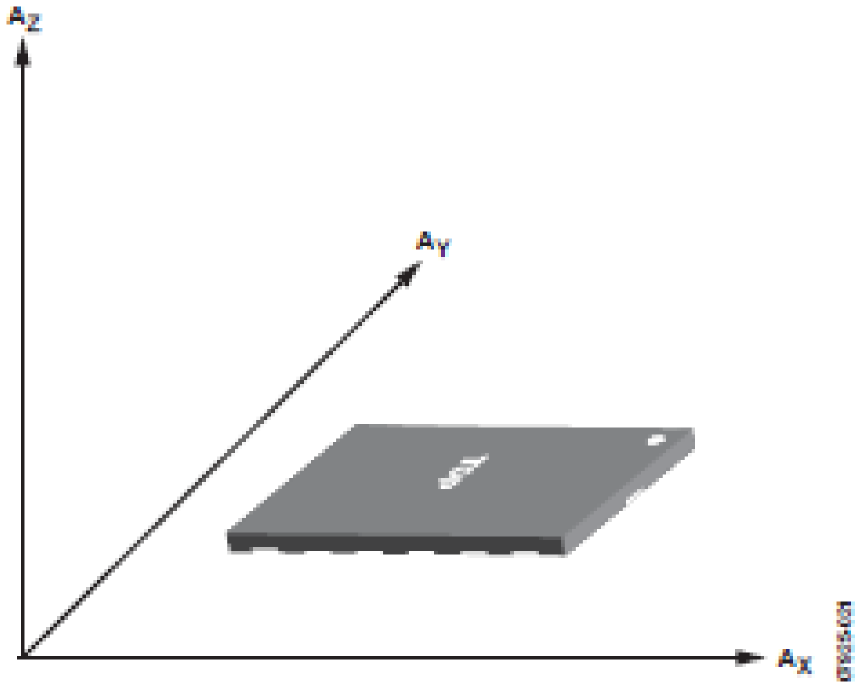
# ENGR 498: Design for the Internet of Things – Signal Processing and Step Analysis

Dr. Jason Forsyth
Department of Engineering
James Madison University

# Accelerometer Measurements



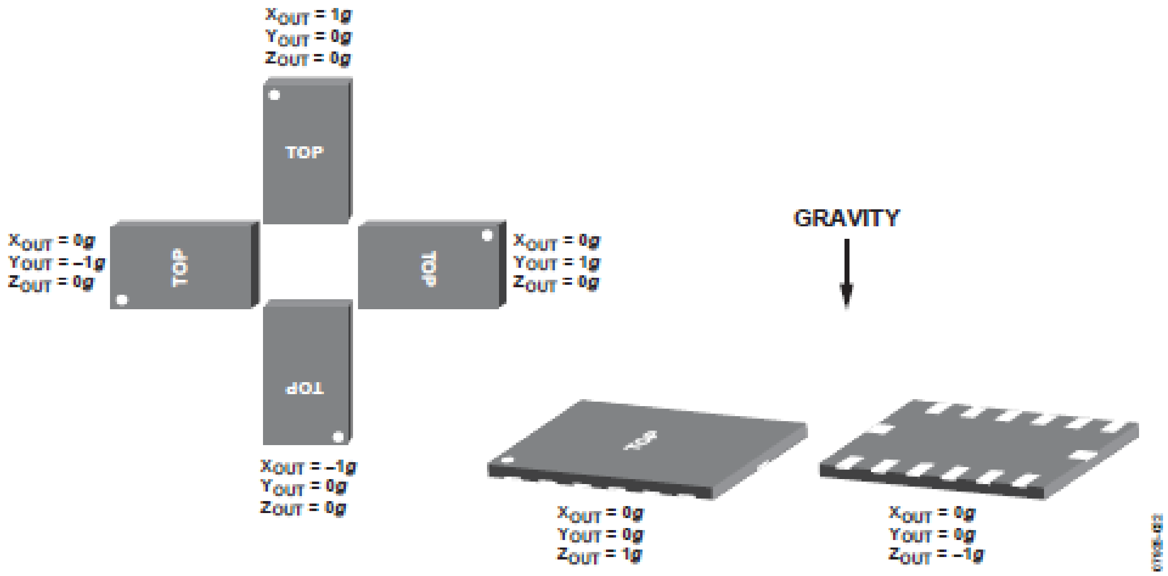
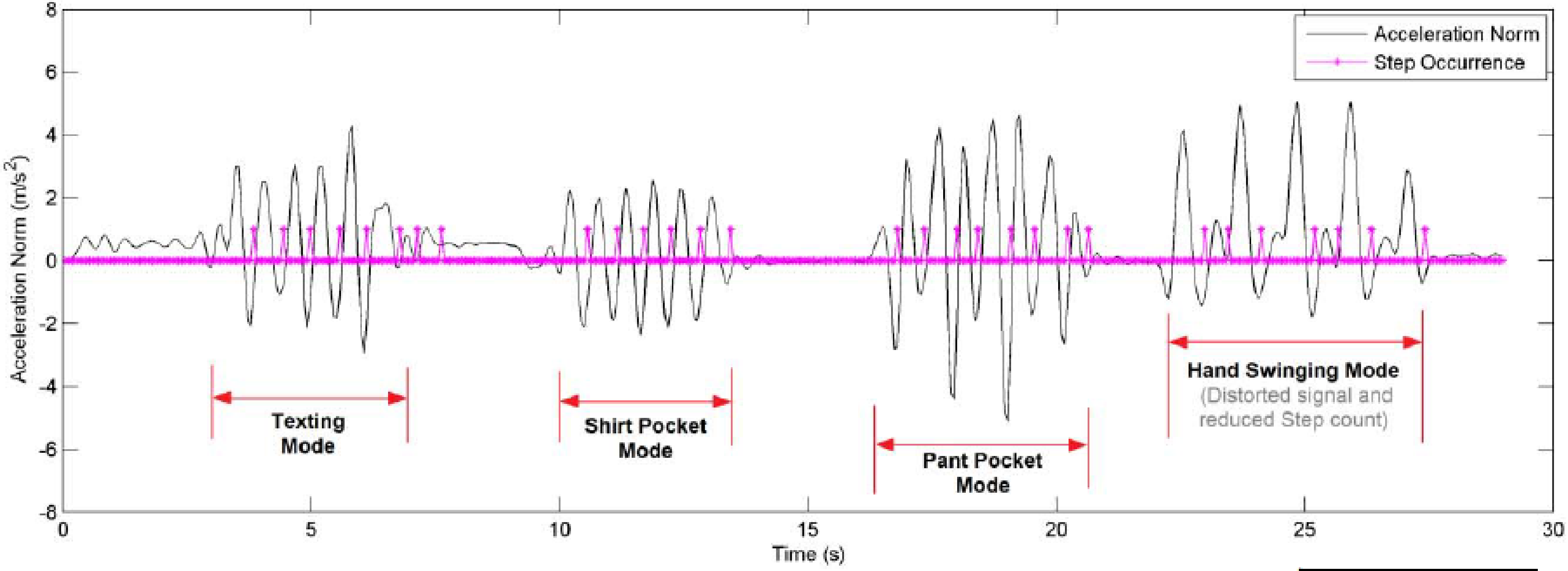




