Course: Regulation and control

Lecture 12: Controller Design I – PID

Lecturer: Chalachew Werku

Course Outline & Learning Objectives

- Understand the role of a controller in a feedback loop.
- Learn the principles of P, I, and D control actions.
- Identify the advantages and drawbacks of each control mode.
- Master the Ziegler-Nichols tuning methods.

The Big Picture: Revisiting the Basic Feedback Control Loop

Key Components:

- Setpoint (R)
- Controller (U)
- Process (G)
- Disturbance (D)
- Output (C)
- Feedback (H)

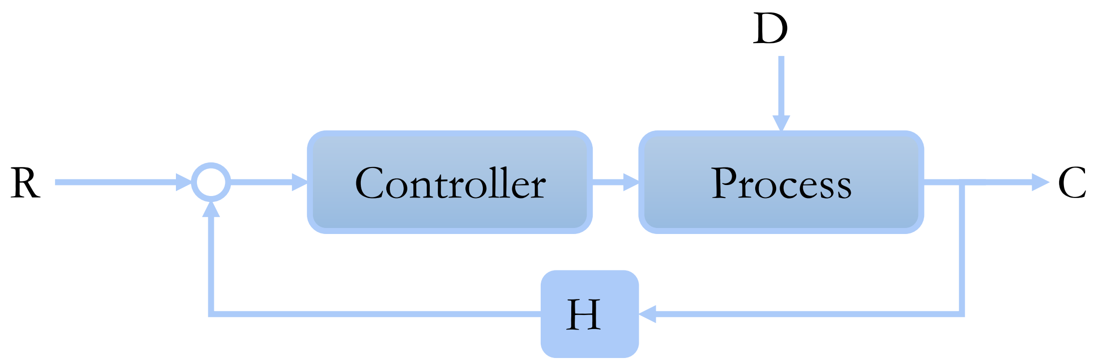


Fig 1: Feedback control system [1]

Objective: Make Output (C) follow Setpoint (R) despite disturbances (D).

Role of a Controller

1. Reduce Error (E = R - C): Make the difference between the **R**eference and the **C**ontrolled output as close to zero as possible.

2. Ensure System Stability: Prevent the system from oscillating uncontrollably, shaking apart, or diverging from the setpoint.

3. Achieve Desired Performance: How quickly the system reaches the desired setpoint, How much it temporarily exceeds the setpoint before settling, How close it finally gets to the setpoint after settling.

Introducing the PID Controller

- The "workhorse" of industrial control.

PID Control Law:

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{\partial e(t)}{\partial dt}$$

- Proportional (P): Present
- Integral (I): Past
- Derivative (D): Future

Proportional Control (P-Control)

- Concept: "React to the present error."
- Equation: $u(t) = K_p * e(t)$
- Simple and fast.
- A simple analogy: "Harder you push, the closer you get."

Fig 2: Harder you push, the closer you get. [2]

Visual Example of P-Control

- Step Response with varying K_p:
  - **Low K_p:** Slow, no oscillations, large steady-state error.
  - **Medium K_p:** Faster, some overshoot smaller error.
  - **High K_p:** Very fast, large oscillations potentially unstable.

