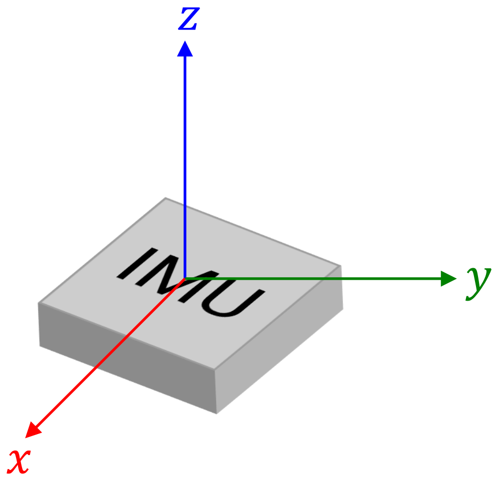
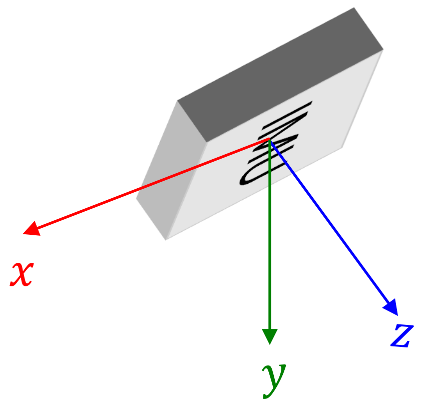
ENAE 788M
Hands-On Autonomous Aerial Robotics

IMU BASICS, ATTITUDE ESTIMATION USING COMPLEMENTARY AND MADGWICK FILTERS

What is an IMU?



What is an IMU?



Can measure

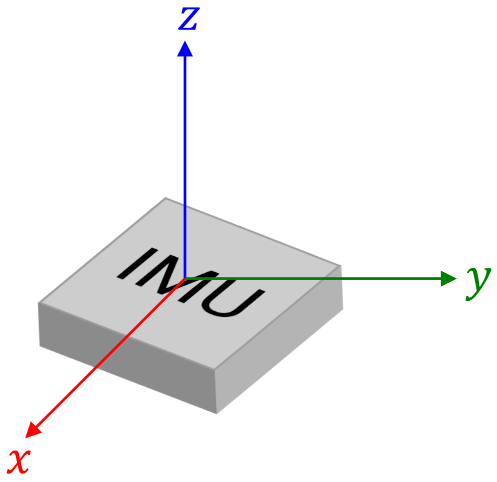
- Angular Velocities (Gyroscope) ω
- Linear Accelerations (Accelerometer) a
- Magnetic Field (Magnetometer/Compass) m

A combination of Gyroscope and Accelerometer is called an Inertial Measurement Unit or IMU or a 6-DoF IMU

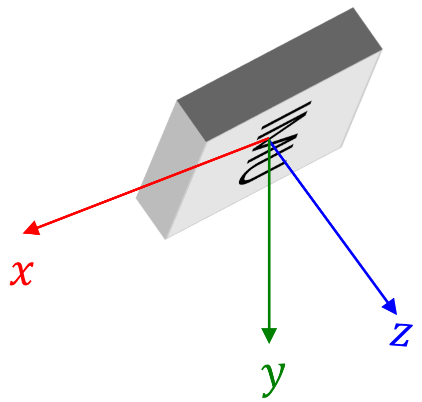
A combination an IMU and a Compass is called a Magnetic, Angular rate and Gravity sensor or MARG or Attitude and Heading Reference System or AHRS or a 9-DoF IMU

