http://elec3004.com

# Advanced PID Control

ELEC 3004: **Systems**: Signals & Controls
Dr. Surya Singh

Lecture 20: Part I

elec3004@itee.uq.edu.au

http://robotics.itee.uq.edu.au/~elec3004/

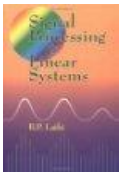
May 15, 2019

---

## Lecture Schedule:

| Week | Date | Lecture Title |
|---|---|---|
| 1 | 27-Feb | Introduction |
| | 1-Mar | Systems Overview |
| 2 | 6-Mar | Systems as Maps & Signals as Vectors |
| | 8-Mar | Systems: Linear Differential Systems |
| 3 | 13-Mar | Sampling Theory & Data Acquisition |
| | 15-Mar | Aliasing & Antialiasing |
| 4 | 20-Mar | Discrete Time Analysis & Z-Transform |
| | 22-Mar | Second Order LTID (& Convolution Review) |
| 5 | 27-Mar | Frequency Response |
| | 29-Mar | Filter Analysis |
| 6 | 3-Apr | Digital Filters (IIR) & Filter Analysis |
| | 5-Apr | PS 1: Q & A |
| 7 | 10-Apr | Digital Windows |
| | 12-Apr | Digital Filter (FIR) |
| 8 | 17-Apr | Active Filters & Estimation |
| | 19-Apr | **Holiday** |
| | 24-Apr | |
| | 26-Apr | |
| 9 | 1-May | Introduction to Feedback Control |
| | 3-May | Servoregulation & PID Control |
| 10 | 8-May | State-Space Control |
| | 10-May | Guest Lecture: FFT |
| **11** | **15-May** | **Advanced PID & & FFT Processes** |
| | 17-May | State Space Control System Design |
| 12 | 22-May | Digital Control Design |
| | 24-May | Shaping the Dynamic Response |
| 13 | 29-May | System Identification & Information Theory & Information Space |
| | 31-May | Summary and Course Review |

## Follow Along Reading:



**B. P. Lathi**
***Signal processing and linear systems***
**1998**
**TK5102.9.L38 1998**

**G. Franklin,**
**J. Powell,**
**M. Workman**
*Digital Control of Dynamic Systems*
1990

**TJ216.F72 1990**
[Available as UQ Ebook]

Today

➔ **State-space** 🡐 **[A stately idea! ☺]**

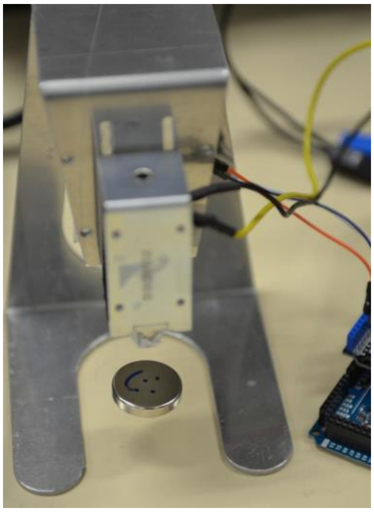
- FPW
  - Chapter 4: Discrete Equivalents to Continuous
  - Transfer Functions: The Digital Filter

- Lathi Ch. 13
  - § 13.2 Systematic Procedure for Determining State Equations
  - § 13.3 Solution of State Equations

***Next Time***

---

## ✪ This WEEK: Lab 4 – LeviLab II ✪



- AKA “Revenge of the TUNING!”

# PID

## When Can PID Control Be Used?

When:

- “Industrial processes” such that the demands on the performance of the control are not too high.
  - Control authority/actuation
  - Fast (clean) sensing
- PI: Most common
  - All stable processes can be controlled by a PI law (modest performance)
  - First order dynamics

PID (PI + Derivative):

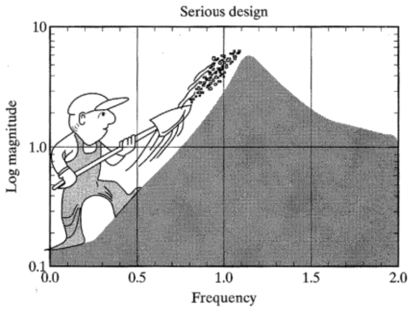
- Second order
  (A double integrator cannot be controlled by PI)
- Speed up response
  When time constants differ in magnitude
  (Thermal Systems)

**Something More Sophisticated:**

- Large time delays
- Oscillatory modes between inertia and compliances

## Seeing PID – No Free Lunch

- The energy (and sensitivity) moves around (in this case in “frequency”)



- Sensitivity reduction at low frequency unavoidably leads to sensitivity increase at higher frequencies.

