CAS 765 Fall'15
Mobile Computing and Wireless Networking

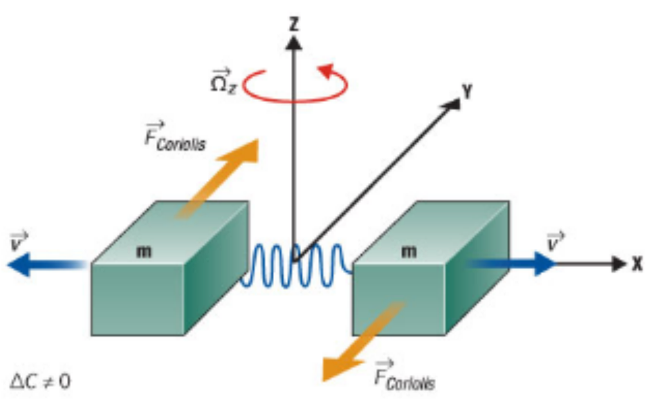
Rong Zheng

Sensor & Sensor Data Processing

Part II Inertial Measurement Units (IMU)

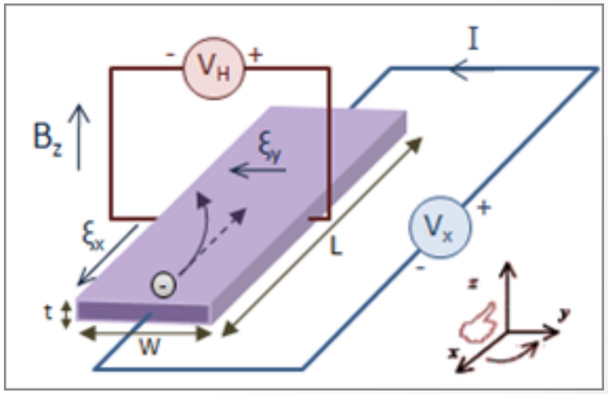
IMUs

- Includes
  - Accelerometer measures acceleration
  - Gyro measures Corilios force due to rotation, or angular velocity
  - Magnetometer reports the magnetic field
  - All in <x, y, z>
- Microelectromechanical systems (MEMS) sensors
- Single chip solution: 9-DOF IMU sensors available in the market



An Illustration of Coriolis Effect for Gyro

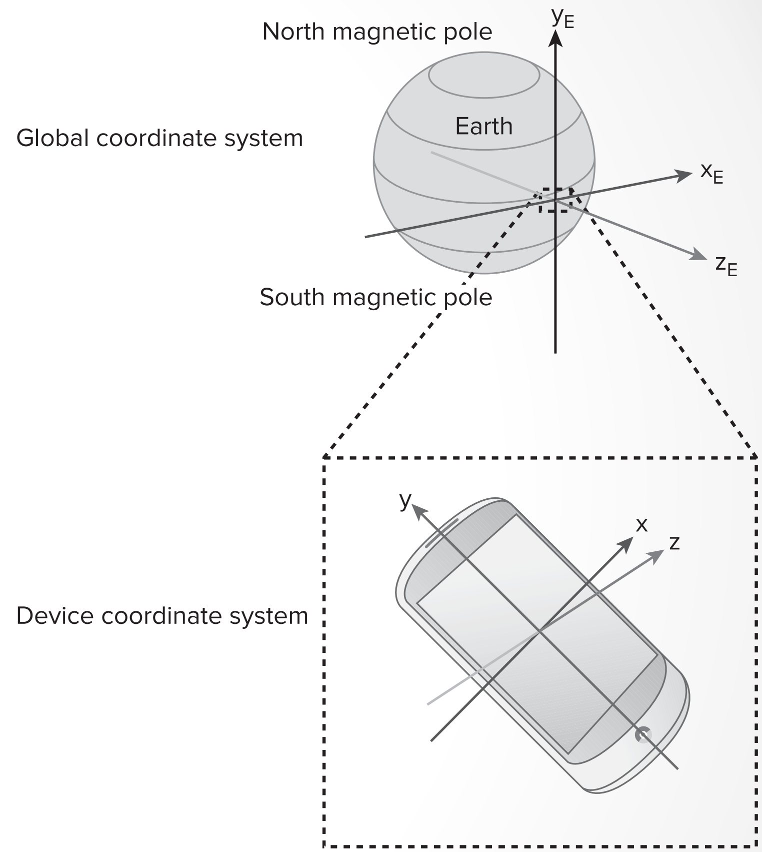
http://electroiq.com/blog/2010/11/introduction-to-mems-gyroscopes/



An Illustration of Hall Effect for Magnetometer (from Wikipedia)

Device Attitude/Pose

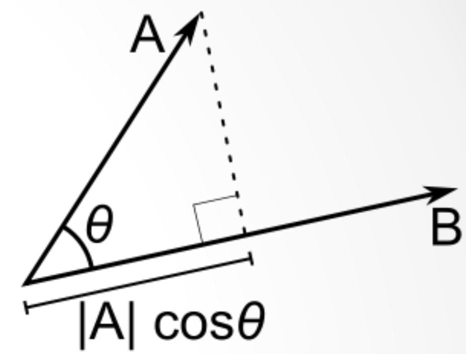
- *Global coordinate system* x_E, y_E, z_E, and
- A *device coordinate system* x, y, z
- Need to know the rotation from the device coordinate system to the global coordinate system
- First, how to represent rotation?

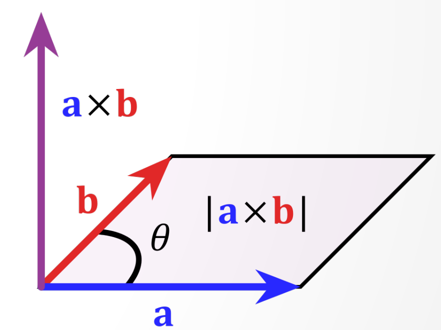


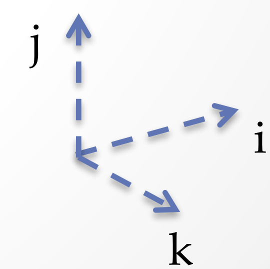
Review: Vector Product

- $A = [x_1, y_1, z_1]$, $B = [x_2, y_2, z_2]$
- Dot product $A \bullet B = x_1x_2+y_1y_2+z_1z_2$
- Cross product A x B =

$[y_1z_2 - z_1y_2, z_1x_2 - x_1z_2, x_1y_2 - y_1x_2]$





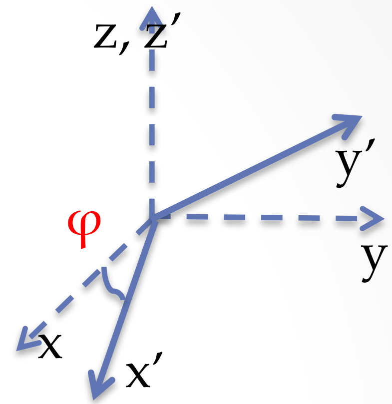


$$(x_1 i + y_1 j + z_1 k)\text{x}(x_2 i + y_2 j + z_2 k)$$
$$= (y_1 z_2 - z_1 y_2)i + (z_1x_2 - x_1z_2)j + (x_1 y_2 - y_1 x_2)k$$

