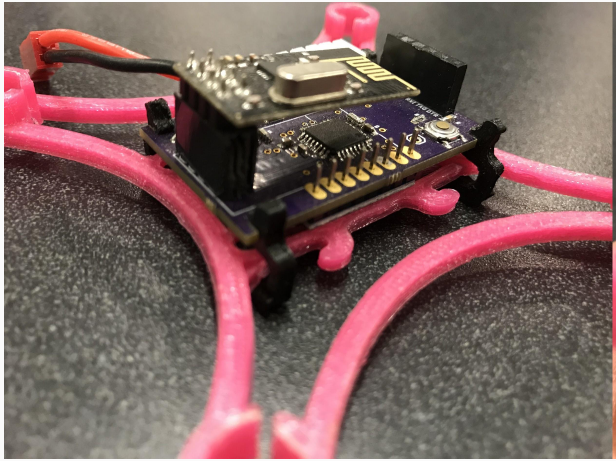
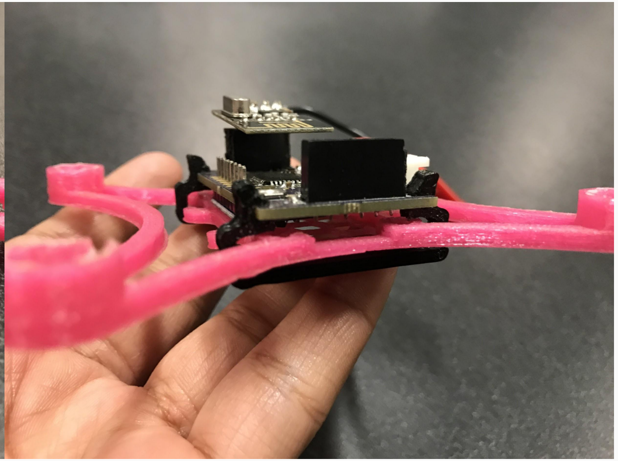
Advanced Projects

Lecture 5: Flight Software

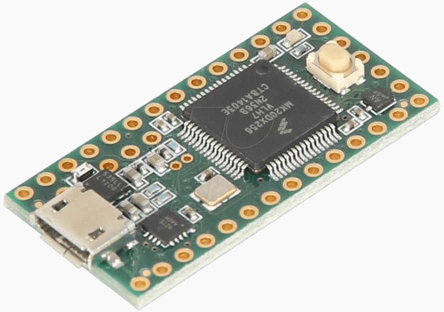
Announcements

- You should be wrapping up assembly of your board.
- Uploading code onto the ATmega requires burning the bootloader.
- To borrow a frame for testing, talk to us.
    - You may have to file the frames for the board to fit in properly.
- When testing the motors, TAKE THE PROPELLERS OFF.

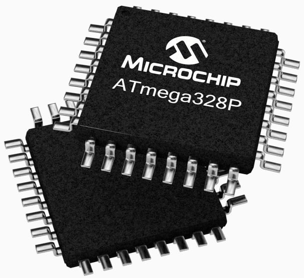



Burning the Bootloader

- In order to upload code, you first have to burn the bootloader using a Teensy LC



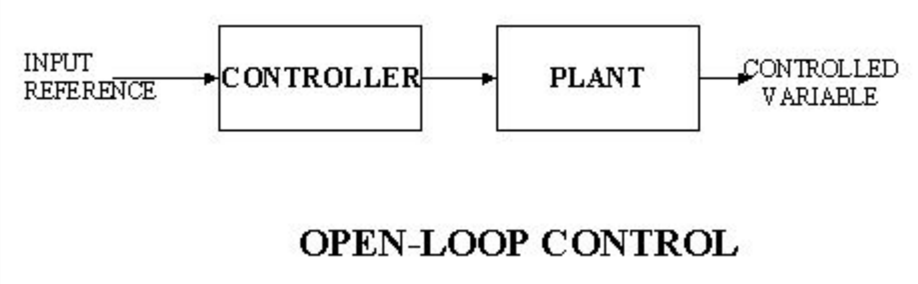
MOSI	⟷	MOSI
MISO	⟷	MISO
SCK	⟷	SCK
RESET (D10)	⟷	RESET
VCC	⟷	VCC
GND	⟷	GND



- SPI pins are already broken out for radio modules
  - Manually make contact for the reset pin
- Only has to be done once
  - Can program using the FTDI Tx and Rx pins