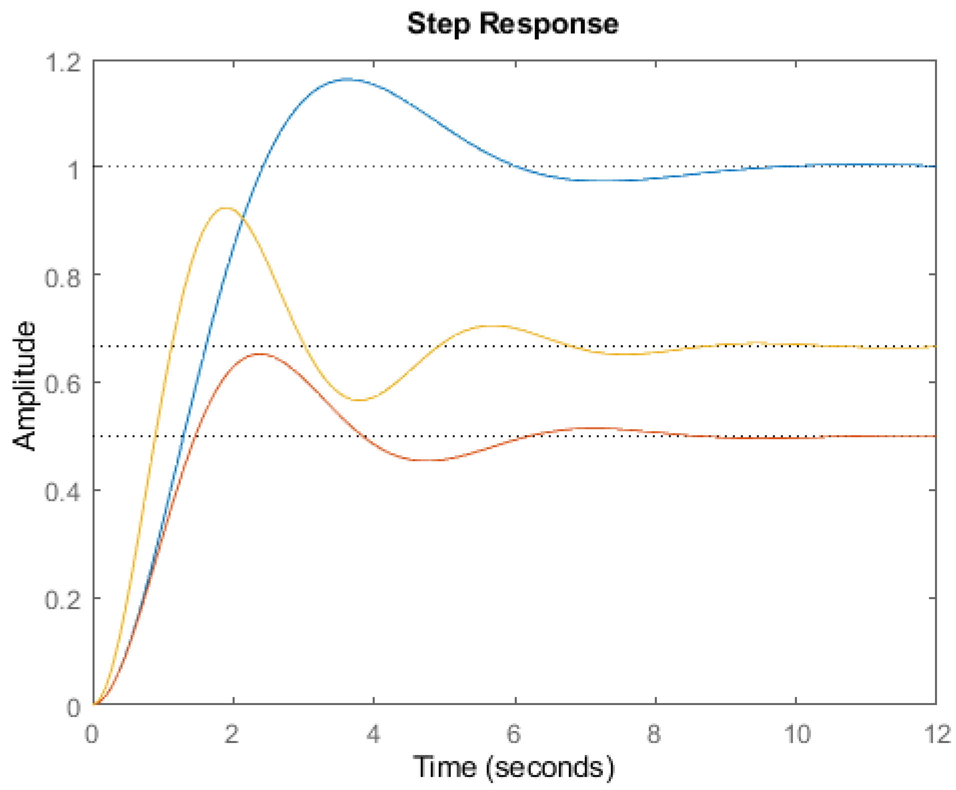


Fig 3: Effect of different proportional gain. [3]

The Problem with P-Control Only

- Steady-State Error (Offset)
- **Why?** A control effort is needed to maintain the output. With P-only, effort is proportional to error. Therefore, zero error means zero effort, which is often not enough to hold the setpoint.

- **Example:** Liquid Tank Level Control.
  - **Scenario:** Setpoint = 50 cm. Outflow increases.

- **P-Controller Response:** Level drops and settles at, e.g., 45 cm. The error of 5 cm is needed to generate just enough control effort (valve opening) to match the new outflow.

Introducing Integral Action (I-Control).

- Concept: "React to the accumulation of past errors.“

$$u_i = K_i \int e(t)dt$$

- It sums up the error over time.

Fig 4: Accumulating past errors [4]

How Integral Action Eliminates Steady-State Error

- Key Insight: As long as any error exists (positive or negative), the integral term keeps changing.
- It keeps adjusting the output until the error is driven to zero.
- $K_p=1$
- $K_p=1$ and $K_i=1$

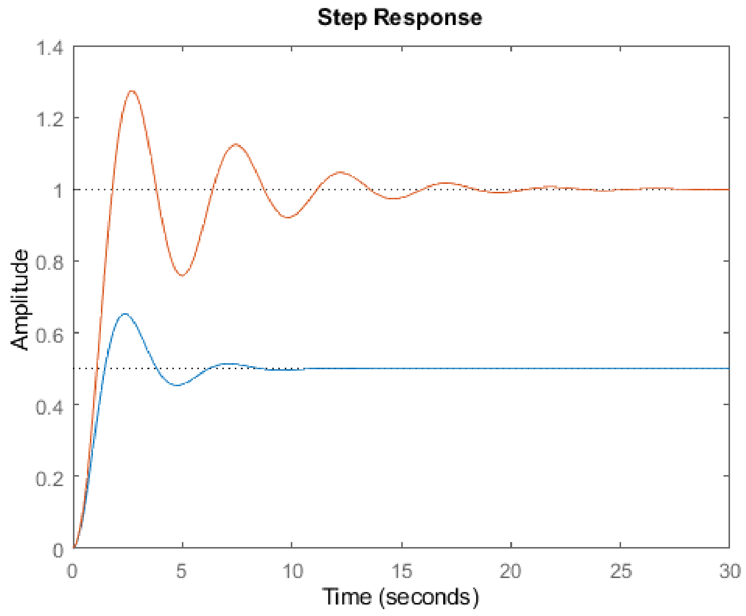


Fig 5: P vs PI controller. [5]

The Side-Effect of Integral Action: Integral Windup

- **What is it?** When the actuator saturates (e.g., a valve is fully open), the error remains large. The integrator "winds up," accumulating a huge value.
- **Problem:** When the setpoint is reached, it takes a long time to "unwind," causing large overshoot and sluggish response.

