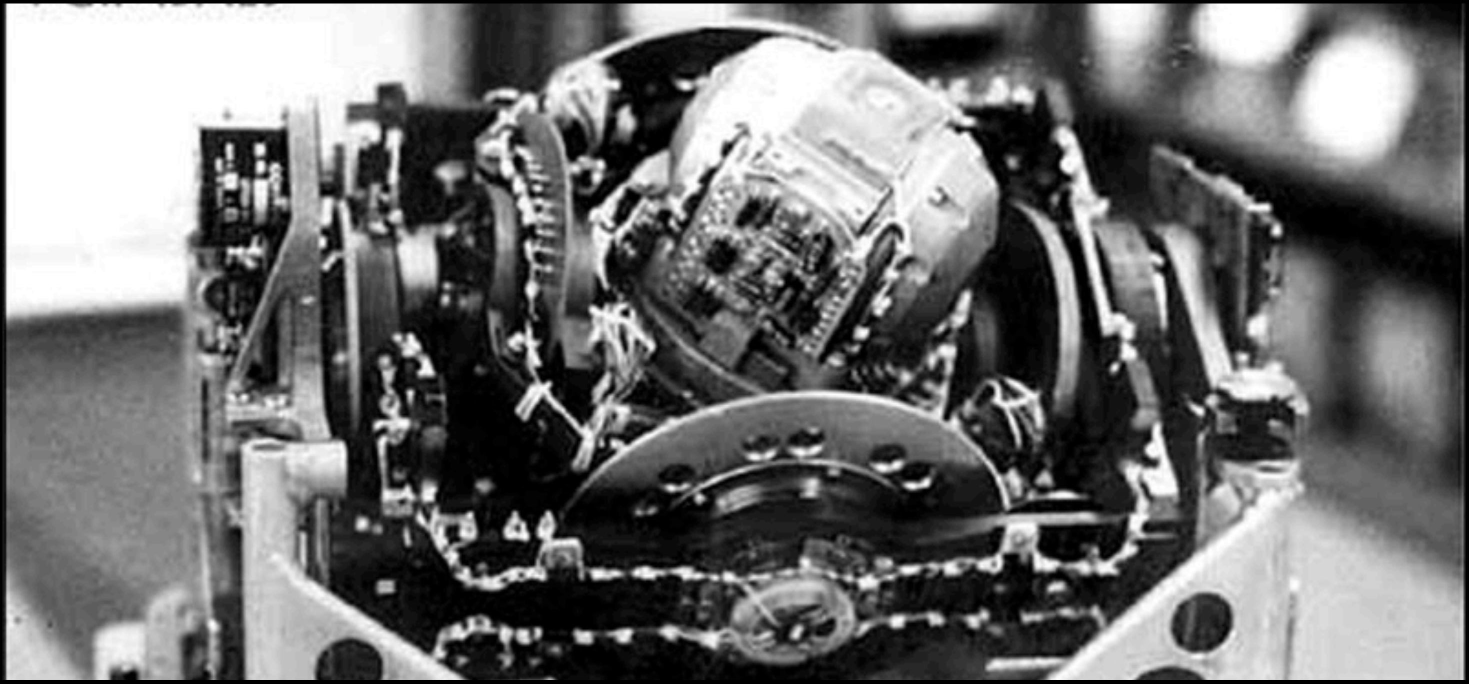
E80
Experimental Engineering

Lecture 9

Inertial Measurement

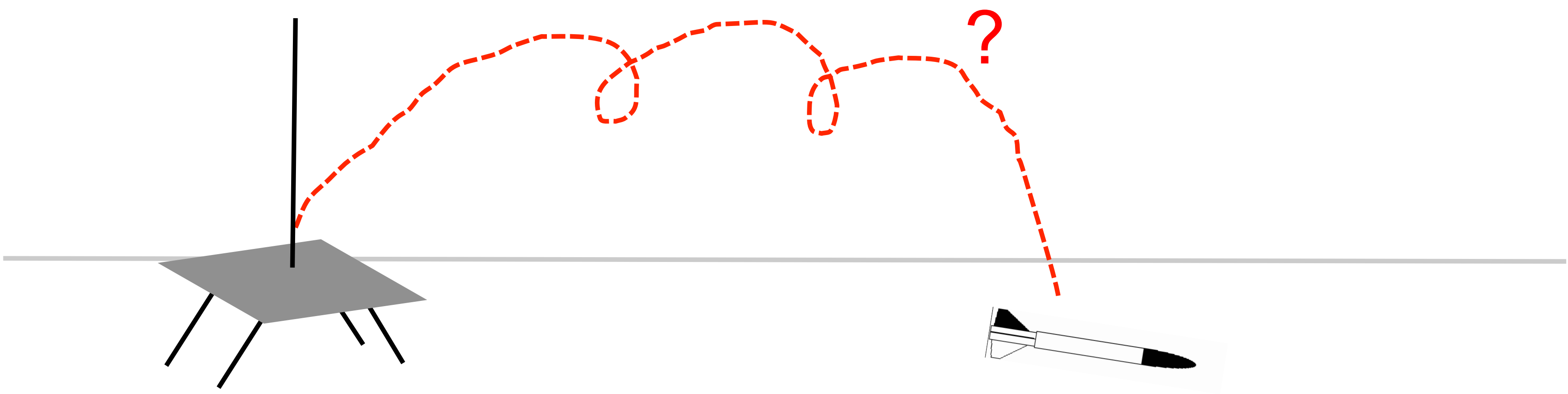


http://www.volker-doormann.org/physics.htm

Feb. 16, 2016

Christopher M. Clark

Where was the rocket?



Outline

- Sensors
- Representations
- State Estimation
- Example Systems
- Bounding with KF

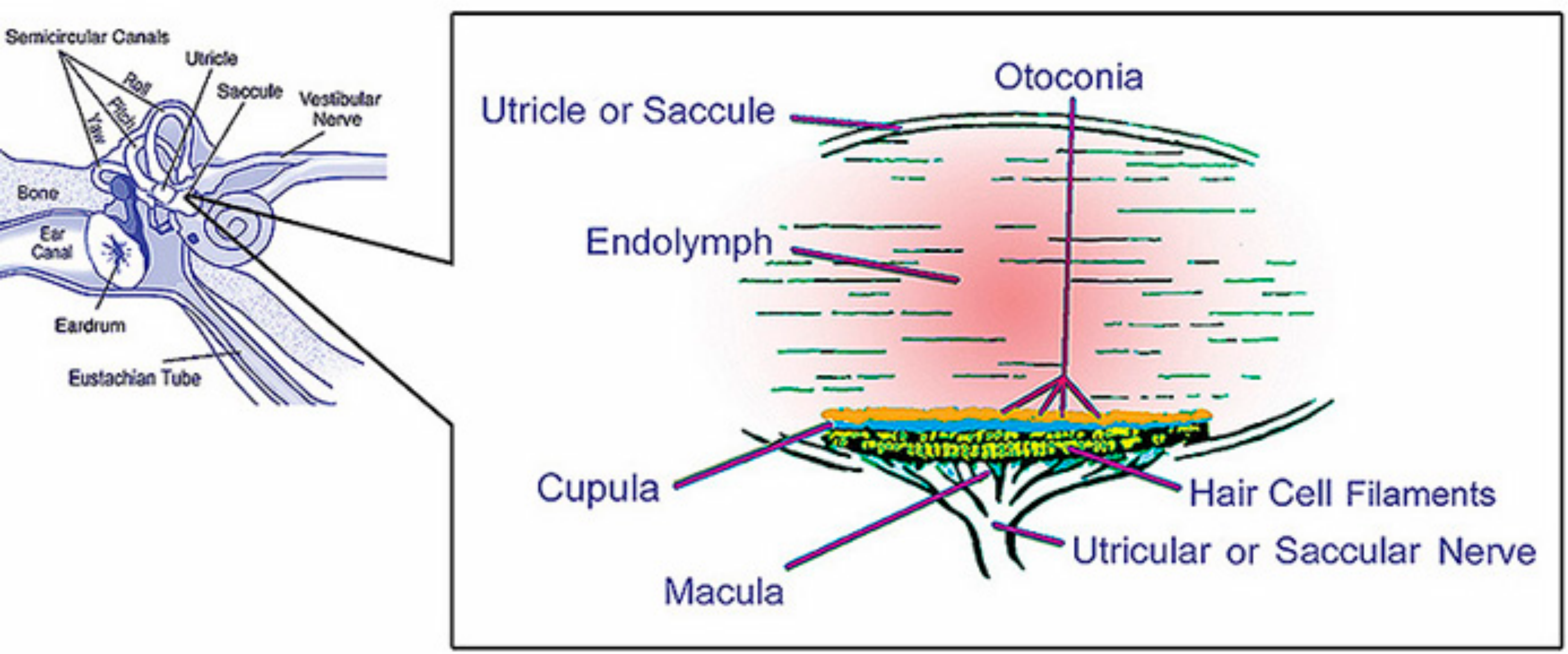
Outline

- Sensors
  - People
  - Accelerometers
  - Gyroscopes
- Representations
- State Estimation
- Example Systems
- Bounding with KF

Sensors

- People

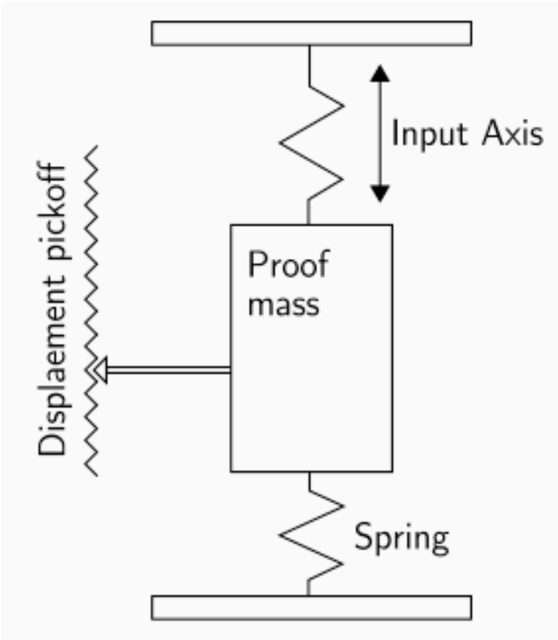
http://en.wikipedia.org/wiki/File:Bigotolith.jpg



