下部かドライバ軸を適切に支持することが，熟練者における効率的なネジ締めに関わっていること」，「中学生を対象としたドライバ技能指導を行う際の指導の要点は，脱落を防げる程度の押圧力，ネジを回せる程度の回す力，ネジを締める／緩める瞬間の軸の垂直性である」と示唆している。そのため，ネジの向きが下向きから正面へと変化した時，ドライバ軸の傾きが垂直になるよう調整する手指動作と押圧力との関係性がより重要になると思われる。つまり，生徒に押圧力を意識させ適正な押圧力となるようドライバ操作技能を習得させる時，手指の形が歪になったり，運動負荷が一部の関節にかかる動きに最適化されたりしないよう配慮した指導も重要と考えた。そこで正面からの押圧力を計測できるようにし，手指動作と押圧力との関係も含めたドライバ操作の技能指導を明らかにするため，開発方針 1,6 を選定した。

開発方針 2 は，ねじに向けた押す力を計測する装置に必要となる。開発方針 3〜5 は，手動式ドライバと動力式ドライバによる押圧力の比較や自分と他者の押圧力を比較するといった，定量的な技能学習に必要となる。

そこで，開発方針 1〜5 に従い，装置の形状，計測保存プログラムと押圧力表示プログラムを作成した。開発方針 6 は，評価試験で確認を行った。

### 7.1.1　装置の形状

開発の方針 1 を満たすため，ドライバ操作の押圧力を測定する時，装置を壁に立てかける活用場面を設定した。また，装置の寸法は，一般的な教育施設でも容易に活用できるため，普通教室用机の天板（幅 650mm ×奥行き 450mm）[158] 内に収まるようにした。図 7.1 に，装置の三面図を示す。図 7.2 に，装置を黒板に立てかけた様子を示す。

### 7.1.2　計測保存プログラム

計測保存プログラムは開発の方針 2〜4 に相当する。計測保存プログラムは Arduino IDE[159] で作成し，トグルスイッチを押すと計測が始まり，計測したデータを PC に送るとともに，SD カードにも保存される仕組みとなる。重量を測定できるセンサにはロードセルを用いた。ロードセルの種類は，松本ら [160] の報告において，動力式ドライバにおける押圧力の最大値が 119.6N であったため，20kg(196N) が最大測定値であるロードセル

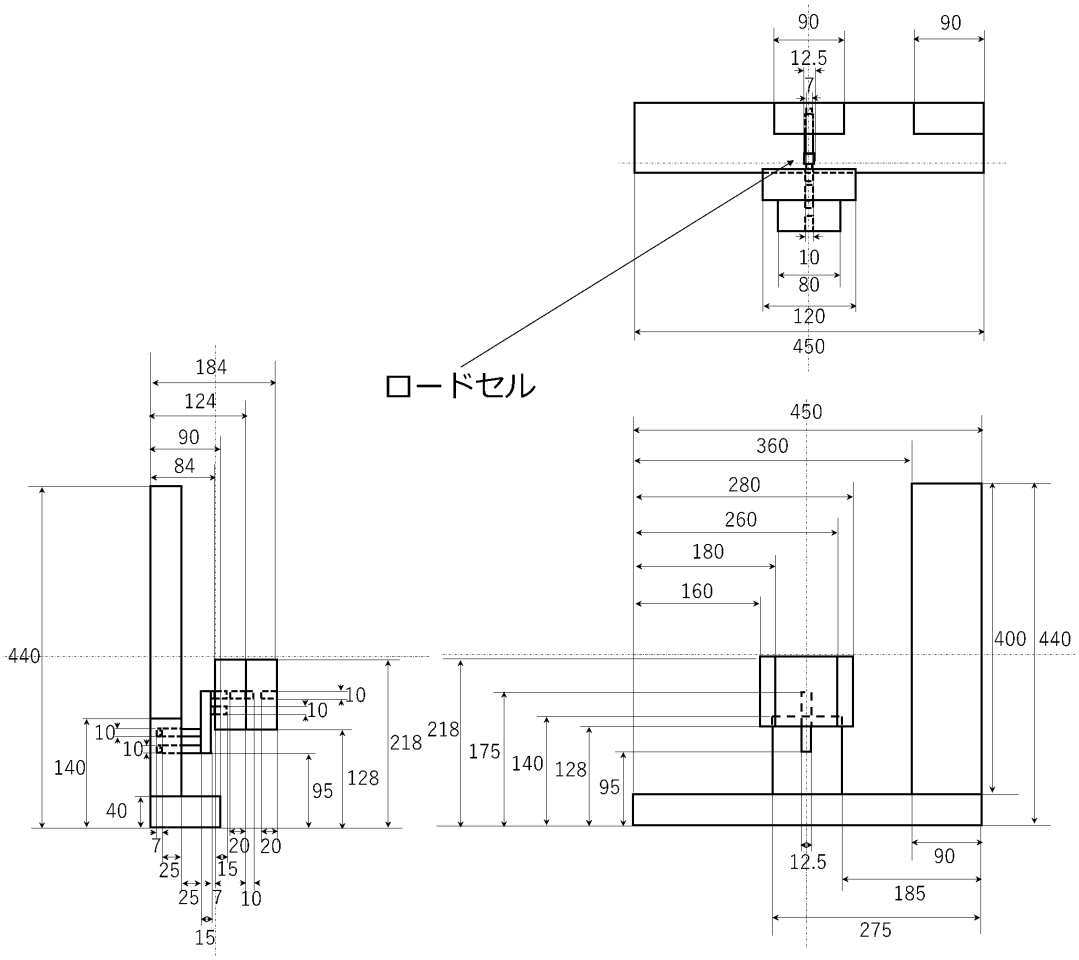


図 7.1　開発した装置の三面図

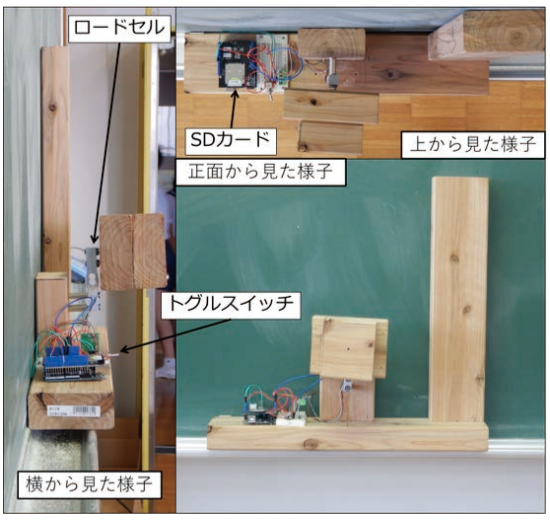


図 7.2　装置を黒板に立てかけた様子

を選択した。SD カードを用いるため，Arduino UNO R3 に SD カードシールドを装着した。また計測値からロードセルに取り付けた角材の重量を除くため，プログラム起動時に重量の初期値を設定する。図 7.3 に，計測保存プログラムのフローチャートを示す。図 7.4 に，計測保存プログラムの回路図を示す。

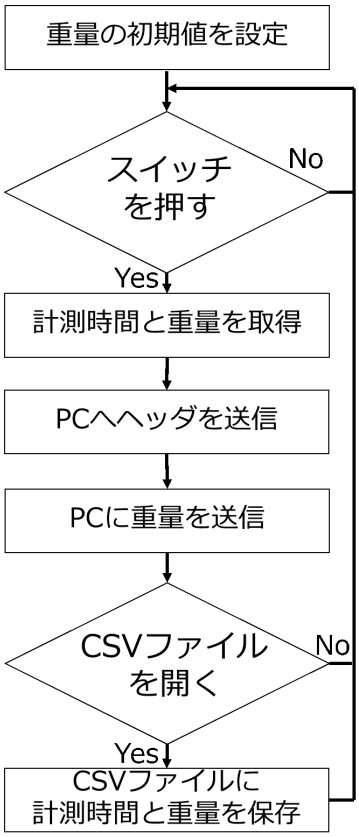


図 7.3　計測保存プログラムのフローチャート

### 7.1.3　押圧力表示プログラム

押圧力表示プログラムは開発の方針 5 に相当する。押圧力表示プログラムは Processing[145] を用いて作成し，押圧力の値が PC の画面上にリアルタイムで表示される仕組みとなる。図 7.5 に，計測時に画面上に表示されるグラフを示す。

図 7.5 の①では，Arduino から受信された重量を基に算出した押圧力の値がリアルタイムで表示される。数値は青色で表示される。これにより，作業者が操作した感覚を数値で確認することができる。図 7.5 の②は，画面上における押圧力の目盛りである。目盛りとその数値は黒色で表示され，押圧力が 27N となる目盛りのみ赤色で表示される。これは，有川らの報告 [57] により「手先が器用な学生が手動式ドライバ操作した時の押圧力が最大

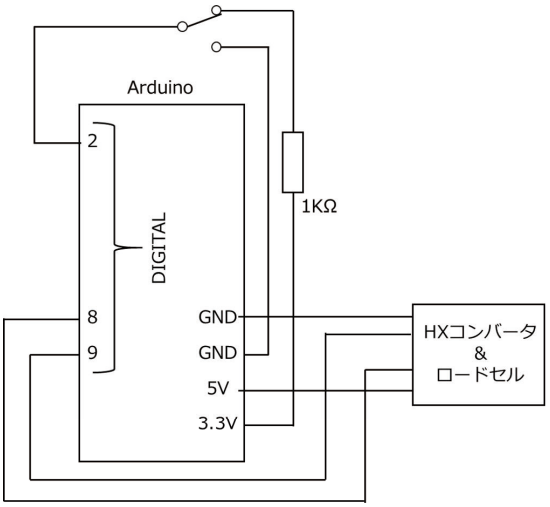


図 7.4　回路図

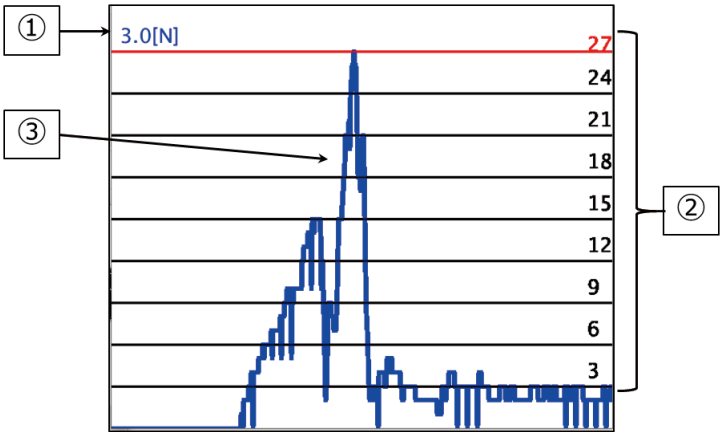


図 7.5　計測時に画面上に表示されるグラフ

35N であったこと」，松本らの報告 [161] により「作業方向によらず，動力式ドライバ操作時の理想的な押圧力を 30N に設定していたこと」を踏まえ，30N を超えないよう技能指導しやすくするために設けた。図 7.5 の③では，時間経過による押圧力の変化をグラフ化している。押圧力グラフは時間とともに左から右にずれていく。また，回路内のスイッチを切ると，グラフが止まるようになっている。これにより，作業者が変わるたびにスイッチを切り替えても，画面上でグラフを比較することができる。

## 7.2　評価試験

開発した装置の計測値の精度と実用性を検証するため，3 種類の評価試験を行った。1 種類目の評価試験（以後，試験 1）は，目標値の押圧力へと変化していく時，その変化が

グラフの特徴として表れることを確かめる目的で行った。2 種類目の評価試験（以後，試験 2）は，計測時に画面上に表示される数値が理論値により近似するため，押圧力の誤差を補正する近似直線の方程式を求める目的で行った。3 種類目の評価試験（以後，試験 3）は，簡易手指動作分析システムと併用できるかを確かめる目的で行った。

### 7.2.1　試験 1 の概要と結果

試験 1 では，目標値の押圧力を 121N に設定し，時間とともに押圧力が変化していく時，値の変化がグラフの特徴として表れるかを確かめた。図 7.6 に試験 1 の様子を示す。

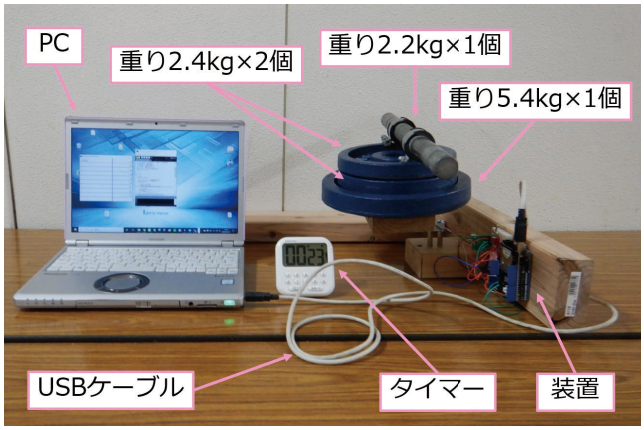


図 7.6　試験 1 の様子

試験 1 の実験方法は，まずタイマーとスイッチを押す。計測部分に計測開始から 5 秒で重り 5.4kg，10 秒で重り 2.4kg，15 秒で重り 2.4kg，20 秒で重り 2.2kg の順に重ねていく。計測開始から 25 秒立った時，スイッチを押し計測を終了した。この工程を 10 回行った。

図 7.7 に，10 回の計測値と理論値の変化をグラフ化したものを示す。図 7.7 より，重りを置く度に，計測値が変化し，理論値のグラフと同じ特徴が表れた。また，押圧力が大きくなるにつれ，実測値と理論値との間の誤差も大きくなることもわかった。

### 7.2.2　試験 2 の概要と結果

試験 2 では，計測時に画面上に表示される数値が理論値により近似するため，押圧力の誤差を補正する近似直線の方程式を求めた。図 7.8 に試験 2 の様子を示す。

試験 2 の実験方法は，まずタイマーとスイッチを押す。計測部分に計測開始から 5 秒で重りを置き，10 秒経過した時，スイッチを押し計測を終了した。この工程を 10 回行った。重りはペットボトルの容器 18g と 5 円玉約 3.75g を用い，300g，600g，901g，1201g，1499g，1799g，2099g，2399g，2699g の重量で行った。

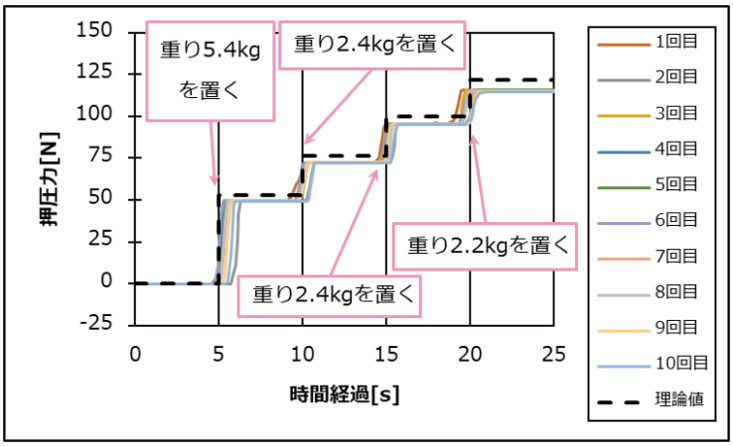


図 7.7　試験 1 の結果

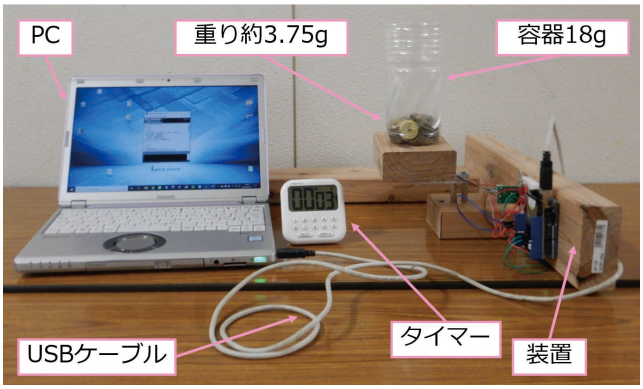


図 7.8　試験 2 の様子

図 7.9 に，押圧力の理論値に対し，10 回の計測結果を平均し算出した実測値との誤差をグラフ化したものを示す。図 7.9 より，押圧力の理論値が大きくなるにつれ，理論値と実測値との誤差が大きくなるとわかった。これは図 7.7 と同様の結果である。計測結果から，押圧力の誤差を補正するための一次関数の方程式は y=0.0642x-0.0192 となった。

### 7.2.3　試験 3 の概要と結果

開発した装置を用い，押圧力を計測する時，簡易手指動作分析システムを用い，大学生を対象に，手指の動作を計測することができるか確かめた。大学生のみを対象にしたのは，簡易手指動作分析システムを用い，手指動作の計測デバイスである Leap Motion を手指の上に設置した計測環境で，下向きのネジをドライバ操作する大学生と中学生の手指動作を計測することができたためである。つまり，大学生で横向きのネジをドライバ操作する手指動作を計測できれば，中学生も計測できると判断した。

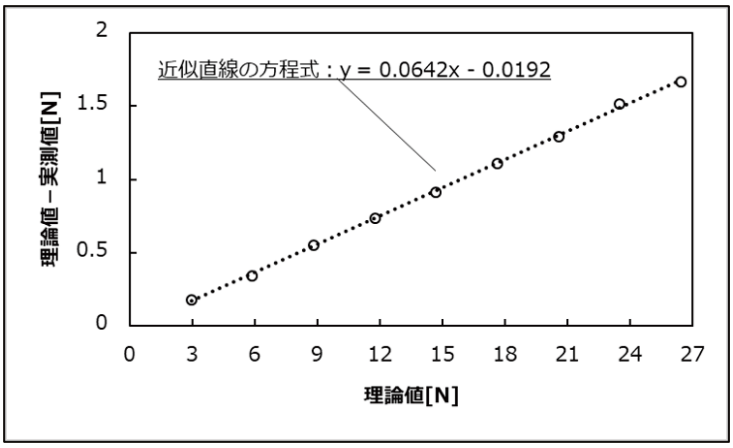


図 7.9　試験 2 の結果

実験方法では，2 台のノート型 PC（OS：windows8.1，windows10）を用意し，それぞれの PC に Leap Motion と装置を USB ケーブルで接続し，計測を行った。計測時，右手で手動式ドライバの柄を回す，左手で手動式ドライバの軸を固定する持ち方であった。図 7.10 に試験 4 の様子を示す。

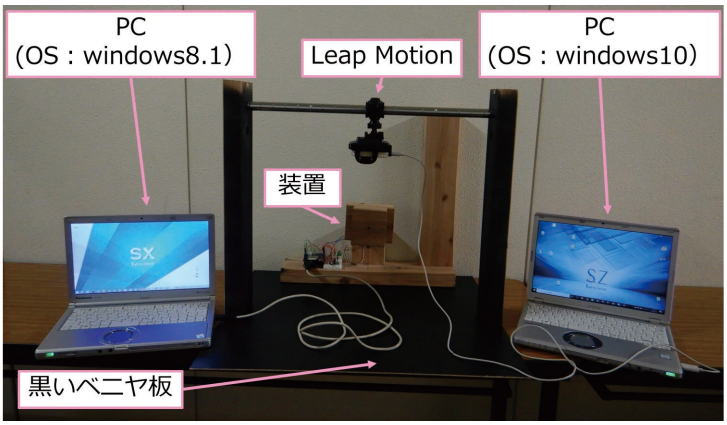


図 7.10　簡易手指動作分析システムとの併用試験する様子

図 7.11 に，左右で同じ親指の関節部位における左右方向の位置座標と押圧力をグラフ化したものを示す。図 7.11 より，右手と押圧力が左手に比べ規則的に変動しているとわかる。これは右手で手動式ドライバを回すのに合わせ押圧力も変化し，左手で手動式ドライバの軸を固定している特徴を捉えることができていると考えられる。そのため，簡易手指動作分析システムと開発した装置をそれぞれ用い，手指の動作と押圧力を計測することができたと言える。

また図 7.11 の写真より，右手の親指指先がドライバの柄を上から押さえる位置，左手

の親指指先がドライバの軸を上から押さえる位置にある持ち方をしているとわかる。そのため，左手親指指先と右手親指指先との位置関係は，ドライバ軸の傾きを検討する目安になると考えた。そこで，最大押圧力が 20N 未満であった試験 4 の大学生では左右の親指指先における位置関係がどうなるかを確認した。

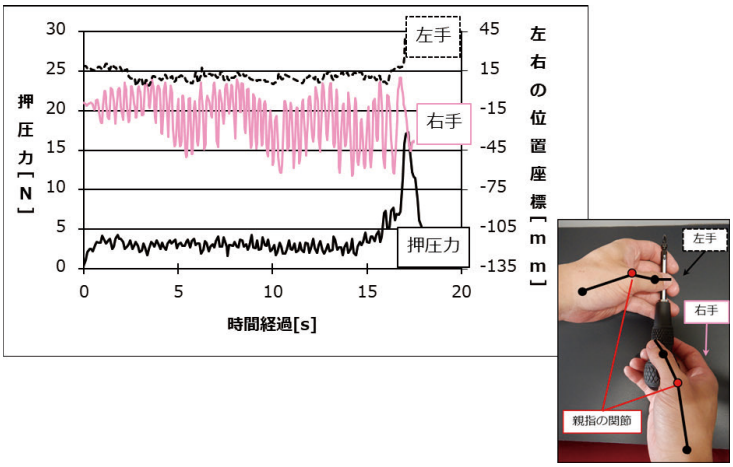





図 7.11　押圧力と左右の位置座標との関係

図 7.12 に，左手親指指先を原点とした，右手親指指先の位置座標を算出し，上と正面から見た右手親指指先の分布をプロットしたものを示す。尚，左手親指指先の位置座標における標準偏差は左右方向で 3mm，上下方向で 6mm，前後方向で 9mm であった。図 7.12 の上から見た様子より，右手の親指指先が左手の親指指先に対し 11°ほど平均してずれているとわかる。図 7.12 の正面から見た様子より，右の親指指先の分布が，理想とするドライバの柄の位置の内側より外側 (左上) に集中しているとわかる。そのため，大学生が行った持ち方では，右手の親指指先が左手の親指指先に対し 11°ほどの角度であり，ドライバの軸ずれが小さければ，20N 未満の最大押圧力でネジを締めることができたと考えられる。

## 7.3　本章のまとめ

本章では，ネジに向けて押す力を測定する装置を開発した。開発した装置と簡易手指動作分析システムを用い，押圧力と手指動作を同時に計測することができた。そこで，次章では，押圧力測定装置と簡易手指動作分析システムを用い，作業しやすいか否かにより，ドライバ操作時の手指動作に違いがあるかを確かめることにした。

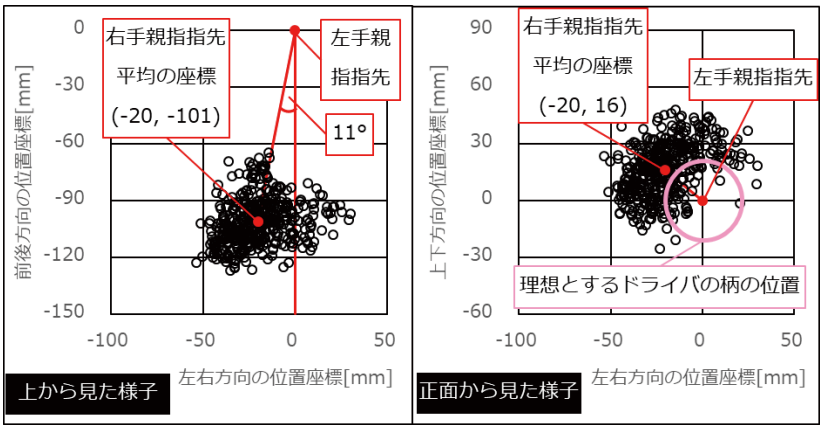


図 7.12　左右における位置座標の関係

# 第 8 章

# 作業しやすさによるドライバ操作時の手指動作の違い

前章では，ドライバ操作時の手指動作とネジに向けて押す力を計測できる環境を設定した。本章では，1.3 節の課題 3 を明らかにするため，作業しやすさの違いにより，ドライバ操作時の手指動作や押圧力にどういった違いがあるかを確かめた。

## 8.1　実験の目的

本実験の目的は，作業しやすさの違いにより，ドライバ操作時の手指動作や押圧力にどういった違いがあるかを確かめることである。作業しやすさに違いが生じやすくするため，ドライバの先端とネジ頭部の溝のサイズが同じか違うかという 2 つの実験条件を用意した。ドライバの先端とネジ頭部の溝のサイズが同じか否かにより，ドライバを操作した時，ネジが外れやすいか否かが変わってくるため，押圧力を無駄に加えすぎたり，押圧力が不足したりするといった違いが確認しやすいと考えた。そこで，対象の中学 1 年生を 2 群にわけ，ドライバの先端とネジ頭部の溝のサイズが同じ中学生群を同群 33 人，違う中学生群を異群 34 人とした。

本実験では，同群と異群の中学生の間で使用経験や握力の大きさで同じような実態であるかを確かめるため，アンケート調査や握力測定を行った。なお，対象の中学 1 年生は技術科の授業で予めネジを締めた後，緩めた状態の穴に，ドライバの先端がネジ頭部の溝と同じネジを締める作業を 1 度行い，技術科授業での使用経験を合わせた。

## 8.2　中学生群の実態

同群と異群の中学生群の間で，実態に違いがあるかを確かめた。事前アンケート調査，握力測定の順に述べていく。

### 8.2.1　アンケート調査

アンケートは選択式の質問 4 つと，自由記述式の質問 1 つの構成になっている。アンケートの質問は下記の通りである。

質問 1　あなたの利き手はどちらですか。
質問 2　ねじをしめたり，ゆるめたりするとき，どちらの手でドライバーを使いますか。
質問 3　中学生になってからの 3 年間，技術科の授業以外でドライバーを使いましたか。
質問 4　いつ，ドライバーを使いましたか。
質問 5　何をする（つくる）ために，ドライバーを使いましたか。（技術科の授業以外で，具体的に書いてください）

利き手の回答とドライバ操作時の利き手の回答は質問 1,2 が 3 択 (①：右利き，②：左利き，③:両利き) になる。ドライバ使用経験の回答は質問 3 が 4 択 (①:よく使っている，②:時々使っている，③:1,2 度使ったことがある，④:全く使わなかった) になる。ドライバ使用時期の回答は質問 4 が 10 択となる。ドライバ使用機会の回答は自由記述式で質問 5 となる。なお，同群の 2 人がすべての質問に対し，無回答であったため，同群

利き手の回答結果を述べる。同群 31 人の中では，右利きが 28 人 (90 %)，左利きが 3 人 (10 %) となった。異群 34 人の中では，右利きが 32 人（94 %），左利きが 2 人 (6 %) となった。そのため，2 群の 90 %が右利きであるとわかった。

ドライバ操作時の利き手の回答結果を述べる。両利きを選んだのは，同群 31 人中 2 人，異群 34 人 3 人であった。両群で 5 人の両利きを選んだ学生は質問 5 において，4 人が家で家具を組み立てるといった木材加工を行っていると記述していた。

ドライバ使用経験の回答結果を述べる。表 8.1 に群ごとに，ドライバ使用経験の回答結果を示す。表 8.1 より，同群と異群との間で回答項目ごとの割合に違いが 5 %程度であるとわかる。そのため，2 群の間でドライバ使用経験に違いがあまりないと判断した。

ドライバ使用時期の回答結果を述べる。表 8.2 に群ごとに，ドライバ使用時期の回答結果を示す。表 8.2 より，同群と異群との間で回答項目ごとの割合に違いが小学 1,2 年時と小学 3,4 年時の学校生活において 10 %程度，小学 3,4 年生時の学校生活でない日常生活・学校生活において 5 %程度の違いがあるとわかる。そのため，異群が同群に比べ，小学 1

表 8.1　ドライバ使用経験の回答結果

| 分類 | よく使っている | 時々使っている | 1,2 度使ったことがある | 全く使わなかった |
|---|---|---|---|---|
| 同群 31 人 | 2 人 (6 %) | 9 人 (29%) | 12 人 (39%) | 8 人 (26%) |
| 異群 34 人 | 2 人 (6%) | 8 人 (24%) | 15 人 (44%) | 9 人 (26%) |

～4 年時の学校生活でドライバを操作した経験をもつ学生が集まっていると判断した。

表 8.2　ドライバ使用時期の回答結果

| 被験者 | 回答時期 | 小学校入学前 | 小学 1,2 年 | 小学 3,4 年 | 小学 5,6 年 | 中学生 |
|---|---|---|---|---|---|---|
| 同群 31 人 | 日常生活 | 4 人 (13%) | 10 人 (32%) | 19 人 (61%) | 24 人 (77%) | 17 人 (55%) |
| | 学校生活 | 1 人 (3%) | 0 人 (0%) | 4 人 (13%) | 11 人 (35%) | 2 人 (6%) |
| 異群 34 人 | 日常生活 | 6 人 (18%) | 10 人 (29%) | 19 人 (56%) | 23 人 (68%) | 20 人 (59%) |
| | 学校生活 | 0 人 (0%) | 3 人 (9%) | 9 人 (26%) | 13 人 (38%) | 1 人 (3%) |

したがって，異群が同群に比べ，家での小学 1～4 年次のドライバ使用時期において，早い段階から扱っていた割合が高い可能性が考えられる。

### 8.2.2　握力測定

握力測定は左右交互に 2 回ずつ実施した。表 8.3 に，群ごとに握力測定の結果を信頼度 95 %の z 推定し算出した下限値と上限値にまとめたものを示す。表 8.3 より握力は群ごとの誤差が左右ともに 1.1kg 未満であるとわかる。そのため，同群と異群に握力差がないと判断した。

表 8.3　各群の握力

| 分類 | 右手 [kg] | 左手 [kg] |
|---|---|---|
| 同群 32 人 | [ 19.9 , 23.5 ] | [ 18.6 , 22.9 ] |
| 異群 32 人 | [ 20.4 , 23.5 ] | [ 18.5 , 21.8 ] |

### 8.2.3　各群の実態

同群と異群は利き手，ドライバ使用経験，および握力について，大きな違いがないと判断した。異群は同群より，小学 1〜4 年次の学校生活において，ドライバ使用した割合が高い可能性がある。そのため，異群が同群に比べ，ネジが外れやすい条件でも対応できる中学生群と判断した。

## 8.3　実験の方法

実験では，押圧力による違いが生じやすくするため，横向き状態の木ネジを締めるドライバ操作を行わせ，実験装置を用い，ドライバ操作時の手指動作と押圧力を計測した。手指動作を計測しやすくするため，黒いネジ ((+)2，直径 3mm，長さ 20mm) と，先端を (+)1 と (+)2 に変える事ができるベッセル社の黒いドライバ [155] を選んだ。中学生が材料固定を気にせず操作に集中できるよう，角材にある直径 2mm の下穴に，ネジを予め 5mm 挿入した状態からネジを締めさせた。作業手順は以下の通りで行った。

1. 中学生は Leap Motion に両手を出し，画面上にアニメーションが表示される。
2. 中学生はドライバを持ち，角材に挿入されたネジ頭部の溝にドライバの先端を固定する。
3. 筆者の合図とともに，ネジをドライバで締め始める。
4. 木ネジを締め終えたら，筆者が合図を出し，ドライバ操作をやめる。60s 以内に締める事ができない場合にも筆者が合図を出し，ドライバ操作をやめる。

筆者が合図した時，実験装置での計測開始・終了を行った。木ネジを締め終えたという筆者の判断は，締め残ったネジの長さが 1.5mm 以内とした。また中学生にはドライバ操作後，ドライバ操作の感想をまとめさせた。図 8.1 に，押圧力測定装置と簡易手指動作分析システムを設置した環境下で，中学生がドライバ操作する様子を示す。

## 8.4　実験の結果

### 8.4.1　締め残ったネジの結果と握力との関係

各群での締め残ったネジの長さを，1.5mm を基準にして分類した結果を表 2 に示す。表 8.4 では群ごとに締め残ったネジの長さを信頼度 95 %の z 推定し算出した下限値と上限値もまとめた。表 8.4 より，同群は異群よりネジを締める事ができた。またネジの長さは両群の間で差が 4mm 前後あり，マンホイットニーの U 検定で $p < 0.01$ となり，両群

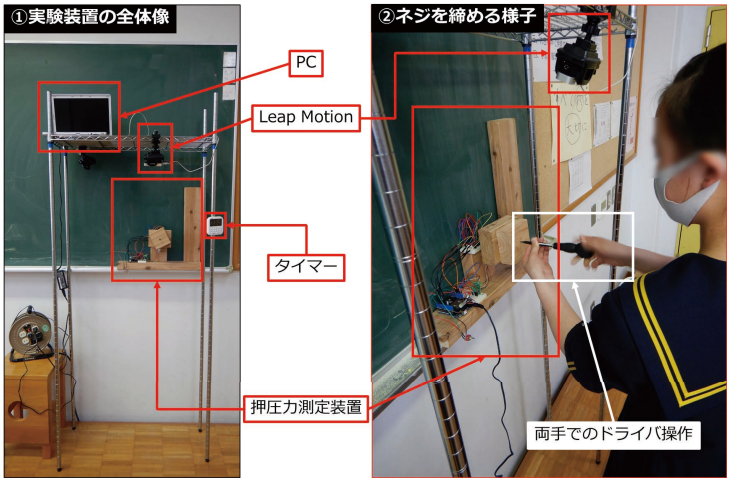


図 8.1　中学生がドライバを操作する様子

の間で有意差があるとわかった。残ったネジの長さ 1.5mm を基準とし分類した異群の 2 群間において，柄を持つ手の握力ではマンホイットニーの U 検定で $p < 0.05$ となり，有意差があるとわかった。なお残ったネジの長さが 1.5mm 以内になった柄を持つ手の握力値は小さい値から，同群が 14.0, 16.0, 16.8kg，異群が 21.8, 25.8, 25.8kg であった。

表 8.4　残ったネジの長さの結果

| 分類 | 残ったネジの分類 | | ネジの長さ |
|---|---|---|---|
| | 1.5mm 以内 | 1.5mm より長い | 信頼区間 [mm] |
| 同群 32 人 | 21 人 (66%) | 11 人 (34%) | [ 1.4 , 3.7 ] |
| 異群 32 人 | 5 人 (16%) | 27 人 (84%) | [ 5.3 , 8.3 ] |

### 8.4.2　両群の共通点

同群と異群では押圧力と押圧力の合計との関係に共通点があった。中学生ごとに作業時の押圧力の第 1〜3 四分位数を算出した。スペアマンの検定により，押圧力の第 1〜3 四分位数と押圧力の合計との間および，押圧力の四分位範囲と押圧力の合計との間ではすべて，$p < 0.05$ で相関係数が 0.30 以上になるとわかった。図 8.2 に同群の押圧力と押圧力の合計との関係を示す。

### 8.4.3　各群でのドライバ操作の要素間における相関

中学生ごとに，作業時の関節各部位と手首の角度データから代表値と散布度を算出し，握力や押圧力との相関をスペアマンの検定で確かめた。代表値は各角度の第 1〜3 四分位

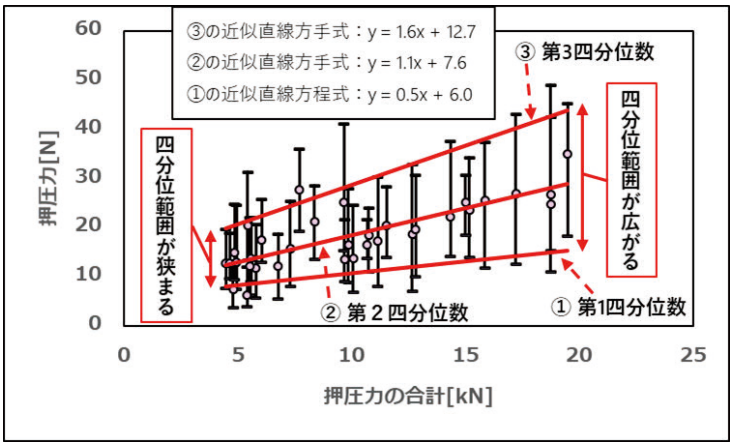


図 8.2　押圧力と押圧力の合計との関係

数，散布度は各角度の四分位範囲とした。角度の代表値 3 つ中 2 つの代表値で $p < 0.05$ と相関係数が $|0.30|$ 以上であった場合，その角度に相関があると判断した。残ったネジの長さとの間で有意差があるドライバ操作時の要素の相関係数は，同群では押圧力の最大値が- 0.50($p < 0.01$)，柄を持つ手の握力が-0.52($p < 0.01$)，軸を持つ手の握力が-0.50($p < 0.01$)，異群では押圧力の合計が 0.50($p < 0.01$) となった。そのため，残ったネジの長さが 0mm に近いほど，同群では押圧力の最大値や握力が大きく，異群では押圧力の合計が小さい傾向にあるとわかった。

握力と関節角度との間に相関関係があるドライバ操作時の要素について，同群を表 8.5 に，異群を表 8.6 へまとめた。表 8.5 より握力が小さく，残ったネジの長さが 0mm に近いほど，表 8.5 の D1, 5 にある関節角度の代表値や散布度が大きく，表 8.5 の D3, 6 にある関節角度の代表値が小さい傾向にあるとわかった。また表 8.5 の D2 にある関節角度の散布度が大きい中学生は，握力と押圧力の最大値が小さい傾向にあるとわかった。表 8.6 より，押圧力の合計が小さい中学生は，表 8.6 の I1～I4 にある関節角度の散布度が小さい傾向にあるとわかった。また，表 8.6 の I2，I4 にある関節角度の散布度が小さい時，押圧力の合計も小さくなる傾向にあるとわかった。なお，手指の関節角度との間で有意差が $p > 0.05$ になったのは，同群では押圧力の合計，異群では握力と残ったネジの長さであり，相関係数も $|0.30|$ 未満だった。

### 8.4.4　生徒の感想

ドライバを締めた感想を記述内容で分類し，両群の間で比較していく。

表 8.7 に，テキストコードが「外れる，抜ける，入る，離れる，はまる，合う，ずれる，くっつく」となり，ドライバとネジのサイズが一致しないことに触れた生徒の感想を示

表 8.5 ドライバ操作時の要素間の相関関係：同群の場合 ($p < 0.05$)

| 角度の種類 | 柄を持つ手指の関節角度 | | | |
|---|---|---|---|---|
| 関節と記述統計量の種類 | D1 人差し指第 1,2 関節の代表値 | D2 中指第 3 関節の散布度 | D3 薬指・小指第 3 関節の代表値 | D4 手首第 1 角度の代表値 |
| 残ったネジの長さ | [ 0.40 , 0.57 ] | | [ -0.42 , -0.32 ] | |
| 柄 (軸) を持つ手の握力 | [ -0.47 , -0.34 ] | -0.35 | [ 0.32 , 0.38 ] | [ 0.42 , 0.50 ] |
| 押圧力の最大値 | | -0.36 | | |
| 角度の種類 | 軸を持つ手指の関節角度 | | | |
| 関節と記述統計量の種類 | D5 人差し指第 2 関節，中指第 2 関節，薬指第 1,2 関節，小指第 1,2 関節の散布度 | | | D6 中指第 1 関節と小指第 1,2 関節の代表値 |
| 残ったネジの長さ | [ 0.40 , 0.52 ] | | | |
| 柄 (軸) を持つ手の握力 | [ -0.39 , -0.31 ] | | | [ 0.32 , 0.52 ] |
| 押圧力の最大値 | | | | |

表 8.6 ドライバ操作時の要素間の相関関係：異群の場合 ($p < 0.05$)

| 角度の種類 | 柄を持つ手指の関節角度 | | 軸を持つ手指の関節角度 | |
|---|---|---|---|---|
| 関節と記述統計量の種類 | I1 人差し指第 1 関節の散布度 | I2 人差し指第 2,3 関節の散布度 | I3 親指・小指第 2 関節の散布度 | I4 人差し指第 1 関節，中指 1,2 関節，薬指・小指第 1 関節の散布度 |
| 押圧力の最大値 | | [ 0.31 , 0.45 ] | | [ 0.38 , 0.47 ] |
| 押圧力の合計 | 0.39 | [ 0.37 , 0.38 ] | [ 0.37 , 0.38 ] | [ 0.35 , 0.50 ] |

す。表 8.7 より，同群は異群より，ドライバとネジのサイズが一致しないことに触れた生徒の数が多いとわかる。

表 8.7　ドライバとネジのサイズが一致しないことに触れた生徒の感想

| 分類 | テキストコード及び記述内容 | 件数 |
|---|---|---|
| 同群 | 外れる・抜ける・入る・離れる・はまる・合う・ずれる：「途中で，はずれたりして少し難しかった」,「ドライバーがぬけたりして大変だった」,「ドライバーがくぎにうまく入ってなくて苦戦した」,「何回か，ネジからドライバが離れてしまって効率が悪いと思った」,「ねじとドライバーがはまらなかった」,「ドライバーとネジが合わなかった」,「ネジとドライバーがずれてしまってタイムロスしてしまった」 | 20 |
| 異群 | 外れる・入る・はまる・合う・ずれる・くっつく：「意外にもネジをしめるのは難しく何回もはずれてしまいましたが，最後まで締めることができたときとてもうれしかったです」,「ねじにねじまわしが入らなかった」,「ドライバーのでこぼこが，なかなかねじにはまらなくて難しかった」,「ねじの『×』の所にドライバを合わせることが大切だと思いました」,「ドライバーとネジが噛み合わなくて，すべってしまった。なので，すべらないように工夫をしたい」,「ドライバーとネジがうまくくっつかなかった」 | 11 |

## 8.5　考察

同群と異群では締め残ったネジの長さで差が 4mm 前後あり，有意差も確認されたため，異群は同群より難易度が高いと言える。また異群では，ネジの長さにより分類した 2 つの群の間で，柄を持つ手の握力に有意差が確認されたため，ドライバの先端とネジ頭部の溝が違う条件でも握力の大きさが影響すると考えられる。そして，条件によらず，押圧力が小さい時，押圧力の合計が小さい傾向にあるが，同群では押圧力の最大値が大きい時，異群では押圧力の合計が小さい時，残ったネジの長さが 0mm に近くなるため，異群の条件は同群の条件より押圧力を小さくする意義に気づきやすいと考えられる。また，同群は異群より，ドライバとネジのサイズが一致しなかったことを生徒の感想に書いてくる割合が大きかった。これは，同群が異群に比べ，押圧力を加えずネジが締まる環境下であったため，ネジ締めをいかに速く行うかに意識が向きやすく，ドライバの先端がネジの頭部にある溝から外れるタイムロスが印象に残ったのではないかと考えられる。

## 8.6　本章のまとめ

本章では，作業しにくいか否かにより，ドライバ操作時の手指動作や押圧力にどういった違いがあるかを確かめるため，中学生を対象に，実験を行った。作業しにくいか否かは，ドライバの先端とネジ頭部の溝が同じサイズか否かといった条件設定をした。ドライバ操作時の手指動作は第 5.4 章において，簡易手指動作分析システムを改良したモーションキャプチャシステムで計測した。押圧力は第章：押圧力測定装置の開発章において開発した測定装置を PC と接続せず起動する，押圧力測定装置で測定した。実験の結果，以下の 4 点が明らかになった。

1. 作業しやすい条件は作業しにくい条件に比べ，ネジを締めることができるとわかった。
2. 作業しにくいか否かによらず，作業中の押圧力が小さい学生は，作業終了時の押圧力の合計が小さいことがわかった。
3. 作業しやすい条件では，押圧力の最大値が大きい程，残ったネジの長さが短く，作業しにくい条件では，押圧力の合計が小さい程，残ったネジの長さが短いということがわかった。
4. 握力が小さい学生は，ドライバの柄を握る手の人差し指が伸び，薬指・小指を柄を覆うように持っていない持ち方となり，ドライバの軸を握る手指の動きが大きくなり，ネジを締めきることができない傾向にあるとわかった。

そこで，本章で得た結果と第 6 で得た結果をもとに，ドライバ操作の技能指導上の要点リストを作成した。次章では，作成したドライバ操作の技能指導上の要点リストについて述べていく。

# 第 9 章

# ドライバ操作の技能指導上の要点リストの作成

前章では，作業しにくいか否かにより，ドライバ操作時の手指動作や押圧力にどういった違いがあるかを確かめた。その結果，握力が小さい学生は，ドライバの柄を握る手の人差し指が伸び，薬指・小指を柄を覆うように持っていない持ち方となり，ドライバの軸を握る手指の動きが大きくなり，ネジを締めきることができない傾向にあるとわかった。本章では，1.3 節の課題 4 を明らかにするため，次章で活用するドライバ操作時の指導上の要点を作成していく。

## 9.1　ドライバ操作の技能指導上の要点リスト

第 6 章と，第 10.4.4 章および，有川ら [57] の研究結果を参考に，ドライバ操作の技能指導上の要点リストを作成していく。作成にあたり，ドライバ操作の技能指導上の要点，開発の方針，作成した要点リストについての順に述べていく。

### 9.1.1　ドライバ操作の要点整理

ドライバ操作の技能指導上の要点を明らかにするため，第 6 章と，第 10.4.4 章および，有川ら [57] の研究結果を整理していく。

第 6 章ではドライバ操作の技能指導における初期段階では，以下の結果を伝える必要があると考えた。

6-1 ：柄を利き手で上から被せるように握る
6-2 ：両手操作で行う
6-3 ：親指の動きを大きく規則的にする

6-4 ：各指の関節各部位と手首における角度変化を大きく動かす

第 10.4.4 章では以下の結果を得ることができた。

10.4.4-1 ：作業しにくいか否かによらず，作業中の押圧力が小さい学生は，作業終了時の押圧力の合計が小さいことがわかった。

10.4.4-2 ：作業しやすい条件では，押圧力の最大値が大きい程，残ったネジの長さが短く，作業しにくい条件では，押圧力の合計が小さい程，残ったネジの長さが短いということがわかった。

10.4.4-3 ：握力が小さい学生は，ドライバの柄を握る手の人差し指が伸び，薬指・小指を柄を覆うように持っていない持ち方となり，ドライバの軸を握る手指の動きが大きくなり，ネジを締めきることができない傾向にあるとわかった。

有川ら [57] の研究結果は以下の通りである。

有-1 ：【両手及び片手での操作に共通する示唆】柄を利き手で上から被せるように握り，脱落しない程度の押す力とねじを回せる程度の回す力を加える。利き手はねじの固さ (回す力に対する抵抗) に応じて適宜持ちかえ，できるだけ同じペースで回す力 (押す力) を加える。

有-2 ：【両手での操作に対する示唆】支え手は柄の下部または軸に添え，軸が真っ直ぐ (垂直に) ぐらつかないように支える。利き手持ちかえの際も (軸がずれ・ぶれないよう) 支える。

有-3 ：【片手での操作に対する示唆】片手では (支える手がないため) 軸のずれ・ぶれが起きやすい。回す力 (押す力) を加える際は，軸が真っ直ぐ (垂直に) ぐらつかないよう意識して，特に回す力を意識して強めに加える。

これらの研究成果から，ドライバ操作の技能指導上初期段階において，伝える要点を以下のように整理した。

- 両手で操作を行う
- 柄を利き手で上から被せるように握る。
- 手首と 5 本の指全てで回す力を加える。
- ネジがずれない程度の押す力を加える。
- 親指を規則的に動かす。
- 支え手は軸が真っ直ぐ (垂直に) ぐらつなかないように支える。

### 9.1.2　要点リスト作成の方針

ドライバ操作の技能指導上の要点リストを作成するに当たり，以下の条件を基本方針とした。

1. 9.1.1 節の要点をまとめた概要をのせる。
2. 握力が小さい学生でもネジ締めできるようにするドライバの持ち方を伝えることができる。
3. 生徒が要点リストを用い，自己評価した結果を残しやすくする。

基本方針 1 は，9.1.1 節の要点を一度に伝えても理解しにくかったり，習熟しにくかったり，習得できている要点が各自で異なっていたりすることが考えられる。そこで，9.1.1 節の要点をまとめた概要を最初にのせ，その細かい説明として 9.1.1 節の要点を記す。

基本方針 2 は，10.4.4-3 より，握力が小さい中学生を中心に，柄を 5 本の指全てでしっかり握らず，軸を持つ手がよく動いてしまうため，その持ち方にならないよう伝える必要があると考えた。そのため，基本方針 2 を設定した。

基本方針 3 は，要点リストにおいて，記載される情報が多く，作業中に意識できなかったり，自己評価しながら練習するのが向く学生がいたりする可能性が考えられる。そこで，基本方針 3 を設定し，自身の学習しやすい方法に合わせられるよう，要点リストに何回でも自己評価できるチェック欄を載せた。

### 9.1.3　作成した要点リストについて

9.1.2 節の方針をもとに作成したドライバ操作の技能指導上の要点リストを，図 9.1 を示す。

図 9.1 にあるポイントの内容の中で「概要，No.1, 2, 3, 5」が第 6 章，「No.4」が有川らの研究 [57] と前章の結果，「No.6」が有川らの研究 [57] からまとめた。No.6 について，「支え手は柄の下部または軸に添え，軸が真っ直ぐ (垂直に) ぐらつかないように支える。」[57] といったまとめ方になっており，第 10.4.4 章の条件下でネジを支える中学生もいなかったが，第 10 章のドライバ操作の練習では下穴にネジを挿入する操作も含まれるため，記述を除いた。図 9.1 にある注意点の中で「No.1」が表 3 の D1，「No.2」が表 8.5 の D2, 3，「No.3」が表 8.5 の D5, 6 の結果からまとめた。柄を持つ手 (図 9.1 の注意点 No.1, 2) では，握力が小さい中学生は人差し指を伸ばしたり，薬指・小指がドライバの柄を覆うように握っていたりして，残ったネジの長さが長い手の形である事を考慮した。また握力が小さい中学生は中指の動きを大きくし，押圧力の最大値が小さくなるため，中指の動きも

**〇ポイント：　ネジをゆるめる際にも手首とすべての指が動かしやすい手の形で，**
**ドライバがぐらつかないようネジを押し，回す。**
↓

| No | チェック項目 | 1回目 | 2回目 | 3回目 | 4回目 | 5回目 |
|---|---|---|---|---|---|---|
| 1 | 両手で行う。 | □OK | □OK | □OK | □OK | □OK |
| 2 | 柄を利き手で上から被せるように握る。 | □OK | □OK | □OK | □OK | □OK |
| 3 | 手首と5本の指全てで回す力を加える。 | □OK | □OK | □OK | □OK | □OK |
| 4 | ネジがずれない程度の押す力を加える。 | □OK | □OK | □OK | □OK | □OK |
| 5 | 親指を規則的に動かす。 | □OK | □OK | □OK | □OK | □OK |
| 6 | 支え手は軸が真っ直ぐ（垂直に）ぐらつなかないようにドライバを支える。 | □OK | □OK | □OK | □OK | □OK |

**〇注意点：　握力がないという自覚がある人は，こんな手の形になっていませんか？**
↓

| No | チェック項目 | 1回目 | 2回目 | 3回目 | 4回目 | 5回目 |
|---|---|---|---|---|---|---|
| 1 | 利き手の人差し指が伸びていませんか？ | □OK | □OK | □OK | □OK | □OK |
| 2 | 利き手中指・薬指・小指が握ったままになっていませんか？ | □OK | □OK | □OK | □OK | □OK |
| 3 | 支え手は軸を握りすぎていませんか？ | □OK | □OK | □OK | □OK | □OK |

**〇他の人・先生の話や，気付いたことは，メモしましょう！**

図 9.1　ドライバ操作の技能指導上の要点リスト

小さくならないように考慮した。ドライバの軸を持つ手 (図 9.1 の注意点 No.3) では，握力が小さい中学生は，人差し指から小指までの 4 本に無駄な動きがあり，残ったネジの長さが長くなる。また，握力が小さい中学生は，中指と小指の関節角度が小さいが，残ったネジの長さが 0mm に近づくわけではない。そのため，支え手で軸を握りすぎてしまい回しにくくなる可能性を考慮し，握りすぎないようにする表記を No.3 に記述した。

## 9.2　本章のまとめ

本章では，ドライバ操作の要点を整理し，要点リスト作成上の基本方針をたて，図 9.1 にあるドライバ操作の技能指導上の要点リストを作成した。次章では，中学生を対象に，作成した要点リストを用い，練習を繰り返した時，どういった変化が起こるのかを明らかにしていく。また，図 9.1 の内容はドライバ操作の初期段階における要点を含み，ドライバ操作の練習をする前後でどちらが効果的か不明である。そのため，次章では図 9.1 を活用するタイミングを変え，技能指導によるドライバ操作の要素間の関係性の変化も分析していく。

