EE324 Lab 4

Shubham Kar, 180070058

February 2021

1 Aim

To make interconnected systems in Scilab using simpler control system blocks and to correctly examine a feedback system using Routh-Hurwitz tables.

2 Question 1

2.1 Part a

The given interconnected system is as shown in Figure 1: This system was coded

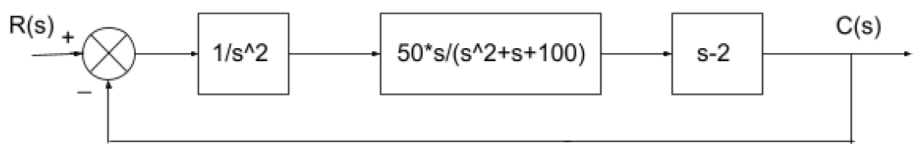

Figure 1: Interconnected system to code in part A

in a straight forward fashion in Scilab as shown in the below listing:

```
clear();
s = poly(0, 's');
t = 0:0.01:10;

//------------Intializing individual blocks------------
sys1 = syslin('c', 1, s^2);
sys2 = syslin('c', 50*s, s^2+s+100);
sys3 = syslin('c', s-2, 1);
//-----------Cascading all the intial blocks-----------
sys4 = sys1*sys2*sys3;
//-------making a unity gain negative feedbcak loop----
```

```
sys5 = syslin('c', 1, 1);
sys = sys4/.sys5;
```

2.2 Part b

The given interconnected system is as shown in Figure 2: In this system, the

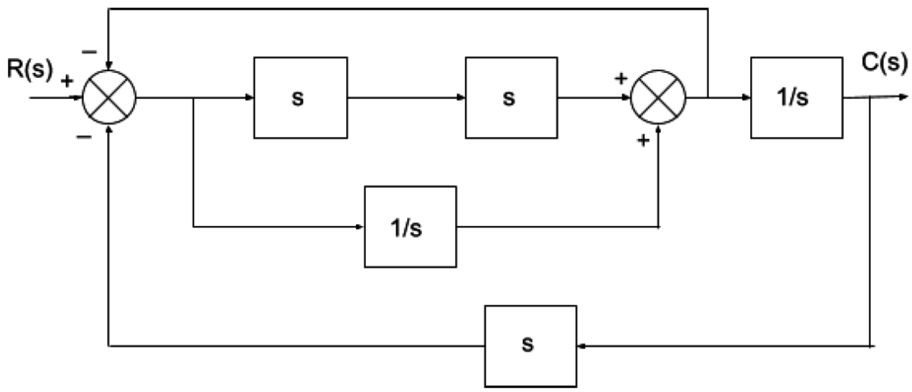

Figure 2: Interconnected system to code in part B

resultant system of due to the inner feedback loop was made first and then it was cascaded and put into a bigger feedback loop as shown in the below listing:

```
clear();
s = poly(0, 's');
t = 0:0.01:10;

//---------------------Intializing individual blocks
    -----------------------
sys1 = syslin('c', s, 1);
sys2 = syslin('c', s, 1);
sys3 = syslin('c', 1, s);
sys4 = syslin('c', 1, s);
sys5 = syslin('c', s, 1);
sys6 = syslin('c', 1, 1);
//------------------Creating the inner feedback loop
    ------------------------
sys7 = ((sys1*sys2)+sys3)/.sys6;
//--Joining the inner feedbakc loop to a bigger one
    after a single cascade--
sys = (sys7*sys4)/.sys5;
```

