

脳波を利用したハンドロボット制御に関する研究

S18174 橋本 昂典

1. はじめに

本研究では、脳波を利用してハンドロボット制御を行う。被験者が手を動かしている間の脳波を計測し、その脳波データを機械学習の手法を用いて分類し、手の動きを判別する。その結果からハンドロボット制御を行い、どの程度行えるかを明らかにする。手の動きとは、以下の4種類の形へ手を動かすことである。全ての指を曲げた“ぐう”の形、親指のみを伸ばした形、人差し指と中指のみを伸ばした“ちょき”の形、全ての指を伸ばした“ばあ”の形である。

本研究では、その準備段階としてハンドロボットの環境構築と制御プログラムの作成を行った。

2. 開発環境と使用機材

本研究ではハードウェアとして Raspberry Pi 4 を使用した。OS は Raspbian を使用した。使用言語は python を使い、統合開発環境は Thonny を使用した。

以下に紹介する機材は全て Hiwonder 社製の物である。ハンドロボットは uHandPi (図1) を使用した。

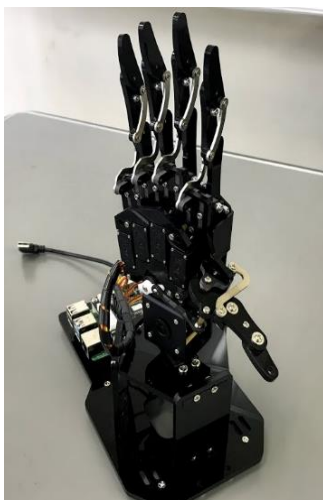


図1 使用したハンドロボット

5つのサーボモータが内蔵されており、指を別々に動かすことができる。また、サーボモータを制

御するための拡張ボードが Raspberry Pi 4 に取り付けられている。これを用いることにより、サーボモータを簡単に制御することができる。

3. デューティ比について

デューティ比とはパルス幅を周期で割った値であり、これを用いてサーボモータを制御する[1][2]。パルスの周期を“T”，パルスの High の時間を“H”とすると、デューティ比は次の式(1)のように求められる。

$$\text{デューティ比}[\%] = (H / T) \times 100 \quad (1)$$

3.1 デューティ比の計算

周波数は 50Hz としているため、周期を求めると、 $1/50 = 0.02[T]$ 、つまり 20ms になる。サーボモータの制御パルス幅は 0.5ms ~ 2.5ms であるため[3]、これをデューティ比に変換すると $0.5/20 \times 100 \sim 2.5/20 \times 100$ つまり 2.5% ~ 12.5% となり[4]、角度と対応させると、デューティ比が 2.5% の -90°、12.5% の時 90° となる。それぞれの値とサーボモータの動きを図に表すと以下の(図2)ようになる[5]。

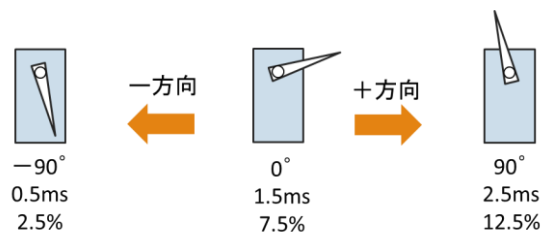


図2 サーボモータの動きと関係する値（上から角度、パルス幅、デューティ比）

また、ハンドロボットのサーボモータはプラス方向に動かすと指を伸ばし、マイナス方向に動かせば指を曲げる仕組みになっている。

3.2 角度からデューティへの計算

プログラム内では動かす角度を 90° などと度数で示すが、この値をデューティ比へ変換する必

要がある。角度からデューティー比を求めるには次の式(2)を用いる[4]。

$$\text{デューティー比} = 2.5 + (12.5 - 2.5) * (\text{回転する角度} + 90) / 180 \quad (2)$$

つまり、「デューティー比 = 最低値 + (最大値 - 最低値) * (回転する角度 + 90) / 180」となる。この式を用い、回転する角度がデューティー比の2.5% ~ 12.5%の間のいくつに値するかを求める。

