EE324 Control Systems Lab

Problem Sheet 2

Param Rathour | 190070049

Autumn Semester 2021-22

Contents

1 Q1

$$a = 49$$

$$b = 16$$

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

(a) Initialising code to build a continuous time LTI system with transfer function $G(s)$

```
--> s = %s;                              // define the symbolic variable 's'
--> a = 49; b = 16;
--> sys = syslin('c' , a/(s+b));
 sys  =
      49
    -----
    16 + s
```

(b) Unit step response to the system with the tranfer function is given by $G(s)$ in the figure 1.1

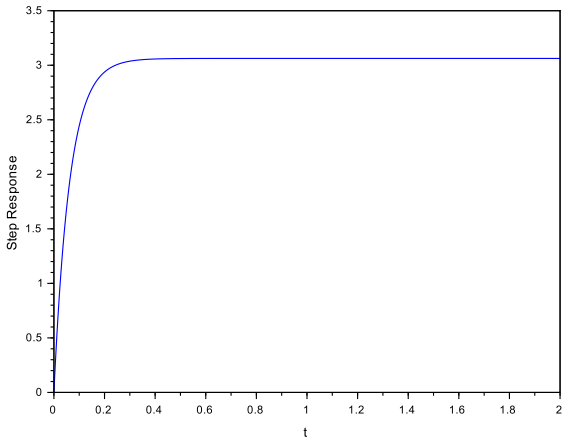


Figure 1.1: Unit Step response to the system with TF $G(s)$

Code,

```
--> step = 0.0001;
--> t = 0:step:2;
--> gp = csim('step', t, sys);
--> plot(t,gp); xlabel("t"); ylabel("Step Response");
--> xs2pdf(0,'Q1b');
```

To find time constant, 2% settling time, rise time from the plot

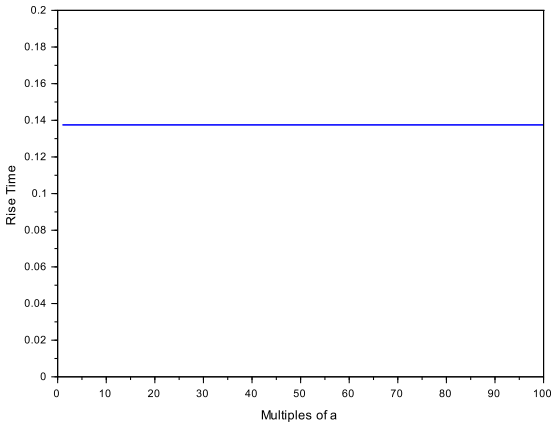
```
--> finalValue = gp($);

--> Tc = 0; Ts = 0; Tr = 0;
--> for x = 1:tmax/step
-->     if gp(x) > 0.9*finalValue
-->         Tr = Tr + x;
-->         break
-->     end
-->     if gp(x) < 0.63*finalValue
-->         Tc = x;
-->     end
-->     if gp(x) < 0.1*finalValue
-->         Tr = -x;
-->     end
--> end

--> Tc = Tc * step;
--> Tr = Tr * step;

--> for x = tmax/step:-1:1
-->     if abs(gp(x)-finalValue) > 0.02*finalValue
-->         Ts = x * step;
-->         break
-->     end
--> end
--> disp(Tc,Ts,Tr)
 Tc  = 0.0621500
 Ts  = 0.2445200
 Tr  = 0.1373400
```

Therefore,

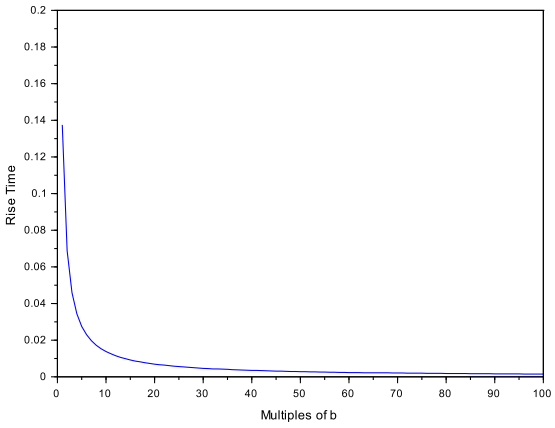
$$\text{Time Constant} = 0.0621500 \quad 2\% \text{ Settling Time} = 0.2445200 \quad \text{Rise Time} = 0.1373400$$