**Source**: Gunter Stein's interpretation of the water bed effect – G. Stein, *IEEE Control Systems Magazine*, 2003.

## Another way to see P I|D

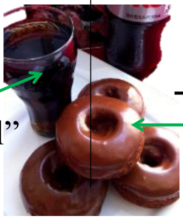
- Derivative

  D provides:
  - High sensitivity
  - Responds to change
  - Adds “damping” & ∴ permits larger $K_P$
  - Noise sensitive
  - Not used alone (∵ its on rate change of error – by itself it wouldn’t get there)

→ “**D**iet Coke of control”

- Integral
  - Eliminates offsets (makes regulation ☺)
  - Leads to Oscillatory behaviour
  - Adds an “order” but instability (Makes a 2nd order system 3rd order)

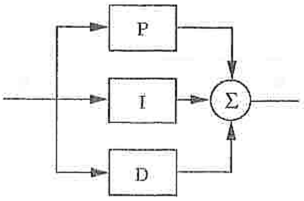
→ “**I**nteresting cake of control”



# PID Implementation

## PID Implementation

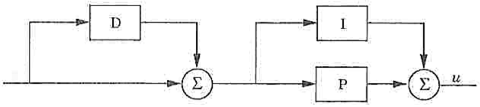
Non-Interacting (Standard Form)



$$C(s) = K\left(1 + \frac{1}{sT_i} + sT_d\right)$$

- Integral time ($T_i$) **does not** influence derivative

Interacting Form (Series Form)

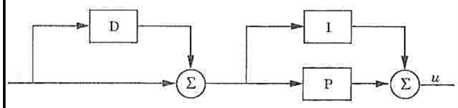


$$C'(s) = K'\left(1 + \frac{1}{sT'_i}\right)(1 + sT'_d)$$

- Note: Different $K$,$T_i$ and $T_d$
- $T_d$ **does** influence $T_i$: *interacting*
- Easier to tune manually

# PID Implementation [2]

Interacting (Series) Form:



$$C'(s) = K'\left(1 + \frac{1}{sT'_i}\right)(1 + sT'_d)$$

Parallel Form:

$$C''(s) = k + \frac{k_i}{s} + sk_d$$

Non-Interacting ➔ Interacting Form:

$T_i > 4T_d$ | $K(s) = K'\left(1 + \frac{1}{sT'_i}\right)(1 + sT'_d)$

$$K' = \frac{K}{2}\left(1 + \sqrt{1 - 4\frac{T_d}{T_i}}\right)$$

$$T'_i = \frac{T_i}{2}\left(1 + \sqrt{1 - 4\frac{T_d}{T_i}}\right)$$

$$T'_d = \frac{T_i}{2}\left(1 - \sqrt{1 - 4\frac{T_d}{T_i}}\right)$$

---

# Implementation of Digital PID Controllers

We will consider the PID controller with an $s$-domain transfer function

$$\frac{U(s)}{X(s)} = G_c(s) = K_P + \frac{K_I}{s} + K_D s. \qquad (13.54)$$

We can determine a digital implementation of this controller by using a discrete approximation for the derivative and integration. For the time derivative, we use the **backward difference rule**

$$u(kT) = \left.\frac{dx}{dt}\right|_{t=kT} = \frac{1}{T}(x(kT) - x[(k-1)T]). \qquad (13.55)$$

