EE324 Lab-1 Report

Yash Sanjeev
180070068

January 2021

Contents

1 Problem 1

1.1 Code

```
a = 68; // Roll Number = 180070068
b = 25; // Name = Yash (Y = 25)

// build the LTI System
s = poly(0, 's');
G = a/(s+b);
sys = syslin('c', G);

// plot the unit step response
time = 0:0.00001:1;
out = csim("step", time, sys);

tau = 1/b; // time constant
settle = log(1/0.02)/b; // 2% settling time
start = log(1/(1-0.10))/b; // rising starts at 10%
fin = log(1/(1-0.90))/b; // rising finishes at 90%
rise = fin - start; // overall rise time

disp("Time constant =", tau);
disp("2% Settling time =", settle);
disp("Rise time", rise);

scf(0);
plot(time, out);
xlabel("time (in secs)");
ylabel("system response");
title("Step Response");

// variation in a
a_range = a:a:100*a;
a_rise = linspace(rise, rise, 100);

scf(1);
plot(a_range, a_rise);
xlabel("a");
ylabel("Rise Time (in secs)");
title("Variation of Rise Time with a");

// variation in b
b_range = b:b:100*b;
b_start = log(1/(1-0.10))./b_range;
b_fin = log(1/(1-0.90))./b_range;
```

```
b_rise = b_fin - b_start;

scf(2);
plot(b_range, b_rise);
xlabel("b");
ylabel("Rise Time (in secs)");
title("Variation of Rise Time with b");
```

1.2 Part 1(a)

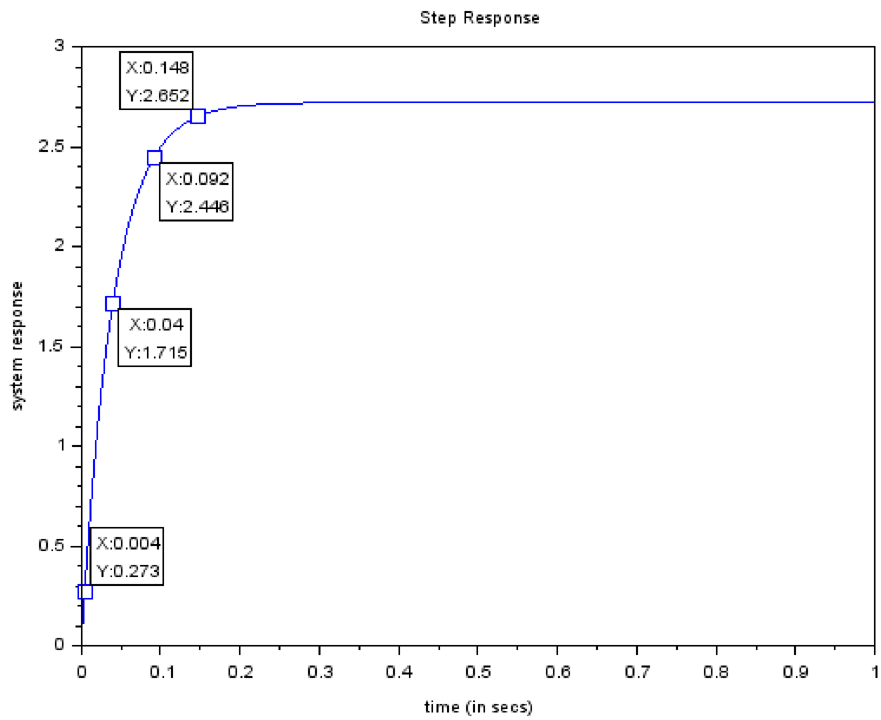
```
a = 68; // Roll Number = 180070068
b = 25; // Name = Yash (Y = 25)

// build the LTI System
s = poly (0 , 's') ;
G = a /( s + b ) ;
sys = syslin ('c', G ) ;
```

- Roll Number = 180070068, hence $a = 68$.
- First Name = **Y**ash, hence $b = 25$.
- According to these values, a continous time LTI has been built.

1.3 Part 1(b)

```
// plot the unit step response
time = 0:0.00001:1;
out = csim("step", time, sys);

tau = 1/b; // time constant
settle = log(1/0.02)/b; // 2% settling time
start = log(1/(1-0.10))/b; // rising starts at 10%
fin = log(1/(1-0.90))/b; // rising finishes at 90%
rise = fin - start; // overall rise time

disp("Time constant =", tau);
disp("2% Settling time =", settle);
disp("Rise time", rise);

scf(0);
plot(time, out);
xlabel("time (in secs)");
ylabel("system response");
title("Step Response");
```

