A picture containing fireworks, dark, water, flying

Description automatically generated

College of Engineering

School of Aeronautics and Astronautics

AAE 564

System Analysis and Synthesis

Homework 13

Output Feedback and Lyapunov Theory

*Author:*

Tomoki Koike

*Supervisor:*

Martin Corless

December 4th, 2020 Friday

Purdue University

West Lafayette, Indiana

**Exercise 1**

Determine whether or not each of the following systems are observable, detectable, or not detectable.

(a)

(b)

(c)

(a)

The state matrices are

The observability matrix

This system is observable, and therefore, this system is detectable.

(b)

The state matrices are

The observability matrix

This system is unobservable. Next, we have to find the unobservable eigenvalues with the PBH test. The eigenvalues are

If ,

This eigenvalue is observable.

If ,

This eigenvalue is unobservable.

The unobservable eigenvalue has a positive real part, and therefore, this system is not detectable.

(c)

The state matrices are

The observability matrix

This system is unobservable. Next, we have to find the unobservable eigenvalues with the PBH test. The eigenvalues are

If ,

This eigenvalue is unobservable.

The unobservable eigenvalue has a positive real part, and therefore, this system is not detectable.

**Exercise 2**

Consider the system described by

Where all quantities are scalars.

1. Is this system observable?
2. Is this system detectable?
3. Does there exist an asymptotic state estimator for this system? If an estimator does not exist, explain why; if one does exist, give an example of one.
4. If the answer to part (c) is yes, illustrate the effectiveness of your observer with a simulation.

(a)

The state matrices are

The observability matrix

This system is unobservable.

(b)

The eigenvalues are

If ,

This eigenvalue is unobservable.

If ,

This eigenvalue is unobservable.

The unobservable eigenvalue has a negative real part, and therefore, this system is detectable.

(c)

Let

we have

Hence,

And is asymptotically stable if

Thus, there exists an asymptotic state estimator.

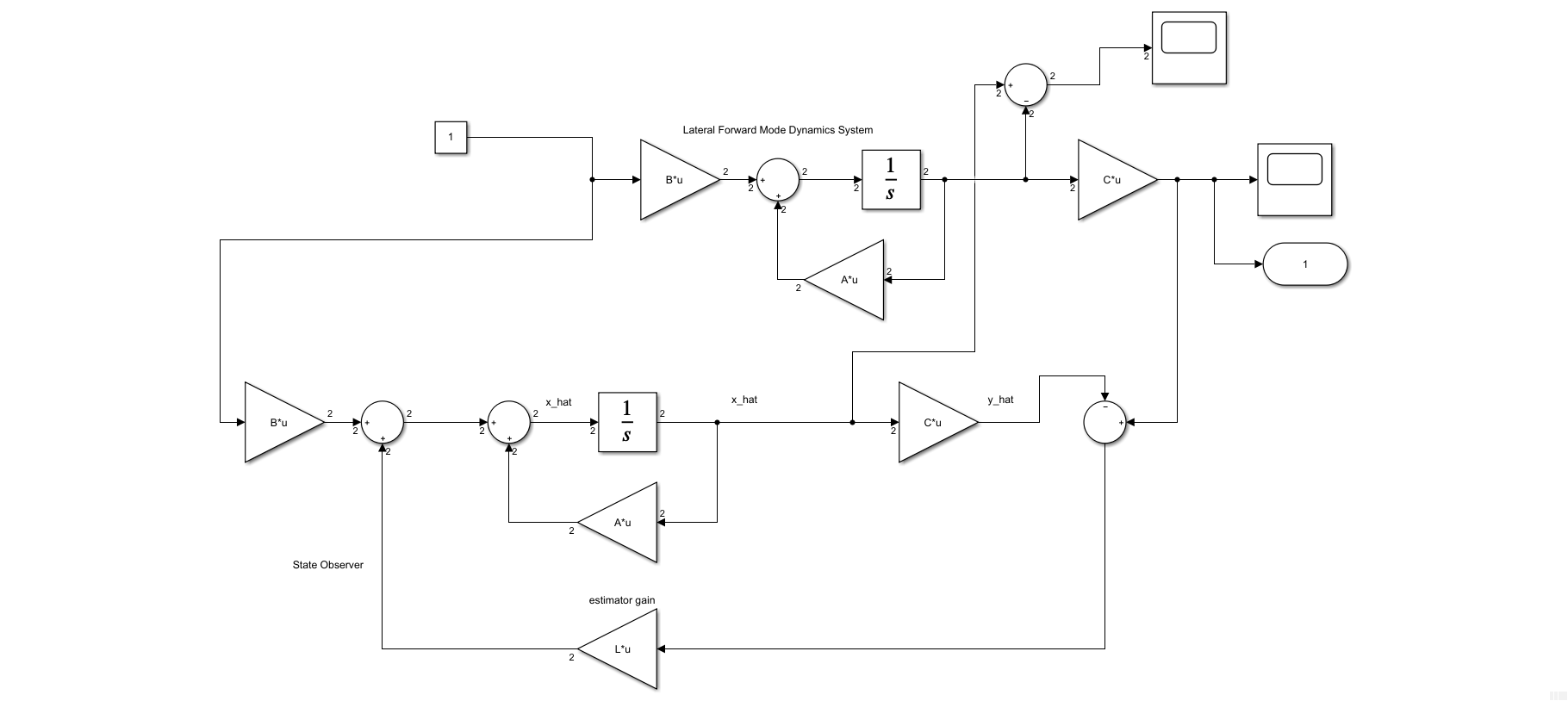
An example of an asymptotic observer is then given by