# 第 10 章

# ドライバ操作の技能指導の評価

前章では，ドライバ操作の指導上の要点を作成した。本章では，1.3 節の課題 4 を明らかにするため，ドライバ操作の指導上の要点を用い，中学生がドライバ操作時の手指動作にどういった変化が起きるか，作業時間が短くなるかを検証した。

## 10.1　実験の目的

試験 2 の目的は，ドライバ操作の技能指導による要素間の関係性の変化を確かめる事である。そのため，試験 1 の中学生群を 2 つに分け，ドライバ操作の技能指導上の要点を見せドライバの練習を繰り返し，握力測定と，ドライバ操作時の手指動作と押圧力の計測，および事後のアンケート調査を行った。調査は 2021 年 4 月〜5 月，私立 S 中学 2 年生計 67 名を対象に，図 4 のリストをドライバ操作練習の前半で活用した前群 34 名と，図 4 のリストをドライバ操作練習の後半で活用した後群 33 名に分類した。

## 10.2　中学生の実態

中学生群の間で握力と試験 1 の結果に違いがあるかを確かめた。握力は左右ともに，前群と後群との間で誤差が 0.6kg 未満であった。前群と後群の分類に合わせ，表 2 の結果を表 10.1 に変換した。表 10.1 より，締め残ったネジの長さは後群が前群に比べ，締め残ったネジの長さが 1.5mm 以内となる人数が多かった。また，前群の異群 (表 5 の②) は後群の異群 (表 5 の④) より，ネジの長さが短かった。しかし前群と後群の同群間 (表 3 の①と③)，異群間 (表 5 の②と④) にはマンホイットニーの U 検定でそれぞれ $p > 0.05$ となり，有意差がなかった。そのため，前群と後群の間には大きな差がないと判断した。

表 10.1　各群における前章の結果

| 群 | 前章での分類 | 残ったネジの分類 | | 信頼区間 [mm] |
|---|---|---|---|---|
| | | 1.5mm 以内 | 1.5mm より長い | |
| 前群 33 人 | ①同群 15 人 | 10 人 (31%) | 5 人 (16%) | [ 0.8 , 3.9 ] |
| | ②異群 18 人 | 1 人 (3%) | 17 人 (53%) | [ 6.0 , 9.1 ] |
| 後群 31 人 | ③同群 17 人 | 11 人 (34%) | 6 人 (19%) | [ 1.1 , 4.4 ] |
| | ④異群 14 人 | 4 人 (13%) | 10 人 (31%) | [ 3.3 , 8.4 ] |

表 10.2　実験計画

| No | 前群 | 後群 |
|---|---|---|
| 1 | 要点を見てドライバ操作の練習 | 何も見ず，ドライバ操作の練習 |
| 2 | 前章の実験・握力測定 | 前章の実験・握力測定 |
| 3 | 何も見ず，ドライバ操作の練習 | 要点を見てドライバ操作の練習 |
| 4 | 前章の実験・握力測定 | 前章の実験・握力測定 |

## 10.3　実験方法の概要

実験では，ドライバ操作の練習と試験 1 の実験・握力測定をそれぞれ 2 回行った後，アンケート調査した。ドライバ操作の練習は 90 分で行った。表 10.2 に実験計画を示す。表 10.2 の No.1〜3 は 1 週間，No.3 と No.4 の間は 2 週間空けて行われた。表 10.2 のドライバ操作の練習は以下の通りに行った。

1. ネジを速く締めきる事を目標にすると説明する。
2. (+)1 のドライバを用い，直径 2mm の下穴を 7 × 7 の計 49 個，10mm 間隔であけた角材に，ネジ ((+)2，直径 3mm，長さ 20mm) を 1 本締めさせる。制限時間 60s で実施し，教員が経過時間を言い，作業にかかった時間と残ったネジの長さを記録させた。
3. (+)2 の先端に付け替えたドライバを用い，制限時間 180s で，筆者が指定した形にネジを締めさせる。この時，4 名班で 2 名がドライバ操作を行い，もう 2 名が相手の動きをタブレットで撮影させる。
4. 動画で自分の動きを確認したり，ドライバ操作時のポイントを話し合ったりさせる。
5. (+)2 のドライバで好きにネジを締める練習をさせる。この時，片方の手で柄をもち，もう一方の手で木を回すドライバ操作を禁止にするよう伝えた。
6. (+)1 の先端に付け替えたドライバを用い，筆者が指定した形にネジを締めさせる。3 と同様に，2 名ずつドライバ操作し，タブレットで動きを撮影させる。

7. 4, 5 の流れで練習させる。
8. ドライバの先端を (+)2 に付け替えさせ，締めたネジを緩めさせる。(+)1 で緩めたい場合，許可した。
9. 2 と同様に，2 の条件下で，60s のネジ締めと，その結果を記録させた。その後，「自分が思うドライバ操作のポイント（情報）」も書かせる。

上記の手順 2, 6 で教員が指定した形を図 5 に示す。表 10.2 にある前群の No.1 と後群の No.3 では，上記の手順 2, 3 の間で，図 10.1 のプリントを配布し，ドライバ操作の要点を伝えた。表 10.2 にある前群の No.3 と後群の No.1 では上記の手順 7,8 の間でドライバ操作の要点を 4 名班で 1 つまとめ，発表させた。相手の動きをタブレットで撮影する様子を図 10.2 に示す。表 10.2 の No.4 をした後，事後アンケートを行った。また試験 1 と表 10.2 の No.2, 4 で一度も 1.5mm 以下にならなかった中学生へ先端が (+)2 のドライバでのネジ締め試験も行った。

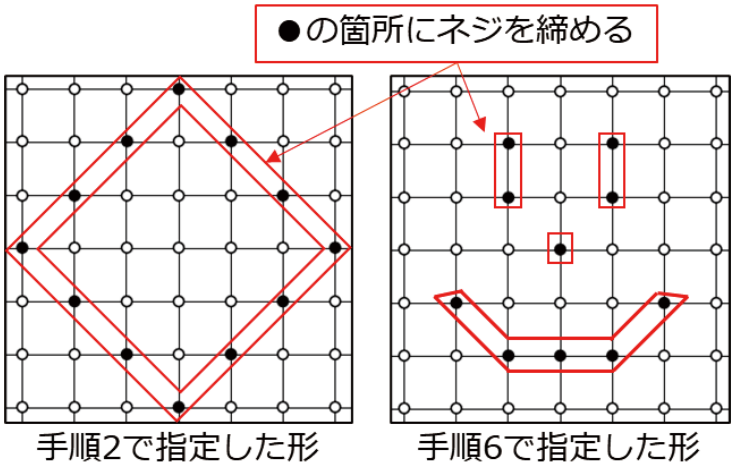


図 10.1　中学生がネジを締める教員が指定した形

## 10.4　実験の結果

### 10.4.1　握力と締め残ったネジの長さの関係

群ごとに握力を信頼度 95 %の z 推定し算出した下限値と上限値にまとめたものを表 10.3 に示す。表 10.3 より握力は左右ともに誤差が 1.5kg 未満であるとわかった。なお，両群で残ったネジの長さが 1.5mm 以内になった柄を持つ手の握力は小さい値から，12.0, 14.5, 15.3kg であった。また両群で残ったネジの長さが 1 度も 1.5mm 以内にならなかった中学生は 4 名おり，柄を持つ手の握力が 14.9, 18.8, 19.0, 30.8kg であった。

図 10.2　相手の動きをタブレットで撮影する様子

表 10.3　握力測定の信頼区間

| 群 | 測定回数 | 右手 [kg] | 左手 [kg] |
|---|---|---|---|
| 前群 | 1 回目 34 人 | [ 19.9 , 23.5 ] | [ 19.5 , 23.2 ] |
| | 2 回目 32 人 | [ 20.3 , 24.3 ] | [ 19.1 , 23.0 ] |
| 後群 | 1 回目 33 人 | [ 20.0 , 23.0 ] | [ 19.5 , 22.5 ] |
| | 2 回目 33 人 | [ 19.6 , 22.8 ] | [ 18.8 , 21.9 ] |

群ごとに締め残ったネジの長さを 1.5mm 以内かに分類した結果を表 8 に示す。表 10.4, 10.5 より，1 回目から 2 回目の計測にかけて，1.5mm より長い中学生は両群ともに減少し，後群での残ったネジの長さが長くなったとわかる。マンホイットニーの U 検定により，2 回目の計測時の後群は 1 回目の後群や 1, 2 回目の計測時の前群との間で $p < 0.01$ となり，1 回目の計測における前群内の同群と異群との間では $p < 0.05$，2 回目の計測における後群の同群と前群の異群との間では $p < 0.01$ となり，有意差があるとわかった。また前群では表 10.1 の異群と 1, 2 回目の計測の異群との間，後群では表 10.1 の異群と 1 回目の計測の異群との間でマンホイットニーの U 検定をした結果，$p < 0.05$ となった。

### 10.4.2　各群でのドライバ操作の要素間における相関

前章と同様の分析方法により，2 回目の計測におけるドライバ操作時の要点間の関係をまとめた。残ったネジの長さとの間で有意差があるドライバ操作時の要素の相関係数は，前群では押圧力の合計が 0.59($p < 0.01$)，柄を持つ手の握力が- 0.45($p < 0.01$)，軸を持

表 10.4　群ごとの残ったネジの長さ：1 回目の計測

| 群 | 前章での分類 | 人数 | 1.5mm 以内 | 1.5mm より長い | 信頼区間 [mm] |
|---|---|---|---|---|---|
| 前群 | 全体 | 34 人 | 22 人 (65%) | 12 人 (35%) | [ 1.6 , 3.6 ] |
| | 同群 | 15 人 | 7 人 (47%) | 8 人 (53%) | [ 2.0 , 5.4 ] |
| | 異群 | 18 人 | 14 人 (78%) | 4 人 (22%) | [ 0.7 , 2.9 ] |
| | 不参加 | 1 人 | 1 人 (100%) | 0 | 1 |
| 後群 | 全体 | 33 人 | 21 人 (64%) | 12 人 (36%) | [ 1.6 , 3.7 ] |
| | 同群 | 17 人 | 11 人 (65%) | 6 人 (35%) | [ 1.2 , 4.2 ] |
| | 異群 | 14 人 | 10 人 (71%) | 4 人 (29%) | [ 0.7 , 4.0 ] |
| | 不参加 | 2 人 | 0 | 2 人 (100%) | 3 , 6.9 |

表 10.5　群ごとの残ったネジの長さ：2 回目の計測

| 群 | 前章での分類 | 人数 | 1.5mm 以内 | 1.5mm より長い | 信頼区間 [mm] |
|---|---|---|---|---|---|
| 前群 | 全体 | 32 人 | 18 人 (53%) | 14 人 (41%) | [ 1.4 , 2.9 ] |
| | 同群 | 14 人 | 7 人 (50%) | 7 人 (50%) | [ 1.9 , 5.1 ] |
| | 異群 | 17 人 | 11 人 (65%) | 6 人 (35%) | [ 1.1 , 2.0 ] |
| | 不参加 | 1 人 | 0 | 1 人 (100%) | 5.7 |
| 後群 | 全体 | 33 人 | 11 人 (33%) | 22 人 (67%) | [ 3.5 , 6.1 ] |
| | 同群 | 17 人 | 5 人 (29%) | 12 人 (71%) | [ 3.3 , 6.9 ] |
| | 異群 | 14 人 | 6 人 (43%) | 8 人 (57%) | [ 2.0 , 6.0 ] |
| | 不参加 | 2 人 | 0 | 2 人 (100%) | 4 , 11 |

つ手の握力が- 0.48($p < 0.01$)，後群では押圧力の最大値が 0.34 ($p < 0.05$)，押圧力の合計が 0.66($p < 0.01$) となった。そのため，残ったネジの長さが 0mm に近いほど，前群では押圧力の合計が小さいが，握力が大きく，異群では押圧力の最大値・合計が小さい傾向とわかった。なお，握力との間に相関関係がある関節角度の中で，他のドライバ操作時の要素との間に相関関係がある関節角度は前群と後群になかった。

### 10.4.3　残ったネジの長さが 1.5mm 以下の群の特徴

表 10.5 より，残ったネジの長さが 1.5mm 以下になった中学生 29 名（以後，到達群）とそれ以外の中学生 36 名（以後，未達群）に分類した。表 10.6 に，到達群と未達群における作業時間，押圧力の合計および人差し指第 1, 2 関節の角度を平均と標準偏差で示す。表 10.6 では角度の平均と標準偏差が 5.3.2 節で算出した代表値 (第 1〜3 四分位数) の集合から求め，群の間でマンホイットニーの U 検定による結果も示した。表 10.6 より，到

達群は未達群に比べ，作業時間や押圧力の合計，人差し指の角度が小さいとわかった。なお，到達群と未到達群との間で握力に有意差はなかった。

表 10.6　両群でのドライバ操作の要素の平均と標準偏差

| 群 | 作業時間 [s] | 押圧力の合計 [kN] | 人差し指 | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 第 1 関節の角度 [°] | | | 第 2 関節の角度 [°] | | |
| | | | 第 1 四分 | 第 2 四分 | 第 3 四分 | 第 1 四分 | 第 2 四分 | 第 3 四分 |
| 到達群 | 34.2 (11.0) | 8.4 (3.5) | 143.0 (5.4) | 150.7 (7.3) | 161.5 (9.0) | 121.4 (13.4) | 136.5 (14.5) | 151.7 (16.1) |
| 未達群 | 54.9 (9.3) | 14.1 (4.0) | 146.1 (6.0) | 156.1 (8.9) | 165.6 (8.2) | 126.6 (15.7) | 144.6 (16.1) | 158.8 (12.8) |
| p | ** | ** | | ** | * | * | * | * |

*：p<0.05　，　**：p<0.01

到達群において，4.3.3 節と同様の分析方法により，握力と関節角度との間で相関関係にあるドライバ操作時の要素間の関係を表 10 に示す。表 10.7 より，柄を持つ手では握力が小さい時，「親指・中指の第 1 関節の角度の散布度と薬指・小指の第 1 関節の角度の代表値」が大きい特徴をもつ一方，「親指・中指の第 1 関節の角度の散布度と薬指・小指の第 1 関節の角度の代表値」が小さい時，作業時間が短く，押圧力の合計・最大値が小さい特徴をもつとわかった。軸を持つ手では握力との相関がないが，5 本すべての関節各部位と手首の角度の散布度が押圧力の最大値との間で，相関係数が 0.39 以上の関係にあるとわかった。なお，残ったネジの長さとの間で有意差があるドライバ操作時の要素の相関係数は柄を持つ手の握力が- 0.34($p < 0.05$)，軸を持つ手の握力が- 0.36($p < 0.05$) となった。

### 10.4.4　技能習熟度別による手先の器用の分類

有川ら [57] らのドライバ操作技能を 4 段階に分類した結果を基に，技能習熟度別によるドライバ操作技能の分類し，表 10.8 に示す。

技能習熟度別によるドライバ操作技能の分類するため，有川ら [57] らのドライバ操作技能にある記述をどう解釈していったのかを説明していく。

ドライバ操作技能「1」と判断する学生は，本研究では 1.5mm まで締めることができなかった，未到達群であると判断した。

ドライバ操作技能「2」と判断する学生は，「手際が悪い (時間を要する)」とあり，これはネジを締めるまで時間がかかったと解釈した。そこで，有川らのドライバ操作技能「1」と判断する学生に比べ，ネジを締めるまでの時間がかかった場合に「手際が悪い (時間を

表 10.7　ドライバ操作時の要素間の相関関係：到達群の場合 ($p < 0.05$)

| 角度の種類 | 柄を持つ手指の関節角度 | | | | |
|---|---|---|---|---|---|
| 関節と記述統計量の種類 | 親指第 1 関節の散布度 | 中指第 1 関節の散布度 | 薬指第 1 関節の代表値 | 小指第 2 関節の代表値 | 小指第 1 関節の代表値 |
| 残ったネジの長さ | 0.31 | | | | |
| 作業時間 | 0.34 | 0.35 | [ 0.39 , 0.44 ] | | [ 0.47 , 0.48 ] |
| 柄を持つ手の握力 | -0.30 | -0.36 | [ -0.41 , -0.38 ] | [ -0.40 , -031 ] | [-0.43 , -0.40 ] |
| 押圧力の最大値 | | | | | [ 0.36 , 0.50 ] |
| 押圧力の合計 | | 0.41 | [ 0.40 , 0.44 ] | | [ 0.40 , 0.53 ] |

要する)」と判断することにした。そのため，締めたネジの長さを 15mm から残ったネジの長さを引いた数値とし，作業時間で割った値を求めた。締めたネジの長さを作業時間で割った値について，ドライバ操作技能「 1 」と判断された学生の最大値が 0.476mm/s であったため，0.476mm/s より小さい場合にはドライバ操作技能「2」と判断した。

ドライバ操作技能「4」と判断する学生は，ドライバ操作技能「3」と判断する学生に比べ，手際のよさだけでなく，「軸ずれは小さく，軸ぶれも少ない」ことも必要である。「軸ずれは小さく，軸ぶれも少ない」結果として，無駄な力を加えることもなく，押圧力の合計も小さいと判断した。そこで，「締めたネジの長さを作業時間で割った値」を押圧力の合計で割った値を求め，ドライバ操作技能「2」より大きいか否かで，ドライバ操作技能「3」か「4」であるかを判断することにした。その結果，「締めたネジの長さを作業時間で割った値」を押圧力の合計で割った値について，ドライバ操作技能「2」と判断された学生の最大値が 0.077mm/(s・kN) であったため，0.077mm/(s・kN) より大きい場合にはドライバ操作技能「3」と判断した。

また，ドライバ操作技能「4」,「3」である学生群は熟練群，ドライバ操作技能「2」,「1」である学生群は未熟練群と呼称していく。表 10.9 にドライバ操作における技能習熟度別の呼称と，分類ごとの人数をまとめた。

表 10.10 に，柄を持つ手の握力と，小指の指先から第 1 関節までの長さにおける平均と標準偏差を，群ごとにまとめた。表 10.10 より，熟達群が未熟練群に比べ，握力が大きく，小指の指先の長さも長いことが分かる。また，熟練群と未熟練群における指の長さおよび握力について，マンホイットニーの U 検定で比較した。その結果，小指の指先から第 1 関節までの長さと握力において，$p < 0.05$ となる，2 群間に違いも出た。つまり，熟練

表 10.8　技能習熟度別による手先の器用の分類

| 項目 | 残ったネジの長さ | 締めたネジの長さ／作業時間 | 締めたネジの長さ／作業時間・押圧力の合計 |
|---|---|---|---|
| 1 操作 (木ねじを締め込む・緩めて外す) を最後まで完了することができない。 | 1.5mm より長い | 0.476mm/s より小さい | 0.077mm/(s・kN) より小さい |
| 2 操作の際，軸ずれが大きく，あるいは軸ぶれが多く，かつ操作の手際が悪い (時間を要する)。 | 1.5mm 以下 | 0.476mm/s より小さい | 0.077mm/(s・kN) より小さい |
| 3 操作の際，軸ずれが大きく，あるいは軸ぶれが多く，または操作にもたつきがある。 | 1.5mm 以下 | 0.476mm/s 以上 | 0.077mm/(s・kN) より小さい |
| 4 操作の際，軸ずれは小さく，軸ぶれも少なく，かつ 操作にもたつきがなく手際がよい。 | 1.5mm 以下 | 0.476mm/s 以上 | 0.077mm/(s・kN) 以上 |

表 10.9　技能習熟度別の呼称

| 分類 | | 項目 | 人数 |
|---|---|---|---|
| 未到達群 | 未熟練群 | 1 操作 (木ねじを締め込む・緩めて外す) を最後まで完了することができない | 36 人 |
| 到達群 | 未熟練群 | 2 操作の際，軸すれが大きく，あるいは軸ぶれが多く，かつ操作の手際が悪い (時間を要する。) | 18 人 |
| 到達群 | 熟練群 | 3 操作の際，軸ずれが大きく，あるいは軸ぶれが多く，または操作にもたつきがある。 | 4 人 |
| 到達群 | 熟練群 | 1 操作の際，軸ずれが小さく，軸ぶれも少なく，かつ操作にもたつきがなく手際がよい | 7 人 |

群は未熟練群に比べ，小指の指先の長さが長く，握力の大きさが大きいと統計的にも言える。また，2 群間で $p < 0.05$ となり，握力に有意差があった組み合わせは，4 と 2，熟練群と 2，熟練群と 1 および，4 と未熟練群であった。

表 10.10　ドライバ操作技能ごとの指の長さと握力

| 分類 | 人数 | 柄を持つ手の握力 | 小指の指先から<br>第 1 関節までの長さ |
|---|---|---|---|
| 4 | 7 人 | 26.06kg<br>(5.23kg) | 15.86mm<br>(1.24mm) |
| 3 | 4 人 | 25.50kg<br>(6.65kg) | 16.53mm<br>(0.95mm) |
| 2 | 18 人 | 20.99kg<br>(3.29kg) | 15.55mm<br>(1.40mm) |
| 1 | 36 人 | 21.00kg<br>(5.09kg) | 15.48mm<br>(1.17mm) |
| 熟練群<br>(4 と 3) | 11 人 | 25.85kg<br>(5.79kg) | 16.11mm<br>(1.19mm) |
| 未熟練群<br>(2 と 1) | 54 人 | 21.00kg<br>(4.57kg) | 15.50mm<br>(1.25mm) |

熟練群と未熟練群における関節角度について，マンホイットニーの U 検定で比較した。その結果，柄を持つ手の関節角度において，中指・薬指・小指の第 1 関節の第 1〜3 四分位数と四分位範囲，および手首の第 1 角度の四分位範囲で，$p < 0.05$ となる，2 群間に違いが出た。つまり，熟練群は未熟練群に比べ，中指・薬指・小指の第 1 関節の角度と角度変化が小さく，手首の第 1 角度の角度変化も小さいとわかった。

ドライバ操作技能「4」と「3」における関節角度をマンホイットニーの U 検定で比較した。その結果，軸を持つ手の関節角度において，中指・小指の第 2,3 関節と薬指 1〜3 関節における四分位範囲で $p < 0.05$ となる，2 群間に違いが出た。つまり，ドライバ操作技能「4」の学生はドライバ操作技能「3」に比べ，中指・小指の第 2,3 関節と薬指 1〜3 関節における角度変化が小さいとわかった。

ドライバ操作技能「2」と「1」における関節角度をマンホイットニーの U 検定で比較した。その結果，柄を持つ手の関節角度において，人差し指の第 1,2 関節の第 2,3 四分位数および，薬指第 1 関節と小指第 1,2 関節の四分位数で $p < 0.05$ となる，2 群間に違いが出た。つまり，ドライバ操作技能の学生「2」はドライバ操作技能の学生「1」に比べ，人差し指第 1,2 関節の角度が小さく，薬指第 1 関節と小指第 1,2 関節の角度変化が大きいとわかった。

第章で行った事前アンケートの質問 4 について，ドライバ操作技能ごとに比較した。選

表 10.11　ドライバ操作技能ごとの使用時期の平均と標準偏差

| 分類 | ドライバ操作技能 | | | |
|---|---|---|---|---|
| | 4 | 3 | 2 | 1 |
| 合計 | 2.8 (2.5) | 3.0 (1.6) | 4.3 (1.5) | 2.7 (1.9) |
| 日常生活 | 2.2 (1.9) | 2.5 (1.1) | 3.2 (1.2) | 2.2 (1.5) |
| 学校生活 | 0.7 (0.7) | 0.5 (0.9) | 1.1 (1.1) | 0.5 (0.8) |

択肢の各使用時期を 1 とし，日常生活・学校生活をそれぞれ合計を 5 とした。表 10.11 にドライバ操作技能ごとの日常生活・学校生活における使用時期の平均と (　) 内に標準偏差を示す。表 10.11 より，ドライバ操作技能「2」の学生群が最もドライバを使用した時期を多く答えているとわかる。学生群の間でマンホイットニーの U 検定を行った。その結果，ドライバ操作技能「2」と「1」の学生群の間において，合計では $p < 0.01$，日常生活では $p < 0.05$ となり，有意差があるとわかった。

### 10.4.5　事後アンケートの結果

図 10.3 に事後アンケートを示す。質問 2 の回答結果，質問 3 と質問 4 をまとめ分析した結果，質問 5，7 の回答結果の順に述べていく。

質問 2 の回答結果を表 10.12 に示す。表 10.12 より，前群と後群ではともに，肯定的な回答「5,4」を選択した割合が 85% 以上であるとわかった。また，ドライバ操作技能での分類ごとにマンホイットニーの U 検定で比較した。その結果，ドライバ操作技能「4」と「3」の間で $p < 0.05$，ドライバ操作技能「3」と「1」の間で $p < 0.01$，ドライバ操作技能「2」と「1」の間で $p < 0.05$ 到達群（ドライバ操作技能「4，3，2」）と未到達群（ドライバ操作技能「1」）の間で $p < 0.01$ となり，2 群間に違いが出た。なお，質問 4 の肯定的回答は前群が 29 人 (91%)，後群が 30 人 (91%)，質問 6 の肯定的回答は前群が 17 人 (54%)，後群が 22 人 (67%) となり，各質問において，どの分類でもマンホイットニーの U 検定で有意差が出なかった。

前群，後群，前群と後群を合計した全体ごとに，質問 3 の項目 1〜10 と質問 4 の回答をもとに CS 分析した結果を図 10.4 に示す。図 10.4 では質問 3 の各項目での肯定的回答の割合である満足率と，質問 3 の各項目と質問 4(総合評価) との相関係数である重要率をもとに，それぞれを偏差値に変換した値となる満足度，重要度を算出し，プロットした。また，質問 3 の各項目の中で改善すべき項目をより明確に判定するため，改善すべき優先順位に応じて得点化した値「改善度」も算出し，改善度が高い項目を優先的に改善すること

クラス： 藤組 ・ 菊組 番号：

**質問 1 中学 2 年生になってから，どのような機会にドライバを使いましたか。（あてはまるもの，すべてに○）**

【 ①電池を交換する ②家具を組み立てる ③機械を分解する ④家具を修理する
⑤学校以外で使わなかった ⑥その他（ ）】

**質問 2 ドライバを使う練習で，これまでよりドライバを使えるようになりましたか。（○は 1 つ）**

【 ①とてもなった ②なった ③どちらともいえない ④あまりならない ⑤全くならない 】

**質問 3 ドライバを使うポイント・注意点は，ドライバを使う参考になりましたか。（○は 1 つずつ）**

| | とても なった | なった | どちらとも いえない | あまり ならない | 全く ならない |
|---|---|---|---|---|---|
| 1. ネジをゆるめる際にも手首とすべての指が動かしやすい手の形で，ドライバがぐらつかないようネジを押し，回す。 | ① | ② | ③ | ④ | ⑤ |
| 2. 両手で行う。 | ① | ② | ③ | ④ | ⑤ |
| 3. 柄を利き手で上から被せるように握る。 | ① | ② | ③ | ④ | ⑤ |
| 4. 手首と 5 本の指全てで回す力を加える。 | ① | ② | ③ | ④ | ⑤ |
| 5. ネジがずれない程度の押す力を加える。 | ① | ② | ③ | ④ | ⑤ |
| 6. 親指を規則的に動かす。 | ① | ② | ③ | ④ | ⑤ |
| 7. 支え手は軸が真っ直ぐ（垂直に）ぐらつかないようにドライバを支える。 | ① | ② | ③ | ④ | ⑤ |
| 8. 利き手の人差し指が伸びていませんか？ | ① | ② | ③ | ④ | ⑤ |
| 9. 利き手中指・薬指・小指が握ったままになっていませんか？ | ① | ② | ③ | ④ | ⑤ |
| 10. 支え手は軸を握りすぎていませんか？ | ① | ② | ③ | ④ | ⑤ |

**質問 4 質問 3 のドライバを使うポイント・注意点は総合的に見て，ドライバを使う参考になりましたか。（○は 1 つ）**

【 ①とてもなった ②なった ③どちらともいえない ④あまりならない ⑤全くならない 】

**質問 5 質問 4 の理由を書いてください。**

**質問 6 タブレットを使ってみて，ドライバを使う参考になりましたか。（○は 1 つ）**

【 ①とてもなった ②なった ③どちらともいえない ④あまりならない ⑤全くならない 】

**質問 7 タブレットを使ってみて，どんな点が良かったか書いてください。**

図 10.3 事後アンケートの問題用紙

表 10.12　質問 2 の回答結果

| 分類 | とても なった | なった | どちらとも いえない | あまり ならない | 全く ならない |
|---|---|---|---|---|---|
| 前群 32 人 | 6 人 (19%) | 22 人 (69%) | 3 人 (9%) | 1 人 (3%) | 0 人 |
| 後群 33 人 | 6 人 (18%) | 22 人 (67%) | 4 人 (12%) | 1 人 (3%) | 0 人 |
| 操作技能 4 7 人 | 0 人 | 7 人 (100%) | 0 人 | 0 人 | 0 人 |
| 操作技能 3 4 人 | 3 人 (75%) | 1 人 (25%) | 0 人 | 0 人 | 0 人 |
| 操作技能 2 18 人 | 7 人 (39%) | 9 人 (50%) | 1 人 (6%) | 1 人 (6%) | 0 人 |
| 操作技能 1 36 人 | 2 人 (6%) | 27 人 (75%) | 6 人 (17%) | 1 人 (3%) | 0 人 |

により，効果が大きく，改善度がマイナスの項目は改善を必要としない。「質問 3 の項目（重要度，満足度，改善度）」を図 10.4 にまとめ，改善度が高い上位 3 つに色をつけた。

図 10.4 より，維持分野（重要度が 50 より小さく，満足度が 50 より大きい領域）に入る質問 3 の項目は群によらず，項目 1,2,5,7 であるとわかった。また，重点維持分野（重要度が 50 より大きく，満足度が 50 より大きい領域）に入る質問 3 の項目は全体が項目 4，前群が 3，後群が 4 と 6 であるとわかった。そして，重点改善分野（重要度が 50 より小さく，満足度が 50 より小さい領域）に入る質問 3 の項目上位 3 つは，全体が項目 8,9,10，前群が項目 6,8,10，後群が項目 3,9,10 であるとわかった。

また，ドライバ操作技能ごとに重要度，満足度，改善度をまとめた結果を表 10.13，10.14 に示す。また，表 10.13，10.14 を比較しやすくするため，ドライバ操作技能ごとに質問 3 の各項目が重点維持分野，維持分野，重点改善分野であるかを表 10.15 に示す。表 10.13 〜10.15 より，ドライバ操作技能によらず，質問 3 の項目 1，2，7 が重点維持分野か維持分野，項目 9 が重点改善分野であるとわかった。また，質問 3 の項目 3 について，ドライバ操作技能「4」,「3」,「2」では重点維持分野になるが，ドライバ操作技能「1」では重点改善分野になるとわかった。

質問 5 の回答結果を表 10.16 に示す。表 10.16 より，質問 4 の選択肢を選んだ理由として，「ドライバ操作技能向上」を記述した生徒の数が 29 人，「問題点がわかる」ことを記述した生徒の数が 8 人，「活用できなかった点」を記述した生徒の数が 8 人いるとわかった。

質問 7 の回答結果を表 10.17 に示す。表 10.17 より，質問 6 の選択肢を選んだ理由として，「自分の動きを客観的に見る」ことを記述した生徒が 37 人，「悪い点や改善点がわか

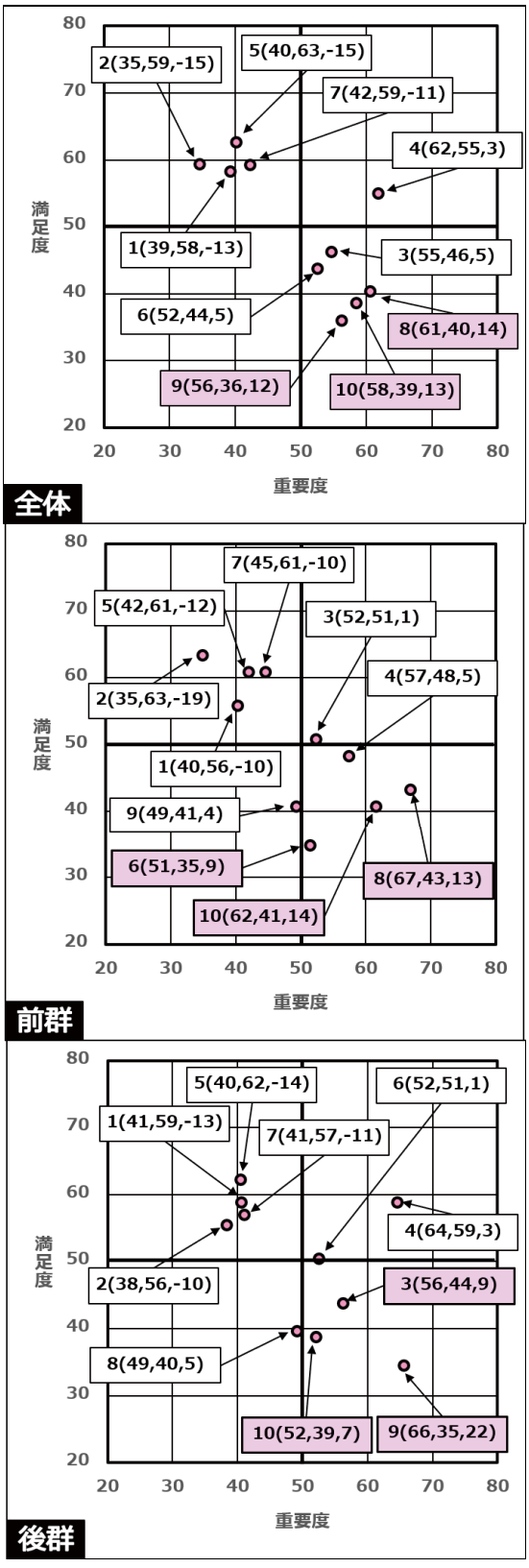


図 10.4　全体，前群，後群における CS 分析の結果

表 10.13　ドライバ操作技能ごとの重要度，満足度，改善度

| 分類 | 操作技能「4」　7 人 | | | 操作技能「3」　4 人 | | |
|---|---|---|---|---|---|---|
| 質問 3 | 重要度 | 満足度 | 改善度 | 重要度 | 満足度 | 改善度 |
| 1 | 41 | 64 | -15 | 57 | 59 | -1 |
| 2 | 35 | 64 | -21 | 32 | 59 | -16 |
| 3 | 52 | 54 | -1 | 57 | 59 | -1 |
| 4 | 58 | 44 | 9 | 51 | 44 | 4 |
| 5 | 37 | 54 | -9 | 51 | 44 | 4 |
| 6 | 59 | 44 | 10 | 51 | 44 | 4 |
| 7 | 43 | 54 | -7 | 57 | 59 | -1 |
| 8 | 62 | 44 | 11 | 32 | 44 | -6 |
| 9 | 54 | 33 | 11 | 57 | 59 | -1 |
| 10 | 59 | 44 | 10 | 57 | 30 | 15 |

表 10.14　ドライバ操作技能「2」と「1」の重要度，満足度，改善度

| 分類 | 操作技能「2」　18 人 | | | 操作技能「1」　36 人 | | |
|---|---|---|---|---|---|---|
| 質問 3 | 重要度 | 満足度 | 改善度 | 重要度 | 満足度 | 改善度 |
| 1 | 55 | 54 | 0 | 36 | 58 | -13 |
| 2 | 63 | 58 | 2 | 34 | 58 | -14 |
| 3 | 57 | 50 | 3 | 56 | 43 | 8 |
| 4 | 49 | 58 | -5 | 61 | 56 | 2 |
| 5 | 28 | 62 | -20 | 48 | 65 | -9 |
| 6 | 49 | 50 | -1 | 47 | 42 | 2 |
| 7 | 49 | 58 | -4 | 44 | 59 | -10 |
| 8 | 45 | 42 | 1 | 63 | 40 | 15 |
| 9 | 61 | 31 | 18 | 55 | 38 | 9 |
| 10 | 45 | 38 | 3 | 56 | 41 | 10 |

る」ことを記述した生徒が 18 人いるとわかった。

## 10.5　理解度に関する評価試験

被験者は技能指導を受け，どれくらい覚えているか確かめるため，理解度に関する評価試験を行った。評価試験は実技指導の最終日から 2 か月が経過した段階で行った。7 月 8 日（木）に評価試験を実施した。参加者は藤組が 33 名，菊組が 32 名であり，藤組と菊組では 1 名ずつ欠席した。評価試験は技術科の定期テストで行った。問題用紙は大問 12 問あり，本評価試験に解答する学生の人数をふやすため，最初の大問に評価試験の問題を入れた。定期テストは事前に教科書やファイルを 10 分間確認した後，40 分の試験を行っ

表 10.15　ドライバ操作技能ごとの分野

| 質問<br>3 | 操作技能「4」<br>7 人 | 操作技能「3」<br>4 人 | 操作技能「2」<br>18 人 | 操作技能「1」<br>36 人 |
|---|---|---|---|---|
| 1 | 維持分野 | 重点維持分野 | 重点維持分野 | 維持分野 |
| 2 | 維持分野 | 維持分野 | 重点維持分野 | 維持分野 |
| 3 | 重点維持分野 | 重点維持分野 | 重点維持分野 | 重点改善分野 |
| 4 | 重点維持分野 | | 維持分野 | 重点維持分野 |
| 5 | 維持分野 | | 維持分野 | 維持分野 |
| 6 | | | 維持分野 | |
| 7 | 維持分野 | 重点維持分野 | 維持分野 | 維持分野 |
| 8 | 重点改善分野 | 維持分野 | | 重点改善分野 |
| 9 | 重点改善分野 | 重点維持分野 | 重点改善分野 | 重点改善分野 |
| 10 | 重点改善分野 | 重点改善分野 | | 重点改善分野 |

た。図 10.5 に問題を示す。図 10.5 は 2 問の記述式となっている。図 10.5 の (あ) では，ドライバ操作時の要点（特に，注意点）が身についたかを確かめるために作問した。図 10.5 の (い) では，ドライバ操作時の要点（特に，ポイント）が身についたか，また，それ以外に学習しているものがあるかを確かめるために作問した。

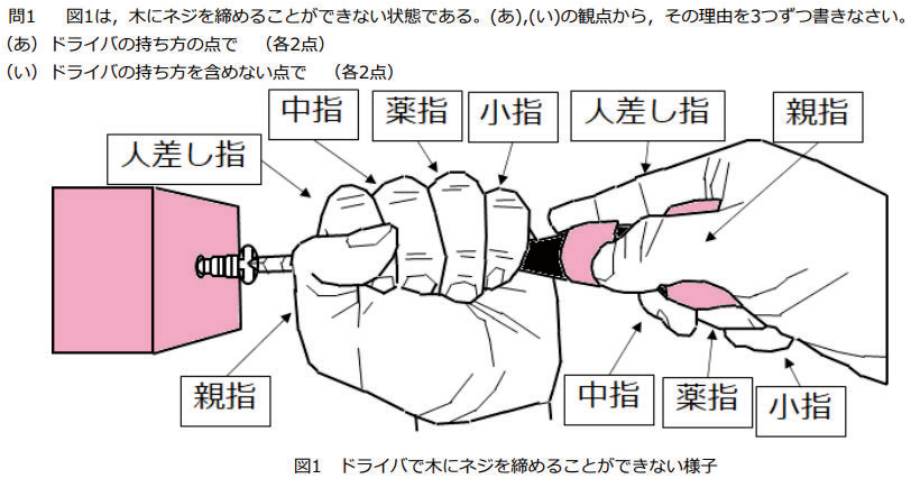


図 10.5　問題

表 10.18 に，(あ) の問題での採点結果と，(い) の問題での採点結果および，(あ) と (い) の問題を合計した採点結果について，ドライバ操作技能ごとに平均と標準偏差にまとめた。

表 10.18 より，(あ) の問題では，熟練群が未熟練群より点数が高いことがわかる。しかし，(い) の問題では，熟練群と未熟練群との間で点数に差がないことがわかる。

表 10.16　質問 5 の回答結果

| 分類 | テキストコード及び記述内容 | 件数 |
|---|---|---|
| ドライバ操作技能向上 | 上手・速く・回す・できる・締める・上達・短い・記録：「最初より上手く使えるようになった」,「書いてあったポイントを見ながらやったら速く回せるようになったから」,「そのポイントを参考を押さえたら，簡単にできたから」,「項目通りにやったらネジを締めやすくなったから」,「上達を感じられたから」,「秒数で比較すると分かりやすいが，10 秒ずつくらいどんどんはめる時間が短くなっているから」,「その方法でやってみたら記録が上がったから」 | 29 |
| 問題点がわかる | 理由・その他：「自分が出来ていない理由が分かったから」,「自分が上手くいかない理由や原因を解決することができたから」,「私が元々出来ていた物もあったが，私ができていない所に気付けたから」,「ポイント，注意点を教えてもらう前にやったので，こうすればよかったんだと自分では見つける事ができないことが知れたから」,「ネジの方を支えながらいつも回してなかったから」,「使い方や締め方を知らずにやっても上手くできないから」,「今まで使い方を知らずに手探りでやっていたので参考になったことが多くあった」 | 8 |
| 活用できなかった点 | できない・(参考に）なる・その他：「意識することはできたけれど，できなかったこともあったから」,「参考にできた部分とあまりできなかった部分とがあったから」,「両手でしっかりと握りこんだ方がとても回しやすかったから。手の持ち方などは本番であまりちゃんと意識出来なかった。気を付けたい」,「(参考に）なった物もあれば，ならなかった物もあったから」,「ポイントなどを気にしてるとうまく回らなくなる」,「活かせていた部分もあるし，活かせなかった部分もあるから」 | 8 |

表 10.18 に合わせ，群ごとにマンホイットニーの U 検定で比較した。その結果，熟練群と未熟練群との間で，(あ）の問題において $p < 0.05$ となり，有意差が生じた。つまり，熟練群は未熟練群に比べ，得点が高いと統計的にも言える。なお，（い）の問題においては，有意差がでなかった。

またドライバ操作技能「4」の学生は全員，（あ）の問題において，図 9.1 の注意点 No.1 にある人差し指を伸ばしていることに触れていた。

表 10.17　質問 7 の回答結果

| 分類 | テキストコード及び記述内容 | 件数 |
|---|---|---|
| 自分の動きを客観的に見る | 分かる・見る・確認・知る・振り返り・見直す・その他：「自分がどうやってドライバを使っているかなどが，動画で分かりやすく見ることができた」,「自分の手の動きを確認できた」,「自分がどのようにドライバを使っているのかを知ることができた」,「自分の振り返りができる」,「自分のねじの使い方を見直せた」,「自分がどのようにやっているか」 | 37 |
| 悪い点や改善点がわかる | 改善・悪い・だめ・できない・反省・よくない：「自分の操作方法を別の視点から見ることができたので，改善点につながった」,「みんなで良い点，悪い点を指摘する際に便利だった」,「カメラを使って自分のだめな所を確認できた」,「自分がどんな感じで回していて，できていること，できていない事が分かりやすかった」,「反省するとこなど詳しく知れた」,「自分の回している姿を見ることで，どこがよくて，どこがよくなかったのかを見てその場でなおすことができた」 | 18 |

## 10.6　考察

試験 2 を行う前では握力の最低値が異群の条件で 21.8kg，同群の条件で 14.0kg 必要であったが，異群の条件で 12〜15.3kg の握力で 1.5mm の長さまでネジを締める事ができるようになった。しかし握力が 30.8kg でもネジを締める事ができなかった事例もあり，身体動作やネジを回す力等と合わせ分析する必要がある。

表 8 より，1 回目の計測では前群内の同群と異群との間でネジの長さに有意差があり，練習前に試験 1 での操作体験があった事が学習効果につながったと考えられる。2 回目の計測では後群での残ったネジの長さが長くなった。これは，自己流のドライバ操作を改善するのが難しかったり，練習 (表 6 の No.3) から 2 週間空いたりしたため，前群に比べ，図 9.1 の指導内容が定着しにくかったと考えられる。そのため，異群の条件で一度ドライバを操作し，指導初期段階にドライバの握り方 (図 9.1) を伝えドライバの練習を繰り返した時，ドライバ操作技能が向上しやすいと考えられる。

到達群は握力と柄を持つ手指の関節角度との関係から握力のない中学生が図 9.1 のポイントの No.3,5 を満たし，1.5mm の長さまでネジを締める事ができたとわかる。その結果，押圧力の合計や人差し指の角度も未達群より小さく，図 9.1 のポイントの No.4 や注意点の No.1 も満たしていると捉えた。軸を持つ手では，関節角度や手首の角度の散布度

表 10.18　テストの結果

| 分類 | 人数 | 問題 | | |
|---|---|---|---|---|
| | | (あ)<br>6 点満点 | (い)<br>6 点満点 | 合計<br>12 点満点 |
| 4 | 7 人 | 4.29 点<br>(1.28 点) | 2.29 点<br>(2.25 点) | 6.57 点<br>(2.44 点) |
| 3 | 4 人 | 3.75 点<br>(1.79 点) | 2.50 点<br>(1.66 点) | 6.25 点<br>(2.86 点) |
| 2 | 18 人 | 2.83 点<br>(1.92 点) | 2.33 点<br>(1.94 点) | 5.17 点<br>(2.97 点) |
| 1 | 36 人 | 2.92 点<br>(1.85 点) | 2.36 点<br>(2.15 点) | 5.28 点<br>(3.00 点) |
| 熟練群<br>(4 と 3) | 11 人 | 4.09 点<br>(1.50 点) | 2.36 点<br>(2.06 点) | 6.45 点<br>(2.61 点) |
| 未熟練群<br>(2 と 1) | 54 人 | 2.89 点<br>(1.87 点) | 2.35 点<br>(2.08 点) | 5.24 点<br>(2.99 点) |

が押圧力の最大値と正の相関にあり，支え手は軸をどう握るかより，軸を固定しやすい定位置を見つけ，軸ブレと過重負荷を防ぐのがよいと考えられる。

ドライバ操作技能ごとに，手指の関節角度や知識を比較した結果，ドライバ操作技能「4」である学生には軸を持つ手指の関節角度において，無駄な動きがなかった。また，未熟練群の中でドライバ操作技能「2」と「1」の学生を比較した結果，ネジを締めることが出来るか否かは，柄を持つ手で人差し指がしっかり柄を固定していたり，薬指と小指も一緒に回していたりすることに違いが生じた。また人差し指で柄を固定するか否かといった知識は，ドライバ操作技能「4」の学生全員が身につけていることも確認できた。また，ドライバ操作技能の習熟度別の学生を比較した時，握力が大きいことや指先の長さが長いことが重要であるとわかった。

# 終章　研究の成果と今後の課題

本研究の目的は，普通教育としての技術教育に必要な手先の器用を明らかにすることである。手先の器用を明らかにするために設定した，第 1 章で示した 4 つの課題（課題 1～課題 4）とその検証結果についてそれぞれ述べていく。

課題 1「手先の器用がどう捉えられているか，どうあるべきかを明らかにし，普通教育としての技術教育において，生徒に獲得させたい手先の器用を定義する」では，普通教育としての技術教育において，生徒に獲得させたい手先の器用を定義するため，手先の器用がどう捉えられているか，どうあるべきかを明らかにした。手先の器用を明らかにするため，辞典や事典，文献にある定義や説明を確認した。その結果，手先の器用には，「速く」「細かく」「正確な」手指を使った動きを意味する「巧緻性」と，「新しい技能習得しやすい性質・才能」といった特徴が代表的なイメージであるとわかった。また，先行研究を確認した結果，手先が器用であるか否かといった自認には，中学校技術・家庭（技術分野）の授業が影響するとわかった。しかし，1990 年代から 2010 年代までの中学校技術・家庭（技術分野）の授業について，家庭でものづくりを経験してこなかった生徒が，実際よりも高い目標の技能を求め，道具を操作する活動が思ったようにできないと感じ，保護者も生徒に器用や道具操作の習熟が身に付いていないと捉えているとわかった。一方，中学校学習指導要領の解説書を確認した結果，技術科教育で習得させたい技能として，学習の初期段階では技能を一つ習得させ，習得した技能を自身の経験や他の技能と関連付け，変化する状況や課題に応じて活用できる技能を習熟させることを求めているとわかった。そのため，普通教育としての技術教育において，生徒に獲得させたい手先の器用は 2 つあり，道具操作のための巧緻性（「速く」「細かく」「正確な」手指を使った動き」）と，「新しい道具操作技能を習得しやすい資質・能力」と定義した。また，この 2 つには学習する順番があり，まず学習の初期段階に道具操作のための巧緻性を習得し，その後，習得した道具操作のための巧緻性をもとに，新しい道具操作に関連させたり，技能習得しやすかったりする資質・能力を養うといった形になる。そのため，課題 2 以降の本研究では，道具操作のための巧緻性（「速く」「細かく」「正確な」手指を使った動き）について，中学生における動きの違いを分析したり，獲得できるかを確認したりしていった。

課題 2「普通教育としての技術教育で生徒に獲得させたい手先の器用や，技術教育で扱われる道具をもとに，手先の器用における手指動作の違いが確認しやすい道具を選定する」では，道具操作のための巧緻性における手指動作の違いが確認しやすい道具を選定するため，「技術・家庭（技術分野）の授業で使用する道具」，「日常生活で使用する道具」，『「速く」「細かく」「正確な」手指を使った動きが求められる道具』，「新しい道具に対応できるよう，操作原理が他の道具の操作でも転移しそうな道具」といった条件にあう道具を明らかにした。「技術・家庭（技術分野）の授業で使用する道具」を明らかにするため，教科書の索引にある道具・機械を整理した。その結果，さしがね，卓上ボール盤，ねじ回し（ドライバ）は調査対象の教科書全てで索引に登場するとわかった。「日常生活で使用する道具」を明らかにするため，日常生活で使用されているかを調べた先行研究を確認した。その結果，ドライバは時代によらず，使用経験の割合が大きい道具であるとわかった。また，ドライバの使用経験が少なければ，ドライバ操作技能が小さくなったり，ドライバ操作技能を 4 段階に分類した評価指標があったりしたため，中学生がドライバ操作時の手指動作を分析しやすいと考えられる。『「速く」「細かく」「正確な」手指を使った動きが求められる道具』を明らかにするため，「速く」「細かく」「正確な」手指を使った動きが，身体全体の動きを使った粗大運動に比べ，道具操作に必要となる道具を検討した。これまでの先行研究では，のこぎりやかんなといった木工具の操作技能を分析しているが，これらの木工具が身体全体の動きや腕の動きが重要であり，手指の細かい動きより握り方が求められる。そのため，これらの木工具は不向きと判断した。「新しい道具に対応できるよう，操作原理が他の道具の操作でも転移しそうな道具」を明らかにするため，どの操作原理であれば転移しそうかを検討した。その結果，小学 6 年生の理科で扱われる「てこの原理」は先行研究において，生活物の仕組みや身体の動かし方への転移につながっていることが確認できたため，てこの原理がが作用するドライバも適していると考えた。以上により，道具操作のための巧緻性における手指動作の違いが確認しやすい道具として，ドライバを選定した。そのため，課題 3 以降の本研究では，ドライバ操作時の手指動作を計測し，分析した。

課題 3「発達段階と作業条件の違いにより，選定した道具を操作した時，手指動作にどういった違いがあるかを確かめる」では，ドライバ操作時の手指動作における特徴を明らかにするため，発達段階と作業条件がそれぞれ違う時のドライバ操作時の手指動作を計測し，分析を行った。発達段階の違いによるドライバ操作時の手指動作の違いを明らかにするため，簡易手指動作分析システムを用い，手指の巧緻性が異なる，大学生群と中学生群がドライバ操作する時の手指動作を計測した。計測結果より，年齢・操作方法によらず，手首の回転が速い学生は，ネジを締めるのが速い傾向にあるとわかった。また，大学生が中学生に比べ，親指の関節各部位の角度が一緒に増減しているとわかった。作業条件の違いによるドライバ操作時の手指動作の違いを明らかにするため，簡易手指動作分析システ

ムと押圧力測定装置を用い，ネジが外れやすい中学生群とネジが外れにくい中学生群がドライバ操作する時の手指動作と押圧力を計測した。計測結果より，ネジの外れやすさによらず，押圧力が小さいほど，最終的な押圧力の合計が小さい傾向にあるとわかった。また，ネジが外れやすい条件では押圧力の合計が小さいほど，締めたネジの長さが長くなり，ネジが外れにくい条件では，押圧力の最大値が大きいほど，締めたネジの長さが長くなる傾向にあるとわかった。そして，握力が大きい中学生がドライバの柄を 5 本の指でしっかり握り，ドライバの軸をもつ手指の動きが小さくなり，ネジを締めきることができる傾向にあるとわかった。以上により，ドライバ操作時の手指動作における特徴として，ネジを速く締めきるためには，「手全体（手首）で回す」，「親指の関節各部位の動きが連動する」，「ドライバの柄を 5 本の指でしっかり握り，ドライバの軸をもつ手指の動きが小さい」ことが求められると言える。また，ネジを速く締めきるためには，押圧力が必要最低限であることも求められると言える。そこで，課題 4 では，ドライバ操作時の手指動作における特徴や，ネジを速く締めきるための特徴をドライバ操作の技能指導で活用した。

課題 4「道具操作時の手指動作の特徴をもとに，道具操作の熟達化を促し，中学生が獲得した手先の器用ごとに，その違いを分析する。」では，中学生が獲得したドライバ操作時の手指動作を分析するため，ドライバ操作時の手指動作における特徴を指導内容に含め，ドライバ操作の技能指導を行った。ドライバ操作の技能指導により，中学生がネジを速く締めきるためのドライバ操作時の手指の形・動作として，次の特徴にまとめることができる。

1. 握力が大きく，指先が長い。
2. 5 本の指すべてで，道具をしっかりと固定する。
3. 手指の動きに無駄な動きがない。

要点 1 は，学生個人の個性（身体的特性）であるため，普通教育としての技術教育で求めるのが難しい。要点 2 は，手指をどんなに速く，細かく，正確に動かそうと考えても，まずは，5 本の指すべてで道具をしっかりと固定することから始める必要性を確認することができる。要点 3 は，道具操作の技能を習得していく中で最適化されていく必要がある。そして，この最適化されていく過程を習得することは，その後新しい道具操作の技能習得をしていく上でも必要な資質・能力ではないかと考える。

以上より，普通教育としての技術教育で必要な手先の器用として，まず，速く細かく正確な手指の動きを身につけるため，5 本の指すべてで道具を固定すること，および道具操作時に無駄な動きがなくなるよう最適化されていく学習過程を身につける必要があると考える。これは，身体全体の動きや腕の動きが重要であった木工具（のこぎりやかんな）においても，手指の細かい動きというより握り方が求められたり，熟練者が初心者に比べ，無駄な動きがないという先行研究 [48] に通じたりする結果である。したがって，ドライバ

操作時の手指動作だけでなく，他の道具にも通用しそうな要点ではないかと考える。

研究の成果をもとに，学習の転移が起こりうるのかを明らかにしていきたい。つまり，新しい道具の操作技能習得に役だったり，設計・計画・創造する活動の精度を高めたりすることにつながるかを確かめていきたい。

# 謝辞

研究を進めるにあたり，諸先生には多くの助言やご指導を頂き，深く感謝いたします。

愛知教育大学・静岡大学大学院教育学研究科共同教科開発学専攻教授の紅林秀治先生には，主指導教員としてご指導いただきました。教育学部 3 年次から修士課程 2 年次までの 4 年間，博士課程の 3 年間の計 7 年間にわたり，ゼミ報告の書き方といった研究の基礎から始まり，博士論文の書き方に至るまで，一から丁寧にご指導をいただきました。また，中学校教員になってからも本研究に関わる実践についてご助言いただき，私の 20 代は先生のご指導，ご助言のおかげで有意義になったと感じております。親身にご指導いただき，深く感謝いたします。

同専攻教授の新保先生，同専攻教授の飯島先生には，副指導教員としてご指導いただきました。新保先生には，博士論文の内容および研究の方法について体育科の観点からご指導，ご助言をいただいただけでなく，新保ジウムへ参加させていただけたことで，私自身の研究を幅広い視点から見直すきっかけとなり，今日に至るまで私にとって大きなモチベーションとなりました。心より御礼申し上げます。飯島先生には，博士論文の構想および論文の書き方について的確なご指導，ご助言をいただきました。このことは，研究の方針を立て，本博士論文を執筆する上で，大変参考となり，私自身の研究視点を見直すきっかけとなりました。心より御礼申し上げます。

この他，愛知教育大学・静岡大学大学院教育学研究科共同教科開発学専攻教授の熊倉啓之先生，同専攻教授の村上陽子先生，同専攻教授の村越真先生，同専攻准教授の郡司賀透先生をはじめ，愛知教育大学・静岡大学大学院教育学研究科共同教科開発学専攻の先生方には，授業においてそれぞれの専門的知見からご指導いただき，教科開発学セミナーの発表では本発表について様々な研究分野の視点からご指摘，ご助言をいただきました。こうしたご指摘，ご助言により，本研究の内容を精査することができました。本当にありがとうございました。

本博士論文を執筆するにあたり，調査にご協力いただきました S 県 S 学園中学校・高等学校の生徒の皆様と先生方，静岡大学教育学部学校教育教員養成課程教科教育学専攻の技術教育専修の卒業生の皆様，家庭科教育専修の卒業生の皆様，静岡大学大学院教育学研究

科の生徒の皆様，卒業生の皆様に改めて感謝申し上げます。

これまで研究を行う上で，共に支えあい，時に励ましやご助言をいただきました愛知教育大学・静岡大学大学院教育学研究科共同教科開発学専攻の学生の皆様，紅林研究室の先輩，同期，後輩の皆様に感謝申し上げます。

最後に，大学進学時から今日まであたたかい励ましをいつも送り続けてくれた家族に心から感謝いたします。本当にありがとうございました.

