

신호 및 시스템 프로젝트 보고서

가속도계를 사용한 걸음 수 측정 신호처리 Matlab 설계 및 실험

201921120 우다현

I. 서론

1. 프로젝트의 목적

주어진 신호에 대한 산업현장에서 사용되는 Matlab 신호처리 기법을 익힌다. 주어진 기술요구서를 잘 이해하고 프로그래밍 및 검증하는 방법을 익힌다.

2. 프로젝트의 개요

걸음 수 검출 알고리즘은 웨어러블 디바이스 환경에서 3 축 가속도계 센서의 x, y, z 축 가속도 신호를 통해 사용자의 걸음 수를 계산하는 알고리즘이다.

알고리즘은 스마트폰 혹은 기기 내에 내장된 가속도 센서를 사용하여 걸음수를 측정한다. 아래와 같이 가속도 센서에서 발생하는 데이터를 수집하여, G 단위 신호 스케일링, 가속도 모멘텀(momentum), 양자화(Quantization) 과정을 거쳐서 걸음 수 측정에 필요한 알고리즘을 거쳐서 걸음 수를 측정하게 된다.

웨어러블 디바이스를 이용한 걸음 수 측정은 현재 매우 대중화되어 있으며, 개개인은 자신의 운동 효과를 디바이스를 통해 확인한다.

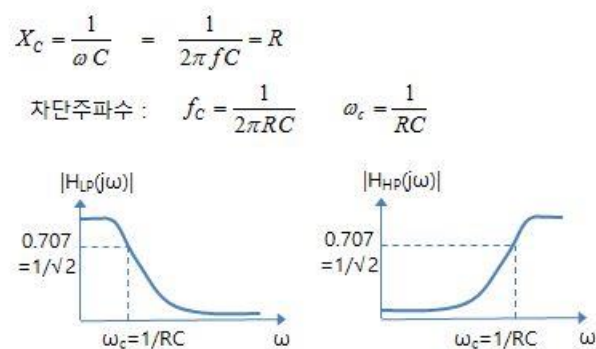
이렇듯 수요가 많고 지속적인 발전을 필요로 하는 분야이므로 걸음 수 측정 알고리즘을 개발하고 분석하는 과제의 의미를 찾을 수 있다.

II. 본론

1. 이론

1) 필터의 차단주파수(cu-toff frequency)

차단주파수(cut-off frequency)란 특정 주파수의 신호를 통과시키거나 차단시키는 대역의 경계점이다. 따라서 노이즈를 제거할 때 사용되는 필터에서 차단주파수를 적절히 설정함으로써 설계자가 원하는 주파수를 차단시키고 통과시킬 수 있게 된다.



2) 필터의 종류

필터는 특정한 신호에서 원하지 않는 신호를 차단하거나 원하는 신호만 통과시킨다. 여러 필터가 존재하는데, 이번 실험에서는 차단주파수보다 낮은 영역의 주파수를 통과시키기 쉬운 필터인 LPF(Low Pass Filte)를 사용했다.

3) 필터의 위상차

신호를 필터링하면 지연이 발생한다. 이는 출력 신호가 입력 신호에 비해 시간적으로 이동하는 것을 의미한다. 이동 크기가 일정할 경우, 신호를 시간적으로 이동하여 지연을 해결할 수 있다.

