PID Control

Classical feedback control

John Bagterp Jørgensen

Department of Applied Mathematics and Computer Science
Technical University of Denmark

02619 Model Predictive Control

Closed-loop continuous-time system in the Laplace domain

- Linear process represented by transfer function

$$Y(s) = G(s)U(s) + H(s)D(s)$$

- Controller

$$U(s) = C(s)E(s) \qquad E(s) = \bar{Y}(s) - Y(s)$$

- Closed-loop system

$$\begin{aligned} Y(s) &= G(s)U(s) + H(s)D(s) \\ &= G(s)C(s)E(s) + H(s)D(s) \\ &= G(s)C(s)\left(\bar{Y}(s) - Y(s)\right) + H(s)D(s) \end{aligned}$$

$$\left[I + G(s)C(s)\right]Y(s) = G(s)C(s)\bar{Y}(s) + H(s)D(s)$$

$$\begin{aligned} Y(s) &= \left[I + G(s)C(s)\right]^{-1} G(s)C(s)\bar{Y}(s) + \left[I + G(s)C(s)\right]^{-1} H(s)D(s) \\ &= G_{cl,\bar{Y}}(s)\bar{Y}(s) + G_{cl,D}(s)D(s) \end{aligned}$$

$$\begin{aligned} G_{cl,\bar{Y}}(s) &= \left[I + G(s)C(s)\right]^{-1} G(s)C(s) \\ G_{cl,D}(s) &= \left[I + G(s)C(s)\right]^{-1} H(s) \end{aligned}$$

Ideal PID controller

- Laplace domain

$$U(s) = C(s)E(s) \qquad C(s) = K_c\left(1 + \frac{1}{T_i s} + T_d s\right)$$

- Time domain

$$u(t) = \overbrace{K_c e(t)}^{\text{P}} + \overbrace{\int_0^t \frac{K_c}{T_i} e(\tau) d\tau}^{\text{I}} + \overbrace{K_c T_d \frac{de}{dt}(t)}^{\text{D}}$$

Ideal PID controller - different parametrization

Standard parametrization:

- Laplace domain

$$U(s) = C(s)E(s) \qquad C(s) = K_c \left(1 + \frac{1}{T_i s} + T_d s\right)$$

- Time domain

$$u(t) = K_c e(t) + \int_0^t \frac{K_c}{T_i} e(\tau) d\tau + K_c T_d \frac{de}{dt}(t)$$

Alternative parametrization:

- Laplace domain

$$U(s) = C(s)E(s) \qquad C(s) = K_P + K_I \frac{1}{s} + K_D s$$

- Time domain

$$u(t) = K_P e(t) + \int_0^t K_I e(\tau) d\tau + K_D \frac{de}{dt}(t)$$

- Parameter relation

$$K_P = K_c \quad K_I = \frac{K_c}{T_i} \quad K_D = K_c T_d$$

- Ideal PID controller with general parametrization

$$u(t) = K_P e(t) + \int_0^t K_I e(\tau) d\tau + K_D \frac{de}{dt}(t)$$

- Practical PID controller (continuous time)

$$u(t) = K_P e(t) + \int_0^t K_I e(\tau) d\tau - K_D \frac{dy}{dt}(t)$$

- Discrete time equivalent practical PID controller

$$u_k = K_P e_k + \sum_{i=0}^{k-1} K_I e_i \Delta t - K_D \frac{y_k - y_{k-1}}{\Delta t}$$

Discrete time practical PID controller - implementation

- Discrete time practical PID controller

$$u_k = K_P e_k + \sum_{i=0}^{k-1} K_I e_i \Delta t - K_D \frac{y_k - y_{k-1}}{\Delta t}$$

- Implementation (start with $k = 0$, $I_0 = 0$, $y_{-1} = y_0$)

$$\begin{aligned}
&\text{Given: } \bar{y}_k, y_k, y_{k-1}, I_k, K_P, K_I, K_D, \Delta t \\
&e_k = \bar{y}_k - y_k \\
&P_k = K_P e_k \\
&D_k = -K_D \frac{y_k - y_{k-1}}{\Delta t} \\
&u_k = P_k + I_k + D_k \\
&I_{k+1} = I_k + K_I e_k \Delta t \\
&\text{Return: } u_k, I_{k+1}
\end{aligned}$$

- Remember that the above equations are for deviation variables

Discrete time practical PID controller - implementation

- Physical variables (u_k, y_k) and deviation variables (U_k, Y_k)

$$U_k = u_k - \bar{u}_k \qquad Y_k = y_k - \bar{y}_k$$

- Discrete time practical PID controller - physical variables

$$u_k = \bar{u}_k + K_P e_k + \sum_{i=0}^{k-1} K_I e_i \Delta t - K_D \frac{y_k - y_{k-1}}{\Delta t}$$

- Implementation (start with $k = 0$, $I_0 = 0$, $y_{-1} = y_0$)

$$\begin{aligned}
&\text{Given: } \bar{u}_k, \bar{y}_k, y_k, y_{k-1}, I_k, K_P, K_I, K_D, \Delta t \\
&e_k = \bar{y}_k - y_k \\
&P_k = K_P e_k \\
&D_k = -K_D \frac{y_k - y_{k-1}}{\Delta t} \\
&u_k = \bar{u}_k + P_k + I_k + D_k \\
&I_{k+1} = I_k + K_I e_k \Delta t \\
&\text{Return: } u_k, I_{k+1}
\end{aligned}$$

- Notice that $\bar{u}_k + I_k$ may be interpreted as an update $\bar{u}$ that will give $\bar{y}$

Constraints and anti-windup

- Input constraints

$$u_{\min} \leq u_k \leq u_{\max}$$

- Anti-windup: Only update the integrator when the constraints are not active
- Implementation (start with $k = 0$, $I_0 = 0$, $y_{-1} = y_0$)

Given: $\bar{u}_k, \bar{y}_k, y_k, y_{k-1}, I_k, K_P, K_I, K_D, \Delta t, u_{\min}, u_{\max}$

$e_k = \bar{y}_k - y_k$

$P_k = K_P e_k$

$D_k = -K_D \dfrac{y_k - y_{k-1}}{\Delta t}$

$v_k = \bar{u}_k + P_k + I_k + D_k$

if $u_{\min} < v_k < u_{\max}$ then $I_{k+1} = I_k + K_I e_k \Delta t, \quad u_k = v_k$

else $u_k = \max\{u_{\min}, \min\{u_{\max}, v_k\}\}$

Return: u_k, I_{k+1}

SISO PID

```
function [u,I] = SISOPID(ubar,ybar,y,yold,I,KP,KI,KD,dt,umin,umax)

e = ybar-y;
P = KP*e;
D = -KD*(y-yold)/dt;
u = ubar + P + I + D;

if (u >= umax)
    u = umax;
elseif (u <= umin)
    u = umin;
else
    I = I + KI*e*dt;
end
```