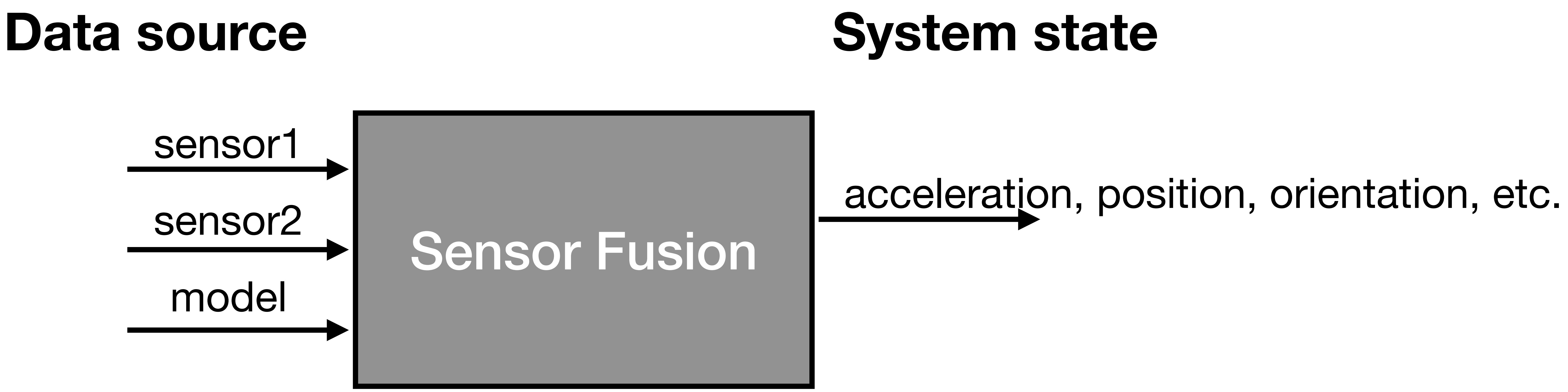
# Sensors II

## Fast Robots, ECE4160/5160, MAE 4190/5190

E. Farrell Helbling, 1/29/25

# Sensor Fusion

- Combine two or more data sources in a way that generates “better” understanding of the system
  - More consistent, accurate, and dependable signal over time

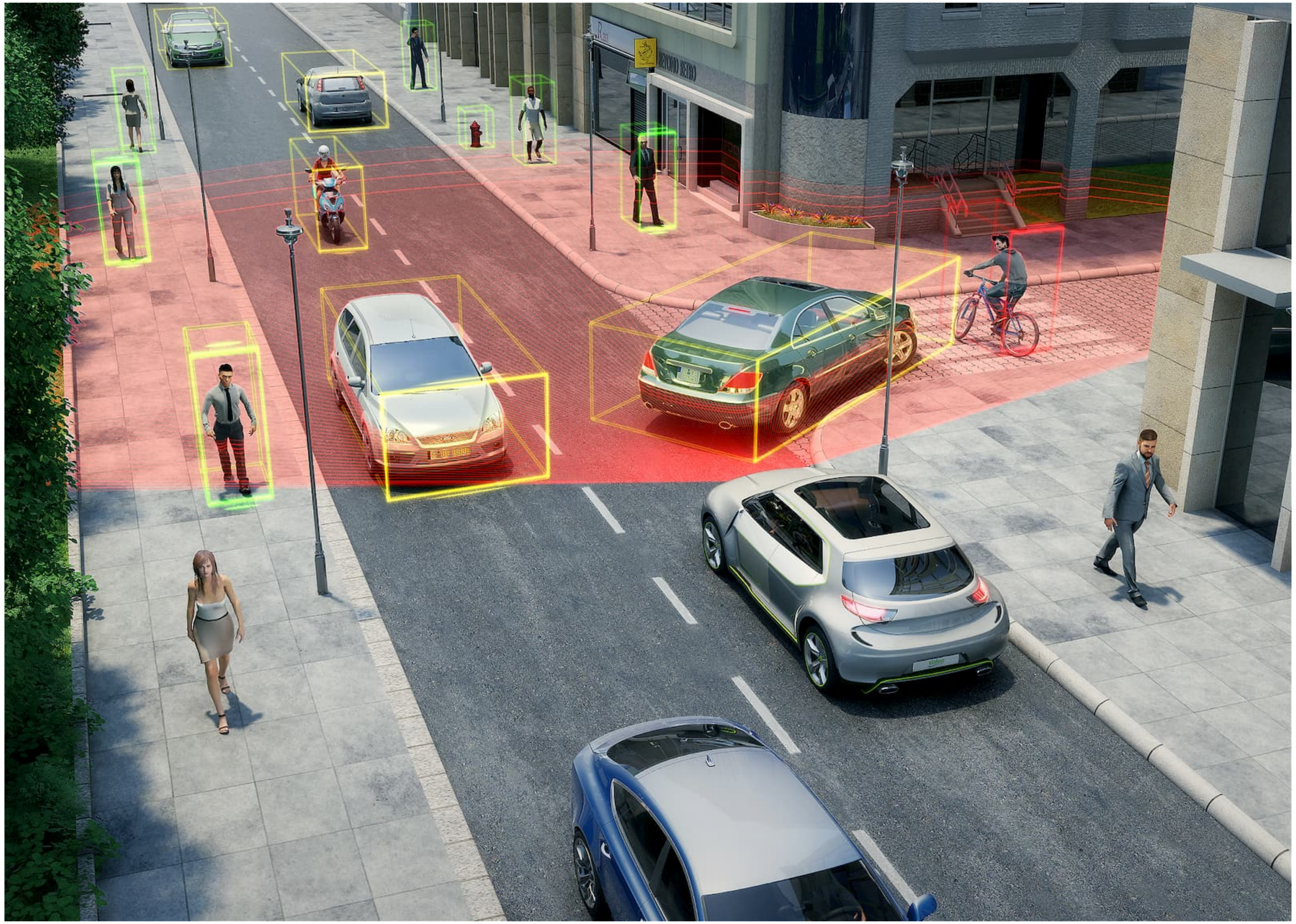


# Sensor Fusion

- Combine two or more data sources in a way that generates “better” understanding of the system
    - More consistent, accurate, and dependable signal over time


- Responsibilities:
    - Self-awareness (where am I? what am I doing? what is my current state?
    - Situational awareness (detection/ tracking)

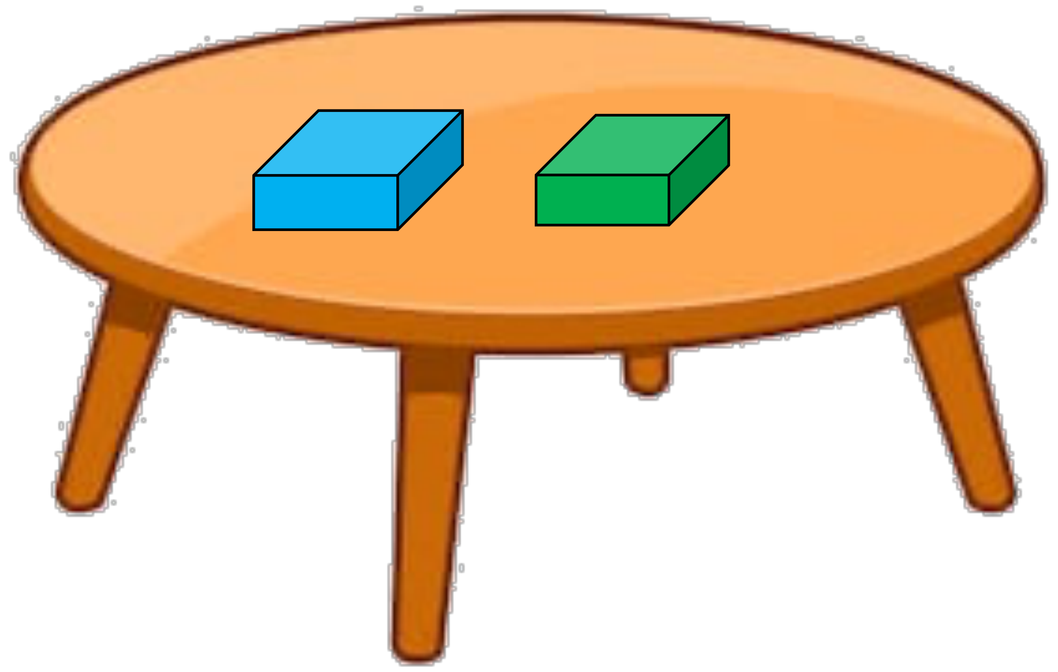
# Situational awareness

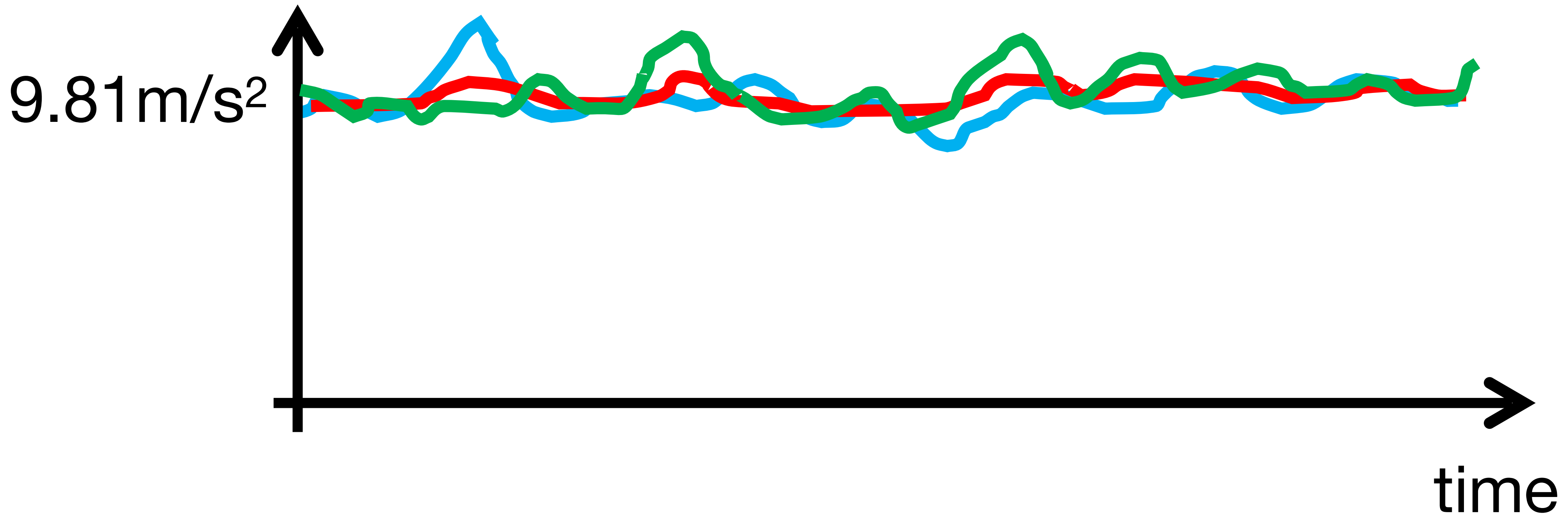


Valeo's LIDAR

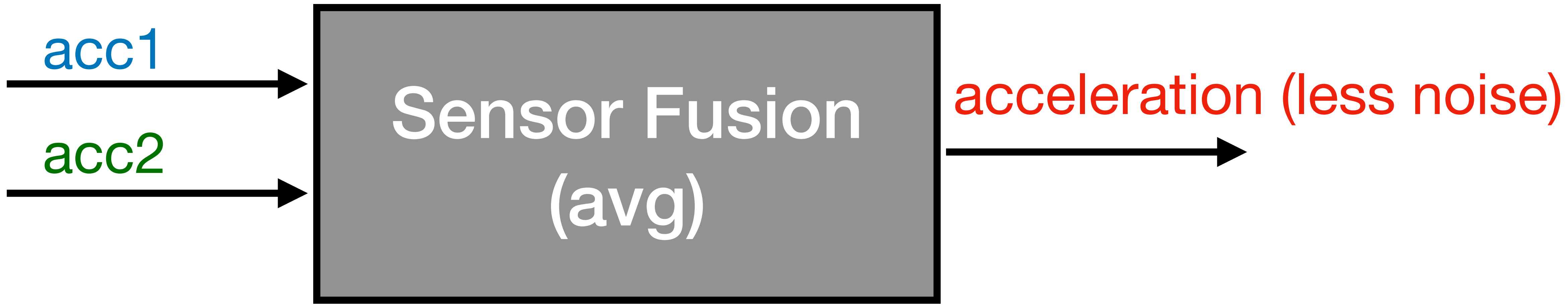
# Sensor Fusion

- Increase the quality of the data: less noise, uncertainty, deviation



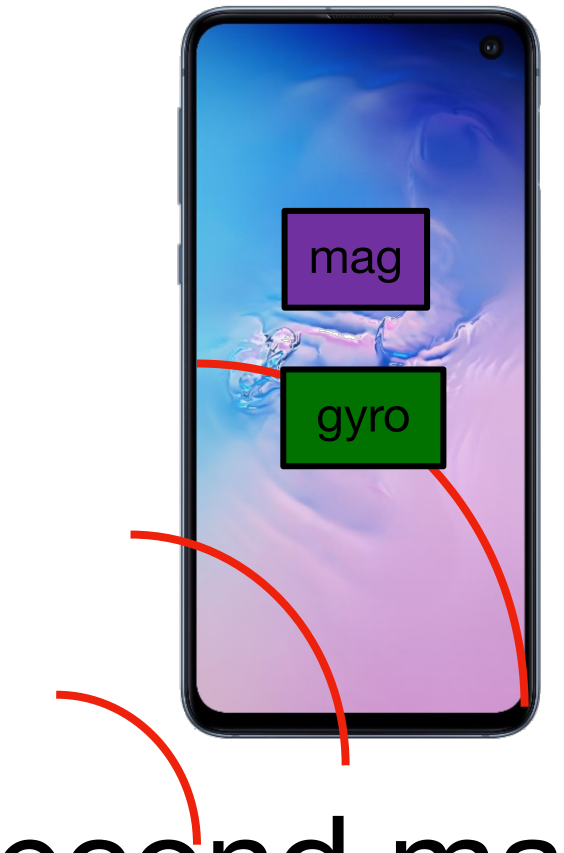


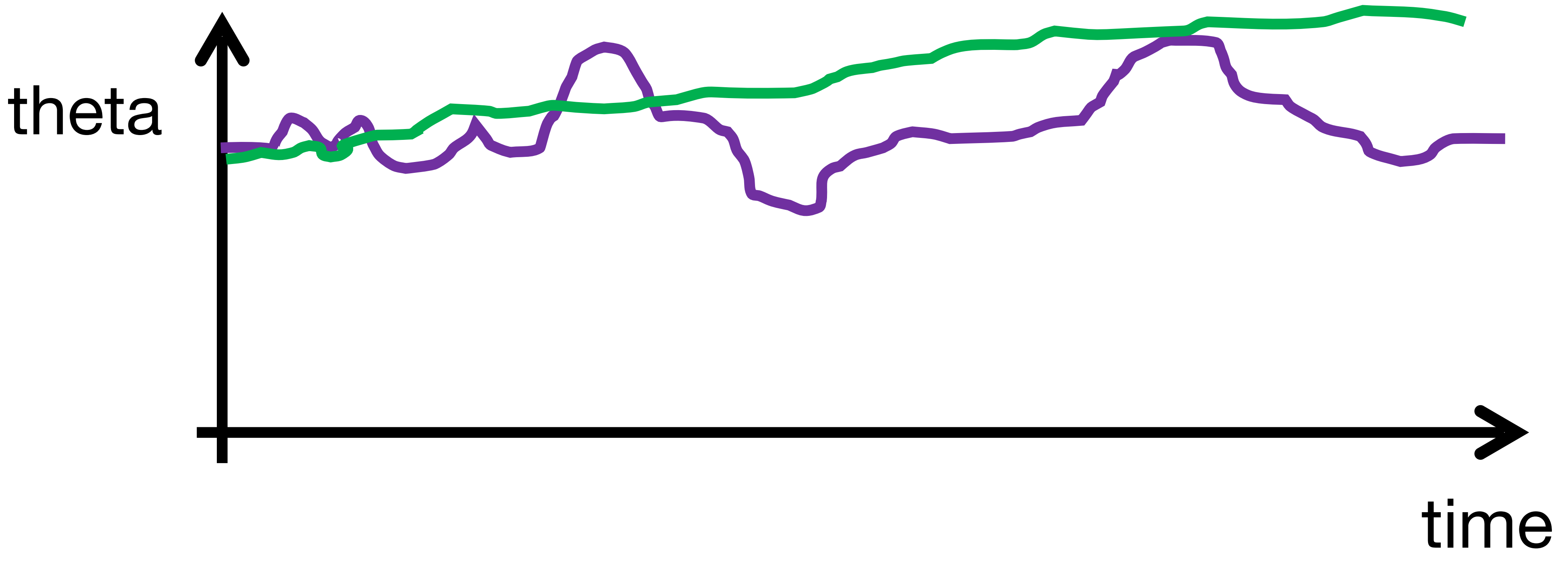

- Adding sensors lowers noise:
  - $n = 1/\sqrt{N}$
  - Only true if the noise is not correlated!



# Sensor Fusion

- Increase the quality of the data: less noise, uncertainty, deviation

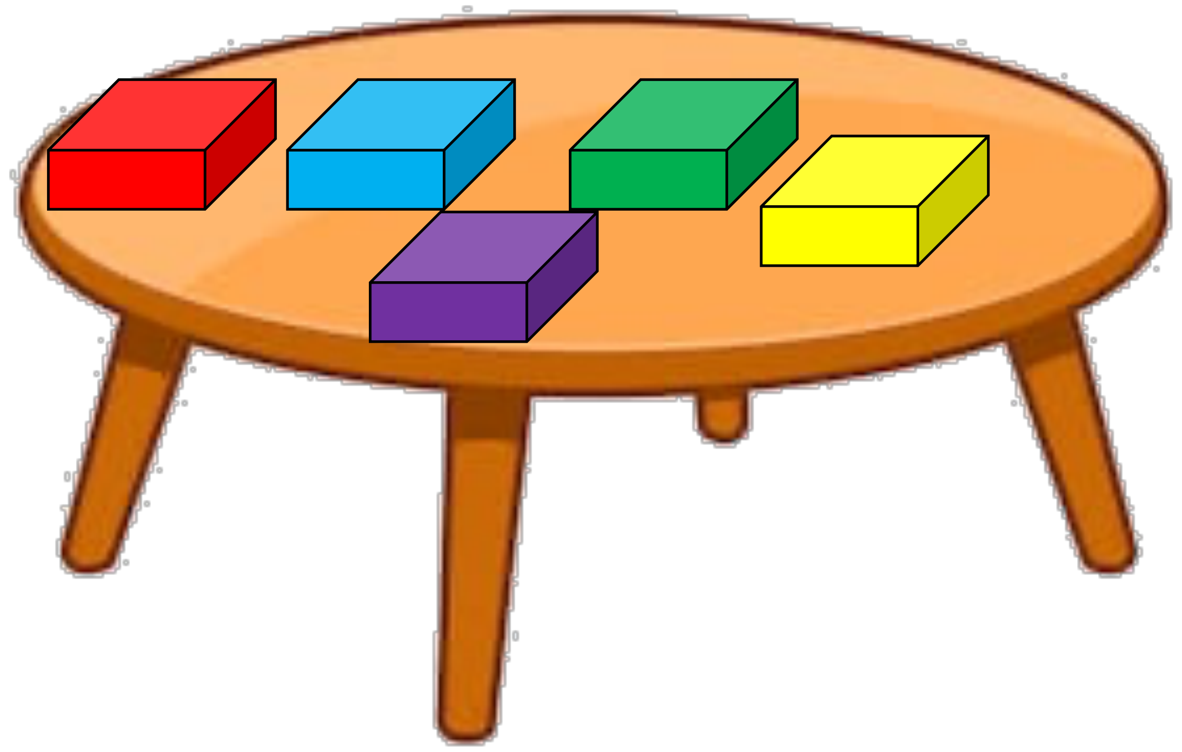


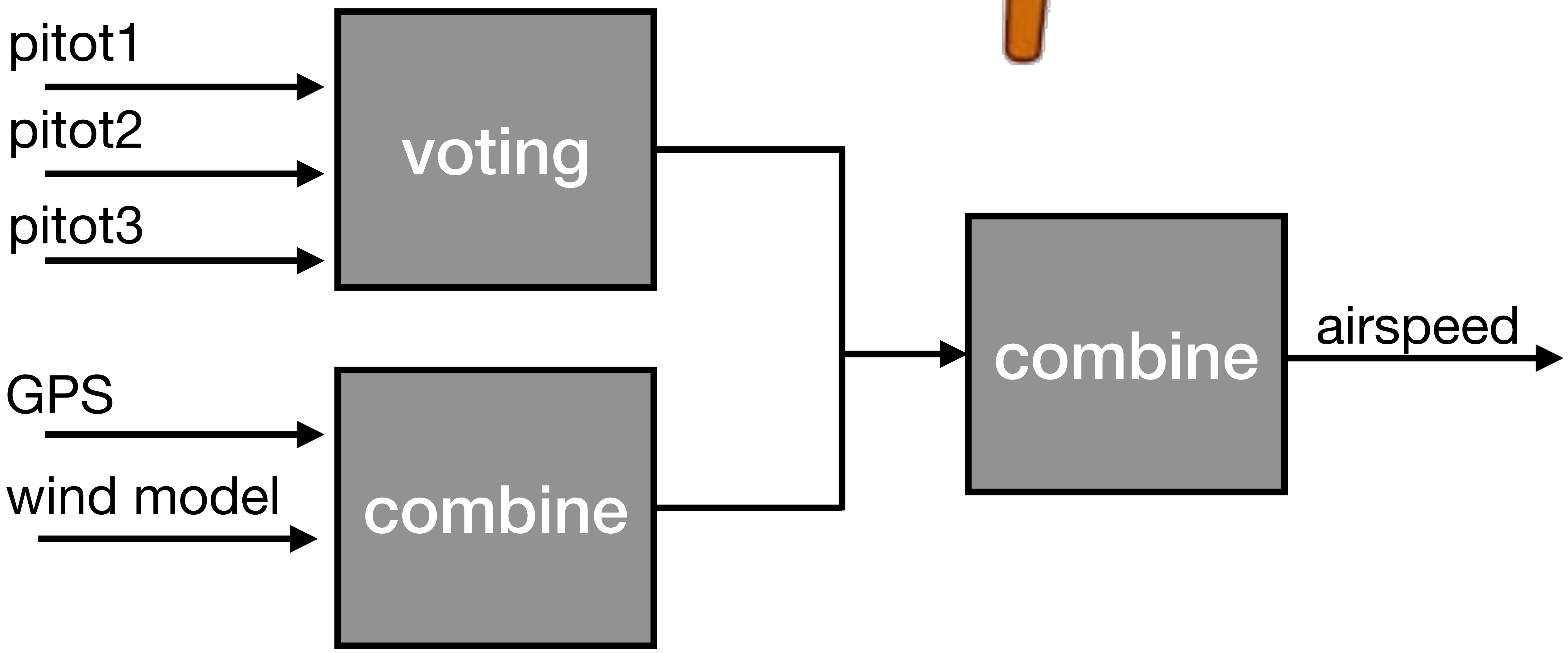




- Add a second mag?
- Move the sensor away from the field
- Low pass filter
- Fuse the mag data with the gyroscope data

