Chapter 5:
Close-loop Control with Feedback

Prof. Li Wen   Zheyuan Gong

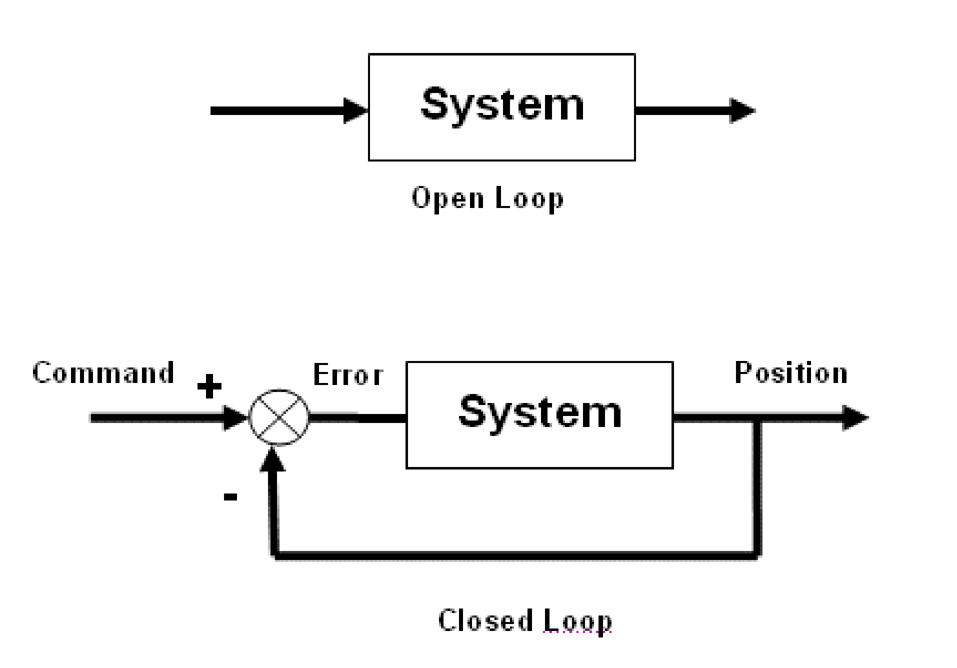
zheyuangong@163.com

Close-loop Control with Feedback

- Aims:
  - Understand the principles of serial communication
  - Understand how to use USART

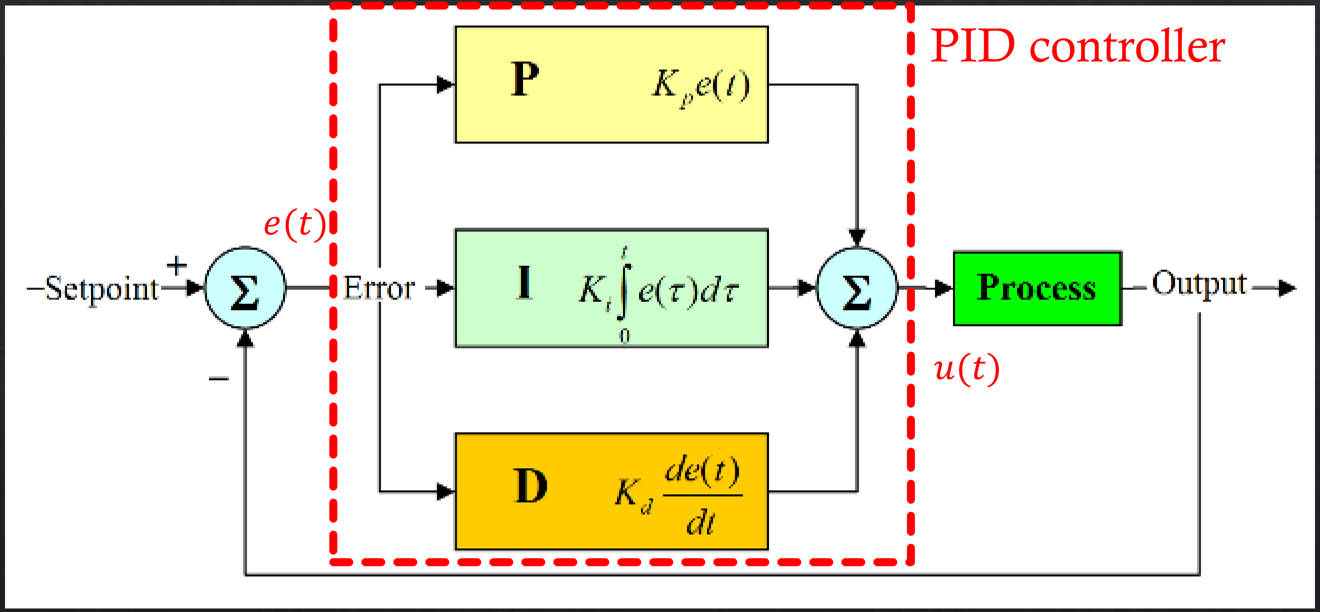
Close-loop Control with Feedback

- Open-loop vs. close-loop



What's PID

- PID controller: proportional–integral–derivative controller



Operate the *error*

What's PID

- PID controller: proportional–integral–derivative controller

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt}$$

- K_p: coefficient for proportional terms
- K_i: coefficient for integral terms
- K_d: coefficient for derivative terms

What do coefficients mean?

- K_p: amplify *error*, faster response
- K_i : stability
- K_d: changing of *error*

How to program PID on computer?

- Continuous to discrete

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt}$$

$$\downarrow$$

$$u(k) = K_p e(k) + K_i [e(k) + e(k-1) + \cdots] + K_d [e(k) - e(k-1)]$$

$$\Delta u(k) = K_p [e(k) - e(k-1)] + K_i e(k) + K_d [e(k) - 2e(k-1) + e(k-2)]$$

How to program PID on computer?

1. Read from ADC $u_i(k)$
2. Get setting value u_r
3. $e(k) = u_i(k) - u_r$
4. $\Delta P_p(k) = K_p \cdot [e(k) - e(k-1)]$
5. $\Delta P_i(k) = K_i \cdot e(k)$
6. $\Delta P_d(k) = K_d \cdot [e(k) - 2e(k-1) + e(k-2)]$
7. $\Delta u(k) = \Delta P_p(k) + \Delta P_i(k) + \Delta P_d(k)$
8. $u(k) = u(k-1) + \Delta u(k)$
9. $e(k-2) \leftarrow e(k-1)$
10. $e(k-1) \leftarrow e(k)$

Algorithm – Pseudocode for PID

```
previous_error = 0
integral = 0
start:
  error = setpoint - measured_value
  integral = integral + error*dt
  derivative = (error - previous_error)/dt
  output = Kp*error + Ki*integral + Kd*derivative
  previous_error = error
  wait(dt)
  goto start
```

Thanks

2016.5.16