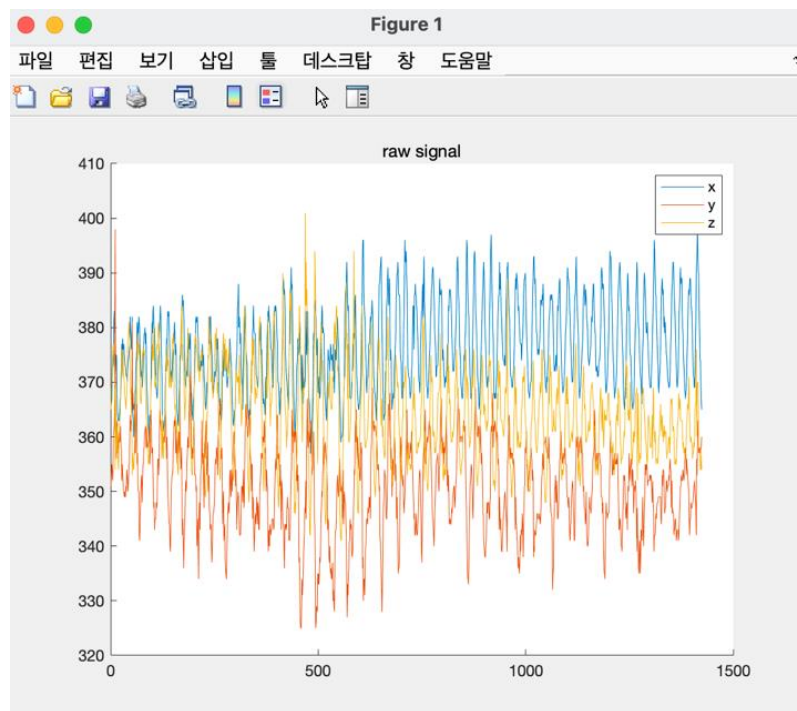
때때로 필터는 특정 주파수 성분을 다른 주파수 성분보다 더 지연시키는데, 이 현상을 위상 지연이라고 한다. 이러한 위상 지연이 있음을 고려해야 한다.

2. 결과에 대한 분석

1) G 단위 신호 스케일링(단위 정량화)

가속도 센서의 임베디드 설정에 따라 측정값의 범위가 달라질 수 있으므로 가속도 크기의 단위 중 하나인 중력(G) 단위 가속도 변환하여 사용한다.

Raw 데이터를 출력하면 다음과 같다. 세로축은 데이터의 크기, 가로축은 시간/sample rate 이다.



스케일링 수식은 다음과 같다.

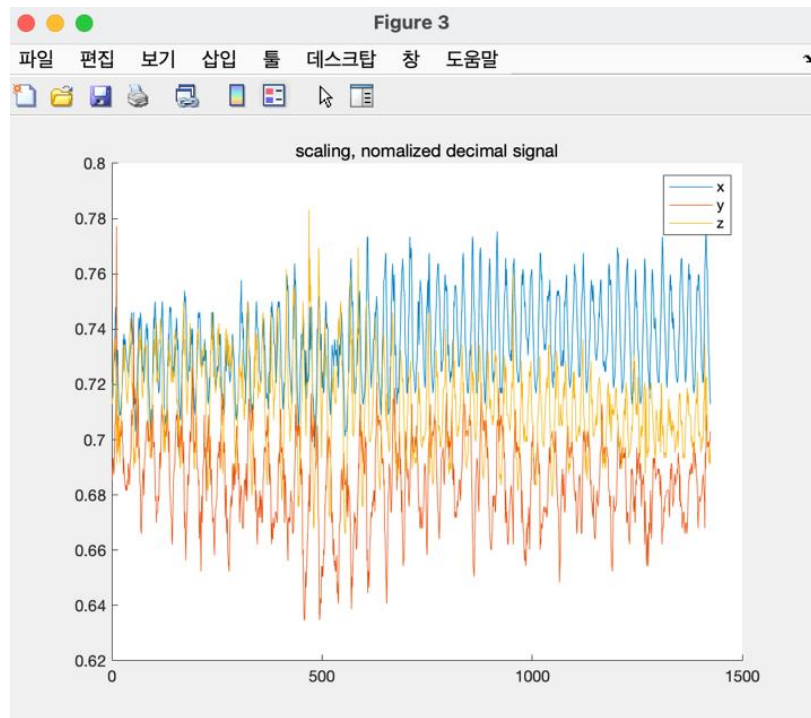
$$G_{divisor} = \frac{G_{saturation} \cdot 2^{(N_{resolution} + 1)}}{N_{resolution}}$$

수식에서 $G_{saturation}$ 은 가속도 센서 측정 범위 설정 값이다. 이 값을 데이터에 맞게 수정해주어야 한다.