*Figure 58. Output Response vs. Orientation to Gravity*

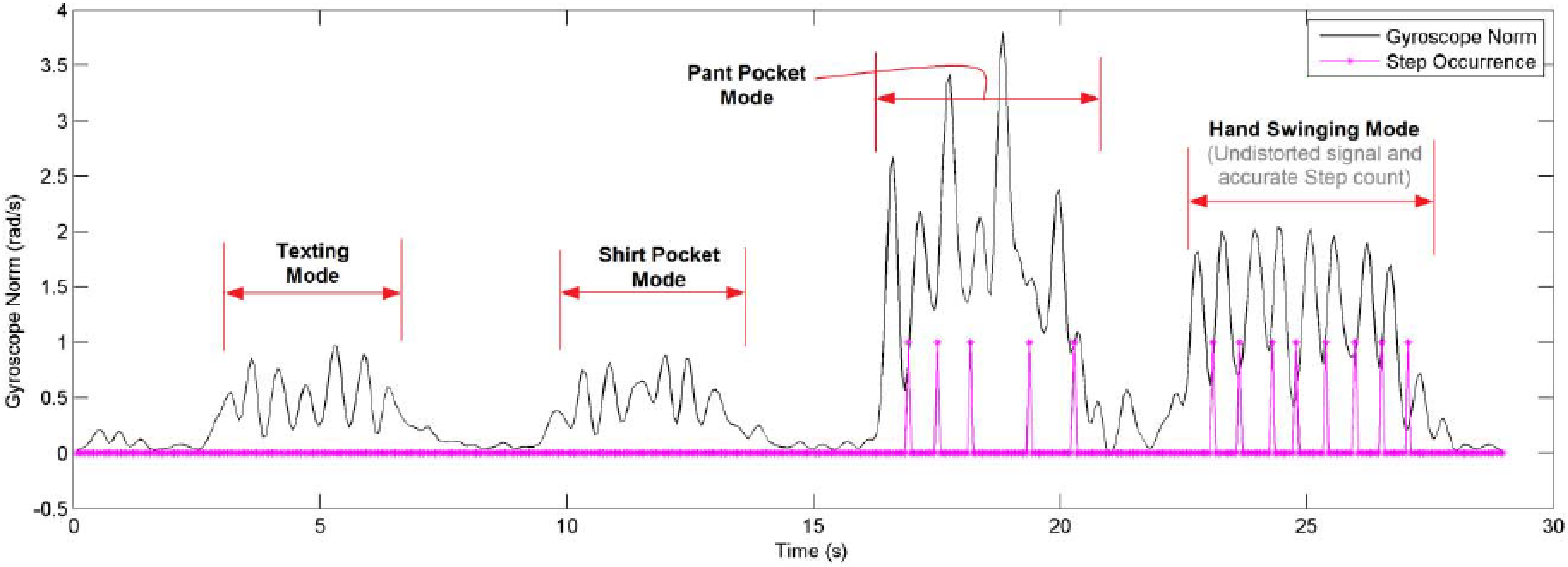
# Anatomy of a Step – Acceleration

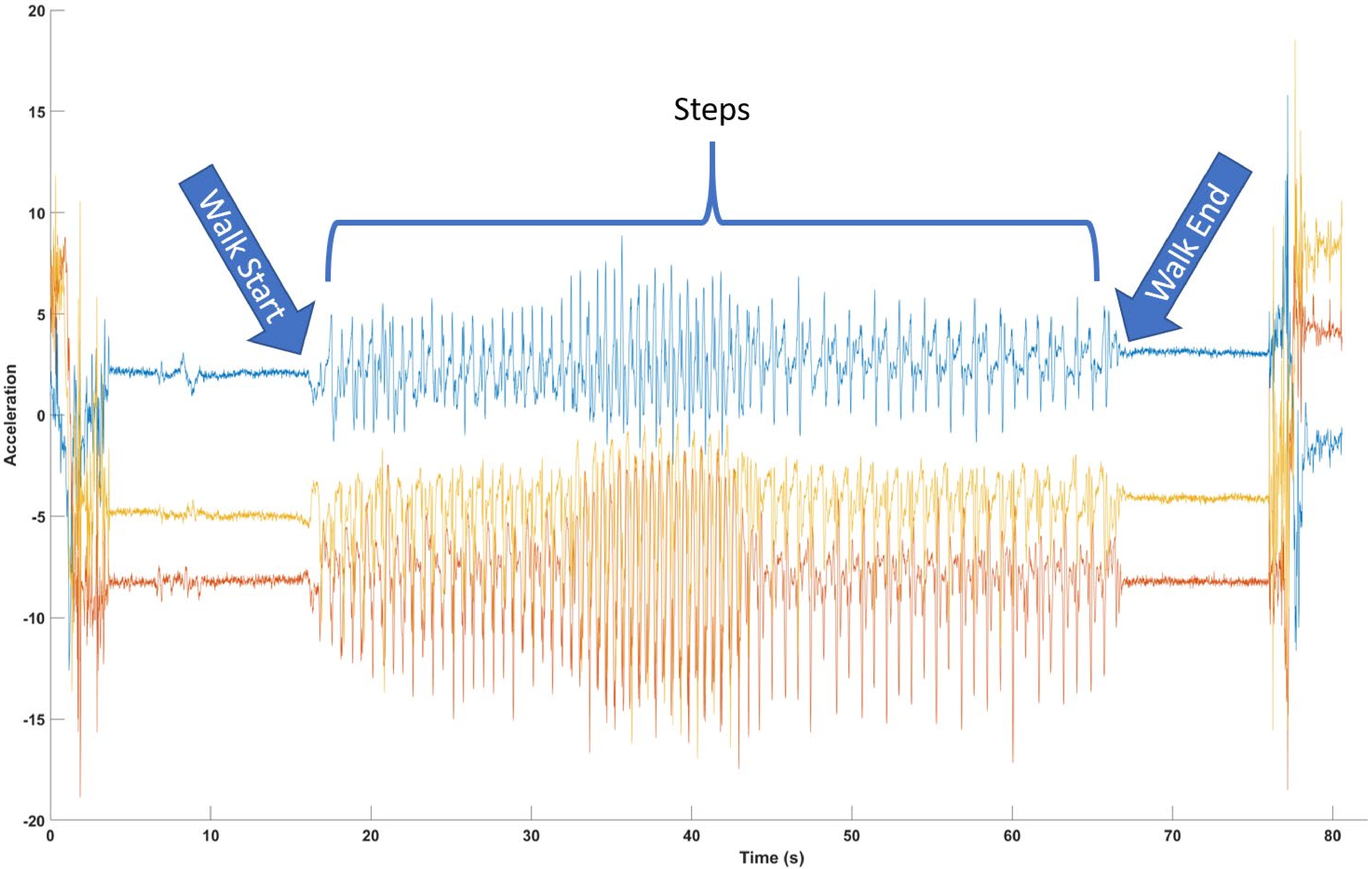
SIDDANAHALLI NINGE GOWDA *et al.*: UMOISP: USAGE MODE AND ORIENTATION INVARIANT SMARTPHONE PEDOMETER 873




$$G_{NORM} = \sqrt{G_x^2 + G_y^2 + G_z^2}$$

# Anatomy of a Step - Gyroscope





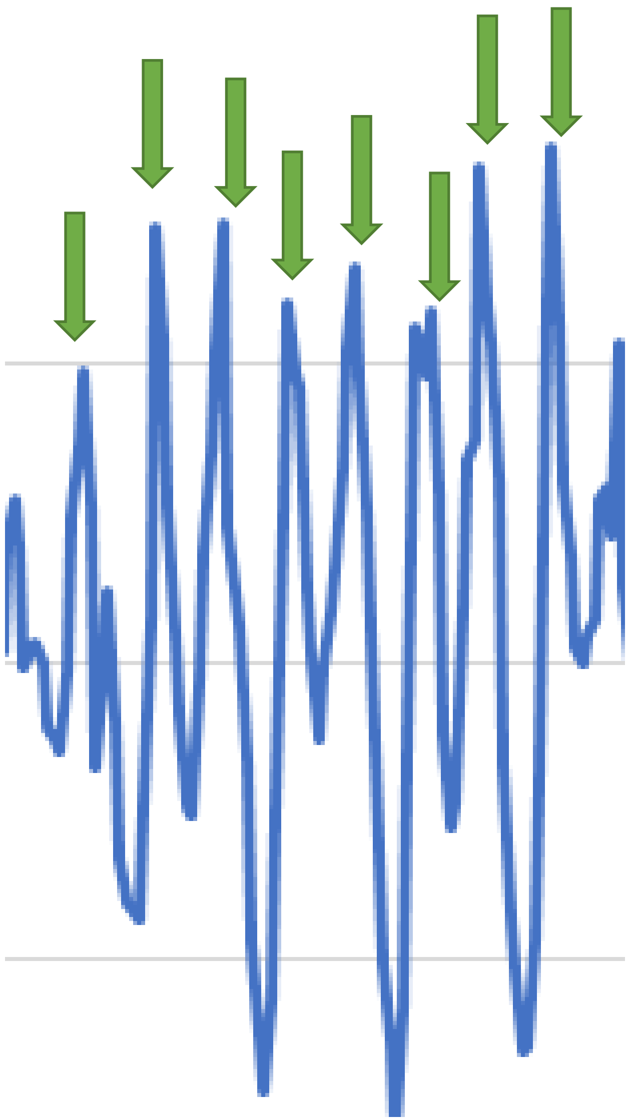
Steps
Walk Start
Walk End
Acceleration
Time (s)
20
15
10
5
0
-5
-10
-15
-20
0
10
20
30
40
50
60
70
80

# Looking for steps

- Acceleration data is very noisy due to random motions
- Walking/steps are very periodic but can vary based upon pace and terrain
  - Consider the impact of stairs; hiking…etc.
- Want to isolate individual steps from noise; find a rhythmic pattern
  - Look for peaks (apex of stride) and/or zero crossings (swinging arm)
- Significantly dependent upon sensing location (hand, waist, foot…etc)

