AmirKabir University of Technology (Tehran Polytechnic)

Multi-Variable Control Systems - Assignment 02 **Mohammad Azimi - 402123100**

Q1-

Note: First Run the corresponing `.m` file to generate all the required variables for this script.

System Description

Let's break down each state of the linear UAV system:

- **Roll Angle Rate (x1):**

This state represents the change rate of the roll angle. Roll angle is the angle of rotation about the longitudinal axis of the aircraft, which is the axis running from the nose to the tail. The roll angle rate describes how quickly the aircraft is rotating around this axis.

- **Pitch Angle Rate (x2):**

Similar to roll angle rate, pitch angle rate represents the change rate of the pitch angle. Pitch angle is the angle of rotation about the lateral axis of the aircraft, which is the axis running from wingtip to wingtip. The pitch angle rate describes how quickly the aircraft is rotating around this axis.

- **Yaw Angle Rate (x3):**

Yaw angle rate represents the change rate of the yaw angle. Yaw angle is the angle of rotation about the vertical axis of the aircraft, which is the axis running from top to bottom through the center of the aircraft. The yaw angle rate describes how quickly the aircraft is rotating around this axis.

- **Forward Velocity along Body X (x4):**

This state represents the velocity component of the UAV in the forward direction along its body frame's x-axis. It indicates how fast the UAV is moving in the direction it is facing.

- **Vertical Velocity along Body Z (x5):**

Similar to forward velocity, this state represents the velocity component of the UAV in the upward direction along its body frame's z-axis. It indicates how fast the UAV is ascending or descending vertically.

- **UAV Altitude (x6):**

This state represents the altitude of the UAV, which is the vertical distance between the UAV and the ground or a reference point. It indicates how high the UAV is flying above the ground or sea level.

- **Roll Angle (x7):**

Roll angle represents the angle of rotation about the longitudinal axis of the aircraft at a given moment. It describes the orientation of the UAV with respect to the horizon in terms of roll.

- **Yaw Angle (x8):**

Yaw angle represents the angle of rotation about the vertical axis of the aircraft at a given moment. It describes the orientation of the UAV with respect to the horizon in terms of yaw.

- **Pitch Angle (x9):**

Pitch angle represents the angle of rotation about the lateral axis of the aircraft at a given moment. It describes the orientation of the UAV with respect to the horizon in terms of pitch.

- **Lateral Velocity along Body Y (x10):**

This state represents the velocity component of the UAV in the sideways direction along its body frame's y-axis. It indicates how fast the UAV is moving sideways.

These states collectively describe the dynamics and kinematics of the UAV in its flight. The control inputs (u1, u2, u3) are used to manipulate these states and control the behavior of the UAV in flight.

Now let’s have glance at what the outputs of the system are supposed.

- **y1 (Output):**

This output is the same as the roll angle (x7). It represents the orientation of the UAV with respect to the horizon in terms of roll.

- **y2 (Output):**

This output is the same as the yaw angle (x8). It represents the orientation of the UAV with respect to the horizon in terms of yaw.

- **y3 (Output):**

This output is the same as the pitch angle (x9). It represents the orientation of the UAV with respect to the horizon in terms of pitch.

These outputs provide information about the orientation of the UAV in flight, which can be useful for monitoring and controlling its trajectory and stability. They directly reflect the corresponding states of the system, providing observable variables for analysis and feedback control.

The state equations of the system can be represented in matrix form as:

$$\dot{x} = Ax + Bu$$

where:

$\dot{x}$ represents the derivative of the state vector with respect to time.

A is the state matrix, which describes how the states evolve over time,

B is the input matrix, which describes how the control inputs affect the system dynamics,

u represents the control input vector.

In this system:

- **State Matrix (A):**

The matrix is a 10x10 matrix that defines the dynamics of the UAV system. Each element of this matrix represents the influence of one state on the rate of change of another state. For example, element represents how state influences the rate of change of state . In other words, the non-zero elements of this matrix indicate the interdependencies among different states of the system.

- **Input Matrix (B):**

The matrix is a 10x3 matrix that defines how the control inputs affect the dynamics of the system. Each column corresponds to one control input, and each row corresponds to one state. For example, element represents how control input affects the rate of change of state .

The state equations encapsulate the dynamic behavior of the UAV system, describing how the system's states evolve over time in response to control inputs and external influences. These equations are fundamental for understanding and modeling the behavior of the UAV in various flight conditions and for designing control strategies to achieve desired performance objectives.

Transfer functions and Char Equations

```
for o=1:numOutputs
    for i=1:numInputs
        fprintf("Transfer Function for Output_%d/Input_%d}= \n", o,i)
        eval(simplify(G(o,i)))
        fprintf("-------------------------------------------------------------
\n")
    end
end
```

```
Transfer Function for Output_1/Input_1}=
ans = 0
-------------------------------------------------------------
Transfer Function for Output_1/Input_2}=
ans =
```

$$\frac{5004822650000000\,s^2+19302723374480500\,s+15127711141163841536}{50000000000000\,s^4+2887000000000000\,s^3+152235817000000000\,s^2+287013076421095000\,s-67941684155290}$$

```
-------------------------------------------------------------
Transfer Function for Output_1/Input_3}=
ans =
```

$$\frac{522965000000000\,s^2+1645331858110000\,s+1139700250389880000}{5000000000000\,s^4+288700000000000\,s^3+15223581700000000\,s^2+28701307642109500\,s-6794168415529}$$

```
-------------------------------------------------------------
Transfer Function for Output_2/Input_1}=
ans = 0
-------------------------------------------------------------
Transfer Function for Output_2/Input_2}=
ans =
```

$$\frac{10510500000000\,s^3+9019895885000\,s^2+6987883141168930\,s+5961033410832500}{s\;(5000000000000\,s^4+288700000000000\,s^3+15223581700000000\,s^2+28701307642109500\,s-6794168415529)}$$

```
-------------------------------------------------------------
Transfer Function for Output_2/Input_3}=
ans =
```

$$\frac{500500000000000\,s^3+1394019627000000\,s^2+6127122087086000\,s+4490753767500000}{s\;(5000000000000\,s^4+288700000000000\,s^3+15223581700000000\,s^2+28701307642109500\,s-6794168415529)}$$

```
-------------------------------------------------------------
Transfer Function for Output_3/Input_1}=
ans =
```

$$\frac{1238489897526886400000000000000\,s^2+12499381603696528850944000000000\,s+1288194169800136471}{1125899906842624000000000000\,s^4+111953305545941964554240000000\,s^3+11056453391533371630377815244800\,s^2+1416053177597400551}$$

```
-------------------------------------------------------------
Transfer Function for Output_3/Input_2}=
ans = 0
-------------------------------------------------------------
Transfer Function for Output_3/Input_3}=
ans = 0
-------------------------------------------------------------
```

Eigenvalues of matrix A are obtained as the following:

```
eig_A
```

```
eig_A = 10×1 complex
             0 +          0i
             0 +          0i
       -1.9421 +     55.078i
       -1.9421 -     55.078i
       -1.8901 +          0i
    0.00023669 +          0i
       -4.9653 +     98.971i
       -4.9653 -     98.971i
    -0.0064037 +   0.014972i
    -0.0064037 -   0.014972i
```

Thus, the characterisic equation of the open-loop system derives using `det(SI - A)`:

```
eval(det(s*eye(numStates) - A))
```

```
ans =
```

$$\frac{(5000000000000\,s^6 + 28870000000000\,s^5 + 15223581700000000\,s^4 + 28701307642109500\,s^3 - 6794168415529\,s^2)\ (11258999068426240000000000000}{56294995342131}$$

Considering a **Negative Unity Feedback** (**-1 feedback**) through all of the system states will lead into the following eigenvalues and charactristic equation.

```
eig_A_prime
```

```
eig_A_prime = 10×1 complex
10^3 ×
  -0.7105 + 1.4075i
  -0.7105 - 1.4075i
  -0.0019 + 0.0484i
  -0.0019 - 0.0484i
  -0.0023 + 0.0000i
  -0.0002 + 0.0011i
  -0.0002 - 0.0011i
   0.0000 + 0.0000i
  -0.0000 + 0.0000i
  -0.0000 + 0.0000i
```

And the characteristic equation for the colsed-loop system (`det(SI - A + BK)`) :

```
det(s*eye(numStates) - A_prime)
```

```
ans =
```

$$s^{10} + \frac{1004595768242435931733\,s^9}{70368744177664000} + \frac{49471504961559905217343212811441253970 7\,s^8}{19807040628566084398385987584 0000} + \frac{1671261387746435908539475589624138180330607949563376}{85070591730234615865843651857942052864000000000}$$

How inputs would affect outputs

Based on the obtained transfer functions for the system, the following results are concluded:

Input 1 affects on output 3.

Input 2 affects on output 1 and 2.

Input 3 affects on output 1 and 2.

