GROUP 18

Aman Saini (2020EEB1155) Ansaf Ahmad (2020EEB1160)
Anirudh Sharma (2020EEB1158)

DYNAMIC RESPONSE OF TRANSFER FUNCTION ON MATLAB PLATFORM

OBJECTIVE

Analysis of dynamic response of a given linear analog system in terms of different performance measures.

TRANSFER FUNCTION OF THE SYSTEM IN CASCADE AND FEEDBACK CONFIGURATIONS.

The given 2nd order Open Loop Transfer function is:

$$G_{OL} = \frac{k}{s^2+3s+10}$$

and the transfer function of PD controller is given as:

$$G_C = 80(s+5)$$

In both the configurations we obtain identical denominator polynomials in resultant transfer function implying similar Root Loci. The difference in step response however, arises from difference in numerator polynomials (which leads to different zeroes for both the transfer functions).

Cascade:

$$Csc = \frac{80k(s+5)}{s^2+(80k+3)s+400k+10}$$

And,

Feedback:

$$FB = \frac{k}{s^2+(80k+3)s+400k+10}$$

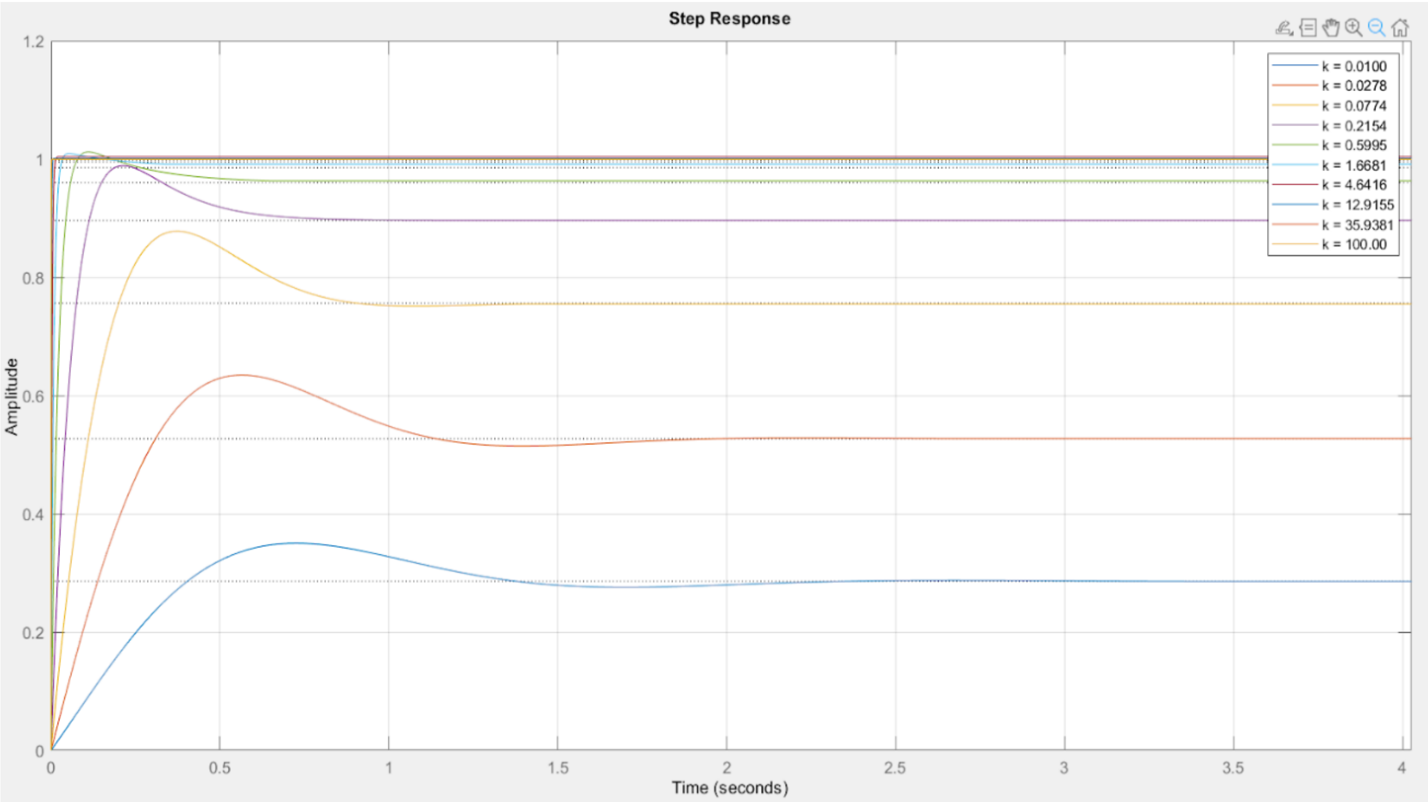
We observe that adding a **LHP zero** to the Transfer function (**x = -5** in this case) adds a scaled derivative term to the step response, which increases the overshoot, decreases the rise time, the settling time is not affected much, in other words, it makes the step response faster.

