EE324 - Control Systems Lab

PROBLEM SHEET 2

Prasann Viswanathan | 190070047

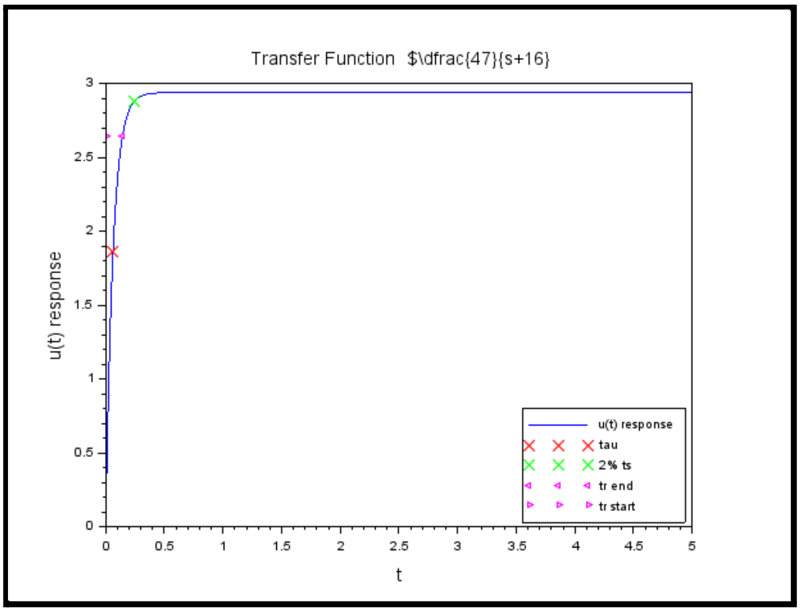
Question 1:

a. For me, a=47 and b=indexof(P)=16. Therefore the code to define the resulting tf is:

```
--> s = poly(0, 's');
--> G = 47/(s+16);
--> sys1 = syslin('c', G)
 sys1  =
    47
   -----
   16 +s
```

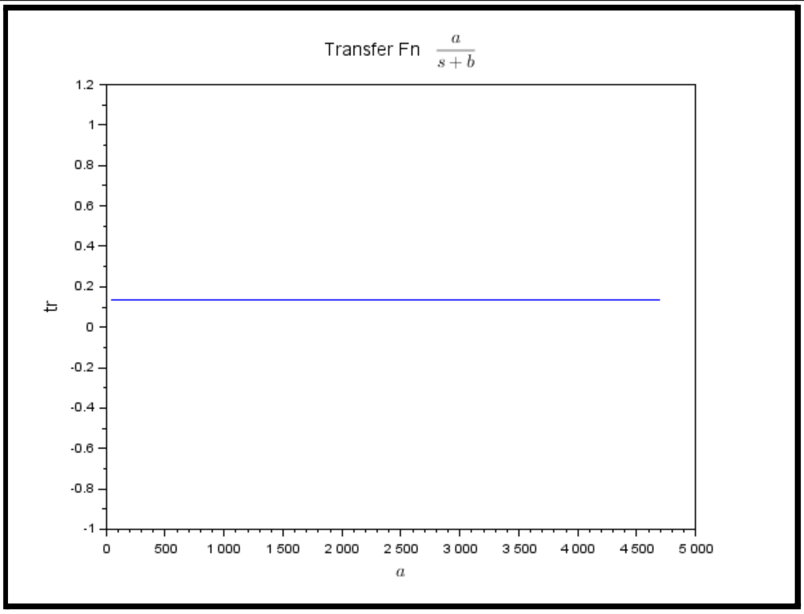
b. Response of my system to a unit step response, marking time constant, 2% settling and rise times:

```
--> t = 0:0.001:5;
--> y1=csim('step', t, sys1);
--> plot(t, y1);
--> tau = 1/16;
--> rise_t1 = log(100/10)/16;
--> rise_t2 = log(100/90)/16;
--> t_s = log(50)/16;
--> plot(tau, csim('step', tau, sys1), 'rx');
--> plot(t_s, csim('step', t_s, sys1), 'gx');
--> plot(rise_t1, csim('step', rise_t1, sys1), 'm<');
--> plot(rise_t2, csim('step', rise_t1, sys1), 'm>');
--> xlabel("t", 'fontsize', 3);
--> ylabel("u(t) response", 'fontsize', 3);
--> title(["Transfer Function", "$\dfrac{47}{s+16}"], 'fontsize', 3);
--> legend(['u(t) response','tau','2% ts','tr end','tr start'],4);
--> xs2png(gcf(), "Q1b.jpg");
```