So, (**Note**: G_{ji} : Transfer Function respecting Y_j and U_i $i.e.$ Output_j and Input_i)

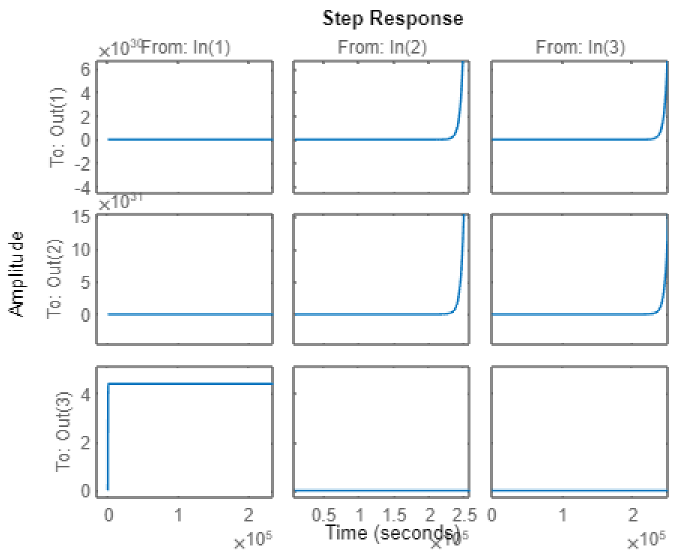
- $Y_1 = f(U_2, U_3) = G_{12}U_2 + G_{13}U_3$
- $Y_2 = f(U_2, U_3) = G_{22}U_2 + G_{23}U_3$

- $Y_3 = f(U_1) = G_{32}U_1$

The afrementioned results could be observed through `step` responses of the system in the following figure.

```
step(sys_ss)
```

```
Warning: The "hold" command is not supported for multi-axes plots. Clearing the previous
plot.
```

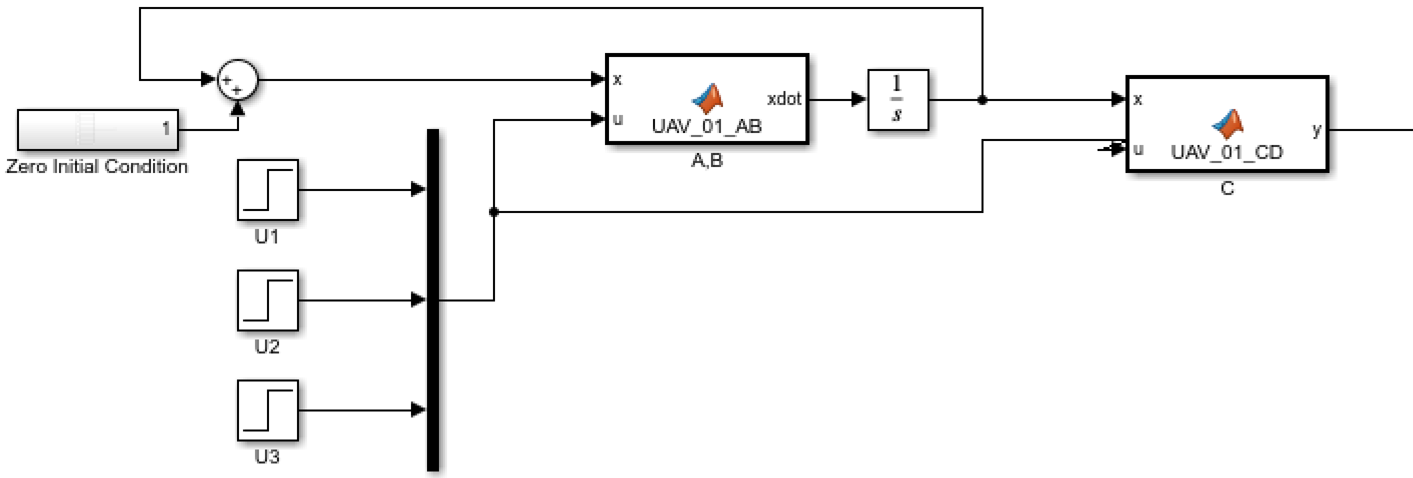


```
open("Q1_UAV.slx")
out = sim("Q1_UAV.slx")
```

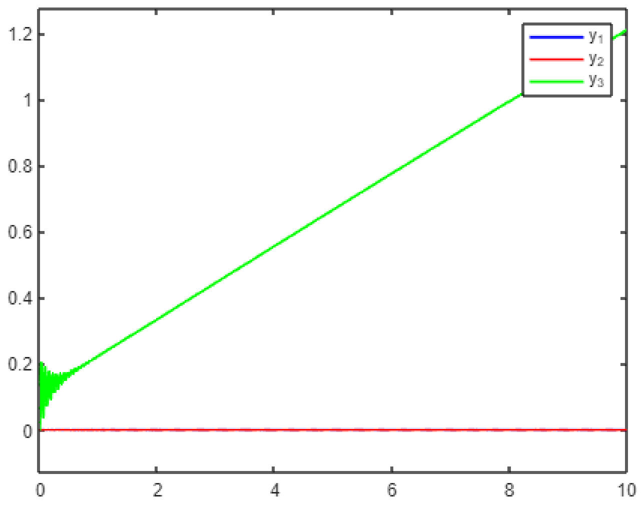
```
out = 
  Simulink.SimulationOutput:

                   tout: [462x1 double] 
                      y: [1x1 timeseries] 

     SimulationMetadata: [1x1 Simulink.SimulationMetadata] 
           ErrorMessage: [0x0 char] 
```

System Simulation in SimuLink

We could implement the system in Simulink as follows:



```
reshaped = reshape(out.y.Data , [3,size(out.y.Data,3)]);
figure
plot(out.tout, reshaped(1,:), Color="blue", LineWidth=1.5)
hold on
plot(out.tout, reshaped(2,:), Color="red", LineWidth=1.25)
plot(out.tout, reshaped(3,:), Color="green", LineWidth=1.5)
legend("y_1", "y_2", "y_3")
```



You can run the system under different inputs and initial conditions and study system behaviour.

Controller Design

Since the eigenvalues of the system include instablity, we may want to desing a controller for the system in order to place its poles where we desire. The required condition for such design is that the Controllability Matrix of the system (`U=ctrb(A,B)`) is full rank.

```
[rankU , numStates]
```

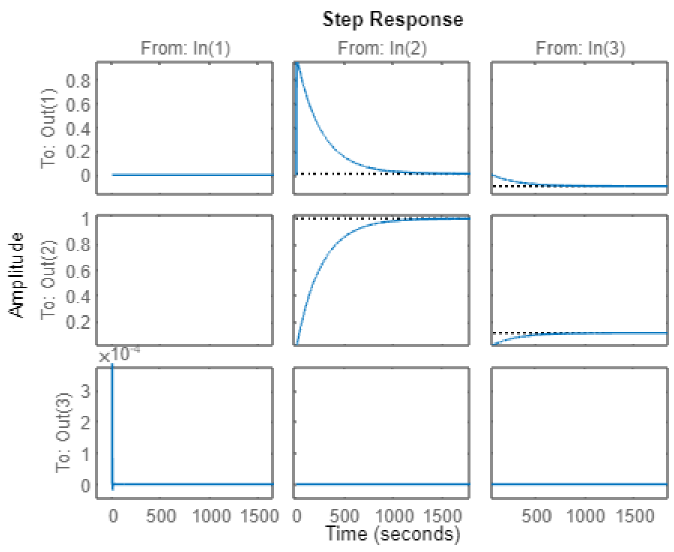
```
ans = 1×2
      4     10
```

According the inequality $4 \neq 10$, the system is not controllable and its poles can not be placed at any desired location. Considering the obtained poles for the system through unity feedback, unity feedback can not at least stabilize the system as it yet still remains one positive poles settled very close to the origin.

All these conditions means that although we can not place poles where we desire but some controllers may be able to place them where at least they are stable. To find such controller, we could take advantage of the command `lqr` and derive an optimal gain matrix `K`, all thanks to MATLAB.

Let's try this out and see if it is able to place system poles where they are stable or not.

```
step(sys_lqr)
```



To conclude, we observed that a negative unity feedback can not stabilize the system. Hence, we used `lqr` MATALB command to find an optimal `K` for the feedback gain so that all poles are placed somewhere on the left-side of the s-plane i.e. their real part are negative, since rank of the controllability matrix is not full ($\rho(U) = 4 < 10$) and the system is not fully controllable.

System Type

We know that the Type of a system equals to the number of poles at the origin. So let's recall the system poles which are the eigenvalues of the matrix `A`.

```
eig_A
```

```
eig_A = 10×1 complex
    0.0000 + 0.0000i
    0.0000 + 0.0000i
   -1.9421 +55.0782i
   -1.9421 -55.0782i
   -1.8901 + 0.0000i
    0.0002 + 0.0000i
   -4.9653 +98.9713i
   -4.9653 -98.9713i
   -0.0064 + 0.0150i
   -0.0064 - 0.0150i
```

Counting poles at the origin:

```
length(find(eig_A == 0))
```

```
ans = 2
```

So the maximum SystemType of which the system could behave is 2 and since the system is MIMO, Type of the system could vary for different transfer functions of the system due to the zero-pole cancelation throughout system realization.

To study each transfer function (**those that are nonzero**), we need to checkout their denominators.

```
for o=1:numOutputs
    for i=1:numInputs
        fprintf("Transfer Function for Output_%d/Input_%d}= \n", o,i)
        eval(simplify(G(o,i)))
        fprintf("-------------------------------------------------------------
\n")
```

```
    end
end
```

```
Transfer Function for Output_1/Input_1}=
ans = 0
-------------------------------------------------------------
Transfer Function for Output_1/Input_2}=
ans =
```

$$\frac{5004822650000000\,s^2 + 19302723374480500\,s + 15127711141163841536}{50000000000000\,s^4 + 288700000000000\,s^3 + 15223581700000000\,s^2 + 287013076421095000\,s - 67941684155290}$$

```
-------------------------------------------------------------
Transfer Function for Output_1/Input_3}=
ans =
```

$$\frac{522965000000000\,s^2 + 1645331858110000\,s + 1139700250389880000}{5000000000000\,s^4 + 28870000000000\,s^3 + 1522358170000000\,s^2 + 28701307642109500\,s - 6794168415529}$$

```
-------------------------------------------------------------
Transfer Function for Output_2/Input_1}=
ans = 0
-------------------------------------------------------------
Transfer Function for Output_2/Input_2}=
ans =
```

$$\frac{10510500000000\,s^3 + 9019895885000\,s^2 + 6987883141168930\,s + 5961033410832500}{s\,(5000000000000\,s^4 + 28870000000000\,s^3 + 1522358170000000\,s^2 + 28701307642109500\,s - 6794168415529)}$$

```
-------------------------------------------------------------
Transfer Function for Output_2/Input_3}=
ans =
```

$$\frac{500500000000000\,s^3 + 1394019627000000\,s^2 + 6127122087086000\,s + 4490753767500000}{s\,(5000000000000\,s^4 + 28870000000000\,s^3 + 1522358170000000\,s^2 + 28701307642109500\,s - 6794168415529)}$$

```
-------------------------------------------------------------
Transfer Function for Output_3/Input_1}=
ans =
```

$$\frac{12384898975268864000000000000000\,s^2 + 12499381603696528850944000000000\,s + 128819416980013647}{1125899906842624000000000000\,s^4 + 111953305545941964554240000000\,s^3 + 110564533915333716303778152448000\,s^2 + 14160531775974005511}$$

```
-------------------------------------------------------------
Transfer Function for Output_3/Input_2}=
ans = 0
-------------------------------------------------------------
Transfer Function for Output_3/Input_3}=
ans = 0
-------------------------------------------------------------
```

It is seen that the system Type is never 2 as we had guessed in the first step. This is due to the zero-pole cancelation. Output 1 with respect to both Input 2 and 3 has the system Type 0 since there is no pole at the origin. Also output 3 has the Type 0 respecting Input 1. On the other hand, Output 2 has the system Type 1 under Input 2 and 3 due to its single pole at the origin.

Singular Values Decomposition

We know that singular values of a matrix `A` are the square roots of the eigenvalues of $A^T A$ or AA^T. in MATLAB, one could use the command `svd` to obtain these values. Remeber that the correpending decomposition for matrix `A` is as follows:

$$A = \mathrm{USV}^T$$

```
UU
```

```
UU = 10×10
   -0.0000    0.0000    0.0003    0.0000    0.8827   -0.0697   -0.0000 ⋯
   -0.0000   -0.0036         0   -0.9998         0   -0.0000    0.0189
   -0.0000    0.0000    0.0011   -0.0000    0.1390    0.9834    0.0000
    0.0039   -0.0047   -0.0000   -0.0188    0.0000    0.0000   -0.9998
    0.0127    0.9999    0.0000   -0.0037   -0.0000    0.0000   -0.0045
   -0.9999    0.0127   -0.0000   -0.0001    0.0000   -0.0000   -0.0039
   -0.0000   -0.0000   -0.0000   -0.0000   -0.4490    0.1674   -0.0000
   -0.0000    0.0000   -0.0004   -0.0000   -0.0021    0.0004   -0.0000
    0.0000    0.0004    0.0000    0.0001   -0.0000   -0.0000    0.0021
   -0.0000   -0.0000    1.0000    0.0000   -0.0004   -0.0011   -0.0000
```

```
VV
```

```
VV = 10×10
   -0.0000   -0.0000    0.0046   -0.0000   -0.9794    0.2021    0.0000 ⋯
    0.0126    0.9999         0    0.0004   -0.0000   -0.0000    0.0000
   -0.0000   -0.0000   -1.0000   -0.0000   -0.0046    0.0005   -0.0000
   -0.0000   -0.0000   -0.0000   -0.0458   -0.0000   -0.0000    0.9990
    0.0004   -0.0004   -0.0000    0.9990    0.0000    0.0000    0.0458
         0         0         0   -0.0000    0.0000    0.0000   -0.0000
    0.0000   -0.0000    0.0039   -0.0000   -0.0018   -0.0089   -0.0000
         0         0         0         0         0         0   -0.0000
   -0.9999    0.0126   -0.0000    0.0004    0.0000   -0.0000   -0.0000
    0.0000    0.0000   -0.0004   -0.0000    0.2021    0.9793   -0.0000
```

```
SS
```

```
SS = 10×10
10^3 ×
    2.5030         0         0         0         0         0         0 ⋯
         0    2.4870         0         0         0         0         0
         0         0    2.4840         0         0         0         0
         0         0         0    0.0040         0         0         0
         0         0         0         0    0.0022         0         0
         0         0         0         0         0    0.0012         0
         0         0         0         0         0         0    0.0000
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0
```

So the singular values of the matrix `A` we have in descending order is:

```
diag(SS)
```

```
ans = 10×1
10^3 ×
    2.5030
    2.4870
    2.4840
    0.0040
    0.0022
    0.0012
    0.0000
    0.0000
    0.0000
    0.0000
```

BIBO stability

As mentioned earlier, eigenvalues of the system dynamics matrix `A` show that the system is not **Bounded-Input-Bounded_Output**. This is also demonstrated in the section that system is simulated in Simulink. There, you could find a plot that with a single step input in one of the three inputs, one of the output is wildly diverging. Hence, the system we have is not **BIBO**.

Model Reduction - Method 1

Following the book instructions to reduce the model using method one "Model Separation" provides us with following results.

```
Az, Bz, Cz
```

```
Az = 10×10 complex
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i ⋯
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i  -1.9421 +55.0782i   0.0000 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i  -1.9421 -
55.0782i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
Bz = 10×3 complex
10^4 ×
   1.0719 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i  -0.0877 + 0.0000i  -0.0661 + 0.0000i
   0.0000 + 0.0000i  -0.0002 + 0.0037i  -0.0007 + 0.2245i
   0.0000 + 0.0000i  -0.0002 - 0.0037i  -0.0007 - 0.2245i
   0.0000 + 0.0000i   0.0113 + 0.0000i   0.0085 + 0.0000i
   0.0000 + 0.0000i   0.0879 + 0.0000i   0.0662 + 0.0000i
   0.0005 - 1.3832i   0.0000 + 0.0000i   0.0000 + 0.0000i
   0.0005 + 1.3832i   0.0000 + 0.0000i   0.0000 + 0.0000i
  -0.5359 - 0.7038i   0.0000 + 0.0000i   0.0000 + 0.0000i
  -0.5359 + 0.7038i   0.0000 + 0.0000i   0.0000 + 0.0000i
Cz = 3×10 complex
   0.0000 + 0.0000i   0.0000 + 0.0000i  -0.0001 - 0.0000i  -0.0001 +
0.0000i ⋯
   0.0000 + 0.0000i   1.0000 + 0.0000i  -0.0004 + 0.0000i  -0.0004 -
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
```

Eigenvalues of matrix `A` had been obtained earlier:

```
eig_A
```

```
eig_A = 10×1 complex
   0.0000 + 0.0000i
   0.0000 + 0.0000i
  -1.9421 +55.0782i
  -1.9421 -55.0782i
  -1.8901 + 0.0000i
   0.0002 + 0.0000i
  -4.9653 +98.9713i
  -4.9653 -98.9713i
  -0.0064 + 0.0150i
  -0.0064 - 0.0150i
```

We could see that none of the system modes could be assumed as nondominant. The most probable modes to be dropped off the system are -4.9655 ± 98.9715. So we are expecting to reduce the system from $n = 10$ to $q = 8$.

