Implementation of a Controller
Alex Spitzer aes368@cornell.edu
12/7/14
Why?

- Abstraction
- Abstraction
- Thrusters driven with electricity
- Webinar will assume method to drive thrusters
- E.g. PWM, simple brushed motors
- 255 full speed forward
- -255 full speed reverse
Abstraction

- Autonomous missions should not be concerned with thruster voltages
- Missions desire vehicle state
- Position and velocity
- Choose abstraction to suite hardware
- CUAUV: Heading, pitch, roll (orientation)
  - Forward velocity, sway velocity, depth
- Can measure absolute orientation (IMU)
- Can measure velocity
- Can measure depth
- Position can be another layer
Single variable control: P loop

- Can measure value
- Can affect value indirectly, continuously
- Thrusters impart force
- e.g. Depth
- First approach:
- See if you need to go up or down
- Force in the opposite direction
- Primitive
- Observation: The closer we are to our desired depth, the less force we should apply
- Scale force by “error” (proportional)
- Force = P * (desired_depth - depth)
Single variable control: PD loop

- Problems with P control: overshooting
- Solution? Add negative force to slow down
- Add proportional to rate of approach
- Derivative
- Force = $P \times (\text{desired\_depth} - \text{depth}) + D \times \frac{d}{dt} (\text{desired\_depth} - \text{depth})$
Single variable control: PID loop

- Problems with PD control: steady state error
- Solution? Add force proportional to time spent in error
- Integral
- Force = \( P \times (\text{desired\_depth} - \text{depth}) + \int I \times (\text{integral}(\text{desired\_depth} - \text{depth})) \)
  \[ D \times (\frac{d}{dt} (\text{desired\_depth} - \text{depth})) \]

- For depth, this “learns” the buoyancy force
- Can be useful to “learn” drag
- Poor man’s system characterization
- Clunky, e.g. No drag, when not moving
System Setup

- PID loops work great in one dimension
- Can also use them for orientation (quaternion PID)
- CUAUV: Quaternion, Vel x, Vel y, depth
- Alternative: one for each DOF
Combining PID loops

- Can interpret output of PID loops as desired acceleration
- Each PID loop requests acceleration to achieve its desire
- Should be independent
- E.g. Quaternion PID loop requests angular acceleration about an axis
- X, Y, and depth PID loops request accelerations in the x, y, and z axes
- Trick: Add acceleration vectors!
- Can add angular accelerations as well!
- Now we have a single desired acceleration vector.
Thruster mapping

- Can convert desired acceleration to desired force or torque.
- $F = ma$, Torque = $I \times \alpha$
- $I$ is the moment of inertia matrix (3 x 3)
- Thrusters output forces
- Each thruster’s contribution can be summed to give total force and torque on the vehicle
- For each thruster: axis of action, position
  - $F = \text{scalar}_\text{force} \times \text{axis}$
  - Torque = position $\times$ $F$
- Each thruster’s axis is column in matrix
- $F = A \times x$, where $x$ is vector of scalar_forces
Thruster mapping

- We have desired $F$, want $x$!
- $x = A^{-1} F$
- Want minimum norm solution
- But wait! $F$ is force from thrusters.
- PID loops don’t care where the force comes from.
- Other forces on vehicle?
  - Buoyancy is huge
  - Drag
  - Outside forces (impossible to anticipate)
- $F = \text{desired\_force} - \text{passive\_forces}$
Force estimation

- Buoyancy force acts through center of buoyancy
- Gravity acts through center of mass
- Buoyancy torque is $r \times F$
Force estimation

- Drag increases with velocity
- Can make model to capture drag due to linear translation, angular rotation
- Force estimation takes work off PID loops
Thruster Characterization

- We would like to give each thruster a force
- In reality we can only give each thruster a voltage
- How does that map to force?
- Can do experiments!
- Curve fit to data points, LUT (look up table)
- This is quite important
Next Level

- Online characterization
- Information is there, don’t do the same things twice expecting different results
- Model updates based on observations
Questions?
aes368@cornell.edu