EE324 Problem Sheet 4

Abhilaksh Kumar, 18D070035

February 8, 2021

The code was written in continuation in a single file. Hence in the code section you may find that variables aren't defined again and again.

Q1

PART a

Simplifying the 3 cascaded systems into 1, it becomes a simple feedback system which can be written as:- For the feedback system

$$H4(s) = H1(s) * H2(s) * H3(s)$$
$$\frac{C(s)}{R(s)} = \frac{H4(s)}{1 + H4(s)}$$

The code for this part is

```
s = poly(0,'s');
t = 0:.001:10;

// PART A
h1 = 1/s^2;
h2 = 50*s/(s^2+s+100);
h3 = s-2;
h4 = h1*h2*h3;

Heq = syslin('c', h4/(1+h4));
```

Part b

First we solve innormost part (2 cascade + series) and then handle the upper feedback loop, after which the system looks quite similar to first part (except there is a transfer function in the bottom loop too). The code for this part is

```
h1 = s;
h2 = 1/s;

h3 = h1*h1+h2;           // inner series + parallel connection
h4 = h3/(h3+1)*h2;        // taking care of upper feedback loop

Heq_b = syslin('c', h1*h4/(1+h1*h4));            // very similar to a part
    (just there is additional multiplication in feedback loop),
```

Part c

Just applying block reduction techniques sequentially.
The code for this part is

```
h1 = s;
h2 = 2*s;
h3 = 1/(s+1);
h4 = 4;

G1 = h1+h2;
G2 = h1/(1+h1);
G3 = h1/(G1);           // 1/s+2
G4 = G1*G2+ h2;
G5 = 4*G3/(1+4*G3);
G6 = G4*G5;

Heq = syslin('c', G6/(1+G6));
```

Q2

Part a

For any value of K the transfer function is $\frac{10k}{s^3+6s^2+8s+10k}$
For any specific value say K = 4, it is $\frac{40}{s^3+6s^2+8s+40}$
The code for this part is

```
s = poly(0,'s');
G = 10/(s*(s+2)*(s+4));

// PART A
k = 4;                             // any arbitrary value chosen
H = k*G/(1+k*G);
disp(H);
```