# 参考文献


[1] 志民一成・耳塚日香里・大石幸史・他 2 名：モーションキャプチャを活用した指揮法指導の可能性: KINECT センサーを用いた簡易動作分析システムを利用して，静岡大学教育学部研究報告（教科教育学篇），第 47 号，pp.131-144(2016)

[2] 薄井洋子：舞踊の学びに対するモーションキャプチャ活用，博士論文，http://hdl,handle.net/10097/60374(閲覧日：2019 年 11 月 12 日)

[3] 寺内定夫・須藤敏昭・森下一期・その他 4 名：子どもの遊びと手の労働，あすなろ書房

[4] 持丸正明・山中龍宏・西田佳史・他 1 名：子供計測ハンドブック，朝倉書店，p.198 (2013)

[5] 坪井章雄・門間正彦・河野豊・他 4 名：健常者における手指巧緻動作と認知機能の関連，厚生の指標，第 60 巻，第 1 号，pp.10-16 (2013)

[6] 日本産業技術教育学会：一般社団法人　日本産業技術教育学会，https://www.jste.jp/main/about.html(閲覧日：2021 年 9 月 8 日)

[7] 大橋和正：人の発達段階を考慮したものづくり教育の体系化，風間書房 (2015)

[8] 紅林秀治：設計を中核とした技術教育の提案，教科開発学論集，第 3 号，pp.151-158(2015)

[9] 紅林秀治：ものづくりをシステムづくりと捉え直す技術教育の検討，教科開発学論集，第 4 号，pp.143-150(2016)

[10] 青木麟太郎・紅林秀治：中等教育プログラムに整合した LED ランタン製作教材の開発の試み，静岡大学教育学部研究報告 (教科教育学篇)，第 52 巻，pp.83-98(2020)

[11] 竹野英敏・塩入睦夫・安東茂樹・その他 116 名：技術・家庭【技術分野】　テクノロジーに希望をのせて，開隆堂，pp.28,288,289(2021)

[12] 中村祐治・古川稔・太田達郎・その他 45 名：New　技術・家庭　技術分野　明日を創造する，教育図書，pp.28,286,287(2021)

[13] 田口浩継・佐藤文子・志村結実・その他 77 名：新しい技術・家庭　技術分野　未来を創る　Technology，東京書籍，pp.33,290,291(2021)

[14] 文部科学省小学校段階における論理的思考力や創造性，問題解決能力等の育成とプログラミング教育に関する有識者会議：小学校段階におけるプログラミング教育の在り方について（議論の取りまとめ）関連（2016），http：//www.mext.go.jp/b_menu/shingi/chousa/shotou/122/index.htm(閲覧日：2021 年 9 月 8 日)

[15] 紅林秀治：小中学校における¡普通教育としてのプログラミング教育¿の展開と課題，情報処理，第 60 巻，第 3 号，pp.248-252(2019)

[16] 大村基将・紅林秀治：ソフトウェア設計に基づく初学者のためのプログラミング学習の家庭に関する考察，教科開発学論集，第 4 号，pp.151-159(2016)

[17] 松秀樹・難波宏司：プログラミング教育の意義に関する研究，園田学園女子大学論文集，第 52 巻，pp.49-60(2018)

[18] 福井昌則・黒田昌克・森山潤：ゲーム・パズルを題材に高校生の創造的態度の育成を図るプログラミング教育の試み，日本教育工学論文誌，第 42 巻，pp.21-24(2018)

[19] 土井康作：技術教育における作業段取りの教育的効果，風間書房 (2004)

[20] 日本産業技術教育学会：会誌「日本産業技術教育学会誌」 ISSN 2434-6101，https://www.jste.jp/main/kaishi/1958_Current.html（閲覧日：2021 年 11 月 15 日）

[21] 中里真之：木材加工の基本工作法について (1) 子供の手鋸による切削の精密さについての調査、日本産業技術教育学会誌，第 16 巻，第 1 号，pp.33-35(1974)

[22] 中里真之：木材加工の基本工作法について (2) スコヤと錐による工作精度の調査－、日本産業技術教育学会誌，第 17 巻，第 1 号，pp.94-96(1974)

[23] 山下晃功・土山球一：手鉋による平削りの学習における実験授業，日本産業技術教育学会誌，第 18 巻，第 1 号，pp.139-143(1976)

[24] 北沢啓：指導内容の基礎化に関する研究 2 －手工具指導の適時性について－，日本産業技術教育学会誌，第 18 巻，第 1 号，pp.220-224(1976)

[25] 城仁士：技能の習熟に関する実験的研究－木材加工（ほぞ加工）技能の多段階的形成の試み—，日本産業技術教育学会誌，第 20 巻，第 2 号，pp.125-129(1978)

[26] 末富正啓：線引き作業による巧緻性の研究 (1)，日本産業技術教育学会誌，第 20 巻，第 2 巻，pp.157-160(1978)

[27] 向山玉雄：技能の習得を早める方法に関する研究，日本産業技術教育学会誌，第 28 巻，第 3 号，pp.24-31(1986)

[28] 向山玉雄・林俊郎：ノコギリ引き作業における技能の習得過程，日本産業技術教育学会誌，第 29 巻，第 3 号，pp.57-64(1987)

[29] 村田昭治・橘田紘洋：巧緻性の発達と技術教育の方法 (2) -のこびきについて-，日本産業技術教育学会誌，第 30 巻，第 1 号，pp.23-27 (1988)

[30] 藤木卓：中学生の 1 本指入力によるキーボード操作技能に関する考察，日本産業技

術教育学会誌，第 31 巻，第 1 号，pp.53-56 (1989)
[31] 山口晴久：技術・家庭科教育におけるハンダ付け指導法について，日本産業技術教育学会誌，第 31 巻，第 1 号，pp.57-60 (1989)
[32] 松原信一・杉山滋：釘打ち技能の測定と評価，日本産業技術教育学会誌，第 32 巻，第 3 号，pp.181-189 (1990)
[33] 江馬諭・山名忠・久保務・その他 1 名：くぎ打ち作業に関する基礎研究，日本産業技術教育学会誌，第 36 巻，第 4 号，pp.260-268 (1994)
[34] 江馬諭・尾高広昭・猿渡彰：中学生のキーボード操作に関する研究，日本産業技術教育学会誌，第 38 巻，第 4 号，pp.231-239 (1996)
[35] 有川誠：ニッパ操作技能の習得における操作反復の効果，日本産業技術教育学会誌，第 50 巻，第 2 号，pp.77-84 (2008)
[36] 大谷忠・畑山翔子：切断加工を工夫し習得する段階における学習過程の最適化，日本産業技術教育学会誌，第 53 巻，第 1 号，pp.1-8 (2011)
[37] 有川誠・野方健治：ドライバー操作におけるメンタルモデルと操作体験との関連，日本産業技術教育学会誌，第 54 巻，第 1 号，pp.11-20 (2012)
[38] 大谷忠：加工技術を習熟し活用する段階における学習過程の最適化，日本産業技術教育学会誌，第 56 巻，第 1 号，pp.1-10 (2014)
[39] 野方健治・有川誠：ドライバー操作のメンタルモデル改善による技能指導法の開発，日本産業技術教育学会誌，第 57 巻，第 2 号，pp.103-112 (2015)
[40] 佐藤次郎：旋盤操作の作業分析，日本産業技術教育学会誌，第 13 巻，第 1 号，pp.55-58(1971)
[41] 末富正啓：角のみ盤の操作にみる子供の技術的認識過程と動作，日本産業技術教育学会誌，第 16 巻，第 1 号，pp.36-42(1974)
[42] 近藤義美：操作能力形成の学習指導法　-工作機械操作能力の転移-，日本産業技術教育学会誌，第 24 巻，第 3 号，pp.77-81 (1982)
[43] 西田好光・篠田功・小川武範・その他 1 名：技術科における安全教育 -卓上ボール盤作業を中心として-，日本産業技術教育学会誌，第 30 巻，第 3 号，pp.207-221 (1988)
[44] 田中通義・安孫子啓：かんな削り動作における身体重心の移動分析について，日本産業技術教育学会誌，第 32 巻，第 4 号，pp.249-255 (1990)
[45] 奥野信一・盛屋勝博・上田正紘：小学校教師のための木工作指導ビデオ教材の制作とその評価，日本産業技術教育学会誌，第 45 巻，第 1 号，pp.39-46 (2003)
[46] 山本利一・森山潤：材料加工学習における生徒の技能習得を把握するためのシンプトムの検討　-金属加工作業時のプロトコル分析を通して-，日本産業技術教育学会誌，第 46 巻，第 3 号，pp.123-133 (2004)

[47] 安田貢・西正明：インターネットを活用した鋸挽き作業の自己確認教材の開発，日本産業技術教育学会誌，第 47 巻，第 4 号，pp.273-280 (2005)

[48] 佐藤英雄・大島郁也：平鉋切削の技能に関する研究，日本産業技術教育学会誌，第 14 巻，第 1 号，pp.9-12(1972)

[49] 佐藤英雄・大島郁也：平鉋切削の技能に関する研究，日本産業技術教育学会誌，第 15 巻，第 1 号，pp.27-29(1973)

[50] 宮崎擴道・長松正康：手びきのこによる切削抵抗の測定方法について，日本産業技術教育学会誌，第 32 巻，第 1 号，pp.11-18 (1990)

[51] 田中通義・篠田功・山下晃功・その他 1 名：かんな削り訓練装置の開発と訓練効果，日本産業技術教育学会誌，第 33 巻，第 4 号，pp.251-256 (1991)

[52] 岡村吉永・栗林誠：刷毛塗りの技能評価に関する研究 -塗り方向の力および塗料量について-，日本産業技術教育学会誌，第 42 巻，第 2 号，pp.83-89 (2000)

[53] 岡村吉永：刷毛を用いた下塗り作業における経験の影響，日本産業技術教育学会誌，第 44 巻，第 2 号，pp.85-90 (2002)

[54] 白石拓也・岡村吉永・弘中誠・その他 1 名：測定器を用いたのこぎりびき学習の指導方法の検討，日本産業技術教育学会誌，第 51 巻，第 1 号，pp.1-6 (2009)

[55] 藤沖零士・田中通義・向井辰徳：平鉋による平面削り指導のための右手の力と左手の力の計測システムの開発と評価，日本産業技術教育学会誌，第 55 巻，第 3 号，pp.207-212 (2013)

[56] 喜多山裕人・永冨一之・藤元嘉安：カッター刃を用いた簡易かんなの開発，日本産業技術教育学会誌，第 59 巻，第 1 号，pp.49-57 (2017)

[57] 有川誠・野方健治：技能測定ドライバーシステムの開発と中学生の巧緻性評価，日本産業技術教育学会誌，第 61 巻，第 2 号，pp.105-113 (2019)

[58] 河原淳夫・宮崎擴道・田中通義：教育用改良かんなの検討　-重量について-，日本産業技術教育学会誌，第 24 巻，第 1 号，pp.127-133 (1982)

[59] 河原淳夫・上田邦夫・江田英俊：教育用工具の心理的評価 (1) -一対比較法による横引きのこぎりの選択とその要因-，日本産業技術教育学会誌，第 29 巻，第 2 号，pp.101-106 (1987)

[60] 平田晴路・河原淳夫：両刃のこぎりの柄の改良について，日本産業技術教育学会誌，第 35 巻，第 2 号，pp.121-127 (1993)

[61] 板垣翔大・安藤明伸・安孫子啓・その他 1 名：かんな掛け動作の学習を支援するスマートフォンアプリケーションの開発と家庭学習における有用性の評価，日本産業技術教育学会誌，第 58 巻，第 1 号，pp.39-48 (2016)

[62] 岡村吉永・阿濱茂樹・平田直樹・その他 1 名：木材加工におけるけがき作業の改善，日本産業技術教育学会誌，第 60 巻，第 3 号，pp.143-148 (2018)

[63] 土井康作：平鉋切削作業動作分析に関する研究 (1)，日本産業技術教育学会誌，第 28 巻，第 1 号，pp.65-69(1986)

[64] 松田健一・辻慎一郎：鋸挽き加工時の女子生徒の作業分析，日本産業技術教育学会誌，第 37 巻，第 2 号，pp.139-146 (1995)

[65] 土井康作：行為と言語を媒介にしてつくられたイメージの技能習熟への影響，日本産業技術教育学会誌，第 37 巻，第 2 号，pp.195-204 (1995)

[66] 河原淳夫・上田邦夫・平田晴路：握圧波形の評価と技能の評価との相関，日本産業技術教育学会誌，第 29 巻，第 3 号，pp.41-48(1987)

[67] 寺田盛紀・中村正寛・塩崎洋子：鉋操作技能の獲得とその指導方法，日本産業技術教育学会誌，第 33 巻，第 3 号，pp.157-163 (1991)

[68] 山本利一・牧野亮哉：垂直な釘打ちを支援する教具の開発 -ホールセンサを利用した簡易釘位置測定装置-，日本産業技術教育学会誌，第 40 巻，第 1 号，pp.9-15 (1998)

[69] 黎子椰・樺澤崇・丘華・その他 1 名：旋盤操作技能教育・訓練用シミュレータの機能開発による性能改善，日本産業技術教育学会誌，第 47 巻，第 3 号，pp.183-192 (2005)

[70] 近藤義美：操作における動作の定位点－手かんなによる繊維方向削り―，日本産業技術教育学会誌，第 18 巻，第 1 号，pp.123-127(1976)

[71] 紅林秀治・小林健太・高山大輝・その他 2 名：KINECT センサーを用いた簡易動作分析システムの開発，日本産業技術教育学会誌，第 55 巻，第 3 号，pp.213-220 (2013)

[72] 千吉良祐弥・西正明：Leap Motion Controller を用いたキー入力指判定機能と AI を用いた重点学習機能を備えたタイピング練習システム「Fingers」の開発，日本産業技術教育学会誌，第 61 巻，第 3 号，pp.223-230 (2019)

[73] 板垣翔大・千田優花・阿部博政・その他 2 名：スマートデバイスと AI の骨格検出を用いたのこぎり引き動作の学習の実践と評価，日本産業技術教育学会誌，第 63 巻，第 3 号，pp.371-378 (2021)

[74] 土井康作：ものづくりにおける児童生徒の器用・不器用感の意識構造に関する研究（第 1 報) 器用・不器用意識と遊びとの関係，鳥取大学教育地域科学紀要，教育・人文科学，第 4 巻，第 2 号，pp.333-354(2003)

[75] 川端博子・日景弥生・鳴海多恵子：手指の器用さに関する意識と実態―東京都内の私立高校生の場合―，日本衣服学会誌，第 47 巻，第 2 号，pp.99-107(2004)

[76] 鳴海多恵子, 日景弥生：手指の運動機能性と被服製作学習に関する研究 第 3 章, 平成 9～11 年度科学研究費補助金報告書 (09680005) (2000)

[77] 尾崎康子：乳幼児期における発達障害の理解と支援　知っておきたいきになる子どもの手先の器用さのアセスメント－ PWL 描画テストの手引と検査用紙, ミネル

ヴァ書房，初版，p.13-17(2018)
[78] 土井康作，高島清隆：中学生の器用・不器用意識と作業パフォーマンス及び自尊感情との関係，産業教育学研究，第 38 巻，第 1 号，pp.43-48(2008)
[79] 森下一期：私にとって器用とは，子どもの遊びと手の労働研究，第 250 巻，pp.2-20(1994)
[80] 土井康作：ものづくりに対する児童生徒の器用・不器用意識，日本産業技術教育学会誌，第 40 巻，第 1 号，pp.23-31(1998)
[81] 土井康作：児童生徒のものづくり経験の意識が器用感，意欲，技術観に及ぼす影響，技術教育研究，第 57 巻，pp.50-59(2001)
[82] 土井康作，奥野信一，横尾恒隆，坂口謙一，田中喜美，近藤義美，木村誠，角和博，森山潤，長谷川雅康：児童生徒のものづくりの教育及び中学校技術科教育に対する意識　小学校 3 年生～高等学校 3 年生を対象とした 10 都県の意識調査，産業教育学研究，第 30 巻，第 1 号，pp.57-63(2000)
[83] 小林正明，前綾佳：小中高生を対象とした工作やモノづくりに関する意識調査について，日本産業技術教育学会第 55 回全国大会，pp.79(2012)
[84] 三宅正太郎，小林正明，岩村充希子：モノづくりに関する意識調査について（その 2）－制作経験や教科の学習などとの関連から－，日本科学教育学会年会論文集，第 40 巻，pp.275-276(2016)
[85] 三宅正太郎，小林正明，岩村充希子：モノづくりに関する意識調査について（その 8）－深いアプローチ活動経験量の違いとの関連を中心に－，日本科学教育学会研究会研究報告，第 33 巻，第 2 号，pp.9-14(2018)
[86] 三山裕久・向山玉雄：「技術科」教育の社会的役割 (3) 中学生の技術科意識の分析，技術教室，産業教育研究連盟，第 598 号，pp.50-55(2002)
[87] 小森祥一・鈴木道義：技術・家庭に関する意識調査，宇都宮大学教育学部教育実践総合センター紀要，第 28 号,pp.103-116(2005)
[88] 吉村光平，中原久志，市原靖士：ものづくり活動に対する保護者の意識に関する調査，日本産業技術教育学会技術教育分科会論文集，技術科教育の研究，Study of Technology Education，第 23 巻，pp.1-7(2018)
[89] 土井康作・舟津雅也：PA32　児童生徒のものづくりの器用・不器用意識に関する因子論的研究，日本教育心理学会総会発表論文集，第 42 回総会発表論文集，p.32(2000)
[90] 金田一京助・佐伯梅友・大石初太郎・その他 1 名：新選国語辞典，第九版，小学館，p.327(2012)
[91] 松村明：大辞泉　上巻あーす，第二版，小学館，p.935(2012)
[92] 新村出：広辞苑　机上版，第 7 版，岩波書店，p.754(2018)
[93] 見坊豪紀・市川孝・飛田良文・その他 3 名：三省堂国語辞典，三省堂，第七版，

p.363(2018)
[94] 小野正弘・市川孝・見坊豪紀・その他 4 名：三省堂現代新国語辞典，三省堂，第六版，p.345(2019)
[95] 西尾実・岩淵悦太郎・水谷静夫・その他 3 名：岩波国語辞典，岩波書店，第八版，pp.369,370(2019)
[96] 中村明：新明解類語辞典，三省堂，p.263(2019)
[97] 山田忠雄・倉持保男・上野善道・その他 3 名：新明解国語辞典，三省堂，第八版，p.380(2020)
[98] 下中弥三郎：心理学事典，平凡社，p.124(1957)
[99] 園原太郎：心理学辞典，ミネルヴァ書房，p.78(1971)
[100] 宮城音弥：岩波心理学小辞典，岩波書店，p.50(1979)
[101] 大中邦彦：新版心理学事典，平凡社，pp.154-155(1981)
[102] 日本スポーツ心理学会：スポーツ心理学事典，大修館書店，p.75(2008)
[103] 谷田貝公昭：子ども心理辞典，一藝社，pp.103,104(2011)
[104] 時実利彦：人間であること，岩波新書，p.93(1977)
[105] 森下一期：私にとって器用とは，子どもの遊びと手の労働研究，第 250 巻，pp.6-7(1994)
[106] 槇塚忠穂：手先の器用さと習熟度の違い―陶芸初心者のロクロ形成―，労働の科学，第 55 巻，第 1 号，pp.55-58(2000)
[107] 久保田競・虫明元・宮井一郎：ライブラリ：脳の世紀：心のメカニズムを探る　6　学習と脳―器用さを獲得する脳―，サイエンス社，初版第 5 刷，p.13(2013)
[108] 久保田競：手と脳　増補新装版，紀伊國屋書店，pp.141-142(2015)
[109] 文部科学省：中学校学習指導要領 (平成 29 年告示) 解説　技術・家庭編，開隆堂，pp.16-17 (2018)
[110] 古川稔：平成 29 年版　中学校新学習指導要領の展開　技術・家庭　技術分野編，明治図書，p.10(2017)
[111] 古川稔・杉山久仁子：平成 29 年改訂　中学校教育課程実践講座　技術・家庭，ぎょうせい，p.10(2017)
[112] 見坊豪紀・市川孝・飛田良文・その他 3 名：三省堂国語辞典，三省堂，第七版，p.485(2018)
[113] 人間工学用語研究会：人間工学事典，日刊工業新聞社，pp.109,110(1983)
[114] 枝光慶江・濱野恵一：知能及び手指の巧緻性の発達に関する研究，児童臨床研究所年報，pp.5-15(1994)
[115] 中村隆一・齋藤宏：基礎運動学，第 5 版，医歯薬出版，p.412(2000)
[116] 和田攻・南裕子・小峰光博・その他 2 名：看護大事典，医学書院，p.938(2002)

[117] 最新医学大辞典編集委員会：最新医学大辞典，第 3 版，医歯薬出版，p.592(2005)
[118] 日本生理人類学会：人間科学の百科事典，丸善出版，pp.160,161(2015)
[119] 日本産業技術教育学会・技術教育分科会：技術科教育概論，九州大学出版会 (2018)
[120] 辺見一男・中坂祐賢：モーションキャプチャ手法を用いた箸使い動作の解析，県立長崎シーボルト大学国際情報学部紀要，第 7 巻，p.19-28(2006)
[121] ラーマンモハッマドモスタフィズル・水戸部一孝・鈴木雅史他 1 名：ピアノ教育用教材としての手指用モーションキャプチャシステムの利用可能性の検討，日本バーチャルリアリティ学会論文誌，第 16 巻，第 1 号，p.83-92(2011)
[122] 尾山匡浩・秋田悠登：小型モーションセンサを用いた指文字単語認識，情報科学技術フォーラム講演論文集，第 19 巻，第 3 分冊，pp.335-336(2020)
[123] 柴田凌輔・中井満：仮想空間でのハンドトラッキングを用いた筆記システムの構築，情報科学技術フォーラム講演論文集，第 19 巻，第 3 分冊，pp.323-324(2020)
[124] 辻天斗・牛田啓太・山口実靖・陳キュウ：手遊びや影絵を用いた 3D モデルの共同的操作システムの開発と評価，研究報告ヒューマンコンピュータインタラクション（HCI），第 2019-HCI-181 巻，第 7 号，pp.1-8(2019)
[125] 佐藤周平・藤村誠：片麻痺治療支援システムのための Leap Motion による手指モーションキャプチャの検討，情報処理学会全国大会講演論文集，第 77 巻，第 1 号，pp.401-402(2015)
[126] 中村祐治・古川稔・太田達郎・その他 45 名：New　技術・家庭　技術分野　明日を創造する　技術ハンドブック，教育図書，p.40(2021)
[127] 安東茂樹・塩入睦夫・竹野英敏：技術・家庭 [技術分野]，開隆堂，pp.262,263(2016)
[128] 佐竹隆顕・市川道和・淡野一郎・その他 8 名：新技術・家庭　技術分野，教育図書，pp.270,271(2016)
[129] 田口浩継・佐藤文子・金子佳代子・その他 62 名：新編　新しい技術・家庭　技術分野　未来を創る Technology，東京書籍，pp.272-275(2016)
[130] 深川和良・福永麻衣果：備品からみる鹿児島県内における中学校技術科の材料と加工に関する一考察：木材加工，鹿児島大学教育学部研究紀要　教科科学編，第 69 巻，pp.59-69(2018)
[131] 中原久志・上之園哲也・森山潤：技術科の材料加工学習における生徒の「癒し」と「ストレス」の感情状況の構造と学習意欲との関連，日本教科教育学会誌，第 39 巻，第 1 号，pp.1-10(2016)
[132] 増田麻人：蒸気タービンを用いた火力発電教材の製作とディベートによってエネルギー資源の問題を考える教育の実践，日本産業技術教育学会誌，第 60 巻，第 4 号，pp.245-250(2018)
[133] 長谷川雅康・廣田拓也：子どもの手の働きと意欲に関する調査：鹿児島の小学生の

事例，鹿児島大学教育学部研究紀要　教育科学編，第 62 巻，pp.167-201 (2010)
[134] 谷田貝公昭・村越晃・生駒恭子・その他 21 名：イラスト版手のしごと　子どもとマスターする 49 の生活技術，合同出版，pp.68,69(1996)
[135] 文部科学省：小学校学習指導要領（平成 29 年告示）解説　理科編，東洋館出版社 (2018)
[136] 下田好行・他 19 名：「キー・コンピテンシー」に基づく学習指導法のモデル開発に関する研究，平成 20 年度科学研究費補助金基盤研究（Ｂ）(課題番号：19330208)　研究成果中間報告書（第 2 次）
[137] 稲田結美：女子の物理学習に対する意識向上のための人体アプローチ，物理教育，第 59 巻，第 3 号
[138] 末富正啓：技術科教育に関する研究―小学生の労働体験とくに道具の使用体験について―，山口大学教育学部研究論叢，第 3 部，芸能・体育・教育・心理，第 25 巻，第 3 号，pp.99-107(1976)
[139] 谷田貝公昭・村越晃・西方穀・その他 10 名：現代の子どもの生活技術の実態 2，日本保育学会大会研究論文集，第 42 巻, pp.444-445 (1989)
[140] 谷田貝公昭・村越晃・伊藤野里子・その他 11 名：現代の子どもの生活技術の実態 5 －1988 年と 1996 年の比較－，日本保育学会大会研究論文集，第 50 巻, pp.576-577 (1997)
[141] 渋谷寿：野外教育における造形活動 (1)―子どもの道具使用実態調査を通して―，名古屋女子大学紀要，第 33 巻，pp.175-185(1987)
[142] 渋谷寿：野外教育における造形活動 (第 2 報)―子どもの道具使用実態調査・親子関係診断調査を通して―，名古屋女子大学紀要，第 34 巻，pp.235-247(1988)
[143] 中村薫：Leap Motion プログラミングガイド，工学社 (2014)
[144] 笹塚和宏：± 10 $\mu$ m で動きを検知 Leap の実体に迫る，日経エレクトロニクス (1113)，pp.59-65 (2013)
[145] Processing：https://processing.org/ (閲覧日：2015 年 9 月 17 日)
[146] 赤間世紀：Processing GUI プログラミング，工学社 (2013)
[147] Leap Motion SDK：https://developer.leapmotion.com/downloads (閲覧日：2016 年 1 月 11 日)
[148] ハッコー：my pen（はんだ付け用），http://www.hakko.com/japan/products/hakko_my-pen_soldering.html(2019.8.17 閲覧)
[149] VR Setup － Leap Motion Developer：https://developer.leapmotion.com/get-started(2019.11.5 閲覧)
[150] 泉田成美・佐藤美佳・工藤詩織・他 4 名：普通箸の操作獲得を目指した訓練の段階付けとして自助具の使用は有効か？，弘前大学医学部保健学科作業療法学専攻卒業

論文集，第 9 巻，pp.25-31 (2013)

[151] 宮崎食器：http://www.mtsco.co.jp/ (閲覧日：2016 年 2 月 7 日)

[152] Bastron：http://www.bastron.com.cn/ (閲覧日 2015 年 10 月 26 日)

[153] タイピング技能検定イータイピング・マスター：https://web.e-typing.ne.jp/mock/ (閲覧日：2016 年 9 月 1 日)

[154] 山岡武邦・白濱弘幸・沖野信一・他 1 名：小学生のための自然理解に通じる知的好奇心を促進させる科学実践と評価 ー「永久ゴマ」の製作を自制としてー，日本科学教育学会年会論文集 (CD-ROM)，pp.351-352 (2016)

[155] VESSEL：ボールグリップ差替ドライバーセット No.220W-3(ビット 3 本組), https://www.vessel.co.jp/product/screwdriver/detail/233072 (最終アクセス日：2020 年 4 月 30 日)

[156] 矢崎潔・小森健司・田口真哉：手の運動を学ぶ 手の役割と手の機能解剖との関係から運動を紐解き，臨床に生かす，三輪書店，pp.53-57 (2019)

[157] 手塚太郎：技能学習における速さと正確さの関係についての心理学的研究，杉山書店，pp.258-265 (1996)

[158] 文部科学省：木材を活用した学校用家具の事例集，第 2　章　調査研究結果 (2)，https://www.mext.go.jp/a_men　u/shisetu/shuppan/06051207/003.html (2020 年 9 月 15 日確認)

[159] 牧野浩二：たのしくできる　Arduino 電子制御 Processing でパソコンと連携，東京電機大学出版局 (2015)

[160] 松本俊之・道用大介・市来嵜治・その他 1 名：タッピンねじ締め作業における押圧力に関する基礎研究，日本人間工学会誌，第 38 巻，第 3 号，pp.143-151 (2002)

[161] 松本俊之・志田敬介・金沢孝：タッピングねじ締め作業における作業方向と作業位置が作業効率に及ぼす影響，日本経営工学会論文誌，第 52 巻，第 6 号，pp.332-343 (2002)

