EE 324 Control Systems Lab

Problem Sheet 2 : Analysis in time domain

Mayur Ware | 19D070070

August 8, 2021

Question 1

$$G(s) = \frac{a}{s+b}$$

My Roll No. is 19D070070. So, a = 70. Also, my name is Mayur. So, b = 13.

$$\therefore G(s) = \frac{70}{s+13}$$

Part (a) : Build LTI System

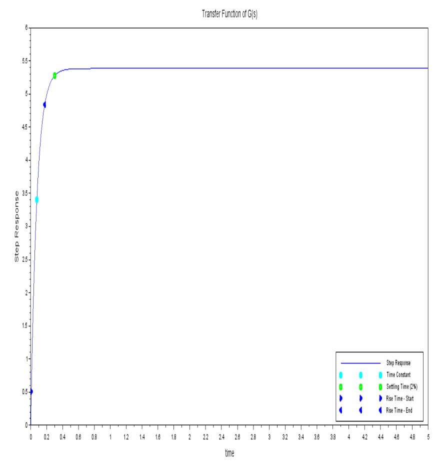
LTI System of G can be made in `SciLab` using `syslin`

```
s = poly(0,'s');
a=70; b=13;
G = syslin('c', a/(s+b));
t = 0:0.01:5;
plot2d(t, csim('step', t, G), style=[color("blue")]);
xgrid(0);
title(["Step Response of G(s)"]);
xlabel("t");
ylabel("Step Response");
```

Part (b) : Unit Step Response

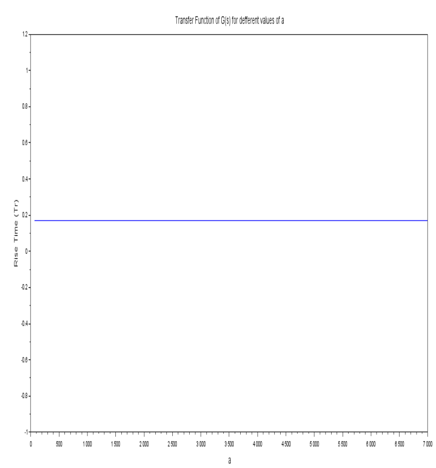
Find and plot time constant, 2% settling time, and rise time of the system. Using `SciLab`, we get :

```
t = 0:(1/260):5;
S = csim('step', t, G);
plot(t, S, style=[color("blue")]);
set(gca(),"data_bounds",[ 1 , 0 .5;5,3]);
time_c = 1/b;
time_c_y = S(21);
plot(time_c, time_c_y, 'c*', 'LineWidth', 4);
time_set = time_c .* log(50);
plot(time_set, S(1+(time_set .* 260)), 'go', 'LineWidth', 4);
time_r_low = time_c .* log(10/9);
time_r_high = time_c .* log(10/1);
plot(time_r_low, S(1+(time_r_low .* 260)), 'o>', 'LineWidth', 4);
plot(time_r_high, S(1+(time_r_high .* 260)), 'o<', 'LineWidth', 4);
legend(['Step Response', 'Time Constant', 'Settling Time (2%)', 'Rise Time - Start',
'Rise Time - End'], 4);
xlabel("time",'fontsize', 3);
ylabel("Step Response",'fontsize', 3);
title(["Transfer Function of G(s) for defferent values of a"], 'fontsize', 3);
```



Part (c) : Dependency of Rise Time (Tr) on 'a'
The rise time for the above system is Tr = $\frac{ln(9)}{b}$. This is independent of a. So, the graph will be a straight line.
Using `SciLab`, we get :

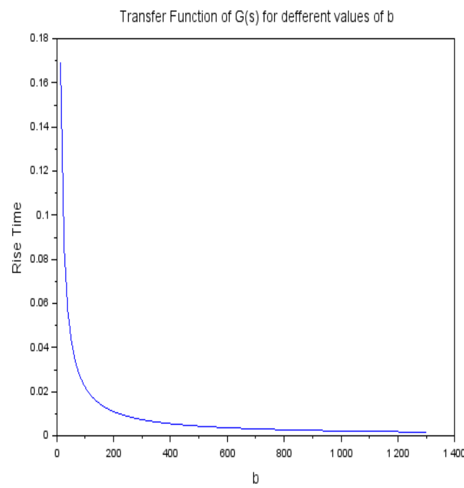
```
P = a:a:100*a;
C = time_r_high     time_r_low;
time_rise = ones(P).*C;
plot(P, time_rise, 'c_');
xlabel("a", 'fontsize', 3);
ylabel("Rise Time (Tr)", 'fontsize', 3);
title(["Transfer Function of G(s) for defferent values of a"], 'fontsize', 3);
```



