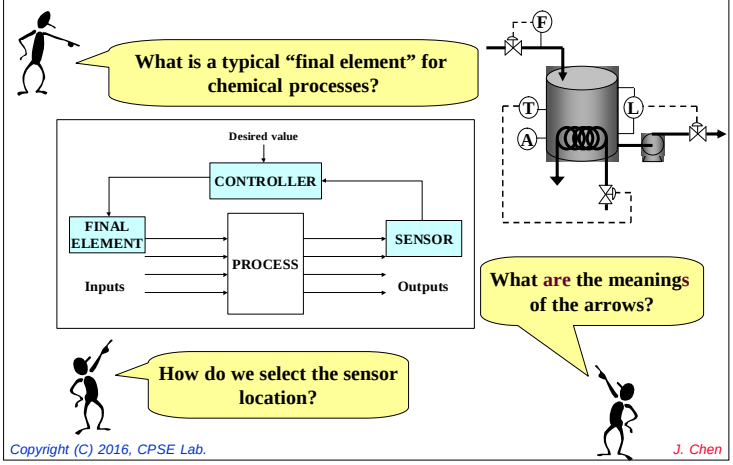
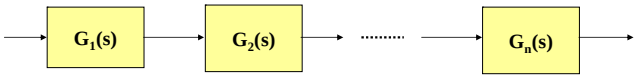
Feedback Control with PID and Its Simulation

陳 榮 輝 (J. Jason Chen)
中原大學化工系
jason@wavenet.cycu.edu.tw
http://wavenet.cycu.edu.tw

Basic Elements in a Feedback Loop



Series (Cascade) Connection

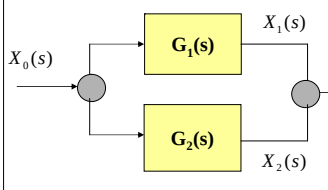


$$X_n(s)=G_n(s)X_{n-1}(s)=G_n(s)G_{n-1}(s)X_{n-2}(s)$$
$$=\cdots=G_n(s)G_{n-1}(s)\cdots G_1(s)X_0(s) \quad i.e. \quad \frac{X_n(s)}{X_0(s)}=\prod_{i=1}^{n}G_i(s)$$

$$G_1(s)=\frac{1}{s} \qquad G_2(s)=\frac{s+1}{s^2+2s+1}$$

```
» tf(1,[1 0])*tf([1 1],[1 2 1])
  Transfer function:
     s + 1
  ---------------
  s^3 + 2 s^2 + s
» sys = series(tf(1,[1 0]),tf([1 1],[1 2 1]))
```


Parallel Connection

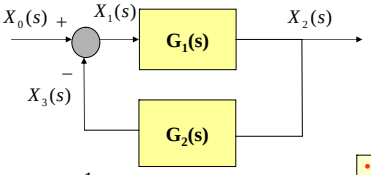


$$X_3(s)=X_1(s)+X_2(s)=G_1(s)X_0(s)+G_2(s)X_0(s)$$
$$\therefore \quad \frac{X_3(s)}{X_0(s)}=G_1(s)+G_2(s)$$

$$G_1(s)=\frac{1}{s} \qquad G_2(s)=\frac{s+1}{s^2+2s+1}$$

```
» tf(1,[1 0])+tf([1 1],[1 2 1])
  Transfer function:
  2 s^2 + 3 s + 1
  ---------------
  s^3 + 2 s^2 + s
» sys = parallel(tf(1,[1 0]),tf([1 1],[1 2 1]))
```


Feedback Connection



$$X_2(s)=G_1(s)X_1(s)=G_1(s)[X_0(s)+X_3(s)]$$
$$=G_1(s)[X_0(s)+G_2(s)X_2(s)]$$
$$\therefore \quad \frac{X_2(s)}{X_0(s)}=\frac{G_1(s)}{1-G_1(s)G_2(s)}$$

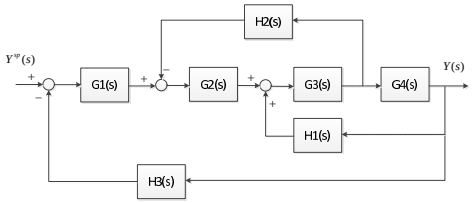
- +1 positive feedback
- -1 negative feedback (default)

$$G_1(s)=\frac{1}{s}$$
$$G_2(s)=\frac{s+1}{s^2+2s+1}$$

G1(s) G2(s)

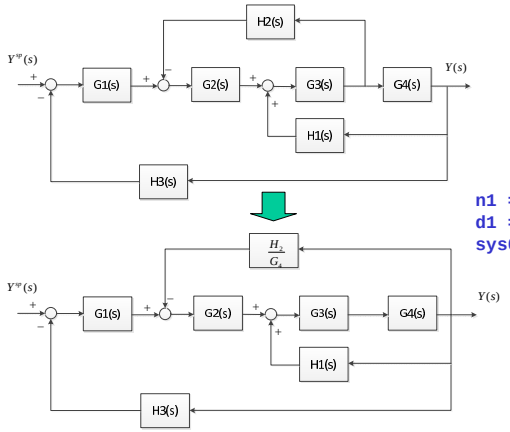
```
» sys = feedback(tf(1,[1 0]),tf([1 1],[1 2 1]))
  Transfer function:
    s^2 + 2 s + 1
  ---------------------
  s^3 + 2 s^2 + 2 s + 1
```