# 付録 A　簡易手指動作分析システムのプログラム

```
import ppopupmenu.*;
import java.awt.Frame;
import java.awt.BorderLayout;
import controlP5.*;
import development.*;
import javax.swing.*; //データ変換確認のボックスの表示に必要


import de.voidplus.leapmotion.*;

LeapMotion leap;

// ファイル書き出し用ライブラリ
PrintWriter outputR;

PrintWriter outputL;


ControlFrame cf;
private ControlP5 cp5_Menubutton,cp5sl1,cp5sl2,cp5button,cp5sl3,cp5_Menu,cp5_Menu2,cp5_Menu3,cp5sl4,cp5sl5,cp5sl6,cp5zoom,cp5sl7,cp5sl8,cp5r;

int Rightspeed = 0;
int Leftspeed = 0;
int Rightspeed2 = 0;
int Leftspeed2 = 0;


Slider sl1,sl2,sl4,sl5,sl6,sl7,sl8;
Slider2D sl3;

Accordion accordion,accordion2,accordion3;
RadioButton cb,rb,rb3;
//CheckBox ;
PPopupMenu ppm1,ppm2;
Range rangeXaxis;
Numberbox zoomValue,MaxAxis,MinAxis;

//透視変換用の変数
//視野角θ
float radS = TWO_PI/360*35;
//カメラの原点からの距離
float camZ = -8000;
//スクリーンの原点からの距離
float scZ = -1000;
//回転
float xRotate = PI/2;
float yRotate = -PI/2;
float zRotate = 0;
//回転右データ 1
float xRotateR1 = 0;
float yRotateR1 = 0;
float zRotateR1 = 0;
//回転左データ 1
float xRotateL1 = 0;
float yRotateL1 = 0;
float zRotateL1 = 0;
//回転右データ 2
float xRotateR2 = 0;
float yRotateR2 = 0;
float zRotateR2 = 0;
//回転左データ 2
float xRotateL2 = 0;
float yRotateL2 = 0;
float zRotateL2 = 0;
```

```
//回転影響係数（環境設定画面でのボタンに関するもの）
int xR =0;
int yR =0;
int zR =0;

//描画用の変数
//線の太さ
float weight = 0.5;
//アニメーションの線の色
int strokeA1 = 255;
int strokeA2 = 0;
int ColorAni =0;
int NextColorAni =0;
//二次関数の線の色
int strokeQ1 = 255;
int strokeQ2 = 0;
//三次関数の線の色
int strokeC1 = 255;
int strokeC2 = 0;
//CSV 形式をデータへ保存
String loadPath1;
int LENGTHF;//ファイルの長さ判別
String [][] csvF;//文字列データ用
float [][]csvF1;//数値データ用
float [][]csvF2;//原点修正データ用
float [] maxF1 = new float[76];  //データの最大値（F1）
float [] minF1 = new float[76];  //データの最小値（F1）
float [] maxF2 = new float[76];  //データの最大値（F2）
float [] minF2 = new float[76];  //データの最小値（F2）
float [] maxF0 =new float[76];
int csvWidth=76;

//CSV 形式をデータへ保存
String loadPath2;
int LENGTHS;//ファイルの長さ判別
String [][] csvS;//文字列データ用
float [][]csvS1;//数値データ用
float [][]csvS2;//原点修正データ用
float [] maxS1 = new float[76];  //データの最大値（S1）
float [] minS1 = new float[76];  //データの最小値（S1）
float [] maxS2 = new float[76];  //データの最大値（S2）
float [] minS2 = new float[76];  //データの最小値（S2）
float [] maxS0 =new float[76];

//CSV 形式をデータへ保存
String loadPath3;
int LENGTHT;//ファイルの長さ判別
String [][] csvT;//文字列データ用
float [][]csvT1;//数値データ用
float [][]csvT2;//原点修正データ用
float [] maxT1 = new float[76];  //データの最大値（T1）
float [] minT1 = new float[76];  //データの最小値（T1）
float [] maxT2 = new float[76];  //データの最大値（T2）
float [] minT2 = new float[76];  //データの最小値（T2）
float [] maxT0 =new float[76];

//CSV 形式をデータへ保存
String loadPath4;
int LENGTHH;//ファイルの長さ判別
String [][] csvH;//文字列データ用
float [][]csvH1;//数値データ用
float [][]csvH2;//原点修正データ用
float [] maxH1 = new float[76];  //データの最大値（H1）
float [] minH1 = new float[76];  //データの最小値（H1）
float [] maxH2 = new float[76];  //データの最大値（H2）
float [] minH2 = new float[76];  //データの最小値（H2）
float [] maxH0 =new float[76];

float timechange =30; //保存したデータの中で最も長い計測時間

int i=2; //右手の列数
int j=2; //左手の列数
int i2=2; //右手の列数
int j2=2; //左手の列数
int z=300;

float LEF_Modi=0;//LENGTHF/360.0 の float 型化
float LES_Modi=0;//LENGTHS/360.0 の float 型化
float i_Modi=0;//i の float 型化
float  j_Modi=0;//j の float 型化

PImage img1,img2,img3,img4,img5,img6;//画像データ

int x_handphoto=-1000,y_handphoto=-1000;

int x_moveA=0,y_moveA=0;
int x_moveQ=0,y_moveQ=0;
int x_moveC=0,y_moveC=0;
```

```
int x=73;
int y,d;

  int e=0;

  int aaa=255;

  int q=0;

int play=0;

int playR2=0;
int playL2=0;

int selectButton=0; //メインループ上でどの機能（アニメーションや変位グラフ etc）を表示する決まる

//各データが保存されたかを確認するための変数
int fitdata1 =0;
int fitdata2 =0;
int fitdata3 =0;
int fitdata4 =0;

int selectLiAJo=0;

//調整画面にある「平行移動」の 2D スライダーの動かす値宣言
float memoriX=60.5;
float memoriY=50.5;

//データを読み込んだ
int SaveData=0;

int addmX1=0,addmY1=0;
int addmX2=0,addmY2=0;
int addmX3=0,addmY3=0;
int addmX4=0,addmY4=0;


int mv1=0,mv2=0,mv3=0,mv4=0,mv5=0;

int r1r=0,r2r=0,l1l=0,l2l=0;

int sel1=0,sel2=0,sel3=0,sel4=0,sel5=0;

int DwEl=0; //円描画に関する変数
int DwAx=1; //変位グラフの縦軸が変化するための変数

int numberbox =0; //zoomvalue の値
int numberboxS =0; //MaxAxis の値
int numberboxT =0; //MinAxis の値


//変位グラフの縦軸に関する変数
int AxisP =100; //縦軸の Max
int AxisM =-100; //縦軸の Min
int DifAxi =0; //縦軸の Max-Min した値

//変位グラフの縦軸の調整に関する変数
int VerAxisIs =0; //関節が変わったときに，値が変わるようにする
//int NumBoxS =0; //numberboxS（縦軸の最大値）と関数内の zoom との差がいくつかを判断
//int NumBoxT =0; //numberboxT（縦軸の最小値）と関数内の zoom との差がいくつかを判断

//変位グラフの線の座標
//右データ 1
float FirPoX1=0; //直線の左側の点（X 座標）
float FirPoY1=0; //直線の左側の点（Y 座標）
float NexPoX1=0; //直線の右側の点（X 座標）
float NexPoY1=0; //直線の右側の点（Y 座標）
float EllipX1 =0; //描画する円（X 座標）
float EllipY1 =0; //描画する円（Y 座標）
//左データ 1
float FirPoX2=0; //直線の左側の点（X 座標）
float FirPoY2=0; //直線の左側の点（Y 座標）
float NexPoX2=0; //直線の右側の点（X 座標）
float NexPoY2=0; //直線の右側の点（Y 座標）
float EllipX2 =0; //描画する円（X 座標）
float EllipY2 =0; //描画する円（Y 座標）
//右データ 2
float FirPoX3=0; //直線の左側の点（X 座標）
float FirPoY3=0; //直線の左側の点（Y 座標）
float NexPoX3=0; //直線の右側の点（X 座標）
float NexPoY3=0; //直線の右側の点（Y 座標）
float EllipX3 =0; //描画する円（X 座標）
float EllipY3 =0; //描画する円（Y 座標）
//左データ 2
float FirPoX4=0; //直線の左側の点（X 座標）
float FirPoY4=0; //直線の左側の点（Y 座標）
float NexPoX4=0; //直線の右側の点（X 座標）
float NexPoY4=0; //直線の右側の点（Y 座標）
float EllipX4 =0; //描画する円（X 座標）
float EllipY4 =0; //描画する円（Y 座標）
```

```
int ThreeGraSepa=0; //3D グラフで平面，目盛り及び軌跡を分ける変数
int DeleteJointXY=1; //XY 平面を選択するか否か
int DeleteJointXZ=1; //XZ 平面を選択するか否か
int DeleteJointYZ=1; //YZ 平面を選択するか否か

int DisplayA=100,DisplayB=200; //表示される値

//LeapMotion で計測するのに使う関数
//////////////////////////////////////
int RightCount =6;
int LeftCount =6;
//////////////////////////////////////

//データの名前と保存先を選択する
  String filenameR  ;
  String filenameL  ;
//LeapMove を実行する前にデータの名前と保存先を決定するための変数
  int FornameR=0;
  int FornameL=0;
//LeapMove で時間を表示するための変数
  int ForLeapTime = 0;
//  long fps =0;
  long fps1 =0;
  long fps2 =0; //目印をつけてからの時間
  long FpsForNoCall =0;//コールバックをなくし繰り返し座標を保存しないための変数
//計測していないときの保存時に，同じ計測時間で保存しないための変数
 int NoCallR =0;
 int NoCallL =0;

int ChangeFingerDirection=0;//普通に計測と裏向き計測の関数

int DataFromAbove=0;
int ScreenUp=0,ScreenRight=0;

////////////////////////////////////////////////////
//関数名：setup
//概要：初期設定
//関連する関数
//     ・ReadImage<<画像ファイルの読み込み>>
//     ・CreateMenu<<サブメニューの作成>>
//     ・CreateToolBar<<ツールバーの作成>>
////////////////////////////////////////////////////
void setup()
{
  size(600, 500,OPENGL); //OPENGL がなくてもよい
  leap = new LeapMotion(this); //ここが EXport Application のできない理由

  frameRate(30);
  ReadImage(); //画像ファイルの読み込み
  CreateEdit(); //編集の作成
  CreateToolBar(); //ツールバーの作成
  //PrepareLeap(); //LeapMotion の設定

 println(9);
   long fps = leap.getTimestamp();
  println(fps,FpsForNoCall); //青木


}

////////////////////////////////////////////////////
//関数名：PrepareLeap
//概要：LeapMotion で動作を計測・保存する準備をする
//関連する関数
//     ・setup<<初期設定>>
////////////////////////////////////////////////////
void PrepareLeap()
{
//  leap = new LeapMotion(this);

 // ファイル名の設定
    selectOutput("左手のデータの名前と保存先を選択してください", "recordingFileSelectedL");

    selectOutput("右手のデータの名前と保存先を選択してください", "recordingFileSelectedR");

}

void recordingFileSelectedR(File selectionR){
 if (selectionR == null) {
    println("Window was closed or the user hit cancel.");
    //保存できていない
    String RightData = filenameR;
    outputR = createWriter(filenameR + ".csv");
    outputR.print( "Timestamp ,Thumb(1),,,Thumb(2),,,Thumb(3),,,Thumb(4),,,"); //時間と親指
    outputR.print("Index(1),,,Index(2),,,Index(3),,,Index(4),,,Index(5),,,"); //人差し指
    outputR.print("Middle(1),,,Middle(2),,,Middle(3),,,Middle(4),,,Middle(5),,,"); //中指
    outputR.print("Ring(1),,,Ring(2),,,Ring(3),,,Ring(4),,,Ring(5),,,"); //薬指
    outputR.print("Pinky(1),,,Pinky(2),,,Pinky(3),,,Pinky(4),,,Pinky(5),,,"); //小指
    outputR.println("Center,Of,Gravity" ); //重心

    outputR.print( " ,x,y,z,x,y,z,x,y,z,x,y,z,"); //親指
```

```
    outputR.print("x,y,z,x,y,z,x,y,z,x,y,z,x,y,z,"); //人差し指
    outputR.print("x,y,z,x,y,z,x,y,z,x,y,z,x,y,z,"); //中指
    outputR.print("x,y,z,x,y,z,x,y,z,x,y,z,x,y,z,"); //薬指
    outputR.print("x,y,z,x,y,z,x,y,z,x,y,z,x,y,z,"); //小指
    outputR.println("x,y,z" ); //重心
  } else {
    println("User selected " + selectionR.getAbsolutePath());
    filenameR = selectionR.getAbsolutePath();
      // 右手ファイルを生成
  outputR = createWriter(filenameR + ".csv");
  outputR.print( "Timestamp ,Thumb(1),,,Thumb(2),,,Thumb(3),,,Thumb(4),,,"); //時間と親指
  outputR.print("Index(1),,,Index(2),,,Index(3),,,Index(4),,,Index(5),,,"); //人差し指
  outputR.print("Middle(1),,,Middle(2),,,Middle(3),,,Middle(4),,,Middle(5),,,"); //中指
  outputR.print("Ring(1),,,Ring(2),,,Ring(3),,,Ring(4),,,Ring(5),,,"); //薬指
  outputR.print("Pinky(1),,,Pinky(2),,,Pinky(3),,,Pinky(4),,,Pinky(5),,,"); //小指
  outputR.println("Center,Of,Gravity" ); //重心

  outputR.print( " ,x,y,z,x,y,z,x,y,z,x,y,z,"); //親指
  outputR.print("x,y,z,x,y,z,x,y,z,x,y,z,x,y,z,"); //人差し指
  outputR.print("x,y,z,x,y,z,x,y,z,x,y,z,x,y,z,"); //中指
  outputR.print("x,y,z,x,y,z,x,y,z,x,y,z,x,y,z,"); //薬指
  outputR.print("x,y,z,x,y,z,x,y,z,x,y,z,x,y,z,"); //小指
  outputR.println("x,y,z" ); //重心
  }
  FornameR =1;
}

void recordingFileSelectedL(File selectionL){
 if (selectionL == null) {
    println("Window was closed or the user hit cancel.");
   //保存できていない
   String LeftData = filenameL;
   outputL = createWriter(filenameL + ".csv");
   outputL.print( "Timestamp ,Thumb(1),,,Thumb(2),,,Thumb(3),,,Thumb(4),,,"); //時間と親指
   outputL.print("Index(1),,,Index(2),,,Index(3),,,Index(4),,,Index(5),,,"); //人差し指
   outputL.print("Middle(1),,,Middle(2),,,Middle(3),,,Middle(4),,,Middle(5),,,"); //中指
   outputL.print("Ring(1),,,Ring(2),,,Ring(3),,,Ring(4),,,Ring(5),,,"); //薬指
   outputL.print("Pinky(1),,,Pinky(2),,,Pinky(3),,,Pinky(4),,,Pinky(5),,,"); //小指
   outputL.println("Center,Of,Gravity" ); //重心

   outputL.print( " ,x,y,z,x,y,z,x,y,z,x,y,z,"); //親指
   outputL.print("x,y,z,x,y,z,x,y,z,x,y,z,x,y,z,"); //人差し指
   outputL.print("x,y,z,x,y,z,x,y,z,x,y,z,x,y,z,"); //中指
   outputL.print("x,y,z,x,y,z,x,y,z,x,y,z,x,y,z,"); //薬指
   outputL.print("x,y,z,x,y,z,x,y,z,x,y,z,x,y,z,"); //小指
   outputL.println("x,y,z" ); //重心
  } else {
    println("User selected " + selectionL.getAbsolutePath());
    filenameL = selectionL.getAbsolutePath();
     // 左手ファイルを生成
  outputL = createWriter(filenameL + ".csv");
  outputL.print( "Timestamp ,Thumb(1),,,Thumb(2),,,Thumb(3),,,Thumb(4),,,"); //時間と親指
  outputL.print("Index(1),,,Index(2),,,Index(3),,,Index(4),,,Index(5),,,"); //人差し指
  outputL.print("Middle(1),,,Middle(2),,,Middle(3),,,Middle(4),,,Middle(5),,,"); //中指
  outputL.print("Ring(1),,,Ring(2),,,Ring(3),,,Ring(4),,,Ring(5),,,"); //薬指
  outputL.print("Pinky(1),,,Pinky(2),,,Pinky(3),,,Pinky(4),,,Pinky(5),,,"); //小指
  outputL.println("Center,Of,Gravity" ); //重心

  outputL.print( " ,x,y,z,x,y,z,x,y,z,x,y,z,"); //親指
  outputL.print("x,y,z,x,y,z,x,y,z,x,y,z,x,y,z,"); //人差し指
  outputL.print("x,y,z,x,y,z,x,y,z,x,y,z,x,y,z,"); //中指
  outputL.print("x,y,z,x,y,z,x,y,z,x,y,z,x,y,z,"); //薬指
  outputL.print("x,y,z,x,y,z,x,y,z,x,y,z,x,y,z,"); //小指
  outputL.println("x,y,z" ); //重心
  }
    FornameL =1;
}


/*  String
CorrectionSavePath(String str_path)
{
  String[] m = match(str_path, ".csv$");
  if(m == null)
  {
    return str_path+".csv";
  }
  return str_path;
}*/

////////////////////////////////////////////////////////
//関数名：ReadImage
//概要：画像ファイルを読み込む.
//関連する関数
//      ・setup<<初期設定>>
////////////////////////////////////////////////////////
void ReadImage()
{
  img1 = loadImage("両手 (for part2 in processing).png");
  img2 = loadImage("手とリープの位置関係.png");
}
```

```
////////////////////////////////////////////////////////
//関数名：CreateEdit
//概要：速度調整や画面調整などの編集機能の作成をする.
//関連する関数
//      ・setup<<初期設定>>
////////////////////////////////////////////////////////
void CreateEdit()
{
  cp5_Menubutton = new ControlP5(this); //サブメニュー宣言
  PFont p = createFont("Meiryo-48",12); //文字フォントの設定
  cp5_Menubutton.setControlFont(p,12); //サブメニューに文字フォント適用
  cf = addControlFrame("３．編集", 400,540); //サブメニューの大きさ設定 //5/12  サブすくｒーサイズ
}

////////////////////////////////////////////////////////
//関数名：CreateToolBar
//概要：読み込みと表示の機能をもつツールバーの作成をする.
//関連する関数
//      ・setup<<初期設定>>
//      ・controlEvent<<左右のデータを選別>>
//      ・radio<<２．表示の実行条件>>
////////////////////////////////////////////////////////
void CreateToolBar()
{
  //１．ファイルの前準備
  cp5_Menu = new ControlP5(this); //ツールバー宣言
  PFont p = createFont("Meiryo-48",12); //文字フォントの設定
  cp5_Menu.setControlFont(p,12); //ツールバーに文字フォント適用
  //１．ファイルの詳細設定
  Group g1 = cp5_Menu.addGroup("１．ファイル") //「ファイル」宣言
                     .setBackgroundColor(color(100, 20)) //背景の設定
                     .setBackgroundHeight(100) //「ファイル」の背景の高さ設定
                     .setHeight(18) //「ファイル」という文字入れる高さ設定
                     ;
  cb = cp5_Menu.addRadioButton("SelectData") //「SelectData」宣言
                .setPosition(10,5) //ボタンの配置
                .setColorForeground(color(0,255,0)) //矢印がボタンに触れると緑色
                .setColorActive(color(255,0,0)) //矢印がボタンを押すと赤色
                .setColorLabel(color(0,0,0)) //文字の色設定（黒色）
                .setSize(15,15) //ボタンの色を大きさ設定
                .setItemsPerRow(1)//1 行当たりのチェックボックスの数
        //      .setSpacingColumn(80)//チェックボックスの列の空白：左右
                .setSpacingRow(10)//チェックボックスの列の空白：上下
                .addItem("右データ１",0)//項目の追加「addItem(theName(string),theValue(float))」
                .addItem("左データ１",1)
                .addItem("右データ２",2)
                .addItem("左データ２",3)
                .moveTo(g1) //SelectData に合わせてボタン移動
                ;
  accordion = cp5_Menu.addAccordion("acc") //「acc」宣言
                      .setPosition(0,0) //acc の位置設定
                      .setWidth(120) //acc の長さ＝１．ファイルの長さ
                      .addItem(g1) //１．ファイルと連動
                      ;
  accordion.open(0); //acc を開いた状態にする
  accordion.setCollapseMode(Accordion.MULTI); //よく分からない!!!!!!!!でも必要

  //２．表示の前準備
  cp5_Menu2 = new ControlP5(this); //ツールバー宣言
  cp5_Menu2.setControlFont(p,12); //ツールバーに文字フォント適用
  //２．表示の詳細設定
  Group g2 = cp5_Menu2.addGroup("２．表示") //「表示」宣言
                      .setBackgroundColor(color(100, 20)) //背景の設定
                      .setBackgroundHeight(100) //「表示」の背景の高さ設定
                      .setHeight(18) //「表示」という文字入れる高さ設定
                      ;
  rb = cp5_Menu2.addRadioButton("radio") //「radio」宣言
                 .setPosition(10,5) //ボタンの配置
                 .setColorForeground(color(0,255,0)) //矢印がボタンに触れると緑色
                 .setColorActive(color(255,0,0)) //矢印がボタンを押すと赤色
                 .setColorLabel(color(0,0,0)) //文字の色設定（黒色）
                 .setItemWidth(15) //ボタンの長さ設定
                 .setItemHeight(15) //ボタンの高さ設定
                 .setItemsPerRow(2)//1 行当たりのチェックボックスの数
                 .setSpacingColumn(150)//チェックボックスの列の空白：左右
                 .setSpacingRow(10)//チェックボックスの列の空白：上下
                 .addItem("１．全て", 6) //項目の追加「addItem(theName(string),theValue(float))」
                 .addItem("５．アニメーションと変位グラフ", 3)
                 .addItem("２．アニメーション", 0)
                 .addItem("６．アニメーションと３Ｄグラフ", 4)
                 .addItem("３．変位グラフ", 1)
                 .addItem("７．変位グラフと３Ｄグラフ", 5)
                 .addItem("４．３Ｄグラフ", 2)
                 .moveTo(g2) //radio に合わせてボタン移動
                 ;
  accordion2 = cp5_Menu2.addAccordion("acc2") //「acc2」宣言
                        .setPosition(123,0) //acc2 の位置設定
                        .setWidth(377) //acc2 の長さ＝２．表示の長さ
                        .addItem(g2) //２．表示と連動
                        ;
```

```
  accordion2.setCollapseMode(Accordion.MULTI);//よく分からない!!!!!!!!でも必要

  //３．計測の前準備
  cp5_Menu3 = new ControlP5(this); //ツールバー宣言
  cp5_Menu3.setControlFont(p,12); //ツールバーに文字フォント適用
  //３．計測の詳細設定
  Group g3 = cp5_Menu3.addGroup("３．計測") //「計測」宣言
                      .setBackgroundColor(color(100, 20)) //背景の設定
                      .setBackgroundHeight(100) //「計測」の背景の高さ設定
                      .setHeight(18) //「計測」という文字入れる高さ設定
                      ;
  rb3 = cp5_Menu3.addRadioButton("MeasurementData") //「MeasurementData」宣言
                 .setPosition(10,5) //ボタンの配置
                 .setColorForeground(color(0,255,0)) //矢印がボタンに触れると緑色
                 .setColorActive(color(255,0,0)) //矢印がボタンを押すと赤色
                 .setColorLabel(color(0,0,0)) //文字の色設定（黒色）
                 .setItemWidth(15) //ボタンの長さ設定
                 .setItemHeight(15) //ボタンの高さ設定
                 .setItemsPerRow(1)//1 行当たりのチェックボックスの数
                 .setSpacingColumn(150)//チェックボックスの列の空白：左右
                 .setSpacingRow(10)//チェックボックスの列の空白：上下
                 .addItem("下から",0)//項目の追加「addItem(theName(string),theValue(float))」
                 .addItem("上から",3)//5/12 追加反対向きで計測できる
                 .addItem("おわり",1)
                 .addItem("目印",2)
                 .moveTo(g3) //MeasurementData に合わせてボタン移動
                 ;
  accordion3 = cp5_Menu3.addAccordion("acc3") //「acc3」宣言
                        .setPosition(503,0) //acc3 の位置設定
                        .setWidth(97) //acc3 の長さ＝３．計測の長さ
                        .addItem(g3) //３．計測と連動
                        ;
   accordion3.setCollapseMode(Accordion.MULTI);//よく分からない!!!!!!!!でも必要



  //ポップアップメニューの設定
  ppm1 = new PPopupMenu(this);//アニメーション時のポップアップメニュー宣言
  ppm1.addMenuItem("右データ 1","J1_End");//(表示させる文字，制御する関数
  ppm1.addMenuItem("左データ 1","J2_End");//(表示させる文字，制御する関数
  ppm1.addMenuItem("右データ 2","J3_End");//(表示させる文字，制御する関数
  ppm1.addMenuItem("左データ 2","J4_End");//(表示させる文字，制御する関数

  ppm2 = new PPopupMenu(this);//3D グラフと，アニメーションと 3D グラフ字のポップアップメニュー宣言
  ppm2.addMenuItem("目盛り表示","J5_End");//(表示させる文字，制御する関数

}

void J1_End()
{
  mv1=1;
  sel1=sel1+1;

}
void J2_End()
{
  mv2=1;
  sel2=sel2+1;

}
void J3_End()
{
  mv3=1;
  sel3=sel3+1;
}
void J4_End()
{
  mv4=1;
  sel4=sel4+1;
}
void J5_End()
{
  mv5=1;
  sel5=sel5+1;
  if(sel5 ==2){sel5 =0;}
}
//////////////////////////////////////////////////////
//関数：MeasurementData
//概要：
//////////////////////////////////////////////////////

void MeasurementData(int theD)
{
  switch(theD)
  {
    //はじめ
    case(0):
    println("test0");
    RightCount=0;
    LeftCount=0;
```

```
    ChangeFingerDirection=0;
    PrepareLeap();
    break;

    //おわり
    case(1):
    println("test1");
    RightCount=6;
    LeftCount=6;
    FornameR =0;
    FornameL =0;
    ForLeapTime = 0;
    SaveData =1;
    ChangeFingerDirection=2;
    break;

    //目印
    case(2):
    println("test2");
    RightCount=0;
    LeftCount=0;
    Mark();
    break;

    //反対向き
    case(3):
    println("test3");
    RightCount=0;
    LeftCount=0;
    ChangeFingerDirection=1;
    PrepareLeap();
    break;
  }
}

void Mark(){
 fps2 = leap.getTimestamp();
  outputR.print(fps2+","); //時間
 outputR.print("Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,"); //親指
 outputR.print("Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,"); //人差し指
 outputR.print("Mark,Mark,Mark,Mark,Mark,Invalid,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,"); //中指
 outputR.print("Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,"); //薬指
 outputR.print("Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,"); //小指
 outputR.println("Mark,Mark,Mark"); //重心

 outputL.print(fps2+","); //時間
 outputL.print("Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,"); //親指
 outputL.print("Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,"); //人差し指
 outputL.print("Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,"); //中指
 outputL.print("Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,"); //薬指
 outputL.print("Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,Mark,"); //小指
 outputL.println("Mark,Mark,Mark"); //重心

 ForLeapTime = 2;
}


//////////////////////////////////////////////////////
//関数：LeapMove
//概要：
//////////////////////////////////////////////////////
void LeapMove(){

  SaveData =0;
  long fps = leap.getTimestamp();
//  println(fps,FpsForNoCall); //青木
  if(ForLeapTime == 0){
    fps1 = leap.getTimestamp();
    ForLeapTime =1;
  }
  fill(0);
if(leap.getTimestamp()==0){}

else{
  text("計測開始から"+nf((fps-fps1)/1000000,1,0)+"秒",10,480);
}
if(ForLeapTime == 2){
  text("目印をつけた時間"+nf((fps2-fps1)/1000000,1,0)+"秒",450,480);
}


if(RightCount ==6){println(3); //青木
}
else{ //println(4); //青木
 if(fps == FpsForNoCall){ //println(fps,FpsForNoCall); //青木
 }
 else{
println(6); //青木
  // ========= HANDS =========
  for (Hand hand : leap.getHands ()) { //^^e2^^91^^a0
```

```
    // ----- BASICS -----
    int     hand_id          = hand.getId();
    PVector hand_position    = hand.getPosition();
    PVector hand_stabilized  = hand.getStabilizedPosition();
    PVector hand_direction   = hand.getDirection();
    PVector hand_dynamics    = hand.getDynamics();
    float   hand_roll        = hand.getRoll();
    float   hand_pitch       = hand.getPitch();
    float   hand_yaw         = hand.getYaw();
    boolean hand_is_left     = hand.isLeft();
    boolean hand_is_right    = hand.isRight();
    float   hand_grab        = hand.getGrabStrength();
    float   hand_pinch       = hand.getPinchStrength();
    float   hand_time        = hand.getTimeVisible();
    PVector sphere_position  = hand.getSpherePosition();
    float   sphere_radius    = hand.getSphereRadius();

    // ----- SPECIFIC FINGER -----
    Finger  finger_thumb     = hand.getThumb();
    Finger  finger_index     = hand.getIndexFinger();
    Finger  finger_middle    = hand.getMiddleFinger();
    Finger  finger_ring      = hand.getRingFinger();
    Finger  finger_pink      = hand.getPinkyFinger();


    // ----- DRAWING -----
       if(ChangeFingerDirection==0){
             hand.draw(false);
             println(1); //青木
       }
       else if(ChangeFingerDirection==1){
       println(2); //青木
    translate(600+ScreenRight,500+ScreenUp);    //立体の中心を画面中央に移動
    scale(-1,-1,1);

    hand.draw(false);
    // hand.drawSphere();
    scale(-1,-1,1);
    translate(-600-ScreenRight,-500-ScreenUp);    //立体の中心を画面中央に移動
  }

    // ========= ARM =========

    if (hand.hasArm()) { //^^e2^^91^^a1
      Arm     arm              = hand.getArm();

    } //^^e2^^91^^a1


    // ========= FINGERS =========

    for (Finger finger : hand.getFingers()) { //^^e2^^91^^a2
      // Alternatives:
      // hand.getOutstrechtedFingers();
      // hand.getOutstrechtedFingersByAngle();

      // ----- BASICS -----

      int     finger_id         = finger.getId();
      PVector finger_position   = finger.getPosition();
      PVector finger_stabilized = finger.getStabilizedPosition();
      PVector finger_velocity   = finger.getVelocity();
      PVector finger_direction  = finger.getDirection();
      float   finger_time       = finger.getTimeVisible();



/*

      if(hand.isLeft() ==true){ //^^e2^^91^^a5

         //文字列を保存
         outputL.print(fps+",");
      } //^^e2^^91^^a5
      */

      // ----- SPECIFIC BONE -----

      Bone    bone_distal       = finger.getDistalBone();
      // or                       finger.get("distal");
      // or                       finger.getBone(0);


      Bone    bone_intermediate = finger.getIntermediateBone();
      // or                       finger.get("intermediate");
      // or                       finger.getBone(1);

      Bone    bone_proximal     = finger.getProximalBone();
      // or                       finger.get("proximal");
      // or                       finger.getBone(2);
```

```
      Bone    bone_metacarpal   = finger.getMetacarpalBone();
      // or                       finger.get("metacarpal");
      // or                       finger.getBone(3);

//println(finger.bone.NextJoint().x);

////////////////////////////////////////
//位置座標取得
//^^e2^^91^^a0 中手骨：骨の下端
//^^e2^^91^^a1 基節骨：骨の上端・下端
//^^e2^^91^^a2 中節骨：なし
//^^e2^^91^^a3 末節骨：骨の上端・下端
////////////////////////////////////////
//Bone bone =null;


////////////////////////////////////////
//右手の計測
////////////////////////////////////////

    RightCount = RightCount + 1 ;
    if(RightCount > 5){RightCount =1;}

  //  println(RightCount);
 if(hand.isRight() ==false){ //^^e2^^91^^a4
 }//^^e2^^91^^a4

 else if(hand.isRight() ==true){ //^^e2^^91^^a4
         switch(RightCount){
           case 1:
         //文字列を保存
         outputR.print(fps+",");

        if(finger.getType()!=0){ //親指 //^^e2^^91^^a6
       //^^e2^^91^^a0 中手骨
       // bone =bone_metacarpal; //中手骨…親指以外の末端にある骨
        outputR.print(bone_metacarpal.getRawPrevJoint().x+","); //上端
        outputR.print(bone_metacarpal.getRawPrevJoint().y+","); //上端
        outputR.print(bone_metacarpal.getRawPrevJoint().z+","); //上端

        } //^^e2^^91^^a6

       //^^e2^^91^^a1 基節骨
        //bone =bone_proximal; //基節骨
        outputR.print(bone_proximal.getRawPrevJoint().x+","); //上端
        outputR.print(bone_proximal.getRawPrevJoint().y+","); //上端
        outputR.print(bone_proximal.getRawPrevJoint().z+","); //上端

        outputR.print(bone_proximal.getRawNextJoint().x+","); //下端
        outputR.print(bone_proximal.getRawNextJoint().y+","); //下端
        outputR.print(bone_proximal.getRawNextJoint().z+","); //下端

       //^^e2^^91^^a3 末節骨
        //bone =bone_distal; //基節骨
        outputR.print(bone_distal.getRawPrevJoint().x+","); //上端
        outputR.print(bone_distal.getRawPrevJoint().y+","); //上端
        outputR.print(bone_distal.getRawPrevJoint().z+","); //上端

        outputR.print(bone_distal.getRawNextJoint().x+","); //下端
        outputR.print(bone_distal.getRawNextJoint().y+","); //下端
        outputR.print(bone_distal.getRawNextJoint().z+","); //下端

           break;

           case 2:

        if(finger.getType()!=0){ //親指 //^^e2^^91^^a6
       //^^e2^^91^^a0 中手骨
        //bone =bone_metacarpal; //中手骨…親指以外の末端にある骨
        outputR.print(bone_metacarpal.getRawPrevJoint().x+","); //上端
        outputR.print(bone_metacarpal.getRawPrevJoint().y+","); //上端
        outputR.print(bone_metacarpal.getRawPrevJoint().z+","); //上端

        } //^^e2^^91^^a6

       //^^e2^^91^^a1 基節骨
        //bone =bone_proximal; //基節骨
        outputR.print(bone_proximal.getRawPrevJoint().x+","); //上端
        outputR.print(bone_proximal.getRawPrevJoint().y+","); //上端
        outputR.print(bone_proximal.getRawPrevJoint().z+","); //上端

        outputR.print(bone_proximal.getRawNextJoint().x+","); //下端
        outputR.print(bone_proximal.getRawNextJoint().y+","); //下端
        outputR.print(bone_proximal.getRawNextJoint().z+","); //下端

       //^^e2^^91^^a3 末節骨
        //bone =bone_distal; //基節骨
        outputR.print(bone_distal.getRawPrevJoint().x+","); //上端
        outputR.print(bone_distal.getRawPrevJoint().y+","); //上端
        outputR.print(bone_distal.getRawPrevJoint().z+","); //上端
```

```
 outputR.print(bone_distal.getRawNextJoint().x+","); //下端
 outputR.print(bone_distal.getRawNextJoint().y+","); //下端
 outputR.print(bone_distal.getRawNextJoint().z+","); //下端

    break;

    case 3:

 if(finger.getType()!=0){ //親指 //^^e2^^91^^a6
//^^e2^^91^^a0 中手骨
 //bone =bone_metacarpal; //中手骨…親指以外の末端にある骨
 outputR.print(bone_metacarpal.getRawPrevJoint().x+","); //上端
 outputR.print(bone_metacarpal.getRawPrevJoint().y+","); //上端
 outputR.print(bone_metacarpal.getRawPrevJoint().z+","); //上端

 } //^^e2^^91^^a6

//^^e2^^91^^a1 基節骨
// bone =bone_proximal; //基節骨
 outputR.print(bone_proximal.getRawPrevJoint().x+","); //上端
 outputR.print(bone_proximal.getRawPrevJoint().y+","); //上端
 outputR.print(bone_proximal.getRawPrevJoint().z+","); //上端

 outputR.print(bone_proximal.getRawNextJoint().x+","); //下端
 outputR.print(bone_proximal.getRawNextJoint().y+","); //下端
 outputR.print(bone_proximal.getRawNextJoint().z+","); //下端

//^^e2^^91^^a3 末節骨
 //bone =bone_distal; //基節骨
 outputR.print(bone_distal.getRawPrevJoint().x+","); //上端
 outputR.print(bone_distal.getRawPrevJoint().y+","); //上端
 outputR.print(bone_distal.getRawPrevJoint().z+","); //上端

 outputR.print(bone_distal.getRawNextJoint().x+","); //下端
 outputR.print(bone_distal.getRawNextJoint().y+","); //下端
 outputR.print(bone_distal.getRawNextJoint().z+","); //下端

    break;

    case 4:

 if(finger.getType()!=0){ //親指 //^^e2^^91^^a6
//^^e2^^91^^a0 中手骨
 //bone =bone_metacarpal; //中手骨…親指以外の末端にある骨
 outputR.print(bone_metacarpal.getRawPrevJoint().x+","); //上端
 outputR.print(bone_metacarpal.getRawPrevJoint().y+","); //上端
 outputR.print(bone_metacarpal.getRawPrevJoint().z+","); //上端

 } //^^e2^^91^^a6

//^^e2^^91^^a1 基節骨
 //bone =bone_proximal; //基節骨
 outputR.print(bone_proximal.getRawPrevJoint().x+","); //上端
 outputR.print(bone_proximal.getRawPrevJoint().y+","); //上端
 outputR.print(bone_proximal.getRawPrevJoint().z+","); //上端

 outputR.print(bone_proximal.getRawNextJoint().x+","); //下端
 outputR.print(bone_proximal.getRawNextJoint().y+","); //下端
 outputR.print(bone_proximal.getRawNextJoint().z+","); //下端

//^^e2^^91^^a3 末節骨
 //bone =bone_distal; //基節骨
 outputR.print(bone_distal.getRawPrevJoint().x+","); //上端
 outputR.print(bone_distal.getRawPrevJoint().y+","); //上端
 outputR.print(bone_distal.getRawPrevJoint().z+","); //上端

 outputR.print(bone_distal.getRawNextJoint().x+","); //下端
 outputR.print(bone_distal.getRawNextJoint().y+","); //下端
 outputR.print(bone_distal.getRawNextJoint().z+","); //下端

    break;

    case 5:

 if(finger.getType()!=0){ //親指 //^^e2^^91^^a6
//^^e2^^91^^a0 中手骨
// bone =bone_metacarpal; //中手骨…親指以外の末端にある骨
 outputR.print(bone_metacarpal.getRawPrevJoint().x+","); //上端
 outputR.print(bone_metacarpal.getRawPrevJoint().y+","); //上端
 outputR.print(bone_metacarpal.getRawPrevJoint().z+","); //上端

 } //^^e2^^91^^a6

//^^e2^^91^^a1 基節骨
 //bone =bone_proximal; //基節骨
 outputR.print(bone_proximal.getRawPrevJoint().x+","); //上端
 outputR.print(bone_proximal.getRawPrevJoint().y+","); //上端
 outputR.print(bone_proximal.getRawPrevJoint().z+","); //上端

 outputR.print(bone_proximal.getRawNextJoint().x+","); //下端
 outputR.print(bone_proximal.getRawNextJoint().y+","); //下端
```

```
        outputR.print(bone_proximal.getRawNextJoint().z+","); //下端

       //^^e2^^91^^a3 末節骨
       // bone =bone_distal; //基節骨
        outputR.print(bone_distal.getRawPrevJoint().x+","); //上端
        outputR.print(bone_distal.getRawPrevJoint().y+","); //上端
        outputR.print(bone_distal.getRawPrevJoint().z+","); //上端

        outputR.print(bone_distal.getRawNextJoint().x+","); //下端
        outputR.print(bone_distal.getRawNextJoint().y+","); //下端
        outputR.print(bone_distal.getRawNextJoint().z+","); //下端

        //^^e2^^91^^a4 重心
        outputR.print(hand.getRawPosition().x+",");
        outputR.print(hand.getRawPosition().y+",");
        outputR.println(hand.getRawPosition().z+",");

        NoCallR =1;
           break;
         }

      } //^^e2^^91^^a4



//////////////////////////////////////////
//左手の計測
//////////////////////////////////////////

    LeftCount = LeftCount + 1 ;
    if(LeftCount > 5){LeftCount =1;}

//    println(RightCount);
 if(hand.isLeft() ==false){ //^^e2^^91^^a4
 }//^^e2^^91^^a4

 else if(hand.isLeft() ==true){ //^^e2^^91^^a4
         switch(LeftCount){
           case 1:
         //文字列を保存
         outputL.print(fps+",");

        if(finger.getType()!=0){ //親指 //^^e2^^91^^a6
       //^^e2^^91^^a0 中手骨
        //bone =bone_metacarpal; //中手骨…親指以外の末端にある骨
        outputL.print(bone_metacarpal.getRawPrevJoint().x+","); //上端
        outputL.print(bone_metacarpal.getRawPrevJoint().y+","); //上端
        outputL.print(bone_metacarpal.getRawPrevJoint().z+","); //上端

        } //^^e2^^91^^a6

       //^^e2^^91^^a1 基節骨
        //bone =bone_proximal; //基節骨
        outputL.print(bone_proximal.getRawPrevJoint().x+","); //上端
        outputL.print(bone_proximal.getRawPrevJoint().y+","); //上端
        outputL.print(bone_proximal.getRawPrevJoint().z+","); //上端

        outputL.print(bone_proximal.getRawNextJoint().x+","); //下端
        outputL.print(bone_proximal.getRawNextJoint().y+","); //下端
        outputL.print(bone_proximal.getRawNextJoint().z+","); //下端

       //^^e2^^91^^a3 末節骨
        //bone =bone_distal; //基節骨
        outputL.print(bone_distal.getRawPrevJoint().x+","); //上端
        outputL.print(bone_distal.getRawPrevJoint().y+","); //上端
        outputL.print(bone_distal.getRawPrevJoint().z+","); //上端

        outputL.print(bone_distal.getRawNextJoint().x+","); //下端
        outputL.print(bone_distal.getRawNextJoint().y+","); //下端
        outputL.print(bone_distal.getRawNextJoint().z+","); //下端

           break;

           case 2:

        if(finger.getType()!=0){ //親指 //^^e2^^91^^a6
       //^^e2^^91^^a0 中手骨
        //bone =bone_metacarpal; //中手骨…親指以外の末端にある骨
        outputL.print(bone_metacarpal.getRawPrevJoint().x+","); //上端
        outputL.print(bone_metacarpal.getRawPrevJoint().y+","); //上端
        outputL.print(bone_metacarpal.getRawPrevJoint().z+","); //上端

        } //^^e2^^91^^a6

       //^^e2^^91^^a1 基節骨
        //bone =bone_proximal; //基節骨
        outputL.print(bone_proximal.getRawPrevJoint().x+","); //上端
        outputL.print(bone_proximal.getRawPrevJoint().y+","); //上端
        outputL.print(bone_proximal.getRawPrevJoint().z+","); //上端
```

```
 outputL.print(bone_proximal.getRawNextJoint().x+","); //下端
 outputL.print(bone_proximal.getRawNextJoint().y+","); //下端
 outputL.print(bone_proximal.getRawNextJoint().z+","); //下端

//^^e2^^91^^a3 末節骨
 //bone =bone_distal; //基節骨
 outputL.print(bone_distal.getRawPrevJoint().x+","); //上端
 outputL.print(bone_distal.getRawPrevJoint().y+","); //上端
 outputL.print(bone_distal.getRawPrevJoint().z+","); //上端

 outputL.print(bone_distal.getRawNextJoint().x+","); //下端
 outputL.print(bone_distal.getRawNextJoint().y+","); //下端
 outputL.print(bone_distal.getRawNextJoint().z+","); //下端

    break;

    case 3:

 if(finger.getType()!=0){ //親指 //^^e2^^91^^a6
//^^e2^^91^^a0 中手骨
 //bone =bone_metacarpal; //中手骨…親指以外の末端にある骨
 outputL.print(bone_metacarpal.getRawPrevJoint().x+","); //上端
 outputL.print(bone_metacarpal.getRawPrevJoint().y+","); //上端
 outputL.print(bone_metacarpal.getRawPrevJoint().z+","); //上端

 } //^^e2^^91^^a6

//^^e2^^91^^a1 基節骨
 //bone =bone_proximal; //基節骨
 outputL.print(bone_proximal.getRawPrevJoint().x+","); //上端
 outputL.print(bone_proximal.getRawPrevJoint().y+","); //上端
 outputL.print(bone_proximal.getRawPrevJoint().z+","); //上端

 outputL.print(bone_proximal.getRawNextJoint().x+","); //下端
 outputL.print(bone_proximal.getRawNextJoint().y+","); //下端
 outputL.print(bone_proximal.getRawNextJoint().z+","); //下端

//^^e2^^91^^a3 末節骨
 //bone =bone_distal; //基節骨
 outputL.print(bone_distal.getRawPrevJoint().x+","); //上端
 outputL.print(bone_distal.getRawPrevJoint().y+","); //上端
 outputL.print(bone_distal.getRawPrevJoint().z+","); //上端

 outputL.print(bone_distal.getRawNextJoint().x+","); //下端
 outputL.print(bone_distal.getRawNextJoint().y+","); //下端
 outputL.print(bone_distal.getRawNextJoint().z+","); //下端

    break;

    case 4:

 if(finger.getType()!=0){ //親指 //^^e2^^91^^a6
//^^e2^^91^^a0 中手骨
// bone =bone_metacarpal; //中手骨…親指以外の末端にある骨
 outputL.print(bone_metacarpal.getRawPrevJoint().x+","); //上端
 outputL.print(bone_metacarpal.getRawPrevJoint().y+","); //上端
 outputL.print(bone_metacarpal.getRawPrevJoint().z+","); //上端

 } //^^e2^^91^^a6

//^^e2^^91^^a1 基節骨
 //bone =bone_proximal; //基節骨
 outputL.print(bone_proximal.getRawPrevJoint().x+","); //上端
 outputL.print(bone_proximal.getRawPrevJoint().y+","); //上端
 outputL.print(bone_proximal.getRawPrevJoint().z+","); //上端

 outputL.print(bone_proximal.getRawNextJoint().x+","); //下端
 outputL.print(bone_proximal.getRawNextJoint().y+","); //下端
 outputL.print(bone_proximal.getRawNextJoint().z+","); //下端

//^^e2^^91^^a3 末節骨
// bone =bone_distal; //基節骨
 outputL.print(bone_distal.getRawPrevJoint().x+","); //上端
 outputL.print(bone_distal.getRawPrevJoint().y+","); //上端
 outputL.print(bone_distal.getRawPrevJoint().z+","); //上端

 outputL.print(bone_distal.getRawNextJoint().x+","); //下端
 outputL.print(bone_distal.getRawNextJoint().y+","); //下端
 outputL.print(bone_distal.getRawNextJoint().z+","); //下端

    break;

    case 5:

 if(finger.getType()!=0){ //親指 //^^e2^^91^^a6
//^^e2^^91^^a0 中手骨
 //bone =bone_metacarpal; //中手骨…親指以外の末端にある骨
 outputL.print(bone_metacarpal.getRawPrevJoint().x+","); //上端
 outputL.print(bone_metacarpal.getRawPrevJoint().y+","); //上端
 outputL.print(bone_metacarpal.getRawPrevJoint().z+","); //上端
```

```
        } //^^e2^^91^^a6

       //^^e2^^91^^a1 基節骨
        //bone =bone_proximal; //基節骨
        outputL.print(bone_proximal.getRawPrevJoint().x+","); //上端
        outputL.print(bone_proximal.getRawPrevJoint().y+","); //上端
        outputL.print(bone_proximal.getRawPrevJoint().z+","); //上端

        outputL.print(bone_proximal.getRawNextJoint().x+","); //下端
        outputL.print(bone_proximal.getRawNextJoint().y+","); //下端
        outputL.print(bone_proximal.getRawNextJoint().z+","); //下端

       //^^e2^^91^^a3 末節骨
        //bone =bone_distal; //基節骨
        outputL.print(bone_distal.getRawPrevJoint().x+","); //上端
        outputL.print(bone_distal.getRawPrevJoint().y+","); //上端
        outputL.print(bone_distal.getRawPrevJoint().z+","); //上端

        outputL.print(bone_distal.getRawNextJoint().x+","); //下端
        outputL.print(bone_distal.getRawNextJoint().y+","); //下端
        outputL.print(bone_distal.getRawNextJoint().z+","); //下端

        //^^e2^^91^^a4 重心
        outputL.print(hand.getRawPosition().x+",");
        outputL.print(hand.getRawPosition().y+",");
        outputL.println(hand.getRawPosition().z+",");

        NoCallL =1;

           break;
         }

      } //^^e2^^91^^a4



    } //^^e2^^91^^a2


  } //^^e2^^91^^a0




//  outputR.println(kakunin);

}
  if(NoCallR ==0){
if(fps==0){}
else{

  outputR.print(fps+","); //時間
 outputR.print("Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,"); //親指
 outputR.print("Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,"); //
人差し指
 outputR.print("Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,"); //
中指
 outputR.print("Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,"); //
薬指
 outputR.print("Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,"); //
小指
 outputR.println("Invalid,Invalid,Invalid"); //重心
}//5/12   fps=0 になるバグ修正のため
  }
 else if(NoCallR ==1){}
   NoCallR =0;

  if(NoCallL ==0){
if(fps==0){}//5/12   fps=0 になるバグ修正のため
else{//5/12   fps=0 になるバグ修正のため
 outputL.print(fps+","); //時間
 outputL.print("Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,"); //親指
 outputL.print("Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,"); //
人差し指
 outputL.print("Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,"); //
中指
 outputL.print("Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,"); //
薬指
 outputL.print("Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,Invalid,"); //
小指
 outputL.println("Invalid,Invalid,Invalid"); //重心
 }//5/12   fps=0 になるバグ修正のため
  }
 else if(NoCallL ==1){}
 NoCallL =0;
}
```

```
  long FpsForNoCall = leap.getTimestamp();

}



//////////////////////////////////////////////////////
//関数：radio
//概要：ツールバーにある「２．表示」でアニメーション，変位グラフ，
//       ３Ｄグラフに関する７種類の表示画面を実行する条件分岐をする．
//関連する関数
//     ・CreateToolBar<<ツールバーの作成>>
//関連する文字
//      ・selectButton
//      ・xRotate,yRoatte,zRotate
//      ・q
//      ・i,j
//      ・Rightspeed,Leftspeed
//      ・sl1.setValue(),sl2.setValue()
//      ・x_moveA,y_moveA
//      ・x_moveQ,y_moveQ
//      ・x_moveC,y_moveC
//      ・sl4.getValue(),sl5.getValue(),sl6.getValue()
//////////////////////////////////////////////////////
void radio(int theD)
{
  switch(theD)
  {
    //１．全て
    case(6):
      selectButton = 6;
      x_moveA = -150;
      y_moveA = 0;
      x_moveQ = 150;
      y_moveQ = 0;
      x_moveC = -150;
      y_moveC = 0;
    break;
    //２．アニメーション
    case(0):
      x_moveA = 0;
      y_moveA = 0;
      selectButton = 1;
//      xRotate = PI/2;
//      yRotate = -PI/2;
//      zRotate = 0;
      sl4.setValue(0);
      sl5.setValue(0);
      sl6.setValue(0);
      q=1;
    break;
    //３．変位グラフ
    case(1):
      selectButton = 2;
//      xRotate = PI/2;
//      yRotate = -PI/2;
//      zRotate = 0;
      sl4.setValue(0);
      sl5.setValue(0);
      sl6.setValue(0);
      x_moveQ = 0;
      y_moveQ = 0;
  //    i=2;
  //    j=2;
  //    Rightspeed=0;
  //    Leftspeed=0;
      q=0;
  //    sl1.setValue(0);
 //     sl2.setValue(0);
    break;
    //４．３Ｄグラフ
    case(2):
      selectButton = 3;
      x_moveC =500;
      y_moveC =0;
//      xRotate = -PI*3/2;
//      yRotate = PI;
//      zRotate = -PI*3/2;
      sl4.setValue(0);
      sl5.setValue(0);
      sl6.setValue(0);
      i=LENGTHF-2;
      j=LENGTHS-2;
      i2=LENGTHT-2;
      j2=LENGTHH-2;
      Rightspeed=0;
      Leftspeed=0;
      Rightspeed2=0;
      Leftspeed2=0;
      q=0;
      sl1.setValue(0);
```

```
      sl2.setValue(0);
      sl7.setValue(0);
      sl8.setValue(0);
    break;
    //５．アニメーションと変位グラフ
    case(3):
      selectButton = 5;
      x_moveA = -150;
      y_moveA = 0;
      x_moveQ = 150;
      y_moveQ = 0;
    break;
    //６．アニメーションと３Ｄグラフ
    case(4):
      selectButton = 4;
      x_moveA =0;
      y_moveA =0;
      q=1;
      i=2;
      j=2;
      Rightspeed=0;
      Leftspeed=0;
      Rightspeed2=0;
      Leftspeed2=0;
      sl1.setValue(0);
      sl2.setValue(0);
      sl7.setValue(0);
      sl8.setValue(0);
      x_moveC =500;
      y_moveC =0;
//      xRotate = -PI*3/2;
//      yRotate = PI;
//      zRotate = -PI*3/2;
      sl4.setValue(0);
      sl5.setValue(0);
      sl6.setValue(0);
    break;
  }
}

////////////////////////////////////////////
//関数名：controlEvent
//概要：「１．ファイル」の左右のデータを選択する
//関連する関数
//      ・fileSelectedR<<右手データ１を編集>>
//      ・fileSelectedL<<左手データ１を編集>>
//      ・fileSelectedNextR<<右手データ２を編集>>
//      ・fileSelectedNextL<<左手データ２を編集>>
//できていないこと：ボタンがうまくかわらない.
////////////////////////////////////////////
void controlEvent(ControlEvent theEvent)
{
  if (theEvent.isFrom("SelectData"))
  {
    for(int i=0; i<4; i++)
    {

      if(cb.getArrayValue(i) == 1.0)
      {
        if(i == 0)
        {
                //データの確認
      JPanel panel = new JPanel();    //パネルを作成
      BoxLayout layout = new BoxLayout( panel, BoxLayout.Y_AXIS );    //メッセージのレイアウトを決定
      panel.setLayout(layout);    //panel に layout を適用
      panel.add( new JLabel( "上から計測したデータですか？" ) );    //メッセージ内容を文字列のコンポーネントとしてパネルに追加

      DataFromAbove = JOptionPane.showConfirmDialog(
      null,    //親フレームの指定
      panel,    //パネルの指定
      "hogehoge",    //タイトルバーに表示する内容
      JOptionPane.YES_NO_OPTION,    //オプションタイプを YES,NO にする
      JOptionPane.INFORMATION_MESSAGE   //メッセージタイプを Information にする
      );
          selectInput("右手データ１を選択してください", "fileSelectedR");
        }
         else if(i == 1)
        {
                //データの確認
      JPanel panel = new JPanel();    //パネルを作成
      BoxLayout layout = new BoxLayout( panel, BoxLayout.Y_AXIS );    //メッセージのレイアウトを決定
      panel.setLayout(layout);    //panel に layout を適用
      panel.add( new JLabel( "上から計測したデータですか？" ) );    //メッセージ内容を文字列のコンポーネントとしてパネルに追加

      DataFromAbove = JOptionPane.showConfirmDialog(
      null,    //親フレームの指定
      panel,    //パネルの指定
      "hogehoge",    //タイトルバーに表示する内容
      JOptionPane.YES_NO_OPTION,    //オプションタイプを YES,NO にする
      JOptionPane.INFORMATION_MESSAGE   //メッセージタイプを Information にする
      );
```

```
          selectInput("左手データ１を選択してください", "fileSelectedL");
        }
         else if(i == 2)
        {
                //データの確認
      JPanel panel = new JPanel();    //パネルを作成
      BoxLayout layout = new BoxLayout( panel, BoxLayout.Y_AXIS );    //メッセージのレイアウトを決定
      panel.setLayout(layout);    //panel に layout を適用
      panel.add( new JLabel( "上から計測したデータですか？" ) );    //メッセージ内容を文字列のコンポーネントとしてパネルに追加

      DataFromAbove = JOptionPane.showConfirmDialog(
      null,    //親フレームの指定
      panel,    //パネルの指定
      "hogehoge",    //タイトルバーに表示する内容
      JOptionPane.YES_NO_OPTION,    //オプションタイプを YES,NO にする
      JOptionPane.INFORMATION_MESSAGE   //メッセージタイプを Information にする
      );
          selectInput("右手データ２を選択してください", "fileSelectedNextR");
        }
         else if(i == 3)
        {
                //データの確認
      JPanel panel = new JPanel();    //パネルを作成
      BoxLayout layout = new BoxLayout( panel, BoxLayout.Y_AXIS );    //メッセージのレイアウトを決定
      panel.setLayout(layout);    //panel に layout を適用
      panel.add( new JLabel( "上から計測したデータですか？" ) );    //メッセージ内容を文字列のコンポーネントとしてパネルに追加

      DataFromAbove = JOptionPane.showConfirmDialog(
      null,    //親フレームの指定
      panel,    //パネルの指定
      "hogehoge",    //タイトルバーに表示する内容
      JOptionPane.YES_NO_OPTION,    //オプションタイプを YES,NO にする
      JOptionPane.INFORMATION_MESSAGE   //メッセージタイプを Information にする
      );
          selectInput("左手データ２を選択してください", "fileSelectedNextL");
        }
      }
    }
  }

}
////////////////////////////////////////////////////////////////////////
//関数名：fileSelectedR
//概要：「右手データ１」にデータを編集する.
//関連する関数
//        ・controlEvent<<データの選択>>
////////////////////////////////////////////////////////////////////////
void fileSelectedR(File selection1)
{
  if (selection1 == null)
  {
    println("Window was closed or the user hit cancel.");
  }
  else
  {
    println("User selected " + selection1.getAbsolutePath());
    loadPath1 = selection1.getAbsolutePath();
    //////CSV 形式をデータへ保存//////////////////
    String lines[] = loadStrings(loadPath1);
    ////calculate max width of csv file//////////////////
    for (int i=0; i < lines.length; i++)
    {
      String [] chars=split(lines[i],',');
    }
    ///create csv array based on # of rows and columns in csv file/////////
    csvF = new String [lines.length][csvWidth]; //文字列データ用
    csvF1 = new float [lines.length][csvWidth]; //数値データ用
    csvF2 = new float [lines.length][csvWidth]; //原点修正データ用
    LENGTHF =lines.length-1; //
    /////parse values into 2d array//////////////
    for (int i=0; i < LENGTHF; i++)
    {
      String [] temp = new String [lines.length];
      temp= split(lines[i], ',');
      for (int j=0; j < csvWidth; j++)
      {
        csvF[i][j]=temp[j];
      }
    }

    //文字データから数値データへ変換
    for (int a = 2; a < LENGTHF; a++)
    {
      for (int b = 0; b < csvWidth; b++)
      {

        if(csvF[a][b].equals("Invalid") && a<3)
        {
          csvF1[a][b] = 0;
        }
        else if(csvF[a][b].equals("Invalid") && a >=3 )
```

```
        {
          csvF1[a][b] = csvF1[a-1][b];
        }
        else if (csvF[a][b].equals("Mark"))
        {
          csvF1[a][b] = csvF1[a-1][b];
        }
        else
        {
          //計測時の様子に合わせ，データ変換
          if ( DataFromAbove == 0 )
          {
            //YES が選択された場合の処理
            if(b%3 == 0){
              //z 軸の値
              csvF1[a][b] = Float.parseFloat(csvF[a][b]);
            }
            else if(b%3 == 1){
              //x 軸の値
              csvF1[a][b] = -1*Float.parseFloat(csvF[a][b]);
            }
            else if(b%3 == 2){
              //y 軸の値
              csvF1[a][b] = 250-1*Float.parseFloat(csvF[a][b]);
            }
          }
          else if ( DataFromAbove == 1 )
          {
            //NO が選択された場合の処理
              csvF1[a][b] = Float.parseFloat(csvF[a][b]);
          }

        }
      }
    }
//原点修正
for (int a = 2; a  < LENGTHF; a++) {
  for (int b = 0; b < csvWidth; b++) {
      csvF2[a][b] = csvF1[a][b]-csvF1[2][b];
    }
  }

if(timechange < csvF2[LENGTHF-2][0]/1000000){timechange = csvF2[LENGTHF-2][0]/1000000;}

//maxF1, minF1, maxF2, minF2 の値を設定
//これをしなければ変位グラフで最大値・最小値にならない。
for(int a=0; a < csvWidth; a++){
 maxF1[a] = csvF1[2][a];
 minF1[a] = csvF1[2][a];
 maxF2[a] = csvF2[2][a];
 minF2[a] = csvF2[2][a];
}

//最大・最小値判別 (F1)
for (int a = 2; a+2 < LENGTHF; a++) {
  for (int b = 0; b < csvWidth; b++) {
  if (maxF1[b] < csvF1[a][b]) maxF1[b] = csvF1[a][b];
  if (minF1[b] > csvF1[a][b]) minF1[b] = csvF1[a][b];
  }
}
//最大・最小値判別 (F2)
for (int a = 2; a+2 < LENGTHF; a++) {
  for (int b = 0; b < csvWidth; b++) {
  if (maxF2[b] < csvF2[a][b]) maxF2[b] = csvF2[a][b];
  if (minF2[b] > csvF2[a][b]) minF2[b] = csvF2[a][b];
  }
}
//絶対値の最大・最小値から最大値判別 (F2)
for (int a = 0; a < csvWidth; a++) {
  if (abs(minF2[a]) < abs(maxF2[a]))
  maxF0[a] = abs(maxF2[a]);
  else
  maxF0[a] = abs(minF2[a]);
  }
  fitdata1 =1; //fileSelectedR にデータを読み込み完了
  SaveData =1; //データを読み込み完了
  }
}
////////////////////////////////////////////////////////////////////////
//関数名：fileSelectedL
//概要：「左手データ 1」にデータを編集する.
//関連する関数
//        ・controlEvent<<データの選択>>
////////////////////////////////////////////////////////////////////////
void fileSelectedL(File selection2) {
 if (selection2 == null) {
    println("Window was closed or the user hit cancel.");
  } else {
    println("User selected " + selection2.getAbsolutePath());
    loadPath2 = selection2.getAbsolutePath();
```

```
  //////CSV 形式をデータへ保存/////////////////
  String lines1[] = loadStrings(loadPath2);


////calculate max width of csv file/////////////////
for (int i=0; i < lines1.length; i++) {
  String [] chars=split(lines1[i],',');

}
///create csv array based on # of rows and columns in csv file/////////
csvS = new String [lines1.length][csvWidth]; //文字列データ用
csvS1 = new float [lines1.length][csvWidth]; //数値データ用
csvS2 = new float [lines1.length][csvWidth]; //原点修正データ用
LENGTHS =lines1.length-1; //
/////parse values into 2d array//////////////
for (int i=0; i < LENGTHS; i++) {
  String [] temp = new String [lines1.length];
  temp= split(lines1[i], ',');
  for (int j=0; j < csvWidth; j++){
    csvS[i][j]=temp[j];
  }
}

//文字データから数値データへ変換
for (int a = 2; a < LENGTHS; a++) {
  for (int b = 0; b < csvWidth; b++) {

  if(csvS[a][b].equals("Invalid") && a<3){
      csvS1[a][b] = 0;
    }
    else if(csvS[a][b].equals("Invalid") && a >=3 ){
      csvS1[a][b] = csvS1[a-1][b];
    }
   else if (csvS[a][b].equals("Mark"))
        {
          csvS1[a][b] = csvS1[a-1][b];
        }
    else{
          //計測時の様子に合わせ，データ変換
          if ( DataFromAbove == 0 )
          {
            //YES が選択された場合の処理
            if(b%3 == 0){
              //z 軸の値
              csvS1[a][b] = Float.parseFloat(csvS[a][b]);
            }
            else if(b%3 == 1){
              //x 軸の値
              csvS1[a][b] = -1*Float.parseFloat(csvS[a][b]);
            }
            else if(b%3 == 2){
              //y 軸の値
