PID Controls

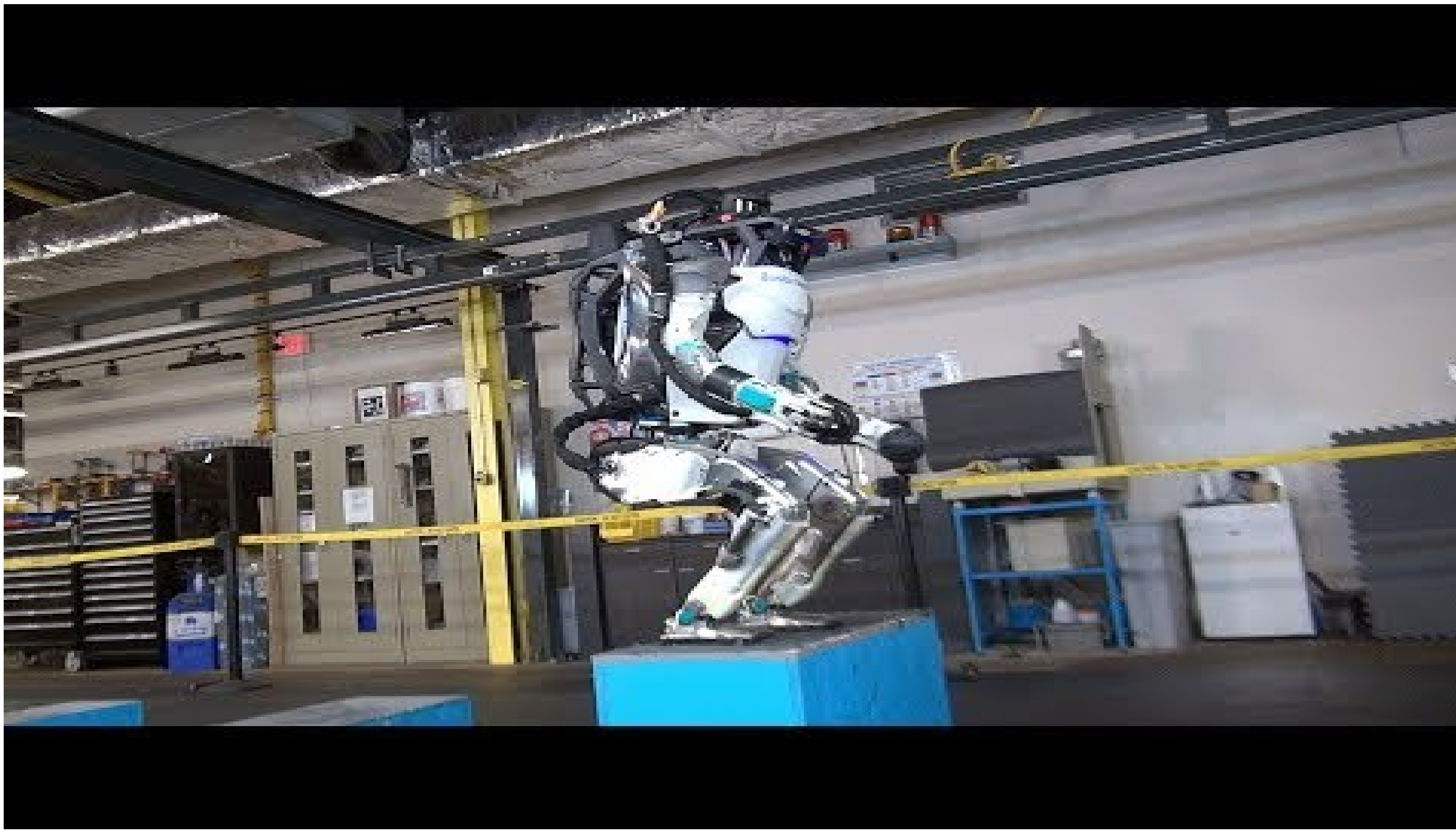
Howie Choset

(thanks to George Kantor, Nathan Michael and Wikipedia)

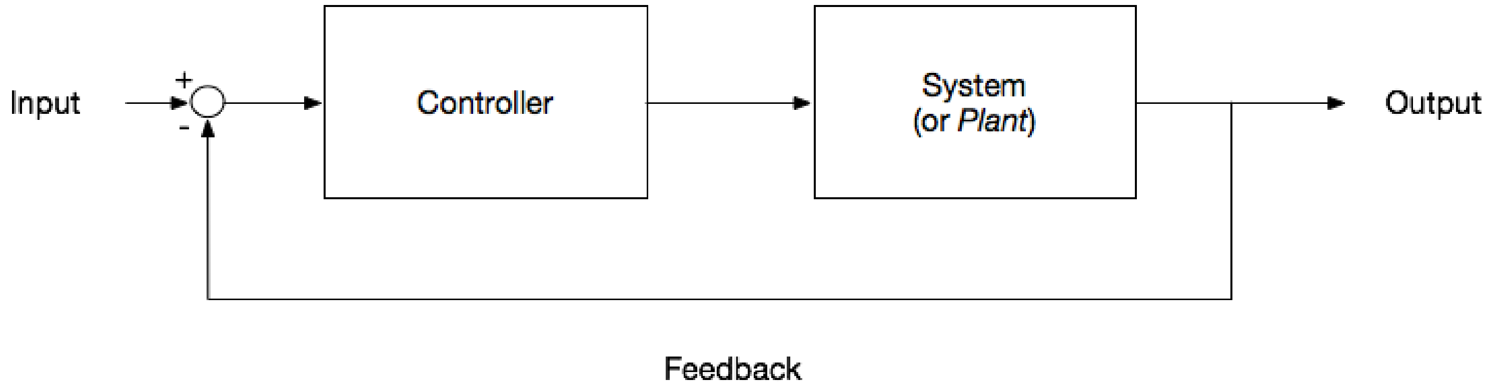
http://www.library.cmu.edu/ctms/ctms/examples/motor/motor.htm

Boston Dynamics

Boston Dynamics

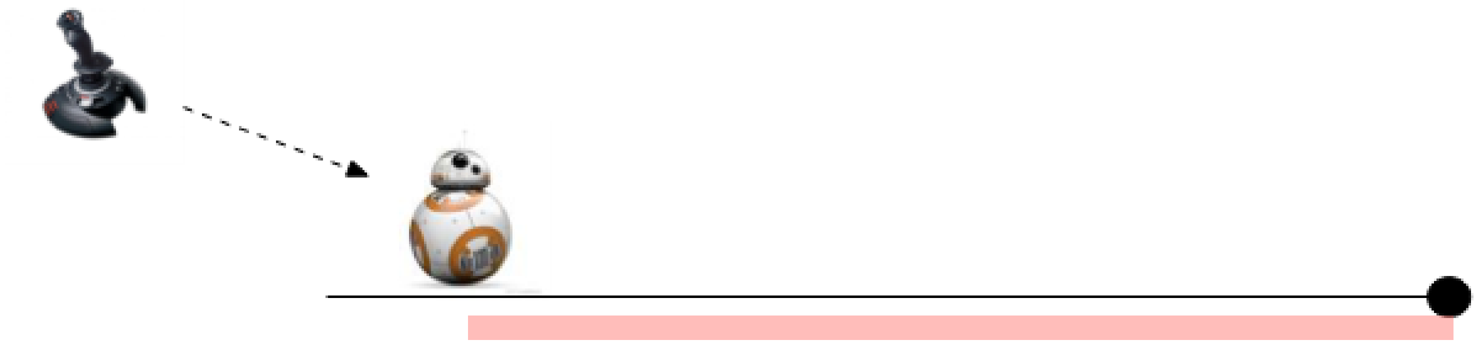


What is a feedback control system?



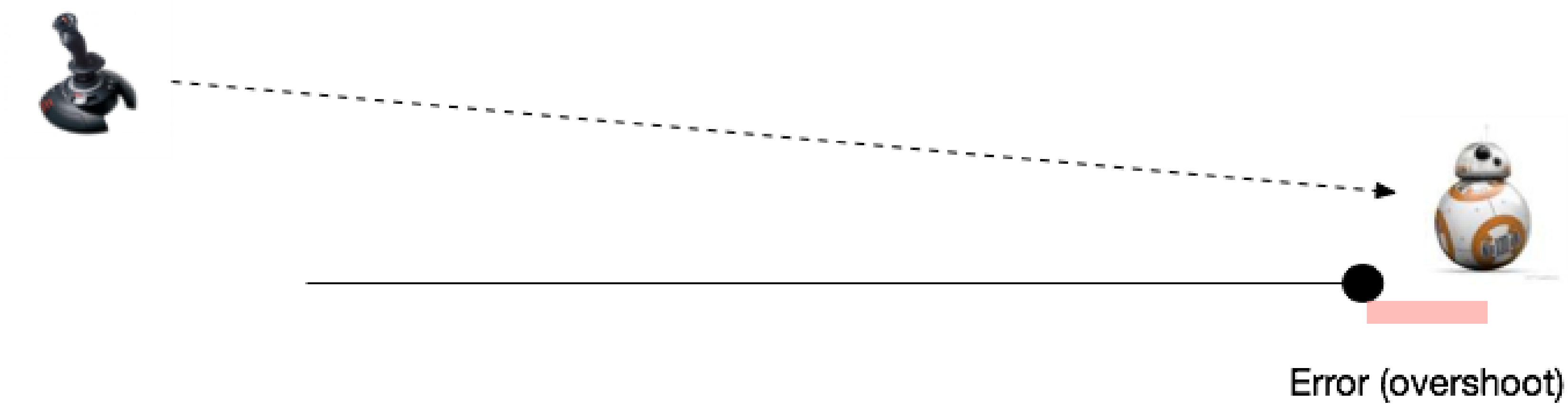
Feedback Control System Example

Feedback Control System Example

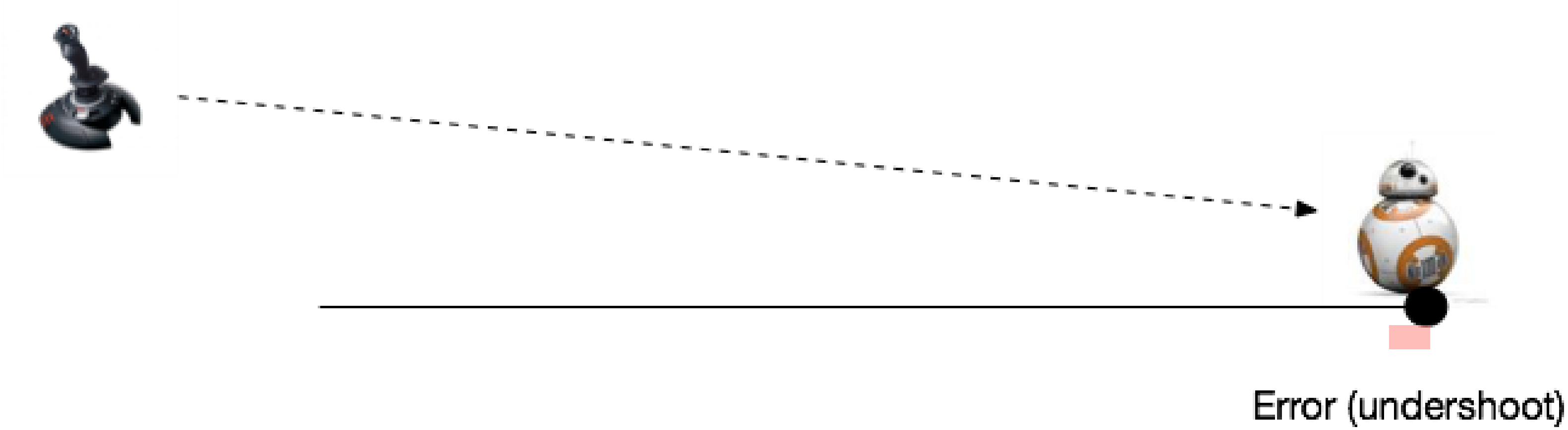


Error between current position and desired position (or goal)

