**EE324 Control Systems Lab**

Problem sheet 10

Tarun Sai Goddu | 190070024

**Question 1**

Given the following state space system

X = AX+BU

Y = CX+DU

Part(i)

Taking A,B,C,D and T as follows

A = , B = , C= , D=2 , T =

Initially G(s) is:

G1 (s) = D + C(sI - A)^-1B

G1(s) = 2s3-6s2-35s+60/s3-6s2-25s+39

After transforming A, B and C, I obtained the same transfer function G1(s) = G2(s)

A = T^-1AT, B = T^-1B, C = CT

G2(s) = 2s3-6s2-35s+60/s3-6s2-25s+39

Part (ii)

Poles of G1s = G2s were calculated as: 8.419, -3.678 and 1.259.Eigenvalues of A were same as that of T^-1AT, which were: 8.419, -3.678 and 1.259. Therefore, here we -1 observe that eigen values of A are the poles of G1s = G2s .

Part (iii)

Taking a proper transfer function G s (from part(i)) as follows)

Gbp (s) = (s+1)(s+5)/(s+3)(s+2)

value of D is non-zero in case of biproper transfer functions, whereas in case of strictly proper transfer function, D = 0. This can be understood as below. I observed that:

lim s → ∞(D + C(sI – A)^-1 B) = D

And for a strictly proper transfer function, as degree of denominator is greater that numerator, hence

lim s → ∞G(s) = 0

lim s → ∞(D+C(sI-A)^-1B)=D

D = 0

**Scilab Code:**

clear;

clc;

s=poly(0,'s');

A = [1,2,5;

1,3,4;

6,1,2];

I = eye(3,3);

B = [1;

2;

1];

C = [1,1,3];

D = 0\*eye(1,1);

T = [1,0,0;

1,2,4;

3,0,2];

G1 = D + C\*(inv(s\*I-A))\*B;

evals1 = spec(A);

*// Checking G(s) after modifying A, B and C*

A = inv(T)\*A\*T;

B = inv(T)\*B;

C = C\*T;

G2 = D + C\*(inv(s\*I-A))\*B;

*// Eigenvalues of A*

evals2 = spec(A);

poles\_G = roots(G1.den);

*// D value*

G\_bp = ((s+1)\*(s+5))/((s+3)\*(s+2));

G\_sp = (s+1)/((s+3)\*(s+2));

sysGbp = syslin('c',G\_bp);

sysGsp = syslin('c',G\_sp);

[A1,B1,C1,D1] = abcd(sysGbp);

[A2,B2,C2,D2] = abcd(sysGsp);

**Question 2**

Given the transfer function G(s):

G(s) = (s+3)/(s^2+5s+4)

We get state space realization as follows:

X = AX+BU

Y = CX+DU

Since values in A,B,C and D were preferred to be integers and scilab's tf2ss was giving float values as entries of A,B,C, and D I switched to MATLAB.

MATLAB Code:

clear;

clc;

syms s;

s=tf('s');

G1 = (s+3)/((s+1)\*(s+4));

[n1,d1] = tfdata(G1,'v');

[A1,B1,C1,D1] = tf2ss(n1,d1)

G2 = (s+1)/(s^2+5\*s+4);

[n2,d2] = tfdata(G2,'v');

[A2,B2,C2,D2] = tf2ss(n2,d2)

**Question 3**

The Transfer function G(s) is written as :

G(s) = D+C(sI-A)-1B

Choosing A as follows:

A =

we observe that eigen values of A are {a1, a2, a3 } and (sI - A)-1

(sI - A)-1 =

Here we see that the term if the element in the kth row of B or kth column of C is , then the term 1/(s-ak) will vanish from the product in term

Hence pole ak will no longer be a pole of G(s).

Case 1: an entry in B is 0

A = , B = , C= , D=0

The transfer function obtained for above choice is:

G(s) = 8s-16/s^2-6s+5

which has poles at s = 1, 5. Thus the pole at s = 3 has been cancelled which is the second entry in diagonal matrix A. Corresponding to the 2 position B's entry was 0, hence the second entry in A (i.e 3) is no longer a pole of G(s).

Case2: an entry in C is 0

A = , B = , C= , D=0

The transfer function obtained for above choice is:

G(s) = 10s-38/s^2-8s+15

which has poles at s = 3, 5. Thus the pole at s = 1 has been cancelled which is the first entry in diagonal matrix A. Corresponding to the 1 position C's entry was 0, hence the first entry in A (i.e 1) is no longer a pole of G(s).

**Scilab Code:**

clear;

clc;

s=poly(0,'s');

A = [1,0,0;

0,3,0;

0,0,5];

I = eye(3,3);

B = [2;

0;

6];

C = [1,4,1];

D = 0\*eye(1,1);

G1 = D + C\*(inv(s\*I-A))\*B;

A = [1,0,0;

0,3,0;

0,0,5];

I = eye(3,3);

B = [2;

1;

6];

C = [0,4,1];

D = 0\*eye(1,1);

G2 = D + C\*(inv(s\*I-A))\*B;

**Question 4**

According to the given conditions we have and assuming D as 0 wlog:

A = , B = , C= , D=0

The expression of G(s) simplifies as:

G(s) = C(sI - A)-1 B

Here I observed that poles of the system are at s = a1 , a3 and a5 , if there's no cancellation from a zero factor in numerator. There can be 3 possible cases of diagonal entries getting repeated

Case i: If we have a1 = a3, here we have pole/zero cancellation when b2(a3 - a5 ) + b3a4 = 0 or a2 = 0

Case ii: If we have a3 = a5,.here we have pole/zero cancellation when c2 (a3 - a1 ) + c1a2 = 0 or a4 =0

Case iii: If we have a5 = a1 , here we have pole/zero cancellation when either a2 = 0 or a4 = 0 or both are zero

Taking an example from each of the above cases:

Case1:

A = , B = , C=

Poles and zeros of G(s) are obtained as: poles = {1, 1, 6} and zeros = {0.7727, 1}

Therefore, pole/zero cancellation happens as denominator and numerator have a common (s-1) factor.

Case2:

A = , B = , C=

Poles and zeros of G(s) are obtained as: poles = {1, 6, 6} and zeros = {0.5455, 6}

Therefore, pole/zero cancellation happens as denominator and numerator have a common (s-6) factor.

Case3:

A = , B = , C=

Poles and zeros of G(s) are obtained as:

poles = {2, 2, 3} and zeros = {2, 2.0455}

Therefore, pole/zero cancellation happens as denominator and numerator have a common (s-2) factor

MATLAB Code:

clear;

clc;

% Case i

A = [1, 0, 0; 0, 1, 4; 0, 0, 6];

B = [1; 3; 5];

C = [1, 2, 3];

D = 0;

[n1,d1] = ss2tf(A,B,C,D);

poles\_1 = roots(d1);

zeros\_1 = roots(n1);

% Case ii

A = [1, 5, 0; 0, 6, 0; 0, 0, 6];

B = [1; 3; 5];

C = [1, 2, 3];

D = 0;

[n2,d2] = ss2tf(A,B,C,D);

poles\_2 = roots(d2);

zeros\_2 = roots(n2);

% Case iii

A = [2, 5, 0; 0, 3, 0; 0, 0, 2];

B = [1; 3; 5];

C = [1, 2, 3];

D = 0;

[n3,d3] = ss2tf(A,B,C,D);

poles\_3 = roots(d3);

zeros\_3 = roots(n3);