              csvS1[a][b] = 250-1*Float.parseFloat(csvS[a][b]);
            }
          }
          else if ( DataFromAbove == 1 )
          {
            //NO が選択された場合の処理
              csvS1[a][b] = Float.parseFloat(csvS[a][b]);
          }
        }
  }
}
//原点修正
for (int a = 2; a  < LENGTHS; a++) {
  for (int b = 0; b < csvWidth; b++) {
    csvS2[a][b] = csvS1[a][b]-csvS1[2][b];

  }
}

if(timechange < csvS2[LENGTHS-2][0]/1000000){timechange = csvS2[LENGTHS-2][0]/1000000;}

//maxS1, minS1, maxS2, minS2 の値を設定
//これをしなければ変位グラフで最大値・最小値にならない。
for(int a=0; a < csvWidth; a++){
 maxS1[a] = csvS1[2][a];
 minS1[a] = csvS1[2][a];
 maxS2[a] = csvS2[2][a];
 minS2[a] = csvS2[2][a];
}

//最大・最小値判別 (S1)
for (int a = 2; a+2 < LENGTHS; a++) {
  for (int b = 0; b < csvWidth; b++) {
  if (maxS1[b] < csvS1[a][b]) maxS1[b] = csvS1[a][b];
  if (minS1[b] > csvS1[a][b]) minS1[b] = csvS1[a][b];
  }
}
//最大・最小値判別 (S2)
for (int a = 2; a+2 < LENGTHS; a++) {
```

```
  for (int b = 0; b < csvWidth; b++) {
  if (maxS2[b] < csvS2[a][b]) maxS2[b] = csvS2[a][b];
  if (minS2[b] > csvS2[a][b]) minS2[b] = csvS2[a][b];
  }
}
//絶対値の最大・最小値から最大値判別
for (int a = 0; a < csvWidth; a++) {
  if (abs(minS2[a]) < abs(maxS2[a]))
  maxS0[a] = abs(maxS2[a]);
  else
  maxS0[a] = abs(minS2[a]);
  }
  fitdata2 =1; //fileSelectedL にデータを保存完了
  SaveData =1; //データを読み込み完了
  }
}
////////////////////////////////////////////////////////////////////////
//関数名：fileSelectedNextR
//概要：「右手データ２」にデータを編集する.
//関連する関数
//       ・controlEvent<<データの選択>>
//出力するデータ
//    fitdata3
////////////////////////////////////////////////////////////////////////
void fileSelectedNextR(File selection3) {
 if (selection3 == null) {
    println("Window was closed or the user hit cancel.");
  } else {
    println("User selected " + selection3.getAbsolutePath());
    loadPath3 = selection3.getAbsolutePath();
  //////CSV 形式をデータへ保存/////////////////
  String lines3[] = loadStrings(loadPath3);
////calculate max width of csv file/////////////////
for (int i=0; i < lines3.length; i++) {
  String [] chars=split(lines3[i],',');

}
///create csv array based on # of rows and columns in csv file////////
csvT = new String [lines3.length][csvWidth]; //文字列データ用
csvT1 = new float [lines3.length][csvWidth]; //数値データ用
csvT2 = new float [lines3.length][csvWidth]; //原点修正データ用
LENGTHT =lines3.length-1; //
/////parse values into 2d array//////////////
for (int i=0; i < LENGTHT; i++) {
  String [] temp = new String [lines3.length];
  temp= split(lines3[i], ',');
  for (int j=0; j < csvWidth; j++){
    csvT[i][j]=temp[j];
  }
}

//文字データから数値データへ変換
for (int a = 2; a < LENGTHT; a++) {
  for (int b = 0; b < csvWidth; b++) {

 if(csvT[a][b].equals("Invalid") && a<3){
      csvT1[a][b] = 0;
    }

    else if(csvT[a][b].equals("Invalid") && a >=3 ){
      csvT1[a][b] = csvT1[a-1][b];
    }
  else if (csvT[a][b].equals("Mark"))
        {
          csvT1[a][b] = csvT1[a-1][b];
        }
    else{
          //計測時の様子に合わせ，データ変換
          if ( DataFromAbove == 0 )
          {
            //YES が選択された場合の処理
            if(b%3 == 0){
              //z 軸の値
              csvT1[a][b] = Float.parseFloat(csvT[a][b]);
            }
            else if(b%3 == 1){
              //x 軸の値
              csvT1[a][b] = -1*Float.parseFloat(csvT[a][b]);
            }
            else if(b%3 == 2){
              //y 軸の値
              csvT1[a][b] = 250-1*Float.parseFloat(csvT[a][b]);
            }
          }
          else if ( DataFromAbove == 1 )
          {
            //NO が選択された場合の処理
              csvT1[a][b] = Float.parseFloat(csvT[a][b]);
          }
        }
  }
```

```
}

//原点修正
for (int a = 2; a  < LENGTHT; a++) {
  for (int b = 0; b < csvWidth; b++) {
    csvT2[a][b] = csvT1[a][b]-csvT1[2][b];
  }
}
if(timechange < csvT2[LENGTHT-2][0]/1000000){timechange = csvT2[LENGTHT-2][0]/1000000;}

//maxT1, minT1, maxT2, minT2 の値を設定
//これをしなければ変位グラフで最大値・最小値にならない。
for(int a=0; a < csvWidth; a++){
 maxT1[a] = csvT1[2][a];
 minT1[a] = csvT1[2][a];
 maxT2[a] = csvT2[2][a];
 minT2[a] = csvT2[2][a];
}

//最大・最小値判別 (T1)
for (int a = 2; a+2 < LENGTHT; a++) {
  for (int b = 0; b < csvWidth; b++) {
  if (maxT1[b] < csvT1[a][b]) maxT1[b] = csvT1[a][b];
  if (minT1[b] > csvT1[a][b]) minT1[b] = csvT1[a][b];
  }
}
//最大・最小値判別 (T2)
for (int a = 2; a+2 < LENGTHT; a++) {
  for (int b = 0; b < csvWidth; b++) {
  if (maxT2[b] < csvT2[a][b]) maxT2[b] = csvT2[a][b];
  if (minT2[b] > csvT2[a][b]) minT2[b] = csvT2[a][b];
  }
}
//絶対値の最大・最小値から最大値判別 (T2)
for (int a = 0; a < csvWidth; a++) {
  if (abs(minT2[a]) < abs(maxT2[a]))
  maxT0[a] = abs(maxT2[a]);
  else
  maxT0[a] = abs(minT2[a]);
  }
  fitdata3 =1; //fileSelectedNextR にデータを保存完了
  SaveData =1; //データを読み込み完了
  }
}
////////////////////////////////////////////////////////////////////////
//関数名：fileSelectedNextL
//概要：「左手データ２」にデータを選択する.
//関連する関数
//    ・controlEvent<<データの選択>>
//出力するデータ
//    fitdata4
////////////////////////////////////////////////////////////////////////
void fileSelectedNextL(File selection4) {
 if (selection4 == null) {
    println("Window was closed or the user hit cancel.");
  } else {
    println("User selected " + selection4.getAbsolutePath());
    loadPath4 = selection4.getAbsolutePath();

  //////CSV 形式をデータへ保存/////////////////
  String lines4[] = loadStrings(loadPath4);

////calculate max width of csv file/////////////////
for (int i=0; i < lines4.length; i++) {
  String [] chars=split(lines4[i],',');

}
///create csv array based on # of rows and columns in csv file/////////
csvH = new String [lines4.length][csvWidth]; //文字列データ用
csvH1 = new float [lines4.length][csvWidth]; //数値データ用
csvH2 = new float [lines4.length][csvWidth]; //原点修正データ用
LENGTHH =lines4.length-1; //
/////parse values into 2d array//////////////
for (int i=0; i < LENGTHH; i++) {
  String [] temp = new String [lines4.length];
  temp= split(lines4[i], ',');
  for (int j=0; j < csvWidth; j++){
    csvH[i][j]=temp[j];
  }
}

//文字データから数値データへ変換
for (int a = 2; a < LENGTHH; a++) {
  for (int b = 0; b < csvWidth; b++) {
   if(csvH[a][b].equals("Invalid") && a < 3){
      csvH1[a][b] = csvH1[a+1][b];
    }
    else if(csvH[a][b].equals("Invalid") && a >=3 ){
      csvH1[a][b] = csvH1[a-1][b];
    }
    else if (csvH[a][b].equals("Mark"))
```

```
        {
          csvH1[a][b] = csvH1[a-1][b];
        }
    else{
          //計測時の様子に合わせ，データ変換
          if ( DataFromAbove == 0 )
          {
            //YES が選択された場合の処理
            if(b%3 == 0){
              //z 軸の値
              csvH1[a][b] = Float.parseFloat(csvH[a][b]);
            }
            else if(b%3 == 1){
              //x 軸の値
              csvH1[a][b] = -1*Float.parseFloat(csvH[a][b]);
            }
            else if(b%3 == 2){
              //y 軸の値
              csvH1[a][b] = 250-1*Float.parseFloat(csvH[a][b]);
            }
          }
          else if ( DataFromAbove == 1 )
          {
            //NO が選択された場合の処理
              csvH1[a][b] = Float.parseFloat(csvH[a][b]);
          }
        }
  }
}
//原点修正
for (int a = 2; a  < LENGTHH; a++) {
  for (int b = 0; b < csvWidth; b++) {
    csvH2[a][b] = csvH1[a][b]-csvH1[2][b];
  }
}
if(timechange < csvH2[LENGTHH-2][0]/1000000){timechange = csvH2[LENGTHH-2][0]/1000000;}

//maxH1, minH1, maxH2, minH2 の値を設定
//これをしなければ変位グラフで最大値・最小値にならない。
for(int a=0; a < csvWidth; a++){
 maxH1[a] = csvH1[2][a];
 minH1[a] = csvH1[2][a];
 maxH2[a] = csvH2[2][a];
 minH2[a] = csvH2[2][a];
}

//最大・最小値判別 (H1)
for (int a = 2; a+2 < LENGTHH; a++) {
  for (int b = 0; b < csvWidth; b++) {
  if (maxH1[b] < csvH1[a][b]) maxH1[b] = csvH1[a][b];
  if (minH1[b] > csvH1[a][b]) minH1[b] = csvH1[a][b];
  }
}
//最大・最小値判別 (H2)
for (int a = 2; a+2 < LENGTHH; a++) {
  for (int b = 0; b < csvWidth; b++) {
  if (maxH2[b] < csvH2[a][b]) maxH2[b] = csvH2[a][b];
  if (minH2[b] > csvH2[a][b]) minH2[b] = csvH2[a][b];
  }
}
//絶対値の最大・最小値から最大値判別 (H2)
for (int a = 0; a < csvWidth; a++) {
  if (abs(minH2[a]) < abs(maxH2[a]))
  maxH0[a] = abs(maxH2[a]);
  else
  maxH0[a] = abs(minH2[a]);
  }
  fitdata4 = 1; //fileSelecetdL にデータを保存完了
  SaveData =1; //データを読み込み完了
  }
}


////////////////////////////////////////
//メインループ
//関連する関数
//play を調度よい位置へ or 不必要な状態にしたい。
//関連する関数
//  ・DrawHands<<両手のアニメーション>>
//  ・DrawQuadratic_function<<二次関数のグラフ>>
//  ・DrawCubic_function<<三次関数のグラフ>>
//  ・SpeedUpdate<<両手の速度を更新する>>
////////////////////////////////////////
void draw(){
  background(255,255,255);
  long fps = leap.getTimestamp();
  println(fps,FpsForNoCall); //青木


  if(SaveData == 1){
    if(selectButton == 1){
```

```
      play=2;
      DrawHands(); //両手のアニメーション
    }
    else if(selectButton == 2){
      play=2;
      DrawQuadratic_function(); //二次関数のグラフ
    }
    else if(selectButton == 3){
      play=2;
      DrawCubic_function(); //三次関数のグラフ
    }
    else if(selectButton == 4){ //アニメーションと３Ｄグラフ
      play=2;
      DrawHands(); //両手のアニメーション
      DrawCubic_function(); //三次関数のグラフ
    }
    else if(selectButton == 5){ //アニメーションと二次関数のグラフ
      play=2;
      DrawHands(); //両手のアニメーション
      DrawQuadratic_function(); //二次関数のグラフ
    }
    else if(selectButton == 6){ //すべて
      play=2;
      DrawHands(); //両手のアニメーション
      DrawQuadratic_function(); //二次関数のグラフ
      DrawCubic_function(); //三次関数のグラフ

    }
      SpeedUpdate(); //両手の速度を更新する
    }
    else{
    }
    if(FornameR == 1){
      if(FornameL ==1){

   LeapMove();
      }
    }
  //  println(x,y,e);

//println(xRotate,yRotate,zRotate);

}

////////////////////////////////////////////
//関数名：SpeedUpdate
//概要：両手の速度を更新する。
////////////////////////////////////////////
void SpeedUpdate(){
    i=i+Rightspeed; //右手の速度調整
   j=j+Leftspeed; //左手の速度調整
   i2=i2+Rightspeed2; //右手の速度調整
   j2=j2+Leftspeed2; //左手の速度調整

   noFill();
   stroke(0);
    weight =0.5;
    strokeWeight(weight);
    rect(20,450,560,40);
   line(20,460,580,460);
   line(20,470,580,470);
   line(20,480,580,480);
   Progress();

}
////////////////////////////////////////////
//関数名：Progress
//概要：両手のアニメーションにおける再生の状況を
//       表示する。
//関連する関数
//  ・For_Finite_Animation<<ループの禁止>>
////////////////////////////////////////////
void Progress()
{
  For_Finite_Animation(); //ループの禁止
  fill(0);
  stroke(0);
  text("0",10,445);
  text(nf(timechange,1,1),550,445);
  text(" (秒) ",570,445);
  text(" (秒) ",15,445);
  text("右 1",0,460);
  text("左 1",0,470);
  text("右 2",0,480);
  text("左 2",0,490);


    if(fitdata1 ==1)
    {
    fill(0);
    rect(20,450,((560*csvF2[i][0])/(1000000*timechange)),10);
```

```
    fill(0);
    text(nf(csvF2[i][0]/1000000,1,1),30+(560*csvF2[i][0])/(1000000*timechange),460);
    }
    if(fitdata2 ==1)
    {
    fill(255,255,0);
    rect(20,460,((560*csvS2[j][0])/(1000000*timechange)),10);
    fill(0);
    text(nf(csvS2[j][0]/1000000,1,1),30+(560*csvS2[j][0])/(1000000*timechange),470);
    }
    if(fitdata3 ==1)
    {
    fill(255,255,255);
    rect(20,470,((560*csvT2[i2][0])/(1000000*timechange)),10);
    fill(0);
    text(nf(csvT2[i2][0]/1000000,1,1),30+(560*csvT2[i2][0])/(1000000*timechange),480);
    }
    if(fitdata4 ==1)
    {
    fill(0,0,255);
    rect(20,480,((560*csvH2[j2][0])/(1000000*timechange)),10);
    fill(0);
    text(nf(csvH2[j2][0]/1000000,1,1),30+(560*csvH2[j2][0])/(1000000*timechange),490);
    }
  }

////////////////////////////////////////////
//関数名：DrawHands
//概要：保存した両手の座標からアニメーションを動
//      かす。
//関連する関数
//  ・RightAni<<右手のアニメーション>>
//  ・LeftAni<<左手のアニメーション>>
////////////////////////////////////////////
void DrawHands(){
  selectLiAJo=1;
  if(fitdata1 ==1){
    RightAni(); //右手のアニメーション
  }
  if(fitdata2 ==1){
    LeftAni(); //左手のアニメーション
  }
   if(fitdata3 ==1){
    AnotherRightAni();
  }
   if(fitdata4 ==1){
     AnotherLeftAni();
   }
}

////////////////////////////////////////////
//関数名：RightAni
//概要：保存した右手の座標からアニメーションを動
//      かす。
//関連する関数
//  ・draw3DLineAndJoint
//  ・JudgeColorRight<<右手の配色>>
////////////////////////////////////////////
void RightAni(){
   JudgeColor_Right(); //右手の配色
  if(selectButton == 1){
   r1r=1;
   //右手
   draw3DLineAndJoint(csvF1[i][1]-csvF1[i][x+y],csvF1[i][2]-csvF1[i][x+y+1],csvF1[i][3]-csvF1[i][x+y+2],csvF1[i][4]-csvF1[i][x+y],
   csvF1[i][5]-csvF1[i][x+y+1],csvF1[i][6]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);//親 3 →親 2
     weight =1.5;
   draw3DLineAndJoint(csvF1[i][1]-csvF1[i][x+y],csvF1[i][2]-csvF1[i][x+y+1],csvF1[i][3]-csvF1[i][x+y+2],csvF1[i][13]-csvF1[i][x+y],
   csvF1[i][14]-csvF1[i][x+y+1],csvF1[i][15]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);//親 3 →人 4
     weight =1.0;
   draw3DLineAndJoint(csvF1[i][1]-csvF1[i][x+y],csvF1[i][2]-csvF1[i][x+y+1],csvF1[i][3]-csvF1[i][x+y+2],csvF1[i][16]-csvF1[i][x+y],
   csvF1[i][17]-csvF1[i][x+y+1],csvF1[i][18]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);//親 3 →人 3
     weight =0.5;
   draw3DLineAndJoint(csvF1[i][4]-csvF1[i][x+y],csvF1[i][5]-csvF1[i][x+y+1],csvF1[i][6]-csvF1[i][x+y+2],csvF1[i][7]-csvF1[i][x+y],
   csvF1[i][8]-csvF1[i][x+y+1],csvF1[i][9]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);//親 2 →親 1
     weight =1.5;
   draw3DLineAndJoint(csvF1[i][4]-csvF1[i][x+y],csvF1[i][5]-csvF1[i][x+y+1],csvF1[i][6]-csvF1[i][x+y+2],csvF1[i][16]-csvF1[i][x+y],
   csvF1[i][17]-csvF1[i][x+y+1],csvF1[i][18]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);//親 2 →人 3
     weight =1.5;
   draw3DLineAndJoint(csvF1[i][7]-csvF1[i][x+y],csvF1[i][8]-csvF1[i][x+y+1],csvF1[i][9]-csvF1[i][x+y+2],csvF1[i][10]-csvF1[i][x+y],
   csvF1[i][11]-csvF1[i][x+y+1],csvF1[i][12]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);//親 1 →親先端
     weight =1.0;
   draw3DLineAndJoint(csvF1[i][13]-csvF1[i][x+y],csvF1[i][14]-csvF1[i][x+y+1],csvF1[i][15]-csvF1[i][x+y+2],csvF1[i][28]-csvF1[i][x+y],
   csvF1[i][29]-csvF1[i][x+y+1],csvF1[i][30]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);//人 4 →中 4
     weight =0.5;
   draw3DLineAndJoint(csvF1[i][16]-csvF1[i][x+y],csvF1[i][17]-csvF1[i][x+y+1],csvF1[i][18]-csvF1[i][x+y+2],csvF1[i][19]-csvF1[i][x+y],
   csvF1[i][20]-csvF1[i][x+y+1],csvF1[i][21]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);   //人 3 →人 2
     weight =1.5;
   draw3DLineAndJoint(csvF1[i][16]-csvF1[i][x+y],csvF1[i][17]-csvF1[i][x+y+1],csvF1[i][18]-csvF1[i][x+y+2],csvF1[i][31]-csvF1[i][x+y],
   csvF1[i][32]-csvF1[i][x+y+1],csvF1[i][33]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);//人 3 →中 3
     weight =1.0;
   draw3DLineAndJoint(csvF1[i][19]-csvF1[i][x+y],csvF1[i][20]-csvF1[i][x+y+1],csvF1[i][21]-csvF1[i][x+y+2],csvF1[i][22]-csvF1[i][x+y],
```

```
   csvF1[i][23]-csvF1[i][x+y+1],csvF1[i][24]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);   //人 2 →人 1
     weight =0.5;
   draw3DLineAndJoint(csvF1[i][22]-csvF1[i][x+y],csvF1[i][23]-csvF1[i][x+y+1],csvF1[i][24]-csvF1[i][x+y+2],csvF1[i][25]-csvF1[i][x+y],
   csvF1[i][26]-csvF1[i][x+y+1],csvF1[i][27]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);    //人 1 →人先端
     weight =1.5;
   draw3DLineAndJoint(csvF1[i][28]-csvF1[i][x+y],csvF1[i][29]-csvF1[i][x+y+1],csvF1[i][30]-csvF1[i][x+y+2],csvF1[i][43]-csvF1[i][x+y],
   csvF1[i][44]-csvF1[i][x+y+1],csvF1[i][45]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);//中 4 →薬 4
     weight =1.0;
   draw3DLineAndJoint(csvF1[i][31]-csvF1[i][x+y],csvF1[i][32]-csvF1[i][x+y+1],csvF1[i][33]-csvF1[i][x+y+2],csvF1[i][34]-csvF1[i][x+y],
   csvF1[i][35]-csvF1[i][x+y+1],csvF1[i][36]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);   //中 3 →中 2
     weight =0.5;
   draw3DLineAndJoint(csvF1[i][31]-csvF1[i][x+y],csvF1[i][32]-csvF1[i][x+y+1],csvF1[i][33]-csvF1[i][x+y+2],csvF1[i][46]-csvF1[i][x+y],
   csvF1[i][47]-csvF1[i][x+y+1],csvF1[i][48]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);//中 3 →薬 3
     weight =1.5;
   draw3DLineAndJoint(csvF1[i][34]-csvF1[i][x+y],csvF1[i][35]-csvF1[i][x+y+1],csvF1[i][36]-csvF1[i][x+y+2],csvF1[i][37]-csvF1[i][x+y],
   csvF1[i][38]-csvF1[i][x+y+1],csvF1[i][39]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);   //中 2 →中 1
     weight =1.0;
   draw3DLineAndJoint(csvF1[i][37]-csvF1[i][x+y],csvF1[i][38]-csvF1[i][x+y+1],csvF1[i][39]-csvF1[i][x+y+2],csvF1[i][40]-csvF1[i][x+y],
   csvF1[i][41]-csvF1[i][x+y+1],csvF1[i][42]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);    //中 1 →中先端
     weight =0.5;
   draw3DLineAndJoint(csvF1[i][43]-csvF1[i][x+y],csvF1[i][44]-csvF1[i][x+y+1],csvF1[i][45]-csvF1[i][x+y+2],csvF1[i][58]-csvF1[i][x+y],
   csvF1[i][59]-csvF1[i][x+y+1],csvF1[i][60]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);//薬 4 →小 4
     weight =1.5;
   draw3DLineAndJoint(csvF1[i][46]-csvF1[i][x+y],csvF1[i][47]-csvF1[i][x+y+1],csvF1[i][48]-csvF1[i][x+y+2],csvF1[i][49]-csvF1[i][x+y],
   csvF1[i][50]-csvF1[i][x+y+1],csvF1[i][51]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);   //薬 3 →薬 2
     weight =1.0;
   draw3DLineAndJoint(csvF1[i][46]-csvF1[i][x+y],csvF1[i][47]-csvF1[i][x+y+1],csvF1[i][48]-csvF1[i][x+y+2],csvF1[i][61]-csvF1[i][x+y],
   csvF1[i][62]-csvF1[i][x+y+1],csvF1[i][63]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);//薬 3 →小 3
     weight =0.5;
   draw3DLineAndJoint(csvF1[i][49]-csvF1[i][x+y],csvF1[i][50]-csvF1[i][x+y+1],csvF1[i][51]-csvF1[i][x+y+2],csvF1[i][52]-csvF1[i][x+y],
   csvF1[i][53]-csvF1[i][x+y+1],csvF1[i][54]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);   //薬 2 →薬 1
     weight =1.5;
   draw3DLineAndJoint(csvF1[i][52]-csvF1[i][x+y],csvF1[i][53]-csvF1[i][x+y+1],csvF1[i][54]-csvF1[i][x+y+2],csvF1[i][55]-csvF1[i][x+y],
   csvF1[i][56]-csvF1[i][x+y+1],csvF1[i][57]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);    //薬 1 →薬先端
     weight =1.0;
   draw3DLineAndJoint(csvF1[i][58]-csvF1[i][x+y],csvF1[i][59]-csvF1[i][x+y+1],csvF1[i][60]-csvF1[i][x+y+2],csvF1[i][61]-csvF1[i][x+y],
   csvF1[i][62]-csvF1[i][x+y+1],csvF1[i][63]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);    //小 4 → 3
     weight =0.5;
   draw3DLineAndJoint(csvF1[i][61]-csvF1[i][x+y],csvF1[i][62]-csvF1[i][x+y+1],csvF1[i][63]-csvF1[i][x+y+2],csvF1[i][64]-csvF1[i][x+y],
   csvF1[i][65]-csvF1[i][x+y+1],csvF1[i][66]-csvF1[i][x+y+2],xRotate,yRotate,zRotate); //小 3 →小 2
     weight =1.5;
   draw3DLineAndJoint(csvF1[i][64]-csvF1[i][x+y],csvF1[i][65]-csvF1[i][x+y+1],csvF1[i][66]-csvF1[i][x+y+2],csvF1[i][67]-csvF1[i][x+y],
   csvF1[i][68]-csvF1[i][x+y+1],csvF1[i][69]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);   //小 2 →小 1
     weight =1.0;
   draw3DLineAndJoint(csvF1[i][67]-csvF1[i][x+y],csvF1[i][68]-csvF1[i][x+y+1],csvF1[i][69]-csvF1[i][x+y+2],csvF1[i][70]-csvF1[i][x+y],
   csvF1[i][71]-csvF1[i][x+y+1],csvF1[i][72]-csvF1[i][x+y+2],xRotate,yRotate,zRotate);   //小 1 →小先端
     weight =0.5;
   r1r=0;
  }
  else {
   r1r=1;
   //右手
   draw3DLineAndJoint(csvF1[i][1],csvF1[i][2],csvF1[i][3],csvF1[i][4],csvF1[i][5],csvF1[i][6],xRotate,yRotate,zRotate);//親 3 →親 2
     weight =1.5;
   draw3DLineAndJoint(csvF1[i][1],csvF1[i][2],csvF1[i][3],csvF1[i][13],csvF1[i][14],csvF1[i][15],xRotate,yRotate,zRotate);//親 3 →人 4
     weight =1.0;
   draw3DLineAndJoint(csvF1[i][1],csvF1[i][2],csvF1[i][3],csvF1[i][16],csvF1[i][17],csvF1[i][18],xRotate,yRotate,zRotate);//親 3 →人 3
     weight =0.5;
   draw3DLineAndJoint(csvF1[i][4],csvF1[i][5],csvF1[i][6],csvF1[i][7],csvF1[i][8],csvF1[i][9],xRotate,yRotate,zRotate);//親 2 →親 1
     weight =1.5;
   draw3DLineAndJoint(csvF1[i][4],csvF1[i][5],csvF1[i][6],csvF1[i][16],csvF1[i][17],csvF1[i][18],xRotate,yRotate,zRotate);//親 2 →人 3
     weight =1.5;
   draw3DLineAndJoint(csvF1[i][7],csvF1[i][8],csvF1[i][9],csvF1[i][10],csvF1[i][11],csvF1[i][12],xRotate,yRotate,zRotate);//親 1 →親先端
     weight =1.0;
   draw3DLineAndJoint(csvF1[i][13],csvF1[i][14],csvF1[i][15],csvF1[i][28],csvF1[i][29],csvF1[i][30],xRotate,yRotate,zRotate);//人 4 →中 4
     weight =0.5;
   draw3DLineAndJoint(csvF1[i][16],csvF1[i][17],csvF1[i][18],csvF1[i][19],csvF1[i][20],csvF1[i][21],xRotate,yRotate,zRotate);   //人 3 →人 2
     weight =1.5;
   draw3DLineAndJoint(csvF1[i][16],csvF1[i][17],csvF1[i][18],csvF1[i][31],csvF1[i][32],csvF1[i][33],xRotate,yRotate,zRotate);//人 3 →中 3
     weight =1.0;
   draw3DLineAndJoint(csvF1[i][19],csvF1[i][20],csvF1[i][21],csvF1[i][22],csvF1[i][23],csvF1[i][24],xRotate,yRotate,zRotate);   //人 2 →人 1
     weight =0.5;
   draw3DLineAndJoint(csvF1[i][22],csvF1[i][23],csvF1[i][24],csvF1[i][25],csvF1[i][26],csvF1[i][27],xRotate,yRotate,zRotate);    //人 1 →人
先端
     weight =1.5;
   draw3DLineAndJoint(csvF1[i][28],csvF1[i][29],csvF1[i][30],csvF1[i][43],csvF1[i][44],csvF1[i][45],xRotate,yRotate,zRotate);//中 4 →薬 4
     weight =1.0;
   draw3DLineAndJoint(csvF1[i][31],csvF1[i][32],csvF1[i][33],csvF1[i][34],csvF1[i][35],csvF1[i][36],xRotate,yRotate,zRotate);   //中 3 →中 2
     weight =0.5;
   draw3DLineAndJoint(csvF1[i][31],csvF1[i][32],csvF1[i][33],csvF1[i][46],csvF1[i][47],csvF1[i][48],xRotate,yRotate,zRotate);//中 3 →薬 3
     weight =1.5;
   draw3DLineAndJoint(csvF1[i][34],csvF1[i][35],csvF1[i][36],csvF1[i][37],csvF1[i][38],csvF1[i][39],xRotate,yRotate,zRotate);   //中 2 →中 1
     weight =1.0;
   draw3DLineAndJoint(csvF1[i][37],csvF1[i][38],csvF1[i][39],csvF1[i][40],csvF1[i][41],csvF1[i][42],xRotate,yRotate,zRotate);    //中 1 →中
先端
     weight =0.5;
   draw3DLineAndJoint(csvF1[i][43],csvF1[i][44],csvF1[i][45],csvF1[i][58],csvF1[i][59],csvF1[i][60],xRotate,yRotate,zRotate);//薬 4 →小 4
     weight =1.5;
   draw3DLineAndJoint(csvF1[i][46],csvF1[i][47],csvF1[i][48],csvF1[i][49],csvF1[i][50],csvF1[i][51],xRotate,yRotate,zRotate);   //薬 3 →薬 2
     weight =1.0;
   draw3DLineAndJoint(csvF1[i][46],csvF1[i][47],csvF1[i][48],csvF1[i][61],csvF1[i][62],csvF1[i][63],xRotate,yRotate,zRotate);//薬 3 →小 3
```

```
      weight =0.5;
    draw3DLineAndJoint(csvF1[i][49],csvF1[i][50],csvF1[i][51],csvF1[i][52],csvF1[i][53],csvF1[i][54],xRotate,yRotate,zRotate);   //薬 2 →薬 1
      weight =1.5;
    draw3DLineAndJoint(csvF1[i][52],csvF1[i][53],csvF1[i][54],csvF1[i][55],csvF1[i][56],csvF1[i][57],xRotate,yRotate,zRotate);    //薬 1 →薬
先端
      weight =1.0;
    draw3DLineAndJoint(csvF1[i][58],csvF1[i][59],csvF1[i][60],csvF1[i][61],csvF1[i][62],csvF1[i][63],xRotate,yRotate,zRotate);    //小 4 → 3
      weight =0.5;
    draw3DLineAndJoint(csvF1[i][61],csvF1[i][62],csvF1[i][63],csvF1[i][64],csvF1[i][65],csvF1[i][66],xRotate,yRotate,zRotate); //小 3 →小 2
      weight =1.5;
    draw3DLineAndJoint(csvF1[i][64],csvF1[i][65],csvF1[i][66],csvF1[i][67],csvF1[i][68],csvF1[i][69],xRotate,yRotate,zRotate);   //小 2 →小 1
      weight =1.0;
    draw3DLineAndJoint(csvF1[i][67],csvF1[i][68],csvF1[i][69],csvF1[i][70],csvF1[i][71],csvF1[i][72],xRotate,yRotate,zRotate);   //小 1 →小先
端
      weight =0.5;
    r1r=0;
   }
}
////////////////////////////////////////////////
//関数名：AnotherRightAni
//概要：保存した右手の座標からアニメーションを動
//       かす。
//関連する関数
//   ・draw3DLineAndJoint
//   ・JudgeColorRight<<右手の配色>>
//本当は DrawHands の中で行いたい
////////////////////////////////////////////////////
void AnotherRightAni()
{
    AnotherJudgeColor_Right(); //右手の配色
   if(selectButton == 1){
    r2r=1;
    //右手
    draw3DLineAndJoint(csvT1[i2][1]-csvT1[i2][x+y],csvT1[i2][2]-csvT1[i2][x+y+1],csvT1[i2][3]-csvT1[i2][x+y+2],csvT1[i2][4]-csvT1[i2][x+y],
    csvT1[i2][5]-csvT1[i2][x+y+1],csvT1[i2][6]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);//親 3 →親 2
      weight =1.0;
    draw3DLineAndJoint(csvT1[i2][1]-csvT1[i2][x+y],csvT1[i2][2]-csvT1[i2][x+y+1],csvT1[i2][3]-csvT1[i2][x+y+2],csvT1[i2][13]-csvT1[i2][x+y],
    csvT1[i2][14]-csvT1[i2][x+y+1],csvT1[i2][15]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);//親 3 →人 4
      weight =1.5;
    draw3DLineAndJoint(csvT1[i2][1]-csvT1[i2][x+y],csvT1[i2][2]-csvT1[i2][x+y+1],csvT1[i2][3]-csvT1[i2][x+y+2],csvT1[i2][16]-csvT1[i2][x+y],
    csvT1[i2][17]-csvT1[i2][x+y+1],csvT1[i2][18]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);//親 3 →人 3
      weight =0.5;
    draw3DLineAndJoint(csvT1[i2][4]-csvT1[i2][x+y],csvT1[i2][5]-csvT1[i2][x+y+1],csvT1[i2][6]-csvT1[i2][x+y+2],csvT1[i2][7]-csvT1[i2][x+y],
    csvT1[i2][8]-csvT1[i2][x+y+1],csvT1[i2][9]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);//親 2 →親 1
      weight =1.0;
    draw3DLineAndJoint(csvT1[i2][4]-csvT1[i2][x+y],csvT1[i2][5]-csvT1[i2][x+y+1],csvT1[i2][6]-csvT1[i2][x+y+2],csvT1[i2][16]-csvT1[i2][x+y],
    csvT1[i2][17]-csvT1[i2][x+y+1],csvT1[i2][18]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);//親 2 →人 3
      weight =1.5;
    draw3DLineAndJoint(csvT1[i2][7]-csvT1[i2][x+y],csvT1[i2][8]-csvT1[i2][x+y+1],csvT1[i2][9]-csvT1[i2][x+y+2],csvT1[i2][10]-csvT1[i2][x+y],
    csvT1[i2][11]-csvT1[i2][x+y+1],csvT1[i2][12]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);//親 1 →親先端
      weight =0.5;
    draw3DLineAndJoint(csvT1[i2][13]-csvT1[i2][x+y],csvT1[i2][14]-csvT1[i2][x+y+1],csvT1[i2][15]-csvT1[i2][x+y+2],csvT1[i2][28]-csvT1[i2][x+y],
    csvT1[i2][29]-csvT1[i2][x+y+1],csvT1[i2][30]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);//人 4 →中 4
      weight =1.0;
    draw3DLineAndJoint(csvT1[i2][16]-csvT1[i2][x+y],csvT1[i2][17]-csvT1[i2][x+y+1],csvT1[i2][18]-csvT1[i2][x+y+2],csvT1[i2][19]-csvT1[i2][x+y],
    csvT1[i2][20]-csvT1[i2][x+y+1],csvT1[i2][21]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);   //人 3 →人 2
      weight =1.5;
    draw3DLineAndJoint(csvT1[i2][16]-csvT1[i2][x+y],csvT1[i2][17]-csvT1[i2][x+y+1],csvT1[i2][18]-csvT1[i2][x+y+2],csvT1[i2][31]-csvT1[i2][x+y],
    csvT1[i2][32]-csvT1[i2][x+y+1],csvT1[i2][33]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);//人 3 →中 3
      weight =0.5;
    draw3DLineAndJoint(csvT1[i2][19]-csvT1[i2][x+y],csvT1[i2][20]-csvT1[i2][x+y+1],csvT1[i2][21]-csvT1[i2][x+y+2],csvT1[i2][22]-csvT1[i2][x+y],
    csvT1[i2][23]-csvT1[i2][x+y+1],csvT1[i2][24]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);   //人 2 →人 1
      weight =1.0;
    draw3DLineAndJoint(csvT1[i2][22]-csvT1[i2][x+y],csvT1[i2][23]-csvT1[i2][x+y+1],csvT1[i2][24]-csvT1[i2][x+y+2],csvT1[i2][25]-csvT1[i2][x+y],
    csvT1[i2][26]-csvT1[i2][x+y+1],csvT1[i2][27]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);    //人 1 →人先端
      weight =1.5;
    draw3DLineAndJoint(csvT1[i2][28]-csvT1[i2][x+y],csvT1[i2][29]-csvT1[i2][x+y+1],csvT1[i2][30]-csvT1[i2][x+y+2],csvT1[i2][43]-csvT1[i2][x+y],
    csvT1[i2][44]-csvT1[i2][x+y+1],csvT1[i2][45]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);//中 4 →薬 4
      weight =0.5;
    draw3DLineAndJoint(csvT1[i2][31]-csvT1[i2][x+y],csvT1[i2][32]-csvT1[i2][x+y+1],csvT1[i2][33]-csvT1[i2][x+y+2],csvT1[i2][34]-csvT1[i2][x+y],
    csvT1[i2][35]-csvT1[i2][x+y+1],csvT1[i2][36]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);   //中 3 →中 2
      weight =1.0;
    draw3DLineAndJoint(csvT1[i2][31]-csvT1[i2][x+y],csvT1[i2][32]-csvT1[i2][x+y+1],csvT1[i2][33]-csvT1[i2][x+y+2],csvT1[i2][46]-csvT1[i2][x+y],
    csvT1[i2][47]-csvT1[i2][x+y+1],csvT1[i2][48]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);//中 3 →薬 3
      weight =1.5;
    draw3DLineAndJoint(csvT1[i2][34]-csvT1[i2][x+y],csvT1[i2][35]-csvT1[i2][x+y+1],csvT1[i2][36]-csvT1[i2][x+y+2],csvT1[i2][37]-csvT1[i2][x+y],
    csvT1[i2][38]-csvT1[i2][x+y+1],csvT1[i2][39]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);   //中 2 →中 1
      weight =0.5;
    draw3DLineAndJoint(csvT1[i2][37]-csvT1[i2][x+y],csvT1[i2][38]-csvT1[i2][x+y+1],csvT1[i2][39]-csvT1[i2][x+y+2],csvT1[i2][40]-csvT1[i2][x+y],
    csvT1[i2][41]-csvT1[i2][x+y+1],csvT1[i2][42]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);    //中 1 →中先端
      weight =1.0;
    draw3DLineAndJoint(csvT1[i2][43]-csvT1[i2][x+y],csvT1[i2][44]-csvT1[i2][x+y+1],csvT1[i2][45]-csvT1[i2][x+y+2],csvT1[i2][58]-csvT1[i2][x+y],
    csvT1[i2][59]-csvT1[i2][x+y+1],csvT1[i2][60]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);//薬 4 →小 4
      weight =1.5;
    draw3DLineAndJoint(csvT1[i2][46]-csvT1[i2][x+y],csvT1[i2][47]-csvT1[i2][x+y+1],csvT1[i2][48]-csvT1[i2][x+y+2],csvT1[i2][49]-csvT1[i2][x+y],
    csvT1[i2][50]-csvT1[i2][x+y+1],csvT1[i2][51]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);   //薬 3 →薬 2
      weight =0.5;
    draw3DLineAndJoint(csvT1[i2][46]-csvT1[i2][x+y],csvT1[i2][47]-csvT1[i2][x+y+1],csvT1[i2][48]-csvT1[i2][x+y+2],csvT1[i2][61]-csvT1[i2][x+y],
    csvT1[i2][62]-csvT1[i2][x+y+1],csvT1[i2][63]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);//薬 3 →小 3
      weight =1.0;
    draw3DLineAndJoint(csvT1[i2][49]-csvT1[i2][x+y],csvT1[i2][50]-csvT1[i2][x+y+1],csvT1[i2][51]-csvT1[i2][x+y+2],csvT1[i2][52]-csvT1[i2][x+y],
```

```
   csvT1[i2][53]-csvT1[i2][x+y+1],csvT1[i2][54]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);   //薬 2 →薬 1
     weight =1.5;
   draw3DLineAndJoint(csvT1[i2][52]-csvT1[i2][x+y],csvT1[i2][53]-csvT1[i2][x+y+1],csvT1[i2][54]-csvT1[i2][x+y+2],csvT1[i2][55]-csvT1[i2][x+y],
   csvT1[i2][56]-csvT1[i2][x+y+1],csvT1[i2][57]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);    //薬 1 →薬先端
     weight =0.5;
   draw3DLineAndJoint(csvT1[i2][58]-csvT1[i2][x+y],csvT1[i2][59]-csvT1[i2][x+y+1],csvT1[i2][60]-csvT1[i2][x+y+2],csvT1[i2][61]-csvT1[i2][x+y],
   csvT1[i2][62]-csvT1[i2][x+y+1],csvT1[i2][63]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);    //小 4 → 3
     weight =1.0;
   draw3DLineAndJoint(csvT1[i2][61]-csvT1[i2][x+y],csvT1[i2][62]-csvT1[i2][x+y+1],csvT1[i2][63]-csvT1[i2][x+y+2],csvT1[i2][64]-csvT1[i2][x+y],
   csvT1[i2][65]-csvT1[i2][x+y+1],csvT1[i2][66]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate); //小 3 →小 2
     weight =1.5;
   draw3DLineAndJoint(csvT1[i2][64]-csvT1[i2][x+y],csvT1[i2][65]-csvT1[i2][x+y+1],csvT1[i2][66]-csvT1[i2][x+y+2],csvT1[i2][67]-csvT1[i2][x+y],
   csvT1[i2][68]-csvT1[i2][x+y+1],csvT1[i2][69]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);   //小 2 →小 1
     weight =0.5;
   draw3DLineAndJoint(csvT1[i2][67]-csvT1[i2][x+y],csvT1[i2][68]-csvT1[i2][x+y+1],csvT1[i2][69]-csvT1[i2][x+y+2],csvT1[i2][70]-csvT1[i2][x+y],
   csvT1[i2][71]-csvT1[i2][x+y+1],csvT1[i2][72]-csvT1[i2][x+y+2],xRotate,yRotate,zRotate);   //小 1 →小先端
     weight =1.0;
   r2r=0;
  }
  else{
   r2r=1;
   //右手
   draw3DLineAndJoint(csvT1[i2][1],csvT1[i2][2],csvT1[i2][3],csvT1[i2][4],csvT1[i2][5],csvT1[i2][6],xRotate,yRotate,zRotate);//親 3 →親 2
     weight =1.0;
   draw3DLineAndJoint(csvT1[i2][1],csvT1[i2][2],csvT1[i2][3],csvT1[i2][13],csvT1[i2][14],csvT1[i2][15],xRotate,yRotate,zRotate);//親 3 →人 4
     weight =1.5;
   draw3DLineAndJoint(csvT1[i2][1],csvT1[i2][2],csvT1[i2][3],csvT1[i2][16],csvT1[i2][17],csvT1[i2][18],xRotate,yRotate,zRotate);//親 3 →人 3
     weight =0.5;
   draw3DLineAndJoint(csvT1[i2][4],csvT1[i2][5],csvT1[i2][6],csvT1[i2][7],csvT1[i2][8],csvT1[i2][9],xRotate,yRotate,zRotate);//親 2 →親 1
     weight =1.0;
   draw3DLineAndJoint(csvT1[i2][4],csvT1[i2][5],csvT1[i2][6],csvT1[i2][16],csvT1[i2][17],csvT1[i2][18],xRotate,yRotate,zRotate);//親 2 →人 3
     weight =1.5;
   draw3DLineAndJoint(csvT1[i2][7],csvT1[i2][8],csvT1[i2][9],csvT1[i2][10],csvT1[i2][11],csvT1[i2][12],xRotate,yRotate,zRotate);//親 1 →親先
端
     weight =0.5;
   draw3DLineAndJoint(csvT1[i2][13],csvT1[i2][14],csvT1[i2][15],csvT1[i2][28],csvT1[i2][29],csvT1[i2][30],xRotate,yRotate,zRotate);//人 4 →
中 4
     weight =1.0;
   draw3DLineAndJoint(csvT1[i2][16],csvT1[i2][17],csvT1[i2][18],csvT1[i2][19],csvT1[i2][20],csvT1[i2][21],xRotate,yRotate,zRotate);   //人 3
→人 2
     weight =1.5;
   draw3DLineAndJoint(csvT1[i2][16],csvT1[i2][17],csvT1[i2][18],csvT1[i2][31],csvT1[i2][32],csvT1[i2][33],xRotate,yRotate,zRotate);//人 3 →
中 3
     weight =0.5;
   draw3DLineAndJoint(csvT1[i2][19],csvT1[i2][20],csvT1[i2][21],csvT1[i2][22],csvT1[i2][23],csvT1[i2][24],xRotate,yRotate,zRotate);   //人 2
→人 1
     weight =1.0;
   draw3DLineAndJoint(csvT1[i2][22],csvT1[i2][23],csvT1[i2][24],csvT1[i2][25],csvT1[i2][26],csvT1[i2][27],xRotate,yRotate,zRotate);    //人
1 →人先端
     weight =1.5;
   draw3DLineAndJoint(csvT1[i2][28],csvT1[i2][29],csvT1[i2][30],csvT1[i2][43],csvT1[i2][44],csvT1[i2][45],xRotate,yRotate,zRotate);//中 4 →
薬 4
     weight =0.5;
   draw3DLineAndJoint(csvT1[i2][31],csvT1[i2][32],csvT1[i2][33],csvT1[i2][34],csvT1[i2][35],csvT1[i2][36],xRotate,yRotate,zRotate);   //中 3
→中 2
     weight =1.0;
   draw3DLineAndJoint(csvT1[i2][31],csvT1[i2][32],csvT1[i2][33],csvT1[i2][46],csvT1[i2][47],csvT1[i2][48],xRotate,yRotate,zRotate);//中 3 →
薬 3
     weight =1.5;
   draw3DLineAndJoint(csvT1[i2][34],csvT1[i2][35],csvT1[i2][36],csvT1[i2][37],csvT1[i2][38],csvT1[i2][39],xRotate,yRotate,zRotate);   //中 2
→中 1
     weight =0.5;
   draw3DLineAndJoint(csvT1[i2][37],csvT1[i2][38],csvT1[i2][39],csvT1[i2][40],csvT1[i2][41],csvT1[i2][42],xRotate,yRotate,zRotate);    //中
1 →中先端
     weight =1.0;
   draw3DLineAndJoint(csvT1[i2][43],csvT1[i2][44],csvT1[i2][45],csvT1[i2][58],csvT1[i2][59],csvT1[i2][60],xRotate,yRotate,zRotate);//薬 4 →
小 4
     weight =1.5;
   draw3DLineAndJoint(csvT1[i2][46],csvT1[i2][47],csvT1[i2][48],csvT1[i2][49],csvT1[i2][50],csvT1[i2][51],xRotate,yRotate,zRotate);   //薬 3
→薬 2
     weight =0.5;
   draw3DLineAndJoint(csvT1[i2][46],csvT1[i2][47],csvT1[i2][48],csvT1[i2][61],csvT1[i2][62],csvT1[i2][63],xRotate,yRotate,zRotate);//薬 3 →
小 3
     weight =1.0;
   draw3DLineAndJoint(csvT1[i2][49],csvT1[i2][50],csvT1[i2][51],csvT1[i2][52],csvT1[i2][53],csvT1[i2][54],xRotate,yRotate,zRotate);   //薬 2
→薬 1
     weight =1.5;
   draw3DLineAndJoint(csvT1[i2][52],csvT1[i2][53],csvT1[i2][54],csvT1[i2][55],csvT1[i2][56],csvT1[i2][57],xRotate,yRotate,zRotate);    //薬
1 →薬先端
     weight =0.5;
   draw3DLineAndJoint(csvT1[i2][58],csvT1[i2][59],csvT1[i2][60],csvT1[i2][61],csvT1[i2][62],csvT1[i2][63],xRotate,yRotate,zRotate);    //小
4 → 3
     weight =1.0;
   draw3DLineAndJoint(csvT1[i2][61],csvT1[i2][62],csvT1[i2][63],csvT1[i2][64],csvT1[i2][65],csvT1[i2][66],xRotate,yRotate,zRotate); //小  3
→小 2
     weight =1.5;
   draw3DLineAndJoint(csvT1[i2][64],csvT1[i2][65],csvT1[i2][66],csvT1[i2][67],csvT1[i2][68],csvT1[i2][69],xRotate,yRotate,zRotate);   //小 2
→小 1
     weight =0.5;
   draw3DLineAndJoint(csvT1[i2][67],csvT1[i2][68],csvT1[i2][69],csvT1[i2][70],csvT1[i2][71],csvT1[i2][72],xRotate,yRotate,zRotate);   //小 1
→小先端
```

```
    weight =1.0;
   r2r=0;
  }
}
////////////////////////////////////////////////
//関数名：LeftAni
//概要：保存した右手の座標からアニメーションを動
//      かす。
//関連する関数
//  ・draw3DLineAndJoint
//  ・JudgeColorLeft<<左手の配色>>
////////////////////////////////////////////////
void LeftAni(){
   JudgeColor_Left(); //左手の配色
  if(selectButton == 1){
   l1l=1;
   //左手
   draw3DLineAndJoint(csvS1[j][1]-csvS1[j][x+y],csvS1[j][2]-csvS1[j][x+y+1],csvS1[j][3]-csvS1[j][x+y+2],csvS1[j][4]-csvS1[j][x+y],
   csvS1[j][5]-csvS1[j][x+y+1],csvS1[j][6]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);//親 3 →親 2
     weight =1.5;
   draw3DLineAndJoint(csvS1[j][1]-csvS1[j][x+y],csvS1[j][2]-csvS1[j][x+y+1],csvS1[j][3]-csvS1[j][x+y+2],csvS1[j][13]-csvS1[j][x+y],
   csvS1[j][14]-csvS1[j][x+y+1],csvS1[j][15]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);//親 3 →人 4
     weight =0.5;
   draw3DLineAndJoint(csvS1[j][1]-csvS1[j][x+y],csvS1[j][2]-csvS1[j][x+y+1],csvS1[j][3]-csvS1[j][x+y+2],csvS1[j][16]-csvS1[j][x+y],
   csvS1[j][17]-csvS1[j][x+y+1],csvS1[j][18]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);//親 3 →人 3
     weight =1.0;
   draw3DLineAndJoint(csvS1[j][4]-csvS1[j][x+y],csvS1[j][5]-csvS1[j][x+y+1],csvS1[j][6]-csvS1[j][x+y+2],csvS1[j][7]-csvS1[j][x+y],
   csvS1[j][8]-csvS1[j][x+y+1],csvS1[j][9]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);//親 2 →親 1
     weight =1.5;
   draw3DLineAndJoint(csvS1[j][4]-csvS1[j][x+y],csvS1[j][5]-csvS1[j][x+y+1],csvS1[j][6]-csvS1[j][x+y+2],csvS1[j][16]-csvS1[j][x+y],
   csvS1[j][17]-csvS1[j][x+y+1],csvS1[j][18]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);//親 2 →人 3
     weight =0.5;
   draw3DLineAndJoint(csvS1[j][7]-csvS1[j][x+y],csvS1[j][8]-csvS1[j][x+y+1],csvS1[j][9]-csvS1[j][x+y+2],csvS1[j][10]-csvS1[j][x+y],
   csvS1[j][11]-csvS1[j][x+y+1],csvS1[j][12]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);//親 1 →親先端
     weight =1.0;
   draw3DLineAndJoint(csvS1[j][13]-csvS1[j][x+y],csvS1[j][14]-csvS1[j][x+y+1],csvS1[j][15]-csvS1[j][x+y+2],csvS1[j][28]-csvS1[j][x+y],
   csvS1[j][29]-csvS1[j][x+y+1],csvS1[j][30]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);//人 4 →中 4
     weight =1.5;
   draw3DLineAndJoint(csvS1[j][16]-csvS1[j][x+y],csvS1[j][17]-csvS1[j][x+y+1],csvS1[j][18]-csvS1[j][x+y+2],csvS1[j][19]-csvS1[j][x+y],
   csvS1[j][20]-csvS1[j][x+y+1],csvS1[j][21]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);   //人 3 →人 2
     weight =0.5;
   draw3DLineAndJoint(csvS1[j][16]-csvS1[j][x+y],csvS1[j][17]-csvS1[j][x+y+1],csvS1[j][18]-csvS1[j][x+y+2],csvS1[j][31]-csvS1[j][x+y],
   csvS1[j][32]-csvS1[j][x+y+1],csvS1[j][33]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);//人 3 →中 3
     weight =1.0;
   draw3DLineAndJoint(csvS1[j][19]-csvS1[j][x+y],csvS1[j][20]-csvS1[j][x+y+1],csvS1[j][21]-csvS1[j][x+y+2],csvS1[j][22]-csvS1[j][x+y],
   csvS1[j][23]-csvS1[j][x+y+1],csvS1[j][24]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);   //人 2 →人 1
     weight =1.5;
   draw3DLineAndJoint(csvS1[j][22]-csvS1[j][x+y],csvS1[j][23]-csvS1[j][x+y+1],csvS1[j][24]-csvS1[j][x+y+2],csvS1[j][25]-csvS1[j][x+y],
   csvS1[j][26]-csvS1[j][x+y+1],csvS1[j][27]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);    //人 1 →人先端
     weight =0.5;
   draw3DLineAndJoint(csvS1[j][28]-csvS1[j][x+y],csvS1[j][29]-csvS1[j][x+y+1],csvS1[j][30]-csvS1[j][x+y+2],csvS1[j][43]-csvS1[j][x+y],
   csvS1[j][44]-csvS1[j][x+y+1],csvS1[j][45]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);//中 4 →薬 4
     weight =1.0;
   draw3DLineAndJoint(csvS1[j][31]-csvS1[j][x+y],csvS1[j][32]-csvS1[j][x+y+1],csvS1[j][33]-csvS1[j][x+y+2],csvS1[j][34]-csvS1[j][x+y],
   csvS1[j][35]-csvS1[j][x+y+1],csvS1[j][36]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);   //中 3 →中 2
     weight =1.5;
   draw3DLineAndJoint(csvS1[j][31]-csvS1[j][x+y],csvS1[j][32]-csvS1[j][x+y+1],csvS1[j][33]-csvS1[j][x+y+2],csvS1[j][46]-csvS1[j][x+y],
   csvS1[j][47]-csvS1[j][x+y+1],csvS1[j][48]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);//中 3 →薬 3
     weight =0.5;
   draw3DLineAndJoint(csvS1[j][34]-csvS1[j][x+y],csvS1[j][35]-csvS1[j][x+y+1],csvS1[j][36]-csvS1[j][x+y+2],csvS1[j][37]-csvS1[j][x+y],
   csvS1[j][38]-csvS1[j][x+y+1],csvS1[j][39]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);   //中 2 →中 1
     weight =1.0;
   draw3DLineAndJoint(csvS1[j][37]-csvS1[j][x+y],csvS1[j][38]-csvS1[j][x+y+1],csvS1[j][39]-csvS1[j][x+y+2],csvS1[j][40]-csvS1[j][x+y],
   csvS1[j][41]-csvS1[j][x+y+1],csvS1[j][42]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);    //中 1 →中先端
     weight =1.5;
   draw3DLineAndJoint(csvS1[j][43]-csvS1[j][x+y],csvS1[j][44]-csvS1[j][x+y+1],csvS1[j][45]-csvS1[j][x+y+2],csvS1[j][58]-csvS1[j][x+y],
   csvS1[j][59]-csvS1[j][x+y+1],csvS1[j][60]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);//薬 4 →小 4
     weight =0.5;
   draw3DLineAndJoint(csvS1[j][46]-csvS1[j][x+y],csvS1[j][47]-csvS1[j][x+y+1],csvS1[j][48]-csvS1[j][x+y+2],csvS1[j][49]-csvS1[j][x+y],
   csvS1[j][50]-csvS1[j][x+y+1],csvS1[j][51]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);   //薬 3 →薬 2
     weight =1.0;
   draw3DLineAndJoint(csvS1[j][46]-csvS1[j][x+y],csvS1[j][47]-csvS1[j][x+y+1],csvS1[j][48]-csvS1[j][x+y+2],csvS1[j][61]-csvS1[j][x+y],
   csvS1[j][62]-csvS1[j][x+y+1],csvS1[j][63]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);//薬 3 →小 3
     weight =1.5;
   draw3DLineAndJoint(csvS1[j][49]-csvS1[j][x+y],csvS1[j][50]-csvS1[j][x+y+1],csvS1[j][51]-csvS1[j][x+y+2],csvS1[j][52]-csvS1[j][x+y],
   csvS1[j][53]-csvS1[j][x+y+1],csvS1[j][54]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);   //薬 2 →薬 1
     weight =0.5;
   draw3DLineAndJoint(csvS1[j][52]-csvS1[j][x+y],csvS1[j][53]-csvS1[j][x+y+1],csvS1[j][54]-csvS1[j][x+y+2],csvS1[j][55]-csvS1[j][x+y],
   csvS1[j][56]-csvS1[j][x+y+1],csvS1[j][57]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);    //薬 1 →薬先端
     weight =1.0;
   draw3DLineAndJoint(csvS1[j][58]-csvS1[j][x+y],csvS1[j][59]-csvS1[j][x+y+1],csvS1[j][60]-csvS1[j][x+y+2],csvS1[j][61]-csvS1[j][x+y],
   csvS1[j][62]-csvS1[j][x+y+1],csvS1[j][63]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);    //小 4 → 3
     weight =1.5;
   draw3DLineAndJoint(csvS1[j][61]-csvS1[j][x+y],csvS1[j][62]-csvS1[j][x+y+1],csvS1[j][63]-csvS1[j][x+y+2],csvS1[j][64]-csvS1[j][x+y],
   csvS1[j][65]-csvS1[j][x+y+1],csvS1[j][66]-csvS1[j][x+y+2],xRotate,yRotate,zRotate); //小 3 →小 2
     weight =0.5;
   draw3DLineAndJoint(csvS1[j][64]-csvS1[j][x+y],csvS1[j][65]-csvS1[j][x+y+1],csvS1[j][66]-csvS1[j][x+y+2],csvS1[j][67]-csvS1[j][x+y],
   csvS1[j][68]-csvS1[j][x+y+1],csvS1[j][69]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);   //小 2 →小 1
     weight =1.0;
   draw3DLineAndJoint(csvS1[j][67]-csvS1[j][x+y],csvS1[j][68]-csvS1[j][x+y+1],csvS1[j][69]-csvS1[j][x+y+2],csvS1[j][70]-csvS1[j][x+y],
   csvS1[j][71]-csvS1[j][x+y+1],csvS1[j][72]-csvS1[j][x+y+2],xRotate,yRotate,zRotate);   //小 1 →小先端
```

```
    weight =1.5;
  l11=0;
  }
  else{
   l11=1;
   //左手
   draw3DLineAndJoint(csvS1[j][1],csvS1[j][2],csvS1[j][3],csvS1[j][4],csvS1[j][5],csvS1[j][6],xRotate,yRotate,zRotate);//親 3 →親 2
     weight =1.5;
   draw3DLineAndJoint(csvS1[j][1],csvS1[j][2],csvS1[j][3],csvS1[j][13],csvS1[j][14],csvS1[j][15],xRotate,yRotate,zRotate);//親 3 →人 4
     weight =0.5;
   draw3DLineAndJoint(csvS1[j][1],csvS1[j][2],csvS1[j][3],csvS1[j][16],csvS1[j][17],csvS1[j][18],xRotate,yRotate,zRotate);//親 3 →人 3
     weight =1.0;
   draw3DLineAndJoint(csvS1[j][4],csvS1[j][5],csvS1[j][6],csvS1[j][7],csvS1[j][8],csvS1[j][9],xRotate,yRotate,zRotate);//親 2 →親 1
     weight =1.5;
   draw3DLineAndJoint(csvS1[j][4],csvS1[j][5],csvS1[j][6],csvS1[j][16],csvS1[j][17],csvS1[j][18],xRotate,yRotate,zRotate);//親 2 →人 3
     weight =0.5;
   draw3DLineAndJoint(csvS1[j][7],csvS1[j][8],csvS1[j][9],csvS1[j][10],csvS1[j][11],csvS1[j][12],xRotate,yRotate,zRotate);//親 1 →親先端
     weight =1.0;
   draw3DLineAndJoint(csvS1[j][13],csvS1[j][14],csvS1[j][15],csvS1[j][28],csvS1[j][29],csvS1[j][30],xRotate,yRotate,zRotate);//人 4 →中 4
     weight =1.5;
   draw3DLineAndJoint(csvS1[j][16],csvS1[j][17],csvS1[j][18],csvS1[j][19],csvS1[j][20],csvS1[j][21],xRotate,yRotate,zRotate);   //人 3 →人 2
     weight =0.5;
   draw3DLineAndJoint(csvS1[j][16],csvS1[j][17],csvS1[j][18],csvS1[j][31],csvS1[j][32],csvS1[j][33],xRotate,yRotate,zRotate);//人 3 →中 3
     weight =1.0;
   draw3DLineAndJoint(csvS1[j][19],csvS1[j][20],csvS1[j][21],csvS1[j][22],csvS1[j][23],csvS1[j][24],xRotate,yRotate,zRotate);   //人 2 →人 1
     weight =1.5;
   draw3DLineAndJoint(csvS1[j][22],csvS1[j][23],csvS1[j][24],csvS1[j][25],csvS1[j][26],csvS1[j][27],xRotate,yRotate,zRotate);    //人 1 →人
先端
     weight =0.5;
   draw3DLineAndJoint(csvS1[j][28],csvS1[j][29],csvS1[j][30],csvS1[j][43],csvS1[j][44],csvS1[j][45],xRotate,yRotate,zRotate);//中 4 →薬 4
     weight =1.0;
   draw3DLineAndJoint(csvS1[j][31],csvS1[j][32],csvS1[j][33],csvS1[j][34],csvS1[j][35],csvS1[j][36],xRotate,yRotate,zRotate);   //中 3 →中 2
     weight =1.5;
   draw3DLineAndJoint(csvS1[j][31],csvS1[j][32],csvS1[j][33],csvS1[j][46],csvS1[j][47],csvS1[j][48],xRotate,yRotate,zRotate);//中 3 →薬 3
     weight =0.5;
   draw3DLineAndJoint(csvS1[j][34],csvS1[j][35],csvS1[j][36],csvS1[j][37],csvS1[j][38],csvS1[j][39],xRotate,yRotate,zRotate);   //中 2 →中 1
     weight =1.0;
   draw3DLineAndJoint(csvS1[j][37],csvS1[j][38],csvS1[j][39],csvS1[j][40],csvS1[j][41],csvS1[j][42],xRotate,yRotate,zRotate);    //中 1 →中
先端
     weight =1.5;
   draw3DLineAndJoint(csvS1[j][43],csvS1[j][44],csvS1[j][45],csvS1[j][58],csvS1[j][59],csvS1[j][60],xRotate,yRotate,zRotate);//薬 4 →小 4
     weight =0.5;
   draw3DLineAndJoint(csvS1[j][46],csvS1[j][47],csvS1[j][48],csvS1[j][49],csvS1[j][50],csvS1[j][51],xRotate,yRotate,zRotate);   //薬 3 →薬 2
     weight =1.0;
   draw3DLineAndJoint(csvS1[j][46],csvS1[j][47],csvS1[j][48],csvS1[j][61],csvS1[j][62],csvS1[j][63],xRotate,yRotate,zRotate);//薬 3 →小 3
     weight =1.5;
   draw3DLineAndJoint(csvS1[j][49],csvS1[j][50],csvS1[j][51],csvS1[j][52],csvS1[j][53],csvS1[j][54],xRotate,yRotate,zRotate);   //薬 2 →薬 1
     weight =0.5;
   draw3DLineAndJoint(csvS1[j][52],csvS1[j][53],csvS1[j][54],csvS1[j][55],csvS1[j][56],csvS1[j][57],xRotate,yRotate,zRotate);    //薬 1 →薬
先端
     weight =1.0;
   draw3DLineAndJoint(csvS1[j][58],csvS1[j][59],csvS1[j][60],csvS1[j][61],csvS1[j][62],csvS1[j][63],xRotate,yRotate,zRotate);    //小 4 → 3
     weight =1.5;
   draw3DLineAndJoint(csvS1[j][61],csvS1[j][62],csvS1[j][63],csvS1[j][64],csvS1[j][65],csvS1[j][66],xRotate,yRotate,zRotate); //小 3 →小 2
     weight =0.5;
   draw3DLineAndJoint(csvS1[j][64],csvS1[j][65],csvS1[j][66],csvS1[j][67],csvS1[j][68],csvS1[j][69],xRotate,yRotate,zRotate);   //小 2 →小 1
     weight =1.0;
   draw3DLineAndJoint(csvS1[j][67],csvS1[j][68],csvS1[j][69],csvS1[j][70],csvS1[j][71],csvS1[j][72],xRotate,yRotate,zRotate);   //小 1 →小先
端
     weight =1.5;
  l11=0;
  }
 }
////////////////////////////////////////////////
//関数名：AnotherLeftAni
//概要：保存した右手の座標からアニメーションを動
//       かす。
//関連する関数
//   ・draw3DLineAndJoint
//   ・JudgeColorLeft<<左手の配色>>
//本当は DrawHands の中で行いたい
////////////////////////////////////////////////
void AnotherLeftAni(){
   AnotherJudgeColor_Left(); //左手の配色
  if(selectButton == 1){
   l21=1;
   //左手
   draw3DLineAndJoint(csvH1[j2][1]-csvH1[j2][x+y],csvH1[j2][2]-csvH1[j2][x+y+1],csvH1[j2][3]-csvH1[j2][x+y+2],csvH1[j2][4]-csvH1[j2][x+y],
   csvH1[j2][5]-csvH1[j2][x+y+1],csvH1[j2][6]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);//親 3 →親 2
     weight =1.0;
   draw3DLineAndJoint(csvH1[j2][1]-csvH1[j2][x+y],csvH1[j2][2]-csvH1[j2][x+y+1],csvH1[j2][3]-csvH1[j2][x+y+2],csvH1[j2][13]-csvH1[j2][x+y],
   csvH1[j2][14]-csvH1[j2][x+y+1],csvH1[j2][15]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);//親 3 →人 4
     weight =0.5;
   draw3DLineAndJoint(csvH1[j2][1]-csvH1[j2][x+y],csvH1[j2][2]-csvH1[j2][x+y+1],csvH1[j2][3]-csvH1[j2][x+y+2],csvH1[j2][16]-csvH1[j2][x+y],
   csvH1[j2][17]-csvH1[j2][x+y+1],csvH1[j2][18]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);//親 3 →人 3
     weight =1.5;
   draw3DLineAndJoint(csvH1[j2][4]-csvH1[j2][x+y],csvH1[j2][5]-csvH1[j2][x+y+1],csvH1[j2][6]-csvH1[j2][x+y+2],csvH1[j2][7]-csvH1[j2][x+y],
   csvH1[j2][8]-csvH1[j2][x+y+1],csvH1[j2][9]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);//親 2 →親 1
     weight =1.0;
   draw3DLineAndJoint(csvH1[j2][4]-csvH1[j2][x+y],csvH1[j2][5]-csvH1[j2][x+y+1],csvH1[j2][6]-csvH1[j2][x+y+2],csvH1[j2][16]-csvH1[j2][x+y],
   csvH1[j2][17]-csvH1[j2][x+y+1],csvH1[j2][18]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);//親 2 →人 3
```

```
    weight =0.5;
   draw3DLineAndJoint(csvH1[j2][7]-csvH1[j2][x+y],csvH1[j2][8]-csvH1[j2][x+y+1],csvH1[j2][9]-csvH1[j2][x+y+2],csvH1[j2][10]-csvH1[j2][x+y],
   csvH1[j2][11]-csvH1[j2][x+y+1],csvH1[j2][12]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);//親 1 →親先端
    weight =1.5;
   draw3DLineAndJoint(csvH1[j2][13]-csvH1[j2][x+y],csvH1[j2][14]-csvH1[j2][x+y+1],csvH1[j2][15]-csvH1[j2][x+y+2],csvH1[j2][28]-csvH1[j2][x+y],
   csvH1[j2][29]-csvH1[j2][x+y+1],csvH1[j2][30]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);//人 4 →中 4
    weight =1.0;
   draw3DLineAndJoint(csvH1[j2][16]-csvH1[j2][x+y],csvH1[j2][17]-csvH1[j2][x+y+1],csvH1[j2][18]-csvH1[j2][x+y+2],csvH1[j2][19]-csvH1[j2][x+y],
   csvH1[j2][20]-csvH1[j2][x+y+1],csvH1[j2][21]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);   //人 3 →人 2
    weight =0.5;
   draw3DLineAndJoint(csvH1[j2][16]-csvH1[j2][x+y],csvH1[j2][17]-csvH1[j2][x+y+1],csvH1[j2][18]-csvH1[j2][x+y+2],csvH1[j2][31]-csvH1[j2][x+y],
   csvH1[j2][32]-csvH1[j2][x+y+1],csvH1[j2][33]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);//人 3 →中 3
    weight =1.5;
   draw3DLineAndJoint(csvH1[j2][19]-csvH1[j2][x+y],csvH1[j2][20]-csvH1[j2][x+y+1],csvH1[j2][21]-csvH1[j2][x+y+2],csvH1[j2][22]-csvH1[j2][x+y],
   csvH1[j2][23]-csvH1[j2][x+y+1],csvH1[j2][24]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);   //人 2 →人 1
    weight =1.0;
   draw3DLineAndJoint(csvH1[j2][22]-csvH1[j2][x+y],csvH1[j2][23]-csvH1[j2][x+y+1],csvH1[j2][24]-csvH1[j2][x+y+2],csvH1[j2][25]-csvH1[j2][x+y],
   csvH1[j2][26]-csvH1[j2][x+y+1],csvH1[j2][27]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);    //人 1 →人先端
    weight =0.5;
   draw3DLineAndJoint(csvH1[j2][28]-csvH1[j2][x+y],csvH1[j2][29]-csvH1[j2][x+y+1],csvH1[j2][30]-csvH1[j2][x+y+2],csvH1[j2][43]-csvH1[j2][x+y],
   csvH1[j2][44]-csvH1[j2][x+y+1],csvH1[j2][45]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);//中 4 →薬 4
    weight =1.5;
   draw3DLineAndJoint(csvH1[j2][31]-csvH1[j2][x+y],csvH1[j2][32]-csvH1[j2][x+y+1],csvH1[j2][33]-csvH1[j2][x+y+2],csvH1[j2][34]-csvH1[j2][x+y],
   csvH1[j2][35]-csvH1[j2][x+y+1],csvH1[j2][36]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);   //中 3 →中 2
    weight =1.0;
   draw3DLineAndJoint(csvH1[j2][31]-csvH1[j2][x+y],csvH1[j2][32]-csvH1[j2][x+y+1],csvH1[j2][33]-csvH1[j2][x+y+2],csvH1[j2][46]-csvH1[j2][x+y],
   csvH1[j2][47]-csvH1[j2][x+y+1],csvH1[j2][48]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);//中 3 →薬 3
    weight =0.5;
   draw3DLineAndJoint(csvH1[j2][34]-csvH1[j2][x+y],csvH1[j2][35]-csvH1[j2][x+y+1],csvH1[j2][36]-csvH1[j2][x+y+2],csvH1[j2][37]-csvH1[j2][x+y],
   csvH1[j2][38]-csvH1[j2][x+y+1],csvH1[j2][39]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);   //中 2 →中 1
    weight =1.5;
   draw3DLineAndJoint(csvH1[j2][37]-csvH1[j2][x+y],csvH1[j2][38]-csvH1[j2][x+y+1],csvH1[j2][39]-csvH1[j2][x+y+2],csvH1[j2][40]-csvH1[j2][x+y],
   csvH1[j2][41]-csvH1[j2][x+y+1],csvH1[j2][42]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);    //中 1 →中先端
    weight =1.0;
   draw3DLineAndJoint(csvH1[j2][43]-csvH1[j2][x+y],csvH1[j2][44]-csvH1[j2][x+y+1],csvH1[j2][45]-csvH1[j2][x+y+2],csvH1[j2][58]-csvH1[j2][x+y],
   csvH1[j2][59]-csvH1[j2][x+y+1],csvH1[j2][60]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);//薬 4 →小 4
    weight =0.5;
   draw3DLineAndJoint(csvH1[j2][46]-csvH1[j2][x+y],csvH1[j2][47]-csvH1[j2][x+y+1],csvH1[j2][48]-csvH1[j2][x+y+2],csvH1[j2][49]-csvH1[j2][x+y],
   csvH1[j2][50]-csvH1[j2][x+y+1],csvH1[j2][51]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);   //薬 3 →薬 2
    weight =1.5;