The $z$-transform of Equation (13.55) is then

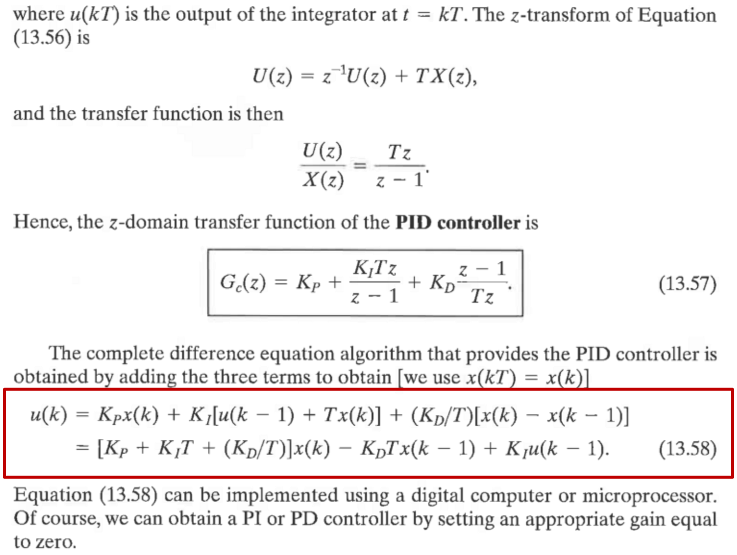
$$U(z) = \frac{1 - z^{-1}}{T}X(z) = \frac{z-1}{Tz}X(z).$$

The integration of $x(t)$ can be represented by the **forward-rectangular integration** at $t = kT$ as

$$u(kT) = u[(k-1)T] + Tx(kT), \qquad (13.56)$$

Source: Dorf & Bishop, Modern Control Systems, §13.9, pp. 1030-1

## Implementation of Digital PID Controllers (2)



where $u(kT)$ is the output of the integrator at $t = kT$. The $z$-transform of Equation (13.56) is

$$U(z) = z^{-1}U(z) + TX(z),$$

and the transfer function is then

$$\frac{U(z)}{X(z)} = \frac{Tz}{z-1}.$$

Hence, the $z$-domain transfer function of the **PID controller** is

$$\boxed{G_c(z) = K_P + \frac{K_I T z}{z-1} + K_D \frac{z-1}{Tz}.} \qquad (13.57)$$

The complete difference equation algorithm that provides the PID controller is obtained by adding the three terms to obtain [we use $x(kT) = x(k)$]

$$\begin{aligned} u(k) &= K_P x(k) + K_I[u(k-1) + Tx(k)] + (K_D/T)[x(k) - x(k-1)] \\ &= [K_P + K_I T + (K_D/T)]x(k) - K_D T x(k-1) + K_I u(k-1). \qquad (13.58) \end{aligned}$$

Equation (13.58) can be implemented using a digital computer or microprocessor. Of course, we can obtain a PI or PD controller by setting an appropriate gain equal to zero.

**Source**: Dorf & Bishop, Modern Control Systems, §13.9, pp. 1030-1

---

## Thus: PID in the Z-Domain

- PID Control may be considered in the Z-Domain as:

$$D(z) = K_p\left(1 + \frac{Tz}{T_I(z-1)} + \frac{T_D(z-1)}{Tz}\right)$$

- And/or as a difference equation as:

$$u(t_k) = u(t_{k-1}) + K_p\left[\left(1 + \frac{\Delta t}{T_I} + \frac{T_D}{\Delta t}\right)e(t_k) + \left(-1 - \frac{2T_D}{\Delta t}\right)e(t_{k-1}) + \frac{T_D}{\Delta t}e(t_{k-2})\right]$$

# PID Intuition

---

## PID Intuition & Tuning

- Tuning – How to get the "magic" values:
  - Ziegler Nichols Methods
  - Lambda Tuning
  - Skogestad's Internal Mode Controller
  - MIGO / AMIGO
  - (Dominant) Pole Placement
  - Auto Tuning (Optimization) …

- Although PID is common it is often poorly tuned
  - The derivative action is frequently switched off!

  (Why ∵ it's sensitive to noise)
  - Also lots of "I" will make the system more transitory & leads to integrator wind-up.

