Step Counter

Mackenzie Wieberg
CSCE 462, Fall 2020

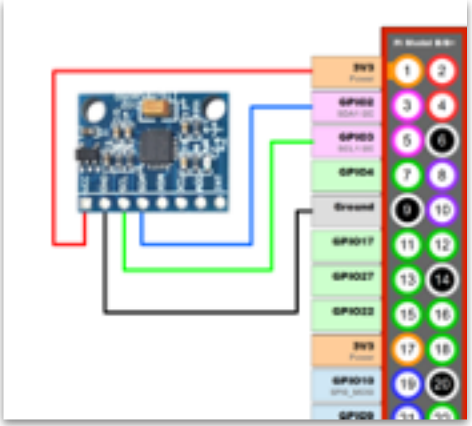
Materials

- Raspberry Pi 3 B+ & power adapter
- MPU-6050
- Jumper wires
- Soldering Iron & solder
- Sweat band or other arm band
- Breadboard & ribbon cable (optional)









MPU-6050

- 3 axis accelerometer and gyroscope
- Uses I2C - power and communication with processor
- 8 pins
  - **VDD**
  - **GND**
  - **SCL**
  - **SDA**
  - XCL
  - XDA
  - ADO
  - INT
- You must solder the pins to the chip

Connections:

MPU-6050	Raspberry Pi 3 B+
VDD	3V3 (pin 1)
GND	GND (pin 9)
SCL	SCL1 (pin 5)
SDA	SDA1 (pin 3)

want to get them "cartoon abstraction"

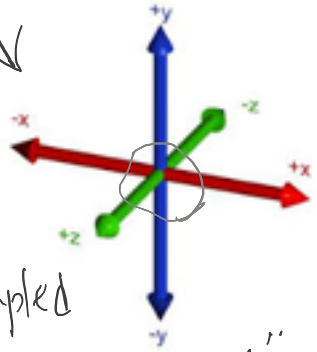
Modeling of "steps"

- Each step is a motion made by a person wearing the sensor
- Numerous ways to characterize different types of "steps"
    - Jogging, walking Q: how many different states it can have?
- Approximation is needed to measure steps
    - Steps:
        - Changes in readings on the x, y, and z axises
        - Both + & - readings represent opposite motion directions
    - Measurement:
        - "Stationary positions": the body rebalances itself for the next step, sensor readings very small
        - "Transitive positions": the motion of a step, large sensor readings

different transitional rates

"connection"

time



sampled "data points"

Data model and processing algorithms

- High sampling rate: 10ms interval, or 100 samples/sec.
  - minimize loss of motion data, but noise is mixed into real data => noise filtering is required
- Differentiation of human stationary vs. transitive motion data
  - Transitive positions: large readings for a period
  - Stationary positions: nearly no readings for a period
- Algorithms

  elimination of unwanted values

  - Filtering rule => small motion data can be discarded, → may introduce discontinuity
  - Smoothing of filtered data to reconstruct human motions
  - Measure steps by counting stationary positions

estimate missed real values

hole

"real" values

Algorithms

Raw Data	Noise Filtering	Smoothing	Is it a step?

Raw Data

- x, y, z
- Directly from the accelerometer
- S = |x| + |y| + |z|
- S is used to measure the level of motion along all directions

* So, this method cannot differentiate the motion direction.

Noise Filtering

- Smax = max[S(n-9), S(n)]
- Only picks the largest motion components collected within the given sample/window length
- Filters S to eliminate minority motion components aka “noise”

"Generalization"

Smoothing

- Moving Average = average[Smax(n), Smax(n+9)]
- Smoothing effect
- Reduces fluctuation

"window size"

what kind of window

Is it a step?

- if S(n-1) < Moving Average(n) and S(n) >= Moving Average(n) and
- If Moving Average(n) > threshold
- Find the peaks of the graph and compare to the threshold to determine if peak counts as a step

Implementation

- Hardware
  - Sensor placement (chest, foot, wrist, etc...)
  - Wiring, prototype board, power
  - Communication between processor and sensor
- Software
  - Language
  - Library
  - etc...

optimal working ranges

Experiment Variations & calibration

robustness

user factor

which type of users

"speak a ~~few~~ words"

known

- Considering factors
  - Human behaviors
  - The type(s) of steps that we aim to measure
    - Their dynamic ranges and thus for algorithm calibration
  - Sensor placement - proxy of the actual steps
- Algorithm calibration (hint: every adjustable parameter)
  - Sampling rate
  - Filtering rule
  - Transitive positions
  - Stationary positions
  - …

train the

actual working

alg.

