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 5

Eigenvalues and Vectors of LTI Systems

*Author:*

Tomoki Koike

*Supervisor:*

Martin Corless

October 2nd, 2020 Friday

Purdue University

West Lafayette, Indiana

**Exercise 1**

Compute the **eigenvalues** and **eigenvectors** of the matrix

From the definition of eigenvalues where ,

Thus, the eigenvalues are

When

For this matrix conduct a Gaussian Elimination.

Cancel the leading column of row 2:

Cancel the leading column of row 3:

Swap the rows 2 and 3:

Divide row 1 by -3 and row 2 by 2:

Cancel column 2 of row 1:

From the matrix, expression for will become

Let be a free variable. Then,

Thus, the corresponding eigenvector is

When

For this matrix conduct a Gaussian Elimination.

Swap rows 1 and 3:

Cancel leading column in row 2:

Cancel leading column in row 3:

Divide row 1 by -3, multiply row 2 and 3 by -3:

Cancel out row 3 with row 2:

Cancel column 2 of row 1:

From the matrix, expression for will become

Let be a free variable. Then,

Thus, the corresponding eigenvector is

When

For this matrix conduct a Gaussian Elimination.

Swap rows 1 and 3:

Cancel leading column in row 2:

Cancel leading column in row 3:

Cancel row 3:

Multiply row 2 by 3/2 and then cancel the second column in row 1:

Divide the first row by -3

From the matrix, expression for will become

Let be a free variable. Then,

Thus, the corresponding eigenvector is

MATLAB Verification

% Exercise 1

A = [2,-3,0;2,-3,0;3,-5,1];

[v, d] = eig(A)

|  |  |
| --- | --- |
| v = 3×3  0 -0.5774 -0.8018  0 -0.5774 -0.5345  1.0000 -0.5774 -0.2673 | d = 3×3  1.0000 0 0  0 -1.0000 0  0 0 0.0000 |

**Exercise 2**

Compute the **eigenvalues** and **eigenvectors** of the matrix

From the definition of eigenvalues where ,

Thus, the eigenvalues are

When ,

Perform Gaussian Elimination on this matrix.

Cancel the leading column of row 4:

Cancel the second column of row 4:

Cancel row 4 with row 3:

Cancel column 2 of row 1 with row 2:

Cancel column 3 or row 2 with row 3:

Cancel column 3 or row 1 with row 3:

Multiply the matrix with -1

From the matrix, expression for will become

Let be a free variable. Then,

Thus, the corresponding eigenvector is

When ,

Cancel the leading column of row 4 with row 1:

Cancel the leading column of row 4 and column 2 of row 1 with row 2:

Cancel row 4 with row 3:

Cancel column 2 of row 1 and 2 with row 3

From the matrix, expression for will become

Let be a free variable. Then,

Thus, the corresponding eigenvector is

When ,

Cancel the leading column of row 4:

Cancel the leading column of row 4 with row 2:

Cancel row 4 with row 3:

Cancel column 2 or row 1 with row 2:

Cancel column 3 of row 2 with row 3:

Cancel column 3 of row 1 with row 3:

Multiply row 1 by , divide row 2 by , and multiply row 3 by

From the matrix, expression for will become

Let be a free variable. Then,

Thus, the corresponding eigenvector is

When ,

MATLAB Verification

% Exercise 2

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

[v, d] = eig(A)

|  |
| --- |
| v = 4×4 complex  -0.5000 + 0.0000i 0.0000 - 0.5000i 0.0000 + 0.5000i -0.5000 + 0.0000i  0.5000 + 0.0000i 0.5000 + 0.0000i 0.5000 - 0.0000i -0.5000 + 0.0000i  -0.5000 + 0.0000i -0.0000 + 0.5000i -0.0000 - 0.5000i -0.5000 + 0.0000i  0.5000 + 0.0000i -0.5000 + 0.0000i -0.5000 + 0.0000i -0.5000 + 0.0000i |
| d = 4×4 complex  -1.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i  0.0000 + 0.0000i 0.0000 + 1.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i  0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 - 1.0000i 0.0000 + 0.0000i  0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 1.0000 + 0.0000i |

**Exercise 3**

Determine whether or not the following matrix is nondefective.

From the definition of eigenvalues where ,

Thus, the eigenvalues are

We must be careful because -1 is a repeated eigenvalue.

When ,

The reduced echelon form of this matrix is

Considering , express this with ,

Let , as a free variable,

Thus, the corresponding eigenvector is

When ,

The reduced echelon form of this matrix is

Considering , express this with ,

Let , as a free variable,

Thus, the corresponding eigenvector is

The algebraic multiplicity, and geometric multiplicity, are

Thus, matrix A is defective.