Sensors

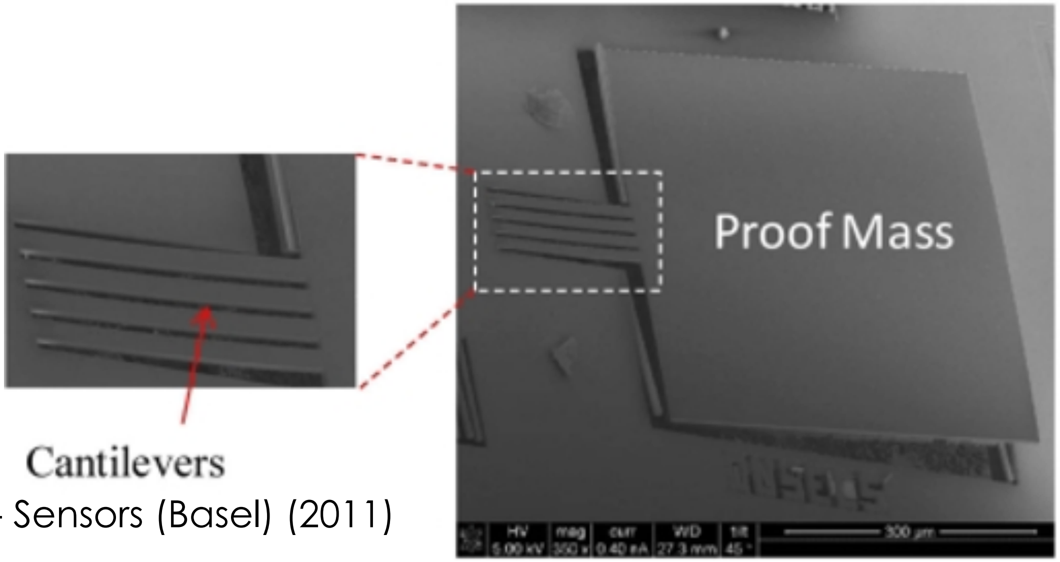
- Accelerometers

$$a = kx/m$$



Sensors

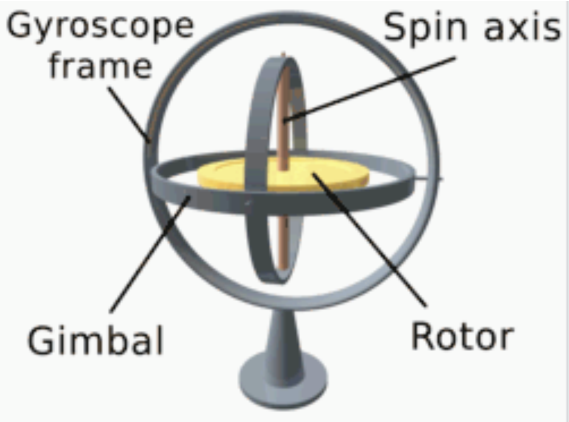
- Accelerometers
  - The accelerometers are typically MEMS based
  - They are small cantilever beams (~100 μm)



Khir MH, Qu P, Qu H - Sensors (Basel) (2011)

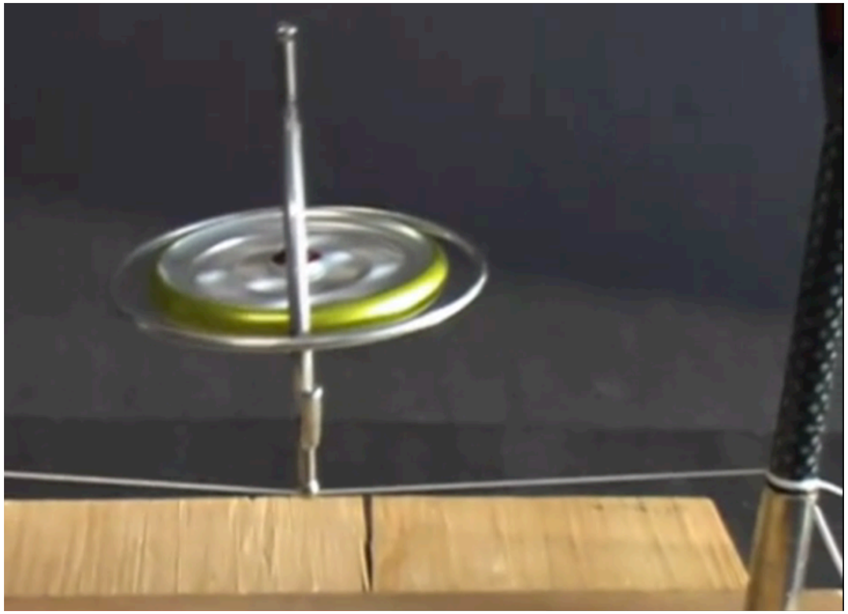
Sensors

- Gyroscopes (original)
  - Mounted on two nested gimbals, the spinning wheel of the gyroscope is free to take any orientation.



  - High spin rate leads to high angular momentum

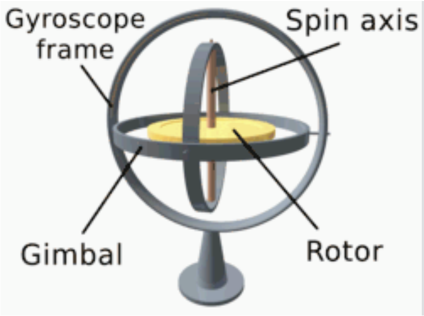
Sensors



http://www.youtube.com/watch?v=cquvA_IpEsA

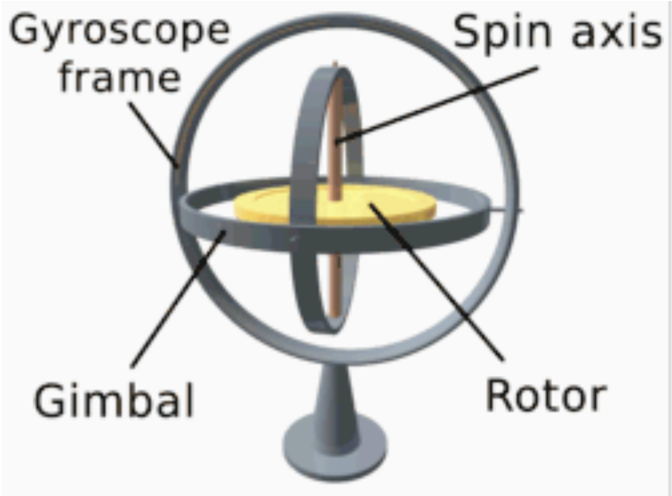
Sensors

- Gyroscopes (original)
  - The gyroscope resists any change of orientation caused by external torques, due to the principle of conservation of angular momentum.
  - The orientation of the gyroscope remains nearly fixed, regardless of any motion of the platform.



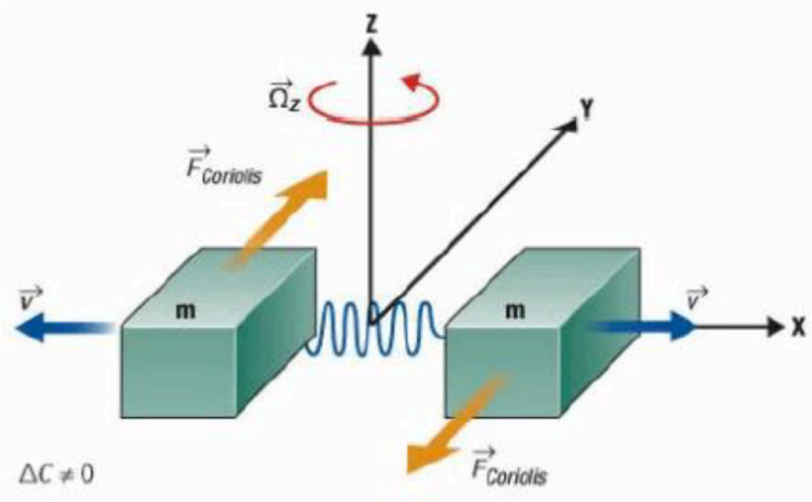
Sensors

- Gyroscopes (original)
  - Gyroscope orientation can be obtained by measuring the angles between adjacent gimbals.



Sensors

- MEMS (Microelectromechanical Systems) Gyroscopes
  - Two masses oscillate back and forth from the center of rotation with velocity v.
  - A rotation will cause a Coriolis force in this coordinate frame.



