**Chapter 6 – Solutions to Instructor Reserve Problems**

**I-1 Instructor**

**Program:**

den=[1 1 -6 0 1 1 -6]

A=roots(den)

**Computer response:**

den =

1 1 -6 0 1 1 -6

A =

-3.0000

2.0000

-0.7071 + 0.7071i

-0.7071 - 0.7071i

0.7071 + 0.7071i

0.7071 - 0.7071i

**I-2 Instructor**

**Program:**

%-det([si() si();sj() sj()])/sj()

%Template for use in each cell.

syms e %Construct a symbolic object for

%epsilon.

%%%%%%%%%%%%%%%%%%%%%%%%%%$$$$$$$$$%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

s5=[1 3 3 0 0] %Create s^5 row of Routh table.

%%%%%%%%%%%%%%%%%%%%%%%%%%%$$$$$$$$$%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

s4=[-1 -3 -2 0 0] %Create s^4 row of Routh table.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if -det([s5(1) s5(2);s4(1) s4(2)])/s4(1)==0

s3=[e...

-det([s5(1) s5(3);s4(1) s4(3)])/s4(1) 0 0];

%Create s^3 row of Routh table

%if 1st element is 0.

else

s3=[-det([s5(1) s5(2);s4(1) s4(2)])/s4(1)...

-det([s5(1) s5(3);s4(1) s4(3)])/s4(1) 0 0];

%Create s^3 row of Routh table

%if 1st element is not zero.

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if -det([s4(1) s4(2);s3(1) s3(2)])/s3(1)==0

s2=[e ...

-det([s4(1) s4(3);s3(1) s3(3)])/s3(1) 0 0];

%Create s^2 row of Routh table

%if 1st element is 0.

else

s2=[-det([s4(1) s4(2);s3(1) s3(2)])/s3(1) ...

-det([s4(1) s4(3);s3(1) s3(3)])/s3(1) 0 0];

%Create s^2 row of Routh table

%if 1st element is not zero.

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

if -det([s3(1) s3(2);s2(1) s2(2)])/s2(1)==0

s1=[e ...

-det([s3(1) s3(3);s2(1) s2(3)])/s2(1) 0 0];

%Create s^1 row of Routh table

%if 1st element is 0.

else

s1=[-det([s3(1) s3(2);s2(1) s2(2)])/s2(1) ...

-det([s3(1) s3(3);s2(1) s2(3)])/s2(1) 0 0];

%Create s^1 row of Routh table

%if 1st element is not zero

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

s0=[-det([s2(1) s2(2);s1(1) s1(2)])/s1(1) ...

-det([s2(1) s2(3);s1(1) s1(3)])/s1(1) 0 0];

%Create s^0 row of Routh table.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

's3' %Display label.

s3=simplify(s3); %Simplify terms in s^3 row.

pretty(s3) %Pretty print s^3 row.

's2' %Display label.

s2=simplify(s2); %Simplify terms in s^2 row.

pretty(s2) %Pretty print s^2 row.

's1' %Display label.

s1=simplify(s1); %Simplify terms in s^1 row.

pretty(s1) %Pretty print s^1 row.

's0' %Display label.

s0=simplify(s0); %Simplify terms in s^0 row.

pretty(s0) %Pretty print s^0 row.

**Computer response:**

s5 =

1 3 3 0 0

s4 =

-1 -3 -2 0 0

ans =

s3

+- -+

| e, 1, 0, 0 |

+- -+

ans =

s2

+- -+

| 3 e - 1 |

| - -------, -2, 0, 0 |

| e |

+- -+

ans =

s1

+- -+

| 2 |

| 2 e - 3 e + 1 |

| - --------------, 0, 0, 0 |

| 3 e - 1 |

+- -+

ans =

s0

+- -+

| -2, 0, 0, 0 |

+- -+

>>

**I-3 Instructor**



Since at the *s3* row we obtain a row of zeros, we move back to the *s4* row and construct the even polynomial:

.