2.3 Part c

The given interconnected system is as shown in Figure 3: In this system, a

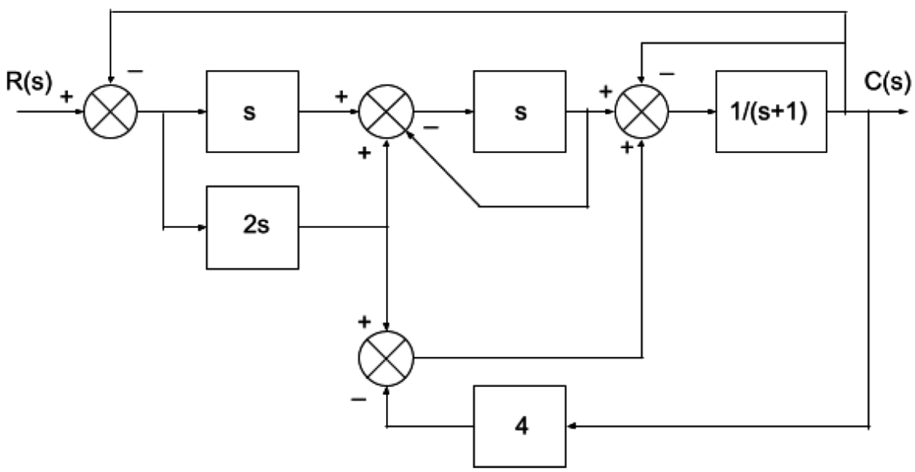

Figure 3: Interconnected system to code in part C

similar approach was followed as in part B, except that the block with a proportionality constant of 4 in it was shifted to the last feedback loop to make the system easier to visualize as shown in the below listing:

```
clear();
s = poly(0, 's');
t = 0:0.01:10;

//-------------------Intializing individual blocks-------------------
sys1 = syslin('c', s, 1);
sys2 = syslin('c', s, 1);
sys3 = syslin('c', (2*s), 1);
sys4 = syslin('c', 1, (s+1));
//-----------Shifting constant 4 to the last feedback loop-----------
sys5 = syslin('c', 5, 1);
sys6 = sys2/.syslin('c',1,1);
sys7 = sys4/.sys5;
//-----Parallely adding and cascading the respective components------
sys8 = (sys1 + sys3)*sys6;
sys9 = (sys8 + sys3)*sys7;
sys = (sys9)/.syslin('c',1,1);
```

3 Question 2

The normal negative feedback loop with unity gain and a proportionality constant of K is given by:

$$H(s) = \left(\frac{KG(s)}{1+KG(s)}\right) = \left(\frac{10K}{s^3+6s^2+8s+10K}\right) \tag{1}$$

This was coded in Scilab using the feedback operator /.. Next, we stored all the pole values obtained for each K and plotted them on the complex plane. The resulting plot is shown in Figure 4.

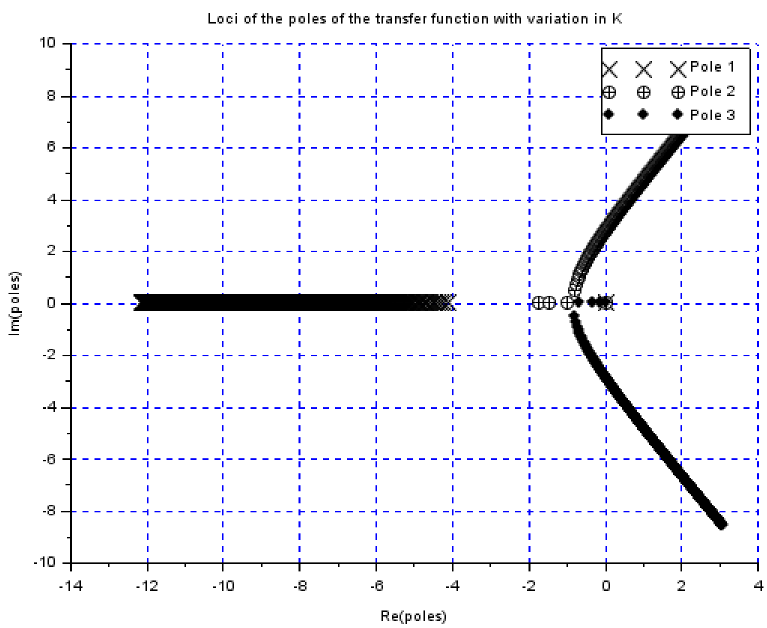

Figure 4: Pole plot of the feedback system $H(s)$

We can see that 2 of the poles(complementary to each other) cross the Left Half plane at a certain value of K. Then, they go on increasing in the right direction. This value of K can be obtained easily by plotting the real value of the one of the LHP crossing poles with respect to K. This gives us the plot shown in Figure 5

The critical value of K comes out to be 4.8 i.e. at this value of K, the poles of the system all lie on the LHP, but two of the poles are on the imaginary axis. This brings the system on the verge of instability. Upon checking with Routh tables, we again obtain that at the value of 4.8, the transfer function has the denominator which has no poles in ORHP but afterwards, it always has 2 poles

