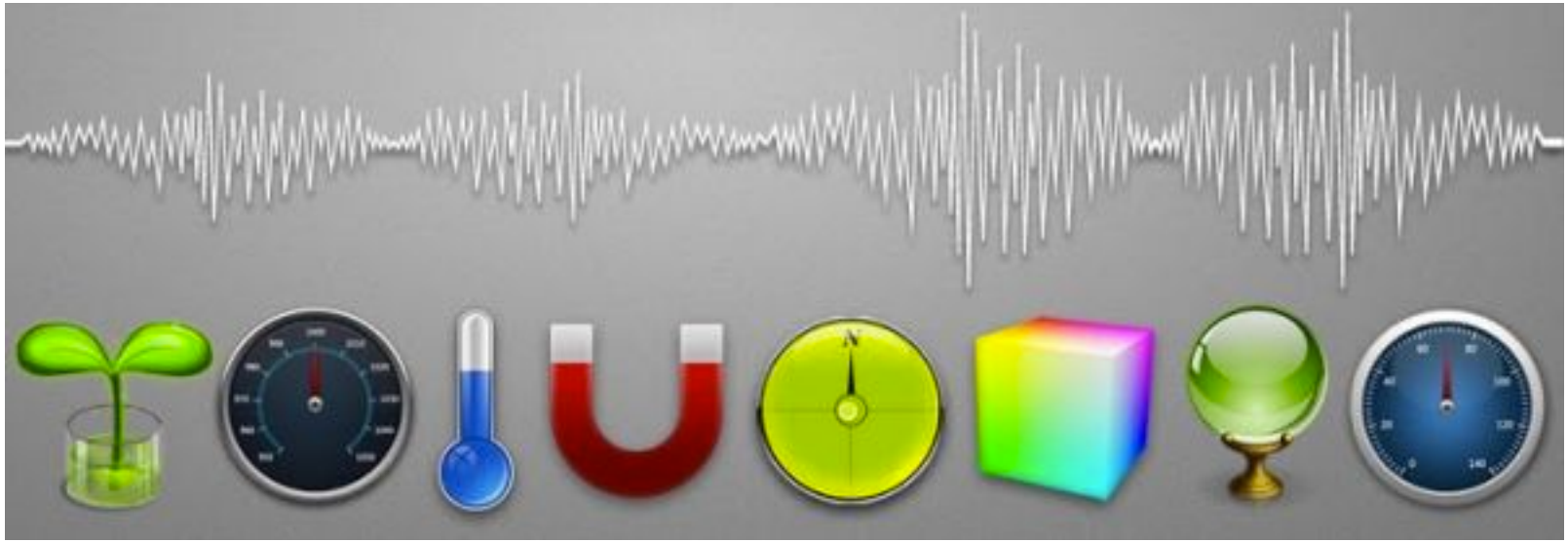
Mobile sensors
CS 407



Various types of sensors

- Accelerometer
- Gyroscope
- Magnetic field
- Light
- Proximity
- Temperature
- Pressure
- Humidity

Accelerometer

- Impact of force applied to the device
  - F = m * a
- Units: m/s^2
- When device is stationary, we get +1g in one axis
- Can give you device orientation

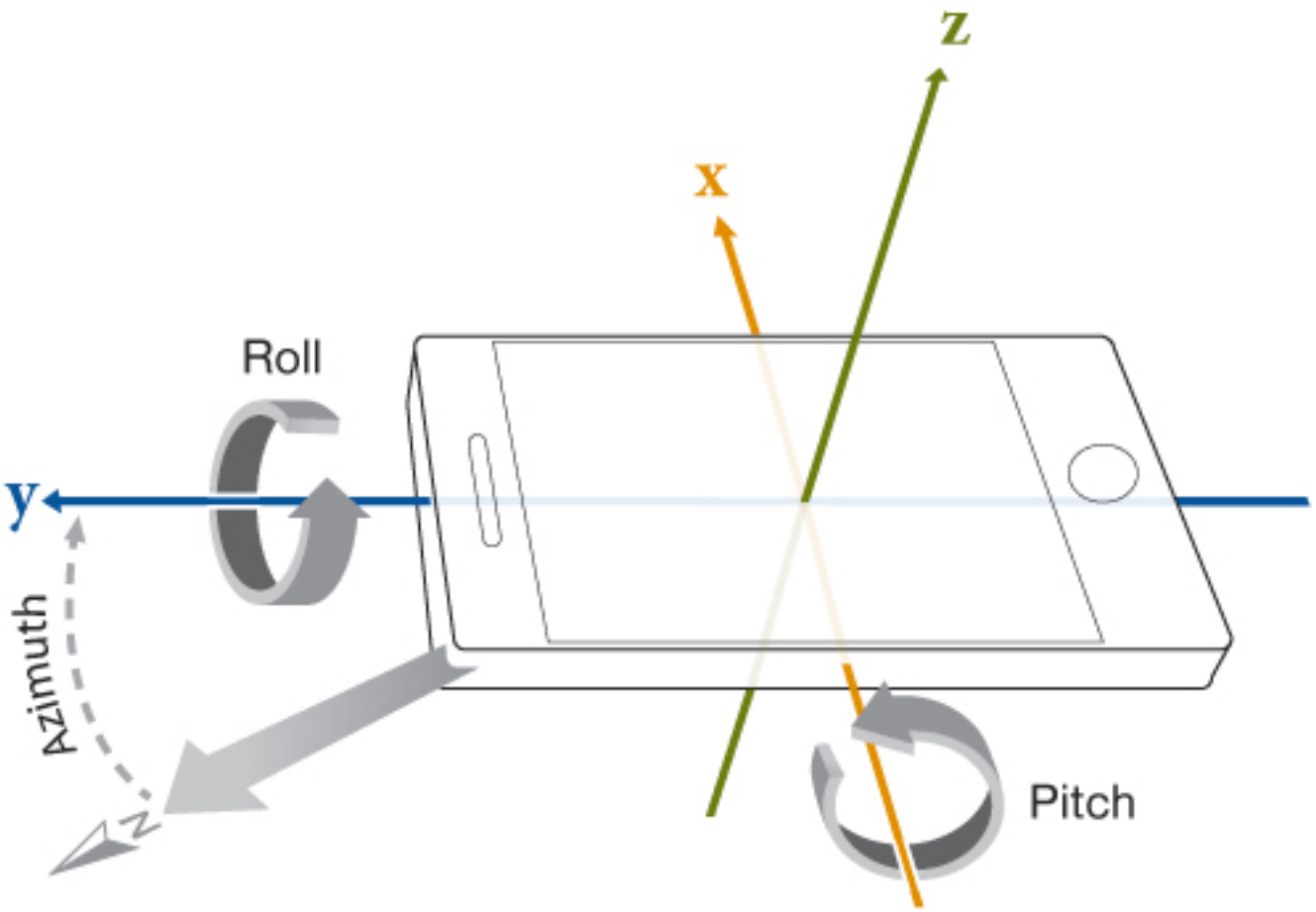
How about displacement?