Experiment Variations & calibration

- One of the most critical step to match the algorithm with human behaviors

- Experiment 1: BASELINE
    - Threshold = 15
    - Sample Length = 10 ms
- Experiment 2: Vary Threshold
    - 2a
        - Threshold = 9
        - Sample Length = 10 ms
    - 2b
        - Threshold = 21
        - Sample Length = 10 ms

- Experiment 3: Vary Sample Length
    - 3a
        - Threshold = 15
        - Sample Length = 1 ms
    - 3b
        - Threshold = 15
        - Sample Length = 100 ms

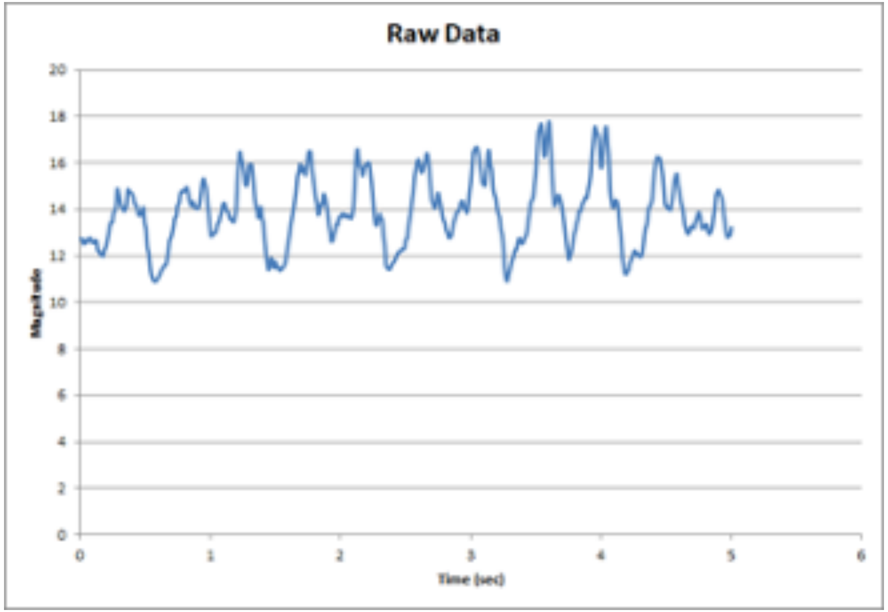
Raw data filtering effect

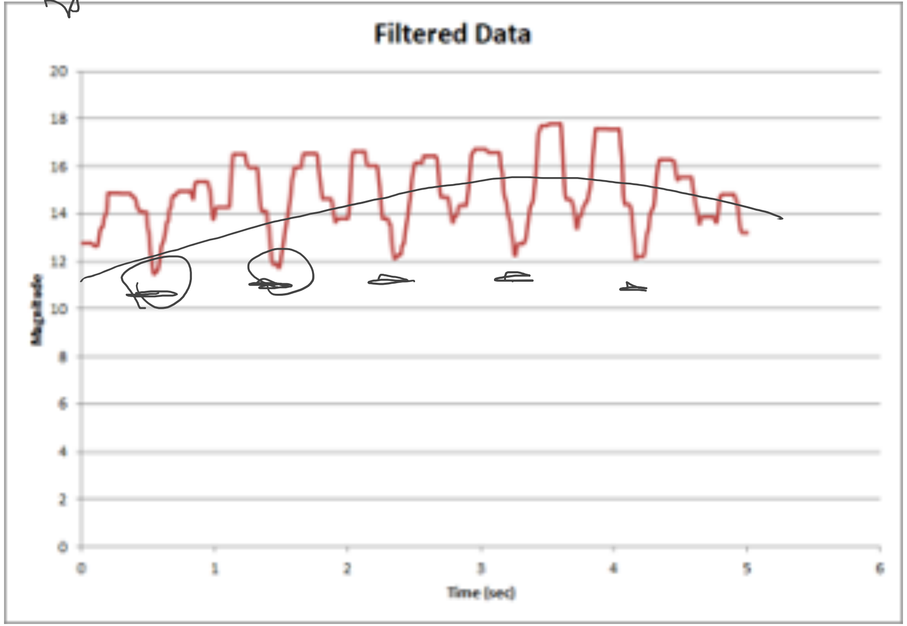
key challenge

how to make this useable

Raw data graph

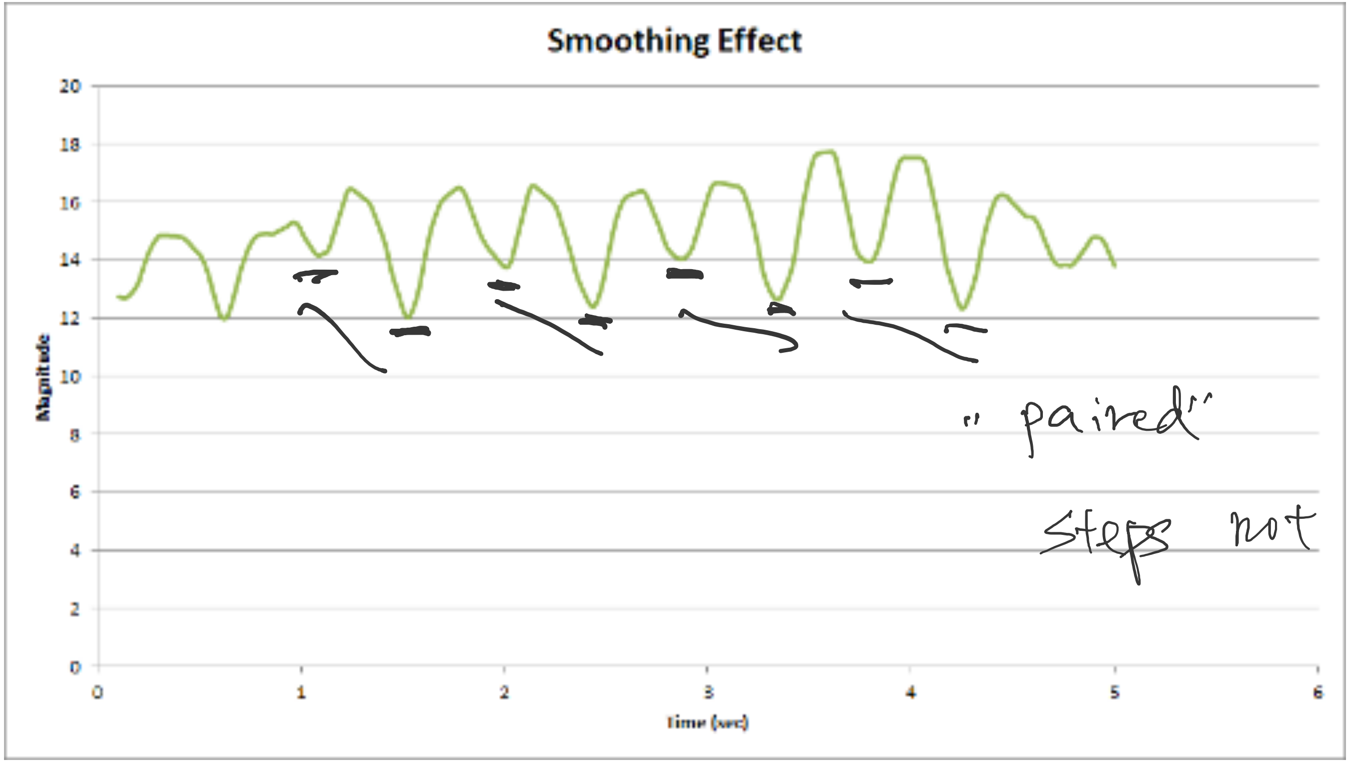
Filtered graph





Smoothing effect

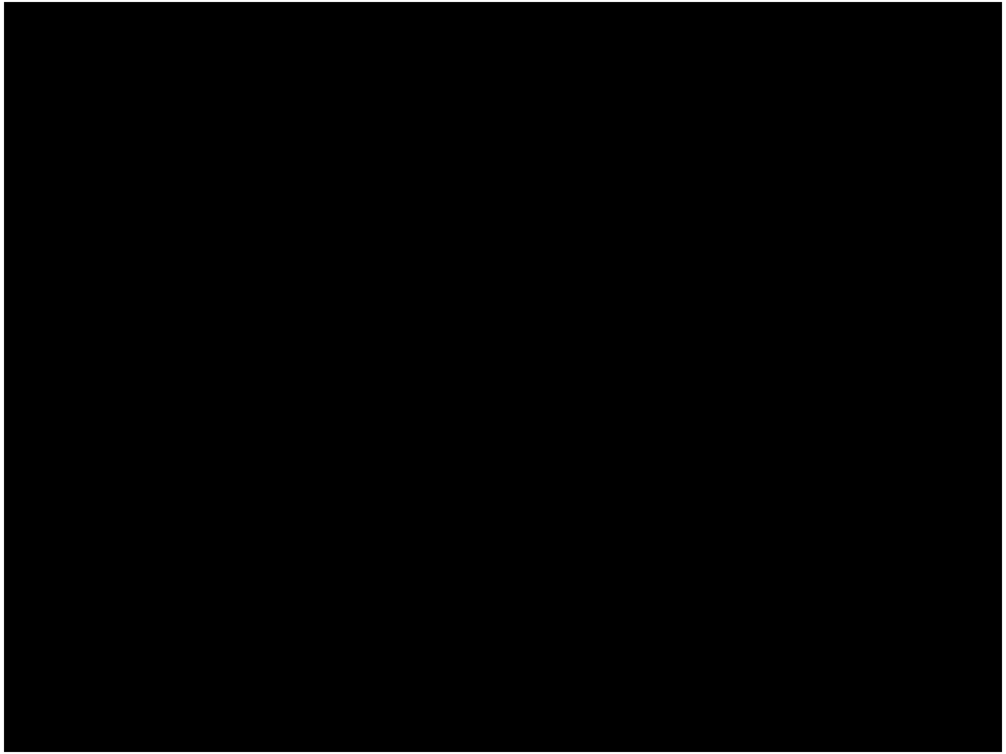
~~intert~~ interpretation → real world actions



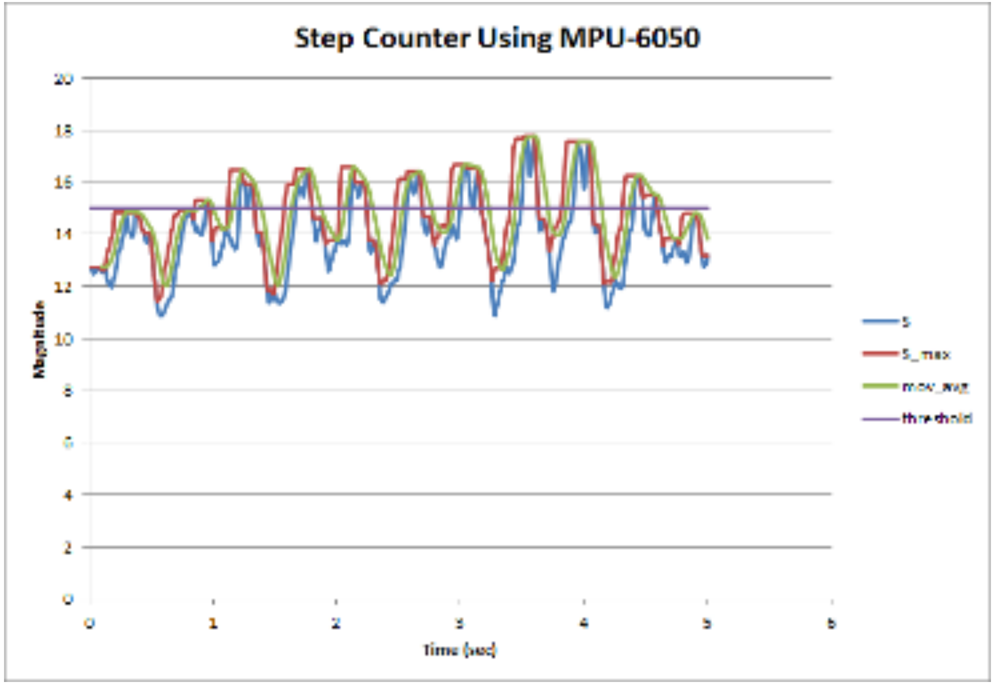
"paired"

steps not uniform

Experiment 1: BASELINE VIDEO



Optimally Calibrated Use-case
(threshold = 15, sample/window length = 10 ms)