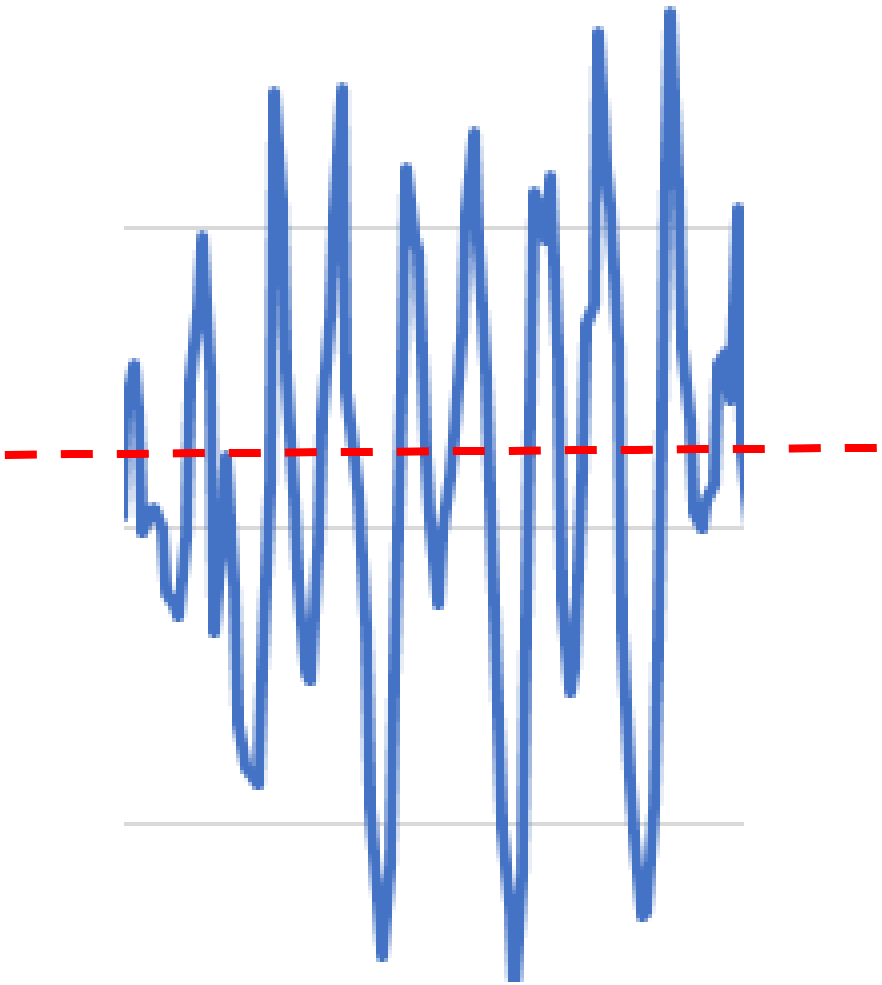
# Counting Steps via Peaks

- How many maxima are detected?
- Would want to pair maxima with minima to ensure full stride
- Peaks can only occur so quickly
  - There’s a maximum walking pace
  - Help reduce/remove noise



# Counting Steps via Zero Crossing

- How many times does the signal cross a “zero” threshold?
- Accel will be at 1g when idle
- Need to detect rising and falling across threshold
  - Should occur relatively close in time



# Stride Frequency

RESEARCH ARTICLE Journal of Experimental Biology (2016) 219, 851-858 doi:10.1242/jeb.133488

**Table 2. Average stride lengths (SL) and stride frequencies (SF)**

| | Straight | | CCW | | CW | |
|---|---|---|---|---|---|---|
| | SL (m) | SF (Hz) | SL (m) | SF (Hz) | SL (m) | SF (Hz) |
| Non-amputees | 3.73±0.23 | 2.16±0.10 | 3.52±0.19* | 2.10±0.12* | 3.48±0.17* | 2.08±0.10* |
| Right leg amputation | 3.86±0.39 | 2.08±0.09 | 3.67±0.23* | **2.07±0.08** | 3.62±0.19* | **1.96±0.08*** |
| Left leg amputation | 3.47±0.20 | 2.06±0.16 | 3.41±0.16 | 1.94±0.14* | 3.33±0.20* | 2.00±0.15 |
| | Straight | | AL outside | | AL inside | |
| | SL (m) | SF (Hz) | SL (m) | SF (Hz) | SL (m) | SF (Hz) |
| All amputees | 3.69±0.37 | 2.07±0.12 | 3.52±0.27* | **2.04±0.12** | 3.53±0.20* | **1.95±0.10*** |

Shouldn't expect more than 2-3 per second

# Signal Processing and Detection

- Separate concerns into Signal Processing (filtering, averages…etc.) and Detection (based upon the signal, did a step occur?)

- Sampling rate should be sufficient to capture step waveform
  - At least 50Hz

- Derivative=> remove signal drift, extract energy

- Integrate=> sum signal and help identify peaks

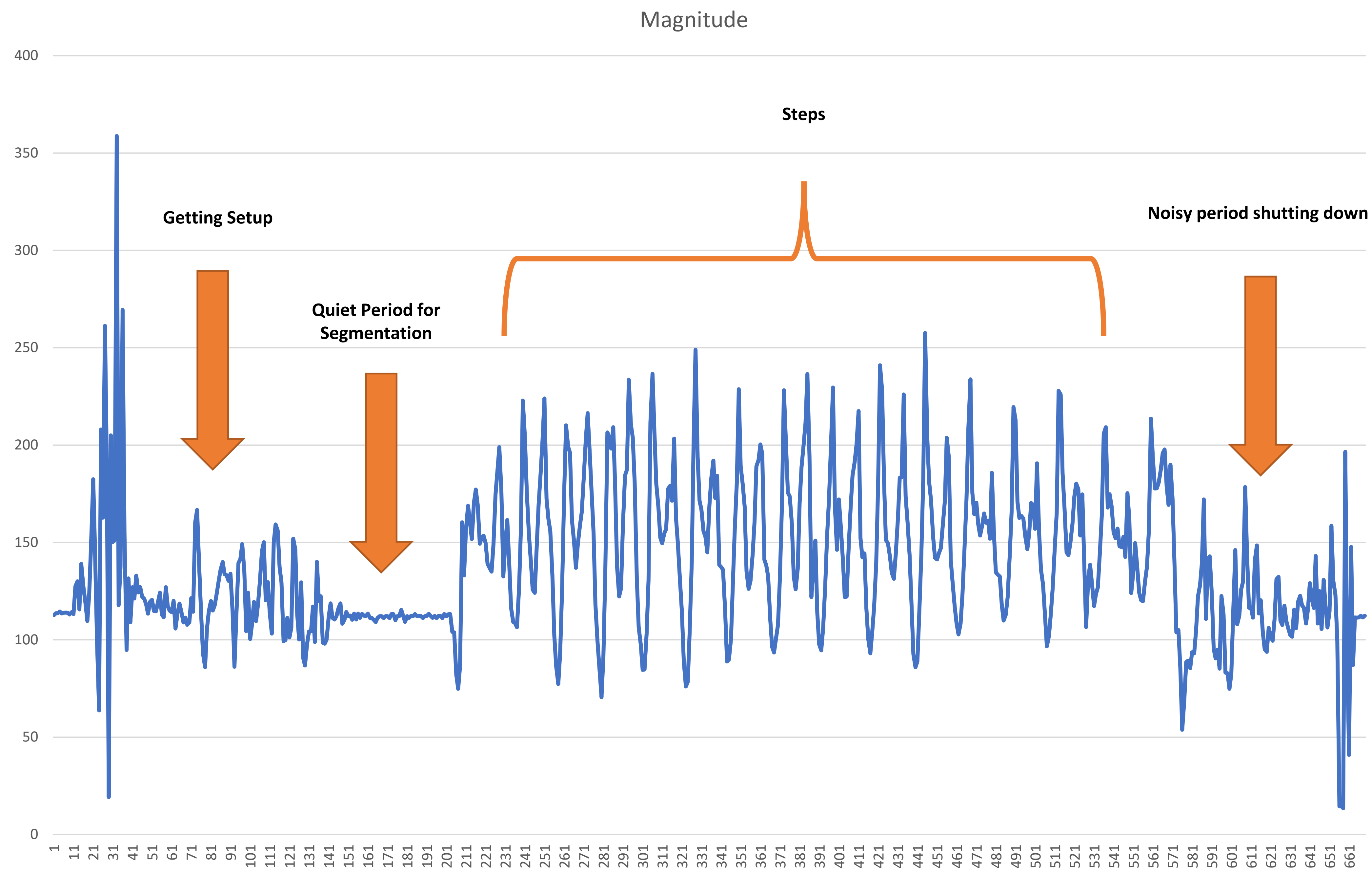
# Pipeline and Detection

# Capturing Acceleration Data

- Simple method is to stream out of Arduino in comma separated value (CSV)
  - Can be imported into Excel or MATLAB