MATLAB Verification

% Exercise 3

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

[v, d] = eig(A)

ech1 = rref(real(d(1,1))\*eye(3) - A);

ech2 = rref(real(d(3,3))\*eye(3) - A);

|  |  |
| --- | --- |
| v = 3×3 complex  0.5774 - 0.0000i 0.5774 + 0.0000i -0.5774 + 0.0000i  0.5774 - 0.0000i 0.5774 + 0.0000i 0.5774 + 0.0000i  0.5774 + 0.0000i 0.5774 + 0.0000i -0.5774 + 0.0000i | |
| d = 3×3 complex  1.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i  0.0000 + 0.0000i 1.0000 - 0.0000i 0.0000 + 0.0000i  0.0000 + 0.0000i 0.0000 + 0.0000i -1.0000 + 0.0000i | |
| ech1 = 3×3  1.0000 0 -1.0000  0 1.0000 -1.0000  0 0 0 | ech2 = 3×3  1 0 -1  0 1 1  0 0 0 |

**Exercise 4**

What is the companion matrix whose eigenvalues are ?

From the eigenvalues, we can find the following polynomial equation

Thus, the companion matrix becomes

MATLAB Verification

% Exercise 4

u = [1, 6, 11, 6];

C = compan(u)

|  |
| --- |
| C = 3×3  -6 -11 -6  1 0 0  0 1 0 |

**Exercise 5**

What is the real 2 x 2 companion matrix with eigenvalues ?

From the eigenvalues, we can find the following polynomial equation

Thus, the companion matrix becomes

MATLAB Verification

% Exercise 5

u = [1, -2, 2];

C = compan(u)

|  |
| --- |
| C = 2×2  2 -2  1 0 |

**Exercise 6**

Suppose is a real square matrix and the vectors

are eigenvectors of corresponding to eigenvalues -1 and 2 and , respectively. What is the response of the system to the following initial conditions.

is a nondefective matrix.

Thus, for each (a)~(d) we only have to find the coefficients that satisfy

(a).

We must solve the relation

Therefore, we solve the augmented matrix

The row reduced echelon form turns out to be

The coefficients become

Hence,

(b).

We must solve the relation

Therefore, we solve the augmented matrix

The row reduced echelon form turns out to be

The coefficients become

Hence,

(c).

We must solve the relation

Therefore, we solve the augmented matrix

The row reduced echelon form turns out to be

The coefficients become

Hence,

(d).

We must solve the relation

Therefore, we solve the augmented matrix

The row reduced echelon form turns out to be

The coefficients become

Hence,

Rule out all the imaginary terms

MATLAB Verification

% Exercise 6

A1 = [1,1,1,1,2; -1,2,1j,-1j,-2; 1,4,-1,-1,2; -1,8,-1j,1j,-2];

E1 = rref(A1)

A2 = [1,1,1,1,-1; -1,2,1j,-1j,-2; 1,4,-1,-1,-4; -1,8,-1j,1j,-8];

E2 = rref(A2)

A3 = [1,1,1,1,0; -1,2,1j,-1j,3; 1,4,-1,-1,3; -1,8,-1j,1j,9];

E3 = rref(A3)

A4 = [1,1,1,1,1; -1,2,1j,-1j,1; 1,4,-1,-1,-1; -1,8,-1j,1j,-1];

E4 = rref(A4)

|  |  |
| --- | --- |
| E1 = 4×5  1 0 0 0 2  0 1 0 0 0  0 0 1 0 0  0 0 0 1 0 | E2 = 4×5  1 0 0 0 0  0 1 0 0 -1  0 0 1 0 0  0 0 0 1 0 |
| E3 = 4×5  1 0 0 0 -1  0 1 0 0 1  0 0 1 0 0  0 0 0 1 0 | |
| E4 = 4×5 complex  1.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i  0.0000 + 0.0000i 1.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i  0.0000 + 0.0000i 0.0000 + 0.0000i 1.0000 + 0.0000i 0.0000 + 0.0000i 0.5000 - 0.5000i  0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 1.0000 + 0.0000i 0.5000 + 0.5000i | |

**Exercise 7**

Consider a discrete-time LTI system described by .

1. Suppose that the vectors

and

are eigenvectors of corresponding to eigenvalues -2 and 3, respectively. What is the response of the system to the initial condition

1. Suppose is a real matrix and the vector

is an eigenvector of corresponding to the eigenvalue . What is the response of the system to the initial condition

(a).

Assuming that the matrix is nondefective, we can express the given initial conditions with the linear function with coefficients . This is because we can say that matrix is 4 x 4 and has 4 linearly independent eigenvectors. We do not know the other two eigenvalues and corresponding eigenvectors but we can still express the following.

Then,

A combination of satisfying this relation is

Thus, the response becomes,

(b).

Similar to part (a), assume a nondefective real matrix to have 4 linearly independent eigenvectors. We have 2 known eigenvalues and eigenvectors, so we can express the following,

Then,

A combination of satisfying this relation is

Thus, the response becomes,

**Exercise 8**

(You may use MATLAB) Recall the 2-pendulum cart system. Consider the equilibrium configurations defined by

Consider state space representations of the linearizations corresponding to the following combinations of parameters and equilibrium conditions:

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

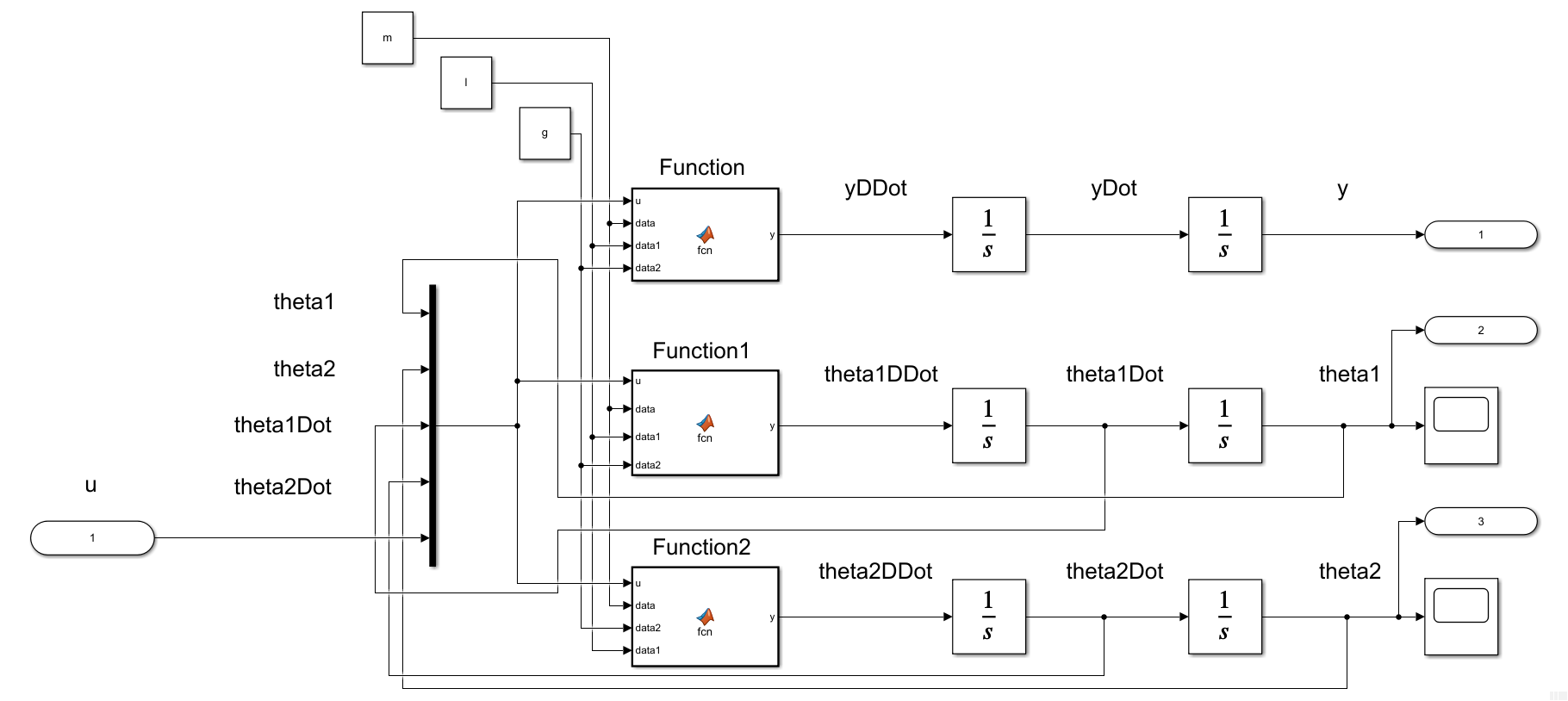
1. Determine the eigenvalues of all the linearized systems L1-L8.
2. Compare the behavior of the nonlinear system with that of the linearized system for cases L7 and L8. Illustrate your results with time histories of , , and .

Defined parameter sets:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| 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 |

(a).

The 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)))...