## PID Intuition

$$u(t) = K\left[e(t) + \frac{1}{T_i}\int e(s)\,ds + T_d\frac{de(t)}{dt}\right]$$

- P:
  - Control action is proportional to control error
  - It is necessary to have an error to have a non-zero control signal

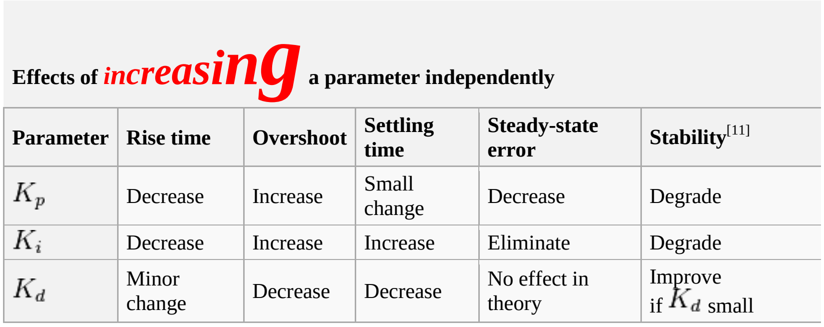
- I:
  - The main function of the integral action is to make sure that the process output agrees with the set point in steady state

## PID Intuition

$$u(t) = K\left[e(t) + \frac{1}{T_i}\int e(s)\,ds + T_d\frac{de(t)}{dt}\right]$$

- P:
- I:
- D:
  - The purpose of the derivative action is to improve the closed loop stability.
  - The instability “mechanism” “controlled” here is that because of the process dynamics it will take some time before a change in the control variable is noticeable in the process output.
  - The action of a controller with proportional and derivative action may e interpreted as if the control is made proportional to the *predicted* process output, where the prediction is made by extrapolating the error by the tangent to the error curve.

# Effects of Increasing Gain

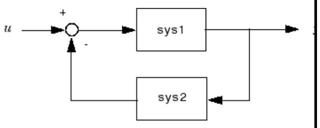


**Effects of *increasing* a parameter independently**

| Parameter | Rise time | Overshoot | Settling time | Steady-state error | Stability[11] |
|---|---|---|---|---|---|
| $K_p$ | Decrease | Increase | Small change | Decrease | Degrade |
| $K_i$ | Decrease | Increase | Increase | Eliminate | Degrade |
| $K_d$ | Minor change | Decrease | Decrease | No effect in theory | Improve if $K_d$ small |

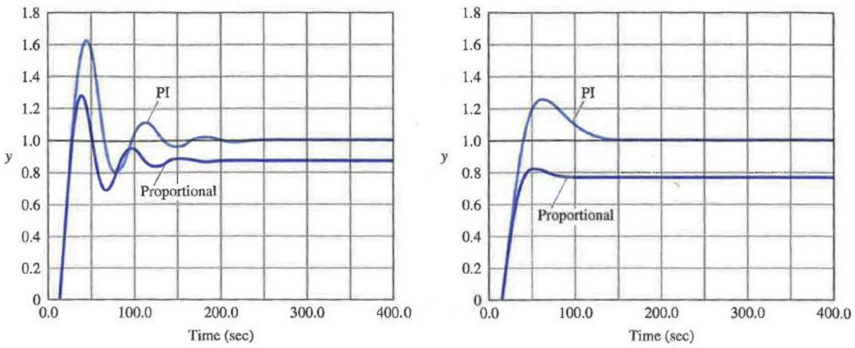
## Matlab helps with PID tuning:

```
G_system=[???]; H=[1];
D_compensator = pidtune(G_system, 'PIDF')
CL_system = feedback(series(D_compensator,G_system), H)
step(CL_system)
```



