Discrete PID Controllers

Contents

This exercise sheet follows the Chapter 6 lecture notes on PID controllers. The tasks move from discrete calculus and z-transforms to the discrete PID transfer function, the incremental PID form, simulation of PI control, closed-loop pole placement, and integral windup with back-calculation anti-windup.

Task A — Warm-up: z-transform of a discrete integral and derivative

Let (e_k) *be an error sequence and let* T_s *be the sampling period. Use one-sided z-transforms and assume zero initial conditions, in particular* $I_{-1} = 0$ *and* $e_{-1} = 0$*.*

The discrete integral is defined by backward Euler:

$$I_k = I_{k-1} + T_s e_k.$$

The discrete derivative is defined by backward difference:

$$D_k = \frac{e_k - e_{k-1}}{T_s}.$$

1. *Apply the z-transform to the integral equation and derive* $\frac{I(z)}{E(z)} = \frac{T_s z}{z-1}$.
2. *Apply the z-transform to the derivative equation and derive* $\frac{D(z)}{E(z)} = \frac{z-1}{T_s z}$.
3. *Explain in one sentence why the derivative formula is causal, while the forward-Euler derivative* $D_k = \frac{e_{k+1} - e_k}{T_s}$ *is not causal.*

Task B — Derive the PID transfer function

The discrete-time PID controller is

$$u_k = K_p e_k + K_i I_k + K_d D_k,$$

where

$$I_k = I_{k-1} + T_s e_k, \quad D_k = \frac{e_k - e_{k-1}}{T_s}.$$

1. *Use the results from Exercise 1 to derive the transfer function* $C(z) = \frac{U(z)}{E(z)}$.
2. *Show that*

$$C(z) = K_p + K_i \frac{T_s z}{z-1} + K_d \frac{z-1}{T_s z}.$$

3. *Identify which part of* $C(z)$ *corresponds to the proportional, integral, and derivative contributions.*

Task C — From position form to incremental form

The PID controller in position form is

$$u_k = K_p e_k + K_i T_s \sum_{j=0}^{k} e_j + K_d \frac{e_k - e_{k-1}}{T_s}.$$

1. *Write the same equation for* u_{k-1}.
2. *Subtract* u_{k-1} *from* u_k.
3. *Show that the controller can be written as*

$$u_k = u_{k-1} + K_p(e_k - e_{k-1}) + K_i T_s e_k + K_d \frac{e_k - 2e_{k-1} + e_{k-2}}{T_s}.$$

4. *Explain what previous values must be stored in memory to implement this incremental form.*

Task D — Simulate a PI controller on a first-order plant

Consider the plant

$$x_{k+1} = 0.95x_k + 0.15u_k, \quad y_k = x_k.$$

The reference is constant, $r_k = 10$, and the initial condition is $x_0 = 5$.

A PI controller is used:

$$u_k = K_p e_k + K_i I_k, \quad e_k = r_k - y_k,$$

with

$$I_k = I_{k-1} + T_s e_k, \quad T_s = 1.$$

Use

$$K_p = 1.5, \quad K_i = 0.3,$$

and assume $I_{-1} = 0$.

1. *Compute e_0, I_0, u_0, and x_1.*
2. *Continue the computation for $k = 1, 2, 3$.*
3. *Explain why the integral term helps remove steady-state error.*
4. *Compare this with a pure proportional controller $u_k = K_p e_k$. Does the pure proportional controller necessarily reach $y = 10$? Explain.*

Task E — Closed-loop design in discrete time

You are given the unstable discrete-time system

$$G(z) = \frac{5}{z - 1.2}.$$

You want to control it using a PI controller. The discrete-time PI controller is

$$C(z) = K_p + K_i \frac{Tz}{z - 1},$$

where T is the sampling period.

You would like the poles of the closed-loop system to be placed at

$$z_1 = 0.4, \quad z_2 = 0.5.$$

Assume unity negative feedback.

> *Hint*
>
> *Because the plant is first order and the PI controller adds one integrator pole at $z = 1$, the closed-loop system becomes second order. That is why we choose two desired closed-loop poles instead of one.*

1. *Write the closed-loop transfer function*

   $$\frac{Y(z)}{R(z)} = \frac{C(z)G(z)}{1 + C(z)G(z)}.$$

2. *Derive the characteristic equation of the closed-loop system.*
3. *Find K_p and K_i such that the closed-loop poles are at 0.4 and 0.5.*
4. *Write the controller as a difference equation that computes u_k from u_{k-1}, e_k, and e_{k-1}.*
5. *Can this controller be implemented? Explain why.*

Task F — Saturation and integral windup

Consider the saturated plant

$$x_{k+1} = 0.9x_k + 0.1u_k^{\mathrm{sat}}, \quad y_k = x_k,$$

where

$$u_k^{\mathrm{sat}} = \mathrm{clip}(u_k, -1, 1).$$

The reference is $r_k = 0.5$ *and the initial state is* $x_0 = 0$.

A PI controller is used:

$$u_k = K_p e_k + K_i I_k,$$

with

$$K_p = 5, \quad K_i = 1, \quad T_s = 1.$$

The integrator is updated by

$$I_k = I_{k-1} + T_s e_k,$$

with $I_{-1} = 0$.

1. *Compute* e_0, I_0, u_0, u_0^{sat}, *and* x_1.
2. *Continue for* $k = 1, 2, 3$.
3. *Explain why the actuator saturates even though the steady-state input needed to hold* $y = 0.5$ *is only* $u = 0.5$.
4. *Explain why the integrator can become too large during saturation.*
5. *Now replace the integrator update with back-calculation:*

   $$I_k = I_{k-1} + T_s e_k + K_{\mathrm{antiwindup}}(u_k^{\mathrm{sat}} - u_k),$$

   with $K_{\mathrm{antiwindup}} = 0.5$. *Compute the first two iterations again and compare the integrator values with the no-anti-windup case.*