Differentiating, gives:



Replace the row of zeros with the coefficients 16, 16, 0 and divide by 16 for convenience.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *s4* | 4 | 8 | 1 |  |
| *s3* | 0 16 1 | 0 16 1 | 0 | ROZ |
| *s2* | 4 | 1 | 0 |  |
| *s1* | 3/4 | 0 |  |  |
| *s0* | 1 |  |  |  |

Even (4): 4 jω poles

**b.** Using MATLAB, we write the following M-file and obtain the poles of *T(s)*, listed below.

**Program:**

numg=1;

deng=[4 0 8 0 0];

G=tf(numg,deng);

'T(s)'

T=feedback(G,1)

'Poles of T(s)'

pole(T)

**Computer Response:**

ans =

T(s)

T =

1

-----------------

4 s^4 + 8 s^2 + 1

Continuous-time transfer function.

ans =

Poles of T(s)

ans =

0.0000 + 1.3660i

0.0000 - 1.3660i

0.0000 + 0.3660i

0.0000 - 0.3660i

Thus, all four poles are located on the j axis and the system is marginally stable.

**I-4 Instructor**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| s8 | 1 | 12 | 45 | 82 | 84 |
| s7 | 1 | 5 | 10 | 12 |  |
| s6 | 1 | 5 | 10 | 12 |  |
| s5 | 3 | 10 | 10 |  | ROZ |
| s4 | 5 | 20 | 36 |  |  |
| s3 | -5 | -29 |  |  |  |
| s2 | -1 | 4 |  |  |  |
| s1 | -49 |  |  |  |  |
| s0 | 4 |  |  |  |  |

Even (6): 2 rhp, 2 lhp, 2 j; Rest (2): 0 rhp, 2 lhp, 0 j ; Total: 2 rhp, 4 lhp, 2 j

**I-5 Instructor**

**Program:**

numg=8;

deng=[1 -2 -1 2 4 -8 -4 0];

'G(s)'

G=tf(numg,deng)

'T(s)'

T=feedback(G,1)

'Poles of T(s)'

pole(T)

**Computer response:**

ans =

G(s)

Transfer function:

8

-----------------------------------------------

s^7 - 2 s^6 - s^5 + 2 s^4 + 4 s^3 - 8 s^2 - 4 s

ans =

T(s)

Transfer function:

8

---------------------------------------------------

s^7 - 2 s^6 - s^5 + 2 s^4 + 4 s^3 - 8 s^2 - 4 s + 8

ans =

Poles of T(s)

ans =

-1.0000 + 1.0000i

-1.0000 - 1.0000i

-1.0000

2.0000

1.0000 + 1.0000i

1.0000 - 1.0000i

1.0000

Thus, there are 4 rhp poles and 3 lhp poles.

**I-6 Instructor**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| s5 | 1 | -7 | -18 |  |
| s4 | 1 | -7 | -18 |  |
| s3 | 4 | -14 | 0 | ROZ |
| s2 | -3.5 | -18 | 0 |  |
| s1 | -34.57 | 0 | 0 |  |
| s0 | -18 | 0 | 0 |  |

Even (4): 1 rhp, 1 lhp, 2 j; Rest (1): 1 lhp; Total: 1 rhp, 2 lhp, 2 j

**I-7 Instructor**

T(s) = . For a second-order system, if all coefficients are positive, the roots will be in the lhp. Thus, K > -1.

**I-8 Instructor**

**Program:**

K=[-6:0.00005:0];

for i=1:length(K);

dent=[(1+K(i)) (8\*K(i)-6) (8+15\*K(i))];

R=roots(dent);

A=real(R);

B=max(A);

if B>0

R

K=K(i)

break

end

end

K=[6:-0.00005:0];

for i=1:length(K);

dent=[(1+K(i)) (8\*K(i)-6) (8+15\*K(i))];

R=roots(dent);

A=real(R);

B=max(A);

if B>0

R

K=K(i)

break

end

end

**Computer response:**

R =

1.0e+005 \*

2.7999

-0.0000

K =

-1.0000

R =

0.0001 + 3.3166i

0.0001 - 3.3166i

K =

0.7500

**I-9 Instructor**

**a.**  . Therefore, .

Making a Routh table,

|  |  |  |
| --- | --- | --- |
| s3 | 1 | 6-3K |
| s2 | 4+K | 4+2K |
| s1 |  | 0 |
| s0 | 4+2K | 0 |

From s1 row: K = 1.57, -4.24; From s2 row: - 4 < K; From s0 row: - 2 < K. Therefore,

- 2 < K < 1.57.

