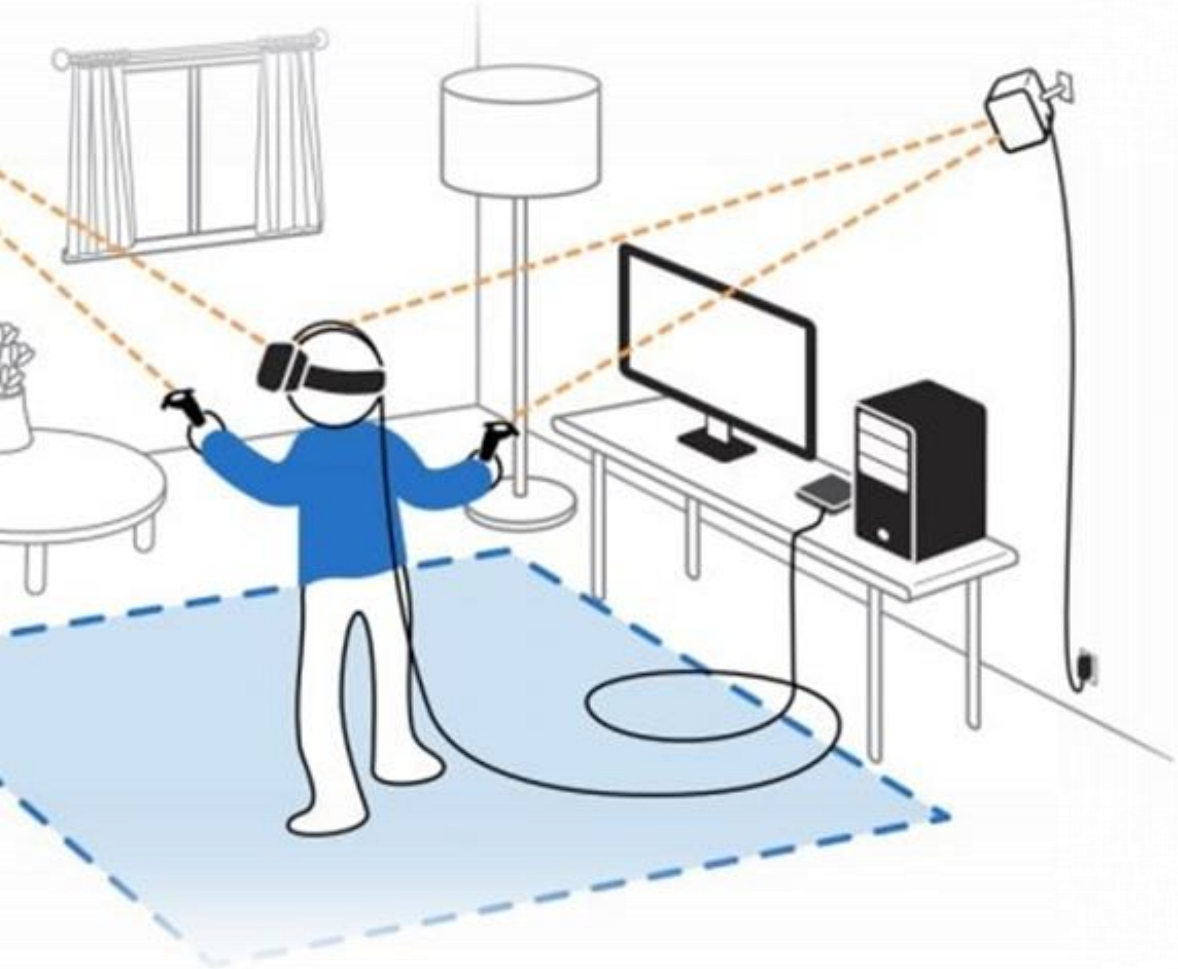
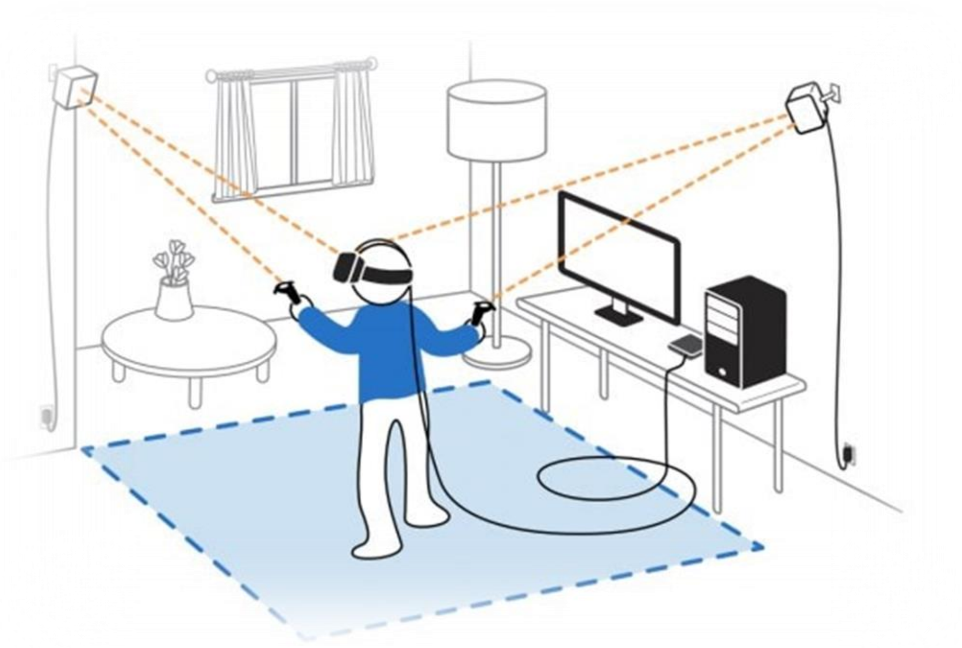


