ECE 4160/5160
MAE 4960/5960

Prof. Kirstin Hagelskjær Petersen
kirstin@cornell.edu

Fast Robots

Suggested seating (to find a lab partner…)

Tuesday lab (seats 1-5)	Wednesday lab (seats 6-10)	Thursday lab (seats 10 -)

Bluetooth!

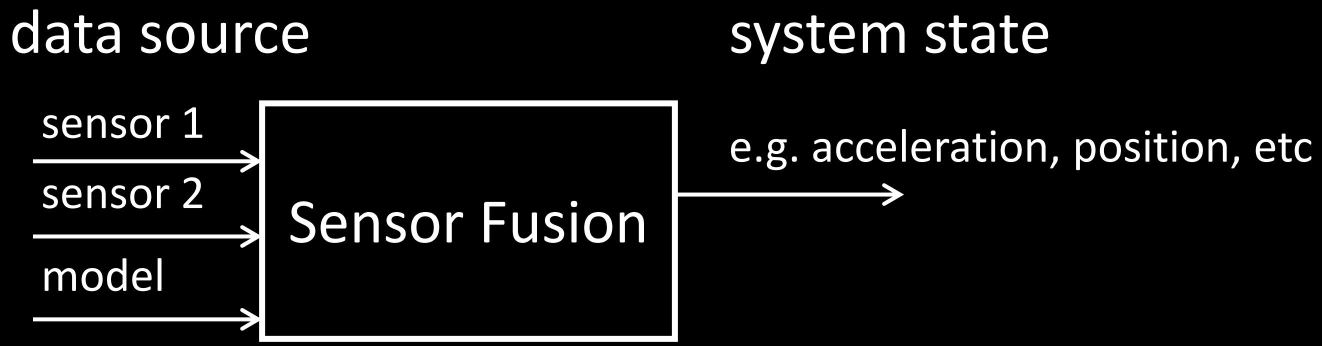
- Couple of confounding issues…
  - Windows 11 / Mac ARM processors don't work well
  - The USB Bluetooth adapters we gave you have issues with USB 3(!)
  - New lab computers
  - New Windows 10 API?
- Possible solutions…
  - Use the built-in Bluetooth on your laptop
  - Use the Bluetooth adapter with USB 2.0
  - We're getting new USB 3.0-compatible Bluetooth adapters for the lab machines
  - Try and try again(!)
- Feel free to drop by any of the lab hours
  - Tu-We-Th 2.40-5.10pm
  - Saturdays 2-6pm
  - …check the lab schedule on cei-lab.github.io/FastRobots-2023/ -> Lab schedule

Prof. Kirstin Hagelskjær Petersen
kirstin@cornell.edu

WHY SENSOR FUSION?

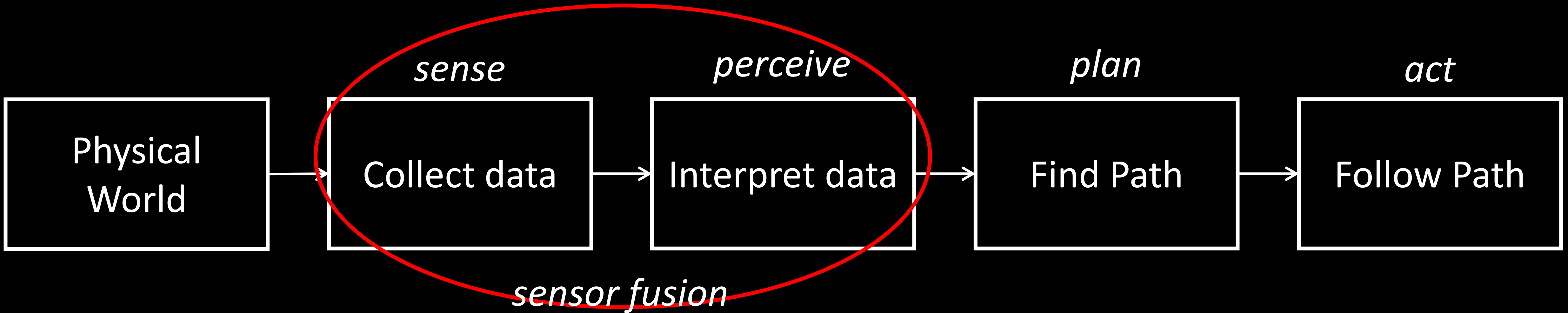
Intro to Sensor Fusion

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



Intro to Sensor Fusion

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

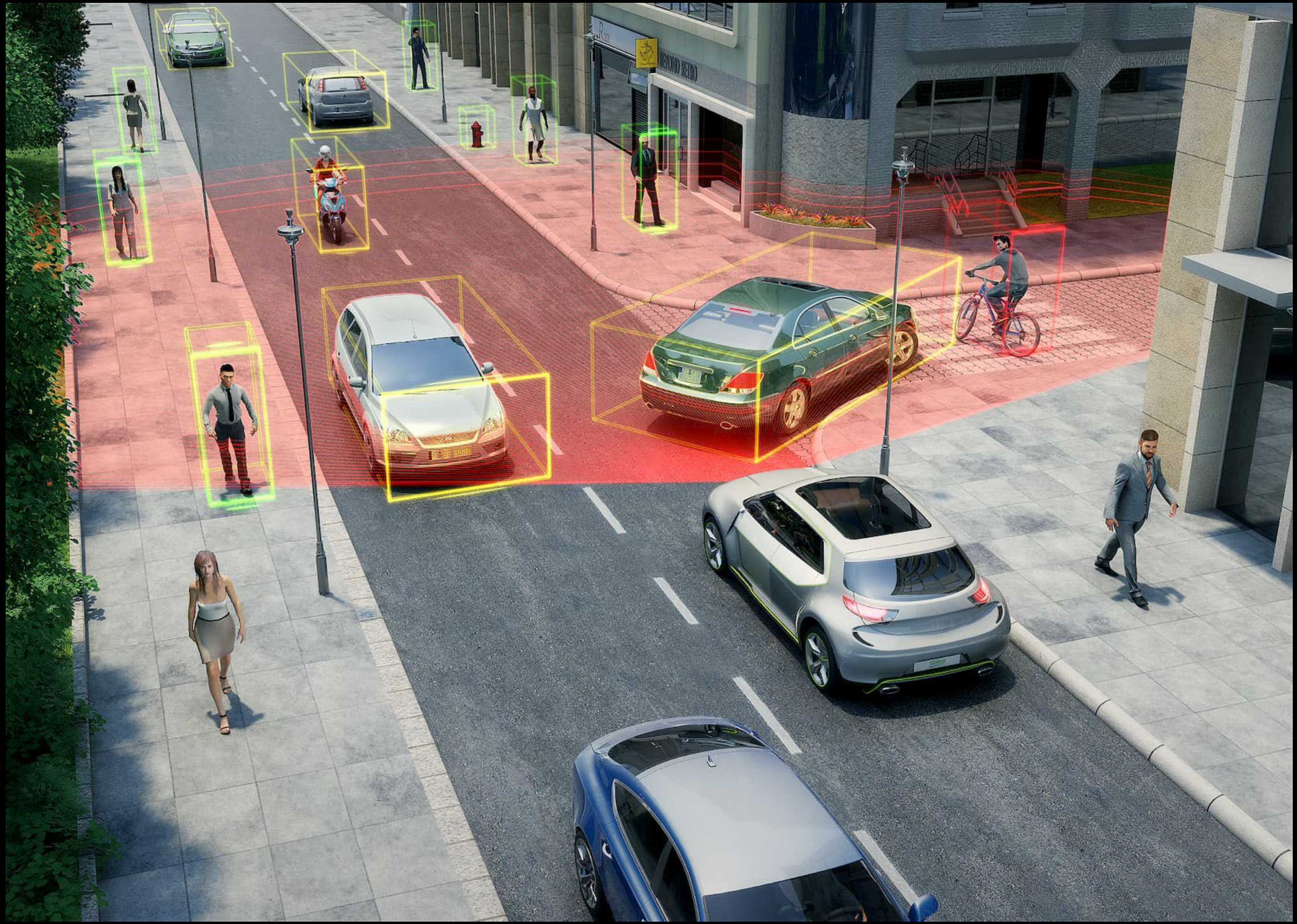


Responsibility:

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

Intro to Sensor Fusion

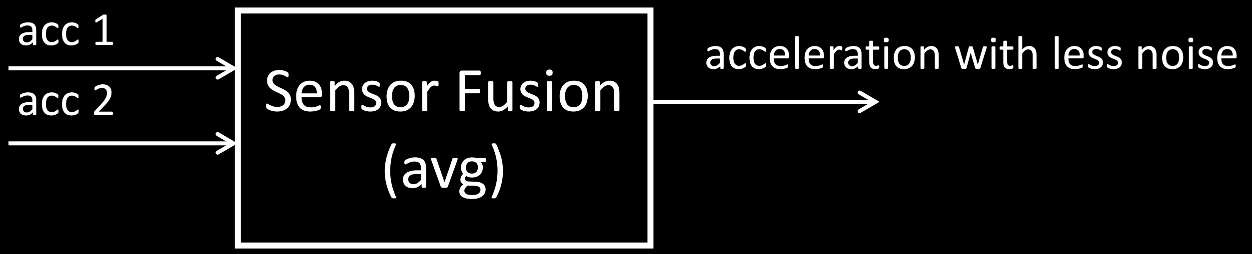
- Example of situational awareness:

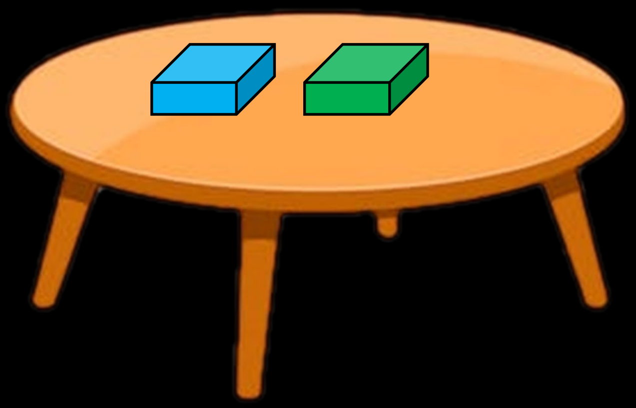


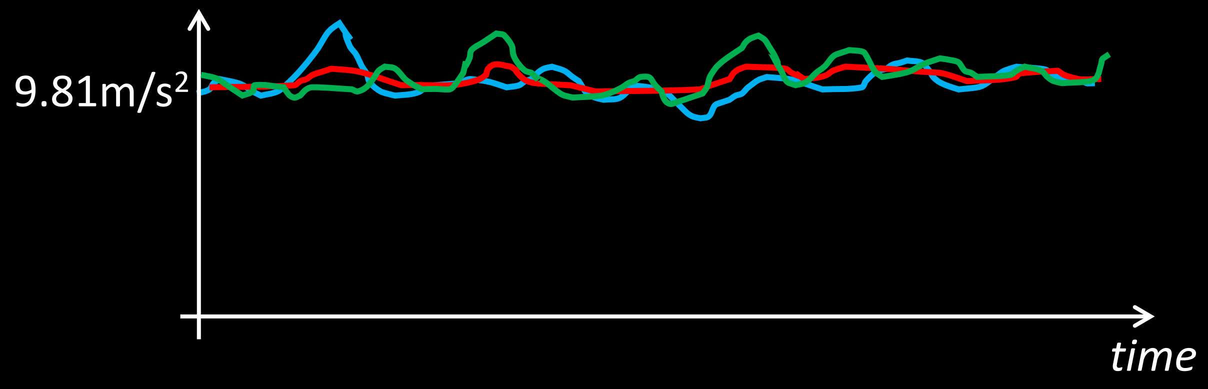
Valeo's LIDAR

Intro to Sensor Fusion

1. Increase the quality of the data
   - Less noise, uncertainty, deviations



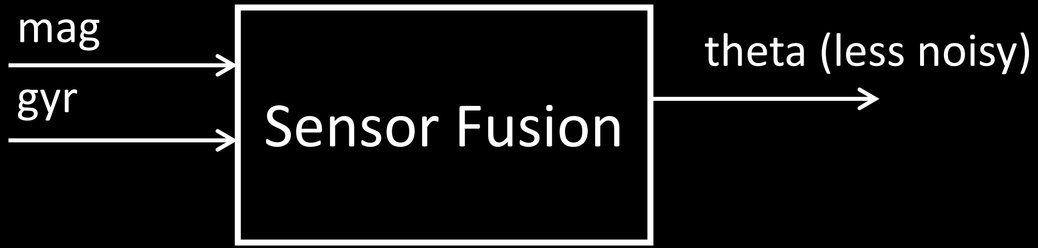


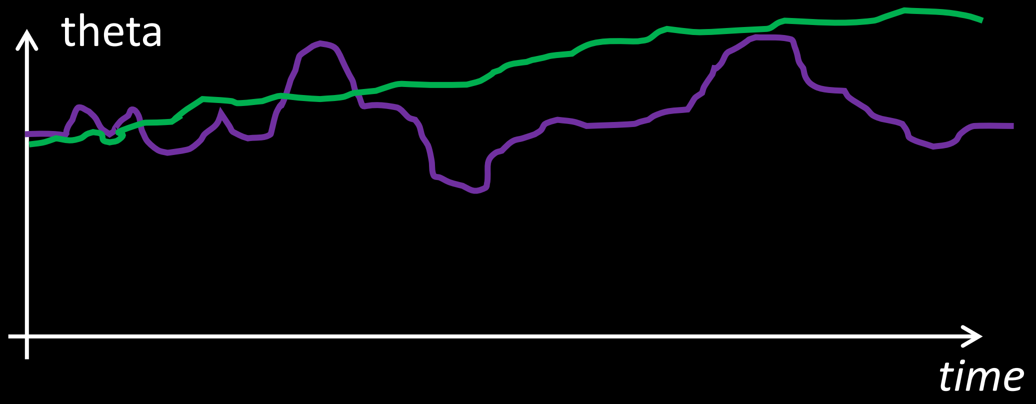


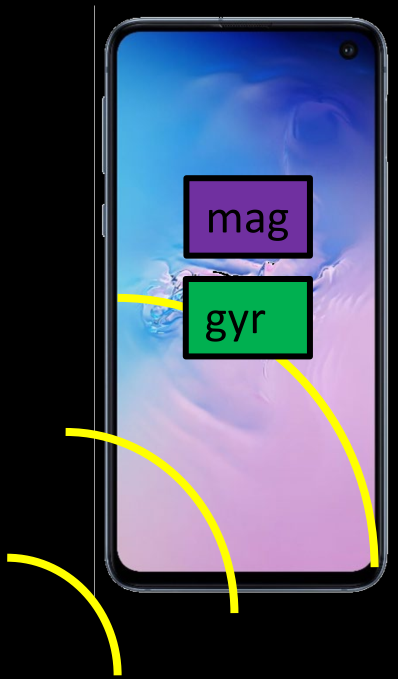
- Adding sensors lowers noise: $n = 1/(\sqrt{N})$
  - 4 identical sensors = ½ noise
    - (Only if the noise is not correlated!)

Intro to Sensor Fusion

1. Increase the quality of the data
   - Less noise, uncertainty, deviations



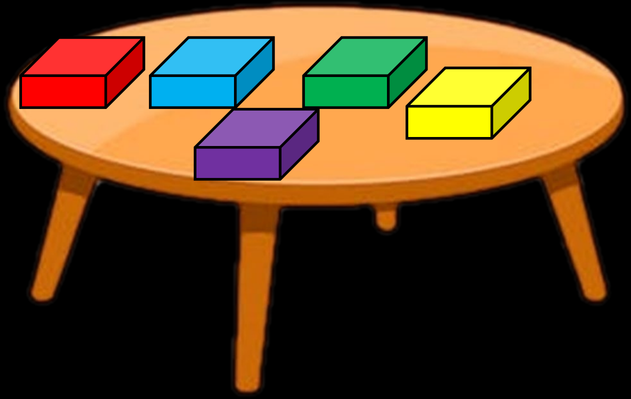


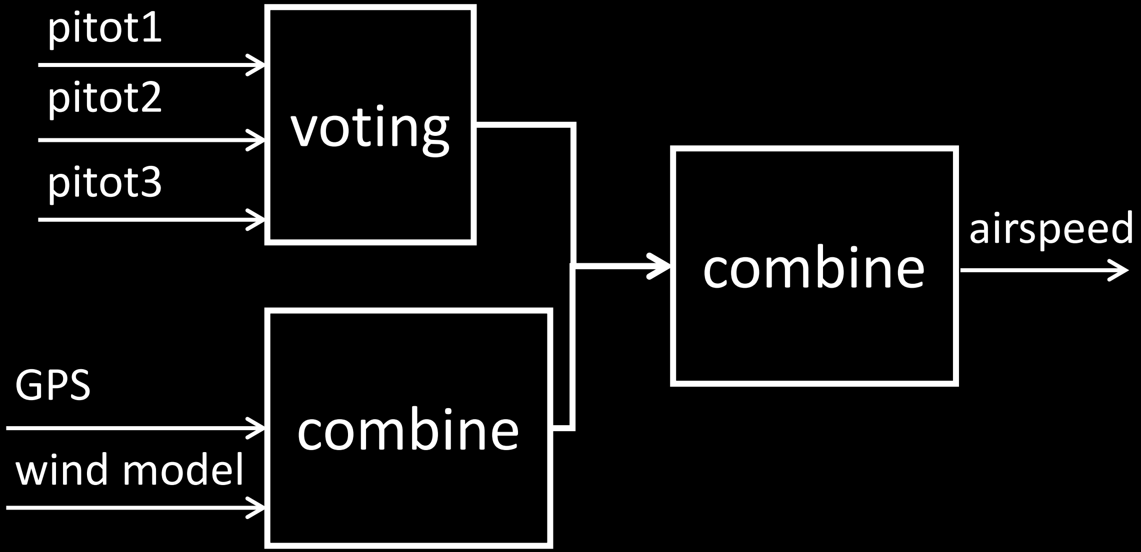


- You can add a 2nd magnetometer to decrease noise
- But some of the noise is correlated
  - Magnetic fields
- Sol 1: Move the sensor away from the magnetic field
- Sol 2: Low pass filter (introduces lag)
- Sol 3: Fuse the mag data with gyr data

