MODERN CONTROL PROJECT

Prof. Arash Sadeghzadeh

ABSTRACT

Analyzing a quadcopter using its state space. Designing state feedback and linear observer for it.

Sepehr Kerachi

97242139

I. Introduction

In this report, a quadcopter [1] has been considered. Given its transfer function matrix, the following analysis have been done:

1. A realization has been provided
2. Its controllability has been discussed
3. Its observability has been discussed
4. Its stability in different methods have been discussed

Then a

1. state-feedback
2. integral-state-feedback
3. linear observer

has been designed using its controllable state space matrices.

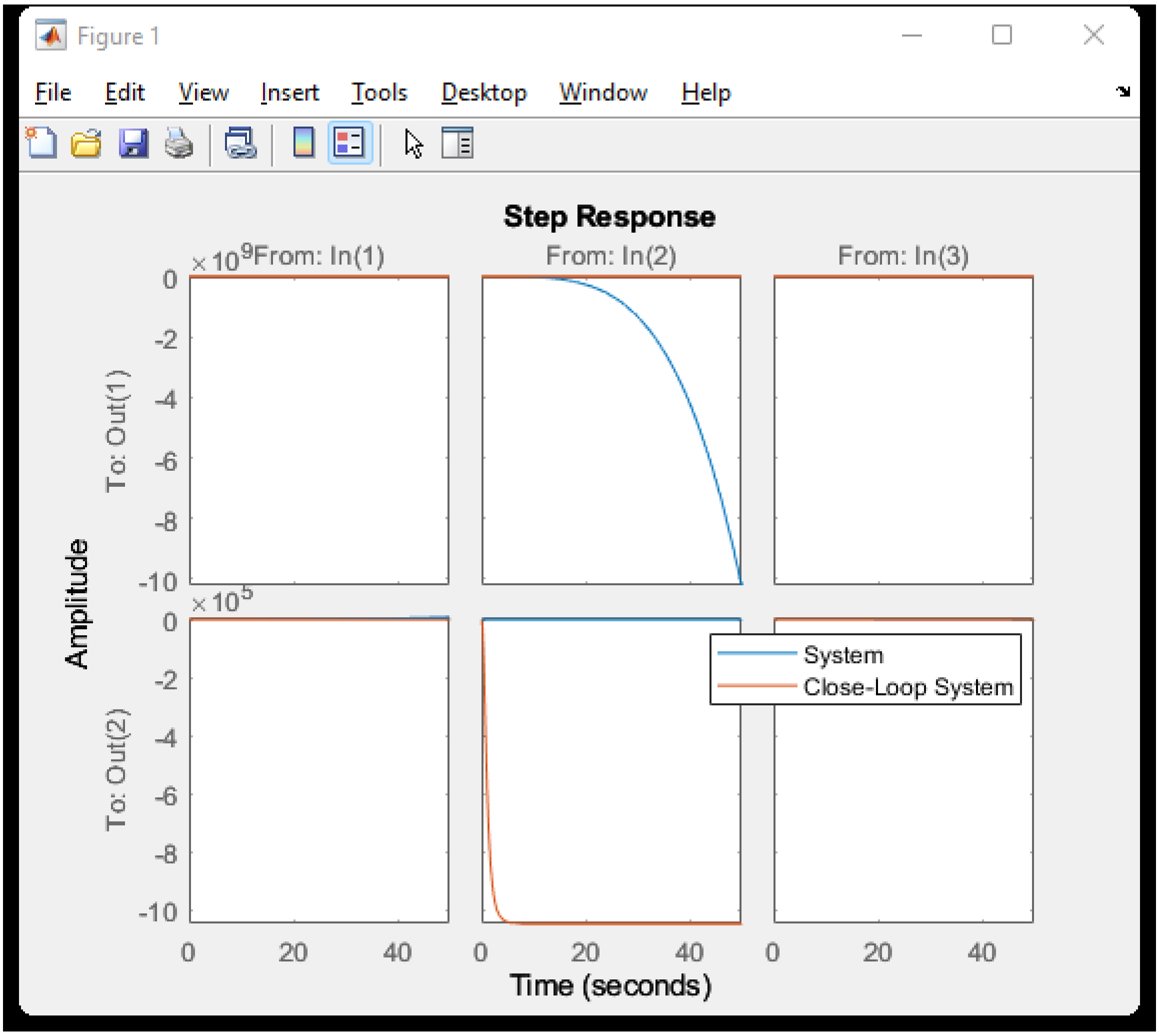
The systems transfer function have been given with the help of the efforts in [1]:

$$H(s) = \begin{bmatrix} 0 & -\frac{39200}{s^4} & 0 \\ \frac{50}{s^2} & 0 & -\frac{1}{s^2} \end{bmatrix}$$

This transfer function has been defined using codes bellow:

```
% system
clc
numerators = {0 -39200 0; [50/9 0 0] 0 [-1 0 0]};
denominators = [1 0 0 0 0];
system = tf(numerators,denominators);
step(system)
pause
```

And its step response is:



II. Analysis

A canonical realization has been provided using the following codes:

```
% 1. realization
clc
state_space = canon(system, "modal");
A = state_space.A
B = state_space.B
C = state_space.C
D = state_space.D
pause
```

