1a. The first order model with time delay can be simplified written in Laplace domain as

$$\frac{T(s)}{u(s)} = P(s) = e^{-\tau s} \cdot \frac{K_u}{60 \cdot S + 1}$$

When the process gain is 0.42 and time constant is 60, the process can be written as

$$P(s) = e^{-\tau s} \cdot \frac{0.42}{60 \cdot s + 1}$$

We can try the PI controller with proportional gain equal to 2, and integral term equal to 300.

$$C_{PI} = \frac{2}{300s}$$

We can close the loop and test the different time delays using in Matlab as below:

```
%Question 1 a
clear all; close all; clc
P=tf(0.42,[60,1],'InputDelay',30)
P1=tf(0.42,[60,1],'InputDelay',100)
P2=tf(0.42,[60,1],'InputDelay',200)
P3=tf(0.42,[60,1],'InputDelay',300)

C=tf(2,[300,0])
T = feedback(P*C,1)
T1 = feedback(P1*C,1)
T2 = feedback(P2*C,1)
T3 = feedback(P3*C,1)
step (T,4000)
hold on
step (T1,4000)
hold on
step (T2,4000)
hold on
step (T3,4000)
legend ('delay=30','delay=100','delay=200','delay=300')
```

The results were plot in Figure 1. The results shown that a large time delay may lead to larger oscillation and take more time to stabilize.

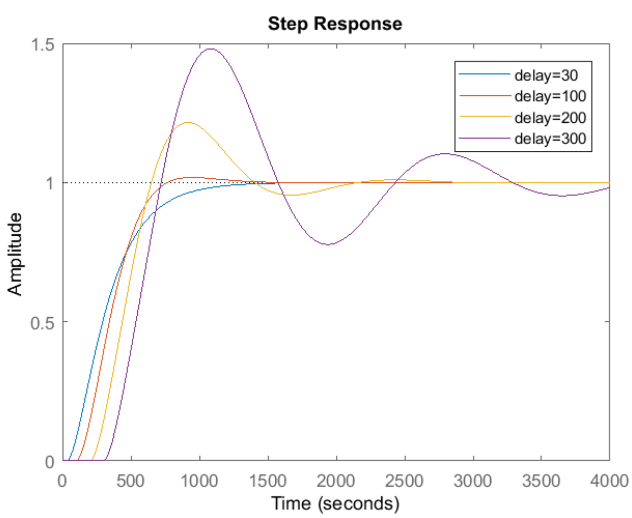
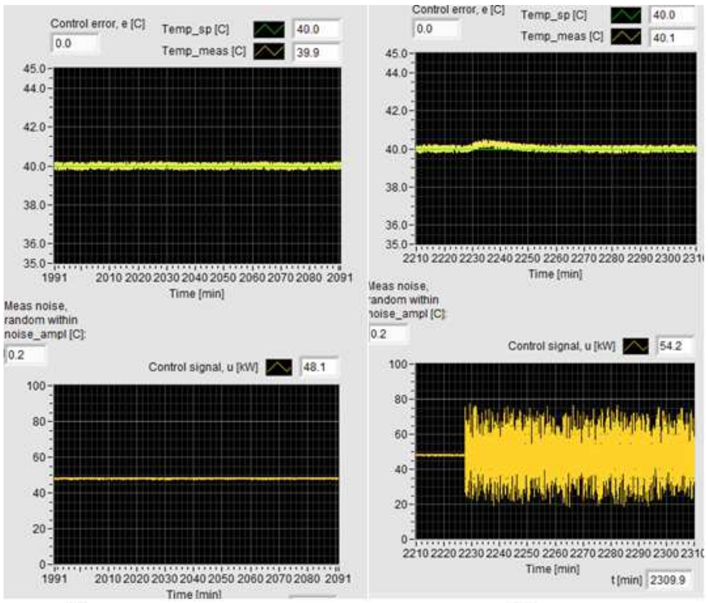


Figure 1 Using PI control for a first order process with different time delay.