We will be using a simple 6-DoF IMU for attitude estimation

Attitude Estimation



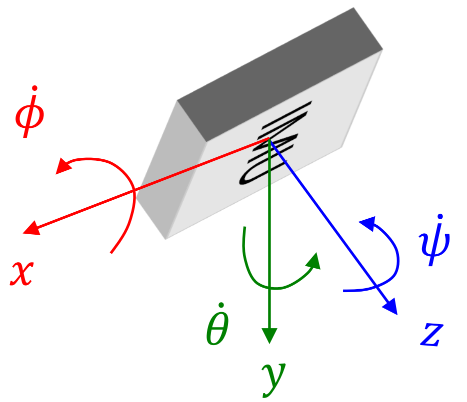
Attitude Estimation



Estimate the orientation in World frame!
Orientation is also called **Attitude**

Orientation can be parametrized as either Euler Angles, Roll Pitch and Yaw or Quaternions

Attitude Estimation from an Ideal Gyroscope



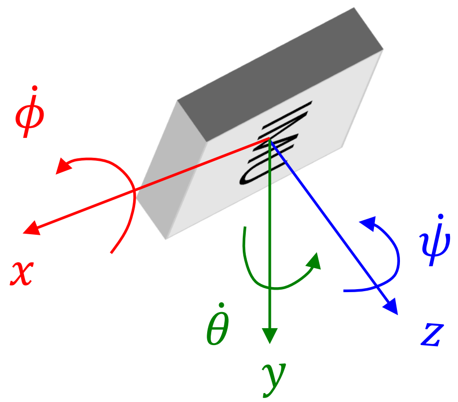
We want to estimate $[\phi \quad \theta \quad \psi]_{t+1}{}^{T}$
We have $\omega_t = [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}\,]_t{}^{T}$

How do we do it?
Numerical Integration!

There's a catch!
Impossible to obtain Attitude without knowing Initial conditions as Numerical Integration is not possible!

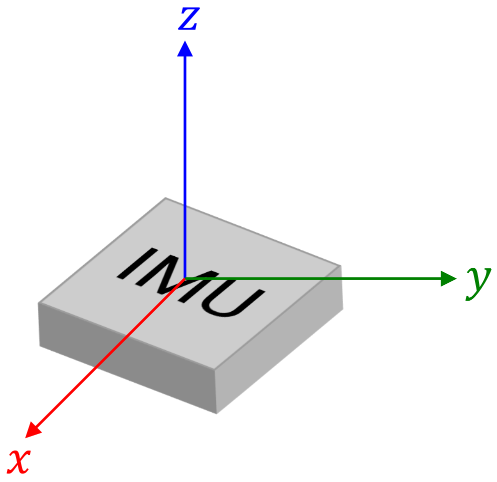
Assume that initial conditions are known, how do you get an initial condition?
Other sensors, starting from rest and so on!

Attitude Estimation from an Ideal Gyroscope

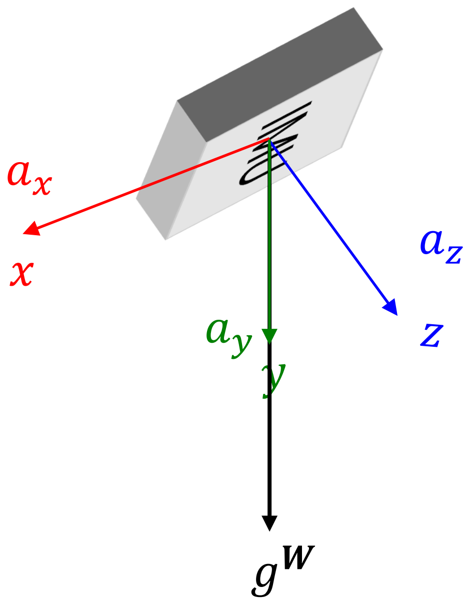


$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}_{t+1} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}_{t} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}_{t} \delta t$$

Attitude Estimation from an Ideal Accelerometer



Attitude Estimation from an Ideal Accelerometer



Again, let's assume that our IMU/Accelerometer is only rotating!

We want to estimate $[\phi \quad \theta \quad \psi]_{t+1}{}^T$

We have $a_t = [a_x \quad a_y \quad a_z]_t{}^T$

How do we do it?