Sensors

- MEMS Gyroscopes
  - Their deflection y is measured, to establish a force

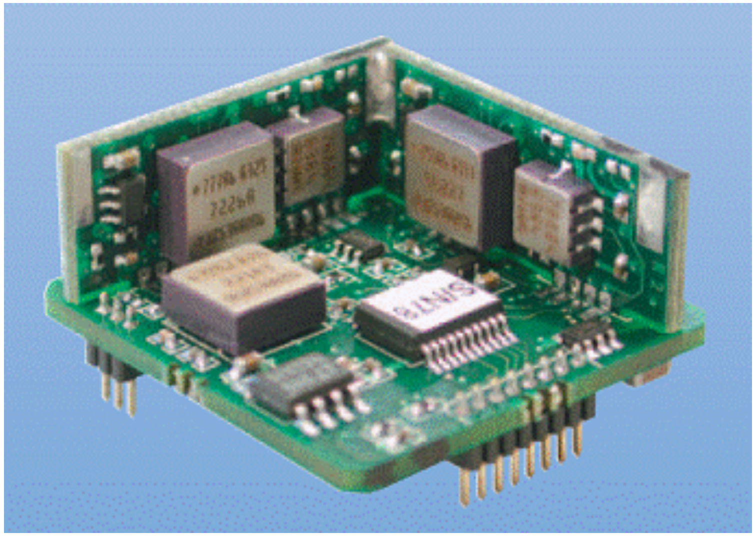
$$F_{Coriolis} = k\, y$$

  - The acceleration is obtained since the mass is known

$$-2m\, |\boldsymbol{\Omega} \times \mathbf{v}| = F_{Coriolis}$$

Sensors

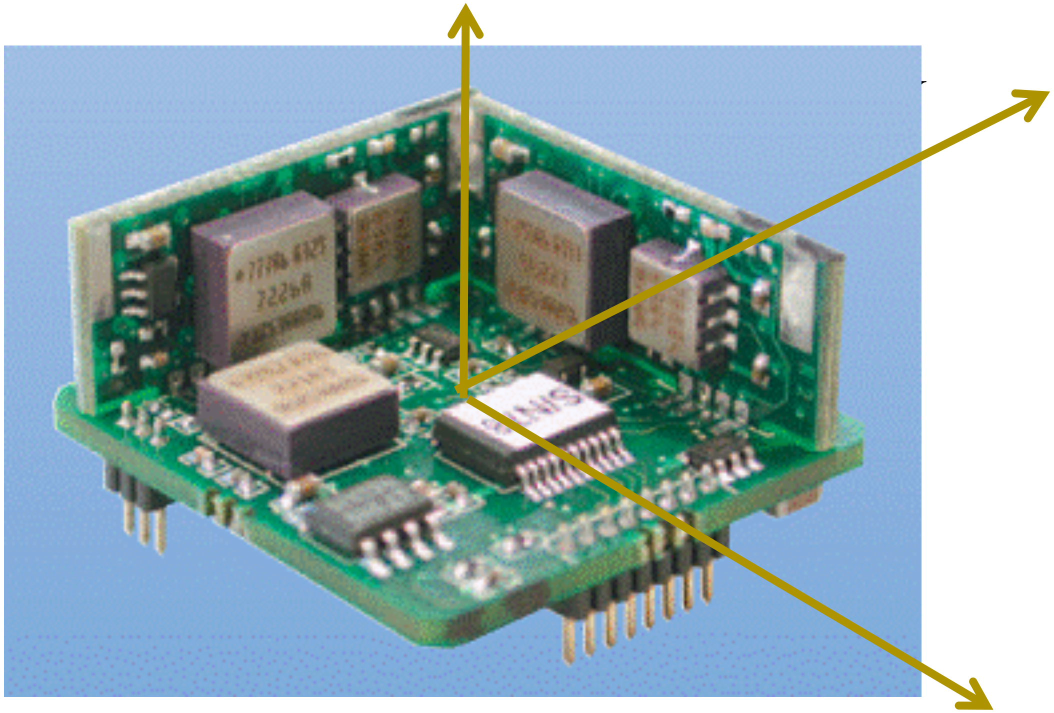
- Inertial Measurement Units
  - 3 Accelerometers
  - 3 Gyroscopes
  - 3 Magnetometers(?)



www.barnardmicrosystems.com

Sensors

- Inertial Measurement Units



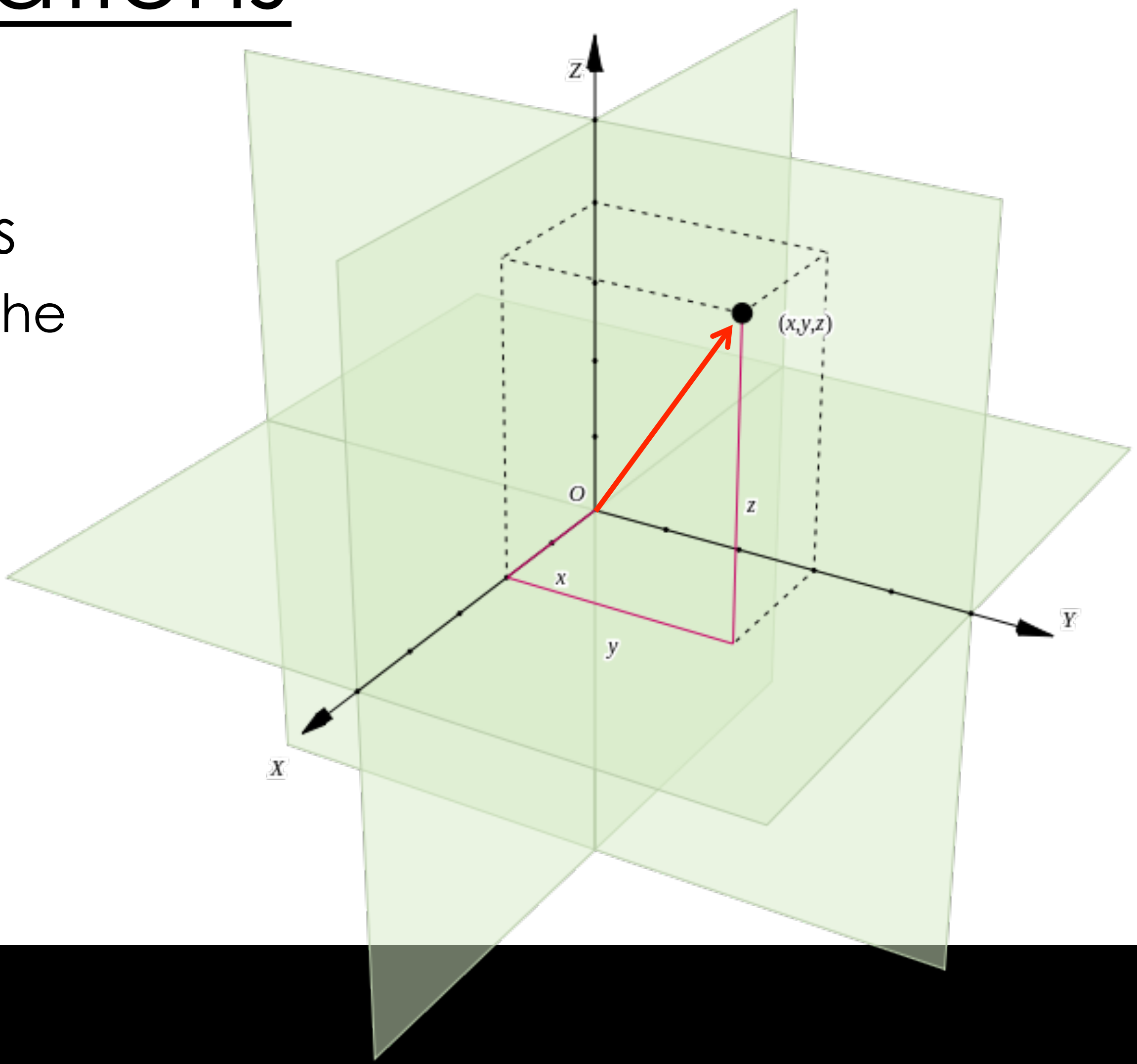
Outline

- Sensors
- Representations
  - Cartesian Coordinate Frames
  - Transformations
- State Estimation
- Example Systems
- Bounding with KF

Representations

- Cartesian Coordinate Frames
  - We can represent the 3D position of a vehicle with the vector

$$\mathbf{r} = [\, x \; y \; z \,]^T$$



Representations

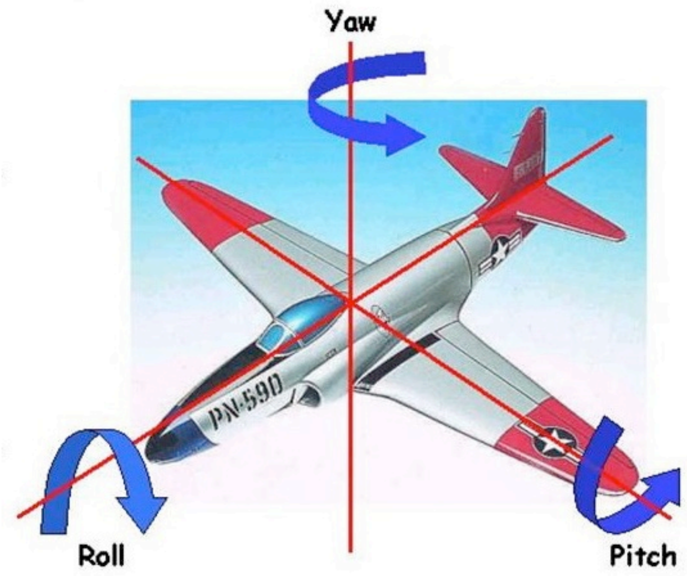
- Euler Angles
  - We can represent the 3D orientation of a vehicle with the vector

$$\phi = [\ \alpha\ \beta\ \gamma\ ]^T$$

Yaw - α

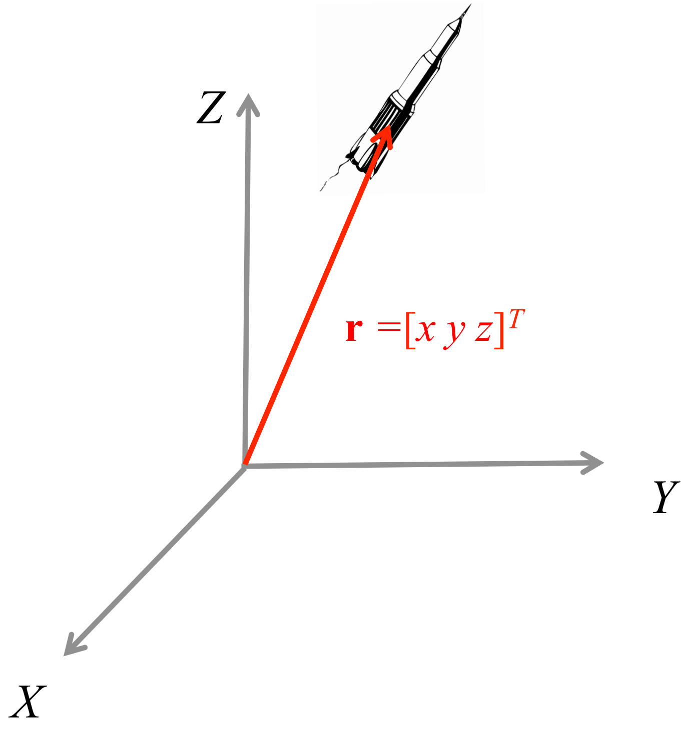
Pitch - β

Roll – γ



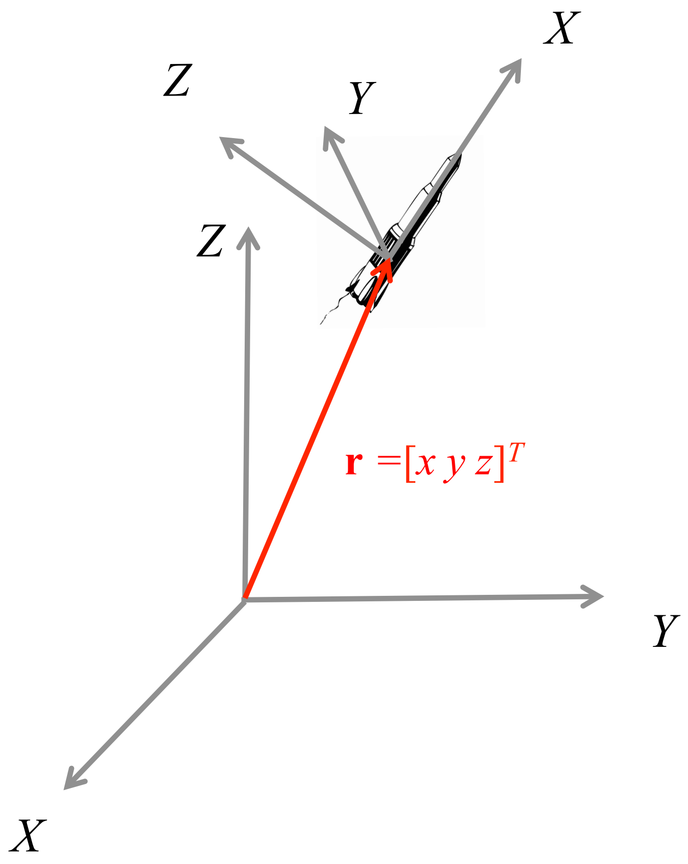
Representations

- Where do we place the origin?
  - We can fix the origin at a specific location on earth, e.g. a rocket's launch pad.
  - This is called the **global** or inertial coordinate frame