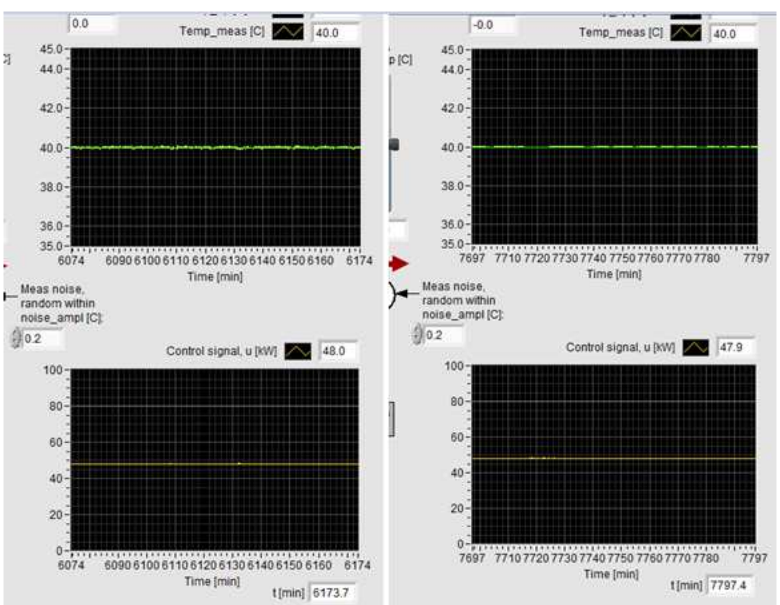
1b. The integral term Ti = 300. According to Ziegler and Nichols method, we let the derivative term Td= 75. When we add random temperature noise, the behavior of the control signal using PI controller and PID controller are shown in Figure 2. The variation of temperature are almost the same, but a larger oscillation of control signal were found from PID controller.



PI PID

Figure 2 Behavior of control signal when add random error of measurement to PI and PID controller.

To reduce the variation of temperature and control signal, the filter time contant for PI control can be Tf = 15 s. To reach the same smoothing results, the filter time contant for PID controller can be Tf = 200 s. The results for applying filter for both PI controller and PID controller are shown in Figure 3.

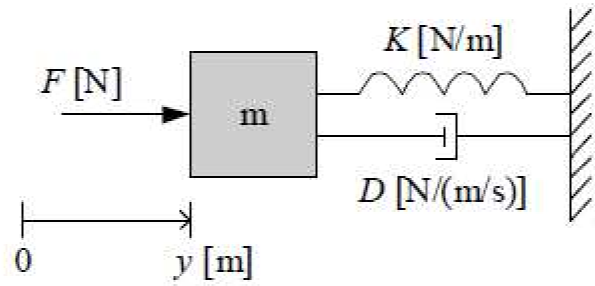


PI control PID control

Figure 3 Results of applying measurement filter for PI controller (Tf=15 s) and PID controller (Tf= 200s).

1c. When using the On/off controller, the mean value of control error is zero.

For the mass-spring damper system,



The damper force is proportional to the speed $F_d(t) = D\dot{y}(t)$, and the spring force is is proportional to the position $F_s(t) = Ky(t)$. The Force balance according to Newton's 2nd Law:

$$m\ddot{y} = F(t) - D\dot{y}(t) - Ky(t)$$

According to laplace transform, the upper equation can be written as below:

$$ms^2Y(s) = F(s) - DsY(s) - KY(s)$$

Then we get:

$$H(s) = \frac{Y(s)}{F(s)} = \frac{1}{ms^2 + Ds + K}$$

2b. To make a step respond as Figure 2.8. When the step of F is from 0 to 4, and m=20, D=4, and K=2.

The Matlab code using "lsim" function is shown as below:

```
clear all, close all,clc
m=20,D=4, K=2
H=tf(1,[m,D,K])
t=0:1:50
u=4*ones(51,1)
u(1:5)=zeros(5,1)
lsim(H,u,t)
```

And the plot from Matlab is showm in Figure

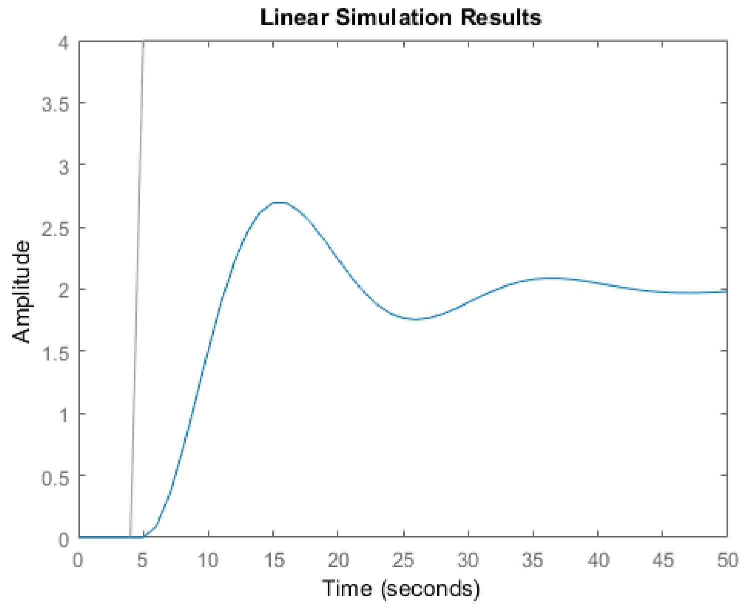


Figure 4 Step respond of the mass-spring-damper system, when the forse was changes from 0 to 4.

2c. The Snapshot of time-position plot, Simulink block diagram, and Matlab script are shown in Figure 5, Figure 6, and Figure 7, respectively. The separate file of Matlab script and Simulink model will also be attached.

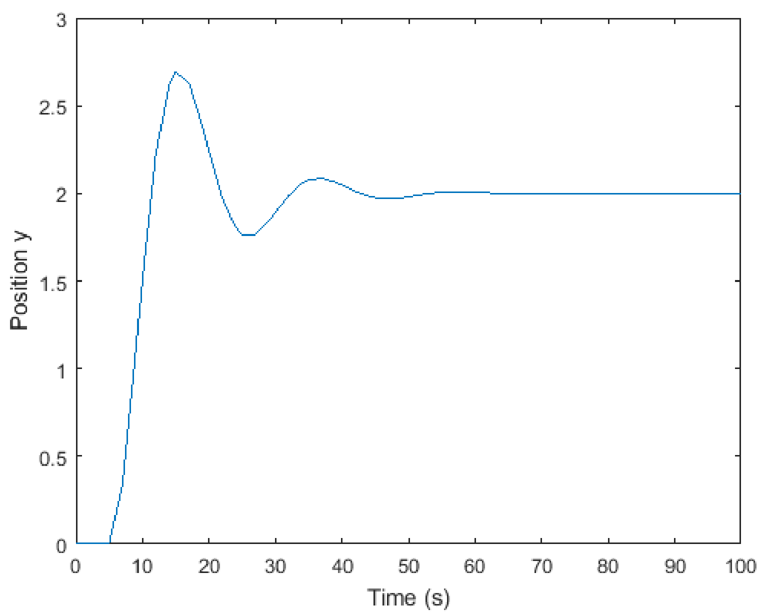


Figure 5 The step respond of mass-spring-damper system.