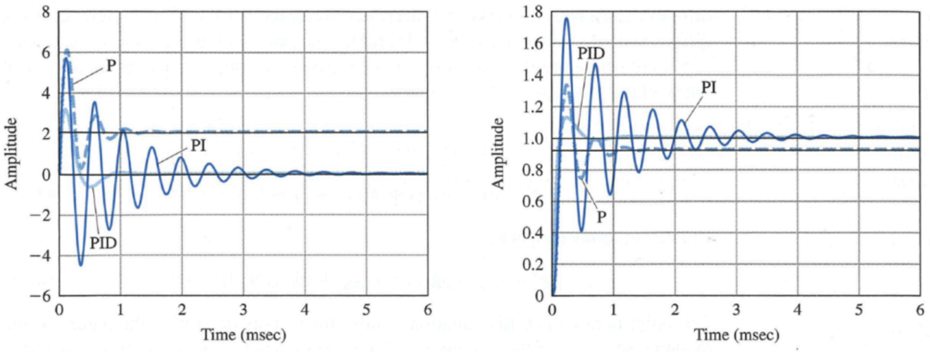
---

# PID Intuition: P and PI



## PID Intuition: P and PI and PID

- Responses of P, PI, and PID control to



(a) step disturbance input (b) step reference input

---

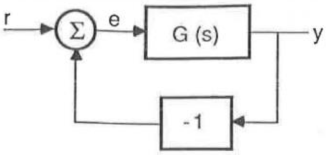
# PID
# (Tuning!)

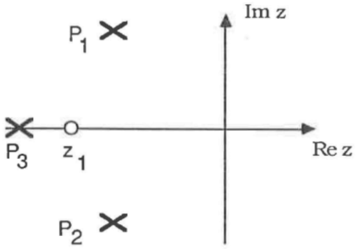
# Rule-Based Empirical Tuning

- Proportional Gain Increased ⇡:
  - Decrease Stability
- Integration Time is Increased ⇡:
  - Error decays more rapidly
- Derivative Time is Increased ⇡:
  - Increasing stability
- Integration Time is Decreased ⇣:
  - Decreases stability

---

# PI Control Tuning Via Dominant Pole





$$p_1 = -\zeta\omega_0 + i\omega_0\sqrt{1-\zeta^2} = -\sigma + i\omega$$

$$p_2 = -\zeta\omega_0 - i\omega_0\sqrt{1-\zeta^2} = -\sigma - i\omega$$

This implies

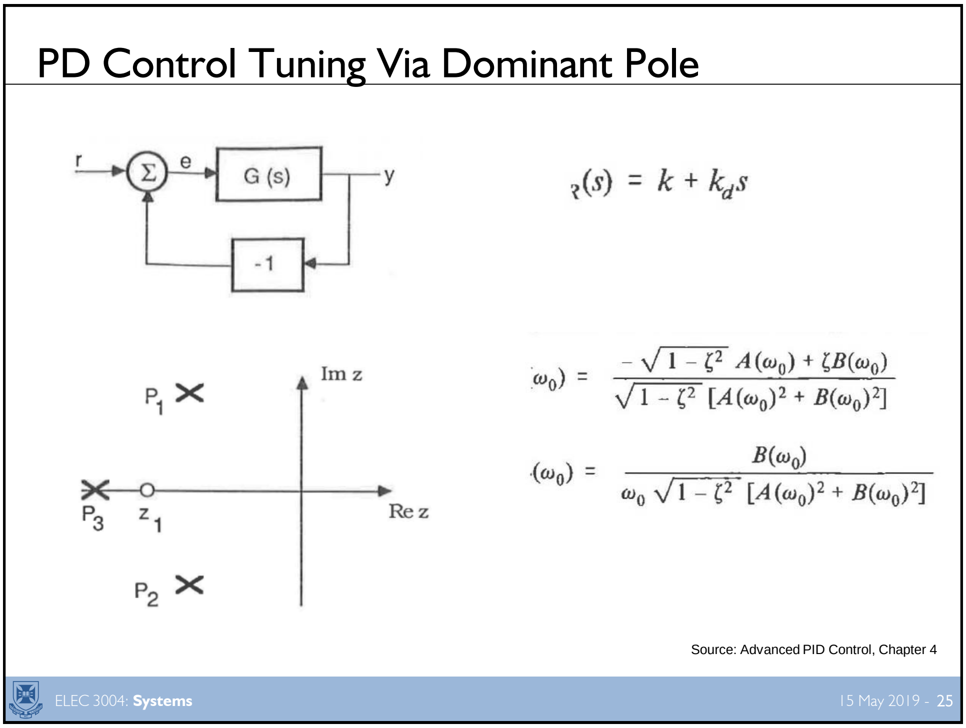
$$\begin{cases} 1 + \left[k + \dfrac{k_i}{p_1}\right] G_p(p_1) = 0 \\ 1 + \left[k + \dfrac{k_i}{p_2}\right] G_p(p_2) = 0 \end{cases}$$

where $G = G_R G_P$, and $G_P$ is the transfer functio equation is linear in $k$ and $k_i$. It has a solution if is

$$k(\omega_0) = -\frac{\sqrt{1-\zeta^2}\ A(\omega_0) + \zeta B(\omega_0)}{\sqrt{1-\zeta^2}\ [A(\omega_0)^2 + \zeta B(\omega_0)^2]}$$

$$k_i(\omega_0) = -\frac{\omega_0 B(\omega_0)}{\sqrt{1-\zeta^2}\ [A(\omega_0)^2 + B(\omega_0)^2]}$$