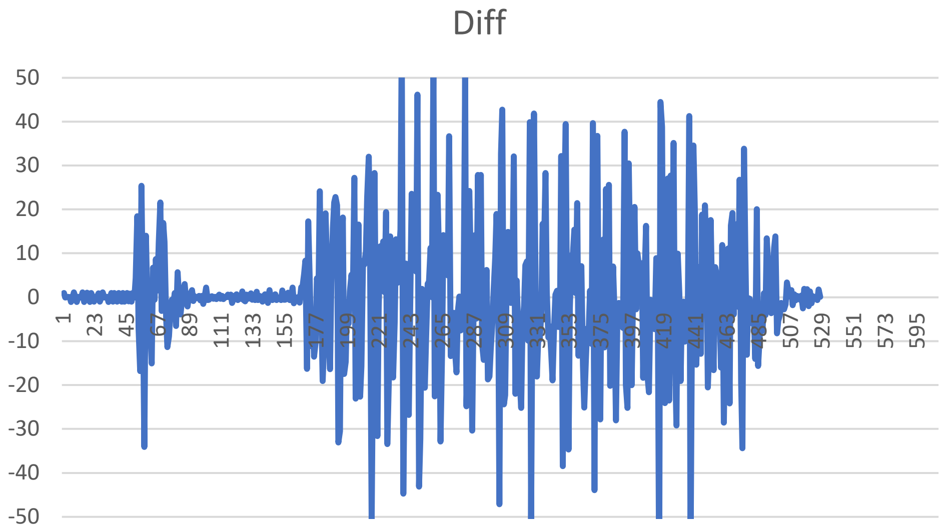
- Leave clear “quiet” areas in the signal to help with segmentation and see when the steps begin/end

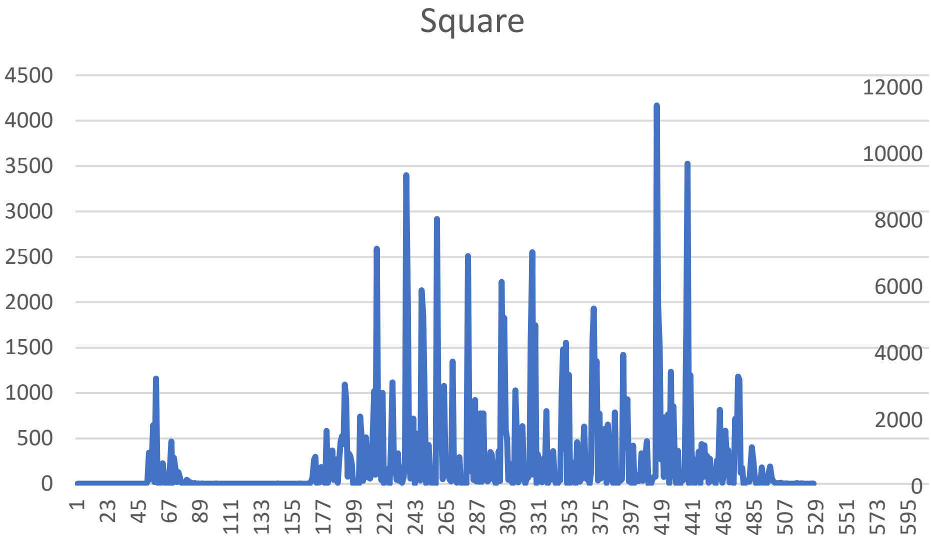
- Capture only the data you need for a particular experiment
  - Really difficult to parse through 5 minutes of data to find a 30s activity

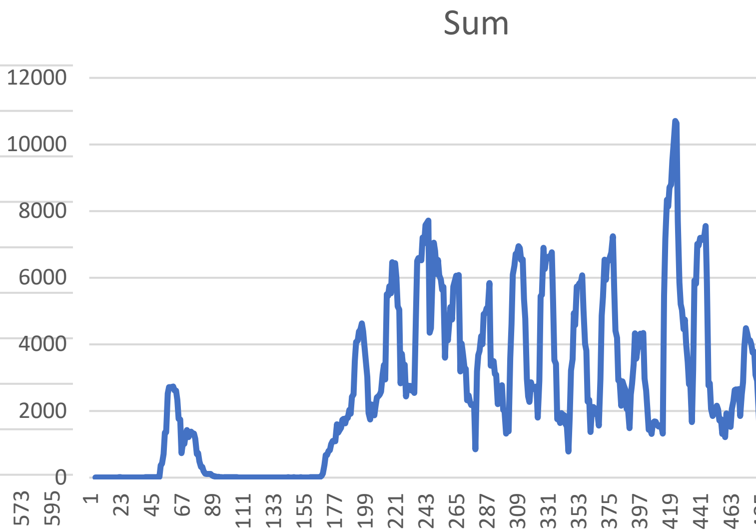

Magnitude
Steps
Getting Setup
Quiet Period for Segmentation
Noisy period shutting down
400
350
300
250
200
150
100
50
0
1
11
21
31
41
51
61
71
81
91
101
111
121
131
141
151
161
171
181
191
201
211
221
231
241
251
261
271
281
291
301
311
321
331
341
351
361
371
381
391
401
411
421
431
441
451
461
471
481
491
501
511
521
531
541
551
561
571
581
591
601
611
621
631
641
651
661

# Some practical tips on filtering

- Work with the magnitude of acceleration to each individual vector
  - Allows step detection regardless of orientation
- Try diff, sum, then square operations
- Sample data sets on website



