PID CONTROLLER

Author
Anshul Jain
University of Colorado at Boulder
2024

Contents

1 Introduction

A PID Controller (Proportional-Integral-Derivative Controller) is the most widely used control strategy in industrial applications. It works by adjusting the output based on the error between reference value and measured output.

1.1 Components of a PID Controller

The PID Controller has three components to it, which work together to control the process variable.

1. Proportional (P)
   - Produces an output that is proportional to the current error value.
   - The equation for the proportional component is:
     $$P = K_p * e(t),$$
     where, e(t) is the error at time t, and K_p is the Proportional Gain.
   - The proportional term works to reduce the error, but it does not necessarily drive the error to zero.
   - A high K_p value reduces the steady-state error, but very high K_p values lead to overshooting and oscillations.
2. Integral (I)
   - The integral term aims to eliminate the steady-state error by integrating the error signal, by finding the cumulative sum of past errors over time.
   - The equation for the integral component is:
     $$I = K_i * \int e(t)dt,$$
     where, K_i is the Integral Gain.
   - The integral term can reduce the error to zero over time, however small or persistent the error is.
   - However, integral wind-up can lead to constantly increasing output if the cumulated error integral is too large. This requires the use proper anti-windup mechanisms.
3. Derivative (D)
   - The derivative term affects the rate of change of error. This gives us control on future error behavior and helps prevent overshoot.
   - The equation for the derivative component is:
     $$D = K_d * \frac{d}{dt}e(t),$$
     where, K_d is the Derivative Gain.

- The derivative term stabilizes the system by controlling the rate of change of errors and helps reduce overshoot.
- It can also amplify high-frequency noise in the system, which may require filtering.

2 PID Control Law

The control law for PID controller gives the control signal u(t) as:

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

where, u(t) is the control output that will be applied back to the system.

3 PID Controller in Industry

A PID Controller uses the above given law to adjust the control output in real-time based on the error signal.

3.1 Error Calculation

Error is calculated at each time step, where the controller calculates the error as the difference between the reference value and the measured value.

3.2 Proportional Response

The controller adjusts the output by an amount proportional to the error. If the error is large, the controller responds quickly. However, this term alone might not always bring the system to the setpoint perfectly.

3.3 Integral Response

Over time, the integral term sums the errors and applies a correction based on the accumulated error. This ensures that the steady-state error (constant bias or offset) is eliminated. For example, if a system always lags by a small amount, the integral term will increase the controller output until the error is zeroed.

3.4 Derivative Response

The derivative term anticipates future errors by evaluating the rate of change of the error. This helps in stabilizing the system by slowing down the response before it overshoots the setpoint, reducing oscillations.

4 Tuning PID Controller

In simple words, tuning any controller means to select the appropriate parameters that govern the control law of the controller in order to achieve the desired system performance.

To tune a PID controller, we will have to select appropriate values for K_p, K_i, and K_d. There are several methods to tune a PID Controller:

1. Manual Tuning
   - Start with K_i and $K_d = 0$, and increase K_p until the system starts to respond to changes in the input.
   - Then adjust K_i to eliminate error.
   - Finally, introduce K_d to reduce overshoot and oscillations.
2. Ziegler-Nicholas Method
   - This method is similar to manual tuning in the sense that we increase K_p first. In this case, K_p is increased until the system reaches steady oscillations, which is the case of Critical Gain.
   - Then estimate the appropriate values of K_p, K_i, and K_d based on the critical gain.
3. Software-based Tuning
   - Many modern software solutions are available that offer adaptive PID tuning which adjust the PID parameters in real-time based on the system feedback.

5 Practical Considerations

Some things to keep in mind while dealing with a PID controller (almost all the time) are:

5.1 Anti-Windup

Wind-up, in the first place, occurs when the system output saturates and cannot go beyond its maximum/minimum limits, and still the integral term keeps accumulating the error signal. The system cannot respond to the 'extra' accumulated error signal leading to instability. As a result, anti-windup mechanisms are incorporated.

5.2 Noise and Derivative Term

We have seen that the derivative term can amplify high-frequency noise, especially if the sensors are noisy. To mitigate this, the derivate term needs to be filtered, usually a Low-Pass Filter is used.

5.3 Sampling Time

The PID Controller is sampled at discrete times which is selected by the control system designer. If the sampling time is too large, system response will be very slow, on the other hand, if the sampling time is very less, the system can get unstable due to excessive control actions.

5.4 Performance Metrics

To validate the design of the controller, some performance metrics need to be balanced. Usually used parameters are:

- Sampling Time
- Overshoot
- Rise time
- Steady-state error

The aim of the controller is to bring the system to the setpoint (desired system performance) quickly while minimizing overshoot and oscillations.

5.5 Application-Specific Adjustments

It is to be kept in mind that control system design is specific to a system, and there is no perfect control algorithm for any one system. The parameters for every system vary and accordingly the controller needs to be adjusted. Example: a temperature control system has the time constant of the thermal system as an important parameter, a speed control system has inertia as an important parameter.

6 Example Industrial Applications

6.1 Temperature Control

In a thermal system, a PID Controller can be used to adjust the heating element's power to keep the temperature at the desired value, compensating for fluctuations in the ambient temperature and heat loss.

6.2 Motor Speed Control

In a motor, a PID Controller can be used to adjust either the current or voltage supplied to the motor to maintain its speed at a constant rpm, compensating for load changes and friction.

6.3 Flow Control

In industrial chemical processes or fluid transport systems, a PID Controller can be used to adjust the valve positions to maintain the fluid flow rate constant, compensating for pressure fluctuations or resistances.

6.4 Position Control

In mechanical or robotic systems, a PID Controller can be used to adjust the position of any actuator to reach the setpoint, compensating for external forces such as friction.

7 Advanced PID Variants

Some industrial applications require advanced control system design. Some advanced versions of a PID Controller can be:

1. PI (Proportional-Integral) Controller: In systems where the derivative term introduces too much noise, only a PI Controller is used ($K_d = 0$) to control the system. Quick response time is more crucial in such cases compared to fine-tuned damping.

2. PD (Proportional-Derivative) Controller: Some systems do not actually have any steady-state error, such as position control system. A PD Controller is sufficient in such systems.

3. Adaptive PID Controller: In cases where the dynamics of the system change with time, an adaptive PID controller is used such that it adjusts it gains in real-time based on feedback from the system and environment.