Tracking Systems I

Isayas Adhanom, Computer Science

What is tracking?

- Keeping track of motion and objects of interest in the physical world.
- What do we need to track in VR?



Objects We Need Track

- The head
- The controllers/hand
- The eye
- The entire body
- Facial features
- Physical objects in the environment

Categories of Tracking in VR

The user's sense organs

Eg: eyes, ears

The user's other body parts

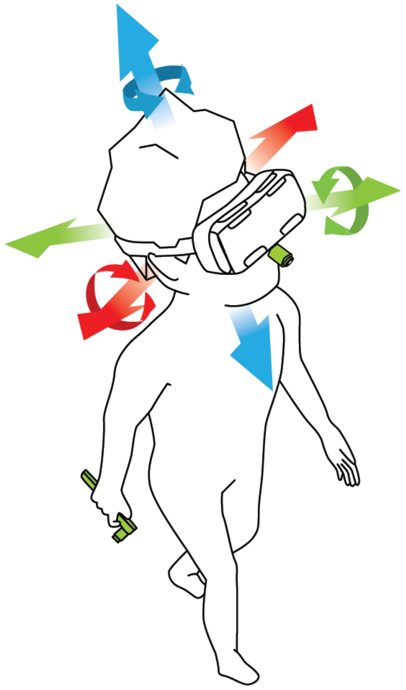
Eg: facial features, hand gestures

The rest of the environment

Eg: physical objects in the environment

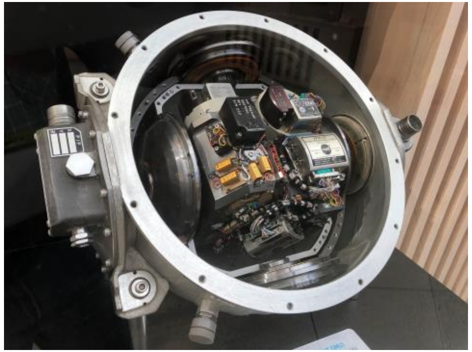
Tracking in VR

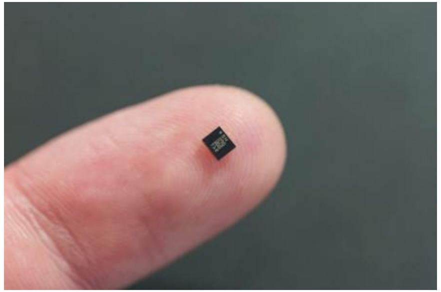
- Track orientation and position of an object.
- Orientation: yaw, pitch and roll movements
- Position: translational movements.



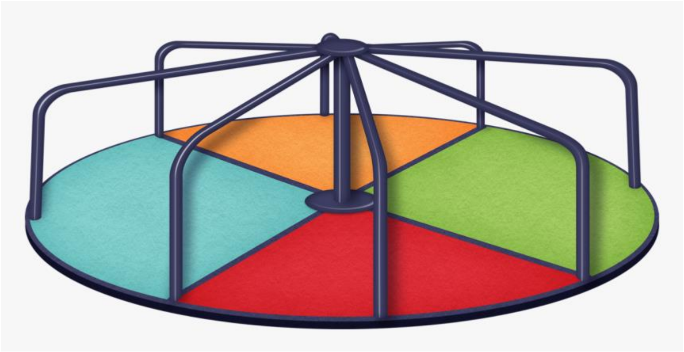
Tracking Orientation in VR

- The main application is determining the viewpoint orientation.
- Relies mainly on the angular velocity readings of an IMU.
- IMU are cheap, small and reliable.
- IMU’s use MEMS technology.

