Digital Control Exercise 3

Gao Sun sungao12@student.hh.se

Abstract

This exercise is about the PID controller implementation, among which only the P-controller and the PI-controller are covered. Sysquake is used to simulate a water filling tank process. First, we tried to control the process manually. Secondly, the P-control was studied by choosing the different proportional gain. But we found that when there was a step disturbance entering the process, there would be a stationary error, which is the drawback of the P controller. To solve this problem, the integrating part is added, PI controller is derived from this. We learned two ways to do the parameterization for the PI controller, which are Ziegler-Nichols tuning rule and Dead-beat tuning rule. Then we compared the difference of PI controller with and without the anti-windup. Finally, the PI controller with scalar T-polynomial was also tested for its feature.

1. Introduction

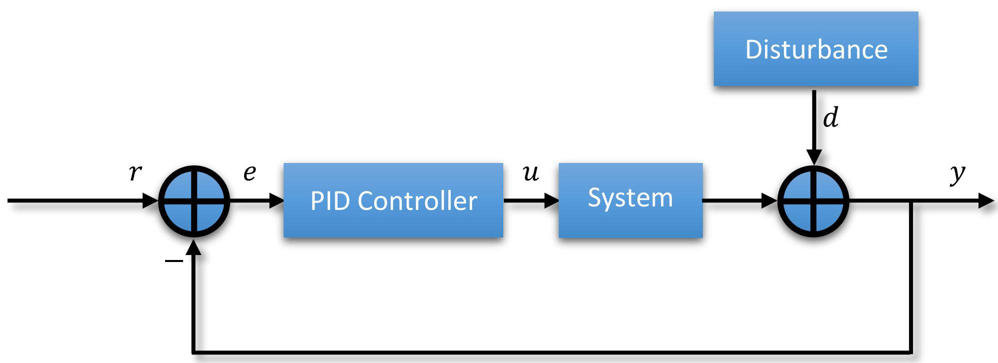


Fig. 1.1 General PID controller

A PID controller is a generic control loop feedback mechanism (controller) widely used in industrial control systems. A PID controller calculates an "error" value as the difference between a measured process variable and a desired setpoint. The controller attempts to minimize the error by adjusting the process control inputs[1]. A typical PID controller structure is shown in Fig. 1.1. In this exercise, the task is to control the water level in the tank by adjust the water flow input to the tank. In the figure, signal r refers to the required water level we need to control to get. Signal y refers to the actual output water level. Signal e refers to the error of the system, i.e. difference between the required water level and the actual water level. Signal u refers to actual controlling action, which is to describe the velocity of the water flow input to the tank (equivalently the water pump voltage). Signal d refers to the disturbance of the system, which is the outflow of the tank in this exercise.

2. Experiment

The experiment in this exercise can be divided into 3 main parts: manual control, P-control and PI-control.

2.1 Manual control

To get some feeling for the tank process dynamics reacts for different inputs, first we tried to control the process manually. We chose the gain of the proportional control $K = 0$ to make the system into an open loop system:

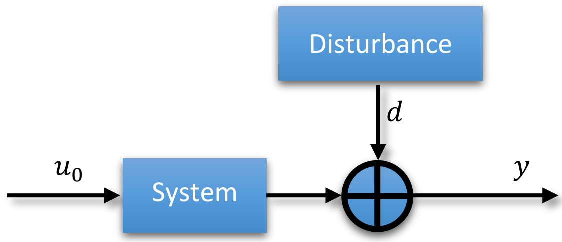


Fig. 2.1.1 Manual control system

Now the system should be like the system shown in Fig. 2.1.1. It's a quite simple-structured system. It is only affected by the signal u_0 given manually and the disturbance produced by the outflow. What we need to do is to tune the parameter of u_0 to make the water level correspond to the reference mark as the red arrow at the tank.

When the reference level is 100, we tune the u_0 to approximately 10 so that the requirement is meet.

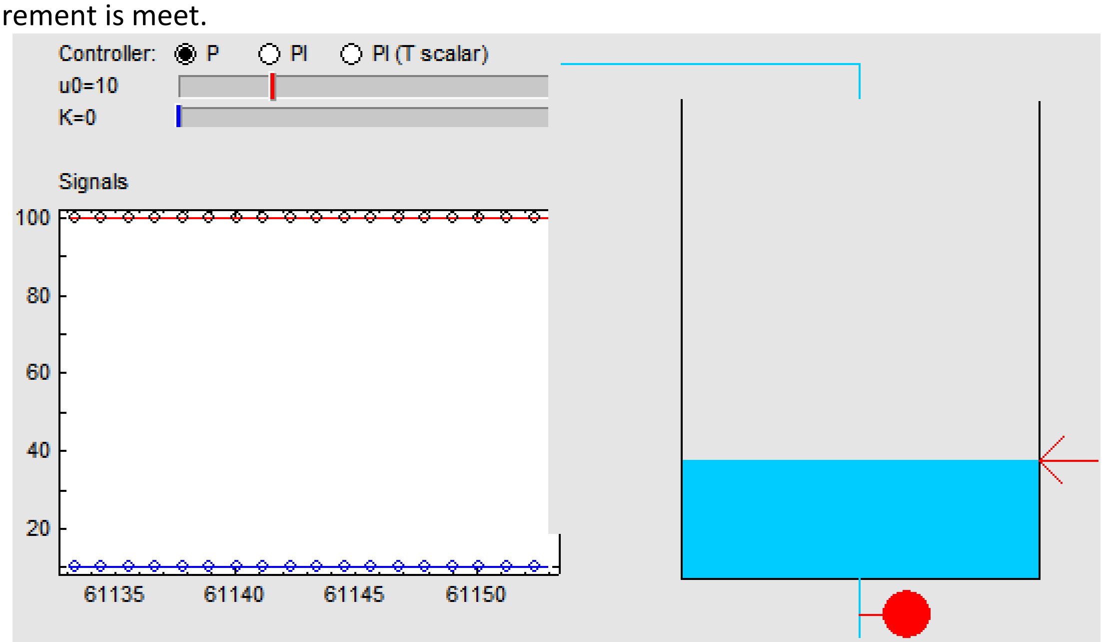


Fig. 2.1.2 Condition of tank simulator when reference level is 100

When the reference level is 200, we tune the u_0 to approximately 14.0738 so that the requirement is met.

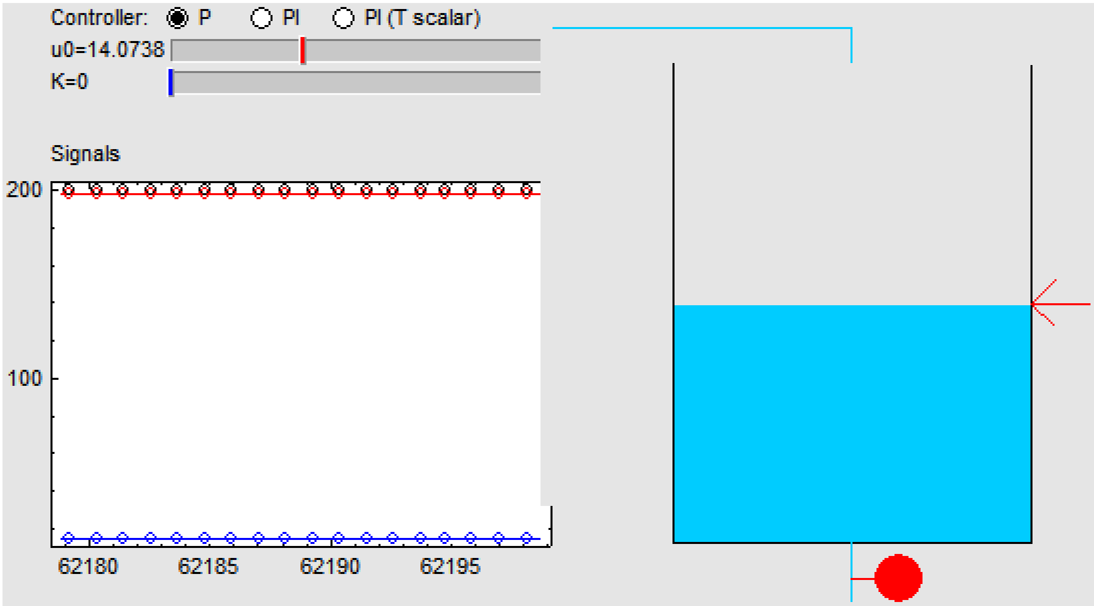


Fig. 2.1.3 Condition of tank simulator when reference level is 200

When the reference level is 300, we tune the u_0 to approximately 17.2978 so that the requirement is meet.