Part b

The plot is as follows:-

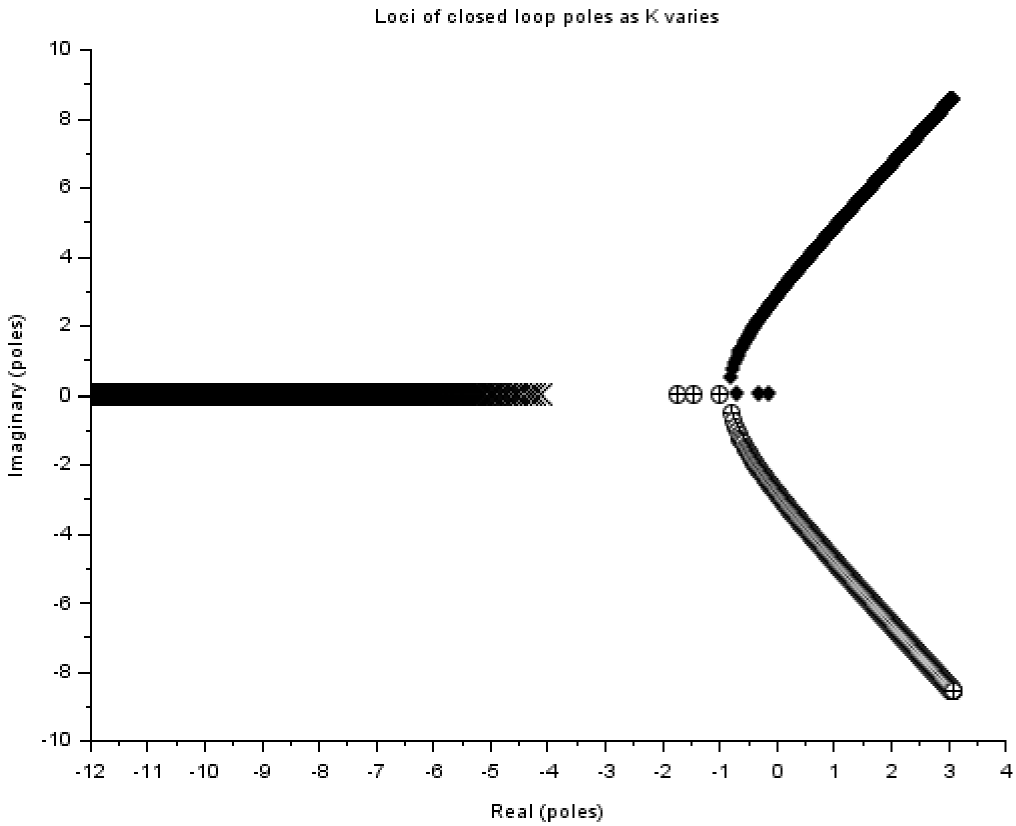


Figure 1: Loci of closed loop poles

The code for this part is

```
k_range = .1:.1:100.1;                                        // excluding k =0 case, as H =0 in that case
// for each k we get 3 poles since denominator of H is cubic
poles = zeros(3,length(k_range));

for i = 1:length(k_range)
    k = k_range(i);
    H = syslin('c',k*G/(1+k*G));
    [_,p, _] = tf2zp(H)
    poles(:,i)  = p;
end

plot2d(real(poles'), imag(poles'), [-2,-3,-4]);
a = gca()
a.data_bounds = [-12,-9;4,9];
title('Loci of closed loop poles as K varies')
xlabel('Real (poles)');
ylabel('Imaginary (poles)');
```

Part c

The critical value of $K_{critical} = 4.8$ The code for this part is

```
k1 = k_range(find(real(poles(2,:))>= 0))(1);
k2 = k_range(find(real(poles(3,:))>= 0))(1);
k3 = k_range(find(real(poles(1,:))>= 0))(1);
k_critical = min(k1,k2,k3);
```

Part d

The RH table for polynomial in denominator i.e. $s^3 + 6s^2 + 8s + 10k$ (for K ¿ 0) is (as calculated manually)

1	8
6	10k
$\frac{48-10k}{6}$	0
10k	0

We know that in order to have a stable system all roots should be in OLHP which implies

$$48 - 10k > 0$$
$$k < 4.8$$

Hence, 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 makes it negative increasing the number of ORHP poles. Also at k = 4.8, the poles lie on the imaginary axis, thus bringing the system on the verge of instability. Afterwards however, it always has poles in ORHP. So, the critical value of K has to be 4.8.

Q3

The Routh tables obtained are as follows The code for this part is

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

// PART a
G1 = s^5+3*s^4+5*s^3+4*s^2+s+3;
[routh_table_1 ,B] = routh_t(G1);

// PART b
G2 = s^5+6*s^3+5*s^2+8*s+20;
[routh_table_2 ,B] = routh_t(G2);

// PART c
G3 = s^5-2*s^4+3*s^3-6*s^2+2*s-4;
```

Var - routh_table_1

	1	2	3
1	1	5	1
2	3	4	3
3	3.6667	0	0
4	4	3	0
5	-2.75	0	0
6	3	0	0

```
--> routh_table_2
 routh_table_2  =


              1                   6      8
              -                   -      -
              1                   1      1

             eps                  5      20
             ---                  -      --
              1                   1      1

          -5 +6eps           -20 +8eps   0
          --------           ---------   -
             eps                 eps     1

     -25 +50eps -8eps²           20      0
     -----------------           --      -
         -5 +6eps                1       1

 -2.274D-13 -160eps -64eps²       0      0
 --------------------------       -      -
     -25 +50eps -8eps²            1      1

              20                  0      0
              --                  -      -
              1                   1      1

--> routh_table_1
```

Var - routh_table_3

	1	2	3
1	1	3	2
2	-2	-6	-4
3	-8	-12	-0
4	-3	-4	0
5	-1.3333	0	0
6	-4	0	0

```
--> routh_table_4
 routh_table_4  =


    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
```

```
[routh_table_3 ,B] = routh_t(G3);