# Sensor Fusion

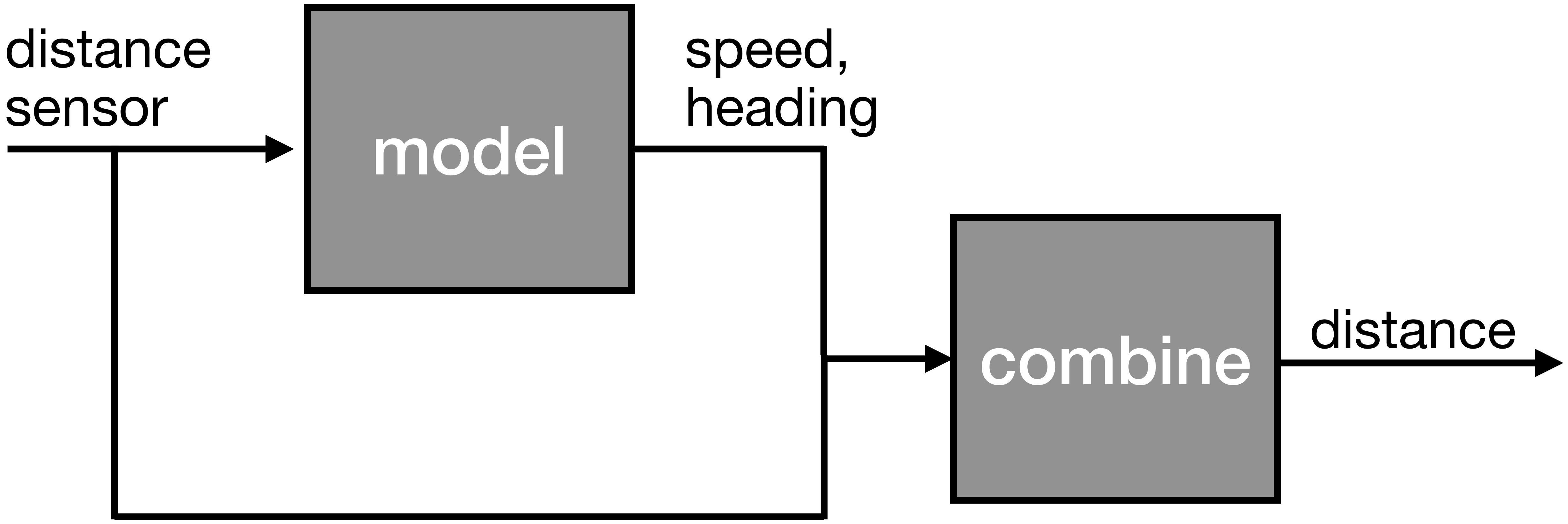
- Increase the quality of the data: less noise, uncertainty, deviation
- Increase data reliability





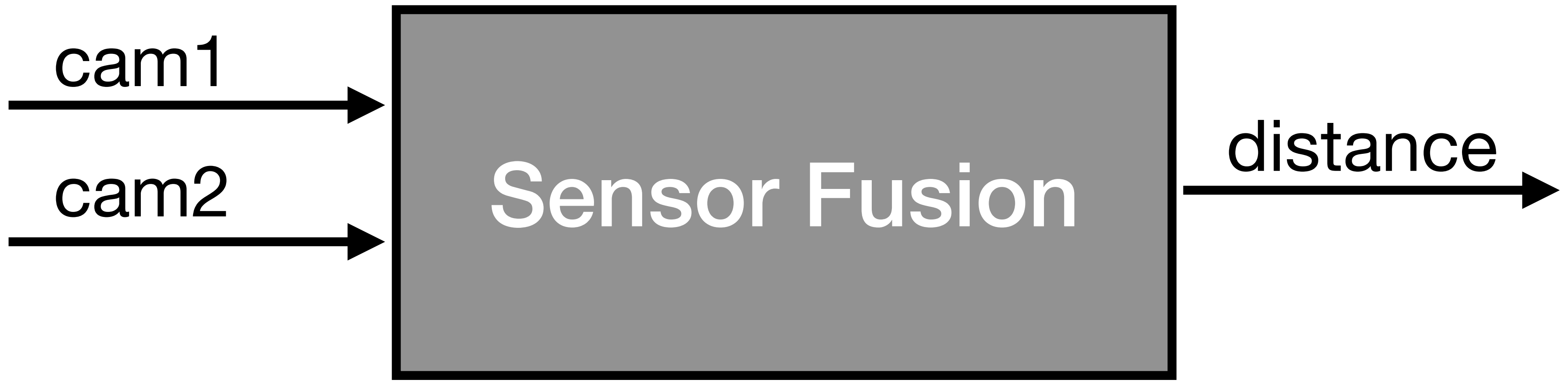
# Sensor Fusion

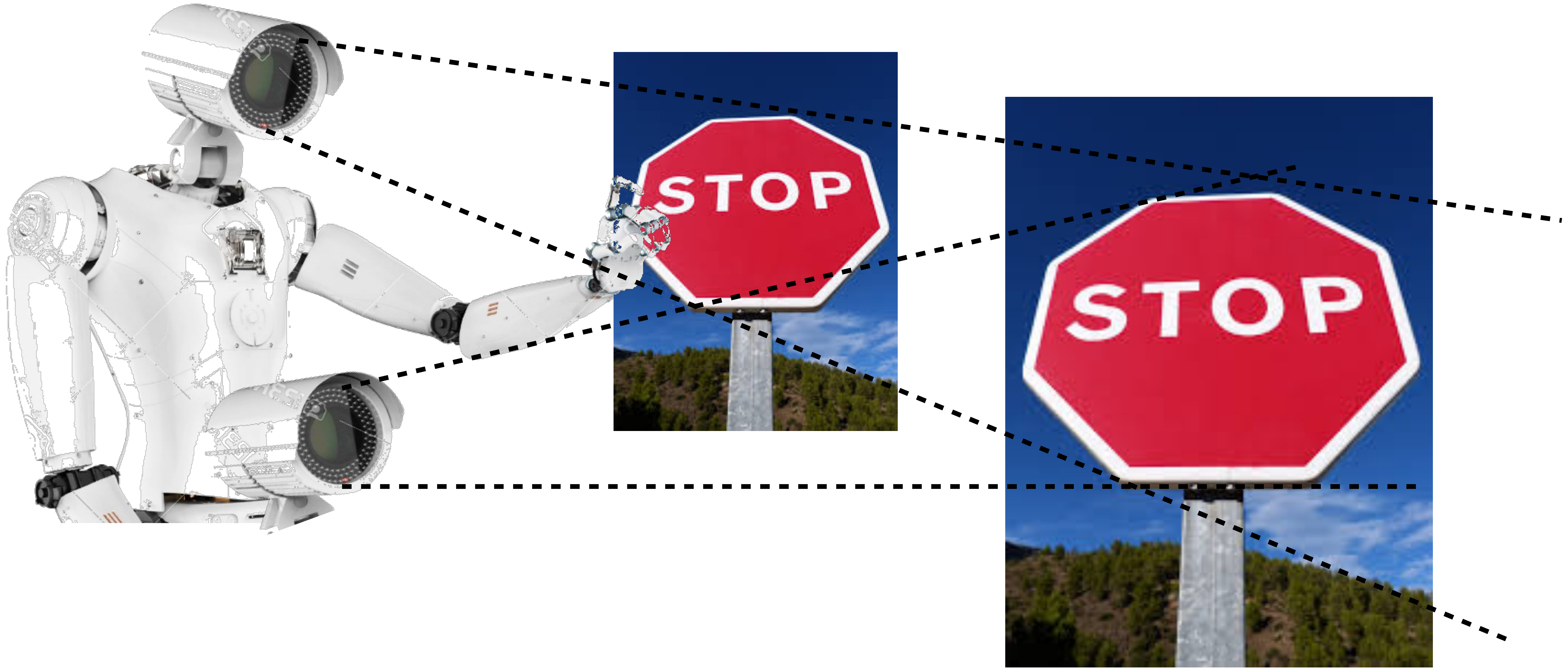
- Increase the quality of the data: less noise, uncertainty, deviation
- Increase data reliability

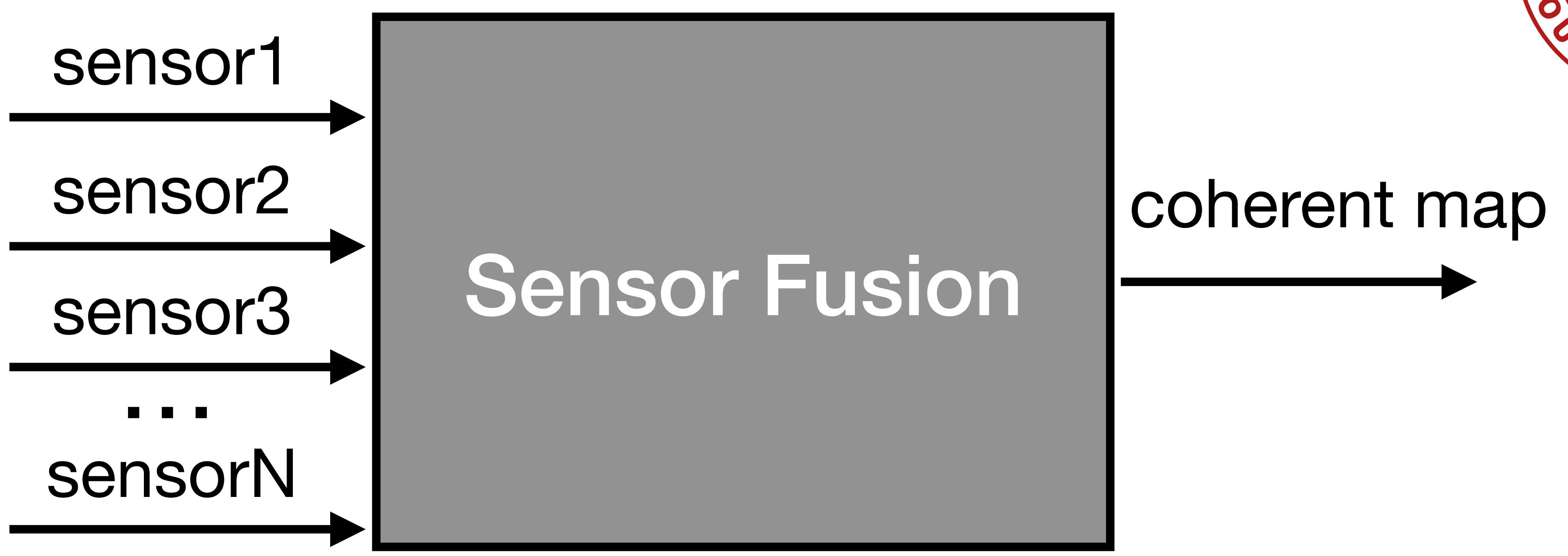
# Sensor Fusion

- Increase the quality of the data: less noise, uncertainty, deviation
- Increase data reliability
- Measure unmeasured states

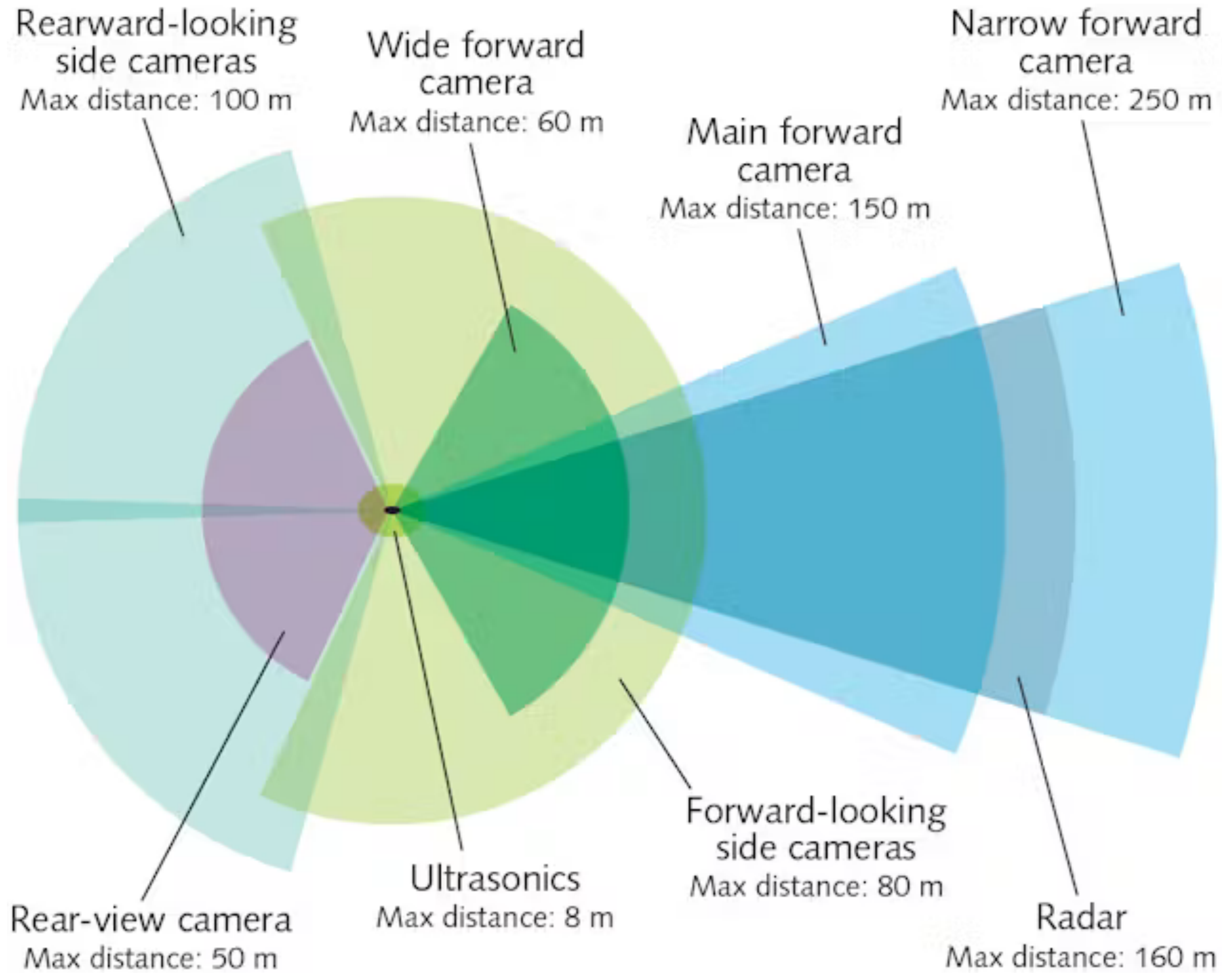


# Sensor Fusion



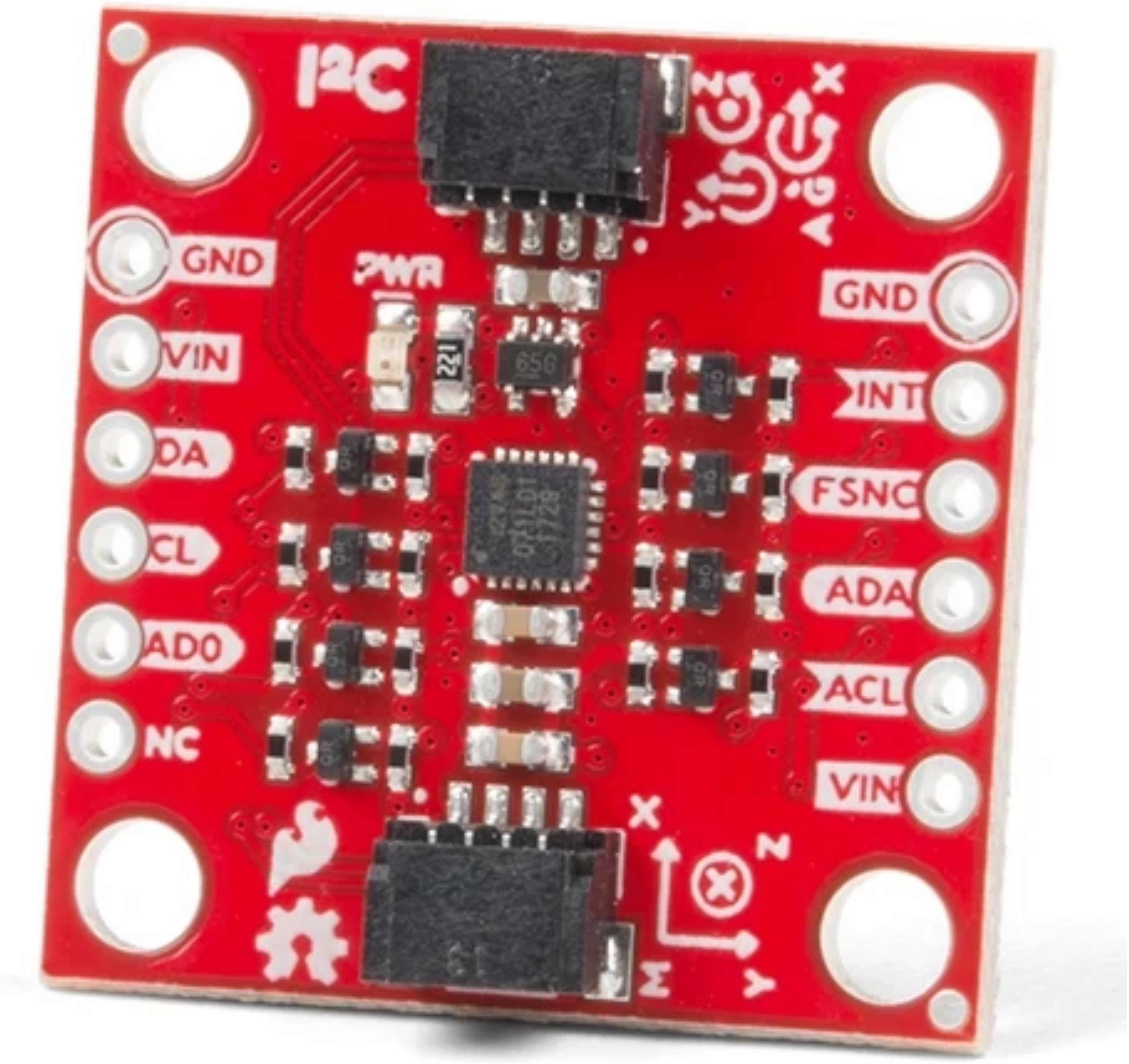

- Increase the quality of the data:
  - less noise, uncertainty, deviation
- Increase data reliability
- Measure unmeasured states
- Increase coverage area