(c) To calculate Rise Time (using plots) I made a function

```
--> function Tr = riseTime(gp, step, tmax)
-->     finalValue = gp($);
-->     for x = 1:tmax/step
-->         if gp(x) > 0.9*finalValue
-->             Tr = Tr + x;
-->             break
-->         end
-->         if gp(x) < 0.1*finalValue
-->             Tr = -x;
-->         end
-->     end
-->     Tr = Tr * step
--> endfunction
```



(a) Variation of Rise Time of $G(s)$ in a



(b) Variation of Rise Time of $G(s)$ in b

Now, to obtain this variation of rise time in a

```
--> aTimes = [];
--> for x = 1:100
-->     systemp = syslin('c' , a*x/(s+b));
-->     gptemp = csim('step', t, systemp);
-->     aTimes($+1) = riseTime(gptemp, step, tmax);
--> end

--> aV = [1:100]
--> plot(aV',aTimes); xlabel("Multiples of a"); ylabel("Rise Time");
--> ax = gca();
--> ax.data_bounds=[1 0;100 0.2];
--> xs2pdf(0,'Q1c');
```

(d) Similarly, to obtain variation of rise time in b

```
--> bTimes = [];
--> for x = 1:100
-->     systemp = syslin('c' , a/(s+b*x));
-->     gptemp = csim('step', t, systemp);
-->     bTimes($+1) = riseTime(gptemp, step, tmax);
--> end

--> bV = [1:100]
--> plot(bV',bTimes); xlabel("Multiples of b"); ylabel("Rise Time");
--> ax = gca();
--> ax.data_bounds=[1 0;100 0.2];
--> xs2pdf(0,'Q1d');
```

For a)

Rise Time is consistent with the formula $T_S = \frac{\ln 9}{b} \approx 0.1373$
Similarly, 2% Settling Time is consistent with the formula $T_S = \frac{\ln 50}{b} \approx 0.2445$ and
Time Constant is consistent with the formula $T_S = \frac{1}{b} = 0.625$

For b) The Rise Time plots are also consistent with the formula
This shows the independence of T_R with a and inverse relation with b

2 Q2

Let's take our underdamped second order continuous time ($0 < \zeta < 1$) system with no zeros with the transfer function as

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{1}{s^2 + 2\zeta s + 1} \quad \text{by taking } \omega_n = 1$$

Initialising code to build a continuous time LTI system with transfer function $G(s)$ with damping ratio $(\zeta) = 0.5$

```
--> s = %s;                              // define the symbolic variable 's'
--> z = 1/2;
--> sys = syslin('c' , 1/(s*s+2*z*s+1));
 sys  =
      1
   --------
   1 + s + s²
```

Step response to the system with the tranfer function is given by $G(s)$ in the figure 2.1

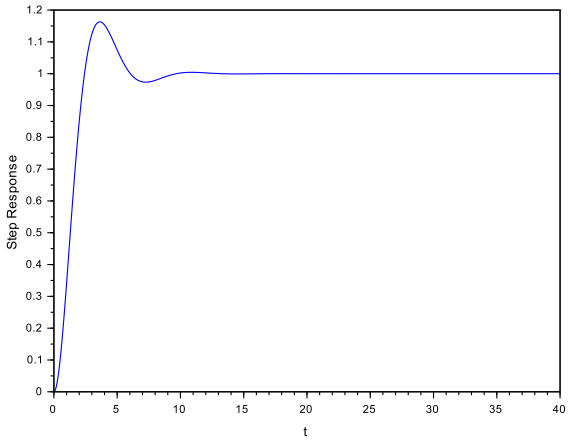


Figure 2.1: Unit Step response to the system with TF $G(s)$

```
--> step = 0.001;
--> t = 0:step:40;
--> gp = csim('step', t, sys);
--> plot(t,gp); xlabel("t"); ylabel("Step Response");
--> xs2pdf(0,'Q2a');
```

The step responses of the standard second order system with variation in Damping Ratio

```
--> t = 0:step:30;
--> z = 0:0.25:2
--> y = [];
--> dims = 1;
--> for x = 1:9
-->     zeta = z(x);
-->     sys = syslin('c' , 1/(s*s+2*zeta*s+1));
-->     y= cat(dims, y, csim('step', t, sys));
--> end
--> plot(t',y'); xlabel("t"); ylabel("Step Response")
--> legend(['$\zeta = 0$';'$\zeta = 0.25$';'$\zeta = 0.5$';'$\zeta = 0.75$';'$\zeta = 1$';
            '$\zeta = 1.25$';'$\zeta = 1.5$';'$\zeta = 1.75$';'$\zeta = 2$']);
--> xs2pdf(0,'Q2b');
```

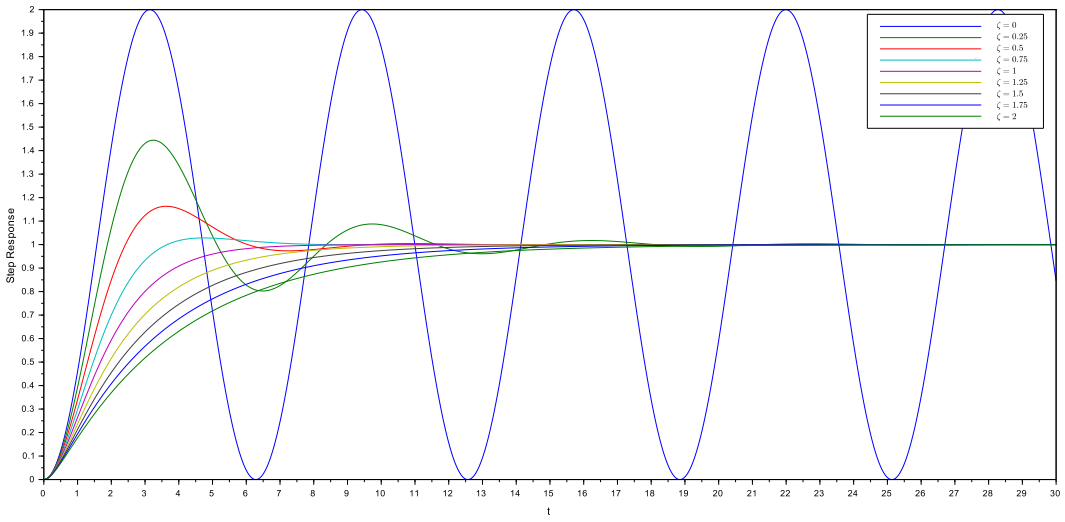


Figure 2.2: Step response to the system with TF $G(s)$ when ζ varies from 0 to 2 in steps of 0.25

With increase in Damping Ratio ζ (assuming same ω_n)

- % Overshoot value decreases from final value at $\zeta = 0$ to 0 at $\zeta = 1$ then there is no % Overshoot
- Rise Time increases as ζ increases
- Peak Time increases as ζ increases and it is only defined for undamped & underdamped systems
- 2 % Settling remains decreases from ∞ at $\zeta = 0$ to minimum at $\zeta = 1$ then increases at higher ζ

3 Q3

Plots of monotonically increasing continuous time systems with tf $G_1(s)$ (1^{st} order), $G_2(s)$ (2^{nd} order)

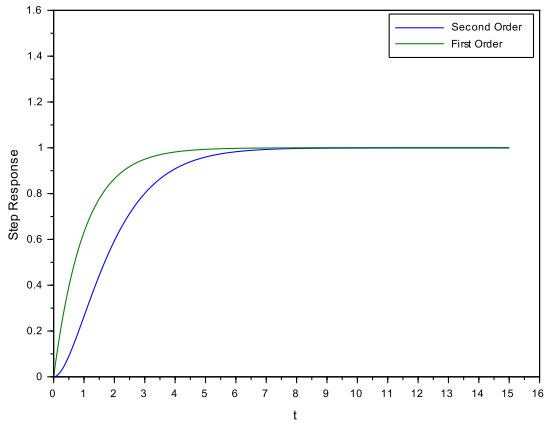
```
--> s = %s;                              // define the symbolic variable 's'
--> a = 1;
--> y = [];
--> dims = 1;
--> t = 0:step:15;
--> sys2 = syslin('c' , a/(s^2 + 2*s +a));
--> y = cat(dims, y, csim('step', t, sys2));
--> sys1 = syslin('c' , a/(s+a));
--> y = cat(dims, y, csim('step', t, sys1));
--> plot(t',y'); xlabel("t"); ylabel("Step Response")
--> legend(['Second Order';'First Order']);
--> ax = gca();
--> ax.data_bounds=[0 0;15 1.5];
--> xs2pdf(0,'Q3a');
```