According to the following equations, we could separate the system dynamics.

$$\begin{bmatrix} \dot{\underline{z}}_1 \\ \dot{\underline{z}}_2 \end{bmatrix} = \begin{bmatrix} A_{1z} & 0 \\ 0 & A_{2z} \end{bmatrix} \begin{bmatrix} \underline{z}_1 \\ \underline{z}_2 \end{bmatrix} + \begin{bmatrix} B_{1z} \\ B_{2z} \end{bmatrix} \underline{u}$$

$$y = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \end{bmatrix} + D\underline{u}$$

And a whole bunch of steps will lead to:

$$A_r = T_{11} A_{1z} T_{11}^{-1}$$
$$B_r = T_{11} \left(A_{1z} T_{11}^{-1} T_{12} A_{2z}^{-1} B_{2z} + B_{1z} \right)$$
$$F = T_{21} T_{11}^{-1}$$
$$E = \left(T_{21} T_{11}^{-1} T_{12} - T_{22} \right) A_{2z}^{-1} B_{2z}$$
$$C_r = C_1 + C_2 F$$
$$D_r = C_2 E + D$$

where they all present:

$$\dot{\underline{x}}_1 = A_r \underline{x}_1 + B_r \underline{u}$$
$$\begin{bmatrix} \underline{x}_2 \\ \hline \underline{y} \end{bmatrix} = \begin{bmatrix} F \\ \hline C_r \end{bmatrix} \underline{x}_1 + \begin{bmatrix} E \\ \hline D_r \end{bmatrix} \underline{u}$$

However, in our case, T_{11} is turned out to be SINGULAR, hence noninvertible.

```
inv(T11)
```

```
Warning: Matrix is singular to working precision.
ans = 8×8
    Inf    Inf    Inf    Inf    Inf    Inf    Inf    Inf
    Inf    Inf    Inf    Inf    Inf    Inf    Inf    Inf
    Inf    Inf    Inf    Inf    Inf    Inf    Inf    Inf
    Inf    Inf    Inf    Inf    Inf    Inf    Inf    Inf
    Inf    Inf    Inf    Inf    Inf    Inf    Inf    Inf
    Inf    Inf    Inf    Inf    Inf    Inf    Inf    Inf
    Inf    Inf    Inf    Inf    Inf    Inf    Inf    Inf
    Inf    Inf    Inf    Inf    Inf    Inf    Inf    Inf
```

Conclusively, the system is **irridducible** (at least by method 1).

Q2-

Paper DOI: 10.4172/2168-9695.1000126

Note: First Run the corresponing `.m` file to generate all the required variables for this script.

System Description

Longitudinal State Variables:

- **Longitudinal velocity (u)**: The velocity component along the longitudinal axis of the UAV.
- **Vertical velocity (w)**: The velocity component along the vertical axis of the UAV.
- **Pitch rate (q):** The rate of change of pitch angle of the UAV.
- **Altitude (h):** The altitude or height of the UAV above a reference point.
- **Pitch angle (θ):** The angle between the longitudinal axis of the UAV and the horizontal plane.

Lateral State Variables:

- **Lateral velocity (v):** The velocity component along the lateral axis of the UAV.
- **Roll rate (p):** The rate of change of roll angle of the UAV.
- **Yaw angle (ψ):** The angle between the heading direction of the UAV and a reference direction.
- **Yaw rate (r):** The rate of change of yaw angle of the UAV.
- **Bank angle (φ):** The angle between the lateral axis of the UAV and the horizontal plane.

Longitudinal Input Variables:

- **Elevator deflection (δe):** The control input that adjusts the pitch angle of the UAV by deflecting the elevator control surface.
- **Thrust control (δt):** The control input that regulates the thrust or engine power of the UAV.

Lateral Input Variables:

- **Aileron deflection (δa):** The control input that controls the roll motion of the UAV by deflecting the aileron control surface.
- **Rudder deflection (δr):** The control input that influences the yaw motion of the UAV by deflecting the rudder control surface.

Longitudinal Output Variables:

- **Longitudinal velocity (v):** The velocity component along the longitudinal axis of the UAV.
- **Angle of attack (α):** The angle between the relative airflow and the chord line of the UAV's wing.
- **Pitch rate (q):** The rate of change of pitch angle of the UAV.
- **Pitch angle (θ):** The angle between the longitudinal axis of the UAV and the horizontal plane.
- **Altitude (h):** The altitude or height of the UAV above a reference point.

Lateral Output Variables:

- **Lateral velocity (v):** The velocity component along the lateral axis of the UAV.
- **Bank angle (φ):** The angle between the lateral axis of the UAV and the horizontal plane.
- **Yaw angle (ψ):** The angle between the heading direction of the UAV and a reference direction.
- **Yaw rate (r):** The rate of change of yaw angle of the UAV.

The state equations of the system can be represented in matrix form as:

$$\dot{x} = Ax + Bu$$

where:

$\dot{x}$ represents the derivative of the state vector with respect to time.

A is the state matrix, which describes how the states evolve over time,

B is the input matrix, which describes how the control inputs affect the system dynamics,

u represents the control input vector.

In this system:

- **State Matrix (A):**

The matrix is a 10x10 matrix that defines the dynamics of the UAV system. Each element of this matrix represents the influence of one state on the rate of change of another state. For example, element represents how state influences the rate of change of state . In other words, the non-zero elements of this matrix indicate the interdependencies among different states of the system.

- **Input Matrix (B):**

The matrix is a 10x3 matrix that defines how the control inputs affect the dynamics of the system. Each column corresponds to one control input, and each row corresponds to one state. For example, element represents how control input affects the rate of change of state .

The state equations encapsulate the dynamic behavior of the UAV system, describing how the system's states evolve over time in response to control inputs and external influences. These equations are fundamental for understanding and modeling the behavior of the UAV in various flight conditions and for designing control strategies to achieve desired performance objectives.

Note: The longitudonal and lateral dynamics of the system are coupled together with the following structure:

$$\textbf{coupled dynamics} \cong \begin{bmatrix} \textbf{longitudinal dynamics} & 0 \\ 0 & \textbf{lateral dynamics} \end{bmatrix}$$

Transfer functions and Char Equations

```
for o=1:numOutputs
    for i=1:numInputs
        fprintf("Transfer Function for Output_%d/Input_%d}= \n", o,i)
        eval(simplify(G(o,i)))
        fprintf("-------------------------------------------------------------
\n")
    end
end
```

```
Transfer Function for Output_1/Input_1}=
ans =
```

$$\frac{103931830741478864956654892319928043438080000000000\, s^4 + 31214694895185945659911599744109768636061719}{3245185536584267267831560205762560000000000000000000\, s^5 + 7776892427292001451322264139497629286400000000000000\, s^4 + 81565903709617}$$

```
-------------------------------------------------------------
Transfer Function for Output_1/Input_2}=
ans = 0
-------------------------------------------------------------
Transfer Function for Output_1/Input_3}=
ans = 0
-------------------------------------------------------------
Transfer Function for Output_1/Input_4}=
ans = 0
--------------------------------------------------------------
Transfer Function for Output_2/Input_1}=
ans =
```

$$-\frac{24063589036266998083473731407925626961715544282732953\,6\,s^4+64795178390197296510302184839924074924275532142}{16225927682921336339157801028812800000000000000000\,s^5+38884462136460007256611320697488146432000000000000000\,s^4+407829518548}$$

```
--------------------------------------------------------------
Transfer Function for Output_2/Input_2}=
ans = 0
--------------------------------------------------------------
Transfer Function for Output_2/Input_3}=
ans = 0
--------------------------------------------------------------
Transfer Function for Output_2/Input_4}=
ans = 0
--------------------------------------------------------------
Transfer Function for Output_3/Input_1}=
ans =
```

$$-\frac{5\,s\,(108470326560329133427269899877613568000000000000\,s^3+867875235916513478 68}{405648192073033408478945025720320000000000000\,s^5+9721115534115001814152830174372036608000000000\,s^4+1019573796370214993115 58}$$

```
--------------------------------------------------------------
Transfer Function for Output_3/Input_2}=
ans = 0
--------------------------------------------------------------
Transfer Function for Output_3/Input_3}=
ans = 0
--------------------------------------------------------------
Transfer Function for Output_3/Input_4}=
ans = 0
--------------------------------------------------------------
Transfer Function for Output_4/Input_1}=
ans =
```

$$-\frac{542351632801645667136349499388067840000000000000\,s^3+4339376179582567393411 1}{405648192073033408478945025720320000000000000\,s^5+9721115534115001814152830174372036608000000000\,s^4+1019573796370214993115 58}$$

```
--------------------------------------------------------------
Transfer Function for Output_4/Input_2}=
ans = 0
--------------------------------------------------------------
Transfer Function for Output_4/Input_3}=
ans = 0
--------------------------------------------------------------
Transfer Function for Output_4/Input_4}=
ans = 0
--------------------------------------------------------------
Transfer Function for Output_5/Input_1}=
ans =
```

$$-\frac{-1381309892768376202856928228845491218022400000000\,s^3+623080580504513465254}{50706024009129176059868128215040000000000000000\,s^5+12151394417643752267691037717965045760000000000000\,s^4+1274467245462768741}$$

```
--------------------------------------------------------------
Transfer Function for Output_5/Input_2}=
ans = 0
--------------------------------------------------------------
Transfer Function for Output_5/Input_3}=
ans = 0
--------------------------------------------------------------
Transfer Function for Output_5/Input_4}=
ans = 0
--------------------------------------------------------------
Transfer Function for Output_6/Input_1}=
ans = 0
--------------------------------------------------------------
Transfer Function for Output_6/Input_2}=
ans = 0
--------------------------------------------------------------
Transfer Function for Output_6/Input_3}=
```

ans =

$$-\frac{4140410611795277515421107229245658922286495825920000000\,s^2+45449}{21267647932558653966460912964485513216000000000000000\,s^4+41977232783366968852841891572782926565212160000000000000\,s^3+156677}$$

