③ a) $G(s) = \dfrac{2(s+2)}{s^2+2s-3} e^{-s}$.

NEW QUESTION (updated)

$\Rightarrow G_{approx}(s) = \dfrac{2(s+2)}{s^2+2s-3} \dfrac{\left(1-\frac{s}{2}\right)}{\left(1+\frac{s}{2}\right)}$

$= \dfrac{2(2-s)}{s^2+2s-3}$.

C.E $= 1 + G_c G_p = 0$

$\Rightarrow 1 + K_c \dfrac{2(2-s)}{(s^2+2s-3)} = 0$

-0.2 is a root!

$\Rightarrow K_c = \dfrac{-\left[(-0.2)^2 + 2(-0.2) - 3\right]}{2(2+0.2)}$

$\Rightarrow G_{c1} = 0.76364$

③ c) 2nd order approx: $L = \dfrac{2(s+2)\left(1-\frac{s}{2}+\frac{s^2}{8}\right)}{(s^2+2s-3)\left(1+\frac{s}{2}+\frac{s^2}{8}\right)}$

$\Rightarrow L = \dfrac{64s^3 - 128s^2 + 1024}{32s^4 + 192s^3 + 416s^2 + 128s - 268}$

Using rltool, the value of K_c for which the dominant poles has real part -0.2 is 0.7878.

Dominant poles : $-0.2 + 0.37j$

Question 3 c) updated

For all simulations, the variance of disturbance was set as 0.1

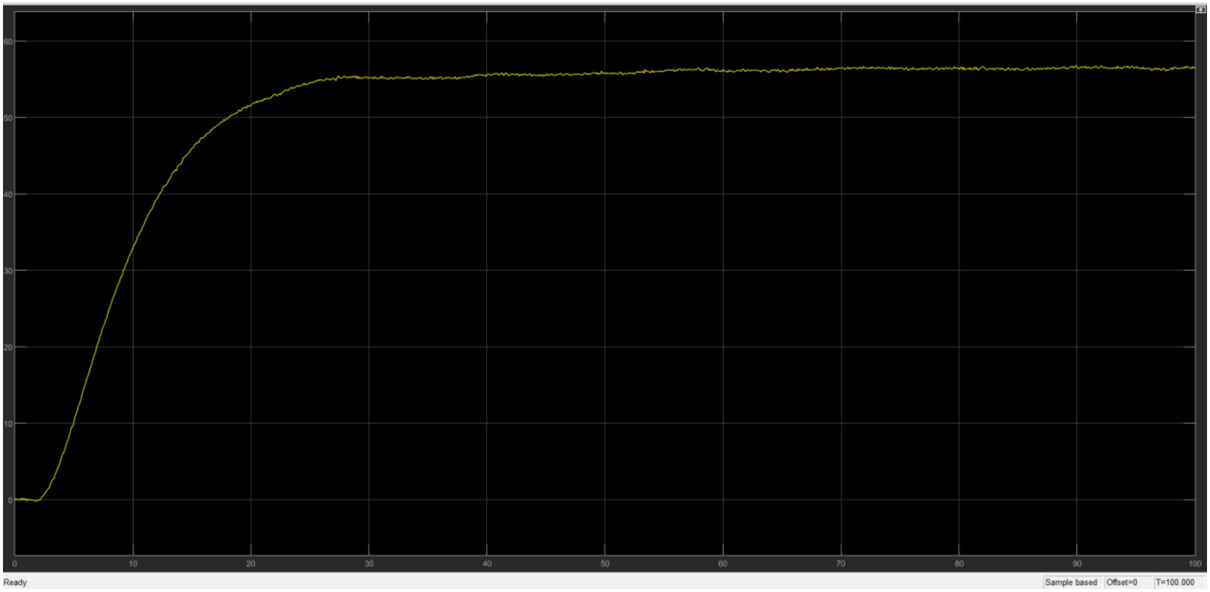

Figure: Plot of step response of the closed loop system for Kc = 0.7636 (part-a)

We see that G_{C1} has an offset of about 55.

For controller design using Pade's second order approximation, we use the rltool on L to get -0.2 as real part of the dominant pole as shown in the below figure

```
Gp_pade_second =

         64 s^3 - 128 s^2 + 1024
  ----------------------------------------
  32 s^4 + 192 s^3 + 416 s^2 + 128 s - 768

Continuous-time transfer function.
```

