Homework 1 in EL2450 Hybrid and Embedded Control Systems

Ali Ghasemi
19980422-8352
aghasemi@kth.se

Yingzhe Tian
20020428-5571
yingzhet@kth.se

January 27, 2025

Task 1

The gain Tap is modeling the valve, referred to as Tap in Figure 1 of the home assignment. A value of zero indicates that the valve is closed.

Task 2

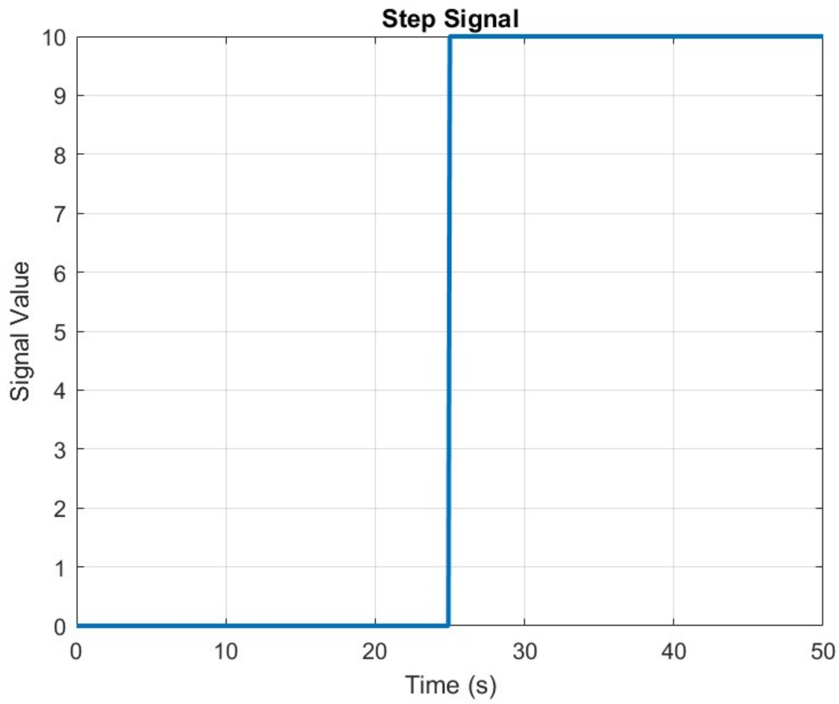
```
uppertank = tf(k_tank,[Tau,1]); % Transfer function for upper tank
lowertank = tf(gamma_tank,[gamma_tank*Tau,1]); % Transfer function for upper tank
G = uppertank*lowertank; % Transfer function from input to lower tank level
```

Task 3

The signal begins at zero seconds with a value of zero. At 25 seconds, the signal jumps to a value of 10. Since the sampling time is zero, the signal is continuous. We can formulate the signal as follows:

$$\text{Reference signal} = 10u(t-25) \tag{1}$$

$$u(t-25) = \begin{cases} 10, & t \geq 25 \\ 0, & t < 25 \end{cases} \tag{2}$$



The variables *uss* (steady-state input) and *yss* (steady-state output) are used to manage the system's behavior around an operating point or steady-state. These values are essential for designing and implementing control systems based on the principle of linearization or incremental control. Many control systems operate around a specific steady-state condition. Linear control methods, such as PID or state-space controllers, require the system to be linearized around an operating point because real-world systems are often nonlinear. The variable *uss* represents the steady-state input required to keep the system at the desired steady-state output *yss*. By using *uss* and *yss*, the system is effectively linearized for small deviations, such as Δu and Δy, around the steady state. Furthermore, the controller works with deviations, such as Δu and Δy, from the steady-state values instead of absolute values. The equations

$$\Delta u = u - uss \quad \text{and} \quad \Delta y = y - yss$$

represent the control signal and system response in terms of changes around the steady state. This approach simplifies control design and analysis by focusing only on deviations.

Task 4

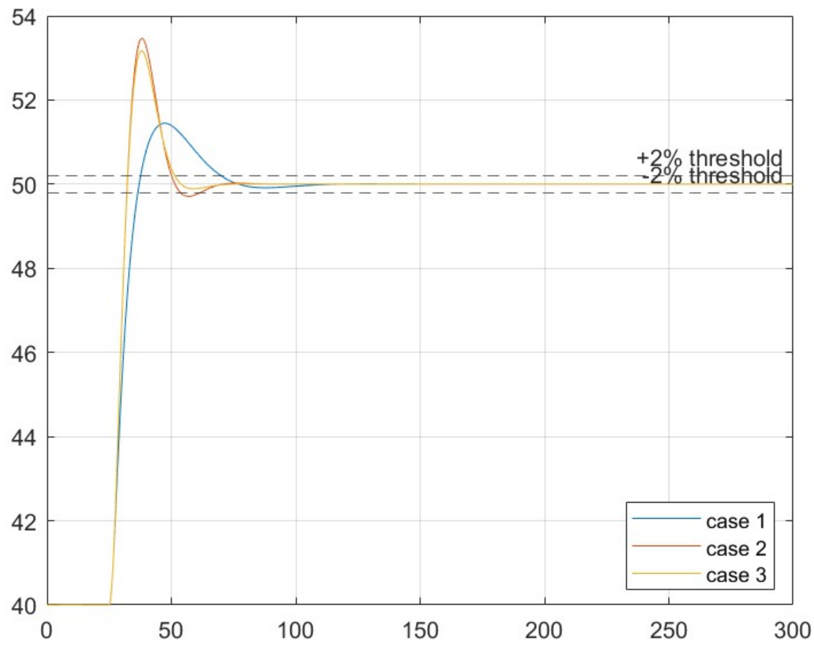
```
% Calculate PID parameters
[K, Ti, Td, N] = polePlacePID(chi, omega0, zeta,Tau,gamma_tank,
    k_tank);
G1 = tf([1], [1]);              % fist term
G2 = tf([1], [Ti, 0]);          % second term
G3 = tf([Td * N, 0], [1, N]); % third term

% Combine the components
F = K * (G1 + G2 + G3);
```

