ELL-700: LINEAR SYSTEMS THEORY

ASSIGNMENT – 2

CONTROLLER DESIGN

Ashwathram S

2023MEZ8467

PhD, Mechanical Engineering

The program takes in three matrices(A,B,C) and then performs the following functions:

- Checks if the system is input stabilizable.
- Checks if the system is output stabilizable.
- Finds the controllable and observable subspaces.
- Performs Controllable decomposition.
- Develops linear state feedback Gains for desired close loop poles.

```
function [reach,unreach,observ,unobs] = Stabilize(A,B,C)

n = size(A,1);
m = size(B,2);
p = size(C,1);

if(n<=0 || m<=0 || p<=0 || n<=max(m,p))
    disp("Invalid dimensions");
else
    %Finding Controllability matrix
    iter = A*B;
    con = [B iter];
    for i = 1:n-2
        iter = A*iter;
        con = [con iter];
    end

    %Finding observability matrix
    iter2 = C*A;
    obs = [C;iter2];
    for j = 1:n-2
        iter2 = iter2*A;
        obs = [obs;iter2];
```

```
    end

    reach = orth(con,1e-4);
    unobs = null(obs,1e-4);
    unreach = null(transpose(reach),1e-4);
    observ = null(transpose(unobs),1e-4);


end
end
```

The Stabilize function is used to find out the controllability and observability matrices.

Once the conditions on matrix dimensions are satisfied, we calculate the controllability and observability matrix. We can find the reachable subspace by finding the image of controllability matrix, and similarly find the unobservable subspace by finding the null-space of observability matrix.

In the first *for* loop, [B AB] is first initiated and A^2B, A^3B.....$A^{n-1}B$ are appended in each loop using *iter*.

In the second *for* loop for the observability matrix, [C;CA] is first initiated and CA^2,CA^3...CA^{n-1} are appended to each row in every loop using iter2.

Thus the controllability and the observability matrices are constructed using the *for* loops.

Once the controllability and the observability matrices are constructed, we use them to find the reachable, unreachable, observable and unobservable subspaces from R^n.

orth function from MATLAB gives the orthonormal basis vectors of the Range space or Image of controllability matrix. A tolerance of 10^{-4} was used to round off all the values which are lesser than the tolerance to 0. This was done after encountering an error which removed some of the vectors from the intersection subspace due to very minor differences.

Unobservable subspace is calculated using the *null* function which gives the null-space of observability matrix in terms of orthonormal vectors.

Then the unreachable and observable subspaces can be determined by taking the complement of the reachable and unobservable subspaces. The relation between the null-space and complement is used to determine the complement subspace.

```
function flagC = isStable(A,B,C)

n = size(A,1);
m = size(B,2);
p = size(C,1);

[V,D] = eig(transpose(A));
```

```
for i = 1:size(V,2)
    BTv = round(transpose(B)*V(:,i),5);
    if ~all(BTv) && real(D(i,i))>0
        flagC = 1;
        return
    end
end
flagC = 0;

end

function flagD = isDetect(A,B,C)

n = size(A,1);
m = size(B,2);
p = size(C,1);

[V,D] = eig(A);

for i = 1:size(V,2)
    Cv = round((C)*V(:,i),5);
    if Cv ==  zeros(p,1) && real(D(i,i))>0
        flagD = 1;
        return
    end
end
flagD = 0;

end
```

Functions flagC and flagD are used to check the input output stability of the system.

In flagC, function calculates the eigenvalues and vectors of transpose(A). In the for loop, it checks if any of the eigenvectors of transpose(A) is in the kernel of transpose(B) and if it is, whether it has a positive real eigen value. If this is true, the system is not stabilizable and flag variable is made 1.

Similarly in flagD, function calculates the eigenvalues and vectors of A, and checks if it lies in the kernel of C and if it does, whether it has positive real eigen value. If it does, then the system is not output stabilizable.

```
%Ackermann function
function K = Ackermann(Ac,Bc,poles)

n = size(Ac,1);
m = size(Bc,2);
In = eye(n);
del_A = eye(n);
%Forming characteristic equation
```

```
for i=1:n
    del_A = del_A*(Ac-(poles(i)*In));
end
const = zeros(1,n);
const(1,n) = 1;

%Forming controllability matrix
iter = Ac*Bc;
    con = [Bc iter];
    for i = 1:n-2
        iter = Ac*iter;
        con = [con iter];
    end
K = const*(con\del_A);       % For SISO K = [0 0 ...1]*inv(con)*A_desired

end
```

The Ackermann function is used to derive the state feedback gain values for the desired closed loop poles. The desired close loop poles are calculated in the main function and is passed on to this function as poles. The desired characteristic equation is formed using cayley Hamilton theorem. Controllability matrix for the decomposed controllable A matrix is formed and it is always invertable. Finally the gain is calculated using formula

K = [0 0 .....1]*$(con)^{-1}$*(A_desired)

Main Function:

```
A = [-7 1 15;-2 0 5;-2 0 4];
B = [65;-5;30];
C = [-1 2 0];

n = size(A,1);
m = size(B,2);
p = size(C,1);

[reach,unreach,observ,unobserv] = Stabilize(A,B,C);

isstab = isStable(A,B,C);

if isstab == 1
    disp("System is unstabilizable based on input conditions");
else
    %Controllable Decomposition
    Tcuc = [reach unreach];
    A_d = Tcuc\(A*Tcuc);
    B_d = Tcuc\B;
    A_dc = A_d(1:size(reach,2),1:size(reach,2))
    B_dc = B_d(1:size(reach,2),:)

    isdetect = isDetect(A,B,C);
```

```
    if isdetect == 1
        disp("System is unstabilizable based on output conditions");
    else
        %For SISO
        if m==1 && p==1
        for i = 1:size(A_dc,1)
            poles_d(i) = input("Enter Desired eigen values:");
        end
        %Calling Ackermann function
        disp("State Feedback Gain: ") % u = -Kx
        K = Ackermann(A_dc,B_dc,poles_d)
        disp("Modified eigen values: ")
        eig(A_dc-(B_dc*K))
        %For MIMO
        else

        end

    end
end
```

The controllability decomposition is done in the main function. The transformation matrix is formed using the controllable and uncontrollable vectors, and the modified controllable A matrix and B matrices are formed, which are then sent to Ackermann to find the required state feedback gains, since the uncontrollable mode's poles cannot be changed.

Sample Output for fully controllable system: