EECS 106B/206B
Discussion 4

GSI: Valmik Prabhu

Action Items

- Concurrent Enrollment students fill out both forms:
  - Student accounts
  - Lab access
- Remember, all students who did not take 106a must finish lab 0 with a GSI before you're allowed to use the robots
  - M 1-2, W 4-5, Th 2-4
- Homework 2 delayed till Wednesday (it has a HW 1 solution in it)
- Lab Due in 13 days!!!!

PID Control

- Proportional
  - Work horse term
  - Causes oscillation
- Derivative
  - Stabilizing influence
  - Decrease overshoot and ringing, but slows response
  - Problem: sensor noise
- Integral
  - Eliminates constant disturbances, but decreases stability
  - Problem: integrator windup

Integrator Windup

- Finite horizon integrator
- Weighted horizon integrator
- Bounds on the integral term
- Turn off the integrator when error is high

First Order and Second Order Controllers

PID:

$$u = (K_P + K_I \frac{1}{s} + K_d s)e$$

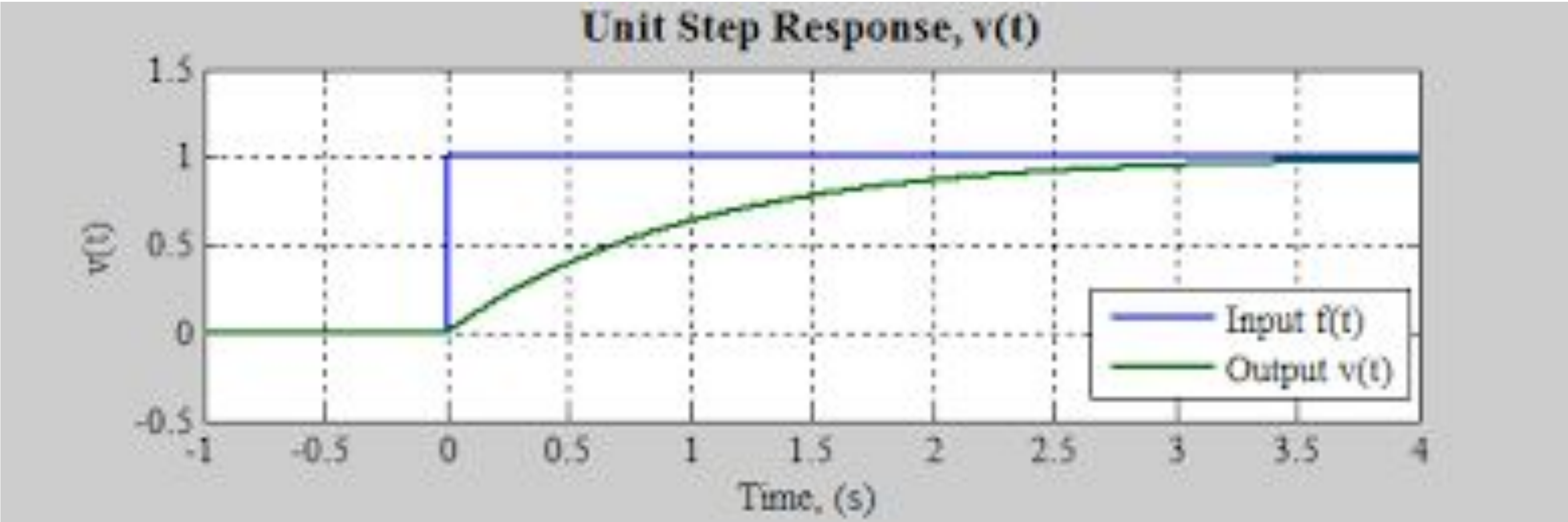
First Order:

$$u = \frac{k_{dc}}{\tau s + 1}$$

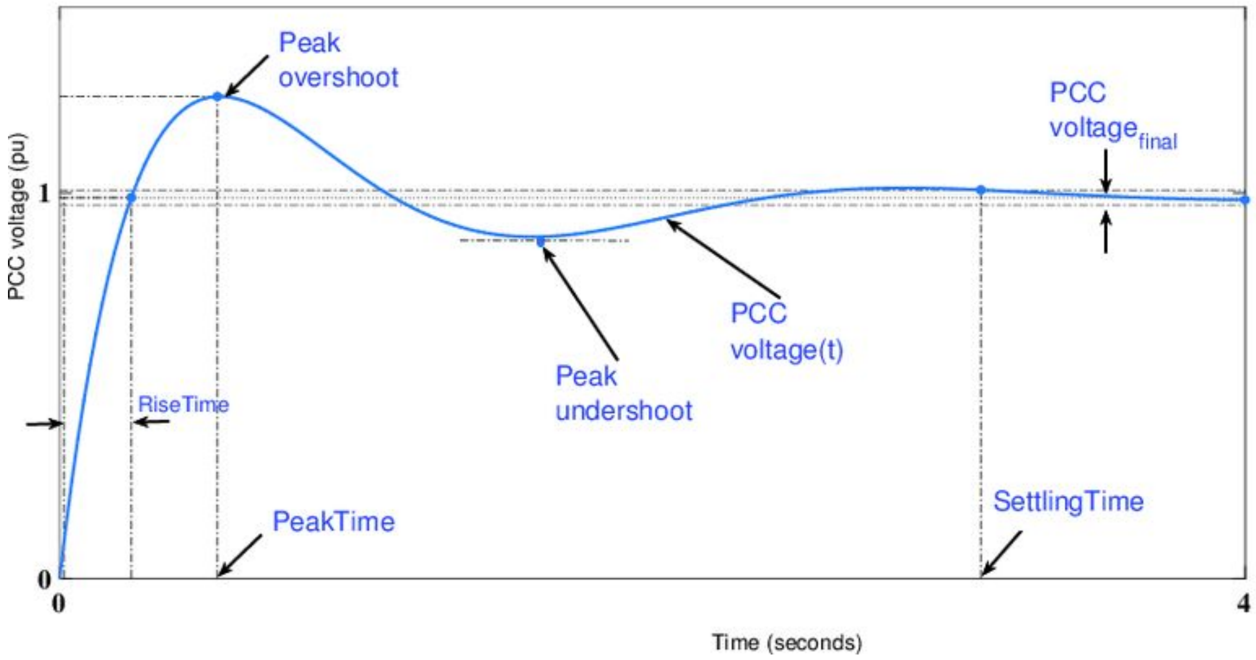
Second Order:

$$u = \frac{k_{dc}\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$

First Order System Response



Second Order System Response

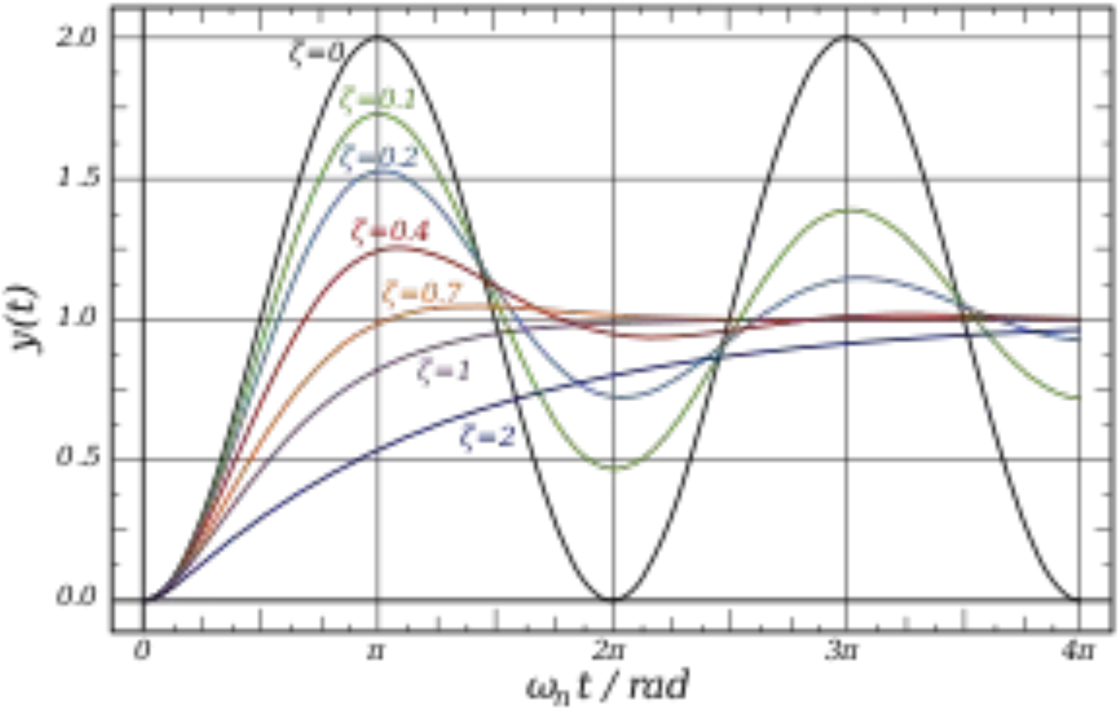


Effect of Damping

Underdamped

Critically Damped

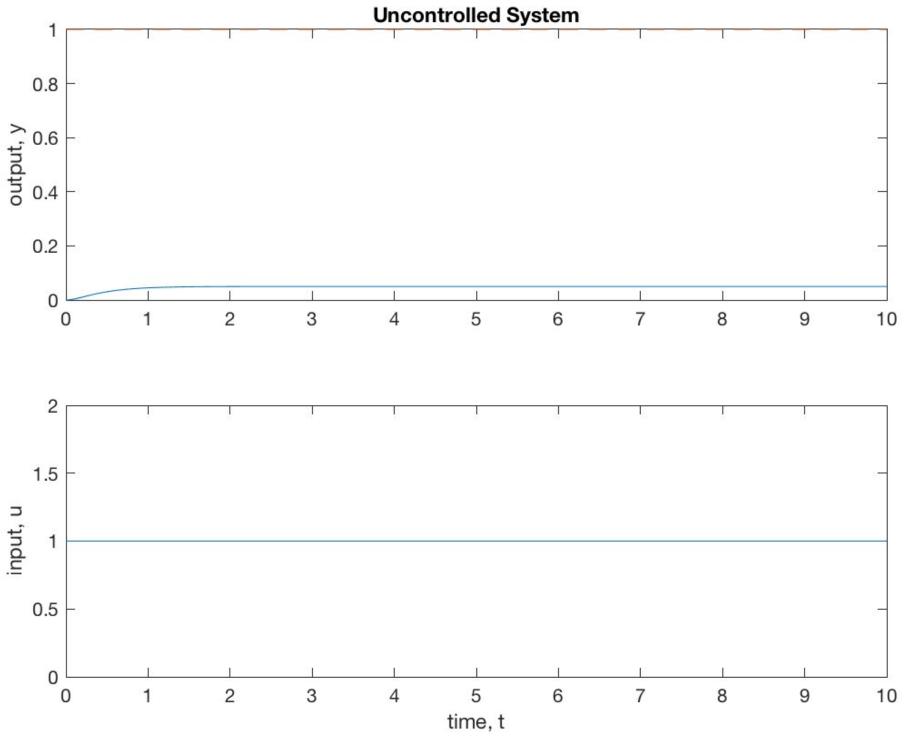
Overdamped



Uncontrolled System

What order system is this?