Feedback Control System Example

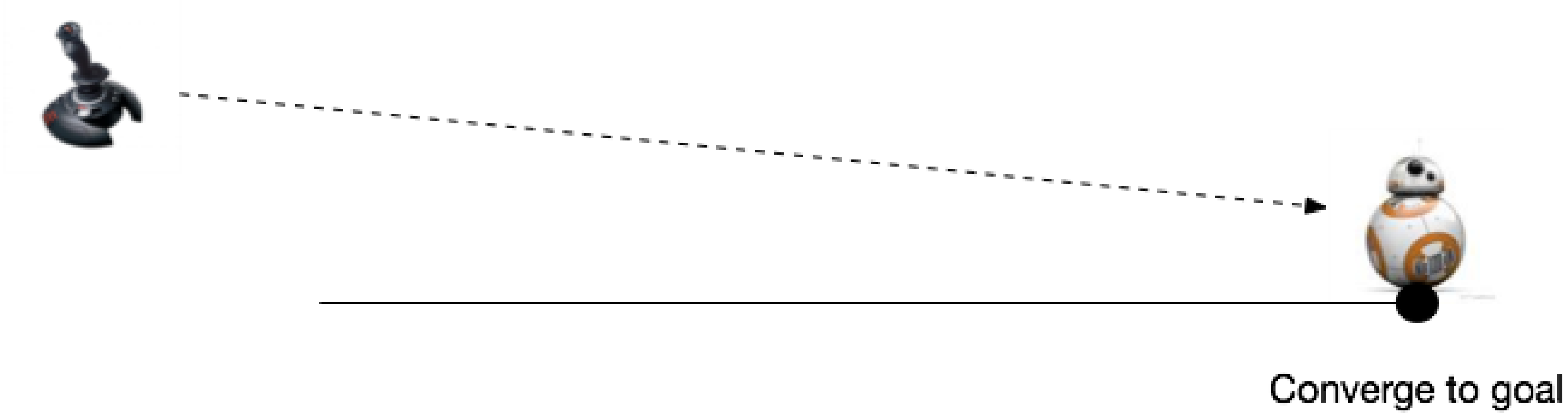


Feedback Control System Example

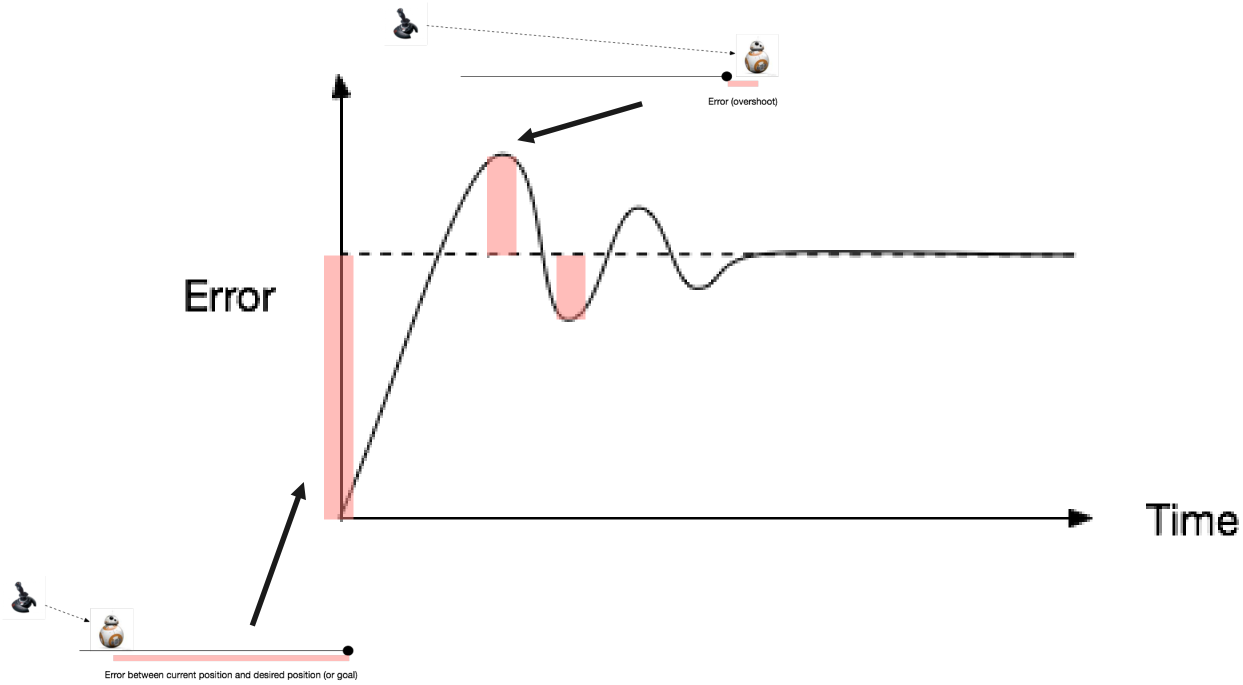


CarnegieMellon
THE ROBOTICS INSTITUTE

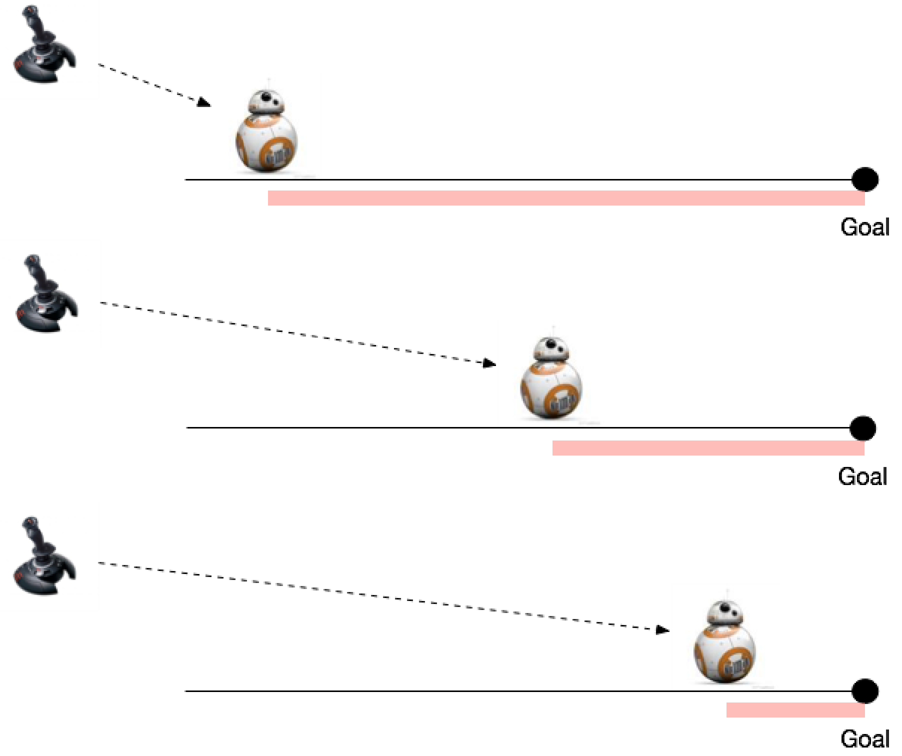
Feedback Control System Example



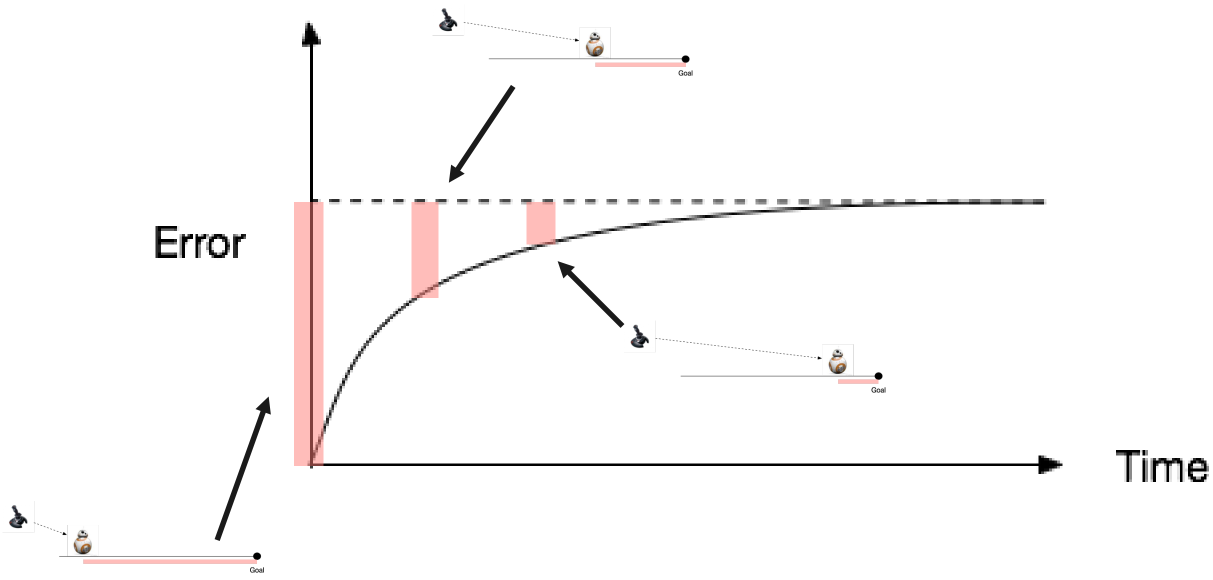
Feedback Control System Example



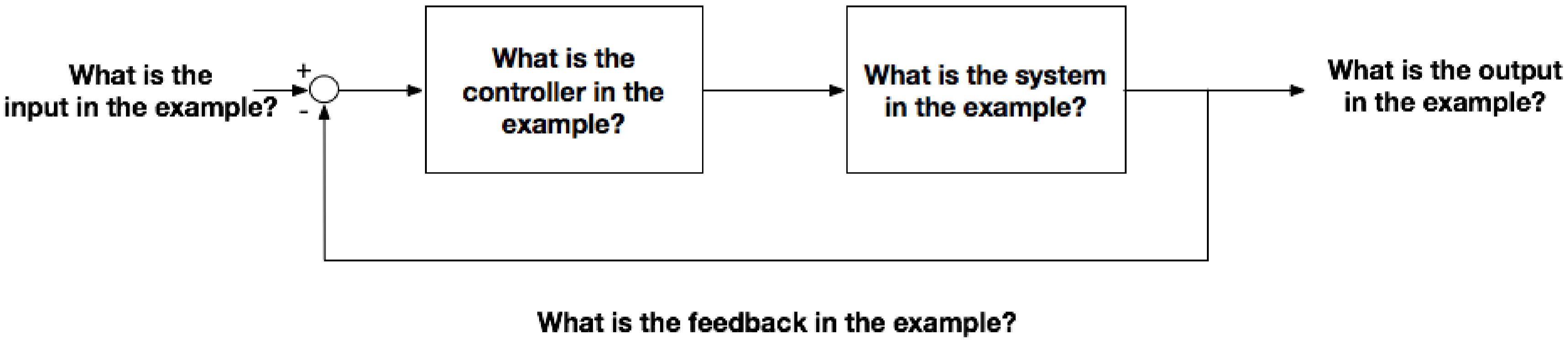
Feedback Control System Example



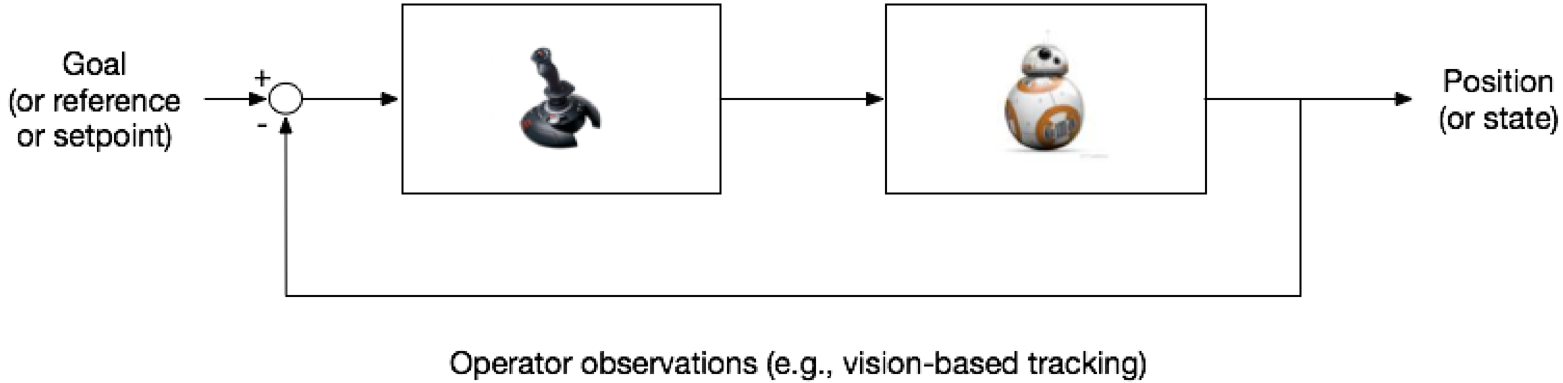
Feedback Control System Example



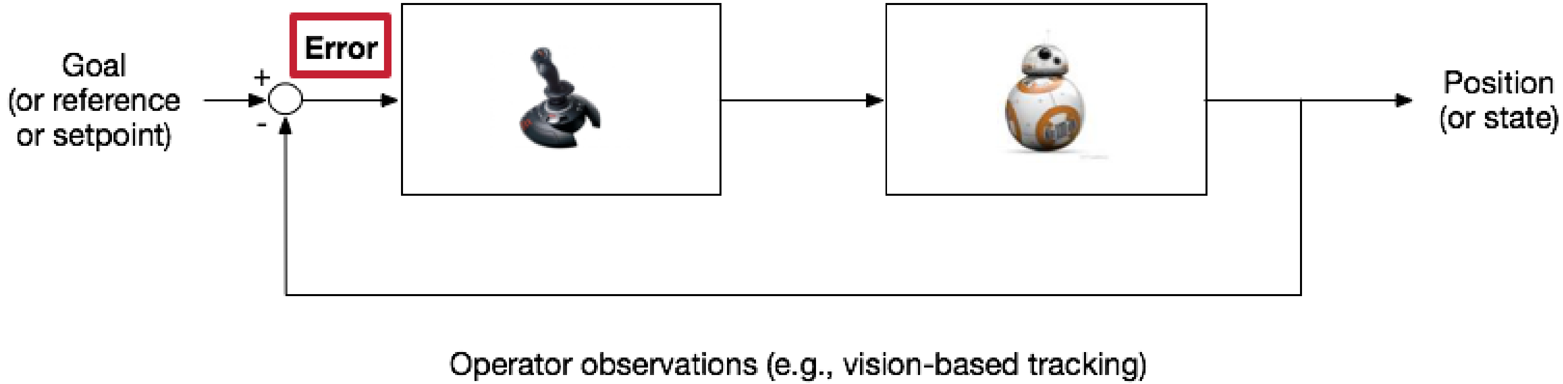
Feedback Control System Example



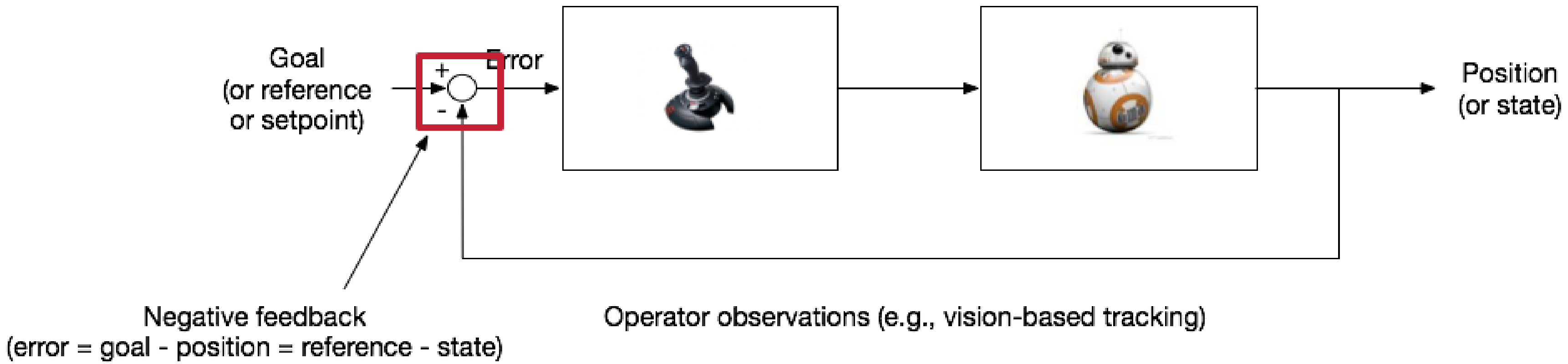
Feedback Control System Example



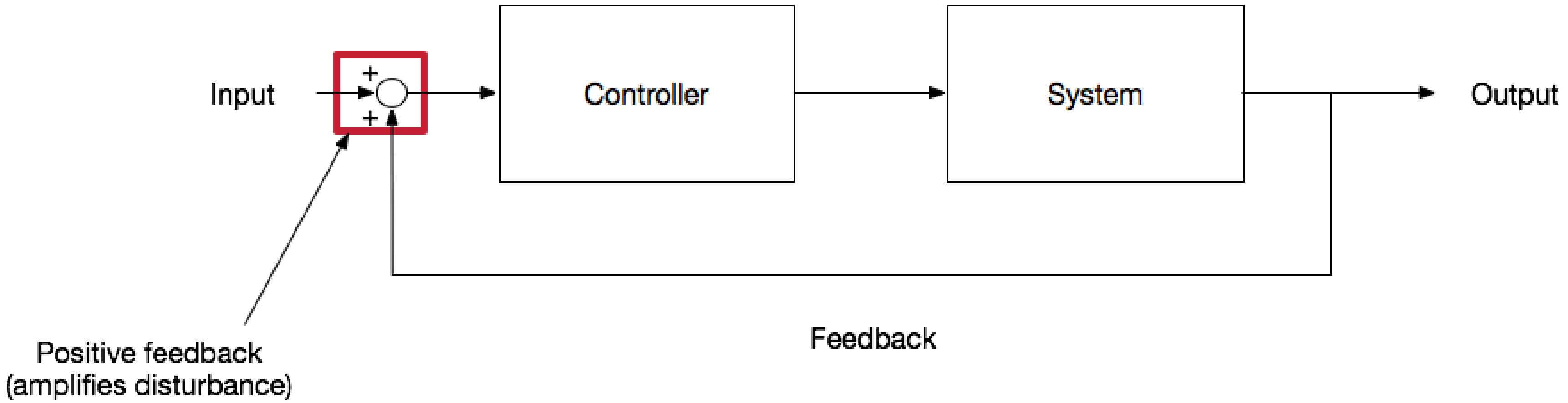
Feedback Control System Example



Feedback Control System Example



Feedback Control System Example



Force Balance

$$F = \frac{d}{dt}(m\dot{x}) = m\ddot{x}$$

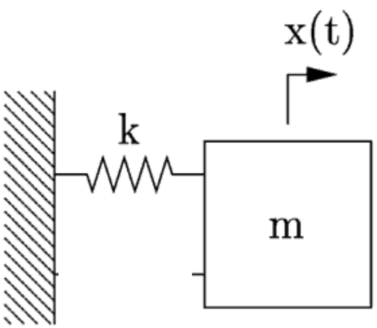
Force = Mass x acceleration

Force Balance

$$F = \frac{d}{dt}(m\dot{x}) = m\ddot{x}$$

Force = Mass x acceleration

.



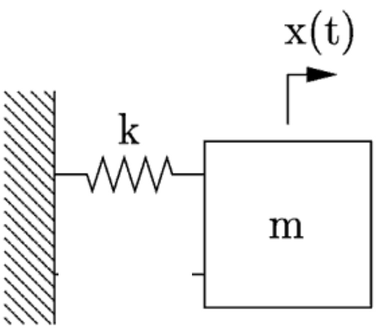
$$F = -kx$$

Force = spring constant x displacement

Force Balance

$$F = \frac{d}{dt}(m\dot{x}) = m\ddot{x}$$

Force = Mass x acceleration



$$F = -kx$$

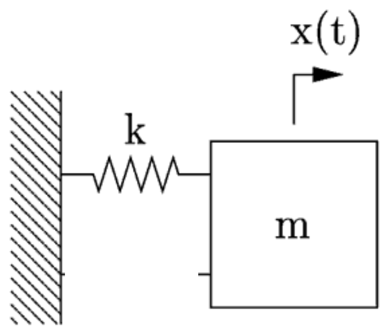
Force = spring constant x displacement

Force Balance

$$F = \frac{d}{dt}(m\dot{x}) = m\ddot{x}$$

Force = Mass x acceleration

$$\ddot{x} + \frac{k}{m}x = 0$$



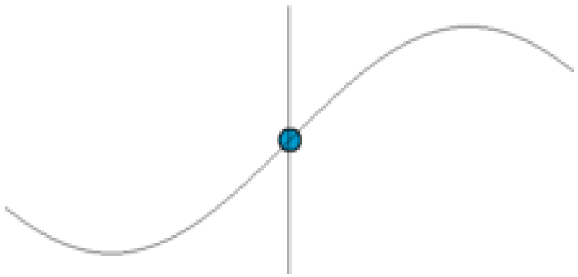
$$F = -kx$$

Force = spring constant x displacement

Standard Form of DEQ

$$\ddot{x} + \frac{k}{m}x = 0$$

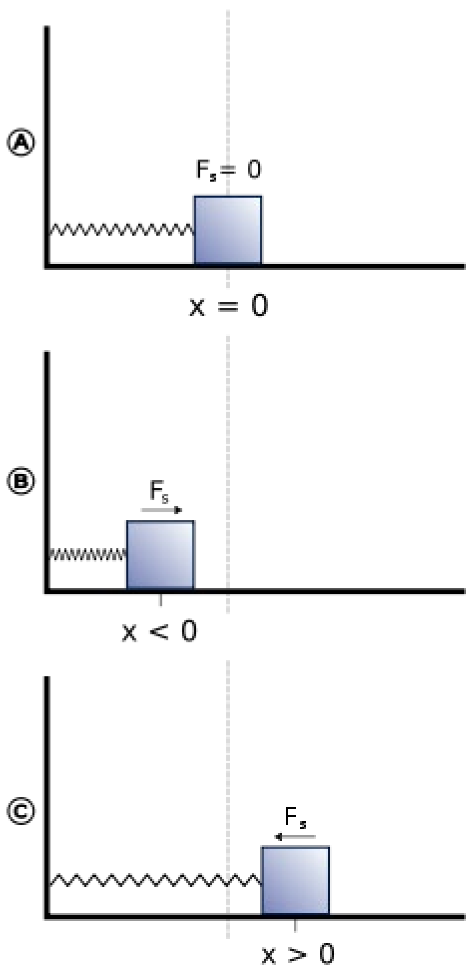
$$\omega_0 = \sqrt{\frac{k}{m}}$$ Natural frequency (rads/sec)