# Sources and references

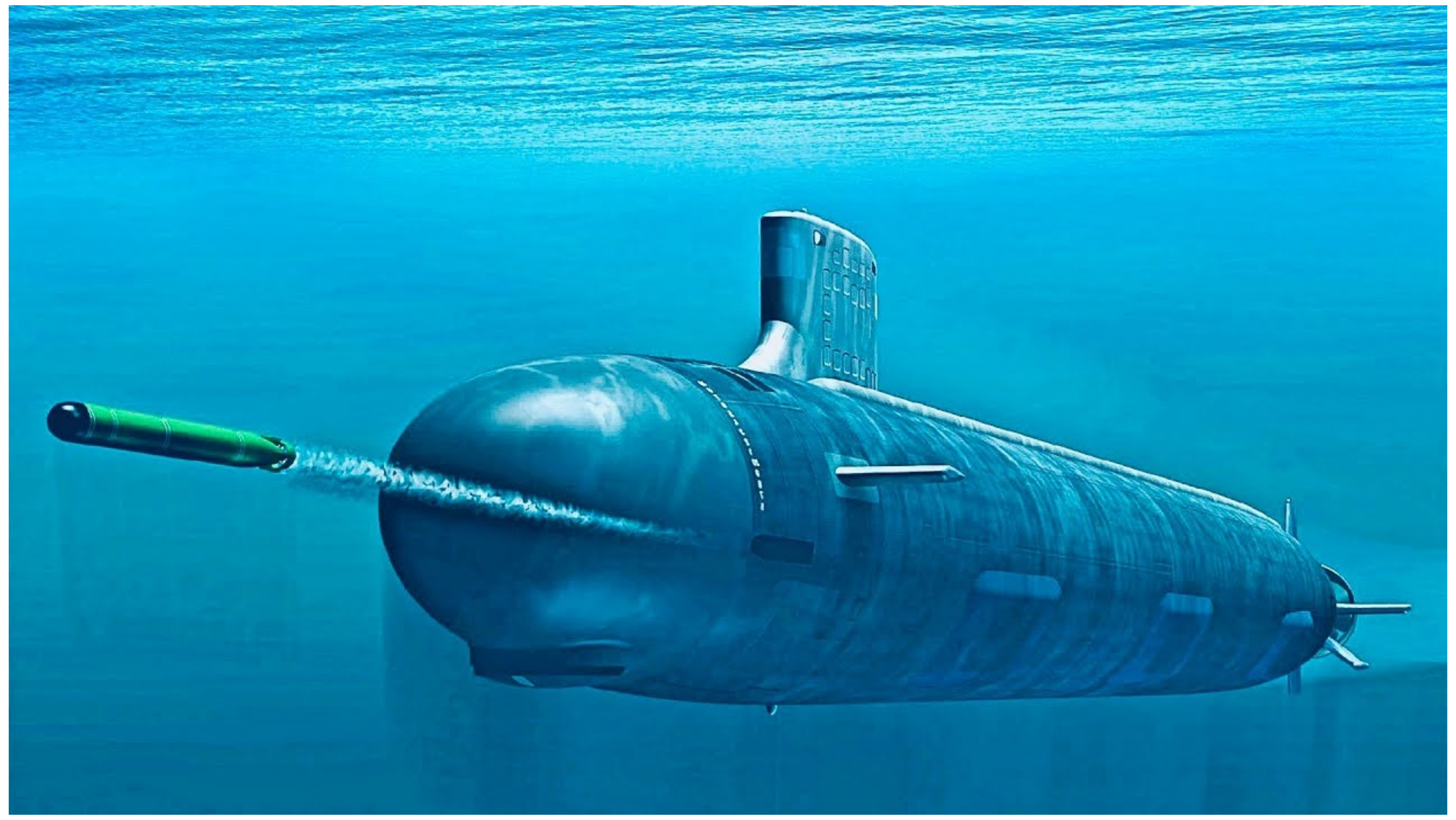

- http://www.cs.cmu.edu/~rasc/Download/AMRobots4.pdf
- https://www.ti.com/lit/ug/sbau305b/sbau305b.pdf?ts=1599417595209&ref_url=https%253A%252F%252Fwww.google.com%252F
- https://hmc.edu/lair/ARW/ARW-Lecture01-Odometry.pdf
- Matlab Tech Talks on Sensor Fusion (https://www.youtube.com/watch?v=6qV3YjFppuc)
- Prof. Kirstin Petersen

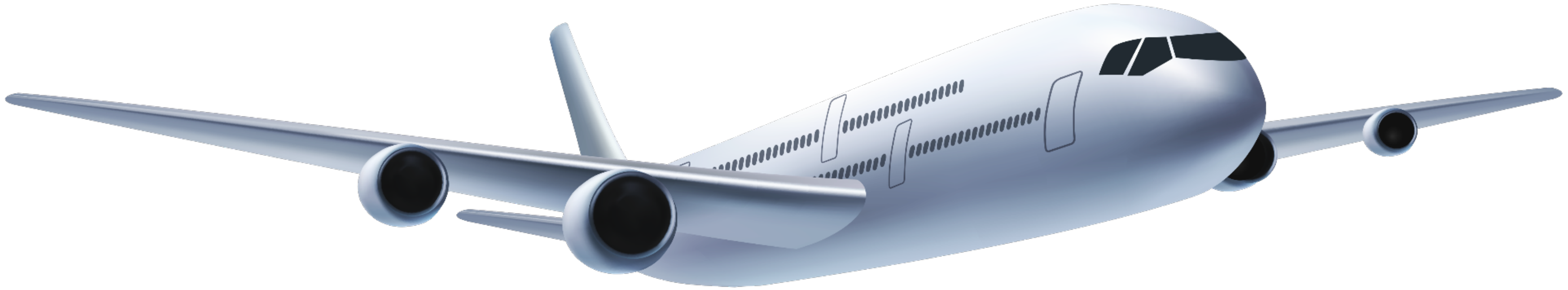
# IMU



# IMU

- Data related to orientation, velocity, and gravity



# IMU

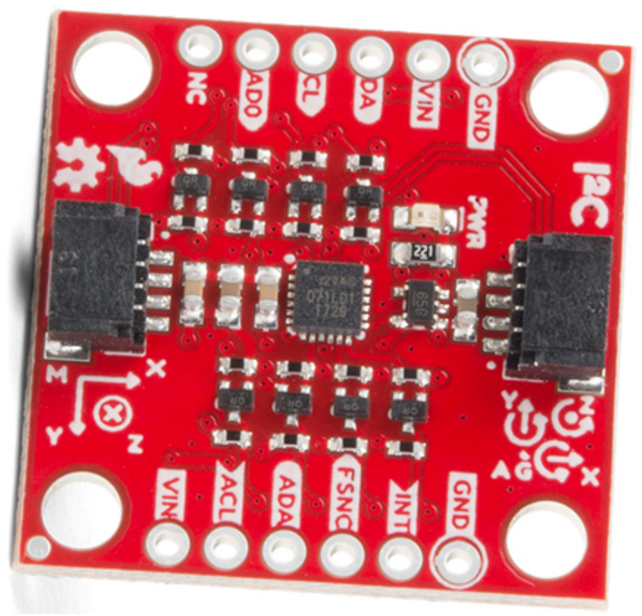
- Accelerometer
  - Linear acceleration, $a = \dot{v}$ [m/s²]
- Gyroscope
  - Angular velocity, $\omega = \frac{\Delta\theta}{\Delta t}$ [°/s]
- Magnetometer
  - Magnetic field strength, [uT] or [Gauss]
- NB: Gravity, magnetic fields, accelerations affect these sensors in many ways!

▶ Track orientation (position)

▶ Track orientation

} Dead reckoning

▶ Get absolute orientation

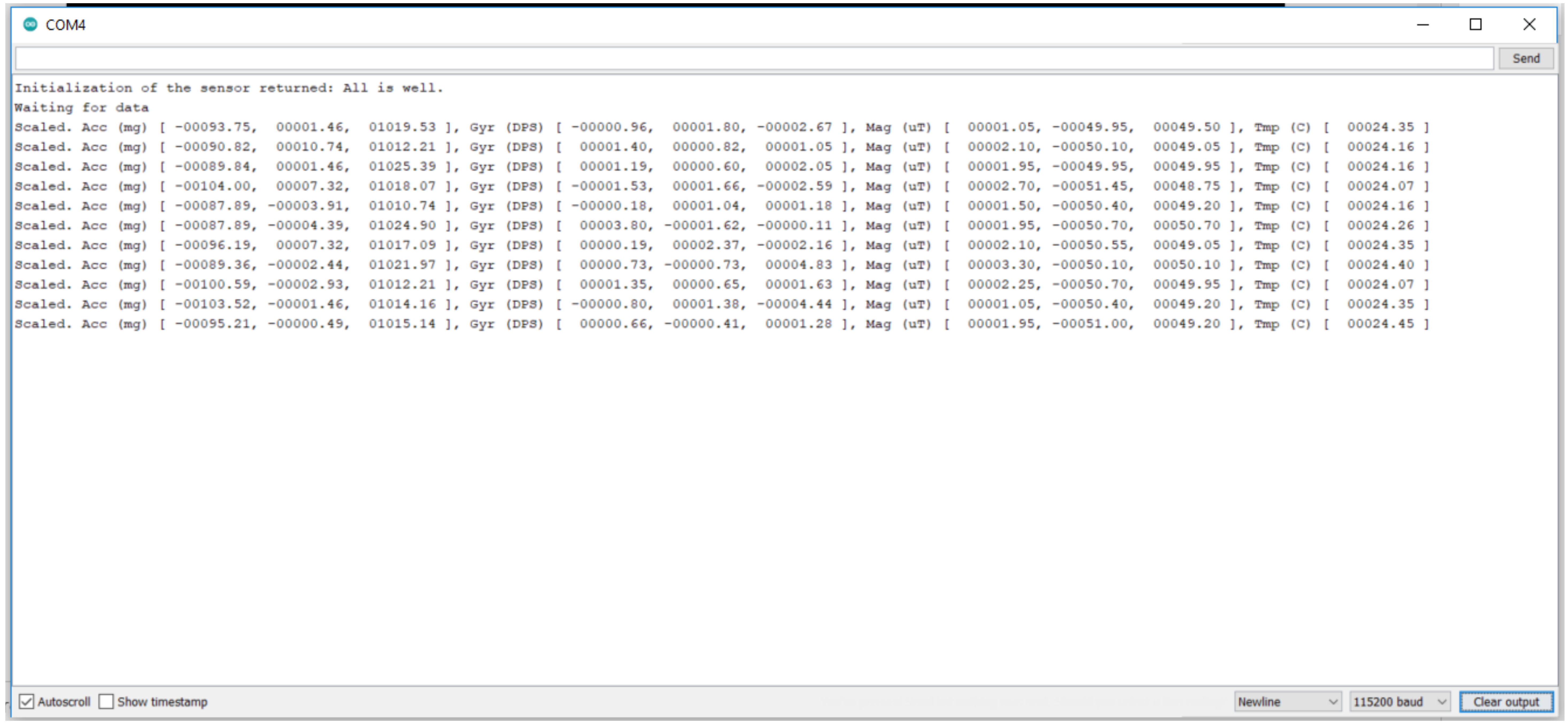


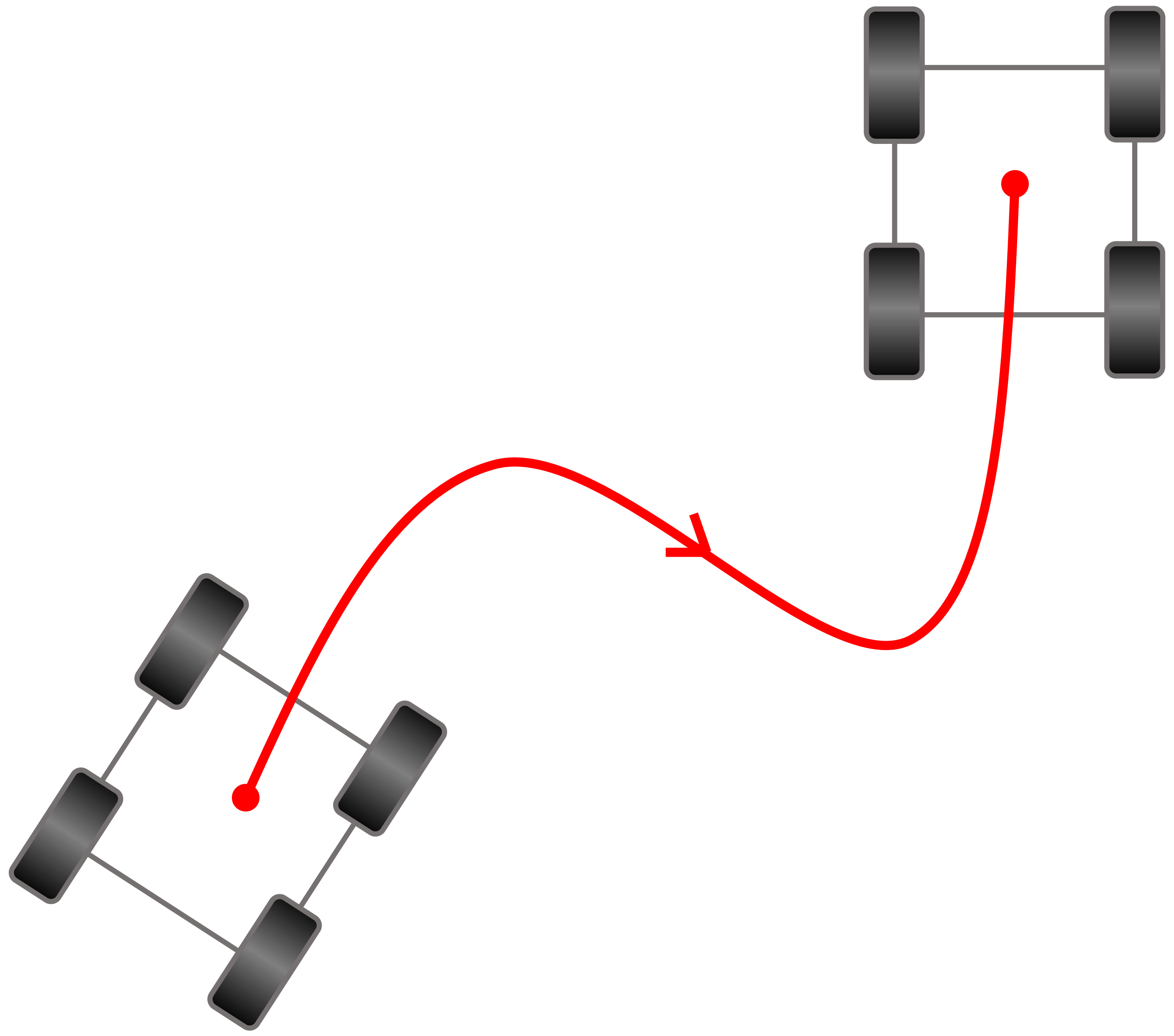
**ICM-20948**
- $16
- Low power
- 9-axis

# IMU

## Demo

- Install Sparkfun 9DOF IMU — ICM 20948 library
- Follow the basics example



# Accelerometer

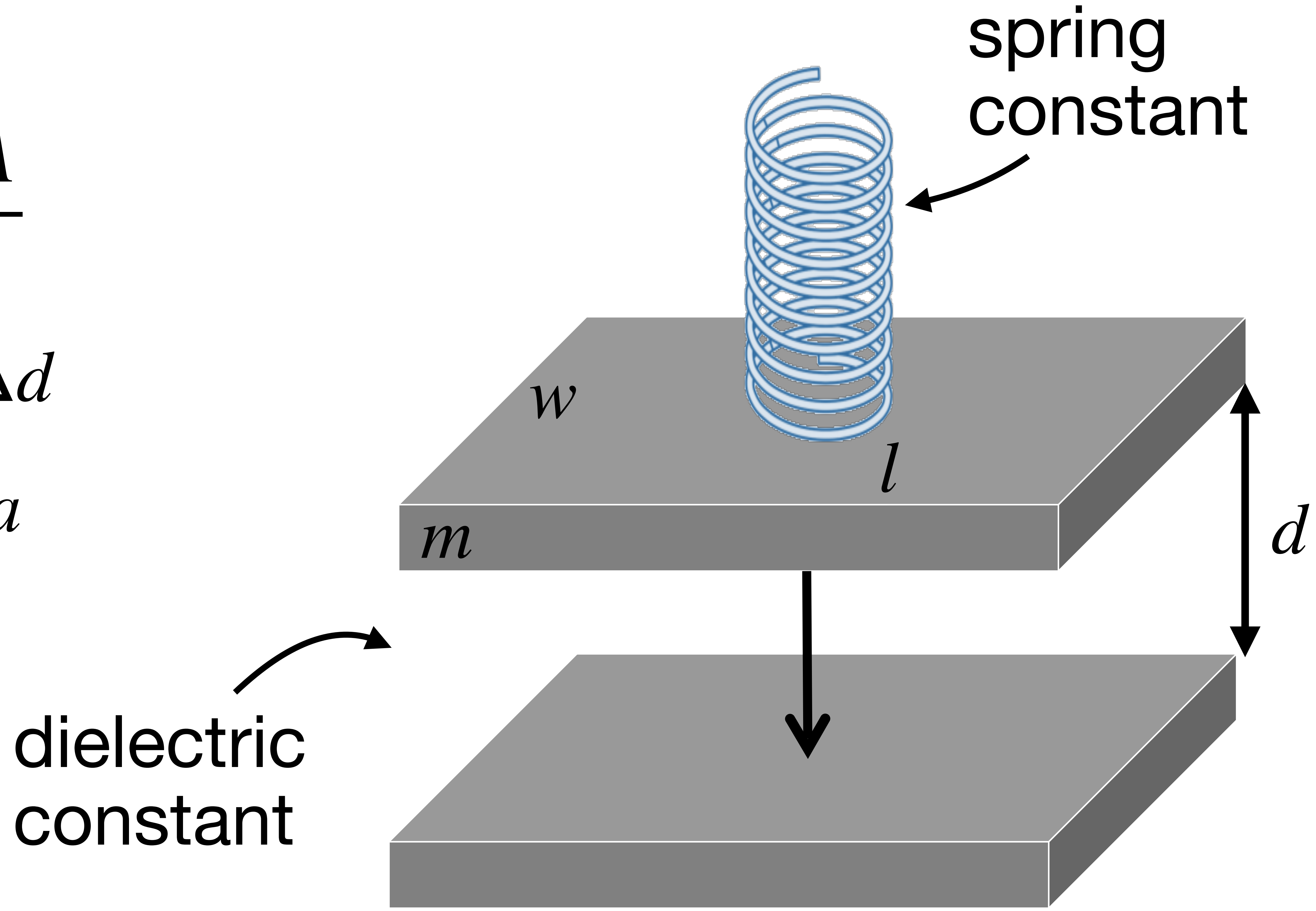
# Accelerometer

## Measure acceleration

$$C = \frac{\epsilon A}{d}$$

$$F = k\Delta d$$

$$F = ma$$



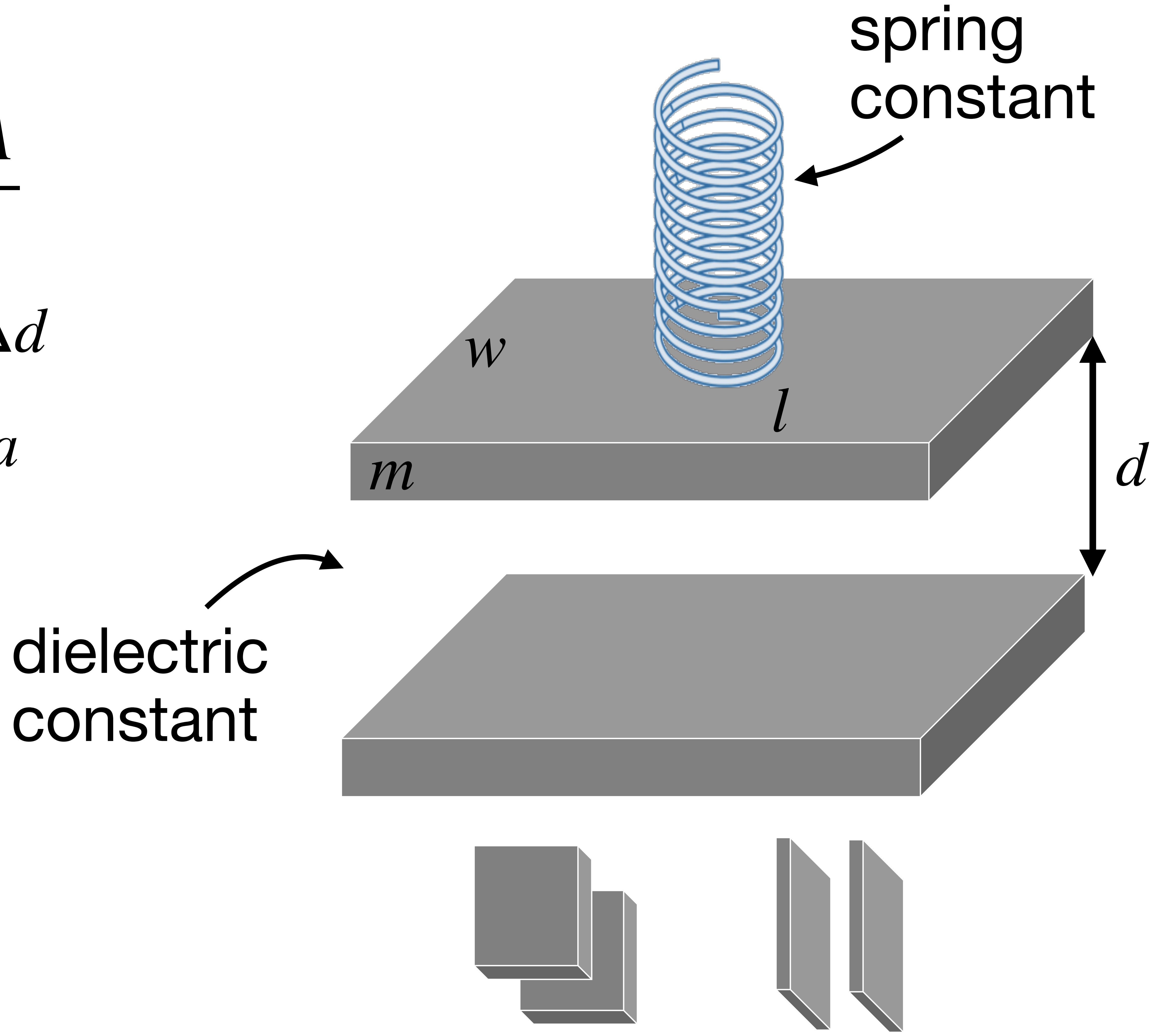
# Accelerometer

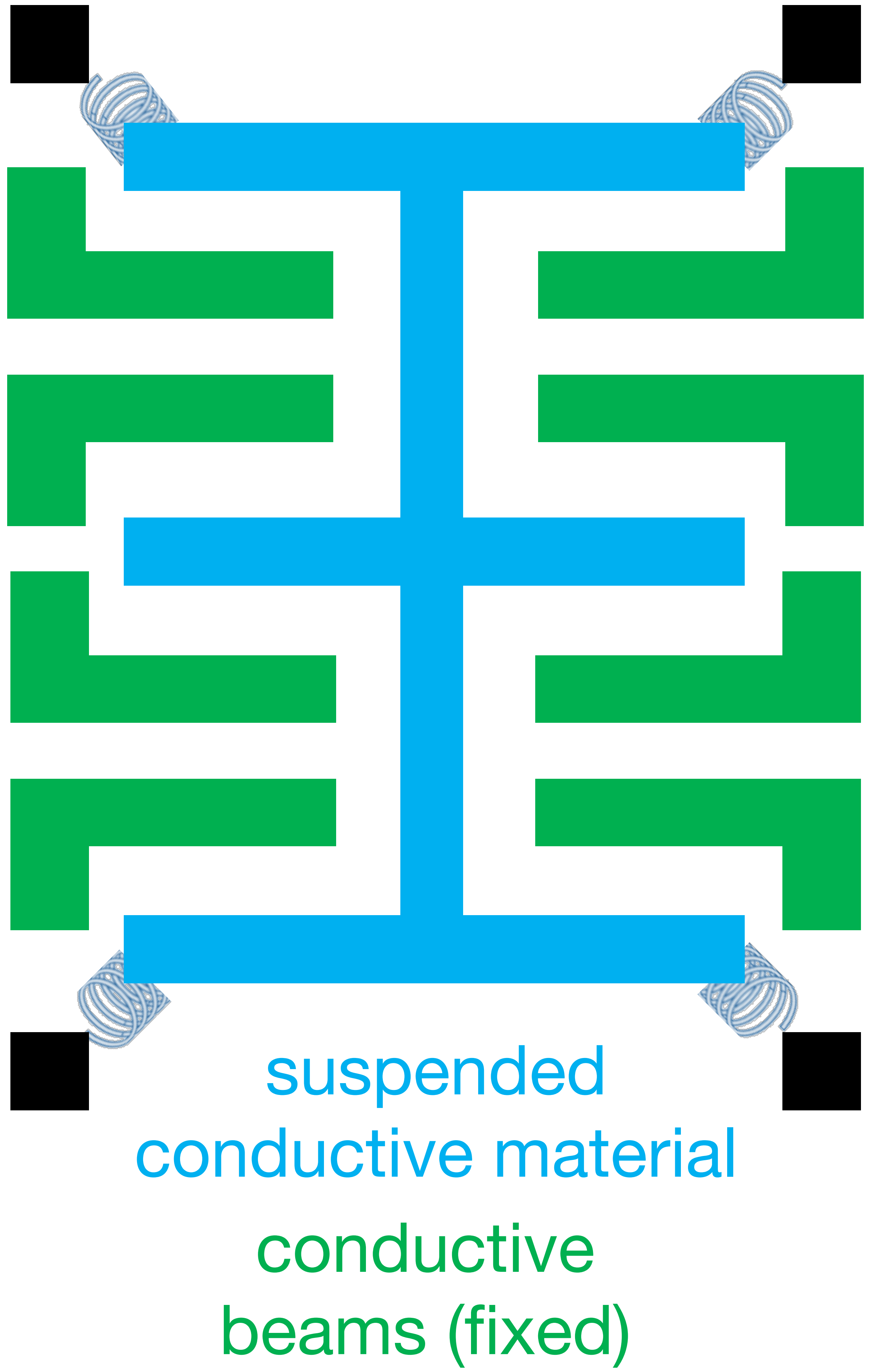
## Measure acceleration in 3D

$$C = \frac{\epsilon A}{d}$$

$$F = k\Delta d$$

$$F = ma$$

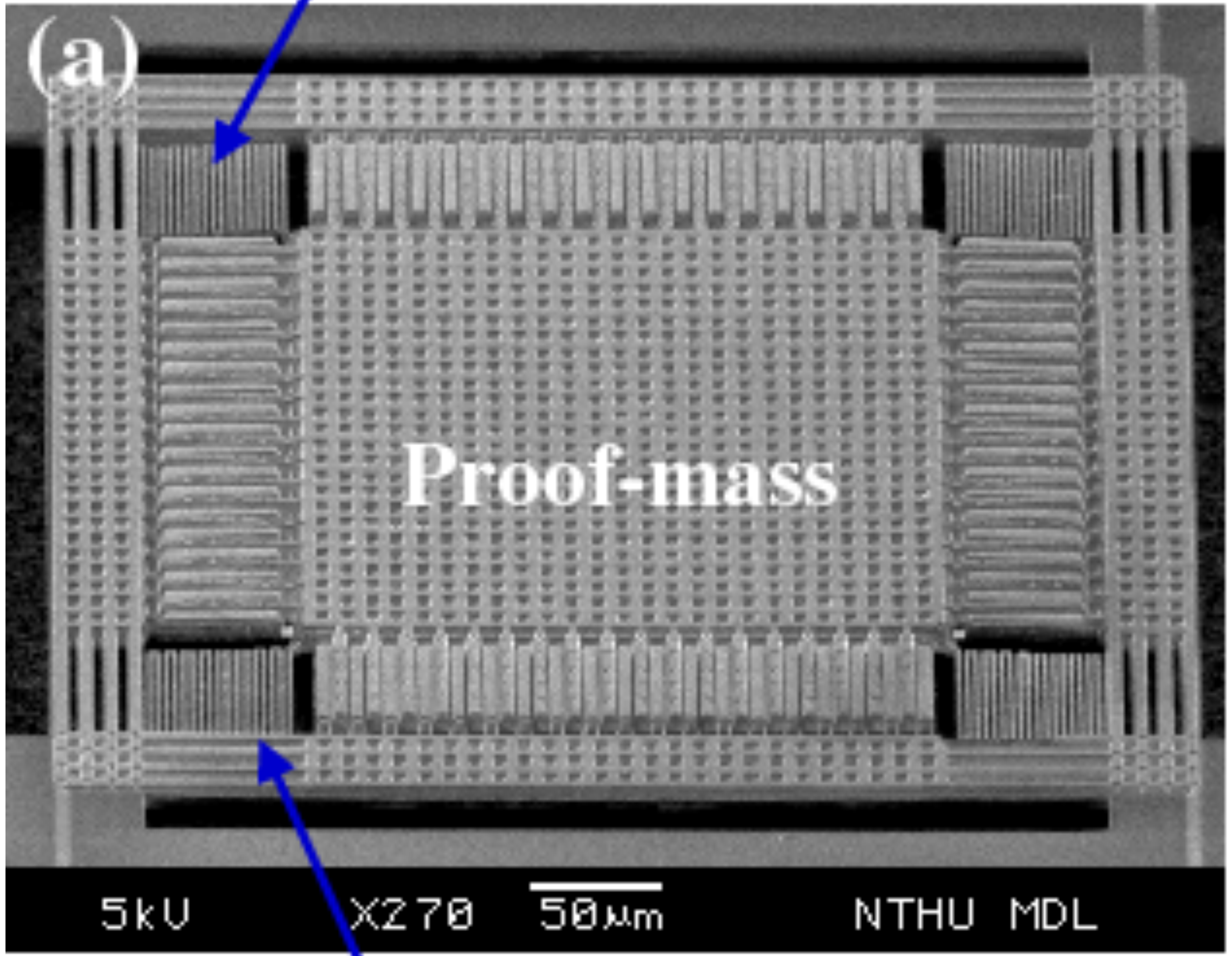


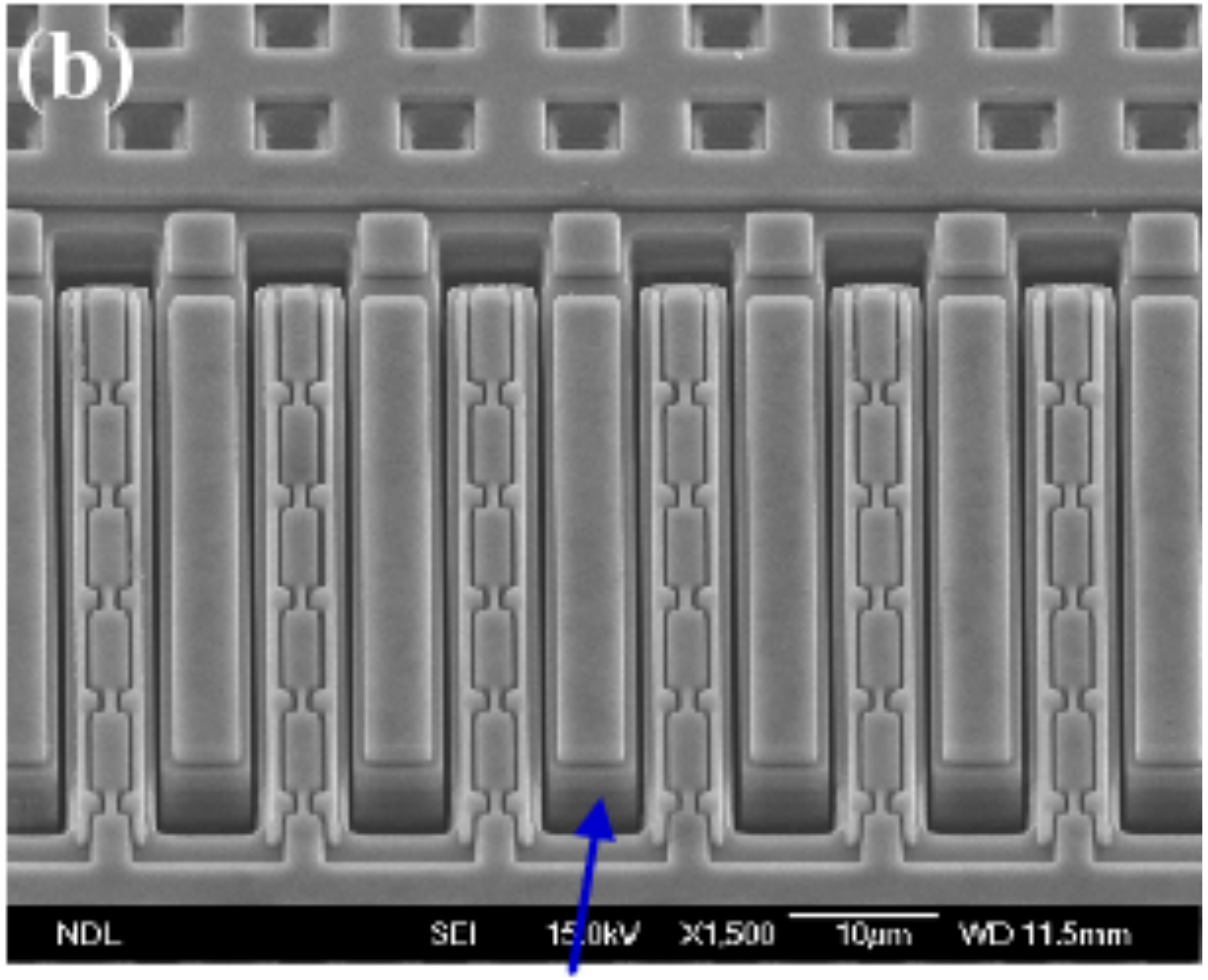



# Accelerometer

## Measure acceleration in 3D

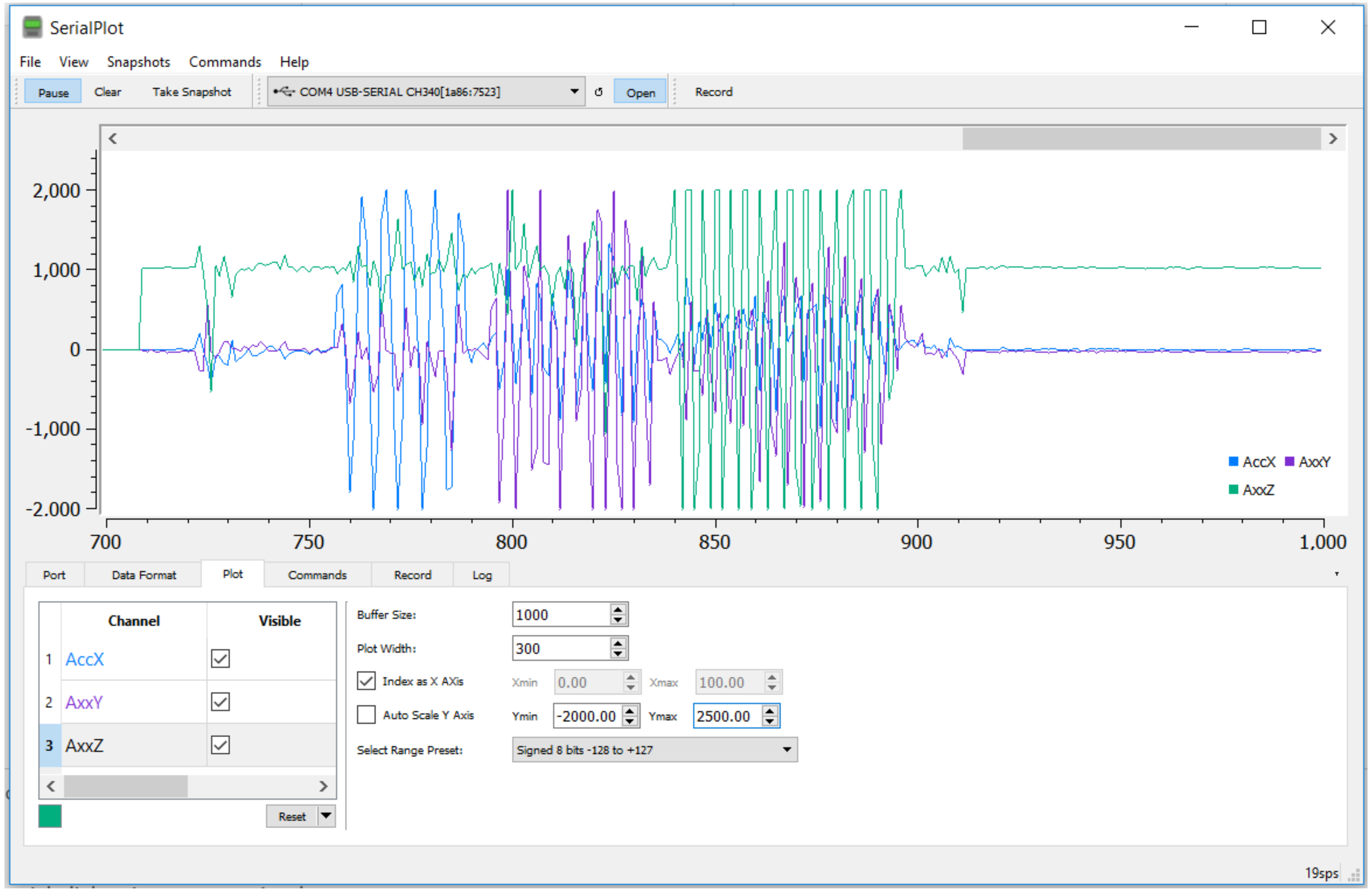
Fang, 2011



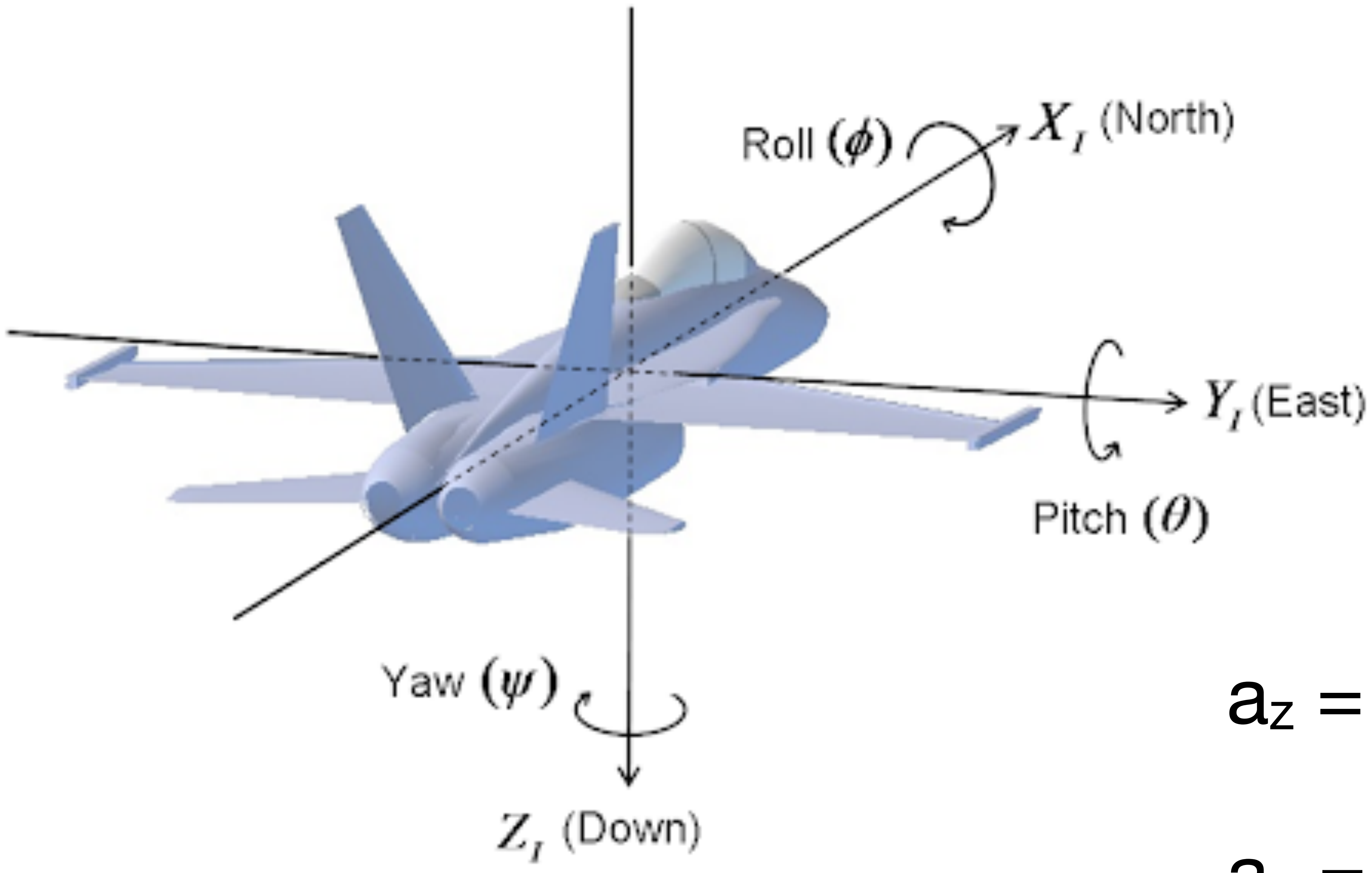
# Accelerometer

## Use Serial Monitor or Serial Plot



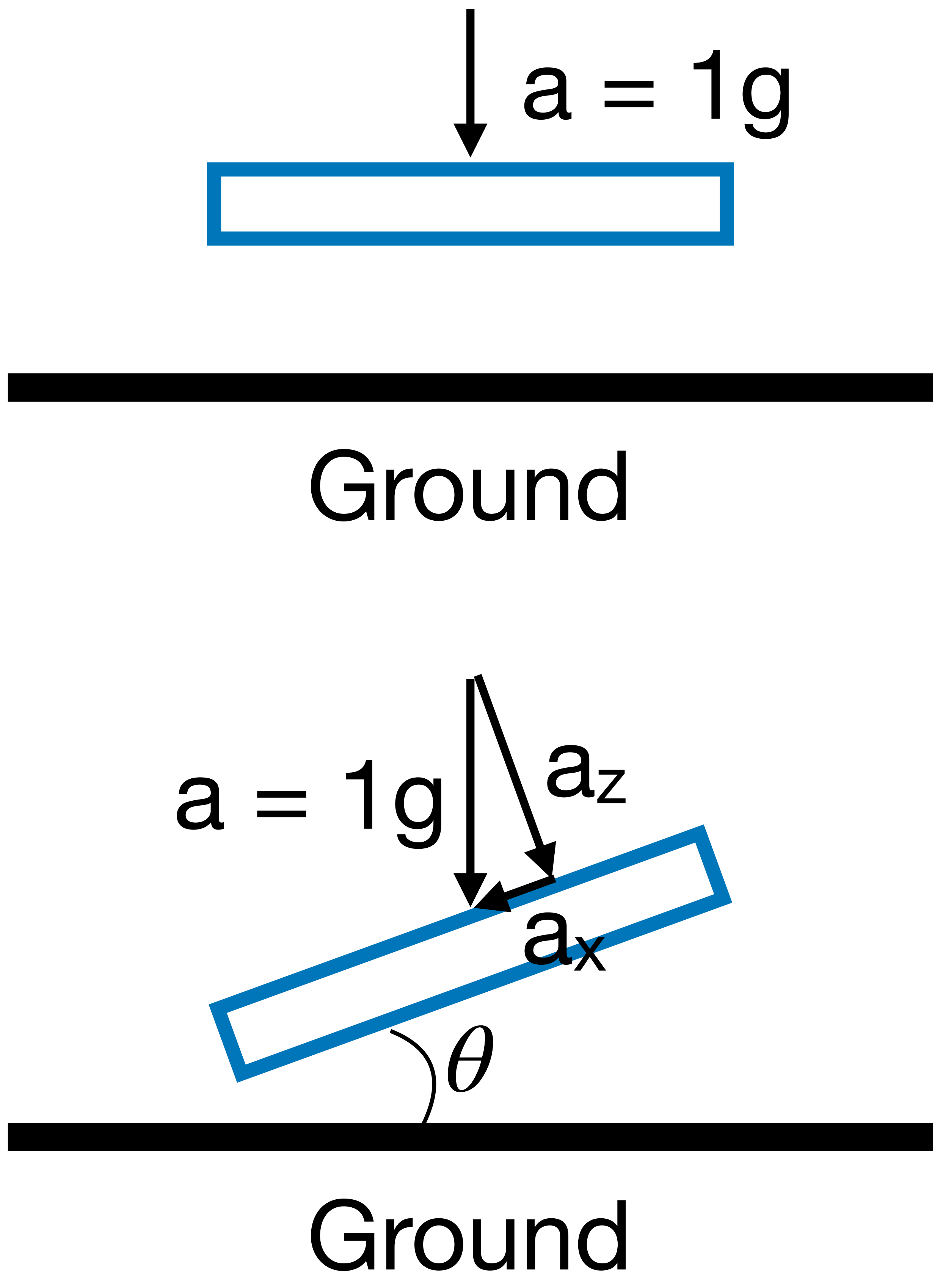