- Can’t tell (looks like first)
- It is second order
- Overdamped



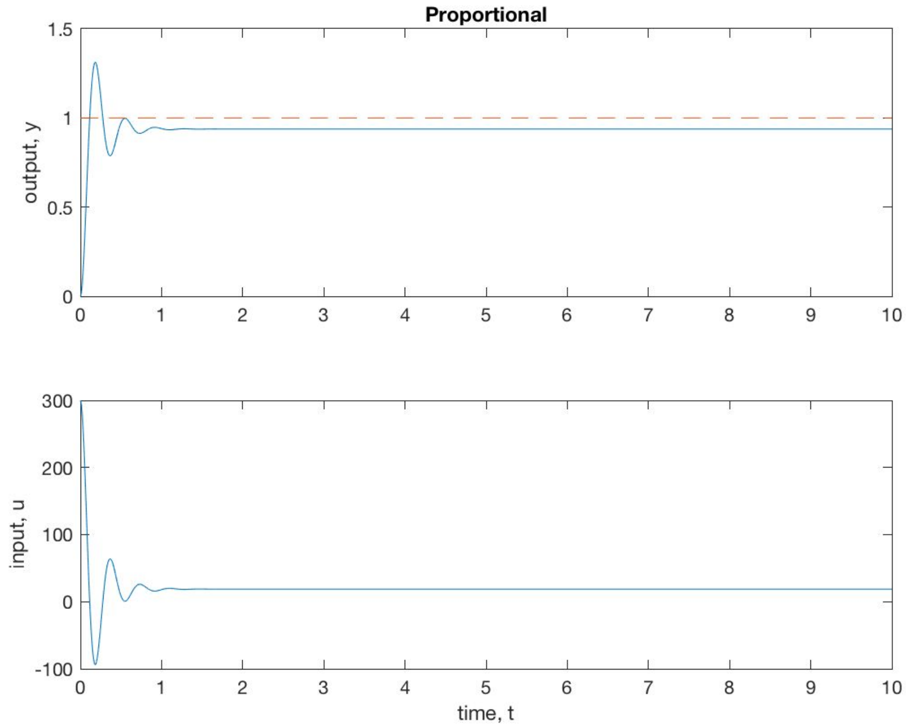
Proportional

What’s the percent overshoot?

- Around 30%

Why does the system settle?

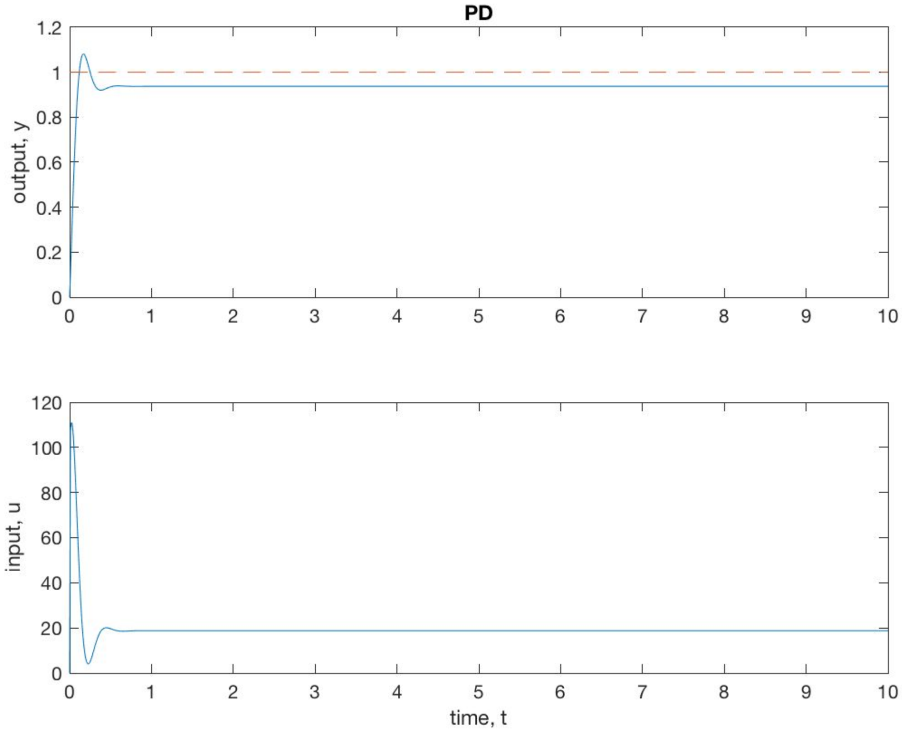
- The system has inherent damping



PD

What is the derivative doing?