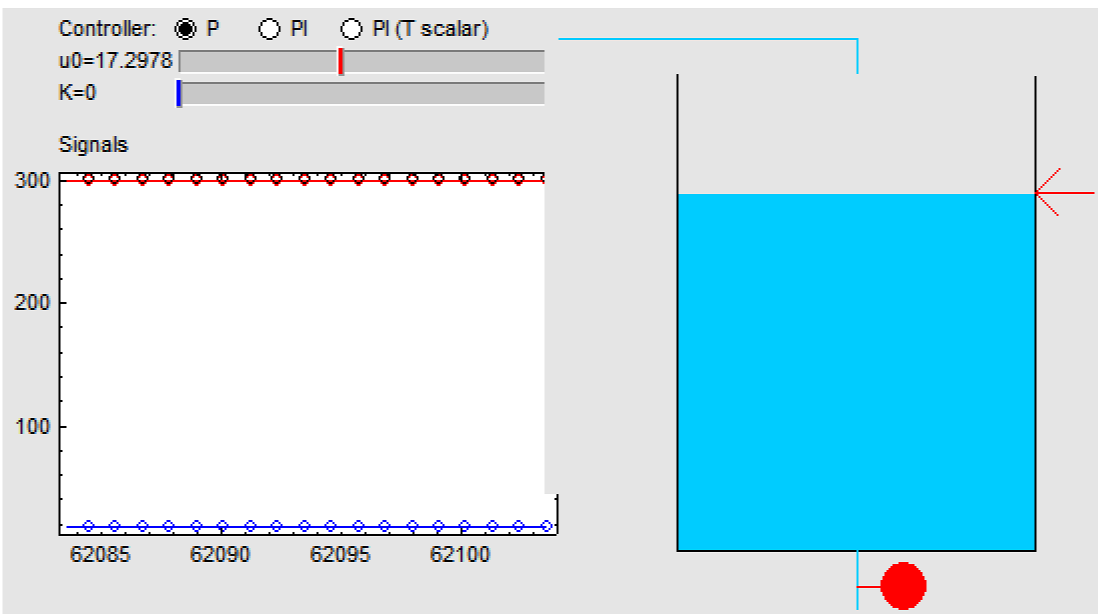


Fig. 2.1.4 Condition of tank simulator when reference level is 300

As we can see, in this system, for any different reference level, we always need to tune the u_0 again and again. Now the situation is that the reference is 100 again, but we open the red valve to increase the disturbance a lot.

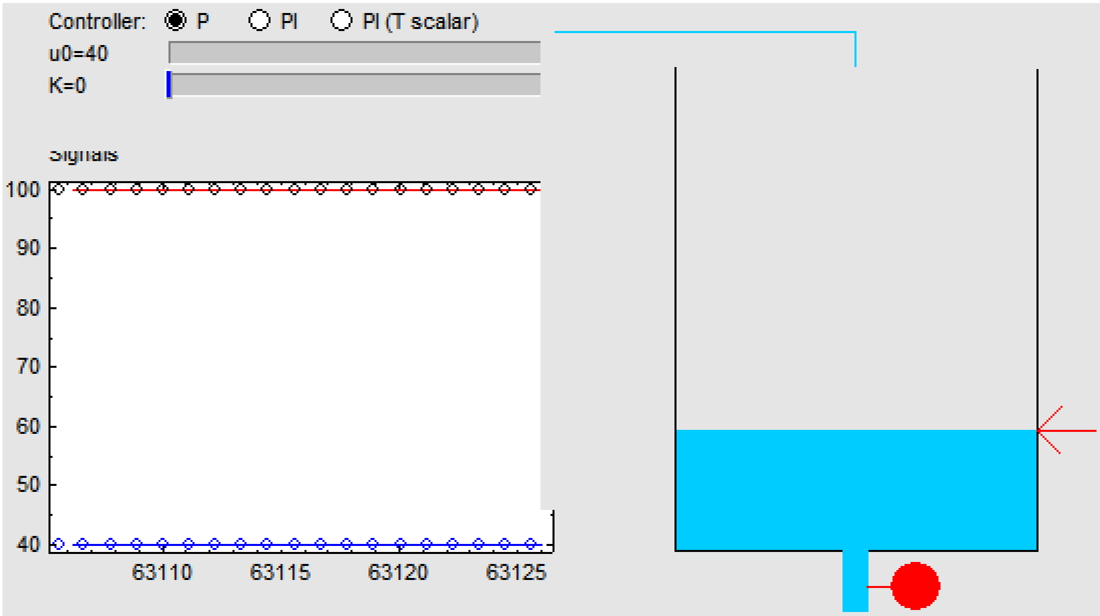


Fig. 2.1.5 Condition of tank simulator with great disturbance

As is shown in the Fig. 2.1.5, now the u_0 need to be increased to 40 to get a new balance. So we can see that this system doesn't have the feature of anti-disturbance.

2.2 P-control

As is stated above, the manual control neither can adjust the water level automatically for different reference levels, nor have the feature of anti-disturbance. To solve this problem, we tried to add the proportional feedback into the system to build a P-control system.

2.2.1 Mathematical model for the P-control system

Then we need to analyze the mathematical model for the system. For easier calculation, we assume that the outflow which is the disturbance in the system is much smaller than the inflow and we just neglect it first.

The relation between the signal y which is the actual water level in the tank and the signal u which is the coming speed of the water from the water pump should be like this:

$$y(k) = y(k-1) + u(k-1)$$

It means that the quantity of the water currently equals the quantity of the water at last time point plus the water comes from the water pump during the last time short period. Then we can convert it like this:

$$y(k) = q^{-1}y(k) + q^{-1}u(k-1)$$

In the polynomial form, the system is:

$$A(q^{-1})y(k) = B(q^{-1})u(k), \qquad A(q^{-1}) = 1 - q^{-1}, \qquad B(q^{-1}) = q^{-1}$$

Next, let's think about the P-control part. First we need to do the same thing as we did in the manual control. We close the valve and set a reference level, and adjust u_0 to get a balance to the reference level. This u_0 is the bias level which should be added to $u(k)$. Assume the proportional gain is K, the control signal is now $u = u_0 + K(r - y)$, so the system block scheme now should be like this:

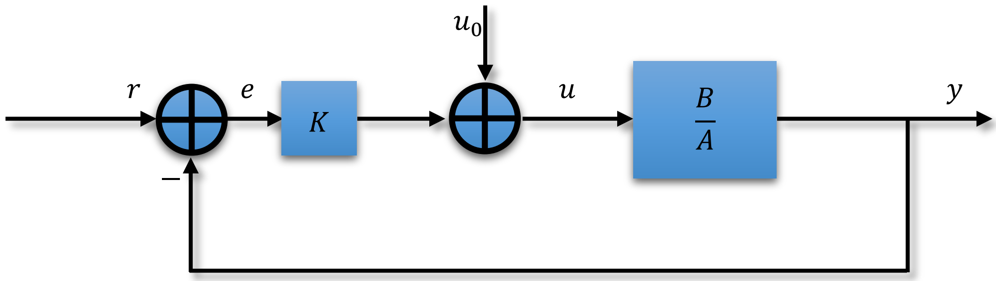


Fig. 2.2.1 P-control system

Then we can calculate the relation between y and r :

$$y(k) = \frac{B}{A}u(k) = \frac{B}{A}(u_0 + K(r(k) - y(k)))$$

$$y(k) = \frac{Bu_0 + BKr(k)}{A + BK} = \frac{q^{-1}u_0 + q^{-1}Kr(k)}{1 + (K - 1)q^{-1}}$$

As u_0 is a constant, it doesn't affect so much in the difference equation. To simplify the formula, we can disregard it.

$$y(k) = \frac{Kq^{-1}}{1 + (K - 1)q^{-1}}r(k)$$

If we use the dead-beat tuning rule, which means we make the proportional gain $K = 1$. In this situation,

$$y(k) = r(k - 1)$$

To verify this, we set $u_0 = 10, K = 1$ and make a step change in the reference level from 82 to 115. The signal $y(k)$ which is the red line gets such a response which shown in Fig. 2.2.2 below. Roughly, the output tracks the reference with one sample delay.