Task 5

χ	ζ	ω_0	T_r	M	T_{set}
0.5	0.7	0.1	7.998	15.698	43.005
0.5	0.7	0.2	4.895	34.459	33.620
0.5	0.8	0.2	4.864	32.667	22.869

According to the table, Case 3 is the best and satisfies all the requirements. These values were extracted using Bilevel measurements in Simulink. The parameter settings are as follows.

State Level Tol (%)	2.00
Upper Ref Level (%)	90.00
Mid Ref Level (%)	50.00
Lower Ref Level (%)	10.00
Settle Seek (s)	50.00



Task 6

The open-loop transfer function is solely determined by the combination of the controller F(s) and the plant G(s), along with any feedback components. In other terms the closed loop transfer function is as follows:

$$\frac{G(s)}{1 + G(s)H(s)} \tag{3}$$

then the open-loop transfer function will be:

$$G(s)H(s) \tag{4}$$

In our model, in the forward path, we have the controller $F(s)$ and the plant $G(s)$. Also, $H(s) = 1$ (When finding the open-loop crossover frequency, we are analyzing the forward path dynamics. This analysis is independent of the reference input. The reference signal is only used in the closed-loop transfer function for analyzing the system's response to input commands).

So the open-loop transfer function for this system is as follows:

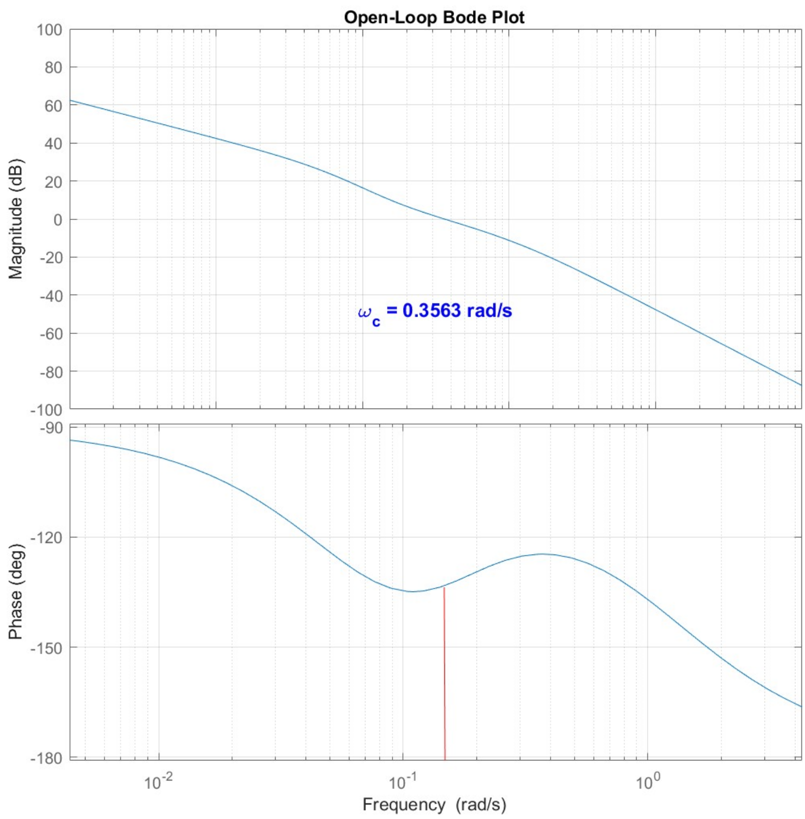
$$T_{\text{open}}(s) = G(s)F(s) = \frac{K_\gamma \left[T_I s(s+N) + (s+N) + T_D N s^2\right]}{T_I s(s+N)(1+\gamma T_I s)(1+\tau s)}. \tag{5}$$

To find the open-loop crossover frequency, we need to solve this equality equation:

$$|T_{\text{open}}(j\omega)| = 1 \tag{6}$$

where the crossover frequency is the frequency (ω_c) at which the magnitude of the open-loop transfer function is equal to 1 (0 dB).

In Bode plots, this point represents the frequency at which the magnitude is equal to one. Alternatively, we can use the MATLAB function: `[mag, phase, omega] = bode(T_open)`. And then find the corresponding frequency where the magnitude is equal to 1.