STEP RESPONSE OF THE SYSTEM (IN FEEDBACK AND CASCADED CONFIGURATION AS 'K' IS VARIED)

The following plots represent the step response of the system in the two configurations as the OLTF gain 'k' is varied on a **logarithmic scale** from **(0.01 to 100).**

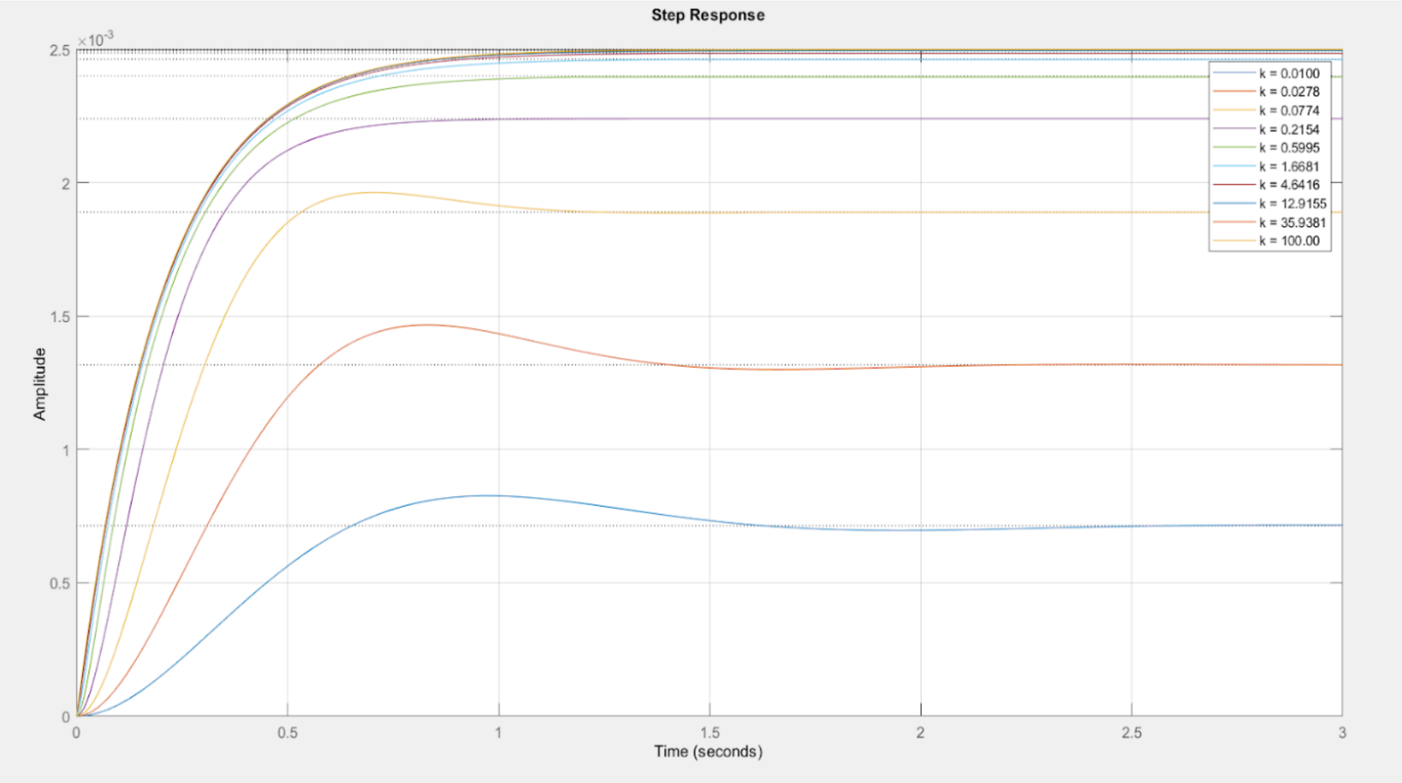
1. CASCADED CONFIGURATION

The presence of zero typically speeds up the response of the system as is evident from the above plot the step response of cascaded configuration progressively approaches the unit step signal as $k \rightarrow \infty$.