- u = a * t
- s = u0 * t + ½ a * t^2

- Dead reckoning

- Does this work?
  - Sometimes, but can be inaccurate since a is not always as accurate

Gyroscope



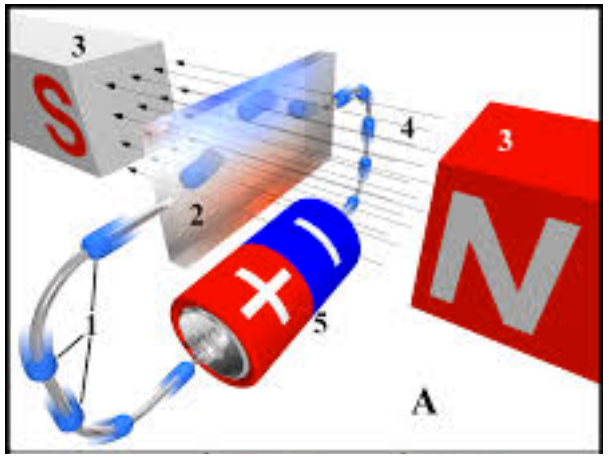
Gyroscope

- Unit: rad/s

Magnetometer

- Hall effect measure of current to evaluate magnetic field

- Unit: Tesla (T) or uT



Light sensor

- Usually located near the front-facing camera
- Unit: Lux

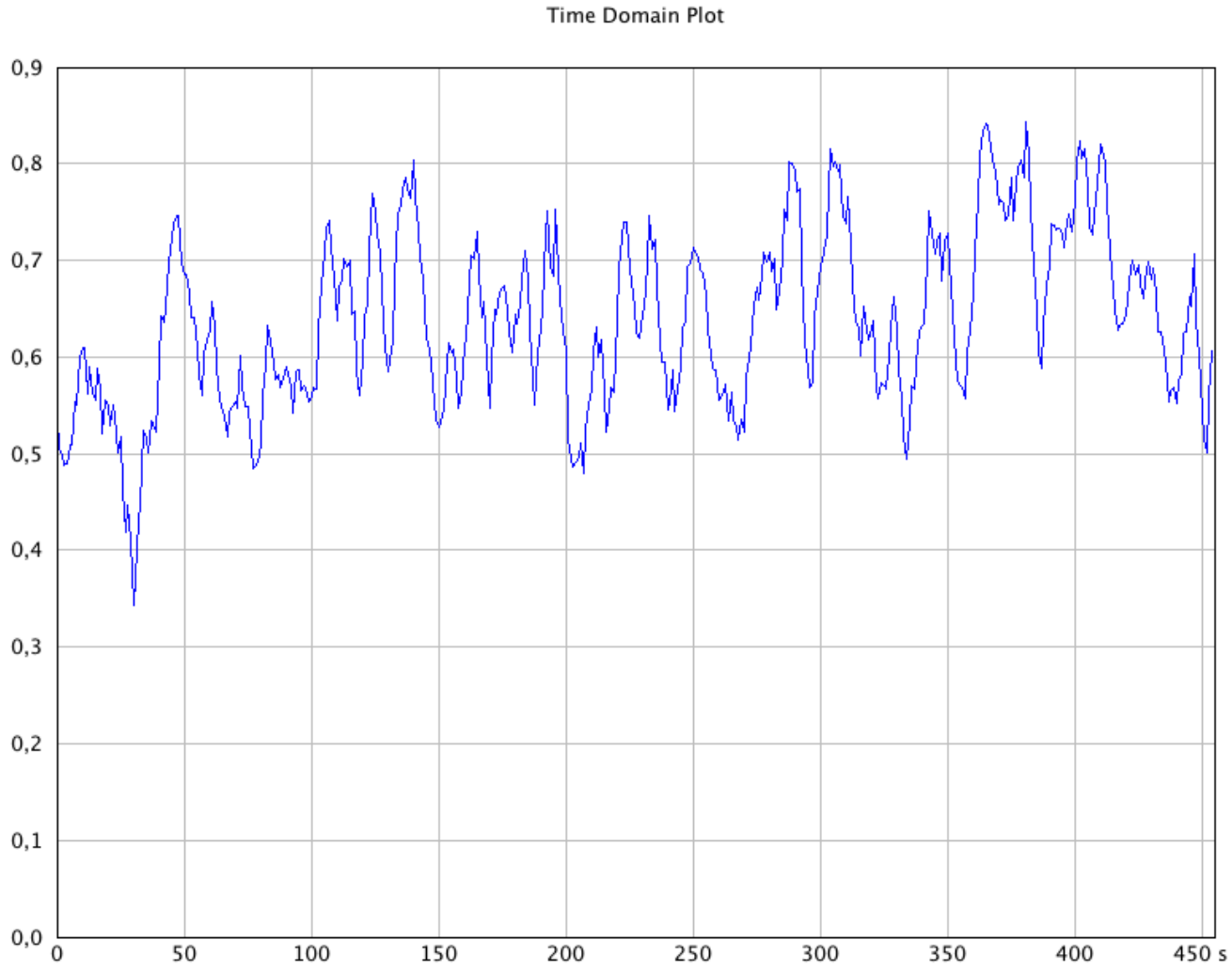
Proximity sensor

- Could measure distance from device in cm
- Could be a binary output
  - Something is close by or not
- Sometimes uses the light sensor to infer this

Sensor framework

- Implemented as a callback
- SensorEventListener()
- Can also indicate how frequent updates are needed

Smoothing of data



Smoothing of data

- Simple moving average

