Digital Control: Exercise 3

Xu Fei feixu12@student.hh.se

1. Manual control.

In this question, we just get some feeling how the tank process will work and how the different inputs will affect the output. Here we choose the option P as controller. The value u_0 can be regard as the balance point for the system.

a) Choose the gain of the P-controller $K = 0$ to turn off the proportional feedback. The process in now running in open loop. The system structure is as following figure.

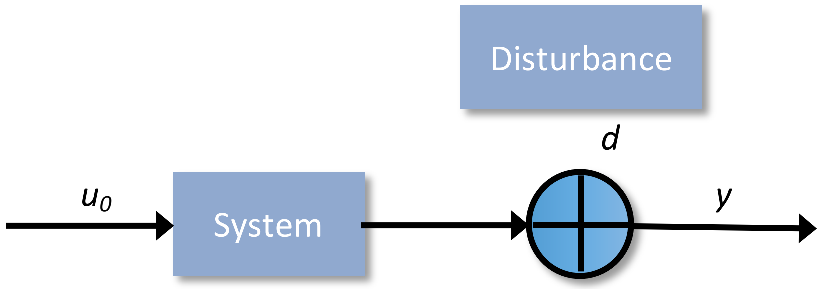

Figure 1.1 Manual control system

We can see from this figure, the system is only affected by u_0 manually and disturbance produced by the drain valve.

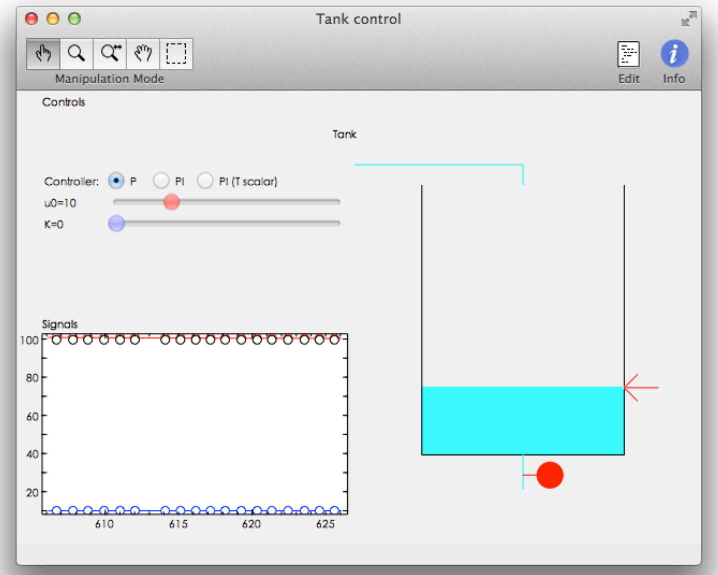

Figure 1.2 P-Controller with reference level = 100

We can see from the figure clearly, when the balance point is 100, $u_0 \approx 10$

can make the system be stable. Then we change the balance point to 150 and 200, and the result is shown as following figure. When the reference level is 150, $u_0 \approx 12.197$ and when the reference level is 200, $u_0 \approx 13.9403$.

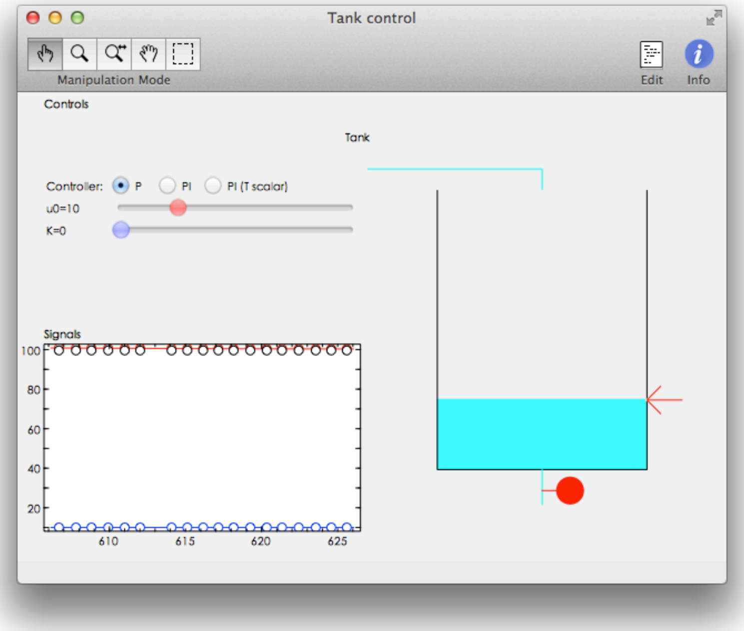

Figure 1.3 P-Controller with reference level = 150

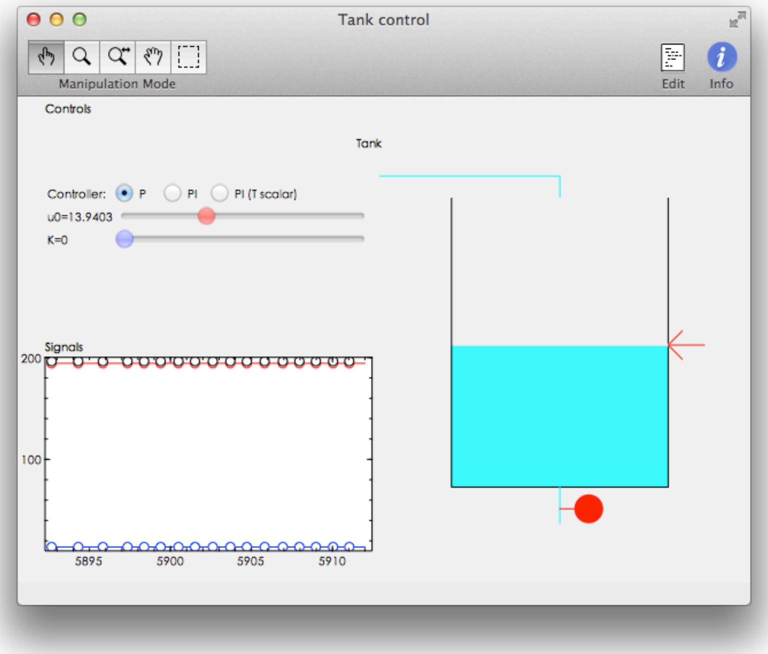

Figure 1.4 P-Controller with reference level = 200

b) Here, I open the red valve to add the disturbance in the system and change the u_0 again to see when the system will balance again. I put the reference level back to 100 again, and the result is shown in figure 2.4. As we seen, the $u_0 = 40$.