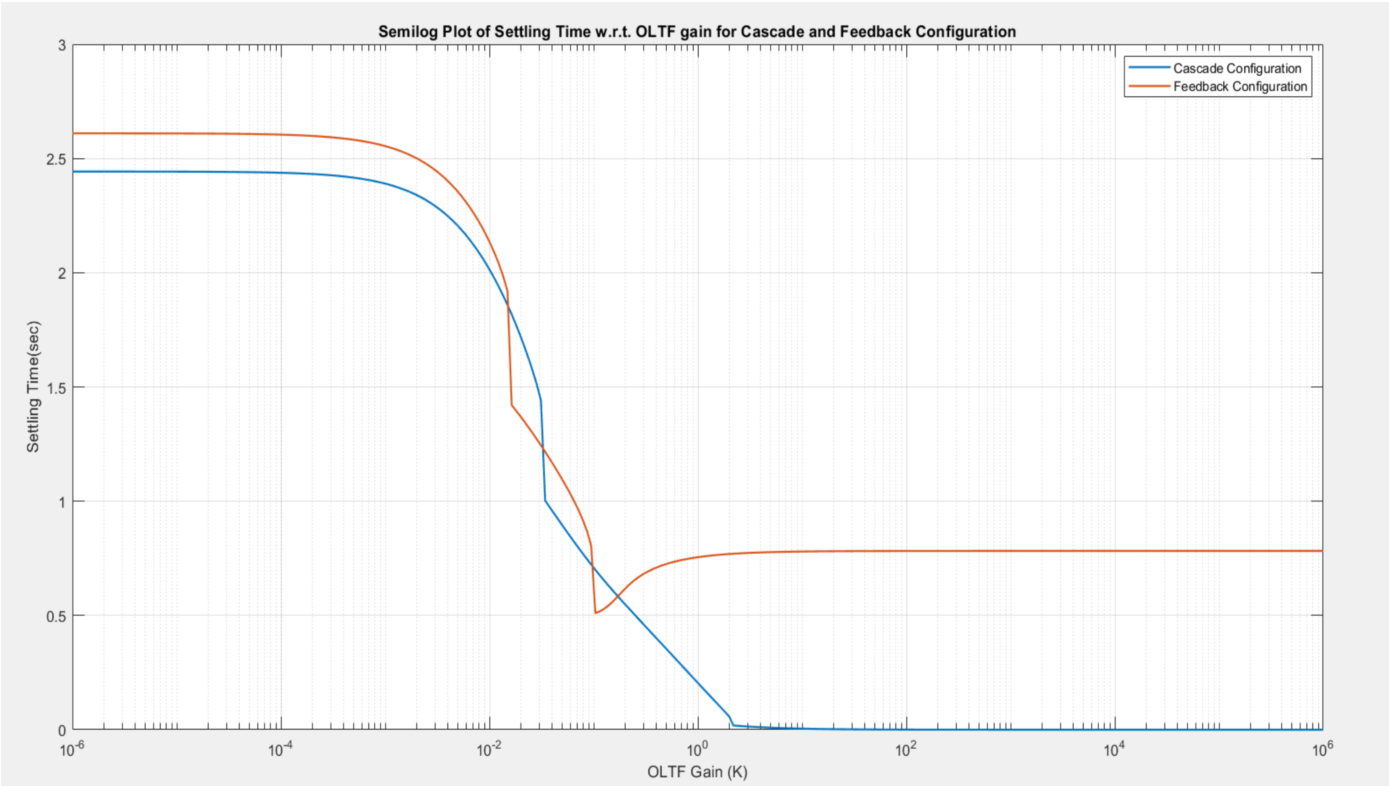
2. FEEDBACK CONFIGURATION

In this configuration, with increase in k oscillations diminish altogether similar to the cascaded controller (since both would have same root loci). And it can be interfered from the response that as 'K' tends to really large values response becomes similar to a first order system.