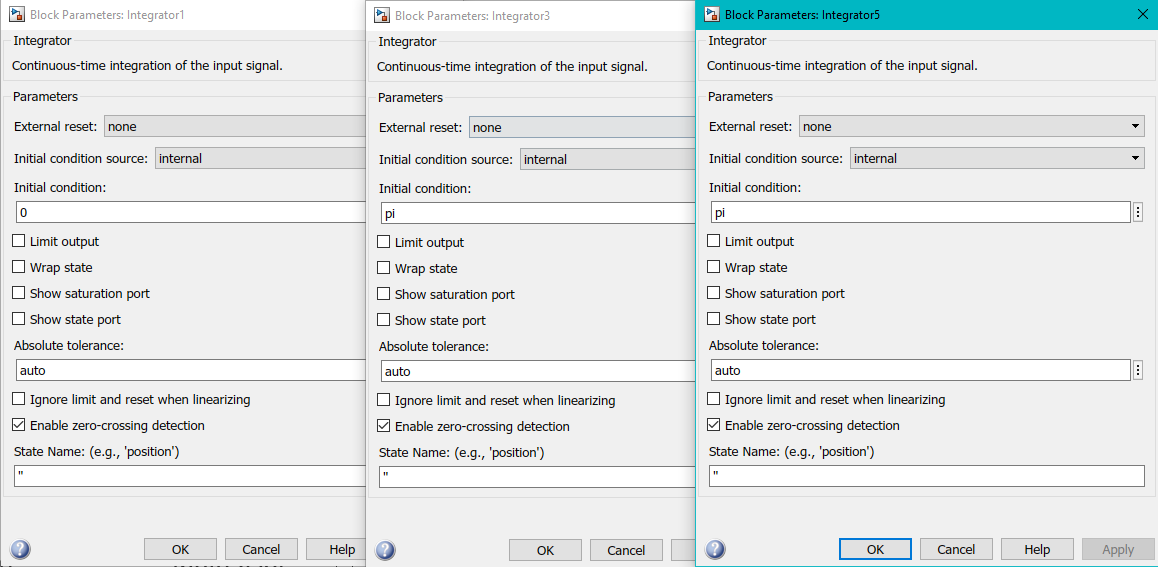
- (m0 + m1 + m2)\*g\*sin(u(2)) + u(5)\*cos(u(2));

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 code to run the linearization and eigenvalue computation is the following

% (a)

global m l g ye t1e t2e

param\_combo = ["L1","L2","L3","L4","L5","L6","L7","L8"];

for i = 1:numel(param\_combo)

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

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

lin\_sys(i).Amat = A;

lin\_sys(i).Bmat = B;

lin\_sys(i).Cmat = C;

lin\_sys(i).Dmat = D;

sys\_ss = ss(A, B, C, D); % get the state space system

sys\_tf = tf(sys\_ss); % get the transfer function

lin\_sys(i).eigVal = pole(sys\_tf); % get the eigenvalues

end

function define\_params(L)

% Function to define parameters

global m l g ye t1e t2e

if L == "L1"

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

ye = 0; t1e = 0; t2e = 0; % E1

elseif L == "L2"

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

ye = 0; t1e = pi; t2e = pi; % E2

elseif L == "L3"

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

ye = 0; t1e = 0; t2e = 0; % E1

elseif L == "L4"

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

ye = 0; t1e = pi; t2e = pi; % E2

elseif L == "L5"

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

ye = 0; t1e = 0; t2e = 0; % E1

elseif L == "L6"

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

ye = 0; t1e = pi; t2e = pi; % E2

elseif L == "L7"

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

ye = 0; t1e = 0; t2e = 0; % E1

elseif L == "L8"

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

ye = 0; t1e = pi; t2e = pi; % E2

else

print('error: did not match any')

end

end

List the state space matrices A, B, C, and D for the linearized system and eigenvalues.

L1:

|  |  |
| --- | --- |
| 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 -0.5000 -1.5000 0 0 0 | B = 6×1  0  0  0  0.5000  0.5000  0.5000 |
| 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 + 1.4142i  0.0000 - 1.4142i  -0.0000 + 1.0000i  -0.0000 - 1.0000i |

L2:

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

|  |
| --- |
| eigVal = 6×1  0  0  -1.4142  -1.0000  1.4142  1.0000 |

L3:

|  |  |
| --- | --- |
| 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 -0.5051 -1.5152 0 0 0 | B = 6×1  0  0  0  0.5000  0.5000  0.5051 |
| 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 + 1.4178i  -0.0000 - 1.4178i  0.0000 + 1.0025i  0.0000 - 1.0025i |