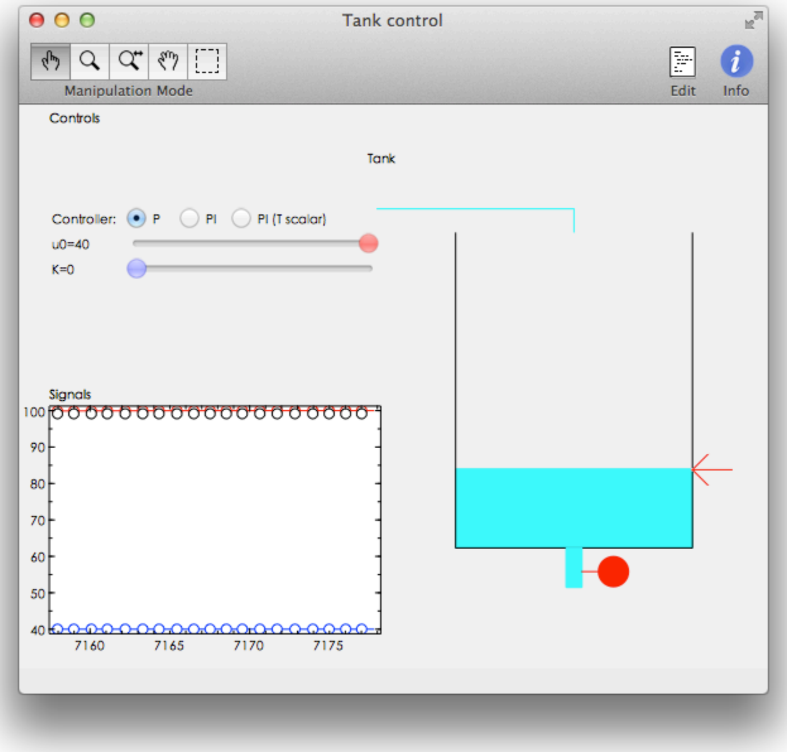

Figure 1.5 P-Controller with reference level = 100 and disturbance

2. P-control.

As we have done with the manual control, we can know that the manual control neither can adjust the reference level automatic nor anti-disturbance. So here I close the valve and choose proportional feedback by increasing K to make the system change to P-control mode.

a) The control signal is now $u = u_0 + K\ (r - y)$, here u_0 is the bias level, and $u = K\ (r - y)$ is the input of syetem. The closed-loop character- istic polynomial is then $A_c = A + BK$ and the closed loop is:

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

Then we can get the structure of the system as figure 2.1

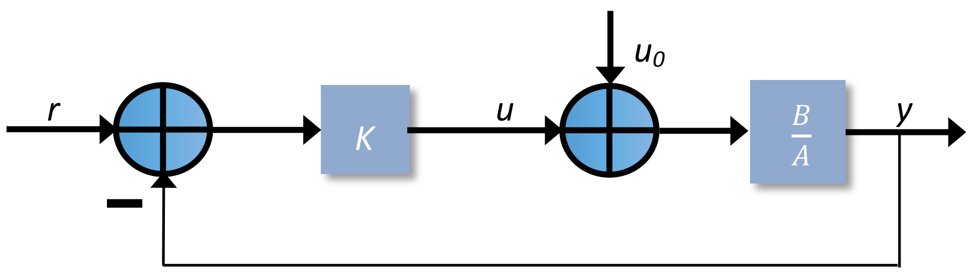

Figure 2.1 P-control system

With this system we can get:

$$y(k)=\frac{BKr(k)+Bu_0}{A+BK}=\frac{Kq^{-1}r(k)+u_0q^{-1}}{1-(1-K)q^{-1}}$$

But here, u_0 is the constant, so we can just ignore it and analysis how u will affect the system.

I choose dead-beat control $K=1$ to make the output track the reference with just one sample delay, $y(k)=u\,(k-1)$. Here we put reference level r from 110 to 140. The result is shown is figure 2.2.

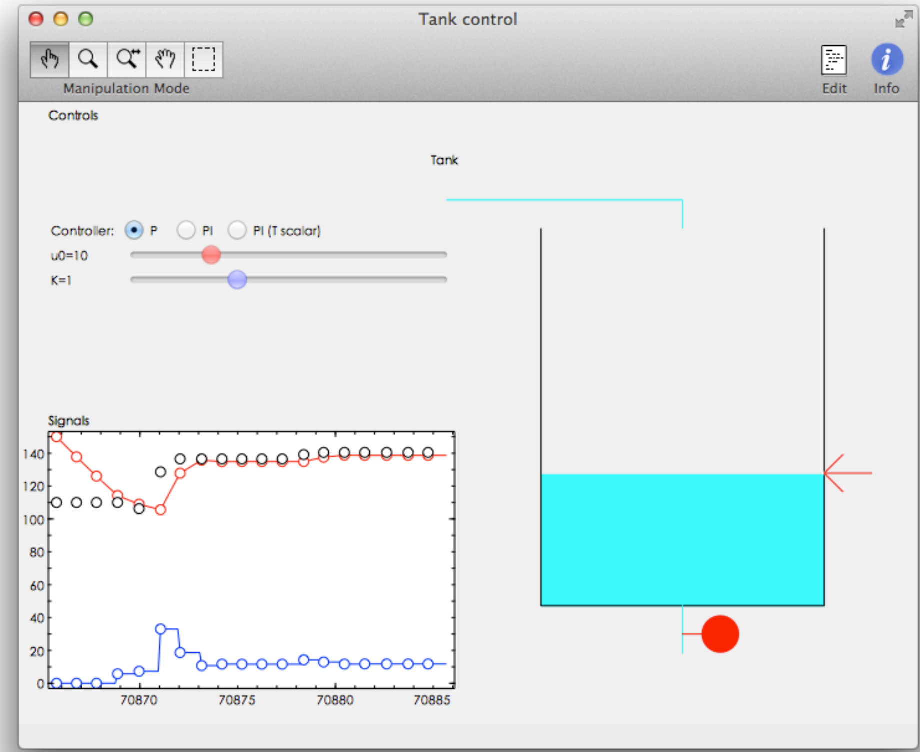

Figure 2.2 P-controller with $r=100$ and $K=100$

We can see the system will follow the changing automatically.

b) Choose different gains to verify that the closed-loop system behaves like a first order system.