**b.** If K = 1.57, the previous row is 5.57s2 + 7.14. Thus, s = ± j1.13.

**c.** From part b,  = 1.13 rad/s.

**37. I-10 Instructor**

**Program:**

K=[0:0.001:200];

for i=1:length(K);

deng=conv([1 -4 8],[1 3]);

numg=[0 K(i) 2\*K(i) 0];

dent=numg+deng;

R=roots(dent);

A=real(R);

B=max(A);

if B<0

R

K=K(i)

break

end

end

**Computer response:**

R =

-4.0000

-0.0000 + 2.4495i

-0.0000 - 2.4495i

K =

5

**a.** From the computer response, (a) the range of K for stability is 0 < K < 5.

**b.** The system oscillates at K = 5 at a frequency of 2.4494 rad/s as seen from R, the poles of the closed-loop system.

**I-11 Instructor**

T(s) =

|  |  |  |  |
| --- | --- | --- | --- |
| s4 | 1 | - 3 | 2K-4 |
| s3 | 3 | K+3 | 0 |
| s2 | - | 2K-4 | 0 |
| s1 |  | 0 | 0 |
| s0 | 2K-4 | 0 | 0 |

For K < -33: 1 sign change; For –33 < K < -12: 1 sign change; For –12 < K < 0: 1 sign change; For

0 < K < 2: 3 sign changes; For K > 2: 2 sign changes. Therefore, K > 2 yields two right-half-plane

poles.

**I-12 Instructor**

The characteristic equation can be calculated using from Mason’s rule, namely

or

The corresponding Routh array is

|  |  |  |
| --- | --- | --- |
|  | 1 | 1 |
|  | 3 |  |
| s |  |  |
| 1 |  |  |

It can readily be seen that when the third row will be of zeros, the system will have 2 complex conjugate poles and a real pole.

**I-13 Instructor**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1 | K2 | 1 |
| s3 | K1 | 5 | 0 |
| s2 |  | 1 | 0 |
| s1 |  | 0 | 0 |
| s0 | 1 | 0 | 0 |

For stability, K1K2 > 5; K12 + 25 < 5K1K2 ; and K1 > 0 . Thus 0 < K12 < 5K1K2 - 25,   
or 0 < K1 < .

**I-14 Instructor**

|  |  |  |  |
| --- | --- | --- | --- |
| s4 | 1 | 1 | 1 |
| s3 | K1 | K2 | 0 |
| s2 |  | 1 | 0 |
| s1 |  | 0 | 0 |
| s0 | 1 | 0 | 0 |

For two j poles, K12 - K1K2 + K22 = 0. However, there are no real roots. Therefore, there is no relationship between K1 and K2 that will yield just two j poles.

**I-15 Instructor**



|  |  |  |
| --- | --- | --- |
| s3 | 1 | 1 |
| s2 | 6 | 2 |
| s |  |  |
| 1 | 2 |  |

No RHP roots



|  |  |  |
| --- | --- | --- |
| s3 | 1 | 2 |
| s2 | 6 | 2 |
| s |  |  |
| 1 | 2 |  |

No RHP roots



|  |  |  |
| --- | --- | --- |
| s3 | 1 | 1 |
| s2 | 4 | 4 |
| s | 8 |  |
| 1 | 4 |  |

Auxiliary equation  , no roots in RHP, but two roots in  axis.



|  |  |  |
| --- | --- | --- |
| s3 | 1 | 2 |
| s2 | 4 | 4 |
| s | 1 |  |
| 1 | 4 |  |

No RHP roots

The interval polynomial has no roots in the RHP.

**I-16 Instructor**

The characteristic equation for the system is  or . The system has two complex conjugate poles at . The arm will oscillate at a frequency rad/sec.

**I-17 Instructor**

Eigenvalues are the roots of the following equation:



Hence, eigenvalues are -3.2824, 1.9133, 6.3691. Therefore, 1 rhp, 2 lhp, 0 j.

**I-18 Instructor**

* + 1. For ,  or . The Routh array is

|  |  |  |
| --- | --- | --- |
|  | -1 |  |
|  | ~~0~~ -2 |  |
| 1 |  |  |

The auxiliary polynomial used in the second row is , that row is replaced with the coefficients of .

The first column has one sign change, so there is one root I the RHP, one in the LHP.

* + 1. For ,  or . The Routh array is

|  |  |  |  |
| --- | --- | --- | --- |
|  | 1 | 0 |  |
|  | ~~0~~ 4 | 0 | 0 |
|  | ~~0~~ ε |  |  |
|  |  |  |  |
| 1 |  |  |  |

The second row was originally a row of zeros, the auxiliary equation used was , so its coefficients were substituted with the coefficients of .

The first column in the array has two sign changes, so the polynomial has two roots in the RHP and two must be in the LHP.