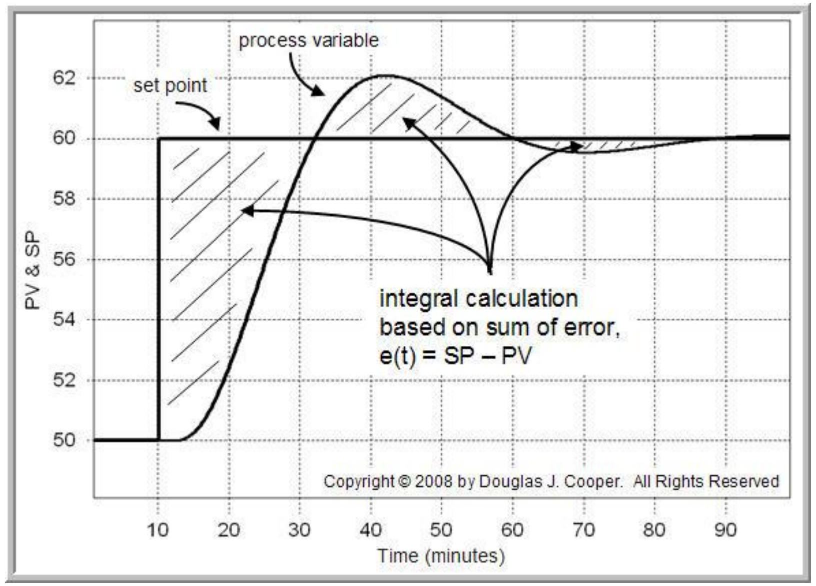


Fig 6: Integral windup [6]

Anti-Windup Strategies

- **Simple Method:** Integrator Clamping
  - Stop integrating when the actuator saturates.
  - Prevents the integral term from growing unnecessarily.
- **Other Methods:**
  - Back-calculation,
  - Conditional integration.

Introducing Derivative Action (D-Control)

- Concept: "React to the prediction of future errors."

$$u_d = K_d \frac{\partial e(t)}{\partial t}$$

- It looks at the rate of change (slope) of the error.

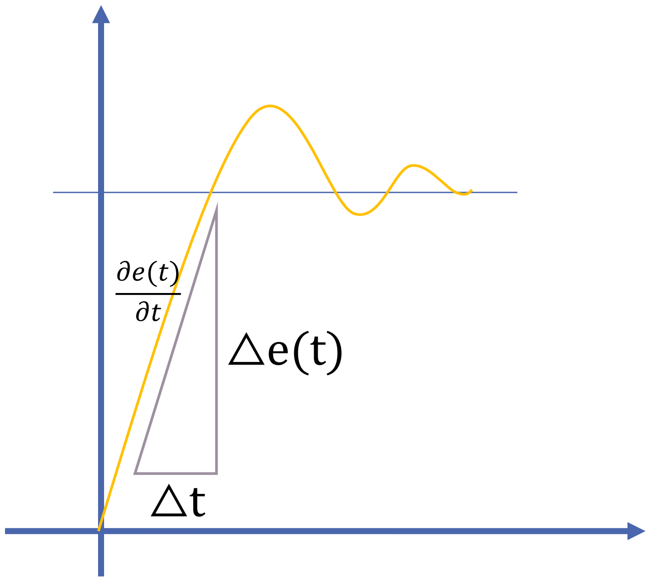


Fig 7: D-control, rate of change of error. [7]

How Derivative Action Damps the System

- **Action:** Opposes rapid changes in the error.
- **Effect:**
  - Reduces overshoot.
  - Damps out oscillations.
  - Increases stability.
- $K_p=1$
- $K_p=1, K_d=1$