If

This is an example of an asymptotic observer.

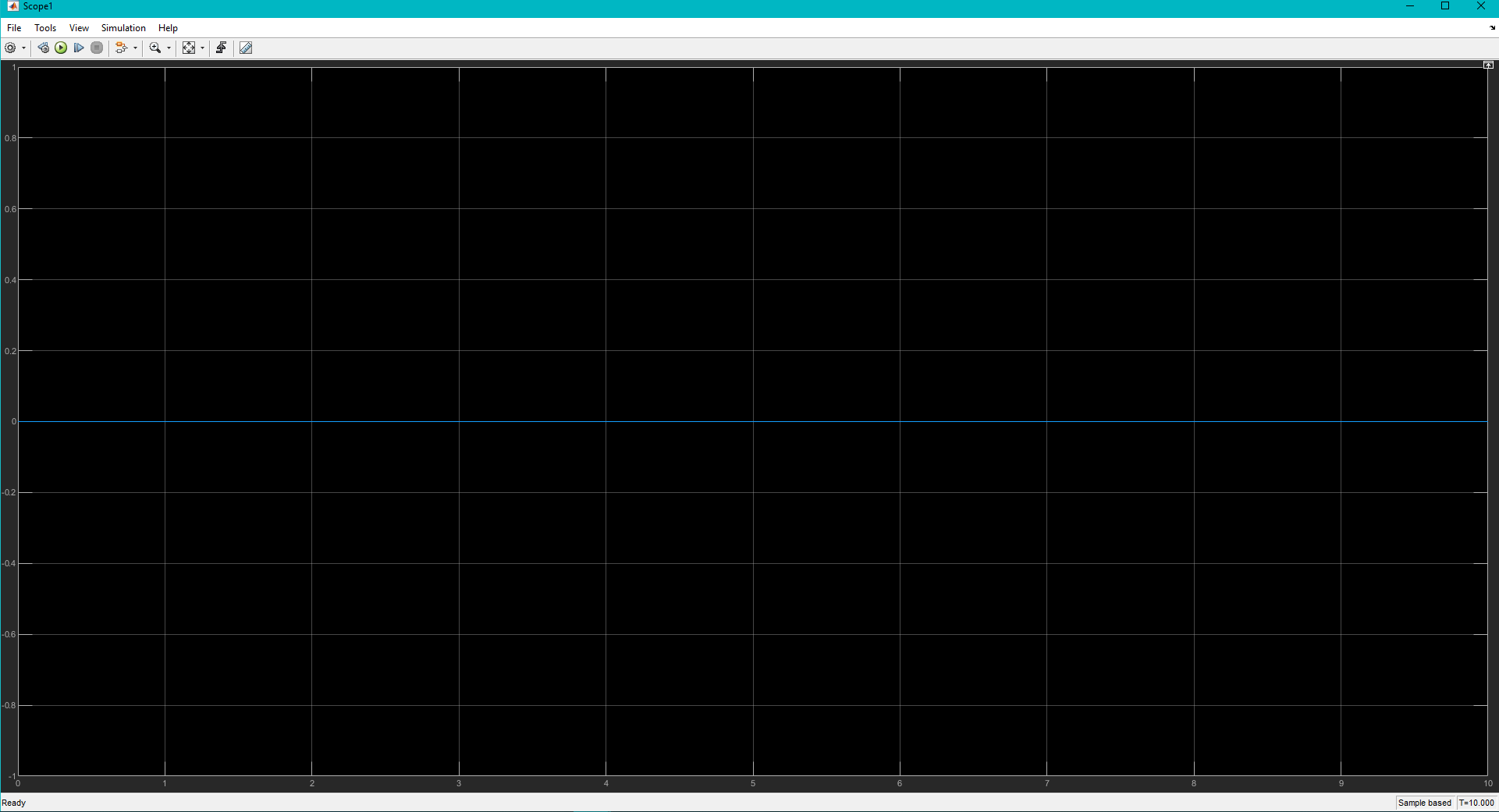
(d)