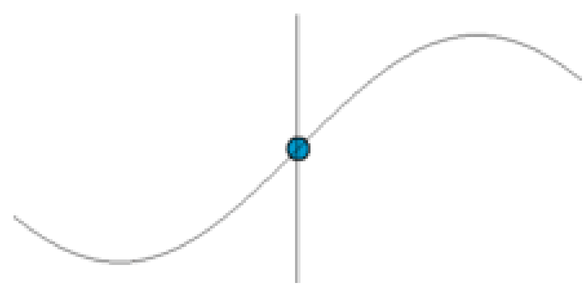


$$\ddot{x} + \omega_0^2 x = 0.$$

$$x(t) = (A\cos(\omega_0 t) + B\sin(\omega_0 t))$$

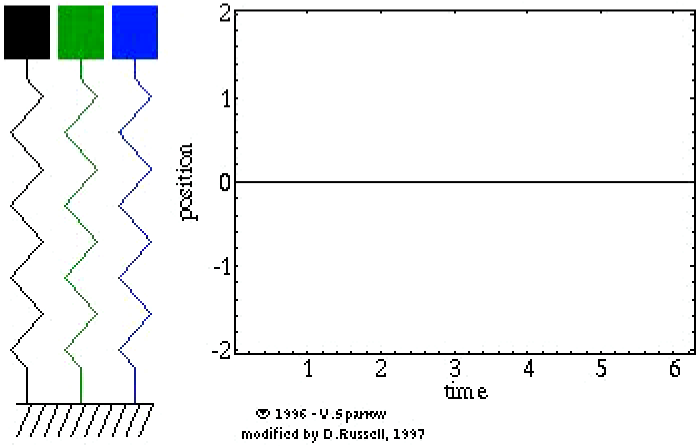
Standard Form of DEQ





$$x(t) = \left(A\cos(\omega_0 t) + B\sin(\omega_0 t)\right)$$

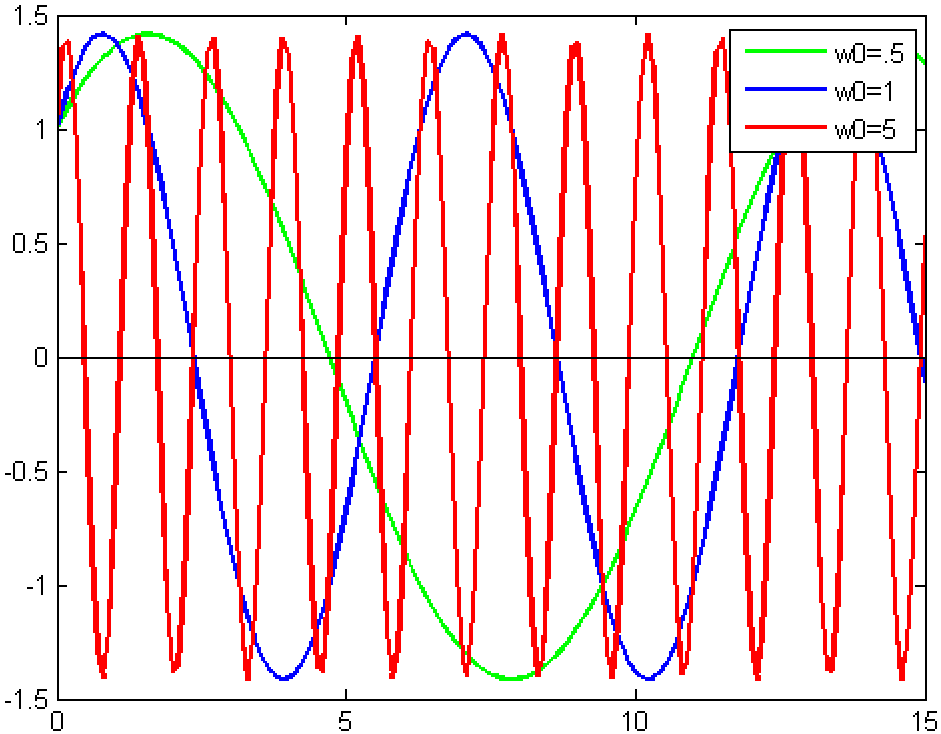
Vary Natural Frequency



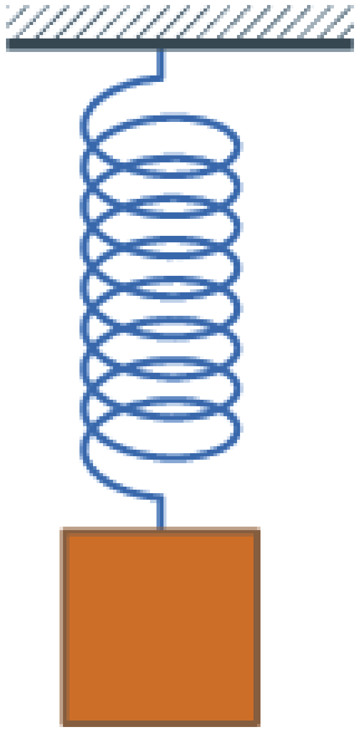
Different Frequencies

$$\boxed{x(t) = A\cos(\omega_0 t) + B\sin(\omega_0 t)}$$

$$\omega_0 = \sqrt{\frac{k}{m}}$$



Can we go forever

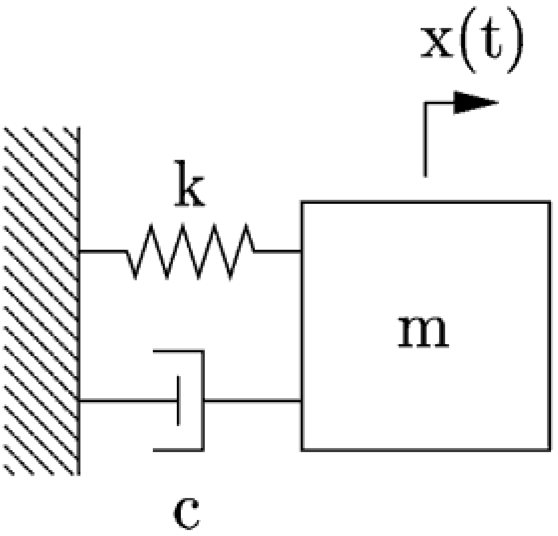


Mass Spring Damper

$$F_s = -kx$$

$$F_d = -cv = -c\frac{dx}{dt} = -c\dot{x}$$

$$F_{tot} = ma = m\frac{d^2x}{dt^2} = m\ddot{x}$$



$$m\ddot{x} = -kx - c\dot{x} \quad \Rightarrow \quad \ddot{x} + \frac{c}{m}\dot{x} + \frac{k}{m}x = 0$$

2nd Order ODE

$$\ddot{x} + \frac{c}{m}\dot{x} + \frac{k}{m}x = 0$$

$$\omega_0 = \sqrt{\frac{k}{m}}$$

Natural (undamped) frequency (rads/sec)

$$\zeta = \frac{c}{2\sqrt{mk}}$$

Damping ratio (unitless)

$$\ddot{x} + 2\zeta\omega_0\dot{x} + \omega_0^2 x = 0$$

2nd Order ODE Solutions

$$\ddot{x} + 2\zeta\omega_0\dot{x} + \omega_0^2 x = 0$$

Recall: ω_0 Natural (undamped) frequency

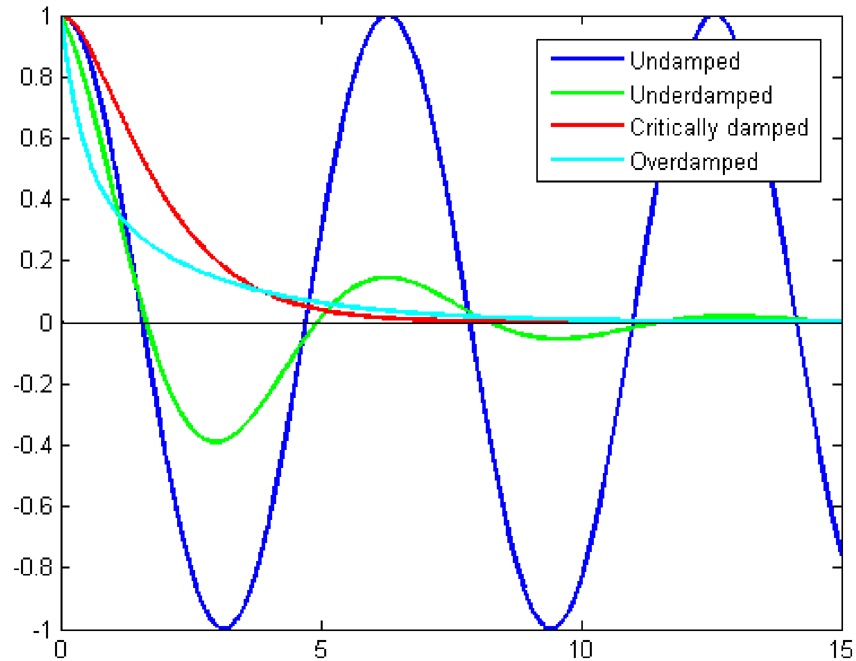
ζ Damping ratio

Solutions:

Critically damped ($\zeta = 1$)

Overdamped ($\zeta > 1$)

Underdamped ($\zeta < 1$)



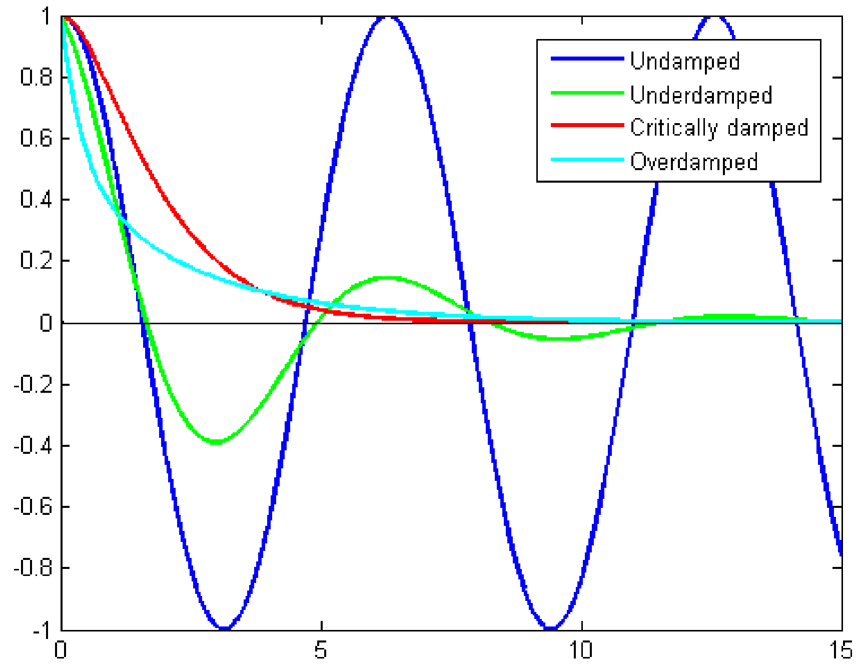
2nd Order ODE Solutions

$$\ddot{x} + 2\zeta\omega_0\dot{x} + \omega_0^2 x = 0$$

Recall: ω_0 Natural (undamped) frequency

ζ Damping ratio

Solutions:



Underdamped ($\zeta < 1$)

$$x(t) = \underbrace{e^{-\zeta\omega_0 t}}_{\text{Decay}}\underbrace{(A\cos(\omega_d t) + B\sin(\omega_d t))}_{\text{Oscillation, damped natural frequency}}$$

$$\omega_d = \omega_0\sqrt{1-\zeta^2}$$

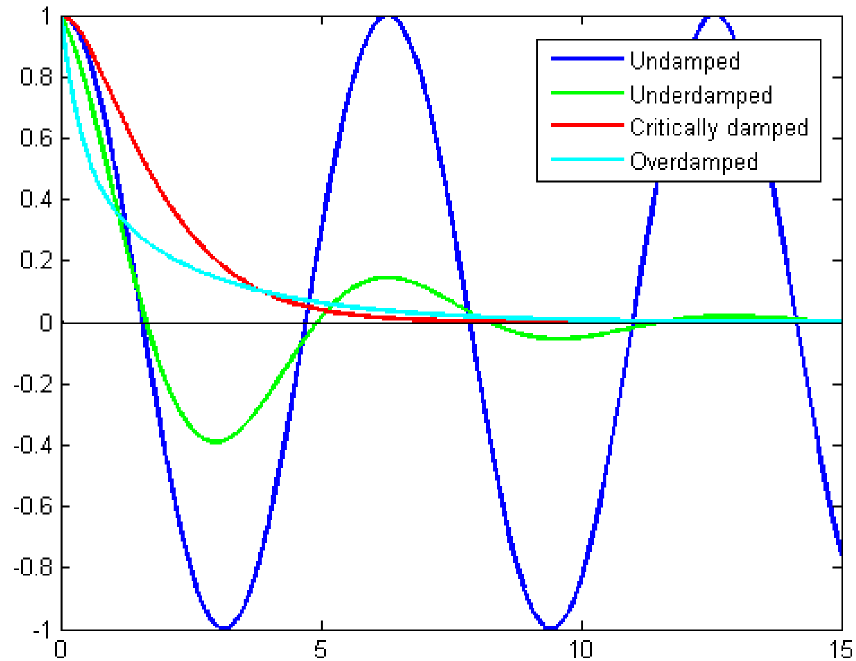
2^{nd} Order ODE Solutions

$$\ddot{x} + 2\zeta\omega_0\dot{x} + \omega_0^2 x = 0$$

Recall: ω_0 Natural (undamped) frequency

ζ Damping ratio

Solutions:



Overdamped $(\zeta > 1)$

$$x(t) = Ae^{\gamma_+ t} + Be^{\gamma_- t}$$

$$\gamma_\pm = \omega_0\left(-\zeta \pm \sqrt{\zeta^2 - 1}\right)$$

Underdamped $(\zeta < 1)$

$$x(t) = \underbrace{e^{-\zeta\omega_0 t}}_{\text{Decay}}\underbrace{(A\cos(\omega_d t) + B}_{\text{Oscillation, damped natural frequency}}\sin(\omega_d t))$$

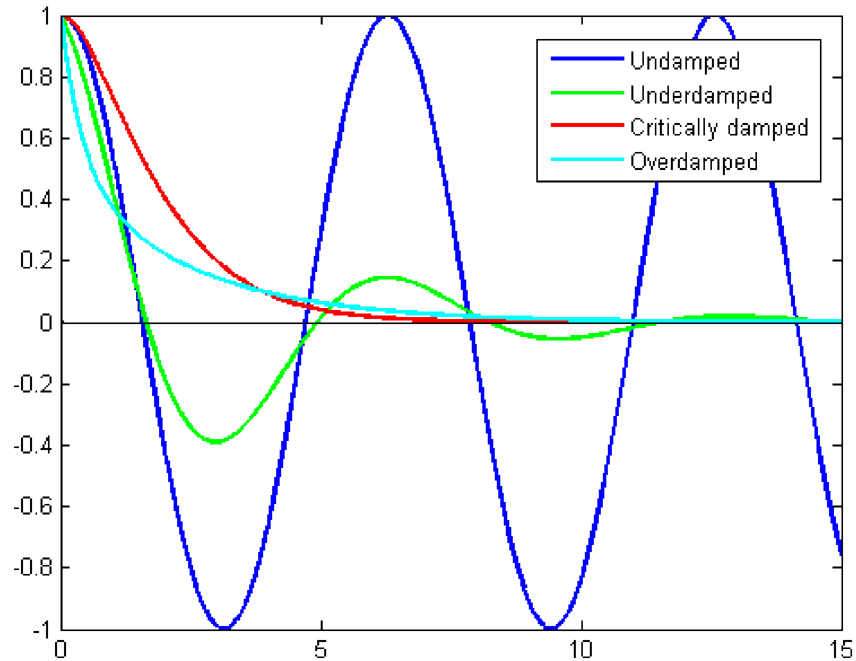
$$\omega_d = \omega_0\sqrt{1 - \zeta^2}$$

