

# 楽しい運動計測実習その1: データ解析編 (C, ver.3.0C)

2018 年 3 月 1 日

## 1 マニュアル作成 (グループ作業): 目標 2 日間

GitHub の bcl-group グループ (<https://github.com/bcl-group>) にリポジトリ”exp-manuals”を作成して、機器操作マニュアルを共同編集で作りましょう。

1. bcl-group への参加招待メールが届いてるはずなので、確認・承認して下さい。
2. リポジトリを作ったら、西井に連絡して下さい
3. マニュアルは筋電計用 (emg-manual.md) とモーションキャプチャ (mc-manual.md) の二種類を作成
4. 画像ファイルは各マニュアル毎に異なるディレクトリ (images-emg, images-mc) に格納
5. 書式は GitHub Flavoured Markdown で記述 (google で調べる)
6. 文章はなるべく簡潔に、しかし具体的に書くこと。
7. 作ったマニュアルは、GitHub 上で (も) 見ることが出来る。

## 2 データ解析 (各個人): 目標 4 日間

以下のデータ解析用プログラムを作りなさい。主に C やシェルスクリプトを使うものとして書いてあるが、python や R で書いても良い。ただし、シェルスクリプトは便利なので書かれている

### 2.1 準備: 目標 2 日間

python をよく知らない人は以下でまず勉強をすること。

1. python を Linux 環境にインストール (裏ページ参照)
2. Aidemy(<https://aidemy.net/>) の以下のコースのうち、よく知らないところをお勉強
  - (a) Python 入門
  - (b) Numpy を用いた数値計算
  - (c) Pandas を用いた数値計算
  - (d) Matplotlib によるデータ可視化
  - (e) データクレンジング (OpenCV の部分は不要)

### 2.2 筋電データの処理: 目標 3 日間

以下のプログラムを作りなさい。

1. 筋電位に 1~40 Hz の通過帯域を持つバンドパスフィルター (3 次バターワース) をかける。ただし、単純にバンドパスフィルターをかけるとデータのピーク位置が実際の時刻より少し遅くなることが多い。そこで、順方向と逆方向の両方から一回づつフィルター処理することで時間的なズレを補正する。この処理は、python ならば scipy の filtfilt 関数を使えば自動で行われる。

2. C 言語で筋電位  $E(t)$  を整流する (絶対値をとる)。
3. C 言語で適当な幅  $\Delta T$  の窓を設定して、その中で積分値を求め、筋肉の活動度を見る指標とする。

つまり、時刻  $t$  における筋肉の活動度 ( $a(t)$ ) は以下で評価することになる。

$$a(t) = \frac{1}{\Delta T} \int_{t-\Delta T/2}^{t+\Delta T/2} |E(t)| dt$$

これらの処理を行う理由を生データから考察せよ。

### 2.2.1 運動軌道データの切り取り

1. 計測によって得られた身体軌道のデータの内容を確認し、フォーマットとデータ (数値) の意味を把握しなさい。フォーマットは、使うソフトウェアの種類によって異なるが、大抵以下のようになっている。
  - (a) ヘッダ部 (始め): 計測条件、データの各列のタイトル
  - (b) データ部: シーン番号、マーカ ID、x 座標, y 座標, z 座標等
  - (c) テール部 (最後): 計測データを統計処理したデータが格納されている。
2. データファイルの名前を `joints.csv` とするとき、このファイルからデータ部のみを取り出した `pos-joints.dat` と、それ以外 (ヘッダ部とテール部) のみを取り出したファイル `info-joints.dat` を作成するシェルスクリプト `getdat.sh` を作りなさい。各出力ファイルの仕様は以下の通りとする。
  - (a) `pos-joints.dat` は、データ部をそのまま取り出す。
  - (b) `info-joints.dat` は、データ部以外を取り出す。(もしあれば) ダブルクォーテーション (") は取り除く。

取得データファイルにおいて、データ部の行頭が数字であることを利用すれば、`egrep` コマンドを使ってデータ部とそれ以外をそれぞれ抽出できる。

### 2.2.2 運動軌道データの処理

データファイル `pos-joints.dat` から、各関節の各座標データを抽出する処理を自動化したい。

1. 以下のようなプログラム `extract.c` を作りなさい。
  - (a) 実行時には以下のような引数を指定できるようにする。
 

```
$ extract <サンプリング周波数> <マーカ数> <入力データファイル名>
```
  - (b) 出力データに書き込む「時刻」は、引数で与えた「サンプリング周波数」と、入力データの「シーン」番号により計算する。
  - (c) 入力ファイルが `name.dat` のとき、出力ファイル名は計測点 ID を使って `1-name.dat`, `2-name.dat`, ... とする。もし、入力ファイル名が `pos-joints.dat` ならば、出力ファイル名は計測点 ID を使って `1-pos-joints.dat`, `2-pos-joints.dat`, ... である。
  - (d) 各出力データファイルのフォーマットは以下の通り。
 

**時刻, x 座標, y 座標, z 座標**
2. 以下のようなシェルスクリプト `extract.sh` を作りなさい。その仕様は以下の通りとする。
  - (a) 実行時には以下のような引数を指定できるようにする。
 

```
$ extract.sh <入力データファイル名>
```
  - (b) 前問の `extract` コマンドを、サンプリング周波数、計測点数、入力データファイル名を指定して呼び出して、各関節の xyz データを抽出する。

- (c) サンプリング周波数とマーカ数は先につくってある `info-*.dat` から抽出する。(例えば `grep` と `cut` を用いて抽出できる。シェルスクリプトで、あるコマンド `cmd` の実行結果を変数 `CMD` に格納するには `CMD='cmd'` とすれば OK)
3. 以下のようなシェルスクリプト `cut23` をつくりなさい。
- (a) 以下のように実行したら、入力ファイルの第 2,3,4 フィールド (xyz 座標データ) のみを抽出したファイルを出力する。(cut コマンドを使う)
- ```
$ cut23 1-pos-joints.dat
```
- (b) 上記のように入力ファイル名が `1-pos-joints.dat` ならば、出力ファイル名は `1-xy-joints.dat` とする。

## 2.3 レポート作成: 目標 1 日間

1. L<sup>A</sup>T<sub>E</sub>X を使って、実験結果をレポートにしない。
2. L<sup>A</sup>T<sub>E</sub>X の使い方は裏ページや、本棚の書籍等を参照のこと。
3. 生データや処理したデータはグラフ (x-t, y-t, x-y 等) にして考察しない。
4. **グラフ作成に表計算ソフトは使わず**, gnuplot, python, R, xmgrace (マニュアル有) 等のグラフ作成ツールを使って作成すること
5. 実験の目的、手法、結果、考察をきちんと書くこと。ただし、考察は数行程度で OK。
6. 結果をわかりやすく示す図表を作ること。絵を描くには LibreOffice 等を使う。表は L<sup>A</sup>T<sub>E</sub>X の表組機能を使うこと。
7. 掲載した図表は必ず本文中で引用して説明すること。
8. 図表の引用は `\label`, `\ref` を使った L<sup>A</sup>T<sub>E</sub>X の自動引用機能を使う事。