Using the following Simulink model



We define

And simulate the model. Then becomes



And the output, being



Since the difference between the estimated states and actual states are zero we can see that the observer is effective.

**Exercise 3**

Consider the system,

where all quantities are scalar. Obtain (by hand) an output feedback controller which results in an asymptotically stable closed loop system.

The state matrices are

The observability matrix

This system is observable, and thus detectable.

The controllability matrix

This system is controllable, and thus stabilizable.

Let the state feedback and observer matrix be

The closed loop system can be characterized as

Now,

Set our desired poles as .

We first find *K* for these desired poles using the Brogan’s algorithm

Find

Compute

Calculate

Where correspond to the columns of .

Find the gains with

Where

Then,

Next, for the same desired poles we will find the observer gains *L*.

The characteristic equation for the desired poles is

Now,

Thus,

Hence,

And

Now we check that it is asymptotically stable

It is asymptotically stable.

Thus, the output feedback controller is

**Exercise 4**

Consider the system

with scalar control input and scalar measured output .

1. Obtain (by hand) an observer-based output feedback controller which results in an asymptotically stable closed loop system.
2. Can all the eigenvalues of the closed loop system be arbitrarily placed?

(a)

The state matrices are

The observability matrix

This system is unobservable.

If ,

This eigenvalue is unobservable.

If ,

This eigenvalue is observable. Now, since the unobservable eigenvalue has a negative real part, this system is detectable.

The controllability matrix

This system is controllable, and thus stabilizable.

Let the state feedback and observer matrix be

Now,

The eigenvalues become asymptotically stable if all the coefficients are positive.

The eigenvalues become asymptotically stable if all the coefficients are positive.

For the *L* matrix we find the positive combinations of that satisfy the following

Which means that can be arbitrary but must satisfy . Thus, we choose

Where

And can be any value so we select

Corresponding to the coefficients , the characteristic equation becomes

Then,

Hence,

The closed loop system can be characterized as

And

Now we check that it is asymptotically stable

It is asymptotically stable.

Thus, the output feedback controller is

(b)

From the relation in part (a),

We can see that the matrix representing these three inequality equations can be represented by the matrix

This shows that the eigenvalues have to be selected so that the following relationship is satisfied

Thus, the desired poles CANNOT be selected arbitrarily.

**Exercise 5**

Consider the system,

where all quantities are scalar. Obtain (by hand) an output feedback controller which results in an asymptotically stable closed loop system.

The state matrices are

The observability matrix

This system is observable, and therefore, detectable.

The controllability matrix

This system is controllable, and thus stabilizable.

Let the state feedback and observer matrix be

Let the desired poles be , the corresponding characteristic equation becomes

Now,

Hence,

The closed loop system can be characterized as

And

Now we check that it is asymptotically stable

It is asymptotically stable.

Thus, the output feedback controller is

**Exercise 6**

(Stabilization of cart pendulum system via output feedback.) consider the cart pendulum system with the displacement as the measured output. Carry out the following for parameter P4 and equilibriums *E1* and *E2*. Illustrate the effectiveness of your controllers with numerical simulations.  
using eigenvalue placement techniques, obtain a output feedback controller which stabilizes the nonlinear system about the equilibrium.   
What is the largest value of (in degrees) for which your controller guarantees convergence of the closed loop system to the equilibrium for initial condition

Where and are the equilibrium values of and .

