**EE324 Control Systems Lab**

Problem sheet 2

Tarun Sai Goddu | 190070024

**Question 1**

My Roll number is 190070024 and “Tarun” is my first name therefore a=24 and b=20, Hence the continuous LTI system with transfer function G(s) = 24/(s+20)

1. LTI Function:

Scilab Code for the same:

s=poly(0,'s')

a = 24

b = 20

G = a/(s+b)

sys = syslin('c',g)

1. Unit Step Response:

Scilab Code for the same:

b= 20

t1 = 1/b

t\_r = 2.2/b

t\_settling = 4/b

r\_timelow = log(10/9)\*(1/b)

r\_timehigh = log(10/1)\*(1/b)

t = 0:0.00002:0.5

resp1 = csim('step',t,sys)

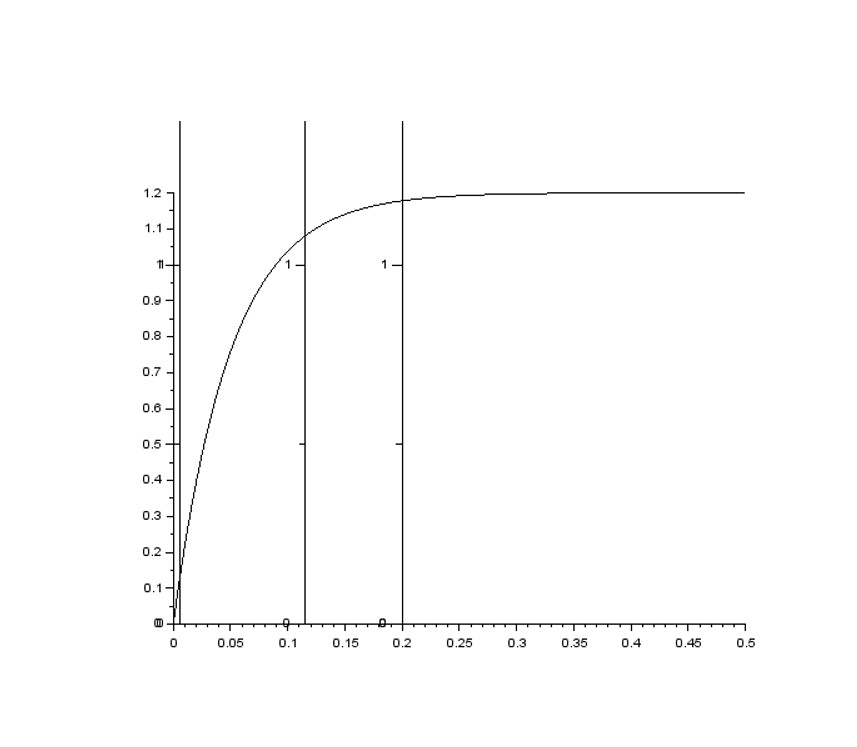
plot2d(t,resp1)

k = drawaxis(x=t\_settling, y =0:3, dir='l',tics='v')

k = drawaxis(x=r\_timelow, y =0:3, dir='l',tics='v')

k = drawaxis(x=r\_timehigh, y =0:3, dir='l',tics='v')

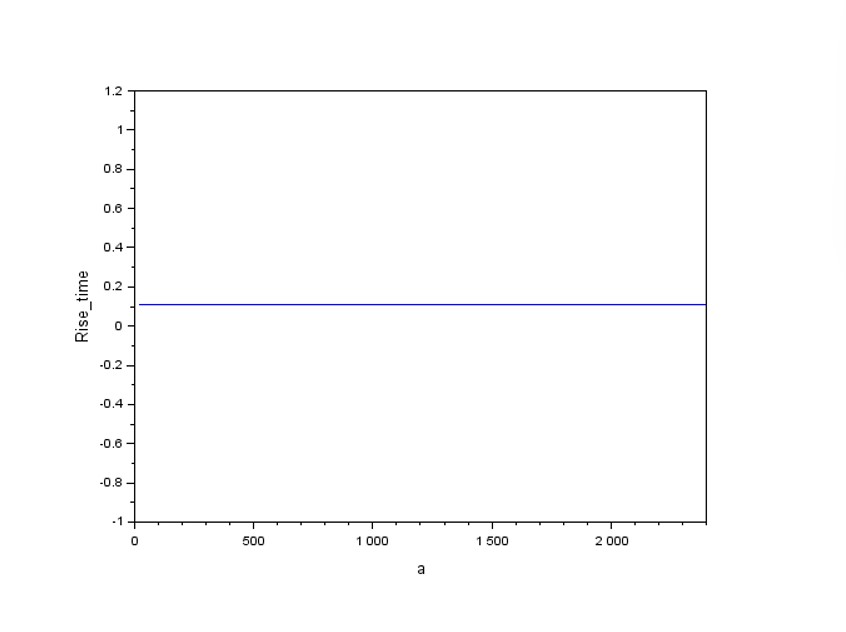
xs2png (0 , "q1b.png")