(스케일 설정). 수식에서 $N_{resolution}$ 은 ADC(Analog to Digital Converter)의 분해능 설정 값이다. 이 값은 측정하는 기기의 ADC 설정에 따라서 달라진다. (측정한 기기의 ADC 범위 : 0~1023). 총 1024 개이며 이는

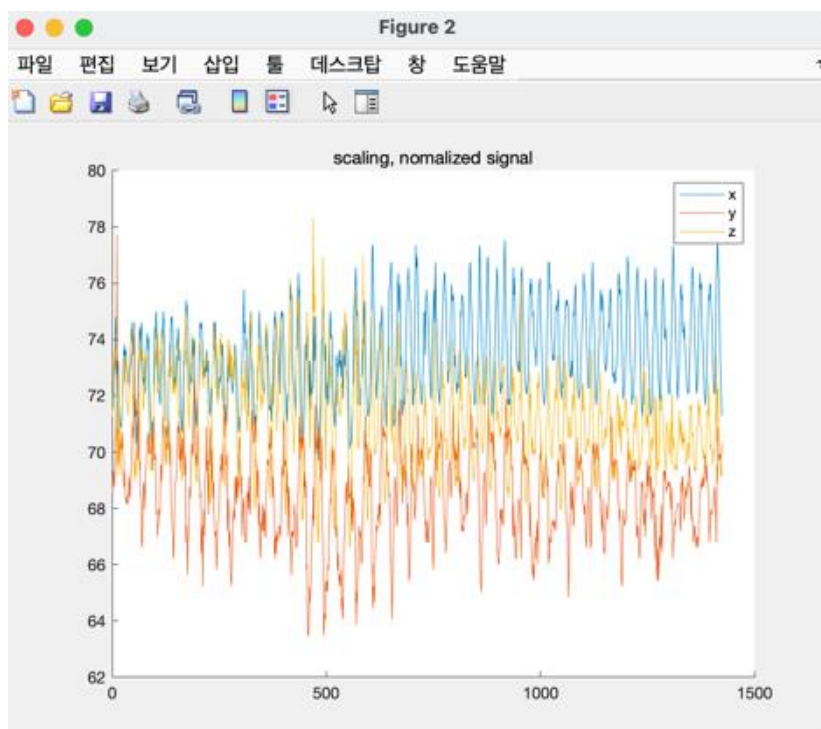
2^{10} 이다. G 값은 raw data 의 max 값이다. 수식에서 G_{divis} 는 위 설정 값에 의해 측정할 수 있는 단위 중력 가속도 크기이다.

스케일링 처리한 가속도 데이터를 출력하면 다음과 같다.



데이터의 오버플로우를 방지하기 위해 0~1 사이의 값으로 조정한다. 이를 위해 데이터를 100 으로 나눈다.

데이터 오버플로우 방지 처리를 한 후 데이터를 출력하면 다음과 같다.

