Robotic Merit Badge Session #4

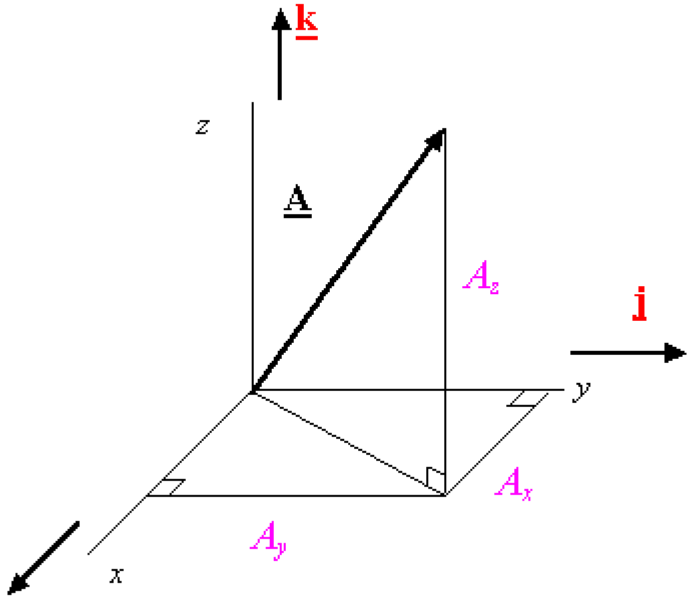
- Merit Badge Counselor: Maurice Ling
- August 17, 2015
- http://github.com/mcli/RoboticsMB

Agenda

- Programming Accelerometer inputs
- Review Competition Rules
- Testing Checklist

Force Vector

- Force is a vector with components in the x, y, and z directions



Accelerometer Principles

- The accelerometer measures acceleration along the x, y, and z axes of the microMagician board.
- When at rest, the accelerometer measures gravitational force.
- What is the relationship between force and acceleration?

Newton's 2nd Law

$$F = ma$$

Force = Mass * Acceleration

- On Earth, $a = g \sim 9.8 \text{ m}/\text{sec}^2$

Accelerometer Measurements

- Measurements from the accelerometer are taken from analog pins 0, 1, and 2 and stored in microM.xaxis, microM.yaxis, and microM.zaxis.
- Raw measurements are from 0 through 1024
- When at rest, these measurements correspond to the gravitational force exerted on the board.
- We can convert these measurements into numbers that we can understand and easily process.

