UNIVERSITY OF SOUTH CAROLINA

CSCE 774 ROBOTIC SYSTEMS

Background

Spring 2017

Ioannis Rekleitis

Evaluation

- 3 Homeworks, 10% each: 30%
  1. ROS project
  2. Bibliography Search
  3. Vision based state estimation
- Final Project: 20%
- Class Participation: 20%
  – Prepare a small report on each paper/topic
- Presentations: 30%

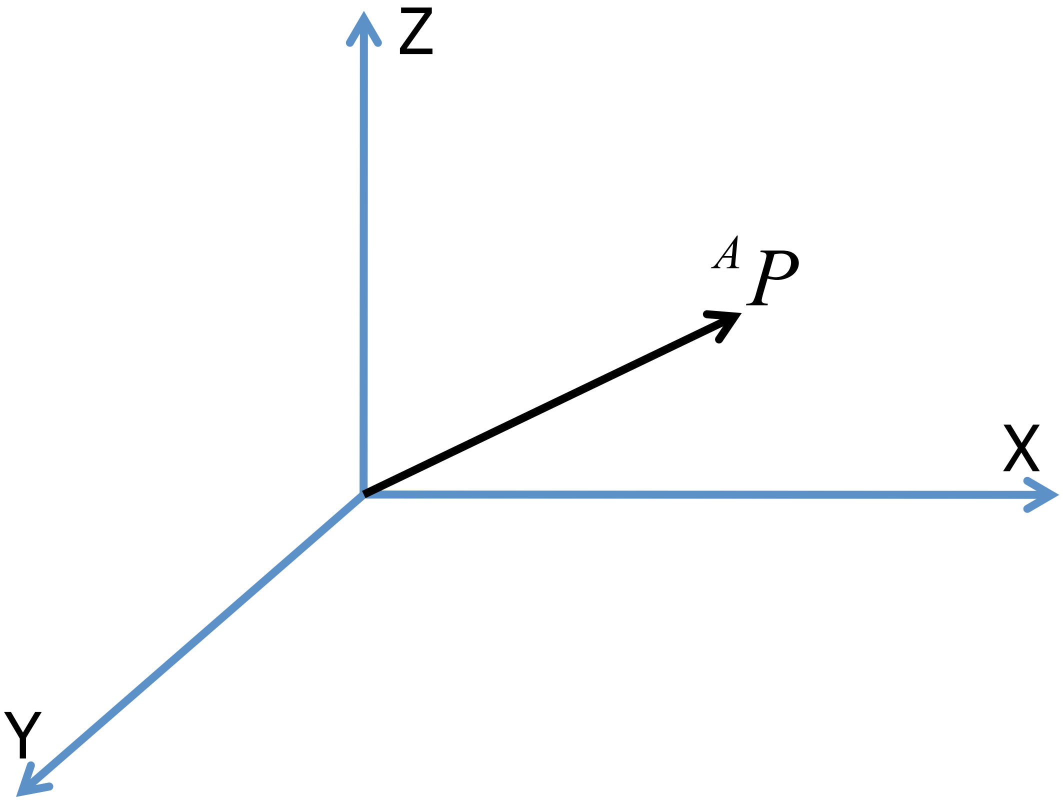
Homeworks/Projects

- Using ROS and OpenCV
- Using Simulations
- Using sensor data from real robots
- Using real robots (TurtleBot)