L2:

|  |  |
| --- | --- |
| 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 0.5051 1.5152 0 0 0 | B = 6×1  0  0  0  0.5000  -0.5000  -0.5051 |
| C = 1×6  1 0 0 0 0 0 | **D = 0** |

|  |
| --- |
| eigVal = 6×1  0  0  1.4178  1.0025  -1.4178  -1.0025 |

L5:

|  |  |
| --- | --- |
| 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 -0.2500 0 0 0  0 -1.5000 -0.2500 0 0 0  0 -0.5000 -1.2500 0 0 0 | B = 6×1  0  0  0  0.5000  0.5000  0.5000 |
| 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 + 1.3229i  -0.0000 - 1.3229i  -0.0000 + 1.0000i  -0.0000 - 1.0000i |

L6:

|  |  |
| --- | --- |
| 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 -0.2500 0 0 0  0 1.5000 0.2500 0 0 0  0 0.5000 1.2500 0 0 0 | B = 6×1  0  0  0  0.5000  -0.5000  -0.5000 |
| C = 1×6  1 0 0 0 0 0 | **D = 0** |

|  |
| --- |
| eigVal = 6×1  0  0  1.3229  1.0000  -1.3229  -1.0000 |

L7:

|  |  |
| --- | --- |
| 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:

|  |  |
| --- | --- |
| 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 |

(b).

For the initial condition,

Thus,

If the ICs defined in HW1 Exercise 7 is used,

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| IC1 | 0 | -10° | 10° | 0 | 0 | 0 |
| IC2 | 0 | 10° | 10° | 0 | 0 | 0 |
| IC3 | 0 | -90° | 90° | 0 | 0 | 0 |
| IC4 | 0 | -90.01° | 90° | 0 | 0 | 0 |
| IC5 | 0 | 100° | 100° | 0 | 0 | 0 |
| IC6 | 0 | 100.01° | 100° | 0 | 0 | 0 |
| IC7 | 0 | 179.99° | 0° | 0 | 0 | 0 |