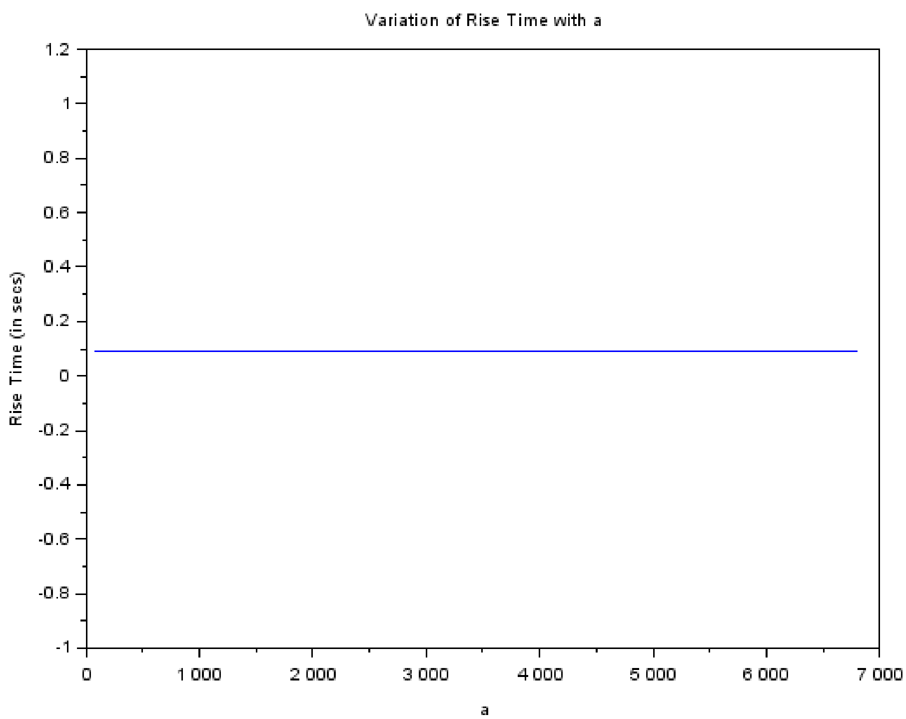
- Rising time taken between 10% and 90% of steady-state value.
- Response $= \frac{a}{b}(1 - e^{-bt})$, hence steady state value is 2.72.
- Rise starts when $y = 0.272$, and stops at $y = 2.448$, for 2% settling time, $y = 0.98 \times 2.72 = 2.665$ and for τ, we need $x = \frac{1}{25} = 0.04$.



1.4 Part 1(c)

```
// variation in a
a_range = a:a:100*a;
a_rise = linspace(rise, rise, 100);

scf(1);
plot(a_range, a_rise);
xlabel("a");
ylabel("Rise Time (in secs)");
title("Variation of Rise Time with a");
```

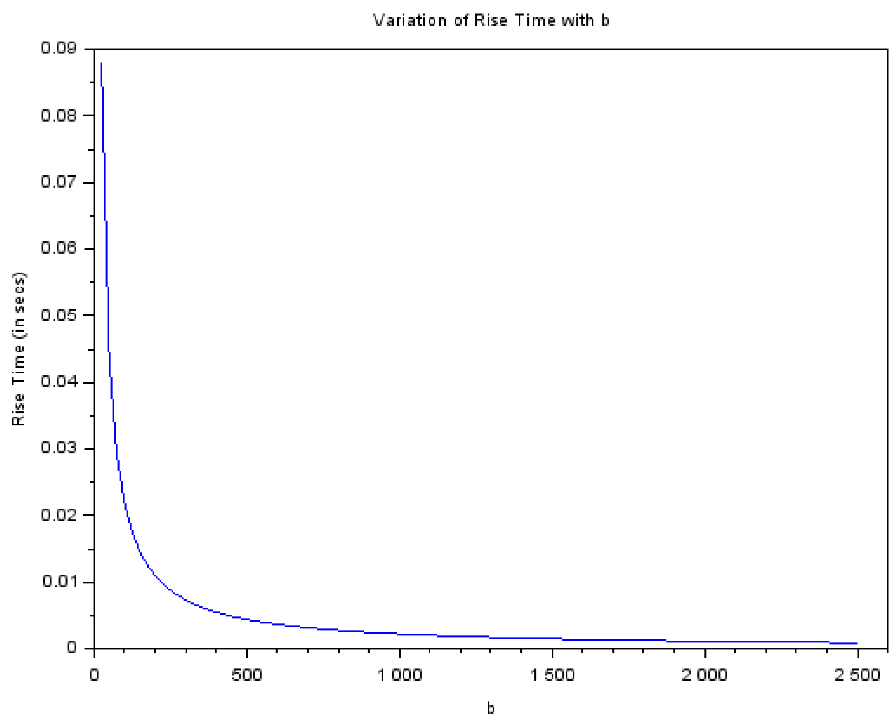
- Since values only depend on b, variation in a produces no effect in rise time.



1.5 Part 1(d)

```
// variation in b
b_range = b:b:100*b;
b_start = log(1/(1-0.10))./b_range;
b_fin = log(1/(1-0.90))./b_range;
b_rise = b_fin - b_start;

scf(2);
plot(b_range, b_rise);
xlabel("b");
ylabel("Rise Time (in secs)");
title("Variation of Rise Time with b");
```

- Rise time $\propto \frac{1}{b}$, the graph drawn is a rectangular hyperbola.