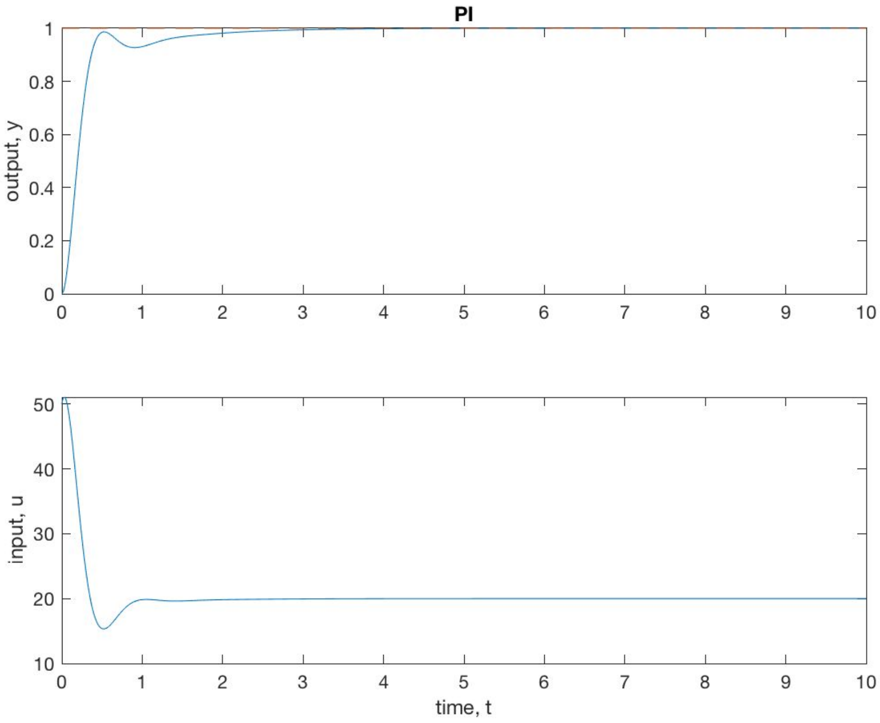
- Reduces overshoot
- Reduces overall energy



PI

What is the integrator doing?

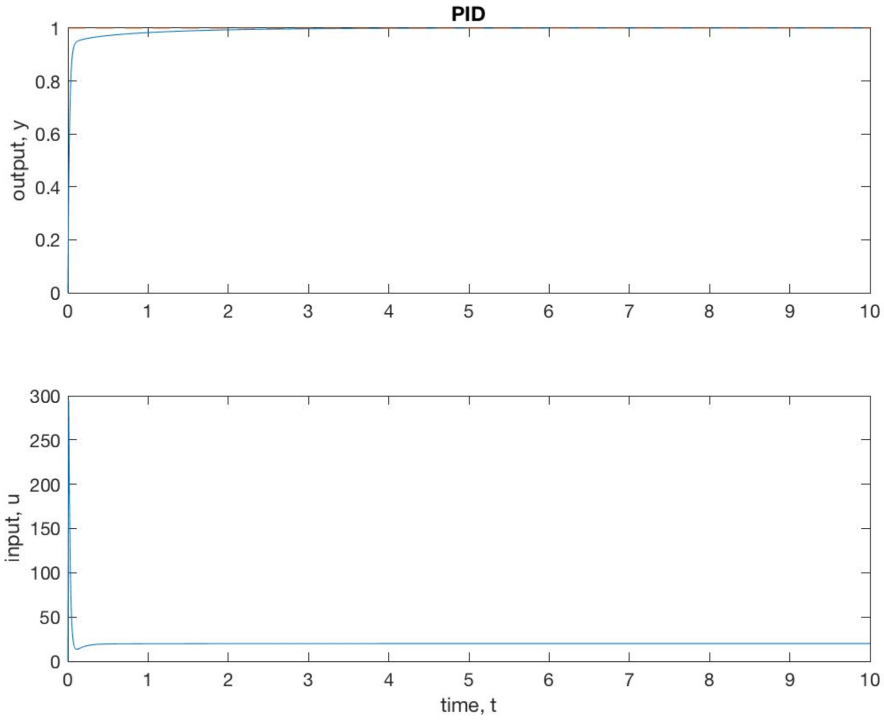
- Reducing static error
- Integrator delay