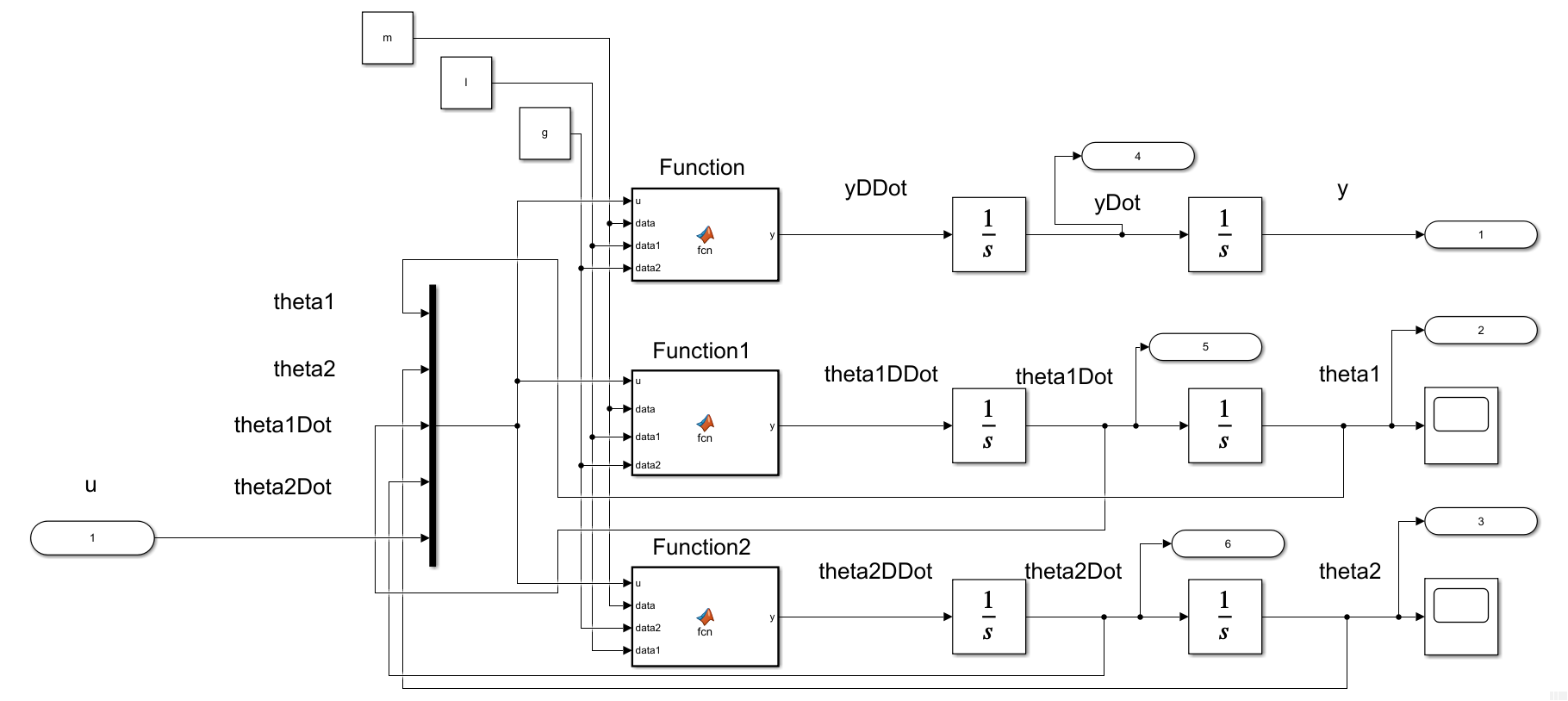
The system equation for the double pendulum cart system is

Have the system be a single output of the displacement y.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| P1 | 2 | 1 | 1 | 1 | 1 | 1 | 0 |
| P2 | 2 | 1 | 1 | 1 | 0.99 | 1 | 0 |
| P3 | 2 | 1 | 0.5 | 1 | 1 | 1 | 0 |
| P4 | 2 | 1 | 1 | 1 | 0.5 | 1 | 0 |

|  |  |  |
| --- | --- | --- |
| L1 | P1 | E1 |
| L2 | P1 | E2 |
| L3 | P2 | E1 |
| L4 | P2 | E2 |
| L5 | P3 | E1 |
| L6 | P3 | E2 |
| L7 | P4 | E1 |
| L8 | P4 | E2 |

The nonlinear Simulink model used for this is shown below,



Embedded MATLAB Block – Function (code)

function y = fcn(u, data, data1, data2)

%{

EMBEDDED MATLAB BLOCK FUNCTION

%}

m0 = data(1); m1 = data(2); m2 = data(3); l1 = data1(1); l2 = data1(2);

g = data2;

num = -m1\*l1\*sin(u(1))\*u(3)\*u(3) - m2\*l2\*sin(u(2))\*u(4)\*u(4)...