The system propagation function is:

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

As we regard this system as a first order system. The pole of this system is $(1-K)$. When $0<K<1$, the step response of the system is shown in figure 2.3.

Here, I set $K=0.2$, so $pole=0.8$. We can see clearly from the figure, the step response is monotonous and get steady-state gain after 25.

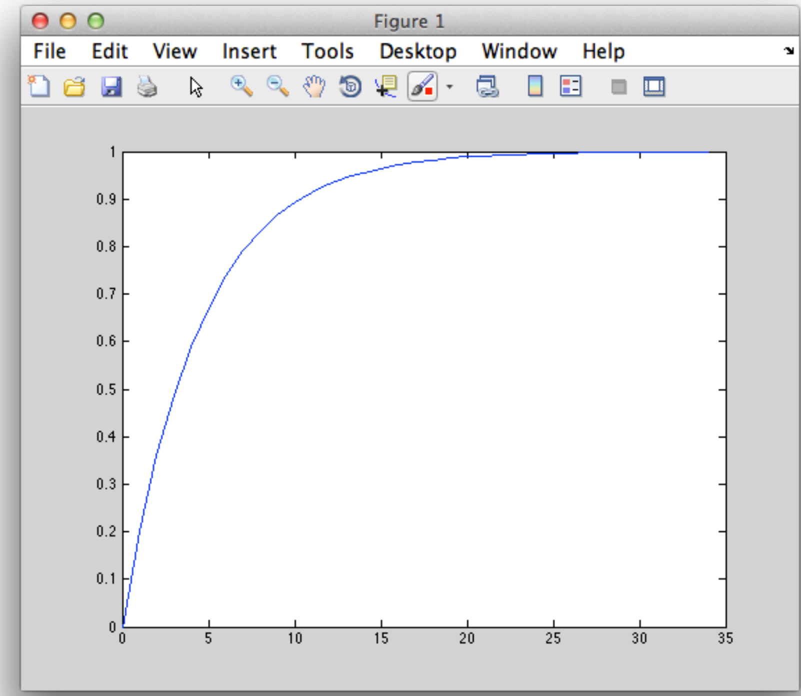

Figure 2.3 Step reponse when $K = 0.2$

The z-plane figure is shown in figure 2.4. We can see the pole point is on the right side of unit circle.

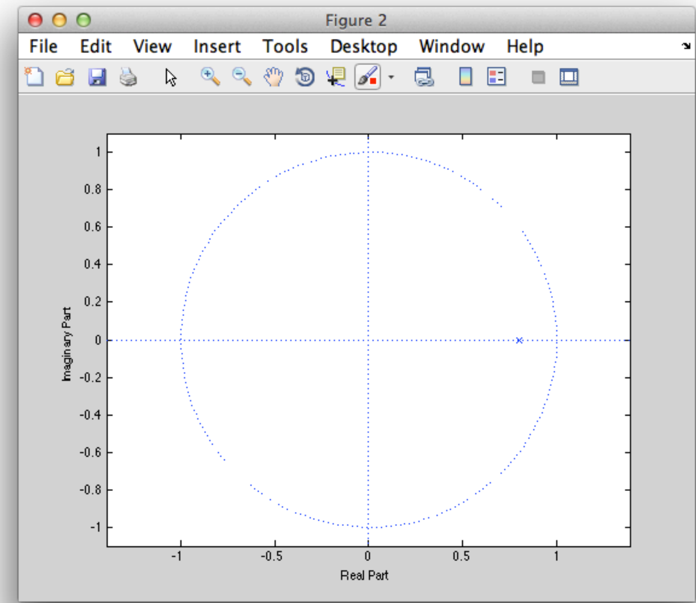

Figure 2.4 z-plane plot when $K = 0.2$

When $1 < K < 2$, $-1 < pole < 0$, the step response is shown is in figure 2.5, z-plane plot is shown in figure 2.6.

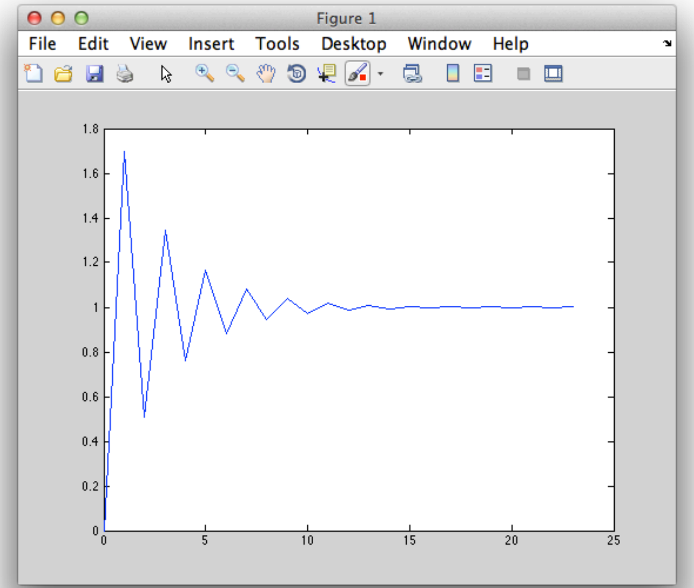

Figure 2.5 Step reponse when $K = 1.7$

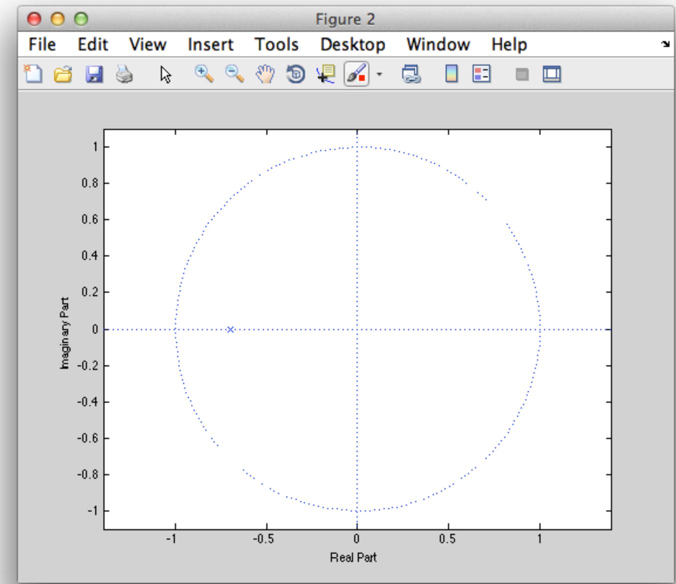

Figure 2.5 z-plane plot when $K = 1.7$

The results are as I expected, the step response is oscillating and get to steady-state gain after 15, the pole point is on the left side of unit circle.