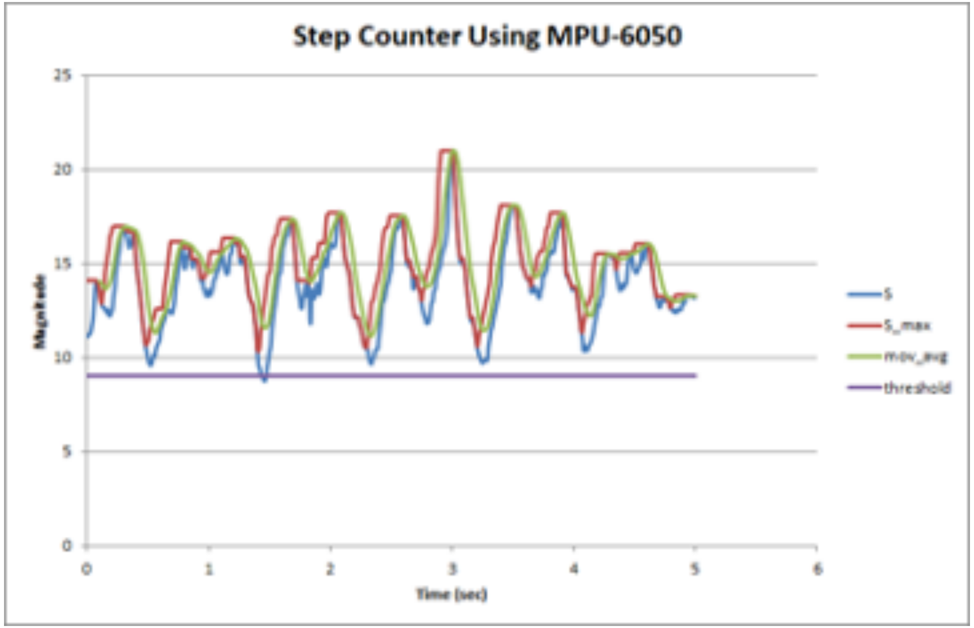


Outcomes:

- Actual Steps: 10
- Counted Steps: 10

Calibrated threshold and sample length allow for accurate step calculation.

Uncalibrated Threshold Use-case #1
(threshold = 9, sample/window length = 10 ms)

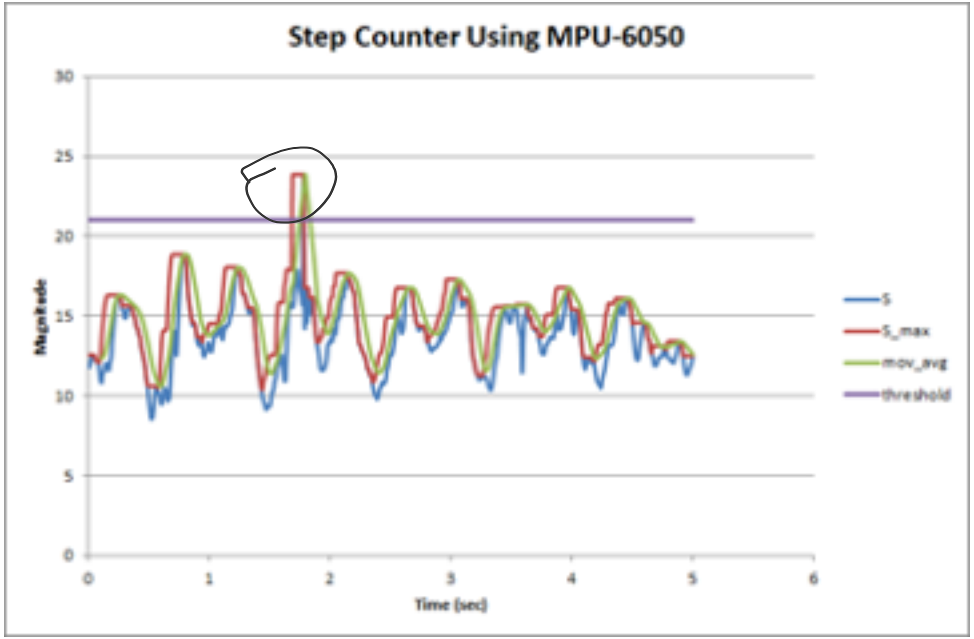


Outcomes:

- Actual Steps: 10
- Counted Steps: 17

Lack of threshold allows micro-movements to be counted as steps.