Euler Angles and Rotation Matrices

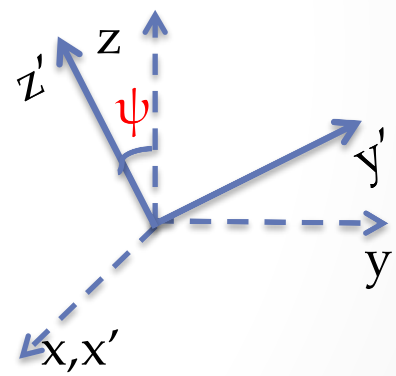
- A rotation of φ radians about the z-axis (yaw/azimuth)

$$R_z(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



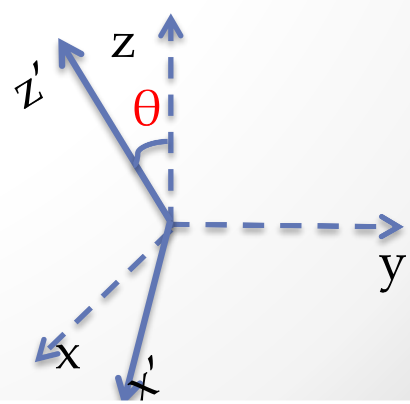
- A rotation of ψ radians about the x-axis (roll)

$$R_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix}$$



- A rotation of θ radians about the y-axis (pitch)

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$



Example

- $\Phi = \pi/2$

$$R_z(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Sequence of Rotations

- Represented by matrix product

$$
\begin{aligned}
R &= R_z(\phi)R_y(\theta)R_x(\psi) \\
&= \begin{bmatrix} \cos\theta\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \cos\theta\sin\phi & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi \\ -\sin\theta & \sin\psi\cos\theta & \cos\psi\cos\theta \end{bmatrix}
\end{aligned}
$$

- Note: not communicative (order matters!), not unique
  - Only 3 degree of freedom (DoF)

$$
R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}
$$

[x', y', z']' = R*[x,y,z]'

Computing Euler Angles

- If $R_{31} \neq \pm 1$

$$
\begin{aligned}
\theta_1 &= -\sin^{-1}(R_{31}) \\
\theta_2 &= \pi - \theta_1 = \pi + \sin^{-1}(R_{31})
\end{aligned}
$$

$$
\begin{aligned}
\psi_1 &= \operatorname{atan2}\left(\frac{R_{32}}{\cos\theta_1}, \frac{R_{33}}{\cos\theta_1}\right) \\
\psi_2 &= \operatorname{atan2}\left(\frac{R_{32}}{\cos\theta_2}, \frac{R_{33}}{\cos\theta_2}\right)
\end{aligned}
$$

$$
\begin{aligned}
\phi_1 &= \operatorname{atan2}\left(\frac{R_{21}}{\cos\theta_1}, \frac{R_{11}}{\cos\theta_1}\right) \\
\phi_2 &= \operatorname{atan2}\left(\frac{R_{21}}{\cos\theta_2}, \frac{R_{11}}{\cos\theta_2}\right)
\end{aligned}
$$

$$
\operatorname{atan2}(y,x) = \begin{cases}
\arctan(\frac{y}{x}) & \text{if } x > 0, \\
\arctan(\frac{y}{x}) + \pi & \text{if } x < 0 \text{ and } y \geq 0, \\
\arctan(\frac{y}{x}) - \pi & \text{if } x < 0 \text{ and } y < 0, \\
+\frac{\pi}{2} & \text{if } x = 0 \text{ and } y > 0, \\
-\frac{\pi}{2} & \text{if } x = 0 \text{ and } y < 0, \\
\text{undefined} & \text{if } x = 0 \text{ and } y = 0.
\end{cases}
$$

- Both (θ_1,ψ_1,ϕ_1) and (θ_2,ψ_2,ϕ_2) are valid solutions
-

Loss of DoF in Euler Angles

- If $R_{31} = \pm 1$, R_{11}, R_{21}, R_{32}, $R_{33} = 0$
  - e.g., $\theta = \pi/2$, $\sin\theta = 1$, $\cos\theta = 0$

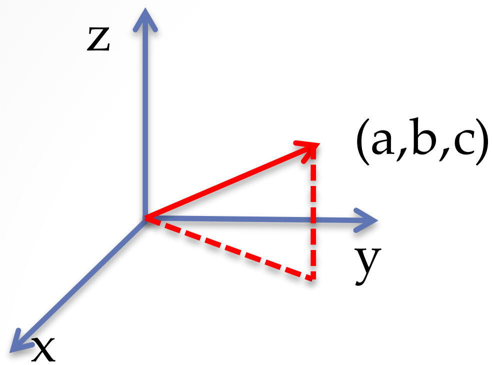
$$\begin{aligned} R &= R_z(\phi) R_y(\theta) R_x(\psi) \\ &= \begin{bmatrix} \cos\theta\cos\phi & \sin\psi\sin\theta\cos\phi - \cos\psi\sin\phi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \cos\theta\sin\phi & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi \\ -\sin\theta & \sin\psi\cos\theta & \cos\psi\cos\theta \end{bmatrix} \end{aligned}$$

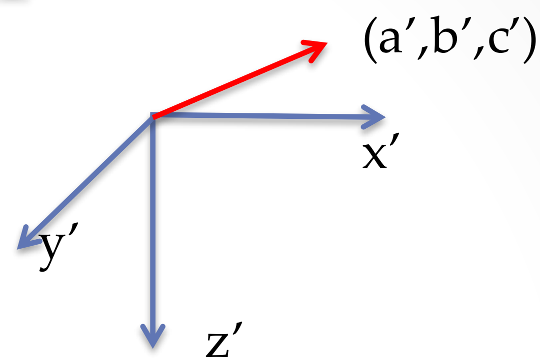
$$\Downarrow$$

$$R = \begin{bmatrix} 0 & sin(\psi - \phi) & cos(\psi - \phi) \\ 0 & cos(\psi - \phi) & -sin(\psi - \phi) \\ -1 & 0 & 0 \end{bmatrix}$$

- Infinite # of solutions!

Example



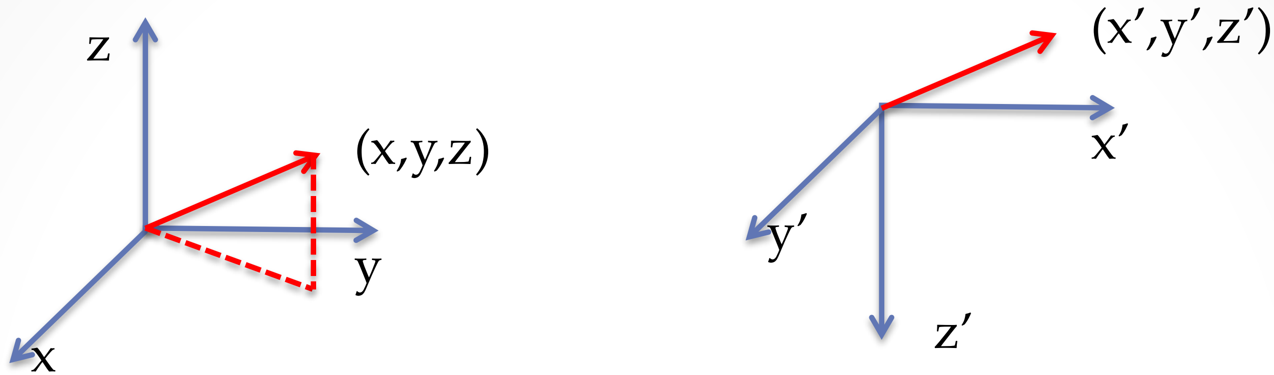


- What is the relation between (a,b,c) and (a',b', c')
- Rotation of the coordination system: rotate π/2 around z and then rotate π around x
- Equivalently, the vector rotates -π/2 around z and then rotate π around x