Intro to Sensor Fusion

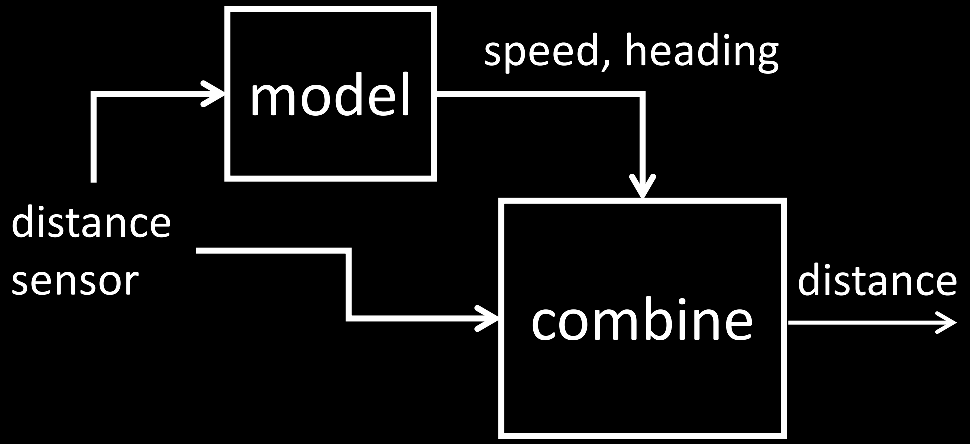
1. Increase the quality of the data
   - Less noise, uncertainty, deviations
2. Increase data reliability





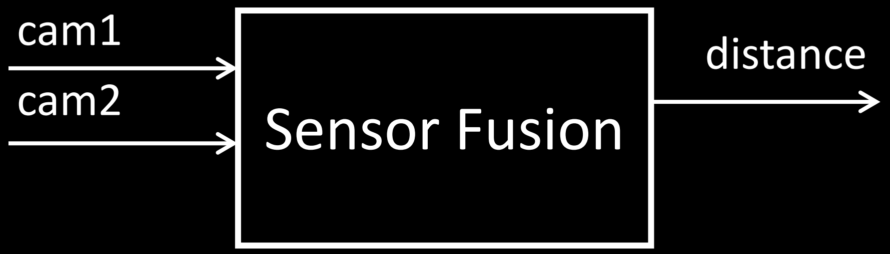
Intro to Sensor Fusion

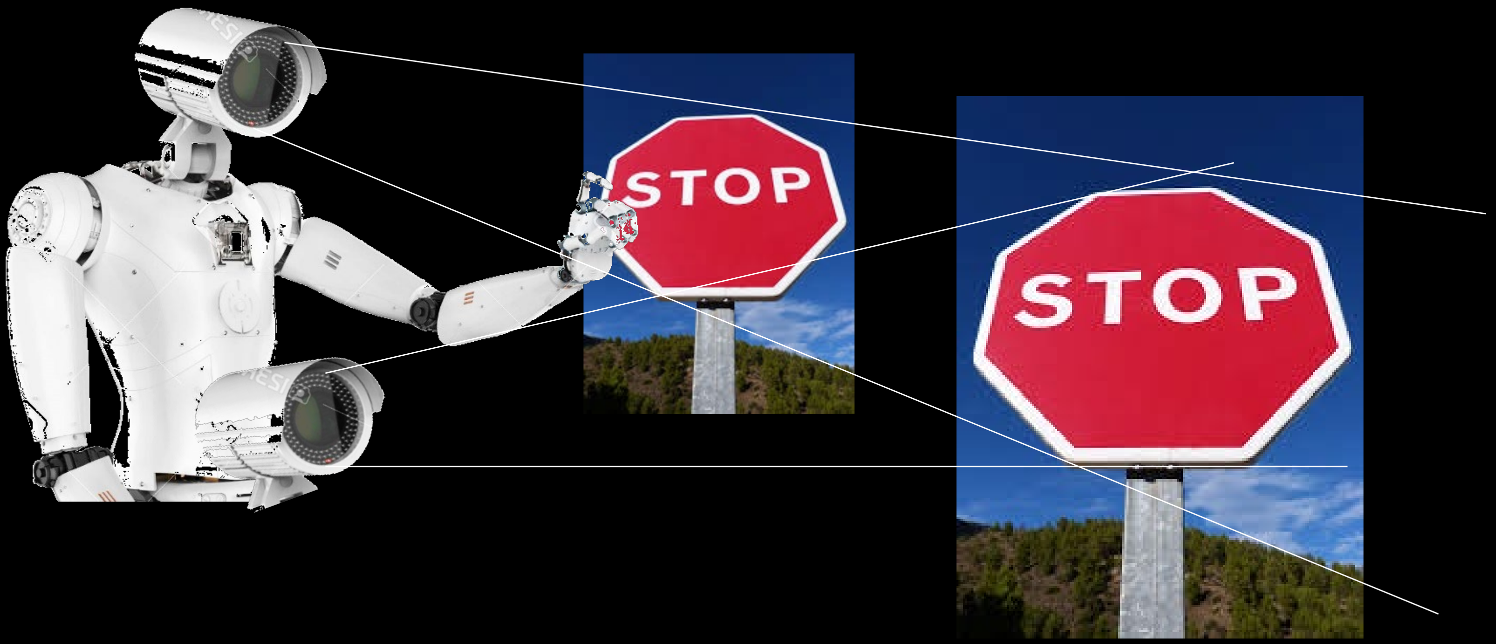
1. Increase the quality of the data
   - Less noise, uncertainty, deviations
2. Increase data reliability



Intro to Sensor Fusion

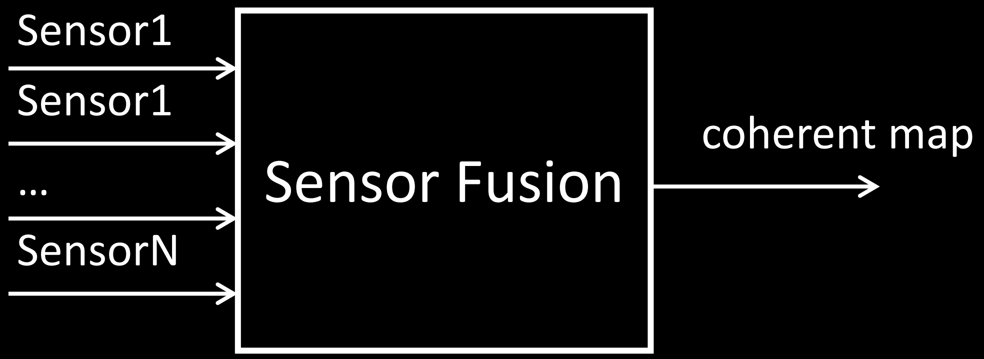
1. Increase the quality of the data
   - Less noise, uncertainty, deviations
2. Increase data reliability
3. You can measure unmeasured states

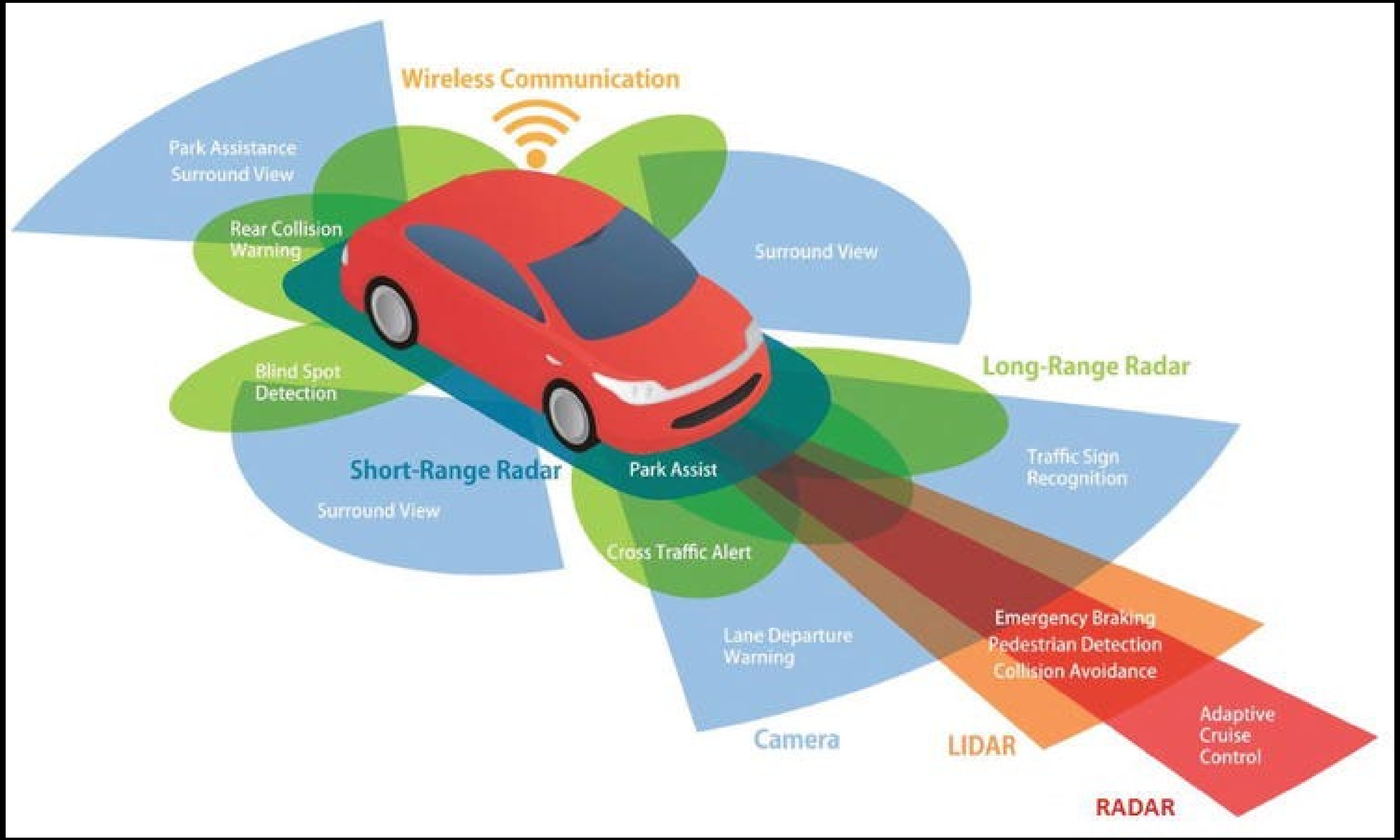




Intro to Sensor Fusion

1. Increase the quality of the data
   - Less noise, uncertainty, deviations
2. Increase data reliability
3. You can measure unmeasured states
4. Increase the coverage area



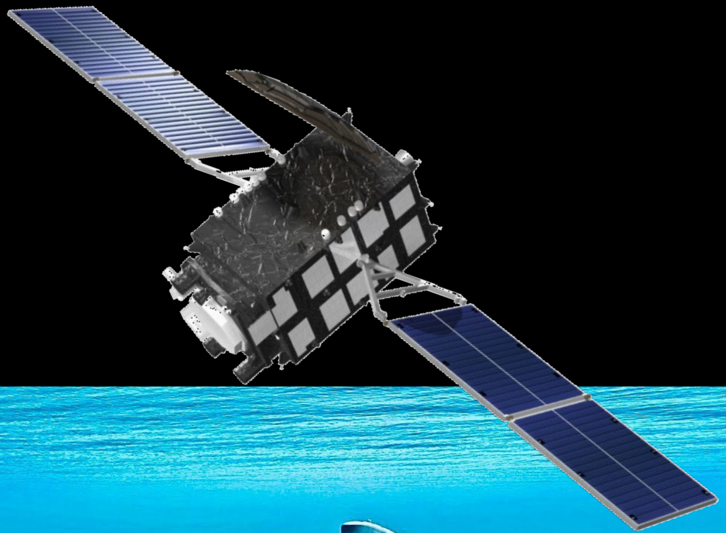


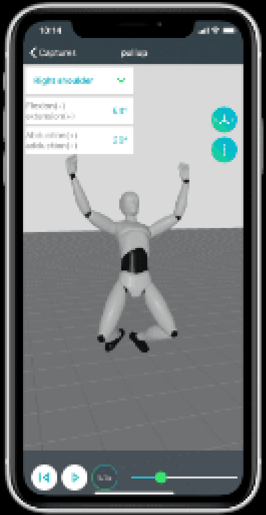
Sources and References

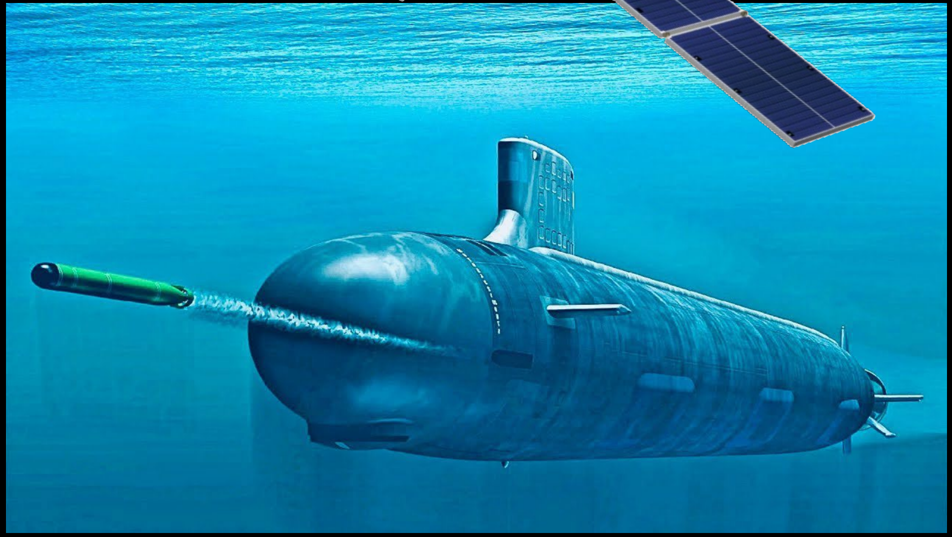
- http://www.cs.cmu.edu/~rasc/Download/AMRobots4.pdf
- https://www.ti.com/lit/ug/sbau305b/sbau305b.pdf?ts=1599417595209&ref_url=https%253A%252F%252Fwww.google.com%252F
- https://hmc.edu/lair/ARW/ARW-Lecture01-Odometry.pdf
- Matlab Tech Talks on Sensor Fusion (https://www.youtube.com/watch?v=6qV3YjFppuc)

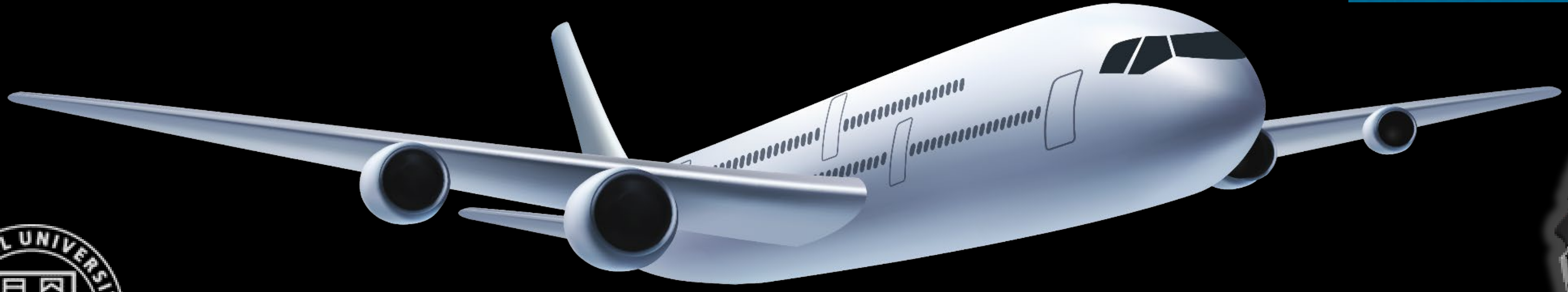
IMU

- Inertial Measurement Unit
  - Data related to orientation, velocity, and gravity









IMU

- Inertial Measurement Unit
- *Accelerometer*
  - Linear acceleration, $a = \dot{v}$ [m/s^2]
- *Gyroscope*
  - Angular velocity, $\omega = \frac{\Delta\theta}{\Delta t}$ [deg/sec]
- *Magnetometer*
  - Magnetic field strength, [uT] or [Gauss], (1 Gauss = 100uT)

→ Track orientation (position)