Position Representation

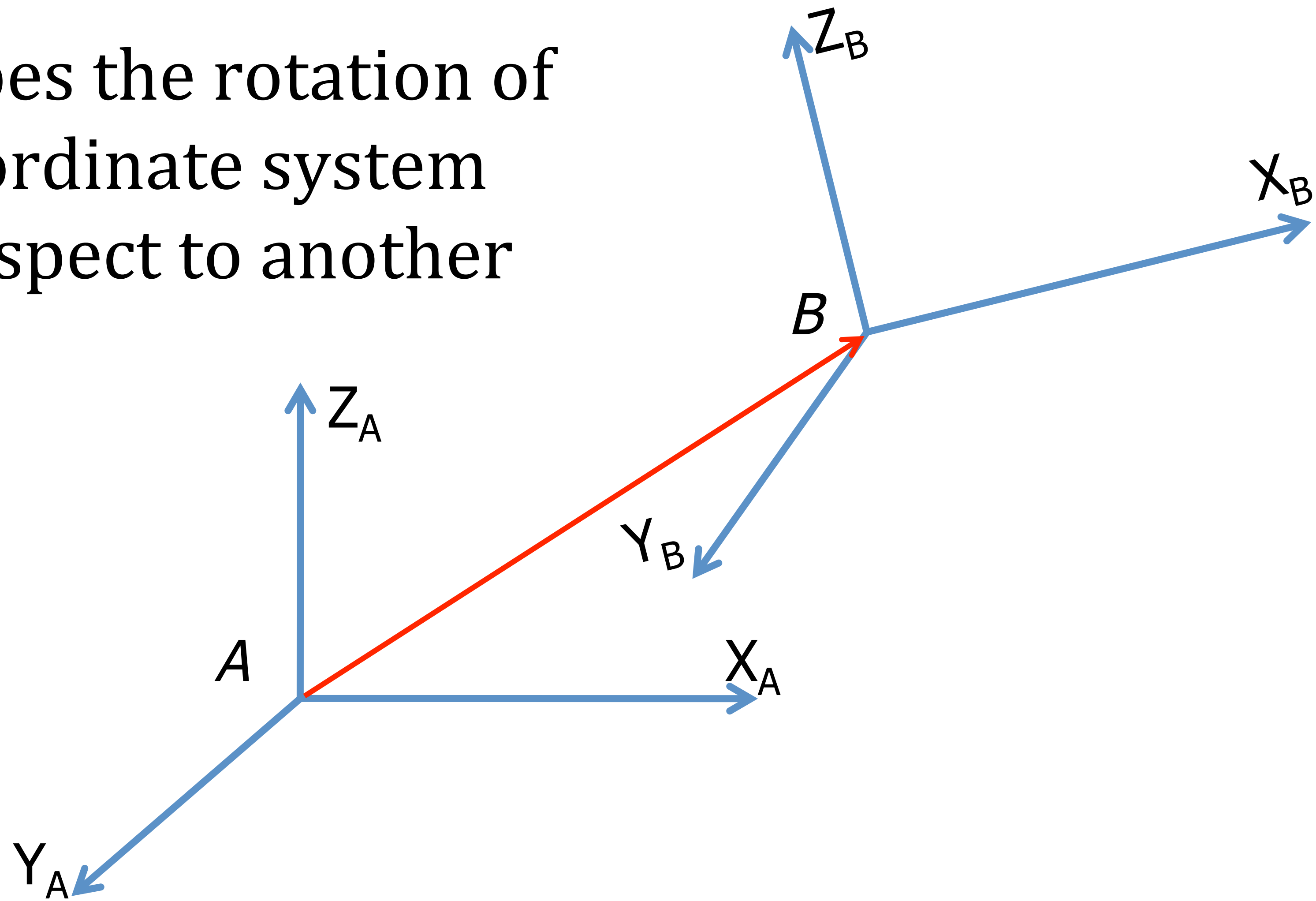
- Position representation is:

$$ {}^{A}P = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} $$



Orientation Representations

- Describes the rotation of one coordinate system with respect to another

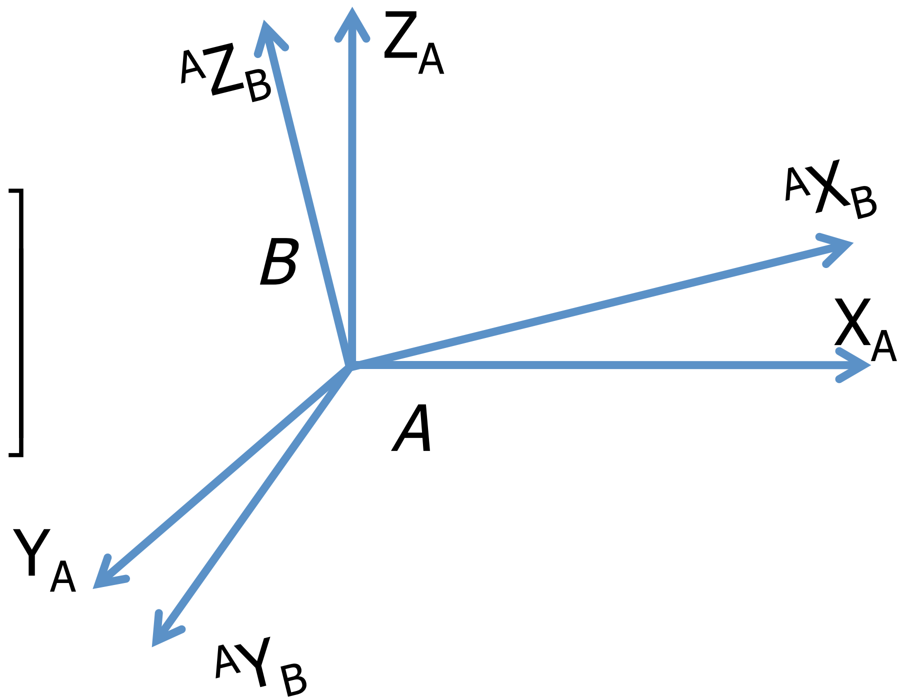


Rotation Matrix

- Write the unit vectors of B in the coordinate system of A.
- Rotation Matrix:

$$ {}_B^A R = \begin{bmatrix} {}^A\hat{X}_B & {}^A\hat{Y}_B & {}^A\hat{Z}_B \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} $$

$$ = \begin{bmatrix} \hat{X}_B \cdot \hat{X}_A & \hat{Y}_B \cdot \hat{X}_A & \hat{Z}_B \cdot \hat{X}_A \\ \hat{X}_B \cdot \hat{Y}_A & \hat{Y}_B \cdot \hat{Y}_A & \hat{Z}_B \cdot \hat{Y}_A \\ \hat{X}_B \cdot \hat{Z}_A & \hat{Y}_B \cdot \hat{Z}_A & \hat{Z}_B \cdot \hat{Z}_A \end{bmatrix} $$



Coordinate System Transformation

$$M = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & T \\ 0_{3\times1} & 1 \end{bmatrix}$$

where R is the rotation matrix and T is the translation vector

Rotation Matrix

- The rotation matrix consists of 9 variables, but there are many constraints. The minimum number of variables needed to describe a rotation is three.

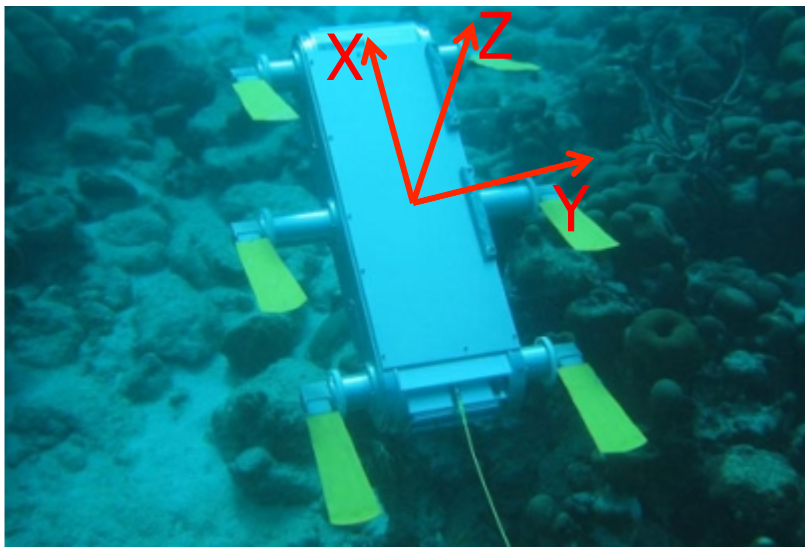
Euler Angles

- **ZYX**: Starting with the two frames aligned, first rotate about the Z_B axis, then by the Y_B axis and then by the X_B axis. The results are the same as with using XYZ fixed angle rotation.

- There are 12 different combination of Euler Angle representations

Euler Angles

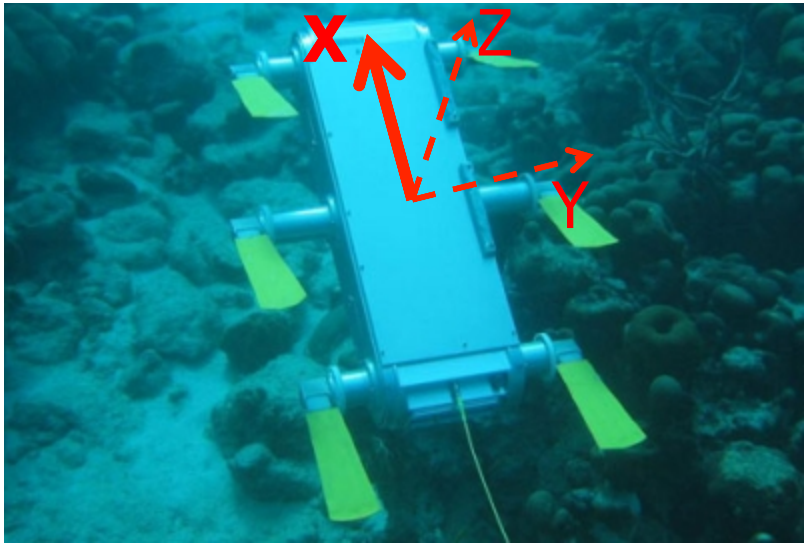
- Traditionally the three angles along the axis are called Roll, Pitch, and Yaw