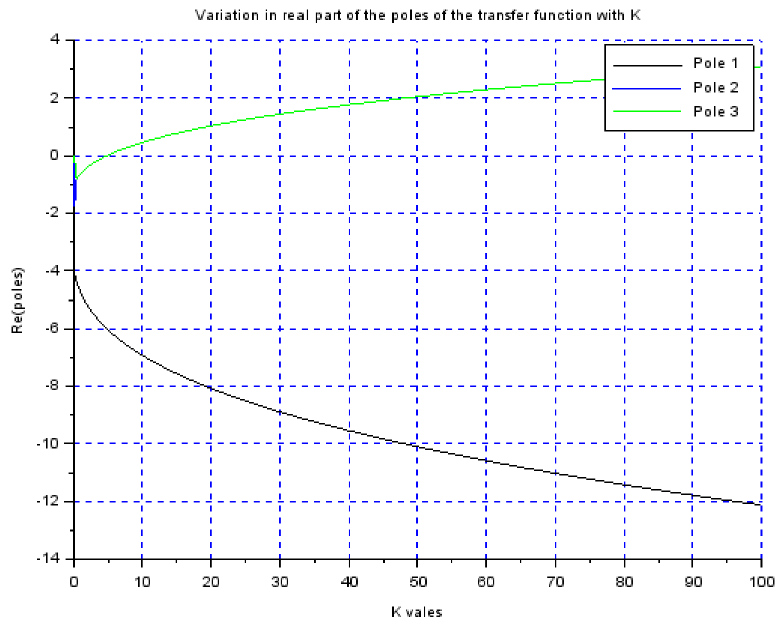

Figure 5: Real part of LHP crossing pole with respect to K

in the ORHP.
Analytically progressing, we have the denominator polynomial as:

$$G(s) \; = \; (s^3 + 6s^2 + 8s + 10K)$$

This gives us the following Routh table as shown in Table 1.

s^3	1	8
s^2	6	10K
s^1	$\left(\frac{48-10K}{6}\right)$	0
s^0	10K	0

Table 1: Routh table of $G(s)$

So, we see that the system is brought to the verge of instability if the first column of of s^1 becomes 0 because after this, increasing the value of K make sit negative increasing the number of ORHP poles. So, the critical value of K has to be 4.8.
The code for generating the same results are shown in the below listing:

```
clear();
s = poly(0, 's');
poles = zeros(length(0:0.1:100), 3);
i = 1;

for K = 0:0.1:100;
    G = (10*K)/(s*(s+2)*(s+4));
    temp_sys = syslin('c', G);
    sys = temp_sys/.syslin('c', 1, 1);
    sys = simp(sys);
    poles(i, :) = (roots(sys.den));
    i = i+1;
end

//---------Plotting real values of one of the pole wrt
     variation in K---------
scf(1);
clf();
plot2d(0:0.1:100, real(poles), 1:3);
xlabel("K vales"); ylabel("Re(poles)");
title("Variation in real part of the poles of the
    transfer function with K");
legend(["Pole 1", "Pole 2", "Pole 3"])
xgrid(2);
xs2png(gcf(), "Q2_real.png");
//----------------------Pole plot with variation in K
    -------------------------
scf(2);
clf();
plot2d(real(poles), imag(poles), [-2, -3, -4]);
xlabel("Re(poles)"); ylabel("Im(poles)");
title("Loci of the poles of the transfer function with
     variation in K");
legend(["Pole 1", "Pole 2", "Pole 3"])
xgrid(2);
xs2png(gcf(), "Q2_poles.png");
//---Finding critical K with a slight tolerance in
    pole values due to precision of calculation----
critical = find(real(poles(:, 2))<1D-10);
K = 0:0.1:100;
//------------------Finding critical value of K using
    Routh tables--------------------
i = 1;
B = zeros(length(0:0.1:100));
for Zeta = 0:0.1:100;
    G = (10)/(s*(s+2)*(s+4));
```

```
    [A, B(i)] = routh_t(G, Zeta);
    i = i+1;
end
critical_1 = find(B==0);
//-----------------Comparing both results
    ------------------------
mprintf("Critical K value = %0.3f\n", K(critical(
    length(critical))));
mprintf("Critical K value = %0.3f\n", K(critical_1(
    length(critical_1))));
```

4 Question 3

The RH tables observed for the polynomials are shown in Figure 6. We observe that the cases where a single zero was observed in a row in the first column, Scilab used the epsilon method to generate the final RH table. This is observed in parts b and d.
In part c, the entire of the third row was supposed to be 0. Scilab automatically put the derivative coefficients of the previous even polynomial encountered in the table for generating the final RH table as observed. Part a is a standard case of finding the Routh table.
The code for generating the Routh tables for the respective polynomials is shown in the below listing:

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

G1 = s^5 + 3*s^4 + 5*s^3 + 4*s^2 + s + 3;
[A, B] = routh_t(G1);
mprintf("Routh table for G1:");
disp(A);
G1 = s^5 + 6*s^3 + 5*s^2 + 8*s + s + 20;
[A, B] = routh_t(G1);
mprintf("Routh table for G2:");
disp(A);
G1 = s^5 - 2*s^4 + 3*s^3 - 6*s^2 + 2*s - 4;
[A, B] = routh_t(G1);
mprintf("Routh table for G3:");
disp(A);
G1 = s^6 + s^5 - 6*s^4 + s^2 + s - 6;
[A, B] = routh_t(G1);
mprintf("Routh table for G4:");
disp(A);
```

```
Routh table for
            2    3    4   5
   3 +s +4s  +5s  +3s  +s

is:
   1.          5.   1.
   3.          4.   3.
   3.6666667   0.   0.
   4.          3.   0.
  -2.75        0.   0.
   3.          0.   0.
```

(a) Part A

```
Routh table for G2:
            2    3   5
  20 +9s +5s  +6s  +s

is:

  1                    6           9
  --                   --          --

  1                    1           1

  eps                  5           20
  ----                 --          ---

   1                   1           1

  -5 + 6eps            -20 + 9eps  0
  ----------           ----------  --

     eps                  eps      1
                  2
  -25 + 50eps - 9eps   20          0
  ------------------   ---         --

      -5 + 6eps        1           1
                   2
  -25 - 90eps - 81eps  0           0
  -------------------  --          --
                  2
  -25 + 50eps - 9eps   1           1

  20                   0           0
  ---                  --          --

  1                    1           1
```

(b) Part B

```
Routh table for G4:
           2     4    5    6
  -6 +s +s  -6s  +s   +s

is:

   1      -6   1    -6
   --     ---  --   ---

   1      1    1    1

   1      0    1    0
   --     --   --   --

   1      1    1    1

   -6     0    -6   0
   ---    --   ---  --

   1      1    1    1

   -24    0    0    0
   ----   --   --   --

    1     1    1    1

   eps    -6   0    0
   ----   ---  --   --

    1     1    1    1

   -144   0    0    0
   -----  --   --   --

   eps    1    1    1

   864    0    0    0
   -----  --   --   --

   -144   1    1    1
```

(c) Part D

```
Routh table for G3:
             2    3    4   5
  -4 +2s -6s  +3s  -2s  +s

is:
    1.          3.    2.
   -2.         -6.   -4.
   -8.         -12.   0.
   -3.         -4.    0.
   -1.3333333   0.    0.
   -4.          0.    0.
```

(d) Part C

Figure 6: RH tables for Question 3

5 Question 4

5.1 Part a

The polynomial needs to have a factor which is an even polynomial and has its highest degree as 4. We took such a polynomial as $(s^4 + 3s^2 + 1)$. Next, we need to make an overall 6 degree polynomial. So, we multiply the 4 degree polynomial with a quadratic polynomial but the quadratic one should no be even. We took the quadratic polynomial as $(s^2 + 5 * s + 5)$. The final 6 degree polynomial we get is:

$$G(s) \;=\; (s^6 + 5s^5 + 8s^4 + 15s^3 + 16s^2 + 5s + 5) \tag{2}$$

Manually calculating the Routh table gives us Table 2:

s^6	1	8	16	5
s^5	5	15	5	0
s^4	5	15	5	0
s^3	$\cancel{0}$ 2	$\cancel{0}$ 3	0	0
s^2	7.5	5	0	0
s^1	$-\frac{5}{3}$	0	0	0
s^0	5	0	0	0

Table 2: Routh table for Question 4 Part A

The resulting Routh table we get in Scilab is shown below in Figure 7: We

1.	8.	16.	5.
5.	15.	5.	0.
5.	15.	5.	0.
20.	30.	0.	0.
7.5	5.	0.	0.
16.666667	0.	0.	0.
5.	0.	0.	0.

Figure 7: RH table obtained in Scilab

observe that except for scaling, the tables are the same.

5.2 Part b

The same idea applies as in part A but here a total degree of is needed. So, we just multiply a polynomial of degree 4 with the quadratic even polynomial considered before. We take that here to be $(s^4 + 2s^3 + s^2 + 5s + 1)$. The resulting 8 degree polynomial is:

$$G(s) \;=\; (s^8 + 2s^7 + 4s^6 + 11s^5 + 5s^4 + 17s^3 + 4s^2 + 5s + 1) \tag{3}$$

Manually calculating the Routh table gives us Table 3:

s^8	1	4	5	4	1
s^7	2	11	17	5	0
s^6	-3	-7	3	2	0
s^5	1	3	1	0	0
s^4	1	3	1	0	0
s^3	$\cancel{0}$ 2	$\cancel{0}$ 3	0	0	0
s^2	3	2	0	0	0
s^1	$\frac{5}{3}$	0	0	0	0
s^0	2	0	0	0	0

Table 3: Routh table for Question 4 Part B

