CSE 490i
Lecture 3
Stable Feedback Control

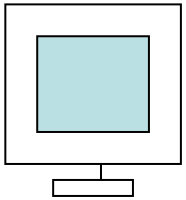
1/16/2007

Announcements

- PS2 due Thursday morning 10:30am
- PS3 (short!) posted today
- Lab1 writeup due this week (before your lab session by hand or electronically)
- Lab2 this week
- TA (Nan)'s office hours:

Friday vs Wednesday (moving assignment due date to Thursday?)

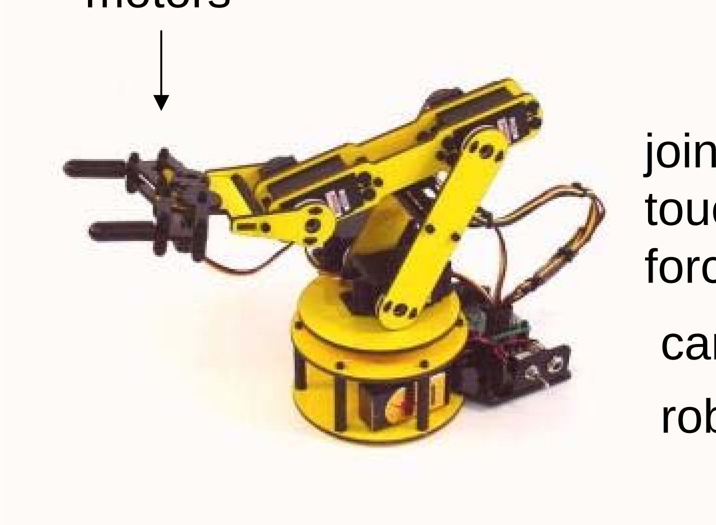
Robot Closed Loop System



computer

electronics (amplifiers, etc)

motors



sensory feedback

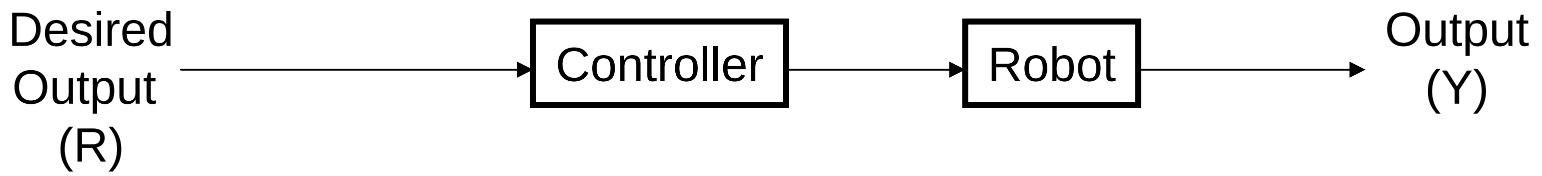
joint angle sensors
touch sensors at gripper
force sensors

camera to see the position

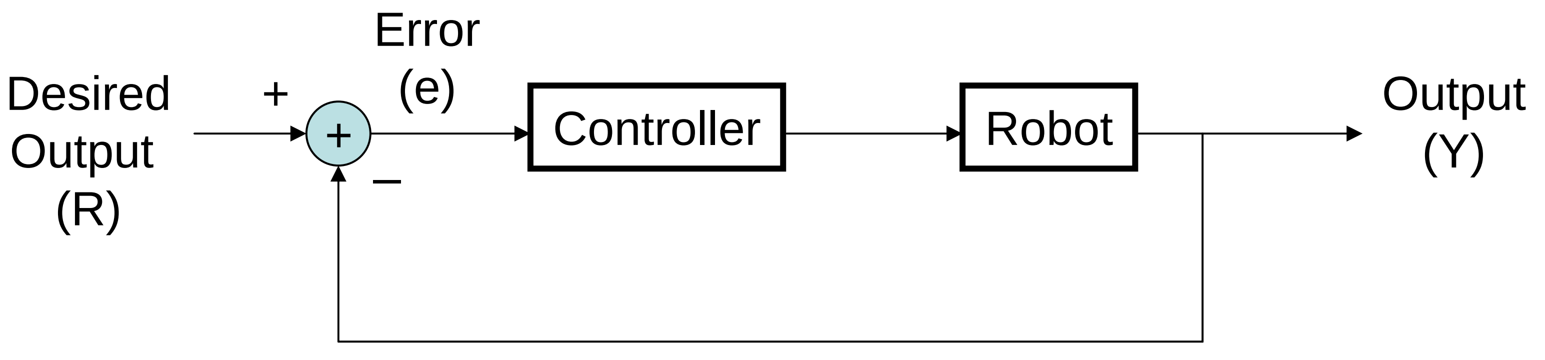
robotics math gives gripper position
based on joint angles

