Case Study

The objective of this case study is to construct a projection operator unknown input observer (UIO) for a discrete time (DT) system.

We design a UIO for the following model of a DT dynamical system:

$$\begin{aligned} x[k+1] &= Ax[k] + B_1u_1[k] + B_2u_2[k] \\ &= \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} x[k] + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} u_2[k] \\ y[k] &= Cx[k] + Dv[k] \\ &= \begin{bmatrix} 0 & 2 & 1 \\ 1 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 1 \\ 0.2 \end{bmatrix} v[k]. \end{aligned}$$

Note that in the above model, we have $B_1 = O$.

Explanation: We prepared the following script to design the UIO:

```
clear all; clc;
% System data
 A=[0.5 0 0;0 0.5 0;0 0 0.5];
B2=[1 1 1]';
C=[0 2 1;1 0 0];
D=[1; 0.2];
disp('rank(obsv(A,C))')
rank(obsv(A,C))
% Dimensions
n = size(A,1);
m2=size(B2,2);
p = size(C,1);
r=size(D,2);
disp('rank(B2)')
rank(B2)
disp('rank(C*B2)')
rank(C*B2)
% Design process
%disp('M')
M=[B2 zeros(n,r)]*pinv([C*B2 D])
disp('rank(M)')
rank(M)
disp('(eye(n)-M*C)*B2')
(eye(n)-M*C)*B2
%disp('A1')
A1 = (eye(n)-M*C)*A
disp('rank(obsv(A1,C))')
rank(obsv(A1,C))
cvx_begin sdp quiet
variable P(n,n) symmetric
variable Y(n,p)
 [-P, A1'*P-C'*Y'; P*A1-Y*C, -P]<= 0
Y*D==0
P >= 0.01*eye(n)
cvx_end
%disp('L')
L = P\Y
% Checking if design objectives satisfied
disp('eig(A1-L*C)')
eig(A1-L*C)
abs(eig(A1-L*C))
disp('M*D')
M*D
disp('L*D')
L*D
```

The UIO has the form

$$\begin{aligned} z[k+1] &= (I_n - MC)(Az[k] + AMy[k] + B_1 u[k]) \\ &\quad + L(y[k] - \hat{y}[k]) \\ \hat{x}[k] &= z[k] + My[k], \end{aligned}$$

where

$$M = \begin{bmatrix} -0.5000 & 2.5000 \\ -0.5000 & 2.5000 \\ -0.5000 & 2.5000 \end{bmatrix} \quad \text{and} \quad L = \begin{bmatrix} 0.1667 & -0.8333 \\ 0.2833 & -1.4167 \\ 0.2667 & -1.3333 \end{bmatrix}.$$

The poles of the UIO are located in the open uint disk in the complex plane at $\{0.0000,\ 0.5000,\ 0.5000\}$.

We also satisfy the conditions:

$$(I_3 - MC)B_2 = O, \quad MD = O, \quad \text{and} \quad LD = O.$$

Therefore, the observation error dynamics are governed by the differential equation,

$$e[k+1] = (A_1 - LC)e[k].$$