PID Control

A Comprehensive Guide to PID Control Systems

CS637: EMBEDDED AND CYBER-PHYSICAL SYSTEMS

Instructor - Indranil Saha

October 28, 2024

Table of Contents I

Introduction

- **PID Controller:** The most widely used control algorithm in industry.
  - Governs the majority of feedback loops in basic form or with minor variations.
- **Implementations:**
  - Standalone devices
  - Direct Digital Control (DDC) systems
  - Hierarchical distributed process control systems
- **Chapter Overview:**
  - **Fundamentals:** Introduction to PID control, fundamental algorithm, and representations.
  - **Controller Behavior:** Intuitive discussion of closed-loop system behavior.
  - **Integral Windup:** Addressing integral windup caused by actuator saturation and prevention methods.

The Principle of Feedback in Control Systems

- **Feedback in Control Systems:**
  - Fundamental concept enabling systems to maintain desired performance despite external disturbances and internal variations.
  - Illustrated by a simple feedback control loop (see Figure 1).

Negative Feedback

- Controller's response opposes the disturbance.
- Essential for stabilizing the system and accurate setpoint tracking.

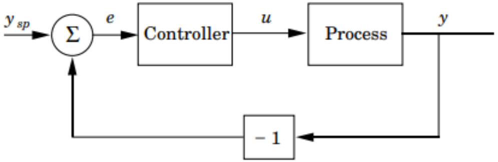

Figure: Block Diagram of a Process with a Feedback Controller

Proportional Action

- **Definition:**
  - Control action proportional to the current error $e(t)$.
  - Simplified PID control law:

    $$u(t) = K_p e(t) + u_b$$

    where K_p is the proportional gain and u_b is the bias.
- **Characteristics:**
  - Reduces oscillations compared to on-off control.
  - Introduces steady-state error.
  - Response improves with higher K_p, but excessive K_p can cause instability.
- **Bias (u_b):**
  - Often set to midpoint of control signal range:

    $$u_b = \frac{u_{\max} + u_{\min}}{2}$$

  - Can be adjusted to eliminate steady-state error at a specific setpoint.

Integral Action

- **Purpose:**
  - Eliminate steady-state error between process output y and setpoint y_{sp}.
- **Control Law:**

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(\tau) d\tau$$

  where T_i is the integral time constant.
- **Characteristics:**
  - Accumulates error over time to adjust control signal.
  - Ensures $e(t)$ approaches zero in steady state.
  - Can introduce slower response and potential oscillations.
- **Steady-State Behavior:**
  - In steady state, $e(t) = 0$ ensures $u(t) = u_b$.
  - Integral action adjusts u_b to eliminate persistent errors.

Derivative Action

- **Purpose:**
  - Enhance stability and improve transient response of the control system.
  - Predict future errors based on the rate of change of the current error.
- **Control Law:**

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt}$$

  where K_d is the derivative gain.
- **Characteristics:**
  - Dampens oscillations and reduces overshoot.
  - Sensitive to high-frequency noise; requires careful tuning.
  - Acts as a predictive mechanism to anticipate future errors.
- **Behavior:**
  - Uses the derivative (rate of change) of the error to adjust $u(t)$.
  - Can be interpreted using a Taylor series expansion:

$$e(t + T_d) \approx e(t) + T_d \frac{de(t)}{dt}$$

  - Control signal becomes proportional to an estimate of the error T_d seconds into the future.

PID Controller Variations

- **Standard (Non-Interacting / Parallel) PID Controller:**

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

- **Interacting (Series) PID Controller:**

$$G_P(s) = K_P\left(1 + \frac{1}{sT_{Pi}}\right)(1 + sT_{Pd})$$

3.png

4.png

Conversion Between Forms & Comparison

- **Conversion from Interacting to Non-Interacting Form:**
  - $$K = K_P \left(\frac{T_{Pi} + T_{Pd}}{T_{Pi}} \right)$$
  - $$T_i = T_{Pi} + T_{Pd}$$
  - $$T_d = \frac{T_{Pi} T_{Pd}}{T_{Pi} + T_{Pd}}$$
- **Condition for Conversion:**
  - $T_i \geq 4 T_d$
  - Ensures stability during conversion

Integrator Windup in PID Control

- **Key Concepts:**
  - Integrator windup occurs when the control output exceeds actuator limits, leading to continuous accumulation of error in the integral term.
  - Effects of windup include prolonged overshoot, oscillations, delayed settling, and reduced control accuracy.

- **Illustration:**
  - The figure below shows the impact of integrator windup on system behavior, highlighting the oscillations and overshoot caused by accumulated integral action.

fig9.jpg

Anti-Windup Mechanisms

- **Methods to Prevent Windup:**
  - **Integrator Clamping:** Temporarily holds the integrator value when saturation is detected.
  - **Back-Calculation:** Adjusts the integrator based on the difference between desired and actual actuator output.
  - **Incremental Algorithms:** Applies incremental control changes, reducing the risk of windup.
  - **Tracking Mode in Digital Controllers:** Dynamically adjusts the integrator based on actuator behavior.
- **Benefits of Anti-Windup Mechanisms:**
  - Improved stability and control performance
  - Reduced oscillations and overshoot
  - Faster settling times

Incremental Algorithms and Back-Calculation

- **Early Development:** Integral action was initially integrated directly with actuators, using motor-driven valves to prevent windup by stopping integration at limits.
- **Digital Transition:** Digital control systems retained analog-like setups, leading to velocity algorithms for incremental control signal changes to avoid windup.
- **Back-Calculation:**
  - When actuator output saturates, back-calculation adjusts the integral term with time constant T_t and feedback gain $\frac{1}{T_t}$.
  - This adjustment resets the integrator at a rate controlled by T_t.
  - Recommended T_t range: $T_d < T_t < T_i$; typically, $T_t = \sqrt{T_i T_d}$.
- **Steady-State Behavior:**
  - At steady state, $e_s = 0$, enabling normal operation without affecting the integrator.
  - During saturation, $e_s \neq 0$ resets the integrator, keeping output within limits.

Tracking Mode and Applications

- **Tracking Mode:**
  - Controllers operate in two modes:
    - **Normal Mode:** Standard PID response to setpoint and output.
    - **Tracking Mode:** Aligns output with a tracking signal, dynamically adjusting as needed.
  - Tracking is inhibited when tracking signal w equals output v, ensuring stability without manual mode switching.
- **Benefits of Tracking:**
  - Enhanced stability with automatic adjustments.
  - Simplified integration for complex control strategies.
  - Improved performance by adapting to external signals.
- **Applications:**
  - **Cascade Control:** Enables inner loop controllers to follow outer loop signals.
  - **Override Control:** Adjusts output based on predefined safety or performance signals.
  - **Setpoint Scheduling:** Allows smooth setpoint changes to reduce sudden control adjustments.

PID Tuning Methods

- **Tuning Overview:** Ziegler-Nichols methods set PID parameters for effective control, focusing on load disturbances, noise, and process dynamics.
- **Step Response Method:**
  - Characterizes open-loop step response to find a and L.
  - Steps:
    - Perform a step test on the open-loop system.
    - Identify max slope; draw tangent to find a and L.
    - Use a and L to calculate controller parameters (Table 1.1).
    - Estimate closed-loop period T_p.
- **Advantages:** Simple, widely used; provides baseline PID parameters.

1.9.1.png

Frequency Response Method

- **Frequency Response Method:**
  - Uses system oscillations in proportional-only mode.
  - Steps:
    - Set PID controller to proportional mode ($T_i = \infty$, $T_d = 0$).
    - Increase K until sustained oscillations occur.
    - Record K_u (ultimate gain) and T_u (ultimate period).
    - Use K_u and T_u to set PID parameters (Table 1.2).
- **Example:**
  - If $K_u = 2$ and $T_u = 5$:
  - $K = 0.6 \times 2 = 1.2$, $T_i = 0.5 \times 5 = 2.5$ sec, $T_d = 0.125 \times 5 = 0.625$ sec
- **Advantages:** Good for dynamic processes; adjusts parameters based on oscillations.

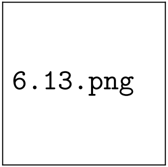

Figure: Frequency Response Method for PID Tuning

Digital Implementation of PID Controllers

- **Discretization:** Continuous PID terms are approximated at discrete sampling instants $\{t_k\}$.
- **Proportional Action:**

$$P(t_k) = K(b \cdot y_{sp}(t_k) - y(t_k))$$

- **Integral Action (Forward Difference):**

$$I(t_{k+1}) = I(t_k) + \frac{Kh}{T_i} e(t_k)$$

- **Derivative Action (Backward Difference):**

$$D(t_k) = \frac{T_d}{T_d + Nh} D(t_{k-1}) - \frac{KT_d N}{T_d + Nh}(y(t_k) - y(t_{k-1}))$$

- **Discrete PID Control Signal:**

$$u(t_k) = P(t_k) + I(t_k) + D(t_k)$$

Velocity Algorithms in PID Control

- **Overview:** Velocity algorithms output the rate of change, ideal for systems with external integrators (e.g., motor-driven).
- **Incremental Form:**

$$\Delta u(t_k) = \Delta P(t_k) + \Delta I(t_k) + \Delta D(t_k)$$

  - Each term (P, I, D) calculated as an increment, allowing efficient computations.
- **Advantages:** Reduced computational load, easy anti-windup, compatible with legacy systems.
- **Non-Integral Controllers:** Uses offset term u_b to hold steady state:

$$\Delta u(t) = Ke(t) + u_b - u(t-h)$$

Feedforward Control with Feedback Integration

- **Concept:** Feedforward (u_{FF}) preempts disturbances, added to feedback (u_{FB}) for combined control:

$$u = u_{FB} + u_{FF}$$

- **Role of Feedforward and Feedback:** - Feedforward handles predictable disturbances; feedback corrects unexpected errors.
- **Anti-Windup Strategy:** Applies to total u to avoid windup when feedforward signals are added.
- **Incremental Approach:**

$$\Delta u = \Delta u_{FB} + \Delta u_{FF}$$

$$u(t) = u(t-h) + \Delta u(t)$$

- Smooth control updates; minimizes windup by synchronizing feedback and feedforward changes.

Applications of PID Control

- **General Use:** Widely used in industry for its simplicity and effectiveness in setpoint tracking, disturbance rejection, and noise handling.
- **When PI Control is Sufficient:**
  - Suitable for systems with first-order dynamics or those with less stringent precision needs.
  - Effective in eliminating steady-state error and providing adequate transient response.
  - Indicators: First-order step response, or Nyquist plot within first and fourth quadrants.
- **When PID Control is Ideal:**
  - Best for systems with second-order dynamics or multiple time constants.
  - Common applications:
    - **Varying time constants**: Derivative action enhances stability.
    - **Temperature control**: Derivative action dampens oscillations, speeds up settling.
    - **High-precision control**: Derivative action improves damping, allowing for higher gains.

Limitations of PID Control

- **Higher-Order Processes:** PID struggles with systems beyond second-order, which may lead to suboptimal performance or instability.
- **Systems with Long Dead Time:** Delays between control action and effect can cause PID to respond too late, leading to oscillations.
- **High-Frequency Noise Sensitivity:** Derivative term amplifies noise, requiring careful tuning and possibly extra filtering.
- **Alternatives:** For these cases, consider advanced controls like Model Predictive Control (MPC) or adaptive strategies to improve stability and performance.

Conclusion

- **PID Controllers:** Widely applicable, offering robust and simple control for many systems.
- **Proportional Control:** Reduces oscillations; may leave steady-state error.
- **Integral Control:** Removes steady-state error; can slow response.
- **Derivative Control:** Improves stability and transient response; sensitive to noise.
- **Tuning Methods:** Essential for optimal performance; includes Step and Frequency Response methods.
- **Digital Implementation:** Requires discretization for continuous-time control.
- **Applications and Limitations:** Versatile but may need alternative approaches for complex dynamics.

Future Directions:

- Integrating advanced control techniques and complex systems.
- Enhancing digital methods for precision and stability.

Thank You!

Questions?