Transfer Function for Output_6/Input_4}=

ans =

$$\frac{82907576231077704243511727935345535985123328000000000000\,s^3+23072973601144813699943505680738230}{265845599156983174580761412056068915200000000000000000\,s^4+524715409792087110660523644659786582065152000000000000000\,s^3+1958}$$

Transfer Function for Output_7/Input_1}=

ans = 0

Transfer Function for Output_7/Input_2}=

ans = 0

Transfer Function for Output_7/Input_3}=

ans =

$$-\frac{66567738028908586915022657578839656366080000000000000000\,s^3+22634260199879421441306965017574830427}{42535295865117307932921825928971026432000000000000000\,s^4+83954465566733937705683783145565853130424320000000000000\,s^3+3133553370}$$

Transfer Function for Output_7/Input_4}=

ans =

$$-\frac{21301676169250747812807250425228690037145600000000000000\,s^3+131631175676651553945666541300559486581}{42535295865117307932921825928971026432000000000000000\,s^4+83954465566733937705683783145565853130424320000000000000\,s^3+3133553370}$$

Transfer Function for Output_8/Input_1}=

ans = 0

Transfer Function for Output_8/Input_2}=

ans = 0

Transfer Function for Output_8/Input_3}=

ans =

$$\frac{48915590244884904122860099818316680396800000000000000000\,s^3+48757741761929453793121026922294268917}{42535295865117307932921825928971026432000000000000000\,s^4+83954465566733937705683783145565853130424320000000000000\,s^3+3133553370}$$

Transfer Function for Output_8/Input_4}=

ans =

$$-\frac{34895956727742239428169065992127830084812800000000000000\,s^3+59143789633158849815588322379197494650}{42535295865117307932921825928971026432000000000000000\,s^4+83954465566733937705683783145565853130424320000000000000\,s^3+3133553370}$$

Transfer Function for Output_9/Input_1}=

ans = 0

Transfer Function for Output_9/Input_2}=

ans = 0

Transfer Function for Output_9/Input_3}=

ans =

$$-\frac{6630330034804473912333447587922183639185489920000000000\,s^2+223706758}{42535295865117307932921825928971026432000000000000000\,s^4+83954465566733937705683783145565853130424320000000000000\,s^3+3133553370}$$

Transfer Function for Output_9/Input_4}=

ans =

$$-\frac{2008321518813410122383772375028649749049769984000000000\,s^2+8180215}{21267647932558653966460912964485513216000000000000000\,s^4+41977232783366968852841891572782926565212160000000000000\,s^3+15667766}$$

Transfer Function for Output_10/Input_1}=

ans = 0

Transfer Function for Output_10/Input_2}=

ans = 0

Transfer Function for Output_10/Input_3}=

```
ans =
```

$$\frac{979290116702595780539659198362699941543936000000000000\, s^3 + 9761299900738276649382829589843312637}{2\, s\, (425352958651173079329218259289710264320000000000000000\, s^4 + 839544655667339377056837831455658531304243200000000000\, s^3 + 31335533}$$

```
------------------------------------------------------------
Transfer Function for Output_10/Input_4}=
ans =
```

$$-\frac{1746542634223499083379861752905997895744880640000000000\, s^3 + 296014667113960043327019553507883460}{5\, s\, (425352958651173079329218259289710264320000000000000000\, s^4 + 839544655667339377056837831455658531304243200000000000\, s^3 + 313355}$$

```
------------------------------------------------------------
```

Eigenvalues of matrix `A` are obtained as the following:

```
eig_A
```

```
eig_A = 10×1 complex
   0.0000 + 0.0000i
 -11.6828 +10.0160i
 -11.6828 -10.0160i
  -0.2991 + 0.6752i
  -0.2991 - 0.6752i
  -0.0006 + 0.0000i
 -16.0483 + 0.0000i
  -0.2199 + 0.0000i
  -1.7347 + 3.2696i
  -1.7347 - 3.2696i
```

Thus, the characterisic equation of the open-loop system derives using `det(SI - A)`:

```
eval(det(s*eye(numStates) - A))
```

```
ans =
```

$$s^{10} + \frac{21851\, s^9}{500} + \frac{522986175445449\, s^8}{655360000000} + \frac{4554263701110603\, s^7}{640000000000} + \frac{6684242411603593\, s^6}{244140625000} + \frac{56600947352863632\, s^5}{762939453125} + \frac{27661707448294182912\, s^4}{476837158203125} + \frac{7052637}{1862}$$

Considering a **Negative Unity Feedback** (**-1 feedback**) through all of the system states will lead into the following eigenvalues and charactristic equation.

```
eig_A_prime
```

```
eig_A_prime = 10×1 complex
10^2 ×
   3.5342 + 0.0000i
  -0.1422 + 0.0770i
  -0.1422 - 0.0770i
  -0.0838 + 0.0000i
   0.0203 + 0.0000i
   0.0027 + 0.0168i
   0.0027 - 0.0168i
  -0.0009 + 0.0010i
  -0.0009 - 0.0010i
   0.0000 + 0.0000i
```

And the characteristic equation for the colsed-loop system (`det(SI - A + BK)`) :

```
det(s*eye(numStates) - A_prime)
```

```
ans =
```

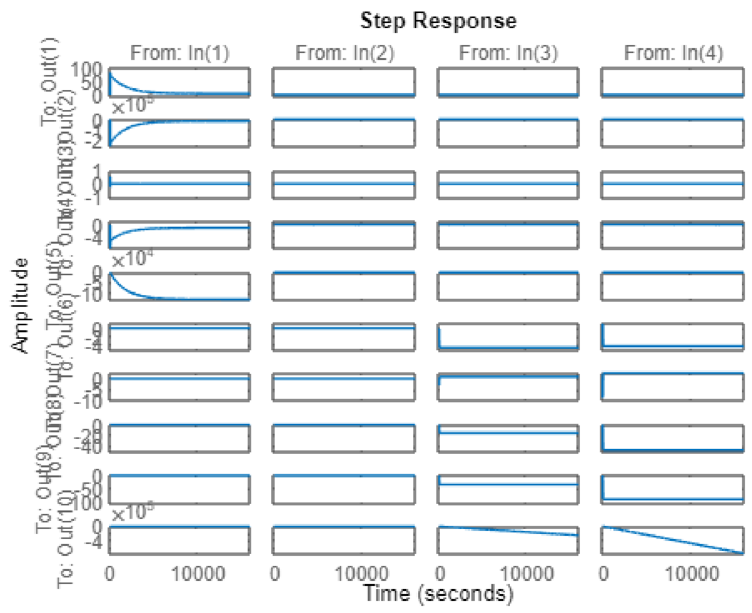
$$s^{10} - \frac{1356791292858470539617419715312148141178244407\, s^9}{42535295865117307932921825928971026432000} - \frac{4999851887546629306181743563934565689229973692 9489\, s^8}{425352958651173079329218259289710264320000000} - \frac{348996044406327803}{23945242826029}$$

How inputs would affect outputs

Based on the obtained transfer functions for the system, the following results are concluded:

The afrementioned results could be observed through `step` responses of the system in the following figure.

```
step(sys_ss)
```

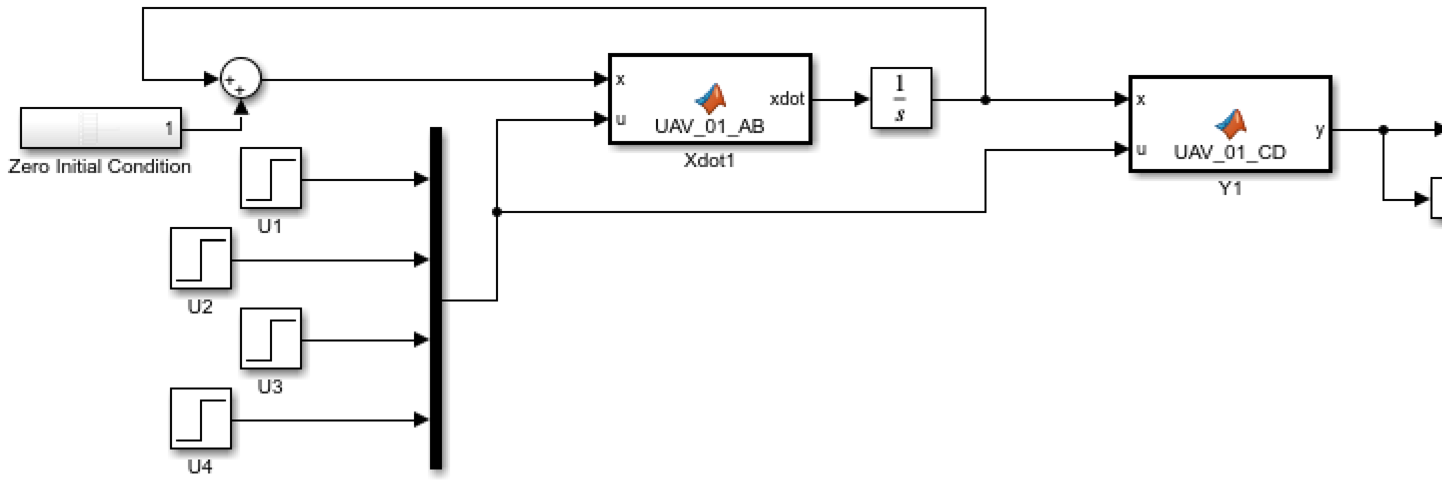


```
open("Q2_UAV.slx")
out = sim("Q2_UAV.slx")
```

```
out = 
  Simulink.SimulationOutput:

                   tout: [106x1 double]
                      y: [1x1 timeseries]

     SimulationMetadata: [1x1 Simulink.SimulationMetadata]
           ErrorMessage: [0x0 char]
```

