EE49001: Control and Electronic System Design

Assignment-10: Effect of Disturbance

Submitted By:

21EE30004: Anirvan Krishna | 21EE30001: Aditya Kumar

System Without Disturbance

$$\mathrm{x}(t+1) = \begin{pmatrix} 1.5 & 0 \\ 1 & -1.5 \end{pmatrix} \mathrm{x(t)} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(t)$$

Considering a system:

$$x(t+1) = Ax(t) + Bu(t)$$

We have:

$A = \begin{pmatrix} 1.5 & 0 \\ 1 & -1.5 \end{pmatrix}$	$B = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

Open Loop Stability:

Eigen values of the matrix $A = \{-1.5, 1.5\}$. Since, one eigen value of matrix A. Therefore, the system is open loop unstable.

System Controllability:

For system controllability, the rank of the controllability matrix must be equal to n. Controllability Matrix, C is defined as:

$$C = [B \quad AB \quad A^2B \quad \dots \quad A^{n-1}B]$$

Where: n denotes the size of the state-space. Here, $n = 2$

$$\mathrm{C} = [B \quad AB]$$

Controllability Matrix (calculated using MATLAB):

$$C = \begin{pmatrix} 1 & 1.5 \\ 0 & 1 \end{pmatrix}$$

Rank of the matrix $= 2$. Therefore, system is controllable.

Determination of $\left(x(t)\right)_0^T$ for $T = 10$ for different initialization

For determining the values of $\left(x(t)\right)_0^T$ using values of $x(0)$ and $u(0)$ can be done by running the following MATLAB script. We take different initializations for $x(0)$and $u(t)$ based on the allowable range. The script runs for various initializations of $x(0)$ and different values $u(t)$ at different time steps and display the results which satisfy the constraint:

$$\begin{pmatrix} -10 \\ -10 \end{pmatrix} \le \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} \le \begin{pmatrix} +10 \\ +10 \end{pmatrix}$$

MATLAB Code

```
clear; close all; clc;

A = [
    1.5, 0;
    1, -1.5;
    ];
B = [1;0];
T = 10;

[all_x0, all_ut] = get_x0_ut(A,B,T);

C = [1;1];
wt = rand(width(C),T)*1-0.5;
xt = gen_xt_wns(A,B,C,all_x0(:,1),all_ut(1,:),wt,T);

cnt = 0;
for i = 1:width(all_x0)
    xt = gen_xt_wns(A,B,C,all_x0(:,i),all_ut(i,:),wt,T);
    cnt = cnt + check_xt(xt);
end

disp("cnt: "+cnt);
disp("all_x0: "+width(all_x0));
disp(all_x0);
disp(all_ut);

%%
function xt = gen_xt_wns(A,B,C,x0,ut,w,T)
    xt = zeros(width(A), T+1);
    xt(:,1) = x0;
    for i = 2:T+1
        xt(:,i) = A*xt(:,i-1) + B*ut(:,i-1) + C*w(:,i-1);
```

```
    end
end

function [x0_b, ut_b] = get_x0_ut(A,B,T)
x0_b = zeros(width(A),1);
ut_b = zeros(1,T);
for i = 1:100000
    ut = rand(1,T)*10 - 5;
    x0 = rand(width(A),1)*20 - 10;
    xt = gen_xt(A, B, x0, ut, T);
    if check_xt(xt)
        x0_b = [x0_b,x0];
        ut_b = [ut_b;ut];
    end
end
x0_b = x0_b(:,2:end);
ut_b = ut_b(2:end,:);
end

function xt = gen_xt(A, B, x0, ut, T)
xt = zeros(width(A),T+1);
xt(:,1) = x0;

for i = 2:T+1
    xt(:,i) = A*xt(:,i-1) + B*ut(:,i-1);
end
end

function y = check_xt(xt)
x = (xt>-10) .* (xt<10);
if sum(x(:)) == height(xt)*width(xt)
    y = 1;
else
```

```
    y = 0;
end
end
```

The results i.e. the different values of $x(0)$and $u(t)$ can be obtained by running the MATLAB file.