Variation of Rise Time with b
Rise Time (in secs)
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
500
1 000
1 500
2 000
2 500
b

2 Problem 2

2.1 Code

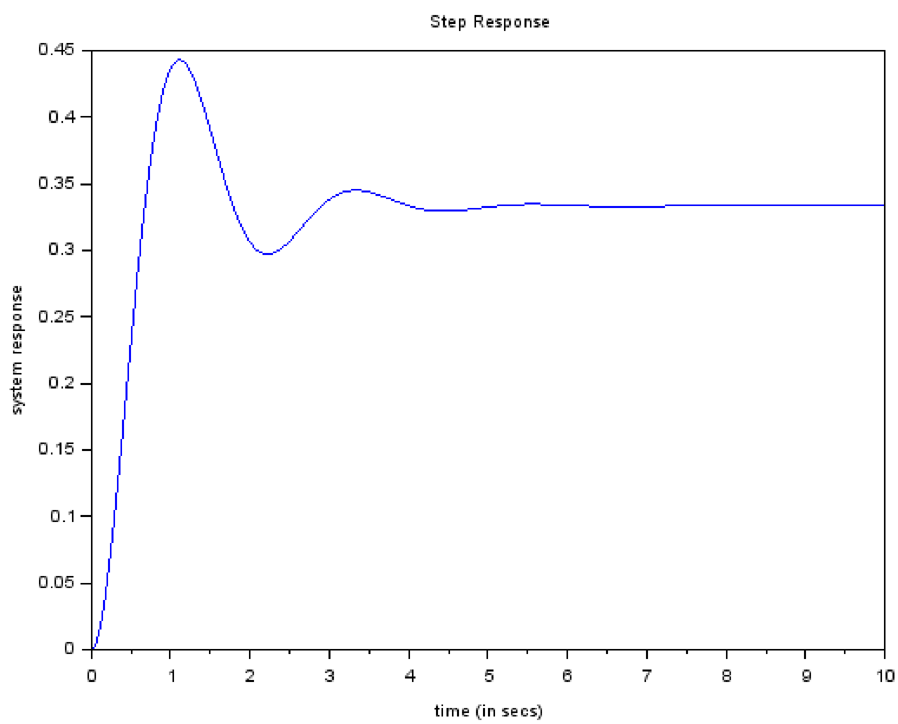
```
s = poly(0, 's');
dr = 1/3; // damping ratio
f = 3; // frequency of oscillation

G = f/(s^2 + 2*dr*f*s + f^2);
sys = syslin('c', G);

// plotting the step response
t = 0:0.002:10;
out = csim("step", t, sys);

scf(0);
plot(t, out);
xlabel("time (in secs)");
ylabel("system response");
title("Step Response");

// variation with damping ratio
dr_range = 0:0.25:2;
out_range = zeros(5001, 9);

for i = 1:size(dr_range)(2)
    dr = dr_range(i);
    G = f/(s^2 + 2*dr*f*s + f^2);
    sys = syslin('c', G);
    out_range(:, i) = csim("step", t, sys);
end

scf(1);
plot2d(t, out_range, 1:9);
legends(['0'; '0.25'; '0.5'; '0.75'; '1'; '1.25'; '1.5'; '1.75'; '2'], 1:9, opt = "?");
xlabel('Time (in secs)');
ylabel('Step Response');
title('Variation of step response with damping ratio');
```

2.2 Part 2(a)

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

```
dr = 1/3; // damping ratio
f = 3; // frequency of oscillation

G = f/(s^2 + 2*dr*f*s + f^2);
sys = syslin('c', G);

// plotting the step response
t = 0:0.002:10;
out = csim("step", t, sys);

scf(0);
plot(t, out);
xlabel("time (in secs)");
ylabel("system response");
title("Step Response");
```

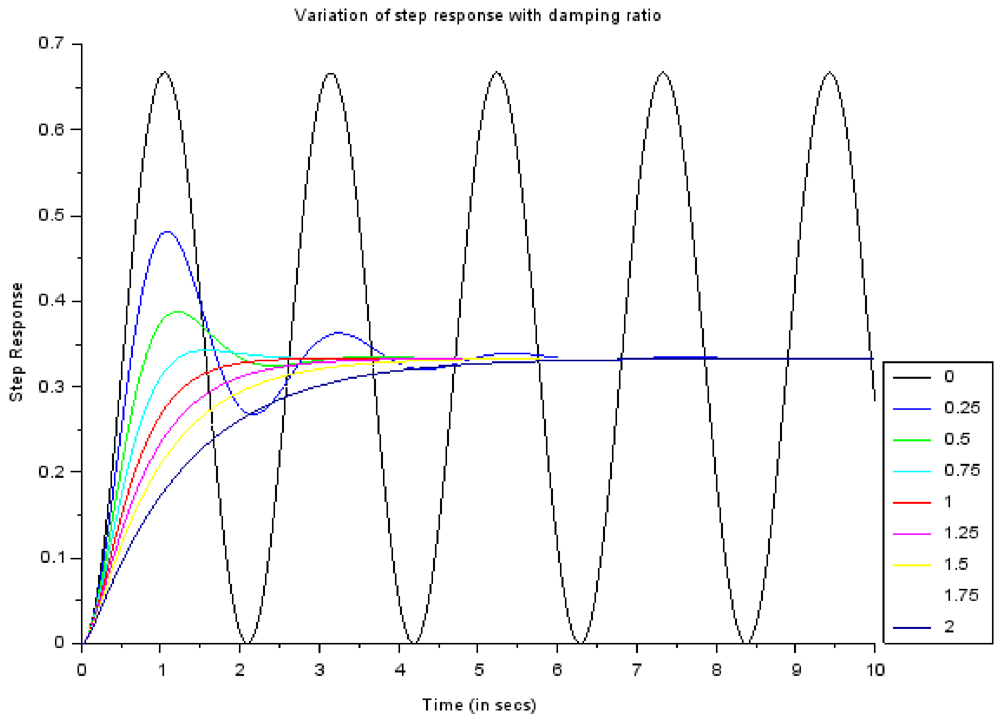
- Since system is underdamped, the graph oscillates about the equilibrium, with frequency as in the code.
- Damping ratio used is $\frac{1}{3}$ and frequency is 3 Hz.