System Simulation in SimuLink

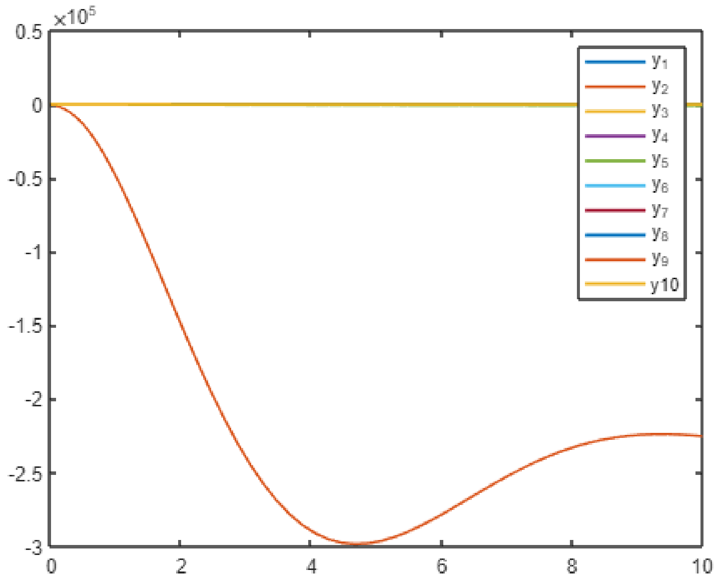
We could implement the system in Simulink as follows:



```
size(out.y.Data)
```

```
ans = 1×3
    10     1   106
```

```
reshaped = reshape(out.y.Data , [10,size(out.y.Data,3)]);
figure
plot(out.tout, reshaped(1,:), LineWidth=1.5)
hold on
plot(out.tout, reshaped(2,:),LineWidth=1.25)
plot(out.tout, reshaped(3,:),LineWidth=1.5)
plot(out.tout, reshaped(4,:),LineWidth=1.5)
plot(out.tout, reshaped(5,:),LineWidth=1.5)
plot(out.tout, reshaped(6,:),LineWidth=1.5)
plot(out.tout, reshaped(7,:),LineWidth=1.5)
plot(out.tout, reshaped(8,:),LineWidth=1.5)
plot(out.tout, reshaped(9,:),LineWidth=1.5)
plot(out.tout, reshaped(10,:),LineWidth=1.5)
legend("y_1", "y_2", "y_3","y_4","y_5","y_6","y_7","y_8","y_9","y10")
```



You can run the system under different inputs and initial conditions and study system behaviour.

Controller Design

Since the eigenvalues of the system include instablity, we may want to desing a controller for the system in order to place its poles where we desire. The required condition for such design is that the Controllability Matrix of the system (`U=ctrb(A,B)`) is full rank.

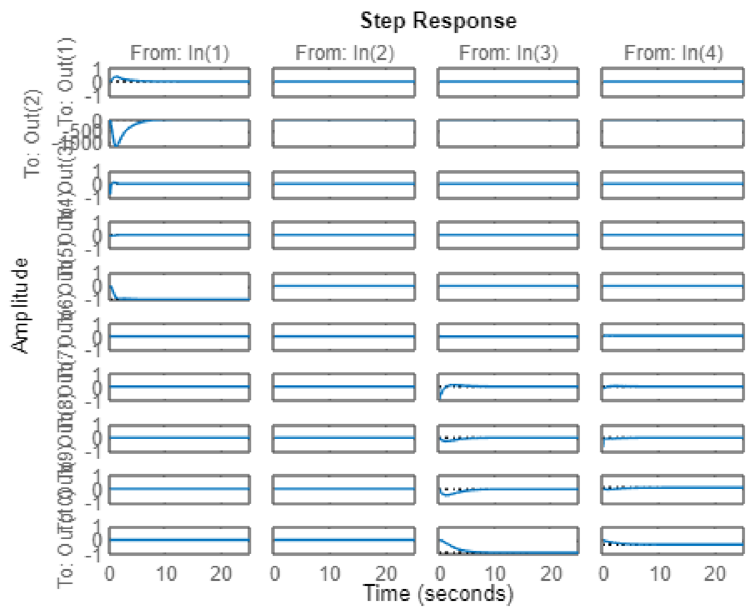
```
[rankU , numStates]
```

```
ans = 1×2
    10    10
```

According to $10 = 10$, the system is controllable and its poles can be placed at any desired location. Considering the obtained poles for the system through unity feedback, unity feedback system poles are stable too.

Let's try this out and see if it is able to place system poles where they are stable or not.

```
step(sys_lqr)
```



To conclude, we observed that a negative unity feedback can stabilize the system. In addition, we used `lqr` MATALB command to find an optimal `K` for the feedback gain.

System Type

We know that the Type of a system equals to the number of poles at the origin. So let's recall the system poles which are the eigenvalues of the matrix `A`.

```
eig_A
```

```
eig_A = 10×1 complex
   0.0000 + 0.0000i
 -11.6828 +10.0160i
 -11.6828 -10.0160i
  -0.2991 + 0.6752i
  -0.2991 - 0.6752i
  -0.0006 + 0.0000i
 -16.0483 + 0.0000i
  -0.2199 + 0.0000i
  -1.7347 + 3.2696i
  -1.7347 - 3.2696i
```

Counting poles at the origin:

```
length(find(eig_A == 0))
```

```
ans = 1
```

So the maximum SystemType of which the system could behave is 1 and since the system is MIMO, Type of the system could vary for different transfer functions of the system due to the zero-pole cancelation throughout system realization.

To study each transfer function (**those that are nonzero**), we need to checkout their denominators.

```matlab
 for o=1:numOutputs
     for i=1:numInputs
         fprintf("Transfer Function for Output_%d/Input_%d}= \n", o,i)
         eval(simplify(G(o,i)))
         fprintf("------------------------------------------------------------
\n")
     end
 end
```

```
Transfer Function for Output_1/Input_1}=
ans =
```

$$\frac{10393183074147886495665489231992804343808000000000\,s^4 + 3121469489518594565991159974410976863606171}{32451855365842672678315602057625600000000000000000\,s^5 + 77768924272920014513222641394976292864000000000000\,s^4 + 81565903709617}$$

```
------------------------------------------------------------
Transfer Function for Output_1/Input_2}=
ans = 0
------------------------------------------------------------
Transfer Function for Output_1/Input_3}=
ans = 0
------------------------------------------------------------
Transfer Function for Output_1/Input_4}=
ans = 0
------------------------------------------------------------
Transfer Function for Output_2/Input_1}=
ans =
```

$$-\frac{240635890362669980834737314079256269617155442827329536\,s^4 + 6479517839019729651030218483992407492427553214}{16225927682921336339157801028812800000000000000000\,s^5 + 38884462136460007256611320697488146432000000000000\,s^4 + 407829518548}$$

```
------------------------------------------------------------
Transfer Function for Output_2/Input_2}=
ans = 0
------------------------------------------------------------
Transfer Function for Output_2/Input_3}=
ans = 0
------------------------------------------------------------
Transfer Function for Output_2/Input_4}=
ans = 0
------------------------------------------------------------
Transfer Function for Output_3/Input_1}=
ans =
```

$$-\frac{5\,s\,(108470326560329133427269899877613568000000000000\,s^3 + 86787523591651347868}{405648192073033408478945025720320000000000000000\,s^5 + 972111553411500181415283017437203660800000000000\,s^4 + 101957379637021499311558}$$

```
------------------------------------------------------------
Transfer Function for Output_3/Input_2}=
ans = 0
------------------------------------------------------------
Transfer Function for Output_3/Input_3}=
ans = 0
------------------------------------------------------------
Transfer Function for Output_3/Input_4}=
ans = 0
------------------------------------------------------------
Transfer Function for Output_4/Input_1}=
ans =
```