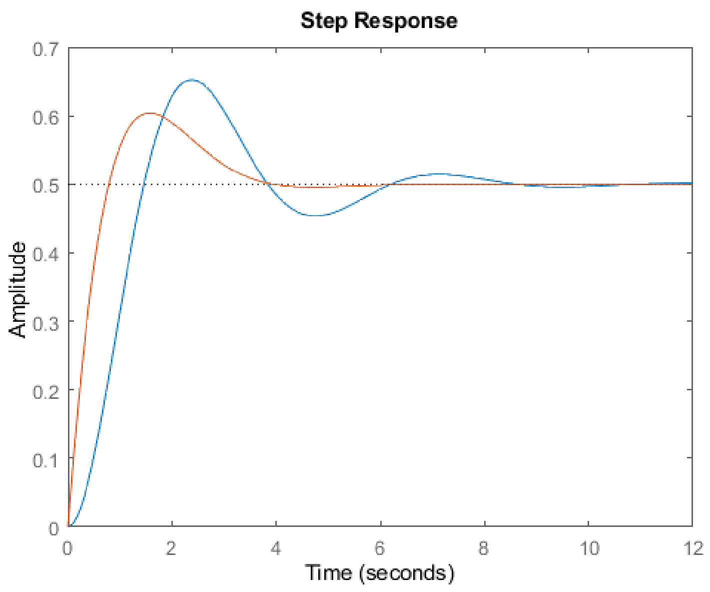


Fig 8: P vs PD controller. [8]

The Critical Weakness of Derivative Action

Noise Amplification

- **Problem:** Derivative action amplifies high-frequency noise.
  - de(t)/dt for a noisy signal can be very large.
  - This can lead to excessive control action and actuator wear.
- **Solution:** Use a "Practical" or "Ideal" derivative with a low-pass filter.
- Practical Derivative Form:

$$\frac{sK_d}{1 + s\left(\frac{K_d}{N}\right)}$$

where N is the filter coefficient (typically 5-20).

Proportional, Integral and Derivative (PID)

- Combines the three controllers and their advantages.
- P gives a direct response, I ensures long-term accuracy, and D anticipates and stabilizes the future. The controller's output is the sum of these three actions, tuned to achieve a fast, accurate, and stable system response.

$$u(t) = k_p e(t) + K_i \int e(t)dt + K_d \frac{\partial e(t)}{\partial t}$$

Example: P, PI, PID Responses

Process: G(s) = 1 / (s + 1)

- P-Control: Steady-state error, possible oscillations.
- PI-Control: No steady-state error, may be oscillatory.
- PID-Control: No steady-state error, fast, damped response.