Say we use IC1 for the initial condition,

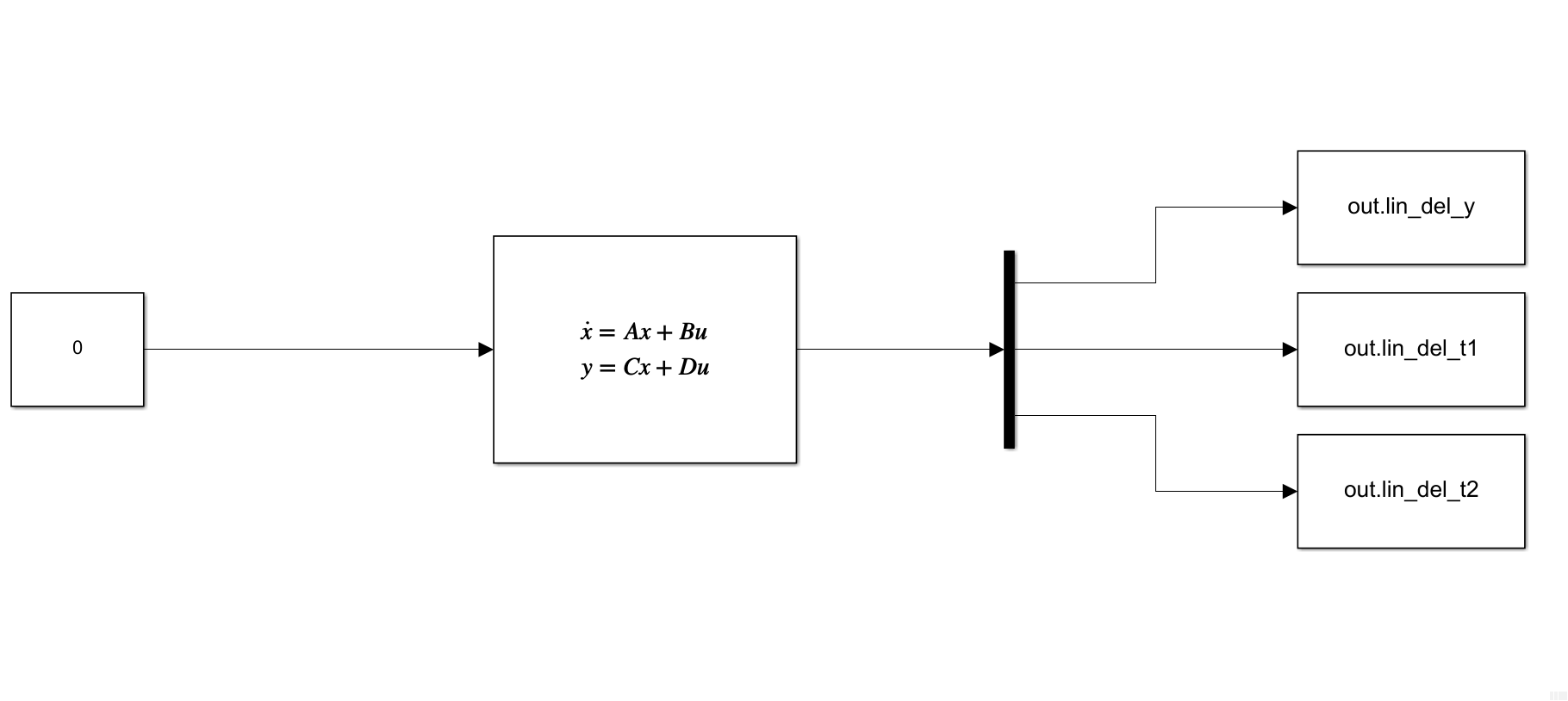
For L7, the initial condition

Then,

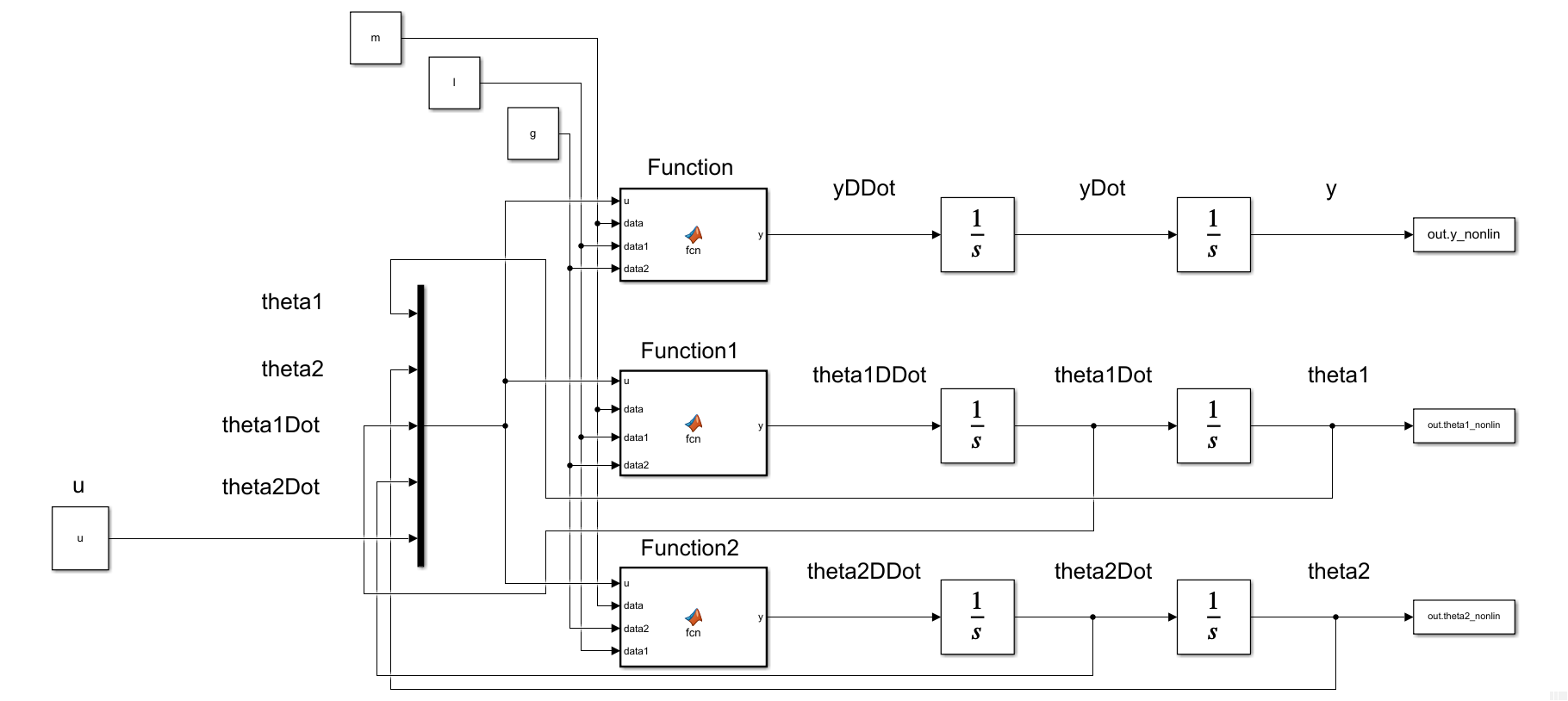
We plug these initial conditions to the following models, (\*making sure that this we have 3 outputs for the displacement, angle 1, and angle 2.)