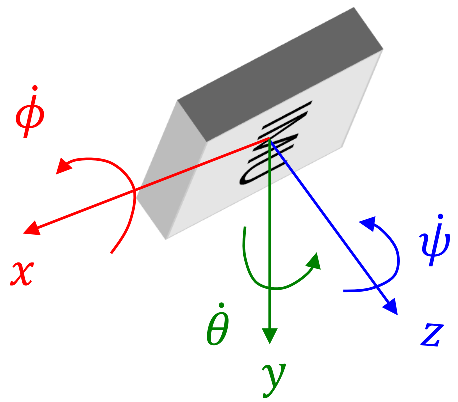
Gravity to the rescue!

$$\phi = \tan^{-1}\left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}}\right)$$

$$\theta = \tan^{-1}\left(\frac{a_x}{\sqrt{a_y^2 + a_z^2}}\right)$$

$$\psi = \tan^{-1}\left(\frac{\sqrt{a_x^2 + a_y^2}}{a_z}\right)$$

Welcome To The Real World!

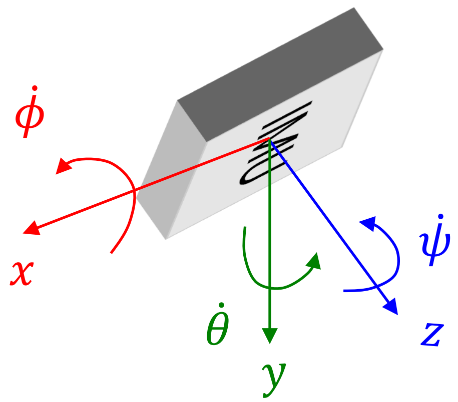


Both Gyroscope and Accelerometer have noise and bias!

Bias: They don't read zero at rest!

Noise: The values don't remain constant when the sensor is not moving!

Gyroscope: Mathematical Model



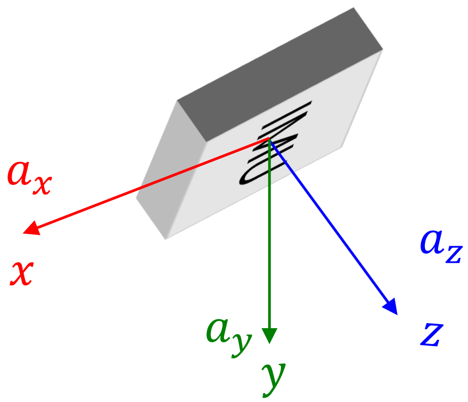
$$\underbrace{\omega}_{\text{Measured Value}} = \overbrace{\widehat{\omega}}^{\text{Ideal Value}} + \underbrace{b_g}_{\text{Bias}} + \overbrace{n_g}^{\text{White Gaussian Noise}}$$

$$\dot{b}_g \sim \mathcal{N}\left(0, \mathbf{Q}_g\right)$$

Bias will overshadow real values when integrating if not compensated for!

Noise will make values drift from real value during integration!

Accelerometer: Mathematical Model



Ideal Value

White Gaussian Noise

$$a = {}^{W}\boldsymbol{R}_{\boldsymbol{B}}^{T}\left(\hat{a} - g^{\boldsymbol{W}}\right) + b_a + n_a$$

Measured Value

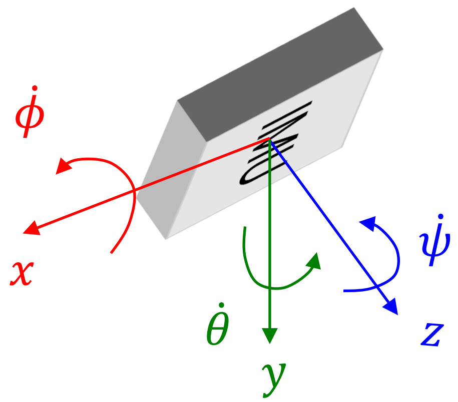
Bias

$$\dot{b}_a \sim \mathcal{N}(0, \mathbf{Q}_a)$$

Accelerometer measures net linear acceleration on the device! Values change when device moves (not just rotates)!

