NAME SC - 32 COURSE ENGR 3030

DATE

LESSON PID 1

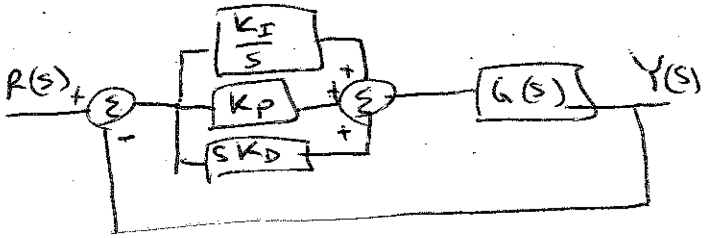
①

PID 1

(NISE CH 9)

Proportional Integral Derivative

P I D



- Today: concepts
- Next time: how to tune (MATLAB)

②

proportional Integral (PI)

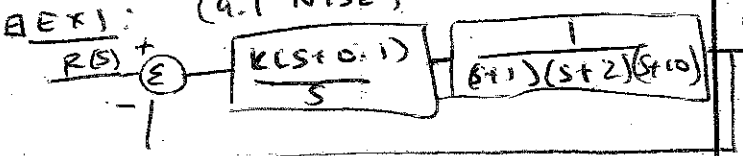
- can be used to improve steady state error
- $K_D = 0$

$$G_c(s) = K_P + \frac{K_I}{s}$$

$$= \frac{K_P\left(s + \frac{K_P}{K_I}\right)}{s}$$

PI controller

EX1: (9.1 NISE)



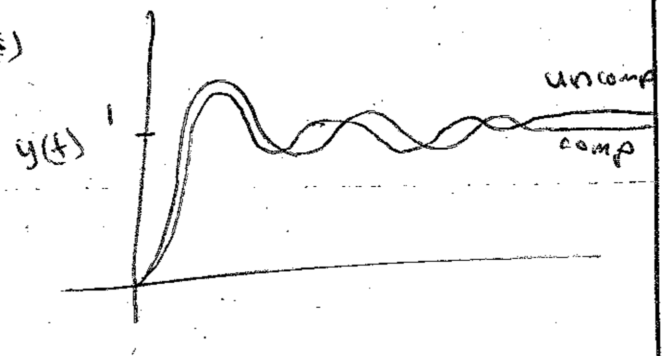
uncompensated

$$e_{ss} = \frac{1}{1+k_p} = 0.108$$

compensated

$K = 158.2$

$e_{ss} = 0$

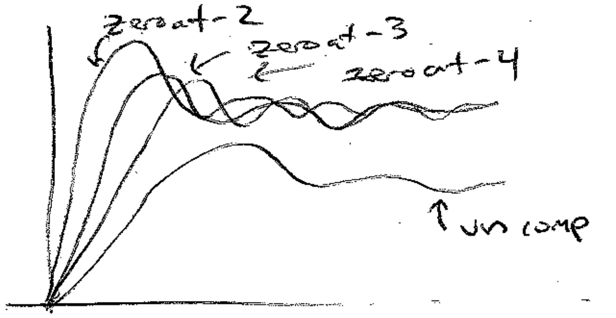


proportional Derivative (PD)

$K_I = 0$

$$G_c(s) = sK_D + K_P$$

- Improve transient response
- place a zero, speeds up response



NAME SC-32 COURSE ENGR 3030

DATE

LESSON PID 1

⑦

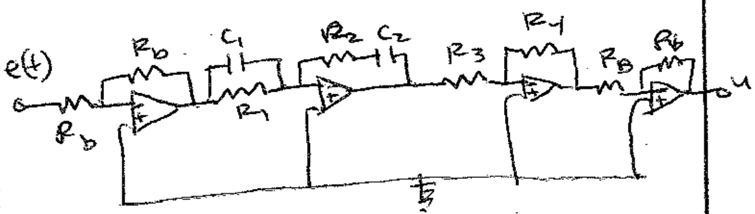
proportional Integral Derivative (PID)

- can fix both transient response and steady state error

$$G_c(S) = K_p + sK_D + \frac{K_p}{S}$$

⑧

Implementation



$$G_c(s) = \left(\frac{R_2 R_4}{R_1 R_3}\right)\left[\frac{(R_1 C_1 s+1)(R_2 C_2 s+1)}{R_2 C_2 s}\right]$$

⑨

$$K_p = \frac{R_4}{R_1 R_3 C_2}(R_1 C_1 + R_2 C_2)$$

$$K_I = \frac{R_4}{R_1 R_3 C_2}$$

$$K_D = \frac{R_2 R_4 C_1}{R_3}$$

$$K_p = K_I(R_1 C_1 + R_2 C_2)$$

can solve for R, C values

⑩

Tuning

- Next time, we will explore tuning
- Use two Ziegler-Nicols tuning methods (1942)
- will write code in MATLAB to help automate the process.
- Notes about Ziegler-Nicols method on Bridges

Exam

HW 5, HW 6, Lab 1 & 2

- Second order time response
- Routh Hurwitz
- Steady state response
- Root locus
- conceptual PID

PID

Method 1

$$K_P = \frac{1.2\,(1)\,(1.45)}{10\,(0.15)} = 1.16$$

$$K_I = \frac{0.6\,(1)\,(1.45)}{10\,(0.15)^2} = 3.867$$

$$K_D = \frac{0.6(1)(1.45)}{10} = 0.087$$

Method 2

rlocus (G , linspace (0, 5, 1000))

$K_{cr} = 3.15$ $\quad f_0 = \frac{\omega_0}{2\pi} = 1.75$

$$K_P = 0.6\,(3.15) = 1.89$$

$$K_I = 1.2\,(1.75)\,(3.15) = 6.615$$

$$K_D = \frac{0.075\,(3.15)}{1.75} = 0.135$$

These don't work! Let's tune!

$MP\% \leq 15\%$

$t_s \leq 1s$

Decrease K_D or decrease K_I

slows us down
extra pole

- Start with K_I, reduce until satisfy t_s (factor of 2 reductions)
- $MP(\%)$ still too big, reduce K_D

last

- Decrease K_p works!

Increase each parameter independently

Parameter	t_r	$mp\%$	settling time	e_{ss}
K_p	Decrease	Increase	Small change	Decrease
K_I	Decrease	Increase	Increase	Eliminate
K_D	Small Change	Decrease	Decrease	Nothing.