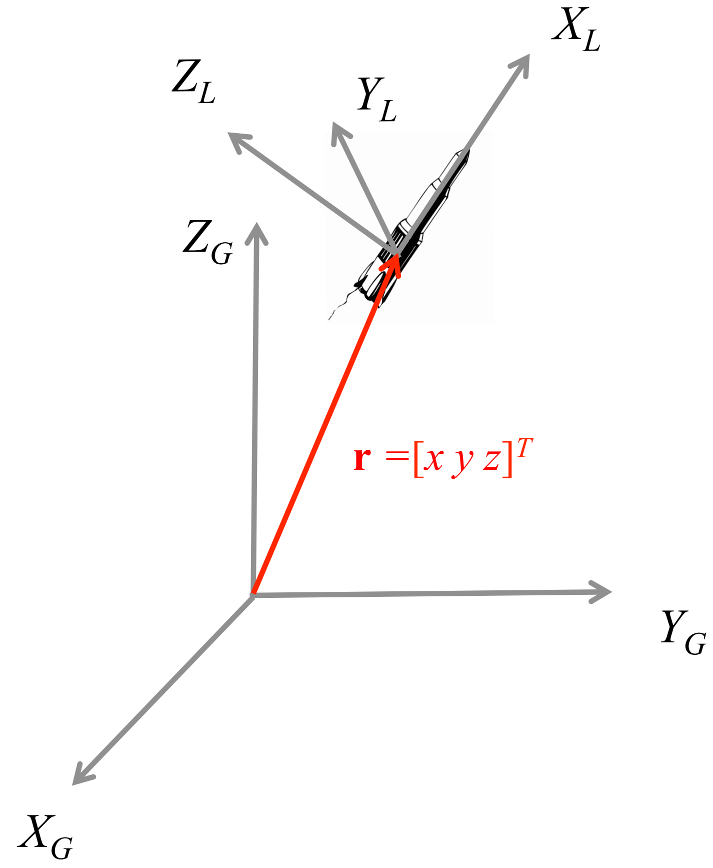
Representations

- Where do we place the origin?
  - We can ALSO fix the origin on a vehicle.
  - This is called the **local** coordinate frame



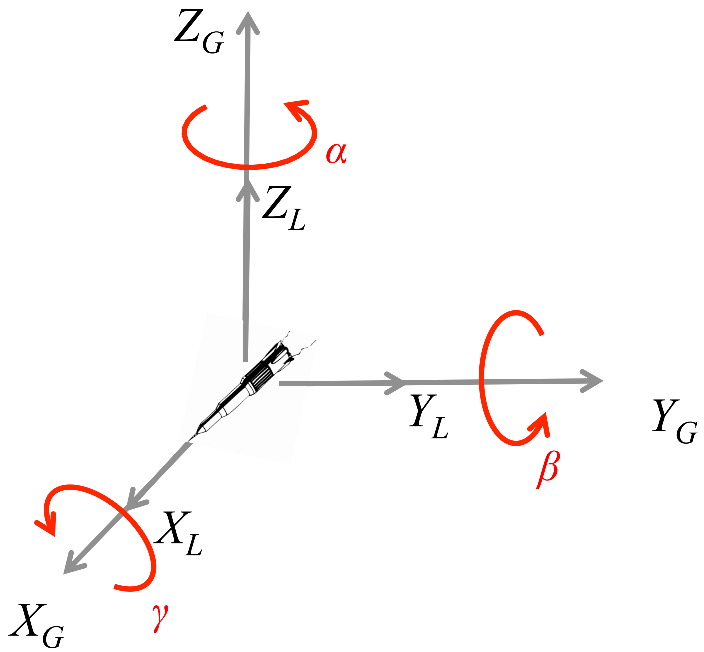
Representations

- Where do we place the origin?
  - We must differentiate between these two frames.
  - What is the real difference between these two frames?
  - A Transformation consisting of a **rotation** and **translation**



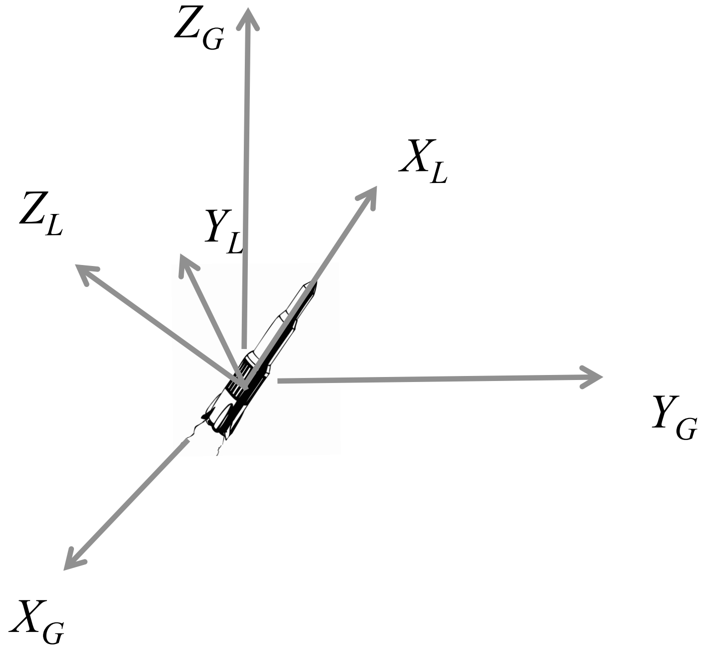
Representations

- Transformations
  - The **rotation** can be about 3 axes (i.e. the roll, pitch, yaw)



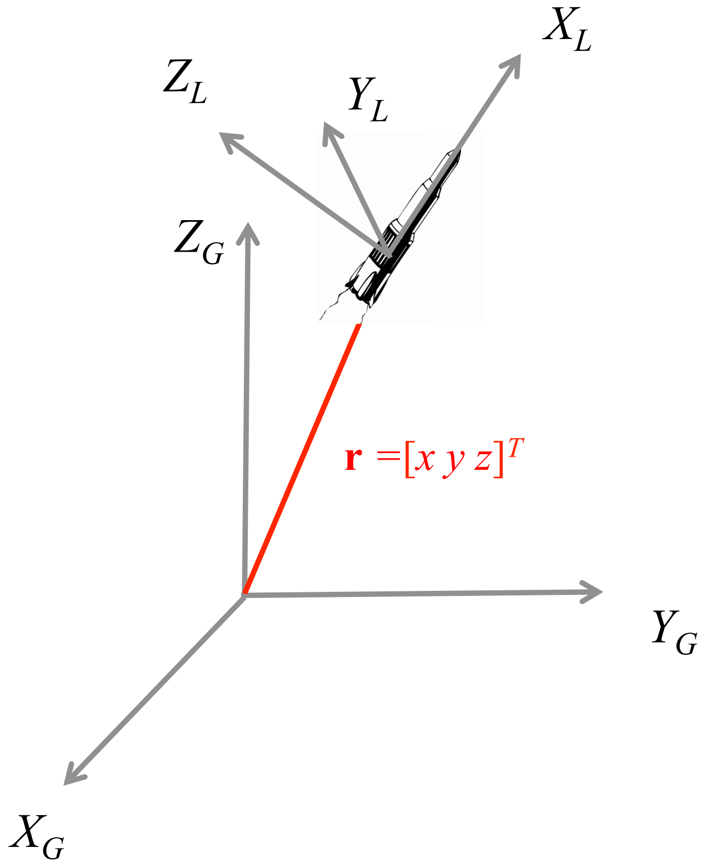
Representations

- Transformations
  - The **rotation** can be about 3 axes (i.e. the roll, pitch, yaw)



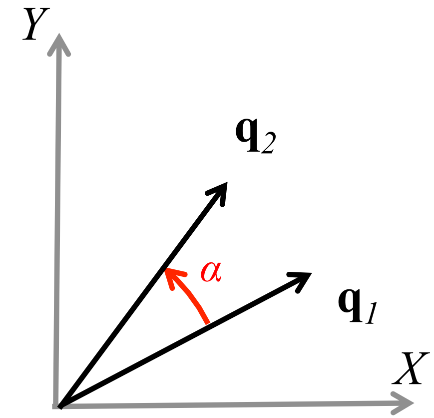
Representations

- Transformations
  - The **translation** can be in three directions



Representations

- Rotations
  - In 2D, it is easy to determine the effects of rotation on a vector

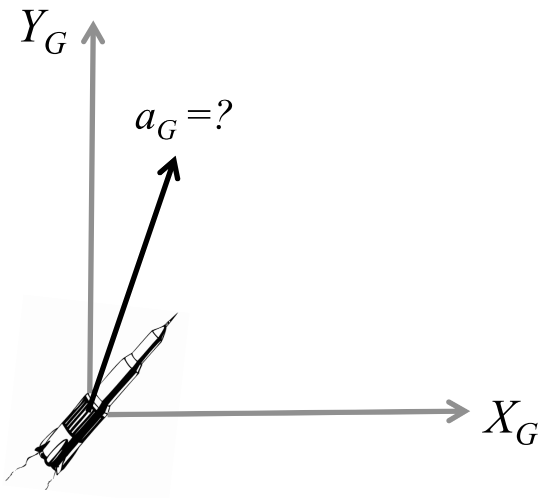


$$\mathbf{q}_2 = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \mathbf{q}_1$$

$$= \mathbf{R}(\alpha)\, \mathbf{q}_1$$

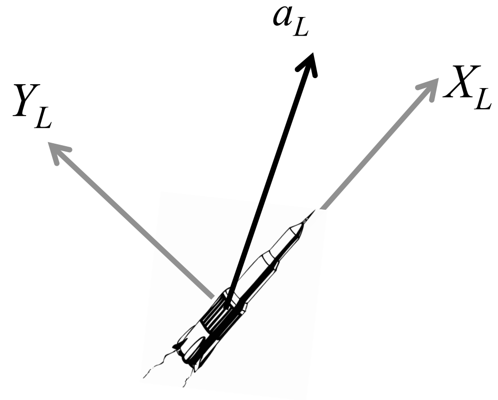
Representations

- Rotations
  - We want to determine the rocket acceleration with respect to the global frame



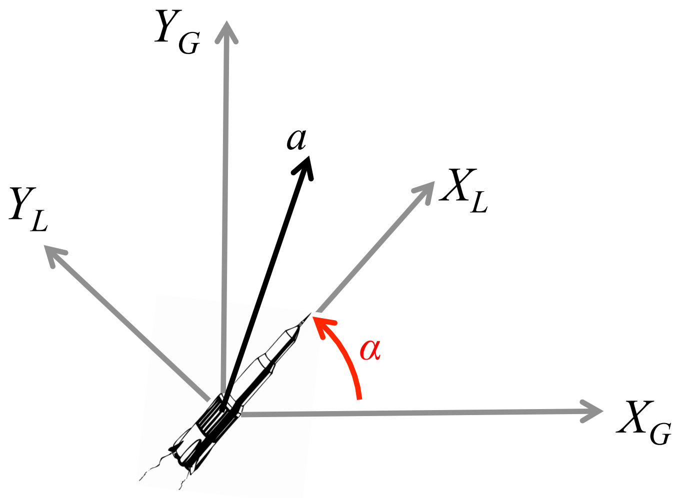
Representations

- Rotations
  - We can measure rocket acceleration in the local frame.