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \bullet \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}$$

$R_x(\pi)$ $\quad$ $R_z(-\pi/2\ )$

Example (cont'd)



- What is the relation between (a,b,c) and (a',b', c')?
- (1, 0, 0) → (0, 1, 0); (0, 1, 0) → (1, 0, 0); (0, 0, 1) → (0, 0, -1);

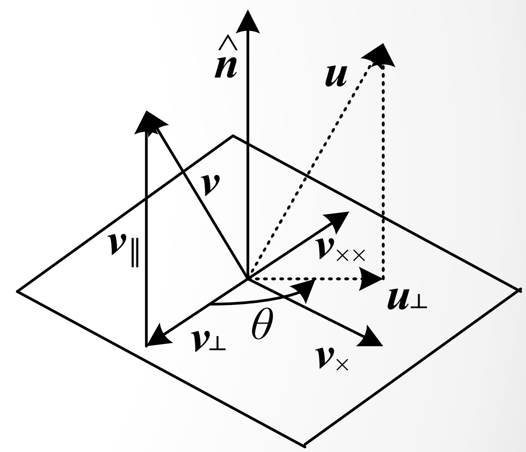
$$\begin{pmatrix} a' \\ b' \\ c' \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & -1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} a' \\ b' \\ c' \end{pmatrix}$$

Axis/angle Representation

- A rotation can be represented by a rotation axis $\hat{n} = [\hat{n}_x, \hat{n}_y, \hat{n}_z]$ and an angle θ
- Rotation matrix known as Rodriguez's formula

$$\boldsymbol{R}(\hat{\boldsymbol{n}}, \theta) = \boldsymbol{I} + \sin\theta[\hat{\boldsymbol{n}}]_\times + (1 - \cos\theta)[\hat{\boldsymbol{n}}]^2_\times,$$



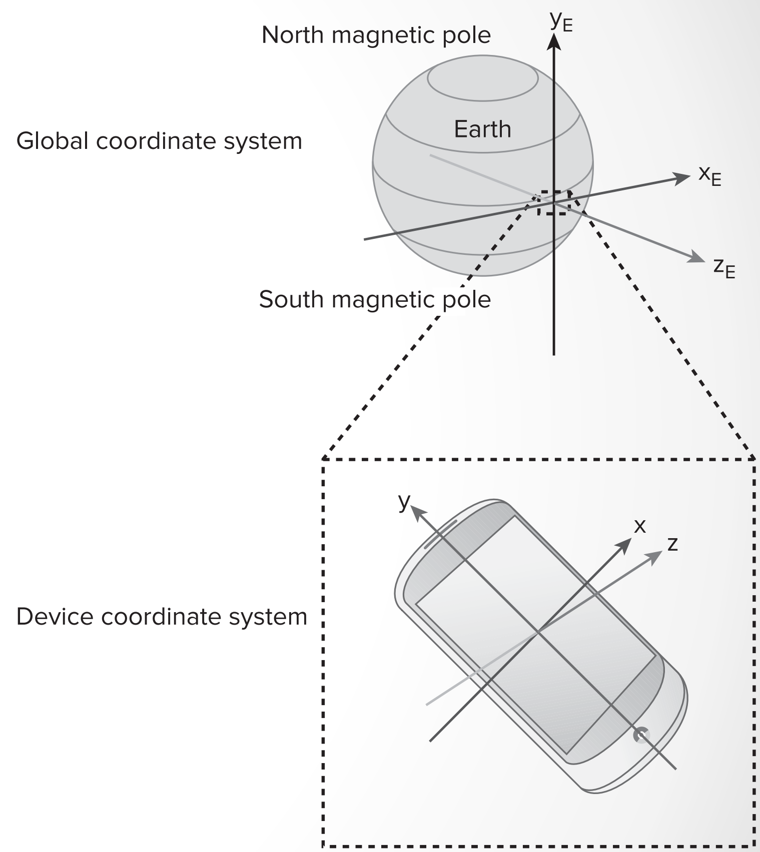
where

$$[\hat{\boldsymbol{n}}]_\times = \begin{bmatrix} 0 & -\hat{n}_z & \hat{n}_y \\ \hat{n}_z & 0 & -\hat{n}_x \\ -\hat{n}_y & \hat{n}_x & 0 \end{bmatrix}$$

how many DoF?

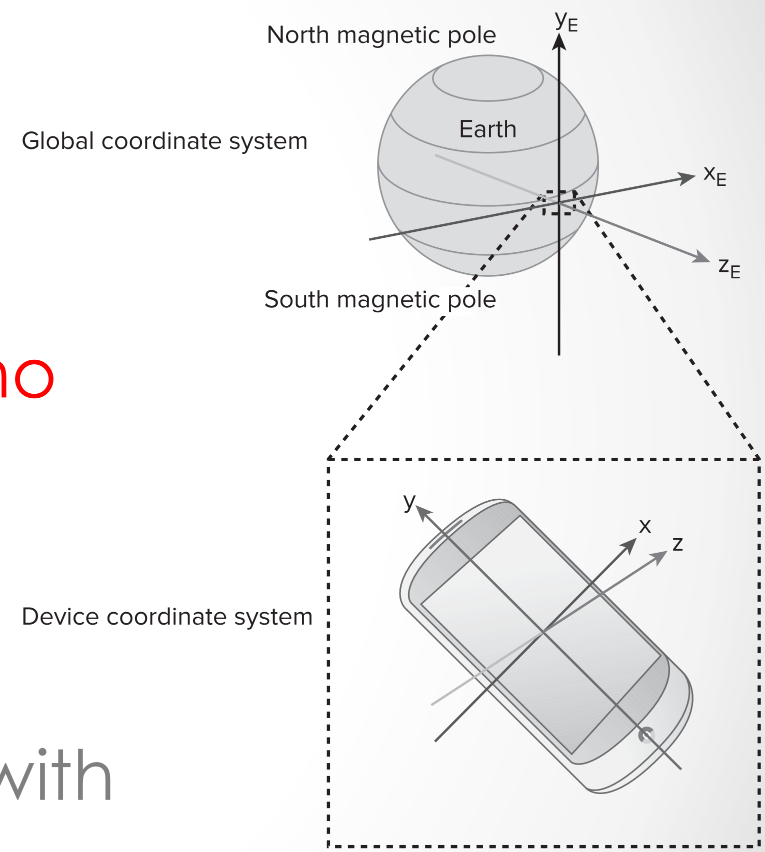
Determining Device Attitude (I)

- Now that we know how to represent rotation, next step is how to infer device attitude (e.g., the rotation matrix that transform a vector from the device frame to the world's [global] coordinate system)
- This is non-trivial due to magnetic interference from the environment
  - We do not always know "true north"