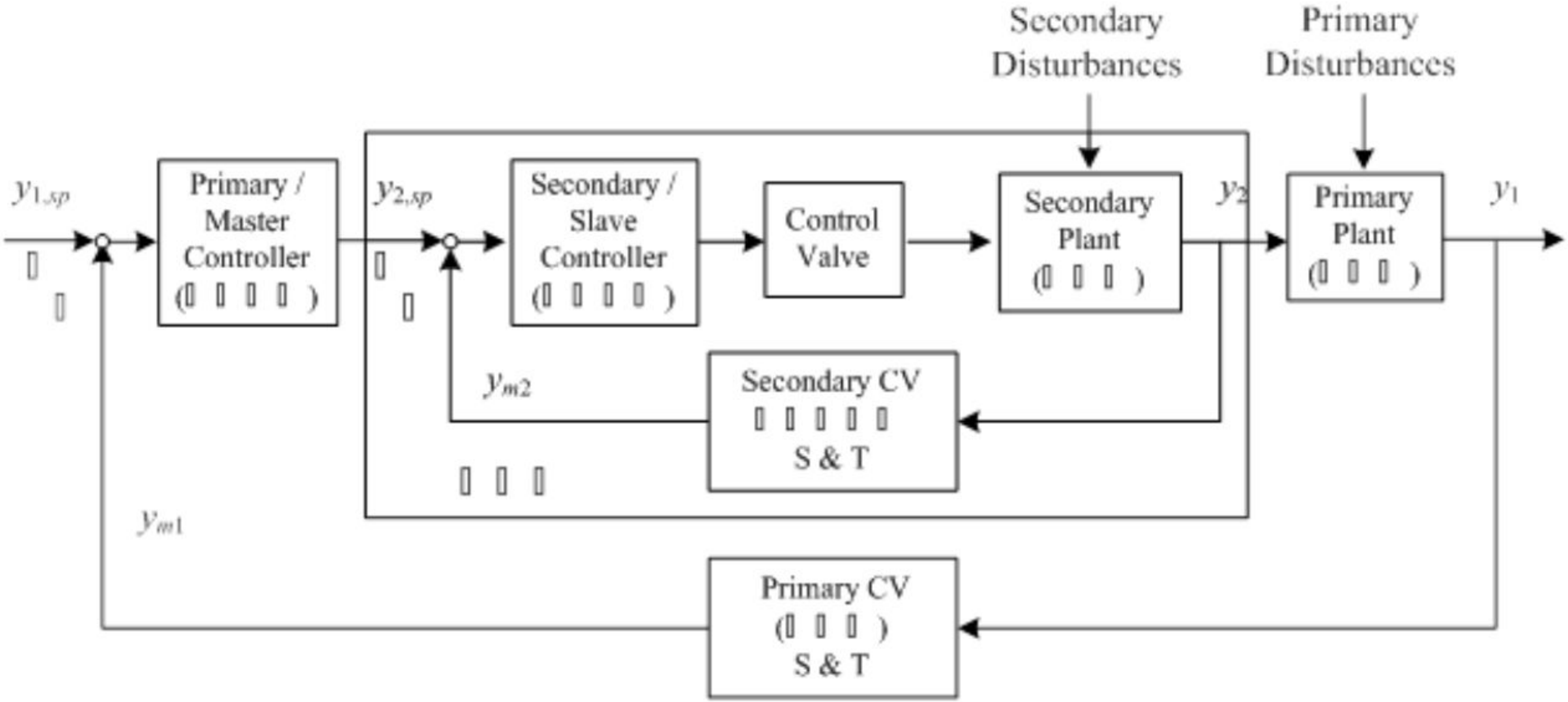
PID

Benefits of PID (here):

- Fast rise time
- Low settling time
- No overshoot



Cascaded Control



Cascaded Control Requirements

Each stage needs to be “faster” than the previous one

- For first order systems, this is just the time constant
- For second order systems, this is the natural frequency

Cascaded Control Example: Quadcopter

Desired: Position Control

- Position Controller
- Velocity Controller
- Acceleration Controller
- Attitude Controller
- Angular Velocity Controller
- Angular Acceleration Controller
- Motor Controller