2^{nd} Order ODE Solutions

$$\ddot{x} + 2\zeta\omega_0\dot{x} + \omega_0^2 x = 0$$

Recall: ω_0 Natural (undamped) frequency

ζ Damping ratio



Solutions:

Critically damped ($\zeta = 1$)

$$x(t) = (A + Bt)e^{-\omega_0 t}$$

Overdamped ($\zeta > 1$)

$$x(t) = Ae^{\gamma_+ t} + Be^{\gamma_- t}$$

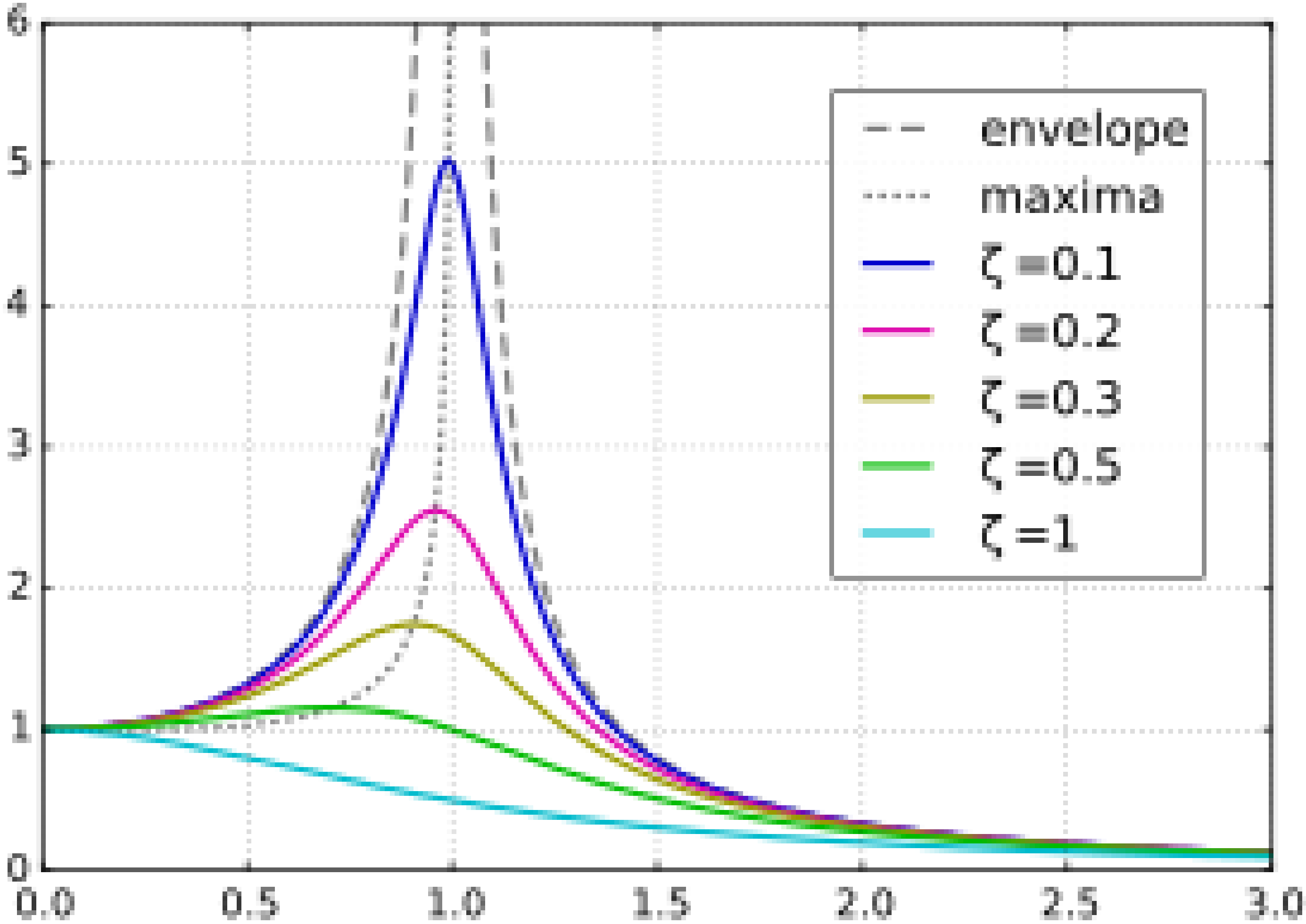
$$\gamma_\pm = \omega_0\left(-\zeta \pm \sqrt{\zeta^2 - 1}\right)$$

Underdamped ($\zeta < 1$)

$$x(t) = \underbrace{e^{-\zeta\omega_0 t}}_{\text{Decay}}\underbrace{(A\cos(\omega_d t) + B\sin(\omega_d t))}_{\text{Oscillation, damped natural frequency}}$$

$$\omega_d = \omega_0\sqrt{1 - \zeta^2}$$

Resonance



Step Response

$$\omega_d = \omega_0 \sqrt{1-\zeta^2}$$

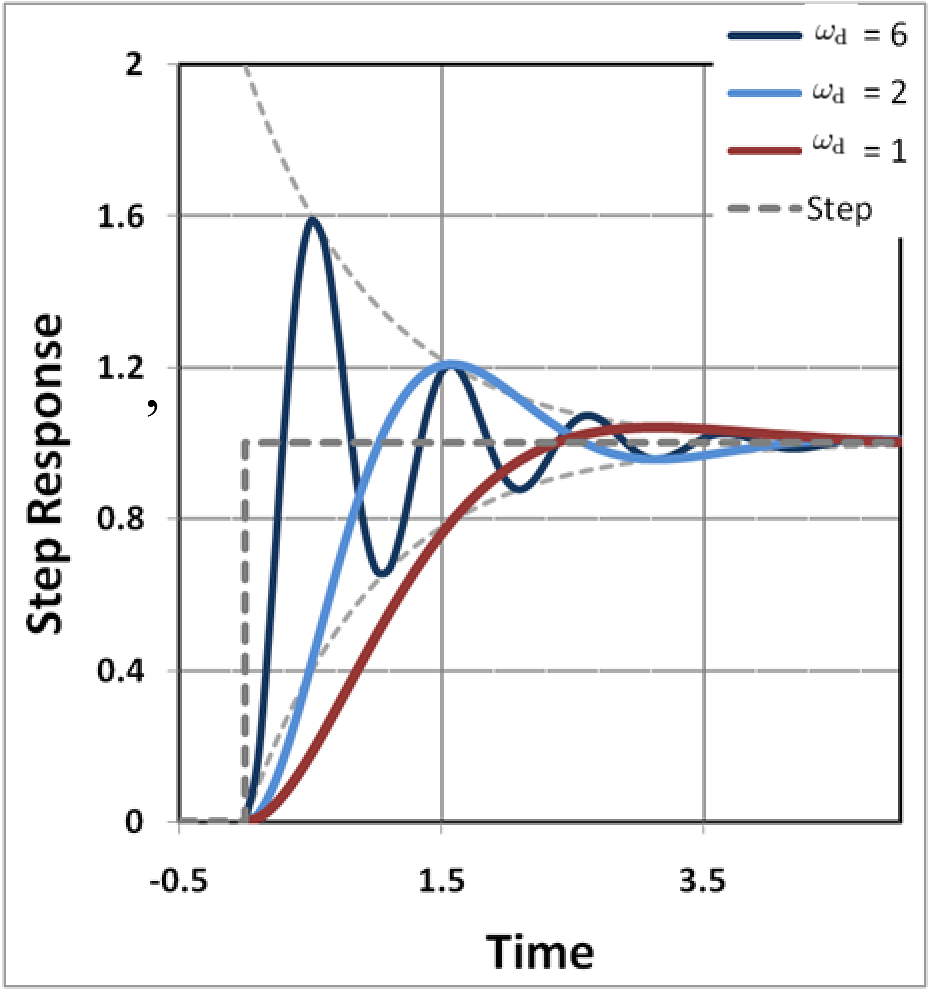
$$\frac{d^2x}{dt^2} + 2\zeta\omega_0 \frac{dx}{dt} + \omega_0^2 x = \frac{F(t)}{m}$$

$$\frac{F(t)}{m} = \begin{cases} \omega_0^2 & t \geq 0 \\ 0 & t < 0 \end{cases}$$

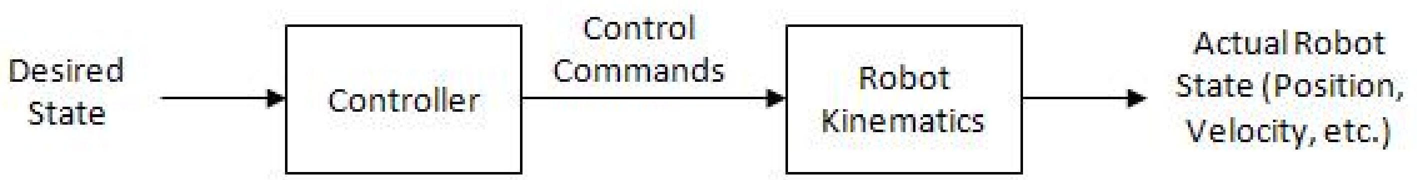
$$x(t) = 1 - e^{-\zeta\omega_0 t} \frac{\sin\left(\sqrt{1-\zeta^2}\,\omega_0 t + \varphi\right)}{\sin(\varphi)}$$

$$\cos\varphi = \zeta$$

As time goes on, x(t) goes to 1

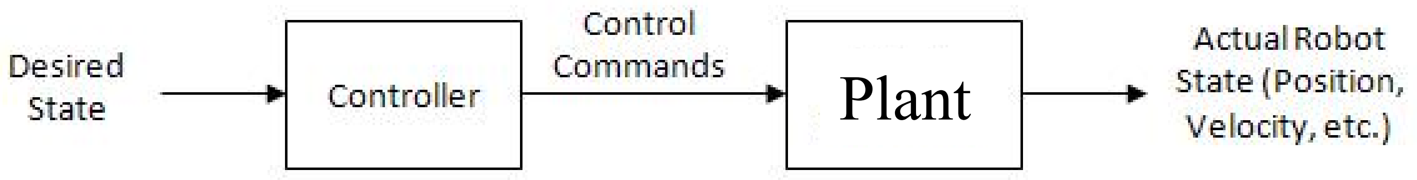


Open Loop Controller



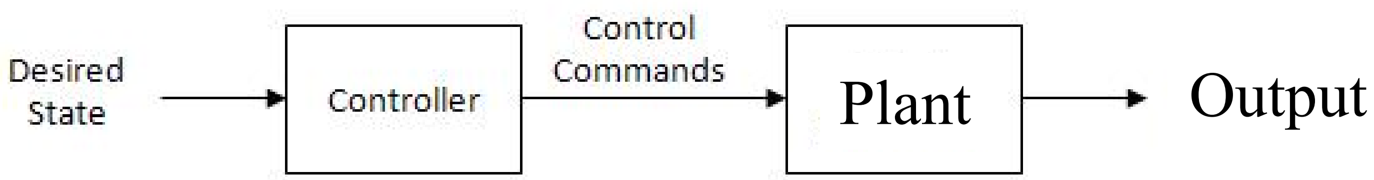
controller tells your system to do something, but doesn’t use the results of that action to verify the results or modify the commands to see that the job is done properly

Open Loop Controller



controller tells your system to do something, but doesn't use the results of that action to verify the results or modify the commands to see that the job is done properly