Determining Device Attitude (II)

- Given <acc_x, acc_y, acc_z> and <mag_x, mag_y, mag_z> from the accelerometer and the magnetometer (in device coordinate)
- Assume 1) stationary device, 2) no magnetic interference, 3) not in north pole
- How to determine the rotation matrix?
- If the device coordinates aligns with the global coordinates what the readings should be?
  - <0, 0, g>, <0, mag_y, 0>



Determining Rotation Matrix

$A = \langle acc_x, acc_y, acc_z \rangle$
$E = \langle mag_x, mag_y, mag_z \rangle$

↓

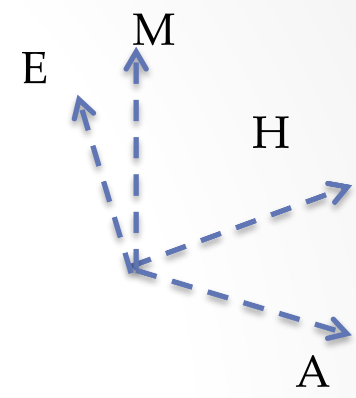
$H = E \times A$

↓

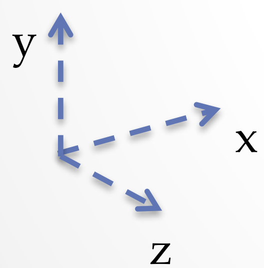
Normalize A, H → A′, H′;
$M = A' \times H'$

↓

$$R = \begin{pmatrix} h_1' & h_2' & h_3' \\ m_1 & m_2 & m_3 \\ a_1' & a_2' & a_3' \end{pmatrix}$$



World coordinate

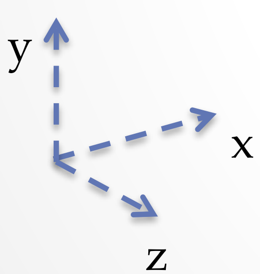


transpose $R' = R^{-1}$

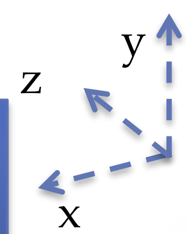
Android Specifics

- Device frame depends on the default orientation
    - Phone – Portrait
    - Table – landscape
    - Device coordinate frame differs from screen coordinate frame: getRotation() and remapCoordinateSystem()
- getOrientation (do not confuse with rotation): return the azimuth (Z), pitch (X), and yaw (Y) wrt inverted world coordinate frame

World coordinate



Inverted world coordinate for orientation (rotate π)



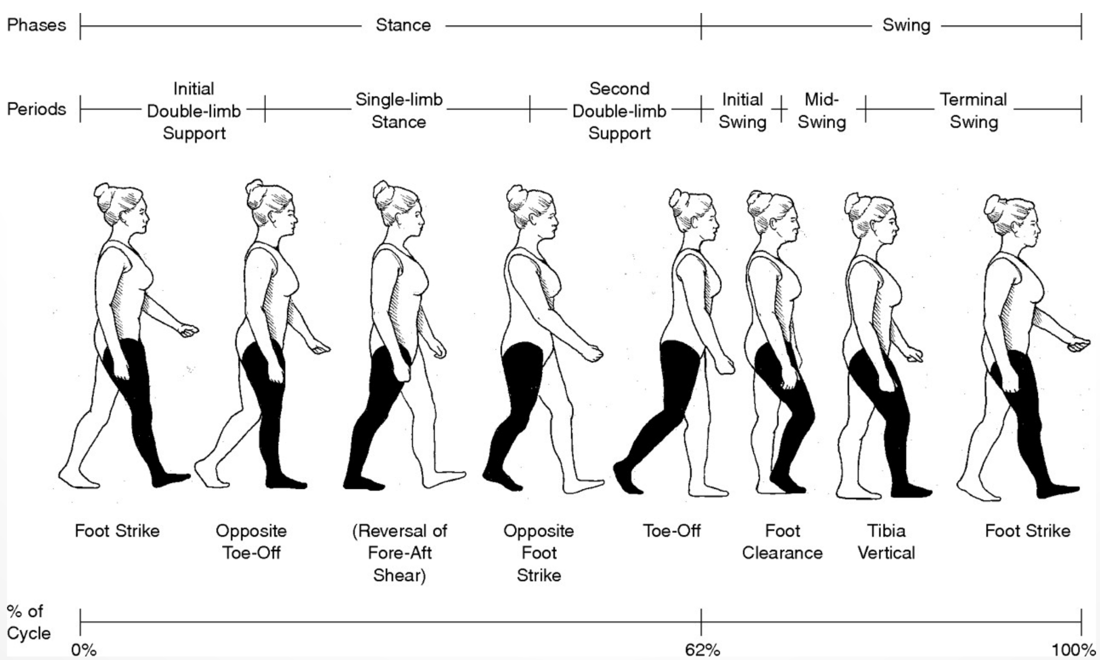
$$R' = \begin{pmatrix} -h_1' & -h_2' & -h_3' \\ m_1 & m_2 & m_3 \\ -a_1' & -a_2' & -a_3' \end{pmatrix}$$

Implementation Notes

- The above rotation matrix R allows transformation from device coordinates to world coordinates
  - R^{-1} is needed for the opposite and the transpose $R' = R^{-1}$
- For stationary devices, may average or apply a low pass filter to get the average acceleration and magnetometer readings (more later)
- Recall the 3 conditions 1) stationary device, 2) no magnetic interference, 3) not in north pole
  - Stationarity can be reasonably inferred from the magnitude of acceleration
  - Near north pole → GPS or magnetometer readings
  - No magnetic interferences difficult to guarantee