Salient points of differences between these two responses

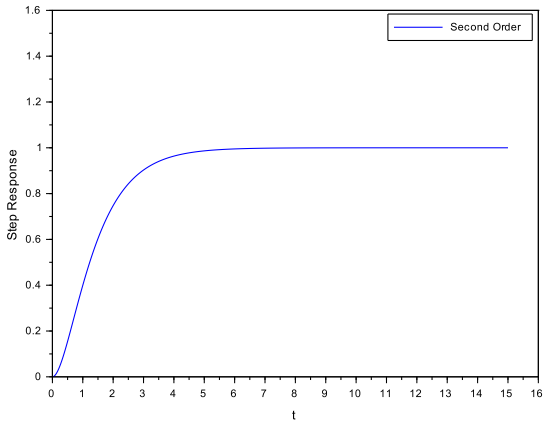
- The second order step responses can oscillate, a first order system can't oscillate as there is just one exponential
- Slope at $t = 0$ is zero for second order systems, whereas it's large for first order systems

When we have repeated poles in the second order system the step response is still monotonic

This is because if a pole is repeated then it has to be real (complex poles are conjugate (tf coefficients are real))



(a) Monotonically increasing systems



(b) A second order system with Repeated poles

Figure 3.1: Comparison of step responses between 1st order & 2nd order systems

```
--> t = 0:step:15;
--> sys = syslin('c' , 1/(s^2 + 2*s +1));
--> gp = csim('step', t, sys2);
--> plot(t',gp); xlabel("t"); ylabel("Step Response")
--> legend(['First Order';'Second Order']);
--> ax = gca();
--> ax.data_bounds=[0 0;15 1.5];
--> xs2pdf(0,'Q3b');
```

4 Q4

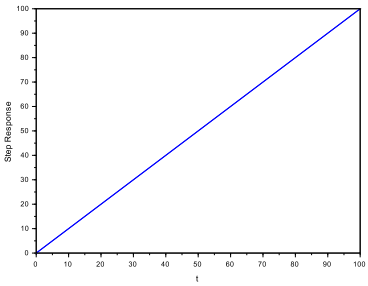
a)

$$G_1(s) = \frac{1}{s}$$

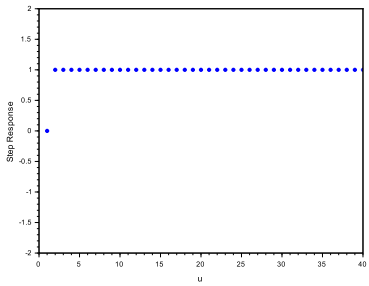
Initialising code to build a continuous time LTI system with transfer function $G_1(s)$

```
--> s = %s;                          // define the symbolic variable 's'
--> sys = syslin('c' , 1/s);
```

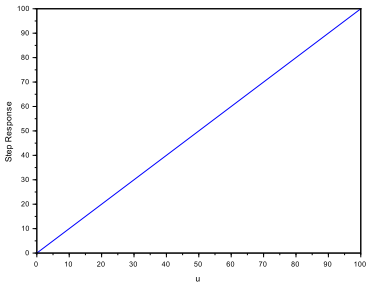
Unit step response to the system with the tranfer function is given by $G_1(s)$ in the figure 4.1a



(a) Continuous $G_1(s)$



(b) Discrete $G_2(s)$ by `dsimul`



(c) Discrete $G_2(s)$ by `csim`

Figure 4.1: Unit Step response to the system

```
--> step = 0.001;
--> t = 0:step:100;
--> gp = csim('step', t, sys);
--> plot(t,gp); xlabel("t"); ylabel("Step Response");
--> xs2pdf(0,'Q4a');
```

b)

$$G_2(z) = \frac{1}{z}$$

Initialising code to build a discrete time system with transfer function $G_2(s)$

```
--> z = %z;                              // define the symbolic variable 'z'
--> h = 1/z;
--> sys = tf2ss(h);
```