- m1\*g\*sin(u(1))\*cos(u(1)) - m2\*g\*sin(u(2))\*cos(u(2))...

+ u(5);

den = m0 + m1 + m2 - m1\*cos(u(1))^2 - m2\*cos(u(2))^2;

y = num / den;

end

Embedded MATLAB Block – Function1 (code)

function y = fcn(u, data, data1, data2)

%{

EMBEDDED MATLAB BLOCK FUNCTION1

%}

m0 = data(1); m1 = data(2); m2 = data(3); l1 = data1(1); l2 = data1(2);

g = data2;

num = -(m1\*l1\*cos(u(1))\*sin(u(1))\*u(3)\*u(3) + m2\*l2\*cos(u(1))\*sin(u(2))\*u(4)\*u(4))...

+ m2\*g\*(sin(u(1))\*cos(u(2))^2 - cos(u(1))\*sin(u(2))\*cos(u(2)))...

- (m0 + m1 + m2)\*g\*sin(u(1)) + u(5)\*cos(u(1));

den = l1\*(m0 + m1 + m2 - m1\*cos(u(1))^2 - m2\*cos(u(2))^2);

y = num / den;

end

Embedded MATLAB Block – Function2 (code)

function y = fcn(u, data, data2, data1)

%{

EMBEDDED MATLAB BLOCK FUNCTION2

%}

m0 = data(1); m1 = data(2); m2 = data(3); l1 = data1(1); l2 = data1(2);

g = data2;

num = -(m1\*l1\*cos(u(2,1))\*sin(u(1))\*u(3)\*u(3) + m2\*l2\*cos(u(2))\*sin(u(2))\*u(4)\*u(4))...

+ m1\*g\*(sin(u(2))\*cos(u(1))^2 - cos(u(2))\*sin(u(1))\*cos(u(1)))...

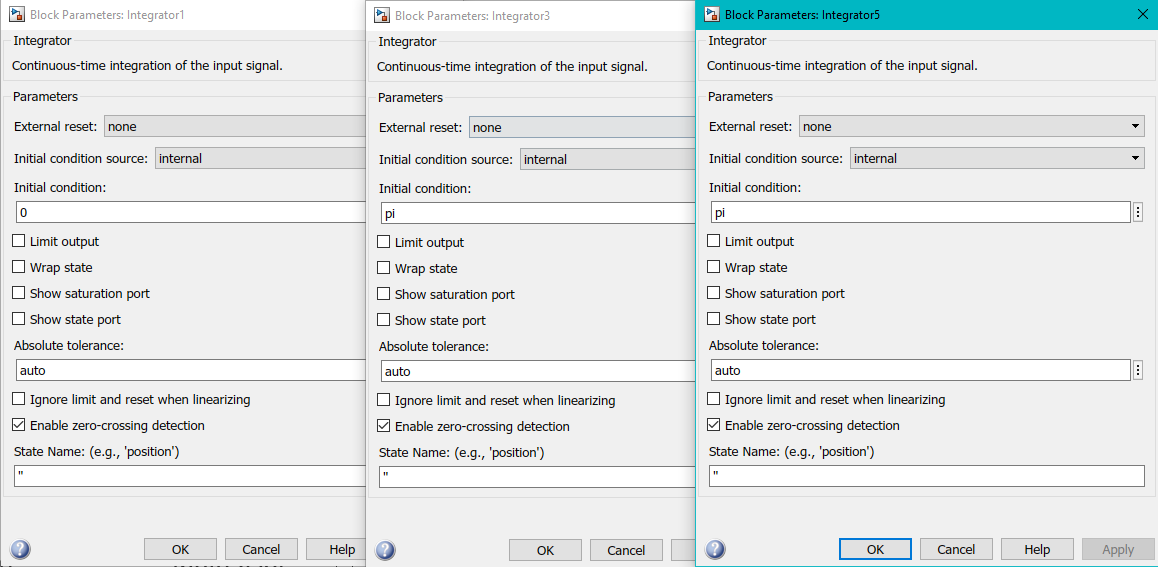
- (m0 + m1 + m2)\*g\*sin(u(2)) + u(5)\*cos(u(2));

den = l2\*(m0 + m1 + m2 - m1\*cos(u(1))^2 - m2\*cos(u(2))^2);

y = num / den;

end

For the conditions E1 and E2, we set the initial conditions of the integrator block of y, , and correspondingly to ; like in the following windows,