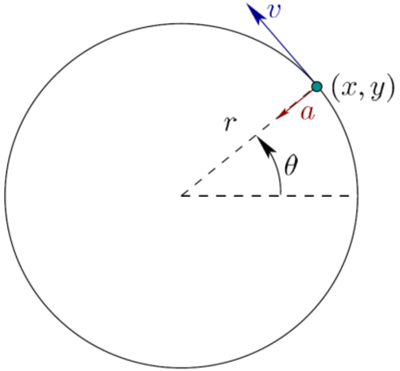




Tracking 2D Orientation



- We use the merry-go-round model.
- A disc rotating along a single axis with constant angular velocity.
- Estimating the angle of rotation based on gyroscope data is sufficient to estimate the orientation.
- Angular velocity $(\omega) = \frac{\theta}{t}$ and $\theta = \omega t$



Tracking 2D Orientation:

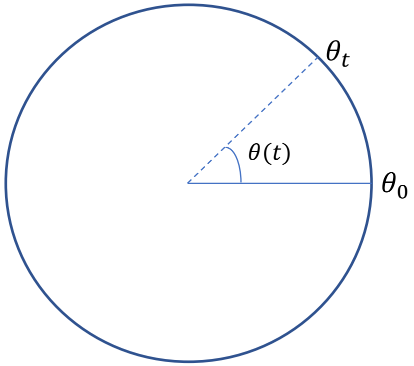
- For constant angular velocity (ω):

$$\text{At } t = 0, \qquad \theta = \theta_0$$

$$\text{At } t > 0, \qquad \theta = \theta_0 + \omega t$$

- For varying ω :

$$\theta(t) = \theta_0 + \int_o^t \omega(t) dt$$



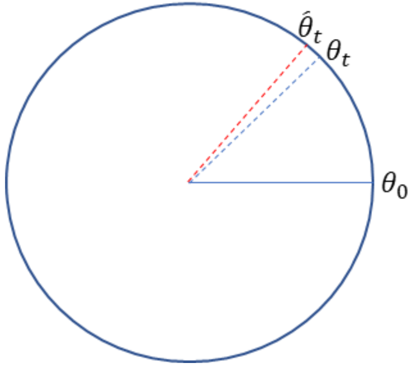
Calibration Error

- Sensors (e.g. gyroscopes) are not perfect.
- The sensor reading ($\widehat{\omega}$) may be different from the true value (ω).
- $\widehat{\omega}$ is the estimate and ω is the true value, and they are usually different.
- The main cause of the difference is *calibration error*.
- For a perfect sensor $\omega_i = \widehat{\omega}_i$
- $\widehat{\omega} \quad = a + b\omega$

