EE324 Control Systems Lab

Problem Sheet 2

Sheel Shah — 19D070052

August 9, 2021

1 Question 1

1.1 Part A

Roll = 19D070052, Name = Sheel

$$\therefore a = 52, b = 19$$

$$\therefore G(s) = \frac{52}{s+19}$$

The following code is used to define this transfer function:

```
s = poly(0, 's')

g_temp = 52 / (s + 19)

g = syslin('c', g_temp)
```

1.2 Part B

The analysis is similar to the transfer function $\frac{1}{s+a}$, with $a = 19$.

$$\tau = \frac{1}{a} = 0.0526316$$

$$T_{rise} = \frac{ln(9)}{a} = 0.1156434$$

$$T_{rise,\ end} = \frac{ln(10)}{a} = 0.1211887$$

$$T_{rise,\ start} = T_{rise,\ end} - T_{rise} = 0.0055453$$

$$T_{settle} = \frac{ln(50)}{a} = 0.2058959$$

Code used:

```
t = 0 : 0.001: 1

step_resp  = csim("step", t, g)

plot(t, step_resp)


// finding parameters

a = 19

tau = 1 / a

y_tau = (52 / 19) * (1 - 1 / %e)

plot(tau, y_tau, "b.")


t_rise = log(9) / a

t_rise_end = log(10) / a

t_rise_start = t_rise_end - t_rise
```

```
y_t_rise_end = (52 / 19) * (1 - 1 / %e ^ log(10))
y_t_rise_start = (52 / 19) * (1 - 1 / %e ^ (log(10) - log(9)))
plot(t_rise_start, y_t_rise_start, "r>")
plot(t_rise_end, y_t_rise_end, "r<")

time_settle = log(50) / a
y_time_settle = (52 / 19) * (1 - 1 / %e ^ log(50))
plot(time_settle, y_time_settle, "b*")
legend(["Step Response", "Time Constant",
    "Rise Time Start", "Rise Time End", "Settle Time"])
// Go to File > Export To to save the image as png.
```

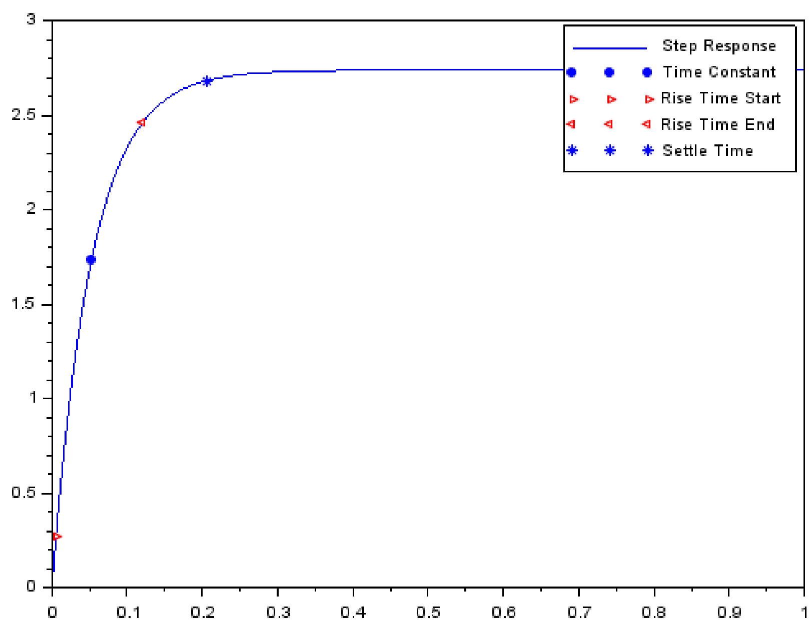

Figure 1: Unit step response of G

1.3 Part C

The following code was used:

```
a_vals = linspace(52, 5200, 100)

// rise time is independent of a

rise_time_variation = linspace(0.1156434, 0.1156434, 100)

plot(a_vals, rise_time_variation)
```