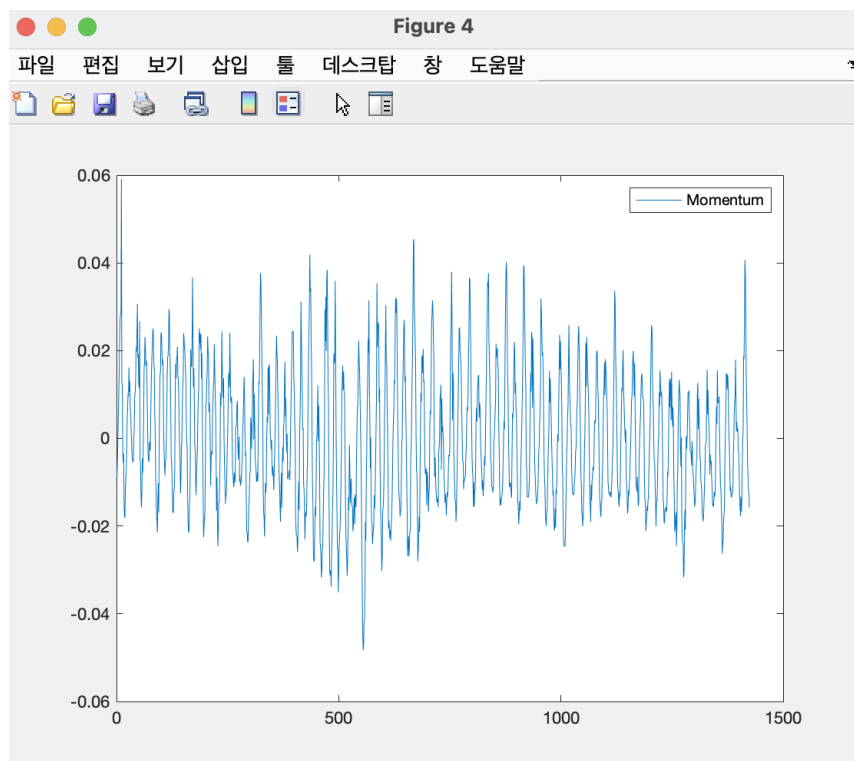


2) 가속도 모멘텀 산출

모멘텀(Momentum)은 운동량으로서 물체의 질량과 속도의 곱으로 표현된다. 가속도 신호에서의 모멘텀은 각 축별 가속도 신호에서 방향성을 제거한 가속도 크기를 뜻한다. 가속도 신호는 중력가속도를 측정하기 때문에 착용방향에 따라서 다른 파형의 신호가 얻어질 수 있다. 가속도 신호의 모멘텀을 산출하면 센서의 방향성을 고려하지 않고 가속도 신호를 처리 할 수 있다. 모멘텀(m,Momentum) 수식은 다음과 같다.

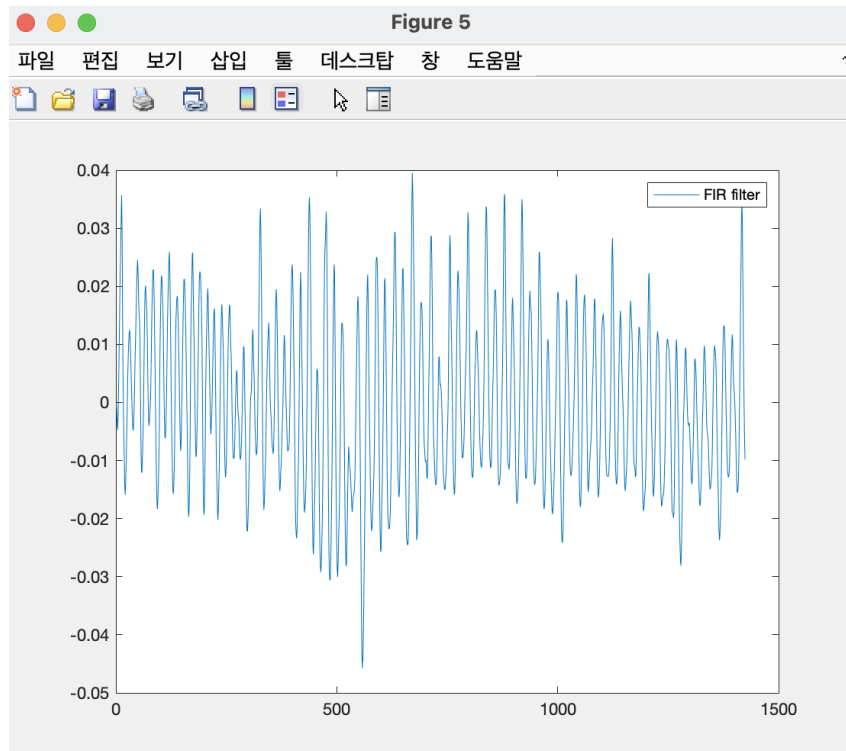
$$m = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

모멘텀을 구한 후 모멘텀의 값을 0 을 기준으로 normalization 시키기 위해 평균값을 구한다. 모멘텀에서 평균값을 뺀 값을 출력한다. 그 결과는 다음과 같다.