Drift Error

- If we use the sensor to estimate the merry-go-round model:

$$d(t) = \theta(t) - \hat{\theta}(t)$$



- $d(t)$ is a function of time and is called **drift error**.
- Drift error is directly proportional angular velocity.
- For VR headsets tracking error increases as the head rotates more quickly.

General Tracking Problems

- Calibration: eliminating calibration errors.
- Integration: integrating measurements based on discrete sensor readings.
- Registration: determining the initial orientation of the tracked object.
- Drift Error: correcting tracking errors that grow over time.

Calibration

- Take many sample readings (thousands) and compare them to the readings from a more accurate sensor.
- Generate a transformation model (function) to map between the inaccurate and accurate readings.
- Transform each raw sensor reading using the transformation function.

Integration

- Sensor outputs usually arrive at a regular sampling rate.
- Discrete time approximation

Instead of this => $\hat{\theta}(t) = \theta_0 + \int_o^t \hat{\omega}_i(t)dt$

We use this => $\hat{\theta}(t) \approx \theta_0 + \sum_{i=1}^{k} \hat{\omega}_i \Delta t$

- Where Δt is fixed sampling rate. Usually 1ms.

Registration

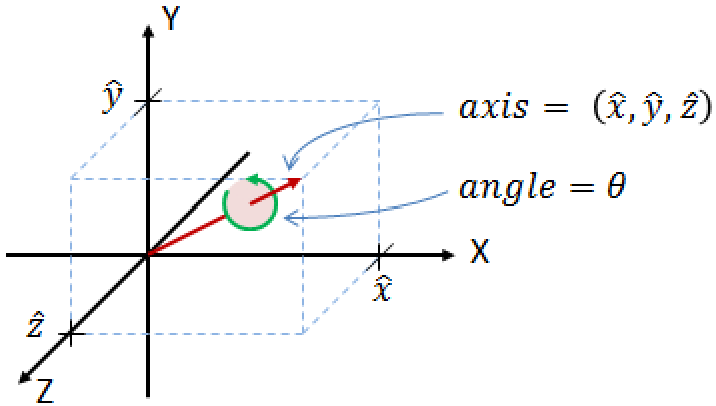
- The initial alignment between the real and virtual worlds.
- Determining the initial orientation θ_0
  - Initial direction of headset when it is turned on.
  - Initial direction of head set when put on head (we need “on head” sensor).
  - Can be defined as a specific direction in the physical world (eg: direction of monitor)
- Some headsets allow the initial orientation to be changed.

Drift Error

- We should not allow drift error to accumulate.
- The first problem is to estimate the drift error,
- We usually need an additional sensor to estimate drift error.
- We then need to remove the error using an approach called filtering.
- The correction should be done fast enough to correct drift but slow enough so that users do not notice it.