# Accelerometer

## Roll, Pitch, Yaw



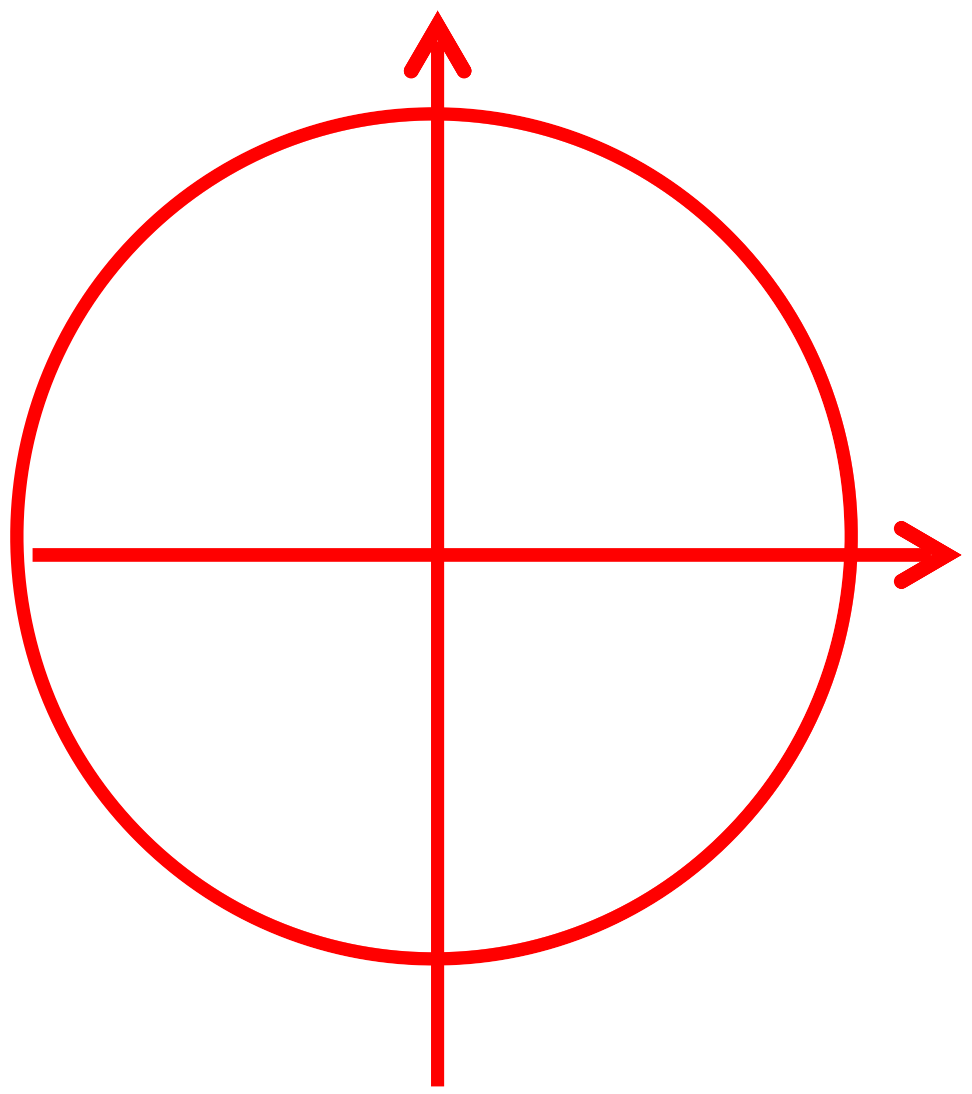

$$a_z = 1g\cos(\theta)$$

$$a_x = 1g\sin(\theta)$$



# atan vs. atan2

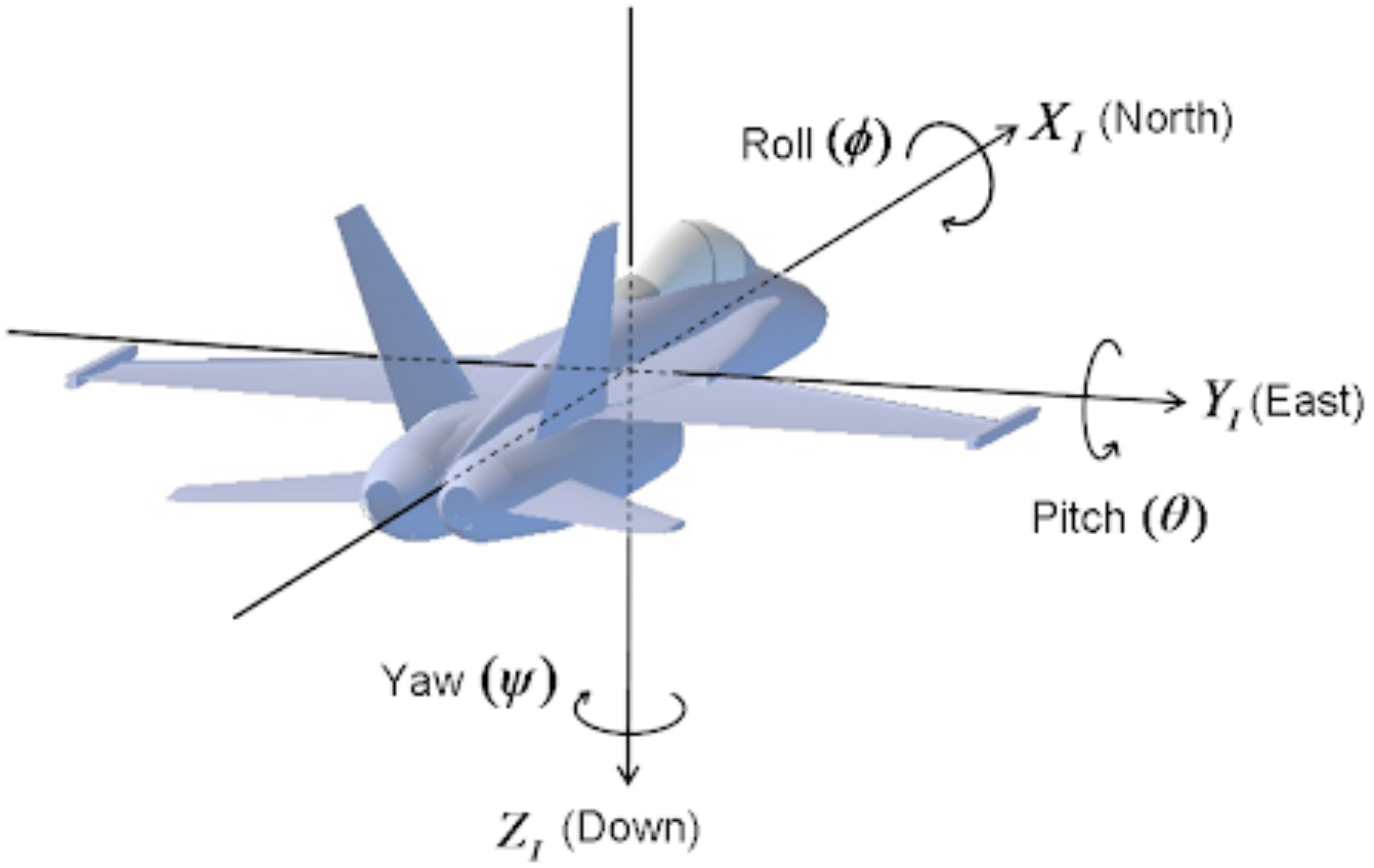
- atan($a_x$ ,$a_z$) returns [-π/2, π/2]
- Instead use atan2($a_x$ ,$a_z$) which returns [-π, π]

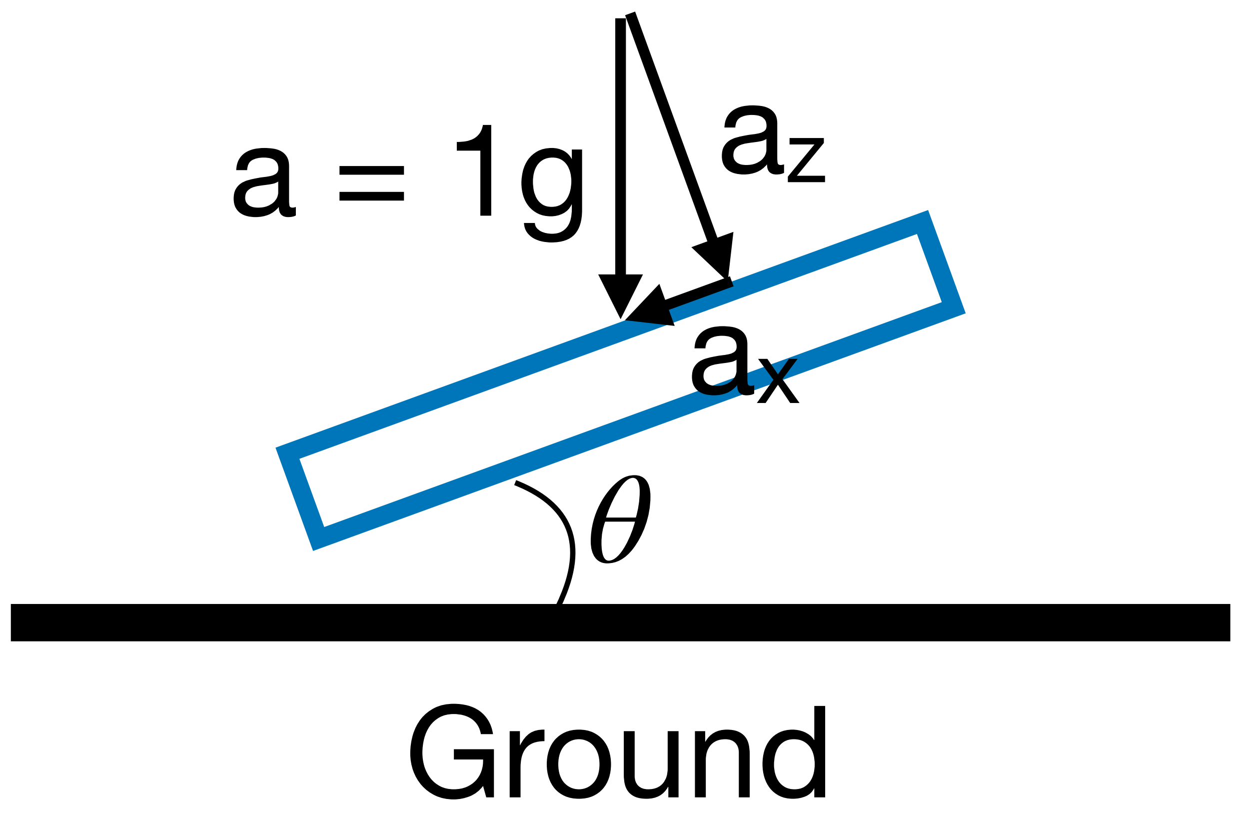


```
float atan2(float x, float y) {
    if (x > 0.0)
        return atan(y/x);
    if (x < 0.0) {
        if (y >= 0.0)
          return (PI + atan(y/x));
        else
          return (-PI + atan(y/x));
        }
    if (y > 0.0) // x == 0
        return PI_ON_TWO;
    if (y < 0.0)
        return -PI_ON_TWO;
    return 0.0; // Should be undefined
}
```