Multi-Loop Reduction



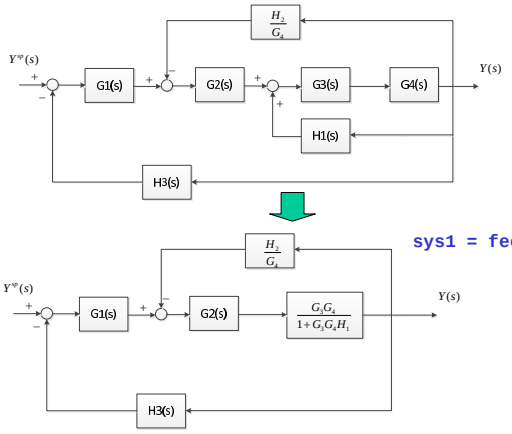
$$G_1(s)=\frac{1}{s+10} \quad G_2(s)=\frac{1}{s+1} \quad G_3(s)=\frac{s^2+1}{s^2+4s+4} \quad G_4(s)=\frac{s+1}{s+6}$$
$$H_1(s)=\frac{s+1}{s+2} \quad H_2(s)=2 \quad H_3(s)=1$$

Multi-Loop Reduction



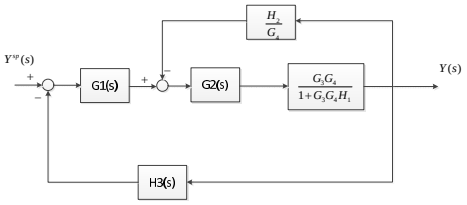

```
n1 = conv(nh2,dg4)
d1 = conv(dh2,ng4)
sys0 = tf(n1,d1)
```


Multi-Loop Reduction

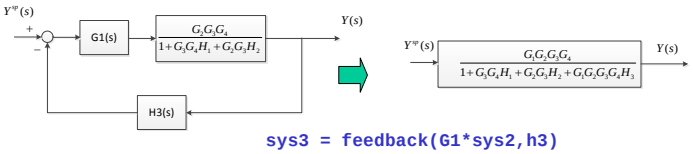



```
sys1 = feedback(G3*G4,H1,1)
```

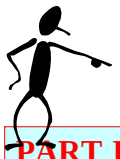

Multi-Loop Reduction




```
sys2 = feedback(G2*sys1,sys0)
```

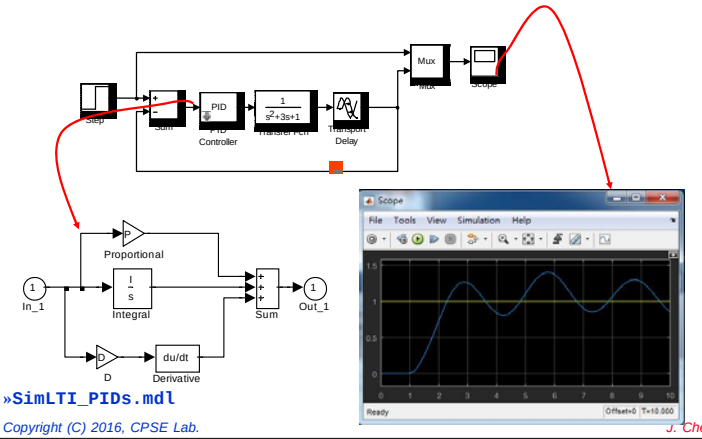



```
sys3 = feedback(G1*sys2,h3)
```

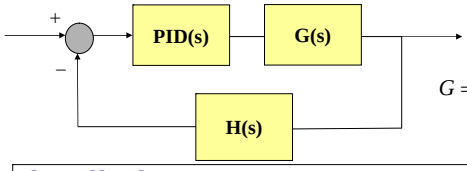
PART IV: FEEDBACK CONTROL LOOP WITH THE PID ALGORITHM

PID for LTI Continuous Process: Simulink




```
»SimLTI_PIDs.mdl
```


PID for LTI Continuous Process: MATLAB



$$G = \frac{e^{-1}}{s^2+3s+1}$$

```
clear all; clc
syms s

% G = 1/(s^2+3s+1)*e^(-1)
num = 1; den = sym2poly(s^2+3*s+1);
G = tf(num,den, 'InputDelay', 1);
H = 1;

%%
Kp = 6; Ki = 0; Kd = 0;
C = pid(Kp,Ki,Kd);
Tc = feedback(C*G,H);
step(Tc)
```

```
»SimLTI_PIDm.mdl
```