$$-\frac{542351632801645667136349499388067840000000000000\,s^3 + 43393761795825673934111}{405648192073033408478945025720320000000000000000\,s^5 + 972111553411500181415283017437203660800000000000\,s^4 + 101957379637021499311558}$$

```
------------------------------------------------------------
Transfer Function for Output_4/Input_2}=
ans = 0
------------------------------------------------------------
Transfer Function for Output_4/Input_3}=
ans = 0
------------------------------------------------------------
Transfer Function for Output_4/Input_4}=
ans = 0
-------------------------------------------------------------
Transfer Function for Output_5/Input_1}=
ans =
```

$$-\frac{-1381309892768376202856928228845491218022400000000\,s^3 + 623080580504513465254}{50706024009129176059868128215040000000000000000\,s^5 + 121513944176437522676910377179650457600000000000\,s^4 + 1274467245462768741}$$

```
-------------------------------------------------------------
Transfer Function for Output_5/Input_2}=
ans = 0
-------------------------------------------------------------
Transfer Function for Output_5/Input_3}=
ans = 0
-------------------------------------------------------------
Transfer Function for Output_5/Input_4}=
ans = 0
-------------------------------------------------------------
Transfer Function for Output_6/Input_1}=
ans = 0
-------------------------------------------------------------
Transfer Function for Output_6/Input_2}=
ans = 0
-------------------------------------------------------------
Transfer Function for Output_6/Input_3}=
ans =
```

$$-\frac{41404106117952775154211072292456589222864958259200000000\,s^2 + 45449}{2126764793255865396646091296448551321600000000000000000\,s^4 + 4197723278336696885284189157278292656521216000000000000000\,s^3 + 156677}$$

```
-------------------------------------------------------------
Transfer Function for Output_6/Input_4}=
ans =
```

$$\frac{829075762310777042435117279353455359851233280000000000000\,s^3 + 230729736011448136999435056807382306}{26584559915698317458076141205606891520000000000000000000\,s^4 + 524715409792087110660523644659786582065152000000000000000\,s^3 + 1958}$$

```
-------------------------------------------------------------
Transfer Function for Output_7/Input_1}=
ans = 0
-------------------------------------------------------------
Transfer Function for Output_7/Input_2}=
ans = 0
-------------------------------------------------------------
Transfer Function for Output_7/Input_3}=
ans =
```

$$-\frac{665677380289085869150226575788396563660800000000000000\,s^3 + 2263426019987942144130696501757483042 7}{4253529586511730793292182592897102643200000000000000000\,s^4 + 83954465566733937705683783145565853130424320000000000\,s^3 + 3133553370}$$

```
-------------------------------------------------------------
Transfer Function for Output_7/Input_4}=
ans =
```

$$-\frac{2130167616925074781280725042522869003714560000000000000\,s^3 + 1316311756766515539456665413005594865 81}{4253529586511730793292182592897102643200000000000000000\,s^4 + 83954465566733937705683783145565853130424320000000000\,s^3 + 3133553370}$$

```
-------------------------------------------------------------
Transfer Function for Output_8/Input_1}=
ans = 0
-------------------------------------------------------------
Transfer Function for Output_8/Input_2}=
ans = 0
-------------------------------------------------------------
Transfer Function for Output_8/Input_3}=
ans =
```

$$\frac{4891559024488490412286009981831668039680000000000000000\,s^3 + 48757741761929453793121026922294268917 7}{4253529586511730793292182592897102643200000000000000000\,s^4 + 83954465566733937705683783145565853130424320000000000\,s^3 + 313355337048}$$

```
-------------------------------------------------------------
Transfer Function for Output_8/Input_4}=
ans =
```

$$-\frac{3489595672774223942816906599212783008481280000000000000\,s^3 + 591437896331588498155883223791974946501}{4253529586511730793292182592897102643200000000000000000\,s^4 + 83954465566733937705683783145565853130424320000000000\,s^3 + 3133553370}$$

```
-------------------------------------------------------------
```

```
Transfer Function for Output_9/Input_1}=
ans = 0
------------------------------------------------------------
Transfer Function for Output_9/Input_2}=
ans = 0
------------------------------------------------------------
Transfer Function for Output_9/Input_3}=
ans =
```

$$-\frac{663033003480447391233344758792218363918548992000000000\, s^2 + 223706758}{4253529586511730793292182592897102643200000000000000000\, s^4 + 839544655667339377056837831455658531304243200000000000\, s^3 + 3133553370}$$

```
------------------------------------------------------------
Transfer Function for Output_9/Input_4}=
ans =
```

$$-\frac{2008321518813410122383772375028649749049769984000000000\, s^2 + 8180215}{21267647932558653966460912964485513216000000000000000000\, s^4 + 419772327833669688528418915727829265652121600000000000\, s^3 + 15667766}$$

```
------------------------------------------------------------
Transfer Function for Output_10/Input_1}=
ans = 0
------------------------------------------------------------
Transfer Function for Output_10/Input_2}=
ans = 0
------------------------------------------------------------
Transfer Function for Output_10/Input_3}=
ans =
```

$$\frac{979290116702595780539659198362699941543936000000000000\, s^3 + 97612999007382766493828295898433126373}{2\, s\ (4253529586511730793292182592897102643200000000000000000\, s^4 + 839544655667339377056837831455658531304243200000000000\, s^3 + 31335533}$$

```
------------------------------------------------------------
Transfer Function for Output_10/Input_4}=
ans =
```

$$-\frac{174654263422349908337986175290599789574488064000000000\, s^3 + 2960146671139600433270195535078834607}{5\, s\ (4253529586511730793292182592897102643200000000000000000\, s^4 + 839544655667339377056837831455658531304243200000000000\, s^3 + 313355}$$

```
------------------------------------------------------------
```

You can see that the nonzero trasfer funcitons of output_10 are of SystemType 1 and all other nonzero transfer functions are for all other outputs are of System Type 0.

Singular Values Decomposition

We know that singular values of a matrix `A` are the square roots of the eigenvalues of A^TA or AA^T. in MATLAB, one could use the command `svd` to obtain these values. Remeber that the correpending decomposition for matrix `A` is as follows:

$$A = \mathrm{USV}^T$$

```
UU
```

```
UU = 10×10
    0.0377         0    0.5000         0    0.0054         0   -0.8602 ⋯
   -0.7055         0   -0.0024         0   -0.7071         0   -0.0396
   -0.0446         0   -0.0000         0    0.0005         0    0.1049
    0.7063         0   -0.0290         0   -0.7066         0    0.0130
   -0.0001         0    0.8656         0   -0.0287         0    0.4972
         0    0.8875         0    0.4441         0    0.1228         0
         0   -0.4425         0    0.8938         0   -0.0327         0
         0    0.1212         0    0.0134         0   -0.9398         0
         0    0.0168         0   -0.0583         0    0.1258         0
         0   -0.0399         0   -0.0185         0    0.2914         0
```

```
VV
```

```
VV = 10×10
    0.0553         0   -0.0189         0   -0.0199         0    0.9981 ⋯
    0.0059         0    0.0761         0    0.9969         0    0.0210
    0.0003         0   -0.9969         0    0.0765         0   -0.0174
   -0.9985         0   -0.0008         0    0.0049         0    0.0554
    0.0000         0    0.0000         0    0.0001         0   -0.0000
         0    0.0273         0   -0.1854         0   -0.4257         0
         0    0.3886         0   -0.8973         0    0.1744         0
         0   -0.8180         0   -0.2887         0    0.4613         0
         0    0.4233         0    0.2778         0    0.7587         0
         0         0         0         0         0         0         0
```

```
SS
```

```
SS = 10×10
   22.3693         0         0         0         0         0         0 ⋯
         0   20.5299         0         0         0         0         0
         0         0   19.6690         0         0         0         0
         0         0         0   15.6502         0         0         0
         0         0         0         0   10.5870         0         0
         0         0         0         0         0    1.5847         0
         0         0         0         0         0         0    0.5285
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0
```

So the singular values of the matrix `A` we have in descending order is:

```
diag(SS)
```

```
ans = 10×1
   22.3693
   20.5299
   19.6690
   15.6502
   10.5870
    1.5847
    0.5285
    0.2192
    0.0000
         0
```

BIBO stability

As mentioned earlier, eigenvalues of the system dynamics matrix `A` show that the system is **Bounded-Input-Bounded_Output**. This is also demonstrated in the section that system is simulated in Simulink if you run the system under various inputs.