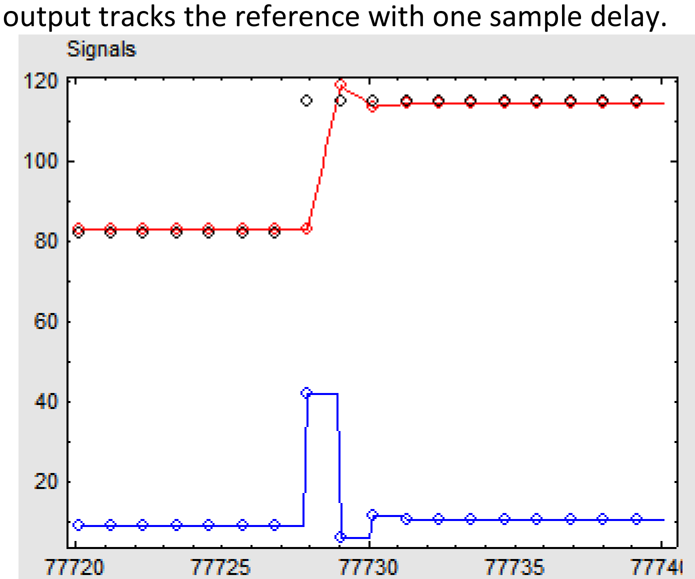


Fig. 2.2.2 Response of the dead-beat tuning P-control system to a step change input

2.2.2 Stability for the P-control system

Form the formula relation between $y(k)$ and $r(k)$, the propagation function is:

$$G(q^{-1}) = \frac{Kq^{-1}}{1+(K-1)q^{-1}}$$

We can regard this closed-loop system as a first order system. The pole of this system is $1-K$. When $0<K<1$, then $0<1-K<1$. So that means the pole locates on the right part inside of the unit circle. In this situation, the step response of the system looks like this:

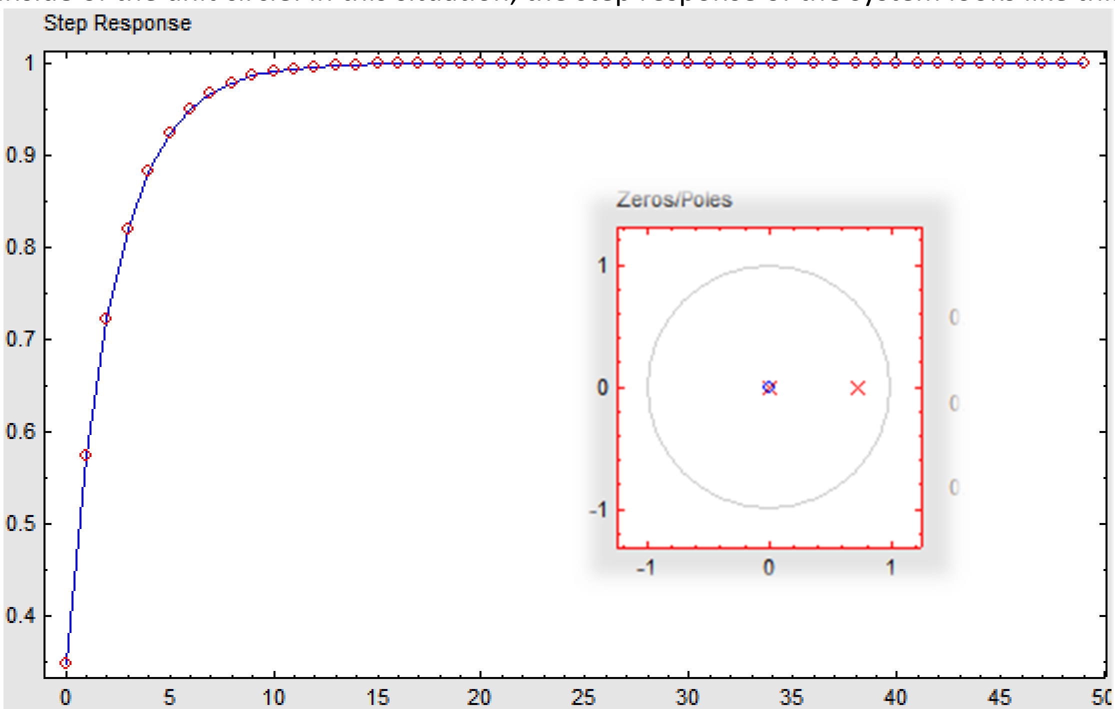


Fig. 2.2.3 Step response of the system when $0<K<1$

As is shown from the figure, the system is stable and the response is monotonous and coming closer and closer to its steady-state gain.

When $1<K<2$, then $-1<1-K<0$. So that means the pole locates on the left part inside of the unit circle. In this situation, the step response of the system looks like this:

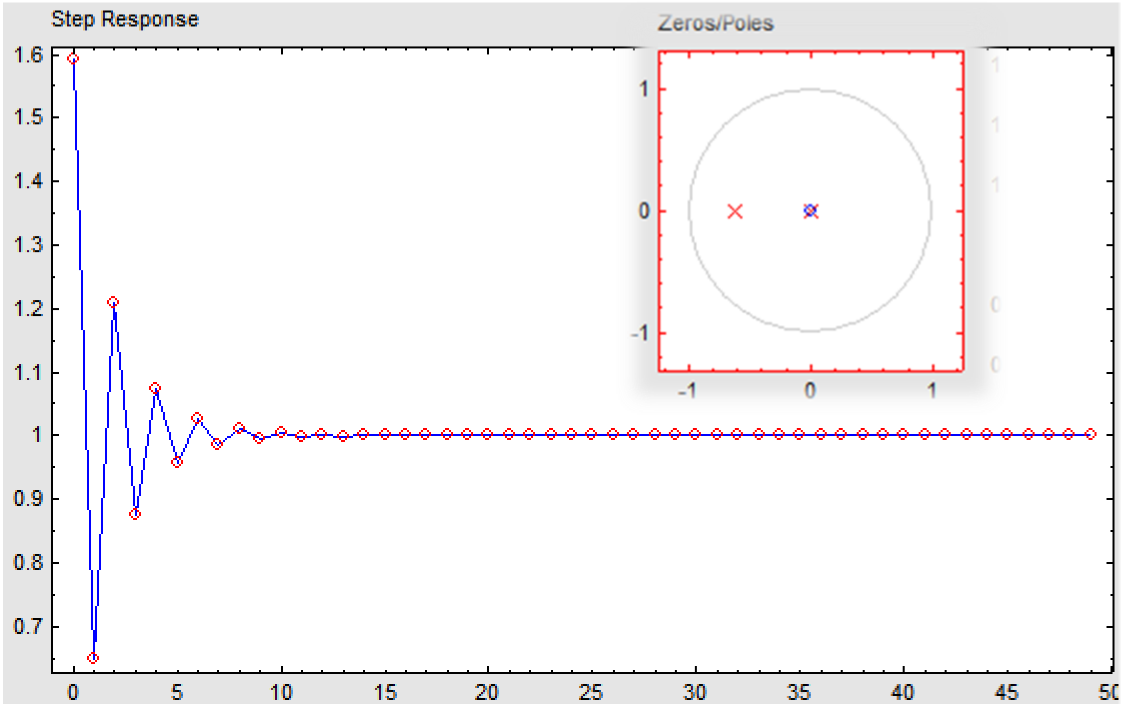


Fig. 2.2.4 Step response of the system when $1 < K < 2$

As is shown from the figure, the system is stable and the response is oscillating and coming closer and closer to its steady-state gain.

When $K > 2$, then $1 - K < -1$. So that means the pole locates to the left part outside of the unit circle. In this situation, the step response of the system looks like this:

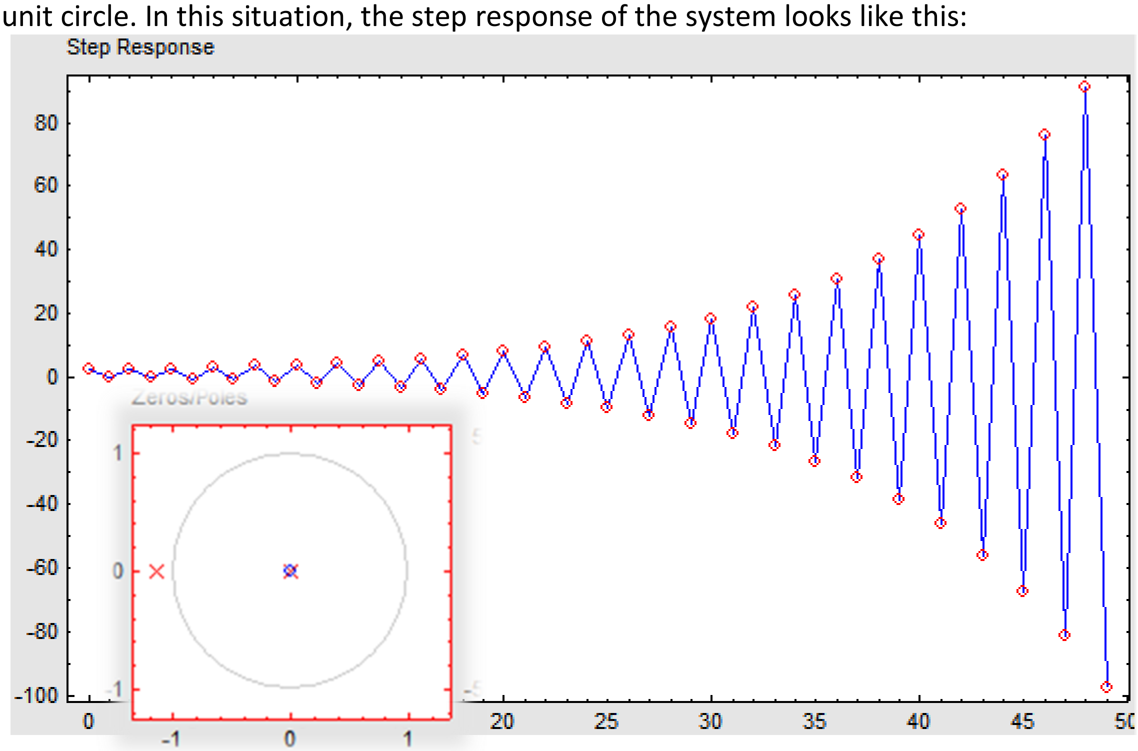


Fig. 2.2.5 Step response of the system when $K > 2$

As is shown from the figure, the system is unstable and the response is oscillating with larger and larger amplitude.