Not great for fast movement as linear acceleration due to translation and rotation get mixed up!

Bias and Noise



Bias varies on device start and with external factors such as temperature!

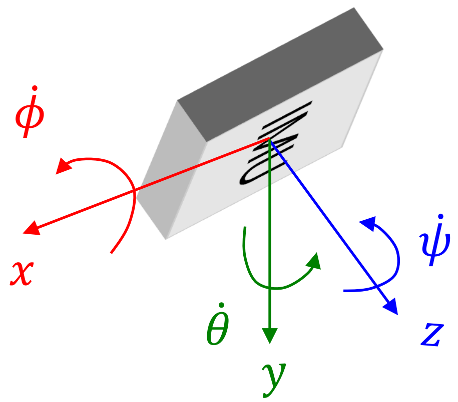
Can be estimated by as a function of expectation of values when device is at rest

Noise varies with external factors such as temperature!

Can be estimated as variance of values when device is at rest

Note that both Bias and Noise change over time and need to be estimated on-the-fly

Attitude Estimation from a Real Gyroscope



We want to estimate $[\phi \quad \theta \quad \psi]_{t+1}{}^{T}$

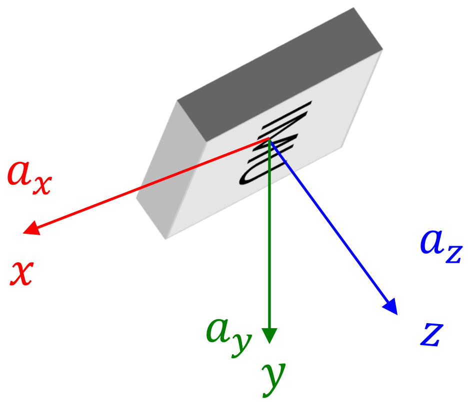
We have $\omega = [\dot{\phi} \quad \dot{\theta} \quad \dot{\psi}]_{t}{}^{T}$

First, estimate Bias

Remove Bias from Initial value (Rest) and Numerically Integrate

Good for fast movement! Drifts like crazy over time!

Attitude Estimation from a Real Accelerometer



We want to estimate $[\phi \quad \theta \quad \psi]_{t+1}{}^T$

We have $a = [a_x \quad a_y \quad a_z]_t{}^T$

First, estimate Bias

Remove Bias from Initial value (Rest) and estimate orientation by decomposing vector a

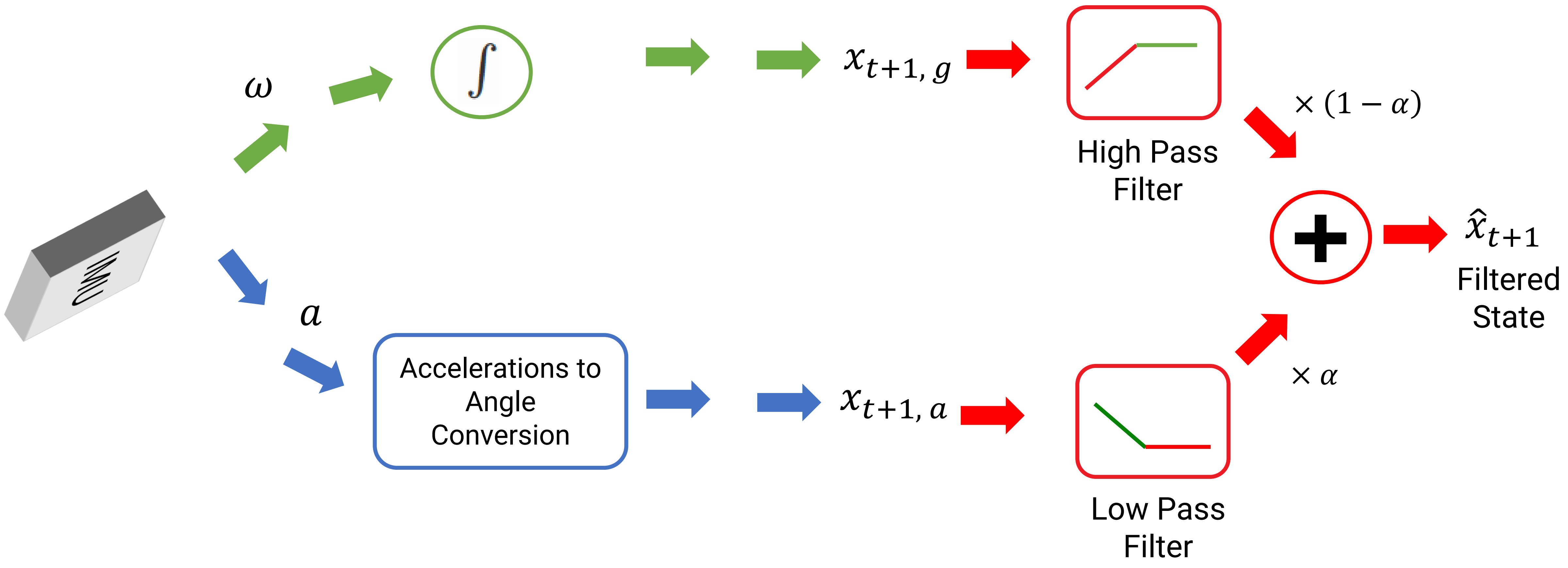
Low pass-filter the values for a better estimate (this will cause the filter values to lag a bit)