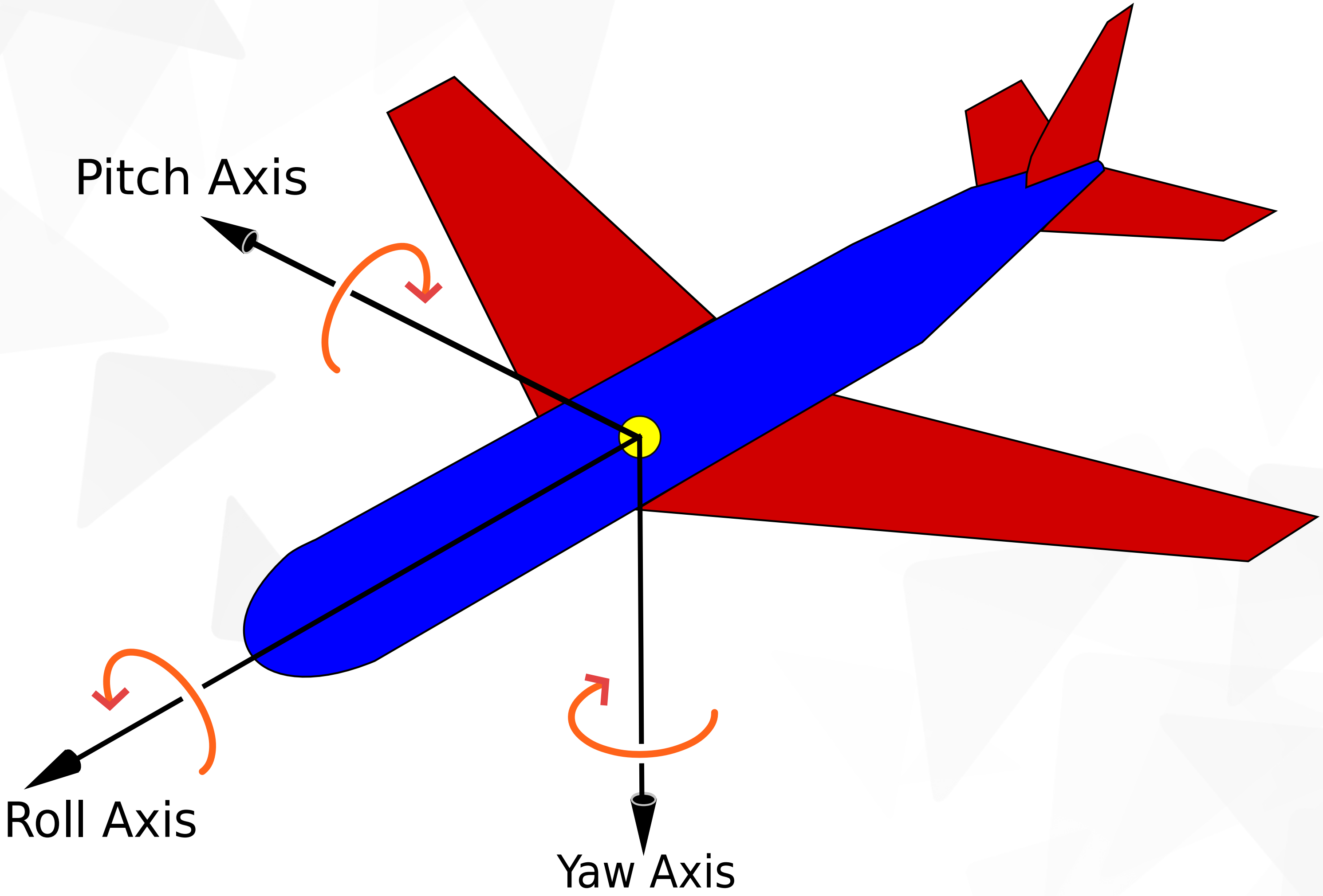
Normalizing the Accelerometer Measurements

- Find the “zero” value of the x, y and z axes by aligning the axes of interest perpendicular to gravity. This is your zero offset.
- Find the measurement value from the zero value corresponding to the gravitational force. This is your scale factor.
- Your normalized measurements will be in units of gravitational force

```
gForce(i) = (measurement(i) -offset(i))/scaleFactor;
```

What about direction?

- How do you represent the direction of the your robot?



Computing the Orientation

- Use trigonometric approximation to get pitch and roll angles.

```
// apply trigonometry to get the pitch and roll:
float pitch = atan(xAxis/sqrt(pow(yAxis,2) +
pow(zAxis,2)));
float roll = atan(yAxis/sqrt(pow(xAxis,2) +
pow(zAxis,2)));

//convert radians into degrees
float pitchInDegrees = pitch * (180.0/PI);
float rollInDegrees = roll * (180.0/PI) ;
```

- Run the AccelerometerOrientationAngles program. Do the output numbers match what you expect?

Robotic Competition Rules

- 5 minute time limit
- 4 Components
  - Design in Engineering Notebook
    - Programming Logic
    - Mechanical design
    - Circuit diagram
    - Testing Checklist & results
  - Loading – manual control
  - Transport Course – automatic navigation
  - Unloading – manual control
- Complete rules on GitHub

Competition Scoring

- +20 Points scored for each segment completed
- +25 Points for Design
- Bonus points:
  - Time completed within 3 minutes
  - Innovative/elegant design (Up to +20)
- Penalty points for:
  - Handling payload
  - Manual intervention
  - Contact with barriers/objects
- Robot must not be moved, turned, or otherwise transported physically by a human.
- Judge panel consisting of industry professionals.

Creating a Test Checklist

- A Test Checklist will help you assess your robot and identify areas you need to change or improve .
- What scenarios can you think of for the competition?

Example Test Checklist

Scenario	Expected Behavior	
Manual Naviation	• Left turn • Right turn • Forward • Reverse • In-place turn	
Loading	• Manually navigate to payload holder • Load payload onto Robot	
Unloading	• Manually navigate to target • Unload payload onto receptacle	
Auto/Manual transition	• Automatic to Manual control. • Manual to Automatic control	
Auto navigation	• U turn • Left turn • Right turn • Go around Island object	

Testing Your Robot

- Test your robot on the different test stations
- Record your results in your engineering notebook
- Write down any improvements and adjustments you need to make.