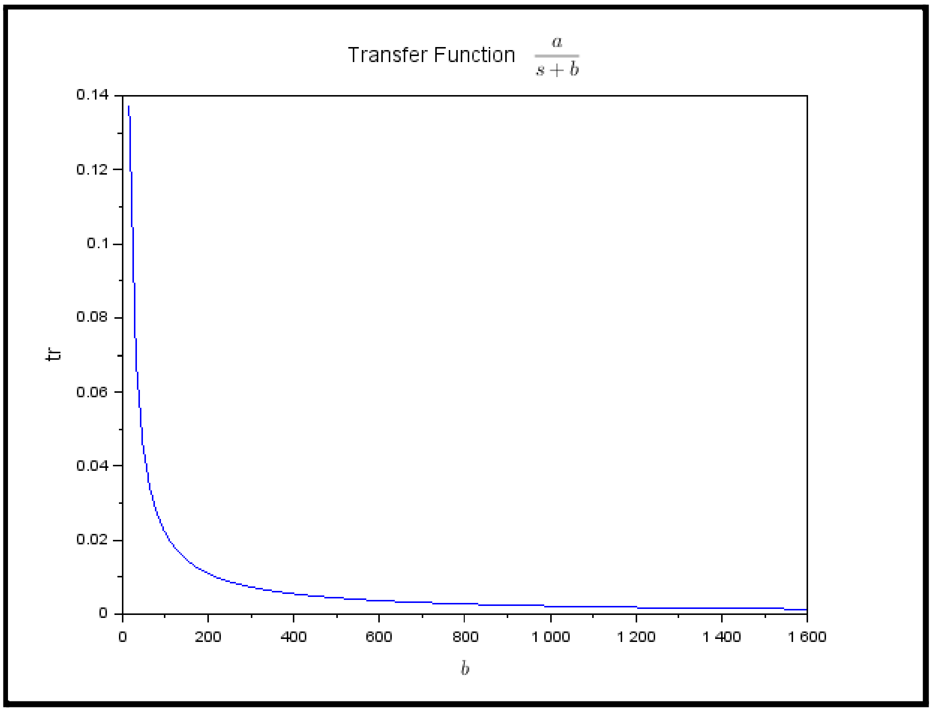
c. To vary the value of a=47 to a=4700, and plot the change in rise time, the code is as:

```
--> a=47; b=16;
--> rise_t1=log(100/10)/b; rise_t2=log(100/90)/b;
--> A_list = a:a:100*a;
--> tr = ones(A_list).*(rise_t1-rise_t2);
--> plot(A_list, tr);
--> xlabel("$a$", 'fontsize', 3); ylabel("tr", 'fontsize', 3);
--> title(["Transfer Fn", "$\dfrac{a}{s+b}$"], 'fontsize', 3);
--> xs2png(gcf(), "Q1c.jpg");
```



d. To vary the value of b=16 to b=1600, and plot the change in rise time, the code is as:

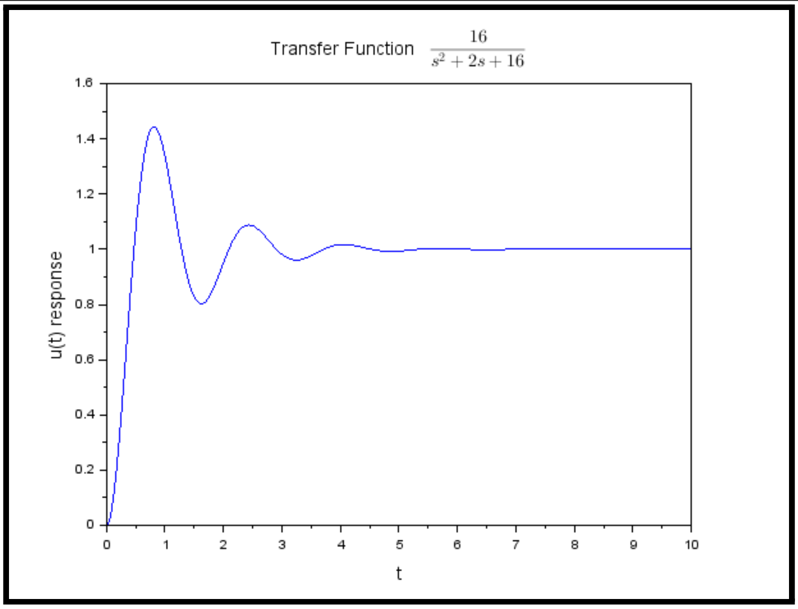
```
--> b=16;
--> B = b:b:100*b;
--> tr = log(9)./B;
--> plot(B, tr);
--> xlabel("$b$", 'fontsize', 3); ylabel("tr", 'fontsize', 3);
--> title(["Transfer Function", "$\dfrac{a}{s+b}$"], 'fontsize', 3);
--> xs2png(gcf(), "Q1d.jpg");
```