$$\hat{x}_{t+1} = (1-\gamma)\hat{x}_t + \gamma\tilde{x}_{t+1}$$

$$\gamma \in [0,1]$$

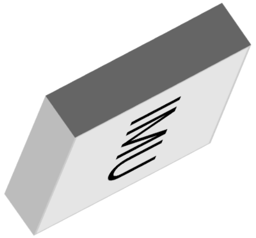
Generally α is chosen to be a small value like 0.2

Best of Both Worlds: Complementary Filter



A Better Way: Madgwick Filter

Gyroscope Model



Let's represent orientation/attitude as a quaternion

$$q = [q_w \quad q_x \quad q_y \quad q_z]$$

Now, given gyroscope measurements ω_{t+1} let's estimate the **orientation increment**

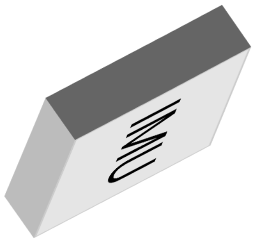
$$\frac{1}{2} {}^I_W\hat{\mathbf{q}}_{est,t} \otimes [0, {}^I\omega_{t+1}]^T$$

The above operation is just adding ω_{t+1} to q on the quaternion manifold

Sebastian OH Madgwick, Andrew JL Harrison, and Ravi Vaidyanathan. "*Estimation of IMU and MARG orientation using a gradient descent algorithm.*" *2011 IEEE international conference on rehabilitation robotics*. IEEE, 2011.

A Better Way: Madgwick Filter

Accelerometer Model



We want to model attitude estimation from the accelerometer measurements by modelling it as an optimization problem instead of direct angle computation

$$\min_{{}^I_W\hat{\mathbf{q}} \in \mathbb{R}^{4\times 1}} f\left({}^I_W\hat{\mathbf{q}}, {}^W\hat{\mathbf{g}}, {}^I\hat{\mathbf{a}}\right)$$

$$f\left({}^I_W\hat{\mathbf{q}}, {}^W\hat{\mathbf{g}}, {}^I\hat{\mathbf{a}}\right) = \widehat{{}^I_W\mathbf{q}^*} \otimes {}^W\hat{\mathbf{g}} \otimes {}^I_W\hat{\mathbf{q}} - {}^I\hat{\mathbf{a}}$$

We can achieve this using Gradient Descent!