2.3 Part 2(b)

```
// variation with damping ratio
dr_range = 0:0.25:2;
out_range = zeros(5001, 9);

for i = 1:size(dr_range)(2)
    dr = dr_range(i);
    G = f/(s^2 + 2*dr*f*s + f^2);
    sys = syslin('c', G);
    out_range(:, i) = csim("step", t, sys);
end

scf(1);
plot2d(t, out_range, 1:9);
legends(['0'; '0.25'; '0.5'; '0.75'; '1'; '1.25'; '1.5
    '; '1.75'; '2'], 1:9, opt = "?");
xlabel('Time (in secs)');
ylabel('Step Response');
title('Variation of step response with damping ratio')
    ;
```

- With increasing damping factor, percentage overshoot decreases. Rise time and peak time remain approximately same.
- The 2% settling time reaches a minimum at critically damped state, and increases when damping factor is less than or greater than 1.


Variation of step response with damping ratio
Step Response
Time (in secs)
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
1
2
3
4
5
6
7
8
9
10
0
0.25
0.5
0.75
1
1.25
1.5
1.75
2

3 Problem 3

3.1 Code

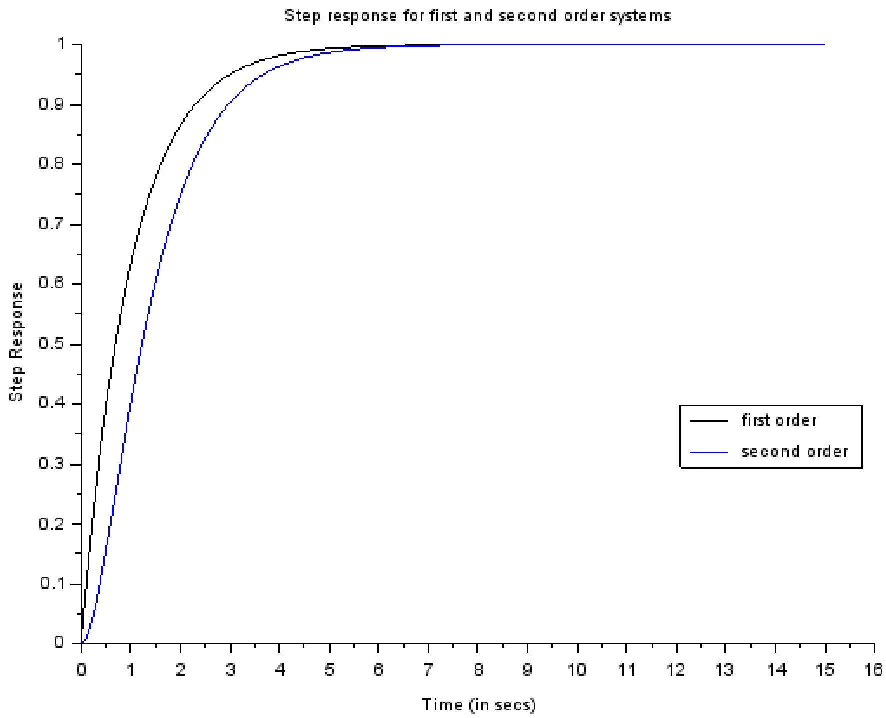
```
s = poly(0, 's');

G1 = 1/(s+1);
sys1 = syslin('c', G1);

G2 = 2/(s^2 + 3*s + 2);
sys2 = syslin('c', G2);

t = 0:0.0003:15;
outs = zeros(50001, 2);

outs(:, 1) = csim('step', t, sys1);
outs(:, 2) = csim('step', t, sys2);

// plotting the step responses
scf(0);
plot2d(t, outs, 1:2);
legends(['first order'; 'second order'], 1:2, opt = "?");
xlabel('Time (in secs)');
ylabel('Step Response');
title('Step response for first and second order systems');

// critically damped case
G3 = 1/(s+1)^2;
sys3 = syslin('c', G3);
out = csim('step', t, sys3);

scf(1);
plot(t, out);
xlabel("Time (in secs)");
ylabel("System response");
title("Step Response for Critically Damped Case");
```