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$

```
A =

         0   32.0000         0         0         0         0         0         0
         0         0  128.0000         0         0         0         0         0
         0         0         0   32.0000         0         0         0         0
         0         0         0         0         0         0         0         0
         0         0         0         0         0   64.0000         0         0
         0         0         0         0         0         0    0.0000         0
         0         0         0         0         0         0         0    0.0005
         0         0         0         0         0         0         0         0

B =

         0         0         0
         0         0         0
         0         0         0
         0    0.5000         0
         0         0         0
    0.3472         0   -0.0625
         0         0         0
         0         0         0

C =

   -0.5981         0         0         0         0         0         0         0
         0         0         0         0    0.2500         0         0         0

D =

     0     0     0
     0     0     0
```

A. Controllability

For knowing whether all state variables are controllable or not, we can use φ_c. The rank of this matrix shows have many of our state variables are controllable.

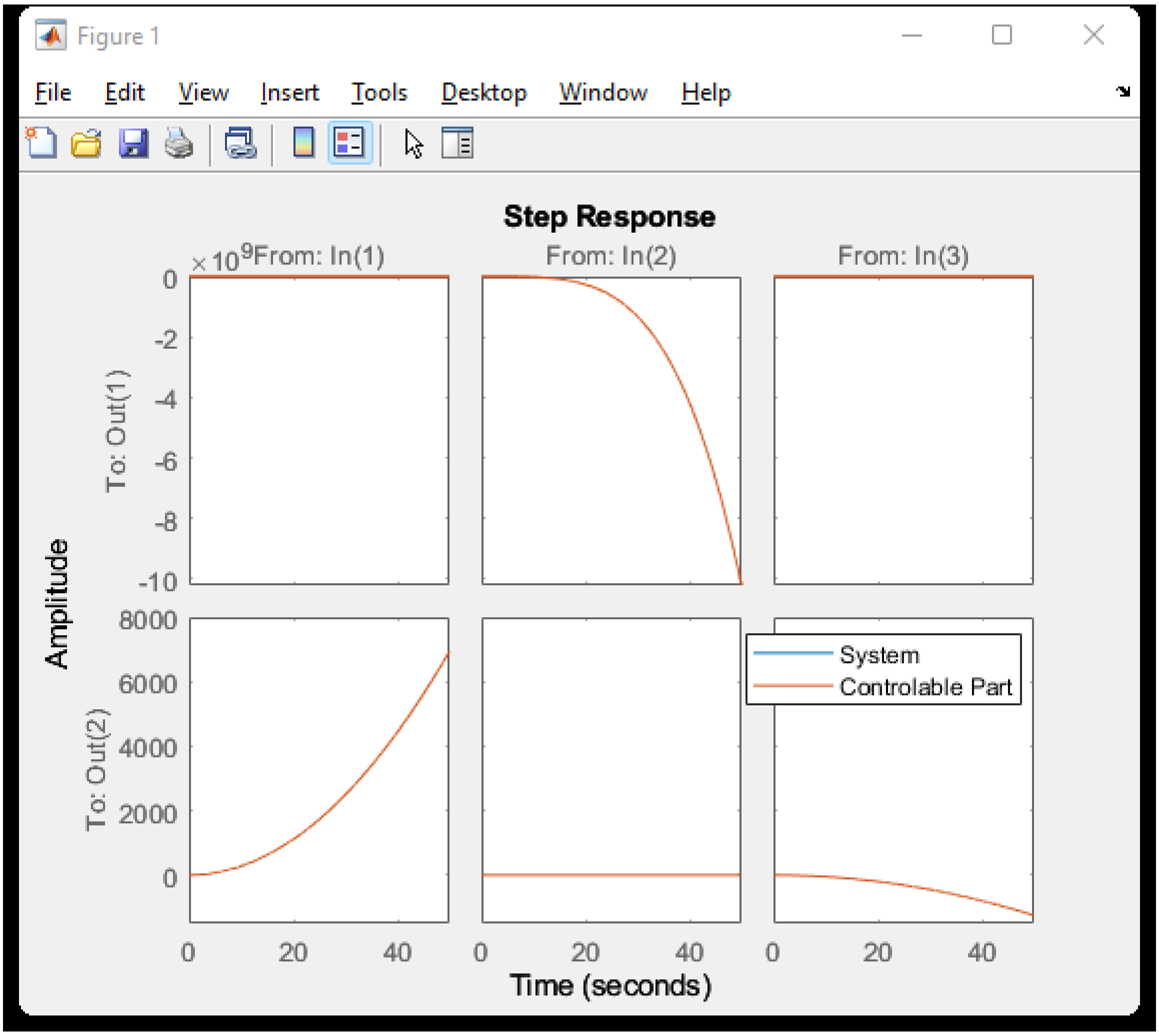
$$\varphi_c = [B \quad AB \quad A^2B]$$

```
clc
phi_c = ctrb(A,B)
rank(phi_c)     % => two uncontrollable modes
```

```
phi_c =

   1.0e+04 *

  Columns 1 through 13

         0         0         0         0         0         0         0         0         0         0    6.5536         0         0
         0         0         0         0         0         0         0    0.2048         0         0         0         0         0
         0         0         0         0    0.0016         0         0         0         0         0         0         0         0
         0    0.0001         0         0         0         0         0         0         0         0         0         0         0
         0         0         0    0.0022         0   -0.0004         0         0         0         0         0         0         0
    0.0000         0   -0.0000         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0         0         0

  Columns 14 through 24

         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0
         0         0         0         0         0         0         0         0         0         0         0


ans =

     6
```

