Indian Institute of Technology Bombay

EE 324: Control Systems Lab

Problem Sheet 2 Report

January 23, 2021

Student Name	*Roll Number*
Mantri Krishna Sri Ipsit	180070032

Contents

1 Question 1

1.1 Part a

My roll number is 180070032 and my first name is "Mantri". So my $a = 32$ and $b = 13$. So a continuous time LTI system with transfer function

$$G(s) = \frac{a}{s+b} = \frac{32}{s+13}$$

is made as follows:

```
s = poly(0, 's');
a = 32;
b = 13;
G = a / (s + b);
S = syslin('c', G);
```

1.2 Part b

The plot for the above LTI system is generated and its properties like time constant, 2% settling time and rise time are found as follows:

```
t = 0:(1/130):5;
y = csim('step', t, S);
scf();
plot(t, y, 'k', 'LineWidth', 2);
set(gca(),"data_bounds",[-0.5,-0.1;5,2.6]);
time_constant = [1/b];
y1 = [y(11)];
plot(time_constant, y1, 'r*', 'LineWidth', 4);
settling_time_2p = time_constant .* log(50);
plot(settling_time_2p, y(1+(settling_time_2p .* 130)), 'b*', 'LineWidth', 4);
rise_time_low = time_constant .* log(10/9);
rise_time_high = time_constant .* log(10/1);
plot(rise_time_low, y(1+(rise_time_low .* 130)), 'm>', 'LineWidth', 4);
plot(rise_time_high, y(1+(rise_time_high .* 130)), 'm<', 'LineWidth', 4);
legend(['Step Response', 'Time Constant', '2% Settling Time', 'Rise Time Start', 'Rise Time End'], 4);
xlabel("time",'fontsize', 3);
ylabel("Step Response",'fontsize', 3);
title(["Transfer Function", "$\frac{a}{s+b}$"], 'fontsize', 3);
xs2png(gcf(), "Q1b.png");
```

Please turn over

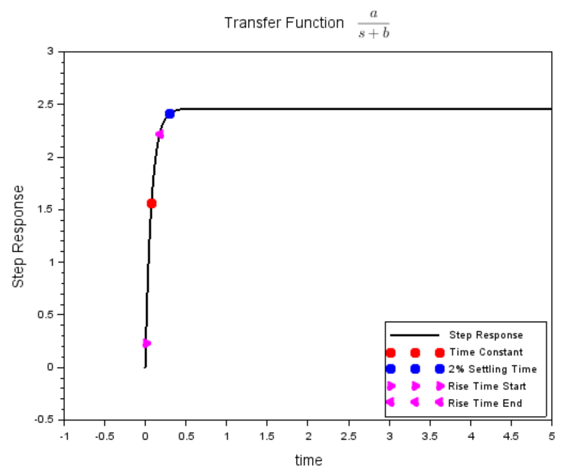


Figure 1: Step Response

1.3 Part c

The rise time for the above system is

$$T_r = \frac{\ln(9)}{b}$$

This is independent of a.

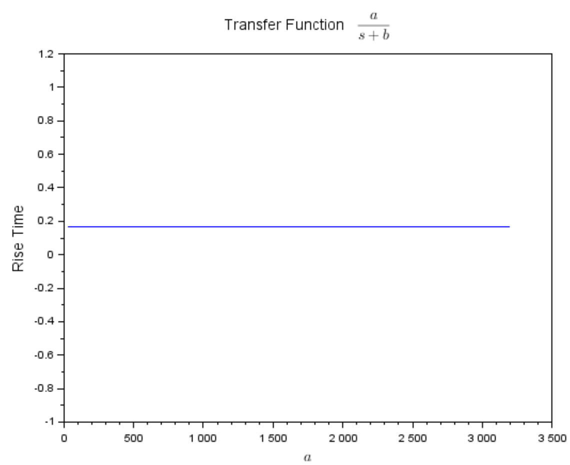


Figure 2: T_r vs a

```
A = a:a:100 * a;
rise_time = ones(A) .* (rise_time_high - rise_time_low)
scf();
plot(A, rise_time);
xlabel("$a$", 'fontsize', 3);
ylabel("Rise Time", 'fontsize', 3);
title(["Transfer Function", "$\frac{a}{s+b}$"], 'fontsize', 3);
xs2png(gcf(), "Q1c.png");
```