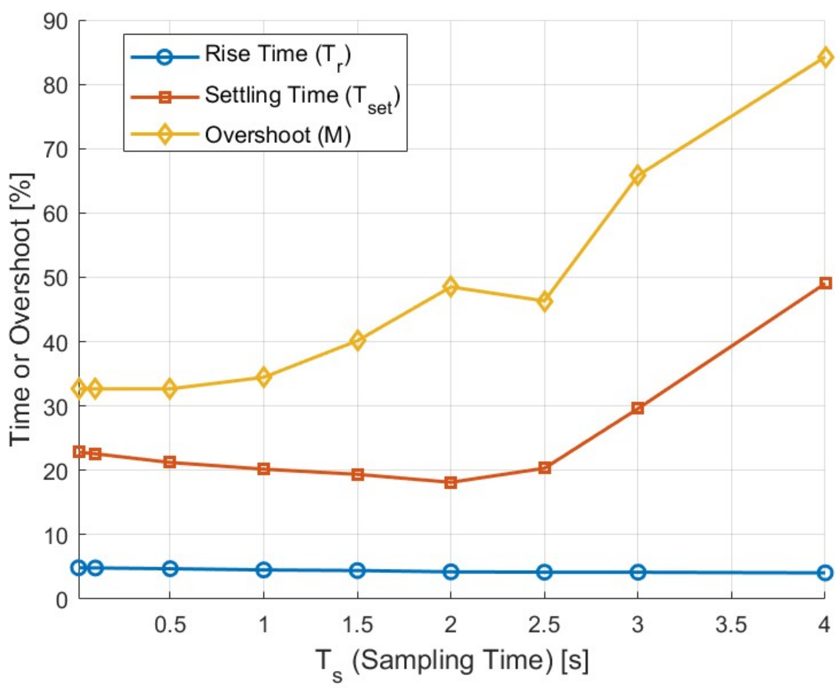


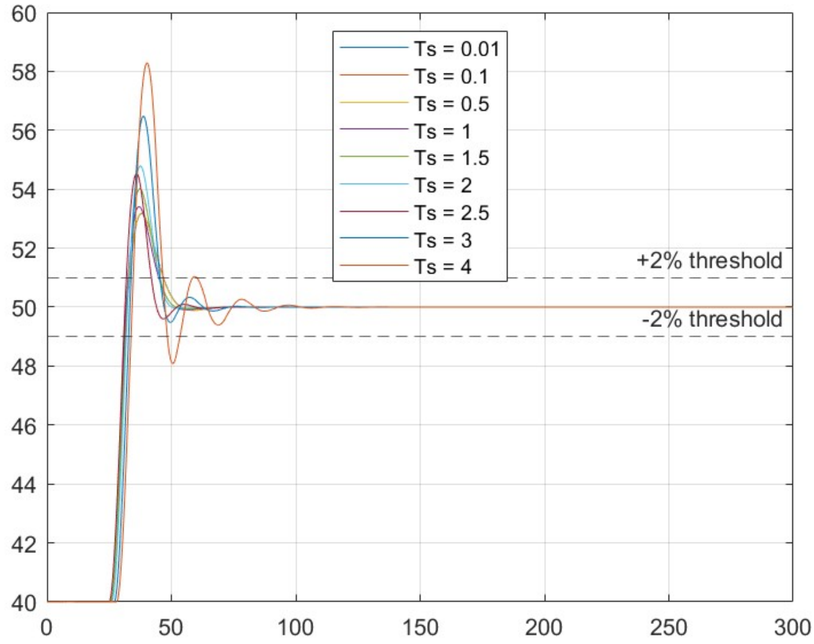
Task 7

We have varied the sampling time values from 0.01 seconds to 4 seconds, and the following results have been obtained:

$T_s(s)$	$T_r(s)$	$T_{\text{set}}(s)$	$M(\%)$
0.01	4.860	22.833	32.667
0.1	4.830	22.569	32.667
0.5	4.715	21.222	32.667
1	4.524	20.189	34.459
1.5	4.429	19.377	40.141
2	4.223	18.140	48.507
2.5	4.173	20.330	46.324
3	4.162	29.594	65.833
4	4.069	49.037	84.259

It can be seen that by decreasing the sampling time, both rise time and settling time increase, while overshoot decreases but does not go below a certain limit.
However, by increasing the sampling time, the rise time decreases, the settling time initially decreases and then increases, and the overshoot generally increases.
Therefore, it seems that a value of 1 is optimal (in terms of meeting the requirements of the question).
For values above 1 second, the requirements of the question are not met at all because the overshoot exceeds 35 %.

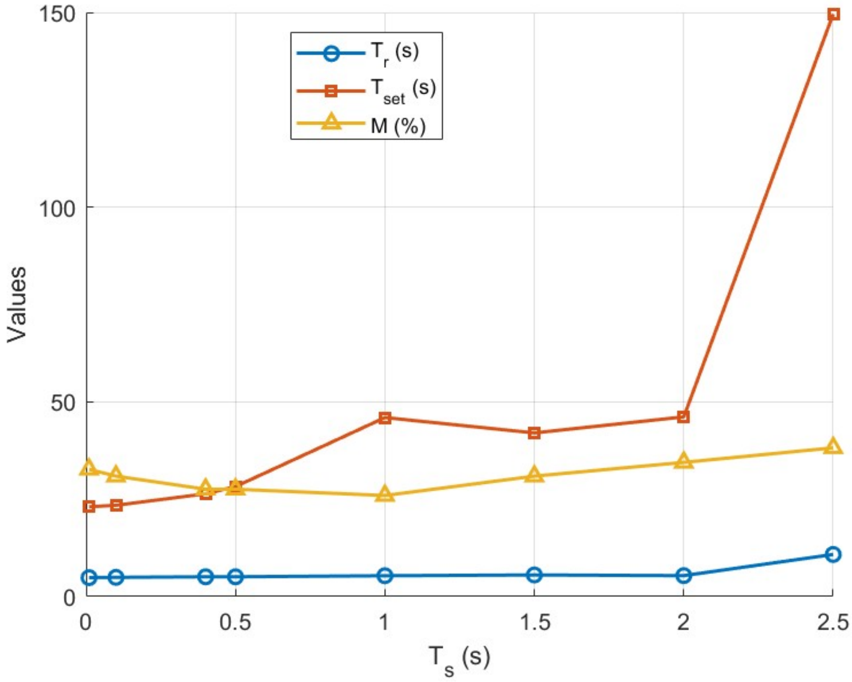




Task 8

We have varied the sampling time values from 0.01 seconds to 2.5 seconds, and the following results have been obtained. For $T_s = 3$ and above, the system becomes completely oscillatory and will not have a settling time.