   draw3DLineAndJoint(csvH1[j2][46]-csvH1[j2][x+y],csvH1[j2][47]-csvH1[j2][x+y+1],csvH1[j2][48]-csvH1[j2][x+y+2],csvH1[j2][61]-csvH1[j2][x+y],
   csvH1[j2][62]-csvH1[j2][x+y+1],csvH1[j2][63]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);//薬 3 →小 3
    weight =1.0;
   draw3DLineAndJoint(csvH1[j2][49]-csvH1[j2][x+y],csvH1[j2][50]-csvH1[j2][x+y+1],csvH1[j2][51]-csvH1[j2][x+y+2],csvH1[j2][52]-csvH1[j2][x+y],
   csvH1[j2][53]-csvH1[j2][x+y+1],csvH1[j2][54]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);   //薬 2 →薬 1
    weight =0.5;
   draw3DLineAndJoint(csvH1[j2][52]-csvH1[j2][x+y],csvH1[j2][53]-csvH1[j2][x+y+1],csvH1[j2][54]-csvH1[j2][x+y+2],csvH1[j2][55]-csvH1[j2][x+y],
   csvH1[j2][56]-csvH1[j2][x+y+1],csvH1[j2][57]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);    //薬 1 →薬先端
    weight =1.5;
   draw3DLineAndJoint(csvH1[j2][58]-csvH1[j2][x+y],csvH1[j2][59]-csvH1[j2][x+y+1],csvH1[j2][60]-csvH1[j2][x+y+2],csvH1[j2][61]-csvH1[j2][x+y],
   csvH1[j2][62]-csvH1[j2][x+y+1],csvH1[j2][63]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);    //小 4 → 3
    weight =1.0;
   draw3DLineAndJoint(csvH1[j2][61]-csvH1[j2][x+y],csvH1[j2][62]-csvH1[j2][x+y+1],csvH1[j2][63]-csvH1[j2][x+y+2],csvH1[j2][64]-csvH1[j2][x+y],
   csvH1[j2][65]-csvH1[j2][x+y+1],csvH1[j2][66]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate); //小 3 →小 2
    weight =0.5;
   draw3DLineAndJoint(csvH1[j2][64]-csvH1[j2][x+y],csvH1[j2][65]-csvH1[j2][x+y+1],csvH1[j2][66]-csvH1[j2][x+y+2],csvH1[j2][67]-csvH1[j2][x+y],
   csvH1[j2][68]-csvH1[j2][x+y+1],csvH1[j2][69]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);   //小 2 →小 1
    weight =1.5;
   draw3DLineAndJoint(csvH1[j2][67]-csvH1[j2][x+y],csvH1[j2][68]-csvH1[j2][x+y+1],csvH1[j2][69]-csvH1[j2][x+y+2],csvH1[j2][70]-csvH1[j2][x+y],
   csvH1[j2][71]-csvH1[j2][x+y+1],csvH1[j2][72]-csvH1[j2][x+y+2],xRotate,yRotate,zRotate);   //小 1 →小先端
    weight =1.0;
   l2l=0;
  }
  else{
   l2l=1;
   //左手
   draw3DLineAndJoint(csvH1[j2][1],csvH1[j2][2],csvH1[j2][3],csvH1[j2][4],csvH1[j2][5],csvH1[j2][6],xRotate,yRotate,zRotate);//親 3 →親 2
    weight =1.0;
   draw3DLineAndJoint(csvH1[j2][1],csvH1[j2][2],csvH1[j2][3],csvH1[j2][13],csvH1[j2][14],csvH1[j2][15],xRotate,yRotate,zRotate);//親 3 →人 4
    weight =0.5;
   draw3DLineAndJoint(csvH1[j2][1],csvH1[j2][2],csvH1[j2][3],csvH1[j2][16],csvH1[j2][17],csvH1[j2][18],xRotate,yRotate,zRotate);//親 3 →人 3
    weight =1.5;
   draw3DLineAndJoint(csvH1[j2][4],csvH1[j2][5],csvH1[j2][6],csvH1[j2][7],csvH1[j2][8],csvH1[j2][9],xRotate,yRotate,zRotate);//親 2 →親 1
    weight =1.0;
   draw3DLineAndJoint(csvH1[j2][4],csvH1[j2][5],csvH1[j2][6],csvH1[j2][16],csvH1[j2][17],csvH1[j2][18],xRotate,yRotate,zRotate);//親 2 →人 3
    weight =0.5;
   draw3DLineAndJoint(csvH1[j2][7],csvH1[j2][8],csvH1[j2][9],csvH1[j2][10],csvH1[j2][11],csvH1[j2][12],xRotate,yRotate,zRotate);//親 1 →親先
端
    weight =1.5;
   draw3DLineAndJoint(csvH1[j2][13],csvH1[j2][14],csvH1[j2][15],csvH1[j2][28],csvH1[j2][29],csvH1[j2][30],xRotate,yRotate,zRotate);//人 4 →
中 4
    weight =1.0;
   draw3DLineAndJoint(csvH1[j2][16],csvH1[j2][17],csvH1[j2][18],csvH1[j2][19],csvH1[j2][20],csvH1[j2][21],xRotate,yRotate,zRotate);   //人 3
→人 2
    weight =0.5;
   draw3DLineAndJoint(csvH1[j2][16],csvH1[j2][17],csvH1[j2][18],csvH1[j2][31],csvH1[j2][32],csvH1[j2][33],xRotate,yRotate,zRotate);//人 3 →
中 3
    weight =1.5;
   draw3DLineAndJoint(csvH1[j2][19],csvH1[j2][20],csvH1[j2][21],csvH1[j2][22],csvH1[j2][23],csvH1[j2][24],xRotate,yRotate,zRotate);   //人 2
→人 1
    weight =1.0;
```

```
   draw3DLineAndJoint(csvH1[j2][22],csvH1[j2][23],csvH1[j2][24],csvH1[j2][25],csvH1[j2][26],csvH1[j2][27],xRotate,yRotate,zRotate);    //人
1 →人先端
     weight =0.5;
   draw3DLineAndJoint(csvH1[j2][28],csvH1[j2][29],csvH1[j2][30],csvH1[j2][43],csvH1[j2][44],csvH1[j2][45],xRotate,yRotate,zRotate);//中 4 →
薬 4
     weight =1.5;
   draw3DLineAndJoint(csvH1[j2][31],csvH1[j2][32],csvH1[j2][33],csvH1[j2][34],csvH1[j2][35],csvH1[j2][36],xRotate,yRotate,zRotate);   //中 3
→中 2
     weight =1.0;
   draw3DLineAndJoint(csvH1[j2][31],csvH1[j2][32],csvH1[j2][33],csvH1[j2][46],csvH1[j2][47],csvH1[j2][48],xRotate,yRotate,zRotate);//中 3 →
薬 3
     weight =0.5;
   draw3DLineAndJoint(csvH1[j2][34],csvH1[j2][35],csvH1[j2][36],csvH1[j2][37],csvH1[j2][38],csvH1[j2][39],xRotate,yRotate,zRotate);   //中 2
→中 1
     weight =1.5;
   draw3DLineAndJoint(csvH1[j2][37],csvH1[j2][38],csvH1[j2][39],csvH1[j2][40],csvH1[j2][41],csvH1[j2][42],xRotate,yRotate,zRotate);    //中
1 →中先端
     weight =1.0;
   draw3DLineAndJoint(csvH1[j2][43],csvH1[j2][44],csvH1[j2][45],csvH1[j2][58],csvH1[j2][59],csvH1[j2][60],xRotate,yRotate,zRotate);//薬 4 →
小 4
     weight =0.5;
   draw3DLineAndJoint(csvH1[j2][46],csvH1[j2][47],csvH1[j2][48],csvH1[j2][49],csvH1[j2][50],csvH1[j2][51],xRotate,yRotate,zRotate);   //薬 3
→薬 2
     weight =1.5;
   draw3DLineAndJoint(csvH1[j2][46],csvH1[j2][47],csvH1[j2][48],csvH1[j2][61],csvH1[j2][62],csvH1[j2][63],xRotate,yRotate,zRotate);//薬 3 →
小 3
     weight =1.0;
   draw3DLineAndJoint(csvH1[j2][49],csvH1[j2][50],csvH1[j2][51],csvH1[j2][52],csvH1[j2][53],csvH1[j2][54],xRotate,yRotate,zRotate);   //薬 2
→薬 1
     weight =0.5;
   draw3DLineAndJoint(csvH1[j2][52],csvH1[j2][53],csvH1[j2][54],csvH1[j2][55],csvH1[j2][56],csvH1[j2][57],xRotate,yRotate,zRotate);    //薬
1 →薬先端
     weight =1.5;
   draw3DLineAndJoint(csvH1[j2][58],csvH1[j2][59],csvH1[j2][60],csvH1[j2][61],csvH1[j2][62],csvH1[j2][63],xRotate,yRotate,zRotate);    //小
4 → 3
     weight =1.0;
   draw3DLineAndJoint(csvH1[j2][61],csvH1[j2][62],csvH1[j2][63],csvH1[j2][64],csvH1[j2][65],csvH1[j2][66],xRotate,yRotate,zRotate); //小  3
→小 2
     weight =0.5;
   draw3DLineAndJoint(csvH1[j2][64],csvH1[j2][65],csvH1[j2][66],csvH1[j2][67],csvH1[j2][68],csvH1[j2][69],xRotate,yRotate,zRotate);   //小 2
→小 1
     weight =1.5;
   draw3DLineAndJoint(csvH1[j2][67],csvH1[j2][68],csvH1[j2][69],csvH1[j2][70],csvH1[j2][71],csvH1[j2][72],xRotate,yRotate,zRotate);   //小 1
→小先端
     weight =1.0;
   l2l=0;
  }
}
////////////////////////////////////////////
//関数名：JudgeColor_Right
//概要：右手が静止状態なら赤色へ、動作状態なら黒
//       色へ変化する。
//関連する関数
//   ・For_Finite_Animation<<ループの禁止>>
////////////////////////////////////////////
void JudgeColor_Right(){
  For_Finite_Animation();  //ループの禁止

  for(int b=0; b< csvWidth; b++){
   if(csvF1[i][b] > csvF1[i+1][b]){ColorAni =strokeA2;}
   if(csvF1[i][b] < csvF1[i+1][b]){ColorAni =strokeA2;}
   if (csvF1[i][b] == csvF1[i+1][b]){ColorAni =strokeA1;}
     //線の色・太さ
  stroke(ColorAni, strokeA2, strokeA2);
  fill(ColorAni, strokeA2, strokeA2);
 }
}
////////////////////////////////////////////
//関数名：AnotherJudgeColor_Right
//概要：右手が静止状態なら赤色へ、動作状態なら黒
//       色へ変化する。
//関連する関数
//   ・For_Finite_Animation<<ループの禁止>>
////////////////////////////////////////////
void AnotherJudgeColor_Right()
{
  For_Finite_Animation();  //ループの禁止

    for(int b=0; b< csvWidth; b++){
   if(csvT1[i2][b] > csvT1[i2+1][b]){ColorAni =strokeA2; NextColorAni=strokeA1;}
   if(csvT1[i2][b] < csvT1[i2+1][b]){ColorAni =strokeA2; NextColorAni=strokeA1;}
   if (csvT1[i2][b] == csvT1[i2+1][b]){ColorAni =strokeA1; NextColorAni=strokeA2;}
     //線の色・太さ
  stroke(ColorAni, strokeA2, NextColorAni);
  fill(ColorAni, strokeA2, NextColorAni);
 }
}
////////////////////////////////////////////
//関数名：JudgeColor_Left
//概要：左手が静止状態なら赤色へ、動作状態なら黒
//       色へ変化する。
```

```
//関連する関数
//   ・For_Finite_Animation<<ループの禁止>>
////////////////////////////////////////////
void JudgeColor_Left()
{
  For_Finite_Animation();  //ループの禁止

   for(int b=0; b< csvWidth; b++){
   if(csvS1[j][b] > csvS1[j+1][b]){ColorAni =strokeA2;}
   if(csvS1[j][b] < csvS1[j+1][b]){ColorAni =strokeA2;}
   if (csvS1[j][b] == csvS1[j+1][b]){ColorAni =strokeA1;}
     //線の色・太さ
  stroke(ColorAni, strokeA2, strokeA2);
  fill(ColorAni, strokeA2, strokeA2);
 }
}
////////////////////////////////////////////
//関数名：AnotherJudgeColor_Left
//概要：左手 2 が静止状態なら赤色へ、動作状態なら黒
//       色へ変化する。
//関連する関数
//   ・For_Finite_Animation<<ループの禁止>>
////////////////////////////////////////////
void AnotherJudgeColor_Left()
{
  For_Finite_Animation();  //ループの禁止

    for(int b=0; b< csvWidth; b++){
   if(csvH1[j2][b] > csvH1[j2+1][b]){ColorAni =strokeA2; NextColorAni=strokeA1;}
   if(csvH1[j2][b] < csvH1[j2+1][b]){ColorAni =strokeA2; NextColorAni=strokeA1;}
   if (csvH1[j2][b] == csvH1[j2+1][b]){ColorAni =strokeA1; NextColorAni=strokeA2;}
     //線の色・太さ
  stroke(ColorAni, strokeA2, NextColorAni);
  fill(ColorAni, strokeA2, NextColorAni);
 }
}
////////////////////////////////////////////
//関数名：For_Finite_Animation
//概要：両手のアニメーションがループしない条件分
//       岐をする。
////////////////////////////////////////////
void For_Finite_Animation()
{
   if(i+1 >= LENGTHF){i=LENGTHF-2; Rightspeed=0; sl1.setValue(0);}
  if(i+1 <= 0){i = 2; Rightspeed=0; sl1.setValue(0); }
 if(j+1 >= LENGTHS){j=LENGTHS-2; Leftspeed=0; sl2.setValue(0);}
 if(j+1 <= 0){j = 2; Leftspeed=0; sl2.setValue(0);}
 if(i2+1 >= LENGTHT){i2=LENGTHT-2; Rightspeed2=0; sl7.setValue(0);}
 if(i2+1 <= 0){i2 = 2; Rightspeed2=0; sl7.setValue(0);}
 if(j2+1 >= LENGTHH){j2=LENGTHH-2; Leftspeed2=0;  sl8.setValue(0);}
 if(j2+1 <= 0){j2 = 2; Leftspeed2=0; sl8.setValue(0);}
}

////////////////////////////////////////////
//関数名：DrawQuadratic_function
//概要：二次関数の時間変位グラフを描画する。
//関連する関数
//   ・DrawBiaxial<<縦軸と横軸の設定>>
//   ・DrawLocus<<軌跡の設定>>
////////////////////////////////////////////
void DrawQuadratic_function()
{
  DrawBiaxial();
  DrawLocus();
}
////////////////////////////////////////////
//関数名：DisplayRectAndScale
//概要：表示するデータ名を囲う四角と縦横軸の目
//       盛りを表すのに使用する不変的なものたち
////////////////////////////////////////////
void DisplayRectAndScale()
{
  fill(255,255,255);
  rect(515,185,80,75); //右手データ 1 や線などを囲う

  //画面右上にある縦横軸の各目盛りについて
  line(460,50,465,45); //縦軸の間隔、左から右
  line(510,50,505,45); //縦軸の間隔、右から左
  line(510,50,515,55); //横軸の間隔、上から下
  line(510,95,515,90); //横軸の間隔、下から上
  fill(strokeQ2,strokeQ2,strokeQ2);
  stroke(strokeQ2,strokeQ2,strokeQ2);
  //縦軸・横軸について
  text("経過時間",520,390); //横軸
  text(" (秒) ",530,405);
  text("(mm)",10,85);  //Y 軸 (縦軸)
}
////////////////////////////////////////////
//関数名：ChangeVerticalAxis
//概要：縦軸の目盛りの表示が変化する
////////////////////////////////////////////
```

```
void ChangeVerticalAxis()
{
    //縦軸の名称
  if(DwAx ==0){text("座標",10,70);} //ただのグラフ
  else if(DwAx ==1){text("変位",10,70);} //時間経過グラフ
/*
  else if(DwAx ==2){text("速度（変位）",10,70);} //時間経過グラフの速度
  else if(DwAx ==3){text("速度（座標）",10,70);} //ただのグラフの速度
  else if(DwAx ==4){text("加速度（変位）",10,70);} //時間経過グラフの加速度
  else if(DwAx ==5){text("加速度（座標）",10,70);} //ただのグラフの加速度
 */
}
////////////////////////////////////////////
//関数名：ChangeNameInRect
//概要：右真ん中にある四角形のデータ名を表示する
//      のかを判断する.
////////////////////////////////////////////
void ChangeNameInRect()
{
  //^^e2^^91^^a0 表示データ
  if(fitdata1 ==1)
  {
    weight =1.0;
    strokeWeight(weight);
    line(520,200,530,200);
    weight =0.5;
    text("右データ 1",535,205);
  }

  if(fitdata2 ==1)
  {
    weight =1.0;
    strokeWeight(weight);
    line(520,215,530,215);
    weight =0.5;
    text("左データ 1",535,220);
  }

  if(fitdata3 ==1)
  {
    stroke(strokeQ2,strokeQ2,strokeQ1);
    fill(strokeQ2,strokeQ2,strokeQ1);
    weight =1.0;
    strokeWeight(weight);
    line(520,230,530,230);
    weight =0.5;
    fill(strokeQ2,strokeQ2,strokeQ2);
    stroke(strokeQ2,strokeQ2,strokeQ2);
    text("右データ 2",535,235);
  }

  if(fitdata4 ==1)
  {
    stroke(strokeQ2,strokeQ2,strokeQ1);
    fill(strokeQ2,strokeQ2,strokeQ1);
    weight =1.0;
    strokeWeight(weight);
    line(520,245,530,245);
    weight =0.5;
    fill(strokeQ2,strokeQ2,strokeQ2);
    stroke(strokeQ2,strokeQ2,strokeQ2);
    text("左データ 2",535,250);
  }
}
////////////////////////////////////////////
//関数名：ChangeHorizontalAxis
//概要：横軸の目盛り線と円をどのような値にする
//      のかを設定する.
////////////////////////////////////////////
void ChangeHorizontalAxis()
{
  //^^e2^^91^^a0Y 軸（縦軸）
  for(int tate =0; tate <9; tate++)
  {
    line(60,50+tate*45,510,50+tate*45); //線描画
    ellipse(60,50+tate*45,4,4); //目盛りの値

    if(SaveData ==0)
    {
      AxisP =100;
      AxisM =-100;
    }

    if(SaveData ==1)
    {
      //位置座標をそのまま表示するグラフの場合
      if(DwAx ==0) //位置座標そのままのグラフ
      {
        //右手の設定
        if(d==0) //右手
        {
```

```
    if(fitdata1 ==1 && fitdata3 ==0)
    {
      AxisP =int(maxF1[x+y+e]+1); //軸の最大値
      AxisM =int(minF1[x+y+e]-1); //軸の最小値
    }
    if(fitdata1 ==0 && fitdata3 ==1)
    {
      AxisP =int(maxT1[x+y+e]+1); //軸の最大値
      AxisM =int(minT1[x+y+e]-1); //軸の最小値
    }
    if(fitdata1 ==1 && fitdata3 ==1)
    {
      //軸の最大値
      if(maxF1[x+y+e] >= maxT1[x+y+e]){AxisP = int(maxF1[x+y+e]+1);}
      if(maxF1[x+y+e] < maxT1[x+y+e]){AxisP = int(maxT1[x+y+e]+1);}
      //軸の最小値
      if(minF1[x+y+e] >= minT1[x+y+e]){AxisM = int(minT1[x+y+e]-1);}
      if(minF1[x+y+e] < minT1[x+y+e]){AxisM = int(minF1[x+y+e]-1);}
    }
  }

  //左手の設定
  if(d==1) //左手
  {
    if(fitdata2 ==1 && fitdata4 ==0)
    {
      AxisP =int(maxS1[x+y+e]+1); //軸の最大値
      AxisM =int(minS1[x+y+e]-1); //軸の最小値
    }
    if(fitdata2 ==0 && fitdata4 ==1)
    {
      AxisP =int(maxH1[x+y+e]+1); //軸の最大値
      AxisM =int(minH1[x+y+e]-1); //軸の最小値
    }
    if(fitdata2 ==1 && fitdata4 ==1)
    {
      //軸の最大値
      if(maxS1[x+y+e] >= maxH1[x+y+e]){AxisP = int(maxS1[x+y+e]+1);}
      if(maxS1[x+y+e] < maxH1[x+y+e]){AxisP = int(maxH1[x+y+e]+1);}
      //軸の最小値
      if(minS1[x+y+e] >= minH1[x+y+e]){AxisM = int(minH1[x+y+e]-1);}
      if(minS1[x+y+e] < minH1[x+y+e]){AxisM = int(minS1[x+y+e]-1);}
    }
  }

}

//初期値を引いた位置座標の変位グラフを表示する場合
if(DwAx ==1) //変位グラフ
{
  //右手の設定
  if(d==0) //右手
  {
    if(fitdata1 ==1 && fitdata3 ==0)
    {
      AxisP =int(maxF0[x+y+e]+1); //軸の最大値
      AxisM =int(-maxF0[x+y+e]-1); //軸の最小値
    }
    if(fitdata1 ==0 && fitdata3 ==1)
    {
      AxisP =int(maxT0[x+y+e]+1); //軸の最大値
      AxisM =int(-maxT0[x+y+e]-1); //軸の最小
    }
    if(fitdata1 ==1 && fitdata3 ==1)
    {
      //軸の最大値・最小値
      if(maxF0[x+y+e] >= maxT0[x+y+e]){AxisP = int(maxF0[x+y+e]+1); AxisM = int(-maxF0[x+y+e]-1);}
      if(maxF0[x+y+e] < maxT0[x+y+e]){AxisP = int(maxT0[x+y+e]+1); AxisM = int(-maxT0[x+y+e]-1);}
    }
  }
  //左手の設定
  if(d==1) //左手
  {
    if(fitdata2 ==1 && fitdata4 ==0)
    {
      AxisP =int(maxS0[x+y+e]+1); //軸の最大値
      AxisM =int(-maxS0[x+y+e]-1); //軸の最小値
    }
    if(fitdata2 ==0 && fitdata4 ==1)
    {
      AxisP =int(maxH0[x+y+e]+1); //軸の最大値
      AxisM =int(-maxH0[x+y+e]-1); //軸の最小値
    }
    if(fitdata2 ==1 && fitdata4 ==1)
    {
      //軸の最大値・最小値
      if(maxS0[x+y+e] >= maxH0[x+y+e]){AxisP = int(maxS0[x+y+e]+1); AxisM = int(-maxS0[x+y+e]-1);}
      if(maxS0[x+y+e] < maxH0[x+y+e]){AxisP = int(maxH0[x+y+e]+1); AxisM = int(-maxH0[x+y+e]-1);}
    }
  }
}
```

```
      //変位グラフから位置座標グラフへ移動する
      if(DwAx ==2){DwAx =0;}

      DifAxi =AxisP-AxisM+numberboxS-numberboxT; //軸の最大値と最小値との差

      text(nf(numberboxS+AxisP-tate*DifAxi/8,1,1),20+x_moveQ,55+tate*45); //縦軸上に表示
    }
  }

  if(VerAxisIs ==1)
  {
    //横軸の最大値・最小値の修正
    MaxAxis.setValue(int(AxisP));
    MinAxis.setValue(int(AxisM));
    VerAxisIs =0;
  }

}
////////////////////////////////////////////
//関数名：DrawBiaxial
//概要：縦軸と横軸の設定
//関連する関数
//      ・ChangeHorizontalAxis<<二次関数における横軸の目盛り線と円の設定>>
////////////////////////////////////////////
void SelectAnalysies()
{
  if(selectButton == 2)
 {
   ChangeHorizontalAxis(); //横軸の目盛り線と円の設定
 }
 if(selectButton == 5)
 {

 }
}
////////////////////////////////////////////
//関数名：DrawBiaxial
//概要：縦軸と横軸の設定
//関連する関数
//    ・DisplayRectAndScale
//    ・ChangeVerticalAxis
//    ・ChangeNameInRect
//    ・SelectAnalysis
////////////////////////////////////////////
void DrawBiaxial()
{
  DisplayRectAndScale(); //データを囲う四角と各軸の目盛りで不変
  ChangeVerticalAxis(); //縦軸の目盛りの表示が変化
  ChangeNameInRect(); //四角内にあるデータ名を表示するか否か

  strokeWeight(weight);

  //二次関数のグラフをどの分析手法で表示するのかを確認
  SelectAnalysies();   //二次関数のグラフをどの分析手法で表示するのかを確認

  //^^e2^^91^^a1 時間推移の軸（横軸）
  if(SaveData ==1)
  {
    for(int yoko =0; yoko <10; yoko++)
    {

      text(nf(rangeXaxis.getArrayValue(0)+(yoko*(rangeXaxis.getArrayValue(1)-rangeXaxis.getArrayValue(0))/9),1,1),50*yoko+x_moveQ+60,425); //
      line(50*yoko+x_moveQ+60,50,50*yoko+x_moveQ+60,410);
      ellipse(50*yoko+x_moveQ+60,410,4,4);
    }
  }
  //縦軸と横軸の間隔について
  text(nf((rangeXaxis.getArrayValue(1)-rangeXaxis.getArrayValue(0))/9,1,1),470,40); //縦軸の目盛り表示
  text(" (秒) ",495,40);
  text(nf(DifAxi/9,1,1),520,75); //横軸
  text(" (mm) ",525,90);
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

////////////////////////////////////////////
//関数名：DrawLocus
//概要：軌跡の設定
////////////////////////////////////////////
void DrawLocus()
{
  For_Finite_Animation();


  //左手
  if(d==1)
  {
    if(fitdata2 ==1)
    {
      for(int m=3;m+1<LENGTHS;m++)
      {
```

```
  if(csvS1[m-1][x+y+e] == csvS1[m][x+y+e])
  {
    stroke(strokeQ1,strokeQ2,strokeQ2);
    fill(strokeQ1,strokeQ2,strokeQ2);
  }
  else
  {
    stroke(strokeQ2,strokeQ2,strokeQ2);
    fill(strokeQ2,strokeQ2,strokeQ2);
  }
  //軌跡グラフの描画
  FirPoX2 =((csvS2[m-1][0]/1000000-rangeXaxis.getArrayValue(0))*450/(rangeXaxis.getArrayValue(1)-rangeXaxis.getArrayValue(0)))+60+x_moveQ;
if(DwAx ==0) //位置座標
{
  FirPoY2 =-((csvS1[m-1][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ;
}
if(DwAx ==1) //変位
{
  FirPoY2 =-((csvS2[m-1][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ;
}
  NexPoX2 =((csvS2[m][0]/1000000-rangeXaxis.getArrayValue(0))*450/(rangeXaxis.getArrayValue(1)-rangeXaxis.getArrayValue(0)))+60+x_moveQ;
if(DwAx ==0) //位置座標
{
  NexPoY2 =-((csvS1[m][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ;
}
if(DwAx ==1) //変位
{
  NexPoY2 =-((csvS2[m][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ;
}
  weight =1.0;
  if(FirPoX2 < 60 ){FirPoX2 = 60; FirPoY2 = 50; NexPoX2 =60; NexPoY2 =50; weight =0.5;}
  if(FirPoY2 < 50){FirPoX2 = 510; FirPoY2 = 50; NexPoX2 =510; NexPoY2 =50; weight =0.5;}
  if(NexPoX2 > 510){FirPoX2 = 510; FirPoY2 = 410; NexPoX2 = 510; NexPoY2 = 410; weight =0.5;}
  if(NexPoY2 > 410){FirPoX2 = 60; FirPoY2 = 410; NexPoY2 = 60; NexPoY2 = 410; weight =0.5;}
  strokeWeight(weight);
  line(FirPoX2,FirPoY2,NexPoX2,NexPoY2);
  if(csvS[m-1][x+y+e].equals("Mark"))
  {
      stroke(strokeQ1,strokeQ1,strokeQ1);
      fill(strokeQ1,strokeQ2,strokeQ2);
      ellipse(FirPoX2,FirPoY2,5,5);
      stroke(strokeQ2,strokeQ2,strokeQ2);
      fill(strokeQ2,strokeQ2,strokeQ2);
  }
  weight =0.5;
       if(DwEl ==1)
       {
         stroke(strokeQ2,strokeQ2,strokeQ2);
         fill(strokeQ2,strokeQ2,strokeQ2);

  //時間経過軸に合わせた移動点の描画
if(DwAx ==0) //位置座標
{
  ellipse(((csvS2[j][0]/1000000-rangeXaxis.getArrayValue(0))*450/(rangeXaxis.getArrayValue(1)-rangeXaxis.getArrayValue(0)))+60+x_moveQ,
         -((csvS1[j][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ,
         5,5);
}
if(DwAx ==1) //変位
{
  ellipse(((csvS2[j][0]/1000000-rangeXaxis.getArrayValue(0))*450/(rangeXaxis.getArrayValue(1)-rangeXaxis.getArrayValue(0)))+60+x_moveQ,
         -((csvS2[j][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ,
         5,5);
}
       }
       if(DwEl ==2)
       {
         DwEl=0;
       }
 }
}
if(fitdata4 ==1)
{
 for(int m=3;m+1<LENGTHH;m++)
 {
   //保存したデータに合わせ軌跡の色を変換
   if(csvH1[m-1][x+y+e] == csvH1[m][x+y+e])
   {
     stroke(strokeQ1,strokeQ2,strokeQ2);
     fill(strokeQ1,strokeQ2,strokeQ2);
   }
   else
   {
     stroke(strokeQ2,strokeQ2,strokeQ1);
     fill(strokeQ2,strokeQ2,strokeQ1);
   }
   //軌跡グラフと点の描画
   //軌跡グラフの描画
  FirPoX4 =((csvH2[m-1][0]/1000000-rangeXaxis.getArrayValue(0))*450/(rangeXaxis.getArrayValue(1)-rangeXaxis.getArrayValue(0)))+60+x_moveQ;
if(DwAx ==0) //位置座標
{
  FirPoY4 =-((csvH1[m-1][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ;
```

```
  }
  if(DwAx ==1) //変位
  {
    FirPoY4 =-((csvH2[m-1][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ;
  }
    NexPoX4 =((csvH2[m][0]/1000000-rangeXaxis.getArrayValue(0))*450/(rangeXaxis.getArrayValue(1)-rangeXaxis.getArrayValue(0)))+60+x_moveQ;
  if(DwAx ==0) //位置座標
  {
    NexPoY4 =-((csvH1[m][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ;
  }
  if(DwAx ==1) //変位
  {
    NexPoY4 =-((csvH2[m][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ;
  }
    weight =1.0;
    if(FirPoX4 < 60 ){FirPoX4 = 60; FirPoY4 = 50; NexPoX4 =60; NexPoY4 =50; weight =0.5;}
    if(FirPoY4 < 50){FirPoX4 = 510; FirPoY4 = 50; NexPoX4 =510; NexPoY4 =50; weight =0.5;}
    if(NexPoX4 > 510){FirPoX4 = 510; FirPoY4 = 410; NexPoX4 = 510; NexPoY4 = 410; weight =0.5;}
    if(NexPoY4 > 410){FirPoX4 = 60; FirPoY4 = 410; NexPoY4 = 60; NexPoY4 = 410; weight =0.5;}
    strokeWeight(weight);
    line(FirPoX4,FirPoY4,NexPoX4,NexPoY4);
    if(csvH[m-1][x+y+e].equals("Mark"))
    {
        stroke(strokeQ1,strokeQ1,strokeQ1);
        fill(strokeQ1,strokeQ2,strokeQ2);
        ellipse(FirPoX4,FirPoY4,5,5);
        stroke(strokeQ2,strokeQ2,strokeQ2);
        fill(strokeQ2,strokeQ2,strokeQ2);
    }
           weight =0.5;
         if(DwEl ==1){
       stroke(strokeQ2,strokeQ2,strokeQ1);
       fill(strokeQ2,strokeQ2,strokeQ1);
    //時間経過軸に合わせた移動点の描画
  if(DwAx ==0) //位置座標
  {
    ellipse(((csvH2[j2][0]/1000000-rangeXaxis.getArrayValue(0))*450/(rangeXaxis.getArrayValue(1)-rangeXaxis.getArrayValue(0)))+60+x_moveQ,
           -((csvH1[j2][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ,
           5,5);
  }
  if(DwAx ==1) //変位
  {
    ellipse(((csvH2[j2][0]/1000000-rangeXaxis.getArrayValue(0))*450/(rangeXaxis.getArrayValue(1)-rangeXaxis.getArrayValue(0)))+60+x_moveQ,
           -((csvH2[j2][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ,
           5,5);
  }
          }
          if(DwEl ==2)
         {
           DwEl=0;
         }

     }
   }
   noFill();
   noStroke();
 }
//右手
 if(d==0)
{
  if(fitdata1 ==1)
  {
   for(int m=3;m+1<LENGTHF;m++)
   {
     if(csvF1[m-1][x+y+e] == csvF1[m][x+y+e])
     {
       stroke(strokeQ1,strokeQ2,strokeQ2);
       fill(strokeQ1,strokeQ2,strokeQ2);
     }
     else
     {
       stroke(strokeQ2,strokeQ2,strokeQ2);
       fill(strokeQ2,strokeQ2,strokeQ2);
      }
     //軌跡グラフと点の描画
     //軌跡グラフの描画
    FirPoX1 =((csvF2[m-1][0]/1000000-rangeXaxis.getArrayValue(0))*450/(rangeXaxis.getArrayValue(1)-rangeXaxis.getArrayValue(0)))+60+x_moveQ;
  if(DwAx ==0) //位置座標
  {
    FirPoY1 =-((csvF1[m-1][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ;
  }
  if(DwAx ==1) //変位
  {
    FirPoY1 =-((csvF2[m-1][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ;
  }
    NexPoX1 =((csvF2[m][0]/1000000-rangeXaxis.getArrayValue(0))*450/(rangeXaxis.getArrayValue(1)-rangeXaxis.getArrayValue(0)))+60+x_moveQ;
  if(DwAx ==0) //位置座標
  {
    NexPoY1 =-((csvF1[m][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ;
  }
  if(DwAx ==1) //変位
```

```
 {
   NexPoY1 =-((csvF2[m][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ;
 }
   weight =1.0;
   if(FirPoX1 < 60 ){FirPoX1 = 60; FirPoY1 = 50; NexPoX1 =60; NexPoY1 =50; weight =0.5;}
   if(FirPoY1 < 50){FirPoX1 = 510; FirPoY1 = 50; NexPoX1 =510; NexPoY1 =50; weight =0.5;}
   if(NexPoX1 > 510){FirPoX1 = 510; FirPoY1 = 410; NexPoX1 = 510; NexPoY1 = 410; weight =0.5;}
   if(NexPoY1 > 410){FirPoX1 = 60; FirPoY1 = 410; NexPoY1 = 60; NexPoY1 = 410; weight =0.5;}
   strokeWeight(weight);
   line(FirPoX1,FirPoY1,NexPoX1,NexPoY1);
   if(csvF[m-1][x+y+e].equals("Mark"))
   {
       stroke(strokeQ1,strokeQ1,strokeQ1);
       fill(strokeQ1,strokeQ2,strokeQ2);
       ellipse(FirPoX1,FirPoY1,5,5);
       stroke(strokeQ2,strokeQ2,strokeQ2);
       fill(strokeQ2,strokeQ2,strokeQ2);
   }
   weight =0.5;
         if(DwEl ==1)
         {
          stroke(strokeQ2,strokeQ2,strokeQ2);
          fill(strokeQ2,strokeQ2,strokeQ2);
   //時間経過軸に合わせた移動点の描画
         if(DwAx ==0) //位置座標
         {
           ellipse(((csvF2[i][0]/1000000-rangeXaxis.getArrayValue(0))*450/(rangeXaxis.getArrayValue(1)-rangeXaxis.getArrayValue(0)))+60+x_moveQ,
          -((csvF1[i][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ,
          5,5);
         }
         if(DwAx ==1) //変位
         {
           ellipse(((csvF2[i][0]/1000000-rangeXaxis.getArrayValue(0))*450/(rangeXaxis.getArrayValue(1)-rangeXaxis.getArrayValue(0)))+60+x_moveQ,
          -((csvF2[i][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ,
          5,5);
         }
         }
         if(DwEl ==2)
        {
          DwEl=0;
        }
 }
}
if(fitdata3 ==1)
{
  for(int m=3;m+1<LENGTHT;m++)
  {
    if(csvT1[m-1][x+y+e] == csvT1[m][x+y+e])
    {
      stroke(strokeQ1,strokeQ2,strokeQ2);
      fill(strokeQ1,strokeQ2,strokeQ2);
    }
    else
    {
      stroke(strokeQ2,strokeQ2,strokeQ1);
      fill(strokeQ2,strokeQ2,strokeQ1);
    }
    //軌跡グラフと点の描画
    //軌跡グラフの描画
   FirPoX3 =((csvT2[m-1][0]/1000000-rangeXaxis.getArrayValue(0))*450/(rangeXaxis.getArrayValue(1)-rangeXaxis.getArrayValue(0)))+60+x_moveQ;
 if(DwAx ==0) //位置座標
 {
   FirPoY3 =-((csvT1[m-1][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ;
 }
 if(DwAx ==1) //変位
 {
   FirPoY3 =-((csvT2[m-1][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ;
 }
   NexPoX3 =((csvT2[m][0]/1000000-rangeXaxis.getArrayValue(0))*450/(rangeXaxis.getArrayValue(1)-rangeXaxis.getArrayValue(0)))+60+x_moveQ;
 if(DwAx ==0) //位置座標
 {
   NexPoY3 =-((csvT1[m][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ;
 }
 if(DwAx ==1) //変位
 {
   NexPoY3 =-((csvT2[m][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ;
 }
   weight =1.0;
   if(FirPoX3 < 60 ){FirPoX3 = 60; FirPoY3 = 50; NexPoX3 =60; NexPoY3 =50; weight =0.5;}
   if(FirPoY3 < 50){FirPoX3 = 510; FirPoY3 = 50; NexPoX3 =510; NexPoY3 =50; weight =0.5;}
   if(NexPoX3 > 510){FirPoX3 = 510; FirPoY3 = 410; NexPoX3 = 510; NexPoY3 = 410; weight =0.5;}
   if(NexPoY3 > 410){FirPoX3 = 60; FirPoY3 = 410; NexPoY3 = 60; NexPoY3 = 410; weight =0.5;}
   strokeWeight(weight);
   line(FirPoX3,FirPoY3,NexPoX3,NexPoY3);
   if(csvT[m-1][x+y+e].equals("Mark"))
   {
       stroke(strokeQ1,strokeQ1,strokeQ1);
       fill(strokeQ1,strokeQ2,strokeQ2);
       ellipse(FirPoX3,FirPoY3,5,5);
       stroke(strokeQ2,strokeQ2,strokeQ2);
       fill(strokeQ2,strokeQ2,strokeQ2);
```

```
        }
        weight =0.5;
              if(DwEl ==1)
              {
           stroke(strokeQ2,strokeQ2,strokeQ1);
           fill(strokeQ2,strokeQ2,strokeQ1);
        //時間経過軸に合わせた移動点の描画
      if(DwAx ==0) //位置座標
      {
        ellipse(((csvT2[i2][0]/1000000-rangeXaxis.getArrayValue(0))*450/(rangeXaxis.getArrayValue(1)-rangeXaxis.getArrayValue(0)))+60+x_moveQ,
                -((csvT1[i2][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ,
                5,5);
      }
      if(DwAx ==1) //変位
      {
        ellipse(((csvT2[i2][0]/1000000-rangeXaxis.getArrayValue(0))*450/(rangeXaxis.getArrayValue(1)-rangeXaxis.getArrayValue(0)))+60+x_moveQ,
                -((csvT2[i2][x+y+e]-AxisM-numberboxT)*360/DifAxi)+410+y_moveQ,
                5,5);
      }
              }
              if(DwEl ==2)
             {
               DwEl=0;
             }
        }
      }
      noFill();
      noStroke();
    }
}


////////////////////////////////////////////
//関数名：DrawCubic_function
//概要：3D グラフでの軌跡を描く。
//関連する関数
//   ・DrawTriaxialAndLeapMotion<<各軸と LeapMotion の描画>>
//   ・DrawGraph<<軌跡の描画>>
////////////////////////////////////////////
void DrawCubic_function()
{
  selectLiAJo=2;
  DrawTriaxialAndLeapMotion(); //各軸と LeapMotion の描画
  DrawGraph(); //軌跡の描画
}

////////////////////////////////////////////
//関数名：DrawTriaxialAndLeapMotion
//概要：X,Y,Z 軸及び LeapMotion を描画する。
//関連する関数
//   ・draw3DLineAndJoint
//   ・Rec3 <<各平面の描画>>
//   ・LineForRec3<<各平面の目盛り線>>
//   ・ElliForRec3<<各平面の目盛りに円を描画>>
////////////////////////////////////////////////
void DrawTriaxialAndLeapMotion()
{
  Rec3(); //各平面の描画
  LineForRec3(); //各平面に目盛り線を描画
  ElliForRec3(); //各平面の目盛りに円を描画
  TextForRec3(); //各平面の目盛りの値を描画
  InfoAboutRec3(); //各平面を分析するための情報
//println(sel5);
}
////////////////////////////////////////////
//関数名：LIneForRec3
//概要：各平面に目盛りの線を引く。
//関連する関数
//   ・ThreeGraSepa
//   ・draw3DLineAndJoint
////////////////////////////////////////////
void LineForRec3()
{
  ThreeGraSepa =1; //直線
  //各平面の目盛り描画
  stroke(strokeC2,strokeC2,strokeC2,50); //黒色
  fill(strokeC2,strokeC2,strokeC2,50);

  //^^e2^^91^^a0X 軸
   for(int memori=1; memori <=10; memori++)
   {
     //左から右へ目盛り線
     //Y 軸が 0～200 と 200～400 で分けている
     draw3DLineAndJoint(-200,memori*20,0,200,memori*20,0,xRotate,yRotate,zRotate);
     draw3DLineAndJoint(-200,200+memori*20,0,200,200+memori*20,0,xRotate,yRotate,zRotate);

     //下から上へ目盛り線
     //X 軸が 0～200 と 0～-200 で分けている

     draw3DLineAndJoint(memori*20,0,0,memori*20,400,0,xRotate,yRotate,zRotate);
```

```
      draw3DLineAndJoint(-memori*20,0,0,-memori*20,400,0,xRotate,yRotate,zRotate);
      }

  //^^e2^^91^^a1Y 軸
   draw3DLineAndJoint(0,0,0,0,400,0,xRotate,yRotate,zRotate); //原点から Y 軸に伸びる線
   for(int memori=0; memori <=10; memori++){
        draw3DLineAndJoint(memori*20,0,-200,memori*20,0,200,xRotate,yRotate,zRotate);
        draw3DLineAndJoint(-memori*20,0,-200,-memori*20,0,200,xRotate,yRotate,zRotate);
        draw3DLineAndJoint(-200,0,memori*20,200,0,memori*20,xRotate,yRotate,zRotate);
        draw3DLineAndJoint(200,0,-memori*20,-200,0,-memori*20,xRotate,yRotate,zRotate);
//  ThreeGraSepa =0; //直線と円 (Y 軸の値)
        draw3DLineAndJoint(0,200+memori*20,0,0,200+memori*20,0,xRotate,yRotate,zRotate);
     }

  //^^e2^^91^^a2Z 軸
   for(int memori=1; memori <=10; memori++)
   {
     //後ろから前へ目盛り線
     //Y 軸が 0 から 200 と 200〜400 で分けている
     draw3DLineAndJoint(0,memori*20,-200,0,memori*20,200,xRotate,yRotate,zRotate);
     draw3DLineAndJoint(0,200+memori*20,-200,0,200+memori*20,200,xRotate,yRotate,zRotate);

     //下から上へ目盛り線
     //X 軸が 0〜200 と 0〜-200 で分けている
     draw3DLineAndJoint(0,0,memori*20,0,400,memori*20,xRotate,yRotate,zRotate);
     draw3DLineAndJoint(0,0,-memori*20,0,400,-memori*20,xRotate,yRotate,zRotate);
     }
}
////////////////////////////////////////////
//関数名：CondJoint
//概要：各平面に目盛りの円を描く際，回転して邪魔にならないようにする。
//関連する関数
//   ・xRotate, yRotate, zRotate //回転角
//   ・DeletJointXY, DelectXZ, DelectYZ //各平面を描画するかを決める
////////////////////////////////////////////
void ConfJoint()
{
  //本当は for 文で作成したかったができなかった.


 //XZ 平面
 if(xRotate == -PI*7/2){DeleteJointXY=0; DeleteJointXZ=1; DeleteJointYZ=0;}
 if(xRotate == -PI*5/2){DeleteJointXY=0; DeleteJointXZ=1; DeleteJointYZ=0;}
 if(xRotate == -PI*3/2){DeleteJointXY=0; DeleteJointXZ=1; DeleteJointYZ=0;}
 if(xRotate > -PI*1/2 -0.2 && xRotate < -PI*1/2 +0.2){DeleteJointXY=0; DeleteJointXZ=1; DeleteJointYZ=0;}
 if(xRotate > PI*1/2 -0.2 && xRotate < PI*1/2 +0.2){DeleteJointXY=0; DeleteJointXZ=1; DeleteJointYZ=0;}


 //println(Rotate);

 if(xRotate > -PI*3-0.2 && xRotate <-PI*3+0.2)
 {
   //XY 平面
   if(yRotate > -PI-0.2 && yRotate <-PI+0.2){DeleteJointXY=1; DeleteJointXZ=0; DeleteJointYZ=0;} //-PI →-360
   if(yRotate > -0.2 && yRotate < 0.2){DeleteJointXY=1; DeleteJointXZ=0; DeleteJointYZ=0;} //0 →-180
   if(yRotate > PI-0.2 && yRotate < PI+0.2){DeleteJointXY=1; DeleteJointXZ=0; DeleteJointYZ=0;} //PI → 0
   if(yRotate > PI*2-0.2 && yRotate < PI*2 +0.2){DeleteJointXY=1; DeleteJointXZ=0; DeleteJointYZ=0;} //2*PI → 180
   if(yRotate > PI*3-0.2 && yRotate <PI*3+0.2){DeleteJointXY=1; DeleteJointXZ=0; DeleteJointYZ=0;} //3*PI → 360

   //YZ 平面
   if(yRotate > -PI/2-0.2 && yRotate < -PI/2+0.2){DeleteJointXY=0; DeleteJointXZ=0; DeleteJointYZ=1;} //-PI/2 →-270
   if(yRotate > PI/2-0.2 && yRotate < PI/2+0.2){DeleteJointXY=0; DeleteJointXZ=0; DeleteJointYZ=1;} //PI/2 →-90
   if(yRotate > PI*3/2-0.2 && yRotate < PI*3/2+0.2){DeleteJointXY=0; DeleteJointXZ=0; DeleteJointYZ=1;} //PI*3/2 → 90
   if(yRotate > PI*5/2-0.2 && yRotate < PI*5/2+0.2){DeleteJointXY=0; DeleteJointXZ=0; DeleteJointYZ=1;} //PI*5/2 →-270
 }
 if(xRotate > -PI*2-0.2 && xRotate < -PI*2+0.2)
 {
   //XY 平面
   if(yRotate > -PI-0.2 && yRotate <-PI+0.2){DeleteJointXY=1; DeleteJointXZ=0; DeleteJointYZ=0;} //-PI →-360
   if(yRotate > -0.2 && yRotate < 0.2){DeleteJointXY=1; DeleteJointXZ=0; DeleteJointYZ=0;} //0 →-180
   if(yRotate > PI-0.2 && yRotate < PI+0.2){DeleteJointXY=1; DeleteJointXZ=0; DeleteJointYZ=0;} //PI → 0
   if(yRotate > PI*2-0.2 && yRotate < PI*2 +0.2){DeleteJointXY=1; DeleteJointXZ=0; DeleteJointYZ=0;} //2*PI → 180
   if(yRotate > PI*3-0.2 && yRotate <PI*3+0.2){DeleteJointXY=1; DeleteJointXZ=0; DeleteJointYZ=0;} //3*PI → 360

   //YZ 平面
   if(yRotate > -PI/2-0.2 && yRotate < -PI/2+0.2){DeleteJointXY=0; DeleteJointXZ=0; DeleteJointYZ=1;} //-PI/2 →-270
   if(yRotate > PI/2-0.2 && yRotate < PI/2+0.2){DeleteJointXY=0; DeleteJointXZ=0; DeleteJointYZ=1;} //PI/2 →-90
   if(yRotate > PI*3/2-0.2 && yRotate < PI*3/2+0.2){DeleteJointXY=0; DeleteJointXZ=0; DeleteJointYZ=1;} //PI*3/2 → 90
   if(yRotate > PI*5/2-0.2 && yRotate < PI*5/2+0.2){DeleteJointXY=0; DeleteJointXZ=0; DeleteJointYZ=1;} //PI*5/2 →-270
 }
 if(xRotate > -PI-0.2 && xRotate <-PI+0.2)//ここがうまくできなかった
 {
   //XY 平面
   if(yRotate > -PI-0.2 && yRotate <-PI+0.2){DeleteJointXY=1; DeleteJointXZ=0; DeleteJointYZ=0;} //-PI →-360
   if(yRotate > -0.2 && yRotate < 0.2){DeleteJointXY=1; DeleteJointXZ=0; DeleteJointYZ=0;} //0 →-180
   if(yRotate > PI-0.2 && yRotate < PI+0.2){DeleteJointXY=1; DeleteJointXZ=0; DeleteJointYZ=0;} //PI → 0
   if(yRotate > PI*2-0.2 && yRotate < PI*2 +0.2){DeleteJointXY=1; DeleteJointXZ=0; DeleteJointYZ=0;} //2*PI → 180
   if(yRotate > PI*3-0.2 && yRotate <PI*3+0.2){DeleteJointXY=1; DeleteJointXZ=0; DeleteJointYZ=0;} //3*PI → 360

   //YZ 平面
   if(yRotate > -PI/2-0.2 && yRotate < -PI/2+0.2){DeleteJointXY=0; DeleteJointXZ=0; DeleteJointYZ=1;} //-PI/2 →-270
```

```
   if(yRotate > PI/2-0.2 && yRotate < PI/2+0.2){DeleteJointXY=0; DeleteJointXZ=0; DeleteJointYZ=1;} //PI/2 →-90
   if(yRotate > PI*3/2-0.2 && yRotate < PI*3/2+0.2){DeleteJointXY=0; DeleteJointXZ=0; DeleteJointYZ=1;} //PI*3/2 → 90
   if(yRotate > PI*5/2-0.2 && yRotate < PI*5/2+0.2){DeleteJointXY=0; DeleteJointXZ=0; DeleteJointYZ=1;} //PI*5/2 →-270
 }
if(xRotate > -0.2 && xRotate < 0.2)
 {
   //XY 平面
   if(yRotate > -PI-0.2 && yRotate <-PI+0.2){DeleteJointXY=1; DeleteJointXZ=0; DeleteJointYZ=0;} //-PI →-360
   if(yRotate > -0.2 && yRotate <0.2){DeleteJointXY=1; DeleteJointXZ=0; DeleteJointYZ=0;} //0 →-180
   if(yRotate > PI-0.2 && yRotate < PI+0.2){DeleteJointXY=1; DeleteJointXZ=0; DeleteJointYZ=0;} //PI → 0
   if(yRotate > PI*2-0.2 && yRotate < PI*2 +0.2){DeleteJointXY=1; DeleteJointXZ=0; DeleteJointYZ=0;} //2*PI → 180
   if(yRotate > PI*3-0.2 && yRotate <PI*3+0.2){DeleteJointXY=1; DeleteJointXZ=0; DeleteJointYZ=0;} //3*PI → 360

   //YZ 平面
   if(yRotate > -PI/2-0.2 && yRotate < -PI/2+0.2){DeleteJointXY=0; DeleteJointXZ=0; DeleteJointYZ=1;} //-PI/2 →-270
   if(yRotate > PI/2-0.2 && yRotate < PI/2+0.2){DeleteJointXY=0; DeleteJointXZ=0; DeleteJointYZ=1;} //PI/2 →-90
   if(yRotate > PI*3/2-0.2 && yRotate < PI*3/2+0.2){DeleteJointXY=0; DeleteJointXZ=0; DeleteJointYZ=1;} //PI*3/2 → 90
   if(yRotate > PI*5/2-0.2 && yRotate < PI*5/2+0.2){DeleteJointXY=0; DeleteJointXZ=0; DeleteJointYZ=1;} //PI*5/2 →-270
 }
 //yRotate で見やすいようにしたかったができなかった。
/*
 //普通に見づらいから消去
   if(yRotate > -PI-0.2 && yRotate <-PI+0.2){DeleteJointXY=1; DeleteJointXZ=1; DeleteJointYZ=0;} //-PI →-360
   if(yRotate > -PI/2-0.2 && yRotate < -PI/2+0.2){DeleteJointXY=0; DeleteJointXZ=1; DeleteJointYZ=1;} //-PI/2 →-270
   if(yRotate > -0.2 && yRotate < 0.2){DeleteJointXY=1; DeleteJointXZ=1; DeleteJointYZ=0;} //0 →-180

   if(yRotate > PI/2-0.2 && yRotate < PI/2+0.2){DeleteJointXY=0; DeleteJointXZ=1; DeleteJointYZ=1;} //PI/2 →-90

   if(yRotate > PI-0.2 && yRotate < PI+0.2){
   if(zRotate > PI*3/2-0.2 && zRotate < PI*3/2+0.2)
   {
   DeleteJointXY=0; DeleteJointXZ=0; DeleteJointYZ=0;
 }} //PI → 0 //なかなかできないので

   if(yRotate > PI*3/2-0.2 && yRotate < PI*3/2+0.2){DeleteJointXY=0; DeleteJointXZ=1; DeleteJointYZ=1;} //PI*3/2 → 90

   if(yRotate > PI*2-0.2 && yRotate < PI*2 +0.2){DeleteJointXY=1; DeleteJointXZ=1; DeleteJointYZ=0;} //2*PI → 180

   if(yRotate > PI*5/2-0.2 && yRotate < PI*5/2+0.2){DeleteJointXY=0; DeleteJointXZ=1; DeleteJointYZ=1;} //PI*5/2 →-270

   if(yRotate > PI*3-0.2 && yRotate <PI*3+0.2){DeleteJointXY=1; DeleteJointXZ=1; DeleteJointYZ=0;} //3*PI → 360


*/

 }


////////////////////////////////////////////
//関数名：ElliForRec3
//概要：各平面に目盛りの線を引く。
//関連する関数
//   ・ThreeGraSepa
//   ・draw3DLineAndJoint
////////////////////////////////////////////
void ElliForRec3()
{
  ThreeGraSepa =0; //直線
  //各平面の目盛り描画
  stroke(strokeC2,strokeC2,strokeC2,50); //黒色
  fill(strokeC2,strokeC2,strokeC2,50);

  DeleteJointXY=1; //XY 平面を選択するか否か
  DeleteJointXZ=1; //XZ 平面を選択するか否か
  DeleteJointYZ=1; //YZ 平面を選択するか否か

  ConfJoint(); //回転角チェックして各平面を描画するか決める

  if(DeleteJointXY==1)
  {
    //^^e2^^91^^a0XY
    for(int memori=1; memori <=10; memori++)
    {
      //左から右へ目盛り線
      //Y 軸が 0～200 と 200～400 で分けている
      draw3DLineAndJoint(-200,memori*20,0,200,memori*20,0,xRotate,yRotate,zRotate);
      draw3DLineAndJoint(-200,200+memori*20,0,200,200+memori*20,0,xRotate,yRotate,zRotate);

      //下から上へ目盛り線
      //X 軸が 0～200 と 0～-200 で分けている
      draw3DLineAndJoint(memori*20,0,0,memori*20,400,0,xRotate,yRotate,zRotate);

      draw3DLineAndJoint(-memori*20,0,0,-memori*20,400,0,xRotate,yRotate,zRotate);
    }
    fill(strokeC1,strokeC2,strokeC2);
    //XY 平面
    //Y 軸
    draw3DLineAndJoint(200,100,0,200,200,0,xRotate,yRotate,zRotate);
    draw3DLineAndJoint(-200,100,0,-200,200,0,xRotate,yRotate,zRotate);
    draw3DLineAndJoint(200,300,0,-200,300,0,xRotate,yRotate,zRotate);
```

```
    //X 軸
    draw3DLineAndJoint(0,0,0,0,400,0,xRotate,yRotate,zRotate);
    draw3DLineAndJoint(100,0,0,-100,0,0,xRotate,yRotate,zRotate);
    draw3DLineAndJoint(100,400,0,-100,400,0,xRotate,yRotate,zRotate);
    draw3DLineAndJoint(200,0,0,200,400,0,xRotate,yRotate,zRotate);
    draw3DLineAndJoint(-200,0,0,-200,400,0,xRotate,yRotate,zRotate);
  }
//  ConfJoint();
  fill(strokeC2,strokeC2,strokeC2);
if(DeleteJointXY==0){}
if(DeleteJointXZ==1){

  //^^e2^^91^^a1XZ
   draw3DLineAndJoint(0,0,0,0,400,0,xRotate,yRotate,zRotate); //原点から Y 軸に伸びる線
   for(int memori=0; memori <=10; memori++){
        draw3DLineAndJoint(memori*20,0,-200,memori*20,0,200,xRotate,yRotate,zRotate);
        draw3DLineAndJoint(-memori*20,0,-200,-memori*20,0,200,xRotate,yRotate,zRotate);
        draw3DLineAndJoint(-200,0,memori*20,200,0,memori*20,xRotate,yRotate,zRotate);
        draw3DLineAndJoint(200,0,-memori*20,-200,0,-memori*20,xRotate,yRotate,zRotate);
//  ThreeGraSepa =0; //直線と円 (Y 軸の値)
        draw3DLineAndJoint(0,memori*20,0,0,memori*20,0,xRotate,yRotate,zRotate);
        draw3DLineAndJoint(0,200+memori*20,0,0,200+memori*20,0,xRotate,yRotate,zRotate);
     }
   fill(strokeC1,strokeC2,strokeC2);
 //XZ 平面
  //X 軸
  draw3DLineAndJoint(100,0,200,-100,0,200,xRotate,yRotate,zRotate);
  draw3DLineAndJoint(100,0,-200,-100,0,-200,xRotate,yRotate,zRotate);
  draw3DLineAndJoint(0,0,200,200,0,200,xRotate,yRotate,zRotate);
  draw3DLineAndJoint(0,0,-200,-200,0,-200,xRotate,yRotate,zRotate);
  draw3DLineAndJoint(200,0,-200,-200,0,200,xRotate,yRotate,zRotate);

  //Z 軸
  draw3DLineAndJoint(200,0,100,200,0,-100,xRotate,yRotate,zRotate);
  draw3DLineAndJoint(-200,0,100,-200,0,-100,xRotate,yRotate,zRotate);
  draw3DLineAndJoint(200,0,0,-200,0,0,xRotate,yRotate,zRotate);
  }
if(DeleteJointXZ==0){}
//  ConfJoint();
  fill(strokeC2,strokeC2,strokeC2);

if(DeleteJointYZ==1){

  //^^e2^^91^^a2YZ
   for(int memori=1; memori <=10; memori++)
   {
     //後ろから前へ目盛り線
     //Y 軸が 0 から 200 と 200〜400 で分けている
     draw3DLineAndJoint(0,memori*20,-200,0,memori*20,200,xRotate,yRotate,zRotate);
     draw3DLineAndJoint(0,200+memori*20,-200,0,200+memori*20,200,xRotate,yRotate,zRotate);

     //下から上へ目盛り線
     //X 軸が 0〜200 と 0〜-200 で分けている
     draw3DLineAndJoint(0,0,memori*20,0,400,memori*20,xRotate,yRotate,zRotate);
     draw3DLineAndJoint(0,0,-memori*20,0,400,-memori*20,xRotate,yRotate,zRotate);
     }
   fill(strokeC1,strokeC2,strokeC2);

  //YZ 平面
  //Y 軸
  draw3DLineAndJoint(0,100,200,0,200,200,xRotate,yRotate,zRotate);
  draw3DLineAndJoint(0,100,-200,0,200,-200,xRotate,yRotate,zRotate);
  draw3DLineAndJoint(0,300,200,0,300,-200,xRotate,yRotate,zRotate);

  //Z 軸
  draw3DLineAndJoint(0,0,0,0,400,0,xRotate,yRotate,zRotate);
  draw3DLineAndJoint(0,0,100,0,0,-100,xRotate,yRotate,zRotate);
  draw3DLineAndJoint(0,400,100,0,400,-100,xRotate,yRotate,zRotate);
  draw3DLineAndJoint(0,0,200,0,400,200,xRotate,yRotate,zRotate);
  draw3DLineAndJoint(0,0,-200,0,400,-200,xRotate,yRotate,zRotate);
  }

  fill(strokeC2,strokeC2,strokeC2);
if(DeleteJointYZ==0){}
}

////////////////////////////////////////////
//関数名：TextForRec3
//概要：目盛りの値を描画する。
//関連する関数
//     ・CondJoint
//     ・draw3DLineAndJoint
////////////////////////////////////////////

void TextForRec3()
{

   //各軸の値
  ThreeGraSepa =2; //XZ 軸の値

  ConfJoint(); //回転角チェックして各平面を描画するか決める
```

```
if(DeleteJointXZ==1)
{
  //XZ 平面
  //X 軸
  DisplayA=100;
  DisplayB=-100;
  draw3DLineAndJoint(100,0,200,-100,0,200,xRotate,yRotate,zRotate);
  draw3DLineAndJoint(100,0,-200,-100,0,-200,xRotate,yRotate,zRotate);

  DisplayA=0;
  DisplayB=200;
  draw3DLineAndJoint(0,0,200,200,0,200,xRotate,yRotate,zRotate);

  DisplayB=-200;
  draw3DLineAndJoint(0,0,-200,-200,0,-200,xRotate,yRotate,zRotate);

  DisplayA=200;
  DisplayB=-200;
  draw3DLineAndJoint(200,0,-200,-200,0,200,xRotate,yRotate,zRotate);

  //Z 軸
  DisplayA=100;
  DisplayB=-100;
  draw3DLineAndJoint(200,0,100,200,0,-100,xRotate,yRotate,zRotate);
  draw3DLineAndJoint(-200,0,100,-200,0,-100,xRotate,yRotate,zRotate);

  DisplayA=0;
  DisplayB=0;
  draw3DLineAndJoint(200,0,0,-200,0,0,xRotate,yRotate,zRotate);
}


if(DeleteJointXY ==1)
{
  //XY 平面
  //Y 軸
  DisplayA=100;
  DisplayB=200;
  draw3DLineAndJoint(200,100,0,200,200,0,xRotate,yRotate,zRotate);
  draw3DLineAndJoint(-200,100,0,-200,200,0,xRotate,yRotate,zRotate);

  DisplayA=300;
  DisplayB=300;
  draw3DLineAndJoint(200,300,0,-200,300,0,xRotate,yRotate,zRotate);

  //X 軸
  DisplayA=0;
  DisplayB=0;
  draw3DLineAndJoint(0,-15,0,0,400,0,xRotate,yRotate,zRotate);

  DisplayA=100;
  DisplayB=-100;
  draw3DLineAndJoint(100,-15,0,-100,-15,0,xRotate,yRotate,zRotate);
  draw3DLineAndJoint(100,400,0,-100,400,0,xRotate,yRotate,zRotate);

  DisplayA=200;
  DisplayB=200;
  draw3DLineAndJoint(200,-15,0,200,415,0,xRotate,yRotate,zRotate);

  DisplayA=-200;
  DisplayB=-200;
  draw3DLineAndJoint(-200,-15,0,-200,415,0,xRotate,yRotate,zRotate);

  DisplayA=0;
  DisplayB=400;
  draw3DLineAndJoint(200,0,0,200,400,0,xRotate,yRotate,zRotate);
  draw3DLineAndJoint(-200,0,0,-200,400,0,xRotate,yRotate,zRotate);


}
if(DeleteJointYZ ==1){
  //YZ 平面
  //Y 軸
  DisplayA=100;
  DisplayB=200;
  draw3DLineAndJoint(0,100,200,0,200,200,xRotate,yRotate,zRotate);
  draw3DLineAndJoint(0,100,-200,0,200,-200,xRotate,yRotate,zRotate);

  DisplayA=300;
  DisplayB=300;
  draw3DLineAndJoint(0,300,200,0,300,-200,xRotate,yRotate,zRotate);

  //Z 軸
  DisplayA=0;
  DisplayB=0;
  draw3DLineAndJoint(0,-15,0,0,400,0,xRotate,yRotate,zRotate);

  DisplayA=100;
  DisplayB=-100;
  draw3DLineAndJoint(0,-15,100,0,-15,-100,xRotate,yRotate,zRotate);
```

```
  draw3DLineAndJoint(0,400,100,0,400,-100,xRotate,yRotate,zRotate);

  DisplayA=200;
  DisplayB=200;
  draw3DLineAndJoint(0,-15,200,0,415,200,xRotate,yRotate,zRotate);

  DisplayA=-200;
  DisplayB=-200;
  draw3DLineAndJoint(0,-15,-200,0,415,-200,xRotate,yRotate,zRotate);

  DisplayA=0;
  DisplayB=400;
  draw3DLineAndJoint(0,0,200,0,400,200,xRotate,yRotate,zRotate);
  draw3DLineAndJoint(0,0,-200,0,400,-200,xRotate,yRotate,zRotate);

}
}


////////////////////////////////////////////
//関数名：InfoAboutRec3
//概要：各平面の各目盛りの値が 20 × 20(mm) であることをを描画する。
//関連する関数
//
////////////////////////////////////////////
void InfoAboutRec3()
{
  text("20(mm)",503,50); //単位表示
  text("20 (mm)",553,85); //単位表示
  line(550,65,555,70); //下の左から右
  line(585,65,580,70); //下の右から左
  line(550,30,545,35); //左の上から下
  line(550,65,545,60); //左の下から上
  fill(255,255,255);
  rect(550,30,35,35);
}
////////////////////////////////////////////
//関数名：Rec3
//概要：3D 空間における各平面を描画する。
//関連する関数
//   ・ThreeGraSepa
//   ・draw3DQuad
////////////////////////////////////////////
void Rec3()
{
  //各平面の描画（配色）
  //^^e2^^91^^a0X 軸（赤に配色）
  ThreeGraSepa =2; //X 軸
  draw3DQuad(
  -200,0,0,
  200,0,0,
  200,400,0,
  -200,400,0,
  xRotate,yRotate,zRotate);

  //^^e2^^91^^a1Y 軸（青に配色）
  ThreeGraSepa =3; //Y 軸
  draw3DQuad(
  -200,0,-200,
  200,0,-200,
  200,0,200,
  -200,0,200,
  xRotate,yRotate,zRotate);

  //^^e2^^91^^a2Z 軸（緑に配色）
  ThreeGraSepa =4; //Z 軸
  draw3DQuad(
  0,0,-200,
  0,0,200,
  0,400,200,
  0,400,-200,
  xRotate,yRotate,zRotate);
}


////////////////////////////////////////////
//関数名：DrawGraph
//概要：3D 空間における軌跡を描画する。
//関連する関数
//   ・draw3DLineAndJoint
////////////////////////////////////////////
void DrawGraph()
{
  ThreeGraSepa =1; //軌跡

  //左手
  if(d==1)
  {
    if(fitdata2 ==1)
    {
      for(int m=3;m<j+1;m++)
```

```
    {
      selectLiAJo=2;
      stroke(strokeC2,strokeC2,strokeC2);
      fill(strokeC2,strokeC2,strokeC2);
      draw3DLineAndJoint(csvS1[m-1][x+y],
                         csvS1[m-1][x+y+1],
                         csvS1[m-1][x+y+2],
                         csvS1[m][x+y],
                         csvS1[m][x+y+1],
                         csvS1[m][x+y+2],
                         xRotate,
                         yRotate,
                         zRotate
                         );
     }
     selectLiAJo=3;
     stroke(strokeC2,strokeC2,strokeC2);
     fill(strokeC1,strokeC2,strokeC2);
     draw3DLineAndJoint(csvS1[j][x+y],
                        csvS1[j][x+y+1],
                        csvS1[j][x+y+2],
                        csvS1[j][x+y],
                        csvS1[j][x+y+1],
                        csvS1[j][x+y+2],
                        xRotate,
                        yRotate,
                        zRotate
                        );
  }
  if(fitdata4 ==1)
  {
    for(int m=3;m<j2+1;m++)
    {
      selectLiAJo=2;
      stroke(strokeC2,strokeC2,strokeC1);
      fill(strokeC2,strokeC2,strokeC1);
      draw3DLineAndJoint(csvH1[m-1][x+y],
                         csvH1[m-1][x+y+1],
                         csvH1[m-1][x+y+2],
                         csvH1[m][x+y],
                         csvH1[m][x+y+1],
                         csvH1[m][x+y+2],
                         xRotate,
                         yRotate,
                         zRotate
                         );
     }
     selectLiAJo=3;
     stroke(strokeC2,strokeC2,strokeC2);
     fill(strokeC1,strokeC2,strokeC2);
     draw3DLineAndJoint(csvH1[j2][x+y],
                        csvH1[j2][x+y+1],
                        csvH1[j2][x+y+2],
                        csvH1[j2][x+y],
                        csvH1[j2][x+y+1],
                        csvH1[j2][x+y+2],
                        xRotate,
                        yRotate,
                        zRotate
                        );
  }
}
//右手
 if(d==0)
{
  if(fitdata1 ==1)
  {
    for(int m=3;m<i+1;m++)
    {
      selectLiAJo=2;
      stroke(strokeC2,strokeC2,strokeC2);
      fill(strokeC2,strokeC2,strokeC2);
      draw3DLineAndJoint(csvF1[m-1][x+y],
                         csvF1[m-1][x+y+1],
                         csvF1[m-1][x+y+2],
                         csvF1[m][x+y],
                         csvF1[m][x+y+1],
                         csvF1[m][x+y+2],
                         xRotate,
                         yRotate,
                         zRotate
                         );
    }
    selectLiAJo=3;
    stroke(strokeC2,strokeC2,strokeC2);
    fill(strokeC1,strokeC2,strokeC2);
    draw3DLineAndJoint(csvF1[i][x+y],
                       csvF1[i][x+y+1],
                       csvF1[i][x+y+2],
                       csvF1[i][x+y],
                       csvF1[i][x+y+1],
```

```
                        csvF1[i][x+y+2],
                        xRotate,
                        yRotate,
                        zRotate
                        );
    }
    if(fitdata3 ==1)
    {
     for(int m=3;m<i2+1;m++)
     {
       selectLiAJo=2;
       stroke(strokeC2,strokeC2,strokeC1);
       fill(strokeC2,strokeC2,strokeC1);
       draw3DLineAndJoint(csvT1[m-1][x+y],
                          csvT1[m-1][x+y+1],
                          csvT1[m-1][x+y+2],
                          csvT1[m][x+y],
                          csvT1[m][x+y+1],
                          csvT1[m][x+y+2],
                          xRotate,
                          yRotate,
                          zRotate
                          );
     }
     selectLiAJo=3;
     stroke(strokeC2,strokeC2,strokeC2);
     fill(strokeC1,strokeC2,strokeC2);
     draw3DLineAndJoint(csvT1[i2][x+y],
                        csvT1[i2][x+y+1],
                        csvT1[i2][x+y+2],
                        csvT1[i2][x+y],
                        csvT1[i2][x+y+1],
                        csvT1[i2][x+y+2],
                        xRotate,
                        yRotate,
                        zRotate
                        );
    }
  }
}


////////////////////////////////////////////
//関数名：draw3DLineAndJoint
//概要：関節と関節間が透視変換を通して描画される。
////////////////////////////////////////////
//透視変換 x1, y1, z1, x2, y2, z2, rotateX, rotateY, rotateZ
void draw3DLineAndJoint(float x1,float y1, float z1, float x2, float y2, float z2, float rx, float ry, float rz)
{
  //Y 軸の回転量を反映
  float z1cash = z1;
  float z2cash = z2;
  z1 = x1 * sin(ry) - z1cash * cos(ry);
  z2 = x2 * sin(ry) - z2cash * cos(ry);
  x1 = x1 * cos(ry) + z1cash * sin(ry);
  x2 = x2 * cos(ry) + z2cash * sin(ry);
  //X 軸の回転量を反映
  z1cash = z1;
  z2cash = z2;
  z1 = y1 * sin(rx) - z1cash * cos(rx);
  z2 = y2 * sin(rx) - z2cash * cos(rx);
  y1 = y1 * cos(rx) + z1cash * sin(rx);
  y2 = y2 * cos(rx) + z2cash * sin(rx);
  //Z 軸の回転量を反映
  float x1cash = x1;
  float x2cash = x2;
  float y1cash = y1;
  float y2cash = y2;
  x1 = x1 * sin(rz) + y1cash * cos(rz);
  x2 = x2 * sin(rz) + y2cash * cos(rz);
  y1 = y1 * sin(rz) - x1cash * cos(rz);
  y2 = y2 * sin(rz) - x2cash * cos(rz);
  //透視変換
  float x1b = width/2  + x1 * (camZ + z1) * tan(radS/2) / (scZ + z1) * tan(radS/2);