# Accelerometer

## Roll, Pitch, ~~Yaw~~




$a_z = 1g\cos(\theta)$

$a_x = 1g\sin(\theta)$

$\theta = \text{atan2}(a_x, a_z)$

$\phi = \text{atan2}(a_y, a_z)$

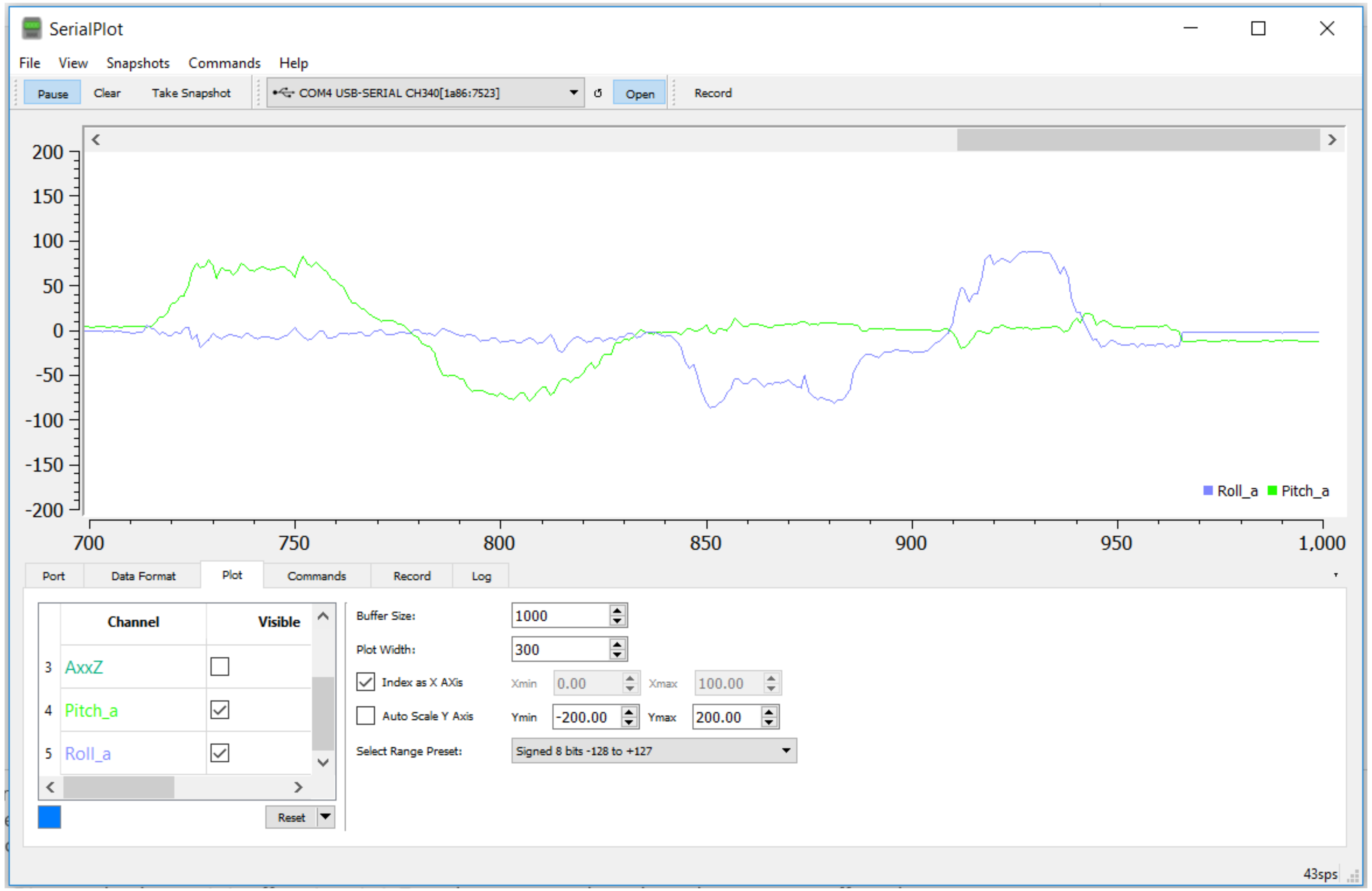
**Can we estimate yaw?**

# Accelerometer

## Pitch and Roll

$$\theta = \mathrm{atan2}(a_x, a_z)$$

$$\phi = \mathrm{atan2}(a_y, a_z)$$



# Accelerometer

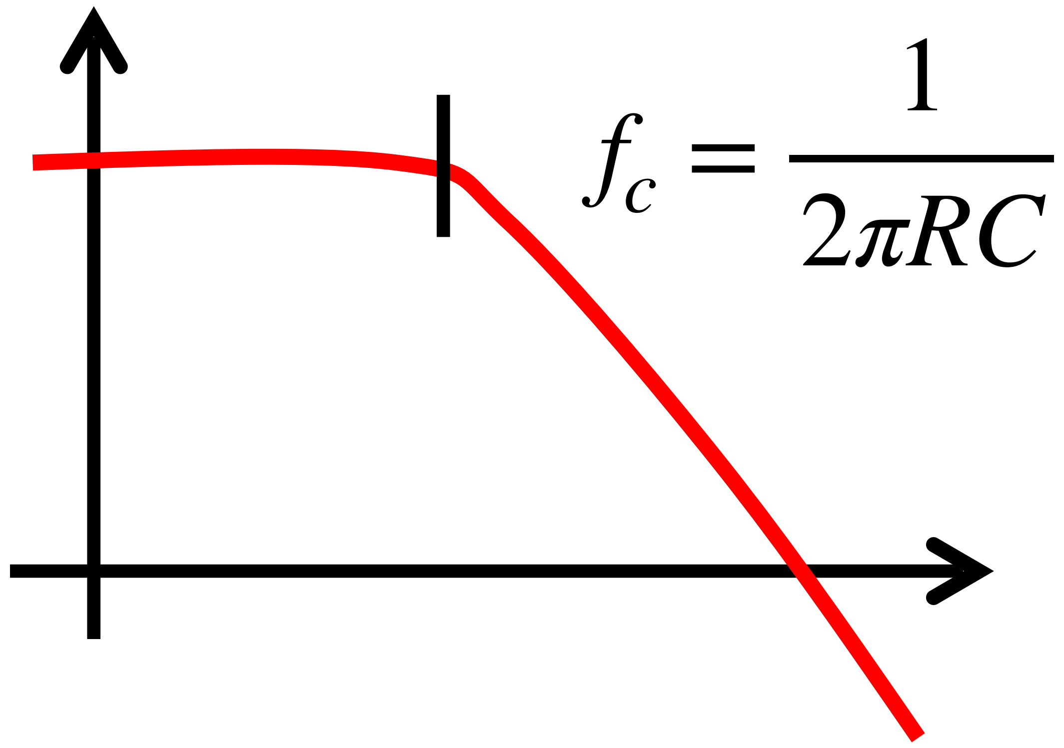
## Roll and Pitch

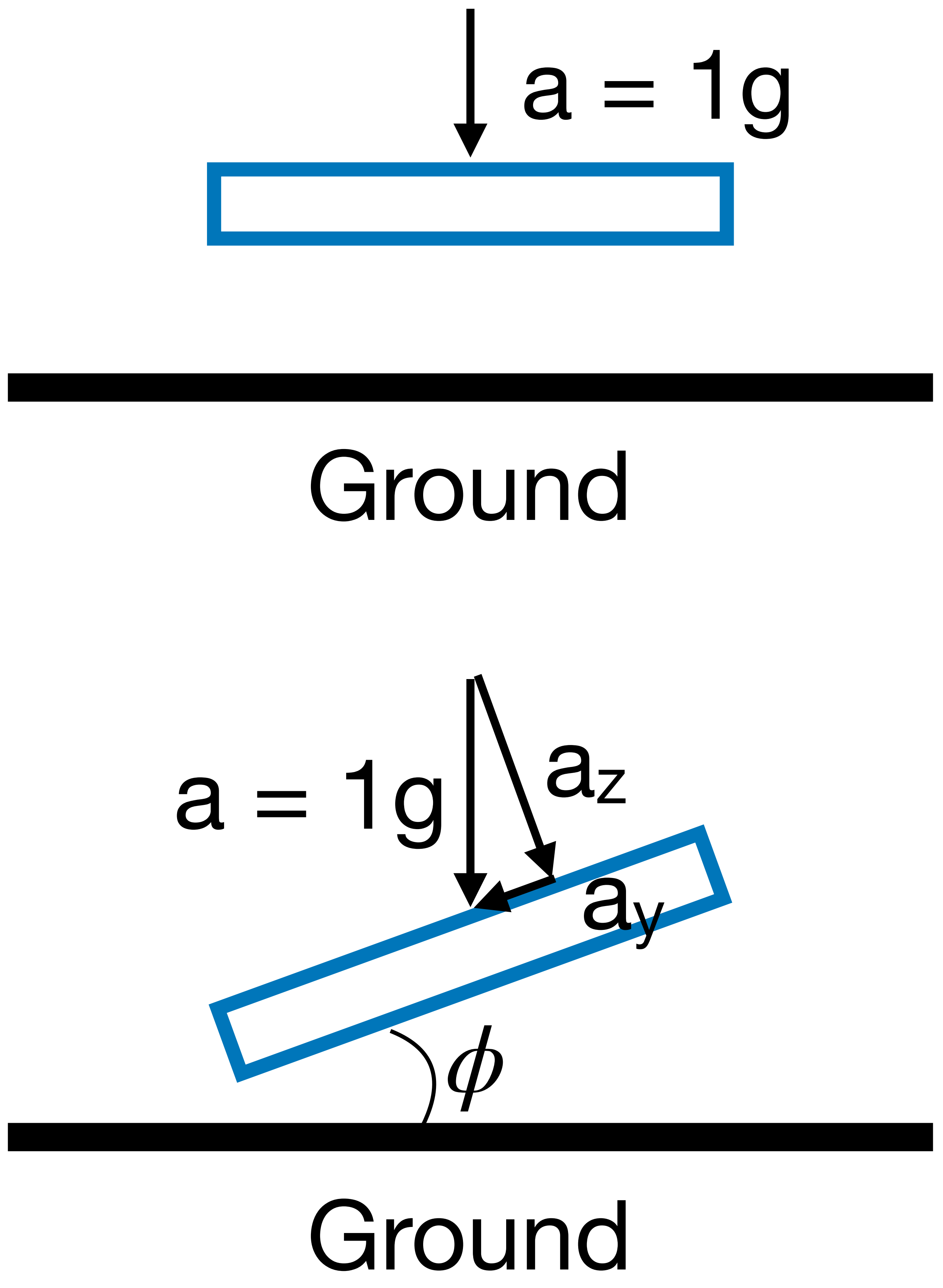
- Good (very accurate on average) vs. bad (noisy)
- Low pass filter

$$\theta_{LPF}[n] = \alpha * \theta_{RAW} + (1 - \alpha) * \theta_{LPF}[n-1]$$
$$\theta_{LPF}[n-1] = \theta_{LPF}[n]$$

- Think of your frequency like an RC low-pass filter:

$$\alpha = \frac{T}{T + RC}$$



# Accelerometer

## Dead Reckoning

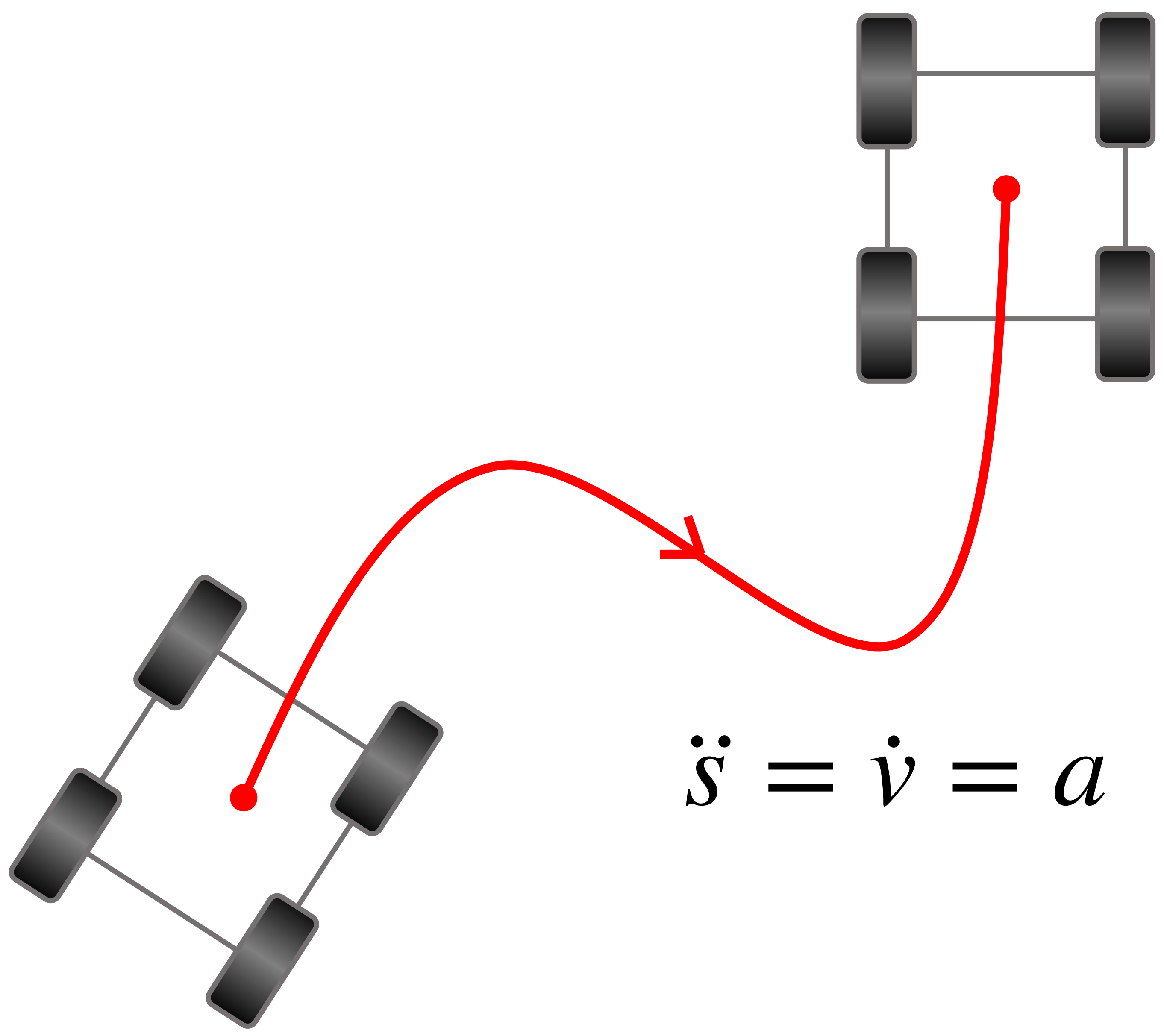
- Use the accelerometer to do dead reckoning?

$$v = \int a$$

$$s = \iint a$$

```
v[k+1] = v[k] + a[k]*dt
s[k+1] = s[k] + v[k]*dt
```

- If you do this at home, remember unit conversion! Accelerometer output is in mg (1g ~9.81 m/s$^2$)



$$\ddot{s} = \dot{v} = a$$

# Accelerometer

## Dead Reckoning

- Issue: Distinguishing sensor acceleration from gravity
  - **Solution 1**: Calibrate the offset
  - **Solution 2**: Low pass filter
  - **Solution 3**: Minimum signal cut-off

Errors only accumulate, and they grow fast!

www.chrobotics.com

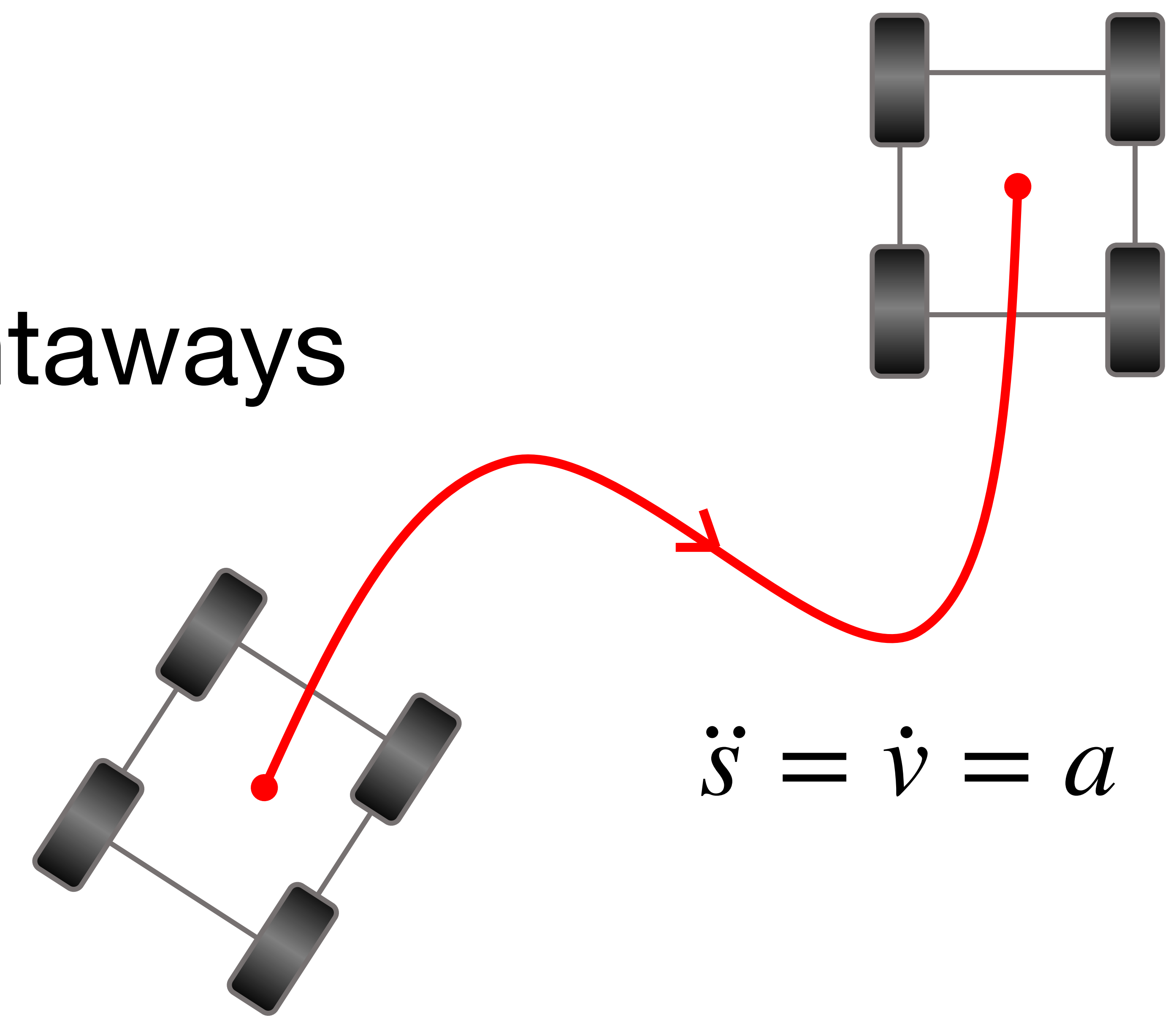
| Angle Error (degrees) | Acceleration Error (m/s/s) | Velocity Error (m/s) at 10 seconds | Position Error (m) at 10 seconds | Position Error (m) at 1 minute | Position Error (m) at 10 minutes | Position Error (m) at 1 hour |
|---|---|---|---|---|---|---|
| 0.1 | 0.017 | 0.17 | 1.7 | 61.2 | 6120 | 220 e 3 |
| 0.5 | 0.086 | 0.86 | 8.6 | 309.6 | 30960 | 1.1 e 6 |
| 1.0 | 0.17 | 1.7 | 17 | 612 | 61200 | 2.2 e 6 |
| 1.5 | 0.256 | 2.56 | 25.6 | 921.6 | 92160 | 3.3 e 6 |
| 2.0 | 0.342 | 3.42 | 34.2 | 1231.2 | 123120 | 4.4 e 6 |
| 3.0 | 0.513 | 5.13 | 51.3 | 1846.8 | 184680 | 6.6 e 6 |
| 5.0 | 0.854 | 8.54 | 85.4 | 3074.4 | 307440 | 11 e 6 |

Table 1 - A summary of velocity and position errors caused by attitude estimation error.

# Accelerometer

## Dead Reckoning

- Issue: Distinguishing sensor acceleration from gravity
  - **Solution 1**: Calibrate the offset
  - **Solution 2**: Low pass filter
  - **Solution 3**: Minimum signal cut-off
  - **Solution 4**: Stop periodically and zero the velocity
  - **Solution 5**: Use in combination with ToF on straightaways
  - **Solution 6**: Buy a more expensive IMU
  - etc…



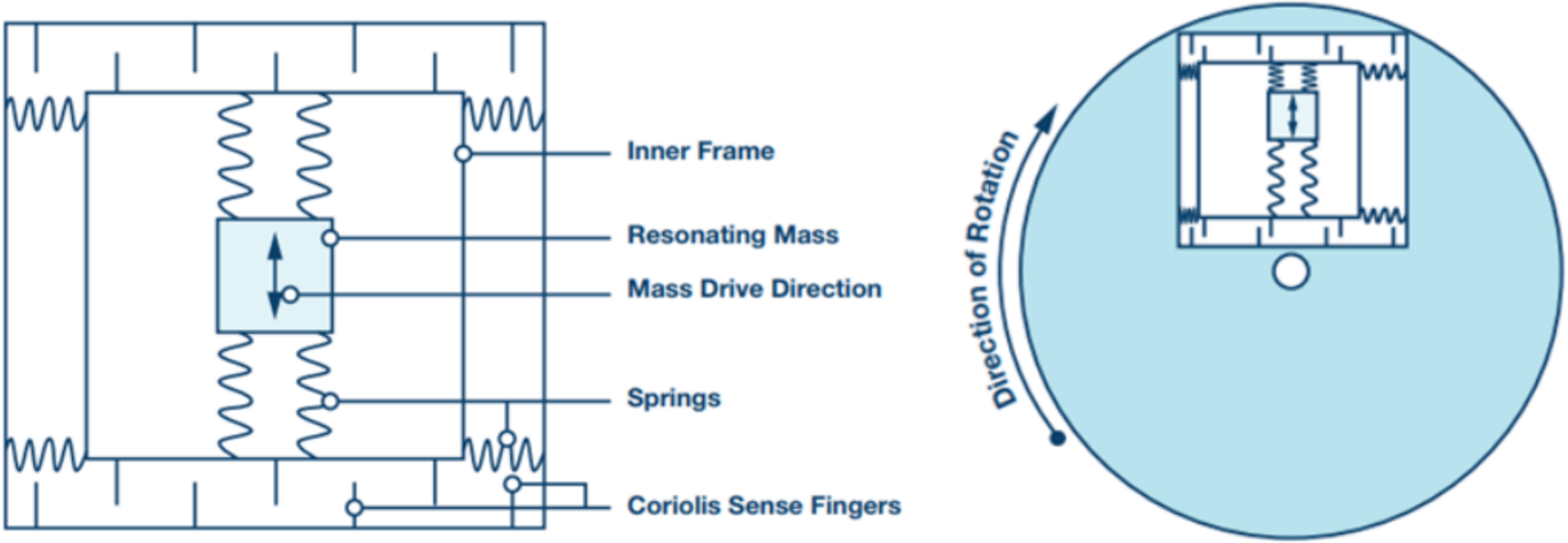
$$\ddot{s} = \dot{v} = a$$

# Gyroscope

# Gyroscope

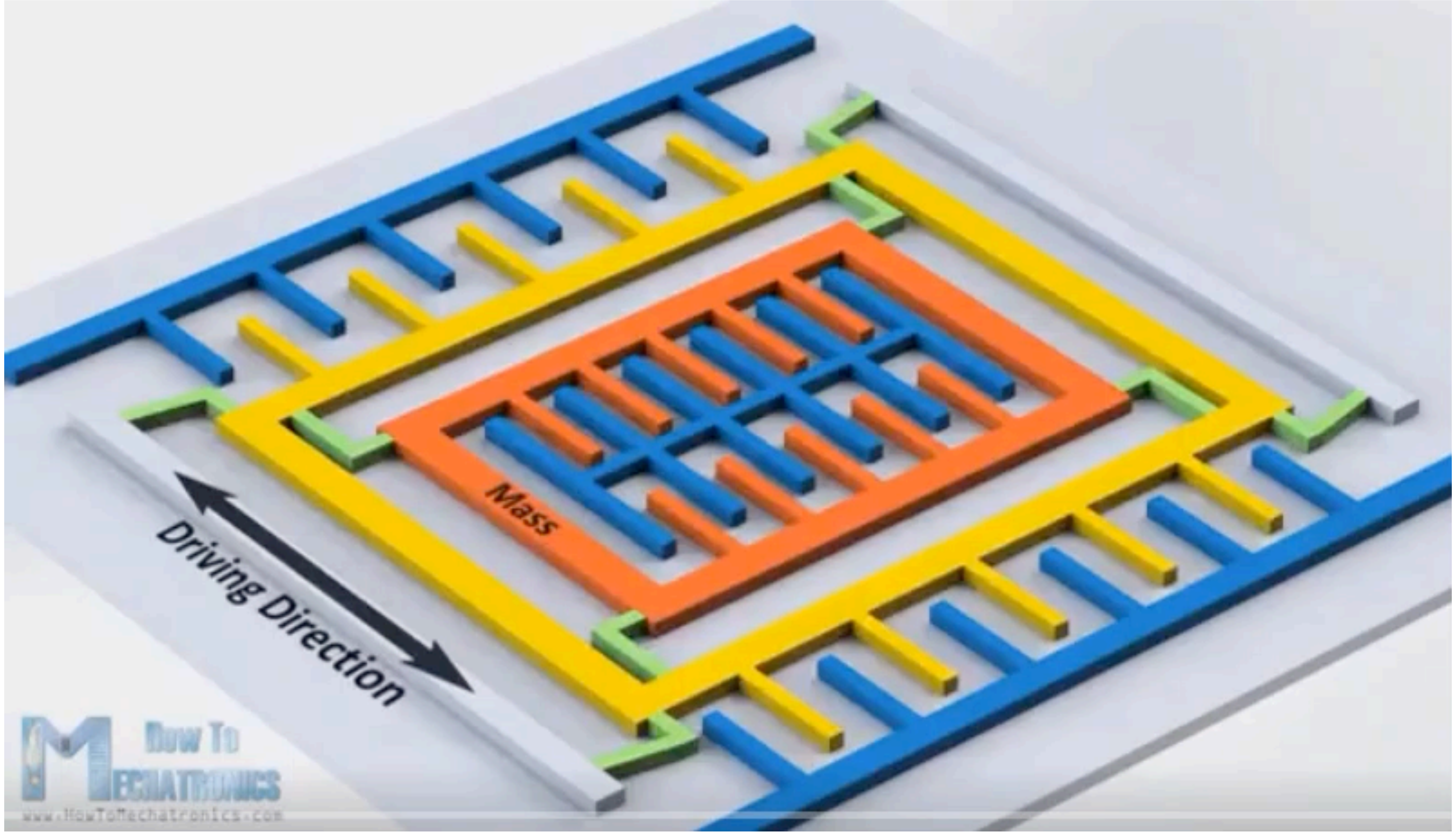
## Measure rate of angular change [°/s]