Euler Angles

- Traditionally the three angles along the axis are called Roll, Pitch, and Yaw

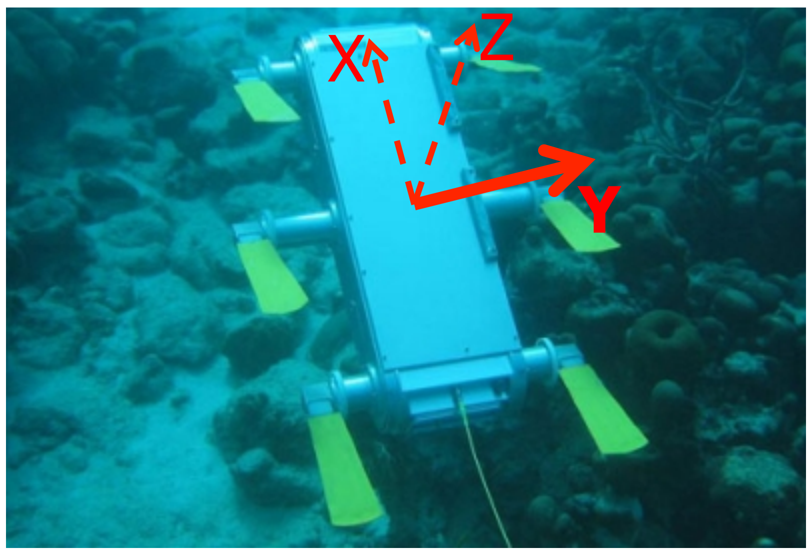
Roll



Euler Angles

- Traditionally the three angles along the axis are called Roll, Pitch, and Yaw

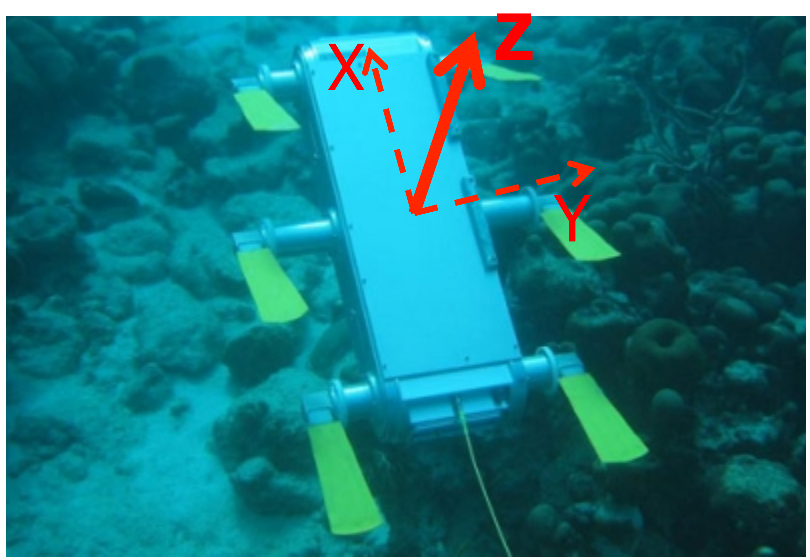
Pitch



Euler Angles

- Traditionally the three angles along the axis are called Roll, Pitch, and Yaw

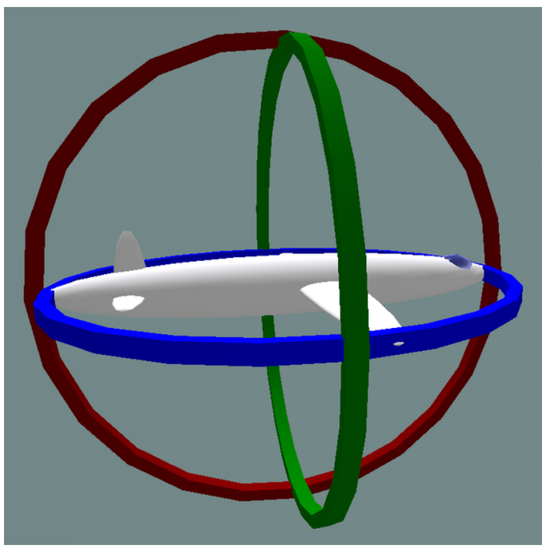
Yaw

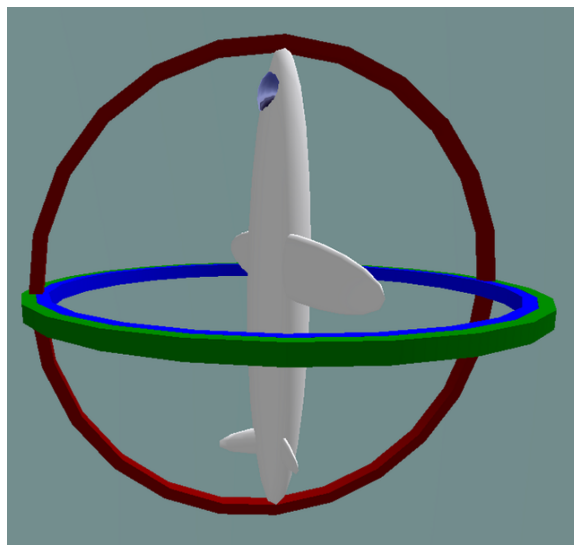


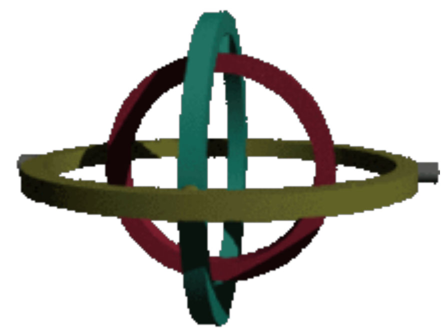
Euler Angle concerns: Gimbal Lock

Using the **ZYZ** convention

- (90°, 45°, −105°) ≡ (−270°, −315°, 255°)   multiples of 360°