1.4 Part d

The rise time for the above system is

$$T_r = \frac{\ln(9)}{b}$$

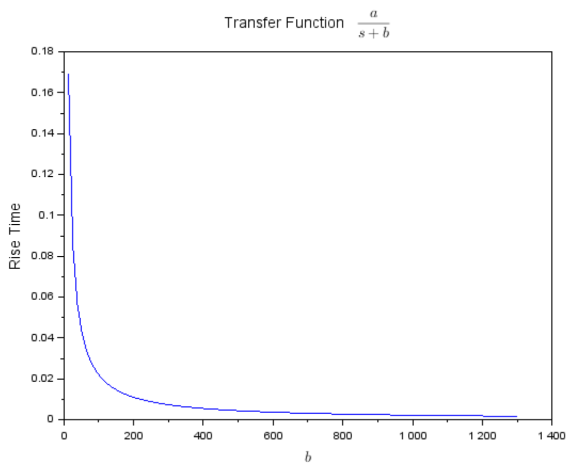


Figure 3: T_r vs a

```
B = b:b:100 * b;
rise_time = (1./B) .* log(9);
scf();
plot(B, rise_time);
xlabel("$b$", 'fontsize', 3);
ylabel("Rise Time", 'fontsize', 3);
title(["Transfer Function", "$\frac{a}{s+b}$"], 'fontsize', 3);
xs2png(gcf(), "Q1d.png");
```

2 Question 2

The example for a under-damped second order continuous time system with no zeros that I took is

$$G(s) = \frac{100}{s^2 + 5s + 100}$$

The damping ratio $\zeta = 0.25$ and the natural frequency $\omega_n = 10$ rad/s. The step response was plotted using the following comands:

```
G2 = 100 / (s^2 + 5*s + 100) // natural frequency = 10 rad/s, zeta = 0.25
S2 = syslin('c', G2);
t = 0:0.005:5
scf();
plot(t, csim('step', t, S2), 'LineWidth', 2);
xlabel("time",'fontsize', 3);
ylabel("Step Response",'fontsize', 3);
title(["Transfer Function", "$\frac{100}{s^2 + 5s + 100}$"], 'fontsize', 3)
xs2png(gcf(), "Q21.png");
```

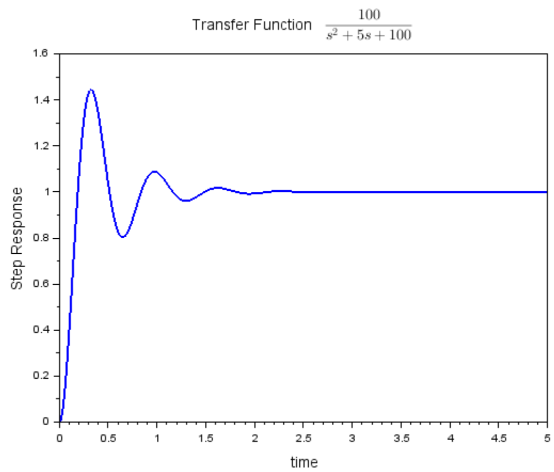


Figure 4: Step response

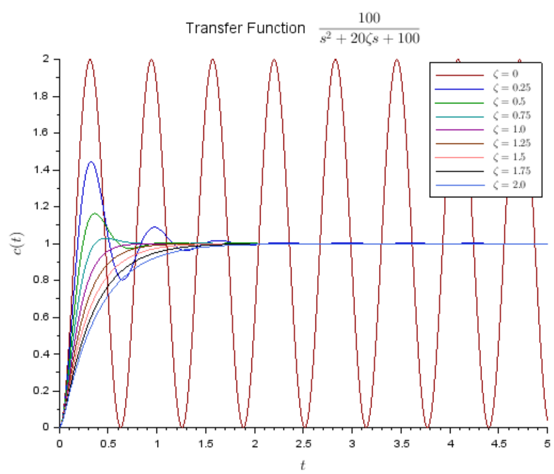


Figure 5: Step response

When the damping ratio is varied from 0 to 2 in the steps of 0.25, the step response changes as follows: The above plot was generated using the following code:

```
zeta = 0:0.25:2;
wn = 10;
scf();
plotcolors = ["scilabred4", "scilab blue2", "scilab green4", "scilab cyan4",...
"scilabmagenta4", "scilabbrown4", "scilabpink4", "black", "royalblue"];
for i=1:size(zeta, 2)
    G = wn^2 / (s^2 + 2 * zeta(i) * wn * s + wn^2);
    S = syslin('c', G);
    plot2d(t, csim('step', t, S), style=[color(plotcolors(i))]);
    xlabel("$t$", 'fontsize', 3);
ylabel("$c(t)$", '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", "$\frac{100}{s^2 + 20\zeta s + 100}$"], 'fontsize', 3);
xs2png(gcf(), "Q22.png");
```