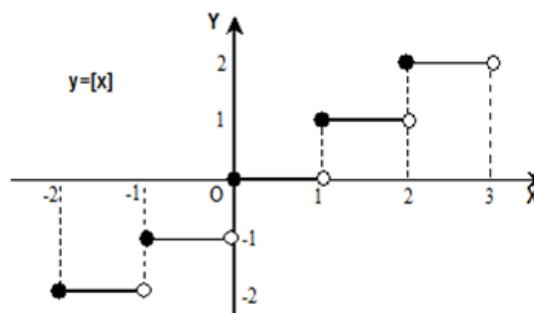
3) 모멘텀 신호 전처리(노이즈 제거)

모멘텀 신호 그래프에서 잡음 성분을 제거하기 위해 LPF(Low Pass Filter)를 적용한다. Designfilt 함수를 사용했다. 세부적으로는 Sampling Frequency = 32Hz, Order = 5, Cut-off Frequency = 0.5Hz 로 설정했다.



4) 모멘텀 신호 양자화

가우스 함수(Gauss,[x])



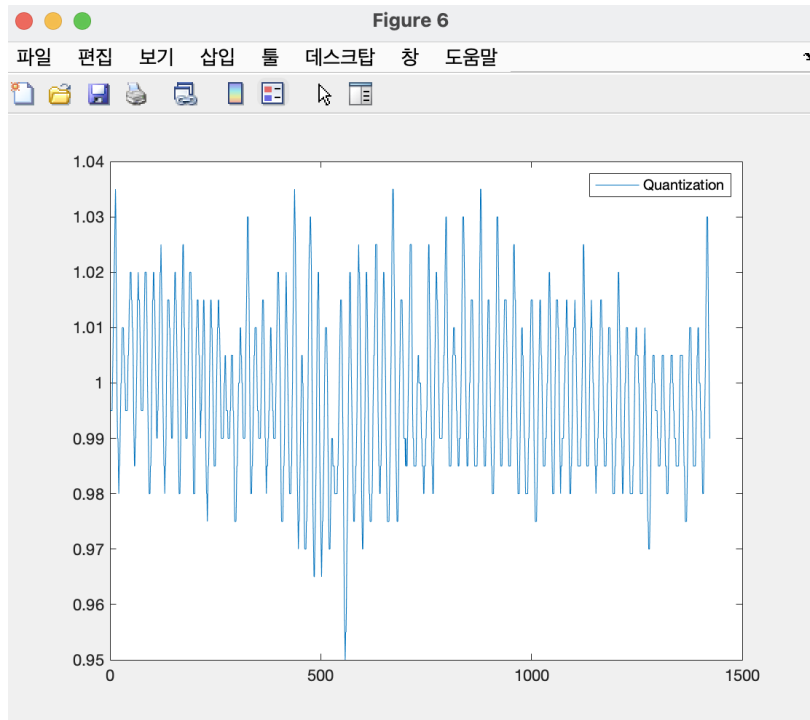
양자화된 모멘텀 값 수식(m_q)

$$m_q[i] = \text{Gauss}((m_f[i] + Q_{bias})/Q_{level}) \times Q_{level}$$

수식에서 m_q 는 양자화된 모멘텀의 값이다. 수식에서 m_f 는 필터링된 모멘텀의 값이다. 수식에서 Q_{level} 은 양자화 시 나누는 단계를 말한다. 수식에서 Q_{bias} 는 양자화 시 [0.9, 0]의 값들을 0으로 만들어주기 위해

더해준다. 걸음 수 검출을 위해 Q_{level} 과 Q_{bias} 를 적절한 값으로 설정한다. 우리 조는 양자화 시 나누는 단계. Level factor 는 데이터가 최소 5 단계 이상으로 분리될 수 있게 해야 하므로 Q_{level} 은 0.05, 양자화 시 $[0.9, 0]$ 의 값들을 0 으로 만들어주기 위해 Q_{bias} 는 1 로 설정했다.