The system state space matrices and poles computed from the linearization for L7 and L8 are the following

L7 (P4 & E1):

|  |  |
| --- | --- |
| A = 6×6  0 0 0 1.0000 0 0  0 0 0 0 1.0000 0  0 0 0 0 0 1.0000  0 -0.5000 -0.5000 0 0 0  0 -1.5000 -0.5000 0 0 0  0 -1.0000 -3.0000 0 0 0 | B = 6×1  0  0  0  0.5000  0.5000  1.0000 |
| C = 1×6  1 0 0 0 0 0 | **D = 0** |

|  |
| --- |
| eigVal = 6×1 complex  0.0000 + 0.0000i  0.0000 + 0.0000i  0.0000 + 1.8113i  0.0000 - 1.8113i  -0.0000 + 1.1042i  -0.0000 - 1.1042i |

L8 (P4 & E2):

|  |  |
| --- | --- |
| A = 6×6  0 0 0 1.0000 0 0  0 0 0 0 1.0000 0  0 0 0 0 0 1.0000  0 -0.5000 -0.5000 0 0 0  0 1.5000 0.5000 0 0 0  0 1.0000 3.0000 0 0 0 | B = 6×1  0  0  0  0.5000  -0.5000  -1.0000 |
| C = 1×6  1 0 0 0 0 0 | **D = 0** |

|  |
| --- |
| eigVal = 6×1  0  0  -1.8113  -1.1042  1.8113  1.1042 |

Using the following MATLAB code we can simulate the nonlinear system controlled by the output feedback controller. The *K* and *L* gains are computed using the pole placement with arbitrarily selected poles.

close all; clear all; clc;

set(groot, 'defaulttextinterpreter','latex');

set(groot, 'defaultAxesTickLabelInterpreter','latex');

set(groot, 'defaultLegendInterpreter','latex');

addpath(genpath("C:\Users\Tomo\Desktop\studies\2020-Fall\AAE564\matlab\_simulink"))

% System requirements

p = [-1, -1.22, -1.5, -2, -2.3, -2.7];

global m l g ye theta1e theta2e

warning('off');

param\_combo = ["L7","L8"]; %["L7","L8"]

for i = 1:numel(param\_combo)

define\_params(param\_combo(i));

xe = trim('db\_pend\_cart\_lin');

[A, B, C, D] = linmod('db\_pend\_cart\_lin',xe);

lin\_sys(i).Amat = A;

lin\_sys(i).Bmat = B;

lin\_sys(i).Cmat = C;

lin\_sys(i).Dmat = D;

% Compute the gains

K = -place(A, B, p);

lin\_sys(i).K = K;

L = -place(A', C', p)';

lin\_sys(i).L = L;

% Output feedback matrices

Ac = A + B\*K + L\*C;

Bc = -L;

[B\_rows, B\_cols] = size(Bc);

Cc = K;

[C\_rows, C\_cols] = size(Cc);

Dc = zeros(C\_rows, B\_cols);

ICc = zeros(B\_rows, 1);

% Plotting

% Initialize figure

fig = figure('Renderer',"painters", 'Position', [10 10 900 1000]);

delta\_max = "0";

inc\_history = [];

while true

% delta = linspace(0,deg2rad(str2double(delta\_max(i))),50)

delta = str2double(delta\_max);

% Initial conditions

dyi = 0;

u = 0;

yi = ye + dyi;

theta1i = theta1e - deg2rad(delta);

theta2i = theta2e + deg2rad(delta);

IC\_ss = [yi, theta1i, theta2i, 0, 0, 0];

% Simulate

simout = sim('db\_pend\_cart\_lin\_outputFeedback');

lin\_sys(i).simout = simout;

% Plot

time = simout.tout;

data = simout.res.signals.values;

y = data(:,1);

theta1 = data(:,2);

theta2 = data(:,3);

if i == 1

if abs(theta1(end)-theta1e) > 0.1 || abs(theta2(end)-theta2e) > 0.1

break;

elseif abs(theta1(end)-theta1e) > 0.08 || abs(theta2(end)-theta2e) > 0.08

inc = 0.01;

elseif abs(theta1(end)-theta1e) > 0.05 || abs(theta2(end)-theta2e) > 0.05

inc = 0.1;

elseif abs(theta1(end)-theta1e) > 0.01 || abs(theta2(end)-theta2e) > 0.01

inc = 0.5;

elseif abs(theta1(end)-theta1e) > 0.001 || abs(theta2(end)-theta2e) > 0.001

inc = 0.8;