Step Counts

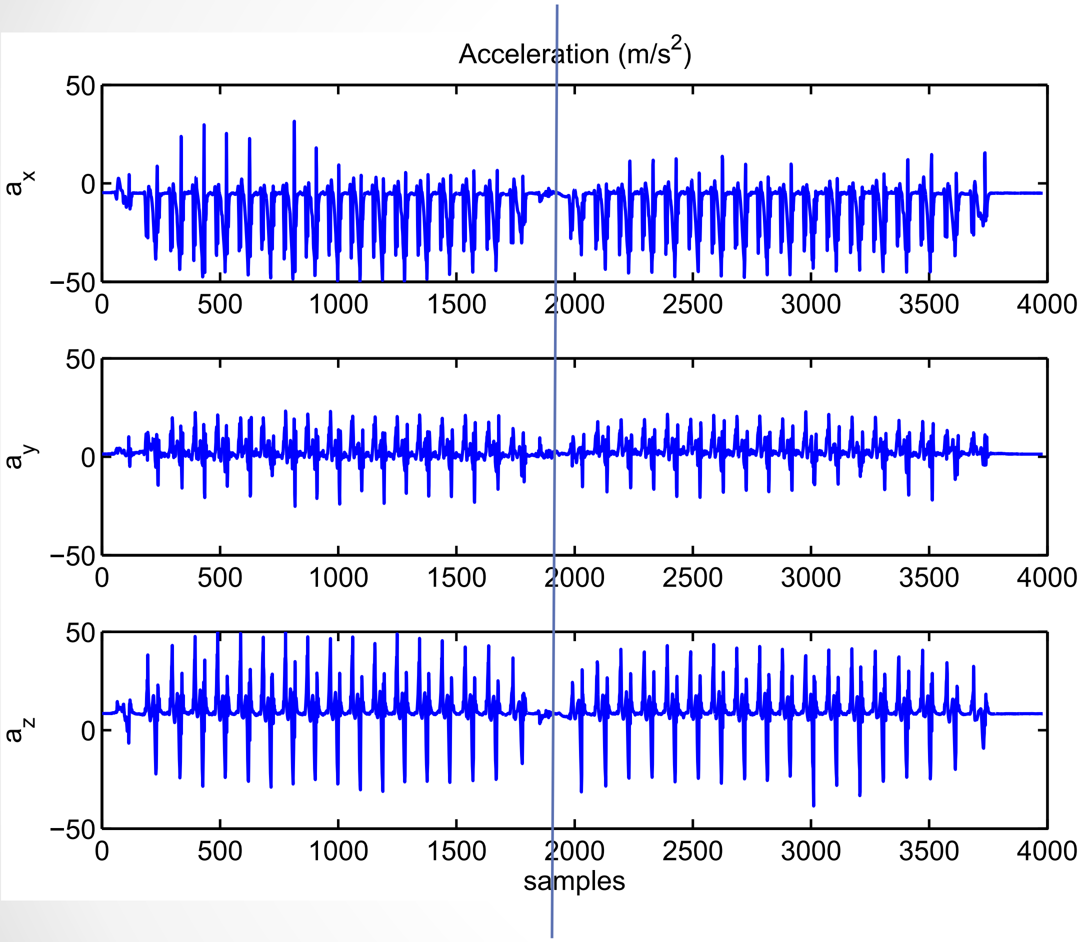
- Nowadays most wearable can do step counts – some are more accurate than the others
  - Why? How?
- Human gait cycle
  - Stance phase: when one foot touch the ground
  - Swing phase: the foot leaves the ground
  - [in jogging/running, both feet may be off the ground]



More Terminologies

- Stride: two consecutive heel strike of the same foot
- Stride length: distance traveled in one stride
- Step: successive heel strikes of opposite feet
- Step length: distance between heel strike of one limb and heel strike of the other limb
- Step width: distance we keep our feet apart when we walk (2 – 4 inches)
- Cadence: Walking speed, or number of steps taken per minute

Step Counting Using Accelerometer Data



- Acceleration and deceleration easily identifiable along all axises
- Noisy

Turn

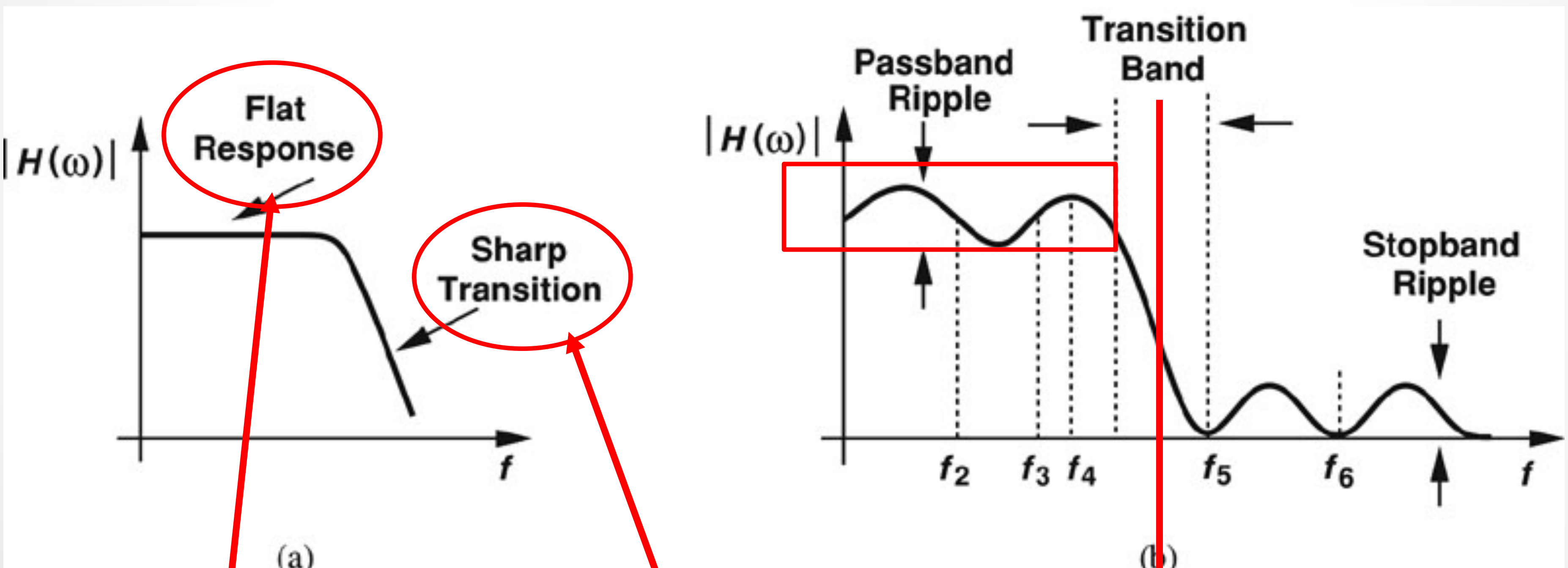
A.R. Jiménez, F. Seco, C. Prieto and J. Guevara, A Comparison of Pedestrian Dead-Reckoning Algorithms using a Low-Cost MEMS IMU

Algorithmic Sketch

Compute linear acceleration	Remove the gravity component
Compute magnitude	$acc = \sqrt{acc_x^2 + acc_y^2 + acc_z^2}$
Apply low-pass filter (LPF)	
Square, LPF	Optional
Find Peaks	Peak to peak → one step

Low-pass Filter (LPF)

- A low-pass filter is a filter that passes signals with a frequency lower than a certain cutoff frequency f_c and attenuates signals with frequencies higher than the cutoff frequency



Must not alter
the desired signal!