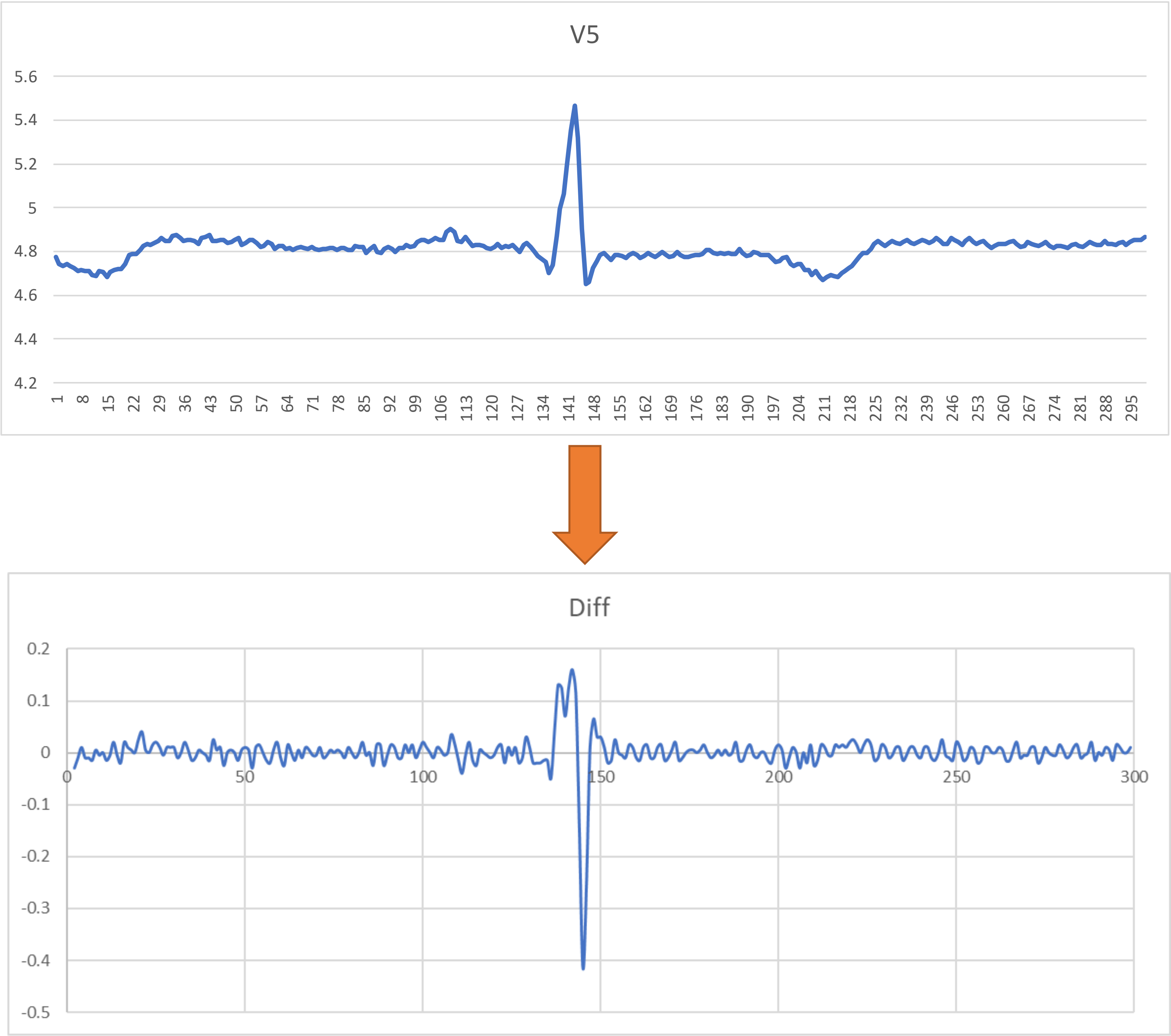



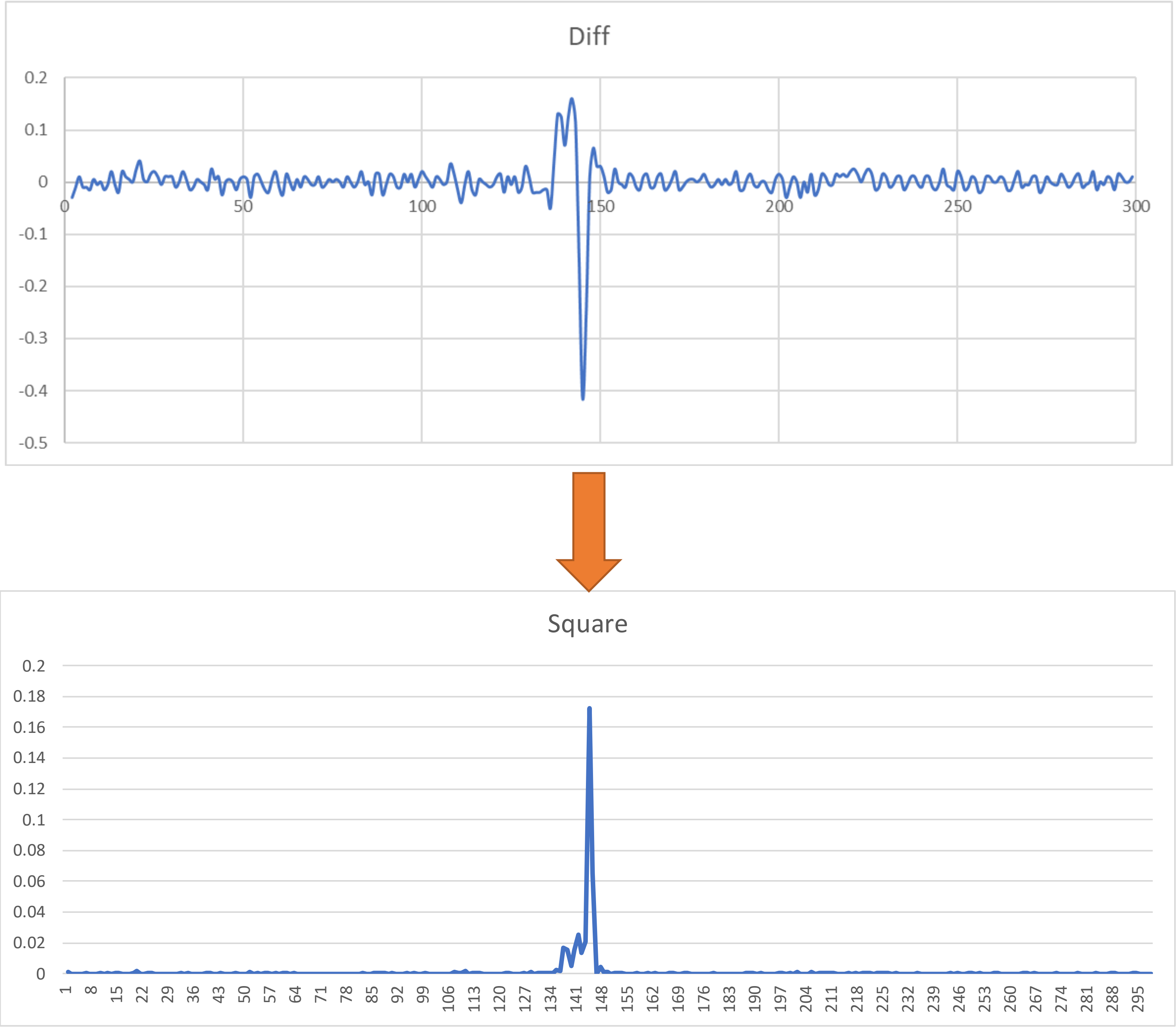
# Pan-Thompkins Algorithm for QRS Detection (but we're using it for foot steps…)

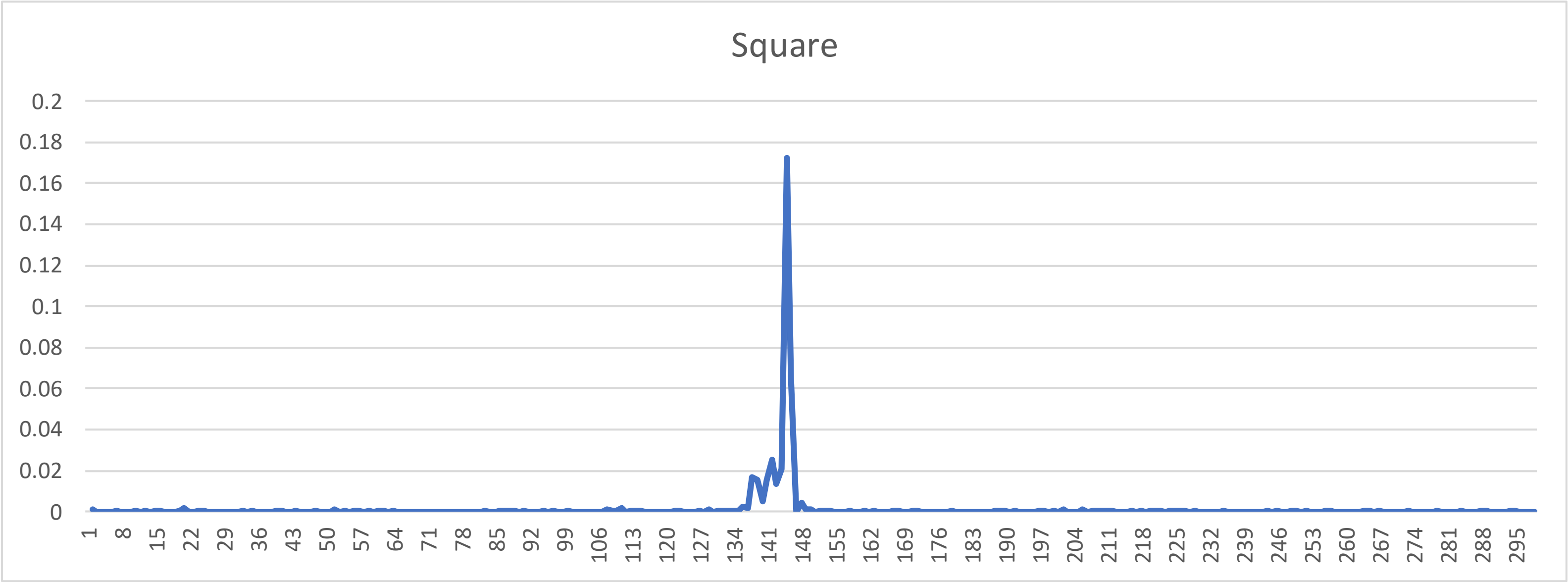
- To detect heart beats, we will follow a standard algorithm to isolate the "energy" from each heart beat. Will create a "pipeline" for processing the data.

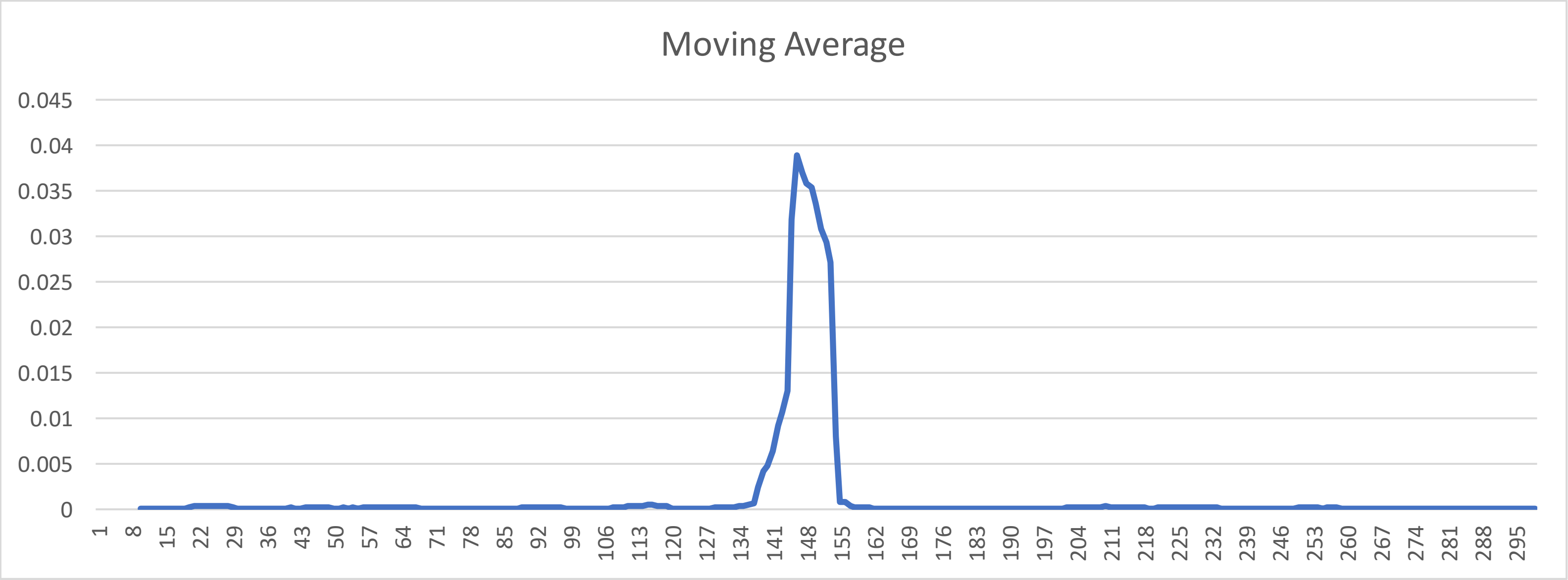
- Each signal is an array/vector of sequential values from [1-n]: $x_1, x_2, \ldots, x_n$. Each operation will result in a new array/vector.

