

Homework 1 in EL2450 Hybrid and Embedded Control Systems

Kartik Chari
960807-0174
kartikc@kth.se

Devendra Sharma
950815-5497
devendra@kth.se

2019-1-23

Task 1

The tap gain is set as 0 because the outflow of water from the upper tank is only via a small hole connected to the lower tank and not via the other pipe to which the tap is connected. It can be modeled as a zero gain as there is zero outflow of water through it.

Task 2

The code for the transfer functions of upper tank, lower tank and the overall system is as follows:

```
numUT = [k_tank]; % Numerator of the upper tank
denUT = [Tau 1]; % Denominator of the upper tank
numLT = [gamma_tank]; % Numerator of the lower tank
denLT = [(gamma_tank*Tau) 1]; % Denominator of the lower tank

uppertank = tf(numUT,denUT); % Transfer function for upper tank
lowertank = tf(numLT,denLT); % Transfer function for lower tank
G = uppertank*lowertank; % Transfer function from input to lower tank level
```

Task 3

The reference signal is a step signal with amplitude 10 and time shifted to the right by 25 sec.

$$u(t) := \text{unitstepfunction}$$
$$\text{reference signal} = 10 * u(t - 25)$$

U_{ss} is the Steady-State Input, which means that U_{ss} is the input to the pump when the 2 tank system is in equilibrium. Similarly, Y_{ss} is the Steady-State Output of the system i.e. it is the height of the lower tank at equilibrium (here, considered as 40).

Here, the system has been linearized around the equilibrium point Y_{ss} , which has been obtained around the input $U = U_{ss}$ and the control system has been designed to control the small deviations around this point. Hence, we need U_{ss} and Y_{ss} .

Task 4

The PID Controller parameters were derived by using the following constants:

$$\zeta = 0.4705; \omega_0 = 0.3147; \chi = 0.1 \quad (1)$$

Thus,

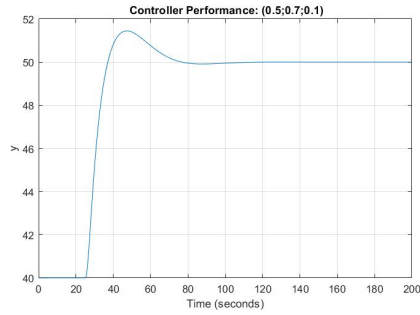
$$K_{pid} = 3.7193; T_i = 17.8170; T_d = 3.1297; N = 0.3353 \quad (2)$$

```
% PID transfer function
n1 = (K*Ti)*((N*Td)+1);
n2 = K*(1+(N*Ti));
n3 = K*N;
numF = [n1 n2 n3]; % Numerator of PID
denF = [Ti N*Ti 0]; % Denominator of PID
F = tf(numF,denF);
```

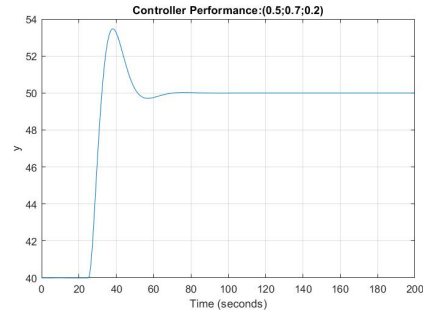
Task 5

χ	ζ	ω	T_r	M	T_{set}
0.5	0.7	0.1	8.8sec	14.4%	45sec
0.5	0.7	0.2	4.93sec	33.7%	25sec
0.5	0.8	0.2	5.02sec	30.5%	26.27sec

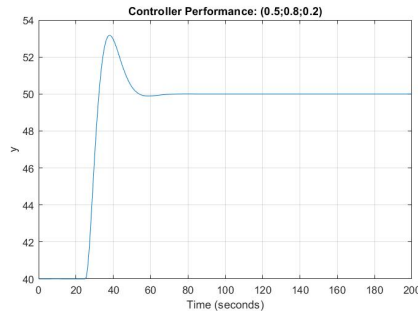
Figure (1) shows the simulation results from which the above time-response characteristics were calculated



(a) Controller Performance for Case 1



(b) Controller Performance for Case 2



(c) Controller Performance for Case 3

Figure 1: **Simulation Results**

The system has some step response characteristics requirements on the closed loop system such as $T_r < 6s$; $M < 35\%$; $T_{set} < 30s$. Thus, from the above table it is evident that the 3rd parameter setting of $\chi = 0.5$; $\zeta = 0.8$; $\omega_0 = 0.2$ gives the best control performance as a control system with less overshoot, small rise-time and small settling-time is desirable.