- (72°, 0°, 0°) ≡ (40°, 0°, 32°)   singular alignment (Gimbal lock)
- (45°, 60°, −30°) ≡ (−135°, −60°, 150°)   bistable flip

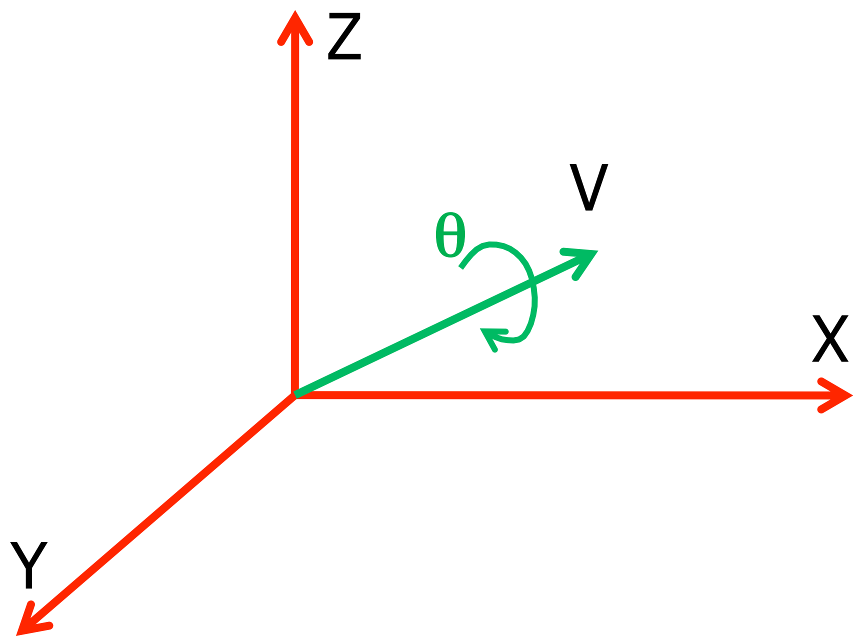






Axis-Angle Representation

- Represent an arbitrary rotation as a combination of a vector and an angle



Quaternions

- Are similar to axis-angle representation
- Two formulations
  - Classical
  - Based on JPL's standards

    W. G. Breckenridge, "Quaternions - Proposed Standard Conventions," JPL, Tech. Rep. INTEROFFICE MEMORANDUM IOM 343-79-1199, 1999.
- Avoids Gimbal lock

- See also: M. D. Shuster, "A survey of attitude representations," Journal of the Astronautical Sciences, vol. 41, no. 4, pp. 439–517, Oct.–Dec. 1993.

Robot Sensors

- Sensors are devices that can sense and measure physical properties of the environment,
  - e.g. temperature, luminance, resistance to touch, weight, size, etc.
  - The key phenomenon is transduction
    - Transduction (engineering) is a process that converts one type of energy to another
- They deliver *low-level* information about the environment the robot is working in.
  - Return an incomplete description of the world.

Robot Sensors

- This information is **noisy** (imprecise).
- Cannot be modelled completely:
  - Reading = *f*(env)   where *f* is the model of the sensor
  - Finding the inverse:
    - ill posed problem (solution not uniquely defined)
    - collapsing of dimensionality leads to ambiguity

Types of sensor

- General classification:
  - **active** *versus* **passive**
    - Active: emit energy in environment
      - More robust, less efficient
    - Passive: passively receive energy from env.
      - Less intrusive, but depends on env. e.g. light for camera
    - Example: stereo vision versus range finder.
  - **contact** *versus* **non-contact**

Sensors

- **Proprioceptive Sensors**
  (monitor state of robot)
  - IMU (accels & gyros)
  - Wheel encoders
  - Doppler radar …

- **Exteroceptive Sensors**
  (monitor environment)
  - Cameras (single, stereo, omni, FLIR …)
  - Laser scanner
  - MW radar
  - Sonar
  - Tactile…

Sensor Characteristics

- All sensors are characterized by various properties that describe their capabilities
  - **Sensitivity**:
    (change of output) ÷ (change of input)
  - **Linearity**: constancy of (output ÷ input)
    - Exception: logarithmic response cameras == wider dynamic range.
  - **Measurement/Dynamic range**:
    difference between min. and max.

Sensor Characteristics

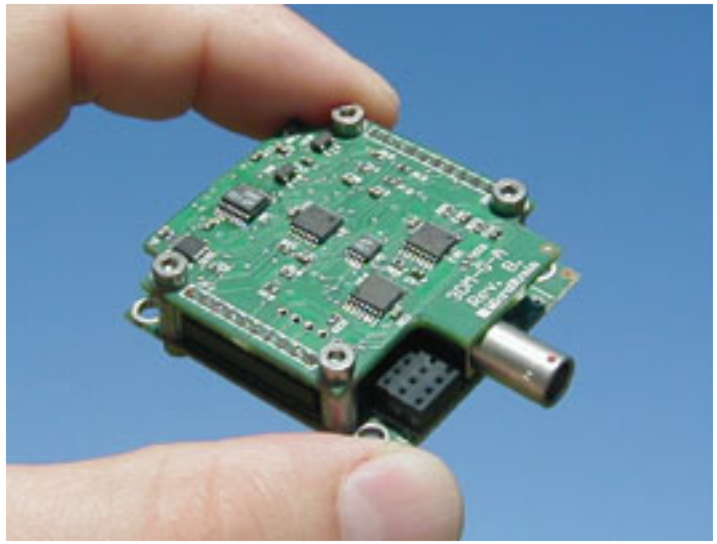
- **Response Time**: time required for a change in input to cause a change in the output
- **Accuracy**: difference between measured & actual
- **Repeatability**: difference between repeated measures
- **Resolution**: smallest observable increment
- **Bandwidth**: result of high resolution or cycle time

Focus on:

- IMU
- Wheel Encoders
- Compass
- Monocular Vision
- Stereo Vision
- RGBd (Kinnect)
- LIDAR

IMU's

- Gyro, accelerometer combination.
- Typical designs (e.g. 3DM-GX1™) use tri-axial gyros to track dynamic orientation and tri-axial DC accelerometers along with the tri-axial magnetometers to track static orientation.
- The embedded microprocessors contains programmable filter algorithms, which blend these static and dynamic responses in real-time.



Why vision?

- Passive (emits nothing).
  - Discreet.
  - Energy efficient.
- Intuitive.
- Powerful (works well for us, right?)
- Long and short range.
- Fast.

So, what's the problem?

- How hard is vision? Why do we think is do-able?

Problems:

- Slow.
- Data-heavy.
- Impossible.
- Mixes up many factors.