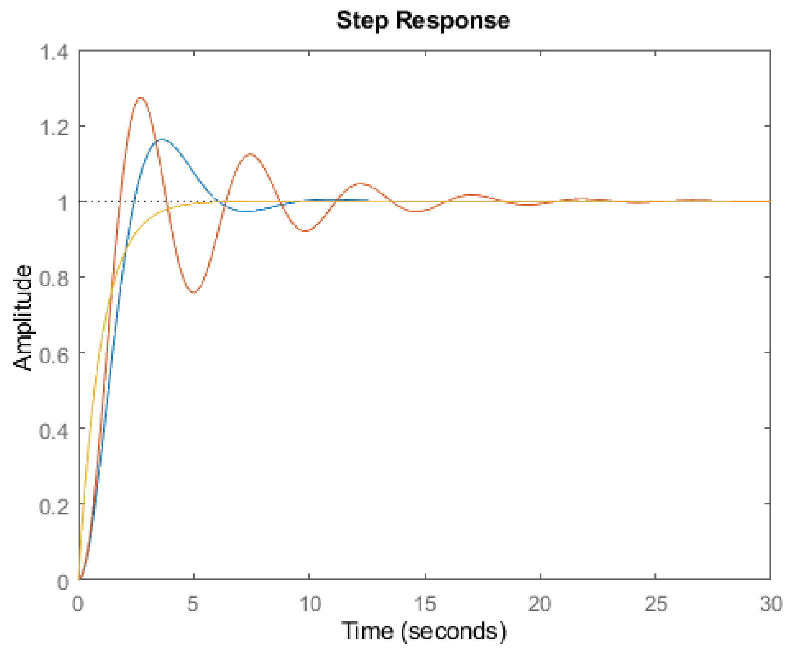


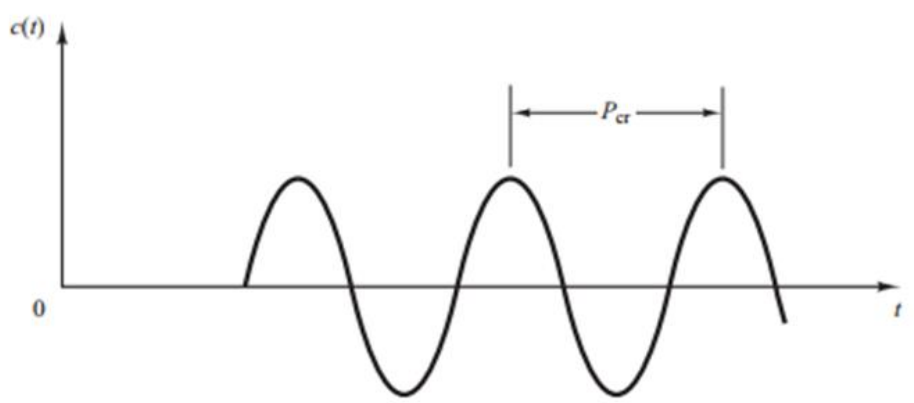
Fig 9: Effect of P, PI and PID controller. [9]

The Need for Tuning

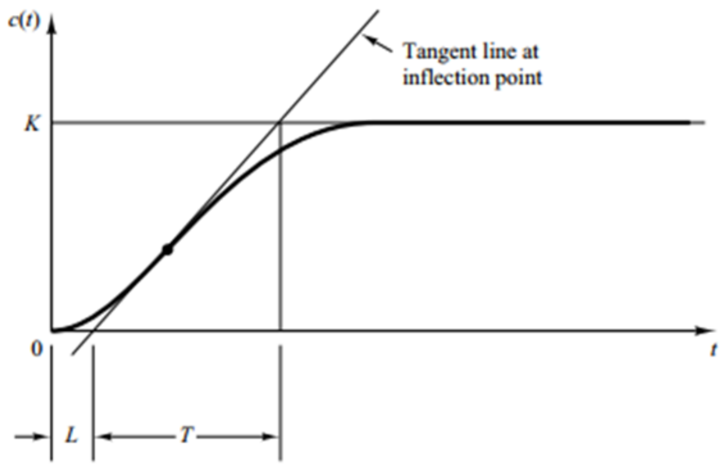
- **Prevent Instability**: Poor tuning can cause dangerous oscillations, shaking, or system failure instead of smooth control.
- **Achieve Performance:** Tuning balances the trade-offs between Speed, Overshoot, and Accuracy.
- **Account for Dynamics:** Every system has unique inertia and response, requiring custom-tailored gains.
- **Ensure Robustness:** Proper tuning makes the system reliable and able to handle disturbances or changes in operating conditions

Introducing the Ziegler-Nichols (Z-N) Tuning Methods

- Two classic empirical methods:



Method 1: Ultimate Cycle Method (Closed-Loop)



Method 2: Reaction Curve Method (Open-Loop)

Fig 10: Z-N tuning methods [10]

Z-N Method 1: The Ultimate Cycle Method

- **Goal:** Find the ultimate gain K_u where the system oscillates continuously.

Steps:

- Set controller to P-only (K_i=0, K_d=0).
- Increase K_p until sustained oscillations occur. This is K_u.
- Measure the oscillation period P_{cr}.

Step 1: Set controller to P-only

- Why remove I and D?
  - **Integral Action:** Can cause saturation and hide the true ultimate gain.
  - **Derivative Action:** Adds damping, which also masks the ultimate gain.
  - We want to find the natural stability boundary of the process.

Step 2: Find Ultimate Gain (K_{cr})

- Gradually increase the proportional gain K_p.
- K_{cr} is the specific gain where the output exhibits sustained, constant-amplitude oscillations.