// PART d
G4 = s^6+s^5-6*s^4+s^2+s-6;
[routh_table_4 ,B] = routh_t(G4);
```

Q4

Part a

We want even polynomial of degree 4 and multiply with a quadratic polynomial which is not even and make it degree 6 to get the resultant answer.

$$G1 = (s^2 + 5 * s + 3) * (s^4 + s^2 + 1)$$
$$= 3 + 5s + 4s^2 + 5s^3 + 4s^4 + 5s^5 + s^6$$

The routh table can be verified using scilab to match the question description

Var - routh_table_1

	1	2	3	
1	1	4	4	3
2	5	5	5	0
3	3	3	3	0
4	12	6	0	0
5	1.5	3	0	0
6	-18	0	0	0
7	3	0	0	0

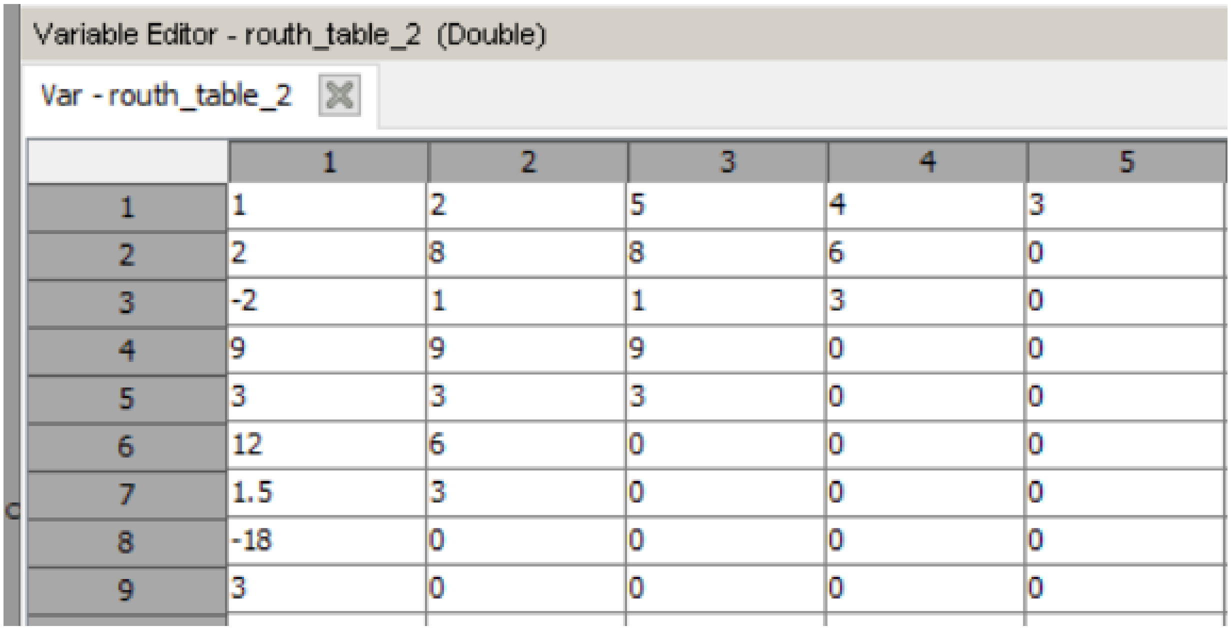
Figure 4: RH table for (a) part Q4

Part b

We want even polynomial of degree 4 and multiply with a quadratic polynomial which is not even and make it degree 6 to get the resultant answer.

$$G2 = (s^4 + 2 * s^3 + s^2 + 6 * s + 3) * (s^4 + s^2 + 1)$$
$$= 3 + 6s + 4s^2 + 8s^3 + 5s^4 + 8s^5 + 2s^6 + 2s^7 + s^8$$

The routh table can be verified using scilab to match the question description



Variable Editor - routh_table_2 (Double)

Var - routh_table_2

	1	2	3	4	5
1	1	2	5	4	3
2	2	8	8	6	0
3	-2	1	1	3	0
4	9	9	9	0	0
5	3	3	3	0	0
6	12	6	0	0	0
7	1.5	3	0	0	0
8	-18	0	0	0	0
9	3	0	0	0	0

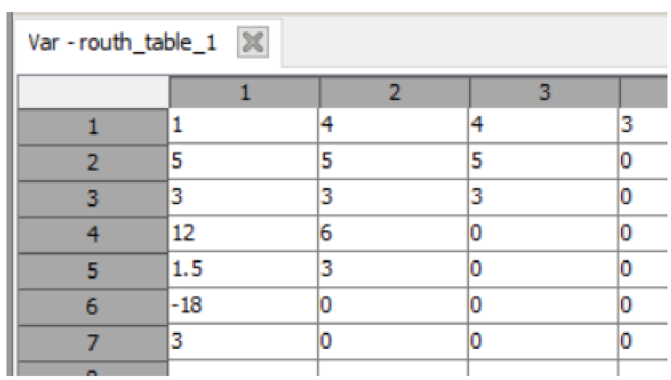
Figure 5: Entire code snippet

Part c

A trivial example that worked was

$$G3 = (s^4 + s^6)$$

The routh table can be verified using scilab to match the question description



Var - routh_table_1

	1	2	3	
1	1	4	4	3
2	5	5	5	0
3	3	3	3	0
4	12	6	0	0
5	1.5	3	0	0
6	-18	0	0	0
7	3	0	0	0

Figure 6: RH table for (c) part Q4

The code for the entire 4th questions is given below:-

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

```

```
// PART a
G1 = (s^2+5*s+3)*(s^4+s^2+1);
[routh_table_1 ,B] = routh_t(G1);
disp(routh_table_1)

// PART b
G2 = (s^4+2*s^3+s^2+6*s+3)*(s^4+s^2+1);
[routh_table_2 ,B] = routh_t(G2);
disp(routh_table_2)

// PART c
G3 = (s^4+s^6);
[routh_table_3 ,B] = routh_t(G3);
disp(routh_table_3)
```