Representations

- Rotations
  - To get the acceleration vector in the Global frame, we rotate the acceleration vector in the local frame by α – the rotation angle between frames

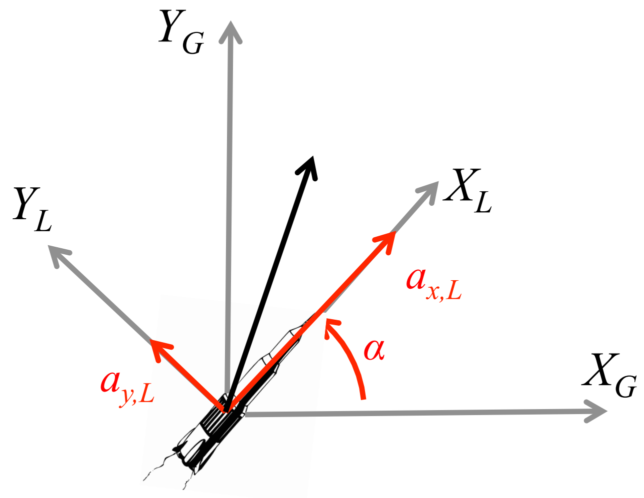


Representations

- Rotations
  - Here, we want to determine how acceleration in one frame relates to acceleration in another.

$$\begin{pmatrix} a_{x,G} \\ a_{y,G} \end{pmatrix} = \begin{pmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{pmatrix} \begin{pmatrix} a_{x,L} \\ a_{y,L} \end{pmatrix}$$

$$\mathbf{a}_G = \mathbf{R}(\alpha)\ \mathbf{a}_L$$

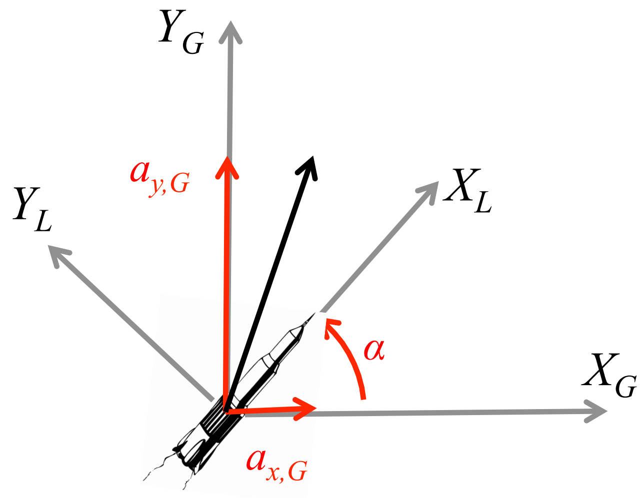


Representations

- Rotations
  - Here, we want to determine how acceleration in one frame relates to acceleration in another.

$$\begin{pmatrix} a_{x,G} \\ a_{y,G} \end{pmatrix} = \begin{pmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{pmatrix} \begin{pmatrix} a_{x,L} \\ a_{y,L} \end{pmatrix}$$

$$\mathbf{a}_G = \mathbf{R}(\alpha)\ \mathbf{a}_L$$

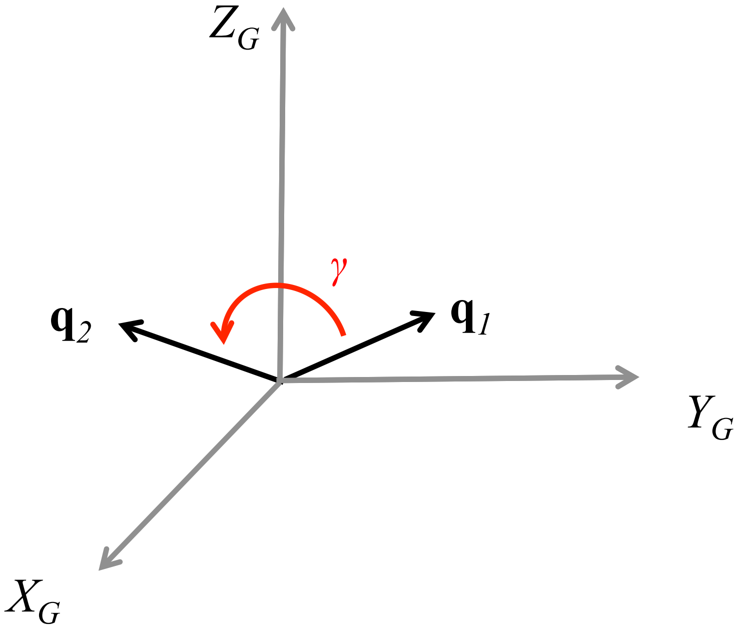


Representations

- Rotations
  - In 3D, we can use similar rotation matrices

$$\mathbf{q}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{pmatrix} \mathbf{q}_1$$

$$= \mathbf{R}_\mathbf{x}(\gamma)\ \mathbf{q}_1$$



Representations

$$\mathbf{R_x}(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{pmatrix}$$

$$\mathbf{R_y}(\beta) = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix}$$

$$\mathbf{R_z}(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Representations

- Rotations
  - For 3 rotations, we can write a general Rotation Matrix

$$\mathbf{R}(\alpha, \beta, \gamma) = \mathbf{R_z}(\alpha)\, \mathbf{R_y}(\beta)\, \mathbf{R_x}(\gamma)$$

  - Hence, we can rotate any vector with the general Rotation Matrix

$$\mathbf{q}_2 = \mathbf{R}(\alpha, \beta, \gamma)\, \mathbf{q}_1$$

Representations

- Rotations
  - Hence, we can rotate any acceleration vector in a local frame through roll, pitch, yaw angles to get the corresponding acceleration vector in the global frame

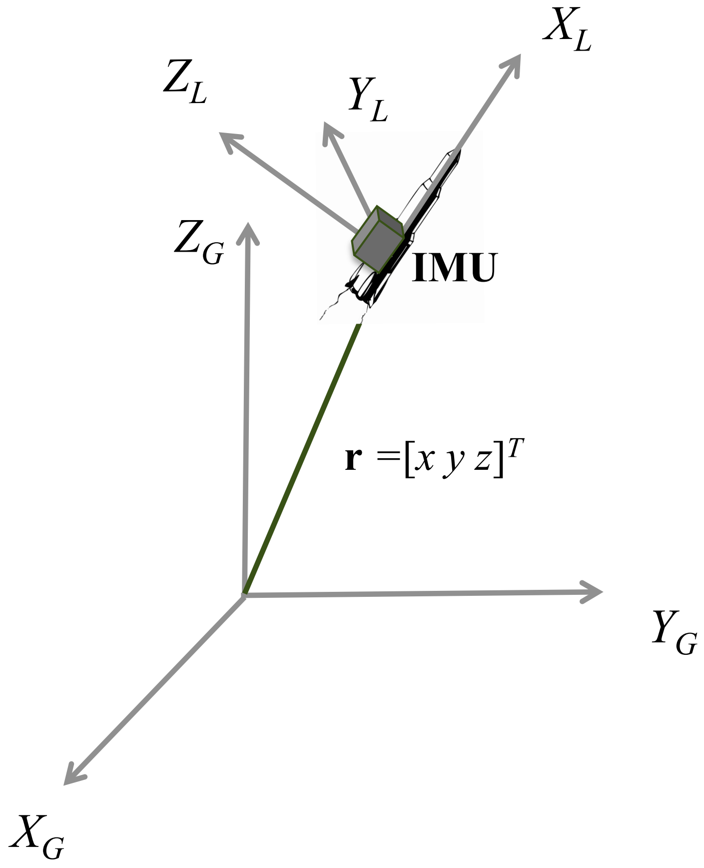
$$\mathbf{a}_G = \mathbf{R}(\alpha, \beta, \gamma)\ \mathbf{a}_L$$

Outline

- Sensors
- Representations
- State Estimation
  - Updating $\mathbf{R}(t)$
  - Updating $\mathbf{r}(t)$
  - Pseudo Code
- Example Systems
- Bounding Errors with KF

State Estimation

- Strapdown Inertial Navigation
  - Our IMU is fixed to the **local** frame
  - We care about the state of the vehicle in the **global** frame



Updating $\mathbf{R}(t)$

Given: $\boldsymbol{\omega}_{\boldsymbol{L}}(t) = [\ \omega_{x,L}(t)\ \omega_{y,L}(t)\ \omega_{z,L}(t)\ ]^T$

Find: $\mathbf{R}(t)$

Updating $\mathbf{R}(t)$

- Lets define a rotational velocity matrix based on our gyroscope measurements $\boldsymbol{\omega}_L = [\omega_{x,L}(t)\ \omega_{y,L}(t)\ \omega_{z,L}(t)]^T$

$$\Omega(t) = \begin{pmatrix} 0 & -\omega_z(t) & \omega_y(t) \\ \omega_z(t) & 0 & -\omega_x(t) \\ -\omega_y(t) & \omega_x(t) & 0 \end{pmatrix}$$