→ Track orientation

} Dead reckoning

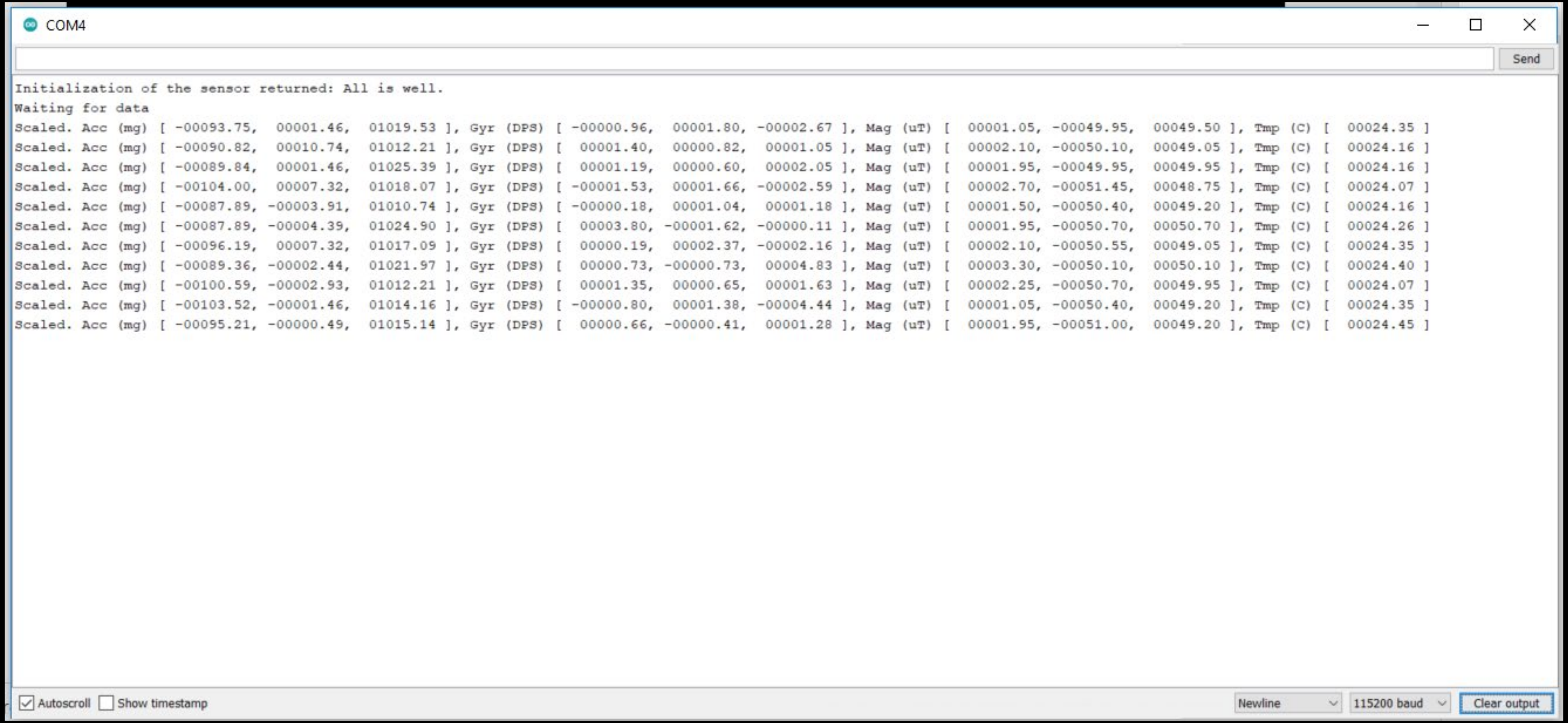
→ Get absolute orientation

- *NB: Gravity, magnetic fields, accelerations affect these sensors in many ways!*

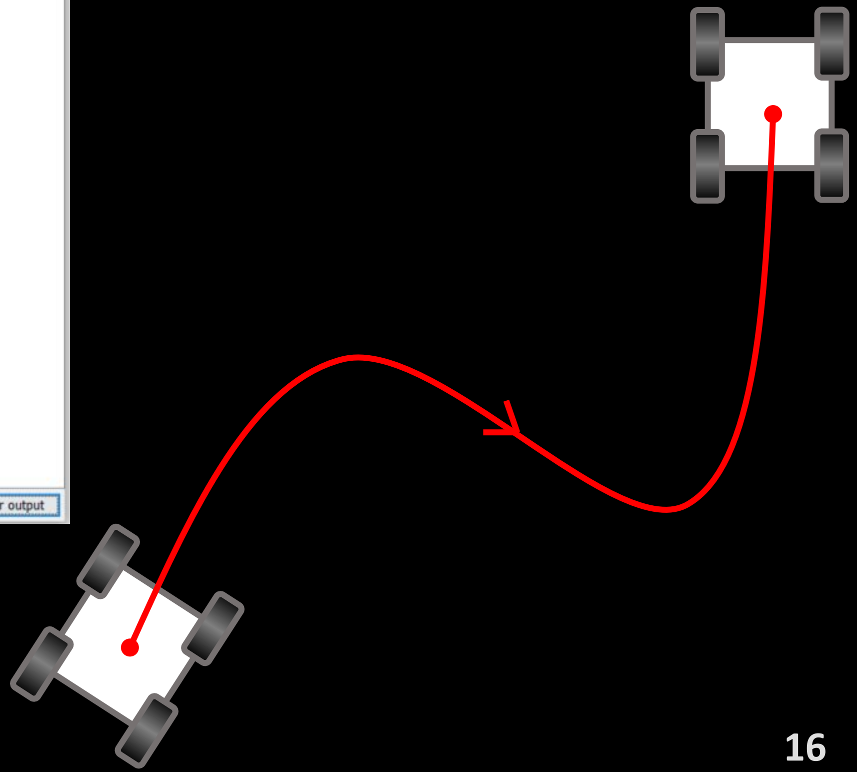
ICM-20948
Lowest power 9-axis IMU
$16

IMU - Demo

- Install Sparkfun 9DOF IMU - ICM 20948 library
- ..\SparkFun_ICM-20948_ArduinoLibrary-master\examples\Arduino\Example1_Basics




```
Initialization of the sensor returned: All is well.
Waiting for data
Scaled. Acc (mg) [ -00093.75,  00001.46,  01019.53 ], Gyr (DPS) [ -00000.96,  00001.80, -00002.67 ], Mag (uT) [  00001.05, -00049.95,  00049.50 ], Tmp (C) [  00024.35 ]
Scaled. Acc (mg) [ -00090.82,  00010.74,  01012.21 ], Gyr (DPS) [  00001.40,  00000.82,  00001.05 ], Mag (uT) [  00002.10, -00050.10,  00049.05 ], Tmp (C) [  00024.16 ]
Scaled. Acc (mg) [ -00089.84,  00001.46,  01025.39 ], Gyr (DPS) [  00001.19,  00000.60,  00002.05 ], Mag (uT) [  00001.95, -00049.95,  00049.95 ], Tmp (C) [  00024.16 ]
Scaled. Acc (mg) [ -00104.00,  00007.32,  01018.07 ], Gyr (DPS) [ -00001.53,  00001.66, -00002.59 ], Mag (uT) [  00002.70, -00051.45,  00048.75 ], Tmp (C) [  00024.07 ]
Scaled. Acc (mg) [ -00087.89, -00003.91,  01010.74 ], Gyr (DPS) [ -00000.18,  00001.04,  00001.18 ], Mag (uT) [  00001.50, -00050.40,  00049.20 ], Tmp (C) [  00024.16 ]
Scaled. Acc (mg) [ -00087.89, -00004.39,  01024.90 ], Gyr (DPS) [  00003.80, -00001.62, -00000.11 ], Mag (uT) [  00001.95, -00050.70,  00050.70 ], Tmp (C) [  00024.26 ]
Scaled. Acc (mg) [ -00096.19,  00007.32,  01017.09 ], Gyr (DPS) [  00000.19,  00002.37, -00002.16 ], Mag (uT) [  00002.10, -00050.55,  00049.05 ], Tmp (C) [  00024.35 ]
Scaled. Acc (mg) [ -00089.36, -00002.44,  01021.97 ], Gyr (DPS) [  00000.73, -00000.73,  00004.83 ], Mag (uT) [  00003.30, -00050.10,  00050.10 ], Tmp (C) [  00024.40 ]
Scaled. Acc (mg) [ -00100.59, -00002.93,  01012.21 ], Gyr (DPS) [  00001.35,  00000.65,  00001.63 ], Mag (uT) [  00002.25, -00050.70,  00049.95 ], Tmp (C) [  00024.07 ]
Scaled. Acc (mg) [ -00103.52, -00001.46,  01014.16 ], Gyr (DPS) [ -00000.80,  00001.38, -00004.44 ], Mag (uT) [  00001.05, -00050.40,  00049.20 ], Tmp (C) [  00024.35 ]
Scaled. Acc (mg) [ -00095.21, -00000.49,  01015.14 ], Gyr (DPS) [  00000.66, -00000.41,  00001.28 ], Mag (uT) [  00001.95, -00051.00,  00049.20 ], Tmp (C) [  00024.45 ]
```