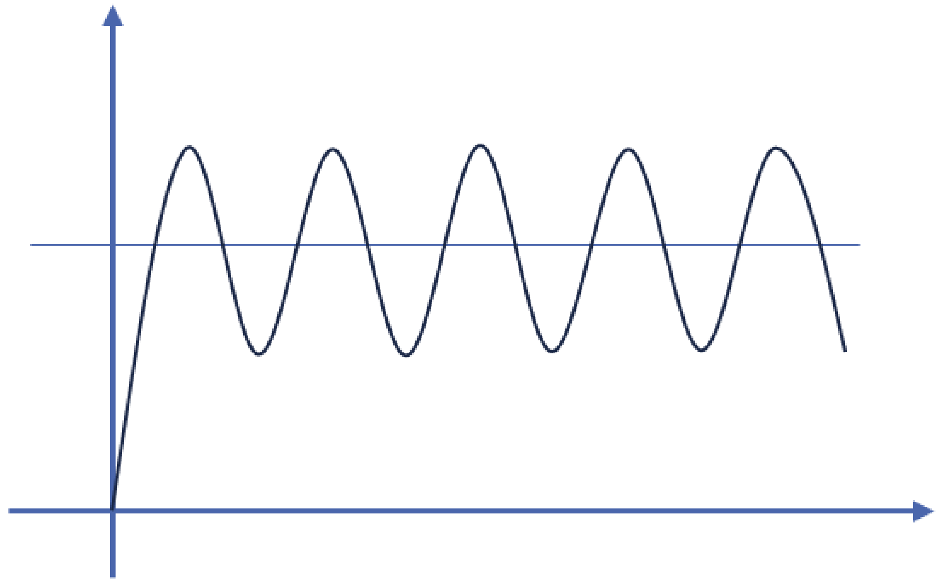


Fig 11: Finding ultimate gain. [11]

Step 3: Measure Ultimate Period (P_{cr})

- From the sustained oscillation plot, measure the time for one complete cycle.
- This is the Ultimate Period, P_{cr}.

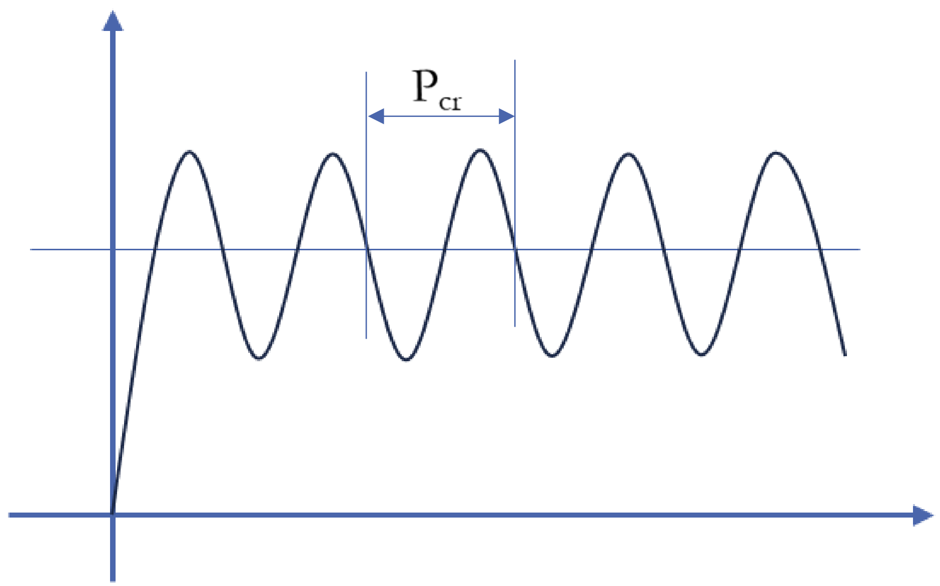


Fig 12: Ultimate period [12]

Step 4: Use Z-N Table

- Use K_{cr} and P_{cr} to calculate PID parameters.
- Ziegler-Nichols Tuning Table (Ultimate Cycle Method):

Controller Type	K_p	T_i	T_d
P	$0.5\ K_{cr}$	-	-
PI	$0.45\ K_{cr}$	$P_{cr}\ /\ 1.2$	-
PID	$0.6\ K_{cr}$	$P_{cr}\ /\ 2$	$P_{cr}\ /\ 8$

Note: $K_i = K_p\ /\ T_i,\ \ K_d = K_p * T_d$

Example: Ultimate Cycle Method

Given: A process transfer function (e.g., $1 / (s(s+1)(s+5))$).

- Task: Find K_{cr} and P_{cr} and calculate PID gains.

- **Method:** Analytical (Routh-Hurwitz) or Simulation (e.g., in MATLAB/Simulink).
- Let's use Routh-Hurwitz
- $1+KG(s)=s^3+6s^2+5s+K$

s^3	1	5
s^2	6	K
s^1	$\frac{30-K}{6}$	0
s^0	K	

Example: finding K_{cr}

- For marginal stability s^1 row must be zero

$$\frac{30 - K}{6} \Longrightarrow K = K_{cr} = 30$$

Finding ultimate period P_{cr}

$6s^2 + K_{cr} = 0$

$s^2 = 30 \rightarrow s = \pm j\sqrt{5}$ rad/sec $= \omega_{cr}$

$$P_{cr} = \frac{2\pi}{\omega_{cr}} = \frac{2\pi}{\sqrt{5}} = 2.81 sec$$

Step 4: Use Z-N Table

- K_{cr}=30 and P_{cr}=2.81

Controller Type	K_p	T_i	T_d
P	0.5 K_{cr}=15	-	-
PI	0.45 K_{cr}=13.5	P_{cr} / 1.2=2.34	-
PID	0.6 K_{cr}=18	P_{cr} / 2=1.4	P_{cr} / 8=0.35