$$s_t = \frac{1}{k}\sum_{n=0}^{k-1} x_{t-n} = \frac{x_t + x_{t-1} + x_{t-2} + \cdots + x_{t-k+1}}{k} = s_{t-1} + \frac{x_t - x_{t-k}}{k},$$

Smoothing of data

- Weighted moving average

$$s_t = \sum_{n=1}^{k} w_n x_{t+1-n} = w_1 x_t + w_2 x_{t-1} + \cdots + w_k x_{t-k+1}.$$

Smoothing data

- Exponentially Weighted Moving Average (EWMA)

$$s_0 = x_0$$
$$s_t = \alpha x_t + (1 - \alpha) s_{t-1},\ t > 0$$

Smoothing data

- Exponentially Weighted Moving Average (EWMA)

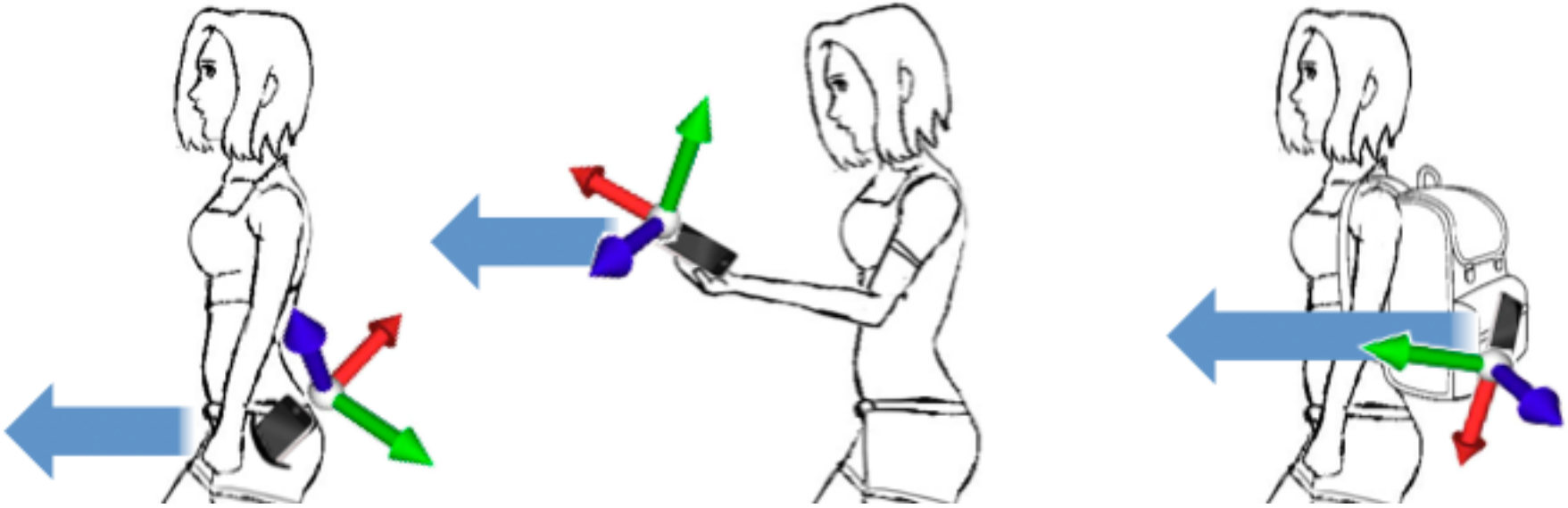
$$s_0 = x_0$$
$$s_t = \alpha x_t + (1-\alpha)s_{t-1},\ t > 0$$

$$\begin{aligned} s_t &= \alpha x_t + (1-\alpha)s_{t-1} \\ &= \alpha x_t + \alpha(1-\alpha)x_{t-1} + (1-\alpha)^2 s_{t-2} \\ &= \alpha\left[x_t + (1-\alpha)x_{t-1} + (1-\alpha)^2 x_{t-2} + (1-\alpha)^3 x_{t-3} + \cdots + (1-\alpha)^{t-1}x_0\right] + (1-\alpha)^t s_0. \end{aligned}$$

Dead reckoning

How to estimate position indoors using sensors?