To verify this, we set $u_0 = 10, K = 0.7$, and make a step change in the reference level from 30 to 105. The signal $y(k)$ which is the red line gets such a response which shown in Fig. 2.2.6 below. Roughly, the response is monotonous and coming closer to the reference level.

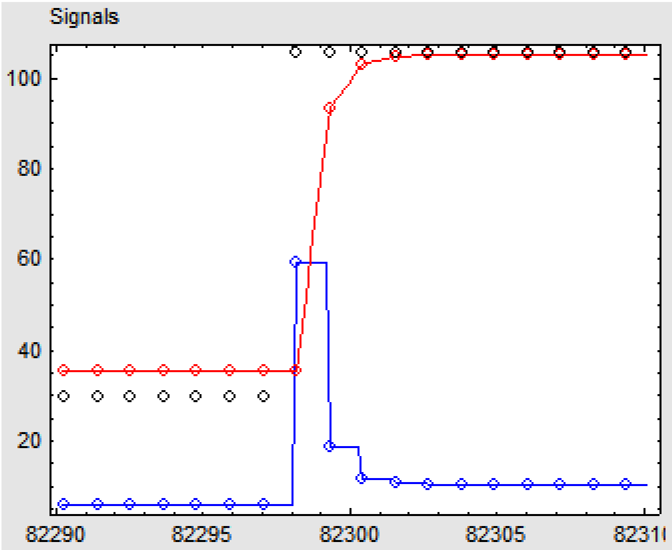


Fig. 2.2.6 Response of the P-control system to a step change input when $K = 0.7$

Then we set $u_0 = 10, K = 1.7$, and make a step change in the reference level from 53 to 123. The signal $y(k)$ which is the red line gets such a response which shown in Fig. 2.2.7 below. Roughly, the response is monotonous and coming closer to the reference level.

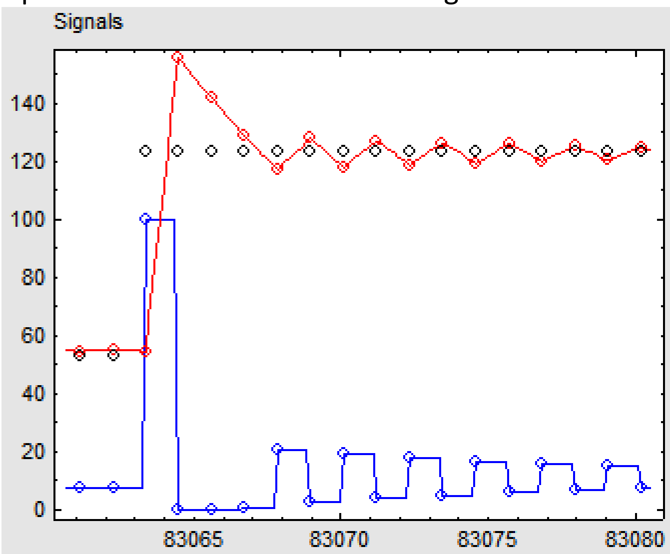


Fig. 2.2.7 Response of the P-control system to a step change input when $K = 1.7$

2.2.3 Stationary error of the P-control system

Now we recall the disturbance which was ignored previously. Including the disturbance, the system block scheme should look like this:

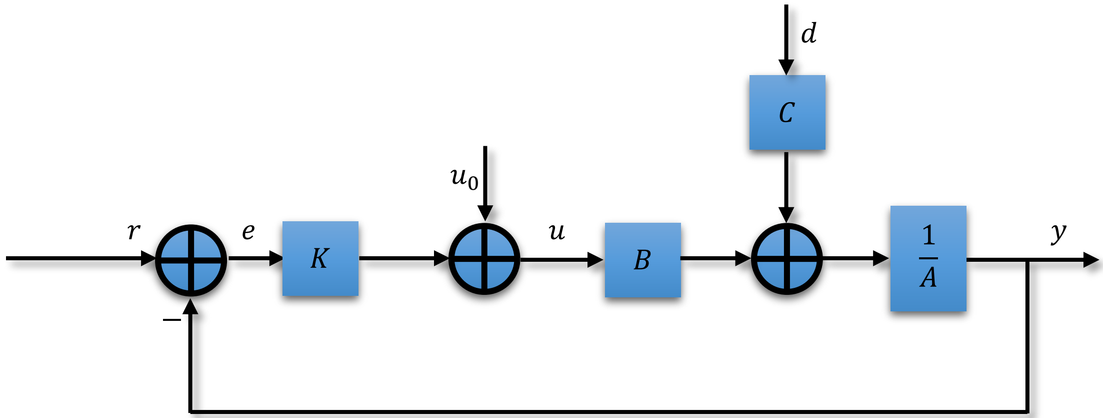


Fig. 2.2.8 P-control system with disturbance

So we can do the following calculation:

$$y = \frac{1}{A}(Bu + Cd) = \frac{1}{A}(BK(r - y) + Cd)$$

$$y = \frac{BK}{A + BK}r + \frac{C}{A + BK}d$$

Now the situation is that after we make the actual water level in the tank correspond to the reference level, we open the valve to create a step disturbance to the system. In this process, it is obviously that the reference level is never changed, so $\frac{BK}{A+BK}r$ can be regarded as a constant. Similar with the situation that we disregarded u_0 in 3.2.1, we can also ignore $\frac{BK}{A+BK}r$ because it doesn't cause much effect in the difference equation:

$$y = \frac{C}{A + BK}d = \frac{C}{1 + (K - 1)q^{-1}}d$$

When $K = 1$,

$$y = Cd$$

Obviously, the output of the system will change to a condition related with the disturbance. When $0 < K < 1$, the output of the system will change monotonous to a condition related with the disturbance. When $1 < K < 2$, the output of the system will oscillate to a condition related with the disturbance. When $K > 2$, the system is not stable.

Anyway, once there is a step disturbance, the output of the system will follow the disturbance. So it will cause a station error to the system. To verify this, first we set $K = 1$ and $u_0 = 13.8382$ to match the reference level of 200 which is shown in Fig. 2.2.9.

Then we turn on the valve to cause a disturbance. The result is that the actual water level in the rank dropped to approximately 161 immediately and after that it stayed on this level stably. So this is the stationary error. The result is shown in Fig. 2.2.10.