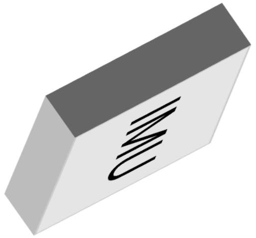
The gradient update is given in the next slide

A Better Way: Madgwick Filter

Accelerometer Model

$$\min_{{}^I_W\hat{\mathbf{q}} \in \mathbb{R}^{4\times1}} f\left({}^I_W\hat{\mathbf{q}}, {}^W\hat{\mathbf{g}}, {}^I\hat{\mathbf{a}}\right)$$

$$f\left({}^I_W\hat{\mathbf{q}}, {}^W\hat{\mathbf{g}}, {}^I\hat{\mathbf{a}}\right) = {}^I_W\hat{\mathbf{q}}^* \otimes {}^W\hat{\mathbf{g}} \otimes {}^I_W\hat{\mathbf{q}} - {}^I\hat{\mathbf{a}}$$



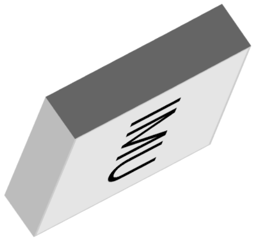
We can achieve this using Gradient Descent!
Compute gradient as follows:

$$\nabla f\left({}^I_W\hat{\mathbf{q}}_{est,t}, {}^W\hat{\mathbf{g}}, {}^I\hat{\mathbf{a}}_{t+1}\right) = J^T\left({}^I_W\hat{\mathbf{q}}_{est,t}, {}^W\hat{\mathbf{g}}\right) f\left({}^I_W\hat{\mathbf{q}}_{est,t}, {}^W\hat{\mathbf{g}}, {}^I\hat{\mathbf{a}}_{t+1}\right)$$

$$f\left({}^I_W\hat{\mathbf{q}}_{est,t+1}, {}^W\hat{\mathbf{g}}, {}^I\hat{\mathbf{a}}_{t+1}\right) = \begin{bmatrix} 2\left(q_2q_4 - q_1q_3\right) - a_x \\ 2\left(q_1q_2 + q_3q_4\right) - a_y \\ 2\left(\frac{1}{2} - q_2^2 - q_3^2\right) - a_z \end{bmatrix} \quad J\left({}^I_W\hat{\mathbf{q}}_{est,t+1}, {}^W\hat{\mathbf{g}}\right) = \begin{bmatrix} -2q_3 & 2q_4 & -2q_1 & 2q_2 \\ 2q_2 & 2q_1 & 2q_4 & 2q_3 \\ 0 & -4q_2 & -4q_3 & 0 \end{bmatrix}$$

A Better Way: Madgwick Filter

Accelerometer Model



The gradient update is given as follows:

$$-\beta \frac{\nabla f\left({}^I_W\hat{\mathbf{q}}_{est,t+1}, {}^W\hat{\mathbf{g}}, {}^I\hat{\mathbf{a}}_{t+1}\right)}{\left|\left|f\left({}^I_W\hat{\mathbf{q}}_{est,t+1}, {}^W\hat{\mathbf{g}}, {}^I\hat{\mathbf{a}}_{t+1}\right)\right|\right|}$$

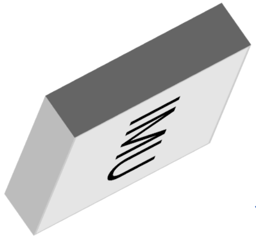
Making certain assumptions about noise, only **one step** of gradient update is enough for convergence!

β is a tunable parameter which models the magnitude of gyroscope error in the direction of accelerometer measurements

The Madgwick Filter

$${}^I\mathbf{a}_{t+1}$$

$$\nabla f\left({}^I_W\hat{\mathbf{q}}_{est,t}, {}^W\hat{\mathbf{g}}, {}^I\hat{\mathbf{a}}_{t+1}\right) = J^T\left({}^I_W\hat{\mathbf{q}}_{est,t}, {}^W\hat{\mathbf{g}}\right) f\left({}^I_W\hat{\mathbf{q}}_{est,t}, {}^W\hat{\mathbf{g}}, {}^I\hat{\mathbf{a}}_{t+1}\right)$$