As ζ increases,

- Rise time increases
- Settling time decreases
- %OS decreases
- Peak time increases

3 Question 3

The systems that I built are

- First Order:

$$G(s) = \frac{12.5}{s + 12.5}$$

- Second Order:

$$G(s) = \frac{100}{s^2 + 25s + 100}$$

- Second Order with Repeated Poles:

$$G(s) = \frac{100}{(s + 10)^2}$$

The step responses are as follows: The plot is generated using the following code:

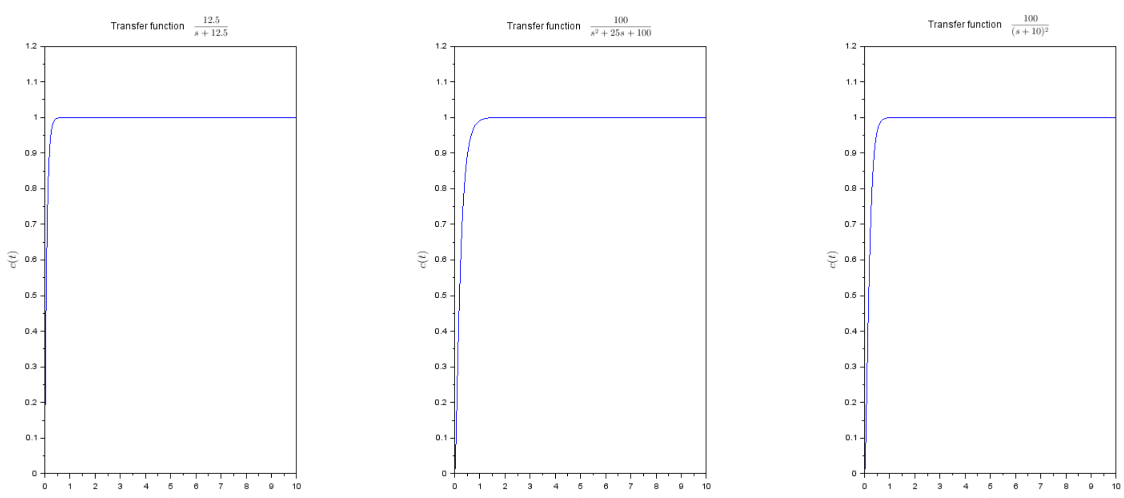


Figure 6: Step responses

```
first_order = 1 / (s+1);
first_order_system = syslin('c', first_order);
second_order = 100 / (s^2 + 25 * s + 100);
second_order_system = syslin('c', second_order);
t = 0:0.01:10;
```

```
scf();
subplot(131),plot(t, csim('step', t, first_order_system));
title(["Transfer function", "$\frac{1}{s+1}$"], 'fontsize', 2);
xlabel("$t$", 'fontsize', 3);
ylabel("$c(t)$", 'fontsize', 3);
subplot(132),plot(t, csim('step', t, second_order_system));
title(["Transfer function", "$\frac{100}{s^2+25s+100}$"], 'fontsize', 2);
xlabel("$t$", 'fontsize', 3);
ylabel("$c(t)$", 'fontsize', 3);
second_order_new = 100 / (s + 10)^2;
second_order_new_system = syslin('c', second_order_new);
subplot(133),plot(t, csim('step', t, second_order_new_system));
title(["Transfer function", "$\frac{100}{(s+10)^2}$"], 'fontsize', 2);
xlabel("$t$", 'fontsize', 3);
ylabel("$c(t)$", 'fontsize', 3);
xs2png(gcf(), "Q3.png");
```

Some salient points of differences:

- For the poles with same real parts, the first order system has a steeper rise than the second order system
- The slope at the origin is higher for the first order system than for the second order system
- The settling time is shorter for the first order system than for the second order system
- The rise time for the first order system is shorter than for the second order system
- The step response is monotonic even when there are repeated poles in the second order system