  float y1b = height/2 + y1 * (camZ + z1) * tan(radS/2) / (scZ + z1) * tan(radS/2);
  float x2b = width/2  + x2 * (camZ + z2) * tan(radS/2) / (scZ + z2) * tan(radS/2);
  float y2b = height/2 + y2 * (camZ + z2) * tan(radS/2) / (scZ + z2) * tan(radS/2);
  //描画開始位置を初期化
  translate(0, 0);
  //描画
  //アニメーション
  if(selectButton == 1)
  {
    if(r1r == 1){
        strokeWeight(weight);
    line(
         -x1b+300+4*sl3.arrayValue()[0]+addmX1, /////////////////
         y1b+4*sl3.arrayValue()[1]-200+addmY1,
         -x2b+300+4*sl3.arrayValue()[0]+addmX1,
         y2b+4*sl3.arrayValue()[1]-200+addmY1
         );
```

```
    ellipse(
            -x1b+300+4*sl3.arrayValue()[0]+addmX1,
            y1b+4*sl3.arrayValue()[1]-200+addmY1,
            3,
            3
            );
    }
    if(11l ==1){
        strokeWeight(weight);

    line(
         -x1b+300+4*sl3.arrayValue()[0]+addmX2, /////////////////
         y1b+4*sl3.arrayValue()[1]-200+addmY2,
         -x2b+300+4*sl3.arrayValue()[0]+addmX2,
         y2b+4*sl3.arrayValue()[1]-200+addmY2
         );
    ellipse(
            -x1b+300+4*sl3.arrayValue()[0]+addmX2,
            y1b+4*sl3.arrayValue()[1]-200+addmY2,
            3,
            3
            );
    }
    if(r2r ==1){
  strokeWeight(weight);

    line(
         -x1b+300+4*sl3.arrayValue()[0]+addmX3, /////////////////
         y1b+4*sl3.arrayValue()[1]-200+addmY3,
         -x2b+300+4*sl3.arrayValue()[0]+addmX3,
         y2b+4*sl3.arrayValue()[1]-200+addmY3
         );
    ellipse(
            -x1b+300+4*sl3.arrayValue()[0]+addmX3,
            y1b+4*sl3.arrayValue()[1]-200+addmY3,
            3,
            3
            );
    }
     if(12l ==1){
         strokeWeight(weight);

    line(
         -x1b+300+4*sl3.arrayValue()[0]+addmX4, /////////////////
         y1b+4*sl3.arrayValue()[1]-200+addmY4,
         -x2b+300+4*sl3.arrayValue()[0]+addmX4,
         y2b+4*sl3.arrayValue()[1]-200+addmY4
         );
    ellipse(
            -x1b+300+4*sl3.arrayValue()[0]+addmX4,
            y1b+4*sl3.arrayValue()[1]-200+addmY4,
            3,
            3
            );
    }




  }
  //３Ｄグラフ
   if(selectButton == 3)
  {

    if(ThreeGraSepa ==0) //円と線
    {
      ellipse(
         -x1b+300+4*sl3.arrayValue()[0],
         y1b+4*sl3.arrayValue()[1]-200,
         3,3);
      ellipse(
         -x2b+300+4*sl3.arrayValue()[0],
         y2b+4*sl3.arrayValue()[1]-200,
         3,3);
       }
    if(ThreeGraSepa ==1) //線描画
   {
     line(
         -x1b+300+4*sl3.arrayValue()[0],
         y1b+4*sl3.arrayValue()[1]-200,
         -x2b+300+4*sl3.arrayValue()[0],
         y2b+4*sl3.arrayValue()[1]-200
          );
   }
  if(ThreeGraSepa ==2) //目盛りの値描画
  {


if(DisplayA == 200 && DisplayB == -200)
{
  if(sel5 ==0){
```

```
       //軸の値
 text("± 200",
       -x1b+300+4*sl3.arrayValue()[0], //x1 =x2
       y1b+4*sl3.arrayValue()[1]-200); //y1 =y4

 text("± 200",
       -x2b+300+4*sl3.arrayValue()[0], //x1 =x2
       y2b+4*sl3.arrayValue()[1]-200); //y1 =y4
  }

}

else{
  if(sel5 ==0){
      //軸の値
 text(DisplayA,
       -x1b+300+4*sl3.arrayValue()[0], //x1 =x2
       y1b+4*sl3.arrayValue()[1]-200); //y1 =y4

 text(DisplayB,
       -x2b+300+4*sl3.arrayValue()[0], //x1 =x2
       y2b+4*sl3.arrayValue()[1]-200); //y1 =y4
  }
}

  }




  }
  //アニメーションと３Ｄグラフ
   if(selectButton == 4)
  {
    if(selectLiAJo == 1)
    {
      line(
           -x1b+300+4*sl3.arrayValue()[0],
           y1b+4*sl3.arrayValue()[1]-200,
           -x2b+300+4*sl3.arrayValue()[0],
           y2b+4*sl3.arrayValue()[1]-200
           );
      ellipse(
              -x1b+300+4*sl3.arrayValue()[0],
              y1b+4*sl3.arrayValue()[1]-200,
              3,
              3
               );
    }
     if(selectLiAJo == 2)
    {
      line(
           -x1b+300+4*sl3.arrayValue()[0],
           y1b+4*sl3.arrayValue()[1]-200,
           -x2b+300+4*sl3.arrayValue()[0],
           y2b+4*sl3.arrayValue()[1]-200
           );
    }
     if(selectLiAJo == 3)
    {
      ellipse(
              -x1b+300+4*sl3.arrayValue()[0],
              y1b+4*sl3.arrayValue()[1]-200,
              3,
              3
              );
    }
    //ここから 3D グラフに関する内容入れるか否か
    if(ThreeGraSepa ==0) //円と線
    {
      ellipse(
         -x1b+300+4*sl3.arrayValue()[0],
         y1b+4*sl3.arrayValue()[1]-200,
         3,3);
      ellipse(
         -x2b+300+4*sl3.arrayValue()[0],
         y2b+4*sl3.arrayValue()[1]-200,
         3,3);
       }
    if(ThreeGraSepa ==1) //線描画
   {
     line(
         -x1b+300+4*sl3.arrayValue()[0],
         y1b+4*sl3.arrayValue()[1]-200,
         -x2b+300+4*sl3.arrayValue()[0],
         y2b+4*sl3.arrayValue()[1]-200
          );
   }
  if(ThreeGraSepa ==2) //目盛りの値描画
  {
```

```
if(DisplayA == 200 && DisplayB == -200)
{
  if(sel5 ==0){
        //軸の値
 text("± 200",
       -x1b+300+4*sl3.arrayValue()[0], //x1 =x2
       y1b+4*sl3.arrayValue()[1]-200); //y1 =y4

 text("± 200",
       -x2b+300+4*sl3.arrayValue()[0], //x1 =x2
       y2b+4*sl3.arrayValue()[1]-200); //y1 =y4
  }

}

else{
  if(sel5 ==0)
  {
      //軸の値
 text(DisplayA,
       -x1b+300+4*sl3.arrayValue()[0], //x1 =x2
       y1b+4*sl3.arrayValue()[1]-200); //y1 =y4

 text(DisplayB,
       -x2b+300+4*sl3.arrayValue()[0], //x1 =x2
       y2b+4*sl3.arrayValue()[1]-200); //y1 =y4
  }
}
  }
  }
  //アニメーションと二次関数のグラフ
   if(selectButton == 5)
  {
    line(
         -x1b+300+4*sl3.arrayValue()[0]+x_moveA,
         y1b+4*sl3.arrayValue()[1]-200+y_moveA,
         -x2b+300+4*sl3.arrayValue()[0]+x_moveA,
         y2b+4*sl3.arrayValue()[1]-200+y_moveA
         );
  ellipse(
          -x1b+300+4*sl3.arrayValue()[0]+x_moveA,
          y1b+4*sl3.arrayValue()[1]-200+y_moveA,
          3,
          3
          );
  }
  //すべて
   if(selectButton == 6)
  {
    if(selectLiAJo == 1)
    {
      line(
           -x1b+300+4*sl3.arrayValue()[0]+x_moveA,
           y1b+4*sl3.arrayValue()[1]-200+y_moveA,
           -x2b+300+4*sl3.arrayValue()[0]+x_moveA,
           y2b+4*sl3.arrayValue()[1]-200+y_moveA
           );
      ellipse(
               -x1b+300+4*sl3.arrayValue()[0]+x_moveA,
               y1b+4*sl3.arrayValue()[1]-200+y_moveA,
               3,
               3
               );
    }
     if(selectLiAJo == 2)
    {
      line(
           -x1b+300+4*sl3.arrayValue()[0]+x_moveC,
           y1b+4*sl3.arrayValue()[1]-200+y_moveC,
           -x2b+300+4*sl3.arrayValue()[0]+x_moveC,
           y2b+4*sl3.arrayValue()[1]-200+y_moveC
           );
    }
     if(selectLiAJo == 3)
    {
      ellipse(
               -x1b+300+4*sl3.arrayValue()[0]+x_moveC,
               y1b+4*sl3.arrayValue()[1]-200+y_moveC,
               3,
               3
               );
    }
  }
}

//3 次元における平面の描画関数
//透視変換 x1, y1, z1, x2, y2, z2, x3, y3, z3, x4, y4, z4, rotateX, rotateY, rotateZ
void draw3DQuad(float x1, float y1, float z1,
float x2, float y2, float z2,
float x3, float y3, float z3,
```

```
float x4, float y4, float z4,
float rx, float ry, float rz)
{
  //Y 軸の回転量を反映
  float z1cash = z1;
  float z2cash = z2;
  float z3cash = z3;
  float z4cash = z4;
  z1 = x1 * sin(ry) - z1cash * cos(ry);
  z2 = x2 * sin(ry) - z2cash * cos(ry);
  z3 = x3 * sin(ry) - z3cash * cos(ry);
  z4 = x4 * sin(ry) - z4cash * cos(ry);
  x1 = x1 * cos(ry) + z1cash * sin(ry);
  x2 = x2 * cos(ry) + z2cash * sin(ry);
  x3 = x3 * cos(ry) + z3cash * sin(ry);
  x4 = x4 * cos(ry) + z4cash * sin(ry);
  //X 軸の回転量を反映
  z1cash = z1;
  z2cash = z2;
  z3cash = z3;
  z4cash = z4;
  z1 = y1 * sin(rx) - z1cash * cos(rx);
  z2 = y2 * sin(rx) - z2cash * cos(rx);
  z3 = y3 * sin(rx) - z3cash * cos(rx);
  z4 = y4 * sin(rx) - z4cash * cos(rx);
  y1 = y1 * cos(rx) + z1cash * sin(rx);
  y2 = y2 * cos(rx) + z2cash * sin(rx);
  y3 = y3 * cos(rx) + z3cash * sin(rx);
  y4 = y4 * cos(rx) + z4cash * sin(rx);
  //Z 軸の回転量を反映
  float x1cash = x1;
  float x2cash = x2;
  float x3cash = x3;
  float x4cash = x4;
  float y1cash = y1;
  float y2cash = y2;
  float y3cash = y3;
  float y4cash = y4;
  x1 = x1 * sin(rz) + y1cash * cos(rz);
  x2 = x2 * sin(rz) + y2cash * cos(rz);
  x3 = x3 * sin(rz) + y3cash * cos(rz);
  x4 = x4 * sin(rz) + y4cash * cos(rz);
  y1 = y1 * sin(rz) - x1cash * cos(rz);
  y2 = y2 * sin(rz) - x2cash * cos(rz);
  y3 = y3 * sin(rz) - x3cash * cos(rz);
  y4 = y4 * sin(rz) - x4cash * cos(rz);
  //透視変換
  float x1b = width/2  + x1 * (camZ + z1) * tan(radS/2) / (scZ + z1) * tan(radS/2);
  float y1b = height/2 + y1 * (camZ + z1) * tan(radS/2) / (scZ + z1) * tan(radS/2);
  float x2b = width/2  + x2 * (camZ + z2) * tan(radS/2) / (scZ + z2) * tan(radS/2);
  float y2b = height/2 + y2 * (camZ + z2) * tan(radS/2) / (scZ + z2) * tan(radS/2);
  float x3b = width/2  + x3 * (camZ + z3) * tan(radS/2) / (scZ + z3) * tan(radS/2);
  float y3b = height/2 + y3 * (camZ + z3) * tan(radS/2) / (scZ + z3) * tan(radS/2);
  float x4b = width/2  + x4 * (camZ + z4) * tan(radS/2) / (scZ + z4) * tan(radS/2);
  float y4b = height/2 + y4 * (camZ + z4) * tan(radS/2) / (scZ + z4) * tan(radS/2);
  //描画開始位置を初期化
  translate(0, 0);

  //^^e2^^91^^a0X 軸（赤に配色）
  if(ThreeGraSepa ==2) //X 軸
  {
    stroke(strokeC1,strokeC2,strokeC2,50); //赤色
    fill(strokeC1,strokeC2,strokeC2,50); //赤色
  }
  //^^e2^^91^^a1Y 軸（青に配色）
  if(ThreeGraSepa ==3) //Y 軸
  {
    stroke(strokeC2,strokeC1,strokeC2,50); //青色
    fill(strokeC2,strokeC1,strokeC2,50); //青色
  }
  //^^e2^^91^^a2Z 軸（緑に配色）
  if(ThreeGraSepa ==4) //Z 軸
  {
    stroke(strokeC2,strokeC2,strokeC1,50); //緑色
    fill(strokeC2,strokeC2,strokeC1,50); //緑色
  }

  //描画
     quad(
            -x1b+300+4*sl3.arrayValue()[0], //x1 =x2
            y1b+4*sl3.arrayValue()[1]-200, //y1 =y4
            -x2b+300+4*sl3.arrayValue()[0], //x2 =x1
            y2b+4*sl3.arrayValue()[1]-200, //y2 =y3
            -x3b+300+4*sl3.arrayValue()[0], //x3 =x3
            y3b+4*sl3.arrayValue()[1]-200, //y3 =y2
            -x4b+300+4*sl3.arrayValue()[0], //x4 =x3
            y4b+4*sl3.arrayValue()[1]-200 //y4 =y1
            );
  stroke(strokeC2,strokeC2,strokeC2); //黒色
  noFill();
```

```
  quad(
            -x1b+300+4*sl3.arrayValue()[0], //x1 =x2
            y1b+4*sl3.arrayValue()[1]-200, //y1 =y4
            -x2b+300+4*sl3.arrayValue()[0], //x2 =x1
            y2b+4*sl3.arrayValue()[1]-200, //y2 =y3
            -x3b+300+4*sl3.arrayValue()[0], //x3 =x3
            y3b+4*sl3.arrayValue()[1]-200, //y3 =y2
            -x4b+300+4*sl3.arrayValue()[0], //x4 =x3
            y4b+4*sl3.arrayValue()[1]-200 //y4 =y1
            );



}

// ========= CALLBACKS =========

void leapOnInit() {
  // println("Leap Motion Init");
}
void leapOnConnect() {
  // println("Leap Motion Connect");
}
void leapOnFrame() {
  // println("Leap Motion Frame");
}
void leapOnDisconnect() {
  // println("Leap Motion Disconnect");
}
void leapOnExit() {
  // println("Leap Motion Exit");
}




//////////////////////////////////////////////////////
//
//
//////////////////////////////////////////////////////
ControlFrame addControlFrame(String theName, int theWidth, int theHeight) {
  Frame f = new Frame(theName);
  ControlFrame p = new ControlFrame(this, theWidth, theHeight);
  f.add(p);
  p.init();
  f.setTitle(theName);
  f.setSize(p.w, p.h);
  f.setLocation(50, 100);
  f.setResizable(false);
  f.setVisible(true);
  return p;
}


// the ControlFrame class extends PApplet, so we
// are creating a new processing applet inside a
// new frame with a controlP5 object loaded
public class ControlFrame extends PApplet
{
  int w, h;
  int abc = 255;
  int R1=0,R2=0;
  int L1=0,L2=0;
  //////////////////////////////////////////////////////
  //関数名：setup
  //概要：「3．設定」の初期設定を行う.
  //関連する関数
  //       ・Button_Advanced_Setting<<再生機能と軸の設定>>
  //////////////////////////////////////////////////////
  public void setup()
  {
    size(w, h);
    background(abc);
    frameRate(30);
    Button_Advanced_Setting();
  }
  //////////////////////////////////////////////////////
  //関数名：Button_Advanced_Setting
  //概要：再生機能と軸の設定を行う.
  //関連する関数：
  //         ・setup<<「3．設定」の初期設定>>
  //////////////////////////////////////////////////////
  public void Button_Advanced_Setting()
  {
    PFont p = createFont("Meiryo-48",12);
    //スライダーの設定
    cp5_Menubutton = new ControlP5(this);
    cp5_Menubutton.setControlFont(p,8);
```

```
 //速度調整（Right,Left）の設定
cp5sl1 = new ControlP5(this);//速度調整のボタン（右データ 1）の設定
sl1 = cp5sl1.addSlider("R1")
            .setPosition(10,135)
            .setSize(100,20)
            .setRange(-5,5)
            .setValue(0)
            .setNumberOfTickMarks(11)
            .setSliderMode(Slider.FLEXIBLE)
            ;
cp5sl2 = new ControlP5(this);//速度調整のボタン（右データ 1）の設定
sl2 = cp5sl2.addSlider("L1")
            .setPosition(10,165)
            .setSize(100,20)
            .setRange(-5,5)
            .setValue(0)
            .setNumberOfTickMarks(11)
            .setSliderMode(Slider.FLEXIBLE)
            ;
cp5sl7 = new ControlP5(this);//速度調整のボタン（右データ 2）の設定
sl7 = cp5sl7.addSlider("R2")
            .setPosition(10,195)
            .setSize(100,20)
            .setRange(-5,5)
            .setValue(0)
            .setNumberOfTickMarks(11)
            .setSliderMode(Slider.FLEXIBLE)
            ;
cp5sl8 = new ControlP5(this);//速度調整のボタン（左データ 2）の設定
sl8 = cp5sl8.addSlider("L2")
            .setPosition(10,225)
            .setSize(100,20)
            .setRange(-5,5)
            .setValue(0)
            .setNumberOfTickMarks(11)
            .setSliderMode(Slider.FLEXIBLE)
            ;



 //軸回転の設定
 //X 軸回転の設定
cp5sl4 = new ControlP5(this);//X 軸回転の設定
sl4 = cp5sl4.addSlider("xR")
            .setPosition(45,10)
            .setSize(160,20)
            .setRange(-360,360)
            .setNumberOfTickMarks(721)
            .setValue(0)
            ;
 //Y 軸回転の設定
cp5sl5 = new ControlP5(this);//X 軸回転の設定
 sl5 = cp5sl5.addSlider("yR")
            .setPosition(45,40)
            .setSize(160,20)
            .setRange(-360,360)
            .setNumberOfTickMarks(721)
            .setValue(0)
            ;
 //Z 軸回転の設定
cp5sl6 = new ControlP5(this);//X 軸回転の設定
 sl6 = cp5sl6.addSlider("zR")
            .setPosition(45,70)
            .setSize(160,20)
            .setRange(-360,360)
            .setNumberOfTickMarks(721)
            .setValue(0)
            ;

//ボタンの設定
cp5button = new ControlP5(this);
cp5button.setControlFont(p,9);
//再生と停止のボタンの設定
cp5button.addButton("停止")//ボタン停止の生成
         .setPosition(220,230)//setPosition(theX,theY)
         .setSize(27,20);//setSize(width,height)
cp5button.addButton("再生")//ボタン再生の生成
         .setPosition(250,230)//setPosition(theX,theY)
         .setSize(27,20);//setSize(width,height)
cp5button.addButton("円描画")//ボタン円描画の生成
         .setPosition(290,230)//setPosition(theX,theY)
         .setSize(35,20);//setSize(width,height)
cp5button.addButton("たて軸変化")//ボタン円描画の生成
         .setPosition(330,230)//setPosition(theX,theY)
         .setSize(55,20);//setSize(width,height)

//軸のボタンの設定
cp5button.addButton("X") //ボタン X の生成
         .setPosition(30,460)
         .setSize(15,15);
cp5button.addButton("Y") //ボタン Y の生成
```

```
            .setPosition(63,378)
            .setSize(15,15);
    cp5button.addButton("Z") //ボタン Z の生成
            .setPosition(33,433)
            .setSize(15,15);
    //視点回転をやり直すボタンの設定
    cp5button.addButton("X 軸") //ボタン X 軸の生成
            .setPosition(10,10)
            .setSize(25,20);
    cp5button.addButton("Y 軸") //ボタン Y 軸の生成
            .setPosition(10,40)
            .setSize(25,20);
    cp5button.addButton("Z 軸") //ボタン Z 軸の生成
            .setPosition(10,70)
            .setSize(25,20);
    //視点変化のボタンの宣言
    cp5button.addButton("上") //前ボタンの生成
            .setPosition(10,100)
            .setSize(20,20);
    cp5button.addButton("下") //後ボタンの生成
            .setPosition(35,100)
            .setSize(20,20);
    cp5button.addButton("右") //右ボタンの生成
            .setPosition(60,100)
            .setSize(20,20);
    cp5button.addButton("左") //左ボタンの生成
            .setPosition(85,100)
            .setSize(20,20);
    cp5button.addButton("前") //上ボタンの生成
            .setPosition(110,100)
            .setSize(20,20);
    cp5button.addButton("後") //下ボタンの生成
            .setPosition(135,100)
            .setSize(20,20);

    cp5button.addButton("0 に戻す") //リセットボタンの生成
            .setPosition(235,70)
            .setSize(40,15);

/*  //データの変換ボタンの設定
    cp5button.addButton("0 に戻す") //下から計測に合わせる
            .setPosition(235,70)
            .setSize(40,15);
    cp5button.addButton("0 に戻す") //上から計測に合わせる
            .setPosition(235,70)
            .setSize(40,15);
  */

    //ズームの設定
    cp5zoom = new ControlP5(this);
    zoomValue = cp5zoom.addNumberbox("numberbox")
           .setPosition(285,40)
           .setSize(40,15)
           .setMultiplier(10.10) // set the sensitifity of the numberbox
           .setDirection(Controller.HORIZONTAL) // change the control direction to left/right
           .setValue(0);

    MaxAxis = cp5zoom.addNumberbox("numberboxS")
           .setPosition(230,265)
           .setSize(50,18)
           .setScrollSensitivity(1.1)
           .setValue(0);
    MinAxis = cp5zoom.addNumberbox("numberboxT")
           .setPosition(230,327)
           .setSize(50,18)
           .setScrollSensitivity(1.1)
           .setValue(0);

    //2D スライダーの設定
    //アニメーションの移動
    cp5sl3 = new ControlP5(this);
    cp5sl3.setControlFont(p,12);
    sl3 = cp5sl3.addSlider2D("平行移動")
                .setPosition(250,100)
                .setSize(120,100)
                .setArrayValue(new float[] {memoriX, memoriY})
         //.disableCrosshair()
         ;
  cp5r = new ControlP5(this);
  rangeXaxis = cp5r.addRange("rangeController")
             // disable broadcasting since setRange and setRangeValues will trigger an event
             .setBroadcast(false)
             .setPosition(220,355)
             .setSize(167,10)
             .setHandleSize(10)
         //    .setNumberOfTickMarks(5)
             .setRange(0,30)
             .setRangeValues(0,30)
             // after the initialization we turn broadcast back on again
             .setBroadcast(true)
             .setColorForeground(color(255,40))
```

```
            .setColorBackground(color(255,40))
            ;


       }
    ////////////////////////////////////////////////
    //関数名：draw
    //概要：
    //関連する関数
    //         ・Progress<<アニメーションの再生状況>>
    //         ・Right,Left<<速度調整のボタンの設定>>
    //関連する単語
    //         ・Rightspeed,LeftSpeed
    ////////////////////////////////////////////////

  public void draw()
  {
    if(SaveData ==0)
    {
      SelectPoint();
    }
    //時間経過を表す数直線上
    stroke(255,255,255); //輪郭が白色
    rect(6,485,50,10); //background の代わり，さいせい
    rect(95,505,40,10);//background の代わり，はやさ
    rect(312,485,88,26); //background の代わり，列数等
    rect(0,595,400,70); //時間経過で重要
    stroke(0);    //輪郭が黒色
    fill(255,255,255); //白色に塗る
    rect(5,5,215,120); //軸回転付近の四角形
    rect(230,5,160,85); //ズーム付近の四角形（枠）
    rect(215,225,65,30); //再生と停止のボタン付近の四角形
    rect(285,225,105,30); //グラフのボタン付近の四角形
    fill(0); //黒色に塗る
    rect(215,260,175,110); //軌跡グラフの軸設定付近の四角形
    rect(5,130,170,125); //速度調整用ボタン付近の四角形
    rect(230,95,160,125); //軸調整

    ellipse(260,45,20,20); //ズームの小さい四角形
    ellipse(360,45,40,40); //ズームの大きい四角形
    line(280,25,330,25); //ズーム付近の→（まっすぐ）
    line(330,25,320,20); //ズーム付近の→（上）
    line(330,25,320,30); //ズーム付近の→（下）
    line(280,70,330,70); //ズーム付近の←（まっすぐ）
    line(280,70,290,65); //ズーム付近の←（上）
    line(280,70,290,75); //ズーム付近の←（下）

    fill(255,255,255);
     stroke(255,255,255);    //輪郭が白色

    rect(285,265,100,80); //軌跡グラフ
    rect(250,265,30,18); //縦軸の数値 Max
    rect(250,327,30,18); //縦軸の数値 Min

    stroke(0);    //輪郭が黒色
    line(285,305,385,305); //軌跡グラフの縦軸
    line(285,285,385,285); //軌跡グラフの縦軸
    line(285,325,385,325); //軌跡グラフの縦軸
    line(335,265,335,345); //軌跡グラフの横軸
    line(310,265,310,345); //軌跡グラフの横軸
    line(360,265,360,345); //軌跡グラフの横軸


    fill(0);
    PFont p = createFont("Meiryo-48",15); //文字フォントの設定
    textFont(p);
    text("回転移動",158,115);
    text("拡大",280,20);
    text("縮小",300,85);

    p = createFont("Meiryo-48",9); //文字フォントの設定
    textFont(p);
//    text("(mm)",322,273);
//    text("(mm)",322,353);
//    text("(秒)",370,345);
    fill(255,255,255);
    text("（右データ 1）",120,150);
    text("（左データ 1）",120,180);
    text("（右データ 2）",120,210);
    text("（左データ 2）",120,240);


    Rightspeed = R1;   //速さ調節（右手）
    Leftspeed = L1;    //速さ調節（左手）
    Rightspeed2 = R2;   //速さ調節（右手）
    Leftspeed2 = L2;    //速さ調節（左手）

    if(selectButton == 1) //アニメーション
    {
    xRotate = PI*1/2 + TWO_PI*sl4.getValue()/360; //少し誤差あり，分母 57
```

```
  yRotate = PI     + TWO_PI*sl5.getValue()/360;
  zRotate = PI*1/2 + TWO_PI*sl6.getValue()/360;
  }
  if(selectButton == 2) //変位グラフ
  {
  xRotate =  PI/2 + TWO_PI*sl4.getValue()/360; //少し誤差あり，分母 57
  yRotate = -PI/2 + TWO_PI*sl5.getValue()/360;
  zRotate =    0 + TWO_PI*sl6.getValue()/360;
  }
  if(selectButton == 3) //3D グラフ
  {
  xRotate =  PI*1/2 + TWO_PI*sl4.getValue()/360; //少し誤差あり，分母 57
  yRotate =  PI     + TWO_PI*sl5.getValue()/360;
  zRotate =  PI*1/2  + TWO_PI*sl6.getValue()/360;
  }
  if(selectButton == 4) //アニメーションと 3D グラフ
  {
  xRotate =  PI*1/2 + TWO_PI*sl4.getValue()/360; //少し誤差あり，分母 57
  yRotate =  PI     + TWO_PI*sl5.getValue()/360;
  zRotate =  PI*1/2 + TWO_PI*sl6.getValue()/360;
  }
  rangeXaxis.setMax(timechange); //X 軸の値変化

}
////////////////////////////////////////////////////////
//関数名：numberbox
//概要：ズーム機能の調整をする.
//関連する関数
//         ・Button_Advanced_Setting<<>>
////////////////////////////////////////////////////////
public void numberbox(int zoom)
{
  camZ = -8000 -zoom;
}
////////////////////////////////////////////////////////
//関数名：numberboxS
//概要：変位グラフの縦軸（最大値）の調整をする.
////////////////////////////////////////////////////////
public void numberboxS(int zoom)
{
  numberboxS = zoom -AxisP;

}
////////////////////////////////////////////////////////
//関数名：numberboxT
//概要：変位グラフの縦軸（最小値）の調整をする.
////////////////////////////////////////////////////////
public void numberboxT(int zoom)
{

  numberboxT = zoom -AxisM;
}

////////////////////////////////////////////////////////
//関数名：SelectPoint
//概要：手の画像に直線と円を描く.
//関連する関数
//         ・controlEvent<<「両手再生」で実行>>
//困っていること
//         ・SelectPoint を controlEvent で実行したかったができなかった.
////////////////////////////////////////////////////////
public void SelectPoint()
{
  image(img2,0,375,img2.width*15/60,img2.height*15/60);
  image(img1,185,375,img1.width*15/60,img1.height*15/60);
  //右手
  line(335,475,300,427); //親指
  line(337,430,328,385); //人差し指
  line(353,430,353,380); //中指
  line(363,434,372,390); //薬指
  line(370,440,386,411); //小指
  line(370,465,370,440);//小指
  line(335,475,345,470); //親指→人差し指
  line(320,455,337,430); //親指→人差し指
  line(335,475,337,430); //親指→人差し指
  line(353,430,337,430); //人差し指→中指
  line(345,470,353,465); //人差し指→中指
  line(353,465,362,465); //中指→薬指
  line(353,430,363,434); //中指→薬指
  line(362,465,370,465); //薬指→小指
  line(363,434,370,440); //薬指→小指
  //左手
  line(245,475,283,427); //親指
  line(245,430,258,385); //人差し指
  line(230,430,230,380); //中指
  line(218,435,205,385); //薬指
  line(208,445,190,405); //小指
  line(206,465,208,445); //小指
  line(215,470,206,465); //薬指
  line(218,435,208,445); //薬指
  line(245,475,234,473); //親指→人差し指
```

```
line(245,475,245,430); //親指→人差し指
line(262,455,245,430); //親指→人差し指
line(232,473,223,473); //人差し指→中指
line(245,430,230,430); //人差し指→中指
line(223,473,215,470); //中指→薬指
line(230,430,218,435); //中指→薬指
//右手
ellipse(335,475,7,7); //親指 3
ellipse(320,455,7,7); //親指 2
ellipse(310,440,7,7); //親指 1
ellipse(300,427,7,7); //親指先端
ellipse(345,470,7,7); //人差し指 4
ellipse(337,430,7,7); //人差し指 3
ellipse(333,410,7,7); //人差し指 2
ellipse(330,395,7,7); //人差し指 1
ellipse(328,385,7,7); //人差し指先端
ellipse(353,465,7,7); //中指 4
ellipse(353,430,7,7); //中指 3
ellipse(353,405,7,7); //中指 2
ellipse(353,390,7,7); //中指 1
ellipse(353,380,7,7); //中指先端
ellipse(362,465,7,7);//薬指 4
ellipse(363,434,7,7);//薬指 3
ellipse(368,410,7,7);//薬指 2
ellipse(370,400,7,7); //薬指 1
ellipse(372,390,7,7);//薬指先端
ellipse(370,465,7,7);//小指 4
ellipse(370,440,7,7);//小指 3
ellipse(378,428,7,7);//小指 2
ellipse(383,420,7,7); //小指 1
ellipse(386,411,7,7);//小指先端
ellipse(353,450,7,7); //重心
//左手
ellipse(245,475,7,7); //親指 3
ellipse(262,455,7,7); //親指 2
ellipse(274,440,7,7); //親指 1
ellipse(283,427,7,7); //親指先端
ellipse(234,473,7,7); //人差し指 4
ellipse(245,430,7,7); //人差し指 3
ellipse(253,405,7,7); //人差し指 2
ellipse(255,395,7,7); //人差し指 1
ellipse(258,385,7,7); //人差し指先端
ellipse(223,473,7,7); //中指 4
ellipse(230,430,7,7); //中指 3
ellipse(230,405,7,7); //中指 2
ellipse(230,392,7,7); //中指 1
ellipse(230,380,7,7); //中指先端
ellipse(215,470,7,7); //薬指 4
ellipse(218,435,7,7); //薬指 3
ellipse(212,410,7,7); //薬指 2
ellipse(209,397,7,7); //薬指 1
ellipse(205,385,7,7); //薬指先端
ellipse(206,465,7,7); //小指 4
ellipse(208,445,7,7); //小指 3
ellipse(200,425,7,7); //小指 2
ellipse(195,415,7,7); //小指 1
ellipse(190,405,7,7); //小指先端
ellipse(230,455,7,7); //重心

rect(180,131,30,120); //データ選択の円を囲う四角形
rect(185,136,20,20); //右データ 1 の円付近の四角形
rect(185,166,20,20); //左データ 1 の円付近の四角形
rect(185,196,20,20); //右データ 2 の円付近の四角形
rect(185,226,20,20); //左データ 2 の円付近の四角形
//円の色判定
//右データ 1 の円
if(fitdata1 ==0)
{
  fill(255,255,255);
}
if(fitdata1 ==1)
{
  fill(255,0,0);
}
ellipse(195,146,10,10);
//左データ 1 の円
if(fitdata2 ==0)
{
  fill(255,255,255);
}
if(fitdata2 ==1)
{
  fill(255,0,0);
}
ellipse(195,176,10,10);
//右データ 2 の円
if(fitdata3 ==0)
{
  fill(255,255,255);
}
if(fitdata3 ==1)
```

```
  {
    fill(255,0,0);
  }
  ellipse(195,206,10,10);
  //左データ 2 の円
  if(fitdata4 ==0)
  {
    fill(255,255,255);
  }
  if(fitdata4 ==1)
  {
    fill(255,0,0);
  }
  ellipse(195,236,10,10);

  fill(255,255,255);

}
////////////////////////////////////////////
//関数名：Progress
//概要：両手のアニメーションにおける再生の状況を
//      表示する。
//関連する関数
//  ・For_Finite_Animation<<ループの禁止>>
////////////////////////////////////////////
public void Progresss()
{
  For_Finite_Animationn(); //ループの禁止
  stroke(0);
  line(20,610,20,615);
  line(20,610,360,610);
  ellipse(20 +(340*i/(LENGTHF-2)),610,6,6);
  text("S",5,614);
  text("F",365,614);
  if(fitdata1 ==1)
  {
    text(nf(csvF2[i][0]/1000000,1,2),20+(340*i/(LENGTHF-2)),605);
    text(nf(csvF2[LENGTHF-3][0]/1000000,1,2),350,624);
    line(20,650,20,655);
    line(20,650,360,650);
    ellipse(20 +(340*j/(LENGTHS-2)),650,6,6);
  }
  text("S",5,654);
  text("F",365,654);
  if(fitdata2 ==1){
    text(nf(csvS2[j][0]/1000000,1,2),20+(340*j/(LENGTHS-2)),645);
    text(nf(csvS2[LENGTHS-3][0]/1000000,1,2),350,664);
  }
}
////////////////////////////////////////////
//関数名：For_Finite_Animation
//概要：両手のアニメーションがループしない条件分
//      岐をする。
////////////////////////////////////////////
public void For_Finite_Animationn()
{
  if(i+1 >= LENGTHF)
  {
    i=2;
    Rightspeed=0;
    sl1.setValue(0);
  }
   if(i+1 <= 0)
  {
    i = 2;
    Rightspeed=0;
    sl1.setValue(0);
  }
   if(j+1 >= LENGTHS)
  {
    j=2;
    Leftspeed=0;
    sl2.setValue(0);
  }
   if(j+1 <= 0)
  {
    j = 2;
    Leftspeed=0;
    sl2.setValue(0);
  }
   if(i2+1 >= LENGTHT)
  {
    i2=2;
    Rightspeed2=0;
    sl7.setValue(0);
  }
   if(i2+1 <= 0)
  {
    i2 = 2;
    Rightspeed2=0;
    sl7.setValue(0);
```

```
  }
   if(j2+1 >= LENGTHH)
  {
    j2=2;
    Leftspeed2=0;
    sl8.setValue(0);
  }
   if(j2+1 <= 0)
  {
    j2 = 2;
    Leftspeed2=0;
    sl8.setValue(0);
  }
}
////////////////////////////////////////////////
//関数名：controlEvent
//概要：再生機能や軸の変化を管理する.
////////////////////////////////////////////////
public void controlEvent(ControlEvent theEvent)
{
  if(theEvent.getController().getName()=="再生")
  {
      Rightspeed =1;
      Leftspeed =1;
      if(fitdata1 ==1)
      {
      sl1.setValue(2);
      }
      if(fitdata2 ==1)
      {
    sl2.setValue(2);
      }
      if(fitdata3 ==1)
      {
        sl7.setValue(2);
      }
      if(fitdata4 ==1)
      {
      sl8.setValue(2);
      }


    SelectPoint();
  }
  if(theEvent.getController().getName()=="停止")
  {
    Rightspeed =0;
    Leftspeed =0;
    sl1.setValue(0);
    sl2.setValue(0);
    sl7.setValue(0);
    sl8.setValue(0);
  }
   if(theEvent.getController().getName()=="円描画")
   {
     DwEl= DwEl +1;
   }
   if(theEvent.getController().getName()=="たて軸変化")
   {
     DwAx= DwAx +1;
     VerAxisIs =1; //変位グラフの縦軸を調整する変数
   }
   if(theEvent.getController().getName()=="X")
  {
    e=0; //csv[x+y+e] で関節を表現
    VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
  if(theEvent.getController().getName()=="Y")
  {
    e=1; //csv[x+y+e] で関節を表現
    VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
  if(theEvent.getController().getName()=="Z")
  {
    e=2; //csv[x+y+e] で関節を表現
    VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
  if(theEvent.getController().getName()=="xR")
  {
    xRotate =  xRotate +xR/360;
  }
  if(theEvent.getController().getName()=="yR")
  {
    yRotate =  yRotate +yR/360;
  }
  if(theEvent.getController().getName()=="zR")
  {
    zRotate =  zRotate +zR/360;
```

```
  }

  if(theEvent.getController().getName()=="X 軸")
  {
    sl4.setValue(0);
  }
  if(theEvent.getController().getName()=="Y 軸")
  {
    sl5.setValue(0);
  }
  if(theEvent.getController().getName()=="Z 軸")
  {
    sl6.setValue(0);
  }
  if(theEvent.getController().getName()=="上")
  {
    sl4.setValue(0);
    sl5.setValue(0);
    sl6.setValue(0);
  }
  if(theEvent.getController().getName()=="下")
  {
    sl4.setValue(180);
    sl5.setValue(180);
    sl6.setValue(0);
  }
  if(theEvent.getController().getName()=="右")
  {
    sl4.setValue(90);
    sl5.setValue(-90);
    sl6.setValue(0);
  }
  if(theEvent.getController().getName()=="左")
  {
    sl4.setValue(90);
    sl5.setValue(90);
    sl6.setValue(0);
  }
  if(theEvent.getController().getName()=="前")
  {
    sl4.setValue(90);
    sl5.setValue(180);
    sl6.setValue(0);
  }
  if(theEvent.getController().getName()=="後")
  {
    sl4.setValue(90);
    sl5.setValue(0);
    sl6.setValue(0);
  }
  if(theEvent.getController().getName()=="0 に戻す")
  {
   camZ = -8000;
   numberbox=0;
   zoomValue.setValue(0);  //表示される値を変えたかったができなかった.
  }
}
 int kx=-901,ky=-900;
 int timeforRigDa1 =0,timeforLefDa1=0,timeforRigDa2 =0,timeforLefDa2=0;
////////////////////////////////////////////////
//関数名：mousePressed
//概要：ボタンを押したとき，変化する.
////////////////////////////////////////////////
public void mousePressed()
{
  /////右手/////////////////////////////////////////////////////////////////////////////////
  //親指////
   if(mouseX>=331 && mouseX<=339 && mouseY>=471 && mouseY<=479)
  {
    //前回の円削除
   fill(255,255,255);
   ellipse(kx,ky,7,7);
   x=10;
   y=-9;
   d=0;
   kx=335;
   ky=475;
   //円描画
   fill(255,0,0);
   stroke(0);
   ellipse(kx,ky,7,7);
   VerAxisIs =1; //変位グラフの縦軸を調整する変数

 }
  if(mouseX>=316 && mouseX<=324 && mouseY>=451 && mouseY<=459)
 {
   //前回の円削除
   fill(255,255,255);
   ellipse(kx,ky,7,7);
   x=10;
   y=-6;
```

```
    d=0;
    kx=320;
    ky=455;
    //円描画
    fill(255,0,0);
    stroke(0);
    ellipse(kx,ky,7,7);
    VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
   if(mouseX>=306 && mouseX<=314 && mouseY>=436 && mouseY<=444)
  {
    //前回の円削除
    fill(255,255,255);
    ellipse(kx,ky,7,7);
    x=10;
    y=-3;
    d=0;
    kx=310;
    ky=440;
    //円描画
    fill(255,0,0);
    stroke(0);
    ellipse(kx,ky,7,7);
    VerAxisIs =1; //変位グラフの縦軸を調整する変数


  }
   if(mouseX>=296 && mouseX<=304 && mouseY>=423 && mouseY<=431)
  {
    //前回の円削除
    fill(255,255,255);
    ellipse(kx,ky,7,7);
    x=10;
    y=0;
    d=0;
    kx=300;
    ky=427;
    //円描画
    fill(255,0,0);
    stroke(0);
    ellipse(kx,ky,7,7);
    VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
  //人差し指
   if(mouseX>=341 && mouseX<=349 && mouseY>=466 && mouseY<=476)
  {
    //前回の円削除
    fill(255,255,255);
    ellipse(kx,ky,7,7);

         x=25;  y=-12;  d=0;
   kx=345;  ky=470;

   //円描画
 fill(255,0,0);
      stroke(0);
 ellipse(kx,ky,7,7);
     VerAxisIs =1; //変位グラフの縦軸を調整する変数

   }
  if(mouseX>=333 && mouseX<=341 && mouseY>=426 && mouseY<=436){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

         x=25;  y=-9;  d=0;
   kx=337;  ky=430;

   //円描画
 fill(255,0,0);
      stroke(0);
 ellipse(kx,ky,7,7);
     VerAxisIs =1; //変位グラフの縦軸を調整する変数

   }
 if(mouseX>=329 && mouseX<=337 && mouseY>=406 && mouseY<=414){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

         x=25;  y=-6; d=0;
   kx=333;  ky=410;

   //円描画
 fill(255,0,0);
      stroke(0);
 ellipse(kx,ky,7,7);
     VerAxisIs =1; //変位グラフの縦軸を調整する変数
```

```
        }
    if(mouseX>=326 && mouseX<=334 && mouseY>=391 && mouseY<=399){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

         x=25;  y=-3; d=0;
   kx=330;  ky=395;

   //円描画
 fill(255,0,0);
      stroke(0);
 ellipse(kx,ky,7,7);
     VerAxisIs =1; //変位グラフの縦軸を調整する変数

   }
if(mouseX>=324 && mouseX<=332 && mouseY>=381 && mouseY<=389){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

         x=25;  y=0; d=0;
   kx=328;  ky=385;

   //円描画
 fill(255,0,0);
      stroke(0);
 ellipse(kx,ky,7,7);
     VerAxisIs =1; //変位グラフの縦軸を調整する変数

   }
   //中指
if(mouseX>=349 && mouseX<=357 && mouseY>=461 && mouseY<=469){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

         x=40;  y=-12; d=0;
   kx=353;  ky=465;

   //円描画
 fill(255,0,0);
      stroke(0);
 ellipse(kx,ky,7,7);
     VerAxisIs =1; //変位グラフの縦軸を調整する変数

   }
if(mouseX>=349 && mouseX<=357 && mouseY>=426 && mouseY<=434){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

         x=40;  y=-9; d=0;
   kx=353;  ky=430;

   //円描画
 fill(255,0,0);
      stroke(0);
 ellipse(kx,ky,7,7);
     VerAxisIs =1; //変位グラフの縦軸を調整する変数

   }
if(mouseX>=349 && mouseX<=357 && mouseY>=401 && mouseY<=409){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

         x=40;  y=-6; d=0;
   kx=353;  ky=405;

   //円描画
 fill(255,0,0);
      stroke(0);
 ellipse(kx,ky,7,7);
     VerAxisIs =1; //変位グラフの縦軸を調整する変数

   }
    if(mouseX>=349 && mouseX<=357 && mouseY>=386 && mouseY<=394){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

         x=40;  y=-3; d=0;
   kx=353;  ky=390;

   //円描画
 fill(255,0,0);
      stroke(0);
 ellipse(kx,ky,7,7);
     VerAxisIs =1; //変位グラフの縦軸を調整する変数
```

```
    }
  if(mouseX>=349 && mouseX<=357 && mouseY>=376 && mouseY<=384){
 //前回の円削除
  fill(255,255,255);
  ellipse(kx,ky,7,7);

          x=40;  y=0; d=0;
    kx=353;  ky=380;

    //円描画
  fill(255,0,0);
       stroke(0);
  ellipse(kx,ky,7,7);
      VerAxisIs =1; //変位グラフの縦軸を調整する変数

    }
//薬指
  if(mouseX>=358 && mouseX<=366 && mouseY>=461 && mouseY<=469){
 //前回の円削除
  fill(255,255,255);
  ellipse(kx,ky,7,7);

          x=55;  y=-12; d=0;
    kx=362;  ky=465;

    //円描画
  fill(255,0,0);
       stroke(0);
  ellipse(kx,ky,7,7);
      VerAxisIs =1; //変位グラフの縦軸を調整する変数

    }
  if(mouseX>=359 && mouseX<=367 && mouseY>=430 && mouseY<=438){
 //前回の円削除
  fill(255,255,255);
  ellipse(kx,ky,7,7);

          x=55;  y=-9; d=0;
    kx=363;  ky=434;

    //円描画
  fill(255,0,0);
       stroke(0);
  ellipse(kx,ky,7,7);
      VerAxisIs =1; //変位グラフの縦軸を調整する変数

    }
  if(mouseX>=364 && mouseX<=372 && mouseY>=406 && mouseY<=414){
 //前回の円削除
  fill(255,255,255);
  ellipse(kx,ky,7,7);

          x=55;  y=-6; d=0;
    kx=368;  ky=410;

    //円描画
  fill(255,0,0);
       stroke(0);
  ellipse(kx,ky,7,7);
      VerAxisIs =1; //変位グラフの縦軸を調整する変数

    }
  if(mouseX>=366 && mouseX<=374 && mouseY>=396 && mouseY<=404){
 //前回の円削除
  fill(255,255,255);
  ellipse(kx,ky,7,7);

          x=55;  y=-3; d=0;
    kx=370;  ky=400;

    //円描画
  fill(255,0,0);
       stroke(0);
  ellipse(kx,ky,7,7);
      VerAxisIs =1; //変位グラフの縦軸を調整する変数

    }
  if(mouseX>=368 && mouseX<=376 && mouseY>=386 && mouseY<=394){
 //前回の円削除
  fill(255,255,255);
  ellipse(kx,ky,7,7);

          x=55;  y=0; d=0;
    kx=372;  ky=390;

    //円描画
  fill(255,0,0);
       stroke(0);
  ellipse(kx,ky,7,7);
      VerAxisIs =1; //変位グラフの縦軸を調整する変数
```

```
    }
//小指
  if(mouseX>=366 && mouseX<=374 && mouseY>=461 && mouseY<=469){
 //前回の円削除
  fill(255,255,255);
  ellipse(kx,ky,7,7);

          x=70;  y=-12; d=0;
    kx=370;  ky=465;

    //円描画
  fill(255,0,0);
       stroke(0);
  ellipse(kx,ky,7,7);
      VerAxisIs =1; //変位グラフの縦軸を調整する変数

    }
  if(mouseX>=366 && mouseX<=374 && mouseY>=436 && mouseY<=444){
 //前回の円削除
  fill(255,255,255);
  ellipse(kx,ky,7,7);

          x=70;  y=-9; d=0;
    kx=370;  ky=440;

    //円描画
  fill(255,0,0);
       stroke(0);
  ellipse(kx,ky,7,7);
      VerAxisIs =1; //変位グラフの縦軸を調整する変数

    }
  if(mouseX>=374 && mouseX<=382 && mouseY>=424 && mouseY<=432){
 //前回の円削除
  fill(255,255,255);
  ellipse(kx,ky,7,7);

          x=70;  y=-6; d=0;
    kx=378;  ky=428;

    //円描画
  fill(255,0,0);
       stroke(0);
  ellipse(kx,ky,7,7);
      VerAxisIs =1; //変位グラフの縦軸を調整する変数

    }
  if(mouseX>=379 && mouseX<=387 && mouseY>=416 && mouseY<=424){
 //前回の円削除
  fill(255,255,255);
  ellipse(kx,ky,7,7);

          x=70;  y=-3; d=0;
    kx=383;  ky=420;

    //円描画
  fill(255,0,0);
       stroke(0);
  ellipse(kx,ky,7,7);
      VerAxisIs =1; //変位グラフの縦軸を調整する変数

    }
  if(mouseX>=382 && mouseX<=390 && mouseY>=407 && mouseY<=415){
 //前回の円削除
  fill(255,255,255);
  ellipse(kx,ky,7,7);

          x=70;  y=0; d=0;
    kx=386;  ky=411;

    //円描画
  fill(255,0,0);
       stroke(0);
  ellipse(kx,ky,7,7);
      VerAxisIs =1; //変位グラフの縦軸を調整する変数

    }
    //重心
  if(mouseX>=349 && mouseX<=357 && mouseY>=446 && mouseY<=454){
 //前回の円削除
  fill(255,255,255);
  ellipse(kx,ky,7,7);

          x=73;  y=0; d=0;
    kx=353;  ky=450;

    //円描画
  fill(255,0,0);
       stroke(0);
  ellipse(kx,ky,7,7);
      VerAxisIs =1; //変位グラフの縦軸を調整する変数
```

```
  }

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

//左手
//親指
 if(mouseX>=241 && mouseX<=249 && mouseY>=471 && mouseY<=479){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

        x=10;  y=-9;  d=1;
   kx=245;  ky=475;

   //円描画
 fill(255,0,0);
      stroke(0);
 ellipse(kx,ky,7,7);
    VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
 if(mouseX>=258 && mouseX<=266 && mouseY>=451 && mouseY<=459){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

        x=10;  y=-6;  d=1;
   kx=262;  ky=455;

   //円描画
 fill(255,0,0);
     stroke(0);
 ellipse(kx,ky,7,7);
 VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
 if(mouseX>=270 && mouseX<=278 && mouseY>=436 && mouseY<=444){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

        x=10;  y=-3;  d=1;
   kx=274;  ky=440;

   //円描画
 fill(255,0,0);
     stroke(0);
 ellipse(kx,ky,7,7);
    VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
 if(mouseX>=279 && mouseX<=287 && mouseY>=423 && mouseY<=431){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

        x=10;  y=0;  d=1;
   kx=283;  ky=427;

   //円描画
 fill(255,0,0);
     stroke(0);
 ellipse(kx,ky,7,7);
    VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
//人差し指
 if(mouseX>=230 && mouseX<=238 && mouseY>=469 && mouseY<=477){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

        x=25;  y=-12;  d=1;
   kx=234;  ky=473;

   //円描画
 fill(255,0,0);
     stroke(0);
 ellipse(kx,ky,7,7);
    VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
 if(mouseX>=241 && mouseX<=249 && mouseY>=426 && mouseY<=434){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

        x=25;  y=-9;  d=1;
   kx=245;  ky=430;
```

```
    //円描画
  fill(255,0,0);
       stroke(0);
  ellipse(kx,ky,7,7);
      VerAxisIs =1; //変位グラフの縦軸を調整する変数

    }
  if(mouseX>=249 && mouseX<=257 && mouseY>=401 && mouseY<=409){
 //前回の円削除
  fill(255,255,255);
  ellipse(kx,ky,7,7);

          x=25;  y=-6;  d=1;
    kx=253;  ky=405;

    //円描画
  fill(255,0,0);
       stroke(0);
  ellipse(kx,ky,7,7);
      VerAxisIs =1; //変位グラフの縦軸を調整する変数

    }
    if(mouseX>=251 && mouseX<=259 && mouseY>=391 && mouseY<=399){
 //前回の円削除
  fill(255,255,255);
  ellipse(kx,ky,7,7);

          x=25;  y=-3;  d=1;
    kx=255;  ky=395;

    //円描画
  fill(255,0,0);
       stroke(0);
  ellipse(kx,ky,7,7);
      VerAxisIs =1; //変位グラフの縦軸を調整する変数

    }
  if(mouseX>=254 && mouseX<=262 && mouseY>=381 && mouseY<=389){
 //前回の円削除
  fill(255,255,255);
  ellipse(kx,ky,7,7);

          x=25;  y=0;  d=1;
    kx=258;  ky=385;

    //円描画
  fill(255,0,0);
       stroke(0);
  ellipse(kx,ky,7,7);
      VerAxisIs =1; //変位グラフの縦軸を調整する変数

    }
//中指
  if(mouseX>=219 && mouseX<=227 && mouseY>=469 && mouseY<=477){
 //前回の円削除
  fill(255,255,255);
  ellipse(kx,ky,7,7);

          x=40;  y=-12;  d=1;
    kx=223;  ky=473;

    //円描画
  fill(255,0,0);
       stroke(0);
  ellipse(kx,ky,7,7);
      VerAxisIs =1; //変位グラフの縦軸を調整する変数

    }
  if(mouseX>=226 && mouseX<=234 && mouseY>=426 && mouseY<=434){
 //前回の円削除
  fill(255,255,255);
  ellipse(kx,ky,7,7);

          x=40;  y=-9;  d=1;
    kx=230;  ky=430;

    //円描画
  fill(255,0,0);
       stroke(0);
  ellipse(kx,ky,7,7);
      VerAxisIs =1; //変位グラフの縦軸を調整する変数

    }
  if(mouseX>=226 && mouseX<=234 && mouseY>=401 && mouseY<=409){
 //前回の円削除
  fill(255,255,255);
  ellipse(kx,ky,7,7);

          x=40;  y=-6;  d=1;
    kx=230;  ky=405;
```

```
  //円描画
 fill(255,0,0);
     stroke(0);
 ellipse(kx,ky,7,7);
    VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
 if(mouseX>=226 && mouseX<=234 && mouseY>=388 && mouseY<=396){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

        x=40;  y=-3;  d=1;
  kx=230;  ky=392;

  //円描画
 fill(255,0,0);
     stroke(0);
 ellipse(kx,ky,7,7);
    VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
 if(mouseX>=226 && mouseX<=234 && mouseY>=376 && mouseY<=384){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

        x=40;  y=0;  d=1;
  kx=230;  ky=380;

  //円描画
 fill(255,0,0);
     stroke(0);
 ellipse(kx,ky,7,7);
    VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
  //薬指
 if(mouseX>=211 && mouseX<=219 && mouseY>=466 && mouseY<=474){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

        x=55;  y=-12;  d=1;
  kx=215;  ky=470;

  //円描画
 fill(255,0,0);
     stroke(0);
 ellipse(kx,ky,7,7);
    VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
 if(mouseX>=214 && mouseX<=222 && mouseY>=431 && mouseY<=439){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

        x=55;  y=-9;  d=1;
  kx=218;  ky=435;

  //円描画
 fill(255,0,0);
     stroke(0);
 ellipse(kx,ky,7,7);
    VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
 if(mouseX>=208 && mouseX<=216 && mouseY>=406 && mouseY<=414){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

        x=55;  y=-6;  d=1;
  kx=212;  ky=410;

  //円描画
 fill(255,0,0);
     stroke(0);
 ellipse(kx,ky,7,7);
    VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
  if(mouseX>=205 && mouseX<=213 && mouseY>=393 && mouseY<=401){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

        x=55;  y=-3;   d=1;
  kx=209;  ky=397;
```

```
  //円描画
 fill(255,0,0);
      stroke(0);
 ellipse(kx,ky,7,7);
     VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
 if(mouseX>=201 && mouseX<=209 && mouseY>=381 && mouseY<=389){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

        x=55;  y=0;  d=1;
  kx=205;  ky=385;

  //円描画
 fill(255,0,0);
      stroke(0);
 ellipse(kx,ky,7,7);
     VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
  //小指
 if(mouseX>=202 && mouseX<=210 && mouseY>=461 && mouseY<=469){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

        x=70;  y=-12;  d=1;
  kx=206;  ky=465;

  //円描画
 fill(255,0,0);
      stroke(0);
 ellipse(kx,ky,7,7);
     VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
 if(mouseX>=204 && mouseX<=212 && mouseY>=441 && mouseY<=449){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

        x=70;  y=-9;  d=1;
  kx=208;  ky=445;

  //円描画
 fill(255,0,0);
      stroke(0);
 ellipse(kx,ky,7,7);
     VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
 if(mouseX>=196 && mouseX<=204 && mouseY>=421 && mouseY<=429){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

        x=70;  y=-6;  d=1;
  kx=200;  ky=425;

  //円描画
 fill(255,0,0);
      stroke(0);
 ellipse(kx,ky,7,7);
     VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
  if(mouseX>=191 && mouseX<=199 && mouseY>=411 && mouseY<=419){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

        x=70;  y=-3;  d=1;
  kx=195;  ky=415;

  //円描画
 fill(255,0,0);
      stroke(0);
 ellipse(kx,ky,7,7);
     VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
 if(mouseX>=186 && mouseX<=194 && mouseY>=401 && mouseY<=409){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

        x=70;  y=0;  d=1;
  kx=190;  ky=405;
```

```
  //円描画
 fill(255,0,0);
     stroke(0);
 ellipse(kx,ky,7,7);
    VerAxisIs =1; //変位グラフの縦軸を調整する変数

  }
 if(mouseX>=226 && mouseX<=234 && mouseY>=451 && mouseY<=459){
//前回の円削除
 fill(255,255,255);
 ellipse(kx,ky,7,7);

        x=73;  y=0;  d=1;
  kx=230;  ky=455;

  //円描画
 fill(255,0,0);
     stroke(0);
 ellipse(kx,ky,7,7);

  }

 //  fill(255,255,255);

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

 if(mouseX>=190 && mouseX<=210 && mouseY>=136 && mouseY<=156)
 {

    timeforRigDa1 = timeforRigDa1 +1;
    if(timeforRigDa1 ==1)
    {
      //円描画
      fill(255,0,0);
      stroke(0);
      ellipse(195,146,10,10);
      fitdata1 =1;
    }
    if(timeforRigDa1 ==2)
   {
      fill(255,255,255);
      ellipse(195,146,10,10);
      timeforRigDa1=0;
      fitdata1 =0;
    }



  }
 if(mouseX>=190 && mouseX<=210 && mouseY>=166 && mouseY<=186)
 {

  timeforLefDa1 = timeforLefDa1 +1;
  if(timeforLefDa1 ==1)
  {
    //円描画
    fill(255,0,0);
    stroke(0);
    ellipse(195,176,10,10);
    fitdata2 =1;
  }
  if(timeforLefDa1 ==2)
  {
    fill(255,255,255);
    ellipse(195,176,10,10);
    timeforLefDa1=0;
    fitdata2 =0;
  }



  }
////////////////////////////////////////////////////////////////////////

 if(mouseX>=190 && mouseX<=210 && mouseY>=196 && mouseY<=216)
 {

  timeforRigDa2 = timeforRigDa2 +1;
  if(timeforRigDa2 ==1)
  {
    //円描画
    fill(255,0,0);
    stroke(0);
    ellipse(195,206,10,10);
    fitdata3 =1;
  }
  if(timeforRigDa2 ==2)
  {
    fill(255,255,255);
    ellipse(195,206,10,10);
```

```
      timeforRigDa2=0;
      fitdata3 =0;
    }



    }
  if(mouseX>=190 && mouseX<=210 && mouseY>=226 && mouseY<=246)
  {
    timeforLefDa2 = timeforLefDa2 +1;
    if(timeforLefDa2 ==1)
    {
      //円描画
      fill(255,0,0);
      stroke(0);
      ellipse(195,236,10,10);
      fitdata4 =1;
    }
    if(timeforLefDa2 ==2)
    {
      fill(255,255,255);
      ellipse(195,236,10,10);
      timeforLefDa2=0;
      fitdata4 =0;
    }



    }



      fill(255,255,255);

}







  private ControlFrame() {}

  public ControlFrame(Object theParent, int theWidth, int theHeight) {
    parent = theParent;
    w = theWidth;
    h = theHeight;
  }

  public ControlP5 control() {
    return cp5_Menubutton;
  }


  ControlP5 cp5_Menubutton;

  Object parent;


}
void mousePressed()
{
  if(mouseButton == RIGHT)
  {
    if(selectButton ==1)
    {
      ppm1.show();
    }
    if(selectButton ==3)
    {
      ppm2.show();
    }
    if(selectButton ==4)
    {
      ppm2.show();
    }

  }
  if(play ==2){
      if(fitdata1 ==1){
    //時間経過を軸で表す
    if(mouseX >= 20 && mouseX <= 20 +((560*csvF2[LENGTHF-2][0])/(1000000*timechange)) && mouseY >= 450 && mouseY <= 460)
    {
      //前回の円削除
      fill(255,255,255);
      rect(20,450,((560*csvF2[i][0])/(1000000*timechange)),1);
      LEF_Modi = (LENGTHF-2)*(1000000*timechange) / (560*csvF2[LENGTHF-2][0]);
      i_Modi = (mouseX-20)*LEF_Modi;
```

```
      i =int(i_Modi);
    }
      }
      if(fitdata2 ==1){
    //時間経過を軸で表す
    if(mouseX >= 20 && mouseX <= 20 +((560*csvS2[LENGTHS-2][0])/(1000000*timechange)) && mouseY >= 460 && mouseY <= 470)
    {
      //前回の円削除
      fill(255,255,255);
      rect(20,460,((560*csvS2[j][0])/(1000000*timechange)),1);
      LEF_Modi = (LENGTHS-2)*(1000000*timechange) / (560*csvS2[LENGTHS-2][0]);
      i_Modi = (mouseX-20)*LEF_Modi;
      j =int(i_Modi);
    }
      }
      if(fitdata3 ==1){
    //時間経過を軸で表す
    if(mouseX >= 20 && mouseX <= 20 +((560*csvT2[LENGTHT-2][0])/(1000000*timechange)) && mouseY >= 470 && mouseY <= 480)
    {
      //前回の円削除
      fill(255,255,255);
      rect(20,470,((560*csvT2[i2][0])/(1000000*timechange)),1);
      LEF_Modi = (LENGTHT-2)*(1000000*timechange) / (560*csvT2[LENGTHT-2][0]);
      i_Modi = (mouseX-20)*LEF_Modi;
      i2 =int(i_Modi);
    }
      }
      if(fitdata4 ==1){
    //時間経過を軸で表す
    if(mouseX >= 20 && mouseX <= 20 +((560*csvH2[LENGTHH-2][0])/(1000000*timechange)) && mouseY >= 480 && mouseY <= 490)
    {
      //前回の円削除
      fill(255,255,255);
      rect(20,480,((560*csvH2[j2][0])/(1000000*timechange)),1);
      LEF_Modi = (LENGTHH-2)*(1000000*timechange) / (560*csvH2[LENGTHH-2][0]);
      i_Modi = (mouseX-20)*LEF_Modi;
      j2 =int(i_Modi);
    }
      }
    }

}


void mouseDragged()
{
  if(selectButton ==3)
  {
    if(mv5 ==1)
    {
      if(sel5 ==1)
      {
      DeleteJointXY=0; DeleteJointXZ=0; DeleteJointYZ=0;
      }
      else{
        sel5 =0;
      }
    }
  }//ピカピカ
  if(selectButton ==1)
  {
    if(mv1 ==1)
    {
      if(sel1 == 1){
        addmX1=(mouseX/4);
        addmY1=(mouseY/4);
      }
      else{
        addmX1=0;
        addmY1=0;
        sel1 =0;
      }
    }
    if(mv2 ==1)
    {
      if(sel2 ==1){
        addmX2=(mouseX/4);
        addmY2=(mouseY/4);
      }
      else{
        addmX2=0;
        addmY2=0;
        sel2=0;
      }
    }
    if(mv3 ==1)
    {
            if(sel3 ==1){
    addmX3=(mouseX/4);
    addmY3=(mouseY/4);
            }
```

```
            else{
        addmX3=0;
        addmY3=0;
        sel3=0;
      }

    }
    if(mv4 ==1)
    {
                if(sel4 ==1){

    addmX4=(mouseX/4);
    addmY4=(mouseY/4);
                }
                else{
        addmX4=0;
        addmY4=0;
        sel4=0;
      }

    }
  }
}

void keyPressed()
{
   //X,Y,Z 軸回転
 //  else if(keyCode == BACKSPACE){xRotate =1.6; yRotate=-1.501; zRotate=0; Rightspeed =0; Leftspeed =0;  camZ = -8000; scZ = -1000;}
   //拡大・縮小
    if(key == 'c'){camZ = camZ +100;}
   else if(key == 'v'){camZ = camZ -100;}
   //前後左右
   else if(keyCode == UP){ScreenUp = ScreenUp -10;}
   else if(keyCode == DOWN){ScreenUp = ScreenUp +10;}
   else if(keyCode == RIGHT){ScreenRight = ScreenRight +10;}
   else if(keyCode == LEFT){ScreenRight = ScreenRight -10;}


 }
```

