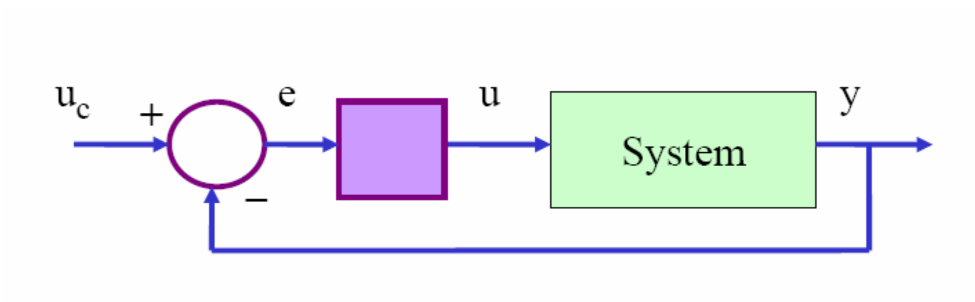
Digital PID Control

Jee-Hwan Ryu

School of Mechanical Engineering
Korea University of Technology and Education



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

$$U(s) = \left[K_P + K_I / s + K_D s\right]$$

In general, there are a number of ways of implementing integration and derivatives digitally

Backward Integration

$$D_I(z) = K_I \frac{T}{z-1}$$

Forward Integration

$$D_I(z) = K_I \frac{Tz}{z-1}$$

Bilinear-transformation Integration

$$D_I(z) = K_I \frac{T}{2} \frac{z+1}{z-1}$$

Most common method of approximation derivative

$$\left.\frac{de(t)}{dt}\right|_{t=T} = \frac{e(kT) - e((k-1)T)}{T}$$

$$D_D(z) = K_D \frac{z-1}{Tz}$$

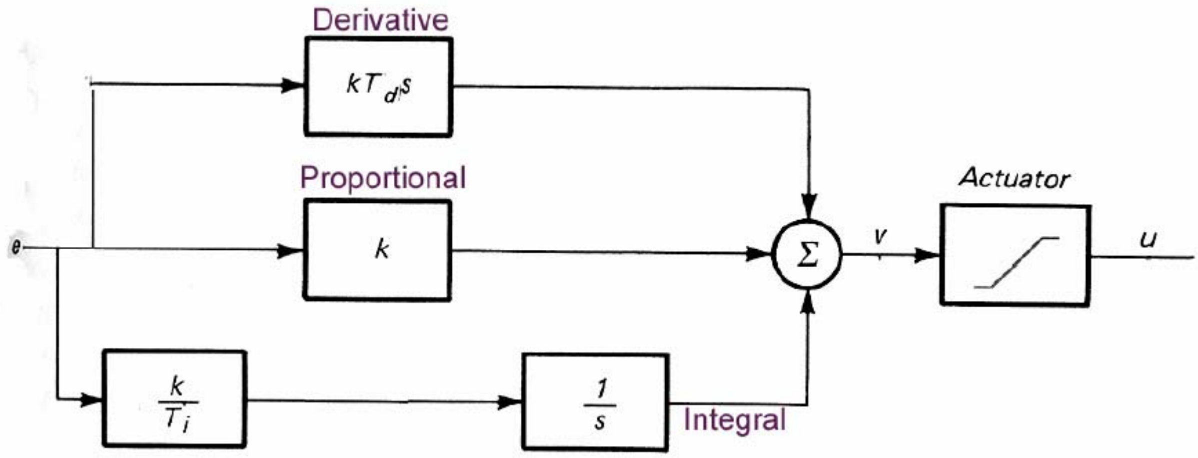
Integrator Windup

- A controller having integral action, in combination with an actuator that becomes saturated, can lead to trouble
- If the control error is so large that the integrator saturates the actuator, then the actuator will remain saturated even if the system output changes
- The integrator can keep on integrating, up to a very large magnitude term
- Then, when something happens to reduce the error (such as a command change), it can take a long time for the integrator to return to a reasonable value
- This is called "integrator windup"

Integrator Windup

- How can we compensate for this?
  - One idea: clip the value of the integrator
    - Problems with this?
  - Another idea: make the integrator leaky
    - Problems with this?

- Can implement "anti-windup" by **measuring** the actuator output and providing a compensating feedback when it is saturated



- Can implement “anti-windup” by **measuring** the actuator output and providing a compensating feedback when it is saturated