1. Unit Step Response
2. Part C:

Rise Time for a system is given by Tr = 2.2/b

Transfer function is G(s) = (a)/(s+b)



1. Figure 2: Rise time vs a

The rise time is independent of a , hence there is no variation in rise time with "a”

Scilab Code for the Plot:

Aval = a:a:100\*a;

rise\_time = ones(Aval)\*(r\_timehigh-r\_timelow);

scf( ) ;

plot(Aval , rise\_time);

xlabel("a", 'fontsize', 2.5 );

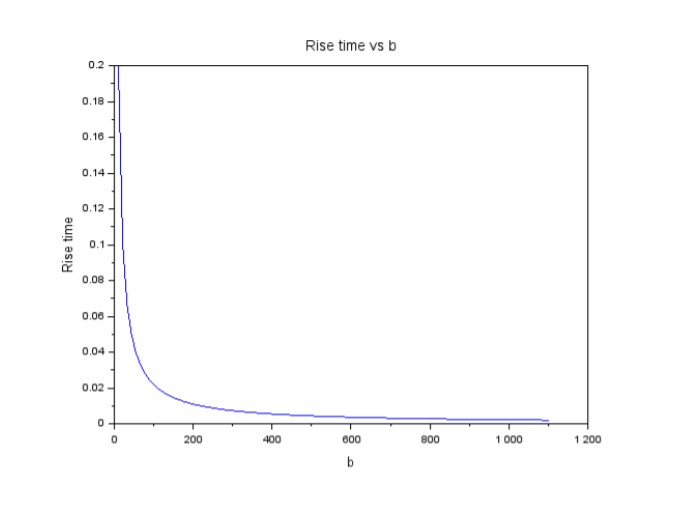
ylabel("Rise\_time" ,'fontsize', 2.5 );

xs2png ( gcf()," 1c.png" )

1. Part D :

Rise Time for a system is given by Tr = 2.2/b

Transfer function is G(s) = (a)/(s+b)



1. Figure 2: Rise time vs b

Scilab Code for the Plot:

Bval = b:b:100\*b;

rise\_time2 = (1/Bval)\*(log(10)-log(10/9))

scf( ) ;

plot(Bval , rise\_time2);

xlabel("b", 'fontsize', 2.5 );

ylabel("Rise\_time" ,'fontsize', 2.5 );

xs2png ( gcf()," 1c.png" )

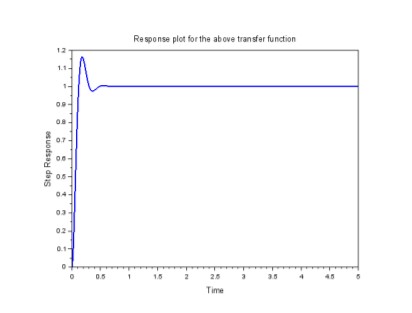
**Question 2**

The example for an under-damped second order continuous time system with no zeros taken is:

G(s) = 400/(s^2+20s+400)

The damping ratio = 0.5 and the natural frequency ωn = 20rad/s.

Scilab Code for finding Poles and zeros of the transfer function:



1. Response Plot

Scilab code for the above transfer function are :

s=poly(0,'s')

G = 400/(s^2+20\*s+400);

sys = syslin(‘c’,G )

t = 0:0.002:5

response = csim(‘step’,t,sys)

scf();

plot(t,response,’LineWidth,2);

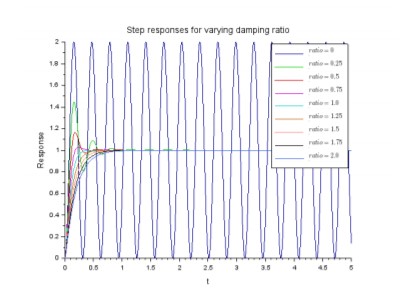
x label(" Time" , ’ fontsize’ , 2 . 5 ) ;

y l a b e l (" Step Response ",’ fontsize’ , 2 . 5) ;

t i t l e (" Response plot for the above transfer function" , ‘fontsize’ , 2.5 ) ;

xs2png ( gcf ( ) , " 2.png " ) ;

After varying the damping ratio from 0 to 2 in the steps of 0.25, the step response plot obtained is as follows