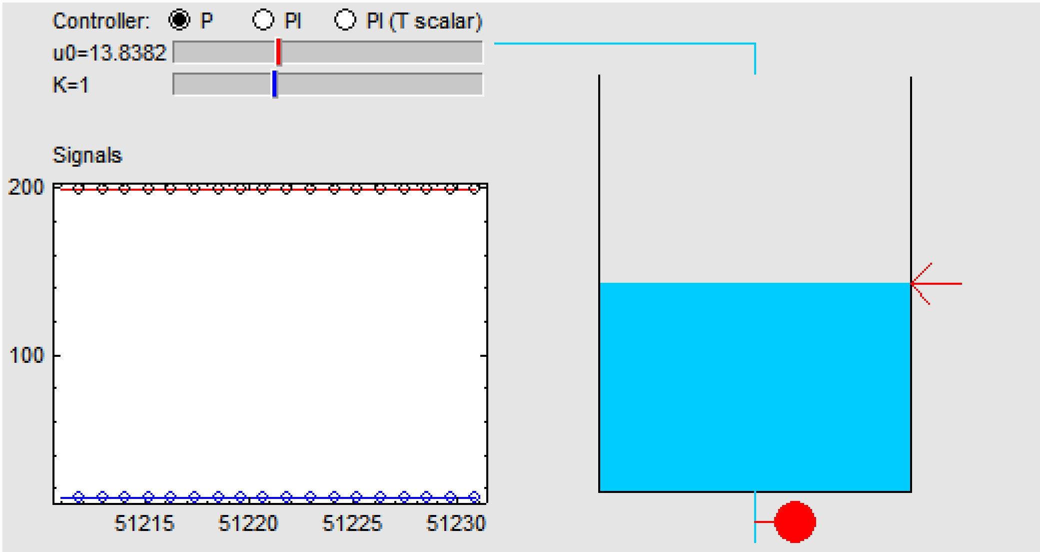


Fig. 2.2.9 P-control system balance without disturbance

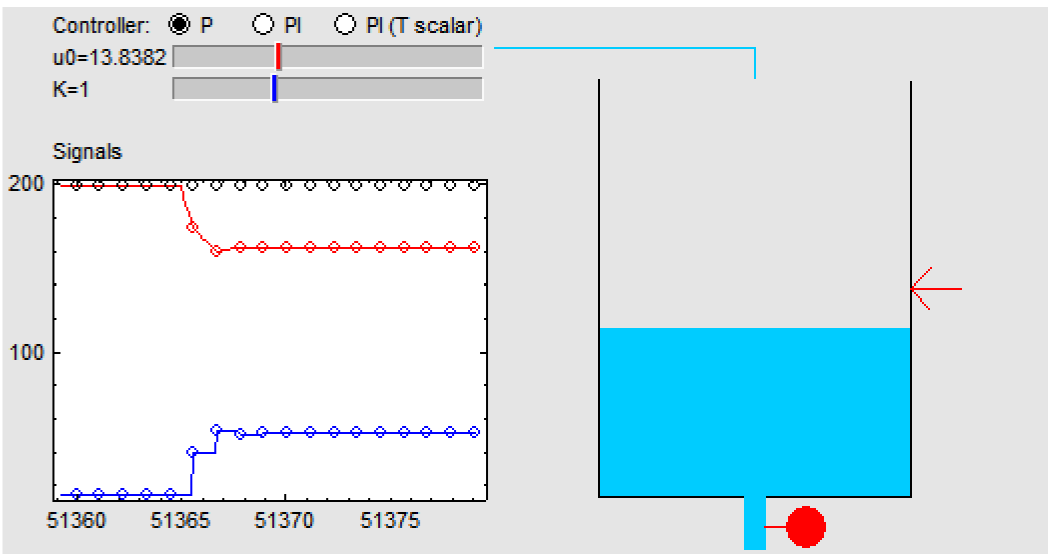


Fig. 2.2.10 P-control system balance with great disturbance

2.3 PI-control

As we can see, the P-controller can control the water level to follow the reference level automatically. But the drawback of the P-controller is also obvious, which is that the P-controller still does not have the feature of anti-disturbance. To solve this problem, we tried to add the integrating part into the P-control system to build a PI-control system.

2.3.1 Mathematical model for the PI-control system

In the continuous-time domain, the PI-control system should be described like:

$$u(t) = Ke(t) + \frac{K}{T_i}\int_0^t e(s)ds$$

Do the Laplace-transform for this equation:

$$U(s) = K(1 + \frac{1}{T_i}\frac{1}{s})E(s)$$

Now we should do the discretization for continuous time because the model we have is discrete. We can approximate the derivative as the difference of two adjacent samples in discrete signal:

$$\frac{d}{dt}y(k) \approx \frac{y(k) - y(k-1)}{k - (k-1)} = y(k) - y(k-1) = y(k) - q^{-1}y(k) = (1 - q^{-1})y(k)$$

So we can get the conclusion:

$$\frac{d}{dt} \approx 1 - q^{-1}$$

As we know, derivative in the Laplace domain is s, so we can say $s \approx 1 - q^{-1}$. In this way, we can get the error propagation function of the PI-controller by replacing s with $1 - q^{-1}$:

$$u(k) = K(1 + \frac{1}{T_i}\frac{1}{1 - q^{-1}})e(k)$$

Now the system block scheme looks like this:

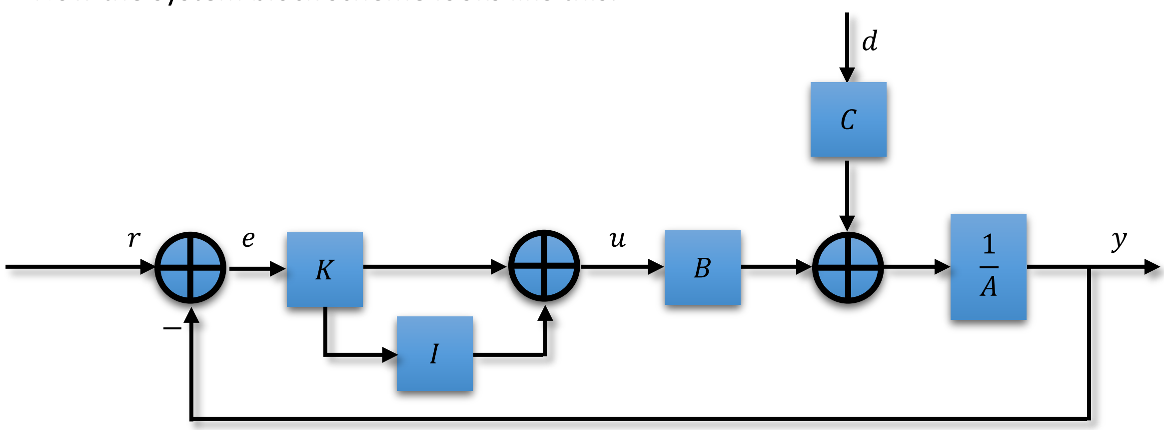


Fig. 2.3.1 PI-control system

As is shown in the figure, the u_0 is not needed because of the integral action. The block I refers to the integrating part:

$$I = \frac{1}{T_i(1 - q^{-1})}$$

Next, we can calculate the propagation function for the whole PI-controller:

$$Ay = Cd + Bu = Cd + B[K(1 + I)(r - y)]$$

$$(1 - q^{-1})y = Cd + q^{-1}K[1 + \frac{1}{T_i(1 - q^{-1})}](r - y)$$

Finally, we can get:

$$y = \frac{(1 - q^{-1})Cd + [\left(K + \frac{K}{T_i}\right)q^{-1} - Kq^{-2}]r}{1 - \left(2 - K - \frac{K}{T_i}\right)q^{-1} + (1 - K)q^{-2}}$$

2.3.2 Dead-beat tuning for PI-control

Dead-beat design corresponds to placing both poles at the origin. That means we should make the roots of the equation $1-\left(2-K-\frac{K}{T_i}\right)q^{-1}+(1-K)q^{-2}$ equals 1 at all the time. So we get the following equations:

$$\begin{cases} 2-K-\dfrac{K}{T_i}=0 \\ 1-K=0 \end{cases}$$

So we can get:

$$\begin{cases} K=1 \\ T_i=1 \end{cases}$$

So the propagation function for the whole PI-controller is:

$$y=(1-q^{-1})Cd+[2q^{-1}-q^{-2}]r$$

Write it in the difference equation:

$$y(k)=2r(k-1)-r(k-2)+C[d(k)-d(k-1)]$$

As the input signal $r(k)$ does not change a lot at most time, which means at most time, we have $r(k-1)=r(k-2)$. Then we can say at most time, $2r(k-1)-r(k-2)=r(k-1)$. So we can simplify $y(k)$ into:

$$y(k)\approx r(k-1)+C[d(k)-d(k-1)]$$

This equation means that if we use the Dead-beat tuning rule to tune the parameter of the PI-control system, the response of the system will follow the input with one sample delay. More important than this, the system has good feature of anti-disturbance. Even a step disturbance d will be eliminated in only one sample because the response always output the difference of the latest two disturbance quantity.

To verify this, we change the tank system to PI mode, and set $K=1, T_i=1$. First is to test the anti-disturbance feature. After the system reaches a balance, we open the valve to cause a step disturbance. The Fig. 2.3.2 below shows how the system responses:

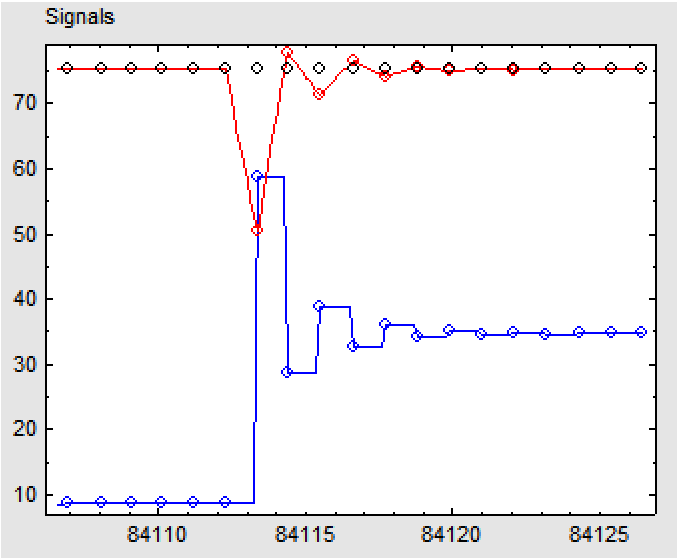


Fig. 2.3.2 Response of dead-beat tuning PI-control system to disturbance

We can see that the system eliminates the disturbance in one sample roughly.