$T_s(s)$	$T_r(s)$	$T_{\text{set}}(s)$	$M(\%)$
0.01	4.862	23.066	32.667
0.1	4.932	23.434	30.921
0.4	5.083	26.385	27.564
0.5	5.082	28.242	27.564
1	5.348	45.961	25.949
1.5	5.545	42.018	30.921
2	5.373	46.134	34.459
2.5	10.809	149.569	38.194



For a discretized controller, increasing the sampling time reduces resolution, increasing rise time and moving poles closer to the unit circle, which weakens damping. Beyond a certain point (e.g., 3 s), the system becomes fully oscillatory with no settling time.
Using a ZOH after a continuous-time controller introduces a delay, reducing the derivative term's effectiveness, increasing overshoot, and potentially leading to instability. In this case, the controller remains continuous, but when the controller is discretized, larger sampling times cause significant approximation errors, quickly destabilizing the system.
In other words, in the first case, where the ZOH is placed after the controller, increasing the sampling time causes the ZOH transfer function itself to approach instability, which is evident from its transfer function:

$$U(s) = \frac{1 - e^{-sT_s}}{s}$$

When T_s increases, the pole in the denominator becomes dominant, and the system becomes unstable.
However, in the case where the controller itself is discretized, the ZOH is effectively placed before the controller, and we are approximating the dynamics of the controller. Due to the approximation error, the dynamic behavior of the discretized controller will differ from the continuous-time case. The smaller the sampling time, the closer the discretized controller will approximate the continuous-time dynamics. However, as the sampling time increases, the approximation error grows, and the system behavior changes accordingly.

Task 9

According to Lecture 4, to give the system a phase margin decrease of 5 to 15 degrees we should have :

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

So with $\omega_c = 0.3563$ for the continuous-time plant (derived in Task 6) this results:

$$T_s \approx 0.1403 \text{ to } 0.393$$

The performance of these two ends of the range has been obtained as follows:

$T_s(s)$	$T_r(s)$	$T_{\text{set}}(s)$	$M(\%)$
0.1403	4.929	24.257	30.921
0.3930	5.004	27.916	29.221

It is obvious that the results are close the continuous case and satisfy the requirements.

Task 10

As we know, a larger phase margin generally indicates a more stable system, and a phase margin close to 0 degrees means that the system is close to oscillatory behavior.
From task 6, the phase margin of the continuous system is approximately 55 degrees ($PM = 180 - \angle T(j\omega_c)$) so with $T_s \approx 2.7\ s$ the system will be close to oscillatory behavior:

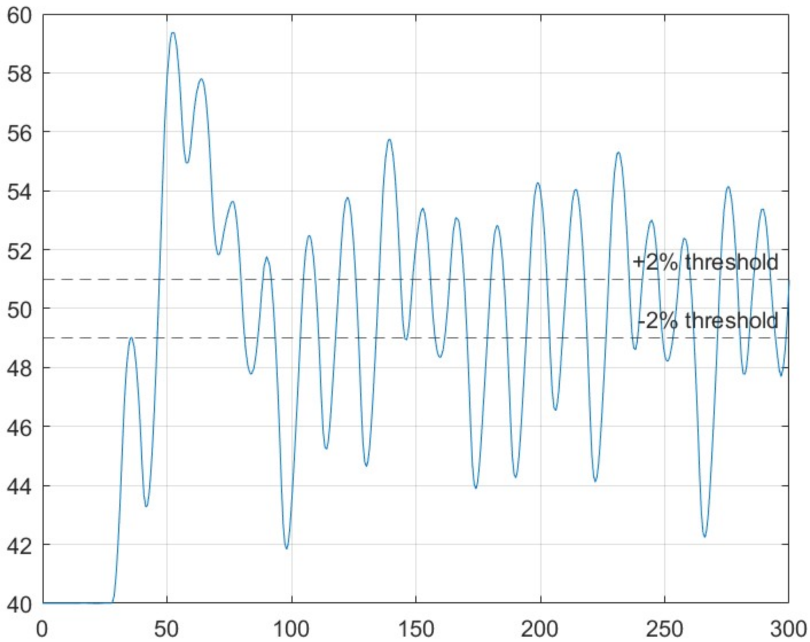
$$T_s\omega_c = 2.7 \times 0.3563 \approx 55^\circ$$

With $T_s = 0.5$ the decrease in phase margin is approximately 10 degrees, which is accepted. The difference from the values mentioned in Lecture 4 is likely due to the fact that, in that case, the reduction in phase margin was considered in a way to ensure the stability of the system under all conditions (even for systems with lower phase margins).

Task 11

Similar to the discussion of Task 10, the sampling time of 4 seconds results in a phase margin of approximately $55 - 80 = -25^\circ$ which apparently results to instability.