Each is its own PID (or lower order) controller

Model-Free vs Model-Based Control

PID (and first/second order controllers) are *model-free*

Use *model-based* control when possible

Feed-forward control

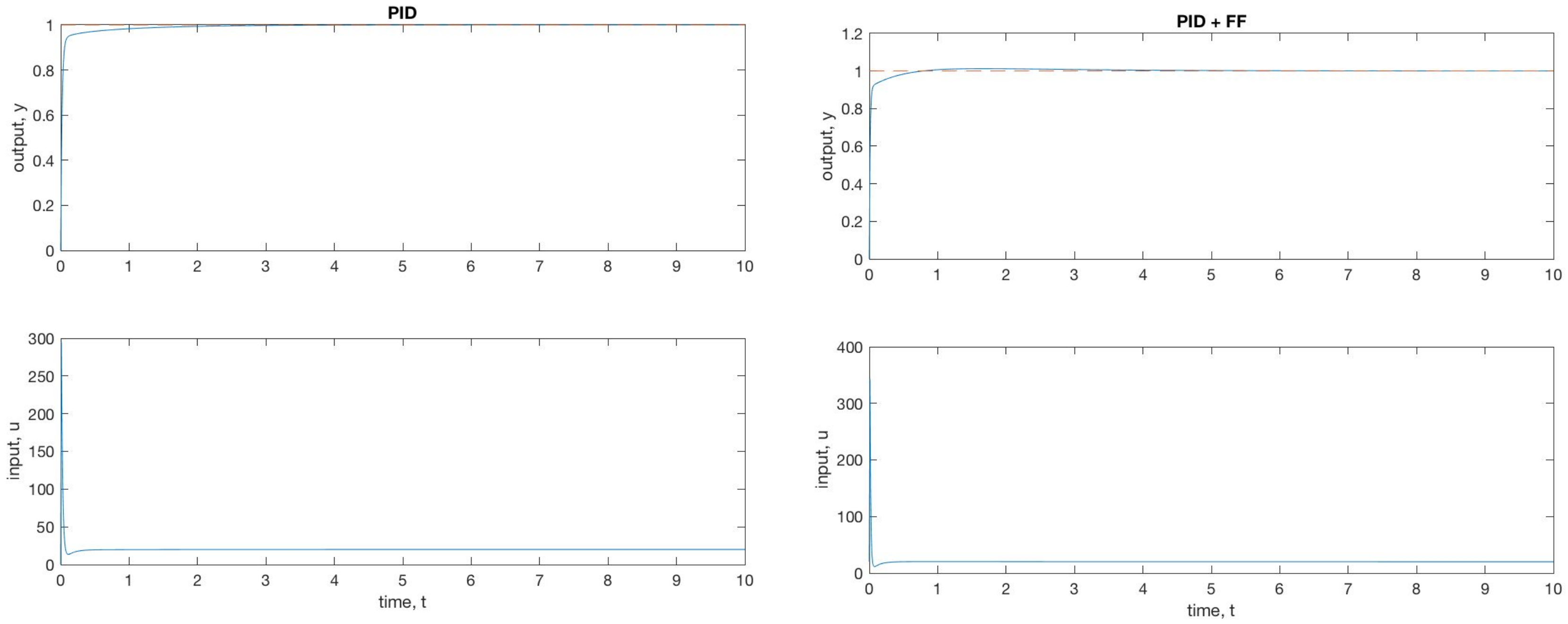
Feed-forward or “ballistic term” does not depend upon error