else

inc = 1;

end

else

if abs(theta1(end)-theta1e) > 0.1 || abs(theta2(end)-theta2e) > 0.1

break;

else

inc = 0.01;

end

end

subplot(3,1,1)

grid on; grid minor; box on;

plot(time,y)

hold on; grid on; grid minor; box on;

ylabel('y [m]')

subplot(3,1,2)

grid on; grid minor; box on;

plot(time,theta1)

hold on; grid on; grid minor; box on;

ylabel('$\theta\_1$ [rad]')

subplot(3,1,3)

grid on; grid minor; box on;

plot(time,theta2)

hold on; grid on; grid minor; box on;

ylabel('$\theta\_2$ [rad]')

delta\_max = num2str(str2double(delta\_max) + inc);

inc\_history = [inc\_history, inc];

end

hold off;

xlabel('time, [sec]')

line1 = param\_combo(i)+' Time Histories for Output Feedback Controlled';

delta\_char = compose("%d", str2double(delta\_max));

line2 = 'Cart Pendulum System for $\delta\in[0,'+delta\_char+'^{\circ}]$ - T. Koike';

title\_string = {line1,line2};

sgtitle(title\_string)

saveas(fig, 'p6\_'+param\_combo(i)+'.png');

end

function define\_params(L)

% Function to define parameters

global m l g ye theta1e theta2e

if L == "L1"

m = [2,1,1]; l = [1,1]; g = 1; % P1

ye = 0; theta1e = 0; theta2e = 0; % E1

elseif L == "L2"

m = [2,1,1]; l = [1,1]; g = 1; % P1

ye = 0; theta1e = pi; theta2e = pi; % E2

elseif L == "L3"

m = [2,1,1]; l = [1,0.99]; g = 1; % P2

ye = 0; theta1e = 0; theta2e = 0; % E1

elseif L == "L4"

m = [2,1,1]; l = [1,0.99]; g = 1; % P2

ye = 0; theta1e = pi; theta2e = pi; % E2

elseif L == "L5"

m = [2,1,0.5]; l = [1,1]; g = 1; % P3

ye = 0; theta1e = 0; theta2e = 0; % E1

elseif L == "L6"

m = [2,1,0.5]; l = [1,1]; g = 1; % P3

ye = 0; theta1e = pi; theta2e = pi; % E2

elseif L == "L7"

m = [2,1,1]; l = [1,0.5]; g = 1; % P4

ye = 0; theta1e = 0; theta2e = 0; % E1

elseif L == "L8"

m = [2,1,1]; l = [1,0.5]; g = 1; % P4

ye = 0; theta1e = pi; theta2e = pi; % E2

else

print('error: did not match any')