www.analog.com



# Gyroscope

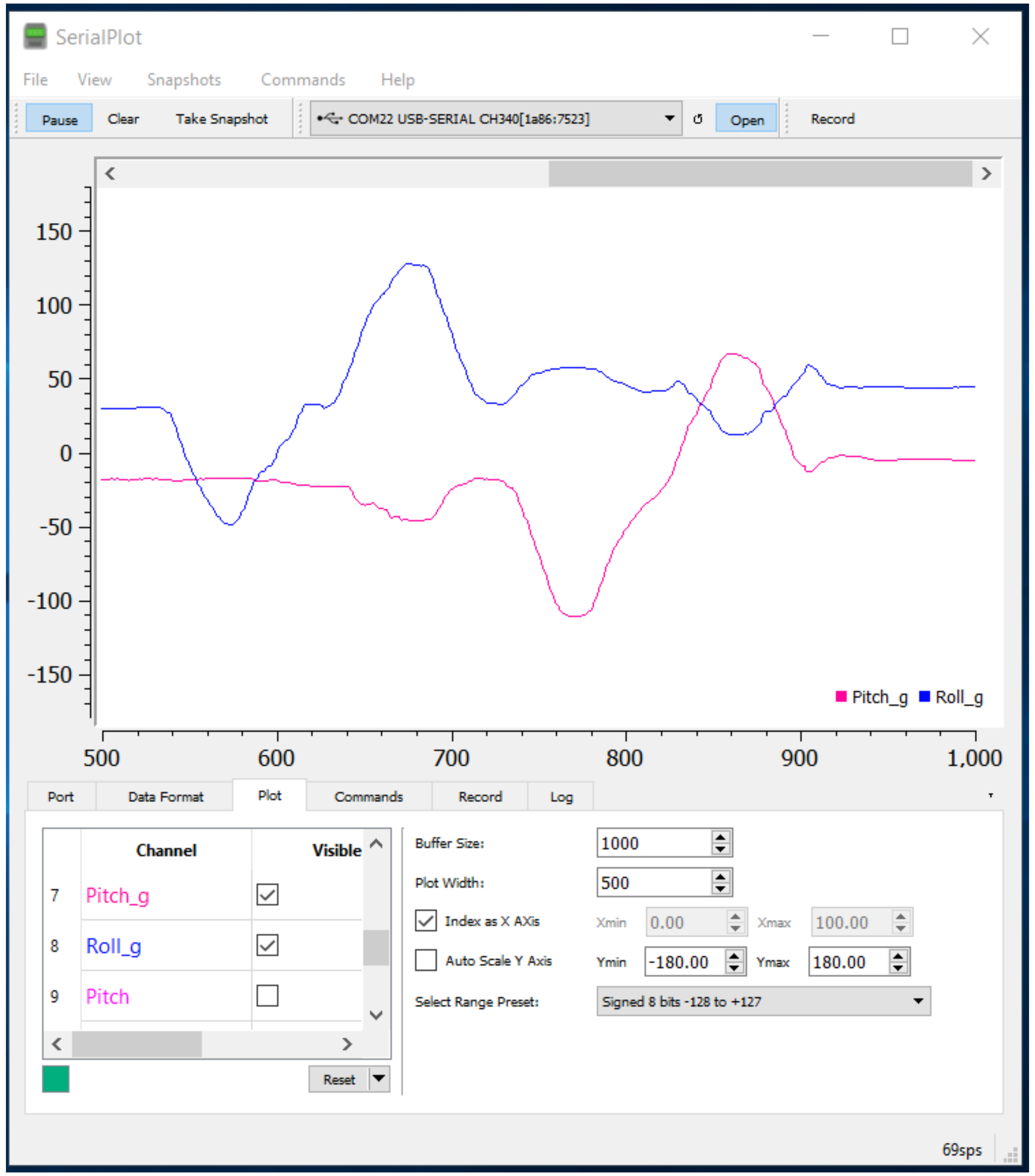
## Measure rate of angular change [º/s]



# Gyroscope

## Measure rate of angular change [°/s]

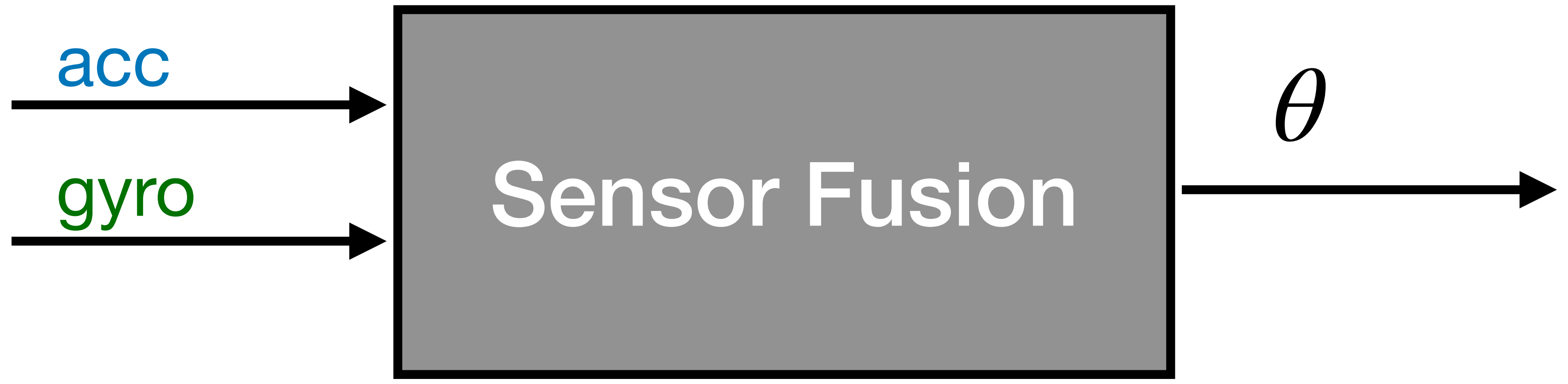
- How to measure angles?
  - $\theta_g = \theta_g + \text{gyro\_reading} \times \text{dt}$
- Drift, but low noise



# Gyroscope

## Measure rate of angular change [º/s]

- How to measure angles?
  - $\theta_g = \theta_g + \text{gyro\_reading} \times \text{dt}$
- Drift, but low noise
- Complementary filter:
  - $\theta = (\theta + \theta_g)(1 - \alpha) + \theta_a \alpha$




**Can we estimate yaw?**

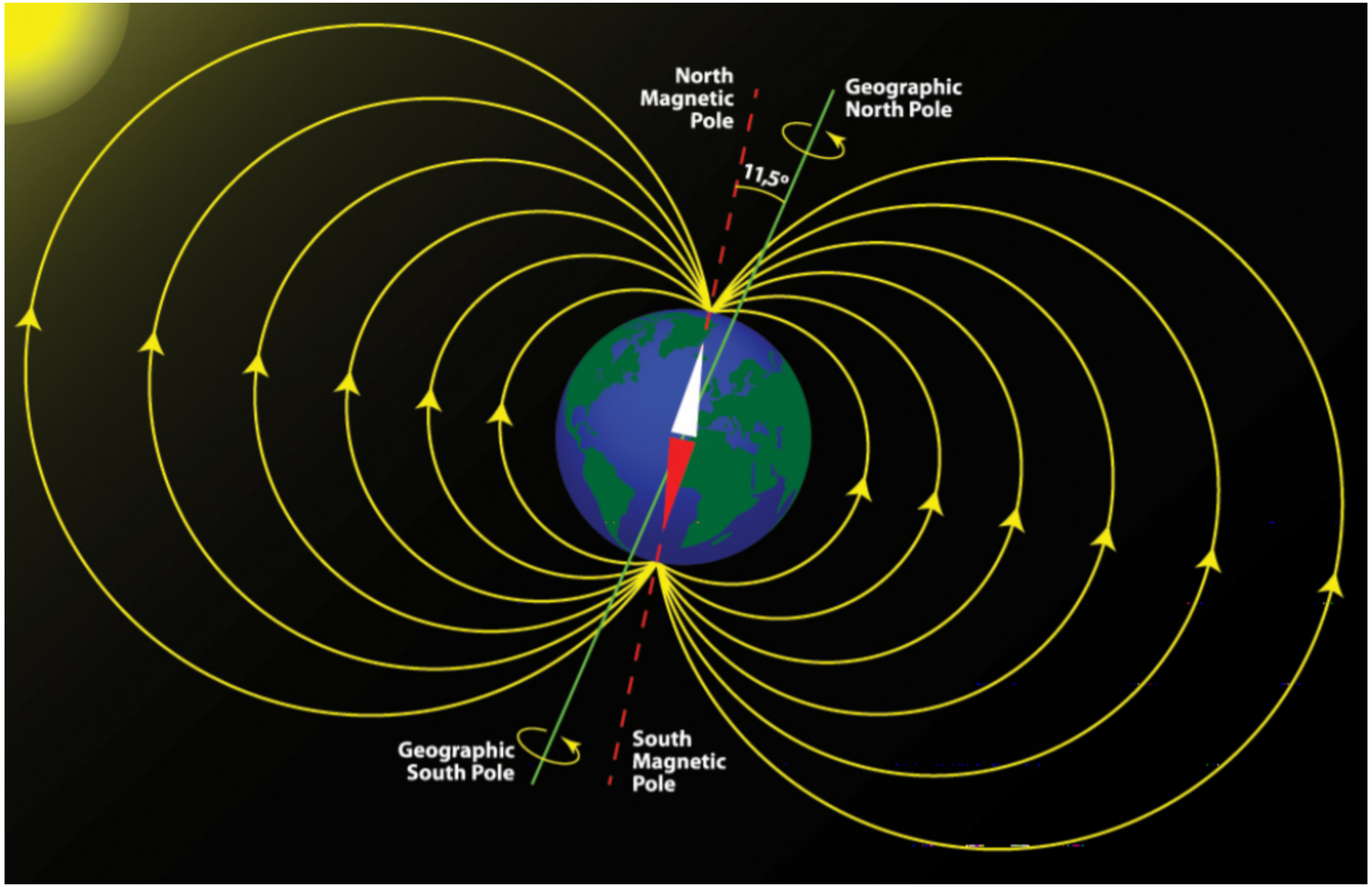
Yes, but no complementary data from the accelerometer

# Magnetometer

# Magnetometer

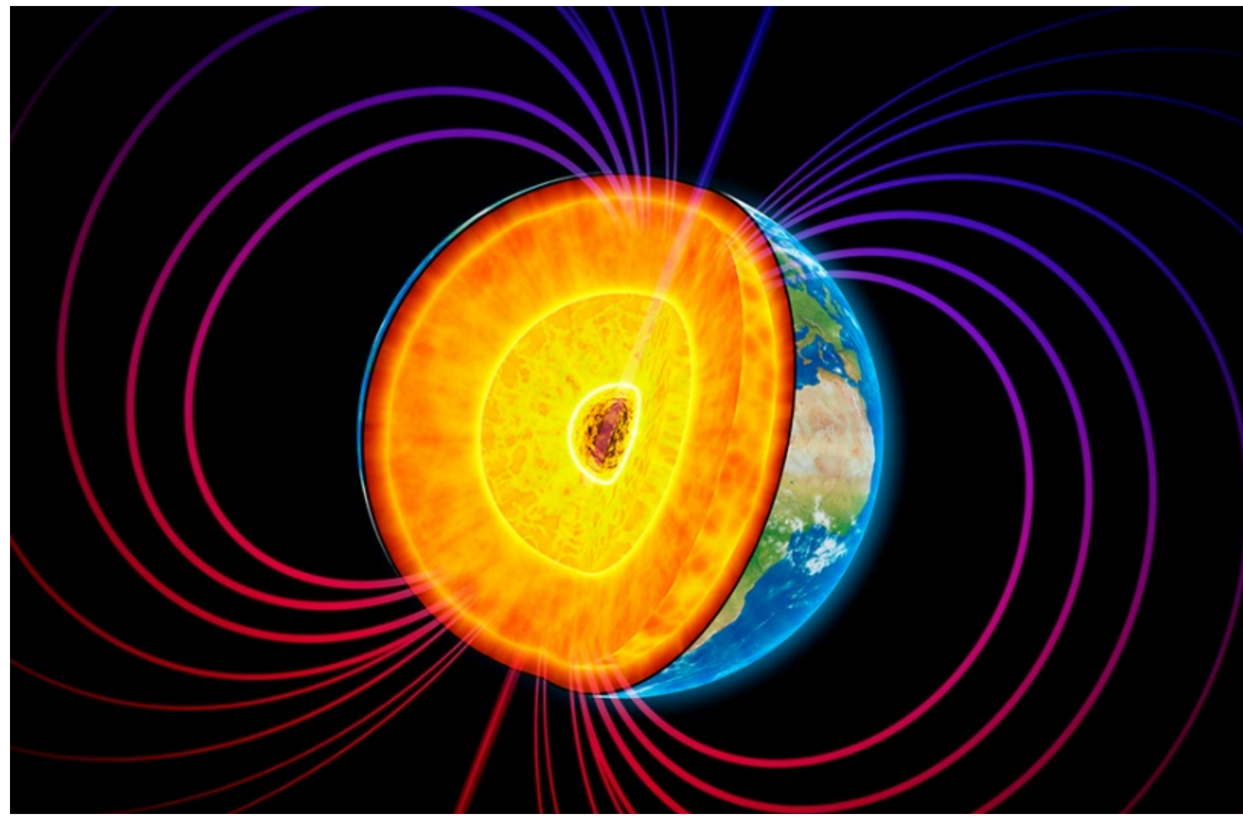
## Measure Earth's magnetic field

- Depends on location, time



# Magnetometer

## Measure Earth's magnetic field



- Depends on location, time
- Distortion due to metal objects or nearby EM fields

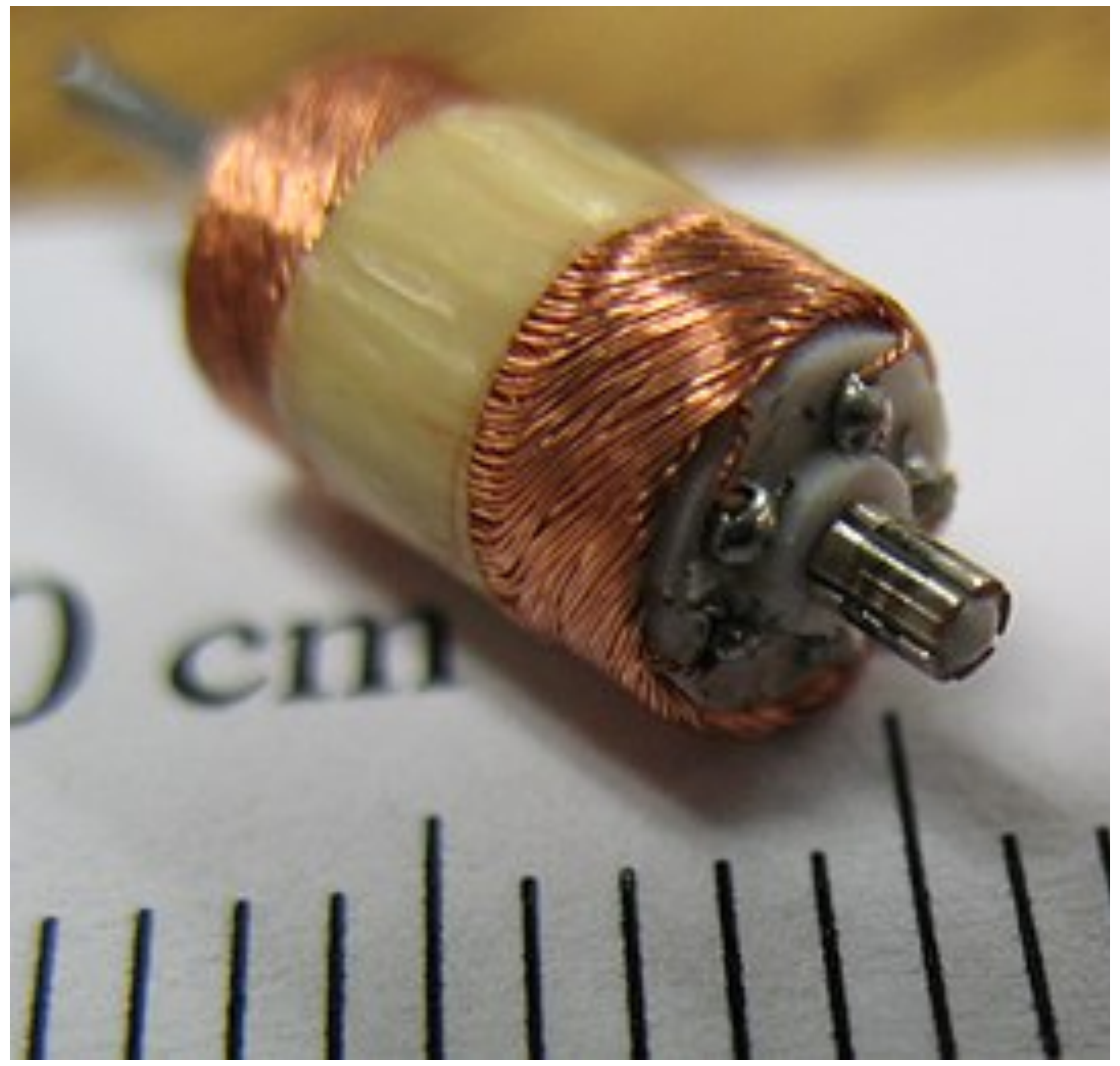

Magnetic Declination

# Magnetometer

## Measure Earth's magnetic field

- Depends on location, time
- Distortion due to metal objects or nearby EM fields

**Examples**



# Magnetometer

## Measure Earth's magnetic field

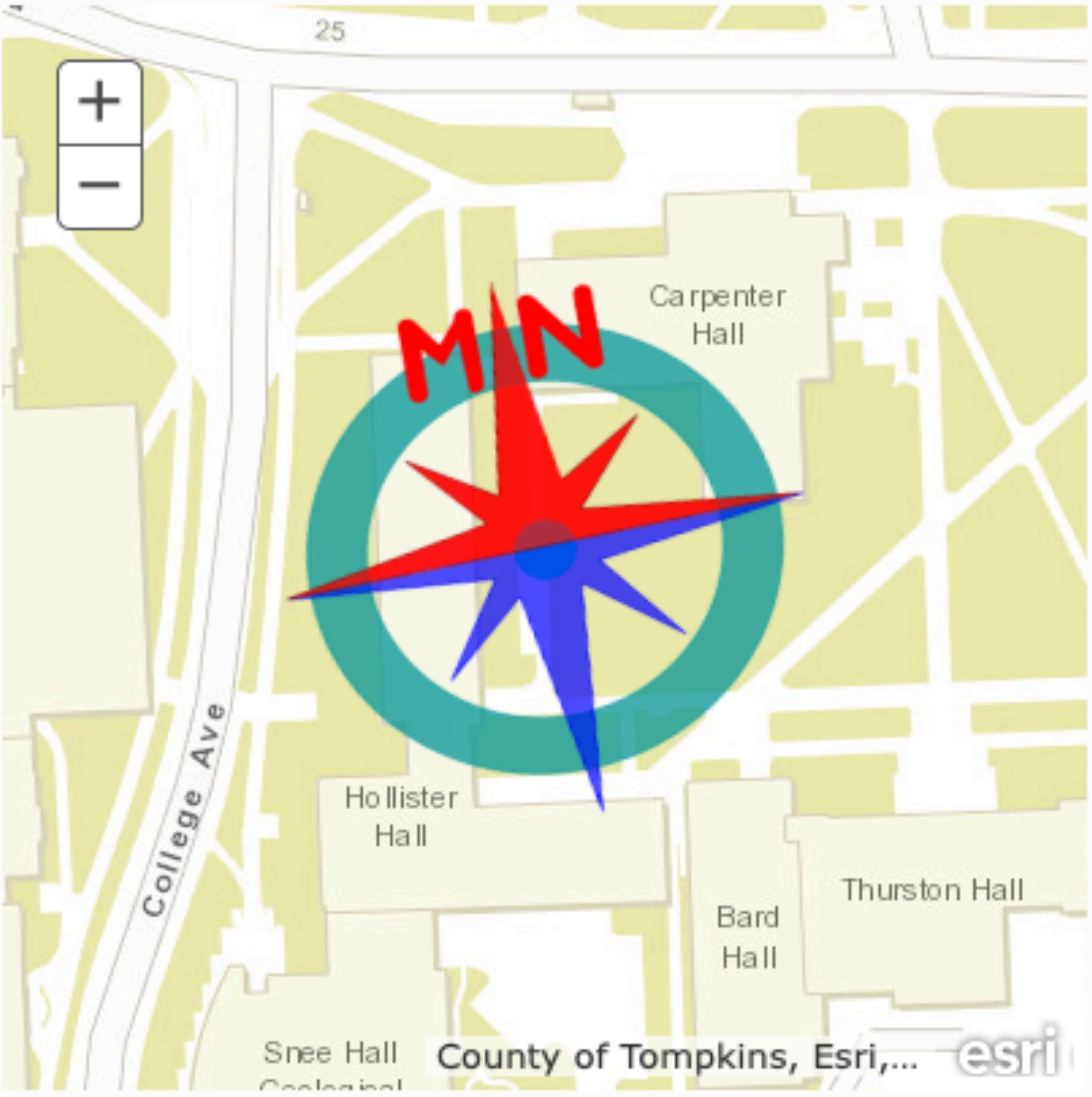
- Magnetic North is along the xmax-axis

| | |
|---|---|
| Model Used: | WMM-2025 |
| Latitude: | 42° 26' 40" N |
| Longitude: | 76° 29' 4" W |
| Elevation: | 0.0 km Mean Sea Level |

| Date | Declination ( + E \| - W ) | Inclination ( + D \| - U ) | Horizontal Intensity | North Comp (+ N \| - S) | East Comp (+ E \| - W) | Vertical Comp (+ D \| - U) | Total Field |
|---|---|---|---|---|---|---|---|
| 2025-01-29 | -11° 37' 27" | 67° 33' 22" | 19,914.2 nT | 19,505.8 nT | -4,012.5 nT | 48,210.8 nT | 52,161.8 nT |
| **Change/year** | 0° 2' 18"/yr | -0° 5' 56"/yr | 41.5 nT/yr | 43.4 nT/yr | 4.7 nT/yr | -135.1 nT/yr | -109.0 nT/yr |
| **Uncertainty** | 0° 23' | 0° 12' | 133 nT | 137 nT | 89 nT | 141 nT | 138 nT |

| | |
|---|---|
| Model Used: | WMM-2025 |
| Latitude: | 42° 26' 40" N |
| Longitude: | 76° 29' 4" W |

| Date | Declination |
|---|---|
| 2025-01-30 | 11° 37' W ± 0° 23' changing by 0° 2' E per year |