Question 2:

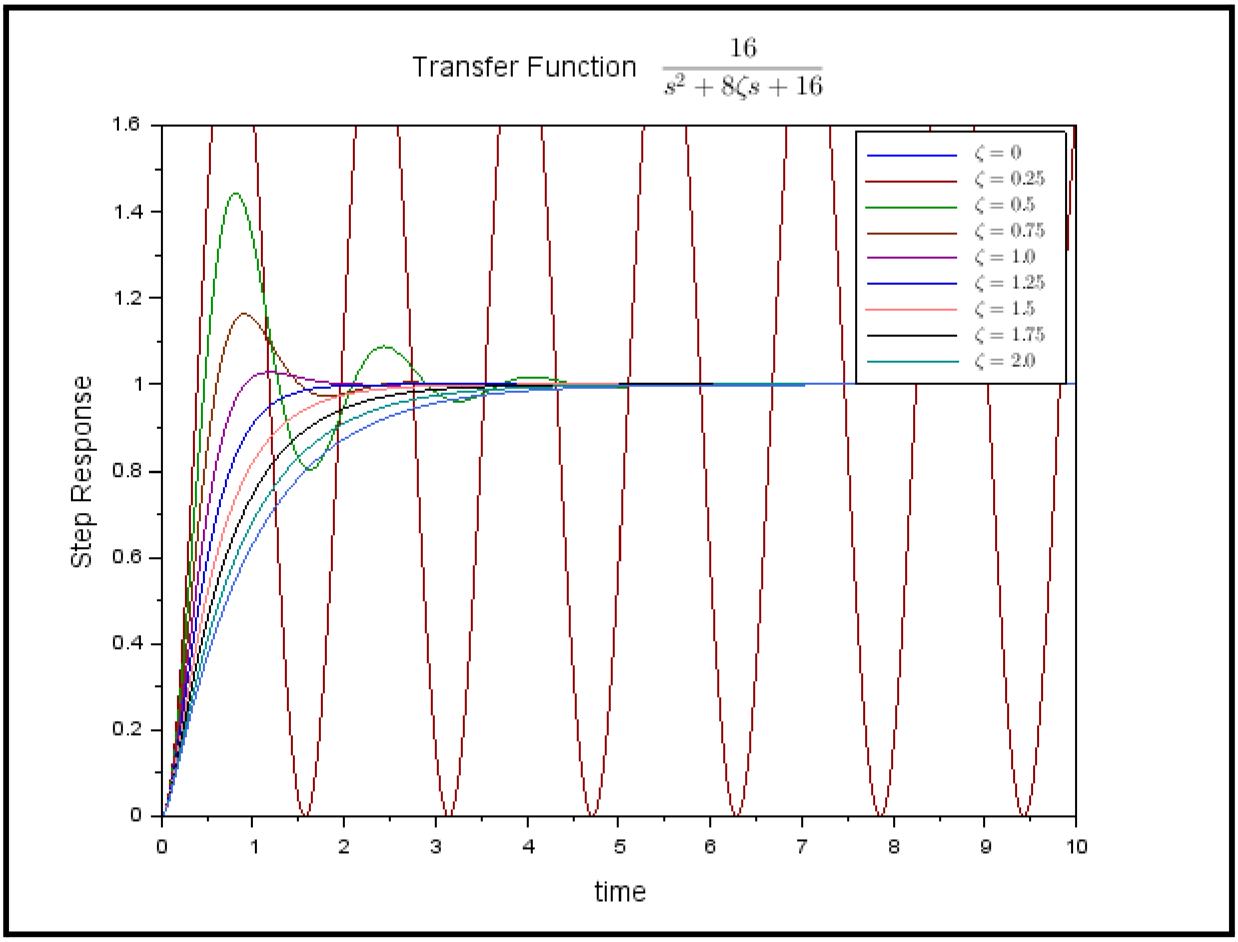
a. I am considering the transfer function **G(s) = 16/(s^2+2s+16)** so we have:
Damping ratio = ¼ and **Natural Frequency =** 4 in this example.

```
--> s=poly(0,'s');
--> G2 = 16/(s^2 + 2*s + 16);
--> sys2 = syslin('c', G2);
--> t = 0:0.001:10;
--> plot(t, csim('step', t, sys2));
--> xlabel("t",'fontsize', 3);
--> ylabel("u(t) response",'fontsize', 3);
--> title(["Transfer Function", "$\dfrac{16}{s^2+2s+16}$"], 'fontsize', 3);
--> xs2png(gcf(), "Q2a.jpg");
```