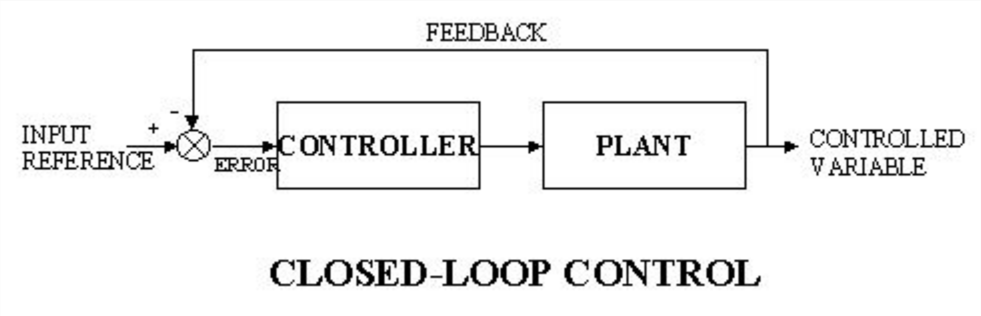
Open-Loop Control

- Using physics we can find equations that govern quadcopter flight
- Requires advanced math/physics
- Requires exact measurements of quad properties
- Does not account for noise or random events
- Open-loop control is not realistic for most use cases



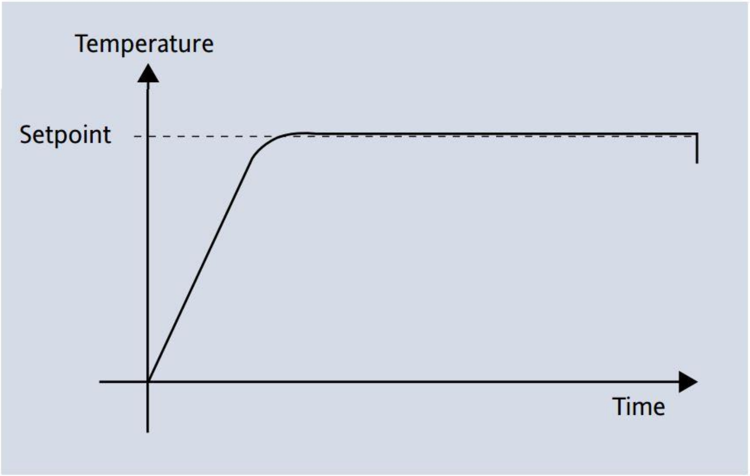
Feedback & Closed-Loop Control

- Sensors can measure the state of our system (e.g. quadcopter orientation)
- Controller can change its behavior to correct for error
- Feedback automatically corrects for noise and random events
- Exact physics are unimportant, only general trends matter (e.g. which direction to apply force)
- Almost all control systems use feedback



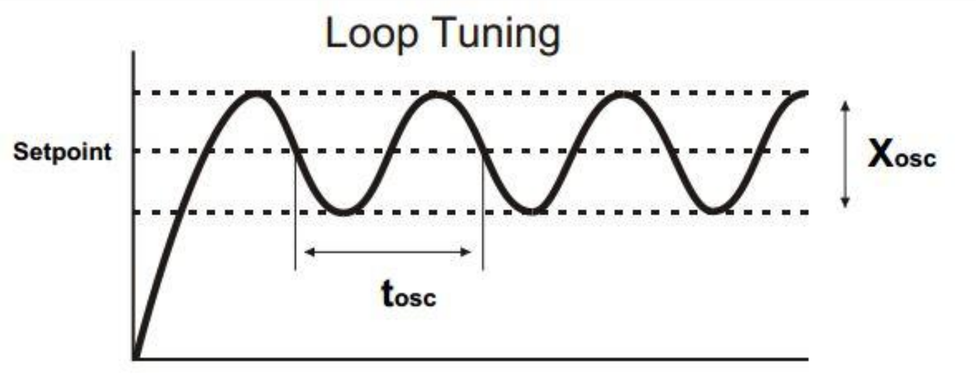
Proportional Control

- Correction should be in the same “direction” as error
- Large error requires large correction, small error requires small correction
- $y = K_p e$
  - y: Output signal
  - e: Error signal
  - K_p: Proportional coefficient
- Coefficient is usually found experimentally



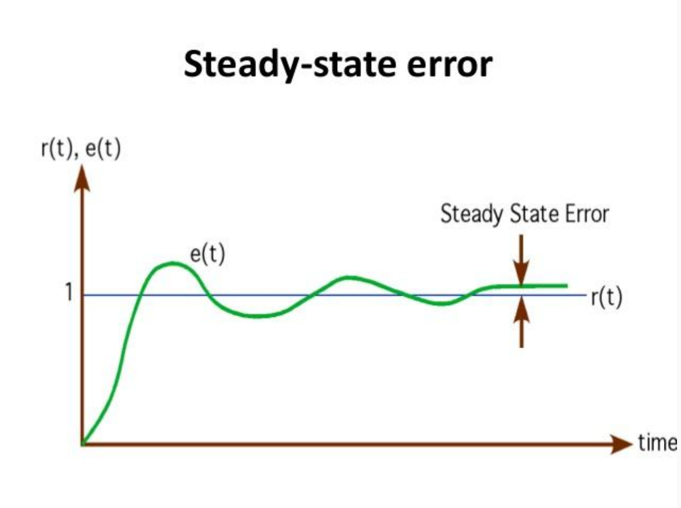
Derivative Control

- Real systems have momentum, causing oscillations around the set point
- If error is growing, we need a larger correction to overcome momentum
- If error is shrinking, we need a smaller correction to stop overshoot
- $y = K_d(e_i - e_{i-1})$
  - $e_i - e_{i-1}$ is the difference between current and previous error, which is a discrete derivative
- Derivative must be combined with proportional to be useful (PD control)