Prof. Kirstin Hagelskjær Petersen
kirstin@cornell.edu

ACCELEROMETER

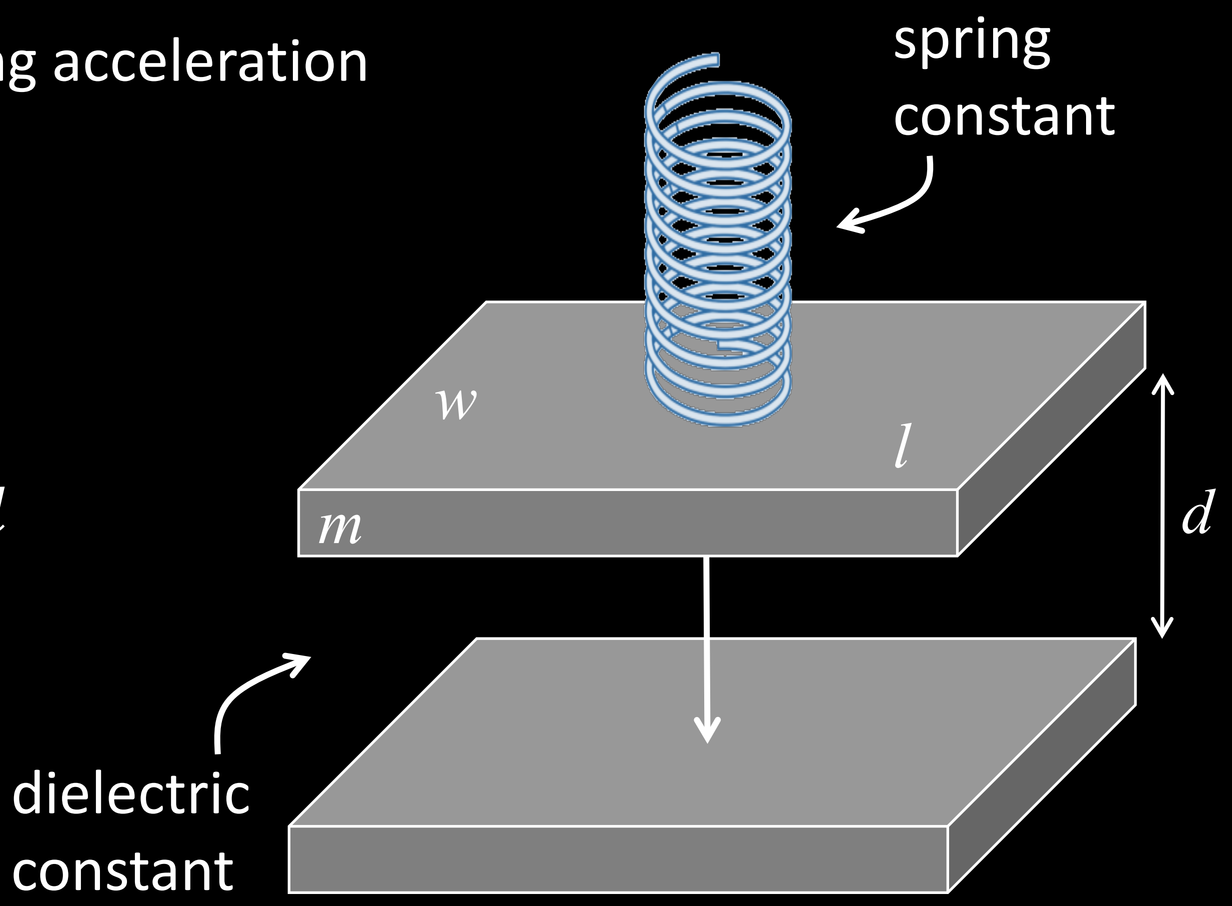
Accelerometer

- Measuring acceleration

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

$$F = k\Delta d$$

$$F = ma$$



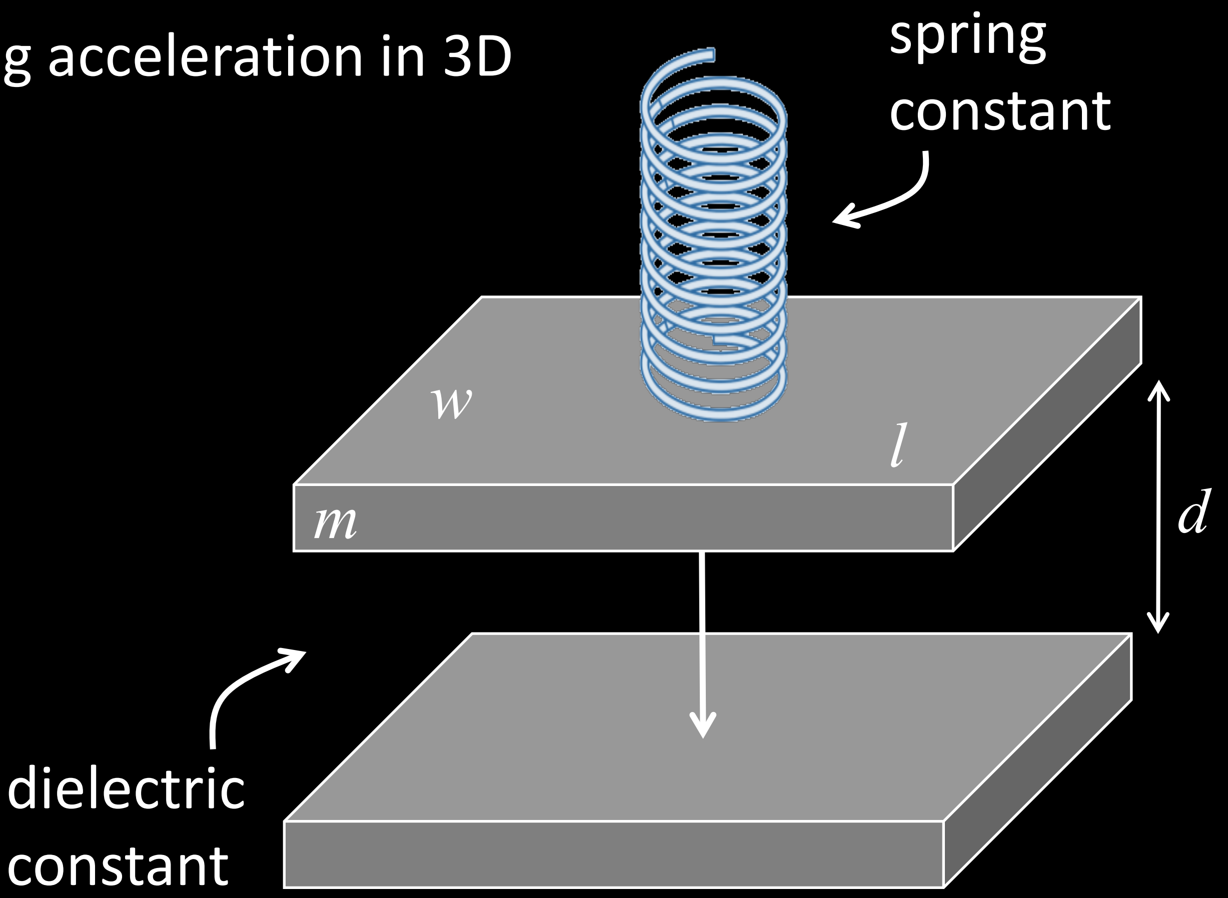
Accelerometer

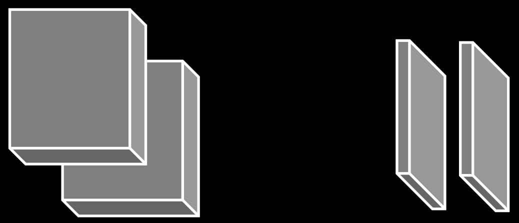
- Measuring acceleration in 3D

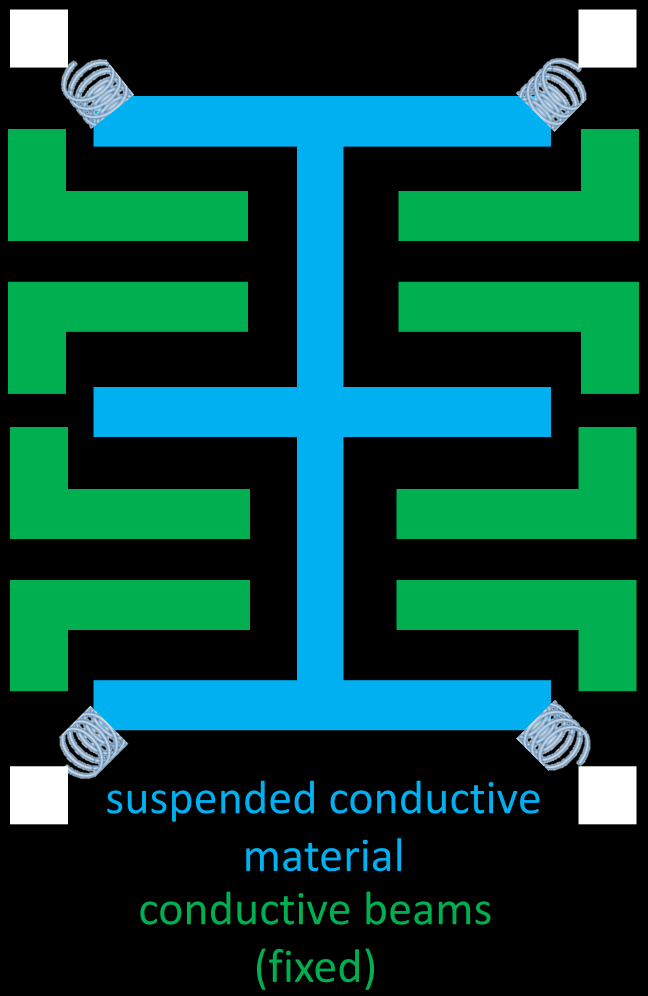
$$C = \frac{\in A}{d}$$

$$F = k\Delta d$$

$$F = ma$$

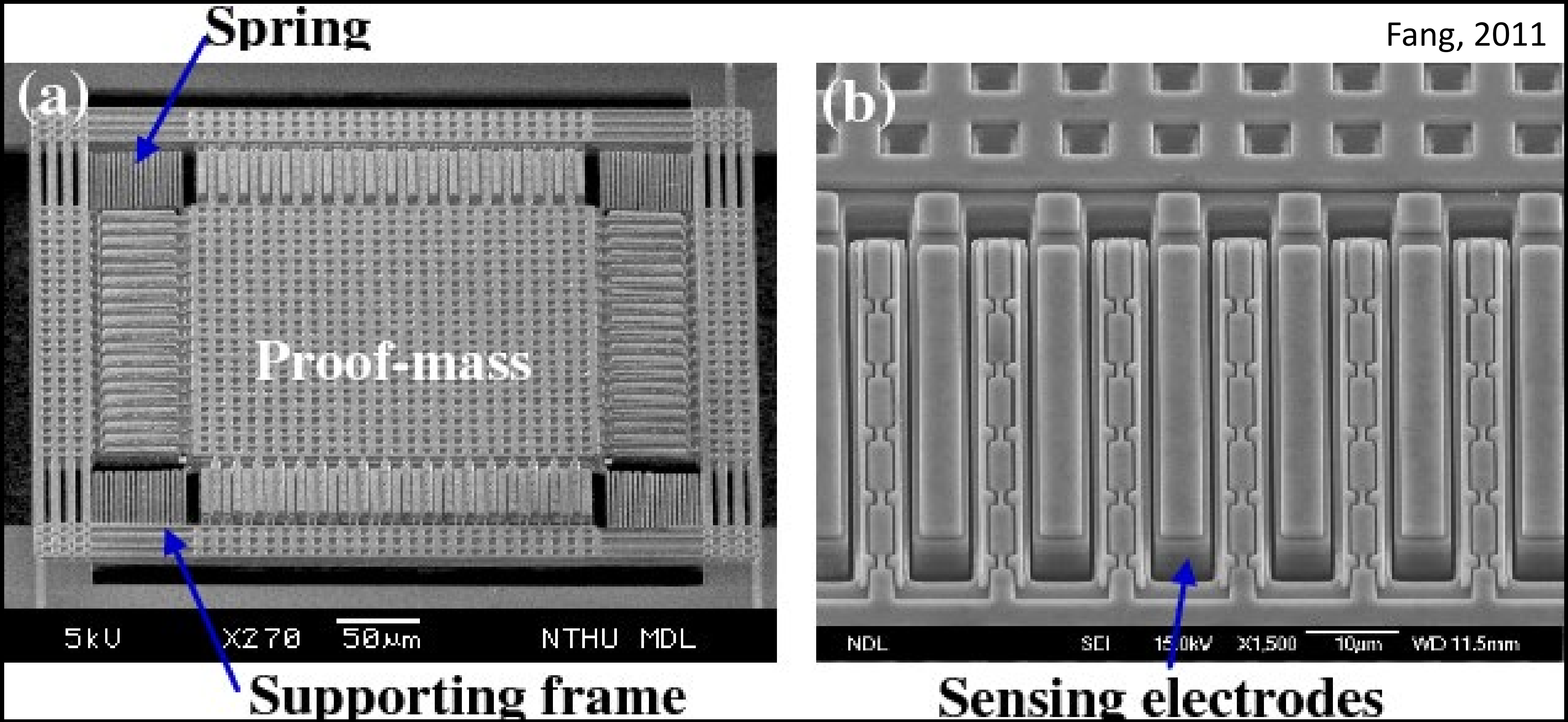






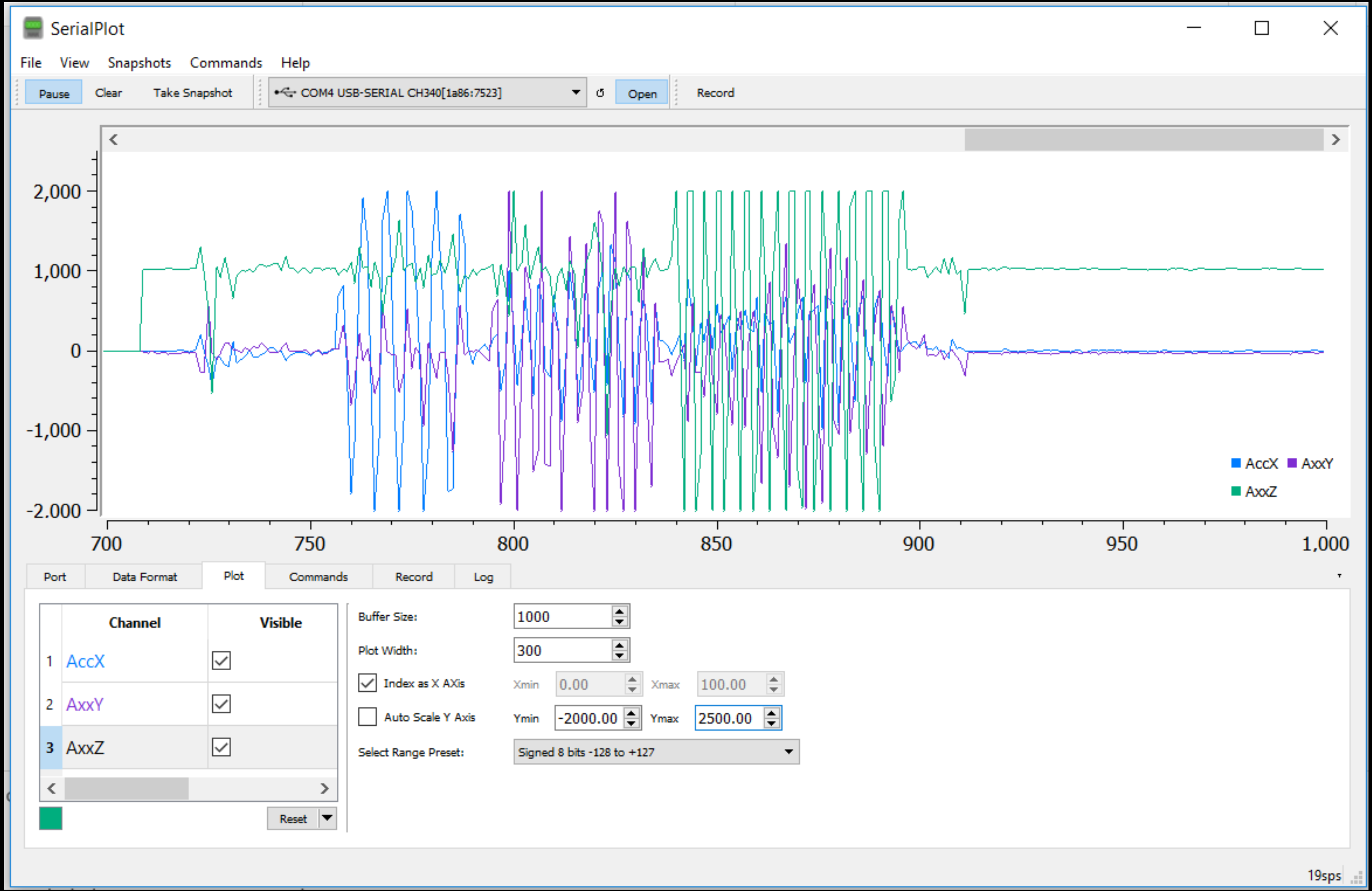
Accelerometer

- Measuring acceleration in 3D
- Micro-Electro-Mechanical Systems



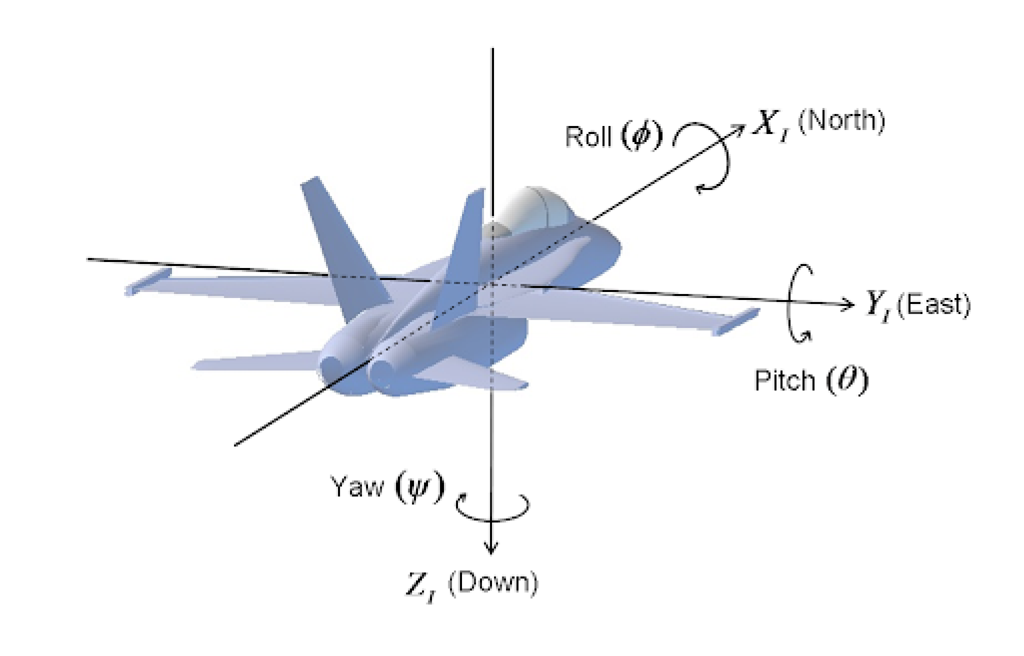
Accelerometer

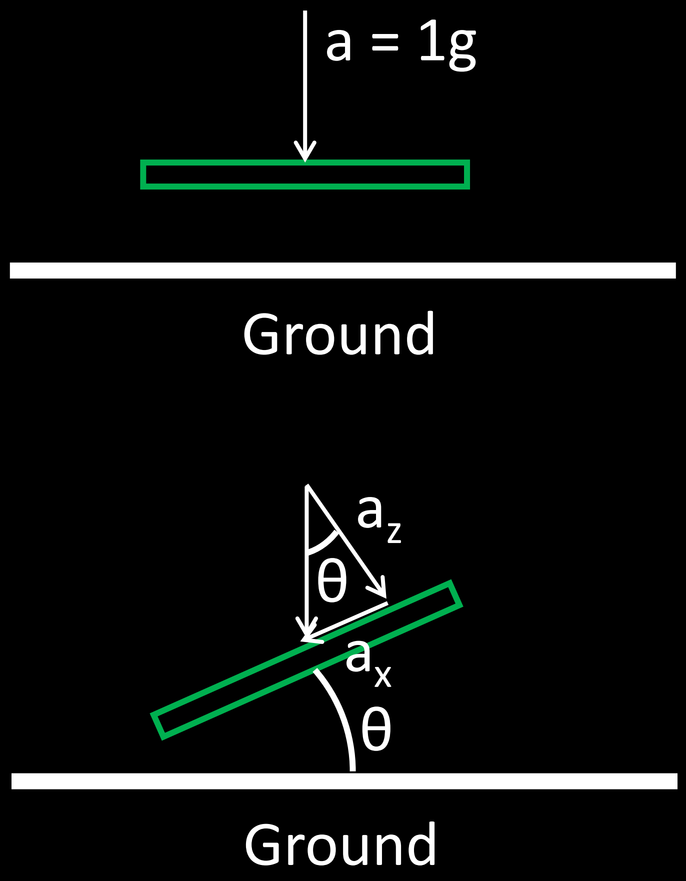
- Measuring acceleration in 3D
- Use Arduino Serial Monitor or a program like SerialPlot to visualize your data



Accelerometer

- How to use the accelerometer to determine roll, tilt, and yaw?

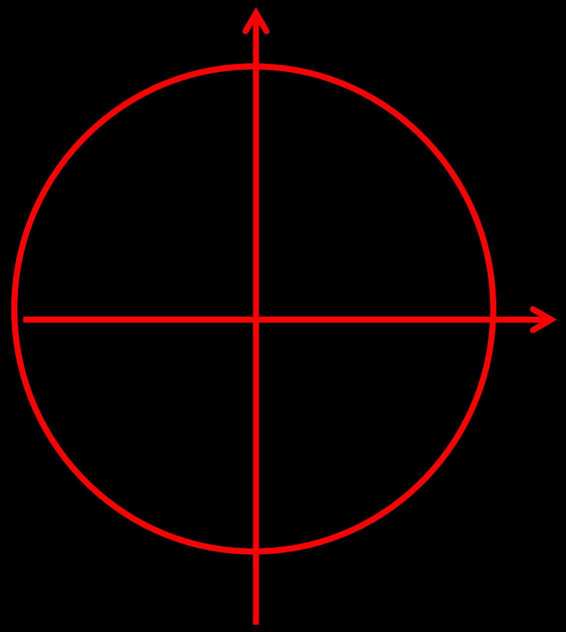




- $a_x = 1g \sin\theta$
- $a_z = 1g \cos\theta$
- *Use atan2!*

Atan vs Atan2

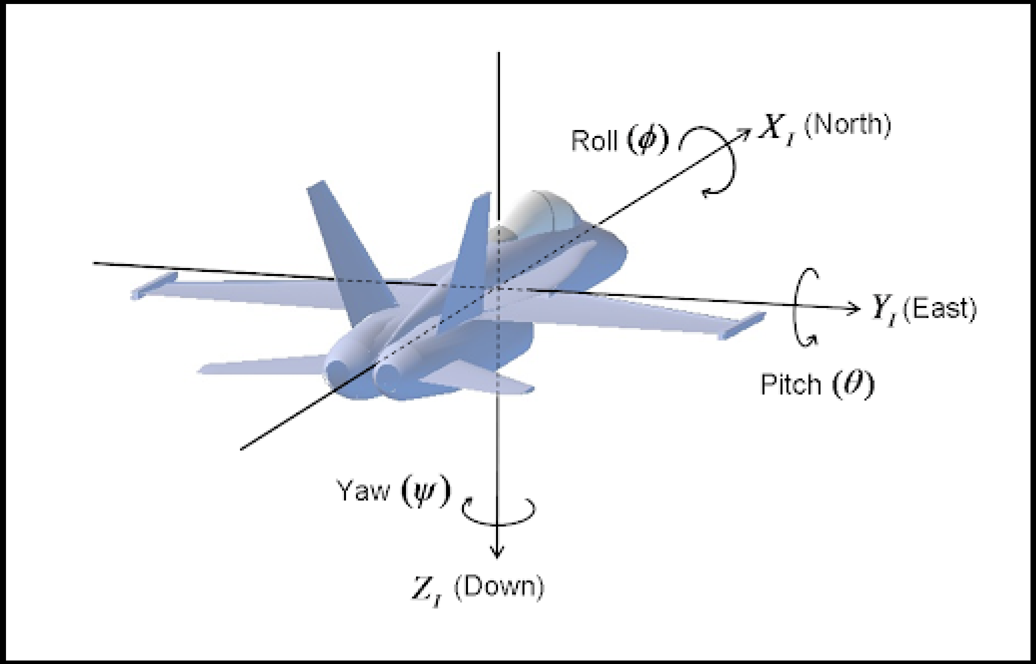
- acos(a_x) returns [0, π]
- atan(a_x, a_z) returns [- $\pi/2$, $\pi/2$]
- Instead use atan2(a_x, a_z) which returns [- π, π]

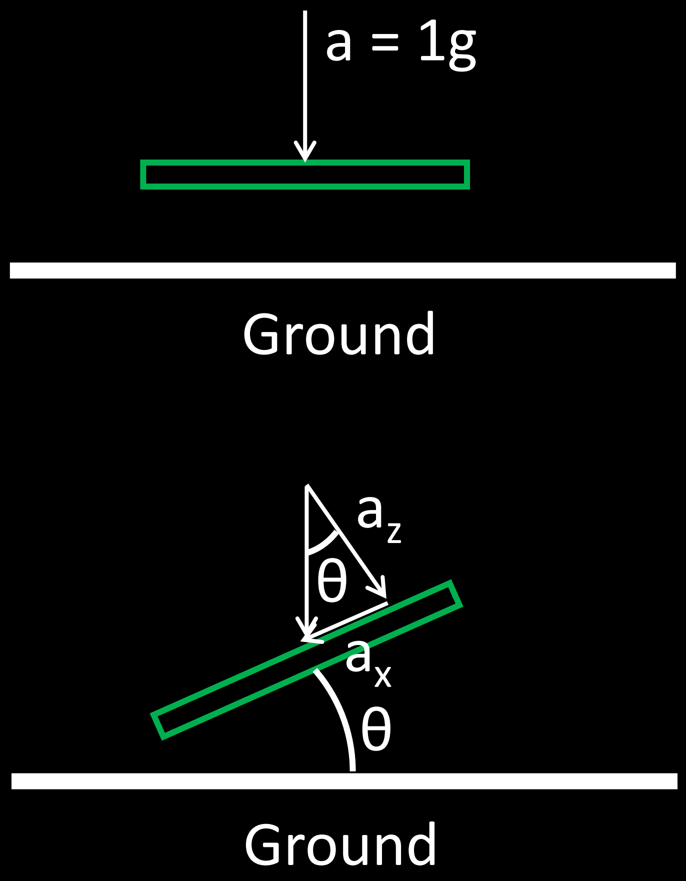


```
float atan2(float x, float y) {
    if (x > 0.0)
        return atan(y/x);
    if (x < 0.0) {
        if (y >= 0.0)
            return (PI + atan(y/x));
        else
            return (-PI + atan(y/x));
    }
    if (y > 0.0) // x == 0
        return PI_ON_TWO;
    if (y < 0.0)
        return -PI_ON_TWO;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

Accelerometer

- How to use the accelerometer to determine roll, tilt, and ~~yaw~~?



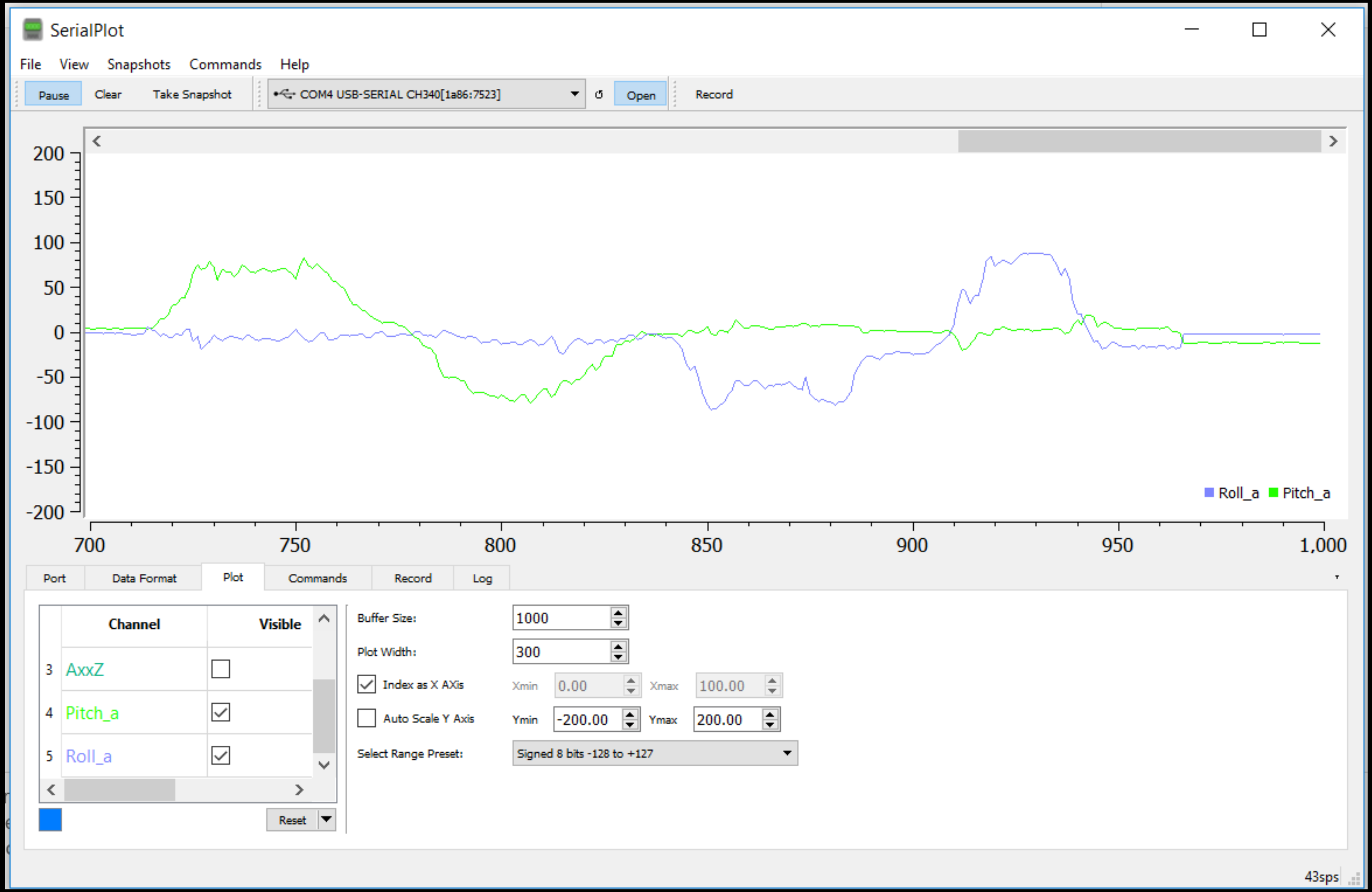


- $a_x = 1g \sin\theta$
- $a_z = 1g \cos\theta$
  - $\theta = \text{atan2}(a_x, a_z)$
    - $\phi = \text{atan2}(a_y, a_z)$

How do you measure yaw with the accelerometer?

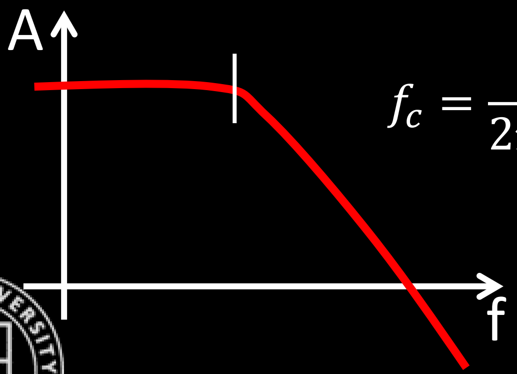
Accelerometer

- Determining tilt and roll
  - $\theta = \text{atan2}(a_x, a_z)$
  - $\phi = \text{atan2}(a_y, a_z)$

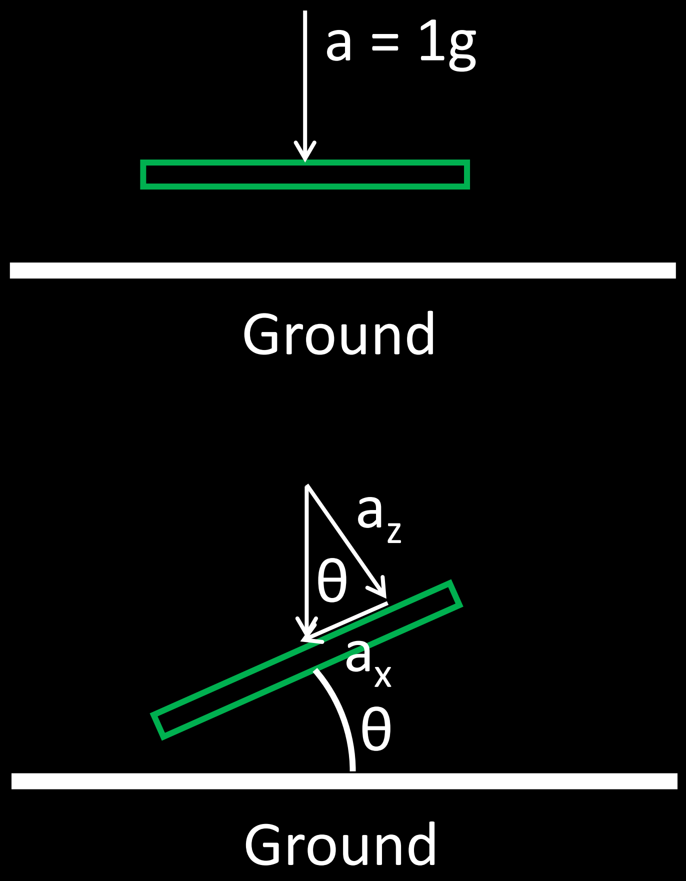


Accelerometer

- Determining tilt and roll
- Good (very accurate on average) vs Bad (noisy)
- Low pass complimentary filter
  - $\theta_{LPF}[n] = \alpha * \theta_{RAW} + (1 - \alpha) * \theta_{LPF}[n-1]$
  - $\theta_{LPF}[n-1] = \theta_{LPF}[n]$
  - Think of it as an RC low-pass filter:
    - $\alpha = \frac{T}{T+RC}$



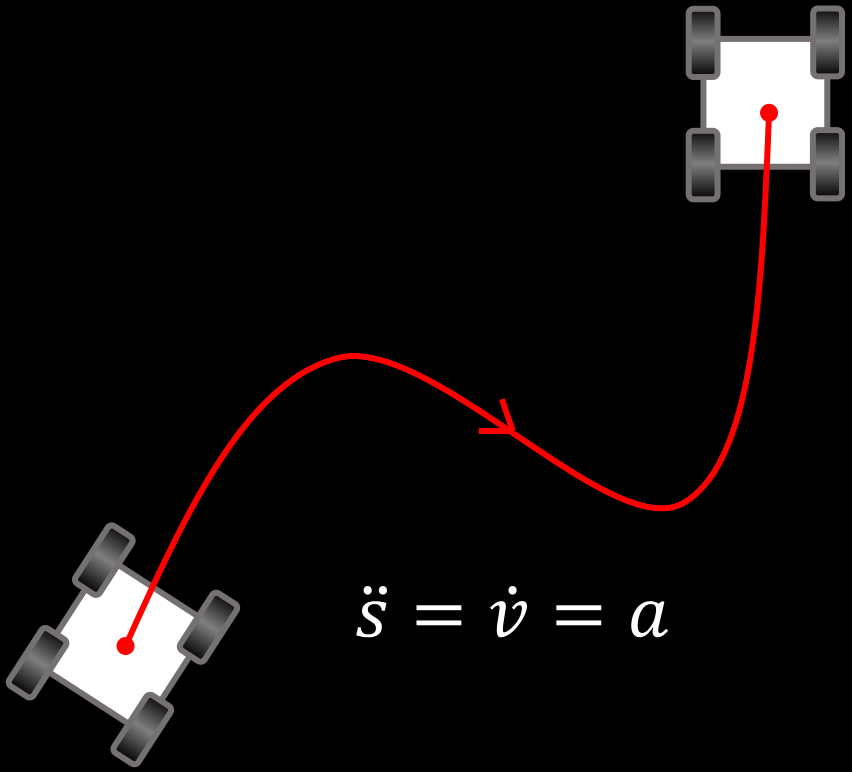
$f_c = \frac{1}{2\pi RC}$



Accelerometer

- Determining tilt and roll ✓
- How to use the accelerometer to do dead reckoning?
  - $v = \int a$
  - s= $\int \int a$
  - `v[k+1]=v[k]+a[k]*dt`
  - `s[k+1]=s[k]+v[k]*dt`

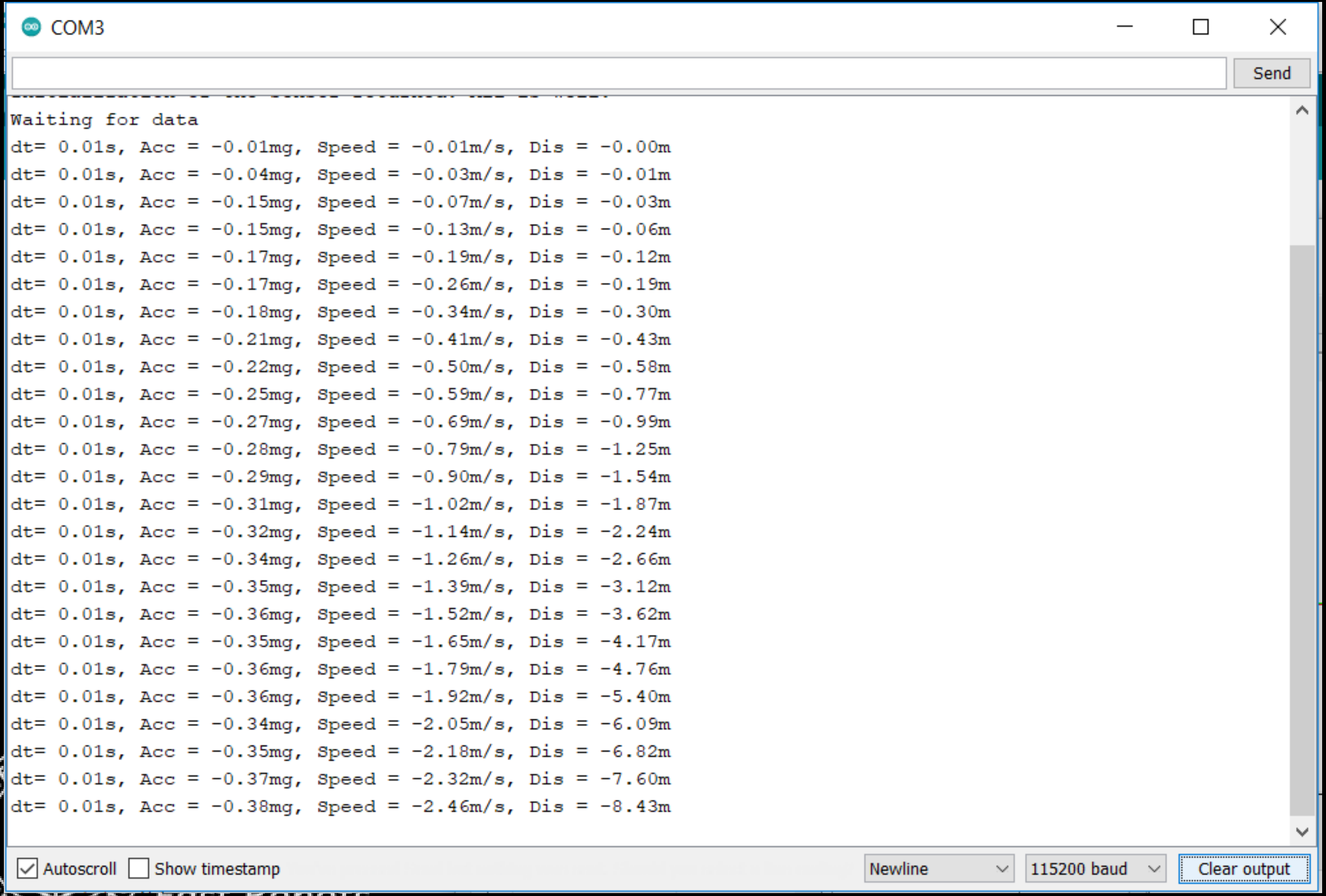
**If you try this at home, remember unit conversion:*
The accelerometer output is in mg (1g = 9.807m/s^2)



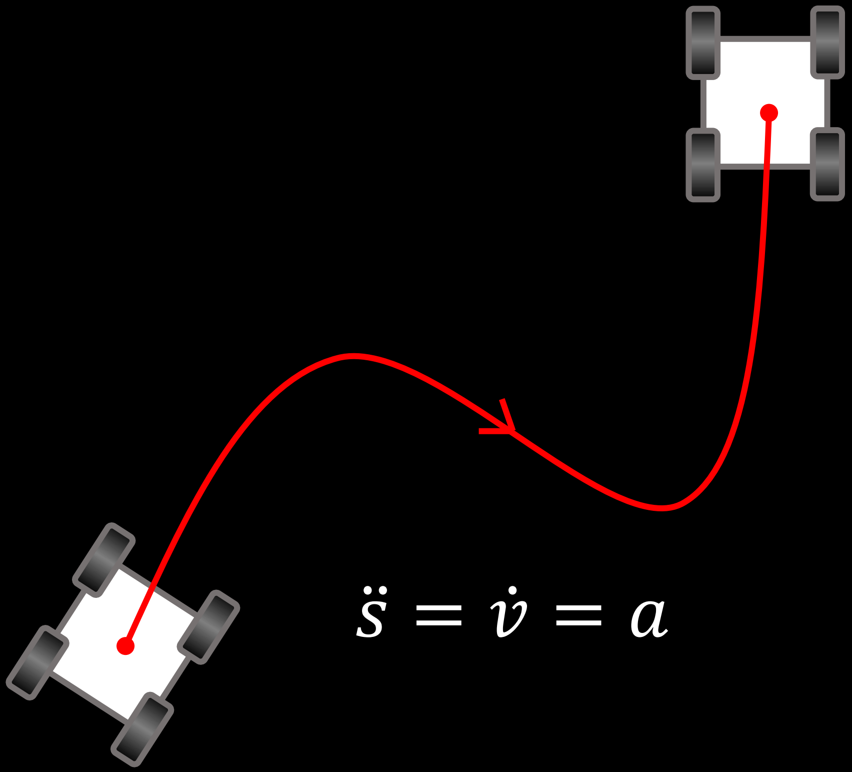
$$\ddot{s} = \dot{v} = a$$

Accelerometer

- Determining tilt and roll ✓
- How to use the accelerometer to do dead reckoning?
  - *Issue:* Distinguishing acceleration of the sensor from gravitational acceleration



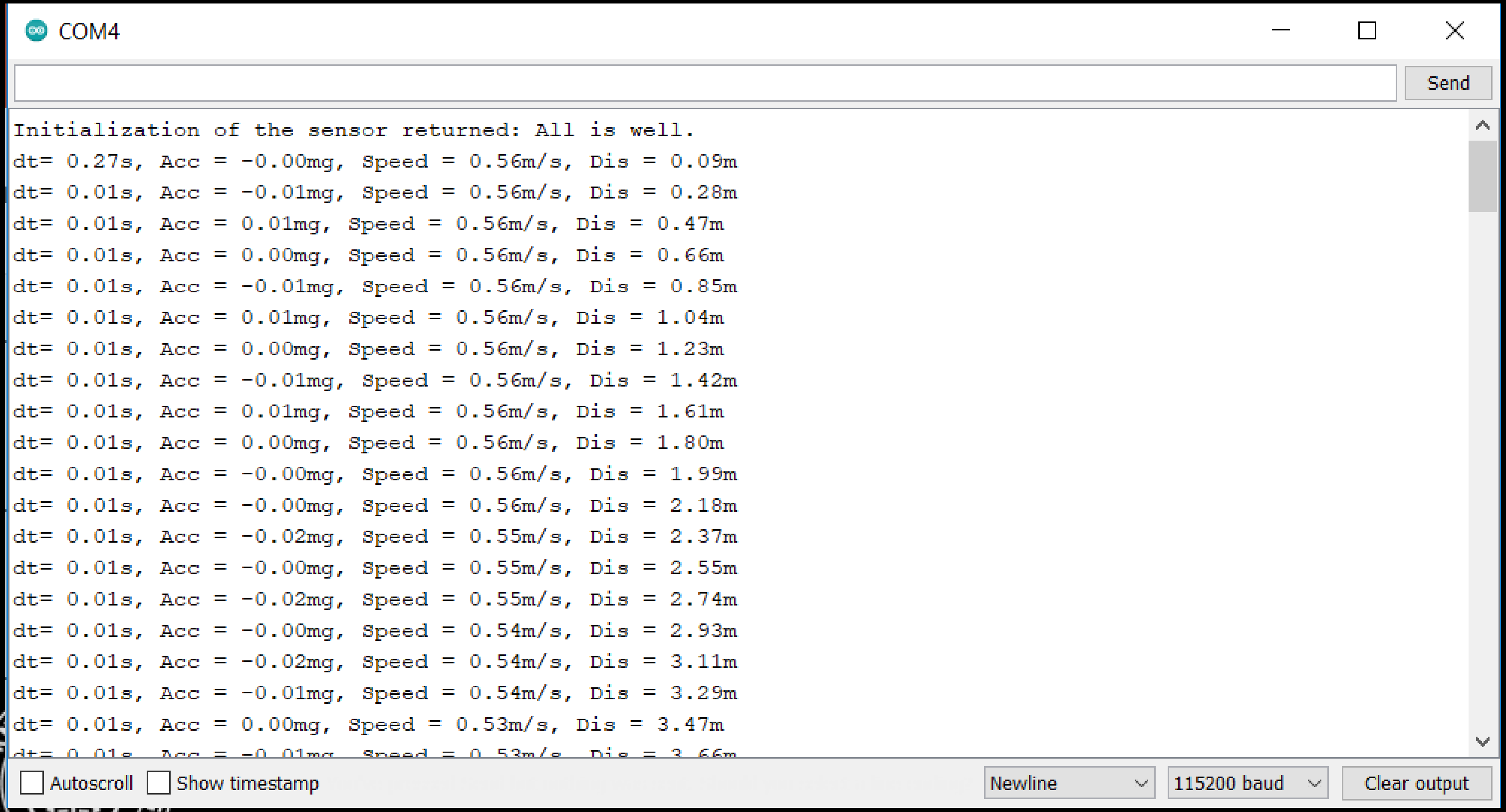

```
Waiting for data
dt= 0.01s, Acc = -0.01mg, Speed = -0.01m/s, Dis = -0.00m
dt= 0.01s, Acc = -0.04mg, Speed = -0.03m/s, Dis = -0.01m
dt= 0.01s, Acc = -0.15mg, Speed = -0.07m/s, Dis = -0.03m
dt= 0.01s, Acc = -0.15mg, Speed = -0.13m/s, Dis = -0.06m
dt= 0.01s, Acc = -0.17mg, Speed = -0.19m/s, Dis = -0.12m
dt= 0.01s, Acc = -0.17mg, Speed = -0.26m/s, Dis = -0.19m
dt= 0.01s, Acc = -0.18mg, Speed = -0.34m/s, Dis = -0.30m
dt= 0.01s, Acc = -0.21mg, Speed = -0.41m/s, Dis = -0.43m
dt= 0.01s, Acc = -0.22mg, Speed = -0.50m/s, Dis = -0.58m
dt= 0.01s, Acc = -0.25mg, Speed = -0.59m/s, Dis = -0.77m
dt= 0.01s, Acc = -0.27mg, Speed = -0.69m/s, Dis = -0.99m
dt= 0.01s, Acc = -0.28mg, Speed = -0.79m/s, Dis = -1.25m
dt= 0.01s, Acc = -0.29mg, Speed = -0.90m/s, Dis = -1.54m
dt= 0.01s, Acc = -0.31mg, Speed = -1.02m/s, Dis = -1.87m
dt= 0.01s, Acc = -0.32mg, Speed = -1.14m/s, Dis = -2.24m
dt= 0.01s, Acc = -0.34mg, Speed = -1.26m/s, Dis = -2.66m
dt= 0.01s, Acc = -0.35mg, Speed = -1.39m/s, Dis = -3.12m
dt= 0.01s, Acc = -0.36mg, Speed = -1.52m/s, Dis = -3.62m
dt= 0.01s, Acc = -0.35mg, Speed = -1.65m/s, Dis = -4.17m
dt= 0.01s, Acc = -0.36mg, Speed = -1.79m/s, Dis = -4.76m
dt= 0.01s, Acc = -0.36mg, Speed = -1.92m/s, Dis = -5.40m
dt= 0.01s, Acc = -0.34mg, Speed = -2.05m/s, Dis = -6.09m
dt= 0.01s, Acc = -0.35mg, Speed = -2.18m/s, Dis = -6.82m
dt= 0.01s, Acc = -0.37mg, Speed = -2.32m/s, Dis = -7.60m
dt= 0.01s, Acc = -0.38mg, Speed = -2.46m/s, Dis = -8.43m
```

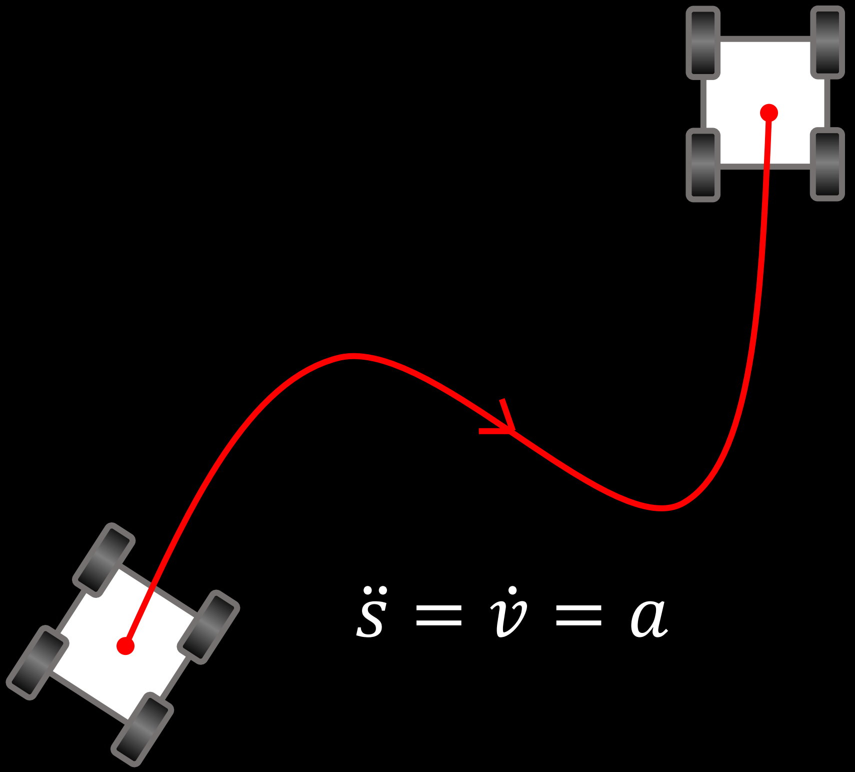


$$\ddot{s} = \dot{v} = a$$

Accelerometer

- Determining tilt and roll √
- How to use the accelerometer to do dead reckoning?
  - *Issue:* Distinguishing acceleration of the sensor from gravitational acceleration
  - *Solution 1*: Calibrate the offset

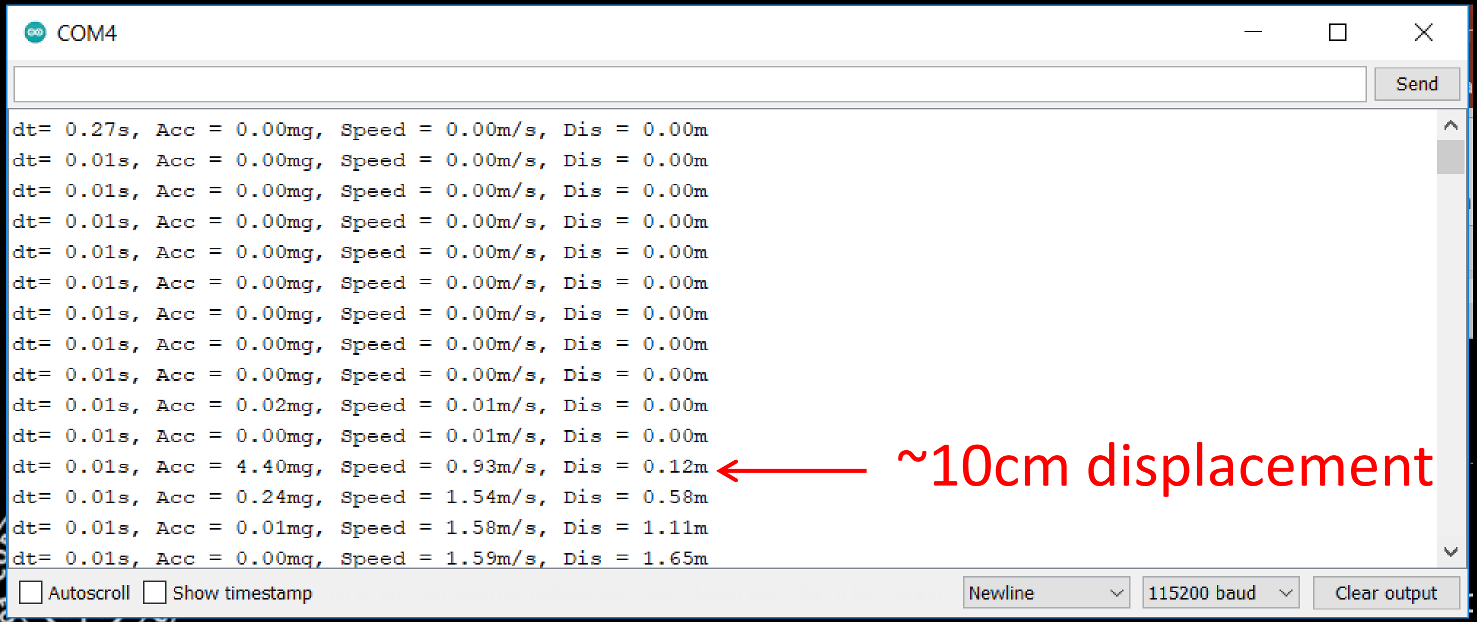




$$\ddot{s} = \dot{v} = a$$

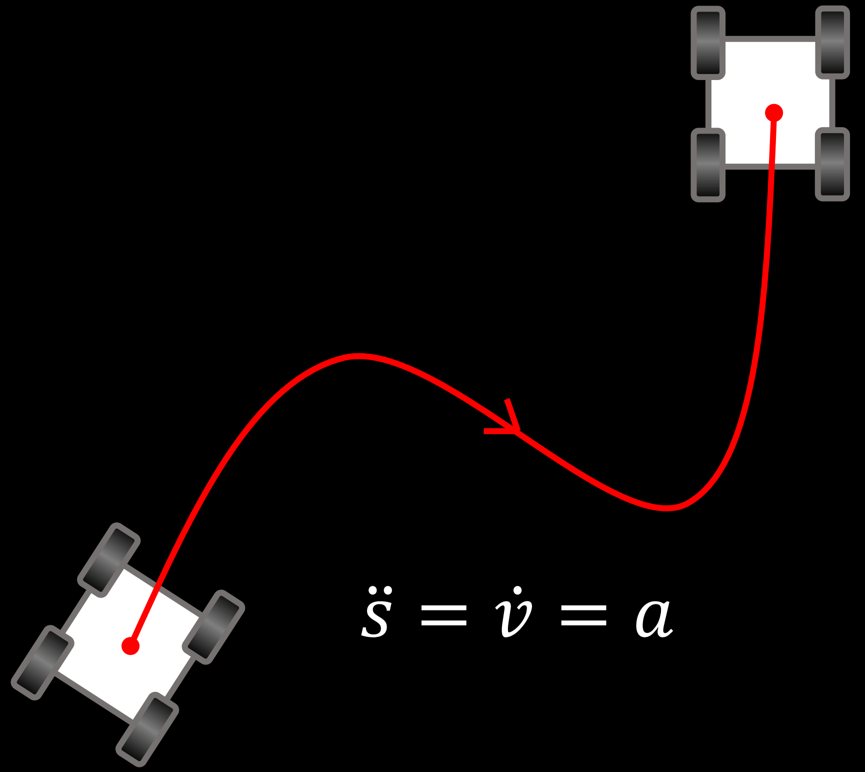
Accelerometer

- Determining tilt and roll √
- How to use the accelerometer to do dead reckoning?
  - *Issue:* Distinguishing acceleration of the sensor from gravitational acceleration
  - *Solution 1*: Calibrate the offset
  - *Solution 2:* Low pass filter the output
  - *Solution 3:* Minimum signal cut-off