양자화를 적용한 데이터 출력하면 다음과 같다. 이때, 모든 신호는 특정 level 의 값만을 갖는다.



5) 피크/밸리(Peak/Valley) 검출

양자화 된 신호로부터 걸음 수를 검출하기 위해 피크와 밸리 포인트를 이용하기 때문에 아래 수식들을 이용하여 피크/밸리 포인트를 검출한다.

$$S[i] = \begin{cases} 1 & m_q[i] > m_q[i-1] \\ 0 & m_q[i] < m_q[i-1] \\ S[i-1] & otherwise \end{cases} \quad P[i] = \begin{cases} peak & S[i] > S[i-1] \\ valley & S[i] < S[i-1] \\ none & otherwise \end{cases}$$

왼쪽 수식에서 S 는 상태를 나타내는 값이다. 현재값과 이전값을 비교하여 상태를 나타내는 값을 설정한다.

오른쪽 수식에서 P 는 해당 지점의 값이 피크/밸리/둘 다 아님 인지를 나타내는 값이다. 이전상태와 현재상태를 비교하여 피크/밸리/둘 다 아님을 결정한다.

코드에서는 상태 변수를 선언한 후 신호가 올라가는 상태일 경우 $state=1$, 신호가 내려가는 상태 일 경우 $state = 0$ 으로 두고 피크와 밸리를 결정했다. 또한, peak 값 혹은 valley 값이 아닐 경우 값을 ylim 밖으로 설정해서 처리했다.



