Control Systems

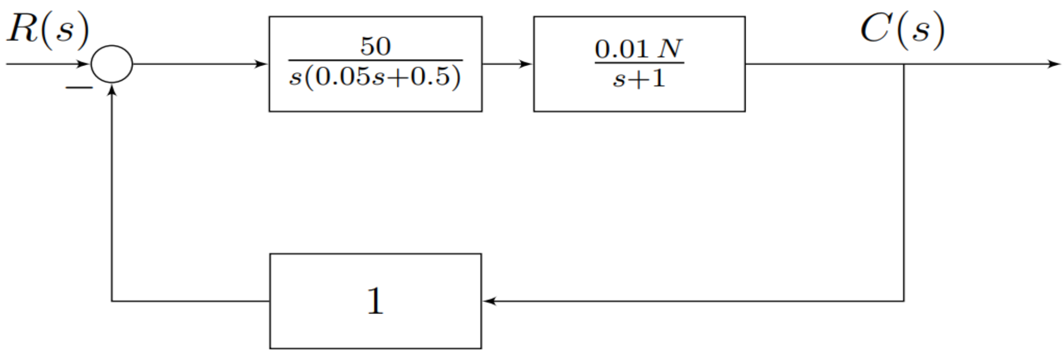
Lab Assessment #4

Date of experiment – 09/09/2023

Group Members: Adarsh Santoria (B21176), Vanshaj Nathani (B20237)

Objective:

Finding an analytically tractable approximation of a higher order system and evaluating its accuracy of approximation.
The higher order system which is being approximated in this lab is -



The above system depicts a liquid level control system, where the level is represented by c(t) and N denotes the number of inlets.

Experiment Design:

As we seen in the pre-lab that the system has to be approximated to 2nd order as its poles are closer to the Y axis, 2nd order approximation is used in the following.

N=10 approximated to 2nd order	ans = $\dfrac{9.15}{s^2 + 0.0773\ s + 9.15}$
N=1 approximated to 2nd order	ans = $\dfrac{0.989}{s^2 + 0.8914\ s + 0.989}$
N=10 Original	$\dfrac{100}{s^3 + 11\ s^2 + 10\ s + 100}$

N=1 Original	tf_org = 10 ------------------------- s^3 + 11 s^2 + 10 s + 10

S.No	N	Peak time - tp (s)	Overshoot - %M (%)
1	1	3.54	20.69
2	10	1.038	96

Note: The derivations/calculations of the above approximations are in the pre-lab pdf which is in the same zip file submitted. Please refer it for the derivation.

1. Peak overshoot and Peak time w.r.t N:

We constructed the 2nd order approximate transfer function by picking the relevant poles and correcting the DC gain. After performing the step analysis, the values of required peak overshoot and peak time are saved and plotted for the required values of N.

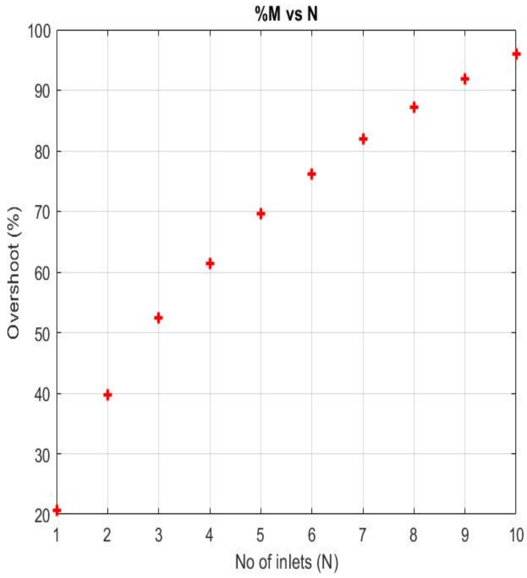
Code:

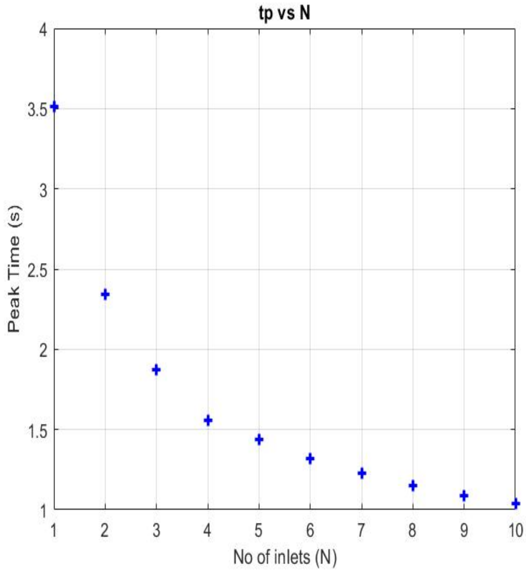
```
%Empty arrays to store the values
tp = [];
mp = [];

%looping the values of n from 1 to 10
for n=1:10
    %finding the zeros and poles
    [z, p, k] = tf2zp([10*n],[1 11 10 10*n]);
    %re-constructing the transfer function of 2nd order
    [b,a] = zp2tf(z,p(2:3),1);
    %adjusting the DC gain and calculating step response
    val = stepinfo(tf(a(3),a));
    % storing the required
    mp(n) = val.Overshoot;
    tp(n) = val.PeakTime;
end

%plotting the figures
figure
plot((1:10), mp,'+','LineWidth',2,Color='r')
xlabel("No of inlets (N)")
ylabel("Overshoot (%)")
title("%M vs N")
grid on
figure
plot((1:10), tp,'+','LineWidth',2,Color='b')
xlabel("No of inlets (N)")
ylabel("Peak Time (s)")
title("tp vs N")
grid on
```

Observations:





1. From these plots we can observe that the overshoot amplitude increases with increase in value of N.
2. This can be understood using a small analogy. Let us consider a case where we have a single hammer punching on a sheet, where the 2nd punch depends on the results of the 1st. On the other hand if we have the same situation with 10 hammers punching at once, though the value of force at 2nd punch might vary according to the result of 1st but initially the amount of shock or punch force experienced by the sheet is higher in the 2nd case.
3. The value of tp decreases wrt N.
4. The both plots may be approximated to some exponential signal (as the shape depicts such relation of high increase or decrease and saturation at later part).
5. The obtained values for N = 1 and 10 closely matches to the values we got in the calculations of prelab which are mentioned in the table above.

2. Step response of Original vs Approximated:

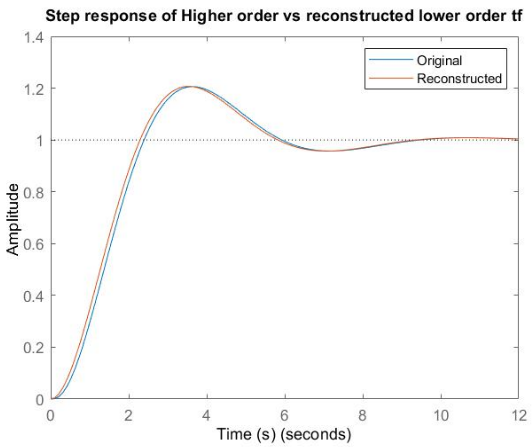
For the given range of N, the smallest and the largest values are 1 and 10 respectively.

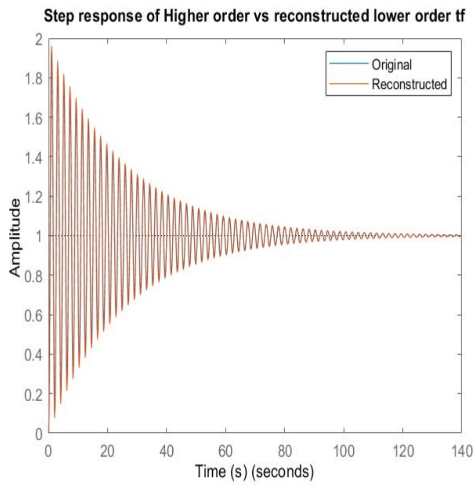
So the plots are made for step response of original and lower order approximated signal for N= 1 and 10.

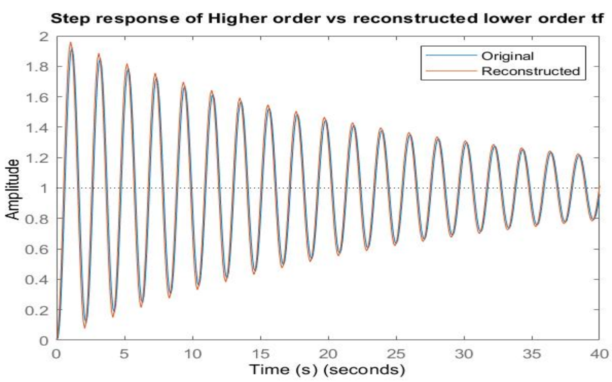
Code:

```
%using a loop for repeating the task
for n=[1,10]
    %original transfer function
    tf_org = tf([10*n],[1 11 10 10*n])
    %re-construcing the transfer function to 2nd order
    [z, p, k] = tf2zp([10*n],[1 11 10 10*n]);
    [b,a] = zp2tf(z,p(2:3),1);
    %adjusting the dc gain
    tf_approx = tf([a(3)],a);
    %plotting the step response of the original and reconstructed tf
    figure
    step(tf_org)
    hold on;
    step(tf_approx)
    xlabel("Time (s)")
    ylabel("Amplitude")
    title("Step response of Higher order vs reconstructed lower order tf")
    legend("Original", "Reconstructed")
end
```