|  |  |
| --- | --- |
| C = 3×6  1 0 0 0 0 0  0 1 0 0 0 0  0 0 1 0 0 0 | D = 3×1  0  0  0 |

The linearized state space model:



The nonlinear model:



(\*The Embedded MATLAB Blocks have the same functions as the previous)

Then the compared response for the two will be plotted out with the following code

% L7

% linear

global m l g ye t1e t2e

define\_params("L7");

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

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

del\_yi = 0; del\_t1i = deg2rad(-10); del\_t2i = deg2rad(10);

u = 0; yi = ye + del\_yi; t1i = t1e + del\_t1i; t2i = t2e + del\_t2i;

IC\_lin = [del\_yi, del\_t1i, del\_t2i, 0, 0, 0];

sim\_lin\_res = sim('ss\_lin\_sys');

time1 = sim\_lin\_res.lin\_del\_y.time;

dy = sim\_lin\_res.lin\_del\_y.signals.values;

dt1 = sim\_lin\_res.lin\_del\_t1.signals.values;

dt2 = sim\_lin\_res.lin\_del\_t2.signals.values;

% non-linear

sim\_nonlin\_res = sim('db\_pend\_cart\_nonlin.slx');

time2 = sim\_nonlin\_res.tout;

y = sim\_nonlin\_res.y\_nonlin.signals.values;

t1 = sim\_nonlin\_res.theta1\_nonlin.signals.values;

t2 = sim\_nonlin\_res.theta2\_nonlin.signals.values;

% Plotting

fig1 = figure('Renderer',"painters", 'Position', [10 10 900 1000]);

subplot(3,1,1)

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

plot(time1,dy)

plot(time2,y)

ylabel('y [m]')

hold off

subplot(3,1,2)

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

plot(time1,dt1)

plot(time2,t1)

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

hold off

subplot(3,1,3)

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

plot(time1,dt2)

plot(time2,t2)

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

xlabel('time, [sec]')

h = legend('linear','nonlinear'); set(h, 'Position', [0.8, 0.05, .1, .025]);

hold off

sgtitle('Time Histories L7 for Nonlinearized and Linearized - T. Koike')

saveas(fig1, 'p8\_L7.png')

A picture containing graphical user interface

Description automatically generated

For L8, we use the same initial condition

But since,

We use these as our initial conditions as well as the A, B, C, D matrices for L8 condition. The code we run for this simulation is the following.

% L8

% linear

global m l g ye t1e t2e

define\_params("L8");

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

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

del\_yi = 0; del\_t1i = deg2rad(-10); del\_t2i = deg2rad(10);

u = 0; yi = ye + del\_yi; t1i = t1e + del\_t1i; t2i = t2e + del\_t2i;

IC\_lin = [del\_yi, del\_t1i, del\_t2i, 0, 0, 0];

sim\_lin\_res = sim('ss\_lin\_sys');

time1 = sim\_lin\_res.lin\_del\_y.time;

dy = sim\_lin\_res.lin\_del\_y.signals.values;

dt1 = sim\_lin\_res.lin\_del\_t1.signals.values;

dt2 = sim\_lin\_res.lin\_del\_t2.signals.values;

% non-linear

sim\_nonlin\_res = sim('db\_pend\_cart\_nonlin.slx');

time2 = sim\_nonlin\_res.tout;

y = sim\_nonlin\_res.y\_nonlin.signals.values;

t1 = sim\_nonlin\_res.theta1\_nonlin.signals.values;

t2 = sim\_nonlin\_res.theta2\_nonlin.signals.values;

% Plotting

fig1 = figure('Renderer',"painters", 'Position', [10 10 900 1000]);

subplot(3,1,1)

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

plot(time1,dy)

plot(time2,y)

ylabel('y [m]')

hold off

subplot(3,1,2)

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

plot(time1,dt1)

plot(time2,t1)

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

hold off

subplot(3,1,3)

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

plot(time1,dt2)

plot(time2,t2)

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

xlabel('time, [sec]')

h = legend('linear','nonlinear'); set(h, 'Position', [0.8, 0.05, .1, .025]);

hold off

sgtitle('Time Histories L7 for Nonlinearized and Linearized - T. Koike')

saveas(fig1, 'p8\_L8.png')

The response becomes

Calendar

Description automatically generated

**Exercise 9**

(You may use MATLAB) This exercise refers to linearizations L7 and L8 of the two pendulum cart system.