Updating $\mathbf{R}(t)$

- It can be shown that the vehicle rotating with velocity $\Omega(t)$ for δt **seconds** will (approximately) yield the resulting rotation Matrix $\mathbf{R}(t+\delta t)$:

$$\mathbf{R}(t+\delta t) = \mathbf{R}(t)\,[\,\mathbf{I} + \Omega(t)\delta t\,]$$

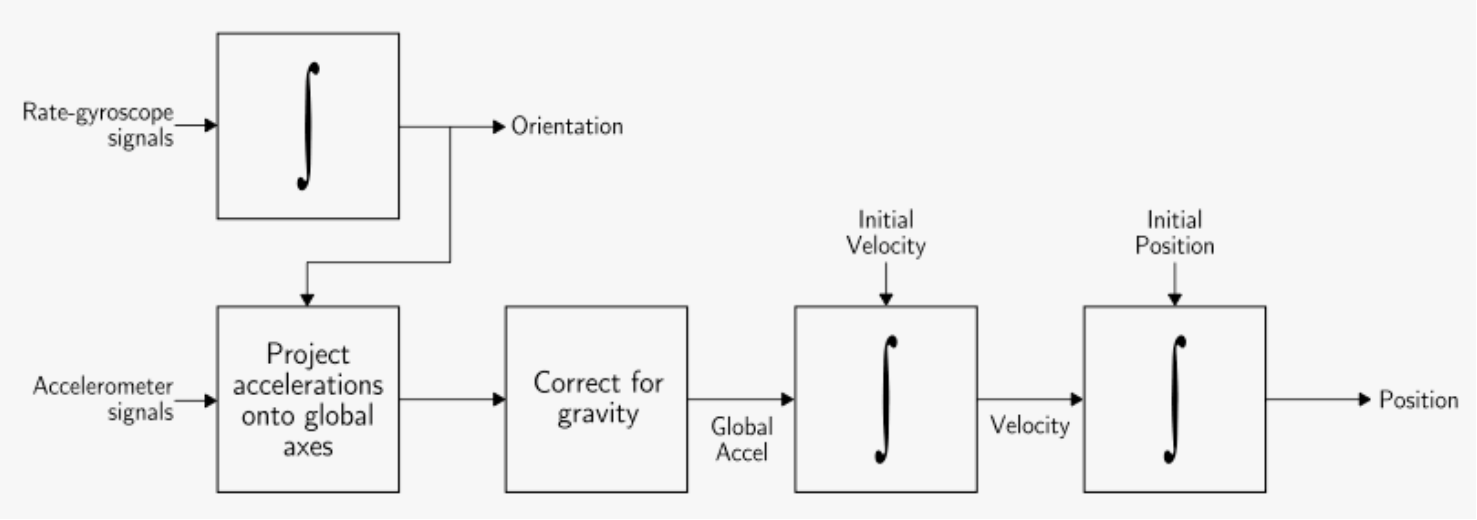
Updating $\mathbf{r}(t)$

Given: $\mathbf{a}_L = [\, a_{x,L}\ a_{y,L}\ a_{z,L}\,]^T$

$\mathbf{R}(t)$

Find: $\mathbf{r}_G = [\, x_G\ \ y_G\ \ z_G\,]^T$

Updating r(t)


Rate-gyroscope signals
Orientation
Initial Velocity
Initial Position
Accelerometer signals
Project accelerations onto global axes
Correct for gravity
Global Accel
Velocity
Position

Updating $\mathbf{r}(t)$

- First, convert to global reference frame

$$\mathbf{a}_G(t) = \mathbf{R}(t)\, \mathbf{a}_L(t)$$

- Second, remove gravity term

$$\mathbf{a}_G(t) = [\; a_{x,G}(t) \quad a_{y,G}(t) \quad a_{z,G}(t) - g \;]^T$$

Updating $\mathbf{r}(t)$

- Third, integrate to obtain velocity

$$\mathbf{v}_G(t) = \mathbf{v}_G(0) + \int_0^t \mathbf{a}_G(\tau)\, d\tau$$

- Fourth, integrate to obtain position

$$\mathbf{r}_G(t) = \mathbf{r}_G(0) + \int_0^t \mathbf{v}_G(\tau)\, d\tau$$

Updating $\mathbf{r}(t)$

- Third, integrate to obtain (approximate) velocity

$$\mathbf{v}_G(t+\delta t) = \mathbf{v}_G(t) + \mathbf{a}_G(t+\delta t)\,\delta t$$

- Fourth, integrate to obtain (approximate) position

$$\mathbf{r}_G(t+\delta t) = \mathbf{r}_G(t) + \mathbf{v}_G(t+\delta t)\,\delta t$$

State Estimation

```
for t = 0 to maxTime
{
        ω(t) = …
        a_L(t) = …

        R(t) = …
        a_G(t) = …
        a_G(t)  = … // subtract gravity
        v_G(t)  = …
        r_G(t)  = …
        …
}
```

State Estimation

```
for t = 0 to maxTime
{
        ω(t) = …
        a_L(t) = …

        R(t) = …
        a_G(t) = …
        a_G(t)  = … // subtract gravity
        v_G(t)  = …
        r_G(t)  = …
        …
}
```

State Estimation

- What about Errors?
  - We could use Error Propagation

$$\mathbf{r}_{G}(t+\delta t) = \mathbf{r}_{G}(t) + \mathbf{v}_{G}(t+\delta t)\,\delta t$$

  - So

$$\mathbf{e}_{\mathbf{r}G}(t+\delta t)^2 = \left(\frac{d\mathbf{r}_G(t+\delta t)}{d\mathbf{r}_G(t)}\right)^2 \mathbf{e}_{\mathbf{r}G}(t)^2 + \left(\frac{d\mathbf{r}_G(t+\delta t)}{d\mathbf{v}_G(t+\delta t)}\right)^2 \mathbf{e}_{\mathbf{v}G}(t+\delta t)^2$$

State Estimation

```
for t = 0 to maxTime
{
        ω(t) = …
        a_L(t) = …

        R(t) = …
        a_G(t) = …
        a_G(t)  = … // subtract gravity
        v_G(t)  = …
        r_G(t)  = …
        e_rG(t)  = …
}
```

State Estimation

- What is bad about the first term?

$$\mathbf{e}_{\mathbf{r}G}(t+\delta t)^2 = \left(\frac{d\mathbf{r}_G(t+\delta t)}{d\mathbf{r}_G(t)}\right)^2 \mathbf{e}_{\mathbf{r}G}(t)^2 + \left(\frac{d\mathbf{r}_G(t+\delta t)}{d\mathbf{v}_G(t+\delta t)}\right)^2 \mathbf{e}_{\mathbf{v}G}(t+\delta t)^2$$

State Estimation

- Errors accumulate!
  - Each error is a function of the error from the previous time step

$$\mathbf{e}_{\mathbf{r}G}(t+\delta t)^2 = \left(\frac{d\mathbf{r}_G(t+\delta t)}{d\mathbf{r}_G(t)}\right)^2 \mathbf{e}_{\mathbf{r}G}(t)^2 + \left(\frac{d\mathbf{r}_G(t+\delta t)}{d\mathbf{v}_G(t+\delta t)}\right)^2 \mathbf{e}_{\mathbf{v}G}(t+\delta t)^2$$

  - For example

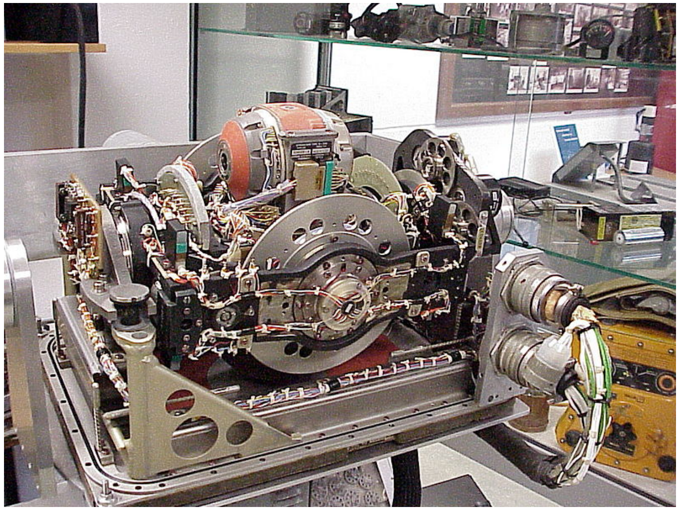
$$\mathbf{e}_{\mathbf{r}G}(t) = f(\mathbf{e}_{\mathbf{r}G}(t-\delta t), \mathbf{e}_{\mathbf{v}G}(t))$$
$$\mathbf{e}_{\mathbf{r}G}(t-\delta t) = f(\mathbf{e}_{\mathbf{r}G}(t-2\delta t), \mathbf{e}_{\mathbf{v}G}(t-\delta t))$$
$$\ldots$$

Outline

- Sensors
- Representations
- State Estimation
- Example Systems
  - LN-3
  - The Jaguar
- Bounding Errors with KF

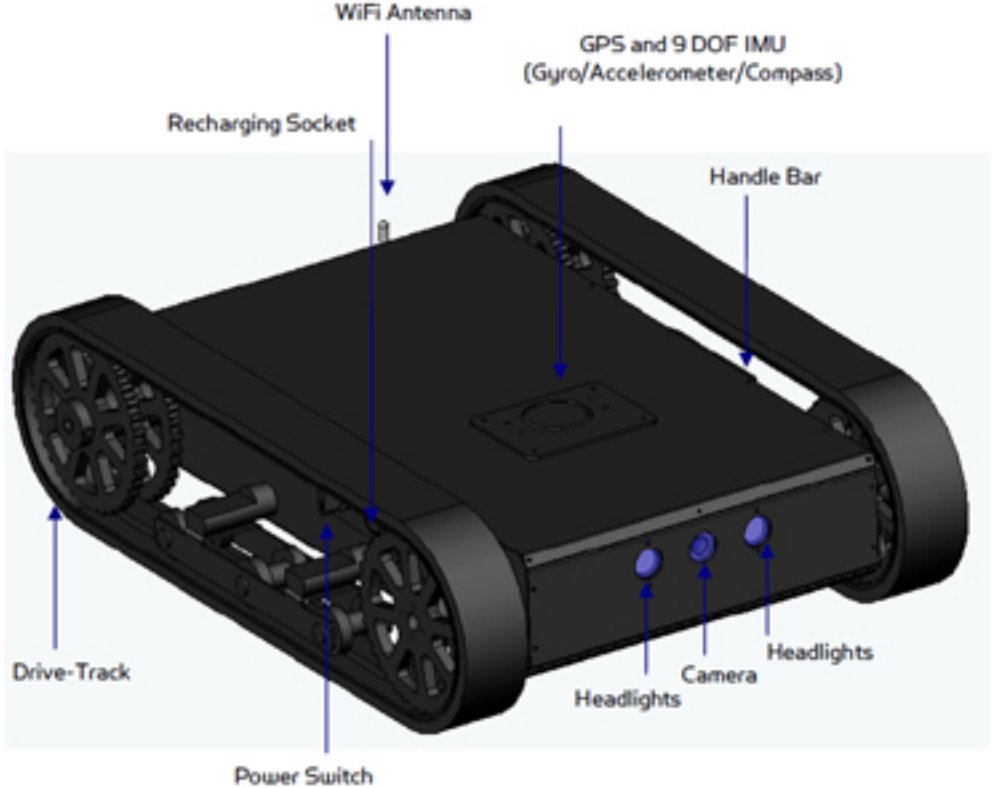
Example Systems

- LN-3 Inertial Navigation System
  - Developed in 1960’s
  - Used gyros to help steady the platform
  - Accelerometers on the platform were used to obtain accelerations in global coordinate frame
  - Accelerations (double) integrated to obtain position