Part (d) : Dependency of Rise Time (Tr) on 'b'
The rise time for the above system is Tr = $\frac{ln(9)}{b}$. As we can notice, Tr is inversly proportional to 'b'. So, the graph will be a decaying function.

```
Q = b:b:100*b;
time_rise = (1./Q) .* log(9);
```

```
plot(Q, time_rise, 'c_');
xlabel("b", 'fontsize', 3);
ylabel("Rise Time", 'fontsize', 3);
title(["Transfer Function of G(s) for defferent values of b"], 'fontsize', 3);
```



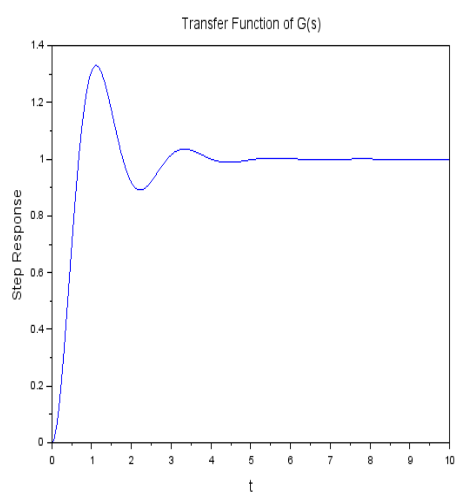
Question 2

By varying damping ratio from 0 to 2 in steps of 0.25. Observe how percentage-overshoot, rise-time, 2% settling time, and peak-time change with change in damping ratio.

$$G(s) = 9/(s^2 + 2 * s + 9)$$

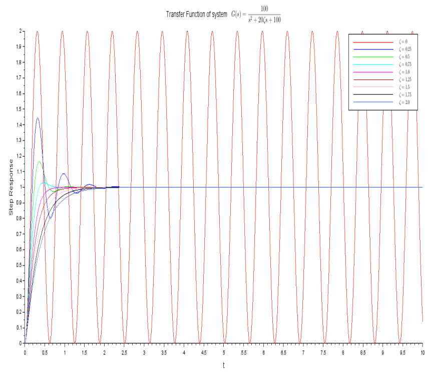
As we can notice, Damping Ratio (ζ) = 1/3 and Natural Frequency (ω_n) = 3.

Part (a) : Plot Step Response




```
G = syslin('c', 9/(s^2+2*s+9));
t = 0:0.005:5
plot(t, csim('step', t, G), style=[color("blue")]);
xlabel("t",'fontsize', 3);
ylabel("Step Response",'fontsize', 3);
title(["Transfer Function of G(s)"], 'fontsize', 3)
```

Part (b) : Varying ζ



As we increase ζ,

1) Peak Time (Tp) increases

2) Rise Time (Tr) increases

3) Settling Time (Ts) decreases

4) Overshoot %OS decreases

```
Z = 0:0.25:2;
Wn = 10;
colors = ["red", "blue", "green", "cyan", "magenta", "brown", "pink", "black", "royalblue"];
for i=1:size(Z, 2)
G = syslin('c', Wn^2 / (s^2 + 2 * Z(i) * Wn * s + Wn^2));
plot2d(t, csim('step', t, G), style=[color(colors(i))], 'LineWidth', 2);
xlabel("t", 'fontsize', 3);
ylabel("Step Response", 'fontsize', 3);
end
title(["Transfer Function of system", "$G(s)=\frac{100}{s^2 + 20\zeta s + 100}$"]);
```

Question 3

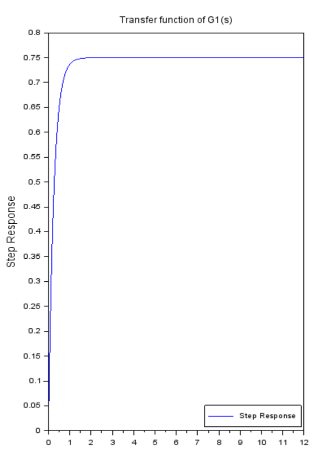
The Systems I have assumed are,

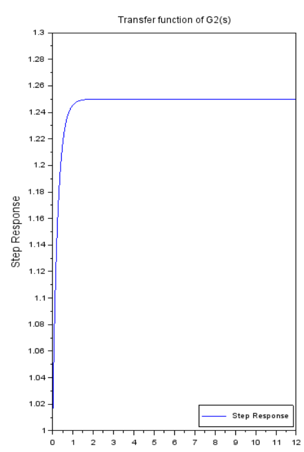
First Order System (G1(s)) $= 3/(s+4)$

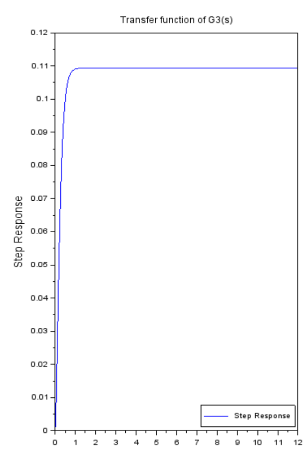
Second Order Overdamped System (G2(s)) $= 5/(s^2+5s+4)$