3.2 Part 3(a)

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

G1 = 1/(s+1);
```

```
sys1 = syslin('c', G1);

G2 = 2/(s^2 + 3*s + 2);
sys2 = syslin('c', G2);

t = 0:0.0003:15;
outs = zeros(50001, 2);

outs(:, 1) = csim('step', t, sys1);
outs(:, 2) = csim('step', t, sys2);

// plotting the step responses
scf(0);
plot2d(t, outs, 1:2);
legends(['first order'; 'second order'], 1:2, opt = "?
    ");
xlabel('Time (in secs)');
ylabel('Step Response');
title('Step response for first and second order
    systems');
```

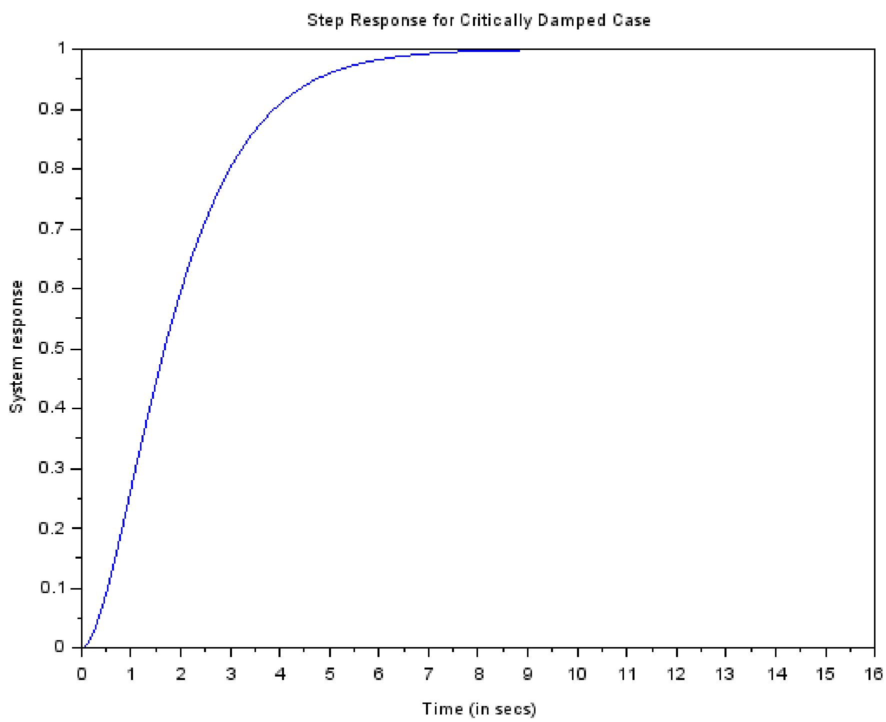
- The first order response has a nonzer derivative at 0, but the second order response has derivative 0.
- The first order response approaches the equilibrium more quickly than the second order response.
- The first order response is concave throughout but the second order response transitions from convex to concave.



3.3 Part 3(b)

```
// critically damped case
G3 = 1/(s+1)^2;
sys3 = syslin('c', G3);
out = csim('step', t, sys3);

scf(1);
plot(t, out);
xlabel("Time (in secs)");
ylabel("System response");
title("Step Response for Critically Damped Case");
```

- For the critically damped case (repeated poles), the response is indeed monotonic.


Step Response for Critically Damped Case
System response
Time (in secs)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

4 Problem 4

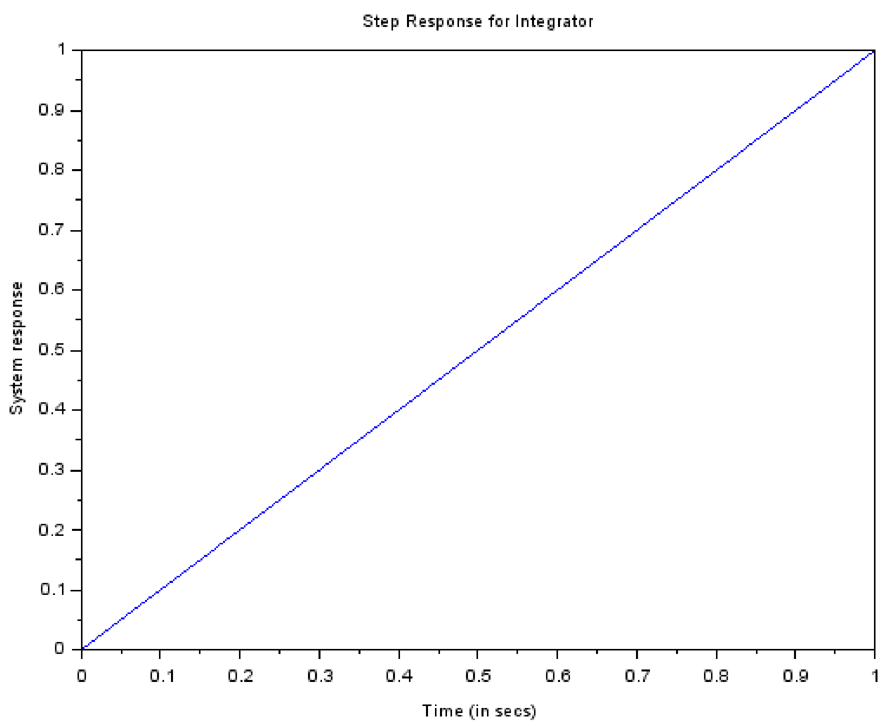
4.1 Code