x_T	-3.5823	0.0884	-4.7470	-1.5480	-2.3629	2.3753	1.5414	-0.4342	2.4709	0.0616
	-1.1107	0.3886	-1.7366	0.3087	-2.2375	1.2498	0.9110	-1.5208	0.9302	0.1750
u_T	3.0684	3.8349	-0.9314	-0.1180	1.1245	2.0531	-0.7709	-3.0131	-0.2468	-3.5873
	-1.3631	1.7624	2.6733	-0.8183	-4.5067	-2.2099	0.3861	2.3122	-1.7007	1.1645
	4.3771	4.6144	-3.2609	1.2868	3.5644	-1.9445	4.7597	-3.5033	-1.7271	2.6443
	-0.8909	4.1301	-0.2884	-1.7129	3.7093	3.5961	-4.0825	3.1211	1.4399	-2.7288
	4.7744	-4.1678	0.5493	2.2589	-0.2818	1.9078	2.1354	2.9590	3.3968	-4.2597
	-0.5809	-2.1710	-4.3481	-1.5459	-2.6328	4.7630	4.8494	2.9204	2.4405	0.4100
	-4.7835	0.5965	4.1940	-2.8728	0.9563	2.1199	3.7086	1.4260	3.1571	3.4340
	3.2376	-1.6577	-1.2915	-4.8198	4.6230	-1.2963	-1.5545	-4.5335	0.0153	3.5500
	-3.8073	-0.6600	3.0898	-3.4423	1.1670	-2.4565	3.8882	1.4468	-4.4105	4.5837
	-2.4979	1.2370	3.1572	-2.7823	4.7883	-0.9389	0.6606	1.2116	-1.3462	-0.8991

System with Disturbance

We need to find the maximum range of $w(t)$ so that for the values of $x(0)$ and $u(t)$, the above equation holds true. This $w(t)$ sequence is randomly generated and is tested against every obtained seed and the number of seeds that evolve correctly are noted. A success rate parameter is defined for the each such iteration and is run for 1000 times and the mean of all such iterations is taken as a metric for further optimization. The upper bound is shrunk with a binary search algorithm that maximizes the stated metric.

External Disturbance

$$\mathrm{x}(t+1) = \begin{pmatrix} 1.5 & 0 \\ 1 & -1.5 \end{pmatrix} \mathrm{x(t)} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(t) + \begin{pmatrix} 1 \\ 1 \end{pmatrix} w(t)$$