PERFORMANCE OF THE SYSTEM IN CASCADE AND FEEDBACK CONFIGURATIONS

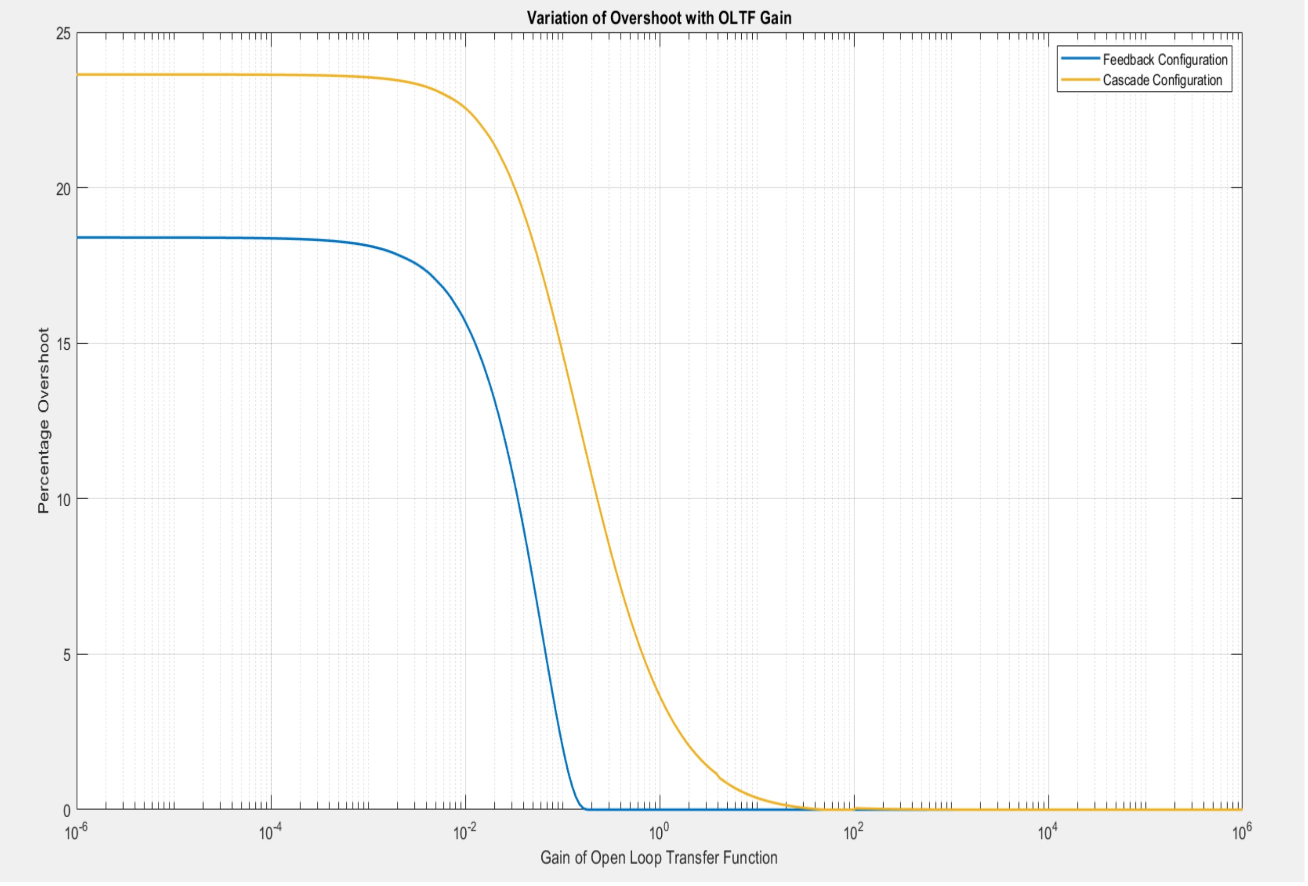
1. SETTLING TIME



Settling time is the time required for an output to reach the steady state value. From the graph above we can observe that the settling time decreases for both the configuration. For feedback configuration, the settling time decreases and eventually settles at a constant value. In cascade configuration, the settling time decreases to very small values at larger k.

2. PERCENTAGE OVERSHOOT

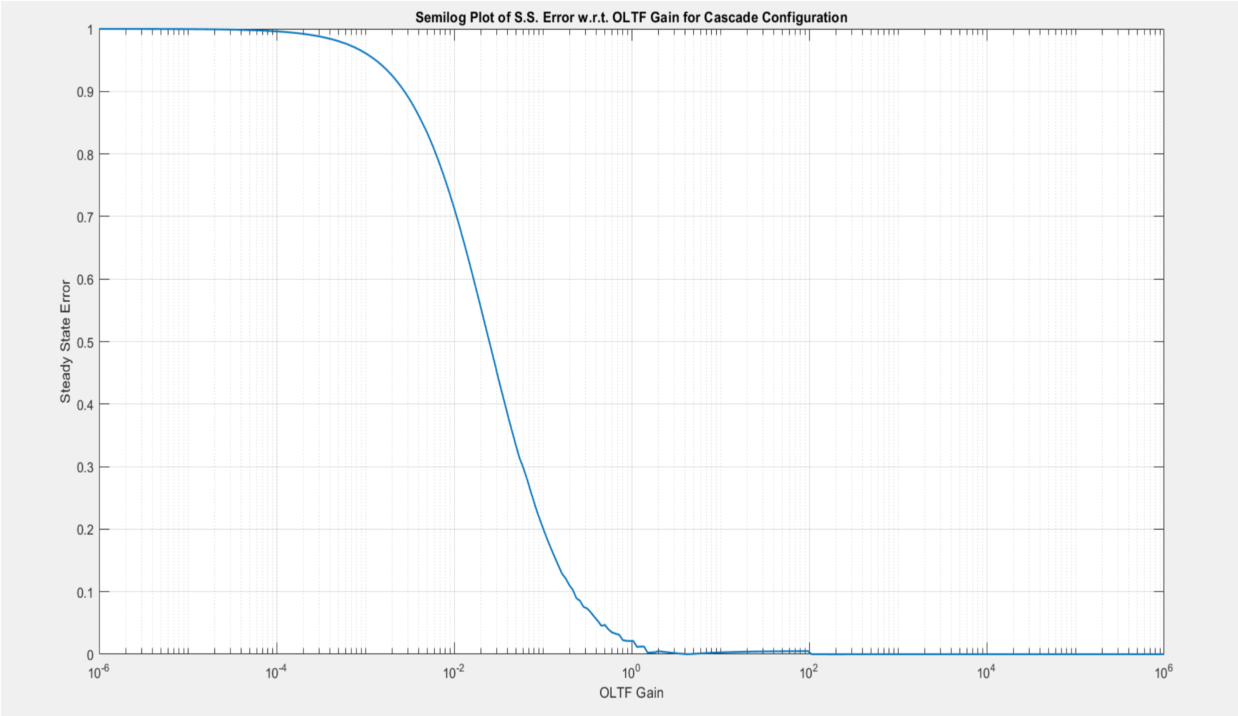
Overshoot is the occurrence of a signal or function exceeding its target. Overshoot decreases for both the configurations which is due to the decrease in the imaginary part of the pole. But at any value of k, the feedback system overshoot is less than cascade one.