```
COM4
Send
dt= 0.27s, Acc = 0.00mg, Speed = 0.00m/s, Dis = 0.00m
dt= 0.01s, Acc = 0.00mg, Speed = 0.00m/s, Dis = 0.00m
dt= 0.01s, Acc = 0.00mg, Speed = 0.00m/s, Dis = 0.00m
dt= 0.01s, Acc = 0.00mg, Speed = 0.00m/s, Dis = 0.00m
dt= 0.01s, Acc = 0.00mg, Speed = 0.00m/s, Dis = 0.00m
dt= 0.01s, Acc = 0.00mg, Speed = 0.00m/s, Dis = 0.00m
dt= 0.01s, Acc = 0.00mg, Speed = 0.00m/s, Dis = 0.00m
dt= 0.01s, Acc = 0.00mg, Speed = 0.00m/s, Dis = 0.00m
dt= 0.01s, Acc = 0.00mg, Speed = 0.00m/s, Dis = 0.00m
dt= 0.01s, Acc = 0.02mg, Speed = 0.01m/s, Dis = 0.00m
dt= 0.01s, Acc = 0.00mg, Speed = 0.01m/s, Dis = 0.00m
dt= 0.01s, Acc = 4.40mg, Speed = 0.93m/s, Dis = 0.12m
dt= 0.01s, Acc = 0.24mg, Speed = 1.54m/s, Dis = 0.58m
dt= 0.01s, Acc = 0.01mg, Speed = 1.58m/s, Dis = 1.11m
dt= 0.01s, Acc = 0.00mg, Speed = 1.59m/s, Dis = 1.65m
Autoscroll  Show timestamp  Newline  115200 baud  Clear output
```

~10cm displacement



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

Accelerometer

- Determining tilt and roll √
- How to use the accelerometer to do dead reckoning?
  - *Issue:* Distinguishing acceleration of the sensor from gravitational acceleration
  - *Solution 1*: Calibrate the offset
  - *Solution 2:* Low pass filter the output
  - *Solution 3:* Minimum signal cut-off

Errors only accumulate, and the … fast!

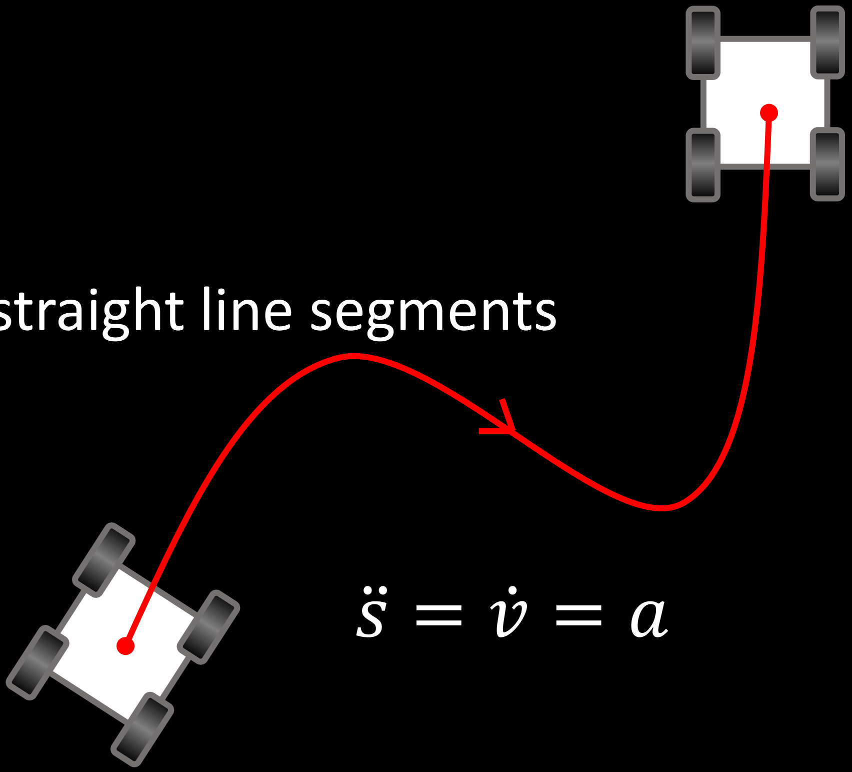
www.chrobotics.com

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	[illegible]	8.6	309.6	30960	1.1 e 6
1.0	0.17	[illegible]	17	[illegible]	[illegible]	2.2 e 6
1.5	0.256	[illegible]	25.6	921.6	92160	3.3 e 6
2.0	0.342	3.4	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

- Determining tilt and roll ✓
- How to use the accelerometer to do dead reckoning?
  - *Issue:* Distinguishing acceleration of the sensor from gravitational acceleration
  - *Solution 1*: Calibrate the offset
  - *Solution 2:* Low pass filter the output
  - *Solution 3:* Minimum signal cut-off
  - *Solution 4:* Stop periodically and zero the velocity
  - *Solution 5:* Use in combination with TOF sensor on straight line segments
  - *Solution 6:* Buy a more expensive IMU
  - etc…



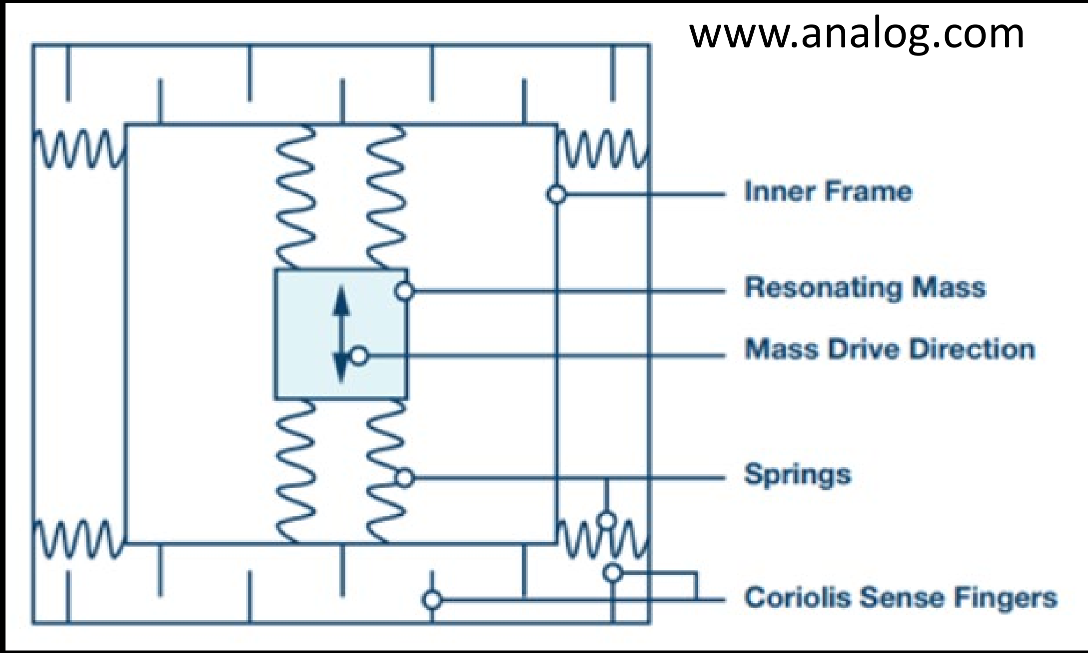
$$\ddot{s} = \dot{v} = a$$

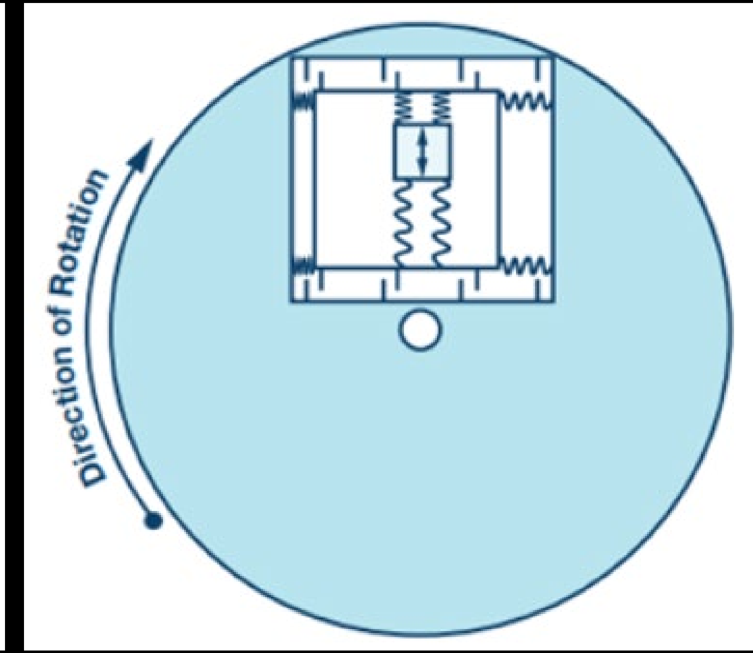
Prof. Kirstin Hagelskjær Petersen
kirstin@cornell.edu

GYROSCOPE

Gyroscopes

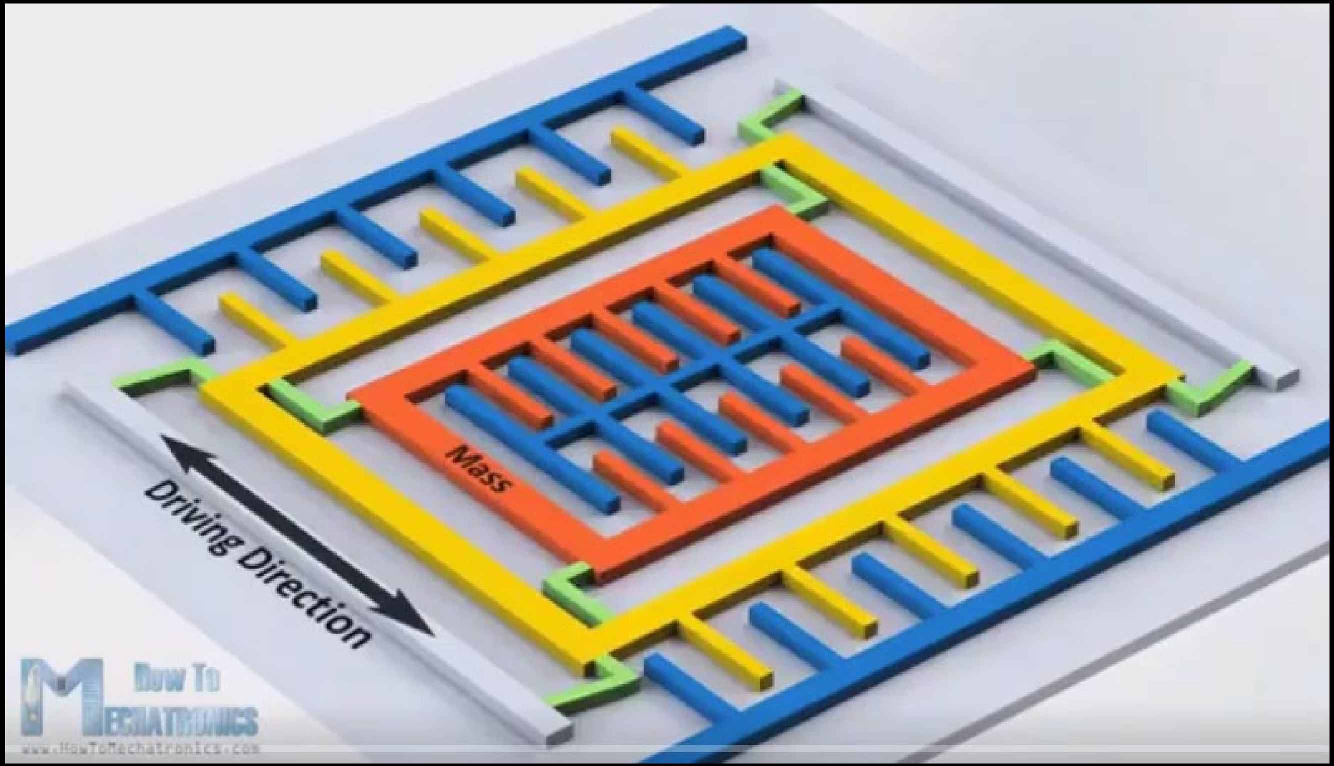
- Measures the rate of angular change [deg/s]





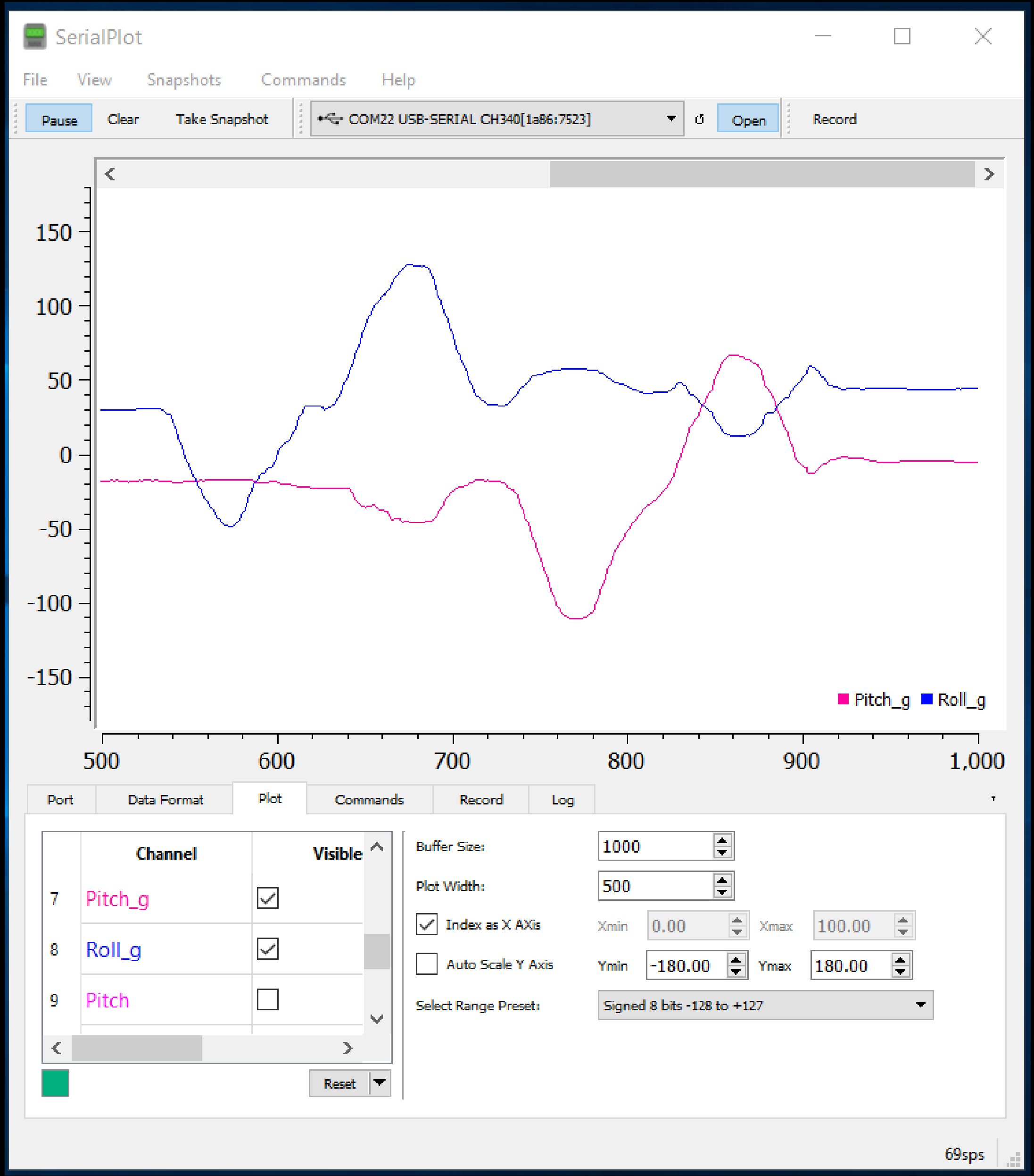
Gyroscopes

- Measures the rate of angular change [deg/s]



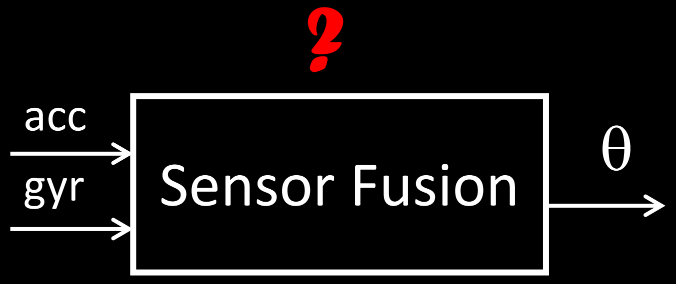
Gyroscopes

- Measures the rate of angular change [deg/s]
- How to use the gyroscope to measure angles?
  - $\theta_g = \theta_g + \text{gyr\_reading} * dt$
- *Drift, but low noise*



Gyroscopes

