

## 0. 前提と方針

M5StickC を用いて歩数計を作る。作った歩数計 (M5StickC) はズボンの前・サイドポケットの中で計測すると限定する。ズボンのポケットなので、ポケットの付いた片足の動きが十分計測できる一方で反対側の足の動きが分からない。歩くとき、片足が動いた後、反対側の足が動くので、片足の「歩数」を検知して、結果を2倍にする方針で行う。

## 1. 考え方とアルゴリズム

歩く時、片足が動くと、もう片足は体を支えるために地面に着いたままである。この時の加速度は低いのに対し、そのあとその足が動き、加速度が高くなるだろう。そのあと再び体を支える足になり、加速度が低くなるという繰り返しである。

加速度が上がったり下がったりするので、ある時間帯において最高の加速度と最低の加速度がある。その最高と最低の加速度の間にうまく閾値を決め、その閾値を超える瞬間を1ステップとかカウントする。閾値の決め方は以下の二つの方法が考えられる：

- (ア) ある時間帯で観測された値において、最高の値と最低の値の平均を取る
  - (イ) ある時間帯で観測された値において、観測されたすべての値の平均を取る
- のいずれかである。

大まかなアルゴリズムは以下通りである。

- ① センサーから3軸の加速度データを取得し、その3つとも加速度ベクトルの足し算からなる合成ベクトルのノルムを計算する。これを現時点の加速度である。
  - ②  $N$  を指定し、この  $N$  回に渡って①で得た加速度上の (ア) または (イ) を計算する。これを判断基準(以下 *baseline*) とする。つまり、 $N$  回加速度が検知するたびに *baseline* が更新される。
  - ③ 現時点の加速度  $a_t$  が  $a_t > baseline$  かつ一個前に観測された加速度  $a_{t-dt}$  が  $a_{t-dt} < baseline$  のとき、1歩としてカウントする。
- ③で得られたのが片足の歩数であるので、実装では③は1歩でなく2歩としてカウントアップする

## 2. 実装

大まかのアルゴリズムは1で説明したが、実装にあたって1つ大きな問題に出会った。静止しているとき、現時点の加速度がノイズにより *baseline* より上がったり下がったりするため、静止しているのにステップとしてカウントされる。対象法として、以下の

二つを実装した

1. ①と②の間にデジタルローパスフィルタを当てた。つまり、 $a_t = (\text{passCoef}) \cdot a_{t-dt} + (1 - \text{passCoef}) \cdot x_t$ 。ただし、ここで  $x_t$  は実際に観測された値であり、 $0 \leq \text{passCoef} \leq 1$  を満たす。
2. ③での一つ目の条件を  $a_t > \text{baseline}$  から  $a_t > \text{baseline} + \epsilon$  に変更する。ただし、 $0 \leq \epsilon \leq \text{baseline}$ 。

M5StickC を起動した際、以下の左のような画面が表示される。これは検知された歩数を表す数字となっている。ボタン A を押せば、画面が以下の右に変わり、 $a_t$  と  $\text{baseline}$  をプロットしたグラフが表示される。また、B ボタンは、ステップカウンタをリセットする機能を追加した。現状がどちらの画面でもリセットは可能である。



### 3. パラメータと実験

上の説とプログラム内の変数名の対応関係は以下通りである：

- $a_t$  のプログラム内変数名 : `accel` (現時点の加速度)
- $\text{baseline}$  のプログラム内変数名 : `baseline`
- プログラム内変数 `step` は歩数を格納する。

以下のパラメータで実験を行う。パラメータの微調整によって、精度が変わる。このパラメータは最適な精度を出すものではないが、時間的に許される範囲ではこれが妥当なパラメータとなっている。

- $N = 25$  (プログラム内変数名 : `BASE_COUNT`)
- $dt = 100\text{ms}$  (プログラム内変数名 : `DELAY`)
- $\text{passCoef} = 0.2$  (プログラム内変数名 : `passCoef`)
- $\epsilon = 0.2 \cdot \text{baseline}$  (プログラム内変数名 : `width`)、現在の  $\text{baseline}$  によって動的に計算される

アルゴリズム(ア) (最大値と最小値の平均を  $\text{baseline}$  にする) で実験 1 0 0 歩を 5 回実験してみた結果、

104, 90, 88, 94, 86

が観測される。一方、アルゴリズム (イ) (移動平均を  $\text{baseline}$  にする) で同じ実験を試してみた結果、

154, 158, 194, 200, 176

が得られる。

#### 4. 考察

アルゴリズム(ア)ではかなり想定した結果に近いものの、アルゴリズム (イ) を使用するならば、片足の動きを検知するものよりも全体の真の歩数に近い。実験では両方ともステップカウントするたびに+2 がカウントアップされる仕組みである。これにより、アルゴリズム (イ) を使うなら+2 ではなく毎回+1 で精度がよくなると言える。しかし、実験結果から、アルゴリズム (イ) の実験データが不安定で分散が大きいのにたいし、アルゴリズム (ア) 毎回安定している結果が出る。そのため、今回提出する M5StickC に搭載されるプログラムはアルゴリズム (ア) を使用したものとする。実際、どちらのアルゴリズムを使うかは提出するプログラム内の useMidpoint (フラグ型変数) を変えるだけである。

#### 5. 参考資料：

<https://qiita.com/ufoo68/items/d0d12c666a19405c5625>

<https://www.hackster.io/610t/m5walker-step-counter-for-m5stickc-99e1b0>