Task 6

The gain cross over frequency of the open loop system (for the third controller configuration) is 0.362 rad/sec with Phase Margin of 55.3° (see fig. 2). In order to calculate it, we need to analyze the Bode Plot. The frequency at which the open loop gain first reaches 1 is the gain cross over frequency (ω_{gc}) i.e it is the frequency at which the Phase Margin is calculated.

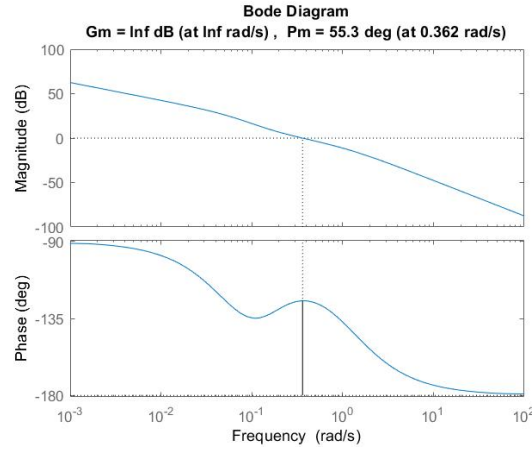
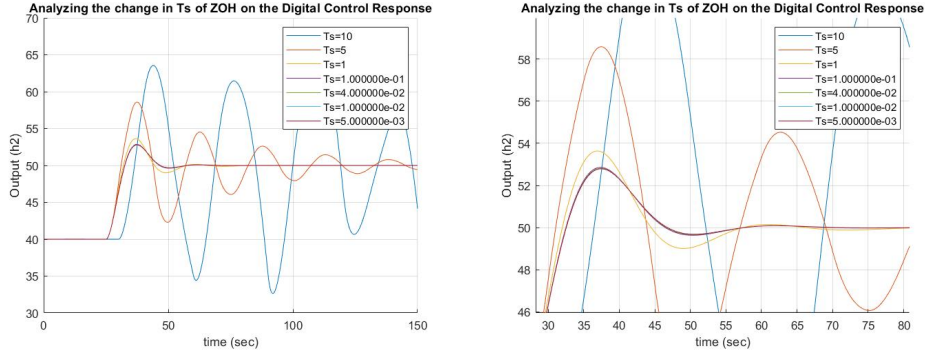
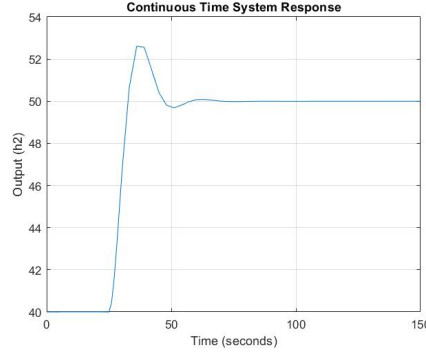


Figure 2: Bode Plot of Open Loop System

Task 7



(a) System Response with ZOH (Var- (b) Peaks of the System Response with
ious T_s) ZOH

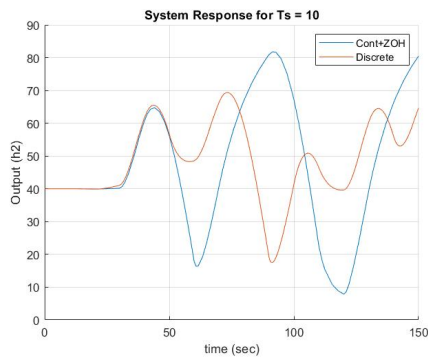


(c) Continuous Time System Response

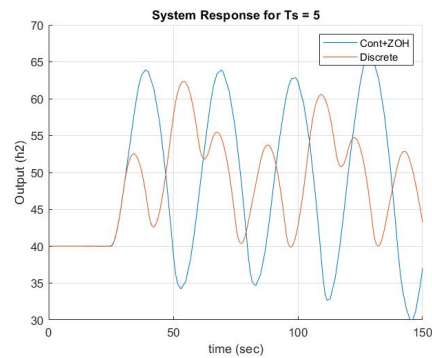
Figure 3: **Differences in Control Performance**

From figure (2), we can see that the system time responses for various sampling times : $T_s = \{10, 5, 1, 0.1, 0.04, 0.01, 0.005\}$ are plotted. It has been observed that for very high sampling time of the ZOH, the time response is very oscillatory with a lot of overshoot . As we further reduce the sampling time, the damping increases leading to reduction in the peak overshoot and the settling time. Around a T_s of 0.005 sec or less, the system starts behaving similar to the continuous time close-loop system but with some more overshoot and delayed settling time mainly due to the delay introduced by the Zero-order Hold.