Open Loop Controller



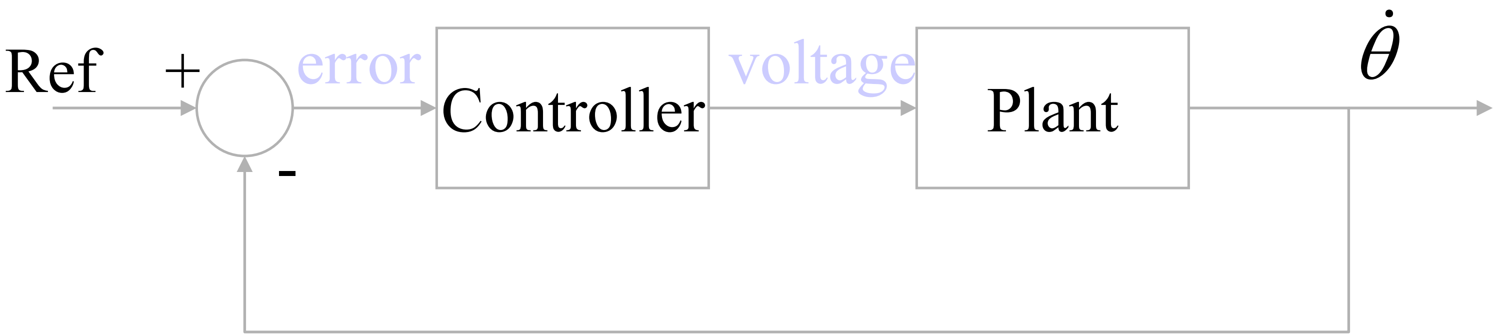
controller tells your system to do something, but doesn't use the results of that action to verify the results or modify the commands to see that the job is done properly

Closed Loop Controller

Give it a velocity command and get a velocity output

Controller Evaluation

Steady State Error
Rise Time (to get to ~90%)
Overshoot
Settling Time (Ring) (time to steady state)
Stability



PID Feedback

$$\frac{d^2x}{dt^2} + 2\zeta\omega_0\frac{dx}{dt} + \omega_0^2 x = \cancel{F(t)} \quad -u(t)$$

$$u(t) = K_P e(t) + K_I \int e(t)\mathrm{d}t + K_D \frac{\mathrm{d}}{\mathrm{d}t} e(t)$$

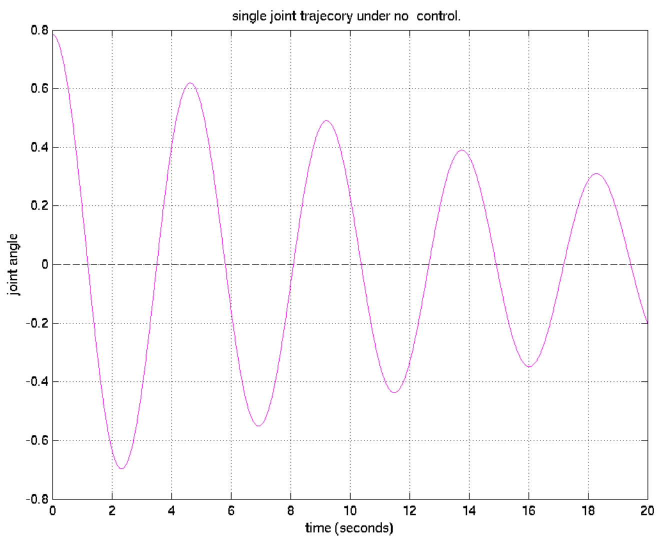
P Feedback

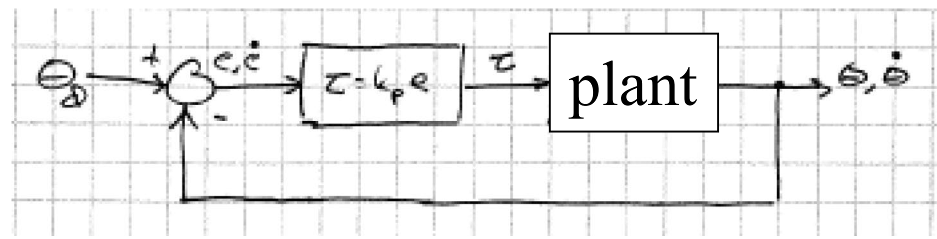
$$\frac{d^2x}{dt^2} + 2\zeta\omega_0\frac{dx}{dt} + \omega_0^2 x = \cancel{F(t)} \; u(t) \longleftarrow -Kx$$

$$\ddot{x} + 2\zeta\omega_0\dot{x} + (\omega_0^2 + K)\, x = 0$$

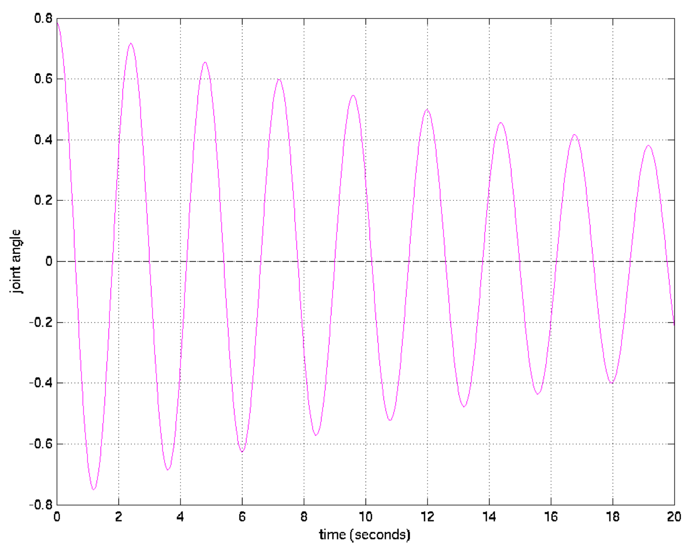
It is like changing the spring constant

Proportional Feedback



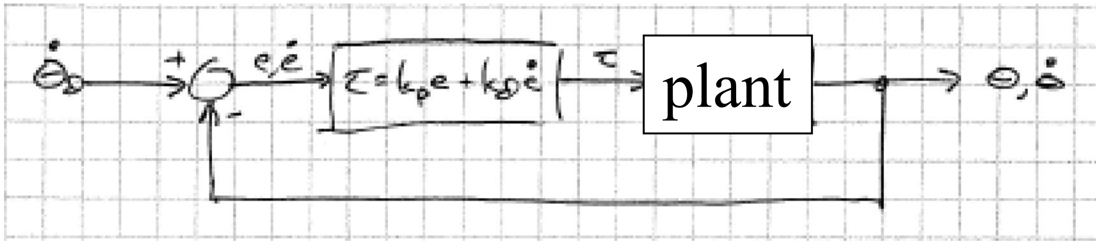


Set desired position to zero

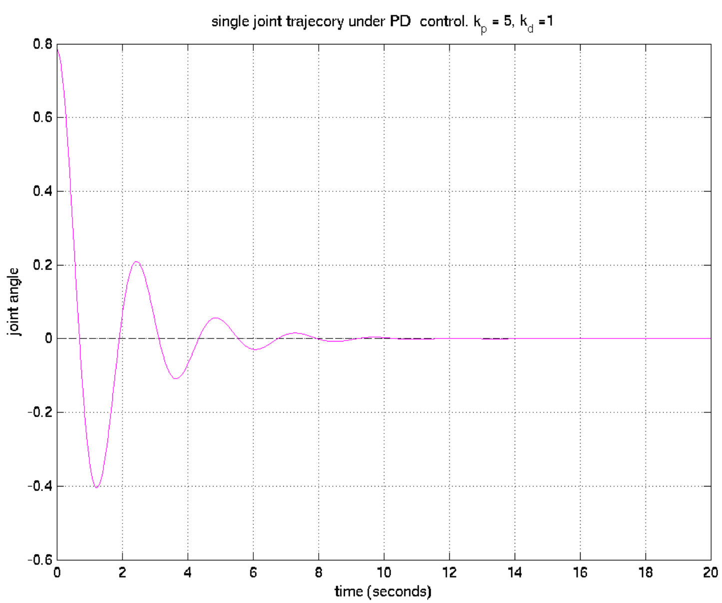


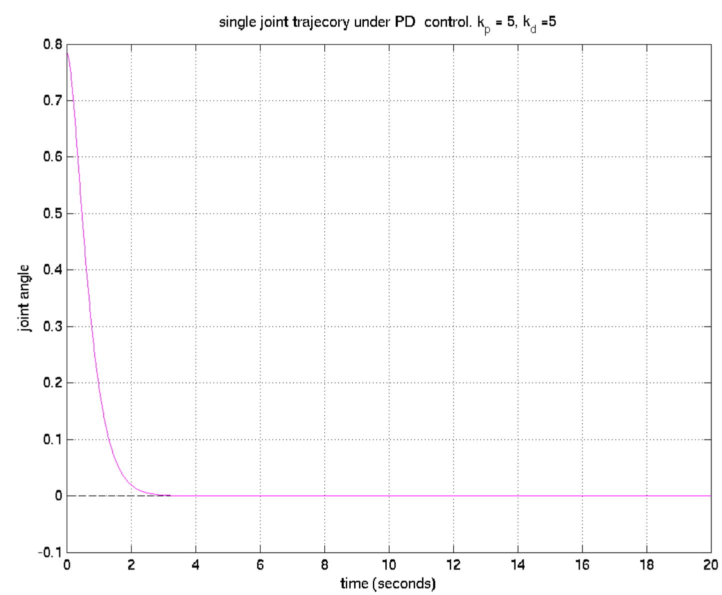
Note that the oscillation dies out at approximately the same rate but has higher frequency. This can be thought of as “stiffening the spring”.

Proportional/Damping



We can increase the damping (i.e., increase the rate at which the oscillation dies out)

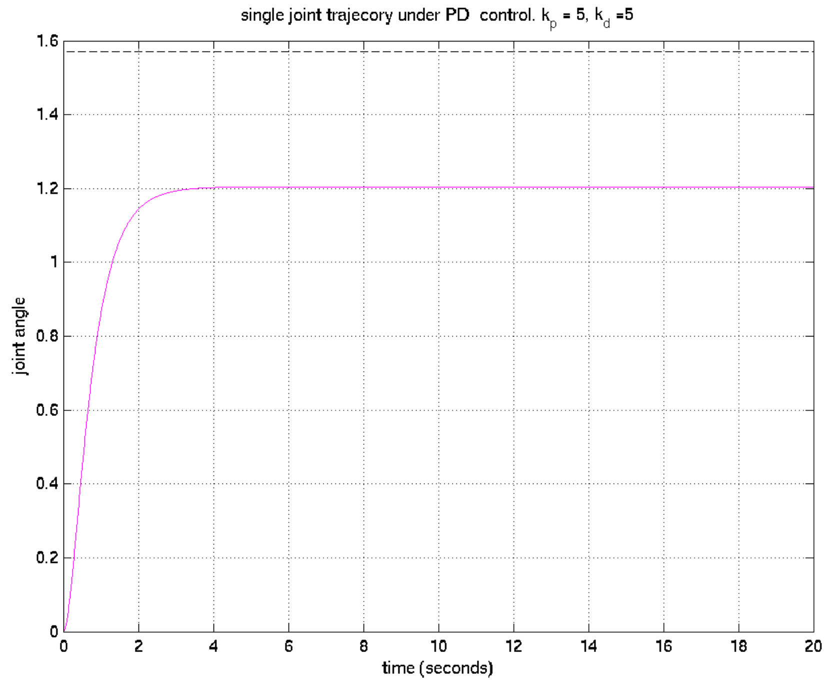




Increasing damping slows everything down (note deriv is an approx and turning the gain high, can cause problems because in a sense it amplifies no

PD works well if desired point is an equilibrium of system, which makes sense because when you are at target, PD does not exert force

Non-zero desired PD



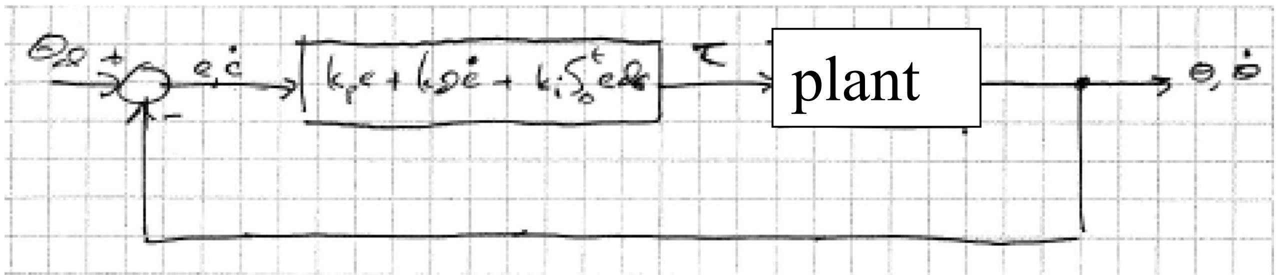
$X_d = 1.6$

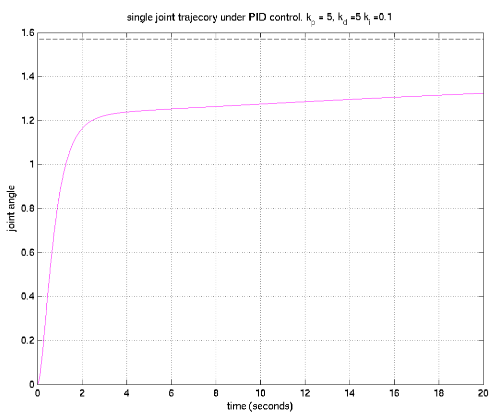
Settle time same

Steady state error!