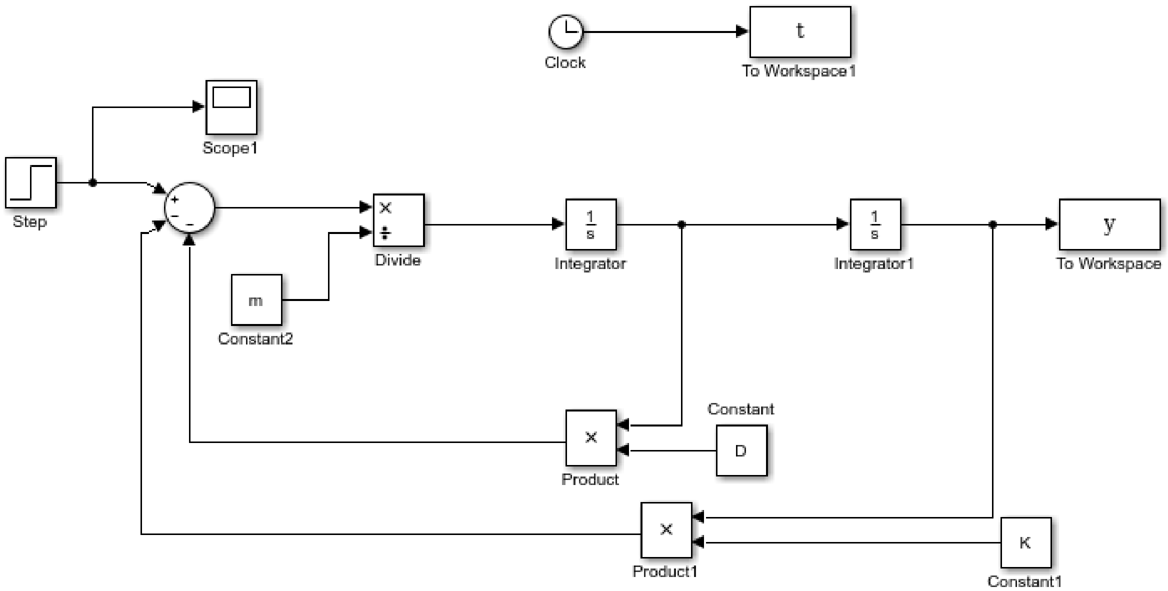


Figure 6 Simulink model of mass-spring-damper system.

```
clear all, close all, clc
m=20,D=4,K=2
options = simset('SrcWorkspace','current');
sim('MassSpringDamper',[],options)
plot(t,y)
xlabel ('Time (s)')
ylabel ('Position y')
```

Figure 7 Matlab cript to control the Simulink model.