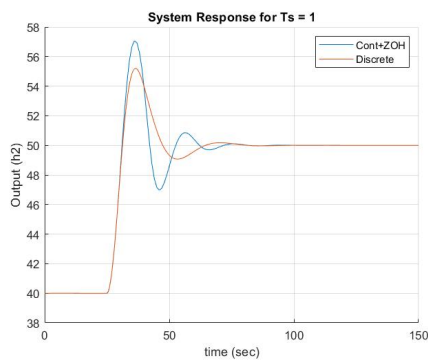
Task 8



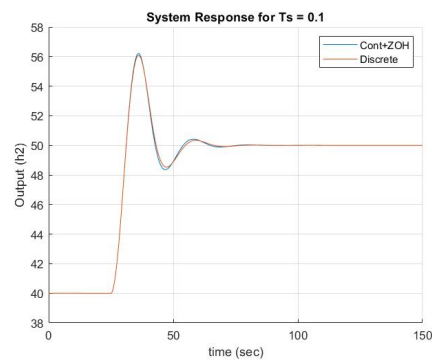
(a) System Response for $T_s = 10$ sec



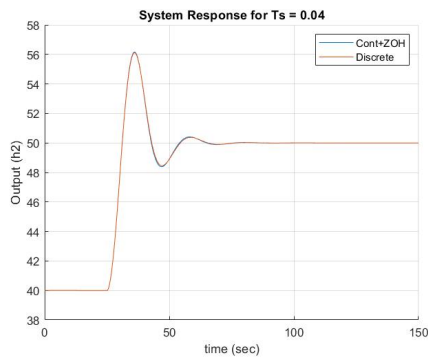
(b) System Response for $T_s = 5$ sec



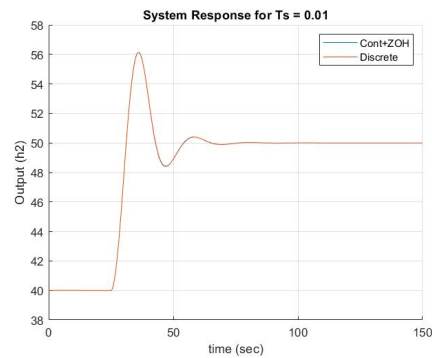
(c) System Response for $T_s = 1$ sec



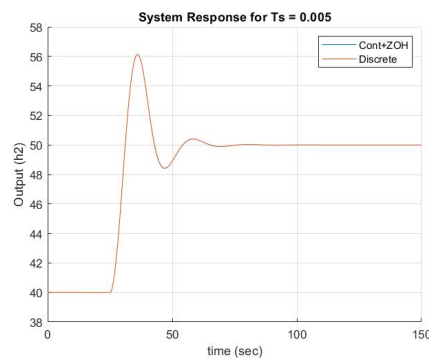
(d) System Response for $T_s = 0.1$ sec



(e) System Response for $T_s = 0.04$ sec



(f) System Response for $T_s = 0.01$ sec



(g) System Response for $T_s = 0.005$ sec

From the simulation results (*see fig.(4)*), we can see that as the Sampling time (T_s) is decreased, the system stability increases which is evident from the sub-figures (a), (b), (c), and (d). For $T_s = 1$ sec, we can see that though the peak overshoot of the discretized controller is lesser than the continuous + ZOH, the response is sluggish. As the T_s is further reduced the Settling Time (T_{set}) of discretized controller reduces and the overall system response comes closer and closer to match that of the Continuous Controller + ZOH. Around $T_s = 0.04$ or lower, we can observe that the response of both the controller overlap each other.

Task 9

When implementing a continuous controller digitally, we should choose the T_s such that

$$T_s * \omega_c \approx 0.05 \text{ to } 0.14$$

In the Task 6, we calculated $\omega_c = 0.362 \text{ rad/sec}$

Thus, T_s lies in the range of ≈ 0.138 and 0.3867

Task 10

Considering the controller parameters with $\chi = 0.5$; $\omega_o = 0.2$; $\zeta = 0.8$

The maximum possible sampling time without affecting the control performance of a system with discretized controller is observed to be around 0.4 sec. But from $T_s = 0.5$ sec to 1.5 sec, the performance of the system deteriorated a little in terms of sluggish nature i.e low T_r and T_{set} but as I increased the T_s above this, it deteriorated exponentially till the point of becoming very oscillatory around $T_s = 5$ sec and diverging above $T_s = 10$ sec. The theoretically calculated maximum T_s was 0.3867 which is very near to what is observed from the simulation results. But depending upon how much compromise in the performance is fine, this upper limit can be extended a little.