- Measures the rate of angular change [deg/s]
- How to use the gyroscope to measure angles?
  - $\theta_g = \theta_g - gyr\_reading*dt$
- *Drift, but low noise*
  - Complimentary to the accelerometer!
- Complimentary filter:
  - $\theta = (\theta + \theta_g*dt)(1-\alpha) + \theta_a\alpha$

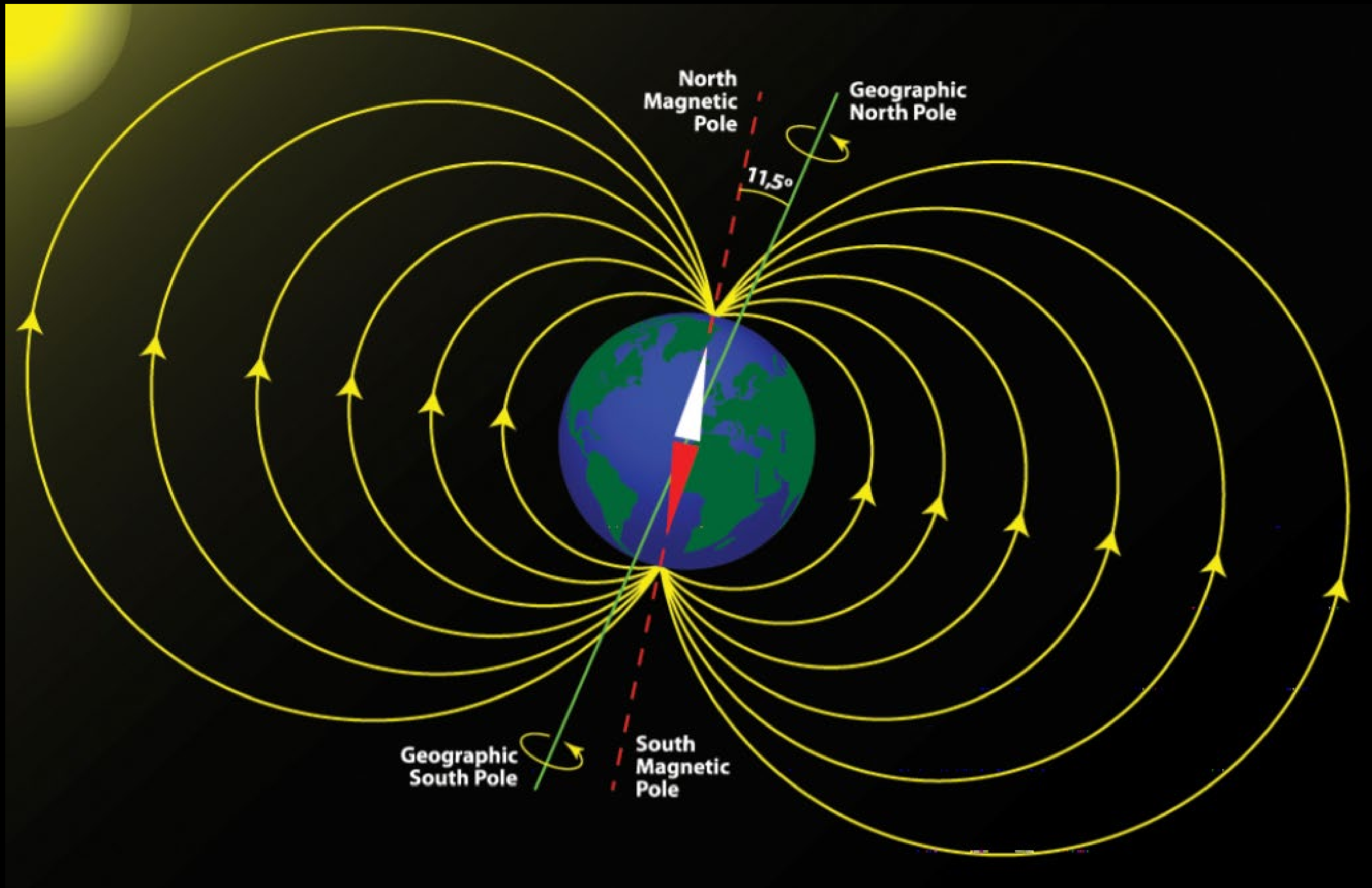


Can we also estimate yaw?

- Yes! (but there is no complementary data from the accelerometer)

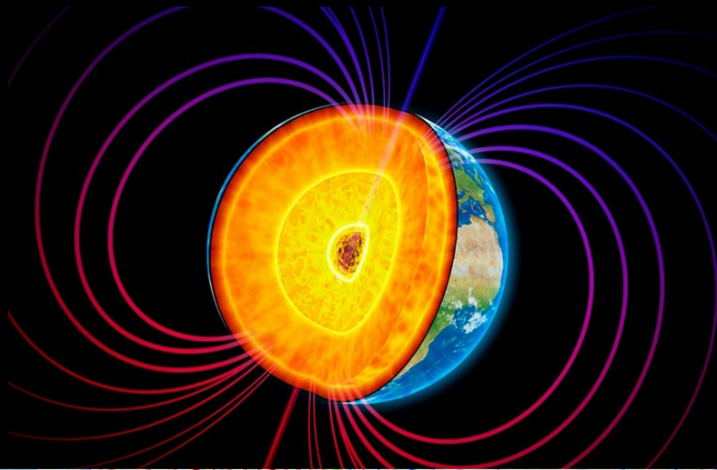
Magnetometer

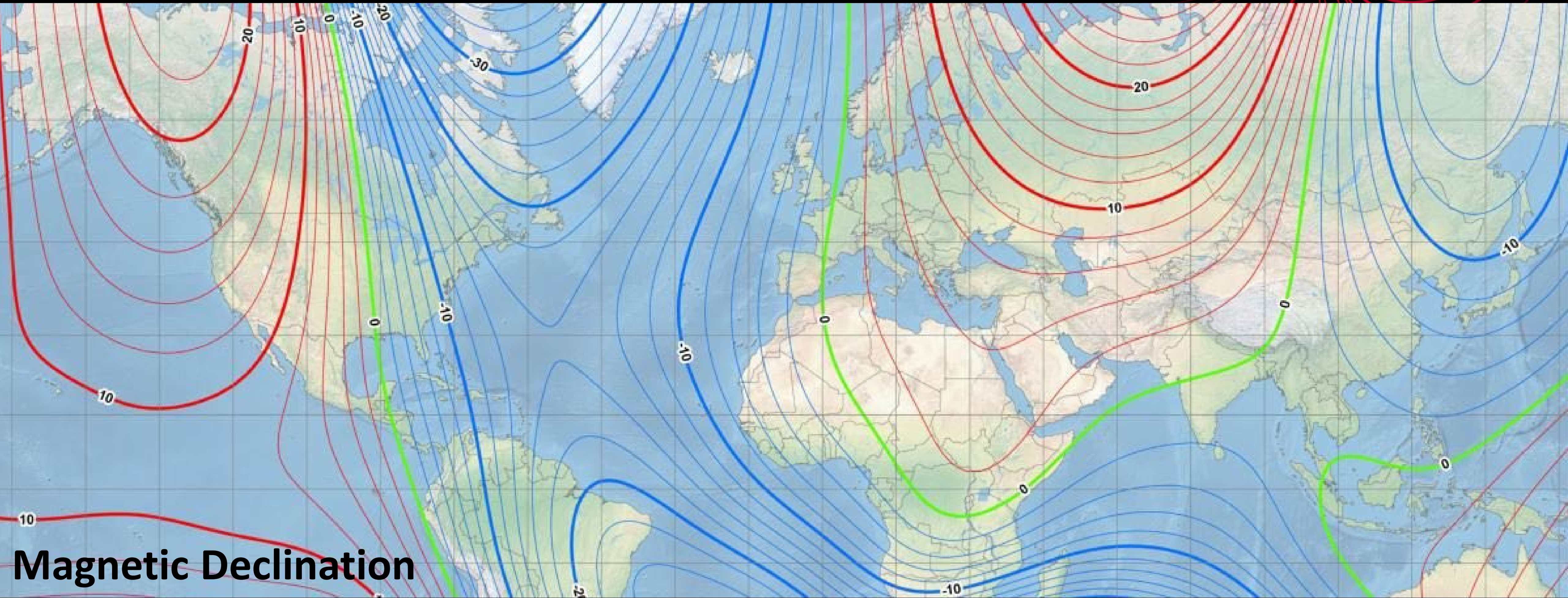
- Measure the Earth's magnetic field [Gauss] or [uT]
- The actual direction depends on latitude, longitude, and time



Magnetometer

- Measure the Earth’s magnetic field [Gauss] or [uT]
- The actual direction depends on latitude, longitude, and time
- Distortions due to metal objects or nearby EM fields

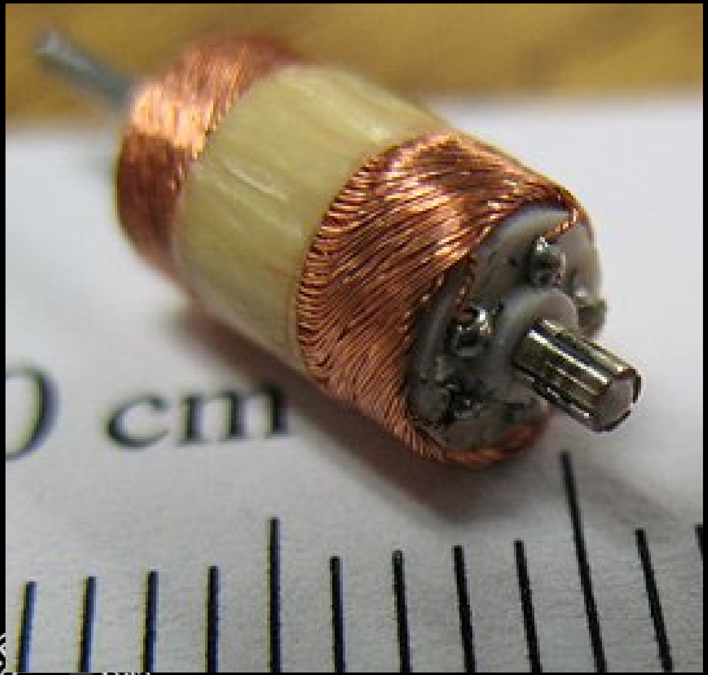




Magnetic Declination

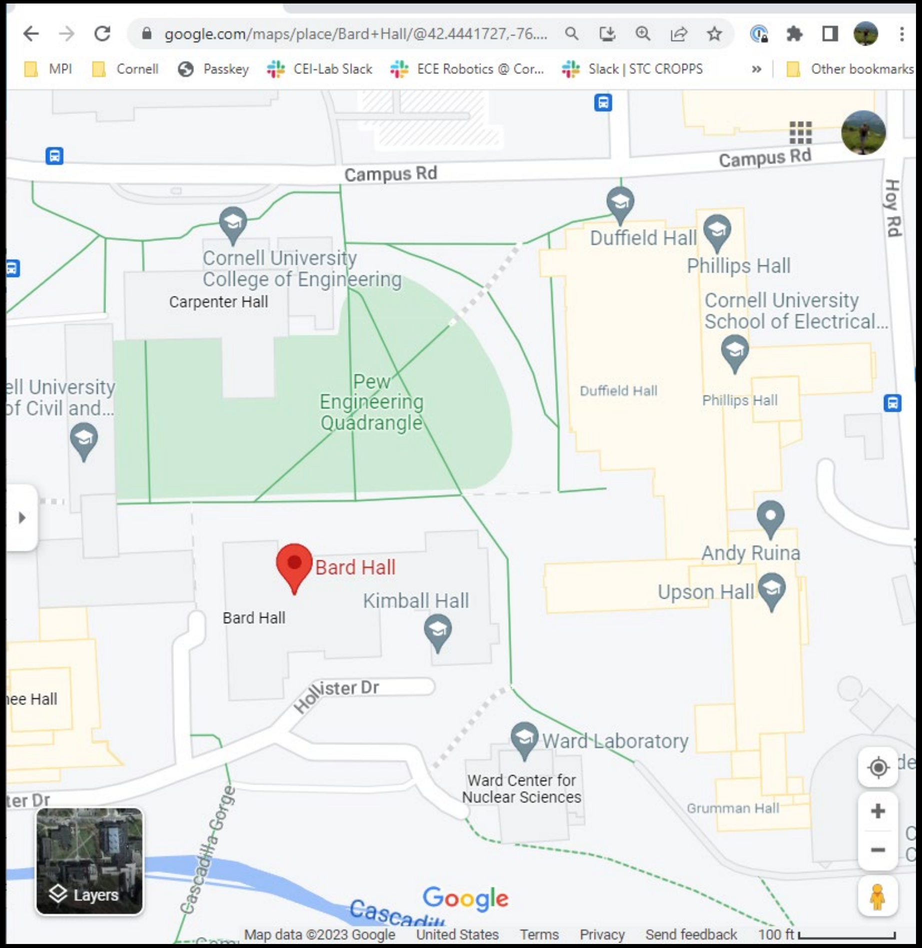
Magnetometer

- Measure the Earth's magnetic field [Gauss] or [uT]
- The actual direction depends on latitude, longitude, and time
- Distortions due to metal objects or nearby EM fields
  - ***Examples?***



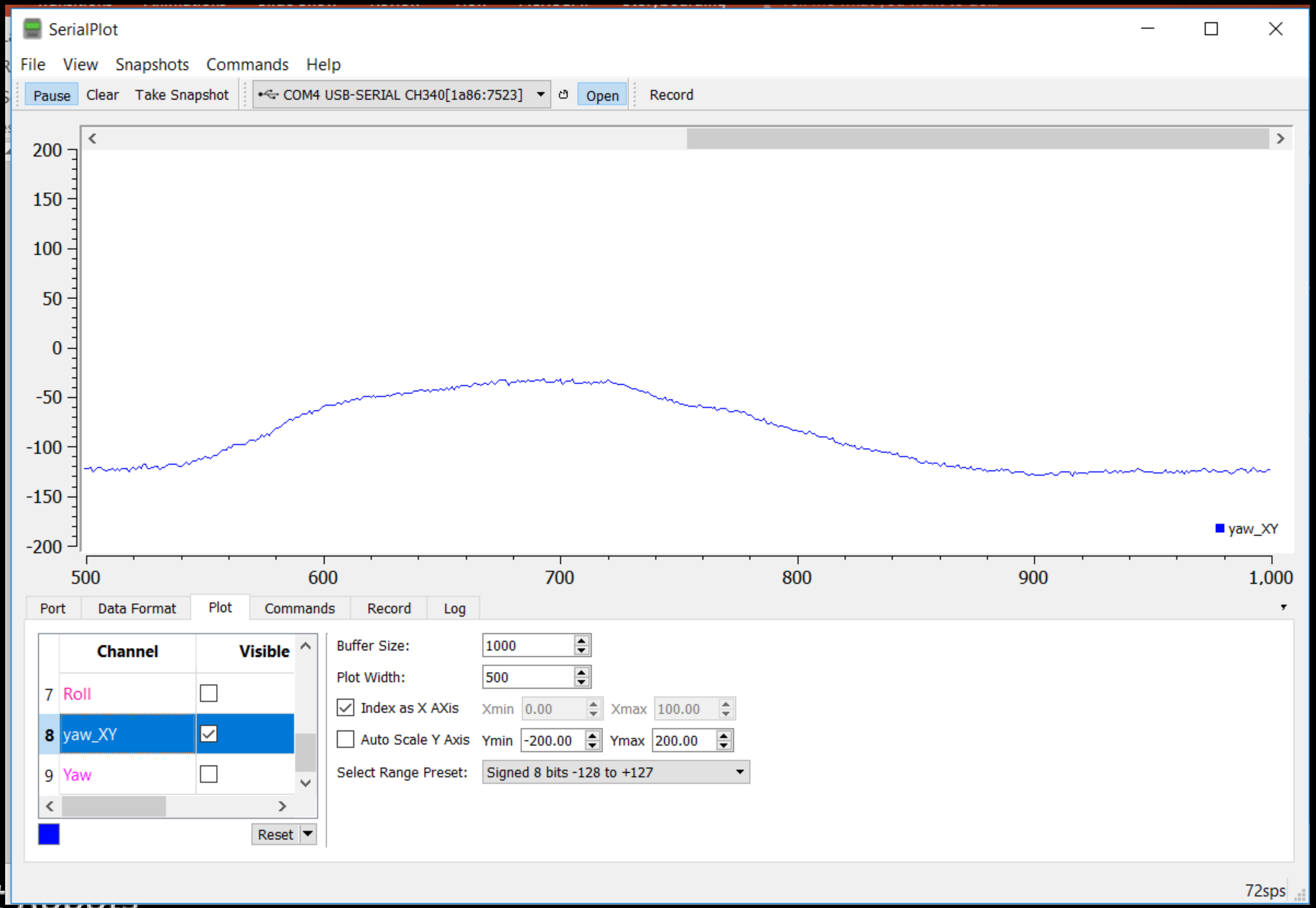
Magnetometer

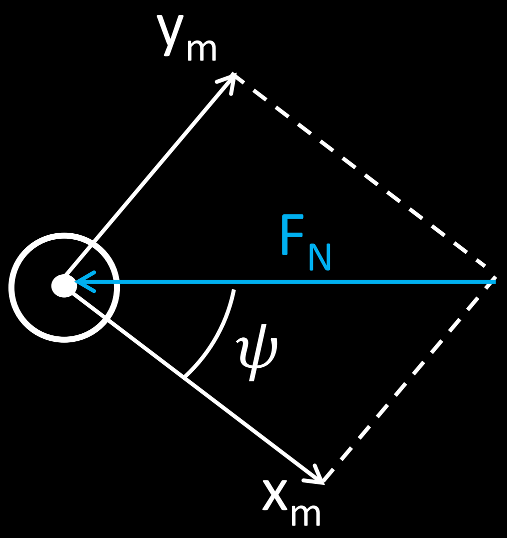
- Measure the Earth's magnetic field [Gauss] or [uT]
  - Magnetic north is along x_{max}-axis



Magnetometer

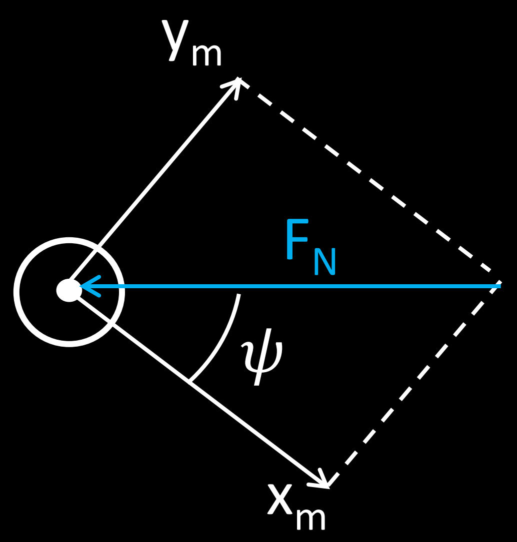
- Measure the Earth's magnetic field [Gauss] or [uT]
- $\psi = \text{atan2}(x_m, y_m)$





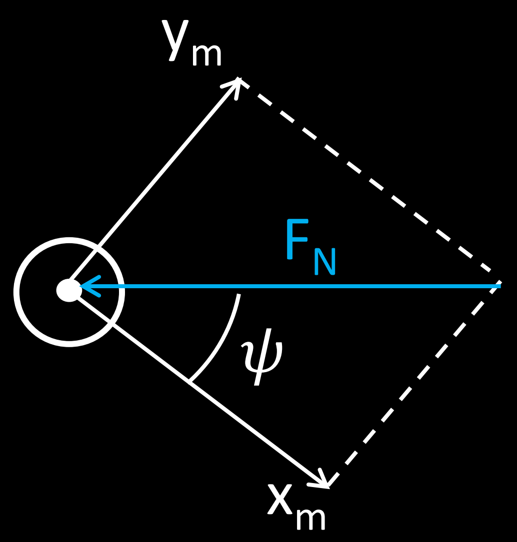
Magnetometer

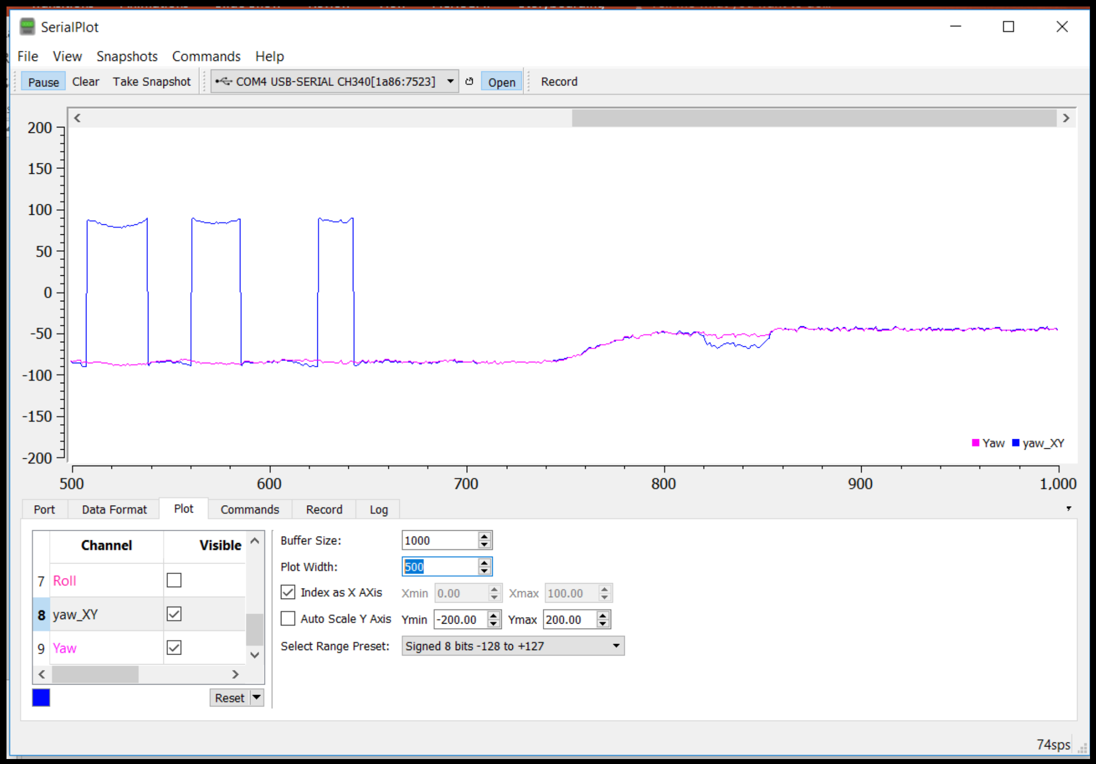
- Measure the Earth's magnetic field [Gauss] or [uT]
- $\psi = \text{atan2}(x_m, y_m)$
- What if you are also experiencing pitch and roll?
  - Fuse accelerometer + gyroscope + magnetometer data
- Tilt-compensated compass
  - $\begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} = R_{x,\phi} R_{y,\theta} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$
  - $\begin{bmatrix} x \\ y \\ z \end{bmatrix} = {R_{x,\phi}}^T {R_{y,\theta}}^T \begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ \sin(\phi)\sin(\theta) & \cos(\phi) & \cos(\theta)\sin(\phi) \\ \cos(\phi)\sin(\theta) & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} \begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix}$
  - $x = y_m*\cos(\phi) - z_m*\sin(\phi);$
  - $y = x_m*\cos(\theta) + y_m*\sin(\phi)*\sin(\theta) + z_m*\cos(\phi)*\sin(\theta);$
  - $\psi = \text{atan2}(x, y)$



Magnetometer

- Measure the Earth's magnetic field [Gauss] or [uT]
- $\psi = \text{atan}\left(\frac{y_m}{x_m}\right)$
- What if you are also experiencing pitch and roll?
  - Fuse accelerometer + gyroscope + magnetometer data
- Tilt-compensated compass





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/