Sharp Transition
in order to attenuate
the interference

Affect selectivity

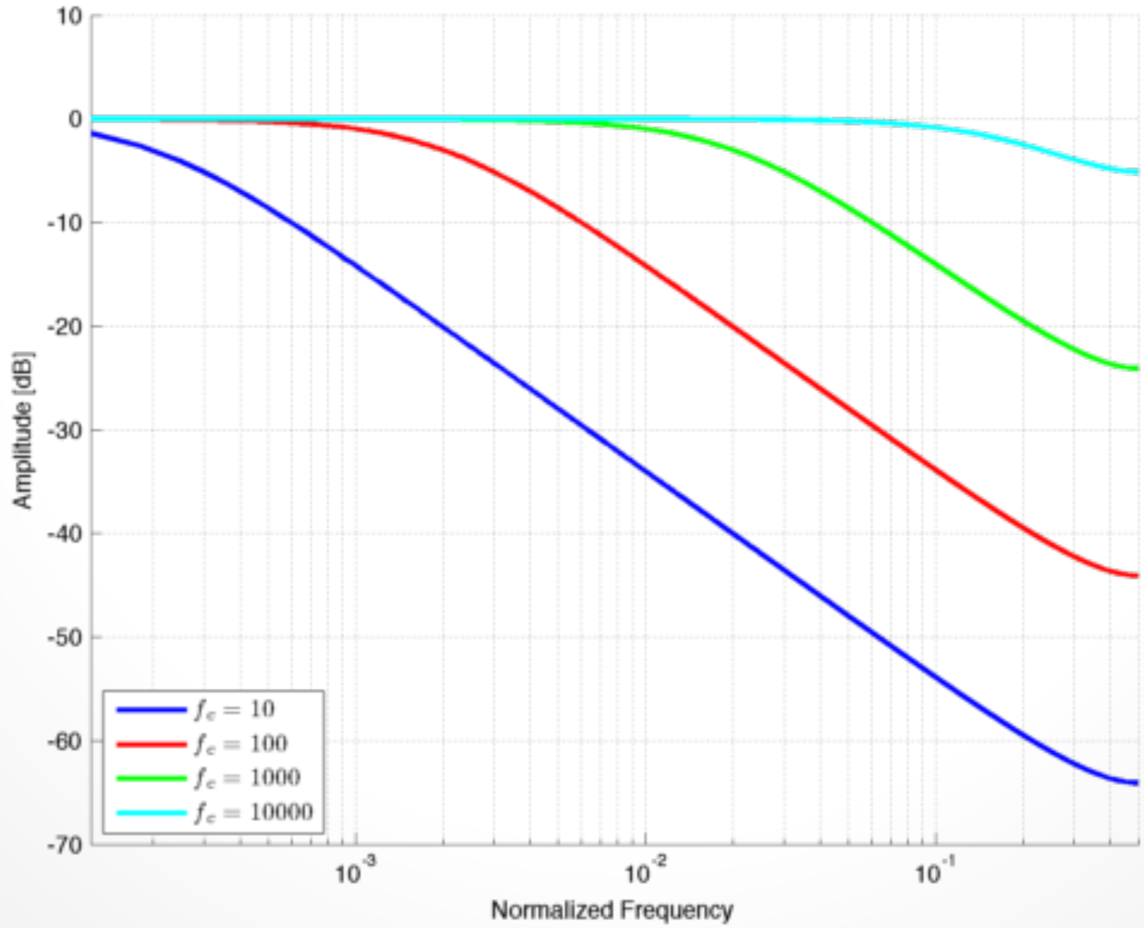
Two Types of Digital LPFs

- Finite impulse response filter (FIR)
  - $y_n = \sum_{k=0}^{P} a_k x_{n-k}$
  - Transfer function $H(z) = \sum_{k=0}^{P} a_k z^{-k}$
  - No feedback
  - Roughly linear phase

- Infinite impulse response filter (IIR)
  - $\sum_{l=0}^{Q} a_l y_{n-l} = \sum_{k=0}^{P} b_k x_{n-k}$
  - Transfer function $H(z) = \frac{\sum_{l=0}^{P} b_l z^{-l}}{\sum_{k=0}^{Q} a_k z^{-k}}$
  - With feedback
  - Can match a particular freq response with relatively fewer parameters than FIR (more computationally efficient)

Commonly Used LPFs

- Exponential moving average (EMA)
  - y(n) = (1-a)x(n)+ay(n-1)
  - Bigger a → more history; smaller a → more current
  - Approximately, a ≈ $\exp(-2\pi f_C/f_s)$, f_C cut-off freq, f_s sampling freq



Commonly Used LPFs

- Butterworth filter
  - Can use Matlab [B, A] = butter(N, W_n, 'low')

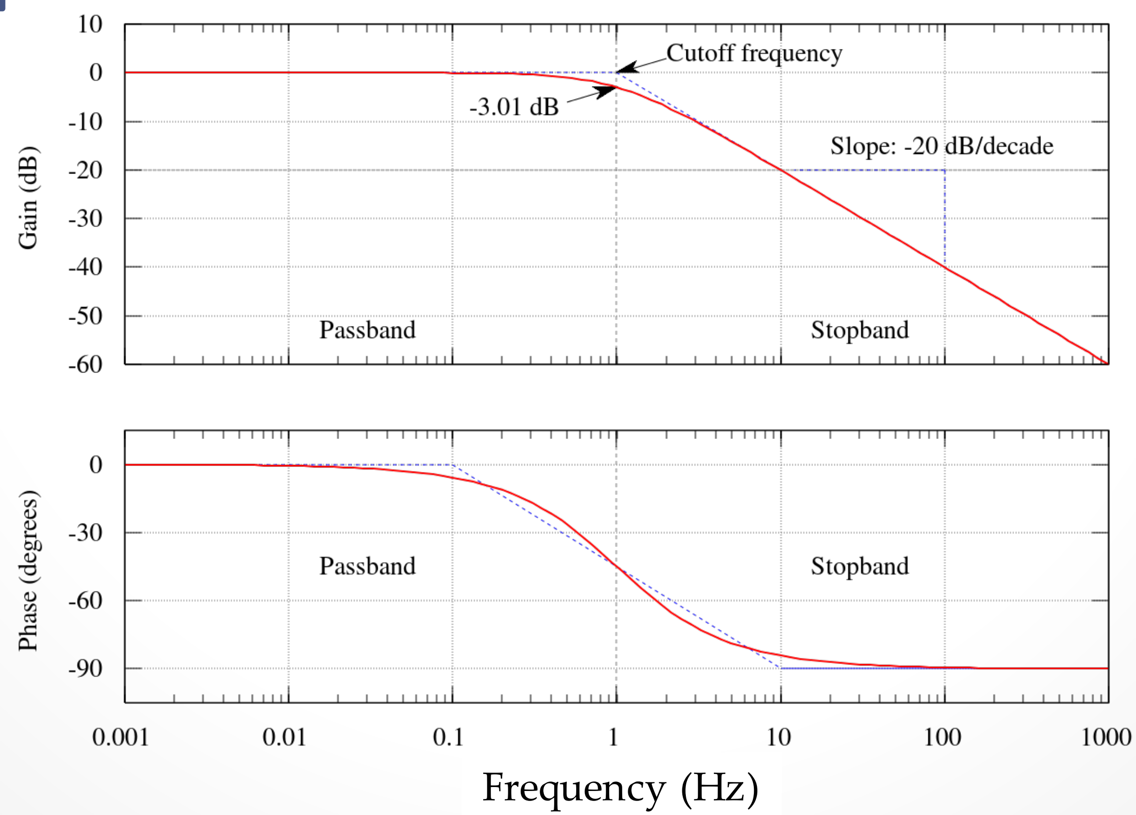
numerator

denumerator

Order

Cut-off freq

0.0 < W_n < 1.0, with 1.0 corresponding to half the sample rate.



Cut-off Frequency?

- Informed guess
  - Step frequency 1 – 3 Hz for walking

Step frequency. Normal gait. Men.