Task 11

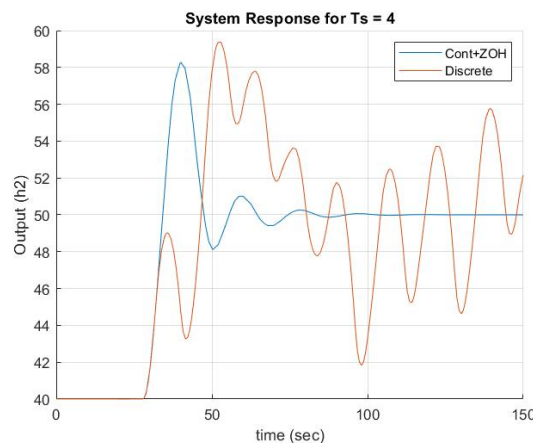


Figure 5: Control Performance for $T_s = 4$ sec

For the mentioned sampling time of 4 sec, the control performance of the discretized controller system is highly oscillatory. On the other hand, if we consider the case with Continuous + ZOH controller, it settles down after some initial oscillations.

Task 12

The Linearized System Equation was first converted from Laplace Domain to Time Domain and then expressed in state-space form to obtain :

$$A = \begin{bmatrix} -1/Tau & 0 \\ 1/Tau & -1/(gamma_tank * Tau) \end{bmatrix}$$

$$B = \begin{bmatrix} k_tank/Tau \\ 0 \end{bmatrix}$$

$$C = [0 \quad 1]$$

$$D = [0]$$

The State-Space form of the Continuous system was discretized with $T_s = 4$ sec and the Discrete-time State Space matrices were derived as follows:

$$\Phi = \begin{bmatrix} 0.7249 & 0 \\ 0.2332 & 0.7249 \end{bmatrix}$$

$$\Gamma = \begin{bmatrix} 0.6017 \\ 0.0916 \end{bmatrix}$$

$$C = [0 \quad 1]$$

$$D = [0]$$

Task 13

The observability and reachability of the discrete-time plant model was checked using MATLAB.

```
% Observability and reachability
Wo = obsv(Phi,C)
isObsv = length(Phi) - rank(Wo)
Wc = ctrb(Phi,Gamma)
isCtrb = length(Phi) - rank(Wc)
```

After executing the above code snippet, we got

$$W_o = \begin{bmatrix} 0 & 0.0142 \\ 0.0142 & 0.0136 \end{bmatrix}$$

$$W_c = \begin{bmatrix} 2.8993 & 1.3037 \\ 6.4741 & 14.6090 \end{bmatrix}$$

Also,

$$isObsv = 0$$

$$isCtrb = 0$$

which means that the Observability matrix and Reachability matrix has full rank and the discrete-time plant is **Observable** and **Reachable**.

Task 14

The reference gain l_r is necessary in order to ensure that the steady state output $y_e = r$ i.e it ensures that we are able to design the dynamics of the system to satisfy our goal.

Task 15

The state-space equation for the dynamic observer is as follows:

$$\Delta \hat{x}(k+1) = \Phi \Delta \hat{x}(k) + \Gamma \Delta u(k) + K [\Delta y - C \Delta \hat{x}(k)] \quad (3)$$

$$\Delta y = C \Delta x(k) \quad (4)$$

$$\therefore \Delta \hat{x}(k+1) = [\Phi - KC] \Delta \hat{x}(k) + \Gamma \Delta u(k) + KC \Delta x(k) \quad (5)$$

$$(6)$$

Now plugging in the Control Law of the form

$$\Delta u(k) = -L \Delta \hat{x}(k) + l_r r(k) \quad (7)$$

We get,

$$\Delta \hat{x}(k+1) = [\Phi - KC - \Gamma L] \Delta \hat{x}(k) + \Gamma l_r r(k) + KC \Delta x(k) \quad (8)$$