To confirm this, we also do this in tank system, we set $K = 0.2$, $u_0 = 10$ and put the reference level r from 50 to 100, the result is shown in figure 2.7. And then we set $K = 1.7$, $u_0 = 10$ and also put the reference level r from 50 to 100, the result is shown in figure 2.8.

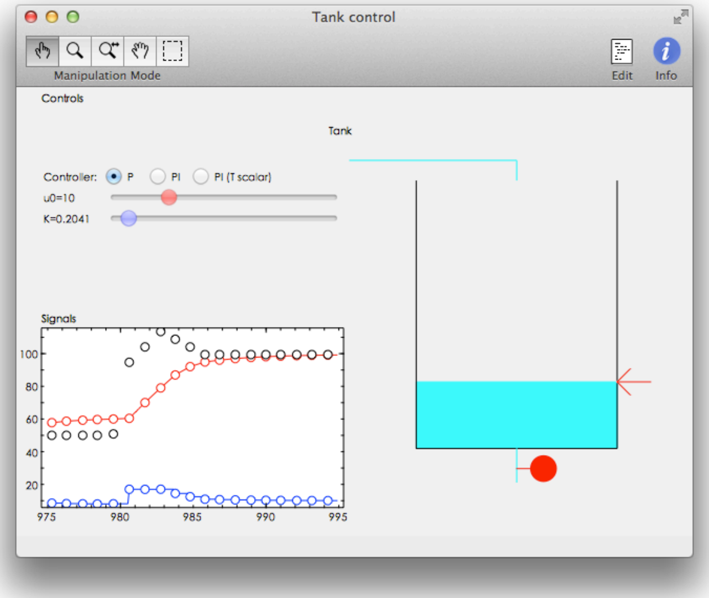

Figure 2.8 P-control with $K = 0.2,\ u_0 = 10,\ r$ from 50 to 100

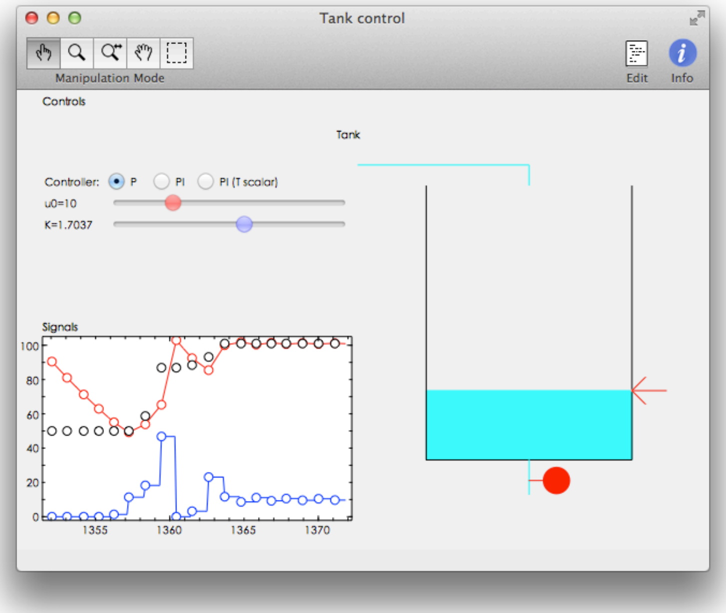

Figure 2.9 P-control with $K = 1.7,\ u_0 = 10,\ r$ from 50 to 100

The result are as I expect, when $K = 0.2$, the water level will increase monotonous until get to reference level, when $K = 1.7$, the water level increase oscillately until get to reference level.

c) The system in a) and b) are without disturbance, now we want to add the disturbance into the system, the model is

$$\begin{cases} Ay = Bu + Cd \\ u = K(r - y) \end{cases} \rightarrow \ y = \frac{BK}{A_c} r + \frac{C}{A_c} d$$

The block scheme is as following:

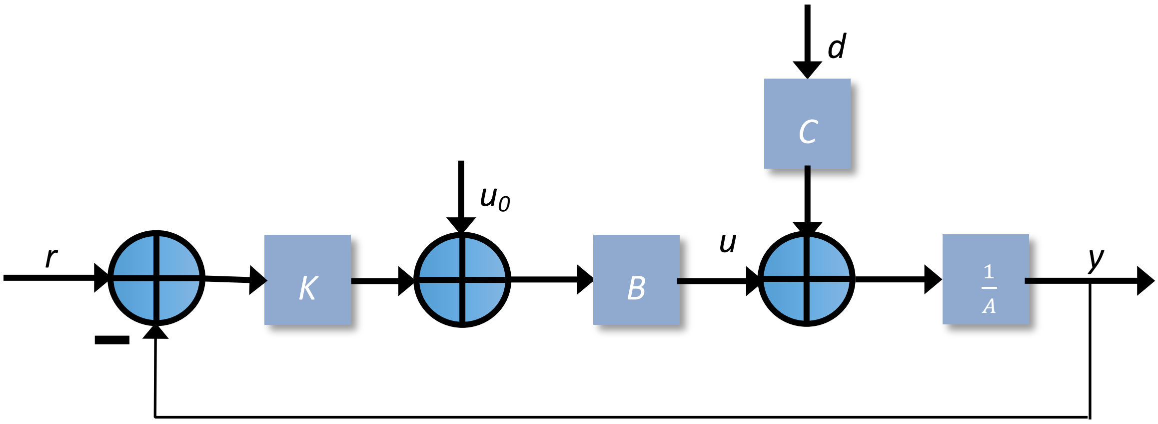

Figure 2.10 P-control system with disturbance.