$T_s(s)$	$T_r(s)$	$T_{\text{set}}(s)$	$M(\%)$
4	16.739	$--$	77.679



Task 12

Using Equation (1) from the instruction file and applying the inverse Laplace transform:

$$\Delta \dot{x}_1(t) = -\frac{1}{\tau}\Delta x_1(t) + \frac{k}{\tau}\Delta u(t) \tag{7}$$

$$\Delta \dot{x}_2(t) = \frac{1}{\tau}\Delta x_1(t) - \frac{1}{\gamma\tau}\Delta x_2(t) \tag{8}$$

$$\Delta y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_1(t) \\ \Delta x_2(t) \end{bmatrix} \tag{9}$$

From these equations, we can write the state-space matrices and equations as follows:

$$A = \begin{bmatrix} -\frac{1}{\tau} & 0 \\ \frac{1}{\tau} & -\frac{1}{\gamma\tau} \end{bmatrix}, \quad B = \begin{bmatrix} \frac{k}{\tau} \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 1 \end{bmatrix} \tag{10}$$

$$\begin{aligned} \Delta \dot{x}(t) &= \begin{bmatrix} -\frac{1}{\tau} & 0 \\ \frac{1}{\tau} & -\frac{1}{\gamma\tau} \end{bmatrix} \Delta x(t) + \begin{bmatrix} \frac{k}{\tau} \\ 0 \end{bmatrix} \Delta u(t), \\ \Delta y(t) &= \begin{bmatrix} 0 & 1 \end{bmatrix} \Delta x(t). \end{aligned} \tag{11}$$

After the discretization of the system, the state-space equations are as follows:

$$\begin{aligned} \Delta x(k+1) &= \Phi \Delta x(k) + \Gamma \Delta u(k), \\ \Delta y(k) &= C \Delta x(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} \Delta x(k). \end{aligned} \tag{12}$$

Where:

$$\Phi = \begin{bmatrix} 0.7249 & 0 \\ 0.2332 & 0.7249 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} 0.6017 \\ 0.0916 \end{bmatrix}. \tag{13}$$

Task 13

Reachability condition: The system is reachable, if and only if, the controllability matrix (W_c) is invertible.

$$W_c = \begin{bmatrix}\Gamma & \Phi\Gamma & \cdots & \Phi^{n-1}\Gamma\end{bmatrix} \tag{14}$$

Observability condition: The system is observable, if and only if, the observability matrix is convertible.

$$W_o = \begin{bmatrix} C \\ C\Phi \\ \vdots \\ C\Phi^{n-1} \end{bmatrix} \tag{15}$$

Using Matlab functions `ctrb(Phi, Gamma)`, `obsv(Phi, C)`, we can compute controllability and observability matrices respectively. To see if these matrices are invertible we can use either `det()` function to check if the determinant of the matrix is non-zero or `rank()` function to see if the matrix is full rank.

$$\begin{aligned} det(W_c) = 0.0844, det(W_o) = -0.2332, \\ rank(W_c) = 2, rank(W_c) = 2 \end{aligned} \tag{16}$$

As a result, the system is both observable and reachable.

Task 14

If the control input u(k) includes a term based on the reference signal, this allows the control input to effectively guide the system to follow the desired reference signal.

$$\Delta u(k) = -L\Delta\hat{x}(k) + l_r r(k) \tag{17}$$

In this equation, l_r is the feedforward gain for the reference signal. By using a dynamic observer to estimate the $\hat{x}$, the relationship between output and reference signal will be as follows:

$$y(k) = C\big(I - \Phi + \Gamma L\big)^{-1}\Gamma\ell_r r(k), (q = 1) \tag{18}$$

(In the z-transform, q is equivalent to the z-variable, which represents the system dynamics in the discrete domain. For steady-state analysis, we examine the system at $q = 1$, which is corresponds to DC gain in other terms at steady state $x(t + 1) = x(t)$)
Furthermore, the equations (19) and (20) show that the process dynamics (A(q), B(q)) and controller dynamics (R(q),S(q)) may cause the output $y(k)$ to have a magnitude mismatch (steady state error) with the reference r(k). the reference gain l_r adjusts the scaling so that $y(k) = r(k)$.

$$y(k) = \frac{B(q)}{A(q)}u(k), \quad u(k) = \frac{S(q)}{R(q)}\big(\ell_r r(k) - y(k)\big) \tag{19}$$

$$\big(A(q)R(q) + B(q)S(q)\big)y(k) = B(q)S(q)\ell_r r(k) \tag{20}$$

Based on equation (18) it can be seen that if the value of l_r to be chosen as follows, then the output will follow the reference exactly and without steady-state error.

$$l_r = \frac{1}{C\big(I - \Phi + \Gamma L\big)^{-1}\Gamma}. \tag{21}$$

Task 15

State-space dynamics in terms of differences ($\Delta x(k)$):

$$\Delta x(k+1) = x(k+1) - x(k) = \Phi \Delta x(k) + \Gamma \Delta u(k) \tag{22}$$

Observer dynamics in terms of differences ($\Delta \hat{x}$):

$$\Delta \hat{x}(k+1) = \Phi \Delta \hat{x}(k) + \Gamma \Delta u(k) + K\big(\Delta y(k) - C\Delta \hat{x}(k)\big) \tag{23}$$