Source: Advanced PID Control, Chapter 4



## PD Control Tuning Via Dominant Pole

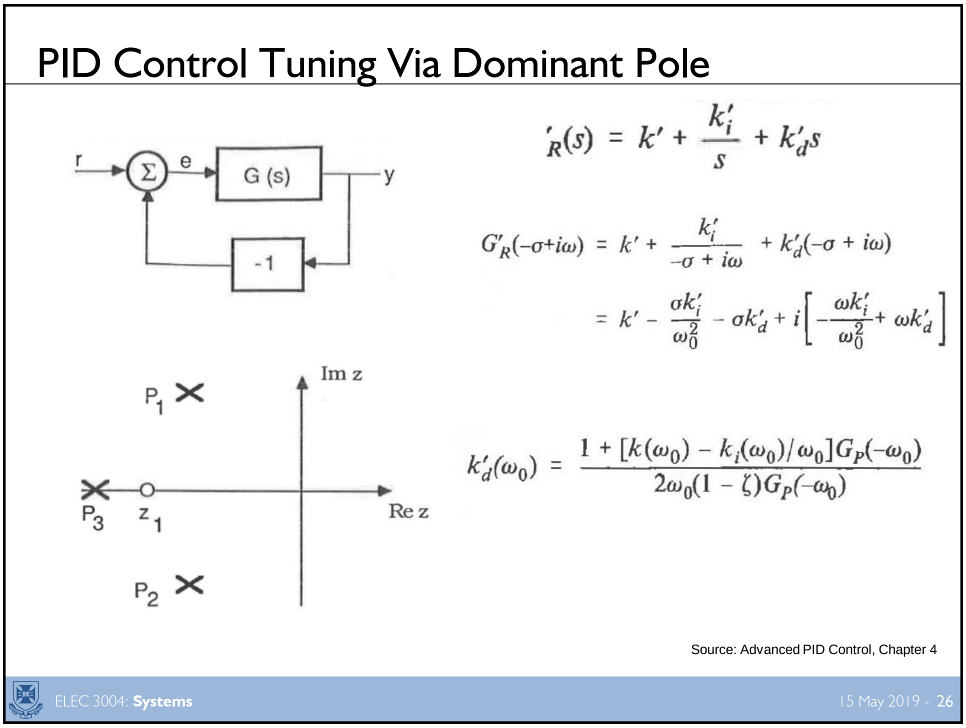
$$(s) = k + k_d s$$

$$\omega_0) = \frac{-\sqrt{1-\zeta^2}\,A(\omega_0) + \zeta B(\omega_0)}{\sqrt{1-\zeta^2}\,[A(\omega_0)^2 + B(\omega_0)^2]}$$

$$(\omega_0) = \frac{B(\omega_0)}{\omega_0\sqrt{1-\zeta^2}\,[A(\omega_0)^2 + B(\omega_0)^2]}$$

Source: Advanced PID Control, Chapter 4

---



## PID Control Tuning Via Dominant Pole

$$'_R(s) = k' + \frac{k_i'}{s} + k_d' s$$

$$G_R'(-\sigma+i\omega) = k' + \frac{k_i'}{-\sigma + i\omega} + k_d'(-\sigma + i\omega)$$