Next, we give a step change for the reference level, the Fig. 2.3.3 below shows how the system responses:

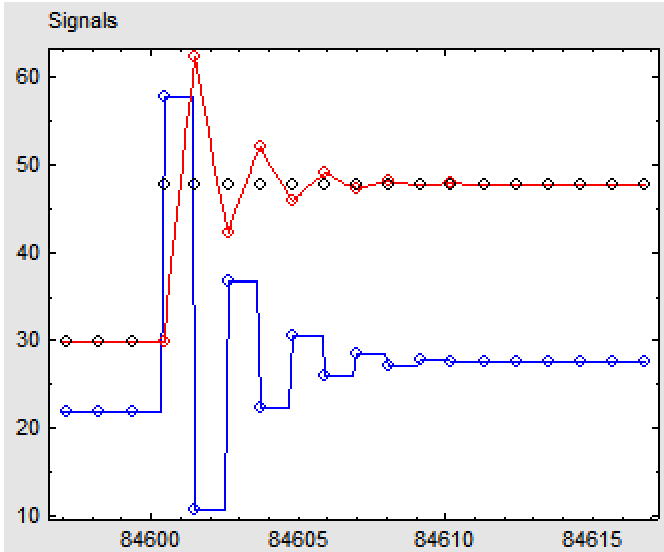


Fig. 2.3.3 Response of dead-beat tuning PI-control system to a step change input

We can see that although at most time, $2r(k-1) - r(k-2) = r(k-1)$, but at the change point of $r(k)$, there will cause a great change to the output $y(k)$, which refers to the peak in Fig. 2.3.3. Then the system will oscillate for some time, and after that the output become stable to the reference level.

2.3.3 Ziegler-Nichols tuning for PI-control

According to the Z-N tuning rule,

$$K = K_C * 0.45$$
$$T_i = T_c/1.2$$

K_C refers to the gain of the P-controller such that the system is on the stability boundary. T_c refers to the oscillation period when this gain is applied. As we have discussed before, $K_C = 2, T_c = 2$. So

$$K = 2 * 0.45 = 0.9$$
$$T_i = \frac{2}{1.2} = \frac{5}{3}$$

Apply them into the propagation function before, we can get:

$$y = \frac{(1-q^{-1})Cd + [(0.9+0.54)q^{-1} - 0.9q^{-2}]r}{1-(2-0.9-0.54)q^{-1} + (1-0.9)q^{-2}}$$

To simplify the equation, we ignore the disturbance in the equation first:

$$y = \frac{1.44q^{-1} - 0.9q^{-2}}{1-0.56q^{-1}+0.1q^{-2}}r = \frac{1.44q^{-1} - 0.9q^{-2}}{[1-(0.2800+0.1470i)q^{-1}][1-(0.2800-0.1470i)q^{-1}]}r$$

So $A_c = 1 - 0.56q^{-1} + 0.1q^{-2}$, $\lambda_1 = 0.2800 + 0.1470i$, $\lambda_2 = 0.2800 - 0.1470i$.

We use Matlab to analyze this system. The step response of $G(q^{-1}) = \frac{1.44q^{-1}-0.9q^{-2}}{1-0.56q^{-1}+0.1q^{-2}}$ is:

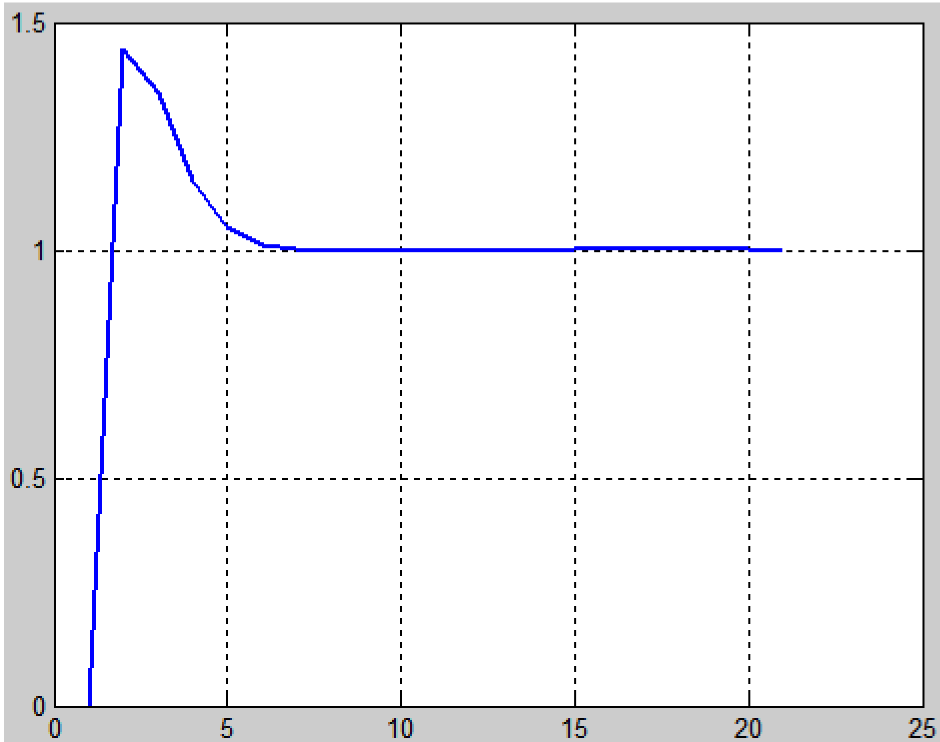


Fig. 2.3.4 Step response of the system

As we see in the Fig. 2.3.4, there is one sample delay in the beginning. Then after one time of obvious oscillation, the system trends to its steady-state gain, which is 1.

In the tank model, we also do the test for this system. After setting $K = 0.9, T_i = 1.67$, we give a step change of the reference level to the system. The following Fig. 2.3.5 shows the response:

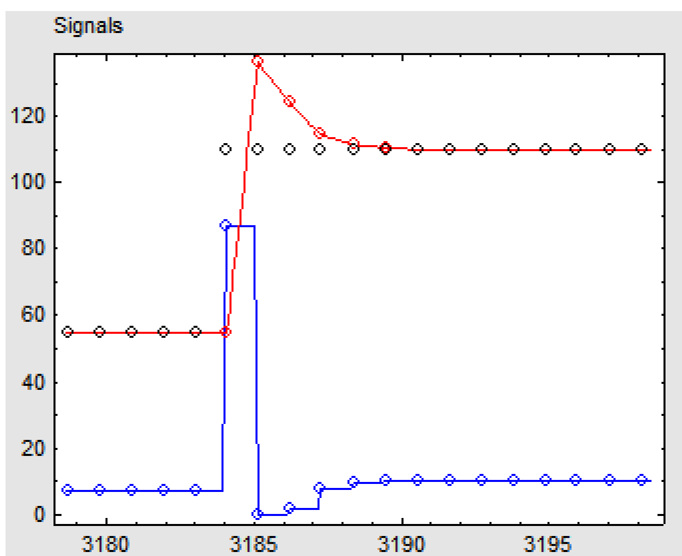


Fig. 2.3.5 Response of Ziegler-Nichols tuning PI-control system to a step change input

As we can see, the response is very similar with the step response curve which we plotted in Matlab.

2.3.4 Anti-windup for PI-control

Now we make a large setpoint change. We try to fill up to a level very close to the top. Then we get the following result:

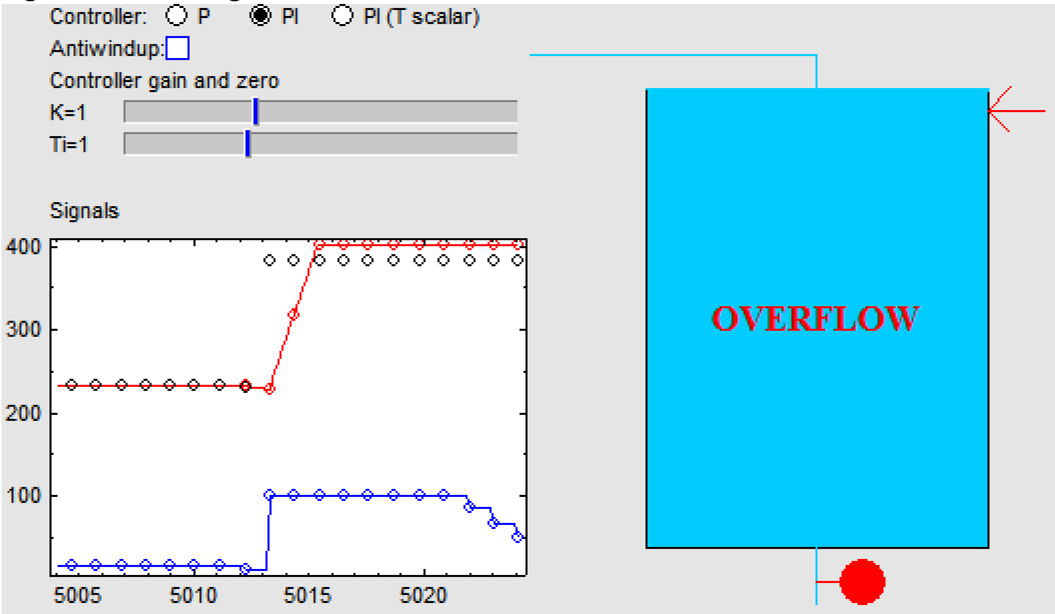


Fig. 2.3.6 Response of the tank PI-control system to a reference level near the top of the tank

As we can see, the tank gets overflow. Also, we try to empty from a full tank to a level close to the bottom.

