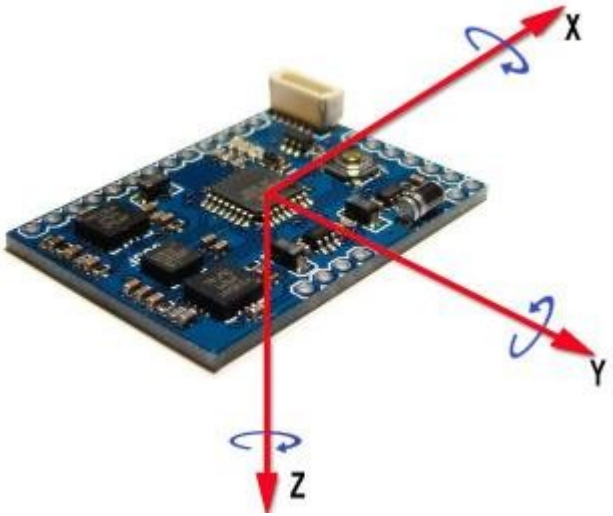
How ArduPlane Works

Andrew Tridgell
ArduPlane development team

ArduPlane Sensors

- Most planes have these sensors:
  - 3-axis gyroscope
  - 3-axis accelerometer
  - 3-axis magnetometer
  - barometer : relative height
  - GPS : 3D position and 3-axis velocity
  - airspeed sensor : indicated airspeed



- Key questions:
  - what is the resolution and saturation point of each sensor?
  - what is the sample rate and lag of each sensor?
  - how is each sensor affected by noise?
  - what types of errors does each sensor have?

How Attitude is Estimated

To determine attitude we need to use "sensor fusion" to combine our sensors. A single sensor won't give us accurate attitude.

- in short term (1-2 seconds) the gyroscope is used
- the roll and pitch is corrected slowly (over 15 seconds or so) using the accelerometer
- the yaw is correctly slowly (over 15 seconds or so)

Why can't we just use the gyros to get attitude?

How many axes do you need for attitude?

Let's see a demo of attitude with just gyros!

Frames of reference

It is important to know the “frame of reference” of each sensor

- “earth frame” means relative to someone standing on the ground facing north
- “body frame” means relative to the aircraft body

Which sensors are in earth frame? Which ones are in body frame?

Let's see a demo of earth and body frame gyros

Controlling the flight surfaces

- How do we control the ailerons?
  - We want the ailerons to control our “roll” in earth frame
  - We want to induce roll when we need to turn
- Basics of a controller
  - Define a “process variable” (e.g. the roll)
  - Define a “setpoint” (e.g. the target roll)
  - Calculate error as difference between variable and setpoint
  - Try to move the surface to reduce the error

A 'P' controller

- Simplest controller is a 'P' controller
  - 'P' stands for 'proportional'
  - output = P * error
- For aileron 'output' is the servo input
  - ArduPlane defines servo inputs as angles from -45 degrees to 45 degrees
- How does the value affect flight?
  - If P is too large, roll will oscillate
  - if P is too small, plane will respond slowly

Let's see a demo of a P aileron controller

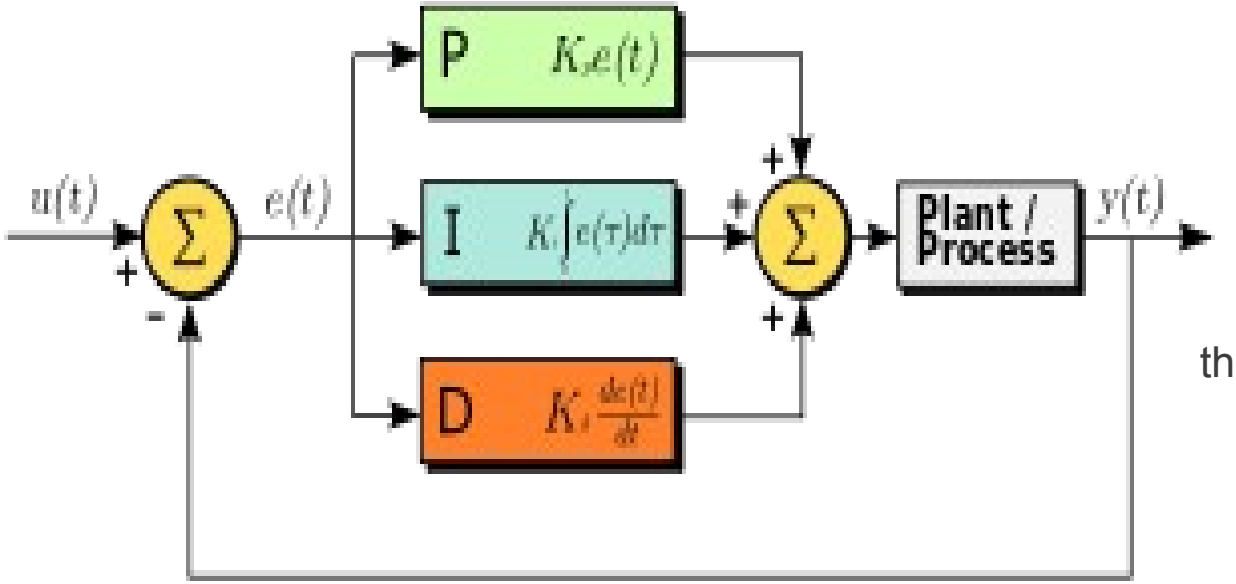
A 'P' controller

- Simplest controller is a 'P' controller
  - 'P' stands for 'proportional'
  - output = P * error
- For aileron 'output' is the servo input
  - ArduPlane defines servo inputs as angles from -45 degrees to 45 degrees
- How does the value affect flight?
  - If P is too large, roll will oscillate
  - if P is too small, plane will respond slowly

Let's see a demo of a P aileron controller

A PID controller

- A 'PID' controller adds 'I' and 'D' terms
  - The 'I' term copes with long term error
  - The 'D' term tries to prevent overshoot
  - APM adds an 'IMAX' term to prevent too much 'I' buildup



thanks to wikipedia (CC-SA)

Let's see what adding 'I' does to the aileron

ArduPlane PID controllers

- ArduPlane uses lots of PID controllers
  - ServoRoll
  - ServoPitch
  - ServoRudder
  - NavRoll
  - NavPitchAirspeed
  - TotalEnergyThrottle
  - NavPitchAltitude
  - WheelSteer
  - AHRS DriftCorrection (PI only)
- What process variable and setpoint would be used for each of these?

Read the code!

- If you are interested in learning more, have a look at the source code
    - http://code.google.com/p/ardupilot-mega/