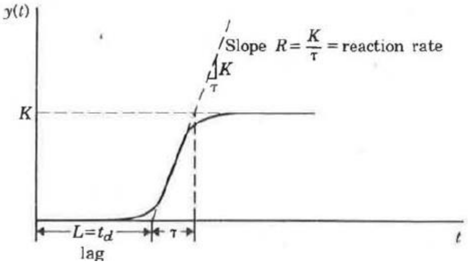
$$= k' - \frac{\sigma k_i'}{\omega_0^2} - \sigma k_d' + i\left[-\frac{\omega k_i'}{\omega_0^2} + \omega k_d'\right]$$

$$k_d'(\omega_0) = \frac{1 + [k(\omega_0) - k_i(\omega_0)/\omega_0]G_P(-\omega_0)}{2\omega_0(1-\zeta)G_P(-\omega_0)}$$

Source: Advanced PID Control, Chapter 4

# Ziegler-Nichols Tuning – Reaction Rate

FPW § 5.8.5 [p.224]



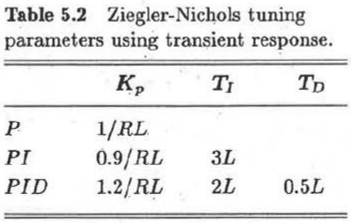
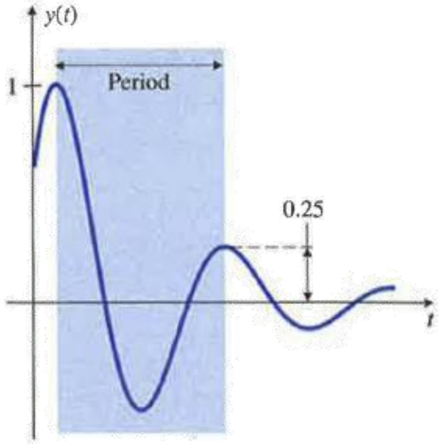


**Table 5.2** Ziegler-Nichols tuning parameters using transient response.

| | $K_p$ | $T_I$ | $T_D$ |
|---|---|---|---|
| $P$ | $1/RL$ | | |
| $PI$ | $0.9/RL$ | $3L$ | |
| $PID$ | $1.2/RL$ | $2L$ | $0.5L$ |

---

# Quarter decay ratio



## Ziegler-Nichols Tuning – Stability Limit Method

FPW § 5.8.5 [p.226]

- Increase $K_P$ until the system has continuous oscillations
  - ≡ $K_U$ : Oscillation Gain for “Ultimate stability”
  - ≡ $P_U$ : Oscillation Period for “Ultimate stability”

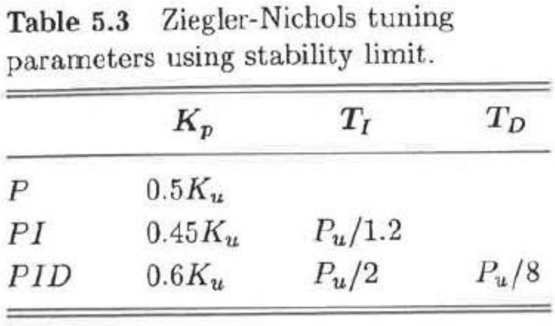
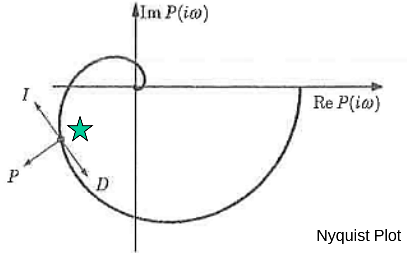


**Table 5.3** Ziegler-Nichols tuning parameters using stability limit.

| | $K_p$ | $T_I$ | $T_D$ |
|---|---|---|---|
| $P$ | $0.5K_u$ | | |
| $PI$ | $0.45K_u$ | $P_u/1.2$ | |
| $PID$ | $0.6K_u$ | $P_u/2$ | $P_u/8$ |

## Ziegler-Nichols Tuning / Intuition



Nyquist Plot

$$C(i\omega_u) = K\left(1 + i\left(\omega_u T_d - \frac{1}{\omega_u T_i}\right)\right) \approx 0.6K_u(1 + 0.467i)$$

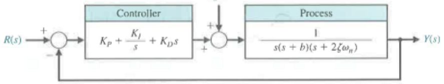
- For a Given Point (★), the effect of increasing P,I and D in the “s-plane” are shown by the arrows above Nyquist plot