Observations:







1. From the N=1 case we can clearly observe that the reconstructed or approximated signal initially lies above the original, after the peak overshoot it went below the original and finally settled above again.
2. Similarly in N=10 case the upper and lower occupancy trends can be clearly visible.
3. Time taken to reach a close limit (where approximated signal is approximately match with the original signal) is less for N=1 case than at N=10 case. Even after 40s there is significant difference (for N=10 case).
4. Coming to the accuracy, the best approximation can be selected in between N=1 and 10 depending upon the use case. This is because if we want the best fit and the average error should be close to zero, N=1 is the better accurate case. But for some analyses we need to study the behaviour of the signal, where having an envelop of the signal helps more for the stability comparisons and else. In this latter case N=10 is the better as the approximated signal is making a envelope or holding the original signal in both the peaks.
5. So keeping a note on above we can state for the best fit (in view of closest match of amplitudes) that accuracy is more for the lower value of N.

3. The Error signal as a metric of evaluation:

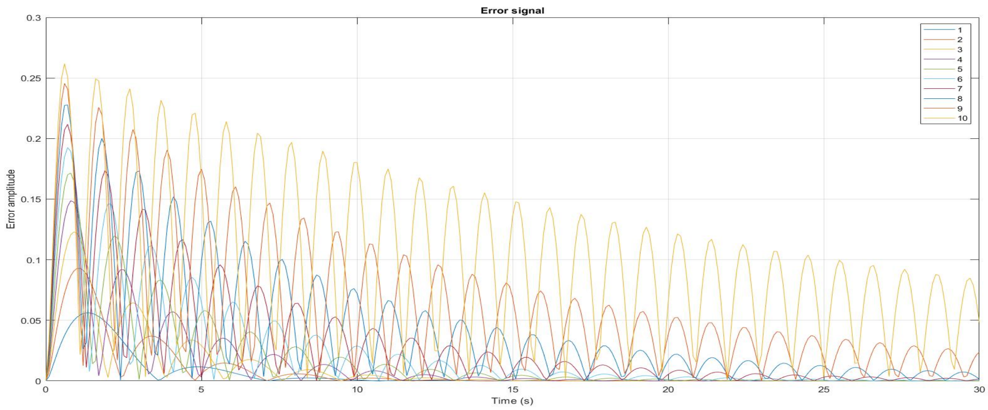
Similarly generating the systems' transfer functions (for both the higher order original and lower order approximate) and calculating the step response, we can calculate the error signal by directly Subtracting The both and applying the absolute to it. The results can be seen below.

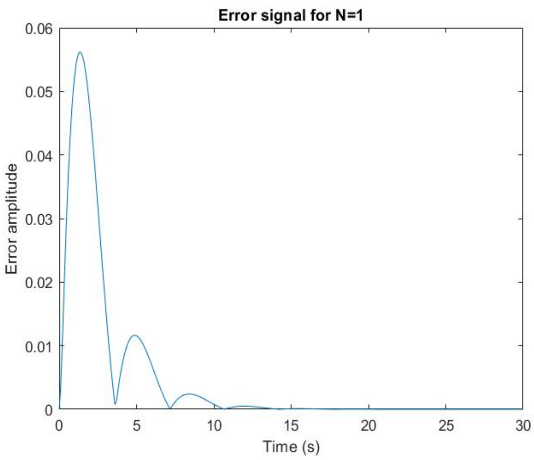
Code:

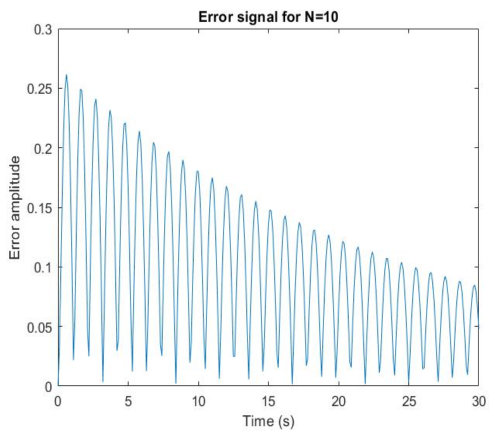
```
%setting the parameters for time vector
t_final = 30;
t_s = 0.1;
%looping the values of n from 1 to 10
for n=1:10
    %original transfer function
    tf_org = tf([10*n],[1 11 10 10*n])
    %reconstructing the tf
    [z, p, k] = tf2zp([10*n],[1 11 10 10*n]);
    [b,a] = zp2tf(z,p(2:3),1);
    tf_approx = tf([a(3)],a);
    %calculating the error signal
    er = abs(step(tf_org,(0:t_s:t_final))-step(tf_approx,(0:t_s:t_final)))
    %plotting the error signal
    plot((0:t_s:t_final),er);
    xlabel("Time (s)")
    ylabel("Error amplitude");
    hold on;
    title("Error signal")
end

grid on;
legend('1','2','3','4','5','6','7','8','9','10');
```

Observations:







1. We can observe that amplitude of the first peak is around 0.055 in case of N=1 and 0.26 in case of N=10.
2. The time period of oscillations too decreases as N increases.
3. The time taken for the error signal to be dead (approximately zero), too increases with increase in N.
4. The magnitude of Error too increases with value of N.
5. We can verify the accuracy comment in the above part using these comparisons.

Improvements and Learnings:

1. We can create a Simulink model of the system by giving a slider parameter for N can compare the change in real time.
2. By defining the required accuracy and other metrices we can make an automated system or an algo code which can suggest the optimal value of N as per the use case.
3. Got the exposure to approximating techniques by which many of the real-world systems (which are usually higher order) can be analysed.
4. Got a small idea that if we can adjust that error by adding any linear component, i.e. breaking the higher order systems to the superposition or combination of lower order systems where one of them can be adjusted for the error.

Prelab Calculations for TF, tp and %M:

a)

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)} \qquad G(s) = \frac{50}{s(0.05s+0.5)} \times \frac{0.01N}{s+1}$$

$$= \frac{N}{2s(s+1)(s+10)\cdot 0.05}$$

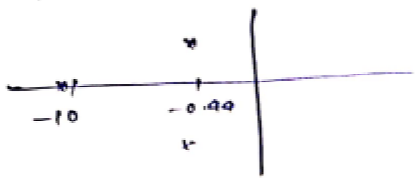
$$= \frac{10N}{s(s+1)(s+10)}$$

$$\frac{C(s)}{R(s)} = \frac{10N}{s(s+1)(s+10)} \times \frac{s(s+1)(s+10)}{s(s+1)(s+10)+10N}$$

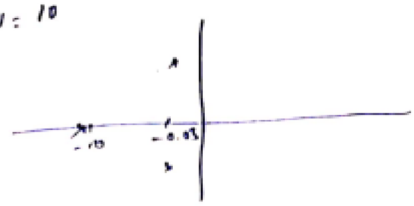
$$\frac{C(s)}{R(s)} = \frac{10N}{s^3+11s^2+10s+10N}$$

c).

N = 1



N: 10



Approximating to 2nd order system.

$H(s) = ?$

$-\zeta\omega_n = -0.4457$

$\omega_n\sqrt{1-\zeta^2} = 0.88916$

$$\frac{\sqrt{1-\zeta^2}}{\zeta} = 1.995$$

$\zeta = 0.448$

$\omega_n = 0.995$

$$H(s) = \frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$$

$$= \frac{0.989}{s^2+s(0.4457)+0.989}$$

$$t_p = \frac{\pi}{\omega_d} = \frac{\pi}{0.88916} \cong 3.54\,s$$

$$\%M = e^{-\pi\zeta/\sqrt{1-\zeta^2}} \times 100 = e^{-\pi/1.995} = 20.69\%$$

$-\zeta\omega_n = -0.03862$

$\omega_n\sqrt{1-\zeta^2} = 3.0255$

$$\frac{\sqrt{1-\zeta^2}}{\zeta} = 78.24$$

$\zeta = 0.0128$

$$\omega_n = \frac{0.03817}{0.0128} = 3.02$$

$$H(s) = \frac{9.12}{s^2+s(0.77)+9.12}$$

$$t_p = \frac{\pi}{\omega_d} = 1.038\,s$$

$$\%M = e^{-\pi\zeta/\sqrt{1-\zeta^2}} = e^{-\pi/78.24} \times 100$$

$\approx 96\ \%$