6 out of 8 state variables are controllable. So, we need to form matrix T in order to easily find which of our state variables are controllable.

```
T = [   0       0       0       0       0       65536   0       0;
        0       0       0       0       2048    0       0       0;
        0       0       0       16      0       0       0       0;
        0       0.5     0       0       0       0       0       0;
        0       0       22.2222 0       0       0       0       0;
        0.3472  0       0       0       0       0       0       0;
        0       0       0       0       0       0       1       0;
        0       0       0       0       0       0       0       1];
T = T';
A_prime = T^(-1) * A * T;
A_11 = A_prime(1:6,1:6);
B_prime = T^(-1) * B;
B_1 = B_prime(1:6,1:3);
C_prime = C * T;
C_1 = C_prime(1:2,1:6);
sys = ss(A_11, B_1, C_1, D)
step(system,sys);
```



The step response of the controllable part of our system is plotted. The controllable state space is:

```
A =

          x1      x2      x3      x4      x5      x6
   x1      0       0       0       0       0       0
   x2   2048       0       0       0       0       0
   x3      0       0       0       0       0       0
   x4      0       0       0       0    5689       0
   x5      0       0   23.04       0       0       0
   x6      0       0       0   46.08       0       0

B =

                 u1           u2           u3
   x1     5.298e-06            0   -9.537e-07
   x2             0            0            0
   x3             0      0.03125            0
   x4             0            0            0
   x5             0            0            0
   x6             0            0            0

C =

             x1         x2         x3         x4         x5         x6
   y1         0          0          0          0          0    -0.2077
   y2         0        512          0          0          0          0

D =

        u1   u2   u3
   y1    0    0    0
   y2    0    0    0

Continuous-time state-space model.
```

B. Observability

The rank of the matrix below shows the number of state variables which are observable.

$$\varphi_c = \begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix}$$

```
% 3. observability
clc
phi_o = obsv(A,C)
rank(phi_o)
pause
```

```
phi_o =

   1.0e+04 *

  -0.0001         0         0         0         0         0         0         0
        0         0         0         0    0.0000         0         0         0
        0   -0.0019         0         0         0         0         0         0
        0         0         0         0         0    0.0016         0         0
        0         0   -0.2450         0         0         0         0         0
        0         0         0         0         0         0    0.0000         0
        0         0         0   -7.8400         0         0         0         0
        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         0         0         0
        0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0
        0         0         0         0         0         0         0         0


ans =

     8
```

C. Stability

If the real part of the eigenvalues of the system are not positive, then the system is stable in Lyapunov stability definition:

```
% 4. stability
clc
%       4.a. lyapunov
eigenvalues = eig(A) % => oscillating stability

%       4.b. asymptotic
eigenvalues % => unstable

%       4.c. BIBO
system % => oscillating stability

%       4.d. T
eigenvalues
system         % => unstable
pause
```

```
eigenvalues =

     0
     0
     0
     0
     0
     0
     0
     0
```

```
system =

  From input 1 to output...
   1:  0

       5.556 s^2
   2:  ---------
          s^4

  From input 2 to output...
       -39200
   1:  ------
        s^4

   2:  0

  From input 3 to output...
   1:  0

       -s^2
   2:  ----
       s^4

Continuous-time transfer function.
```

So, the system is stable in Lyapunov definitions. If the real part of eigenvalues is negative, then the system is asymptotic stable. All the eigenvalues are zero so the system is unstable in asymptotic definitions. BIBO stability require the non-positive real part poles of the system. All the poles are zero then the system is critically stable in BIBO stability definitions.