Uncalibrated Threshold Use-case #2
(threshold = 21, sample/window length = 10 ms)

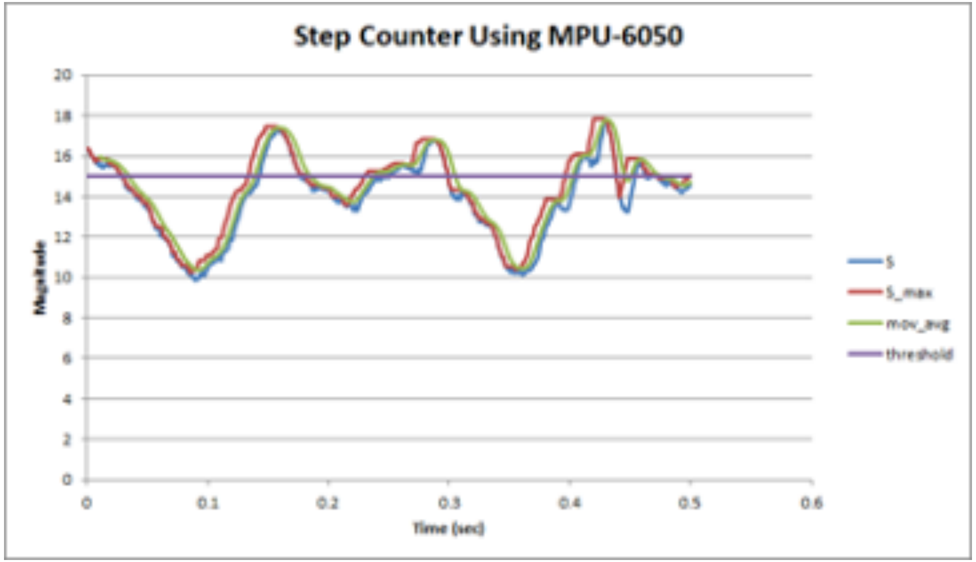


Outcomes:

- Actual Steps: 10
- Counted Steps: 1

Too high of threshold blocks anything from being counted.

Over-Sampled Use-case
(threshold = 15, sample/window length = 1 ms)

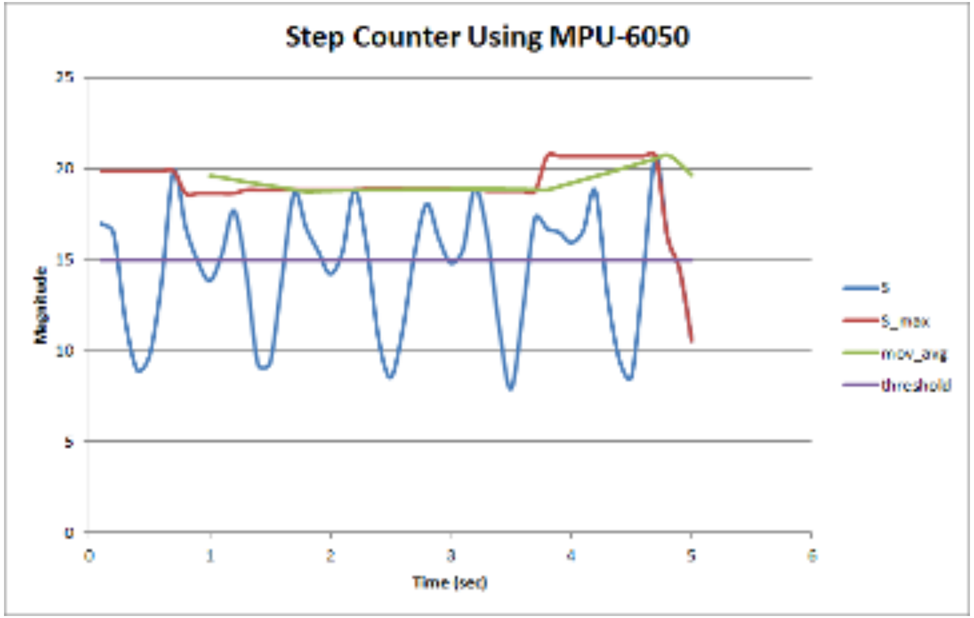


Outcomes:

- Actual Steps: 10
- Counted Steps: 19

Oversampling, too much data to analyze allows too much noise to get through the filter.

Under-Sampled Use-case
(threshold = 15, sample rate from 10ms to 100 ms)



Outcomes:

- Actual Steps: 10
- Counted Steps: 3

Undersampling, not enough data to sample skews the algorithm.

Resources

https://drive.google.com/drive/folders/1-ilULNyBdV8NkojHhglQZhAuE76It2JF?usp=sharing

https://github.tamu.edu/mackenzie-wieberg/step_counter_mpu6050

Purchase MPU-6050