And we adjust u_0 to make $r = y$. Now the r can be regard as a constant, and the system become:

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

As we have known from b), when $0 < K < 1$, the system will change monotonous until get to reference level but depending on disturbance and when $1 < K < 2$, the system will change oscillately until get to reference level but also depending on disturbance. As a result of the disturbance step (opening of the valve) there will be a bias (stationary error).

To verify this, I set reference level to $r = 150$, $K = 1$, $u_0 = 11.5348$, the result is shown in figure 2.11.

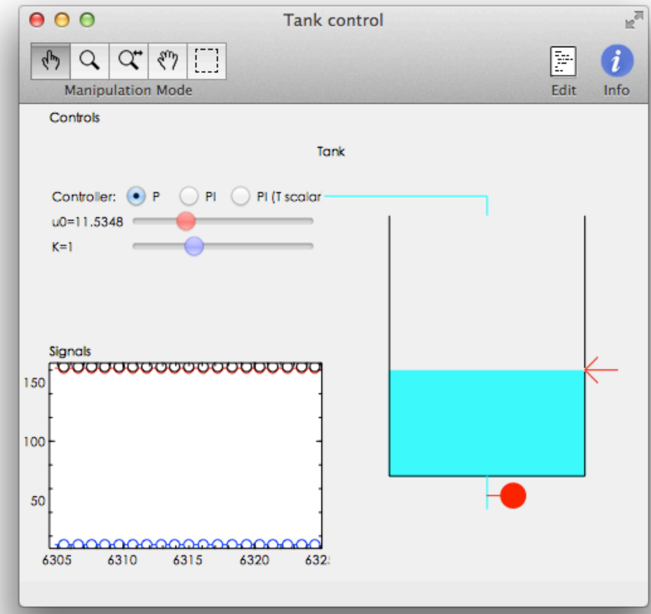

Figure 2.11 P-control with $r = 150$, $K = 1$, $u_0 = 11.5348$

We can see in this figure, the water level will get the reference level and stable on

150. Then I open the red valve and keep the other parameter, the result is shown in figure 2.12.

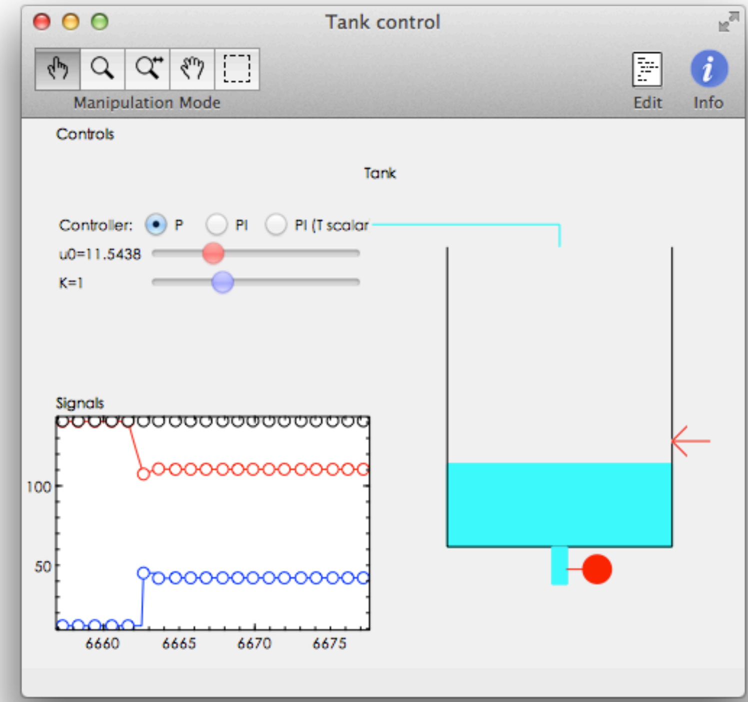

Figure 2.12 P-control with $r = 150,\ K = 1,\ u_0 = 11.5348$ and disturbance

We can see from this figure, the water level will follow the disturbance and decrease until almost 110 and be stable again.

3. PI-control.

From part 2 we can know, P-control can follow the reference level automatically, but the defect is also obviously, it can't anti-disturbance. So, here we will solve this problem by using PI-control.

a) We change the option to PI, and unmark the option antiwindup, now the system should be:

$$R(q^{-1})u(k) = -S(q^{-1})y(k) + T(q^{-1})r(k), \begin{cases} R(q^{-1}) = 1 - q^{-1} \\ S(q^{-1}) = s_0 + s_1 q^{-1} \\ T(q^{-1}) = S(q^{-1}) \end{cases}$$

Or equivalently in recursive form

$$u(k) = u(k-1) - s_0 e(k) - s_1 e(k-1), \quad e(k) = r(k) - y(k)$$

The pole of system should be at 1(integral action) and the zero of system at $z = -s_1/s_0$. The discretization of the PI-control structure is by backward-difference approximation, which is as the following equation:

$$s_0 = K(1 + 1/T_i)$$
$$s_1 = -K$$

The parameterization of the controller makes it possible to change the gain K and the integration time T_i interactively. And now the bias u_0 is unnecessary in the system because the integral action.
The closed-loop characteristic polynomial is:

$$A_c = AR + BS = (1 - q^{-1})^2 + q^{-1}(s_0 + s_1 q^{-1})$$
$$= 1 - (2 - s_0)q^{-1} + (1 + s_1)q^{-2}$$

The system block scheme is shown as following figure:

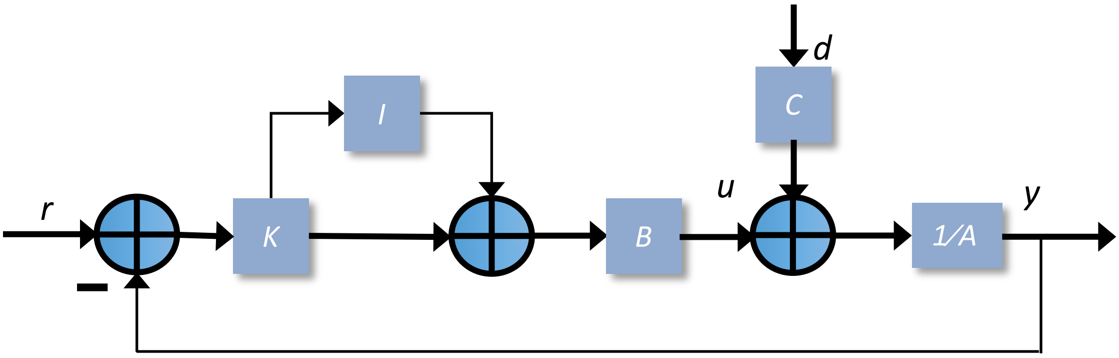

Figure 3.1 PI-control systems

The integral part:

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

Now the propagation function for the whole PI-control system is:

$$Ay = Cd + Bu = Cd + B(k(1 + I)(r - y))$$

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

Dead-beat design corresponds to placing both poles at the origin, so that's means we need make $\begin{cases} \left(2 - K - \frac{K}{T_i}\right) = 0 \\ (1 - K) = 0 \end{cases}$, then we can easily get:

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

Then can simplify as:

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

The differential equation is:

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

As we already know from part2, the reference level doesn't change most time, so $r(k-1) \approx r(k-2)$, so we can tune the close-loop system into

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

From the equation, we can know that the response of system will follow the input as one-step delay and the system is also anti-disturbance, a step disturbance d will be eliminated in only one sample.
To confirm this, I set $K = 1$ and $T_i = 1$, after the system get stable on the reference level, I open the red valve to add one step disturbance into the system,

the result is shown in figure 3.2.

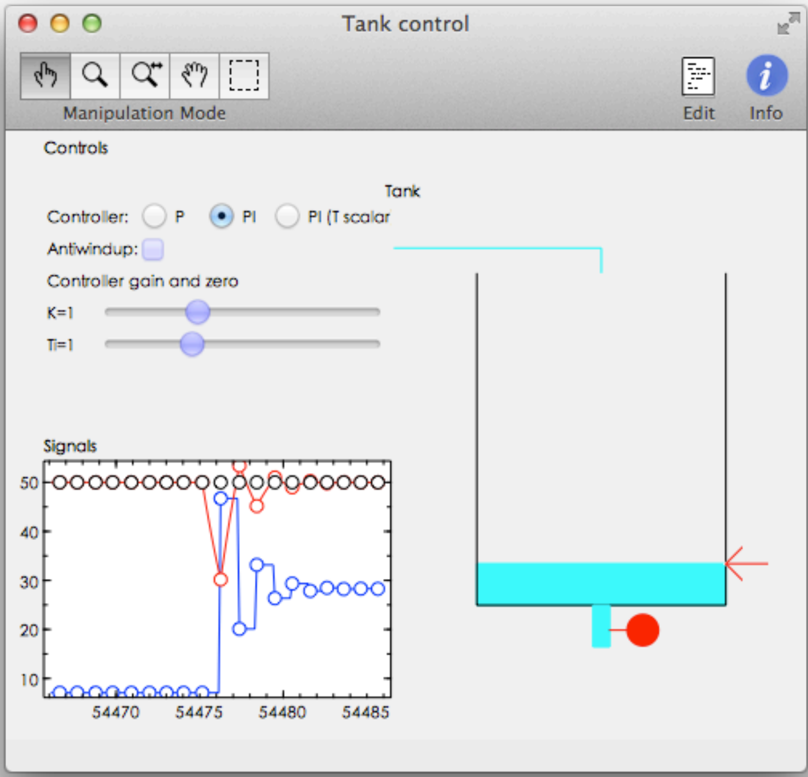

Figure 3.2 PI-control with $K = 1,\ T_i = 1$ and disturbance

From the result we can see after one step disturbance, the system will feed back to reference level again and become stable at this level.

b) Tune the controller according to the Ziegler-Nichols setting in Problem 1d)

According to Ziegler-Nichols rules,

$$\begin{cases} K = K_c \times 0.45 \\ T_i = T_c / 1.2 \end{cases}$$

As we have already known from part 2, K_c is the critical gain and $K_c = 2,\ T_c$ is the corresponding critical period of the oscillation, $T_c = 2$. So we can get:

$$K = 2 \times 0.45 = 0.9$$

$$T_i = \frac{2}{1.2} = \frac{5}{3}$$

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

To simplify the system, we can ignore the disturbance in this system:

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

The step response of q^{-1} is:

$$G(q^{-1}) = \frac{[1.44q^{-1} - 0.9q^{-2}]}{1 - 0.56q^{-1} + 0.1q^{-2}}$$

Then we use Matlab to simulate the system, the result is shown in figure 3.3.

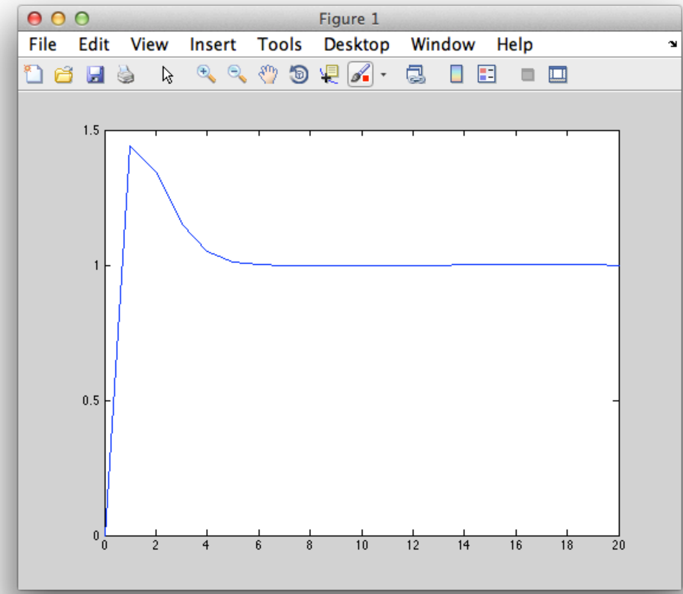

Figure 3.3 Step reponse of PI-control system when $K = 0.9$ and $T_i = \frac{5}{3}$

We can see from the figure that the system have one step reponse and then get stable at the $gain = 1$. $poles = 0.28 \mp 0.1470i$, $A_c = 1 - 0.56q^{-1} + 0.1q^{-2}$. Then I use the tank system to verify this, I set $K = 0.91$ and $T_i = 1.67$, and get the system one step change from 50 to 100, the result is shown in figure 3.4.