4 Question 4

4.1 Part a

The continuous time single-integrator has the transfer function

$$G(s) = \frac{1}{s}$$

The step response was plotted using the following commands:

```
gs = 1/s;
Gs = syslin('c', gs);
scf();
plot(t, csim('step', t, Gs));
title(["Transfer function", "$\frac{1}{s}$"], 'fontsize', 4);
xlabel("$t$", 'fontsize', 3);
ylabel("$c(t)$", 'fontsize', 3);
xs2png(gcf(), "Q4a.png");
```

Please turn over

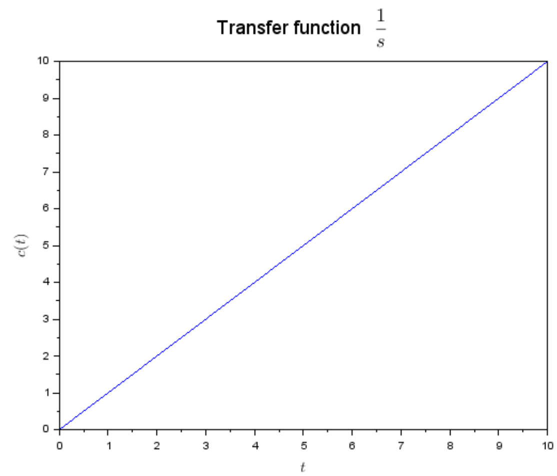


Figure 7: Step response

4.2 Part b

Here we have a discrete-time integrator with the transfer function

$$H(z) = \frac{1}{z}$$

The discrete-time step response for this system is generated using the following code:

```
z=poly(0,'z');
h=1/z;
sl=tf2ss(h);
u1=ones(1,10);
x=dsimul(sl,u1);  //Step response
scf();
plot(x);
set(gca(),"data_bounds",[0,0;10,2]);
xlabel("$n$", 'fontsize', 3);
ylabel("$y[n]$", 'fontsize', 3);
title(["Transfer Function", "$\frac{1}{z}$"], 'fontsize', 4);
xs2png(gcf(), "Q4b.png");
```

Please turn over

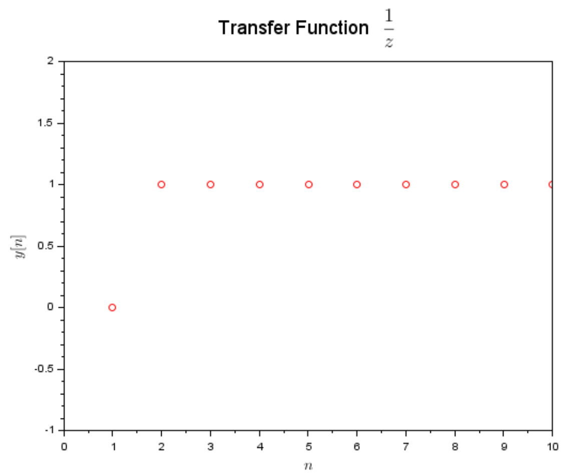


Figure 8: Step response

4.3 Part c

If we give the ratio of two polynomials as input to the `csim`, then Scilab gives the following warning:

```
WARNING: csim:  Input argument 1 is assumed continuous time.
```

If we compare the responses of 4a and 4b, we can see that it is different. This is because in the continuous time, the output is as follows:

$$C(s) = R(s) \times \frac{1}{s} = \frac{1}{s^2}$$
$$\implies c(t) = t\,u(t)$$

In the case of discrete-time, the output is as follows:

$$C(z) = R(z) \times \frac{1}{z} = \frac{1}{z^2 - z}$$

On solving it using partial fractions, we get

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

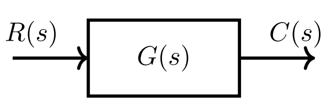
This is expected because a discrete-time system with transfer function z^{-1} introduces a 1 sample delay.