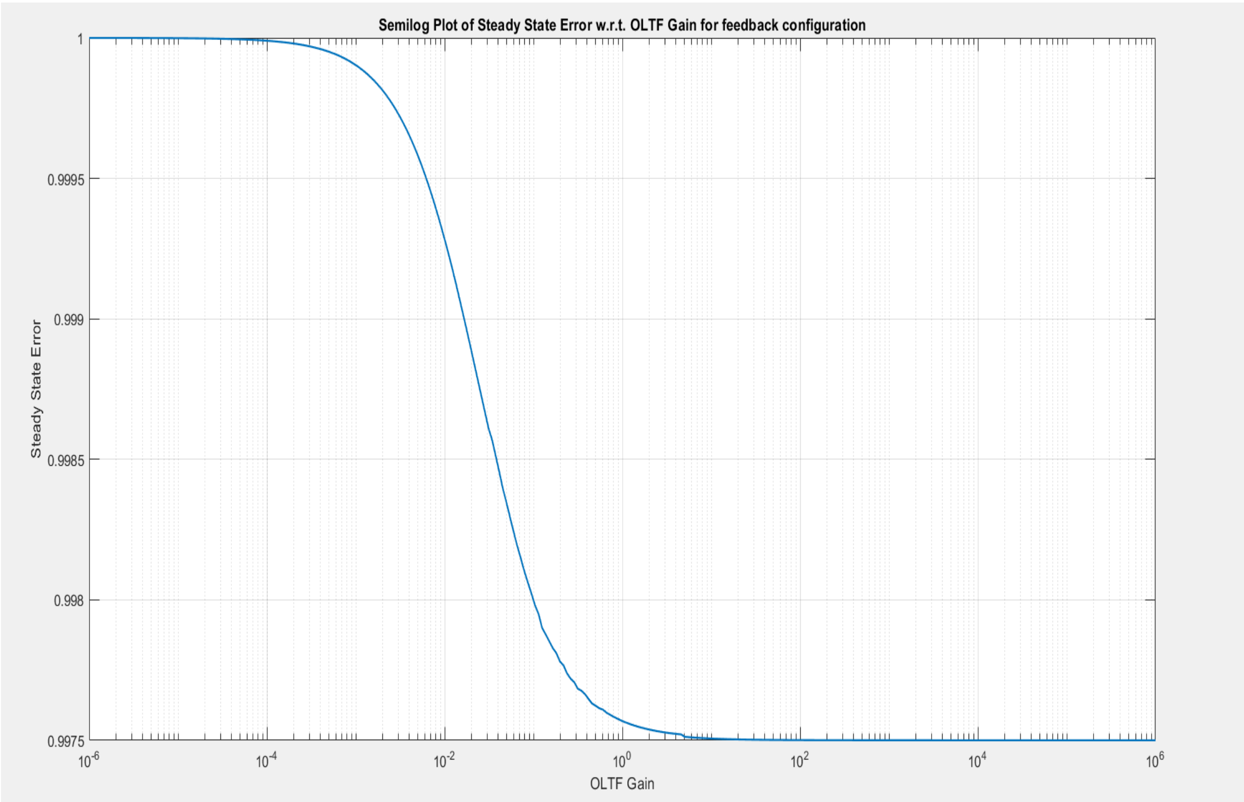
3. STEADY STATE ERROR

Steady-state error is defined as the difference between the input (command) and the output of a system in the limit as time goes to infinity (i.e., when the response has reached steady state).

As evident from the below plots, steady state error for Cascade system eventually decreases to zero with increase in gain. But in case of feedback configuration, error never reaches zero.