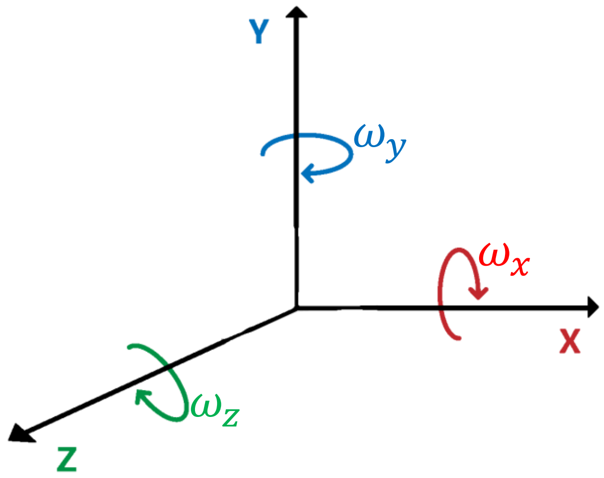
3D Rotation Representation

- Axis angle representation
- Can be represented by a single quaternion value: q = $(\vec{v},\ \theta)$
- We can then use quaternion product to apply transformation.



3D Gyroscope Readings

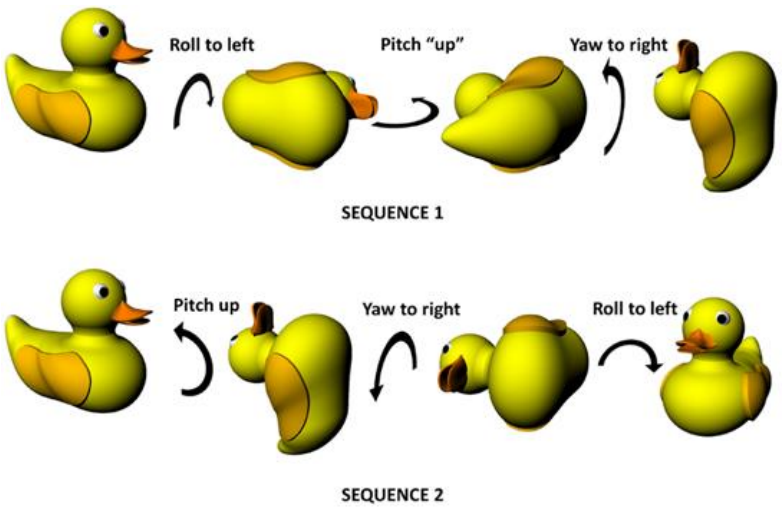
- The gyroscope provides angular velocity readings for the three axis: ω_x, ω_y, ω_z.
- The rotation can be represented using axis-angle representation $(\vec{v}_i,\ \Delta\theta_i)$ (refer chapter 3).
- Let Δq_i be the quaternion representation of $(\vec{v}_i,\ \Delta\theta_i)$.



Integrating Sensor Readings

- Recall our 2D orientation formula

$$\hat{\theta}(t) = \theta_0 + \sum_{i=1}^{k} \widehat{\omega}_i \Delta t$$
$$= \theta_0 + \hat{\theta}_1 + \ldots + \hat{\theta}_{k-1} + \hat{\theta}_k$$



- Our 3D orientation will be:

$$\boxed{\hat{q}_t = \Delta\hat{q}_k \circ \Delta\hat{q}_{k-1} \circ \ldots \circ \Delta\hat{q}_2 \circ \Delta\hat{q}_1 \circ \Delta\hat{q}_0}$$