5 Question 5

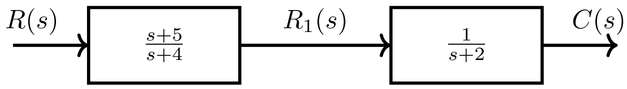
The transfer function that we have is

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

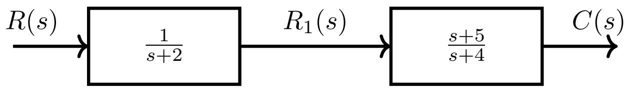
We have three configurations:
Direct:



Series 1:



Series 2:



On plotting the unit step responses for these three configurations by changing the sampling period, we get:

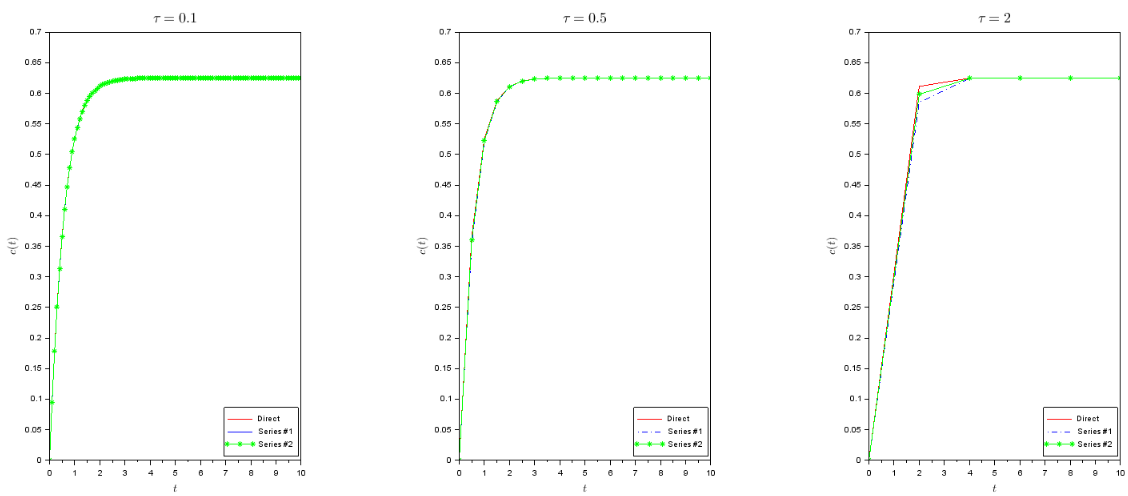


Figure 9: Step Responses

This was plotted using the following code:

```
t1 = 0:0.1:10; // tau = 0.1
t2 = 0:0.5:10; // tau = 0.5
t3 = 0:2:10;   // tau = 2
G1 = (s+5) / ((s+4)*(s+2));
S1 = syslin('c', G1);
G2 = (s+5)/(s+4);
S2 = syslin('c', G2);
G3 = 1/(s+2);
S3 = syslin('c', G3);
y1 = csim('step', t1, S1);
y21 = csim('step', t1, S2);
y22 = csim(y21, t1, S3);
y31 = csim('step', t1, S3);
y32 = csim(y31, t1, S2);
scf();
subplot(131),plot(t1, y1, 'r');
subplot(131),plot(t1, y22, 'b');
```

```
subplot(131),plot(t1, y32, 'g--*');
legend(["Direct", "Series #1", "Series #2"], 4);
title("$\tau = 0.1$", 'fontsize', 4);
xlabel("$t$", 'fontsize', 3);
ylabel("$c(t)$", 'fontsize', 3);
y1 = csim('step', t2, S1);
y21 = csim('step', t2, S2);
y22 = csim(y21, t2, S3);
y31 = csim('step', t2, S3);
y32 = csim(y31, t2, S2);
subplot(132),plot(t2, y1, 'r');
subplot(132),plot(t2, y22, 'b-.');
subplot(132),plot(t2, y32, 'g--*');
legend(["Direct", "Series #1", "Series #2"], 4);
title("$\tau = 0.5$", 'fontsize', 4);
xlabel("$t$", 'fontsize', 3);
ylabel("$c(t)$", 'fontsize', 3);
y1 = csim('step', t3, S1);
y21 = csim('step', t3, S2);
y22 = csim(y21, t3, S3);
y31 = csim('step', t3, S3);
y32 = csim(y31, t3, S2);
subplot(133),plot(t3, y1, 'r');
subplot(133),plot(t3, y22, 'b-.');
subplot(133),plot(t3, y32, 'g--*');
legend(["Direct", "Series #1", "Series #2"], 4);
title("$\tau = 2$", 'fontsize', 4);
xlabel("$t$", 'fontsize', 3);
ylabel("$c(t)$", 'fontsize', 3);
xs2png(gcf(), "Q5.png");
```

We can see that as the sampling period is increased, the plots tend to differ more. As the sampling period is increased, the Series 2 configuration has higher values than Series 1 and the direct implementation has the highest value at any time t.