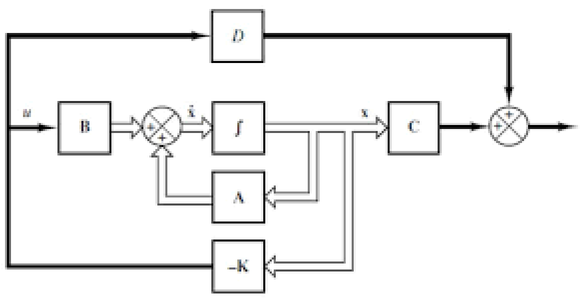
III. Design

D. State-feedback

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$

$$u(t) = V(t) - Kx(t)$$

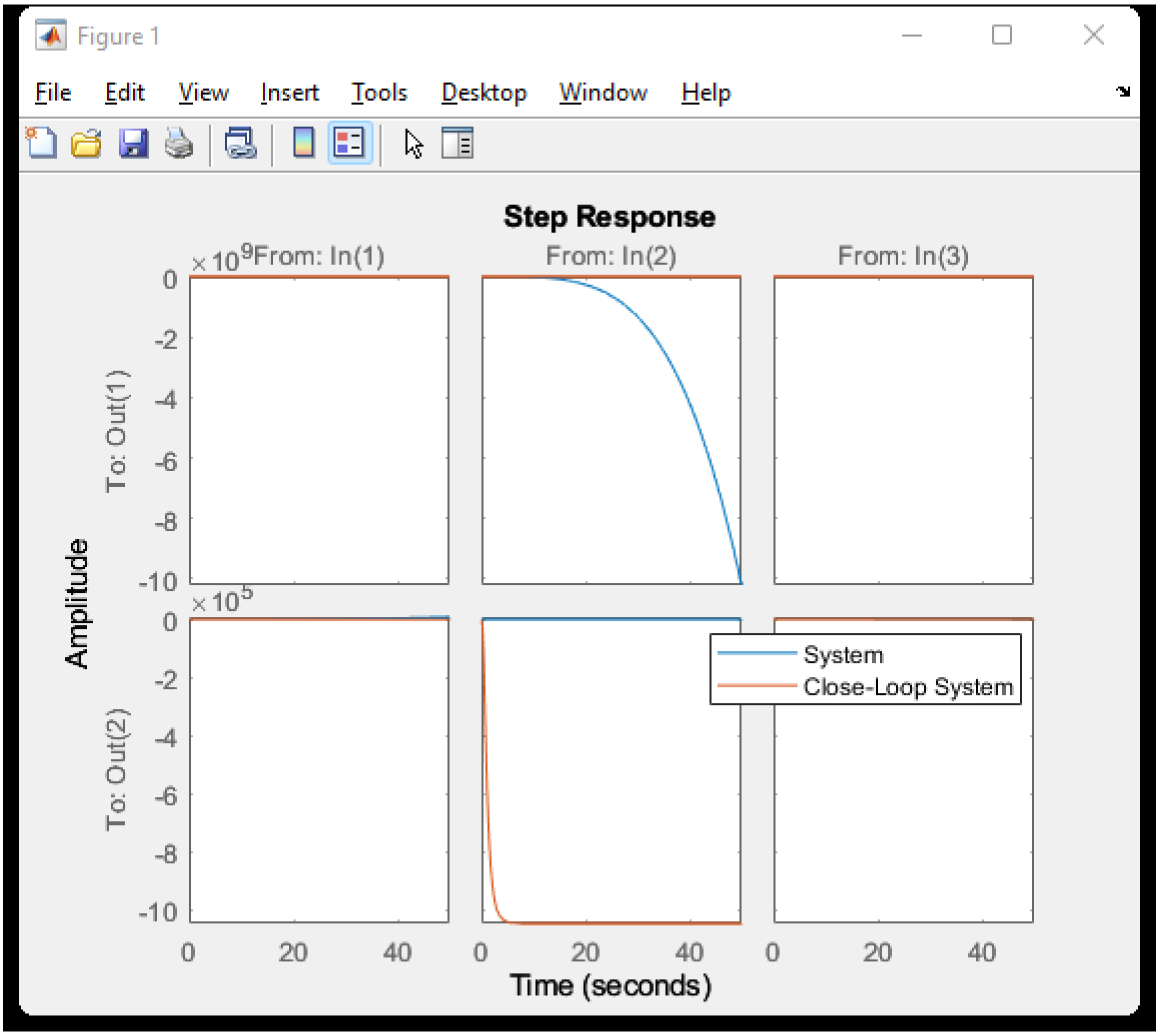
$$\Rightarrow \begin{cases} \dot{x}(t) = Ax(t) + B[V(t) - Kx(t)] \\ y(t) = Cx(t) + Du(t) \end{cases} \Rightarrow \begin{cases} \dot{x}(t) = (A - BK)x(t) + BV(t) \\ y(t) = Cx(t) + Du(t) \end{cases}$$



By setting a proportional feedback for each state variable, we want to move the poles that we can move form (0,0,0,0,0,0) to (-1,-2,-3,-4,-5,-6) so that the system would have an acceptable bandwidth and stability. For MIMO systems the “acker” function didn’t we work so the “place” function have been replaced.

```
% 5.1. state feedback
clc
desired_poles = [-1 -2 -3 -4 -5 -6];
K_prime = place(A_11,B_1,desired_poles);
K_prime = [K_prime zeros(3,2)];
K = K_prime * T^(-1)
A_cl = A - (B * K);
eig(A_cl)
sys_cl = ss(A_cl, B, C, D)
step(system,sys_cl);
legend('System','Close-Loop System','Location','NorthEast');
pause
```

By using the values that have been calculated from "place" function, we create a closed loop system. We plot its step response and check the closed loop poles to check that whether they are on a right place or not.



```
K =

   1.0e+07 *

    0.0002    0.0080    0.3399    1.0442    0.0000    0.0000         0         0
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000         0         0
   -0.0000   -0.0014   -0.0612   -0.1880   -0.0000   -0.0000         0         0


ans =

   -6.0000
   -5.0000
   -1.0000
   -4.0000
   -3.0000
   -2.0000
         0
         0
```