Orient the phone



Put it Flat: Project Device Coordinates

- World Coordinates $(x_w \quad y_w \quad z_w)$
- Tablet Coordinates $(x_t \quad y_t \quad z_t)$
- Rotation Matrix R = $\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix}$
- $\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{bmatrix} * \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix}$

Speed and Distance Estimation

- Speed estimation: $v\downarrow i+1 = v\downarrow i + a\downarrow i \, t\downarrow i$
- Distance estimation: $s\downarrow i = v\downarrow i \, t\downarrow i + 1/2 \, a\downarrow i \, t\downarrow i\uparrow \uparrow 2$

Smoothing data

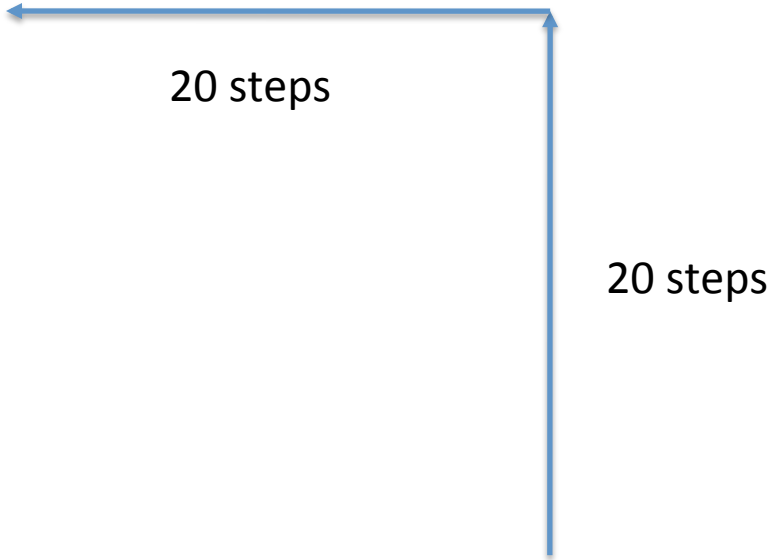
- Exponentially Weighted Moving Average (EWMA)

$$s_0 = x_0$$
$$s_t = \alpha x_t + (1-\alpha)s_{t-1},\ t > 0$$

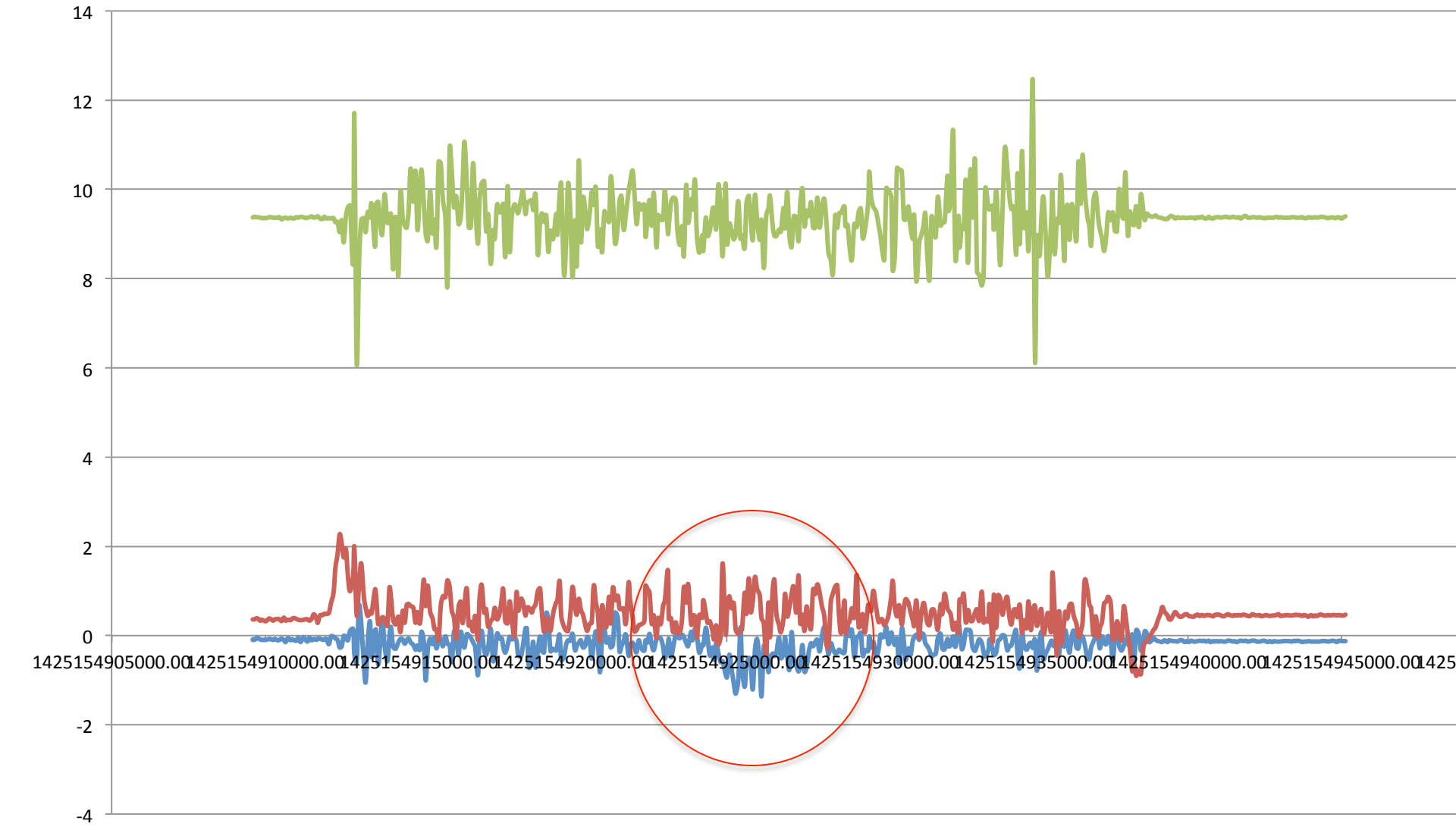
$$\begin{aligned} s_t &= \alpha x_t + (1-\alpha)s_{t-1} \\ &= \alpha x_t + \alpha(1-\alpha)x_{t-1} + (1-\alpha)^2 s_{t-2} \\ &= \alpha\left[x_t + (1-\alpha)x_{t-1} + (1-\alpha)^2 x_{t-2} + (1-\alpha)^3 x_{t-3} + \cdots + (1-\alpha)^{t-1}x_0\right] + (1-\alpha)^t s_0. \end{aligned}$$