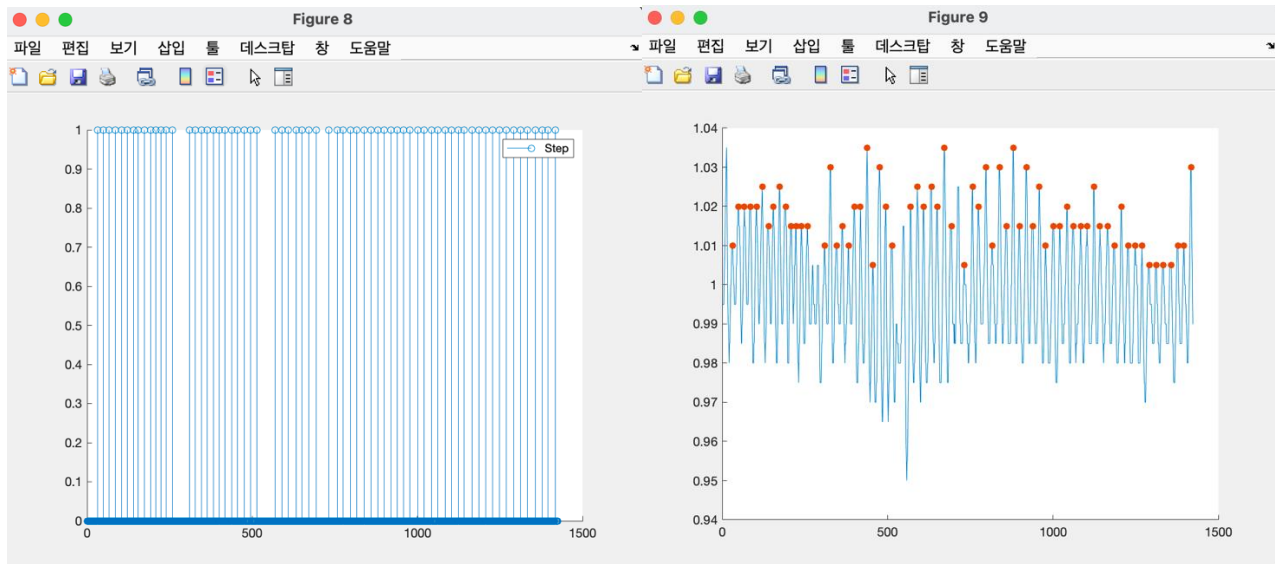
6) 걸음 수 검출

현재의 피크 값, 이전 피크 값, 현재 피크 값과 이전 밸리 값의 차이, 피크사이의 거리를 통해 '걸음 판단 흐름도'를 작성하고 흐름도를 모두 통과하면 걷기 행동이라 판단한다. 걸음으로 판단된 점들을 모두 더하는 것으로 걸음 수를 검출하게 된다. 코드에서 Threshold 값들을 조정한다. Threshold 값은 Active, Swing, Interval long time, Interval short time 이 존재한다. long time 과 short time 을 조정하여 피크 사이의 최대 최소 간격을 조정한다. long time, short time 을 sample rate 와 곱하여 데이터에 맞게 시간을 조정한다. swing 값 조절을 통해 valley 와 peak 의 모멘텀 차이가 걸음걸이의 모멘텀과 일치하는지 확인한다. active 값 조절을 통해 최소 피크 값을 설정해준다.

즉, 양자화된 모멘텀 데이터로부터 걸음 발생을 정의한다. 걸음 발생의 조건은 다음과 같다. 첫 째, 이전 peak 와 현재 peak 모두 ACTIVE 문턱값을 넘어야 한다. 둘 째, 현재 peak 와 이전 valley 의 차이는 SWING

문턱값을 넘어야 한다. 셋째, peak 와 peak 사이 시구간이 LONG_SAMPLE 보다 작고, SHORT_SAMPLE 보다는 커야 한다. 더 자세한 설명은 코드의 주석으로 첨부하였다.

출력 결과는 다음과 같다.



7) 실험 결과

$$R = \frac{N_r - |N_r - N_a|}{N_r} \times 100(\%)$$

인식률 산출 수식

제안된 알고리즘을 바탕으로 걷기와 실험 후 추정된 걸음수와 정확도를 산출했다. R 은 상대정확도로 위 수식을 이용하여 구한다. 수식에서 N_r 은 실제로 측정된 횟수를 의미한다. 수식에서 N_a 는 신호에 대해서 알고리즘이 추정한 횟수를 의미한다. 나눠준 데이터는 실제 71 회의 걷기를 수행한 데이터이다. 인식률 산출 수식에서 N_r 은 71 회를 사용했다.



출력 결과 경과 시간은 0.829466 초, 인식률은 94.4%로 나왔다.

(알고리즘 추정 횟수는 67 회이다. 발걸음수는 step_signal 발생점을 모두 더하는 것으로 알 수 있다.)

III. 결론

본 실험에서는 가속도계를 사용한 걸음 수 측정 신호처리 Matlab 을 설계하고 실험을 진행하였다. 그 결과 주어진 문제와 힌트의 조건들을 모두 만족하는 만족스러운 결과를 얻었다.

알고리즘의 경과 시간은 0.829466 초로 짧았다. 인식률 산출 수식을 바탕으로 알고리즘의 인식률을 산출하면 94.4%로 정확했다.

오차가 발생한 원인으로는 여러 가지를 생각해볼 수 있다. 일반적으로 데이터를 측정하는 과정에서 오차가 있었을 수 있다. 또한, 데이터 양자화 과정에서 변수의 값을 조절할 때 오차가 있었을 수 있다. 각 지점의 Peak 와 Valley 사이의 거리 값의 평균을 통해 산출해서 걸음을 구분할 때 평균거리보다 짧지만 걸음일 경우가 있을 수도 있다.

이러한 오차를 줄이기 위한 방법은 다음과 같다. 데이터를 측정할 때 오차가 없도록 통제 변인을 잘 조절한다. 또한 데이터 양자화 과정에서 값을 더욱 세밀하게 조절한다. 여러 실험을 통해 걸음을 판단하기 위한 거리의 평균값을 최적의 값으로 찾는다.