$$u = K_{ff} + K_{fb}e$$

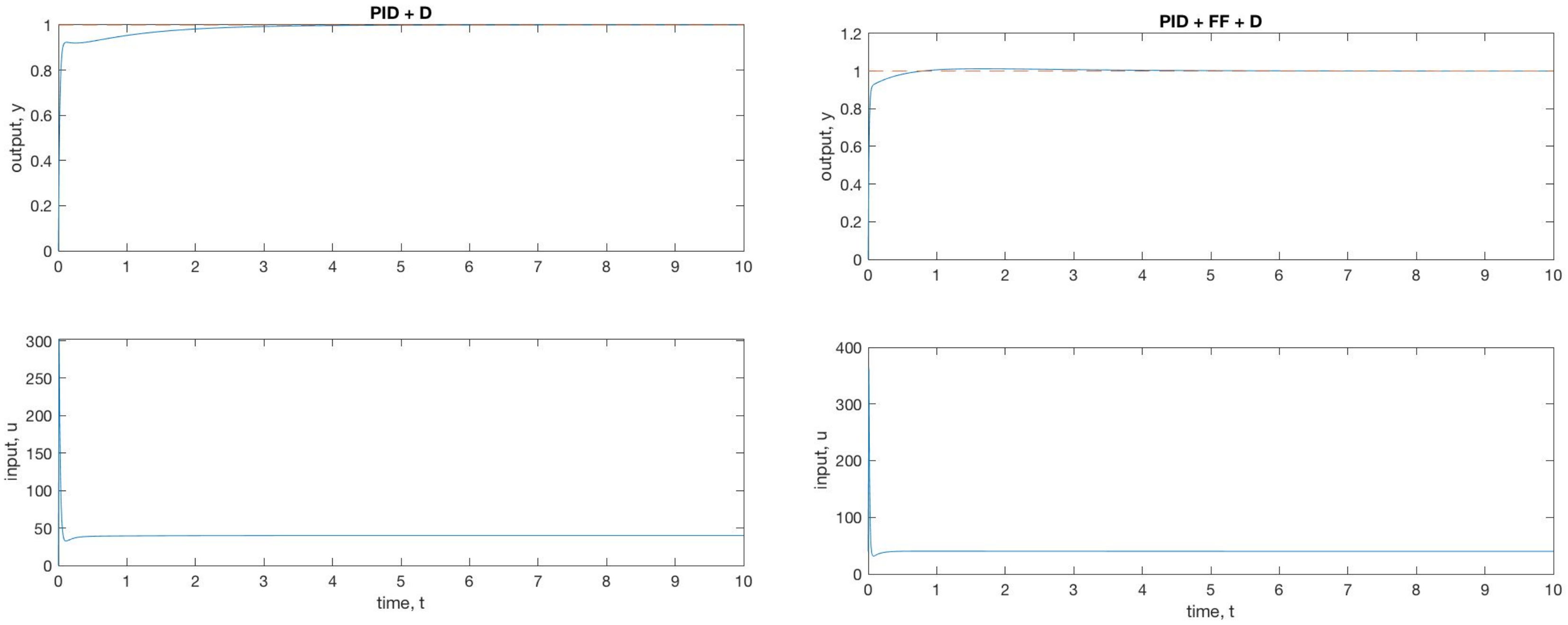
Feed-forward term should do a bulk of your work, with the feedback doing small corrections.

Feed-forward control doesn’t oscillate, and has no stability problems, so it improves system response without decreasing stability.

Feed-forward Control Example



Feed-forward can Counteract Known Disturbance



State Space Realization

$$\Sigma \quad \left\{ \begin{array}{rcl} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) + Du(t) \end{array} \right.$$

State Space Realization: Example

$$y^{[3]} + 3\ddot{y} + 4\dot{y} + 7y = u$$

$$x_1 = y, x_2 = \dot{y}, x_3 = \ddot{y}$$

$$\left[\begin{array}{c} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \hline y \end{array}\right] = \left[\begin{array}{ccc|c} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -7 & -4 & -3 & 1 \\ \hline 1 & 0 & 0 & 0 \end{array}\right] \left[\begin{array}{c} x_1 \\ x_2 \\ x_3 \\ \hline u \end{array}\right]$$

State Feedback Control

Define a controller

$$u = -Kx$$

Since u now depends on x, we can now define

$$A_{cl} = A - BK$$

Since there are n values in K, we can actually control all n eigenvalues.

LQR - Linear Quadratic Regulator

The “optimal” way to find K

Define cost matrices Q, R

$$Q \in \mathbb{R}^{n \times n}, \;\; R \in \mathbb{R}^{u \times u}, \;\; Q, R \succ 0$$

LQR finds a K which minimizes the cost function

$$\int_0^\infty (x^T Q x + u^T R u + 2x^T N u) dt$$

Use command “lqr” in Matlab