If we are asked for PID setting K_p=18, K_i=12.86, and K_d=6.3

Note: $K_i = K_p / T_i, \; K_d = K_p * T_d$

Z-N Method 2: Process Reaction Curve

Open-Loop Method.

- Step 1: Apply a step input to the open-loop process, and record the output response ("S-shaped" curve).
- Step 2: Draw a diagonal line tangent to the response curve
- Step 3: Mark L and T
- Step 4: Use table to adjust the setting.

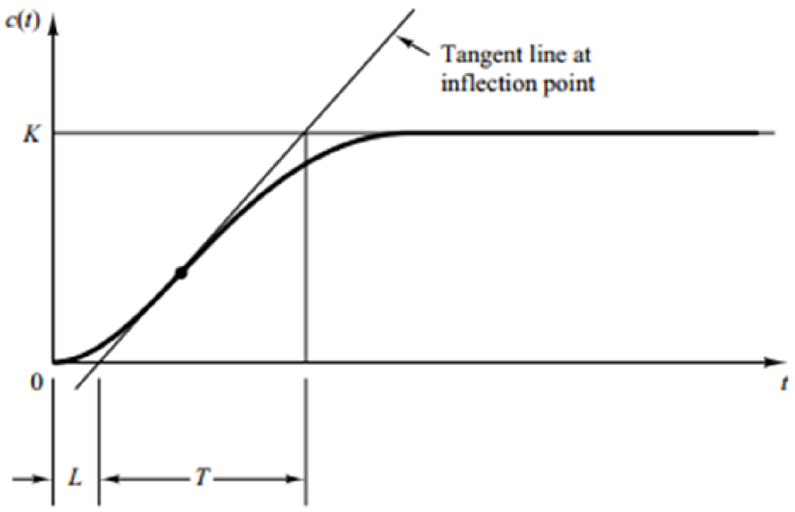


Fig 13: Process reaction curve. [13]

Z-N Method 2: Process Reaction Curve Continued

- Step 1: Apply a step input to the open-loop process, and record the output response ("S-shaped" curve).
- Step 2: Draw a diagonal line tangent to the response curve
- Step 3: Mark L and T

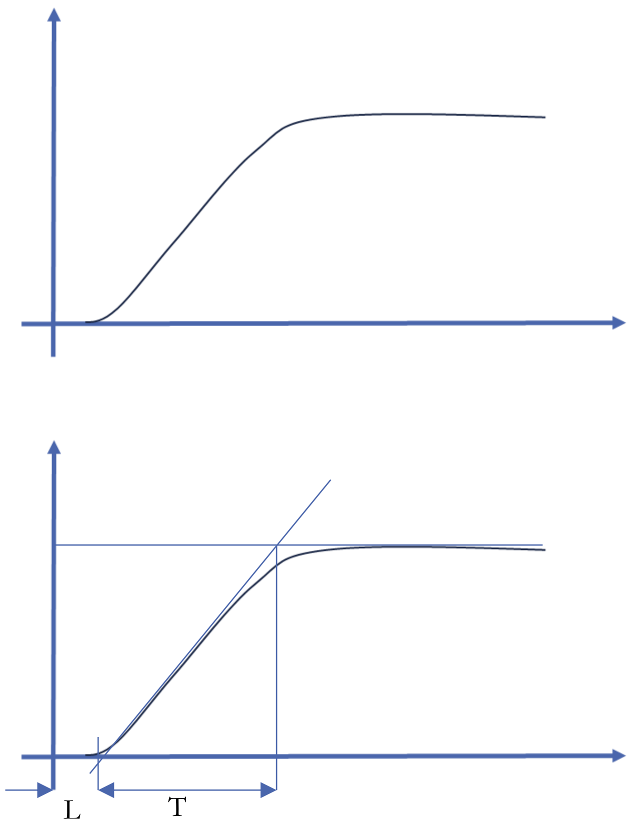


Fig 14: Sketching reaction curve and marking the points [14]

Step 4: Use table to adjust the setting.