The values which have been calculated for "K" are unexpectedly high. Then the matrices of the closed loop system have been printed.

```
sys_cl =

  A = 
               x1          x2          x3          x4          x5          x6          x7          x8
   x1           0          32           0           0           0           0           0           0
   x2           0           0         128           0           0           0           0           0
   x3           0           0           0          32           0           0           0           0
   x4  -0.0005821    -0.03165      -2.128      -14.09  -1.368e-08  -2.528e-07           0           0
   x5           0           0           0           0           0          64           0           0
   x6      -717.4  -2.856e+04  -1.218e+06  -3.743e+06     -0.1643      -6.907   1.526e-05           0
   x7           0           0           0           0           0           0           0   0.0004883
   x8           0           0           0           0           0           0           0           0

  B = 
            u1       u2       u3
   x1        0        0        0
   x2        0        0        0
   x3        0        0        0
   x4        0      0.5        0
   x5        0        0        0
   x6   0.3472        0  -0.0625
   x7        0        0        0
   x8        0        0        0

  C = 
            x1       x2       x3       x4       x5       x6       x7       x8
   y1  -0.5981        0        0        0        0        0        0        0
   y2        0        0        0        0     0.25        0        0        0

  D = 
       u1  u2  u3
   y1   0   0   0
   y2   0   0   0

Continuous-time state-space model.
```

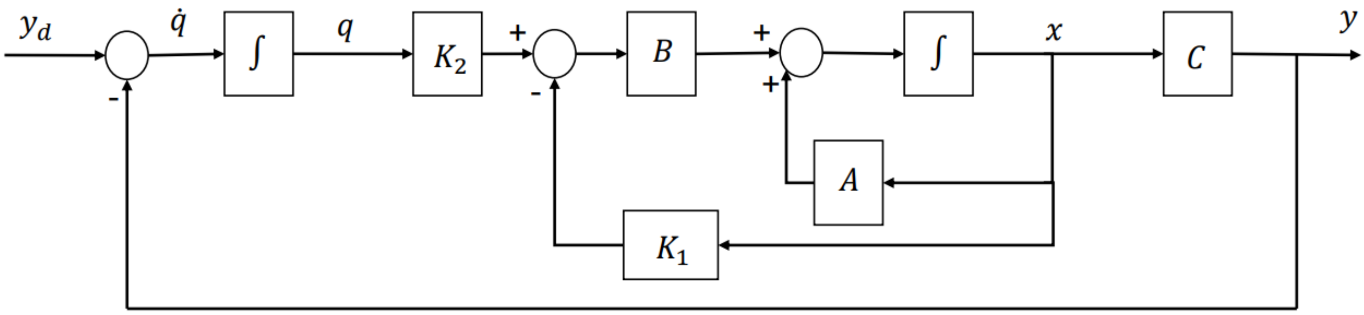
E. Integral State

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t) \end{cases}$$

$$\dot{q}(t) = y_d - y(t) = y_d - Cx(t) \quad \Rightarrow$$

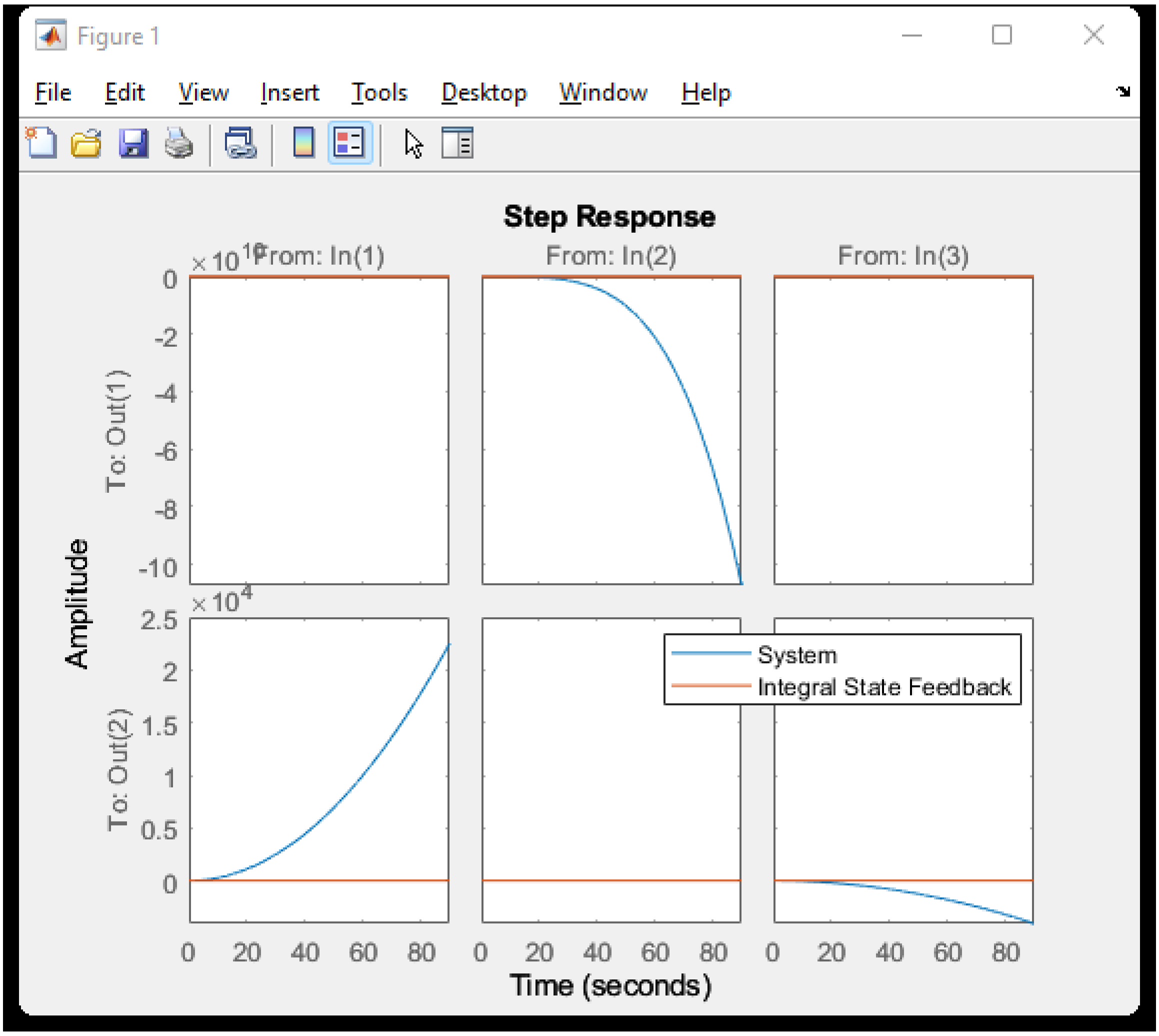
$$\begin{bmatrix} \dot{x}(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ q(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ I \end{bmatrix} y_d$$