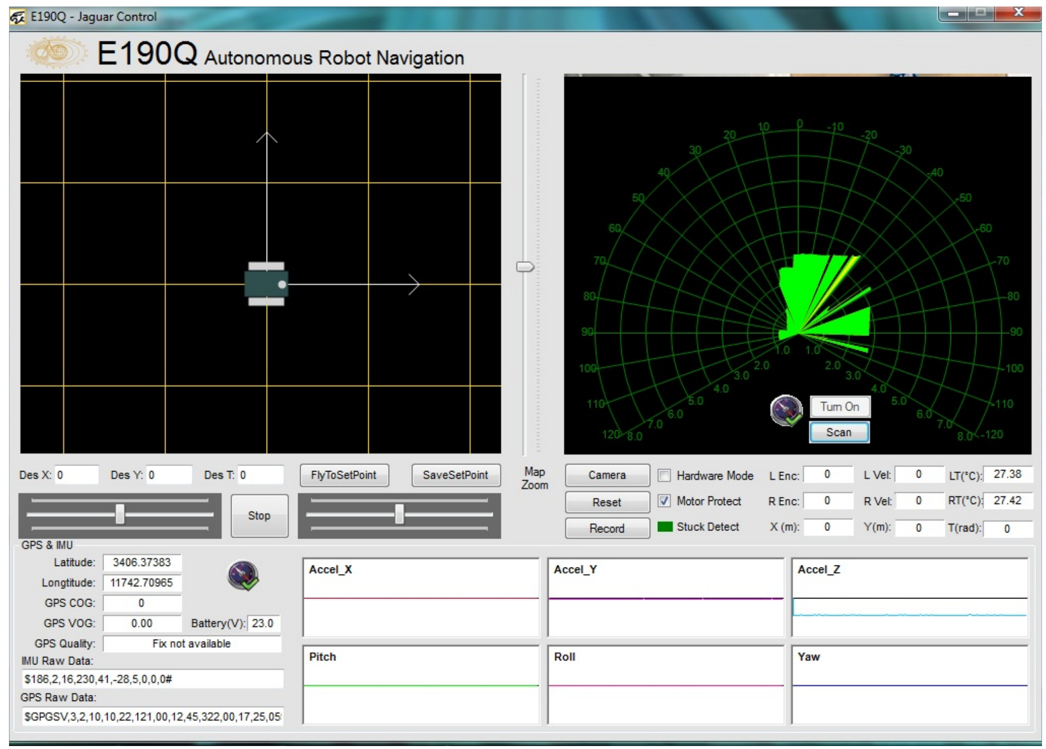
Example Systems

- The Jaguar Lite
  - Equipped with an **IMU**, camera, laser scanner, encoders, GPS



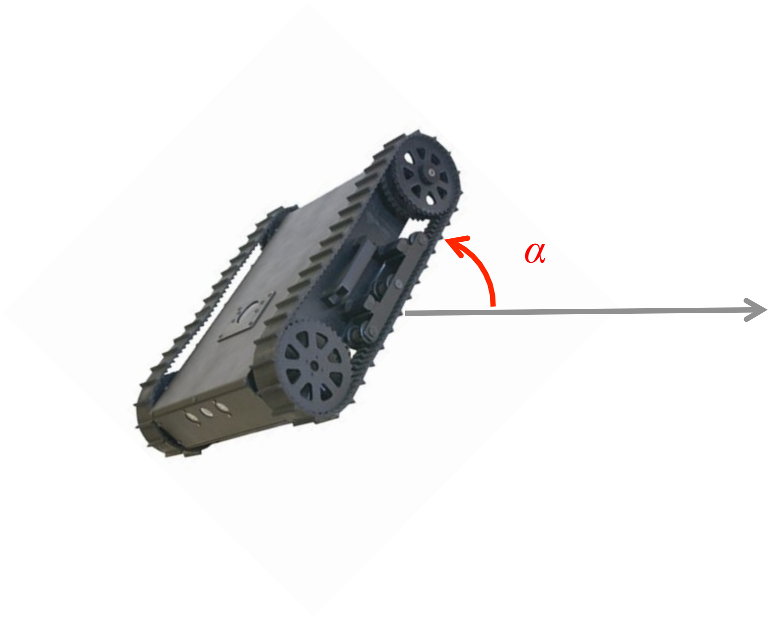
Example Systems

- The Jaguar Lite
  - GUI provides acceleration measurements



Example Systems

- Question:
  - Can we use the accelerometers alone to measure orientation?

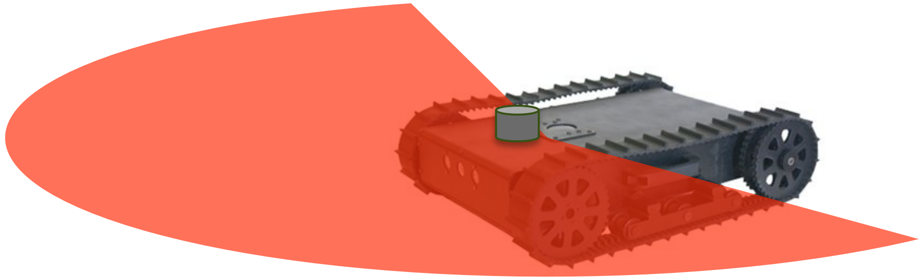


Outline

- Sensors
- Representations
- State Estimation
- Example Systems
- Bounding Errors with KF
  - Exteroceptive Sensing
  - Fusing measurements

Bounding our Errors

- Exteroceptive sensors
  - Drift in inertial navigation is a problem
  - We often use exteroceptive sensors – which measure outside the robot – to bound our errors
  - Examples include vision systems, GPS, range finders



Bounding our Errors

- Exteroceptive sensors
  - We can fuse measurements, e.g. integrated accelerometer measurements and range measurements, by averaging.
  - For example, consider the 1D position estimate of the jaguar.

$$x = 0.5\ (\ x_{IMU} + (x_{wall} - x_{laser})\ )$$

x_{IMU} is the double integrated IMU measurement
x_{wall} is the distance from the origin to the wall
x_{laser} is the central range measurement

Bounding our Errors

- Exteroceptive sensors
  - Lets weight the average, where weights reflect measurement confidence

$$x = \frac{w_{IMU} x_{IMU} + w_{laser}(x_{wall} - x_{laser})}{w_{IMU} + w_{laser}}$$

Bounding our Errors

- Exteroceptive sensors
  - This leads us to a 1D Kalman Filter

$$x_t = x_{IMU,t} + K_t\,[(x_{wall} - x_{laser,t}) - x_{IMU,t}]$$

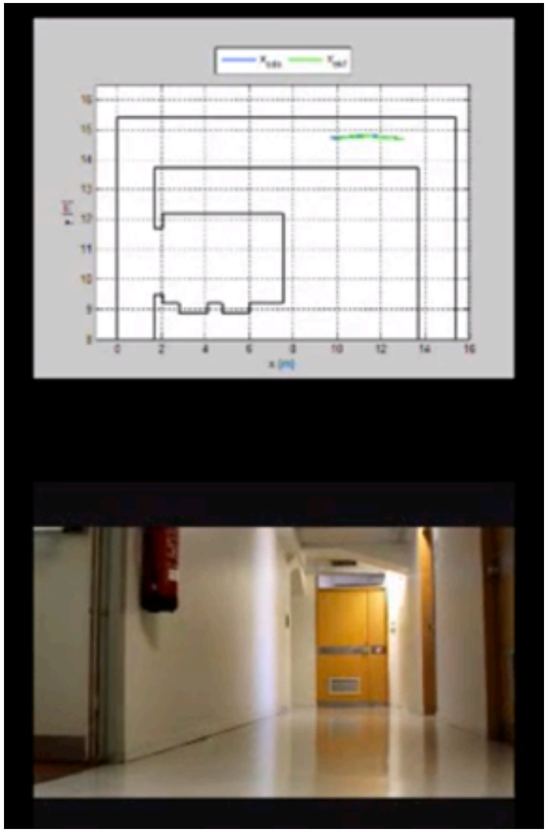
$$K_t = \frac{\sigma^2_{IMU}}{\sigma^2_{IMU} + \sigma^2_{laser}}$$