Controllers covered in the last lecture

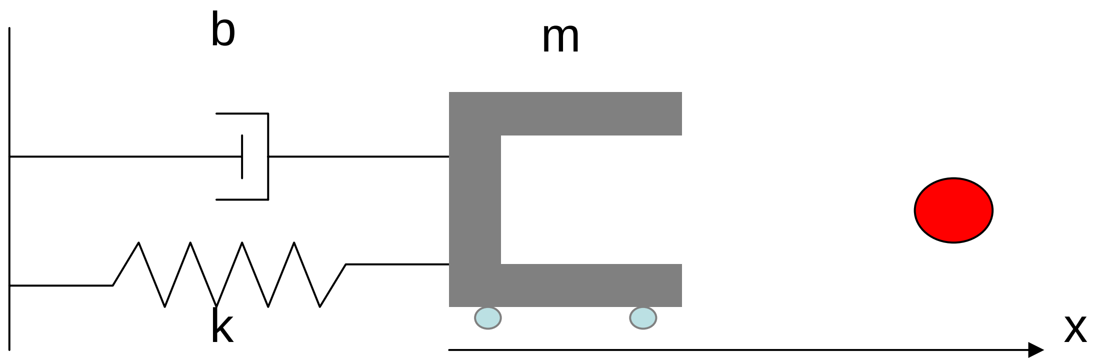
Open Loop Controller



Closed Loop Controller: Specifically we talked about proportional controller



Today: We will compare five different controllers using MATLAB

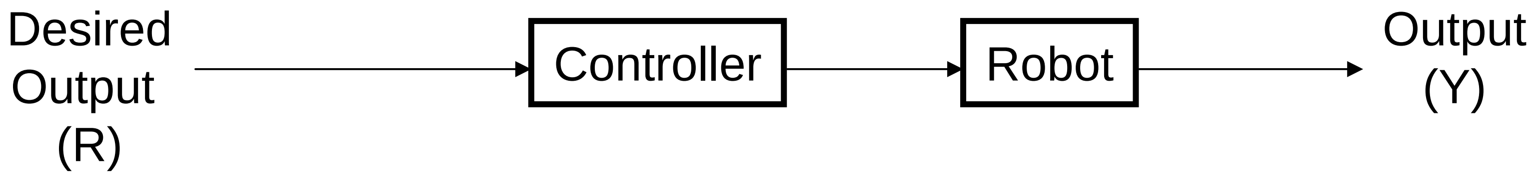


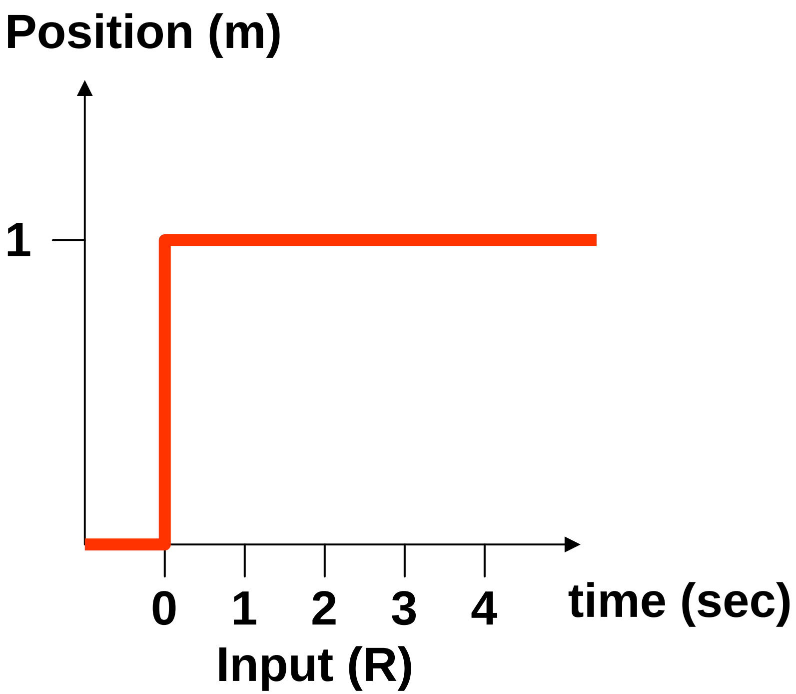
Same silly robotic gripper as the problem set:

Except,

m = 1kg, b = 8 N.s/m, k = 25 N/m

Start with an open loop



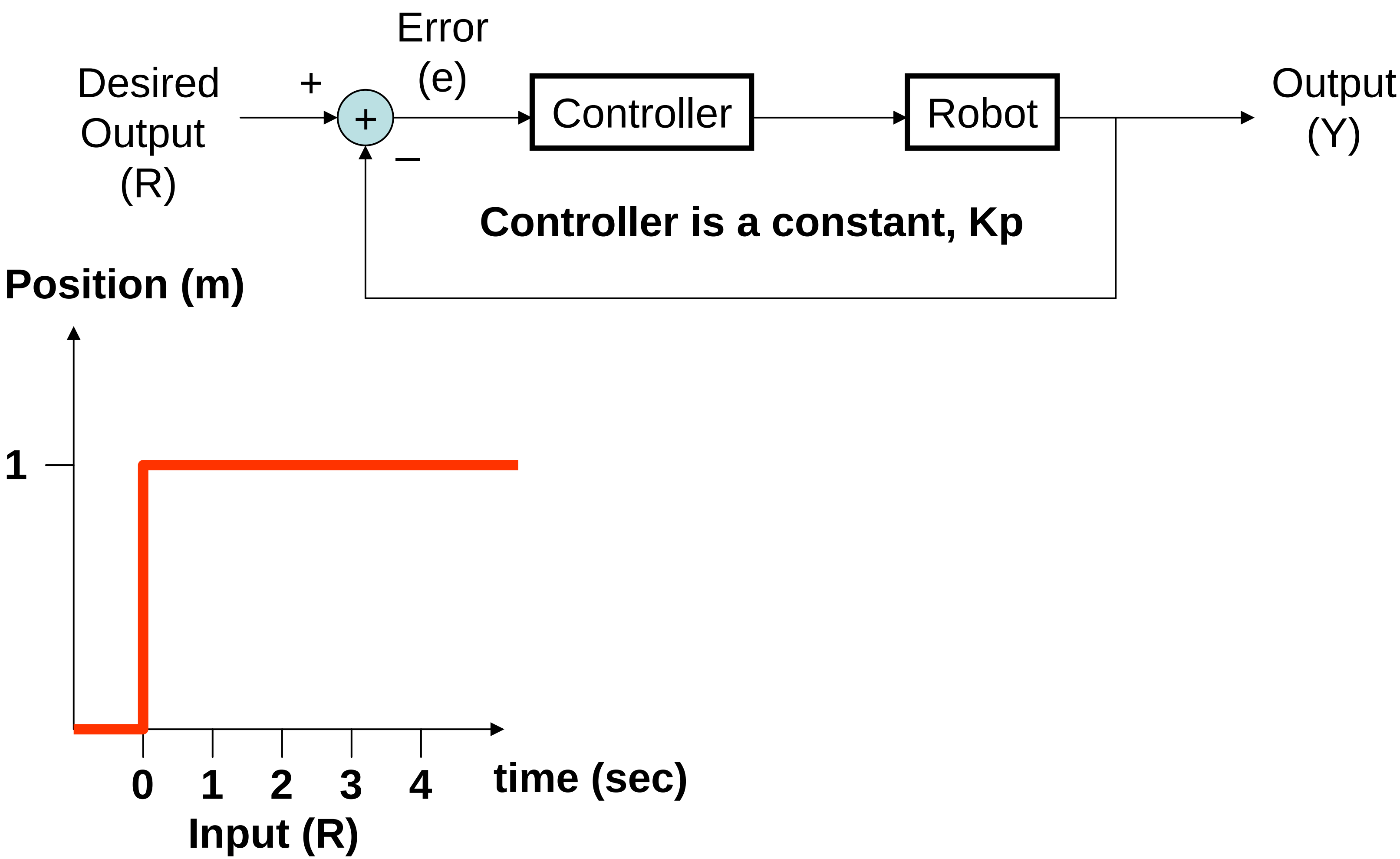


Problems with Open loop control

Cannot reject perturbation

Let's close the loop so that the controller is aware of the current error

Proportional Controller



Problems with P control

Steady state error (the error of the final value) cannot be reduced

It seems to oscillate a lot before settling down for higher gains

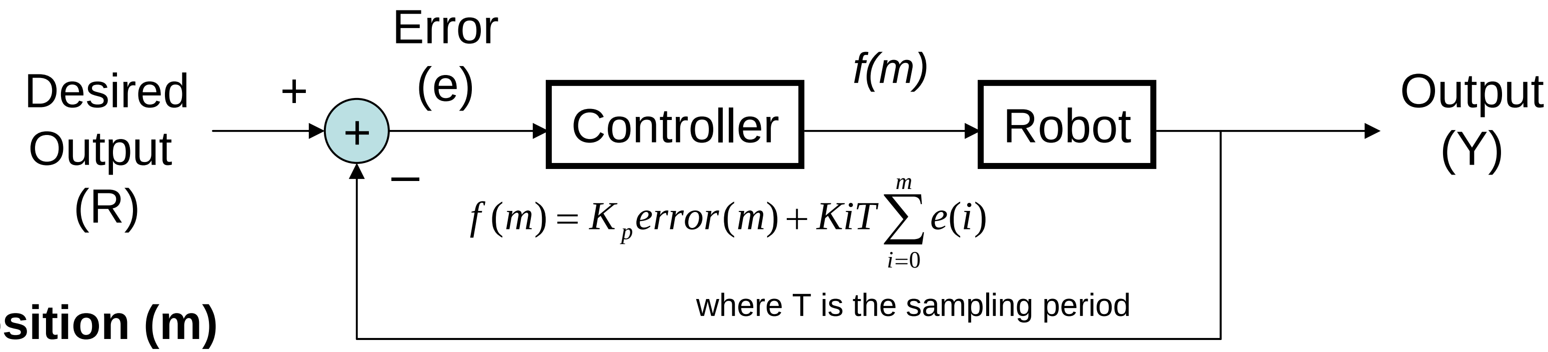