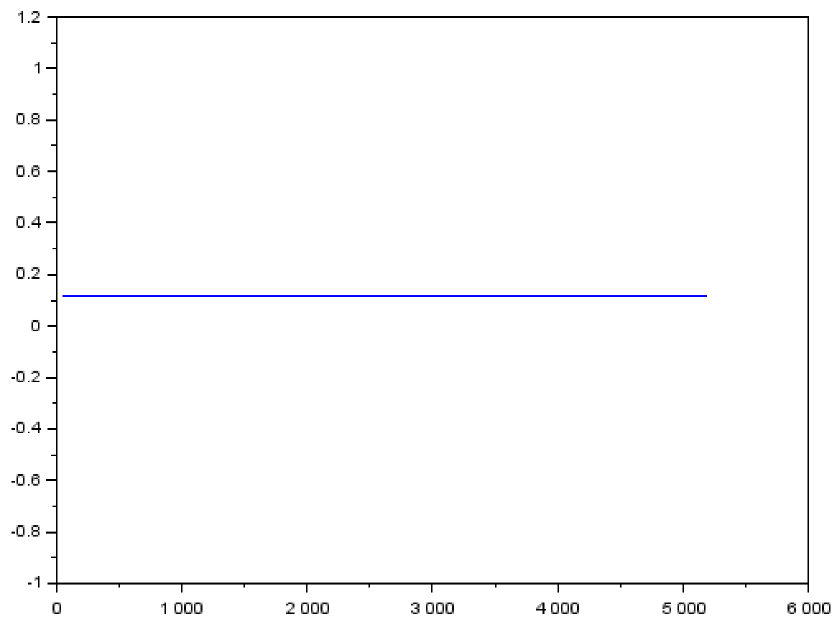

Figure 2: Variation of Rise Time with a

1.4 Part D

The following code was used:

```
b_vals = linspace(19, 1900, 100)
// rise time formula wrt b
rise_time_variation = log(9) ./ b_vals
plot(b_vals, rise_time_variation)
```

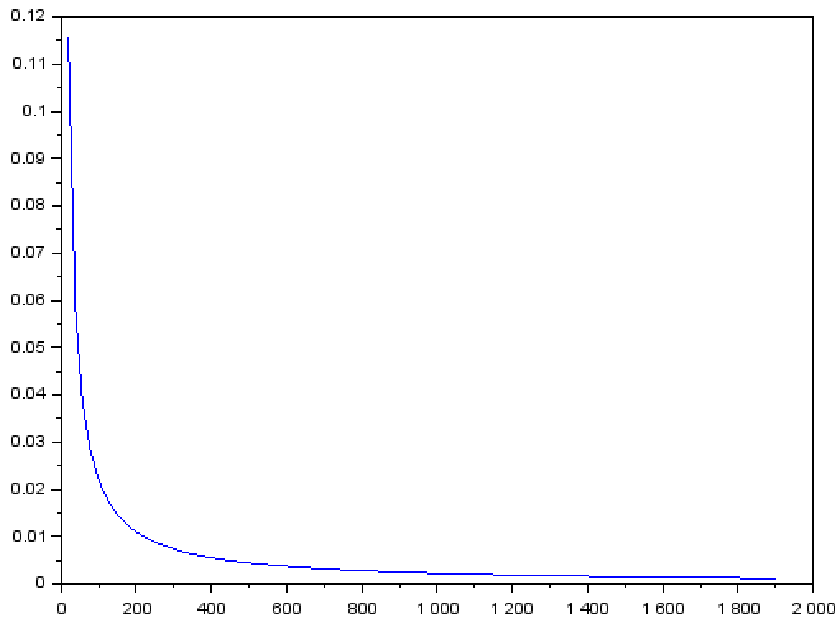

Figure 3: Variation of Rise Time with b

2 Question 2

We use the following transfer function:

$$G(s) = \frac{1}{s^2 + s + 1},\ \zeta = 0.5$$

Code used:

```
s = poly(0,'s')

g_temp = 1 / (s^2 + s + 1)

g = syslin('c', g_temp);

t = 0:0.01:20

step_resp = csim("step", t, g)

plot(t, step_resp)
```

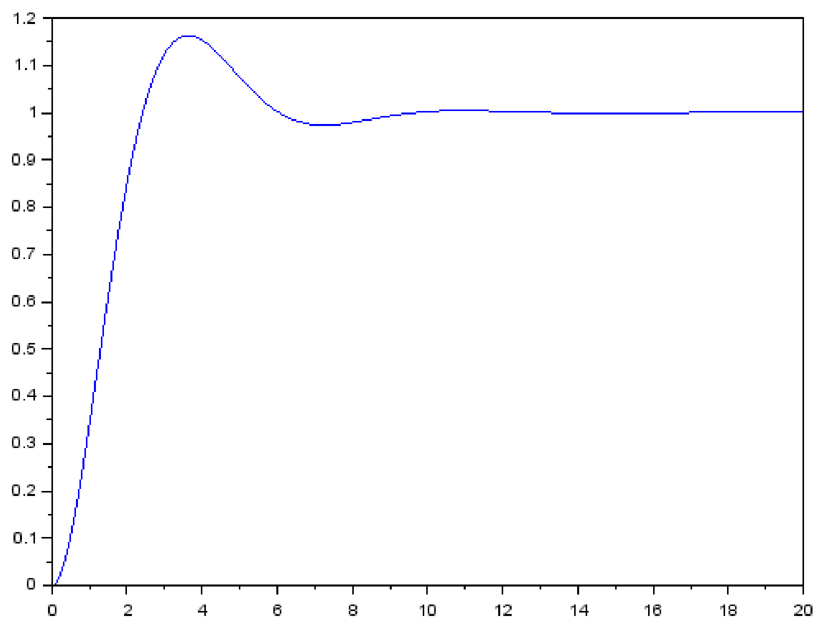

Figure 4: Step response of under-damped G

Code used to vary ζ:

```
for zeta = linspace(0, 2, 9)
    g_temp = 1 / (s^2 + 2*zeta*s + 1)
    g = syslin('c', g_temp)
    step_resp = csim("step", t, g)
    plot2d(t, step_resp)
end
```