# 付録 B　押圧力測定装置のプログラム

## B.1.　計測保存プログラム

```
#include <SD.h>

#define DAT 8
#define CLK 9
#define PIN_SDCS 4 //SD のチップ・セレクト CS(SS) のポート番号

File file; //SD ファイルの定義
float offset=0; //ロードセルの初期値定義

void setup() {
  Serial.begin(9600); //通信速度を 9600bps に設定
  pinMode(CLK, OUTPUT); //CLK(9) を出力に設定
  pinMode(DAT, INPUT); //DAT(8) を入力に設定
  pinMode(2,INPUT); //2 番ピンを入力に設定
  pinMode(4,INPUT); //2 番ピンを入力に設定
  offset = Read(); //初期値の設定
  while(SD.begin(PIN_SDCS)==false){ //SD メモリカードの開始
    delay(1000); //失敗時は 1 秒ごとに繰り返し実施
    }
}

void loop() {
  if(digitalRead(2) ==LOW){ //スイッチが押されていれば
    float data; //荷重の定義
    int Change;
    unsigned long time=millis(); //起動からの時間 [ms] 定義
    data = Read(); //荷重の設定
    Change =(int)-data/5;

    file=SD.open("TEST.CSV",FILE_WRITE); //書きこみファイルのオープン
    if(file == true){ //オープンが成功した場合
if(digitalRead(4) ==LOW){ //スイッチが押されていれば PC と接続
  Serial.write('H'); //開始合図 (ヘッダ) を送信

    Serial.write(-Change/100); //data を送信
    Serial.write(-Change%100); //data を送信
   // Serial.println(Change/100);
  }

      file.print(time); //ファイルに書き込み
      file.print(",");
      file.println(-data,2);
      file.close();
         // Serial.println(data/100);

      }
     else{} //オープンが失敗した場合
      }
  else{ //スイッチが押されていなければ
    delay(100); //失敗時は 0.1 秒遅延
    }
}

float Read(void){
  float volt; //電圧値の定義
  float gram; //荷重の定義
  long data=0; //読みだした 24 ビット用データ定義
  while(digitalRead(DAT)!=0); //DAT が 0 でない時繰り返す
  for(char i=0;i<24;i++) {
    digitalWrite(CLK,1);
    delayMicroseconds(1);
```

```
    digitalWrite(CLK,0);
    delayMicroseconds(1);
    data = (data<<1)|(digitalRead(DAT));
    }
    digitalWrite(CLK,1); //gain=128, PGA の増幅度
    delayMicroseconds(1);
    digitalWrite(CLK,0);
    delayMicroseconds(1);
    data = data^0x800000;
    volt =data*(4.2987/16777216.0/128);
    gram=volt/(0.001*4.2987/20000.0);
    return gram-offset; //荷重から初期値引き戻す
}
```

## B.2.　押圧力表示プログラム

```
import processing.serial.*;

Serial port;
float[] data;

void setup()
{
  size(600,500);
  port =new Serial(this,"COM3",9600); //通信ポートと速度の設定
  background(255); //背景を灰色
  strokeWeight(2); //線の太さを 2pt
  fill(0,0,255); //文字の色を青
  textSize(24); //文字の大きさを 24pt
  port.clear(); //受信データをクリア
  port.write('H'); //Arduino に返信要求を送る
  data = new float [width];

}

void draw()
{
  int Power;
//  println(1);
  if(port.available() > 1)
  {
    //Time = port.read(); //経過時間を読み込む
    Power = port.read(); //力を読み込む

    if(Power =='H')
    {
      int Power1 =port.read();
      int Power2 =port.read();
      int Power3=((Power1*100+Power2)*5);
      int Power4=Power3;
      float(Power4);
      float Power5=(Power4*9.8/1000);
      float Power6=Power5+(0.0642*Power5-0.0192);
     // nfc(Power6,1);

      background(255);
      text(nfc(Power6,1)+"[N]",10,40);
   //   println(Power5,Power6);
      for (int i=0; i<data.length-1; i++)
      {
        data[i] = data[i+1];
      }
      data[data.length-1] = Power6;
    }
  }
  // グラフの描画
  stroke(0,0,255); //線の色を青
  strokeWeight(5); //線の太さを 5pt

  for (int i=0; i<data.length-1; i++)
  {
   // println(convToGraphPoint(data[i]));
    //line( -300+i, -250+convToGraphPoint(data[i]), -300+i+1, -250+convToGraphPoint(data[i+1]) );
    line( i, convToGraphPoint(data[i]), i+1, convToGraphPoint(data[i+1]) );
  }
  stroke(0); //線の色を黒
  fill(0); //文字の色を黒
  strokeWeight(2); //線の太さを 2pt

  for(int i=0; i<8; i++)
  {
    line(0,100+i*50,600,100+i*50);
    text(24-i*3,570,100+i*50-10);
  }
      fill(255,0,0); //文字の色を赤
```

```
      stroke(255,0,0); //線の色を赤
   line(0,50,600,50);
    text(27,570,50);
    fill(0,0,255); //文字の色を青


}

float convToGraphPoint(float value) {
  println(height);
  return (500.0 - value*50/3);
}
```