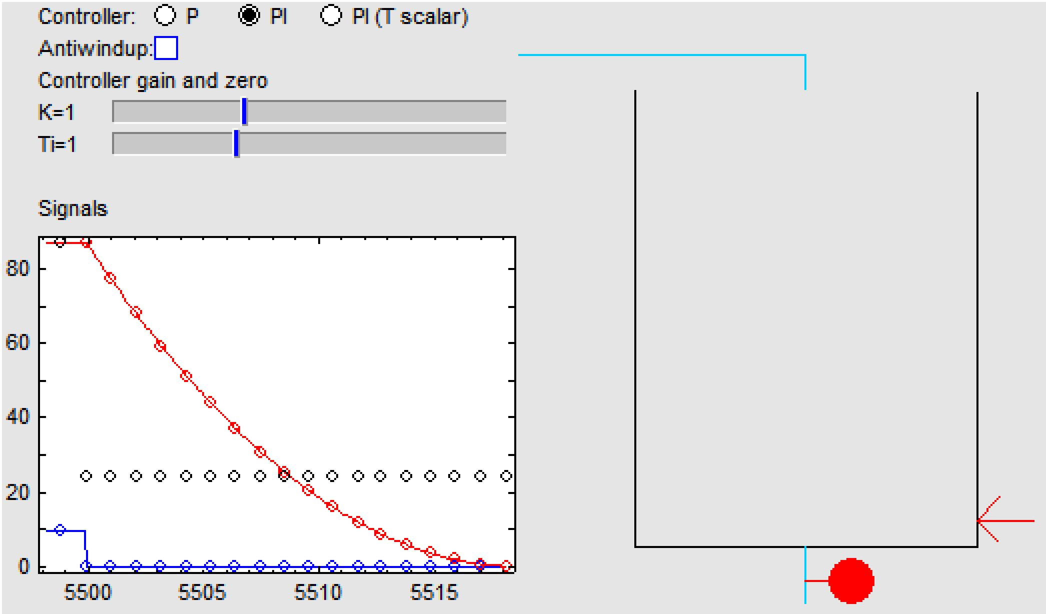


Fig. 2.3.7 Response of the tank PI-control system to a reference level near the bottom of the tank

As we can see, the tank gets empty this time.

This is because the limitation of the control signal causes integrator windup. To improve this, we should set a boundary for the control signal $u(k)$:

$$v(k) = sat[u(k)] = \begin{cases} u_{min} & u(k) < u_{min} \\ u(k) & u_{min} < u(k) < u_{max} \\ u_{max} & u(k) > u_{max} \end{cases}$$

Each time we should calculate the maximum/minimum $u(k)$ we can use. Once $u(k)$ gets out of the boundary, we should make the $u(k)$ just on the boundary.

After we add the anti-windup feature to the system, we do the test again. The following Fig. 2.3.8 shows the result:

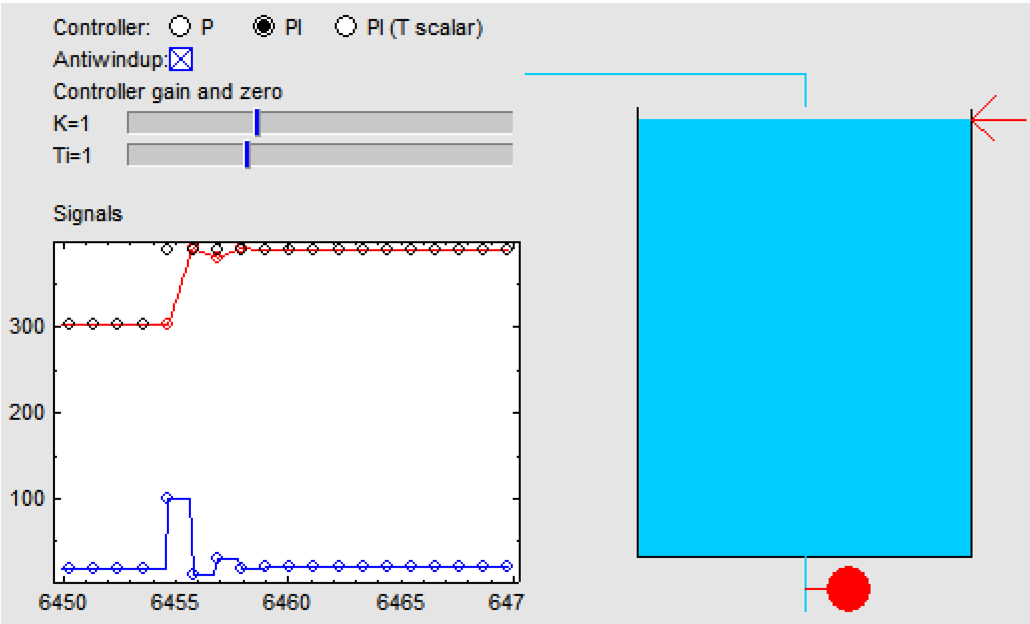


Fig. 2.3.8 Response of the tank PI-control system with the feature of anti-windup to a reference level near the top of the tank

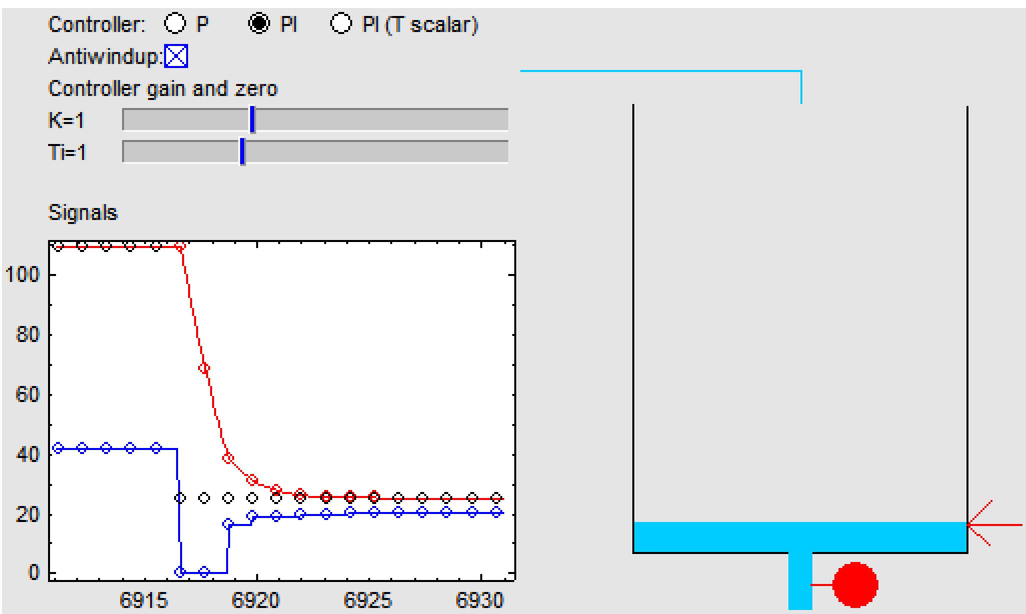


Fig. 2.3.9 Response of the tank PI-control system with the feature of anti-windup to a reference level near the bottom of the tank

Form the red line in the Fig. 2.3.9, we can see that now the system has the feature of anti-windup, which will prevent the output of the system getting out of the limitation boundary.

2.3.5 T-scalar PI-control

As we discussed in 2.3.1, we get the propagation function for the whole system:

$$y = \frac{(1-q^{-1})Cd + [\left(K + \frac{K}{T_i}\right)q^{-1} - Kq^{-2}]r}{1 - \left(2 - K - \frac{K}{T_i}\right)q^{-1} + (1-K)q^{-2}}$$

Now we do some little modification. We change the coefficient of r to $\frac{K}{T_i}q^{-1}$. So we get:

$$y = \frac{(1-q^{-1})Cd + \frac{K}{T_i}q^{-1}r}{1 - \left(2 - K - \frac{K}{T_i}\right)q^{-1} + (1-K)q^{-2}}$$

Next we will test how this change affects the performance of the system. We will use both Dead-beat tuning rule and Ziegler-Nichols tuning rule separately to assign different values for the two parameters K and T_i.

First we use the Dead-beat tuning rule, which means to make $K = 1, T_i = 1$.

$$y = (1-q^{-1})Cd + q^{-1}r = r(k-1) + C[d(k) - d(k-1)]$$

The difference between the previous PI system and the T-scalar system is that in previous PI system,

$$y(k) = 2r(k-1) - r(k-2) + C[d(k) - d(k-1)]$$

One with $r(k-1)$ while another one with $2r(k-1) - r(k-2)$. Although in most of the time, signal $r(k)$ does not change a lot, but in previous PI-control system, once it changes, $r(k-1) \neq r(k-2)$ at that point, which will cause an overshoot. However, in T-scalar PI-control system, this condition never occurs because $y(k)$ always follows $r(k-1)$.