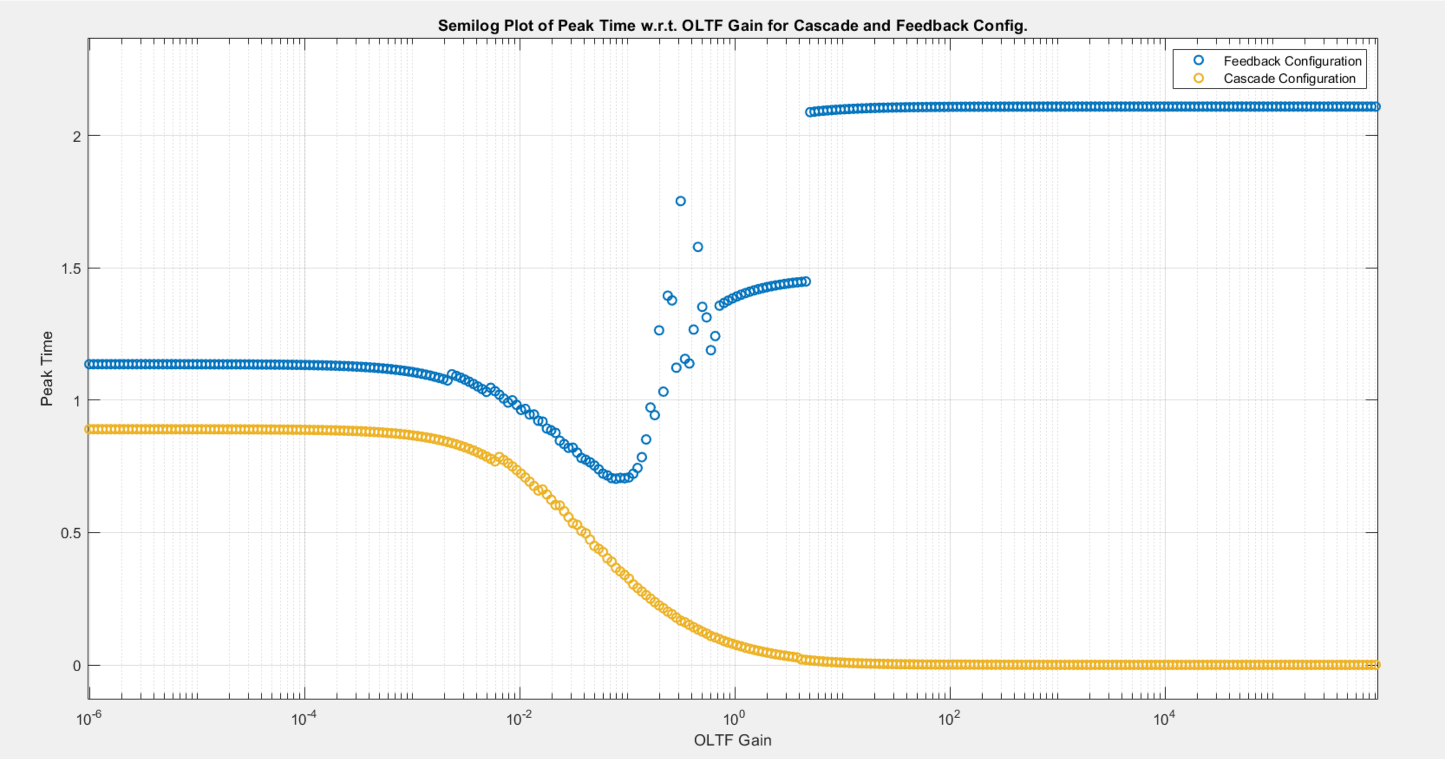


Steady State Error for Cascade configuration



Steady State Error for Feedback configuration

4. PEAK TIME



From the above graph we see that as gain 'K' increases the Peak Time for-

a) **Feedback Configuration:**
Progressively decreases as 'K' increases from 10^{-6} to 10^{-1}. After this value is surpassed, we see a random distribution in the graph till K~0.8. After this we observe an increase in settling time and a discontinuity as K goes from a value of 4 to 5. Eventually the increase in settling time saturates around ~2.1s.

b) **Cascade Configuration:**
As compared to the Feedback configuration, we see a uniform decrease in Settling time in Cascade configuration as 'K' increases. There are no discontinuities in the graph making it a more predictable and stable system.

We can compare our findings on a table for K=10 for both the configurations.

Configuration	Settling time(s)	Overshoot	Steady state error	Peak time(s)
Feedback	0.7798	0	0.9975	2.0986
Cascade	0.0046	0.3809	0.0029	0.0086

Table 1.