Tracking Orientation Incrementally

- Tracking 2D orientation incrementally

$$\hat{\theta}_{current} = \Delta\hat{\theta}_k + \hat{\theta}_{previous}$$

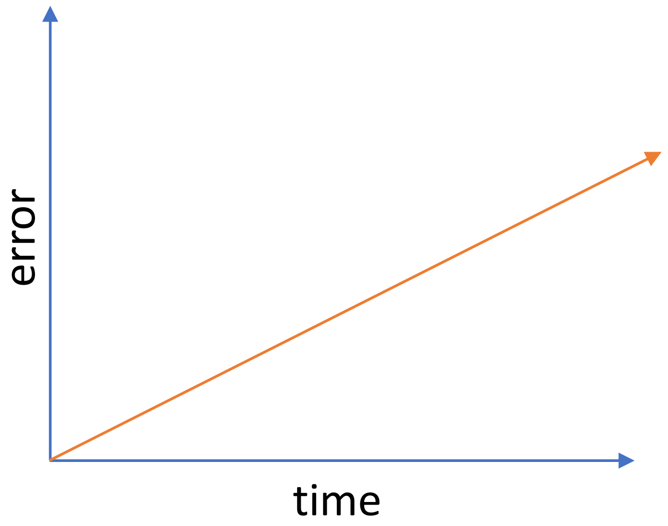
- Tracking 3D orientation incrementally

$$\hat{q}_{current} = \Delta\hat{q}_k \circ \hat{q}_{previous}$$

- What happens if we do this continuously?
    - We end up with big drift errors!

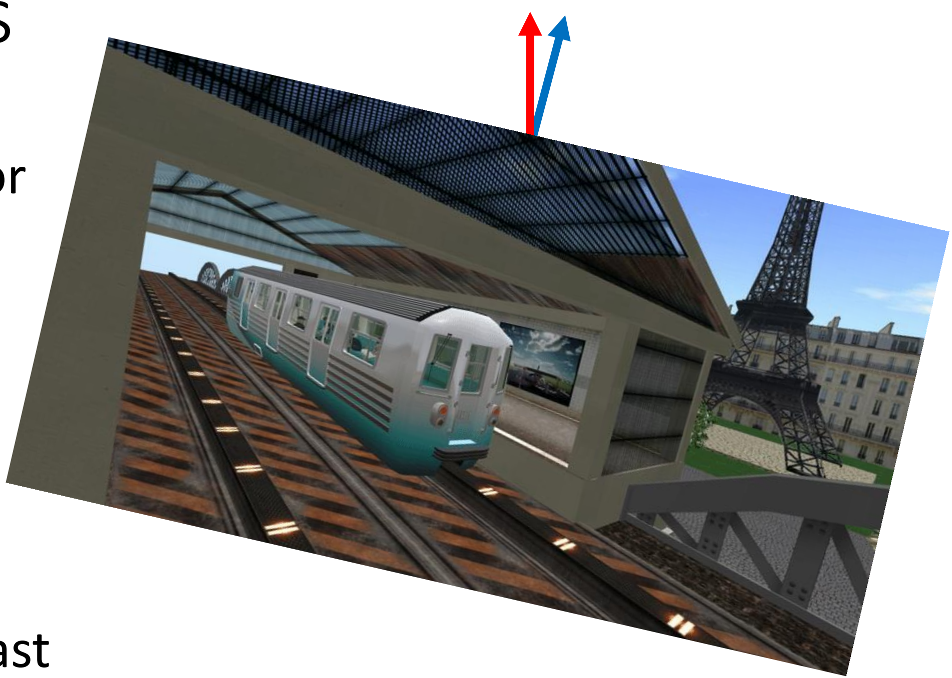
Drift Errors

- Also called dead reckoning errors.
- Drift error for 2D case:
  - $d_k = \theta_k - \hat{\theta}_k$
- Drift error for 3D case
  - $d_k = q_k \circ \hat{q}_k^{-1}$
- Drift error gets worse over time.



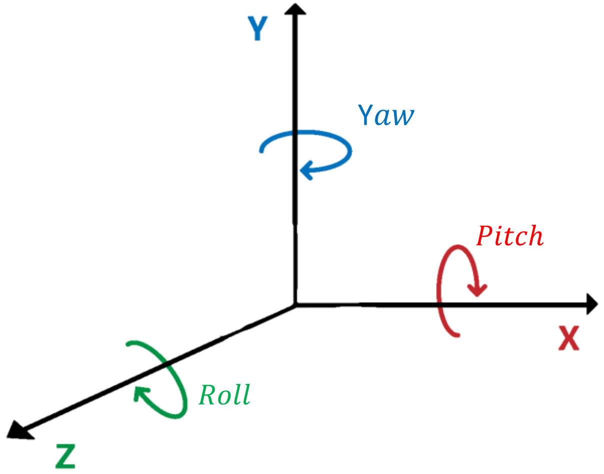
Correcting for Drift Errors

- The procedures we use to correct for drift errors are:
  - Use other sensors to provide a world reference
  - Gradually apply corrections (this is a difficult problem)
- Apply corrections so that they are fast enough to compensate drift, and slow enough to be imperceptible by the user.