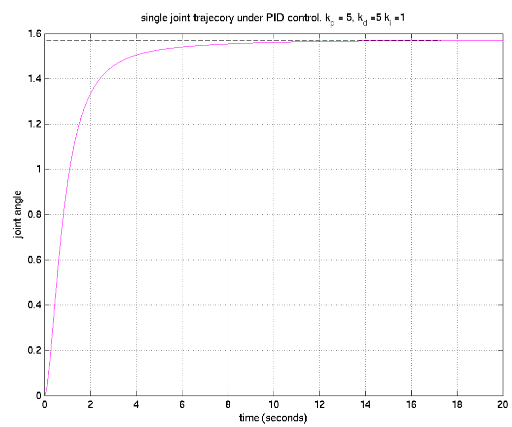
At set point, applying no force so end up settling at equilibrium that balances force due to error and force due to spring (damper goes away in steady state because depends on derivative). Crank up P gain, steady state error gets smaller, but that causes overshoot, oscillations, etc which you don't want

PID Control

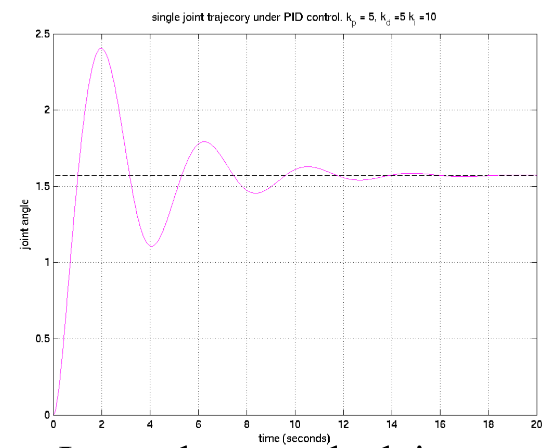




System does its dynamic thing and then gradually integrates to correct for steady state error

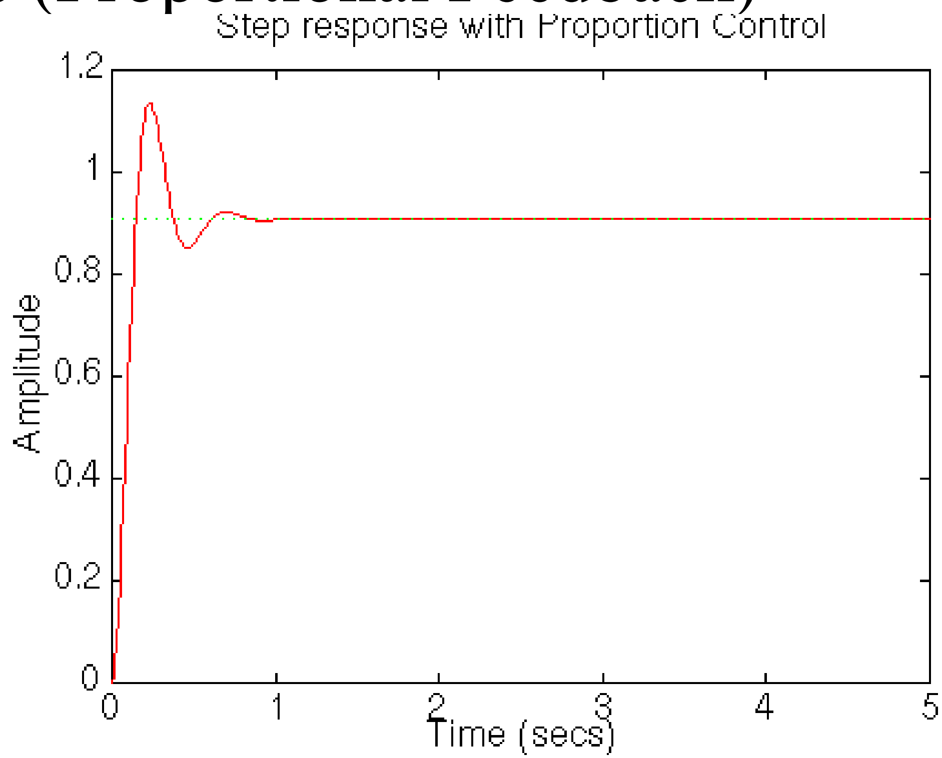


As increase I gain, gets faster, good response



Integral gets so bad, it starts to interfere with other dynamics, lead to unintended motions which could lead to instability

Closed Loop Response (Proportional Feedback)

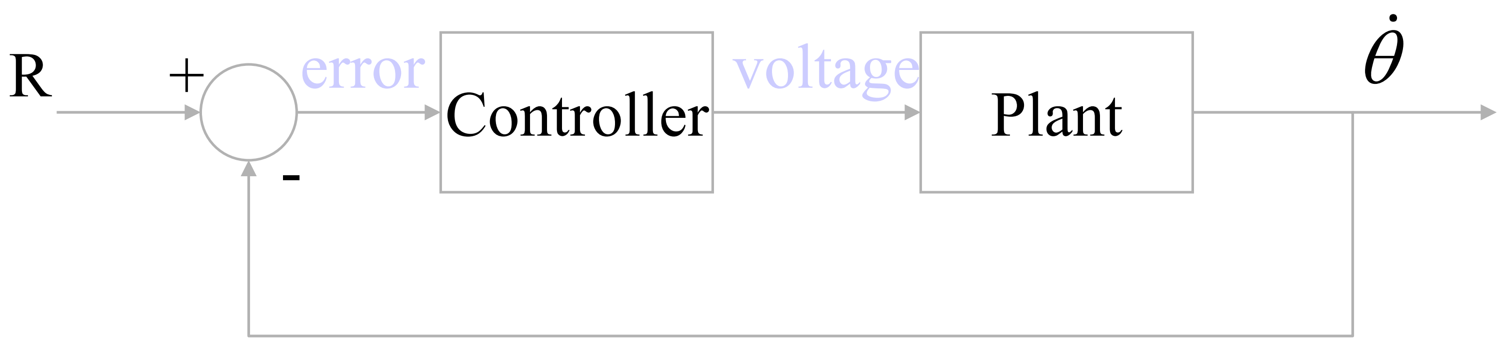


Proportional Control K_p

Easy to implement
Input/Output units agree
Improved rise time

Steady State Error (true)

↑P: ↓Rise Time vs. ↑ Overshoot*
↑P: ↓Rise Time vs. ↓ Settling time*
↑P: ↓Steady state error vs. other problems



Voltage = K_p error

*In some other systems, not mass-spring

Closed Loop Response (PI Feedback)

Proportional/Integral Control

$$K_p + \frac{1}{s} K_I$$

No Steady State Error

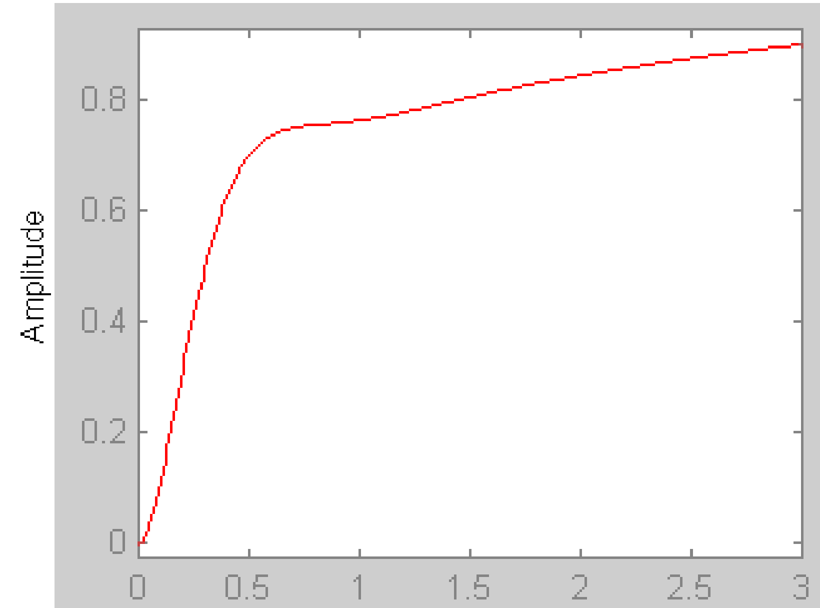
Bigger Overshoot and Settling

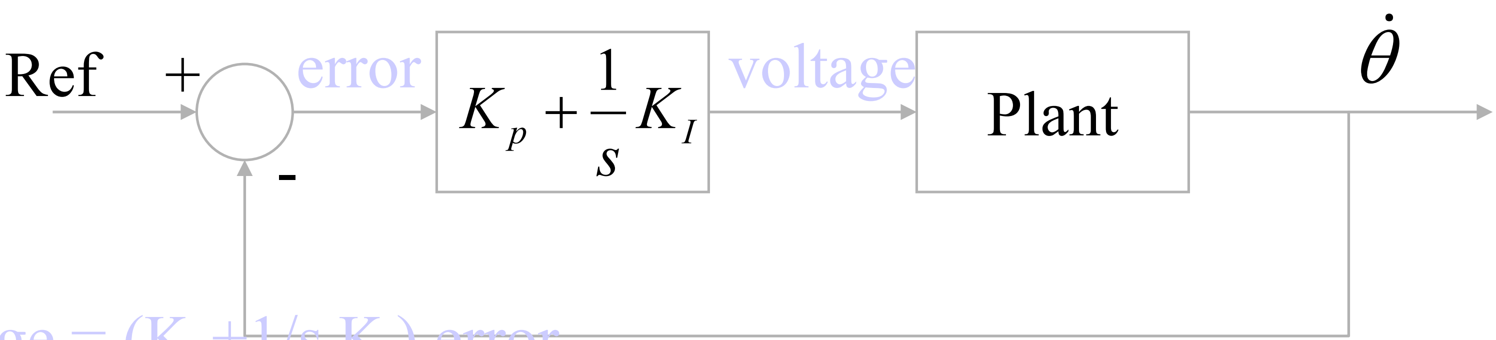
Saturate counters/op-amps

↑P: ↓Rise Time vs. ↑ Overshoot

↑P: ↓Rise Time vs. ↓ Settling time

↑I: ↓Steady State Error vs. ↑Overshoot





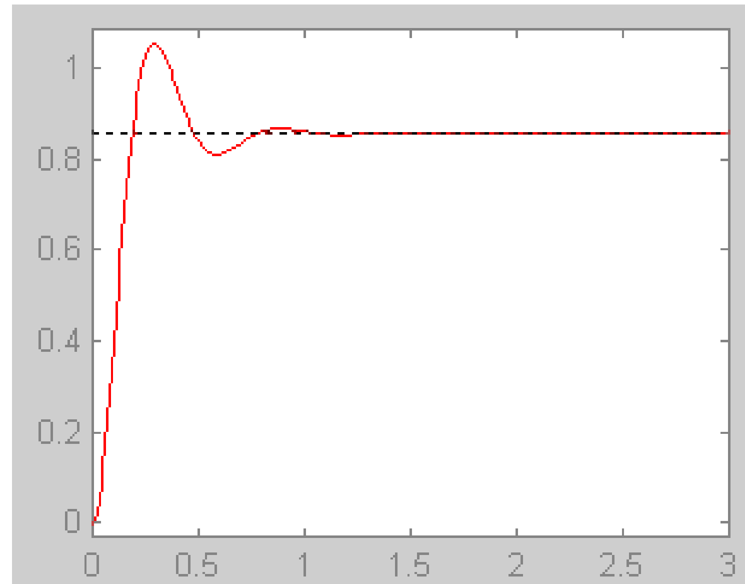
Voltage = (K_p+1/s K_i) error

Closed Loop Response (PID Feedback)