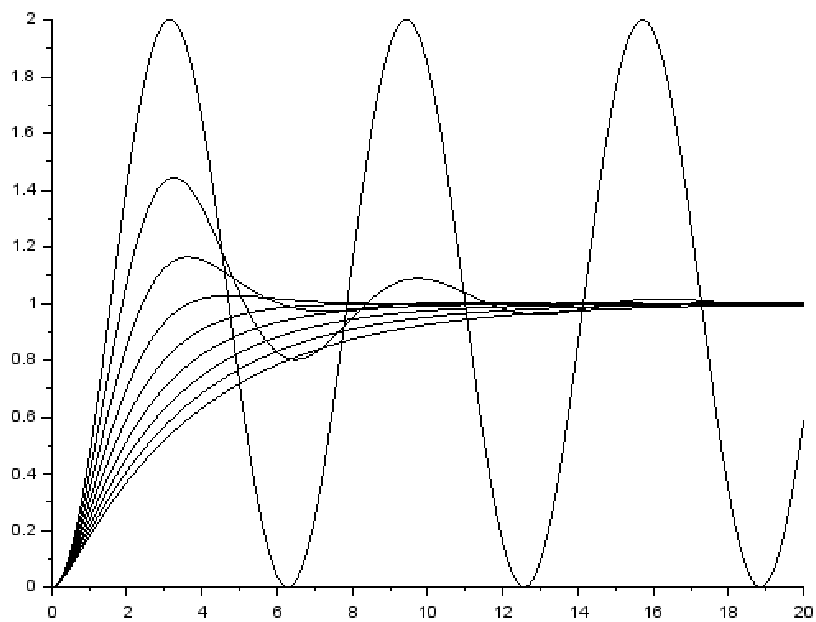

Figure 5: Variation in step response

3 Question 3

Transfer functions used:

$$G1(s) = \frac{1}{s+1},\ G2(s) = \frac{4}{s^2+4.1s+4},\ G3(s) = \frac{4}{s^2+4s+4}$$

Code Used:

```
s = poly(0, 's')

t = 0 : 0.01 : 10

g_temp1 = 1 / (s + 1)

g1 = syslin('c', g_temp1)

g_temp2 = 4 / (s^2 + 4.1*s + 4)

g2 = syslin('c', g_temp2)

g_temp3 = 4 / (s^2 + 4*s + 4)

g3 = syslin('c', g_temp3)


step_resp1 = csim("step", t, g1)

step_resp2 = csim("step", t, g2)

step_resp3 = csim("step", t, g3)


plot(t, step_resp1, "b")

plot(t, step_resp2, "r")

plot(t, step_resp3, "g")
```

I saw the following two differences between the first order and second order systems:

1. Second order systems rise and settle faster

2. Second order systems have a 0 gradient at the origin

3. Repeated poles do result in a monotonic response

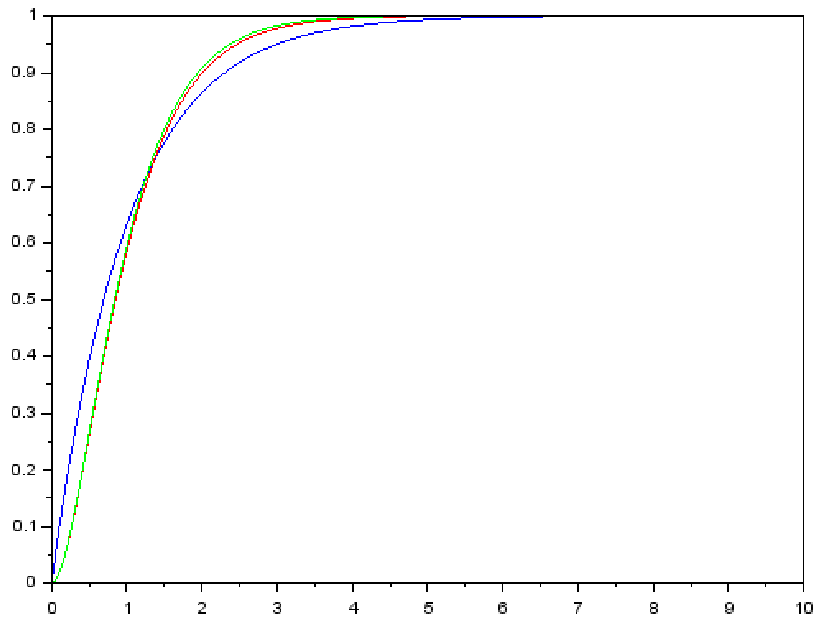

Figure 6: Comparision of responses

4 Question 4

4.1 Part A

Code used:

```
s = poly(0, 's')

g_temp = 1 / s

g = syslin('c', g_temp)

t = 0 : 0.01 : 5

step_resp = csim("step", t, g)

plot(t, step_resp)
```

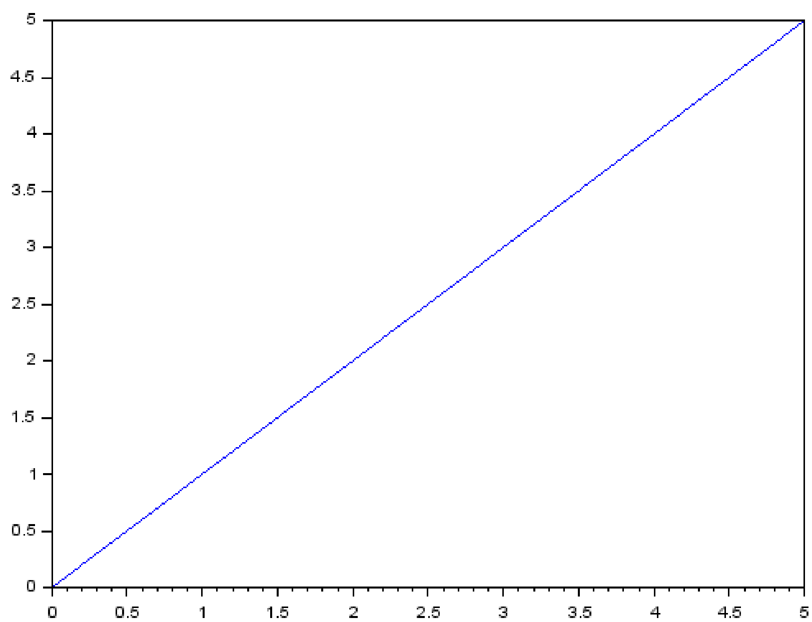

Figure 7: Continuous time response

4.2 Part B

Code used:

```
s = poly(0, 's')
g_temp = 1 / s
g = tf2ss(g_temp)
t = ones(1, 10)
step_resp = dsimul(g, t)
scatter(linspace(1, 10, 10), step_resp)
```

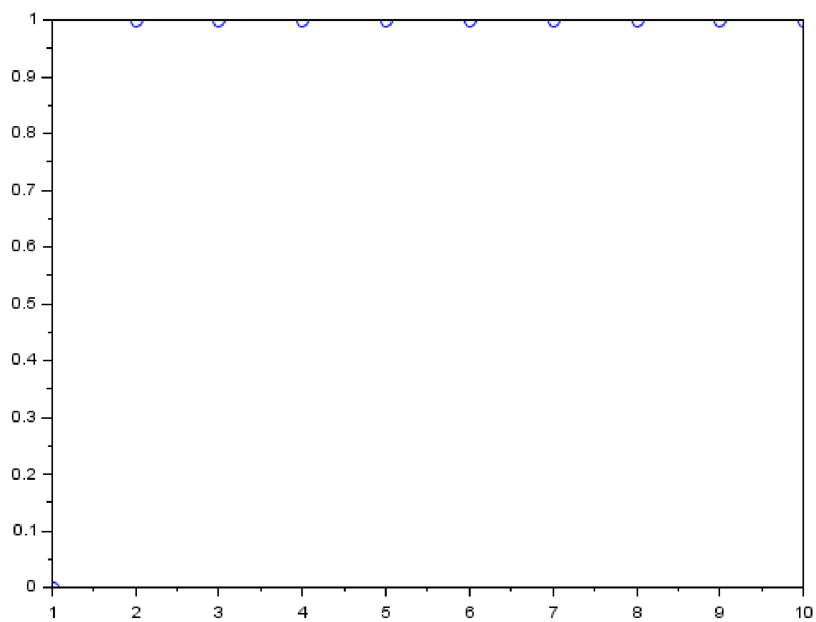

Figure 8: Discrete time response

4.3 Part C

The ratio when given as input to ‘csim‘ throws an error.

The difference in plots is due to the fact that the discrete time system adds a 1 sample delay.

Continuous Time: $y(t) = tu(t)$

Discrete Time: $y[n] = u[n-1]$