(Substitute $y(k) = Cx(k)$):

$$\Delta \hat{x}(k+1) = \Phi \Delta \hat{x}(k) + \Gamma \Delta u(k) + K\big(C\Delta x(k) - C\Delta \hat{x}(k)\big) \tag{24}$$

Substitute control law (19):

$$\Delta x(k+1) = \Phi \Delta x(k) + \Gamma\big(- L\Delta \hat{x}(k) + \ell_r r(k)\big) \tag{25}$$

$$\Delta \hat{x}(k+1) = \big(\Phi - KC\big)\Delta \hat{x}(k) + \Gamma\big(- L\Delta \hat{x}(k) + \ell_r r(k)\big) + KC\Delta x(k) \tag{26}$$

State-space dynamics of the augmented system ($x_a(k)$):

$$x_a(k+1) = \begin{bmatrix} \Phi & -\Gamma L \\ KC & \Phi - \Gamma L - KC \end{bmatrix} x_a(k) + \begin{bmatrix} \Gamma \ell_r \\ \Gamma \ell_r \end{bmatrix} r(k) \tag{27}$$

Where:

$$A_a = \begin{bmatrix} \Phi & -\Gamma L \\ KC & \Phi - \Gamma L - KC \end{bmatrix}, \quad B_a = \begin{bmatrix} \Gamma \ell_r \\ \Gamma \ell_r \end{bmatrix} \tag{28}$$

Task 16

First, we need to verify the system can be written as the form in the task. From first and second rows of the state space system equation (27), we can get the:

$$\Delta x(k+1) = \Phi \Delta x(k) - \Gamma L \Delta \hat{x}(k) + \Gamma \ell_r r(k), \tag{29}$$

$$\Delta \hat{x}(k+1) = KC\Delta x(k) + (\Phi - \Gamma L - KC)\Delta \hat{x}(k) + \Gamma \ell_r r(k). \tag{30}$$

By substituting $\Delta \hat{x}(k+1) = \Delta x(k+1) - \Delta \tilde{x}(k+1)$ in to these two equations and rewriting them:

$$\Delta x(k+1) = (\Phi - \Gamma L)\Delta x(k) + \Gamma L \Delta \tilde{x}(k) + \Gamma \ell_r r(k), \tag{31}$$

$$\Delta \tilde{x}(k+1) = (\Phi - KC)\Delta \tilde{x}(k) + \Gamma \ell_r r(k). \tag{32}$$

In the matrix form the equations can be writen as:

$$\begin{bmatrix} \Delta x(k+1)^T \\ \Delta \tilde{x}(k+1)^T \end{bmatrix} = \begin{bmatrix} \Phi - \Gamma L & \Gamma L \\ 0 & \Phi - KC \end{bmatrix} \begin{bmatrix} \Delta x(k)^T \\ \Delta \tilde{x}(k)^T \end{bmatrix} + \begin{bmatrix} \Gamma l_r \\ 0 \end{bmatrix} r(k) \tag{33}$$

The separation principal in control theory states that the design of the state feedback controller and the observer can be done independently, without interfering with each other. By showing that the controller dynamics and the observer dynamics in the augmented matrix are decoupled, we can prove that for this system separation holds. The augmented matrix has a block triangular structure:

$$A_{\text{aug}} = \begin{bmatrix} \Phi - \Gamma L & \Gamma L \\ 0 & \Phi - KC \end{bmatrix}. \tag{34}$$

For block triangular matrices, the eigenvalues of the matrix is the union of the eigenvalues of the diagonal blocks. This means the eigenvalues of A_{aug} are:

$$\text{Eigenvalues of } A_{\text{aug}} = \text{Eigenvalues of } (\Phi - \Gamma L) \cup \text{Eigenvalues of } (\Phi - KC)$$

This result shows that the eigenvalues of $(\Phi - \Gamma L)$ depend only on feedback gain (L) and the eigenvalues of $(\Phi - KC)$ depend only on observer gain (K). so by choosing eigenvalues (poles) for an specific design of the observer or state feedback controller the design of the other one does not change.

Task 17

The closed-loop transfer function of the system is as follow:

$$Cloop = \frac{G \cdot F}{1 + G \cdot F} \tag{35}$$

Using Matlab function `pole()` we can calculate the poles of the system (before this we use `minreal()` to ensure minimal realization).

Continues Pole	**Value**
1	$-0.5000 + 0.0000i$
2	$-0.5000 - 0.0000i$
3	$-0.1600 + 0.1200i$
4	$-0.1600 - 0.1200i$

Then discrete poles of a continues time system using the relation $z_i = e^{T_s p_i}$ will be identified.

Discrete Pole	**Value**
1	$0.1353 + 0.0000i$
2	$0.1353 - 0.0000i$
3	$0.4677 + 0.2435i$
4	$0.4677 - 0.2435i$

Now the goal is to design a state-feedback controller such that the closed-loop system has the same poles as the continuous-time closed-loop system. So the desired poles are the continues ones, and we want to design the controller gain L and the observer gain K such that the poles of the discrete system move to the poles of the continues system.
To do this, first we should choose the desired poles for design of the observer and controller. Poles for the state feedback (eigenvalues of $(\Phi - \Gamma L)$) determines the stability and performance of the overall system, while poles for the observer (eigenvalues of $(\Phi - KC)$) determine how quickly the estimation error converges to zero.
Based on this, We choose the desired poles for the observer as poles closer to the origin than the state feedback poles to make the observer faster than the system. A fast observer ensures accurate state estimation.