- Three major operations to be performed:
  - Diff: take the difference between sequential points $y_1 = x_1 - x_2$
  - Square: take the result of Diff and perform ^2 operation $z_1 = y_1 * y_1$
  - Moving average: take the result from Diff and average some M values $Q = \sum_1^M z_i$


V5
5.6
5.4
5.2
5
4.8
4.6
4.4
4.2
1 8 15 22 29 36 43 50 57 64 71 78 85 92 99 106 113 120 127 134 141 148 155 162 169 176 183 190 197 204 211 218 225 232 239 246 253 260 267 274 281 288 295
Diff
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
0 50 100 150 200 250 300

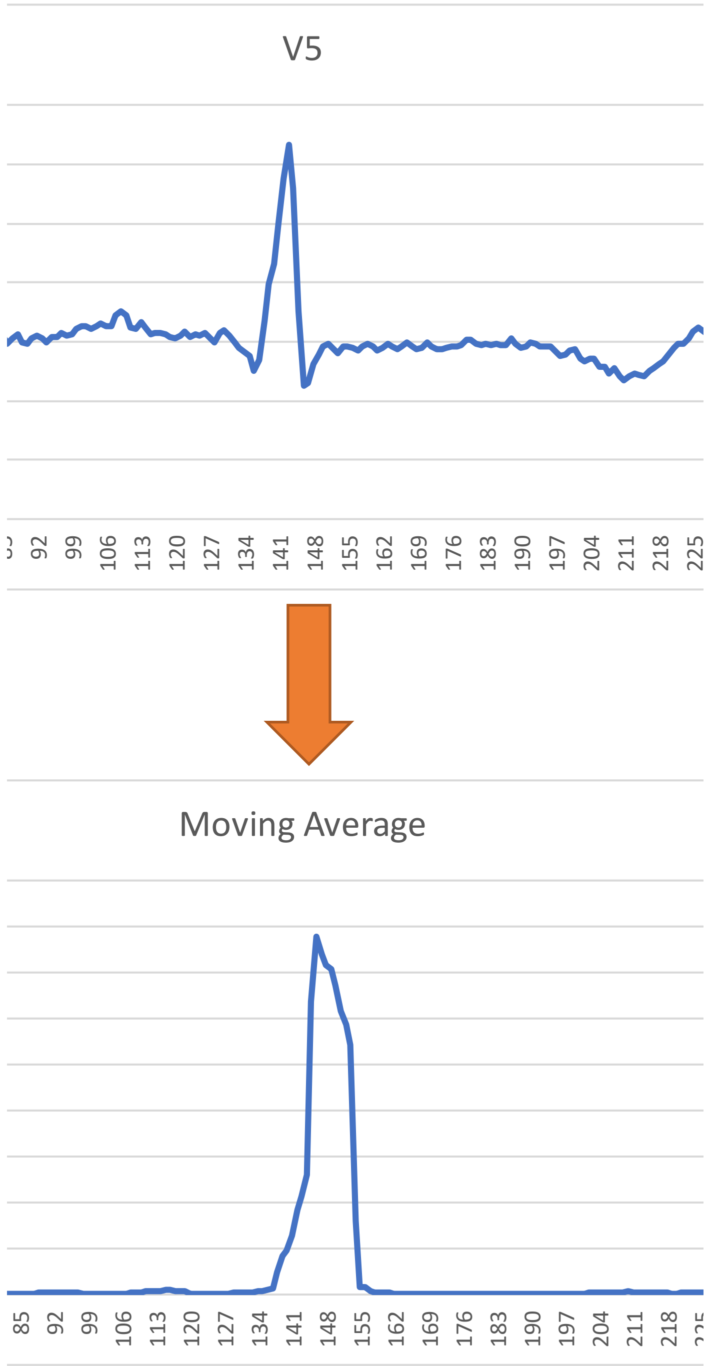

Diff
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
0
50
100
150
200
250
300
Square
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
1
8
15
22
29
36
43
50
57
64
71
78
85
92
99
106
113
120
127
134
141
148
155
162
169
176
183
190
197
204
211
218
225
232
239
246
253
260
267
274
281
288
295


Square
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
1 8 15 22 29 36 43 50 57 64 71 78 85 92 99 106 113 120 127 134 141 148 155 162 169 176 183 190 197 204 211 218 225 232 239 246 253 260 267 274 281 288 295


Moving Average
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
1 8 15 22 29 36 43 50 57 64 71 78 85 92 99 106 113 120 127 134 141 148 155 162 169 176 183 190 197 204 211 218 225 232 239 246 253 260 267 274 281 288 295

# Some important items to notice

- The "peak" is now more prominent, relative to the signal "floor", so it is easier to "see".
- Signal is generally "0" when there is no peak nearby. This makes detection easier.
- Minor noise in signal is removed. However, signal is time delayed due to processing (average filter)



# Practice in Excel then do it in C++

# Signal Analysis in Excel

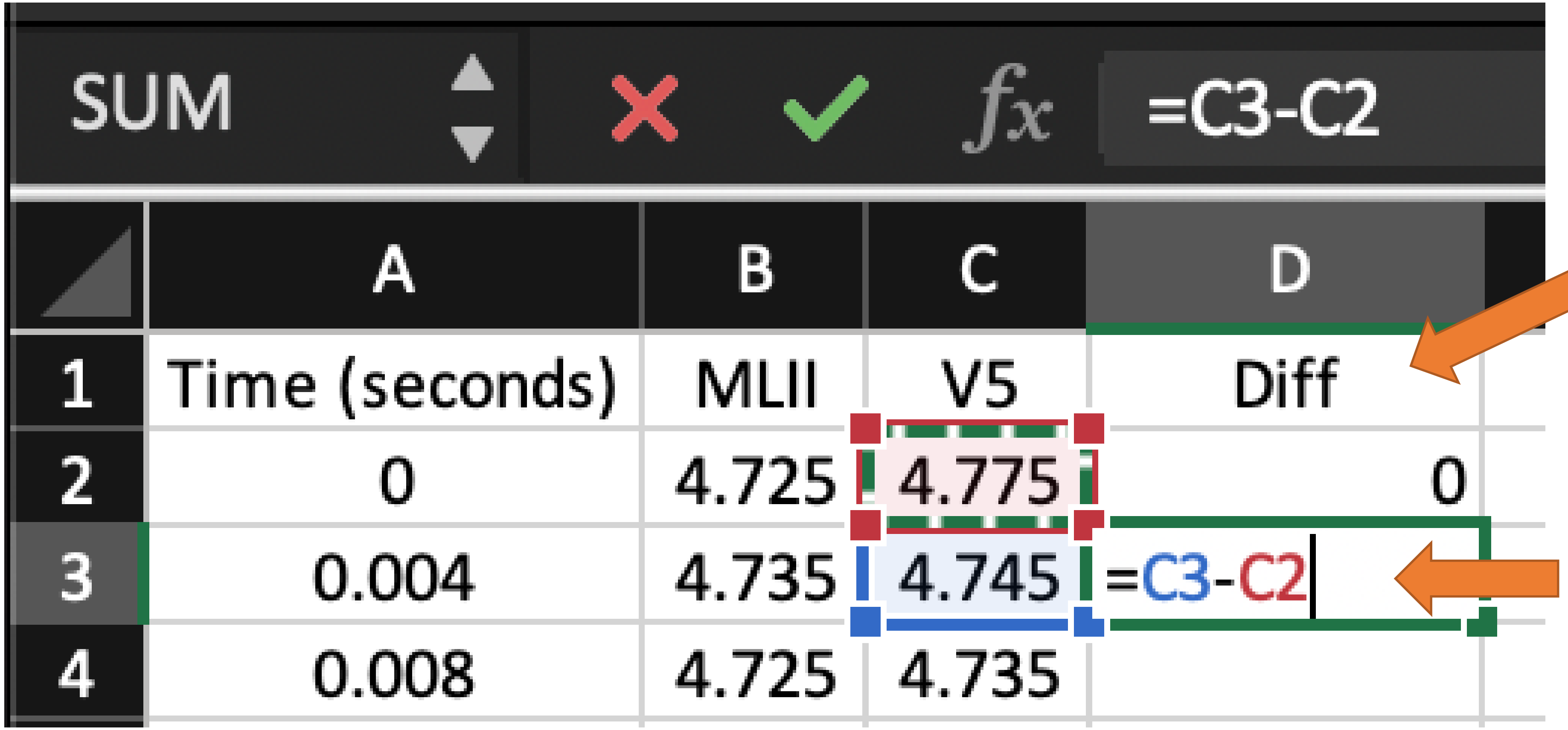
- Leave the original data alone and process each “signal” as a new column. Use the basic +/* operators in Excel and the average() function.