b. Keeping the natural frequency of our system fixed at 4, if we wish to vary the damping ratio from 0 to 2 in steps of 0.25, each time our transfer function remains of the form **G(s) = 16/(s^2+8*zeta*s+16)**
The code to plot these is:

```
--> z = 0:0.25:2; omega = 4;
--> line_colours = [ "scilabred4", "scilab green4", "scilabbrown4", "scilabmagenta4",
"scilab blue2", "scilabpink4", "black", "scilab cyan4", "royalblue"];
--> for i=1:size(z, 2)
  > G = omega^2/(s^2+2*z(i)*omega*s+omega^2);
  > sys = syslin('c', G);
  > plot2d(t, csim('step', t, sys), style=[color(line_colours(i))]);
  > xlabel("time",'fontsize', 3); ylabel("Step Response",'fontsize', 3);
  > end
--> legend(["$\zeta = 0$", "$\zeta = 0.25$", "$\zeta = 0.5$", "$\zeta = 0.75$", "$\zeta
= 1.0$", "$\zeta = 1.25$", "$\zeta = 1.5$", "$\zeta = 1.75$", "$\zeta = 2.0$"]);
--> title(["Transfer Function", "$\dfrac{16}{s^2+8 \zeta s+16}$"], 'fontsize', 3);
--> xs2png(gcf(), "Q2b.jpg");
```



- One can observe that as ζ increases, the rise time and the peak time both increase, till ζ=1, after which these values are not well defined and peak time becomes infinite.
- It is also observed that the settling time decreases till we reach the critically damped condition after which it increases. One must also note that settling time for ζ=0 is infinite.
- The % overshoot decreases from ζ=0 till ζ=1, after which it remains zero for values of ζ >= 1

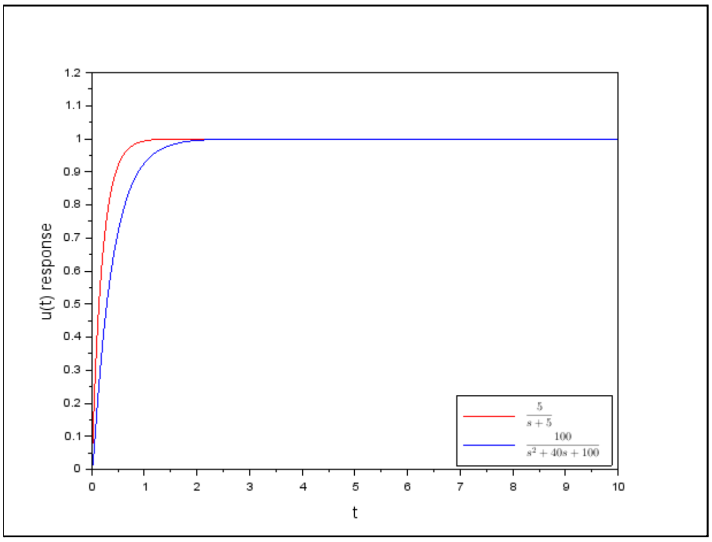
Question 3:

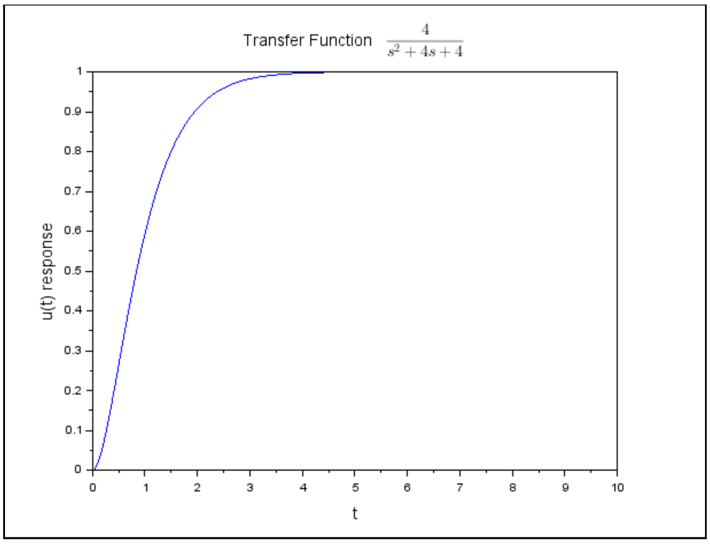
a. Since all first order systems of the form G(s)=a/(s+a) rise monotonically from 0 to 1, let us arbitrarily pick **G(s) = 5/(s+5)**
b. In order to have a second order system which monotonically rises from 0 to 1, the system must be overdamped or have repeated roots. Since the next part of the question requires us to check a system with repeated root, arbitrarily choosing an overdamped system, we have
   G(s) = 100/(s^2+40s+100)   [ζ = 2]
c. Finally, a transfer function with repeated poles we could consider is **G(s) = 4/(s^2+4s+4) [ζ = 1]**

Now, the SciLab code to check the responses of these systems is as follows:

```
--> s = poly(0, 's');
--> G1 = 5/(s+5); sys1 = syslin('c', G1);
--> G2 = 100/(s^2+40*s+100); sys2 = syslin('c', G2);
--> G3 = 4/(s^2+4*s+4); sys3 = syslin('c', G3);
--> t = 0:0.001:10;
--> y1 = csim('step', t, sys1); y2 = csim('step', t, sys2);
--> xlabel("t", 'fontsize', 3); ylabel("u(t) response", 'fontsize', 3);
--> plot(t, y1, "r");
--> plot(t, y2);
--> legend(["$\dfrac{5}{s+5}$", "$\dfrac{100}{s^2+40s+100}$"], 4);
```

```
--> xs2png(gcf(), "Q3ab.jpg");

--> y3 = csim('step', t, sys3); scf();
--> plot(t, y3); xlabel("t", 'fontsize', 3); ylabel("u(t) response", 'fontsize', 3);
--> title(["Transfer Function", "$\dfrac{4}{s^2+4s+4}$"], 'fontsize', 3);
--> xs2png(gcf(), "Q3c.jpg");
```





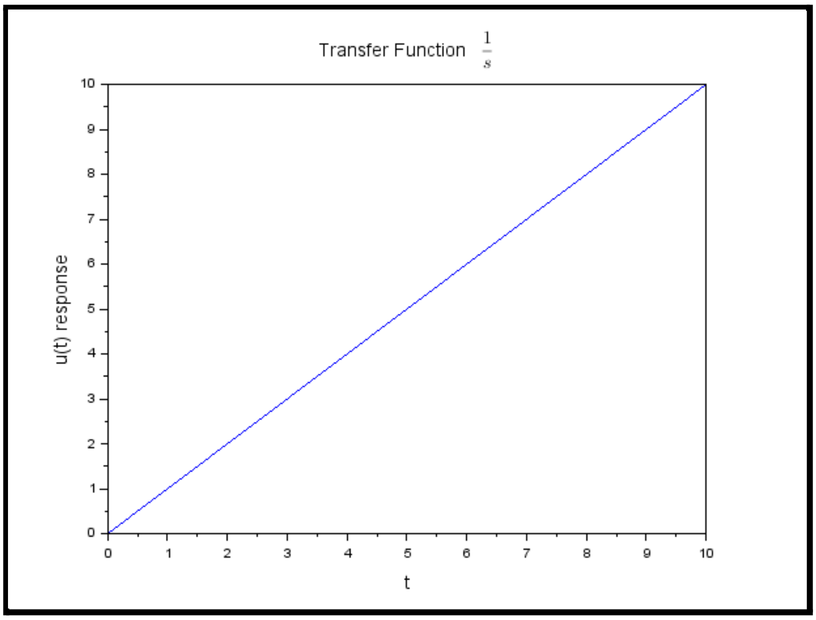
Both systems have a steady state value of 1 and reach it monotonically. The differences are as follows:

1. The second order unit step response has a kink at the origin (not visible in plot) which is absent in the first order unit step response. This happens because the derivative of the second order system's response at origin is 0 unlike that of the first order system at origin.
2. The first order system has a lower rise and settling times compared to the second order system.
3. For the system with repeated poles, we can observe it indeed is **monotonically** increasing from 0 to 1

Question 4:

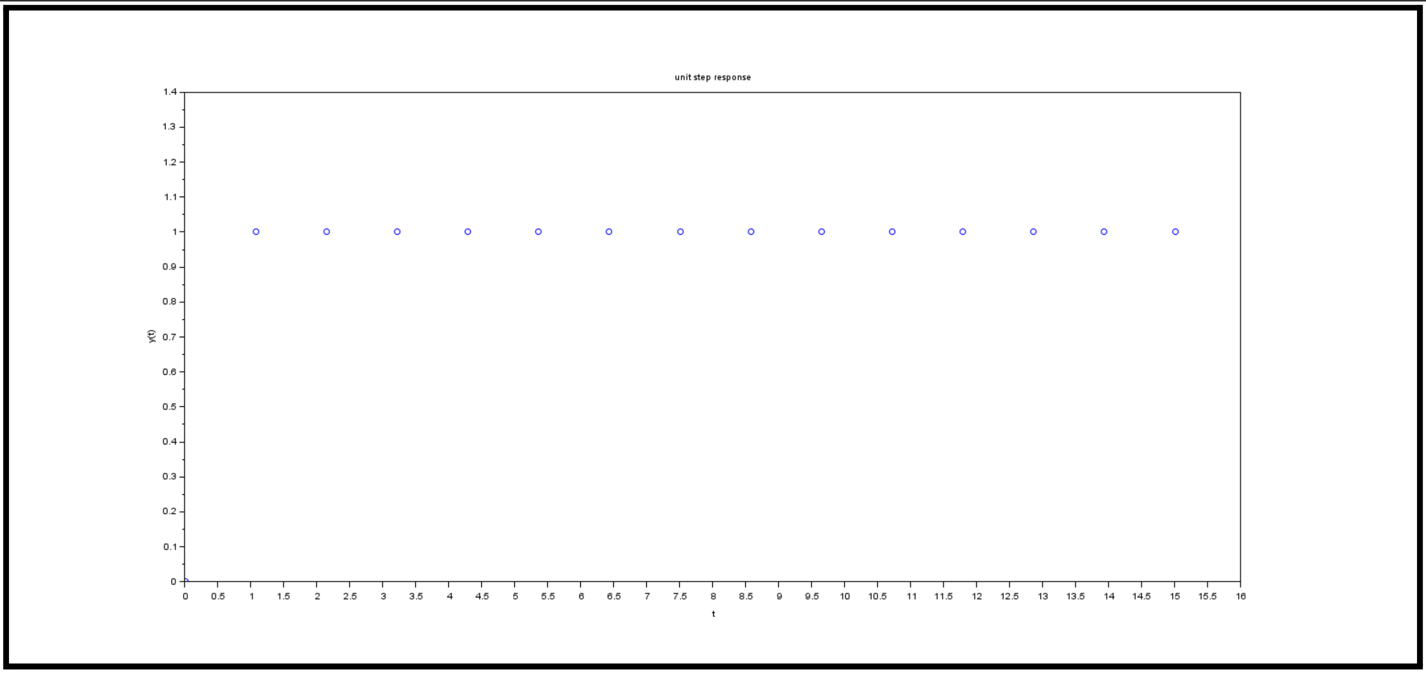
a. The code to plot the unit step response to the transfer function **G(s) = 1/s** is as follows:

```
--> s = poly(0, 's');
--> G = 1/s; sys = syslin('c', G);
--> t = 0:0.001:10;
--> plot(t, csim('step', t, sys));
--> title(["Transfer Function", "$\dfrac{1}{s}$"], 'fontsize', 3);
--> xlabel("t", 'fontsize', 3); ylabel("u(t) response", 'fontsize', 3);
--> xs2png(gcf(), "Q4a.jpg");
```