Unit step response to the system with the tranfer function is given by $G_2(s)$ in the figure 4.1b

```
--> u = ones(1,40);
--> gdp = dsimul(sys, u);
--> plot(gdp,'.'); xlabel("u"); ylabel("Step Response");
--> ax = gca();
--> ax.data_bounds=[0 -2;40 2];
--> xs2pdf(0,'Q4b');
```

The response 4.1b is constant(=1) for $u \geq 1$ unlike 4.1a and 0 for $u = 0$

c)

Initialising code to build a continuous time system with transfer function $G_2(s)$

```
--> z = %z;                              // define the symbolic variable 'z'
--> h = 1/z;
--> sys = tf2ss(h);
```

Unit step response to the system with the tranfer function is given by $G_2(s)$ in the figure 4.1c

```
--> step = 0.001;
--> t = 0:step:100;
--> gp = csim('step', t, sys);
--> plot(t,gp); xlabel("u"); ylabel("Step Response");
--> xs2pdf(0,'Q4c');
```

The plot is different from 4.1b case as `csim` assumes continuous time linear system

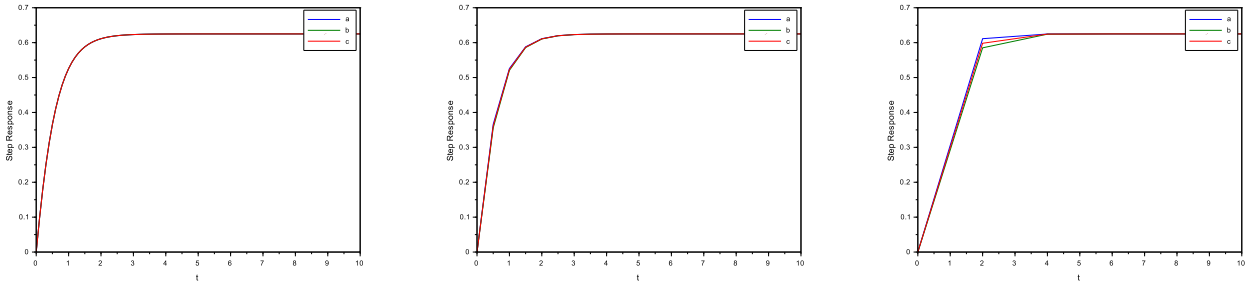
5 Q5

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

Initialising code to build a continuous time LTI system with transfer function $G(s)$

```
--> s = %s;                                          // define the symbolic variable 's'
--> sys1 = syslin('c' , (s+5)/((s+2)(s+4)));
```

Unit step response to the system with the tranfer function is given by $G(s)$ in the figure 5.1 This function plots the



(a) $\tau = 0.1$ (b) $\tau = 0.5$ (c) $\tau = 2$

Figure 5.1: Unit Step response to the system $G(s)$

required response with different τ

```
function y = plots(sys1,sys2,sys3,tau)
    t = [0:tau:10];
    y = cat(dims, y, csim('step', t, sys1));

    temp1 = csim('step', t, sys2);
    y = cat(dims, y, csim(temp1, t, sys3));

    temp2 = csim('step', t, sys3);
    y = cat(dims, y, csim(temp2, t, sys2));

    plot(t',y'); xlabel("t"); ylabel("Step Response")
    legend(['a';'b';'c']);
    filename = "Q5" + string(int(10*tau))
    xs2pdf(0,filename);
endfunction
```

Now the code,

```
s = %s;                              // define the symbolic variable 's'
a = 1;
y = [];
dims = 1;
G1 = (s+5)/((s+2)*(s+4))
G2 = (s+5)/(s+4)
G3 = 1/(s+2)
sys1 = syslin('c' , G1);
sys2 = syslin('c' , G2);
sys3 = syslin('c' , G3);

plots(sys1,sys2,sys3,0.1);
plots(sys1,sys2,sys3,0.5);
plots(sys1,sys2,sys3,2);
```