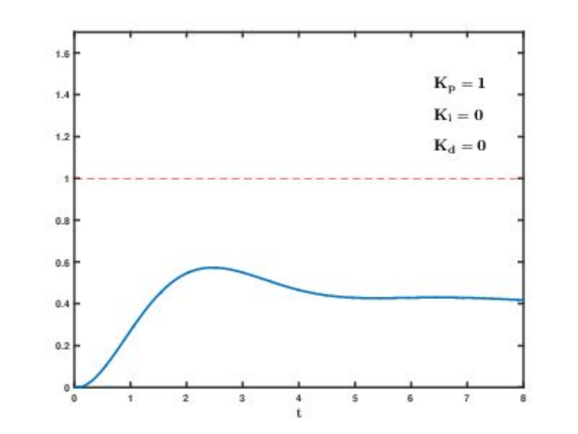
Integral Control

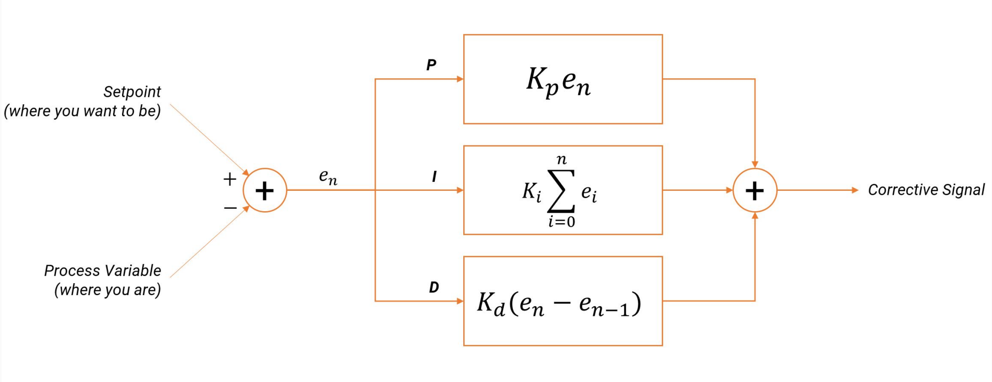
- Real systems also have steady state forces acting on them (e.g. gravity)
    - These forces can counteract correction, creating a steady state error
- To detect and correct for this error, we can use an integral
- $y = K_i \Sigma e$
    - Summation is discrete equivalent of an integral
- Average error must be zero, creates oscillations
    - Avoid large K_i to prevent large oscillations
- Error sum can overflow if too large/small
    - Check for bounds on summation



PID Control & Tuning

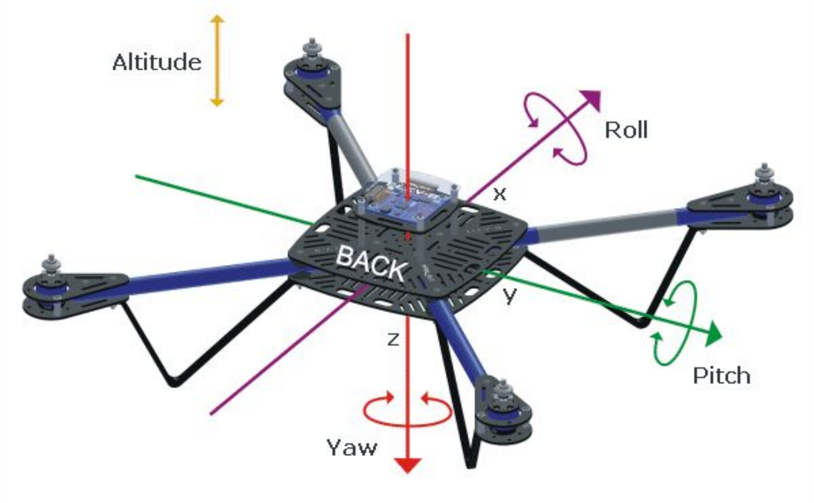
- For complete control, we need to use all three coefficients
- Start with P control, as it is most important
- Increasing K_p and K_i decreases rise time but increases overshoot/oscillation
    - K_i also removes steady state error
- Increasing Kd decreases overshoot and oscillation but increases rise time





Euler Angles

- Quaternions are useful for tracking orientation, not control
- Quaternion can be broken into 3 angles
  - Roll (x-axis)
  - Pitch (y-axis)
  - Yaw (z-axis)
- We use 6 DOF tracker
  - Cannot track yaw
  - Instead use yaw rate from gyroscope



Mapping PID to Motors

- Roll, pitch, and yaw are closed-loop
- Throttle is open-loop