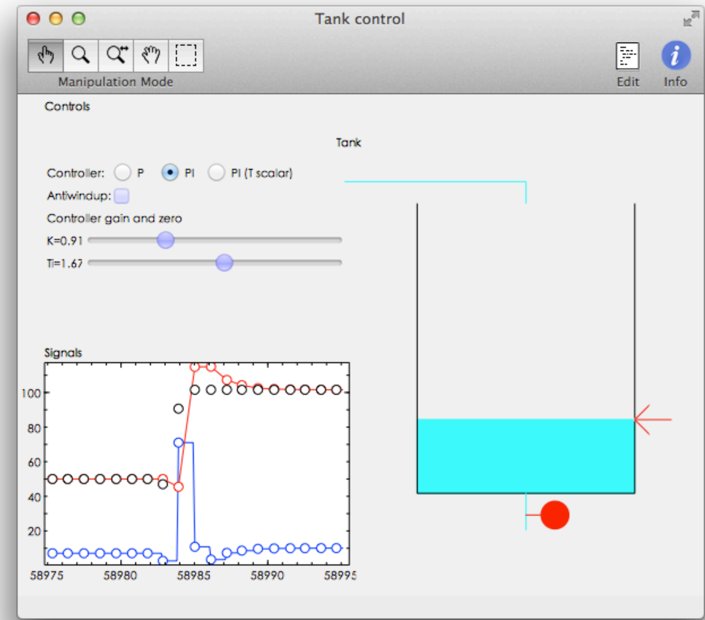

Figure 3.4 PI-control with $K = 0.91$, $T_i = 1.67$

From figure 3.4 we can see in tank system the step response is just as I expect, the signal has one step delay and get stable at the reference level just as the result that we simulate in Matlab.

c) Here, I make a large set point change, first I try to fill up to a level very close to the top and then I try to empty from a full tank to a level close to the bottom.

I set $K = 1,\ T_i = 1$. The result is shown in figure 3.5 and figure 3.6.

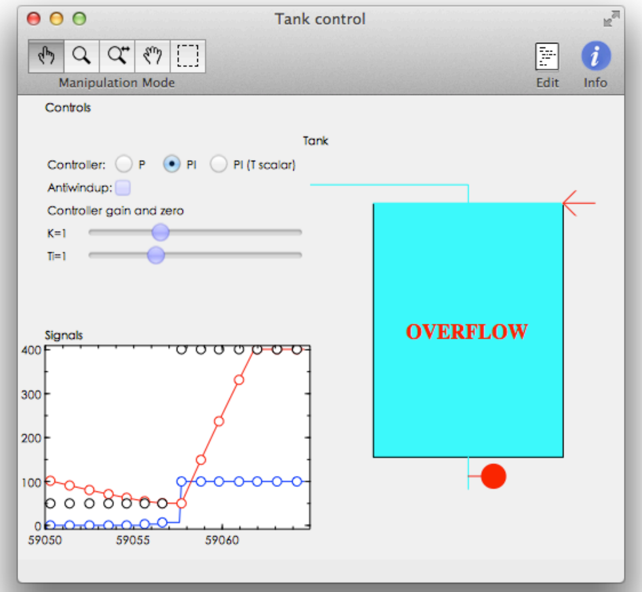

Figure 3.5 PI-control and fill up to the top

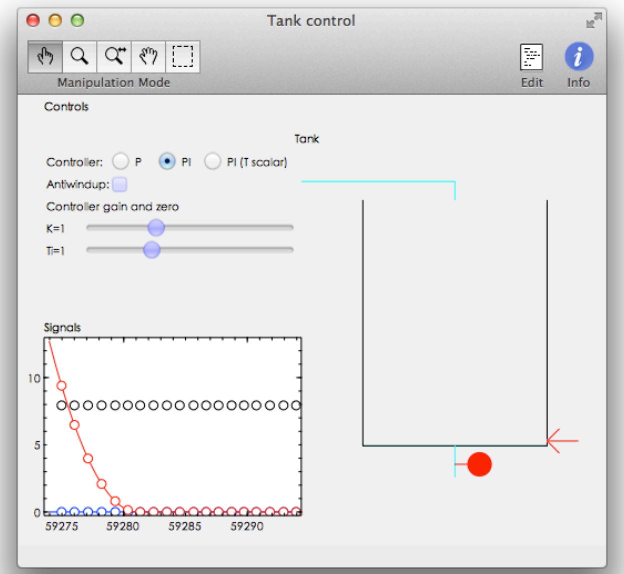

Figure 3.6 PI-control and empty to the bottom

These two results are cause by integrator windup, to solve this problem, we need to add another boundary control signal $b(k)$, and $b(k)$ is:

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

The equation means that when $u(k)$ over the minimal or maximal boundary, we set $u(k)$ correspondingly equal to u_{min} or u_{max} to limit it back in the range.

To verify this I use tank system, choose PI-control mode and mark the option antiwindup. K and T_i are still as before. The result is shown in figure 3.7 and figure 3.8.

