PID Controllers

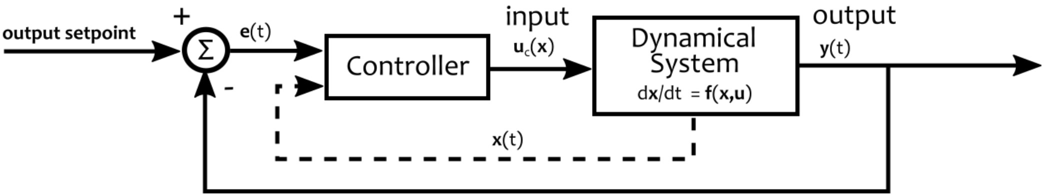
Proportional-Integral-Derivative Controllers

- First used in the beginning of the 1900
- Widespread usage in the industry
- Few parameters to adjust to obtain stability and a good performance

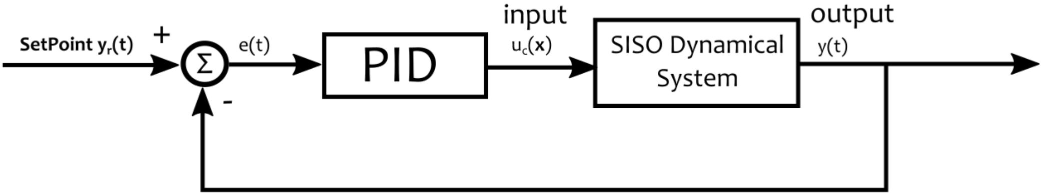
Set-point Control Objective

- Stabilize the system outputs at the values given by the external setpoint
- It is a very typical type of objective for control systems
- It can be said that this is a particular case of the more general reference tracking objective, where the reference is piecewise constant
- leads to an error-based controller, and negative feedback - the output is fed with a negative sign in the controller structure

Set-point Control Objective



PID Controllers



$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}, u);\; y(t) = g(\boldsymbol{x}) \tag{1}$$

$$u_c(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt};\; e(t) = y_r(t) - y(t) \tag{2}$$

P-Controller

$$u_c(t) = k_p e(t) = k_p \big(y_r(t) - y(t) \big) \tag{3}$$

- If the error is zero, the command is zero.
- The bigger the error, the bigger the command.
- Example: control of position using velocity

PI-Controller

$$u_c(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau = k_p\big(y_r(t) - y(t)\big) + k_i \int_0^t \big(y_r(\tau) - y(\tau)\big) d\tau \tag{4}$$

- If the error is zero, the command is the last value of the integral term. the integral terms creates a bias offset in the command required to keep the output constant
- Example: control of velocity using acceleration, car driving or cruise control

PD-Controller

$$u_c(t) = k_p e(t) + k_d \frac{de(t)}{dt} = k_p\big(y_r(t) - y(t)\big) + k_d \frac{d\big(y_r(t) - y(t)\big)}{dt} \qquad (5)$$

- The D-term in a PID controller reacts to fast changes in either the reference or the output
- If the output fastly increases due to a disturbance, the D-term contribution to the command is negative, reducing the command to compensate the disturbance.
- Prevents overshootsing the desired value

More about PIDs

- Choosing of the values k_p, k_i and k_d is called PID tunning
- PIDs can be written also in terms of k, T_i and T_d coefficients

$$u_c(t) = k\left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d\frac{de(t)}{dt}\right) \tag{6}$$

- PIDs can be equipped with saturation - that is limiting the output by a maximum and minimum value

$$u_c(t) = \min\Big(\max\left(u_c(t), \text{min\_value}\right), \text{max\_value})\Big) \tag{7}$$

More about PIDs

- The integral term can accumulate beyond desirable values, causing overshooting for example in big changes of the reference or in PIDs with saturation cases. Anti-windup, limiting and reset strategies for the integral term are often required
- The derivative term has disadvantages in situations when there is a lot of noise in the measurement of the system output value

Digital Implementation

$$e[0] = 0, \;\; I[0] = 0, \;\; D[0] = 0 \tag{8a}$$

$$e[n] = y_r[n] - y[n] \tag{8b}$$

$$I[n] = I[n-1] + \frac{T_s}{2}\left(e[n] + e[n-1]\right) \tag{8c}$$

$$\text{or } I[n] = I[n-1] + T_s e[n] \tag{8d}$$

$$D[n] = \frac{\tau - T_s/2}{\tau + T_s/2} D[n-1] + \frac{1}{\tau + T_s/2}\left(e[n] - e[n-1]\right) \tag{8e}$$

$$\text{or } D[n] = 1/(\tau T_s + 1) D[n-1] + T_s(\tau T_s + 1) e[n] \tag{8f}$$

$$u_c[n] = K_p e[n] + k_i I[n] + k_d D[n], \tag{8g}$$

where T_s is the sampling period, and τ is the time constant of the band-limited derivative term.