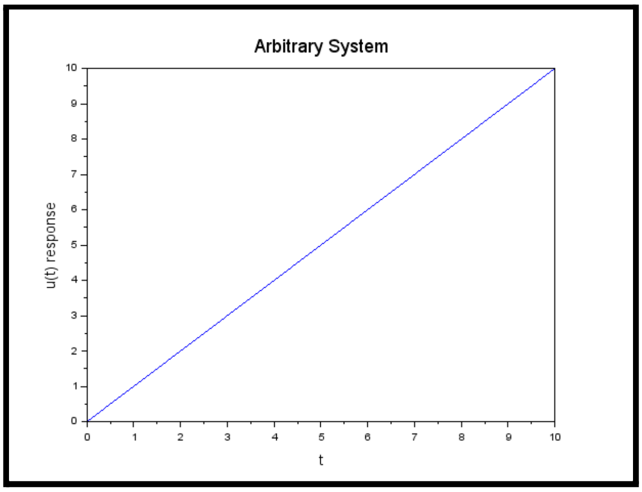
b. For a discrete time transfer function, the response of unit step would be as follows:

```
--> z = poly(0, 'z'); G = 1/z;
--> sys = tf2ss(syslin('d', G));
--> u = ones(1, 15); t = linspace(0, 15, 15);
--> y = dsimul(sys, u);
--> plot(t, y, 'r');
--> plot(t, y, 'o');
--> xtitle('unit step response', 't', 'y(t)');
--> var = gca(); var.data_bounds = [0,0 ; 15, 1.3];
--> xs2png(gcf(), "Q4b.jpg");
```



c. When we build a ratio of some arbitrary system, without defining its nature; SciLab assumes it to be continuous time only. The code for the same is as follows:

```
--> z=poly(0,'z'); G = 1/(z);
--> t = linspace(0,10,1000);
--> y = csim('step', t, G);
WARNING: csim: Input argument #1 is assumed continuous time.
--> plot(t, y);
--> title("Arbitrary System", 'fontsize', 4);
--> xlabel("t", 'fontsize', 3); ylabel("u(t) response", 'fontsize', 3);
--> xs2png(gcf(), "Q4c.jpg");
```



(The plot for this case will be the same as expected from part (a) where the system was continuous.)

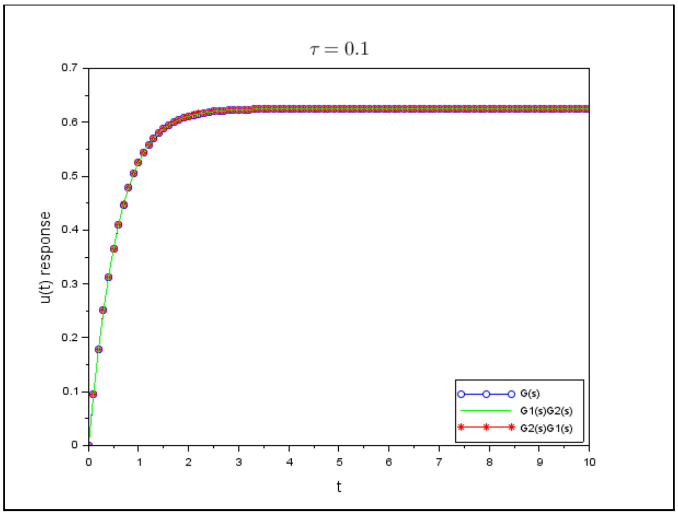
Question 5:

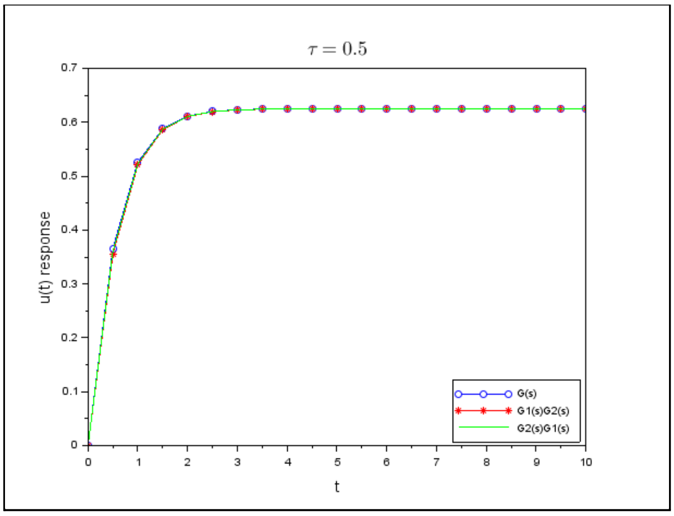
The required transfer function is **G(s) = (s + 5)/[(s + 4)(s + 2)]** is to be constructed in three ways and the response for each is to be plotted. Each of the cases is shown next:

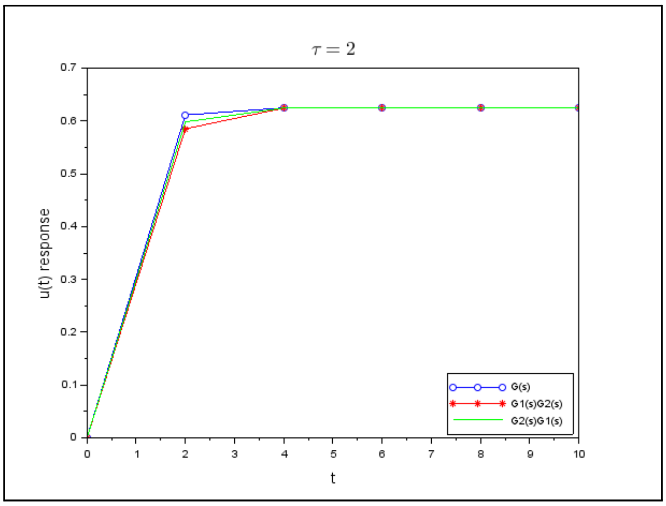
a. **G(s) = (s + 5)/[(s + 4)(s + 2)]** as a single block:
b. **G1(s) = (s+5)/(s+4)** and **G2(s) = 1/(s+2)** in series in that order
c. **G1(s) = 1/(s+2)** and **G2(s) = (s+5)/(s+4)** in series in that order

The combined code to compare all these cases at various resolutions of the input time sampling axis is as follows:

```
--> s = poly(0, 's');
--> t1 = 0:0.1:10; t2 = 0:0.5:10; t3 = 0:2:10;
--> G = (s+5)/((s+4)*(s+2)); G1 = (s+5)/(s+4); G2 = 1/(s+2);
--> sys = syslin('c', G); sys1 = syslin('c', G1); sys2 = syslin('c', G2);

--> y11 = csim('step', t1, sys);
--> y12_t = csim('step', t1, sys1); y_12 = csim(y12_t, t1, sys2);
--> y13_t = csim('step', t1, sys2); y_13 = csim(y13_t, t1, sys1);
--> scf();
--> plot(t1, y_12, 'b-o'); plot(t1, y11, 'r-*'); plot(t1, y_13, 'g');
--> legend(["G(s)", "G1(s)G2(s)", "G2(s)G1(s)"], 4);
--> title("$\tau = 0.1$", 'fontsize', 4);
--> xlabel("t", 'fontsize', 3); ylabel("u(t) response", 'fontsize', 3);
--> xs2png(gcf(), "Q5a.jpg");

--> y21 = csim('step', t2, sys);
--> y22_t = csim('step', t2, sys1); y22 = csim(y22_t, t2, sys2);
--> y23_t = csim('step', t2, sys2); y23 = csim(y23_t, t2, sys1);
--> scf();
--> plot(t2, y21, 'b-o'); plot(t2, y22, 'r-*'); plot(t2, y23, 'g');
--> legend(["G(s)", "G1(s)G2(s)", "G2(s)G1(s)"], 4);
--> title("$\tau = 0.5$", 'fontsize', 4);
--> xlabel("t", 'fontsize', 3); ylabel("u(t) response", 'fontsize', 3);
--> xs2png(gcf(), "Q5b.jpg");

--> y31 = csim('step', t3, sys);
--> y32_t = csim('step', t3, sys1); y32 = csim(y32_t, t3, sys2);
--> y33_t = csim('step', t3, sys2); y33 = csim(y33_t, t3, sys1);
--> scf();
--> plot(t3, y31, 'b-o'); plot(t3, y32, 'r-*'); plot(t3, y33, 'g');
--> legend(["G(s)", "G1(s)G2(s)", "G2(s)G1(s)"], 4);
--> title("$\tau = 2$", 'fontsize', 4);
--> xlabel("t", 'fontsize', 3); ylabel("u(t) response", 'fontsize', 3);
--> xs2png(gcf(), "Q5c.jpg");
```







With increasing sample times we observe discrepancies which are because of the way we study the system and not because of any inherent flaws with the system. The infrequent sampling will cause small differences to propagate through the different transfer function blocks, creating the illusion of error/dissimilarity. **Theoretically**, all three cases should and will have the same response.