$$y(t) = Cx(t)$$



For designing the integral state feedback first, we should check that whether it is possible or not. For this purpose, the M matrix should be full-rank. The step response of the new closed loop system has been plotted and the state space of that have been printed. The feedbacks are with respect to the desired poles.

```
% 5.2. integral state
clc
M = [A_11 B_1;-C_1 zeros(2,3)];
size(M)
rank(M)
A_int = [A_11 zeros(6,2); -C_1 zeros(2,2)];
B_int = [B_1 ; zeros(2,3)];
C_int = [C_1 zeros(2,2)];
rank(ctrb(A_int, B_int))
desired_poles_int = [-0.1 -0.2 -0.3 -0.4 -0.5 -0.6 -0.7 -0.8];
K_int = place(A_int, B_int, desired_poles_int);
K_int_1 = K_int(1:3, 1:6);
K_int_2 = -K_int(1:3, 7:8);
A_int_cl = [A_11-(B_1 * K_int_1)     (B_1 * K_int_2);
            -C_1                      zeros(2,2)];
B_int_cl = [zeros(7,3);[1 1 1]];
C_int_cl = [C_1 zeros(2)];
sys_int = ss(A_int_cl, B_int_cl, C_int_cl, 0)
step(system,sys_int);
legend('System','Integral State Feedback','Location','NorthEast');
pause
```




```
ans =

     8     9


ans =

     8


ans =

     8
```

```
sys_int =

  A =
                x1          x2          x3          x4          x5          x6          x7          x8
   x1         -0.6  -5.371e-05  -8.428e-08  -1.291e-12  -8.557e-09  -1.045e-14  -6.875e-15   5.722e-09
   x2         2048           0           0           0           0           0           0           0
   x3   -7.997e-08  -1.934e-11          -3  -1.579e-05     -0.1541  -9.841e-08  -5.357e-08   4.503e-15
   x4            0           0           0           0        5689           0           0           0
   x5            0           0       23.04           0           0           0           0           0
   x6            0           0           0       46.08           0           0           0           0
   x7            0           0           0           0           0      0.2077           0           0
   x8            0        -512           0           0           0           0           0           0

  B =
       u1  u2  u3
   x1   0   0   0
   x2   0   0   0
   x3   0   0   0
   x4   0   0   0
   x5   0   0   0
   x6   0   0   0
   x7   0   0   0
   x8   1   1   1

  C =
            x1       x2       x3       x4       x5       x6       x7       x8
   y1        0        0        0        0        0  -0.2077        0        0
   y2        0      512        0        0        0        0        0        0

  D =
       u1  u2  u3
   y1   0   0   0
   y2   0   0   0

Continuous-time state-space model.
```

F. Linear Observer

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$\dot{e}(t) = \dot{x}(t) - \dot{\hat{x}}(t) = \hat{A}\hat{x}(t) + \hat{B}u(t) - \hat{A}[x(t) - e(t)] - \hat{B}u(t) - LCx(t)$$