MATLAB Code

```
clear; close all; clc;
A=[1.5 0;1 -1.5];
B=[1;0];
T = 10;


[all_x0, all_ut] = gen_x0_ut_b(A,B,T);




C = [1;1];
```

```
b1 = bin_search_bound(A,B,C,T,all_x0,all_ut,1);
b2 = bin_search_bound(A,B,C,T,all_x0,all_ut,2);
b3 = bin_search_bound(A,B,C,T,all_x0,all_ut,3);
disp("b1: "+b1);
disp("b2: "+b2);
disp("b3: "+b3);


function y = bin_search_bound(A,B,C,T,all_x0,all_ut,idx);
lo = 0;
hi = 1.5;
mid = 0;
for j = 1:30
    if lo>hi
        break;
    end
    mid = (lo+hi)/2;
    scc = zeros(1,1000);
    for i = 1:1000
        if idx==1
            scc(i) = gen_scc(A,B,C,T,all_x0,all_ut,idx,mid,0);
        elseif idx==2
            scc(i) = gen_scc(A,B,C,T,all_x0,all_ut,idx,0,mid);
        elseif idx==3
            scc(i) = gen_scc(A,B,C,T,all_x0,all_ut,idx,mid,mid);
        end
    end
    temp = mean(scc);
    if temp<0.99
        hi = mid;
    elseif temp>0.99
        lo = mid;
```

```
    end

    if(abs(temp-0.99)<1e-5)
        break
    end

    % disp("mean scc: "); disp(temp);
    % disp("m: "); disp(mid);

end
y = mid;
end

function scc = gen_scc(A,B,C,T,x0b,utb,fid,band1,band2)
wt = rand(1,T)*2*band1 - band1;
wpt = rand(1,T)*2*band2 - band2;
cnt = 0;
if fid==1
    for i = 1:width(x0b)
        xt = gen_xt_wns(A,B,C,T,x0b(:,i),utb(i,:),wt);
        cnt = cnt + check_xt(xt);
    end
elseif fid==2
    for i = 1:width(x0b)
        xt = gen_xt_wps(A,B,T,x0b(:,i),utb(i,:),wpt);
        cnt = cnt + check_xt(xt);
    end
elseif fid==3
    for i = 1:width(x0b)
        xt = gen_xt_wpns(A,B,C,T,x0b(:,i),utb(i,:),wpt,wt);
        cnt = cnt + check_xt(xt);
    end
end
```

```
scc = cnt/width(x0b);
end

function xt = gen_xt_wpns(A, B, C, T, x0, ut, wpt, wt)
xt = zeros(height(x0),T+1);
xt(:,1) = x0;
for i = 2:T+1
    xt(:,i) = (A + [wpt(:,i-1), 0; 0,0])*xt(:,i-1) + B*ut(:,i-1) + C*wt(:,i-1);
end
end

function xt = gen_xt_wps(A,B,T,x0,ut,wpt)
xt = zeros(height(x0),T+1);
xt(:,1) = x0;
for i = 2:T+1
    xt(:,i) = (A + [wpt(:,i-1), 0; 0,0])*xt(:,i-1) + B*ut(:,i-1);
end
end

function xt = gen_xt_wns(A,B,C,T,x0,ut,wt)
xt = zeros(height(x0),T+1);
xt(:,1) = x0;
for i = 2:T+1
    xt(:,i) = A*xt(:,i-1) + B*ut(:,i-1) + C*wt(:,i-1);
end
end

function [x0b, utb] = gen_x0_ut_b(A,B,T)
x0b = zeros(width(A),1);
utb = zeros(1,T);
for i = 1:1000000
    x0 = rand(width(A),1)*20 - 10;
    ut = rand(width(B),T)*10 - 5;
```

```
    xt = gen_xt(A,B,T,x0,ut);
    if check_xt(xt)
        x0b = [x0b, x0];
        utb = [utb; ut];
    end
end
x0b = x0b(:,2:end);
utb = utb(2:end,:);
end

function xt = gen_xt(A,B,T,x0,u)
xt = zeros(height(x0), T+1);
xt(:,1) = x0;
for i = 2:T+1
    xt(:,i) = A*xt(:,i-1) + B*u(:,i-1);
end
end

function y = check_xt(xt)
a = (xt>-10) .* (xt<10);
if sum(a(:)) == width(xt)*height(xt)
    y = 1;
else
    y = 0;
end
end
```

The bound for external disturbances: $w \in [-0.0039, 0.0039]$

Parametric Disturbances

$$\mathrm{x}(t+1) = \begin{pmatrix} 1.5 + w^p(t) & 0 \\ 1 & -1.5 \end{pmatrix} \mathrm{x(t)} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(t)$$

The bound for external disturbances: $w \in [-0.0018, 0.0018]$

Both Parametric and External Disturbance

$$x(t+1) = \begin{pmatrix} 1.5 + w^p(t) & 0 \\ 1 & -1.5 \end{pmatrix} x(t) + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(t) + w(t)$$

The bound for external disturbances: $w \in [-0.0016, 0.0016]$

Effect of Disturbance and Parameter Uncertainty

External disturbances and parametric uncertainties both exert significant influences on system behaviour. External disturbances, stemming from environmental factors or external inputs not accounted for in the system model, can introduce noise, bias, or fluctuations, disrupting the system's operation and leading to performance degradation.

On the other hand, parametric uncertainties, arising from variations or lack of precise knowledge in system parameters, can result in variations in system behaviour and response. Even small changes in parameter values can lead to substantial differences in system dynamics, making accurate prediction and control challenging. In combination, external disturbances and parametric uncertainties can amplify each other's effects, further complicating system behaviour and potentially compromising performance.

Example of a Practical System

One practical example where the magnitude of the system state needs to be restricted is in the control of a pendulum. Consider a pendulum used in a stabilization system, such as an inverted pendulum used in a Segway-like vehicle or a pendulum used in a robotic arm. In such systems, the angle of the pendulum from its upright position is a critical state variable that needs to be controlled within certain bounds to ensure stability and prevent the system from toppling over or becoming uncontrollable. For instance, in an inverted pendulum system for a Segway-like vehicle, the angle of the pendulum relative to the vertical needs to be restricted within a certain range to maintain balance and prevent the vehicle from tipping over. The restriction on the magnitude of the system state, in this case, ensures safe and stable operation of the system.