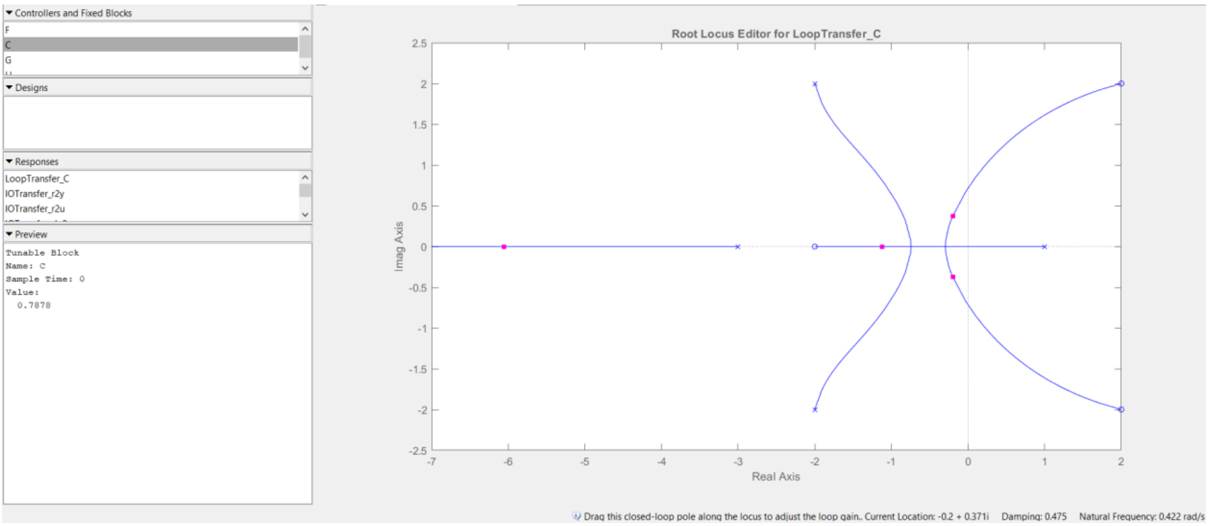

Figure: RL plot with the required roots marked. (along with the K_C value)

K_C = 0.7878

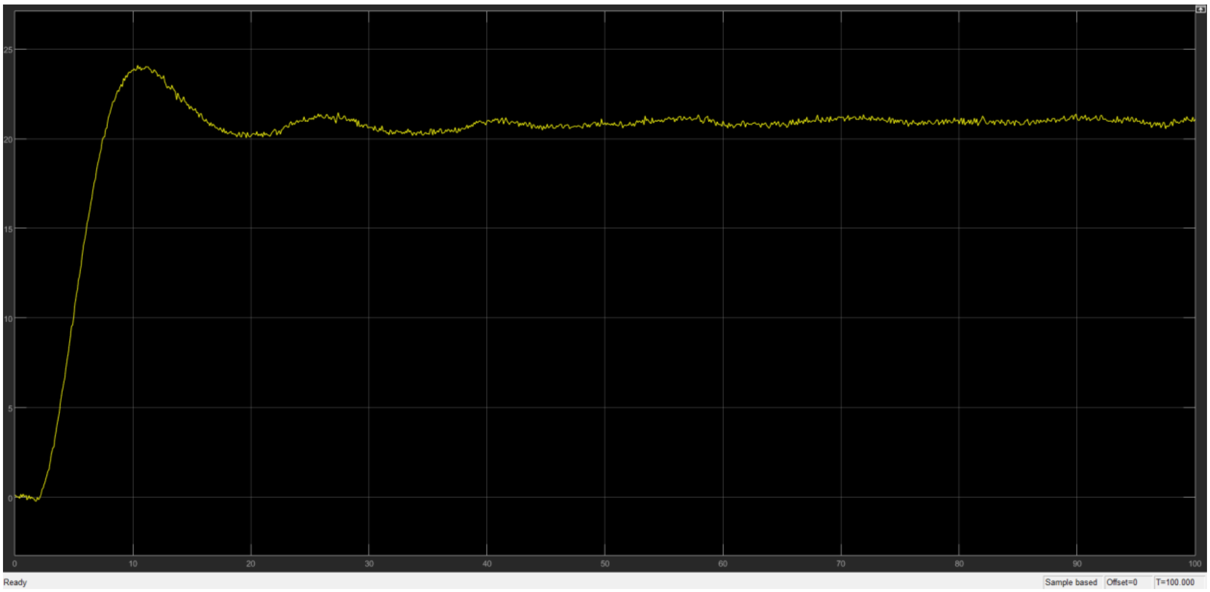

Figure: step response of the closed loop system with K_C = 0.7878

Conclusion: Closed loop system with controller from part b) is unstable as shown earlier. We see that the offset with controller from part a) (~55) is much greater than controller from part c) (~21). So we conclude that controller proposed by utilizing Pade's second order approximation has proven to be more effective in dealing with the actual system.

<u>Code:</u>

```
clear; close all;
%% Setup the system
s = tf('s');
Gp = 2*(s+2)/(s^2+2*s-3)*exp(-s);
%% 3a
Gp_pade = 2*(2-s)/(s^2+2*s-3);
f = @(s)(2*(2-s)/(s^2+2*s-3));
Kc_a = -1/(f(-0.2));
poles_parta = pole(1/(1+Kc_a*Gp_pade))
%% 3c
Gp_pade_second = 2*(s+2)*(1-s/2+s^2/8)/((s^2+2*s-3)*((1+s/2+s^2/8)));
rltool(Gp_pade_second)
%0.7878
```