1. Response Plot

Scilab code for the above transfer function are :

d\_ ratio = 0 : 0 . 25 : 2 ;

wn = 20;

scf( ) ;

colors\_p = [ " scilab blue4 " ," scilab green2 " ," scilabred2 " ," scilab magenta2"

,"scilab cyan2 " ," scilab brown2" ," scilab pink 4 " ," black " , " royal blue " ] ;

for j =1: size ( d\_ratio , 2 )

G = wn^2/( s ^2 + 2∗ d\_ ra tio ( j )∗wn∗ s + wn^2 );

s\_gen = syslin ( ’ c ’ , G) ;resp\_gen = csim ( ’ s tep ’ , t , s\_gen ) ;

plo t2 d ( t , resp\_gen , s t y l e= [ c o l o r ( colors\_p ( j ) ) ] ) ;

end

x l a b e l (" t " , ’ fontsize ’ , 2 . 5 ) ;

y l a b e l (" Response " , ’ fontsize ’ , 2 . 5 ) ;

legend ( [ " $ ratio = 0$" ," $ ratio = 0. 25 $ " ," $ ratio = 0. 5 $ " ," $ ratio = 0. 75 $"

," $ ratio = 1. 0 $ " ," $ ratio = 1.2 5 $ " ," $ ratio = 1. 5 $ " ," $ ratio = 1.75 $ " ," $ ratio = 2. 0 $ " ] ) ;

t i t l e (" Step responses for varying damping ratio " , ’ fontsize ’ , 3 ) ;

xs2png ( gcf ( ) ," q2b.png " ) ;

We find that as ζ increases,

• Peak time increases

• Settling time decreases

• %OS decreases

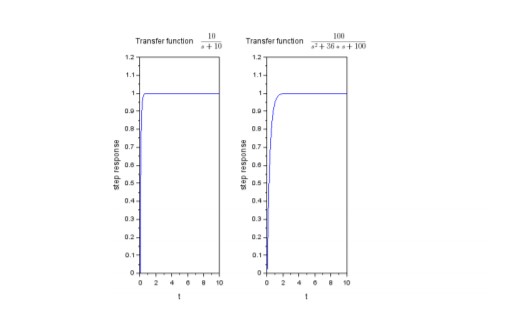
• Rise time increases

**Question 3**

The systems built are :

First order: G1(s) = 10/s+10

Second order : G2(s) = 100/s^2+36∗s+100



1. Response plot for a first order and a second order transfer function

Scilab code for the same is as follows:

first \_ order\_t f = 10/ ( s +10);

s \_ f i r s t = s y s l i n ( ’ c ’ , f i r s t \_ o r d e r \_ t f ) ;

second\_o rde r\_ t f = 100/ ( s ^2 + 36∗ s + 1 0 0 ) ;

s\_second = s y s l i n ( ’ c ’ , second\_o rde r\_ t f ) ;

t = 0 : 0 . 0 2 : 1 0 ;

r e s p1 = csim ( ’ step ’ , t , s \_ f i r s t ) ;

r e s p2 = csim ( ’ step ’ , t , s\_second ) ;

s c f ( ) ;

s u b plo t ( 1 3 1 ) , pl o t ( t , r e s p1 ) ;

x l a b e l ( ’ t ’ , ’ f o n t s i z e ’ , 2 . 5 ) ;

y l a b e l (" s t e p r e s po n s e " , ’ f o n t s i z e ’ , 2 . 5 ) ;

t i t l e ( [ " T ra n s f e r f u n c ti o n " , "$\ f r a c {10}{ s+10}$ " ] , ’ f o n t s i z e ’ , 2 ) ;

s u b plo t (132 ) , pl o t ( t , r e s p2 ) ;

x l a b e l ( ’ t ’ , ’ f o n t s i z e ’ , 2 . 5 ) ;

y l a b e l (" s t e p r e s po n s e " , ’ f o n t s i z e ’ , 2 . 5 ) ;

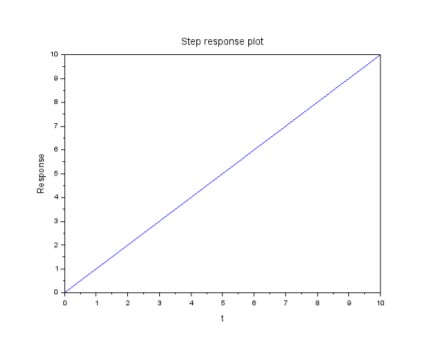
t i t l e ( [ " T ra n s f e r f u n c ti o n " , "$\ f r a c {100}{ s ^2+36∗ s +100}$ " ] , ’ f o n t s i z e ’ , 2 ) ;

xs2png ( g c f ( ) , " q3 . png " ) ;