4. Python による実装

プログラムの概要について説明する。まず初期設定を記述し、それぞれの手の形を作る関数（ぐう、親指のみ伸ばした形、ちょき、ぱあ）を記述した。次にメインプログラムとなる while 文を記述した。本研究は準備段階であるため脳波ではなく、キーボード入力からハンドロボット制御を行う。そのため、while 文中に文字入力プログラムを記述した。そして、ある特定の文字が入力された場合、“ぐう”など、それぞれの手の形を作る関数を呼び出し、実行する処理を記述した。

以下に“ちょき”の形を作るプログラムの一部を示す。while 文の繰り返し処理、中指を伸ばす処理を記載し、それ以外は省略とした。

```
1  flag_choki = False
2  #指を動かす while 文
3  while flag_choki == False:
4      flag_choki = True
5      ...省略...
6  #中指を伸ばす
7  if deg3 < deg_max:
8      deg3 += 1
9      duty3 = 2.5 + (12.5 -
10         2.5)/180*(deg3+90)
11     Servo3.ChangeDutyCycle(duty3)
12     flag_choki = False
```

まず、繰り返し処理について説明する。1行目でブール値“flag_choki”を False とし、while 文に入った後、4行目で True に変更する。7行目の“deg3”は中指の角度を表し、同行の“deg_max”は指を伸ばした状態のサーボモータの角度である。指の角度を変更する必要がある場合、7行目の if 文に入り、12行目でブール値を False とする。これにより、3行目からの while 文を抜ける

ことはなく、指を動かす繰り返し処理を行う。

指の処理が終了し、“deg3”が“deg_max”と等しくなった時、7行目の if 文に入らないため、ブール値は4行目に変更された True のままとなり、3行目の while 文から抜ける。この時点で手の形を作る一連の処理の終了となる。

次に実際に指を動かすプログラムについて説明する。7行目の時点で“deg3”に“deg_min”の値が入っていると仮定した場合、“deg3”は“deg_max”よりも小さいため、7行目の if 文へ入る。8行目はサーボモータをプラス方向に動かすため、“deg3”の値に1を足している。9、10行目は式(2)を用いて、デューティー比を計算し、“duty3”に結果の値を代入している。11行目では中指を動かすサーボモータのデューティー比を変更しており、この命令により、“duty3”で指定された角度へサーボモータを動かしている。そしてこれらの処理が繰り返し行われることにより、ハンドロボットの指を伸ばしている。

逆に指を曲げるプログラムは、7行目の“deg_max”を“deg_min”に、8行目の“+=”を“-=”へ変更するのみで概形は同じである。

5. おわりに

本研究では、ハンドロボットをキーボードから制御するプログラムを作成した。今後は脳波からハンドロボットを制御するため、機械学習の手法を学び、脳波解析についての研究を進める。

6. 参考文献

- [1] 「【ラズパイ電子工作】サーボモータを動かす方法(角度指定)」電機設計人.com, 2021.5.30 時点, <https://denkisekkeijin.com/raspberrypi/pi-servo-angle/>
- [2] 「備忘録 研究の備忘録」, Seesaa BLOG, 2021.6.14 時点, <http://hanmrabi.seesaa.net/article/439356627.html>
- [3] 「lewansoul lfd-01 マイクロサーボ 9 G ミニサーボ with anti-lock 保護 for RC カーロボット(Pack of 5)」, amazon.co.jp, 2021.6.28 時点, <https://www.amazon.co.jp/Lewan-Soul-LFD-01-Micro-Anti-Lock-Protection/dp/B07DLWSX9F>
- [4] 「RaspberryPi でサーボモーター動作」, JellyWare, 2021.6.14 時点, <https://jellyware.jp/kurage/raspi/servo.html>
- [5] 「dotstudio, inc.」, PWM 制御, 2021.6.14 時点, <https://dotstud.io/docs/pulse-width-modulation/>