```
// step response of integrator
s = poly(0, 's');
G1 = 1/s;
sys1 = syslin('c', G1);

t = 0:0.0001:1;
out1 = csim('step', t, sys1);

scf(0);
plot(t, out1);
xlabel("Time (in secs)");
ylabel("System response");
title("Step Response for Integrator");

// discrete system step response
z = poly(0, 'z');
G2 = 1/z;
sys2 = syslin('d', G2);
state = tf2ss(sys2);

n = 0:10;
step = ones(1, 11);
step(1, 1) = 0;
out2 = dsimul(state, step);

scf(1);
plot2d(n, out2, -1, rect = [-1, -0.2, 12, 1.2]);
xlabel("Time (in secs)");
ylabel("Discrete system response");
title("Step Response for Delay System");

// plotting a polynimial directly
p = 1/s;
out3 = csim('step', t, p);

scf(2);
plot(t, out3);
xlabel("Time (in secs)");
ylabel("Polynomial response");
title("Step Response for polynomial 1/s");
```

4.2 Part 4(a)

```
// step response of integrator
s = poly(0, 's');
G1 = 1/s;
sys1 = syslin('c', G1);

t = 0:0.0001:1;
out1 = csim('step', t, sys1);

scf(0);
plot(t, out1);
xlabel("Time (in secs)");
ylabel("System response");
title("Step Response for Integrator");
```

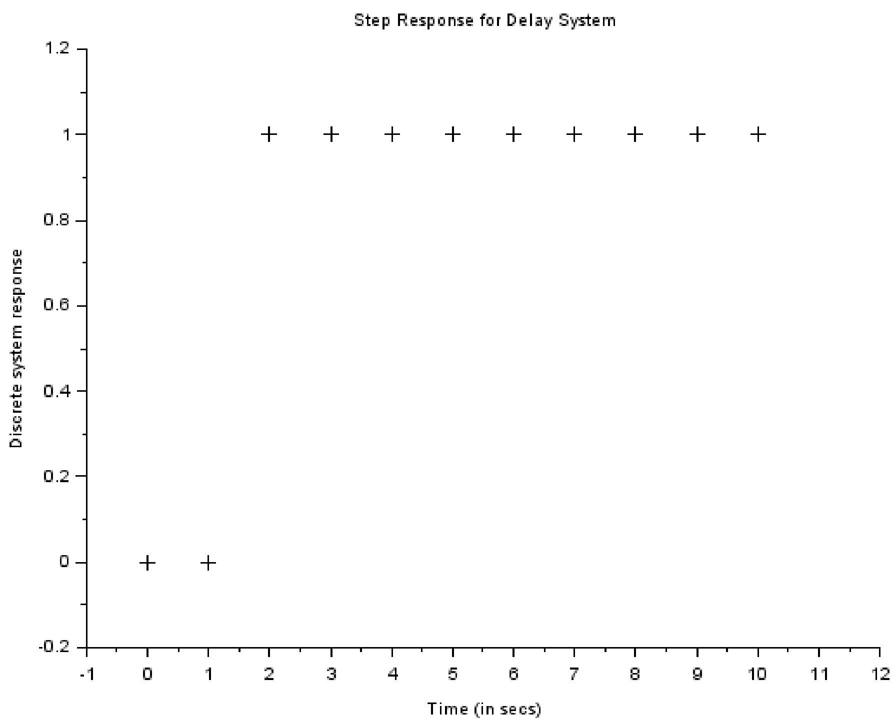
- The step response of the integrator is just a clipped linear response.



4.3 Part 4(b)