**Question 4**

1. Part a

The transfer function for a single-integrator is G(s) = 1/s



1. Response Plot

Scilab code for the same is as follows:

−−> G4 = 1/ s ;

−−> S4 = s y s l i n ( ’ c ’ , G4 ) ;

−−> s c f ( ) ;

−−> r e s p4 = csim ( ’ s tep ’ , t , S4 ) ;

−−> pl o t ( t , r e s p4 ) ;

−−> x l a b e l (" t " , ’ f o n t s i z e ’ , 2 . 5 ) ;

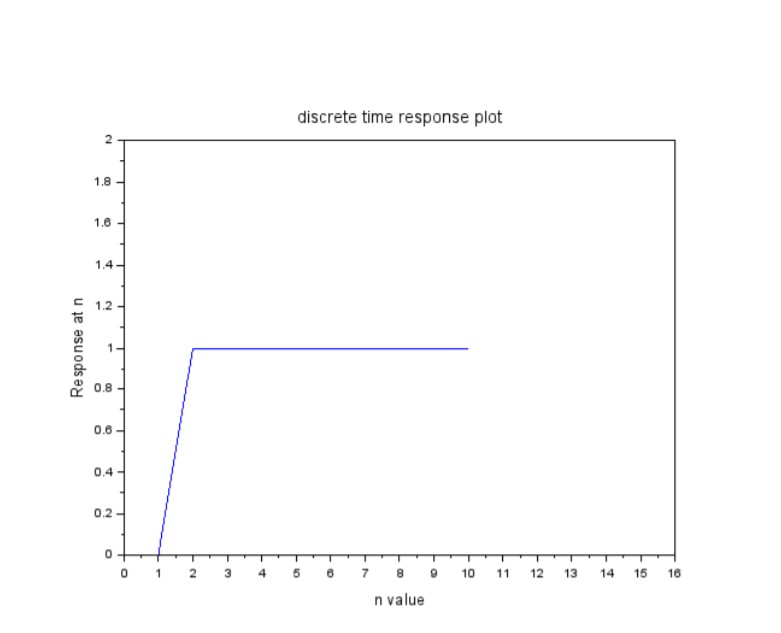
−−> y l a b e l (" Response " , ’ f o n t s i z e ’ , 2 . 5 ) ;

−−> t i t l e (" Step r e s po n s e pl o t " , ’ f o n t s i z e ’ , 3 ) ;

−−> xs2png ( g c f ( ) , " q4 . png " ) ;

1. Part b

The discrete time transfer function is given by H(z) = 1/z



1. Response Plot

Scilab code for the same is as follows:

−−> z = pol y ( 0 , ’ z ’ ) ;

−−> H1 = 1/ z ;

−−> s = t f 2 s s (H1 ) ;

−−> var = one s ( 1 , 1 0 ) ;

−−> val = d simul ( s , var ) ;

−−> s c f ( ) ;

−−> pl o t ( val ) ;

−−> s e t ( gca ( ) , " data\_bounds " , [ 0 , 0 ; 1 5 , 2 ] ) ;

−−> x l a b e l (" n val u e " , ’ f o n t s i z e ’ , 2 . 5 ) ;

−−> y l a b e l (" Response a t n " , ’ f o n t s i z e ’ , 2 . 5 ) ;

−−> t i t l e (" d i s c r e t e time r e s po n s e pl o t " , ’ f o n t s i z e ’ , 3 ) ;

−−> xs2png ( g c f ( ) , " q4b . png " ) ;

1. Part c:

When ratio of two polynomials is given as input to the csim command , then scilab gives the following error :

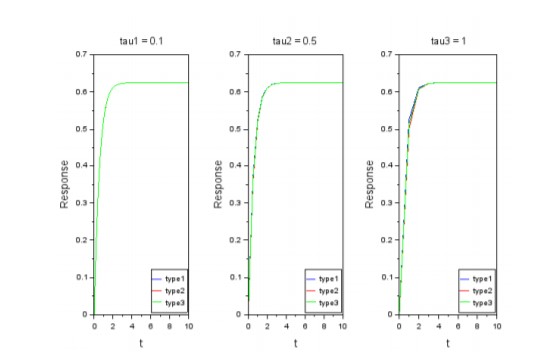
–> resp5 = csim(’step’,t,G/G4)

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

When we compare the results of 4a and 4b , we find a few differences and that is because in 4a step response is calculated in continuous time domain and in 4b it is obtained in discrete time domain.

**Question 5**

The transfer function given is: G(s) = s+5/( (s+4)(s+2))

****

a) Response plots for 3 cases with 3 different tau values

Scilab code for the same is :

−> tau1 = 0 : 0 . 1 : 1 0 ;

−−> tau2= 0 : 0 . 5 : 1 0 ;

−−> tau3 = 0 : 1 : 1 0 ;

−−> s= pol y ( 0 , ’ s ’ ) ;

−−> G1 = ( s +5)/(( s +4)∗( s +2 ) );

−−> G2 = ( s +5)/( s +4);

−−> G3 = 1/ ( s +2);

−−> s y s1 = s y s l i n ( ’ c ’ , G1 ) ;

−−> s y s2 = s y s l i n ( ’ c ’ , G2 ) ;

−−> s y s3 = s y s l i n ( ’ c ’ , G3 ) ;

−−> r e s p1 = csim ( ’ s tep ’ , tau1 , s y s1 ) ;

−−> r e s p21 = csim ( ’ s tep ’ , tau1 , s y s2 ) ;

−−> r e s p22 = csim ( re sp21 , tau1 , s y s3 ) ;

−−> r e s p31 = csim ( ’ s tep ’ , tau1 , s y s3 ) ;

−−> r e s p32 = csim ( re sp31 , tau1 , s y s2 ) ;

−−> s c f ( ) ;

−−> s u b plo t (131 ) , pl o t ( tau1 , re sp1 , ’ b ’ ) ;

−−> s u b plo t ( 1 3 1 ) , pl o t ( tau1 , re sp22 , ’ r ’ ) ;

−−> s u b plo t ( 1 3 1 ) , pl o t ( tau1 , re sp32 , ’ g ’ ) ;

−−> l eg e n d ( [ " type1 " ," type2 " ," type3 " ] , 4 ) ;

−−> t i t l e (" tau1 = 0 . 1 " , ’ f o n t s i z e ’ , 2 . 5 ) ;

−−> x l a b e l (" t " , ’ f o n t s i z e ’ , 3 ) ;

−−> y l a b e l (" Response " , ’ f o n t s i z e ’ , 3 ) ;

> r e s p1 = csim ( ’ s tep ’ , tau2 , s y s1 ) ;

−−> r e s p21 = csim ( ’ s tep ’ , tau2 , s y s2 ) ;

−−> r e s p22 = csim ( re sp21 , tau2 , s y s3 ) ;

−−> r e s p31 = csim ( ’ s tep ’ , tau2 , s y s3 ) ;

−−> r e s p32 = csim ( re sp31 , tau2 , s y s2 ) ;

−−> s u b plo t (132 ) , pl o t ( tau2 , re sp1 , ’ b ’ ) ;

−−> s u b plo t ( 1 3 2 ) , pl o t ( tau2 , re sp22 , ’ r ’ ) ;

−−> s u b plo t ( 1 3 2 ) , pl o t ( tau2 , re sp32 , ’ g ’ ) ;

−−> l eg e n d ( [ " type1 " ," type2 " ," type3 " ] , 4 ) ;

−−> t i t l e (" tau2 = 0 . 5 " , ’ f o n t s i z e ’ , 2 . 5 ) ;

−−> x l a b e l (" t " , ’ f o n t s i z e ’ , 3 ) ;

−−> y l a b e l (" Response " , ’ f o n t s i z e ’ , 3 ) ;

−−> r e s p1 = csim ( ’ s tep ’ , tau3 , s y s1 ) ;

−−> r e s p21 = csim ( ’ s tep ’ , tau3 , s y s2 ) ;

−−> r e s p22 = csim ( re sp21 , tau3 , s y s3 ) ;

−−> r e s p31 = csim ( ’ s tep ’ , tau3 , s y s3 ) ;

−−> r e s p32 = csim ( re sp31 , tau3 , s y s2 ) ;

−−> s u b plo t (133 ) , pl o t ( tau3 , re sp1 , ’ b ’ ) ;

−−> s u b plo t ( 1 3 3 ) , pl o t ( tau3 , re sp22 , ’ r ’ ) ;

−−> s u b plo t ( 1 3 3 ) , pl o t ( tau3 , re sp32 , ’ g ’ ) ;

−−> l eg e n d ( [ " type1 " ," type2 " ," type3 " ] , 4 ) ;

−−> t i t l e (" tau3 = 1" , ’ f o n t s i z e ’ , 2 . 5 ) ;

−−> x l a b e l (" t " , ’ f o n t s i z e ’ , 3 ) ;

−−> y l a b e l (" Response " , ’ f o n t s i z e ’ , 3 ) ;

−−> xs2png (0 ," q5 . png " ) ;

The plots in all the 3 cases differ