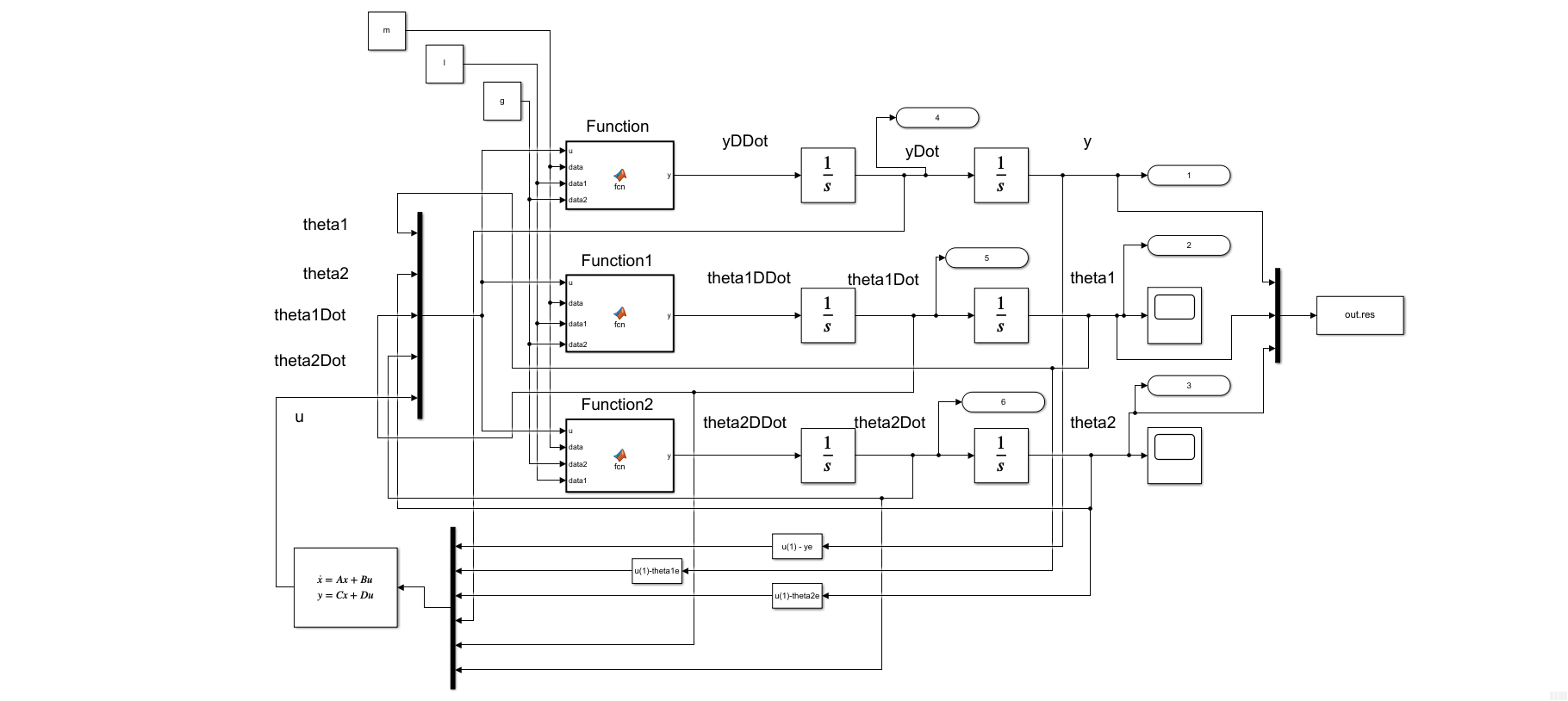
end

end

We implement an output feedback controller by finding the *K* and *L* gains using eigenvalue placement. The controller is defined as

and the poles are

The Simulink model used for the output feedback controlled system is



The simulation results are the following

L7 (P4 & E1):

Chart

Description automatically generated

The results show that .

L8 (P4 & E2):

Diagram

Description automatically generated

The results show that .

**Exercise 7**

Using the Lyapunov equation determine (by hand) whether or not the system is asymptotically stable for each one of the following *A* matrices.

(a)

(b)

(c)

Check your answers using the MATLAB command lyap.

(a)

Let a Hermitian matrix *P* be

Say

Since,

Then from the Lyapunov equation

Solving these equations, we get

Thus, the system is asymptotically stable.

MATLAB Verification:

% verify

A = [-1, 2; 0, -1];

Q = [1, -1; -1, 2];

P = lyap(A', Q)

eigVal = eig(P)

|  |  |
| --- | --- |
| P = 2×2  0.5000 0  0 1.0000 | eigVal = 2×1  0.5000  1.0000 |

(b)

Let a Hermitian matrix *P* be

Say

Since,

Then from the Lyapunov equation

Solving these equations, we get

Thus, the system is NOT asymptotically stable.

MATLAB Verification:

% verify

A = [-1, 2; 0, 1];

Q = [1, -1; -1, 2];

P = lyap(A', Q)

Error using [**lyap**](matlab:matlab.internal.language.introspective.errorDocCallback('lyap',%20'C:\Program%20Files\MATLAB\R2020a\toolbox\control\control\lyap.m',%2073)) ([line 73](matlab:%20opentoline('C:\Program%20Files\MATLAB\R2020a\toolbox\control\control\lyap.m',73,0)))  
The solution of this Lyapunov equation does not exist or is not unique.

(c)

Let a Hermitian matrix *P* be

Say

Since,

Then from the Lyapunov equation

Solving these equations, we get

Thus, the system is NOT asymptotically stable.

MATLAB Verification:

% verify

A = [1, 2; 0, 1];

Q = [1, -1; -1, 2];

P = lyap(A', Q)

eigVal = eig(P)

|  |  |
| --- | --- |
| P = 2×2  -0.5000 1.0000  1.0000 -3.0000 | eigVal = 2×1  -3.3508  -0.1492 |

**Exercise 8**

Consider the system with disturbance input and performance output described by

Using an appropriate Lyapunov equation, determine (by hand)

for each of the following situations.

(a)

(b)

From the given system we know that

Then

Let

Then solving the Lyapunov equation we get

(a)

With and

(b)

With and