Let's deal with the steady state error first.

The reason why this error cannot be reduced: The controller output is not big enough to overcome the robot's mass, damping, and spring forces.

Raised the gain to solve this problem, but error is going down at the same time, so the output of the controller can't get as large as it needs.

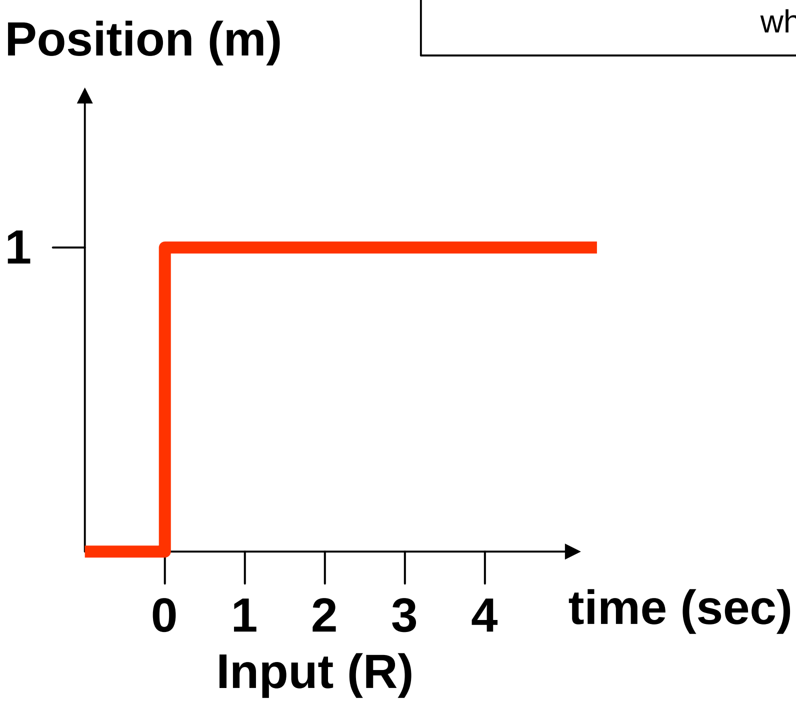
Need a term that gets larger if the error is not completely gone
Integral of error will accumulate to be large if the error is not 0.