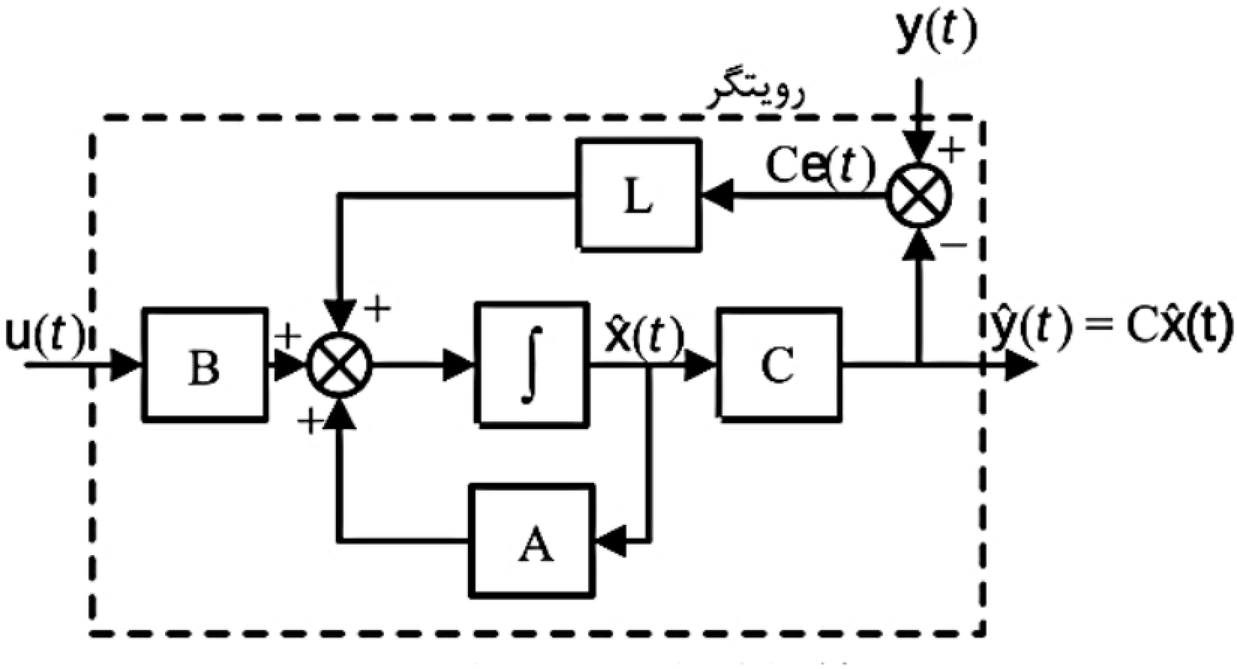
$$= \hat{A}e(t) + \left(A - LC - \hat{A}\right)x(t) + \left(B - \hat{B}\right)u(t)$$

$$\hat{A} = A - LC$$

$$\hat{B} = B$$

$$\dot{\hat{x}}(t) = (A - LC)\hat{x}(t) + Bu(t) + Ly(t)$$

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + Ly\left(t - C\hat{x}(t)\right)$$



Now we want to estimate the state variables using feedback from the output. The desired poles for the observer should be further from the poles of the system so that it can observe the system. We choose matrix be equal to identity so that we would see the inputs in the outputs.

```
% 6. linear observer
clc
desired_poles_obs = [-0.7 -0.8 -0.9 -1.0 -1.1 -1.2];
L = place(A_11', C_1', desired_poles_obs);
A_hat = [A_11   -B_1*K(1:3,1:6); L'*C_1   A_11-L'*C_1-B_1*K(1:3,1:6)];
B_hat = [B_1; zeros(6,3)];
C_hat = eye(12);
sys_hat = ss(A_hat, B_hat, C_hat, 0)
X0 = [1, 2, 3, 4, 5, 6, 0, 0, 0, 0, 0, 0];
t = 0:0.0001:0.5;
u = 0 * t;
U = [u; u; u];
[Y,T,X] = lsim(sys_hat, U, t, X0);
```

Here is the estimators state space system:

sys_hat =

A =

	x1	x2	x3	x4	x5	x6	x7	x8	x9
x1	0	0	0	0	0	0	-0.01095	-0.4358	-18.59
x2	2048	0	0	0	0	0	0	0	0
x3	0	0	0	0	0	0	-3.638e-05	-0.001978	-0.133
x4	0	0	0	0	5689	0	0	0	0
x5	0	0	23.04	0	0	0	0	0	0
x6	0	0	0	46.08	0	0	0	0	0
x7	0	0.0002929	0	0	0	-4.178e-10	-0.01095	-0.4361	-18.59
x8	0	1.548	0	0	0	-1.032e-06	2048	-1.548	0
x9	0	0.008495	0	0	0	1.715e-07	-3.638e-05	-0.01047	-0.133
x10	0	3091	0	0	0	0.1386	0	-3091	0
x11	0	0.5966	0	0	0	1.625e-05	0	-0.5966	23.04
x12	0	3.387e+04	0	0	0	4.152	0	-3.387e+04	0

	x10	x11	x12
x1	-57.12	-2.507e-06	-0.0001054
x2	0	0	0
x3	-0.8808	-8.55e-10	-1.58e-08
x4	0	0	0
x5	0	0	0
x6	0	0	0
x7	-57.12	-2.507e-06	-0.0001054
x8	0	0	1.032e-06
x9	-0.8808	-8.55e-10	-1.873e-07
x10	0	5689	-0.1386
x11	0	0	-1.625e-05
x12	46.08	0	-4.152

B =

	u1	u2	u3
x1	5.298e-06	0	-9.537e-07
x2	0	0	0
x3	0	0.03125	0
x4	0	0	0
x5	0	0	0
x6	0	0	0
x7	0	0	0
x8	0	0	0
x9	0	0	0
x10	0	0	0
x11	0	0	0
x12	0	0	0