Proportional/Integral/Differential

Quick response
Reduced Overshoot

$$K_p + \frac{1}{s}K_I + sK_D$$

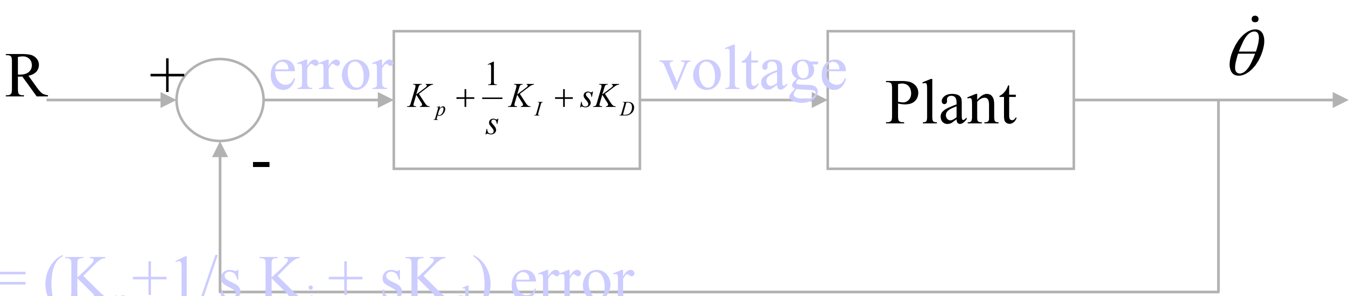


Sensitive to high frequency noise
Hard to tune

↑P: ↓Rise Time vs. ↑ Overshoot
↑P: ↓Rise Time vs. ↓ Settling time

↑I: ↓Steady State Error vs. ↑Overshoot

↑D: ↓Overshoot vs. ↑Steady State Error



Voltage = (K_p+1/s K_i + sK_d) error

Quick and Dirty Tuning

- Tune P to get the rise time you want
- Tune D to get the settling time you want
- Tune I to get rid of steady state error
- Repeat

- More rigorous methods – Ziegler Nichols, Self-tuning,
- Scary thing happen when you introduce the I term
  - Wind up (example with brick wall)
  - Instability around set point

Feed Forward

Volt

Decouples Damping from PID

To compute K_b

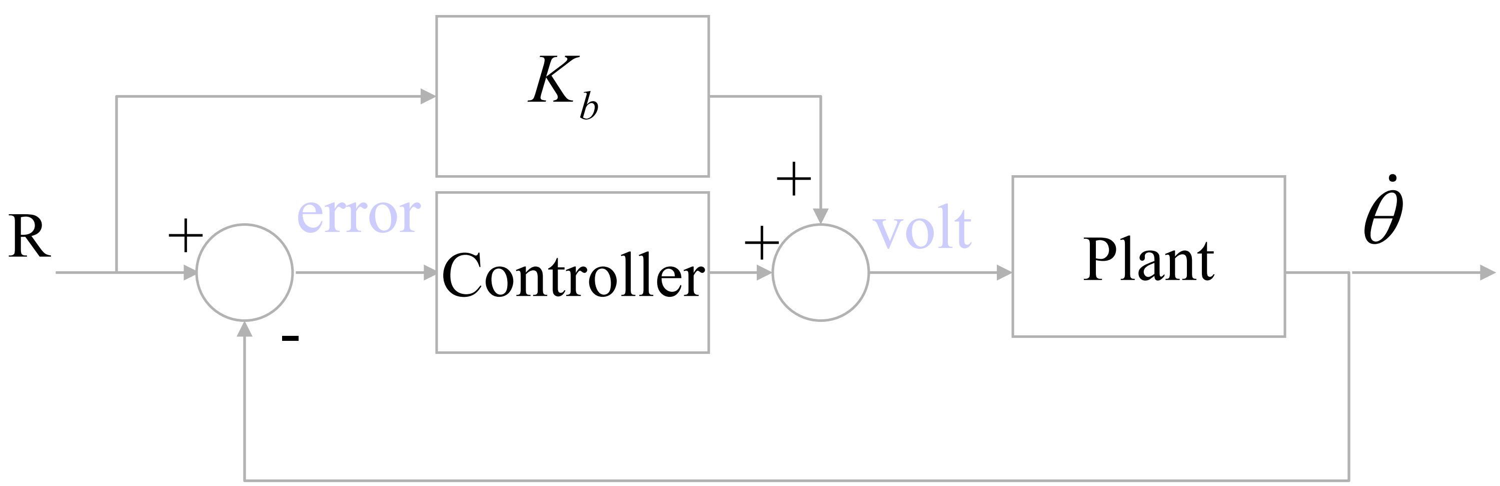
Try different *open* loop inputs and measure output velocities

For each trial i,

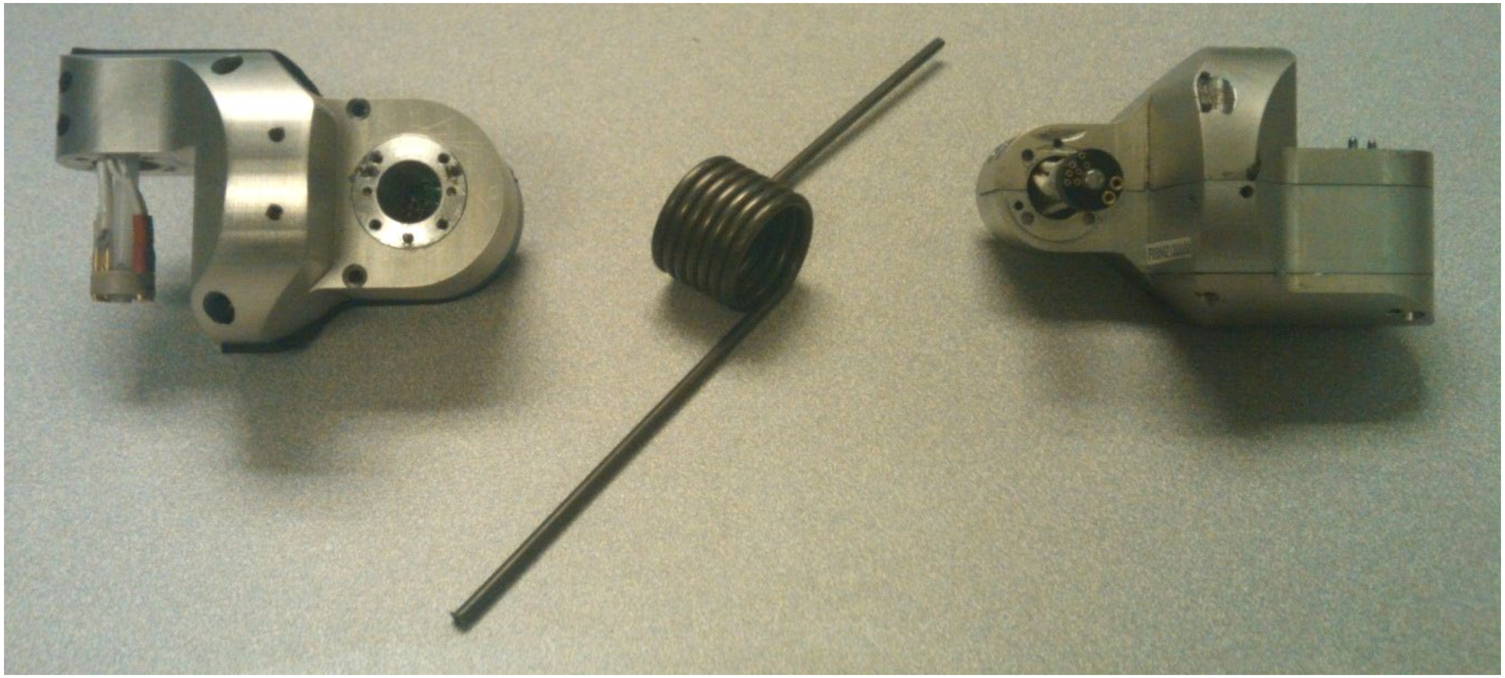
Tweak from there.

$$K_b^i = \frac{u_i}{\dot{\theta}_i}, \qquad K_b = \text{avg}\, K_b^i$$

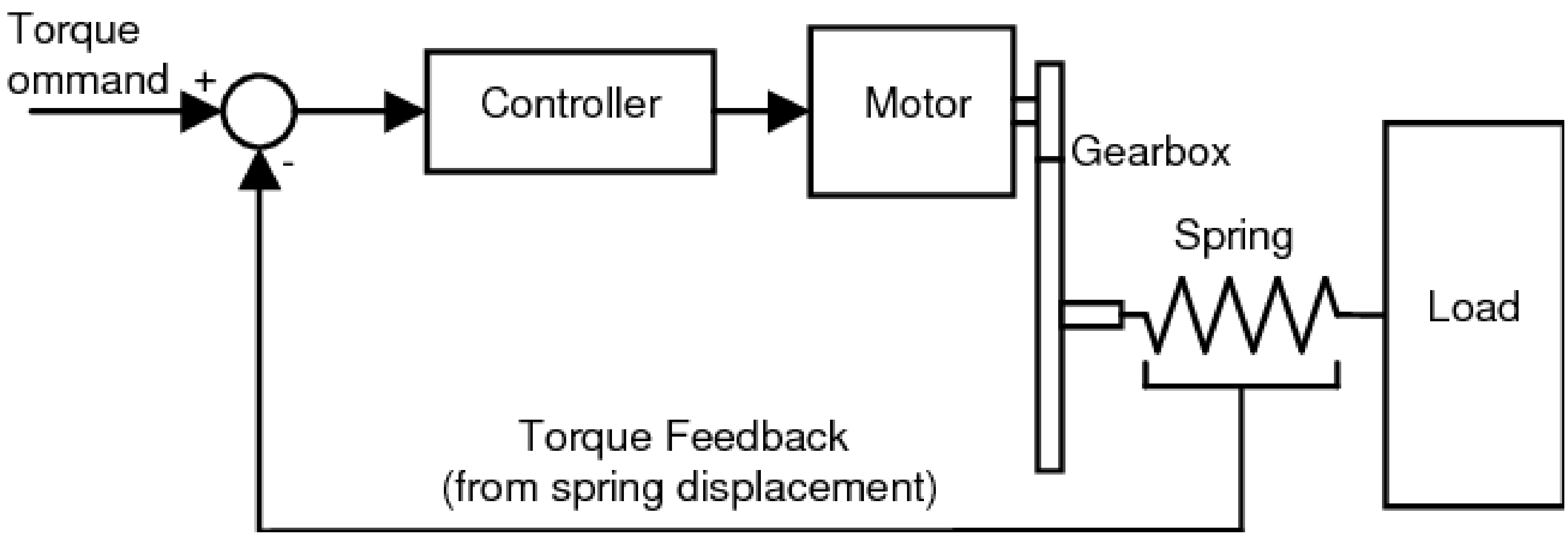
K



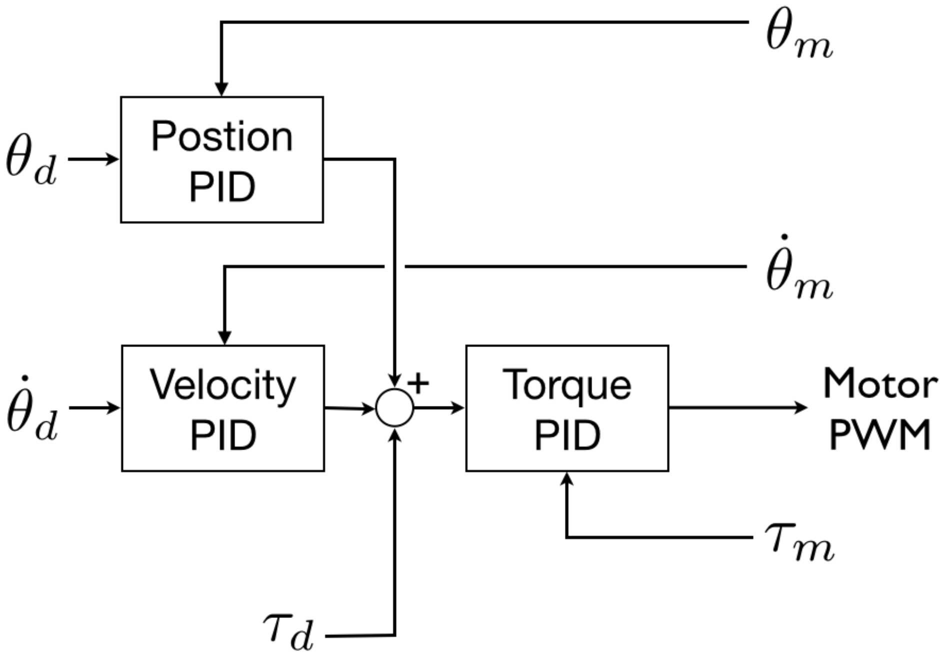
Series Elastic Module



Series Elastic Actuator



Module Control



Snake Monster

Snake Monster