Discrete PID Controller

Rishav

November 2023

Preface

The equation of PID controller consists of integration and derivative of error with respect to time. The ideas of integration and derivative are based on *infinitesimally* small time period as prescribed by the theory of calculus which is the study of *continuous change*. Ideally such continuous forms cannot be realized in software because two consecuitive function calls of PID algorithm does not occur in infinitesimally small time step but on certain, finite, time intervals.

This is an attempt to document the discretization of PID controller for software implementation along with other practical considerations.

Discrete PID

PID controller in continuous form is

$$u(t) = k_p e(t) + k_i \int_0^t e(t)\,dt + k_d \frac{de(t)}{dt}$$

where $u(t)$ is output of PID, $e(t)$ is control error (i.e. desire - feedback), and k_p, k_i, and k_d are positive control gains.

Applying Laplace transform, we get the transfor function of PID in continuous complex domain as function of s. i.e.

$$\frac{U(s)}{E(s)} = k_p + k_i \frac{1}{s} + k_d s.$$

To discretize above equation, we need to transform it from continuous s-domain to discrete z-domain. We will do so using bilinear transformation (or Tustin's method) which relates s and z as

$$s = \frac{2}{T}\frac{z-1}{z+1}.$$

Now the PID equation in s-domain can be rewritten as

$$\begin{aligned}
\frac{U(z)}{E(z)} &= k_p + k_i \frac{T}{2}\frac{z+1}{z-1} + k_d \frac{2}{T}\frac{z-1}{z+1} \\
&= \frac{2Tk_p(z-1)(z+1) + T^2 k_i (z+1)^2 + 4k_d(z-1)^2}{2T(z-1)(z+1)} \\
&= \frac{2Tk_p(z^2-1) + T^2 k_i (z^2+2z+1) + 4k_d(z^2-2z+1)}{2T(z^2-1)} \\
&= \frac{2Tk_p(1-z^{-2}) + T^2 k_i (1+2z^{-1}+z^{-2}) + 4k_d(1-2z^{-1}+z^{-2})}{2T(1-z^{-2})}
\end{aligned}$$

Inverse z-transform converts a function in z-domain to corresponding time domain sequence that we can actually implement in software. If $X(z)$ is a function in z-domain, then its inverse z-transform is expressed as

$$\mathcal{Z}^{-1}\{X(z)\} = x[n]$$

where n is discrete time index.

It is observed that numerator and denominator are divided by z^2 to have all z terms in negative power. This is because it makes inverse z-transform easier because inverse transform of z^{-1} is a delayed unit step response. i.e.

$$\mathcal{Z}^{-1}\{z^{-r}\} = u[n-r] \quad \text{where } r = 1, 2, 3\dots.$$

This implies that

$$\mathcal{Z}^{-1}\{z^{-1}X(z)\} = x[n-1].$$

Finally we are ready to transform PID algorithm to discrete time domain.

$$\begin{aligned}
U(z)(1-z^{-2})2T &= E(z)\left(2Tk_p(1-z^{-2}) + T^2k_i(1+2z^{-1}+z^{-2}) + 4k_d(1-2z^{-1}+z^{-2})\right) \\
U(z)(1-z^{-2})2T &= E(z)\left((2Tk_p + T^2k_i + 4k_d) + (2T^2k_i - 8k_d)z^{-1} + (-2Tk_p + T^2k_i + 4k_d)z^{-2}\right) \\
2Tu[n] - 2Tu[n-2] &= (2Tk_p + T^2k_i + 4k_d)e[n] + (2T^2k_i - 8k_d)e[n-1] + (-2Tk_p + T^2k_i + 4k_d)e[n-2] \\
2Tu[n] &= 2Tu[n-2] + (2Tk_p + T^2k_i + 4k_d)e[n] + (2T^2k_i - 8k_d)e[n-1] + (-2Tk_p + T^2k_i + 4k_d)e[n-2] \\
u[n] &= u[n-2] + \left(k_p + \frac{T}{2}k_i + \frac{2}{T}k_d\right)e[n] + \left(Tk_i - \frac{4}{T}k_d\right)e[n-1] + \left(-k_p + \frac{T}{2}k_i + \frac{2}{T}k_d\right)e[n-2]
\end{aligned}$$