- Use T and L to calculate PID parameters.

Controller Type	K_p	K_i	K_s
P	T/L	-	-
PI	0.9 T/L	(0.9T/L) (0.3/L)	-
PID	1.2 T/L	(1.2T/L)(0.5/L)	0.6T

Comparison of Z-N Methods

- **Pros:**
  - Simple, model-free (or simple model), widely used.
- **Cons:**
  - Aggressive Tuning: Results in a quarter-amplitude decay ratio (~25% overshoot).
  - Not suitable for all processes (e.g., unstable, large dead time).

When to use P, PI, or PID?

Controller Type	When to Use It
P	• Simple systems where a small, constant error is acceptable. • When the process cannot have integral action • For very fast, stable responses where offset doesn't matter.
PI	• Most common in industry. • When steady-state error must be zero. • When the process is noisy
PID	• Systems with slow response and significant inertia or lag. • When you need faster response and less overshoot than PI can provide. • For high-performance, precision systems.

Real-World PID

- **Setpoint Weighting:** Apply derivative and proportional action only on the process variable (PV) to avoid "derivative kick" on setpoint changes.
- **Bump less Transfer:** Smooth switching between manual and automatic control modes.

Beyond Ziegler-Nichols

- Internal Model Control (IMC) Tuning:
  - Model-based method.
  - Provides a tuning parameter (λ) to directly trade-off performance vs. robustness.
  - Generally produces better responses than Z-N.

Common Pitfalls in PID Tuning

- Too much Integral gain → Oscillations and windup.
- Using Derivative on a noisy signal →Excessive control action.
- Not understanding the process physics.

Case Study

- **System:** DC Motor Speed Control.
- **Requirements:** Fast response, zero steady-state error, minimal overshoot.
- **Design Choice:** PI or PID controller.
- **Tuning Process:** Use Reaction Curve method or simulation to find initial gains, then fine-tune.

Summary of Key Takeaways

- PID combines Past (I), Present (P), and Future (D) error.
- P reduces error, I eliminates offset, D reduces overshoot.
- Ziegler-Nichols provides a systematic way to tune.
- Be aware of windup, noise, and other practical issues.

References

[1] Chalachew Werku, 2025, Feedback control system, Self-created

[2] Harder you push, the closer you get, https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcREwpemwf0Yu88xsWTRg6sZOGFo9YNn7dD6DQ&s, accessed on October 28, 2025

[3] Chalachew Werku, 2025, Effect of different proportional gain, Self-created

[4] Accumulating past errors, https://www.shutterstock.com/image-illustration/snowball-effect-metaphor-business-concept-260nw-2386637099.jpg, accessed on October 28,2025

[5] Chalachew Werku, 2025, P vs PI controller, Self-created

[6] Integral windup, https://www.google.com/url?sa=i&url=https%3A%2F%2Fcontrolguru.com%2Fintegral-reset-windup-jacketing-logic-and-the-velocity-pi-form%2F&psig=AOvVaw0CmEgf5W_E9iFhysiUhejQ&ust=1762087140998000&source=images&cd=vfe&opi=89978449&ved=0CBUQjRxqFwoTCIjektL80JADFQAAAAAdAAAAABAE, accessed on October 28,2025

References Continued

[7] Chalachew Werku, 2025, D-control, rate of change of error, Self-created

[8] Chalachew Werku, 2025, P vs PD controller, Self-created

[9] Chalachew Werku, 2025, Effect of P, PI and PID controller, Self-created

[10] Chalachew Werku, 2025, Z-N tuning methods, Self-created

[11] Chalachew Werku, 2025, Finding ultimate gain, Self-created

[12] Chalachew Werku, 2025, Ultimate period, Self-created

[13] Chalachew Werku, 2025, Process reaction curve, Self-created

[14] Chalachew Werku, 2025, Sketching reaction curve and marking the points, Self-created

- Astrom, K.J., and Hagglund, T. "PID Controllers: Theory, Design, and Tuning, International Society of Automation, August 8, 2005"
- Ogata, K. "Modern Control Engineering. (2010, Pearson)"
- Norman S. Nise – Control Systems Engineering (2015, Wiley)