```
// discrete system step response
z = poly(0, 'z');
G2 = 1/z;
sys2 = syslin('d', G2);
state = tf2ss(sys2);

n = 0:10;
step = ones(1, 11);
step(1, 1) = 0;
out2 = dsimul(state, step);

scf(1);
plot2d(n, out2, -1, rect = [-1, -0.2, 12, 1.2]);
xlabel("Time (in secs)");
ylabel("Discrete system response");
title("Step Response for Delay System");
```

- The discrete system with transfer function $\frac{1}{z}$ acts as a delay of one unit, and hence the reponse is a delayed step.

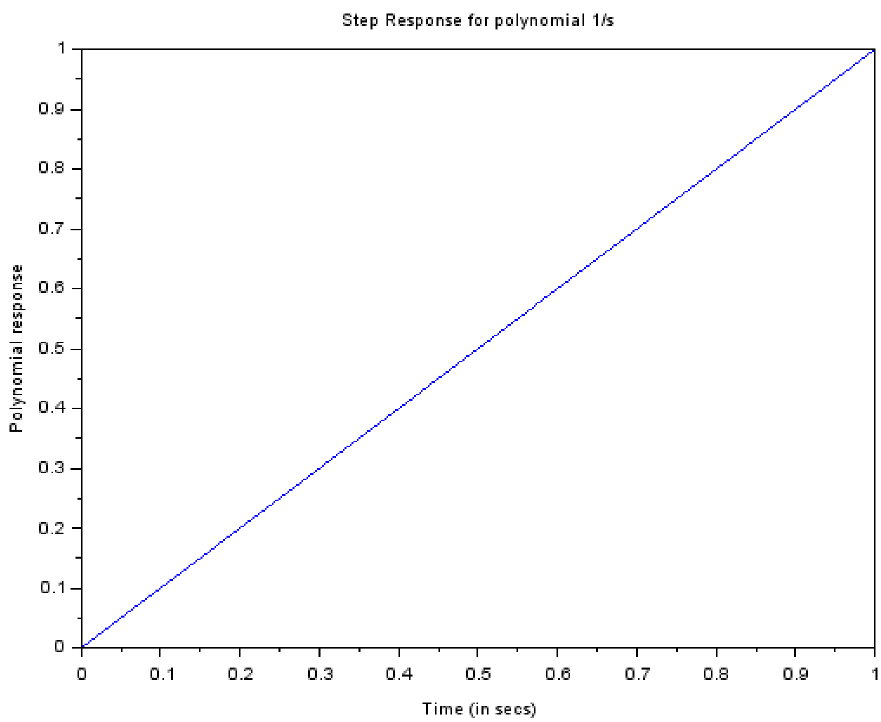


4.4 Part 4(c)

```
// plotting a polynimial directly
p = 1/s;
out3 = csim('step', t, p);

scf(2);
plot(t, out3);
xlabel("Time (in secs)");
ylabel("Polynomial response");
title("Step Response for polynomial 1/s");
```

- The response is similar to the continous case, but we end up with a warning in the console.

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

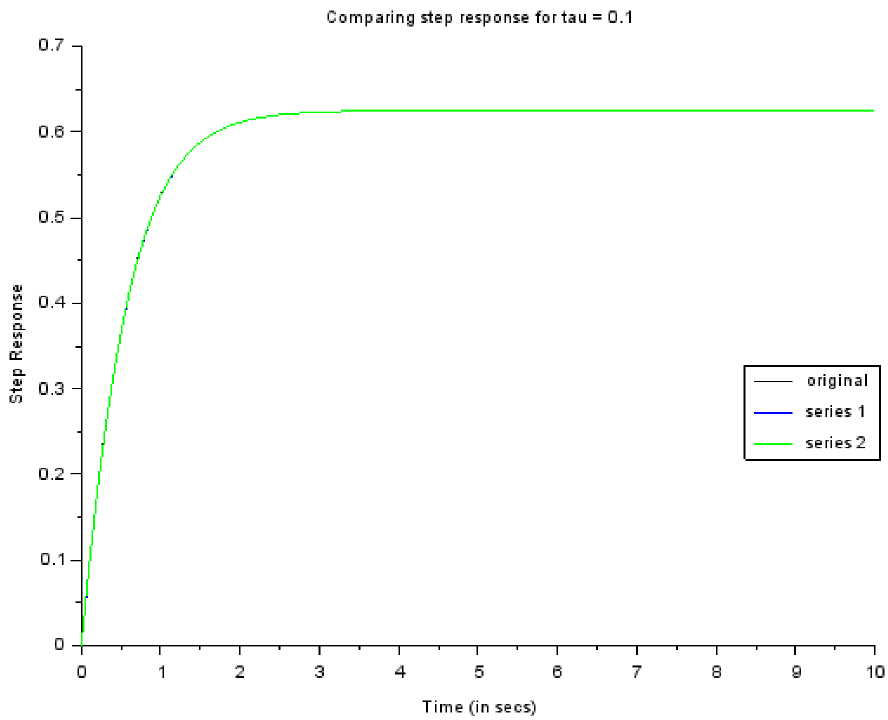

Step Response for polynomial 1/s
Polynomial response
Time (in secs)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

5 Problem 5

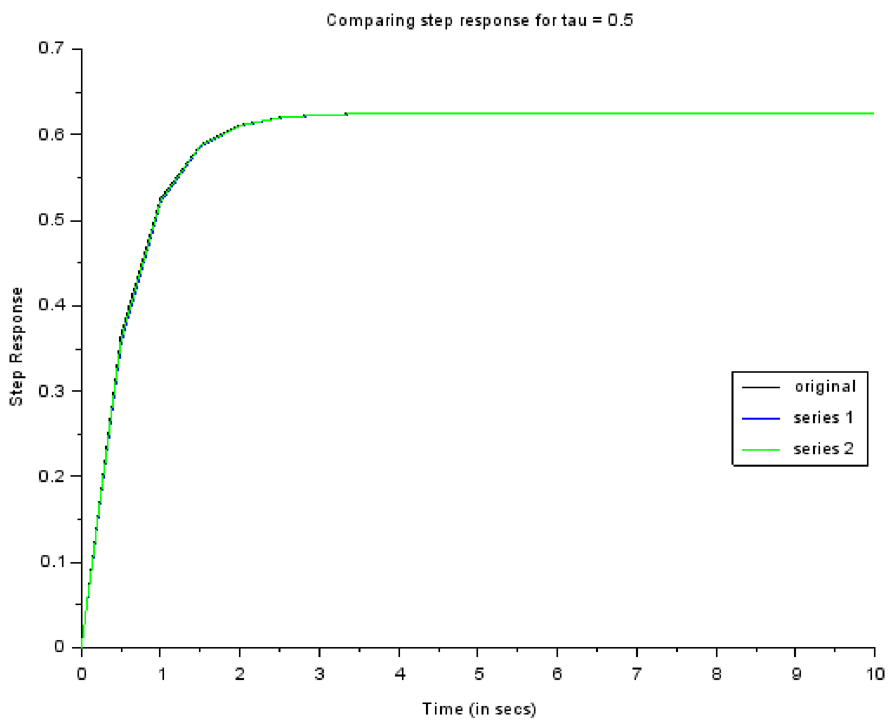
5.1 Code

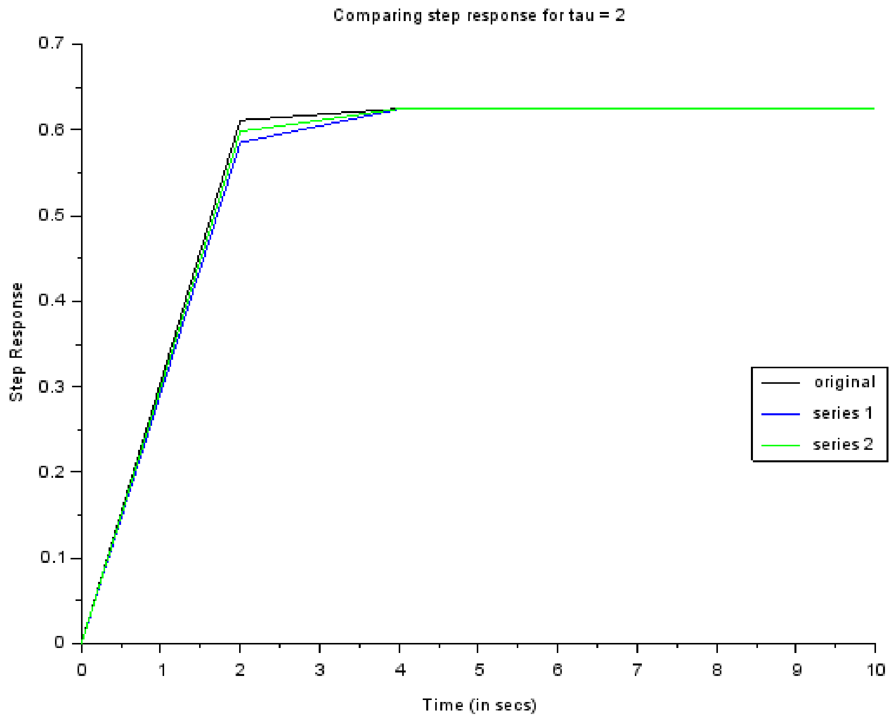
```
s = poly(0, 's');

G1 = (s+5)/((s+4)*(s+2));
sys1 = syslin('c', G1);

G2 = (s+5)/(s+4);
sys2 = syslin('c', G2);

G3 = 1/(s+2);
sys3 = syslin('c', G3);

taus = [0.1, 0.5, 2];

for i = 1:3
    tau = taus(i);
    t = 0:tau:10;

    out1 = csim('step', t, sys1);

    y1 = csim('step', t, sys2);
    out2 = csim(y1, t, sys3);

    y2 = csim('step', t, sys3);
    out3 = csim(y2, t, sys2);

    scf(i);
    plot2d(t, [out1', out2', out3'], 1:3);
    legends(["original", "series 1", "series 2"], 1:3, opt="?");
    xlabel("Time (in secs)");
    ylabel("Step Response");
    title("Comparing step response for tau = " + string(tau));
end
```

5.2 Plots



Comparing step response for tau = 0.1
Step Response
Time (in secs)
original
series 1
series 2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
1
2
3
4
5
6
7
8
9
10


Comparing step response for tau = 0.5
Step Response
Time (in secs)
original
series 1
series 2
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
1
2
3
4
5
6
7
8
9
10



Comparing step response for tau = 2
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Step Response
0
1
2
3
4
5
6
7
8
9
10
Time (in secs)
original
series 1
series 2