To verify this, we set the correct value to the tank model, and then give a step change of the reference level. The Fig. 2.3.10 shows the response:

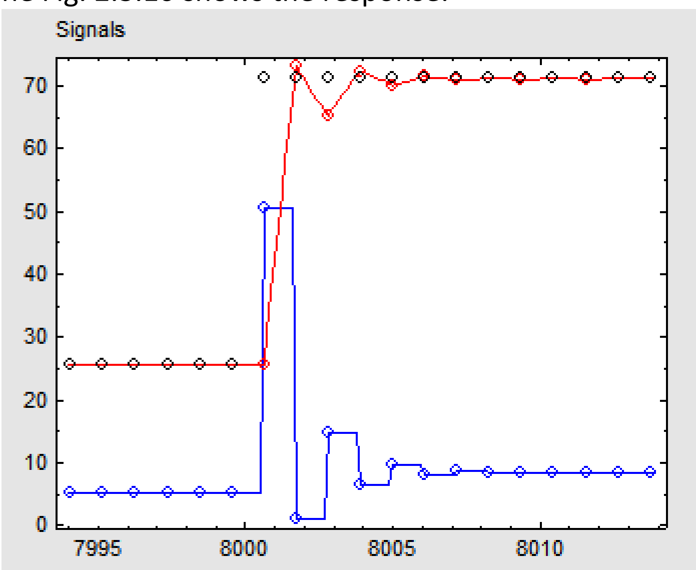


Fig. 2.3.10 Response of the dead-beat tuning T-scalar PI-control system to a step change input

As we see, the overshoot almost disappears, which is what we expect.

Secondly, we use the Ziegler-Nichols tuning rule, which means to make $K = 0.9, T_i = \frac{5}{3}$.

$$y = \frac{(1-q^{-1})Cd + 0.54q^{-1}r}{1-0.56q^{-1}+0.1q^{-2}}$$

Without the disturbance, the step response of the system $G(q^{-1}) = \frac{0.54q^{-1}}{1-0.56q^{-1}+0.1q^{-2}}$ is:

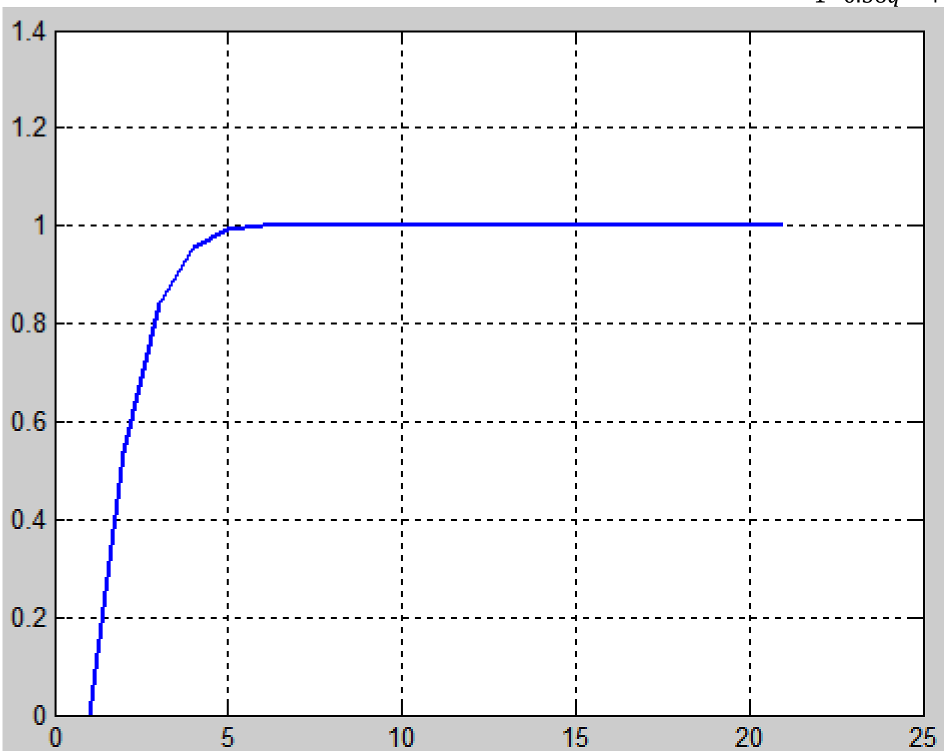


Fig. 2.3.11 Step response of the T-scalar system

Comparing the Fig. 2.3.11 and Fig. 2.3.4, it is obvious that the peak disappears, which also means that there will be no overshoot in step response.

We also do the verification in the tank model which is shown in Fig. 2.3.12.

As we see in the figure, the fact is also that the overshoot also disappears, which is what we expect.

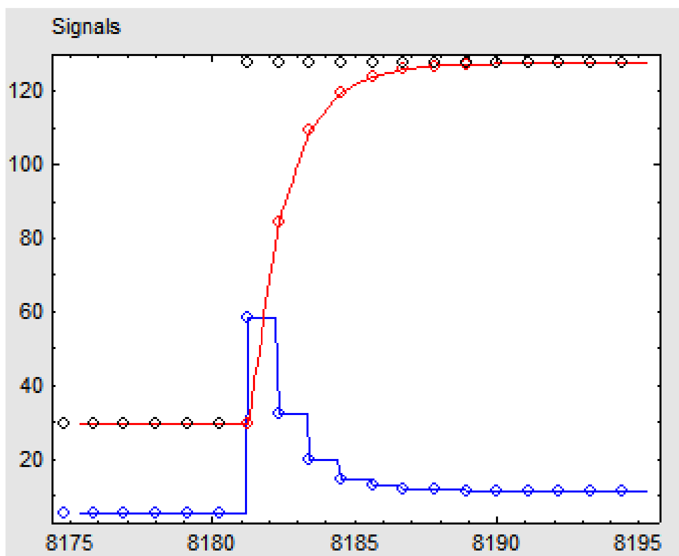


Fig. 2.3.12 Response of the Ziegler-Nichols tuning T-scalar PI-control system to a step change input

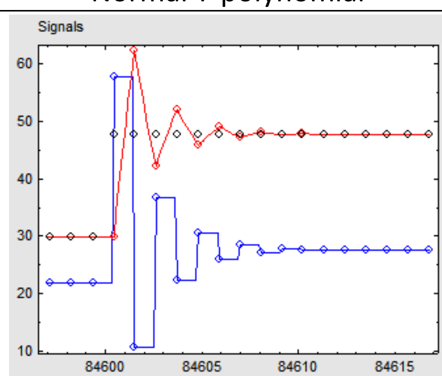
3. Conclusion

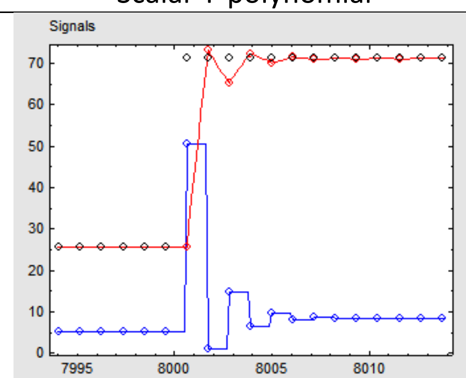
In this exercise, we analyzed P-controller and PI-controller. The conclusion we get can be shown in the following two tables:

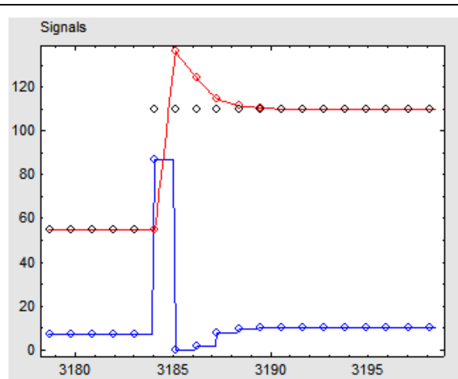
In general:

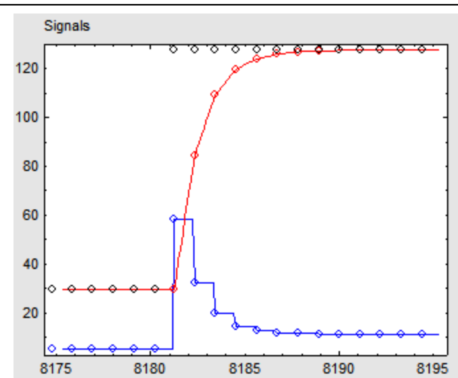
	Automatically follow the input requirement	Anti-disturbance
Manual Control	No	No
P-controller	Yes	No
PI-controller	Yes	Yes

In PI-controller:

<table>
<tr><th></th><th>Normal T-polynomial</th><th>Scalar T-polynomial</th></tr>
<tr><td>Dead-beat tuning rule</td><td>


Oscillate strongly with overshoot</td><td>


Oscillate without overshoot</td></tr>
<tr><td>Ziegler-Nichols tuning rule</td><td>


Oscillate lightly with overshoot</td><td>


Monotonously without overshoot</td></tr>
</table>

Reference

[1] PID controller. http://en.wikipedia.org/wiki/PID_controller