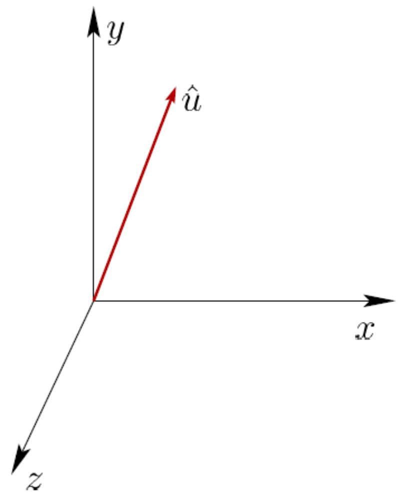
Correcting for Drift Errors

- Separate rotational error into two components:
  - Tilt error (pitch + roll) - we need an “up” sensor
  - Yaw error - we need a “compass”



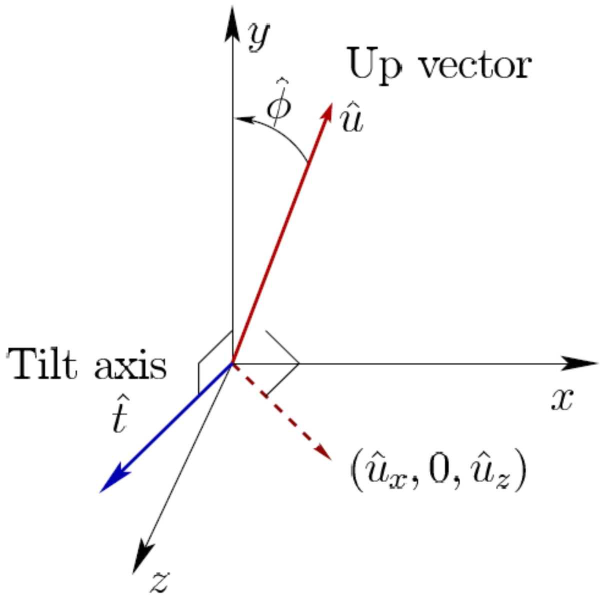
Correcting Tilt Error

- We need a sensor that tells us which way "up" is.
- In practice, the accelerometer is used to measure the "up" direction.
- We can use an accelerometer.
- An accelerometer measures the vector sum of gravity and true linear acceleration



Correcting Tilt Error

- Suppose $\hat{u}$ is the up estimate after applying our orientation estimate $\hat{q}$
- Then because of drift error, it might not be aligned with the y axis.
- So, we must rotate $\hat{u}$ to align it with the y axis.
- The tilt error portion of the drift error is the quaternion $q(\hat{t}, \hat{\phi})$



Filtering

- Allows us to apply the correction gradually.
- We use a complementary filter, which mathematically interpolates between the two estimates.
- $\hat{q}_{corrected} = q(\hat{t}, -\alpha\hat{\phi}) \circ \hat{q}$
- α is a gain parameter that must satisfy 0 < α < 1
- We use a small value of α

Correcting Yaw Error

- We use a compass that always points north to correct yaw error.
- We can use a magnetometer, which measures a 3D magnetic field vector.
- We use the same approach used for tilt correction to find the drift error .
- We apply the changes gradually using a complementary filter.

Problems with Magnetometers

- Earth's magnetic field do not all lie in the horizontal plane.
- The projected vector in the horizontal plane does not point north.
- Magnetometer measures the vector sum of all magnetic field sources.