- Desired poles for Observer: The pair $-0.1600 \pm 0.1200i$.
- Desired poles for state feedback: $-0.5000, -0.5000$.

Now using `acker()` function and by matrices Φ, Γ, and C we can calculate the optimal value of K and L.

$$L = \begin{bmatrix} 2.4435 & 10.6906 \end{bmatrix}, \quad K = \begin{bmatrix} 3.4189 \\ 1.7697 \end{bmatrix}.$$

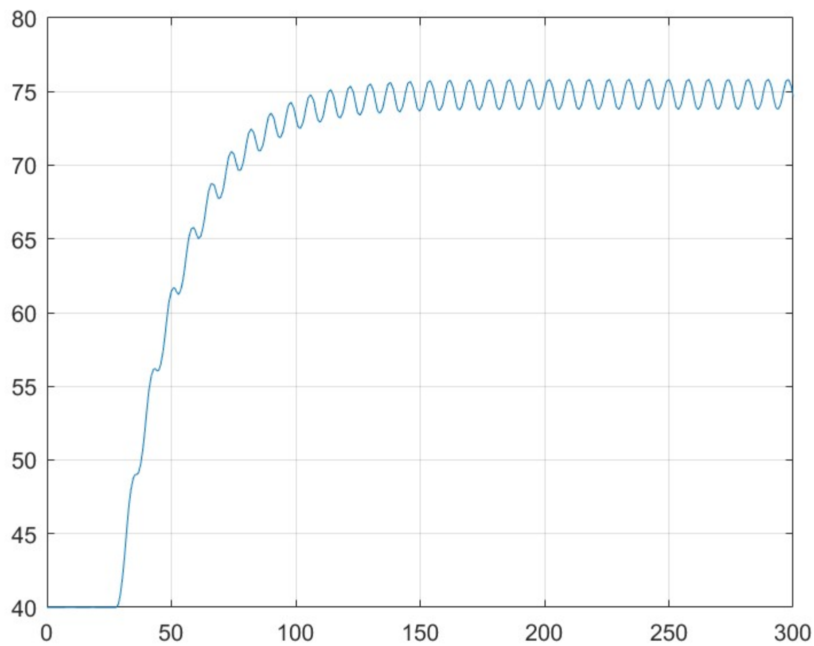
To check the results we can calculate the eigenvalues of augmented matrix A_a.

$$\text{Eigenvalues of } A_a = \begin{bmatrix} -0.1600 + 0.1200i \\ -0.1600 - 0.1200i \\ -0.5000 + 0.0000i \\ -0.5000 - 0.0000i \end{bmatrix}.$$

The eigenvalues are equal to our desired poles.

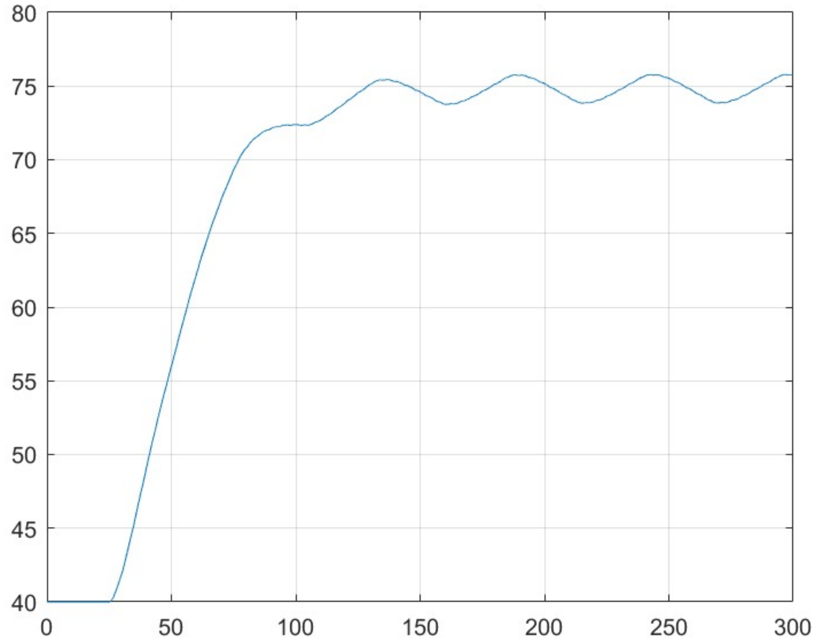
Task 18

Using discrete designed controller, the output signal is as follows:

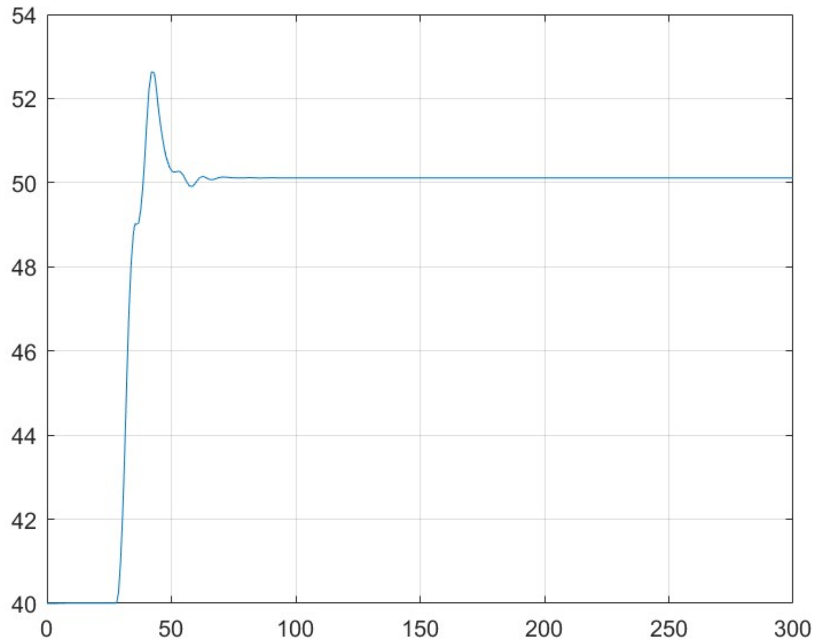


With respect to this results we can see two phenomena:

- **The Oscillation**: We want to compensate this oscillation, by using a smaller sampling time Ts to improve the discrete-time approximation of the continuous system. For example for $Ts = 0.1$ (within the range of tast 9) the result would be:



- **Large Steady-State Error**: This problem is mainly due to The reference gain. l_r is used to scale the control input $u(k)$ so that the system output $y(k)$ tracks the reference $r(k)$ correctly in steady state. When the l_r is too large, the output overshoots and stabilizes at a higher value. Based on our design strategy we put our desired poles for the sate feedback at $[-0.5000, -0.5000]$. But if we would change their place and get them closer to the origin we could achieve a smaller l_r. For example for $Ts = 4$ and $P_L = [-0.1000, -0.1000]$:



which has a very small steady-state error.

Task 19

To calculate the quantization level, we use the formula:

$$\text{Quantization Level} = \frac{\text{Range of Signal}}{\text{Number of Quantization Steps}}$$

Where the Number of Quantization Steps is given by 2^N, where N is the number of bits. So we have:

$$\text{Quantization Level} = \frac{100}{1024} \approx 0.0977$$

Task 20

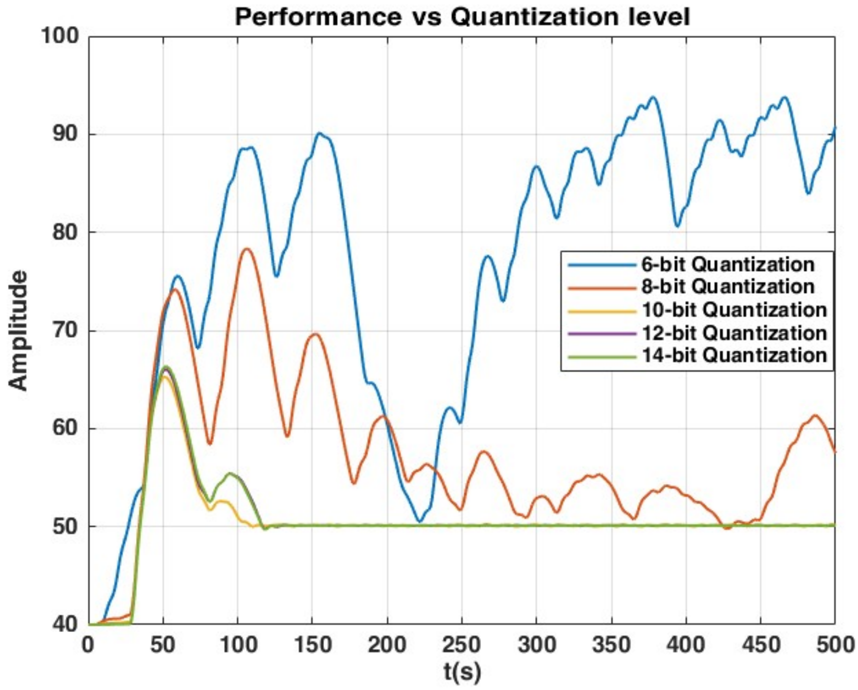
The usage of saturation block can be regarded as:

$$output = \begin{cases} 0 & input \leq 0 \\ input & 0 \leq input \leq 100 \\ 100 & input \geq 100 \end{cases} \tag{36}$$

The reason why we need to add a saturation block is that in reality we have upper and lower limit for the signal, for example the output signal(water height of the second tank),in reality the maximum height of the water tank has a certain limit, in this case we set a saturation of 100.
Both A/D and D/A converters operate within a specific range, if the input signal exceeds this range, the converter cannot process or represent those values, leading to clipping. The Saturation block simulates this behavior by constraining the signal to the valid range, mimicking the physical limitations of real-world converters. We model the ADC as a cascade of ZOH, Quantizer and a saturation block. Similarly the DAC as a cascade of Quantizer, Saturation block and ZOH.

Task 21

The system was simulated by placing the ADC and DAC models in the configuration mentioned in the previous tasks (18,21).
The sampling time $T_s = 4$ seconds was set, and the number of quantization bits varied from 6 to 14.



Simulation results show that when the quantization bit count decreases to 10 or lower, the system's performance significantly degrades and becomes unstable. Additionally, it appears that 12 bits (corresponding to Quantization level of 0.0244) is the optimal choice. Using fewer bits disrupts the system's functionality, while exceeding 12 bits would result in overdesign.

Fewer quantization bits degrade the accuracy and stability of the digital control system. To balance performance and hardware limitations, the number of bits should be chosen such that the quantization error is small enough to not impact system dynamics significantly, Otherwise larger quantization errors from fewer bits can lead to increased overshoot, prolonged settling time and loss of damping, making the system oscillatory or unstable.