$$\sigma^2_x = (1 - K_t)\,\sigma^2_{IMU}$$

Bounding our Errors

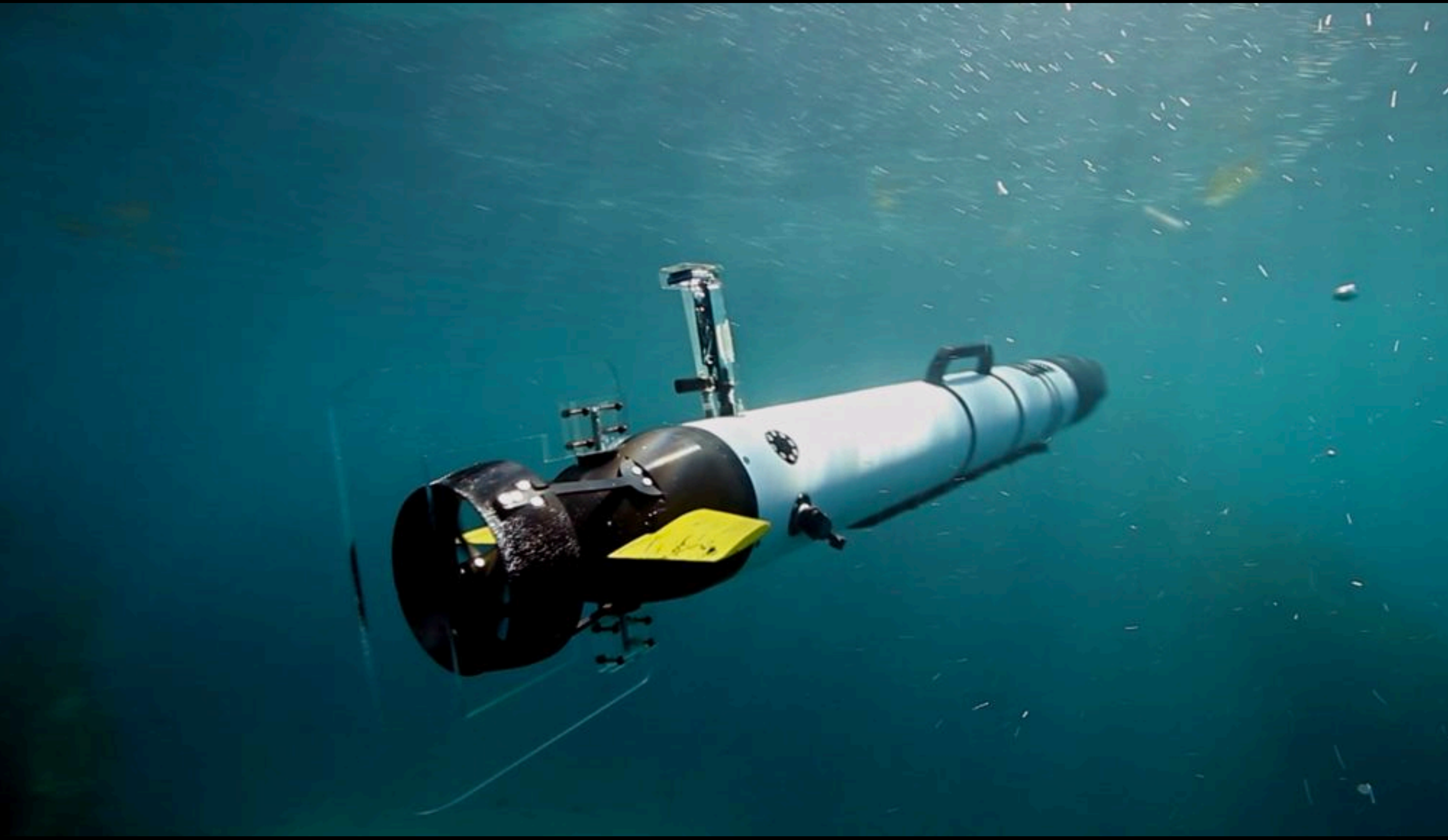
- IMU
  - Higher sampling rate
  - Small errors between time steps (maybe centimeters)
  - Uncertainty increases
  - Large build up over time (unbounded)
- GPS
  - Lower sampling rate
  - Larger errors(maybe meters)
  - Uncertainty decreases
  - No build up (bounded)

Bounding our Errors



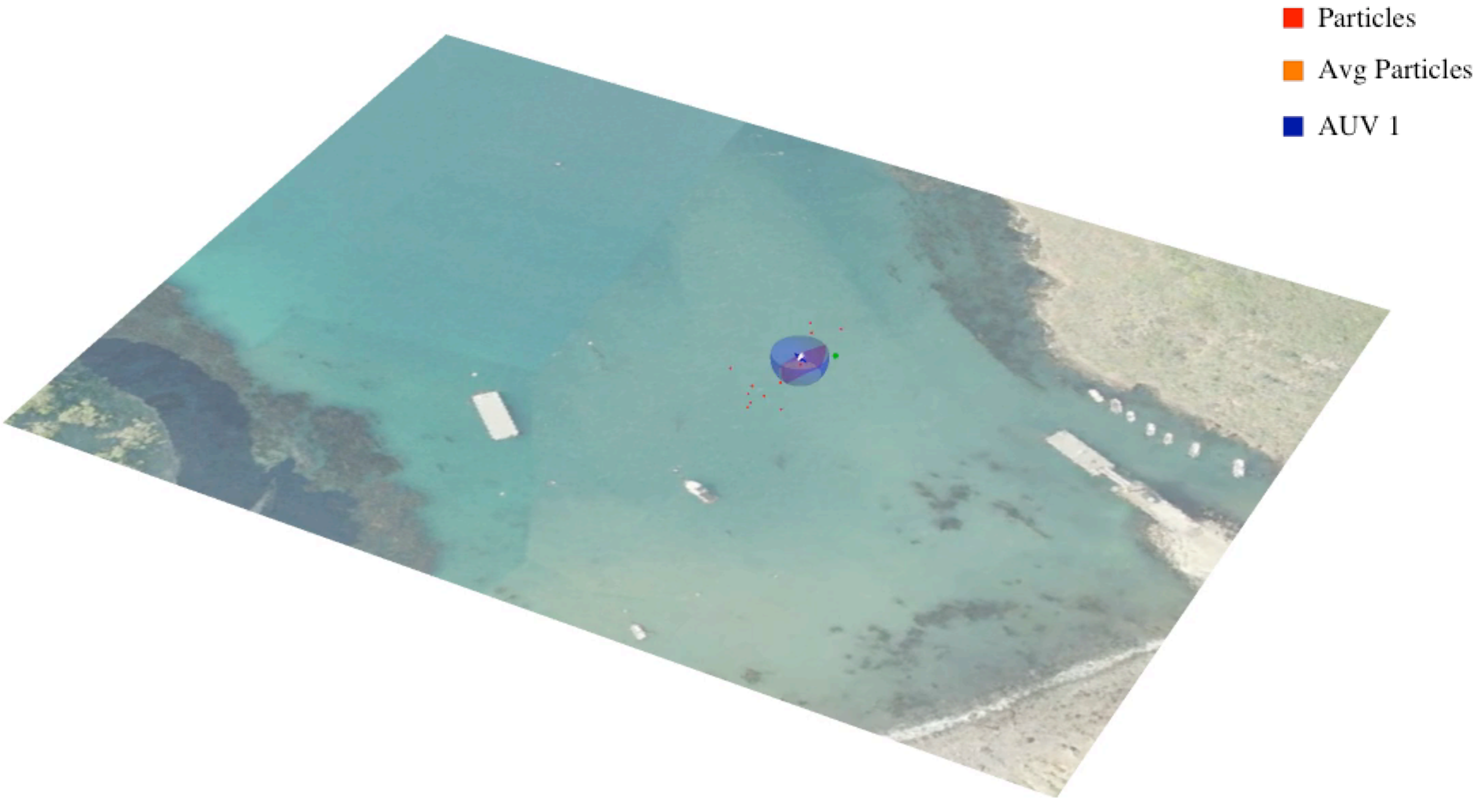
http://www.youtube.com/watch?v=AXGXfD1GMY4

E80
Experimental Engineering

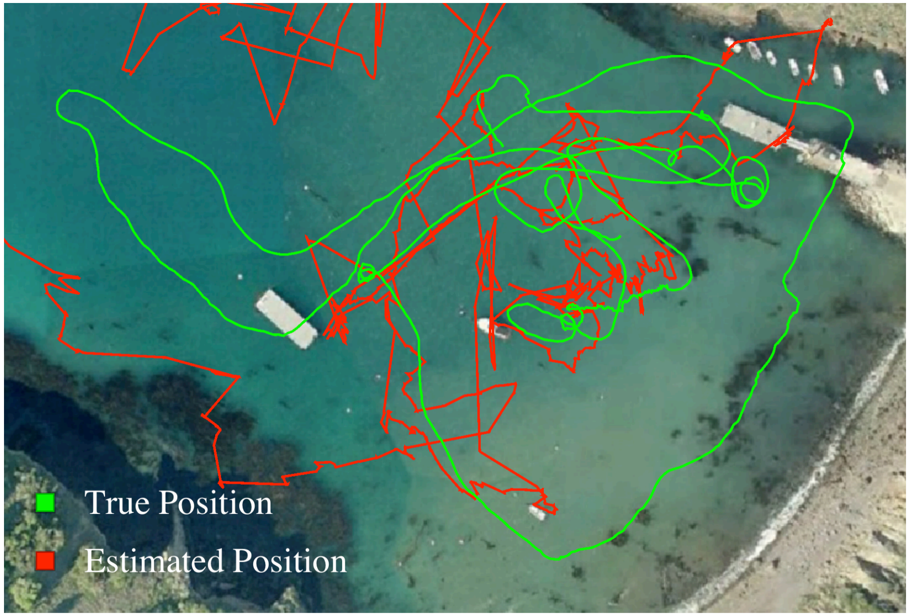


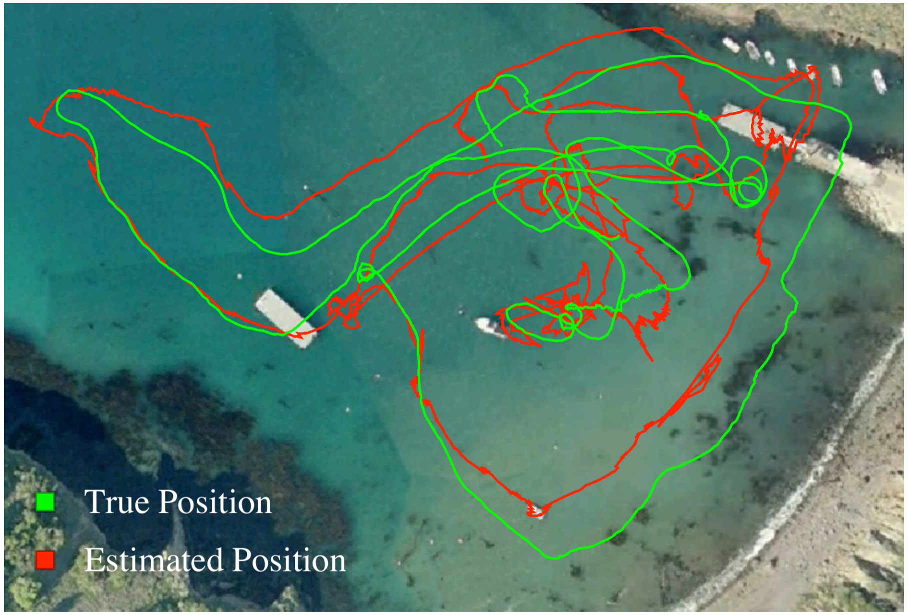
Shark Tagging

Result: Boat Track



Result: Boat Track





E80
Experimental Engineering

Result: Shark Track