Here is a walk pattern

- Android tablet in a cart
- Fixed by tape
- Tablet is flat



Accelerometer

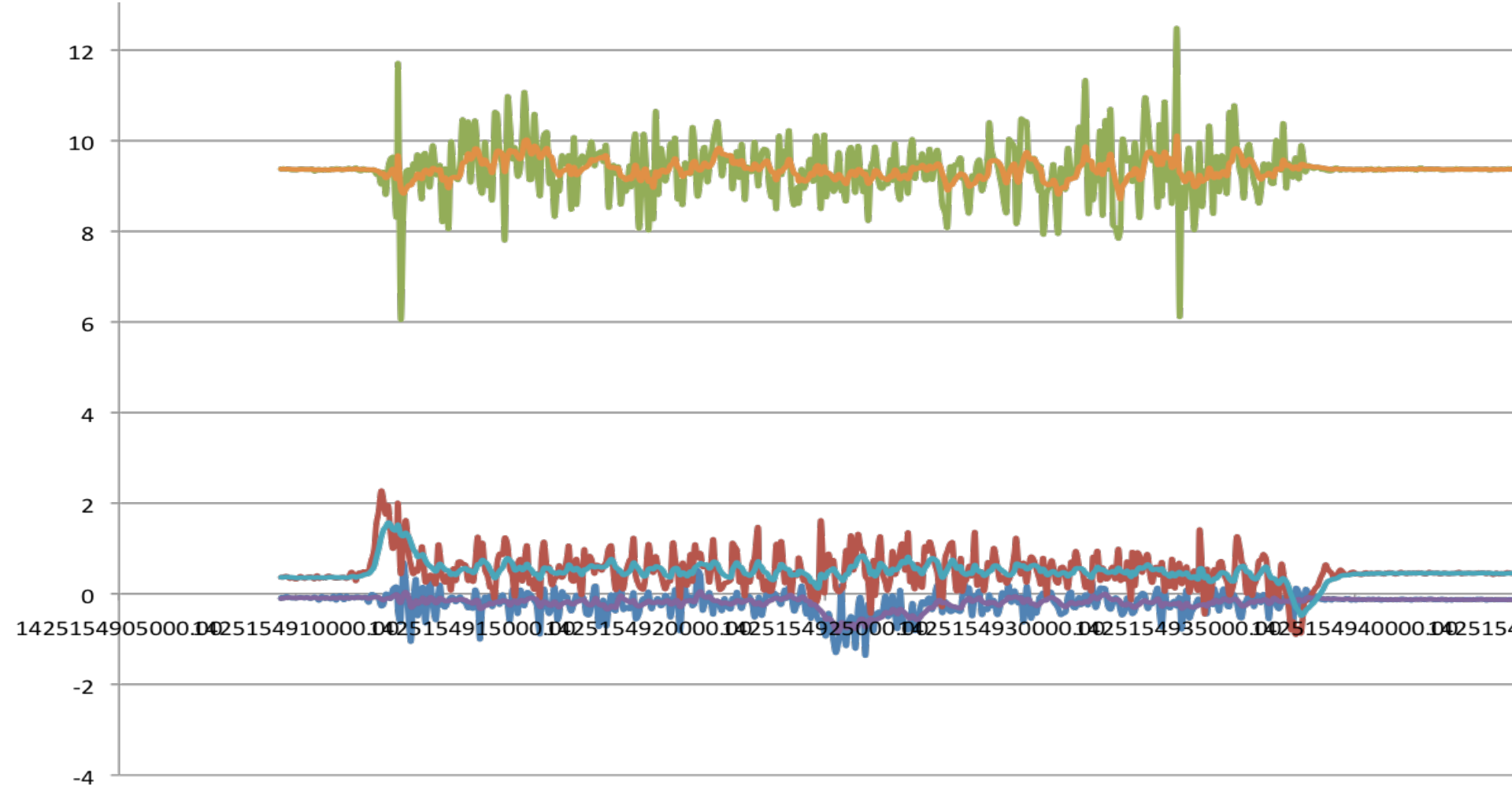


Alpha = 0.8

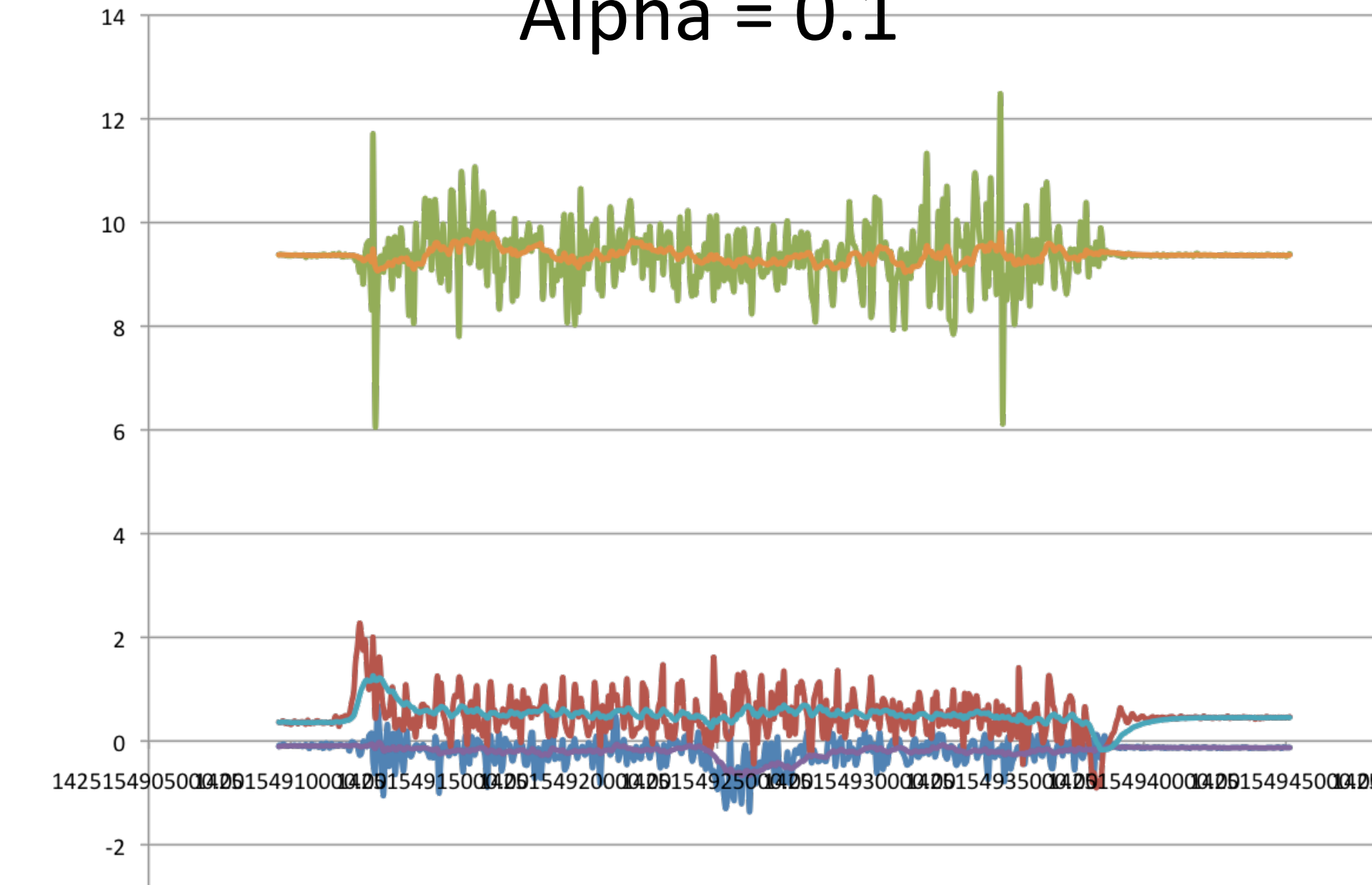
Alpha = 0.5


4
2
0
8
6
4
2
0
2
4
549050000.140251549100000.140251549150000.140251549200000.140251549250000.140251549300000.140251549350000.140251549400000.140251549450000.140251549

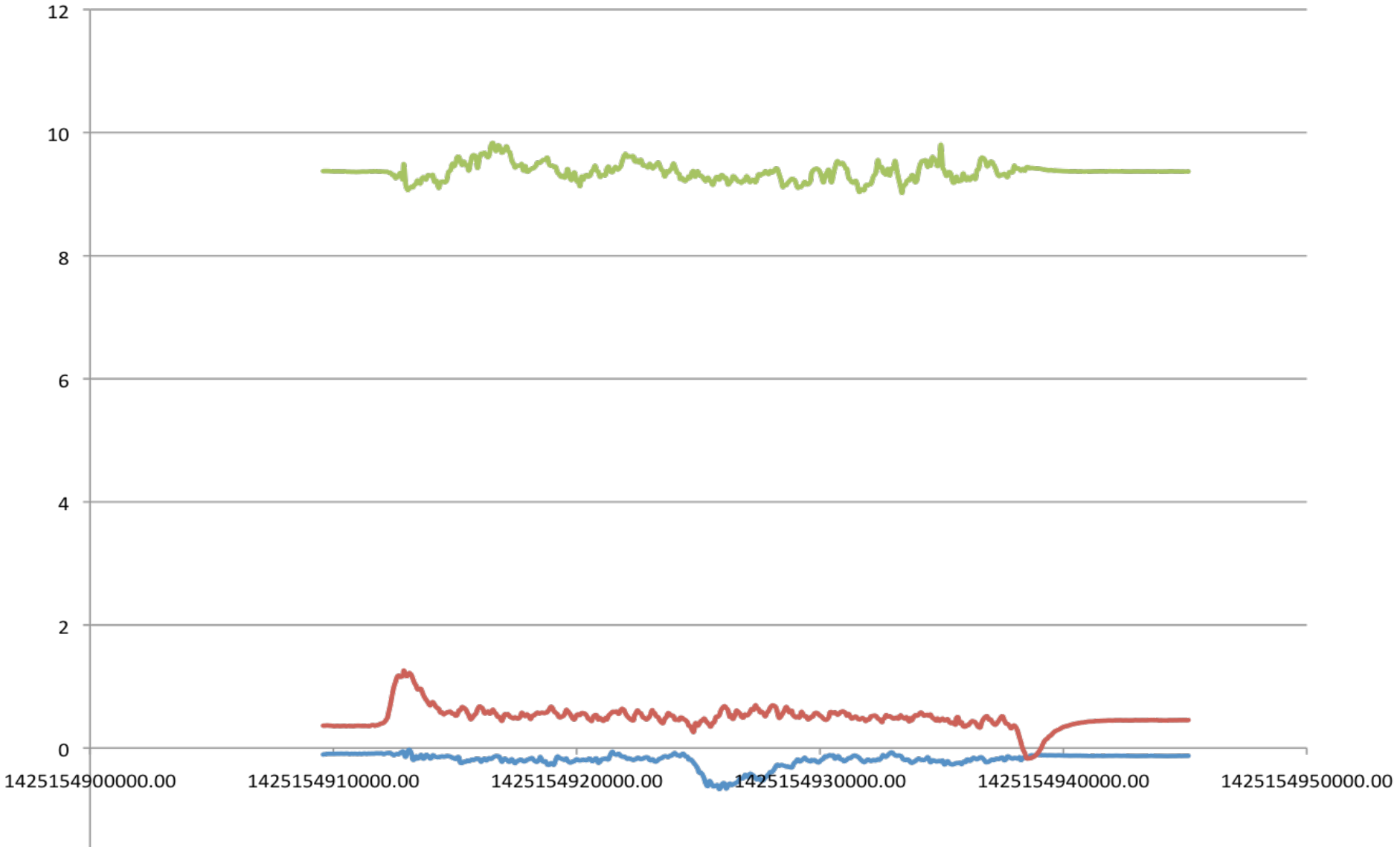
Alpha = 0.2


12
10
8
6
4
2
0
-2
-4
1425154905000.1
1425154910000.1
1425154915000.1
1425154920000.1
1425154925000.1
1425154930000.1
1425154935000.1
1425154940000.1