Model Reduction - Method 1

Following the book instructions to reduce the model using method one "Model Separation" provides us with following results.

```
Az, Bz, Cz
```

```
Az = 10×10 complex
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i ⋯
   0.0000 + 0.0000i -11.6828 +10.0160i   0.0000 + 0.0000i   0.0000 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i -11.6828 -10.0160i   0.0000 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i  -0.2991 +
0.6752i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
```

```
    0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
    0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
    0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
    0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
    0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
Bz = 10×4 complex
10^2 ×
    0.0000 + 0.0000i   0.0000 + 0.0000i  -0.1569 + 0.0000i  -0.4766 +
0.0000i
   -0.0157 + 1.2827i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
   -0.0157 - 1.2827i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
   -0.4758 - 0.3951i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
   -0.4758 + 0.3951i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
    0.7595 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
    0.0000 + 0.0000i   0.0000 + 0.0000i   1.5783 + 0.0000i   0.0268 +
0.0000i
    0.0000 + 0.0000i   0.0000 + 0.0000i   0.2393 + 0.0000i   0.6572 +
0.0000i
    0.0000 + 0.0000i   0.0000 + 0.0000i   0.0297 + 0.2841i   0.2364 -
2.2835i
    0.0000 + 0.0000i   0.0000 + 0.0000i   0.0297 - 0.2841i   0.2364 +
2.2835i
Cz = 10×10 complex
10^3 ×
    0.0000 + 0.0000i   0.0000 - 0.0000i   0.0000 + 0.0000i   0.0003 +
0.0004i ⋯
    0.0000 + 0.0000i   0.1251 + 0.0286i   0.1251 - 0.0286i  -1.0463 -
1.1826i
    0.0000 + 0.0000i  -0.0002 + 0.0005i  -0.0002 - 0.0005i   0.0000 +
0.0000i
    0.0000 + 0.0000i   0.0000 - 0.0000i   0.0000 + 0.0000i   0.0000 -
0.0000i
    0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 - 0.0000i  -0.0009 +
0.0000i
    0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
    0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
    0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
    0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
    0.0010 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
```

Eigenvalues of matrix A had been obtained earlier:

```
eig_A
```

```
eig_A = 10×1 complex
    0.0000 + 0.0000i
  -11.6828 +10.0160i
  -11.6828 -10.0160i
   -0.2991 + 0.6752i
   -0.2991 - 0.6752i
   -0.0006 + 0.0000i
```

```
 -16.0483 + 0.0000i
  -0.2199 + 0.0000i
  -1.7347 + 3.2696i
  -1.7347 - 3.2696i
```

We could see that none of the system modes could be assumed as nondominant. The most probable modes to be dropped off the system are -11.6828 ± 10.0160 and -16.0483. So we are expecting to reduce the system from $n = 10$ to $q = 7$.

According to the following equations, we could separate the system dynamics.

$$\begin{bmatrix} \dot{\underline{z}}_1 \\ \dot{\underline{z}}_2 \end{bmatrix} = \begin{bmatrix} A_{1z} & 0 \\ 0 & A_{2z} \end{bmatrix} \begin{bmatrix} \underline{z}_1 \\ \underline{z}_2 \end{bmatrix} + \begin{bmatrix} B_{1z} \\ B_{2z} \end{bmatrix} \underline{u}$$

$$y = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \end{bmatrix} + D\underline{u}$$

And a whole bunch of steps will lead to:

$$A_r = T_{11} A_{1z} T_{11}^{-1}$$
$$B_r = T_{11} \left(A_{1z} T_{11}^{-1} T_{12} A_{2z}^{-1} B_{2z} + B_{1z} \right)$$
$$F = T_{21} T_{11}^{-1}$$
$$E = \left(T_{21} T_{11}^{-1} T_{12} - T_{22} \right) A_{2z}^{-1} B_{2z}$$
$$C_r = C_1 + C_2 F$$
$$D_r = C_2 E + D$$

where they all present:

$$\dot{\underline{x}}_1 = A_r \underline{x}_1 + B_r \underline{u}$$
$$\begin{bmatrix} \underline{x}_2 \\ \underline{y} \end{bmatrix} = \begin{bmatrix} F \\ C_r \end{bmatrix} \underline{x}_1 + \begin{bmatrix} E \\ D_r \end{bmatrix} \underline{u}$$

The following results are obtained:

```
Ar
```

```
Ar = 7×7 complex
10^2 ×
   0.1968 + 0.0000i  -3.5381 - 0.0000i   0.0001 + 0.0000i   0.0000 +
0.0000i ⋯
   0.0113 + 0.0000i  -0.2030 - 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
   0.3515 + 0.0000i  -6.1303 - 0.0000i   0.0002 + 0.0000i   0.0000 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i  -0.0104 -
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0061 -
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i  -0.0019 +
0.0000i
```

```
    0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i  -0.0000 +
0.0000i
```

Br

```
Br = 7×4 complex
  -2.0893 + 0.0454i   0.0000 + 0.0000i  -0.0000 + 0.0000i  -0.0000 +
0.0000i
  -0.2354 + 0.0028i   0.0000 + 0.0000i  -0.0000 + 0.0000i  -0.0000 +
0.0000i
  -6.5641 + 0.2390i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i  -8.9281 - 0.1813i   5.1502 -
0.0031i
   0.0000 + 0.0000i   0.0000 + 0.0000i   5.8658 - 0.0260i -82.1358 -
0.0004i
   0.0000 + 0.0000i   0.0000 + 0.0000i  -9.7674 + 0.0004i  -0.1660 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.3520 + 0.0094i   0.0060 +
0.0002i
```

Cr

```
Cr = 10×7 complex
10^3 ×
  -0.0000 - 0.0000i   0.0005 + 0.0007i   0.0003 - 0.0004i  -0.0000 +
0.0000i ⋯
  -0.2606 + 0.0589i   3.5440 - 2.1975i  -1.0465 + 1.1826i   0.0017 +
0.0000i
   0.0005 + 0.0011i  -0.0079 - 0.0185i   0.0000 - 0.0000i  -0.0000 +
0.0000i
  -0.0001 - 0.0000i   0.0012 + 0.0006i   0.0000 + 0.0000i   0.0000 +
0.0000i
  -0.0000 + 0.0001i  -0.0008 - 0.0009i  -0.0009 + 0.0000i  -0.0010 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i  -0.0000 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0002 -
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i  -0.0000 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i  -0.0000 +
0.0000i
   0.0010 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 -
0.0000i
```

F

```
F = 3×7 complex
  -0.0121 + 0.0000i   0.6058 - 0.0000i  -0.0001 - 0.0000i   0.0000 +
0.0000i ⋯
  -2.0717 - 0.0000i  36.0963 + 0.0000i  -0.0011 - 0.0000i   0.0000 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i  -0.1868 +
0.0000i
```

E

```
E = 3×4 complex
10^2 ×
  -0.0652 - 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
   1.5139 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
```

```
   0.0000 + 0.0000i   0.0000 + 0.0000i  -0.0976 - 0.0000i  -0.0017 -
0.0000i
```

Hence, the system is reduced from order of 10 to 7 and three nondominant poles are dropped off the system main dynamics.

Model Reduction - Method 2

Following the book instructions to reduce the model using method one "Residuals" provides us with following results.

```
Azz
```

```
Azz = 7×7 complex
  -1.7347 - 3.2696i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i ⋯
   0.0000 + 0.0000i  -0.0043 + 0.0093i   0.0402 + 0.0178i  -0.0001 -
0.0002i
   0.0000 + 0.0000i   0.0021 + 0.0008i   0.0037 - 0.0083i   0.0002 -
0.0001i
   0.0000 + 0.0000i  -0.2557 - 0.5924i  -0.0322 - 0.0143i  -0.2990 -
0.6750i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
```

```
Bzz
```

```
Bzz = 7×4 complex
10^2 ×
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0658 + 0.0000i  -0.4695 -
0.0000i
  -0.9672 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
   0.0000 + 0.2020i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
   0.7735 - 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i  -0.0901 - 0.0000i   0.6466 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i  -0.0576 + 0.0000i   0.4690 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 - 0.1595i  -0.0000 +
1.1412i
```

```
Czz
```

```
Czz = 10×7 complex
10^3 ×
   0.0000 + 0.0000i   0.0003 + 0.0000i   0.0000 - 0.0015i  -0.0000 +
0.0000i ⋯
   0.0000 + 0.0000i  -1.0463 + 0.0000i   0.0000 + 4.7303i   0.0017 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 - 0.0001i  -0.0000 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
   0.0000 + 0.0000i  -0.0009 + 0.0000i   0.0000 + 0.0000i  -0.0010 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
```

```
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
   0.0010 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 +
0.0000i
```

```
eig_A
```

```
eig_A = 10×1 complex
   0.0000 + 0.0000i
 -11.6828 +10.0160i
 -11.6828 -10.0160i
  -0.2991 + 0.6752i
  -0.2991 - 0.6752i
  -0.0006 + 0.0000i
 -16.0483 + 0.0000i
  -0.2199 + 0.0000i
  -1.7347 + 3.2696i
  -1.7347 - 3.2696i
```

```
eig_Azz
```

```
eig_Azz = 7×1 complex
  -1.7347 - 3.2696i
  -0.2991 - 0.6752i
  -0.0006 + 0.0013i
  -0.0000 - 0.0000i
  -1.7224 + 3.2463i
  -0.2198 + 0.0000i
  -0.0006 - 0.0000i
```

we can see that all the dominant poles of the system are almost preserved and the system is reduced from order of 10 to 7. Note that the system responseness would not be exactly the same.