First Order Critically damped System (G3(s)) $= 7/(s+8)^2$

The Unit Step Responses of the systems are as follows :







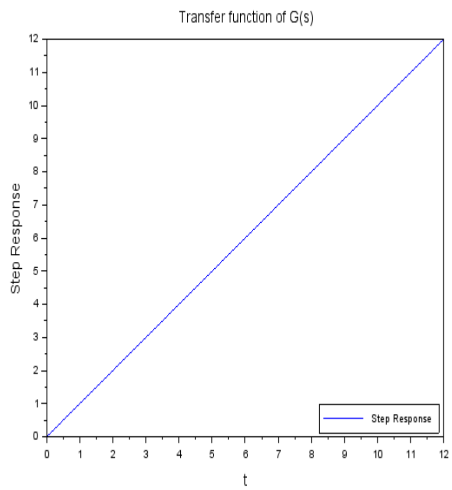
```
G1 =  syslin('c',3/(s+4));
G2= syslin('c', 5/(s^2+5*s+4);
G3 = syslin('c', 7/(s+8)^2);
t = 0:0.01:12;
subplot(1,3,1),plot(t, csim('step', t, G1), style=[color("blue")]);
title(["Transfer function of G1(s)"], 'fontsize', 2);
xlabel("t", 'fontsize', 3);
ylabel("Step Response", 'fontsize', 3);
legend("Step Response", 4);
subplot(1,3,2),plot(t, csim('step', t, G2), style=[color("blue")]);
title(["Transfer function of G2(s)"], 'fontsize', 2);
xlabel("t", 'fontsize', 3);
ylabel("Step Response", 'fontsize', 3);
legend("Step Response", 4);
subplot(1,3,3),plot(t, csim('step', t, G3), style=[color("blue")]);
title(["Transfer function of G3(s)"], 'fontsize', 2);
xlabel("t", 'fontsize', 3);
ylabel("Step Response", 'fontsize', 3);
legend("Step Response", 4);
```

Question 4

Part (a) : Continuous Time Transfer Function

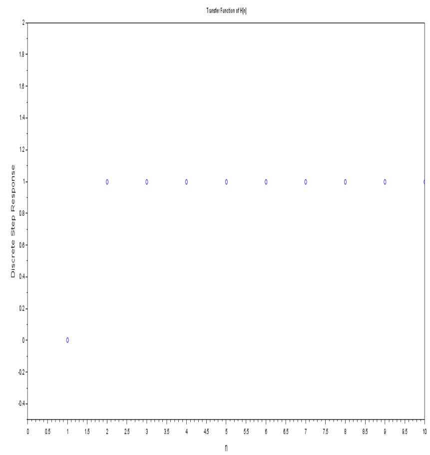
$$G(s) = 1/s$$

Unit Step Response using `SciLab` is as follows :




```
G = syslin('c', 1/s);
plot(t, csim('step', t, G), style=[color("blue")]);
title(["Transfer function of G(s)"], 'fontsize', 4);
xlabel("t", 'fontsize', 3);
ylabel("Step Response", 'fontsize', 3);
legend("Step Response", 4);
```

Part (b) : Discrete Time Transfer Function
Discrete time transfer function $H(z) = 1/z$. The discrete time step response can be given as :




```
z = poly(0,'z');
H = 1/z;
sl=tf2ss(H);
u1=ones(1,10);
plot(dsimul(sl,u1), 'bo');
set(gca(),"data_bounds",[0,−0.5;10,2]);
xlabel("n", 'fontsize', 3);
ylabel("Discrete Step Response", 'fontsize', 3);
title(["Transfer Function of H[n]"]);
```

Part (c) : Fraction

$$P = \frac{G(s)}{H(z)} = z/s$$

When we use `csim` of P, `SciLab` gives the following error :
`WARNING: csim:  Input argument 1 is assumed continuous time.`
This happened because H(z) is a discrete-time transfer function and because of it, P is also an discrete-time transfer function.
Now, as `csim` requires a continuous-time transfer function to work, it gives an error.
4(a) :
$C(s) = R(s) * 1/s = 1/s^2$

$$\implies c(t) = t * u(t)$$

4(b) :
$C(z) = R(z) * 1/z = \frac{1}{z^2 - z}$

$$\implies c[n] = u[n-1]$$

Question 5

The given transfer function is :