The state-space form of the original system is given by

$$\Delta x(k+1) = \Phi \Delta x(k) + \Gamma \Delta u(k) \quad (9)$$

Now, plugging in the Control Law $\Delta u(k)$, we get

$$\Delta x(k+1) = \Phi \Delta x(k) - \Gamma L \Delta \hat{x}(k) + \Gamma l_r r(k) \quad (10)$$

Thus, representing the eq.(5) and eq.(10) in a state-space form, we get

$$\begin{bmatrix} \Delta x(k+1) \\ \Delta \hat{x}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi & -\Gamma L \\ KC & \Phi - KC - \Gamma L \end{bmatrix} \begin{bmatrix} \Delta x(k) \\ \Delta \hat{x}(k) \end{bmatrix} + \begin{bmatrix} \Gamma l_r \\ \Gamma l_r \end{bmatrix} r(k)$$

Comparing this equation with $x_a(k+1) = A_a x_a(k) + B_a r(k)$

We get,

$$A_a = \begin{bmatrix} \Phi & -\Gamma L \\ KC & \Phi - KC - \Gamma L \end{bmatrix}$$

$$B_a = \begin{bmatrix} \Gamma l_r \\ \Gamma l_r \end{bmatrix}$$

Task 16

Let $\Delta \tilde{x}(k) = \Delta x(k) - \Delta \hat{x}(k)$ Thus, substituting for $\Delta \hat{x}(k)$ in eq.(10), we get

$$\Delta x(k+1) = [\Phi - \Gamma L] \Delta x(k) + \Gamma L \Delta \tilde{x}(k) + \Gamma l_r r(k) \quad (11)$$

Now, from eq.(3), eq.(4) and eq.(7), we get

$$\Delta \hat{x}(k+1) = \Phi \Delta \hat{x}(k) - \Gamma L \Delta \hat{x}(k) + \Gamma l_r(k) + KC \Delta \tilde{x}(k) \quad (12)$$

Substituting for $\Delta \hat{x}(k)$ in eq.(12), we get

$$\Delta \tilde{x}(k+1) = [\Phi - KC] \Delta \tilde{x}(k) \quad (13)$$

Expressing the above equations in state-space form, we get

$$\begin{bmatrix} \Delta x(k+1) \\ \Delta \tilde{x}(k+1) \end{bmatrix} = \begin{bmatrix} \Phi - \Gamma L & \Gamma L \\ 0 & \Phi - KC \end{bmatrix} \begin{bmatrix} \Delta x(k) \\ \Delta \tilde{x}(k) \end{bmatrix} + \begin{bmatrix} \Gamma l_r \\ 0 \end{bmatrix} r(k)$$

Let

$$z(k) = \begin{bmatrix} \Delta x(k) \\ \Delta \tilde{x}(k) \end{bmatrix}$$

Thus, the above state-space representation can be re-written as:

$$z(k+1) = \begin{bmatrix} \Phi - \Gamma L & \Gamma L \\ 0 & \Phi - KC \end{bmatrix} z(k) + \begin{bmatrix} \Gamma l_r \\ 0 \end{bmatrix} r(k)$$

Now if we try to find the eigen values of the above system, we get

$$[sI - (\Phi - \Gamma L)][sI - (\Phi - KC)] = 0 \quad (14)$$

Thus pole placement for both can be done independently, which shows that Separation Principle holds.

Task 17

The Simulink model with the Continuous Controller was linearized using the Linear Model Analysis Toolkit and that model was exported to the MATLAB workspace. For this task, we used the 3rd controller specification with $\chi = 0.5$; $\omega_0 = 0.2$; $\zeta = 0.8$. The eigen values of the Linearized Continuous time closed loop system were as follows:

$$p = [-0.5176, -0.4822, -0.1601 + 0.1201i, -0.1601 - 0.1201i]$$

These poles were converted into discrete-time using the equation

$$z_i = e^{T_s p_i} \dots \forall T_s = 4sec. \\ \implies z_i = [0.1261, 0.1453, 0.4675 + 0.2436i, 0.4675 - 0.2436i]$$

Owing to the Separation Principle, the Controller as well as the Observer can be designed independently. Coming to the observer, as the control action depends upon the state estimation by observer, it has to be faster than the controller. Also, theoretically the observer error will decrease if the poles of s-plane are further to the left. On the other hand for the State feedback Controller, the dominant poles in the s-plane are used as they have more impact on the system transient response. The Controller gain **L** and Estimator gain **K** values were calculated using the **acker** function. We have also verified that the **A_a** has the desired poles(z-domain) [0.1261, 0.1453, 0.4675+0.2436i, 0.4675-0.2436i].