Proportional plus Integral (PI) Controller



$$f(m) = K_p error(m) + KiT \sum_{i=0}^{m} e(i)$$

where T is the sampling period



Problems with P control

Steady state error (the error of the final value) cannot be reduced

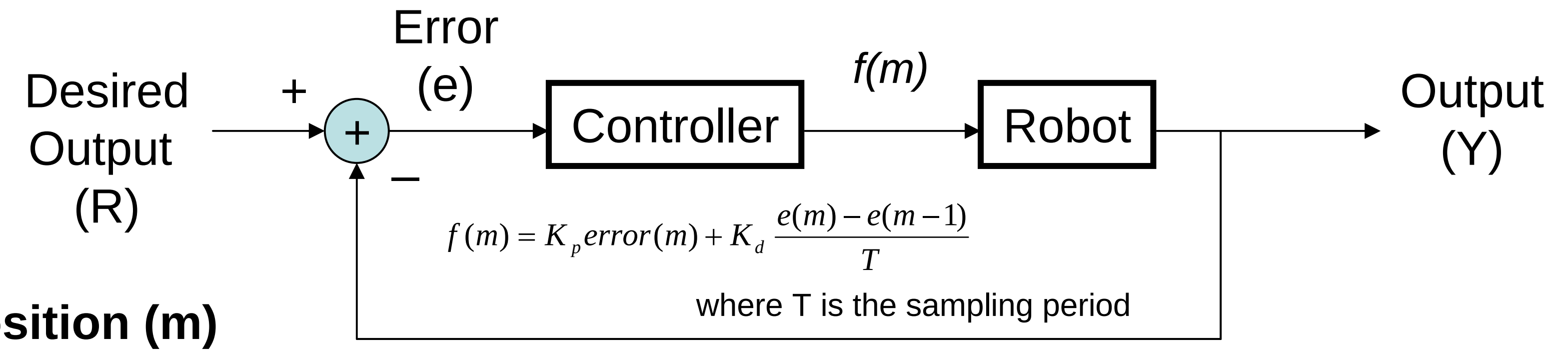
It seems to oscillate a lot before settling down for higher gains

Let's deal with the oscillation issue now

The reason why the output oscillates:
proportional error alone overcompensates.

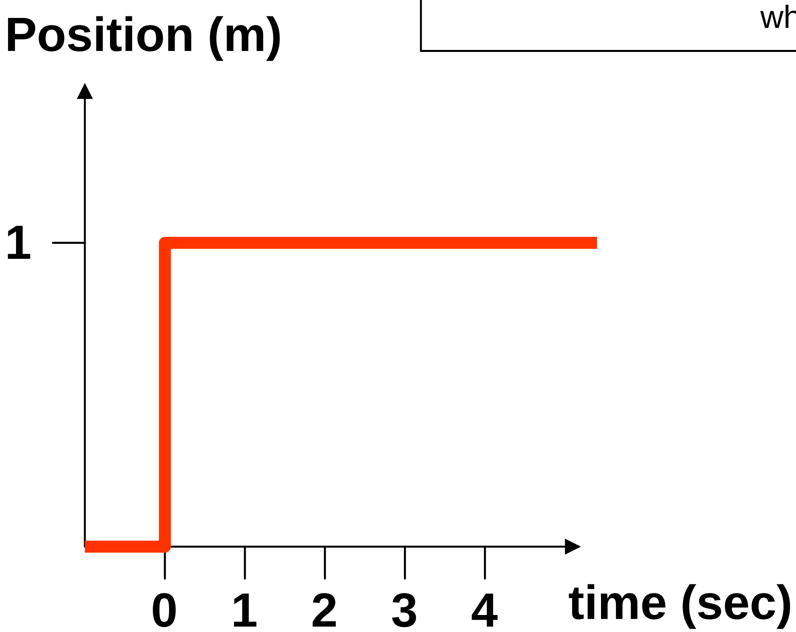
Need a term that can augment the overcompensation
Derivative of error has the right effect

Proportional plus Derivative (PD) Controller



$$f(m) = K_p error(m) + K_d \frac{e(m) - e(m-1)}{T}$$

where T is the sampling period



Input (R)