$$G(s) = \frac{s+5}{(s+4)(s+2)}$$

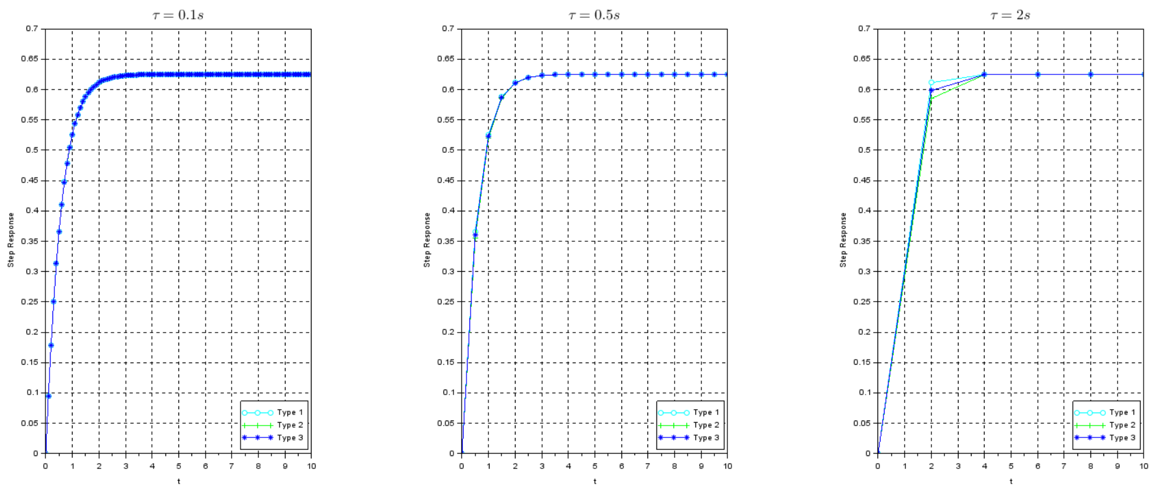
Let's assume,

$$G1(s) = \frac{s+5}{(s+4)(s+2)}$$

$$G2(s) = \frac{s+5}{s+4}$$

$$G3(s) = \frac{1}{s+2}$$

Now,
System 1 : G1(s)
System 2 : G2(s)*G3(s)
System 3 : G3(s)*G2(s)
Unit step responses for all these three configurations by changing the sampling period τ, we get :



With increasing τ, plots can be seen getting differentiated. For $\tau = 2s$, all the three system can be seen.

```
//tau is each of 0.1 seconds, 0.5 seconds, and 2 seconds
tau1 = 0:0.1:10 //Tau1 = 0.1s
tau2 = 0:0.5:10 //Tau2 = 0.5s
tau3 = 0:2:10 //Tau3 = 2s
s = poly(0,'s');
G1 = syslin('c', (s+5)/((s+4)*(s+2)));
G2 = syslin('c', (s+5)/(s+4));
G3 = syslin('c', 1/(s+2));
//Tau = 0.1Sec
Res11 = csim('step', tau1, G1);
Res12 = csim(csim('step', tau1, G2), tau1, G3);
Res13 = csim(csim('step', tau1, G3), tau1, G2);
subplot(1,3,1),plot(tau1, Res11, 'c-o');
subplot(1,3,1),plot(tau1, Res12, 'g-+');
subplot(1,3,1),plot(tau1, Res13, 'b-*');
legend(["Type 1", "Type 2", "Type 3"], 4);
xgrid(0);
title("$\tau = 0.1s$", 'fontsize', 4);
xlabel("t");
ylabel("Step Response");
//Tau = 0.5Sec
Res21 = csim('step', tau2, G1);
Res22 = csim(csim('step', tau2, G2), tau2, G3);
Res23 = csim(csim('step', tau2, G3), tau2, G2);
```

```
subplot(1,3,2),plot(tau2, Res21, 'c--o');
subplot(1,3,2),plot(tau2, Res22, 'g--+');
subplot(1,3,2),plot(tau2, Res23, 'b--*');
legend(["Type 1", "Type 2", "Type 3"], 4);
xgrid(0);
title("$\tau = 0.5s$", 'fontsize', 4);
xlabel("t");
ylabel("Step Response");
//Tau = 2Sec
Res31 = csim('step', tau3, G1);
Res32 = csim(csim('step', tau3, G2), tau3, G3);
Res33 = csim(csim('step', tau3, G3), tau3, G2);
subplot(1,3,3),plot(tau3, Res31, 'c--o');
subplot(1,3,3),plot(tau3, Res32, 'g--+');
subplot(1,3,3),plot(tau3, Res33, 'b--*');
legend(["Type 1", "Type 2", "Type 3"], 4);
xgrid(0);
title("$\tau = 2s$", 'fontsize', 4);
xlabel("t");
ylabel("Step Response");
```

References

1) https://help.scilab.org/
2) https://spoken-tutorial.org/tutorial-search/?search_foss=Scilabsearch_language=English
3) https://spoken-tutorial.org/tutorial-search/?search_foss=Scilabsearch_language=Englishpage=2