- Write the equation once and then copy it down the excel column. This will propagate the equation into each cell.

- Some initial values may be unknown or deleted. Can’t have a result from a 10 data point average after the second result.

| Time (seconds) | MLII | V5 | **Diff** | **Square** | **Moving Average** |
|---|---|---|---|---|---|
| 0 | 4.725 | 4.775 | | | |
| 0.004 | 4.735 | 4.745 | -0.03 | 0.0009 | |
| 0.008 | 4.725 | 4.735 | -0.01 | 1E-04 | |
| 0.012 | 4.715 | 4.745 | 0.01 | 1E-04 | |
| 0.016 | 4.72 | 4.735 | -0.01 | 1E-04 | |
| 0.02 | 4.705 | 4.725 | -0.01 | 0.0001 | |
| 0.024 | 4.7 | 4.71 | -0.015 | 0.000225 | |
| 0.028 | 4.69 | 4.715 | 0.005 | 2.5E-05 | |
| 0.032 | 4.71 | 4.71 | -0.005 | 2.5E-05 | |
| 0.036 | 4.71 | 4.71 | 0 | 0 | 0.000175 |
| 0.04 | 4.695 | 4.695 | -0.015 | 0.000225 | 1E-04 |
| 0.044 | 4.69 | 4.69 | -0.005 | 2.5E-05 | 9.16667E-05 |
| 0.048 | 4.71 | 4.71 | 0.02 | 0.0004 | 0.000125 |
| 0.052 | 4.695 | 4.705 | -0.005 | 2.5E-05 | 0.000116667 |
| 0.056 | 4.705 | 4.685 | -0.02 | 0.0004 | 0.00015 |

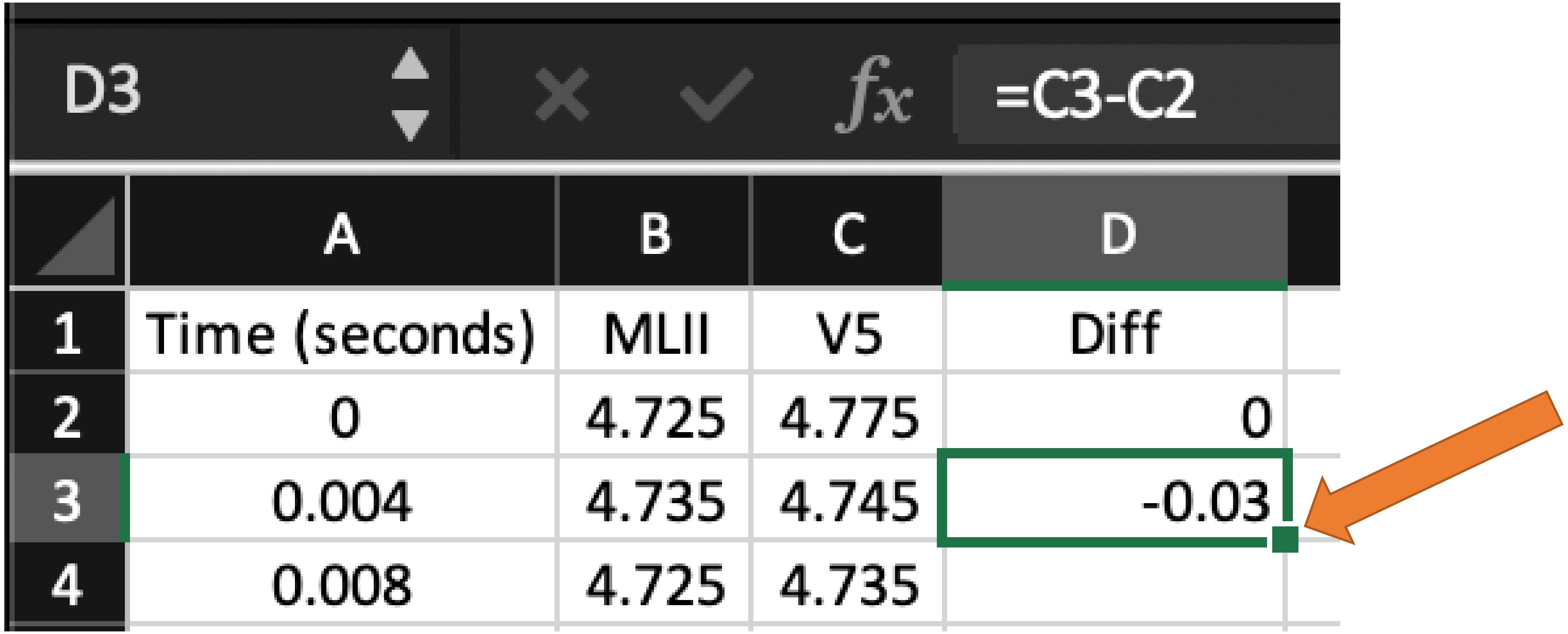
# Creating the First Diff Entry




**1. Give the column a name. Fill in the first value as 0**

**2. In the cell, type "=" and then click the cell in the V5 column. Type "-". Then click the next cell. Press enter when done.**

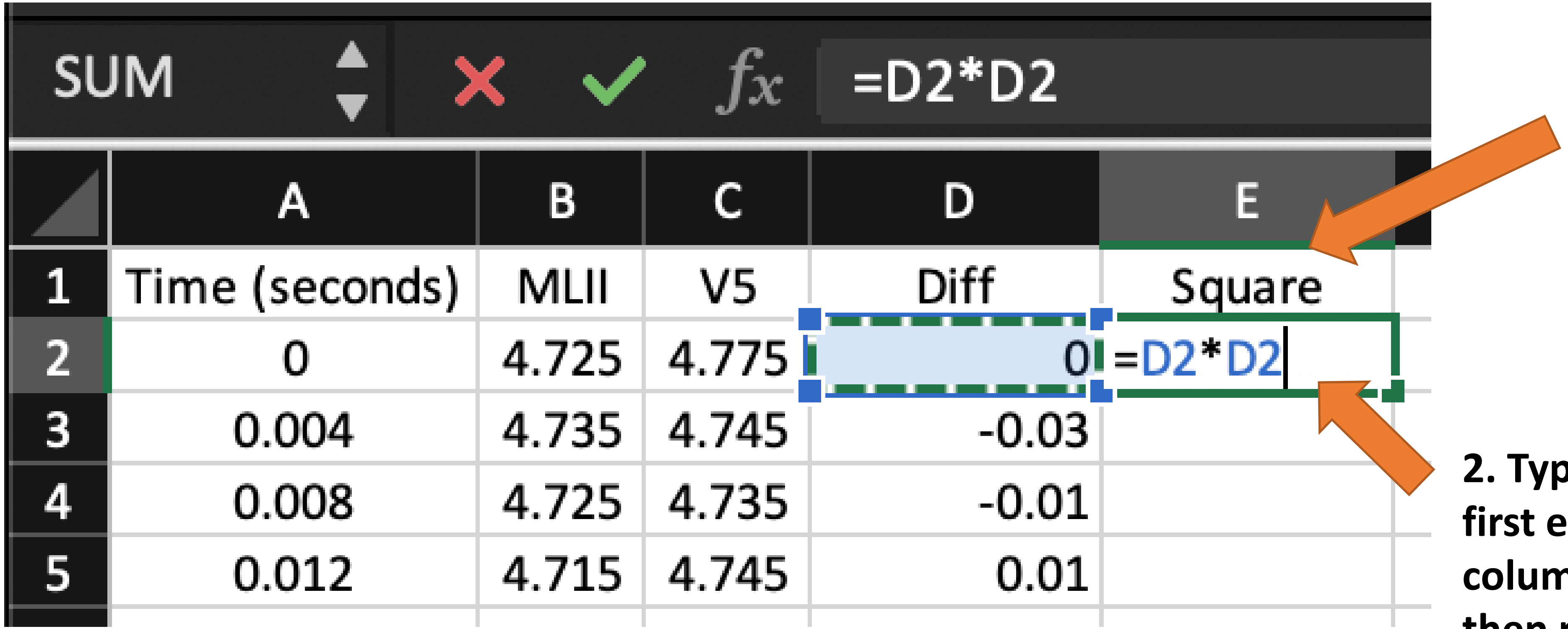
# When Done, double click the bottom right of cell to copy down the entire column.