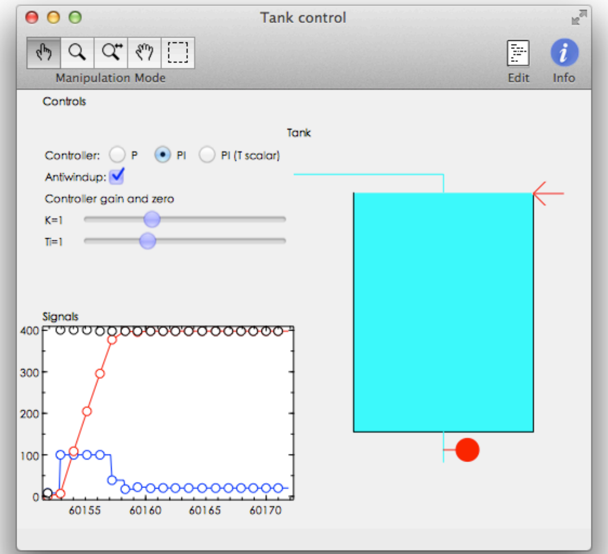

Figure 3.7 PI-control and fill up to the top with antiwindup

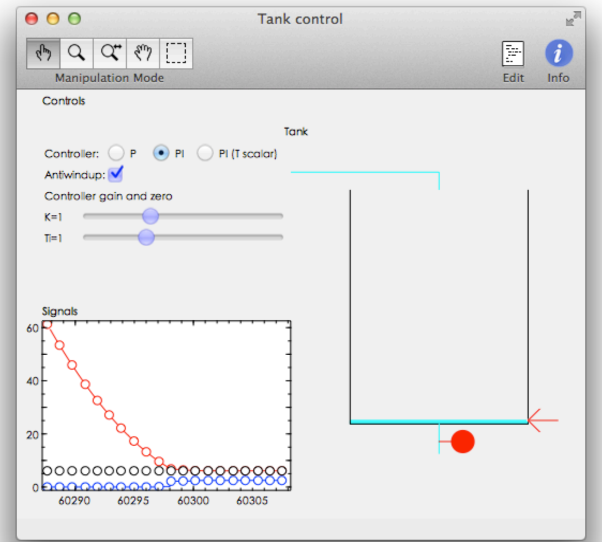

Figure 3.8 PI-control and empty to the bottom with antiwindup

From figure 3.7 and figure 3.8 we can see, now the system can limit the $u(k)$ in the range.

d) Here, we change the controller T-polynomial to the scalar $T = S(1) = s_0 + s_1$. As the result we get in a), the whole system is:

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

$$s_0 = K(1 + 1/T_i)$$

$$s_1 = -K$$

So, we can get $S(1) = K/T_i$, then the whole system become:

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

Then I use both Dead-beat tuning rule and Ziegler-Nichols tuning rule separately to analysis the system.

First I use Dead-beat tuning rule, K and T_i are set as $K = 1$, $T_i = 1$. Then we can get:

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

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

Before in a) part, we have said r almost doesn't change in the process, and we regard $r(k-1) \approx r(k-2)$, but here, even r doesn't change a lot in the whole process, but it still will make a little overshoot at the changing time instance. But, in scalar system, it will never happen because this system always follows $r(k-1)$. To verify this, I use tank system and choose PI-control (T scalar) mode, set $K = 1$, $T_i = 1$, the result is shown in figure 3.9.

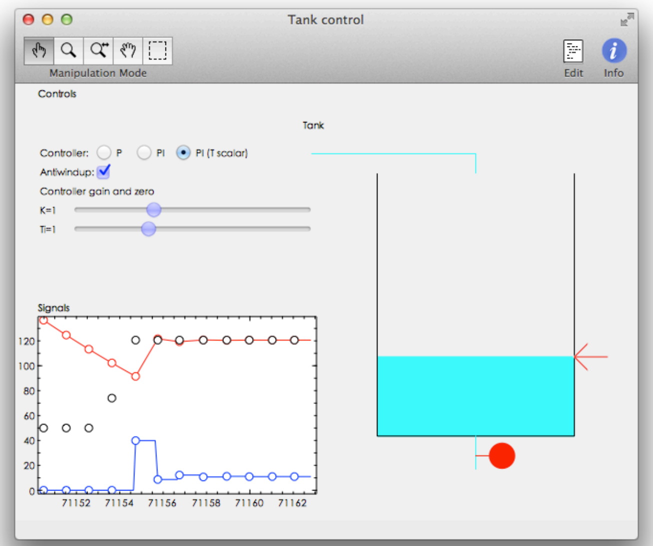

Figure 3.9 PI-control (T-scalar) system when $K = 1$, $T_i = 1$

From the figure we can see there's no overshoot during the changing anymore. Then, I use Ziegler-Nichols tuning rule, K and T_i are set as $K = 0.9,\ T_i = 1.67$, then we can get,

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

Here, we still can ignore the affection by disturbance. The system become:

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

The step response of the system is:

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

The result I simulate in Matlab is shown in figure 3.10. To verify this, I use tank system and choose PI-control (T scalar) mode, set $K = 0.9,\ T_i = 1.68$, the result is shown in figure 3.11.

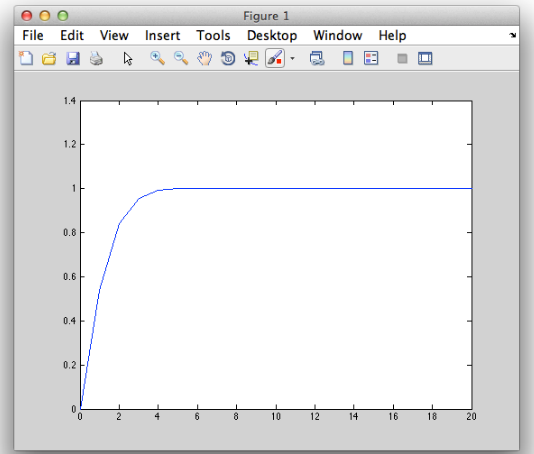

Figure 3.11 Step reponse of PI-control (T-scalar) system

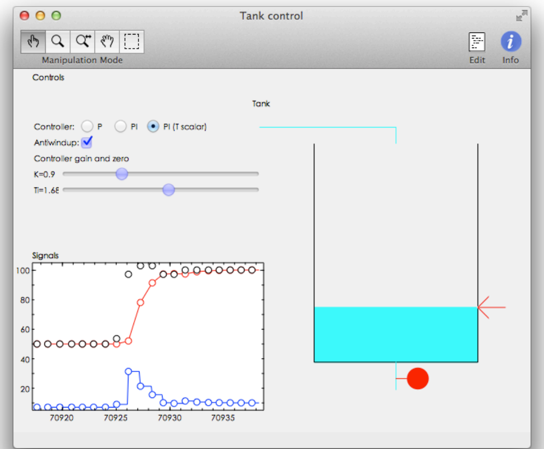

Figure 3.9 PI-control (T-scalar) system when $K = 0.9,\ T_i = 1.68$

As I expect, the monotonous increase didn't overshoot during the changing.

4. Conclusion.

We have analysis manual system, P-control and PI-control system in this exercise, the conclusion is as following table:

Table 4.1 Difference between different control system

	Automatic follow the reference level	**Anti-disturbance**
Manual Control	NO	NO
P - Control	YES	NO
PI - Control	YES	YES

Table 4.2 Difference between different T-polynomial in PI-control

	Normal T-polynomial	**Scalar T-polynomial**
Dead-beat tuning rule	Oscillate with overshoot	Oscillate without overshoot
Ziegler-Nichols tuning rule	Monotonously with one step overshoot	Monotonously without overshoot