CONCLUSIONS

The transfer function of Cascade and Feedback Configuration essentially differs in the gain and zeroes. The cascaded transfer function is given as:

$$Csc(s) = \frac{80k\,(s+5)}{(s^2 + (80k+3)s + 400k + 10)}$$

And, the feedback transfer function is given as:

$$Fb(s) = \frac{k}{(s^2 + (80k+3)s + 400k + 10)}$$

The difference is that the Csc (Cascaded transfer function) has an extra zero at **s = -5** and its **gain is 400 times that of Feedback configuration**.

$$Csc(s) = 80(s+5)Fb(s) = 400\left(Fb(s) \; + \frac{s}{5}Fb(s)\right)$$

If we consider $Y_{feedback}(s)$ and $Y_{cascade}(s)$ to be the Laplace Transform of response of Feedback and Cascade configuration to step input respectively, then,

$$\begin{aligned} Y_{cascade}(s) &= \frac{1}{s} * 400\left(Fb(s) \; + \frac{s}{5}Fb(s)\right) \\ &= 400 * \left(\frac{Fb(s)}{s} + \frac{s}{5}\, Fb(s)\frac{1}{s}\right) \\ &= 400 * \left(Y_{feedback}(s) \; + \frac{s}{5}Y_{feedback}(s)\right) \end{aligned}$$

Taking inverse Laplace transforms we get,

$$y_{cascade}(t) = 400 * \left(y_{feedback}(t) \; + \frac{1}{5}\left(\dot{y}_{feedback}(t)\right)\right)$$

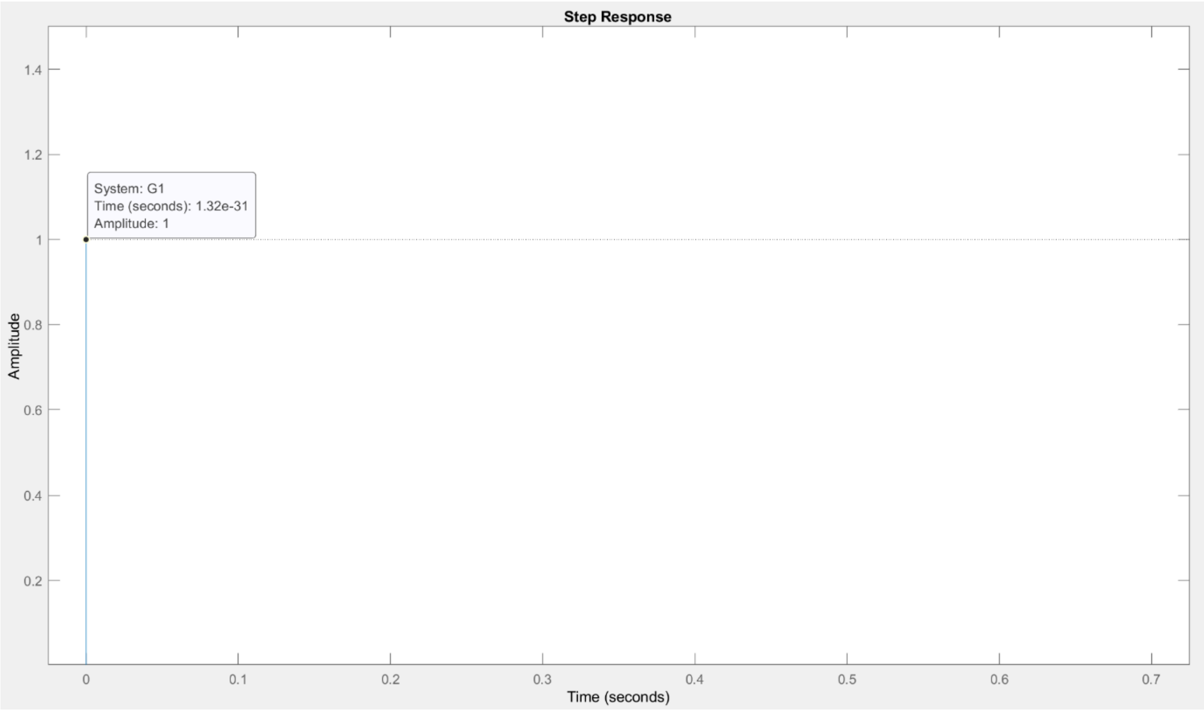
Thus, we can conclude that the difference in the step response arises because of the **scaled derivative term** introduced in time domain response (**which speeds up the response**) and the fact that the whole expression is scaled by **400**, which accounts for the difference is **Steady State Value** between both configurations.

It is also observable from the graph below that as OLTF Gain (k) →∞, the response of **cascaded configuration approaches step signal**. This is because,

$$\lim_{k\to\infty} \text{Csc} = \lim_{k\to\infty} \frac{80k(s+5)}{s^2+(80k+3)s+400k+10} = 1$$