Age years	N	Mean steps/s
10-14	12	2.14
15-19	15	2.02
20-29	15	1.98
30-39	15	2.00
40-49	15	2.01
50-59	15	1.96
60-69	15	1.95
70-79	14	1.91

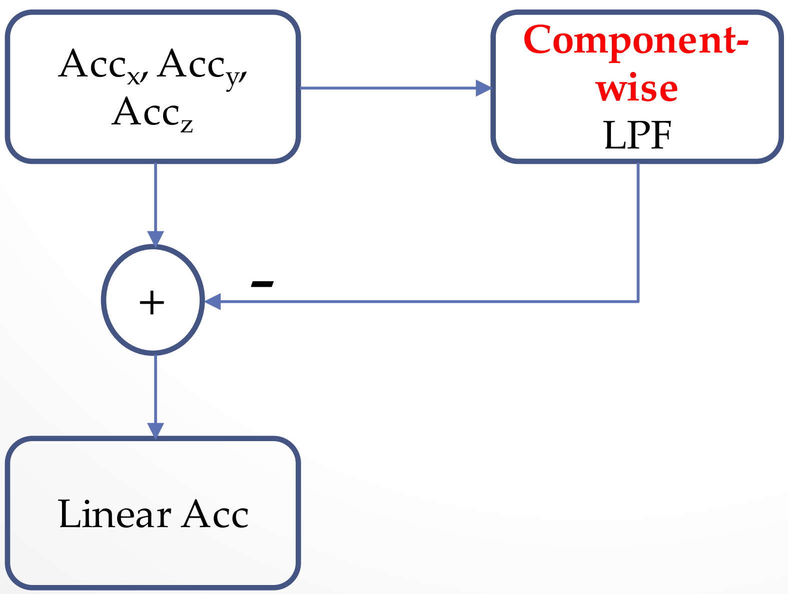
Step frequency. Normal gait. Women.

Age years	N	Mean steps/s
10-14	12	1.97
15-19	15	2.09
20-29	15	2.08
30-39	15	2.13
40-49	15	2.16
50-59	15	2.03
60-69	15	2.06
70-79	15	2.03

- Frequency domain analysis (Libby'09)
  - Perform DFT and find the frequency f_c, where the x% of total energy fall below f_c

Linear Acceleration

- Goal: to remove the (constant) gravity component from acceleration measurements due to motion
- Idea A: gravity is constant
  - apply a high-pass filter
  - Or, apply a low-pass filter and then subtract the resulting signal from the raw signal
  - Cutoff frequency?
    - 0.1Hz would be reasonable for walking. At 50Hz sampling frequency, this is equivalent to a weight $a = \exp(-2\pi f_c/f_s) = 0.9875$

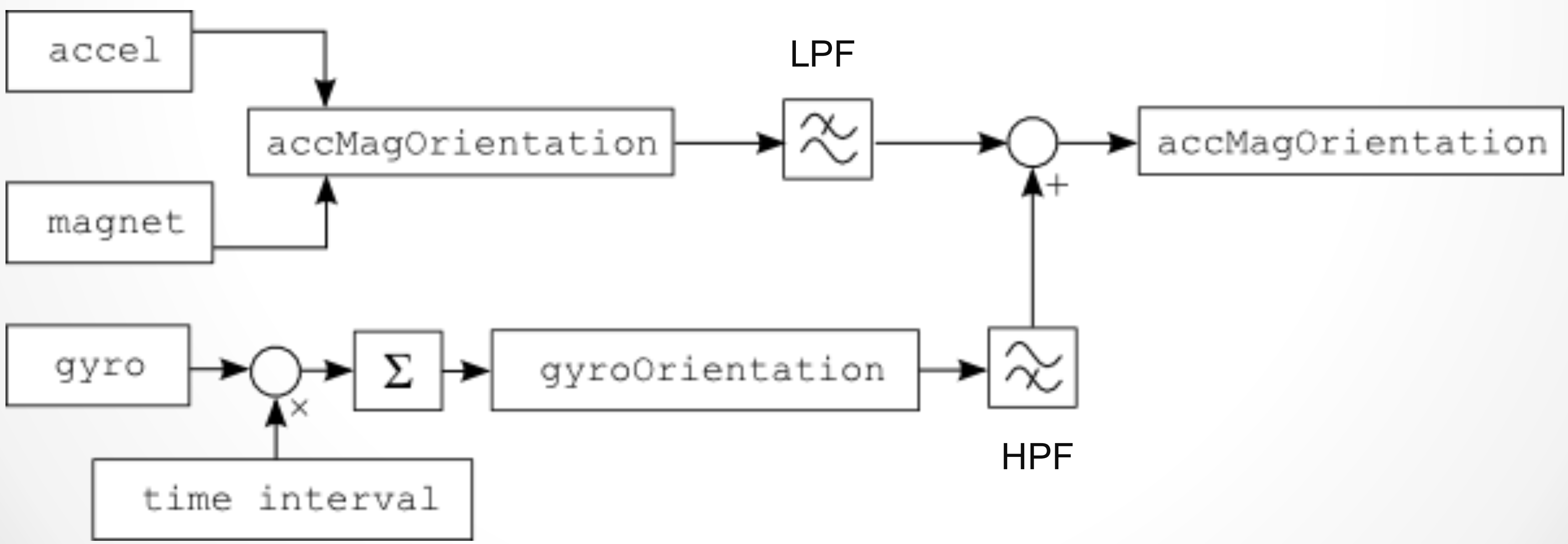


Linear Acceleration (Cont'd)

- Idea B:
  - Determine the tilt angle
  - Subtract the gravity components
  - A rather complex method but it demonstrates how to use gyro to determine device orientation changes during motion
- Recall Gyro measures Corilios force due to rotation, or angular velocity
  - In the device frame
  - $< \omega_x(t), \omega_y(t), \omega_z(t) >$ represents the angular velocity around x, y, z axis at time t
  - Then, the angular changes are $\Delta_x = \omega_x(t)dt$, $\Delta_y = \omega_y(t)dt$, $\Delta_z = \omega_z(t)dt$
  - $\Delta = \sqrt{\Delta_x^2 + \Delta_y^2 + \Delta_z^2}$, $\overline{\Delta_x} = \Delta_x/\Delta$, $\overline{\Delta_y} = \Delta_y/\Delta$, $\overline{\Delta_z} = \Delta_z/\Delta$,

Linear Acceleration (Cont'd)

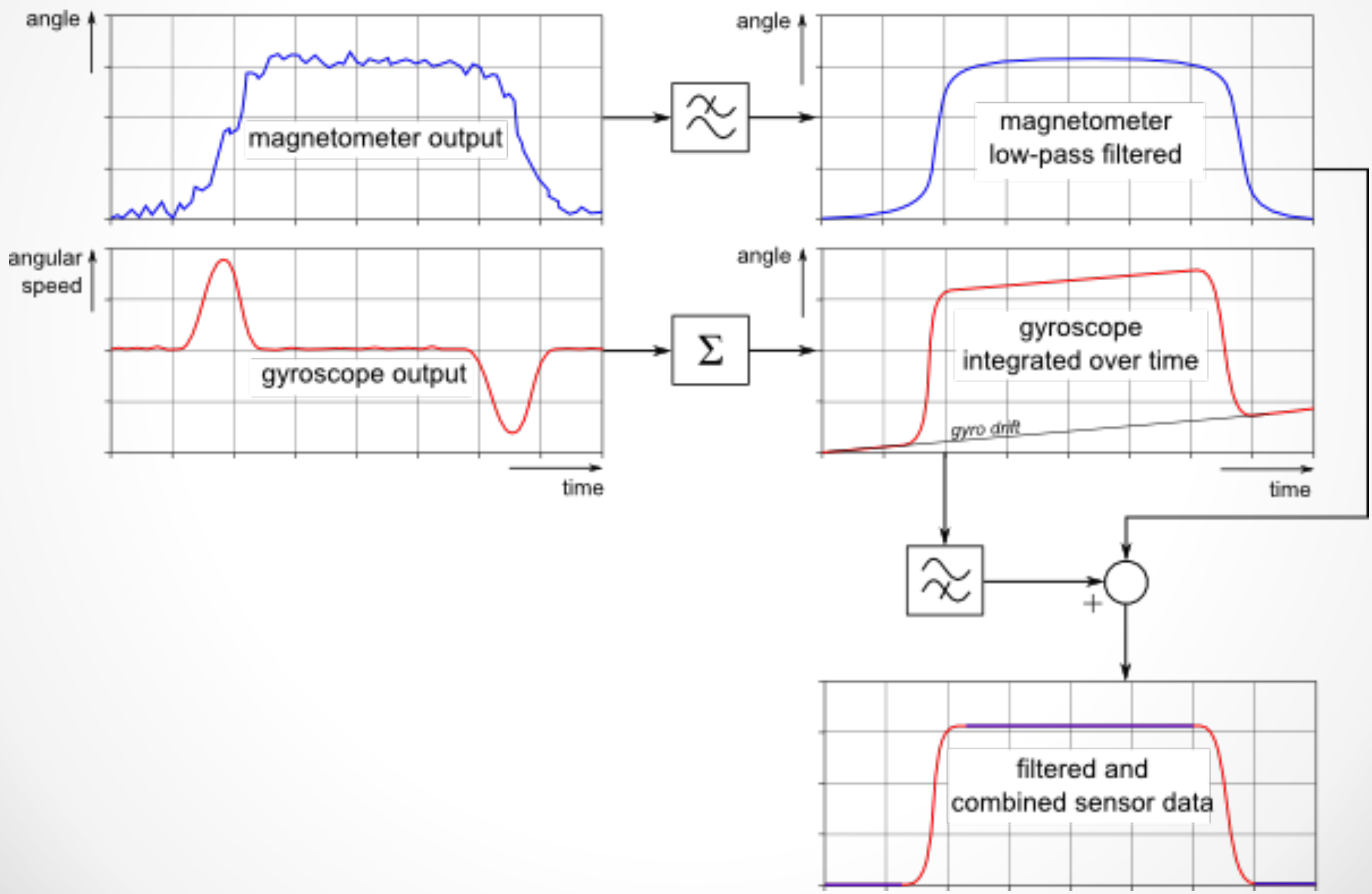
- Derive rotation matrix
- If initial orientation is known → new orientation
  - Can optionally combine the orientation from gyro and the orientation estimated from accelerometer and compass (see below)
- Finally, given the orientation, we can subtract the gravity from accelerometer data



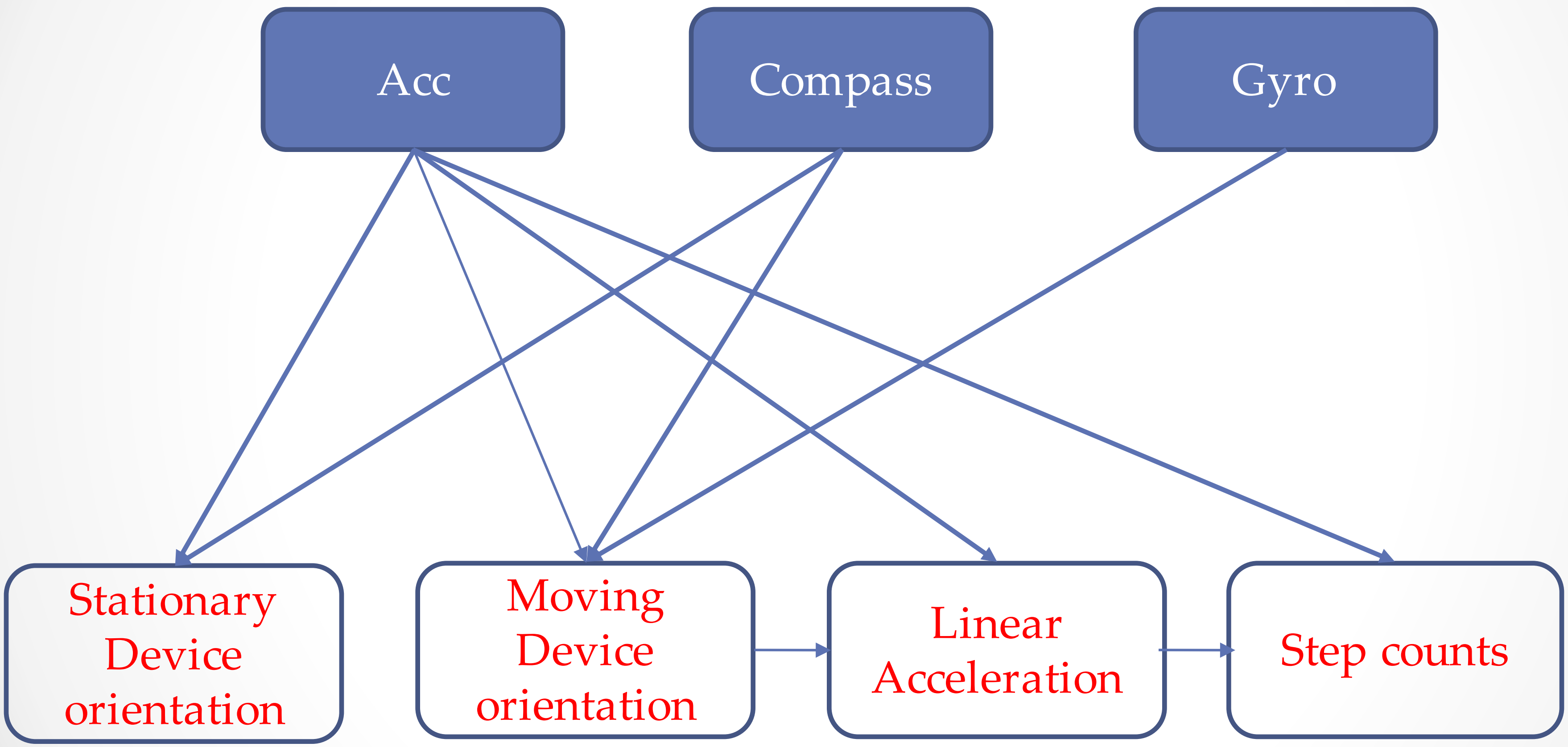
Complimentary filter for device orientation

http://www.codeproject.com/Articles/729759/Android-Sensor-Fusion-Tutorial

Complementary Filters



Summary


Acc
Compass
Gyro
Stationary Device orientation
Moving Device orientation
Linear Acceleration
Step counts

Further Reading

- Greg Milette, Adam Stroud, "Professional Android Sensor Programming", 2012
- Gregory G. Slabaugh, “Computing Euler angles from a rotation matrix”
- A.R. Jime´nez, F. Seco, C. Prieto and J. Guevara, A Comparison of Pedestrian Dead-Reckoning Algorithms using a Low-Cost MEMS IMU