% MATLAB code for Task 17

```
p_controller = [0.4675+0.2436i 0.4675-0.2436i];
p_observer = [0.1261 0.1453];
L = acker(Phi,Gamma,p_controller);
M = acker(Phi',C',p_observer);

% reference gain
num = 1;
den1 = C/(eye(2)-Phi+(Gamma*L));
den = den1 * Gamma;
lr = num / den;

% augmented system matrices
Aa = [Phi -(Gamma*L); M'*C Phi-(Gamma*L)-(M'*C)];
Ba = [Gamma.*lr; Gamma.*lr];
```

Task 18

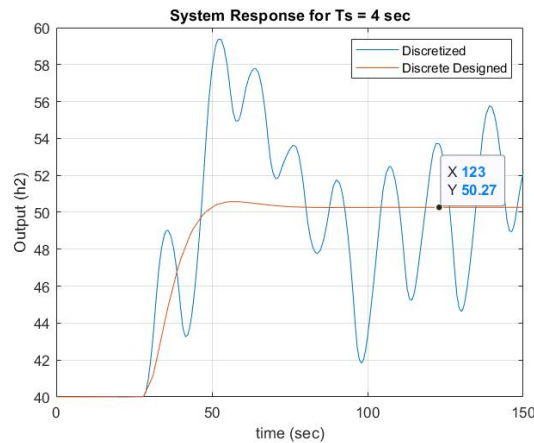


Figure 6: **Control Performance for $T_s = 4$ sec**

From Figure (5), we can see that the performance of the system has improved a lot (oscillations have reduced to a great extent and T_s has also reduced) for the discrete designed controller system. This may be because this controller is specifically designed for the discrete system and has been designed to place the closed loop discrete poles at the desired location by considering State Feedback as well as Estimator design.

The time-response of the discrete designed system settles down having a steady-state error value of **0.27**. There is a small steady-state error in the output which maybe because of the following reasons

- The type number of the system is zero and hence, for a step input the steady-state error is non-zero.
- As the Observer Gain takes large values, there is a possibility of noise amplification.

Task 19

Quantization with 10 bits is $\text{abs}(\frac{0-100}{2^{10}}) = 0.0976$.

Task 20

A Quantization block was successfully constructed in Simulink by connecting a Quantizer block to a Saturation block. The Saturation block is necessary because the only parameter that the Quantizer block has is the Quantization Interval and no upper or lower limit. Thus, in order to limit the signal between the desired limits, we add a Saturation block.

Task 21

Different Quantization levels [12.5, 6.25, 1.56, 0.78125, 0.39, 0.097] corresponding to [3, 4, 6, 7, 8, 10] bits were used to simulate the system.

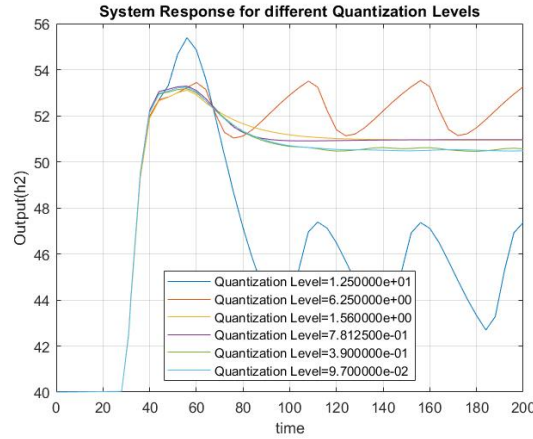


Figure 7: System Response for different Quantization Levels

From figure (6), we can observe that for Quantization levels greater than **0.78125** (corresponding to lower than 7 bits), the control performance starts to degrade. The Simulink model with Quantization Subsystem consisting of Quantizer in series with Saturation is shown below:

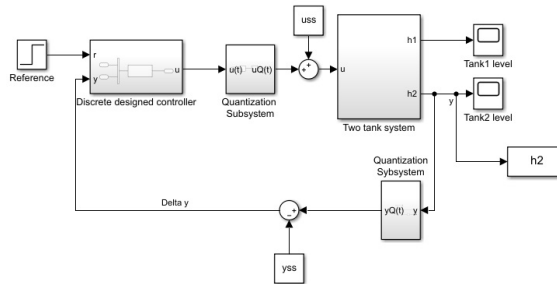


Figure 8: Simulink Tank Model with Quantization blocks