D3 =C3-C2

| | A | B | C | D |
|---|---|---|---|---|
| 1 | Time (seconds) | MLII | V5 | Diff |
| 2 | 0 | 4.725 | 4.775 | 0 |
| 3 | 0.004 | 4.735 | 4.745 | -0.03 |
| 4 | 0.008 | 4.725 | 4.735 | |

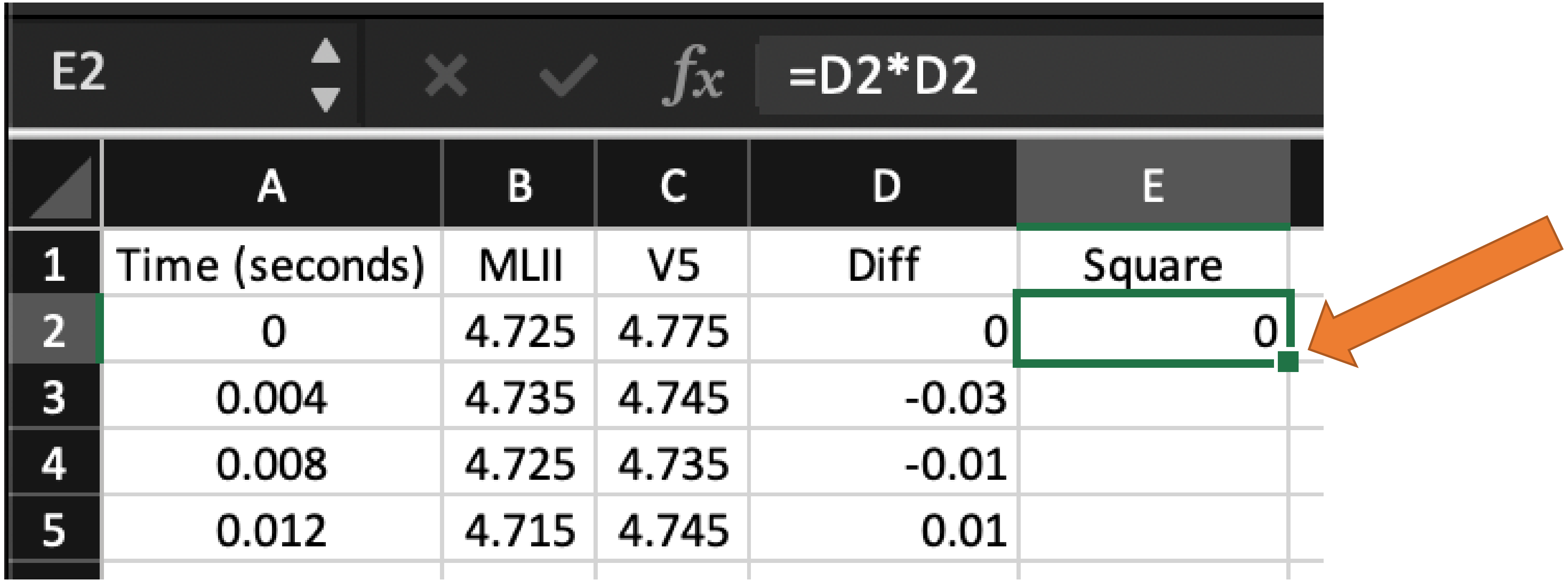
# Creating the Square Column




**1. Give the column a name.**

**2. Type "=". Click the first entry in the Diff column. Type "*" and then re-select the entry again. Press enter**

# When Done, double click the bottom right of cell to copy down the entire column.


E2 =D2*D2

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Time (seconds) | MLII | V5 | Diff | Square |
| 2 | 0 | 4.725 | 4.775 | 0 | 0 |
| 3 | 0.004 | 4.735 | 4.745 | -0.03 | |
| 4 | 0.008 | 4.725 | 4.735 | -0.01 | |
| 5 | 0.012 | 4.715 | 4.745 | 0.01 | |

# Make a Moving Average of 10pts. Can be adjusted.

SUM ✕ ✓ fx =average(E2:E12)

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Time (seconds) | MLII | V5 | Diff | Square | Moving Avg |
| 2 | 0 | 4.725 | 4.775 | 0 | 0 | |
| 3 | 0.004 | 4.735 | 4.745 | -0.03 | 0.0009 | |
| 4 | 0.008 | 4.725 | 4.735 | -0.01 | 1E-04 | |
| 5 | 0.012 | 4.715 | 4.745 | 0.01 | 1E-04 | |
| 6 | 0.016 | 4.72 | 4.735 | -0.01 | 1E-04 | |
| 7 | 0.02 | 4.705 | 4.725 | -0.01 | 0.0001 | |
| 8 | 0.024 | 4.7 | 4.71 | -0.015 | 0.000225 | |
| 9 | 0.028 | 4.69 | 4.715 | 0.005 | 2.5E-05 | |
| 10 | 0.032 | 4.71 | 4.71 | -0.005 | 2.5E-05 | |
| 11 | 0.036 | 4.71 | 4.71 | 0 | 0 | |
| 12 | 0.04 | 4.695 | 4.695 | -0.015 | 0.000225 | =average(E2:E12) |
| 13 | 0.044 | 4.69 | 4.69 | -0.005 | 2.5E-05 | |
| 14 | 0.048 | 4.71 | 4.71 | 0.02 | 0.0004 | |

**This time, when typing in the formula use the built-in average function.**

**Type "=average(" then use the cursor to select a range of data (about 10 pts). Then type ")" to close the formula. Press enter.**

**Expand the formula down the column.**

You should have something like this…

M24

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | Time (seconds) | MLII | V5 | Diff | Square | Moving Avg |
| 2 | 0 | 4.725 | 4.775 | 0 | 0 | |
| 3 | 0.004 | 4.735 | 4.745 | -0.03 | 0.0009 | |
| 4 | 0.008 | 4.725 | 4.735 | -0.01 | 1E-04 | |
| 5 | 0.012 | 4.715 | 4.745 | 0.01 | 1E-04 | |
| 6 | 0.016 | 4.72 | 4.735 | -0.01 | 1E-04 | |
| 7 | 0.02 | 4.705 | 4.725 | -0.01 | 0.0001 | |
| 8 | 0.024 | 4.7 | 4.71 | -0.015 | 0.000225 | |
| 9 | 0.028 | 4.69 | 4.715 | 0.005 | 2.5E-05 | |
| 10 | 0.032 | 4.71 | 4.71 | -0.005 | 2.5E-05 | |
| 11 | 0.036 | 4.71 | 4.71 | 0 | 0 | |
| 12 | 0.04 | 4.695 | 4.695 | -0.015 | [illegible] | 0.00016364 |
| 13 | 0.044 | 4.69 | 4.69 | -0.005 | 2.5E-05 | 0.00016591 |
| 14 | 0.048 | 4.71 | 4.71 | 0.02 | 0.0004 | 0.00012045 |
| 15 | 0.052 | 4.695 | 4.705 | -0.005 | 2.5E-05 | 0.00011364 |
| 16 | 0.056 | 4.705 | 4.685 | -0.02 | 0.0004 | 0.00014091 |
| 17 | 0.06 | 4.7 | 4.705 | 0.02 | 0.0004 | 0.00016818 |
| 18 | 0.064 | 4.71 | 4.715 | 0.01 | 1E-04 | 0.00016818 |
| 19 | 0.068 | 4.71 | 4.72 | 0.005 | 2.5E-05 | 0.00015 |
| 20 | 0.072 | 4.7 | 4.72 | 0 | 0 | 0.00014773 |
| 21 | 0.076 | 4.705 | 4.745 | 0.025 | 0.000625 | 0.00020227 |
| 22 | 0.08 | 4.72 | 4.785 | 0.04 | 0.0016 | 0.00034773 |
| 23 | 0.084 | 4.735 | 4.79 | 0.005 | 2.5E-05 | 0.00032955 |
| 24 | 0.088 | 4.74 | 4.79 | 0 | 0 | 0.00032727 |
| 25 | 0.092 | 4.73 | 4.805 | 0.015 | 0.000225 | 0.00031136 |

# Some practical tips on detection

- Identify peaks or zero crossings and then subsequent event soon after
- If corresponding event did not occur, then likely was a false alarm
- Don’t set static threshold; should float with your data

# A wild assignment appears…

| Participant | Phone Location | Steps Taken | Walking Begins (index) | Walking Ends (index) | Start (s) | End (s) |
|---|---|---|---|---|---|---|
| p1.1 | Handheld | 70 | 1607 | 5393 | 16.07 | 53.93 |
| p1.4 | Purse | 70 | 2110 | 5963 | 21.1 | 59.63 |
| p2.2 | Handheld | 66 | 1469 | 5930 | 14.69 | 59.3 |
| p9.2 | Pants pocket | 70 | 4774 | 8525 | 47.74 | 85.25 |
| p11.3 | Backpack | 76 | 1603 | 6709 | 16.03 | 67.09 |
| p27.1 | Handheld | 68 | 7325 | 11053 | 73.25 | 110.53 |