```
 1                          4            5            4    1
 --                         --           --           --   --

 1                          1            1            1    1

 2                          11           17           5    0
 --                         ---          ---          --   --

 1                          1            1            1    1

 -1.5                       -3.5         1.5          1    0
 -----                      -----        ----         --   --

  1                          1            1           1    1

 6.3333333                  19           6.3333333    0    0
 ----------                 ---          ----------   --   --

     1                      1                1        1    1

 1                          3            1            0    0
 --                         --           --           --   --

 1                          1            1            1    1

 eps                        -8.882D-16   0            0    0
 ----                       -----------  --           --   --

  1                             1        1            1    1

 8.882D-16 + 3eps           1            0            0    0
 -----------------          --           --           --   --

       eps                  1            1            1    1

-7.895D-16 - 0.3333333eps  0            0            0    0
--------------------------  --           --           --   --

            1               1            1            1    1

1                           0            0            0    0
--                          --           --           --   --

1                           1            1            1    1
```

Figure 8: RH table obtained in Scilab for part B

The resulting Routh table we get in Scilab is shown below in Figure 8:
We observe that due to precision issues, we get 'eps' in the RH table generated by Scilab. However, the entire row corresponding to s^3 is very low in magnitude, suggesting that it is actually 0.

5.3 Part c

In this part, we go bottom up and start from the row corresponding to s^3. we need 0 in the first column and then we consider 1 in the second column. Then we select the first 2 elements of s^4 to be 1 and 2 and that of s^5 to be 2 and 4. These values guarantee a 0 in the first column of s^3.
Next, we select the first 2 elements of s^6 such that the first element of s^4 becomes 1. This gives us 1 and 3. We then continue constructing the RH table above s^3 as such and obtain the resulting RH table above s^3 to be as shown in Table 4.

s^6	1	3	5	2.5
s^5	2	4	6	0
s^4	1	2	2.5	0
s^3	0	1	0	0

Table 4: Upper part of the Routh table for Question 4 Part C

The RH table obtained using Scilab is shown below in Figure 9.
We observe that the rows leading upto s^3 are exactly as that manually calculated. The epsilon method is later followed by Scilab but we observe that the first element of s^3 indeed comes out to be 0 in this case. The resulting polynomial is:

$$G(s) \;=\; (s^6 + 2s^5 + 3s^4 + 4s^3 + 5s^2 + 6s + 2.5) \tag{4}$$

The code for generating the above RH tables are shown below in the listing:

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

//-----------All evaluations to be done at s^3--------
//-----------Entire row 0 and total degree 6----------
G1 = (s^2 + 5*s + 5)*(s^4 + 3*s^2 + 1);
[A, b] = routh_t(G1);
disp(A);
//-----------Entire row 0 and total degree 8----------
G2 = (s^4 + 2*s^3 + s^2 + 5*s + 1)*(s^4 + 3*s^2 + 1);
disp(G2);
[A, b] = routh_t(G2);
disp(A);
//----------First Column 0 and total degree 6---------
G3 = (s^6 + 2*s^5 + 3*s^4 + 4*s^3 + 5*s^2 + 6*s + 2.5)
    ;
disp(G3);
[A, b] = routh_t(G3);
disp(A);
```

```
1                    3     5     2.5
--                   --    --    ----

1                    1     1      1

2                    4     6     0
--                   --    --    --

1                    1     1     1

1                    2     2.5   0
--                   --    ----  --

1                    1      1    1

eps                  1     0     0
----                 --    --    --

 1                   1     1     1

-1 + 2eps            2.5   0     0
----------           ----  --    --

   eps                1    1     1
                  2
-1 + 2eps - 2.5eps   0     0     0
-------------------  --    --    --

    -1 + 2eps        1     1     1

2.5                  0     0     0
----                 --    --    --

 1                   1     1     1
```

Figure 9: RH table obtained in Scilab for part C