## PID Example

- A 3rd order plant: b=10, ζ=0.707, $\omega_n$=4

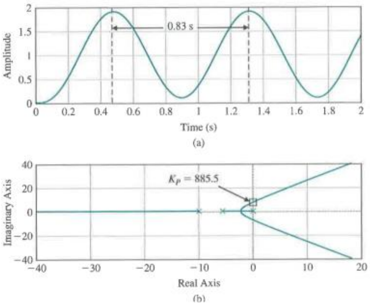
$$G(s) = \frac{1}{s(s+b)(s+2\zeta\omega_n)}$$

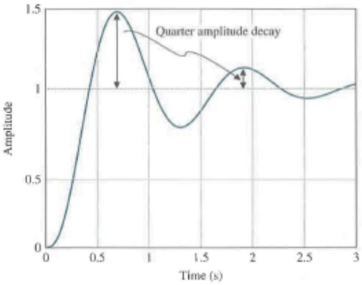
- PID:



- Kp=855:

· 40% Kp = 370





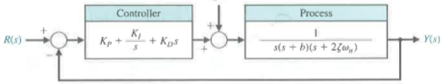
---

## PID Example

- A 3rd order plant: b=10, ζ=0.707, $\omega_n$=4

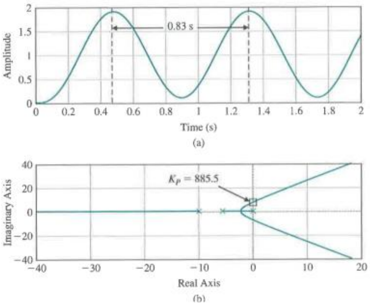
$$G(s) = \frac{1}{s(s+b)(s+2\zeta\omega_n)}$$

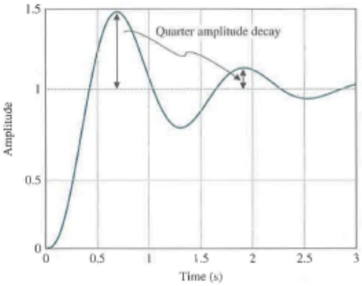
- PID:



- Kp=855:

· 40% Kp = 370





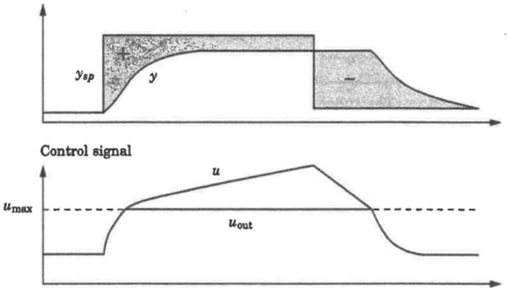
# PID

## (Non-Linear Effects [e.g. saturation])

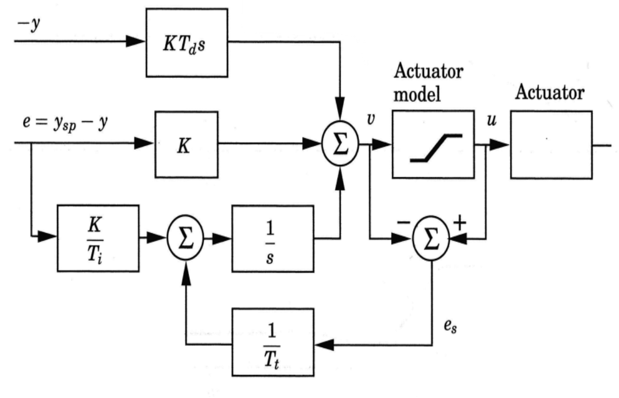
# Integrator Wind-Up

Wind-Up:

- A non-linear effect: motor limitations (speed, hysteresis, etc.) / **saturation**
- When this happens the feedback loop is broken and the system runs as an open loop because the actuator will remain at its limit independently of the process output.
- If a controller with integrating action is used, the error may continue to be integrated if the algorithm. is not properly designed. This means that the integral term may become very large or, colloquially, it “winds up.”
- Ex:



## An Anti Wind-Up Mechanism:



---

## Next Time…

- **Digital Feedback Control**

- Review:
  - Chapter 2 of FPW

- More Pondering??