3a. The Simulink model block diagram is sown as in Figure 8

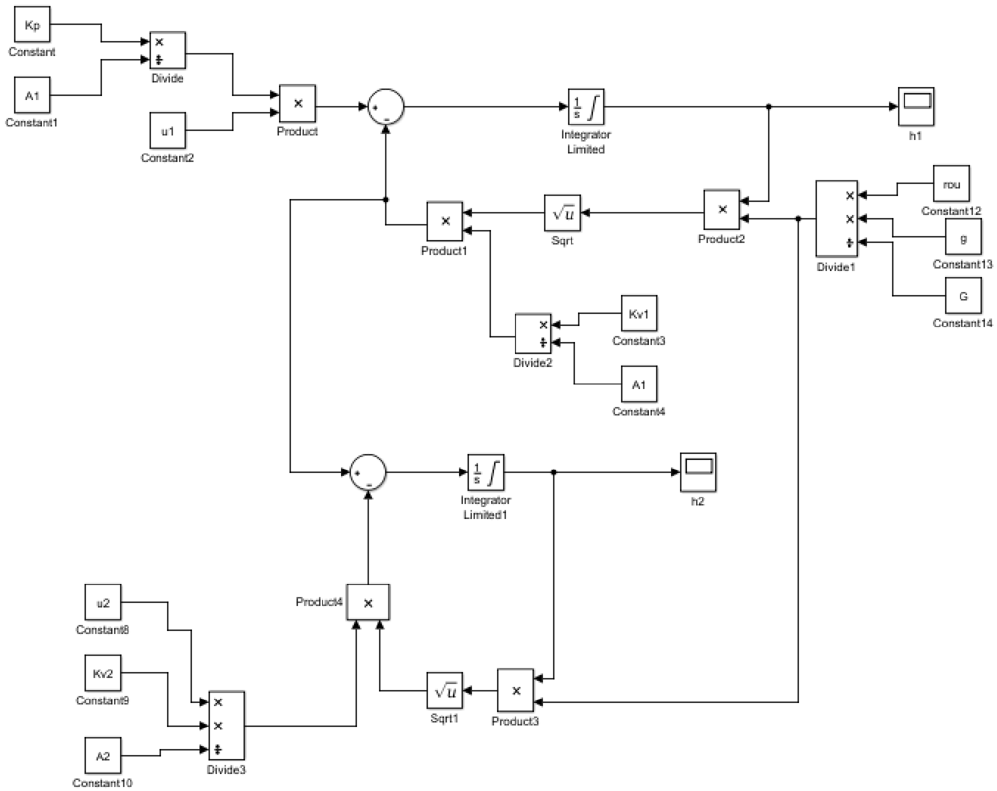


Figure 8 Two tanks system

The responses using "plot" function is shown as in Figure 9 and Figure 10

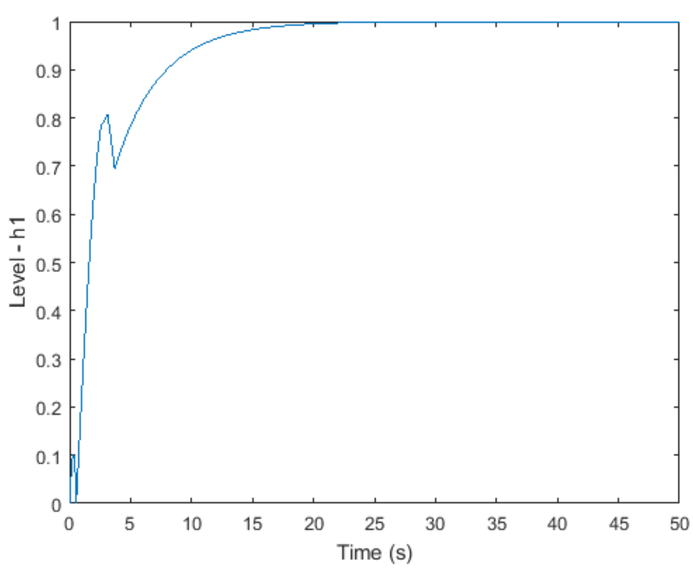


Figure 9 Level h1

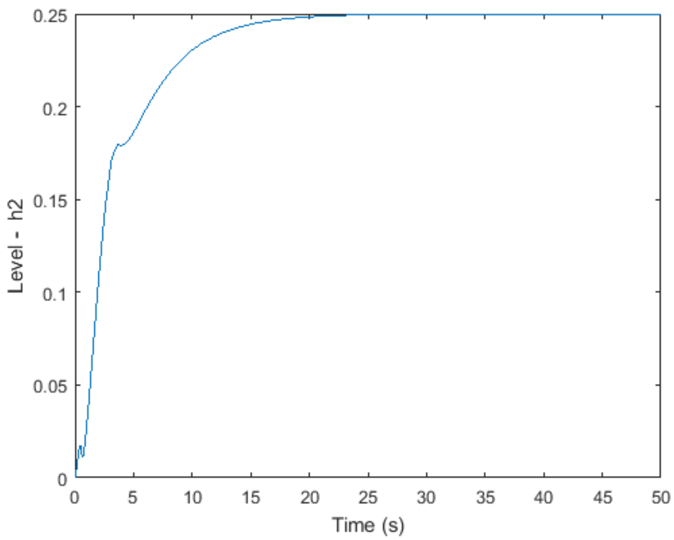


Figure 10 Level h2

3b. When keep u2 constant and u1 changed from zero to 1, the the of h1 and h2 response to the step is shown in Figure 11

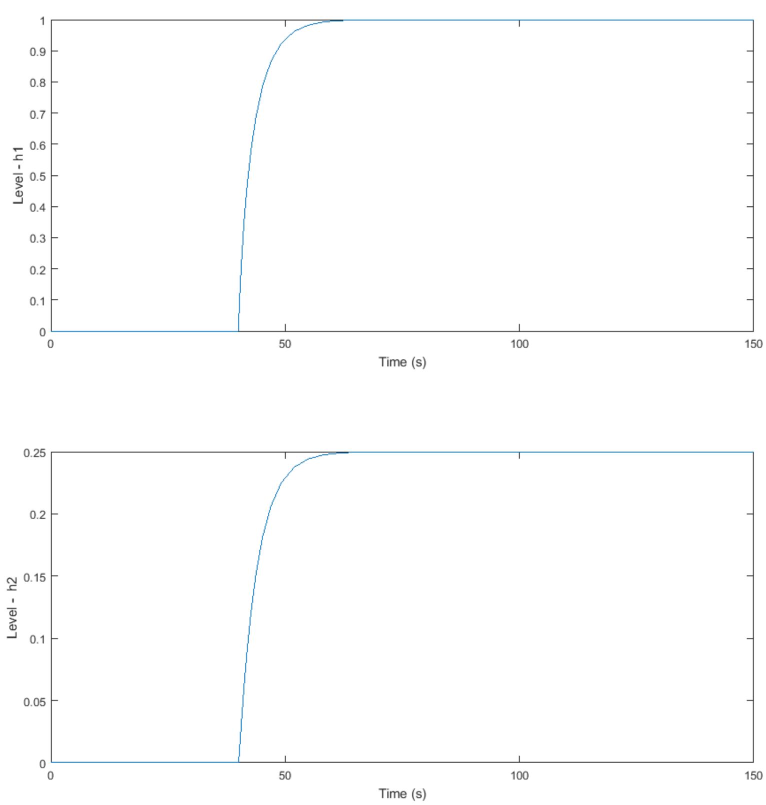


Figure 11 The response of h1 and h2 to a step change of u1 from 0 to 1.

3c. At steady state when u1=1,

$$\dot{h}_1 = \frac{K_p}{A_1}u_1 - \frac{K_{v1}}{A_1}\sqrt{\frac{\rho g}{G}} \cdot \sqrt{h_1} = 0$$

$$\dot{h}_2 = \frac{K_{v1}}{A_1}\sqrt{\frac{\rho g}{G}} \cdot \sqrt{h_1} - \frac{K_{v2}u_2}{A_2}\sqrt{\frac{\rho g}{G}} \cdot \sqrt{h_2} = 0$$

As we the parameter values used in the simulation, Kp=1, A1=2, rou=1, G=1, g=10, Kv1=1/sqrt(10), Kv2=1/sqrt(10), A2=1, u2=1

The analytical calculation results is h1=1, h2=0.25

The simulation results can be verified by analytical calculation.

The two mass balance can be written as the following form after canceling

$$\dot{h}_1 = \frac{K_p}{A_1}u_1 - \frac{K_{v1}}{A_1}\sqrt{\frac{\rho g}{G}} \cdot \sqrt{h_1} = f_1$$

$$\dot{h}_2 = \frac{K_{v1}}{A_2}\sqrt{\frac{\rho g}{G}} \cdot \sqrt{h_1} - \frac{K_{v2}u_2}{A_2}\sqrt{\frac{\rho g}{G}} \cdot \sqrt{h_2} = f_2$$

The first coefficient matrix A can be calculated as partial derivative of function f1 and f2 of h1 and h2, when the system is at steady state.

$$A_{11} = \frac{\partial f_1}{\partial h_1} = -\frac{K_{v1}}{2A_1}\sqrt{\frac{\rho g}{G}} \cdot h_{1s}^{-1/2}$$

$$A_{12} = \frac{\partial f_1}{\partial h_1} = 0$$

$$A_{21} = \frac{\partial f_2}{\partial h_1} = \frac{K_{v1}}{2A_2}\sqrt{\frac{\rho g}{G}} \cdot h_{1s}^{-1/2}$$

$$A_{22} = \frac{\partial f_2}{\partial h_2} = -\frac{K_{v2}u_2}{2A_2}\sqrt{\frac{\rho g}{G}} \cdot h_{2s}^{-1/2}$$

The B matrix entries:

$$B_{11} = \frac{\partial f_1}{\partial u_1} = \frac{K_p}{A_1}$$

$$B_{12} = \frac{\partial f_1}{\partial u_2} = 0$$

$$B_{21} = \frac{\partial f_2}{\partial u_1} = 0$$

$$B_{22} = \frac{\partial f_2}{\partial u_2} = -\frac{K_{v2}}{A_2}\sqrt{\frac{\rho g}{G}} \cdot \sqrt{h_{2s}}$$