Assuming step input,

$$\mathcal{L}_s^{-1}[1/s](t) = u(t)$$



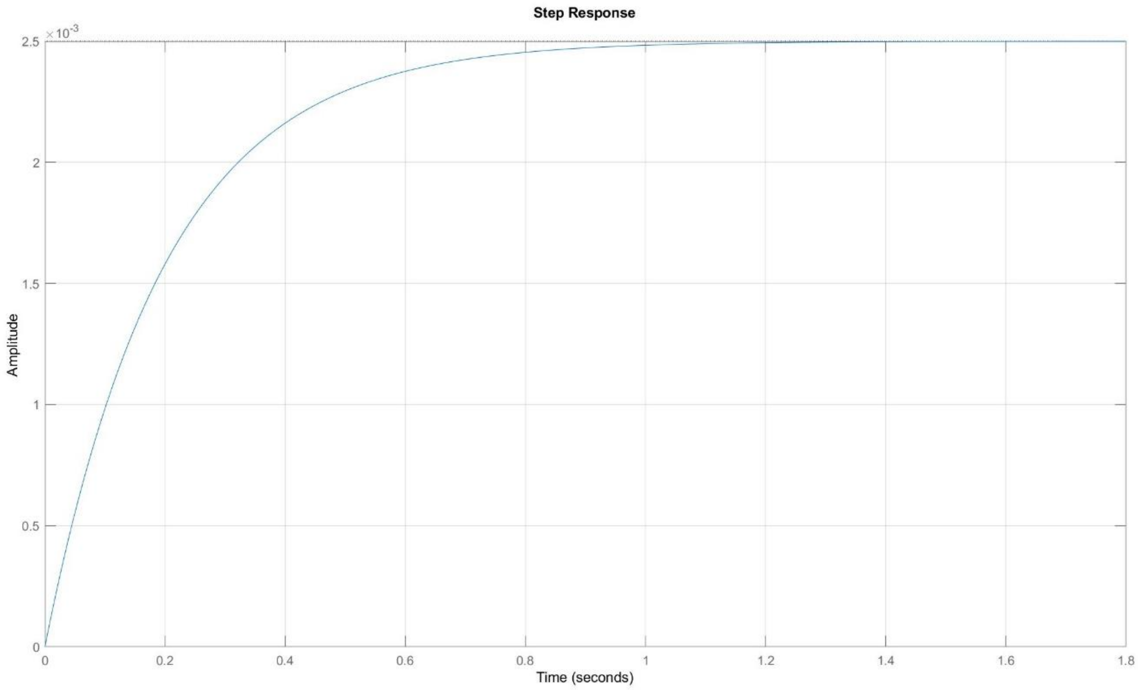
Step response for Cascade configuration

And the response of **feedback configuration approaches** $\boldsymbol{y = \frac{1}{400} - \frac{e^{-5t}}{400}}$. This is because,

$$\lim_{k\to\infty} \text{FB} = \lim_{k\to\infty} \frac{k}{s^2+(80k+3)s+400k+10} = \frac{1}{80s+400}$$

Assuming step input,

$$\mathcal{L}_s^{-1}\left[\frac{1}{s(80s+400)}\right](t) = \frac{1}{400} - \frac{e^{-5t}}{400}$$



Step response for feedback configuration

Hence, we conclude that **Feedback Configuration** is **less desired** as **Cascaded Configuration** provides **faster Settling Time**, **less Peak Time** and **negligible Steady State Error**. Also, the variation of above-mentioned quantities is **very minute w.r.t change in OLTF Gain 'K' in Cascade Configuration**. (For e.g., the variation in Peak Time vs OLTF Gain 'K' is very unpredictable for Feedback Configuration in the range $K=10^{-1}$ to 10^{0} and has a discontinuity at the point where K tends to 5.)

MATLAB CODE USED

```
s = tf('s');
G = tf(1, [1,3,10]); % System transfer function (Assuming K = 1)
G %printing G

C = 80*(s+5); % controller Transfer function
C %printing C (controller tf)

FB = feedback(G, C); %feedback connection
FB %printing FB
step(FB);
%print("Step Characteristics for feedback connection")
stepinfo(FB)

Csc = feedback(G*C, 1); %Cascaded connection with unit negative feedback
Csc %printing Csc
step(Csc);
%print("Step Characteristics for Cascade connection")
stepinfo(Csc)
%Characteristic function is same in both the cases
% i.e., feedback connection and cascaded connection with unit negative feedback

k = [1: 1: 10];
G_array = k*G;
B_arr = G_array; % initializing array equal to G_array
% we will update FB_array later to rep cascade connection with C with unity
% neg feedback, for different values of OLTF gain

for i = 1:10
    B_arr(i) = feedback(G_array(i)*C, 1); %cascade
end
for i = 1:10
    stepinfo(B_arr(i)) %gives characteristics of step response
    step(B_arr(i)); % for plotting the step response
    hold on % prints step response on the same graph
end
legend({'k = 1', 'k = 2', 'k = 3', 'k = 4', 'k = 5','k = 6', 'k = 7', 'k = 8', 'k
= 9', 'k = 10'});
hold off;

k = [1: 1: 10];
G_array = k*G;
A_arr = G_array;
for i = 1:10
    A_arr(i) = feedback(G_array(i), C); %feedback connection
end
for i = 1:10
    stepinfo(A_arr(i))
    step(A_arr(i));
    hold on
end
legend({'k = 1', 'k = 2', 'k = 3', 'k = 4', 'k = 5','k = 6', 'k = 7', 'k = 8', 'k
= 9', 'k = 10'});
```