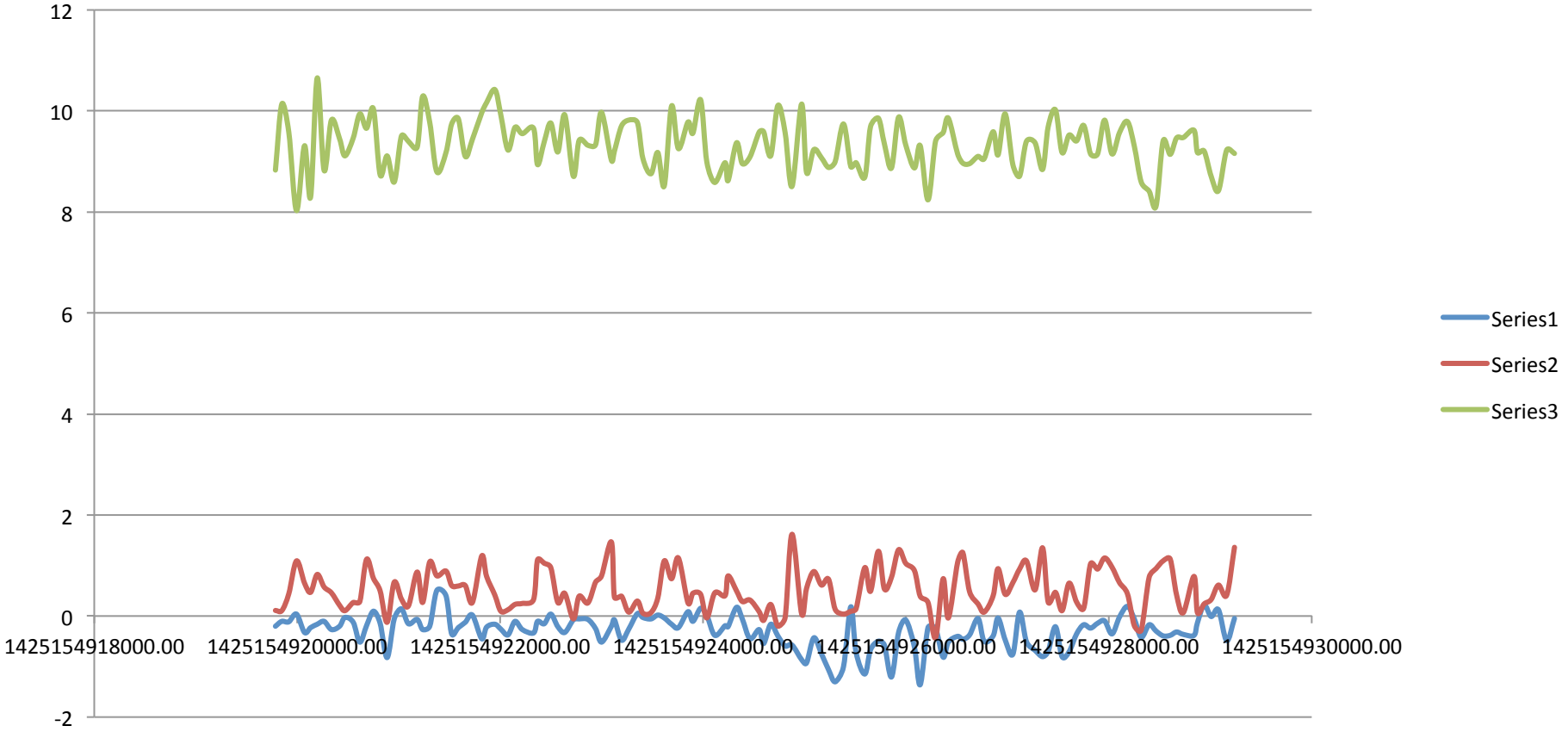
Alpha = 0.1



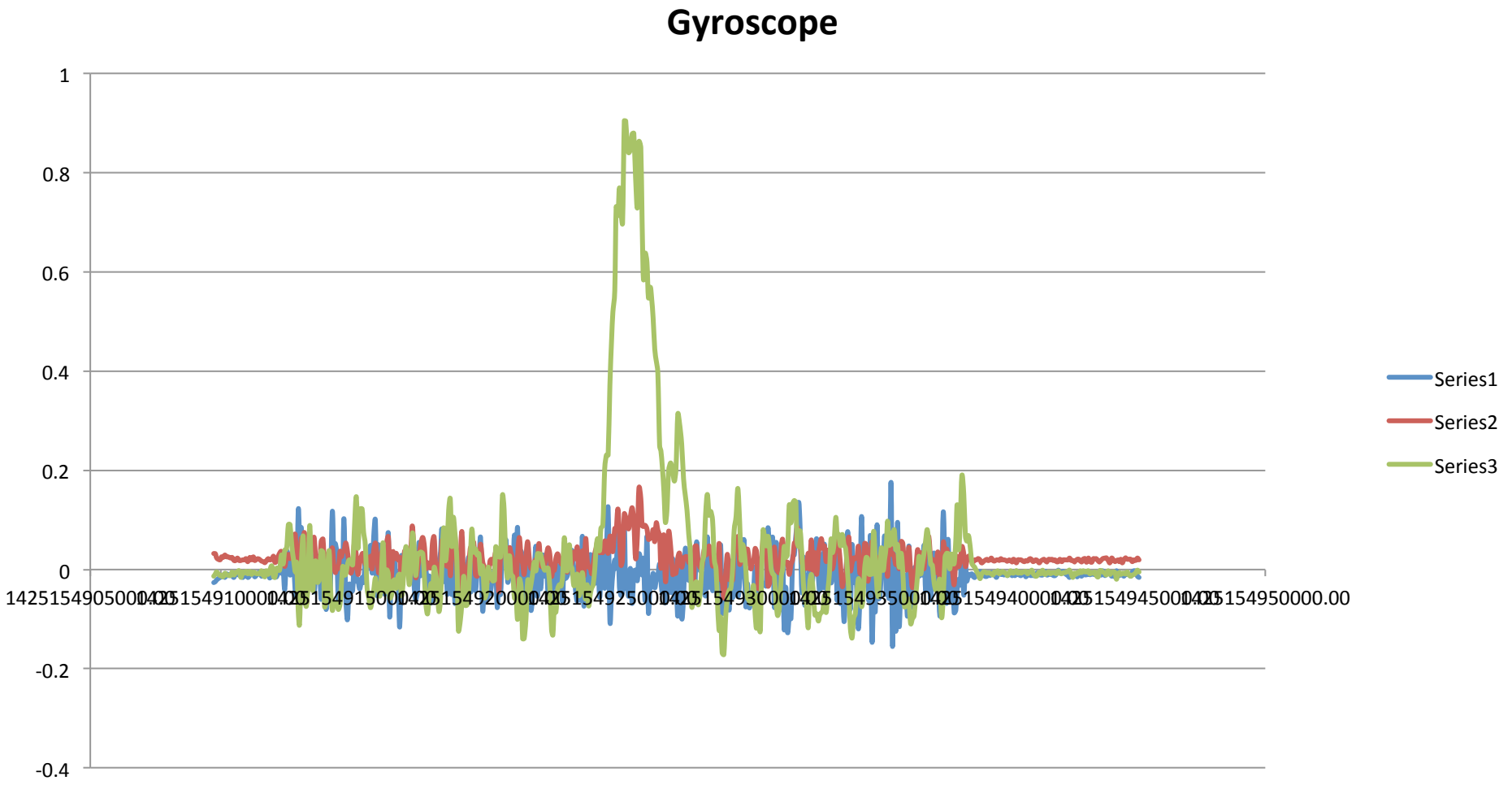
Alpha = 0.1


12
10
8
6
4
2
0
1425154900000.00
1425154910000.00
1425154920000.00
1425154930000.00
1425154940000.00
1425154950000.00

Zoom in


12
10
8
6
4
2
0
-2
1425154918000.00
1425154920000.00
1425154922000.00
1425154924000.00
1425154926000.00
1425154928000.00
1425154930000.00
Series1
Series2
Series3

Gyroscope



Step counting