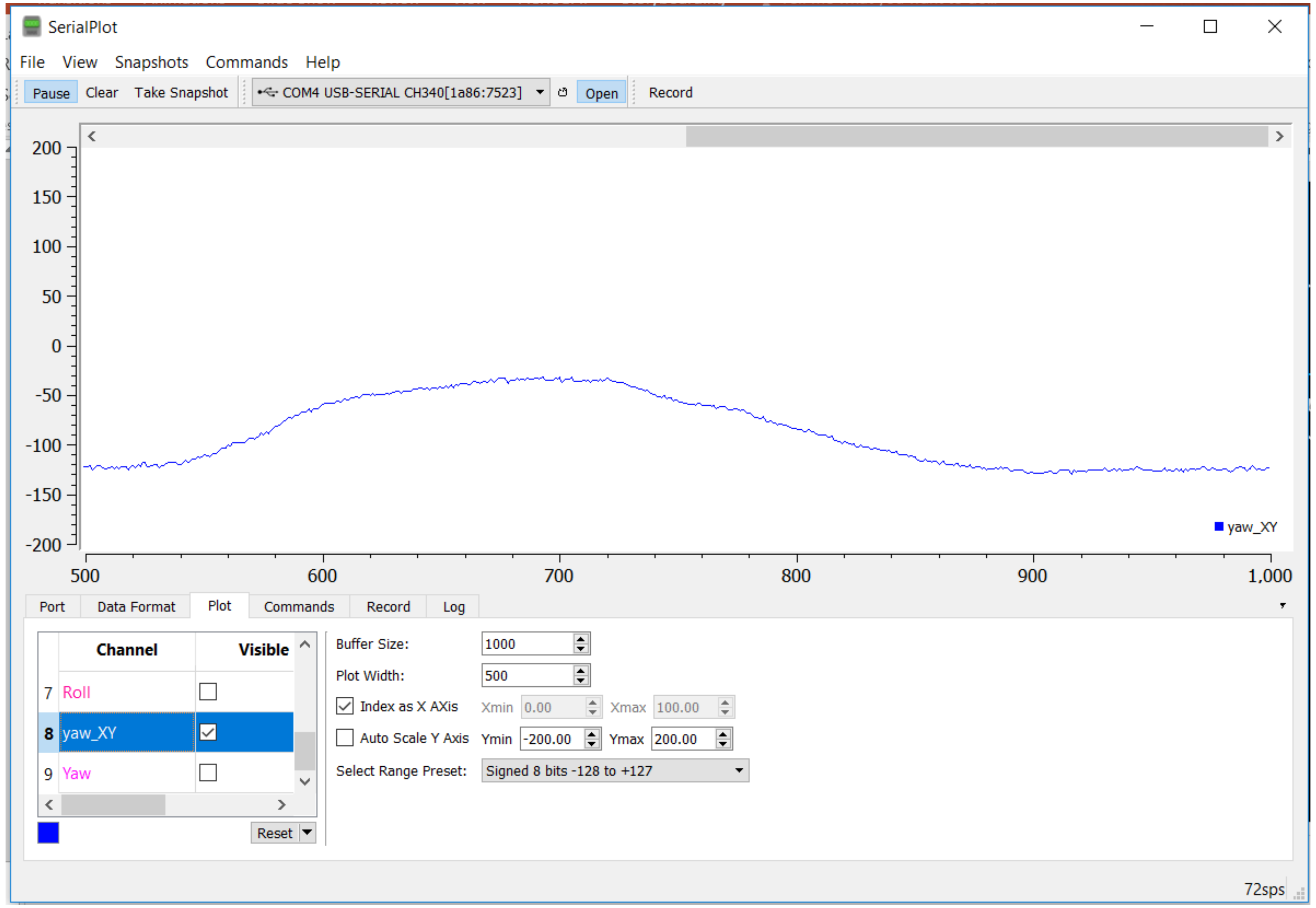


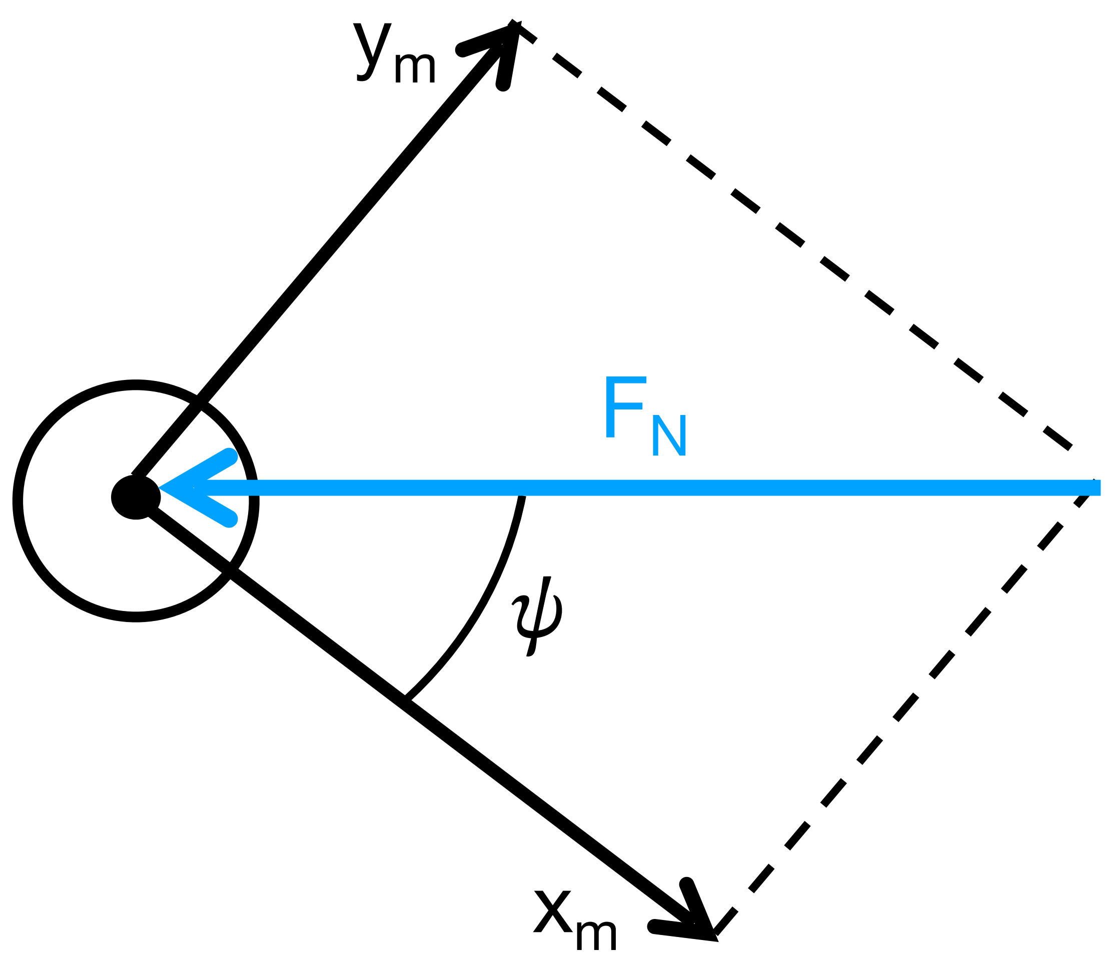

https://www.ngdc.noaa.gov/geomag/calculators/magcalc.shtml#declination

# Magnetometer

## Measure Earth's magnetic field

- $\psi = \text{atan2}(x_m, y_m)$



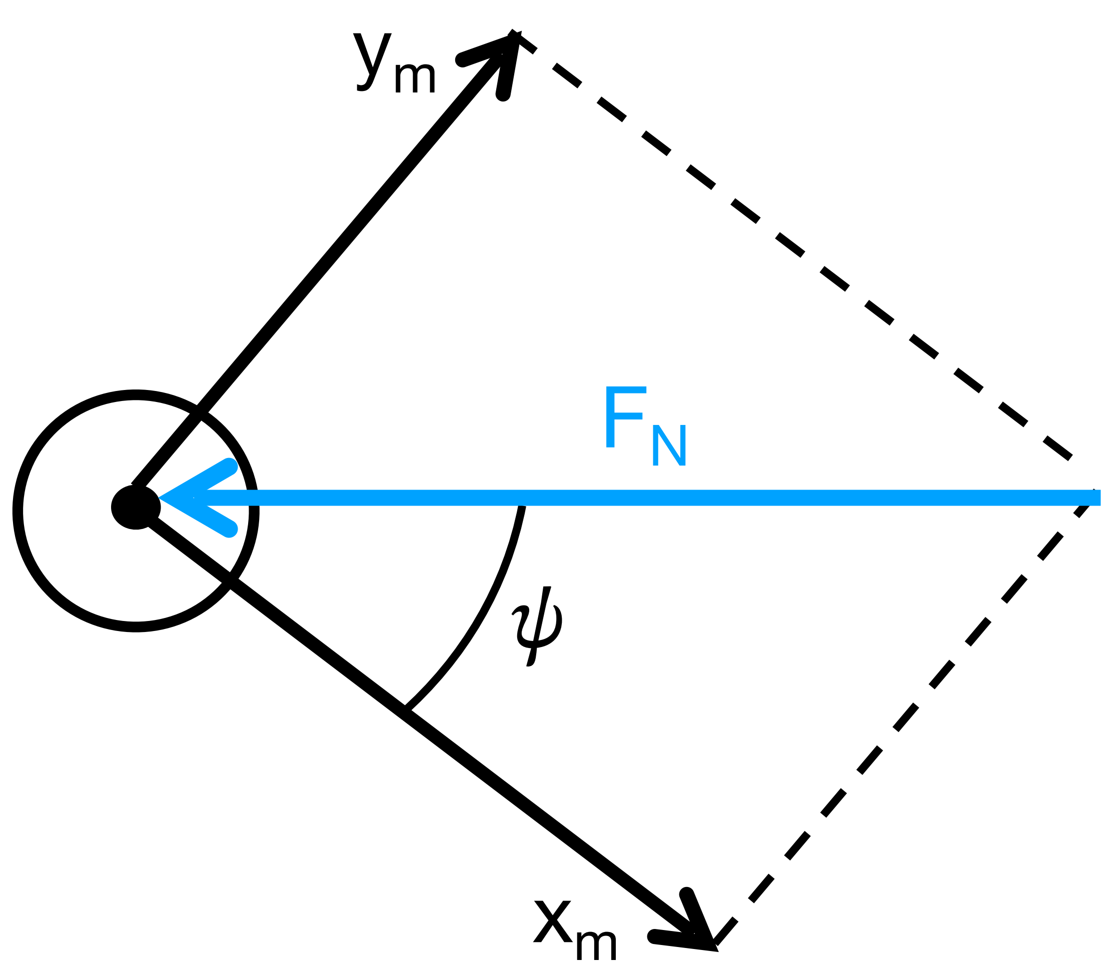
# Magnetometer

## Measure Earth's magnetic field

- $\psi = \text{atan2}(x_m, y_m)$
- How to compensate for tilt? — Fuse data

$$\begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} = R_{x,\phi} R_{y,\theta} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R_{x,\phi}^T R_{y,\theta}^T \begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} = \begin{bmatrix} c_\theta & 0 & c_\phi - s_\theta \\ s_\phi s_\theta & c_\phi & c_\theta s_\phi \\ c_\phi s_\theta & -s_\phi & c_\phi c_\theta \end{bmatrix} \begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix}$$




$$x = y_m \cos(\phi) - z_m \sin(\phi)$$

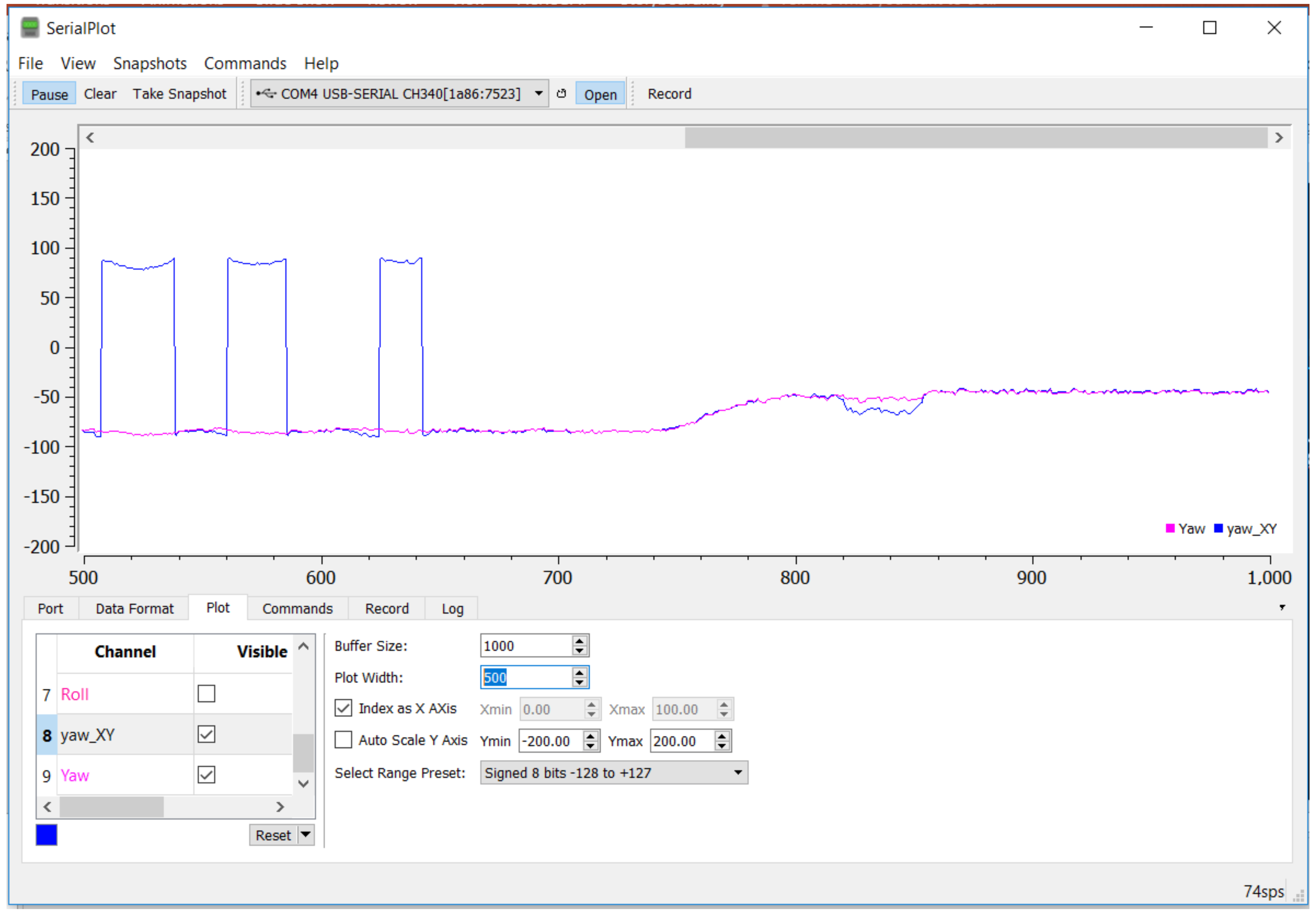
$$y = x_m \cos(\theta) + y_m \sin(\phi)\sin(\theta) + z_m \cos(\phi)\sin(\theta)$$

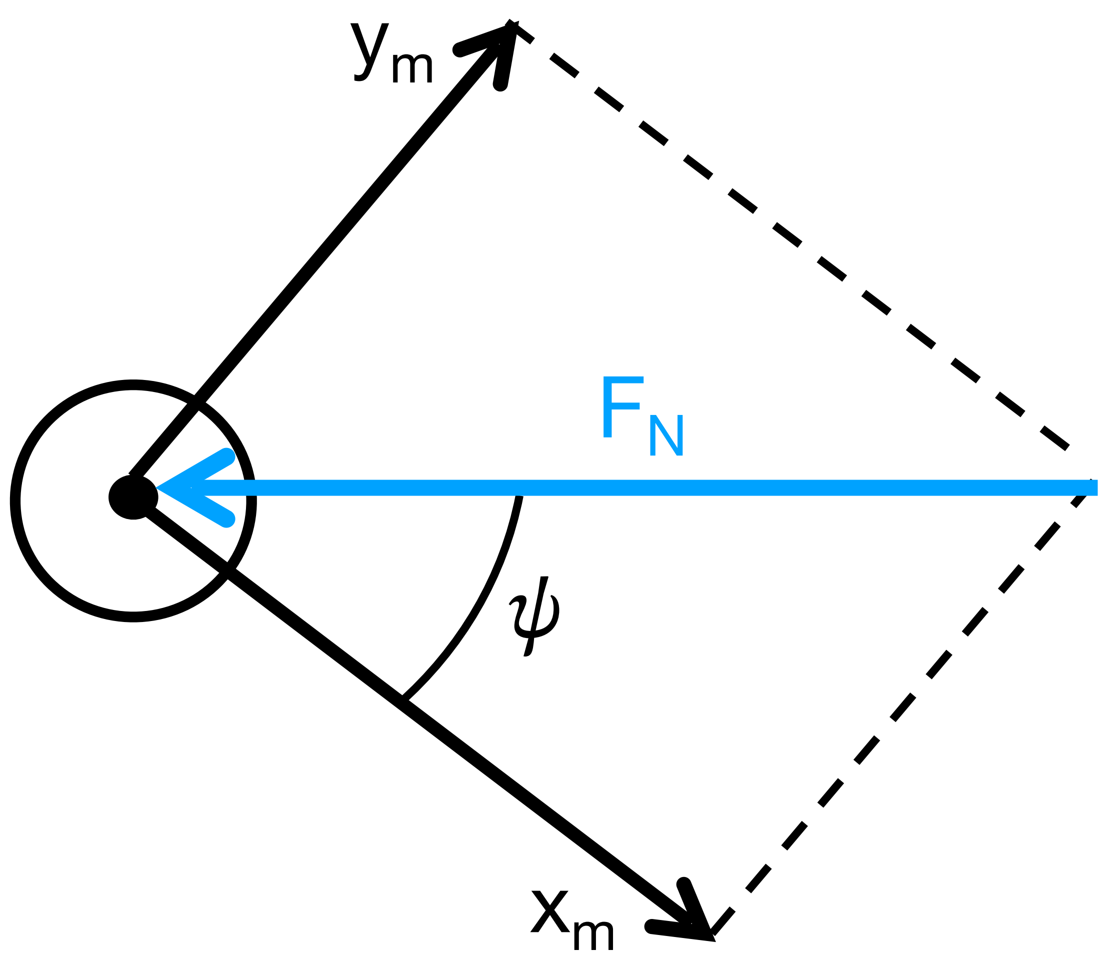
$$\psi = \text{atan2}(x, y)$$

# Magnetometer

## Measure Earth's magnetic field

- $\psi = \text{atan2}(x_m, y_m)$
- How to compensate for tilt? — Fuse data



# Sources and references


- http://www.chrobotics.com/library/accel-position-velocity
- EE 267 Virtual Reality, by Gordon Wetzstein at Stanford University
- Analog.com
- https://toptechboy.com/
- Prof. Kirstin Petersen

# Class Action Items

- If you want to drop the class, please let me know **ASAP** and return your lab kits!
- **January 31st, midnight**: Make a GitHub repository and build your Github page
  - Include: name, photo, a small introduction, and the class number
  - Share **the page link** in the canvas assignment
- **February 4th (8am)** for Lab 401, and **February 5th (8am)** for Labs 402 & 403: Lab 1A and Lab 1B write-ups are due!