Problems with PI and PD controller

PI controller

the steady state error is gone, but

It still oscillated

It didn't settle to the final position fast

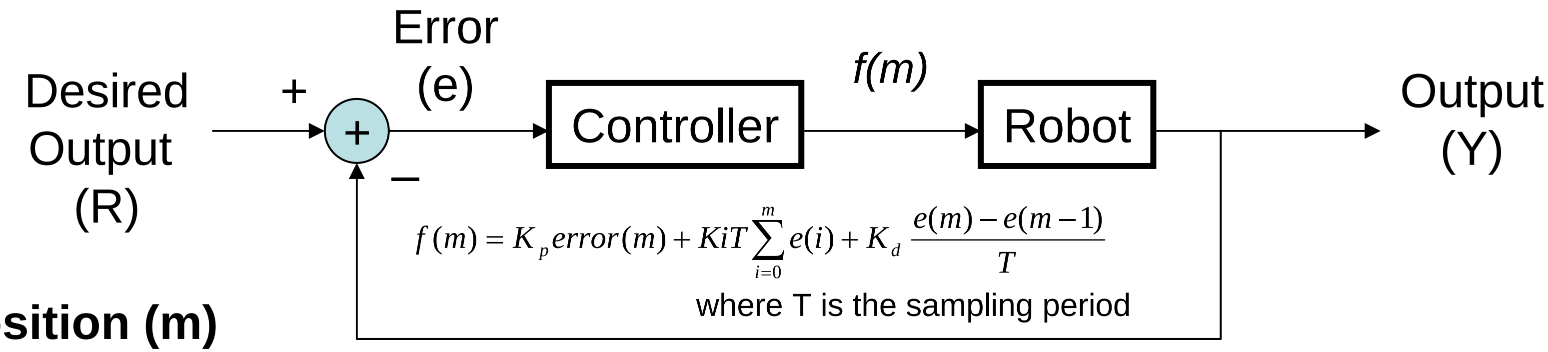
PD controller

the oscillation was reduced and the final value was reached fast, but

It didn't reach the desired output level as the final value

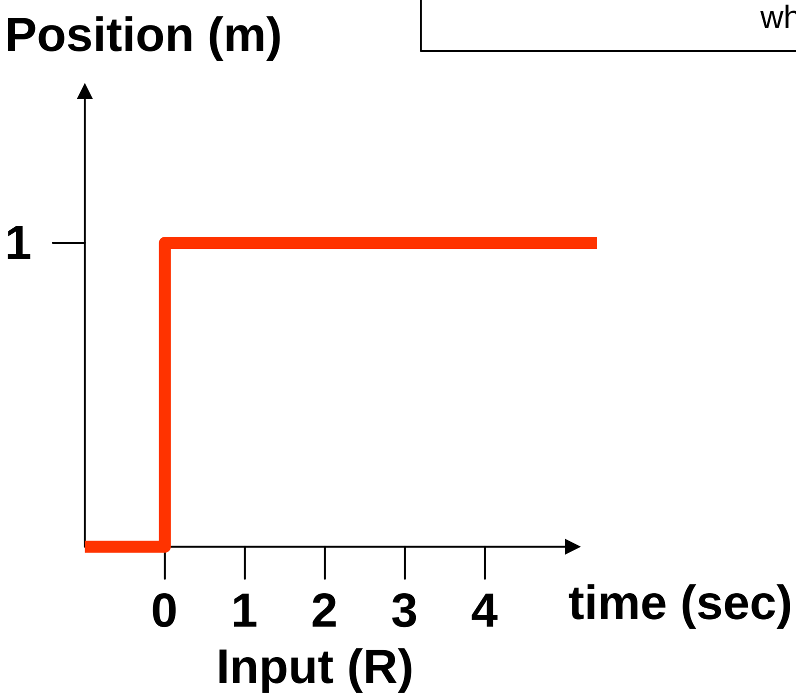
PID controller! Combine them all to take advantage of all terms

PID Controller



$$f(m) = K_p error(m) + KiT\sum_{i=0}^{m} e(i) + K_d \frac{e(m) - e(m-1)}{T}$$

where T is the sampling period



PID seems great but…

- PID control seems perfect, but sometimes you don’t need all the terms.

If the robot is not likely to oscillate, then D term is not as important
If the robot can reach the desired value, then I term is not as important