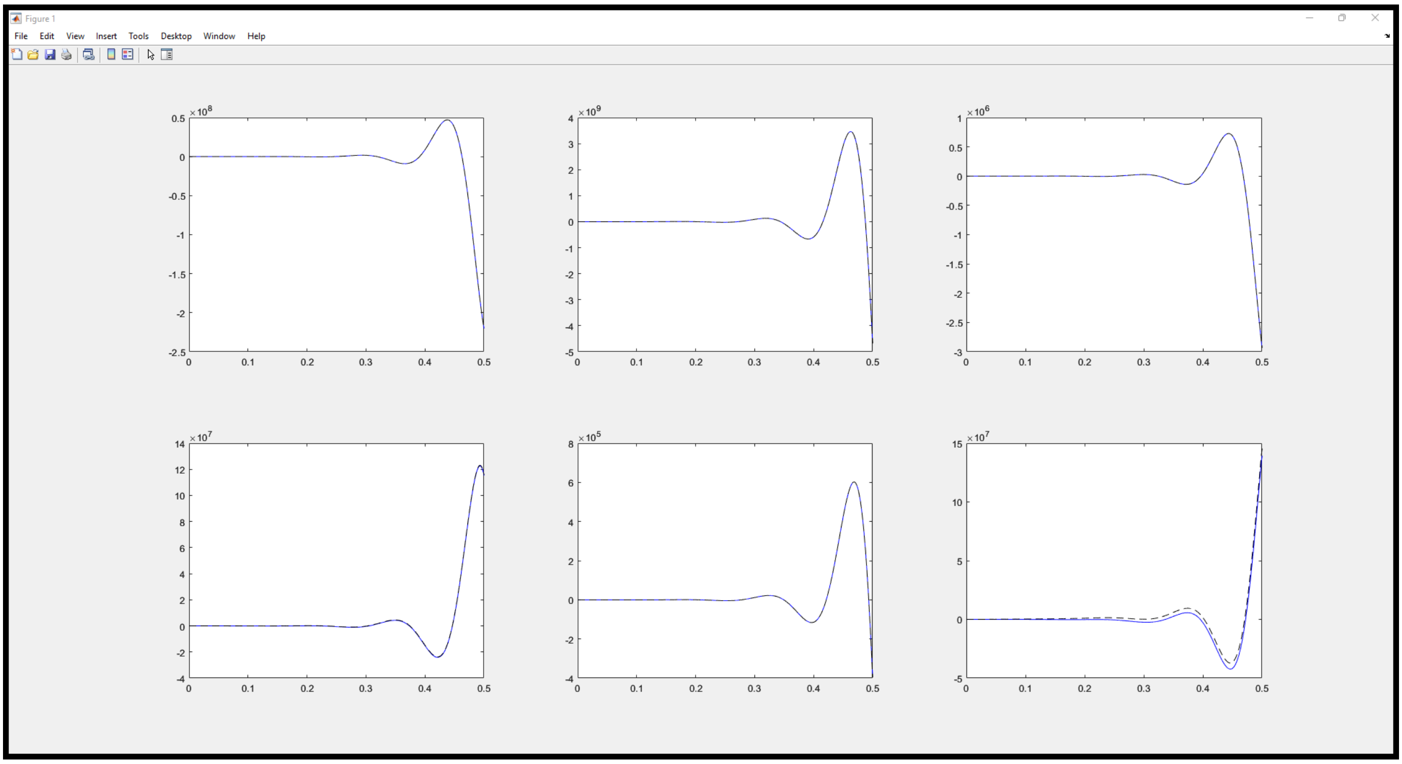
C =

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12
y1	1	0	0	0	0	0	0	0	0	0	0	0
y2	0	1	0	0	0	0	0	0	0	0	0	0
y3	0	0	1	0	0	0	0	0	0	0	0	0
y4	0	0	0	1	0	0	0	0	0	0	0	0
y5	0	0	0	0	1	0	0	0	0	0	0	0
y6	0	0	0	0	0	1	0	0	0	0	0	0
y7	0	0	0	0	0	0	1	0	0	0	0	0
y8	0	0	0	0	0	0	0	1	0	0	0	0
y9	0	0	0	0	0	0	0	0	1	0	0	0
y10	0	0	0	0	0	0	0	0	0	1	0	0
y11	0	0	0	0	0	0	0	0	0	0	1	0
y12	0	0	0	0	0	0	0	0	0	0	0	1

```
D =

       u1  u2  u3
   y1   0   0   0
   y2   0   0   0
   y3   0   0   0
   y4   0   0   0
   y5   0   0   0
   y6   0   0   0
   y7   0   0   0
   y8   0   0   0
   y9   0   0   0
  y10   0   0   0
  y11   0   0   0
  y12   0   0   0

Continuous-time state-space model.
```

And now we plot the real values of each state variable and our stimation:

```
subplot(2,3,1)
plot(t, X(:,1),'b')
hold on
plot(t, X(:,7),'k--')

subplot(2,3,2)
plot(t, X(:,2),'b')
hold on
plot(t, X(:,8),'k--')

subplot(2,3,3)
plot(t, X(:,3),'b')
hold on
plot(t, X(:,9),'k--')

subplot(2,3,4)
plot(t, X(:,4),'b')
hold on
plot(t, X(:,10),'k--')

subplot(2,3,5)
plot(t, X(:,5),'b')
hold on
plot(t, X(:,11),'k--')

subplot(2,3,6)
plot(t, X(:,6),'b')
hold on
plot(t, X(:,12),'k--')
```



We can see that somehow the linear observer can reach the real values in half of a second. This goes back to the choice of the desired poles of the observer.

Refrences:

1. Venkatesh, Praveen, Sanket Vadhvana, and Varun Jain. "Analysis and Control of a Planar Quadrotor." *arXiv preprint arXiv:2106.15134* (2021).