$$-\beta\frac{\nabla f\left({}^I_W\hat{\mathbf{q}}_{est,t+1}, {}^W\hat{\mathbf{g}}, {}^I\hat{\mathbf{a}}_{t+1}\right)}{||f\left({}^I_W\hat{\mathbf{q}}_{est,t+1}, {}^W\hat{\mathbf{g}}, {}^I\hat{\mathbf{a}}_{t+1}\right)||}$$

$${}^I\omega_{t+1}$$

$$\frac{1}{2}\,{}^I_W\hat{\mathbf{q}}_{est,t} \otimes \left[0, {}^I\omega_{t+1}\right]^T$$

$${}^I_W\dot{\mathbf{q}}_{\omega,t+1}$$

$$+$$

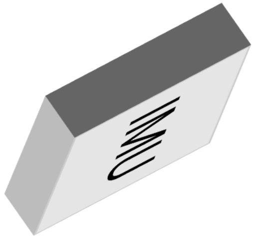
$${}^I_W\dot{\mathbf{q}}_{est,t+1}$$

$$\int$$

$$\Delta t$$

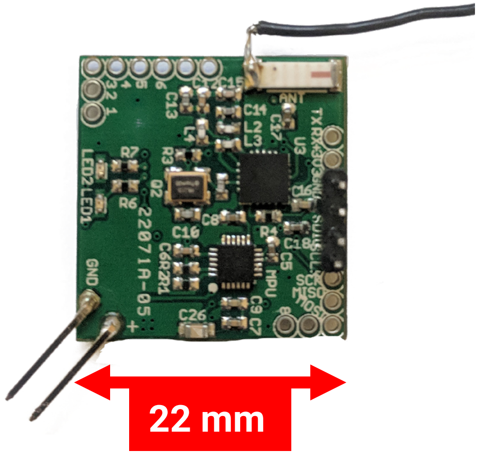
$${}^I_W\mathbf{q}_{est,t+1}$$

The Madgwick Filter



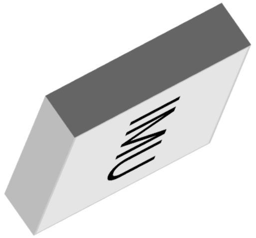
Advantages:
- Easy math
- Converges in one step of gradient descent
- Tracks orientation in quaternion space
- Runs on small embedded hardware like ELKA-R

Disadvantages:
- Doesn’t explicitly track bias
- Cannot explicitly model system dynamics



ELKA-R
Autopilot

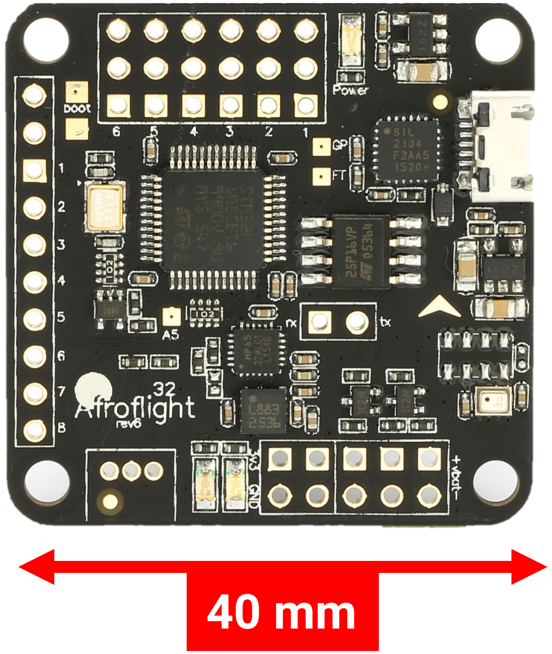
Better Filters: Bayesian Based



Better Filtering can be performed by the class of Bayesian Filters

Some of the common Bayesian filters are
- Kalman Filter
- Extended Kalman Filter
- Unscented Kalman Filter

Newer and more powerful computers can run these filters on-board



Naze32
Autopilot