1. For L7 choose an initial state for the linearized system which results in a periodic solution for the linearized system.
2. For L8 choose an initial state for the linearized system which results in a solution which asymptotically decays to zero for the linearized system.
3. For L8 choose an initial state for the linearized system which results in a solution whose magnitude grows exponentially for the linearized system.

In each case, simulate both the linearized system and the nonlinear system with initial conditions corresponding to your chosen initial state for the linearized system.

(a).

The matrix for this condition is

and the corresponding eigenvalues and eigenvectors are

|  |  |
| --- | --- |
|  |  |
| 0 | 1  0  0  0  0  0 |
| 0 | -1.0000  0  0  0.0000  0  0 |
| -0.0000 + 1.8113i | -0.0000 - 0.0893i  -0.0000 - 0.1284i  -0.0000 - 0.4573i  0.1617 - 0.0000i  0.2326 - 0.0000i  0.8283 + 0.0000i |
| -0.0000 - 1.8113i | -0.0000 + 0.0893i  -0.0000 + 0.1284i  -0.0000 + 0.4573i  0.1617 + 0.0000i  0.2326 + 0.0000i  0.8283 + 0.0000i |
| 0.0000 + 1.1042i | 0.0000 - 0.1040i  0.0000 - 0.5782i  -0.0000 + 0.3247i  0.1148 + 0.0000i  0.6385 + 0.0000i  -0.3585 + 0.0000i |
| 0.0000 - 1.1042i | 0.0000 + 0.1040i  0.0000 + 0.5782i  -0.0000 - 0.3247i  0.1148 - 0.0000i  0.6385 + 0.0000i  -0.3585 - 0.0000i |

Thus, the response is characterized by

To have a periodic response we want to have and to be non-zero and other coefficients to be 0 while the IC to be real.

This, from the Euler’s equation, we can tell is a periodic response. Thus, when , , and

Using the models from Exercise 8, we plug the initial condition above to make the behavior of the linearized system to be periodic. The result follows.

A picture containing calendar

Description automatically generated

% L7

% linear

global m l g ye t1e t2e

define\_params("L7");

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

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

[eigVec, eigVal] = eig(A,"vector");

temp = 0.5\*(eigVec(:,3) + eigVec(:,4));

% ICs

del\_yi = temp(1); del\_t1i = temp(2); del\_t2i = temp(3);

del\_yi\_dot = temp(4); del\_t1i\_dot = temp(5); del\_t2i\_dot = temp(6);

u = 0; yi = ye + del\_yi; t1i = t1e + del\_t1i; t2i = t2e + del\_t2i;

IC\_lin = [del\_yi, del\_t1i, del\_t2i, del\_yi\_dot, del\_t1i\_dot, del\_t2i\_dot];

sim\_lin\_res = sim('ss\_lin\_sys');

time1 = sim\_lin\_res.lin\_del\_y.time;

dy = sim\_lin\_res.lin\_del\_y.signals.values;

dt1 = sim\_lin\_res.lin\_del\_t1.signals.values;

dt2 = sim\_lin\_res.lin\_del\_t2.signals.values;

% non-linear

sim\_nonlin\_res = sim('db\_pend\_cart\_nonlin.slx');

time2 = sim\_nonlin\_res.tout;

y = sim\_nonlin\_res.y\_nonlin.signals.values;

t1 = sim\_nonlin\_res.theta1\_nonlin.signals.values;

t2 = sim\_nonlin\_res.theta2\_nonlin.signals.values;

% Plotting

fig1 = figure('Renderer',"painters", 'Position', [10 10 900 1000]);

subplot(3,1,1)

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

plot(time1,dy)

plot(time2,y)

ylabel('y [m]')

hold off

subplot(3,1,2)

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

plot(time1,dt1)

plot(time2,t1)

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

hold off

subplot(3,1,3)

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

plot(time1,dt2)

plot(time2,t2)

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

xlabel('time, [sec]')

h = legend('linear','nonlinear'); set(h, 'Position', [0.8, 0.05, .1, .025]);

hold off

sgtitle('L7 Periodic Response for Nonlinearized and Linearized - T. Koike')

saveas(fig1, 'p9\_a.png')

(b).

Similar to part (a),

The matrix for this condition is

and the corresponding eigenvalues and eigenvectors are

|  |  |
| --- | --- |
|  |  |
| 0 | 1  0  0  0  0  0 |
| 0 | -1.0000  0  0  0.0000  0  0 |
| -0.0000 + 1.8113i | -0.0000 - 0.0893i  -0.0000 - 0.1284i  -0.0000 - 0.4573i  0.1617 - 0.0000i  0.2326 - 0.0000i  0.8283 + 0.0000i |
| -0.0000 - 1.8113i | -0.0000 + 0.0893i  -0.0000 + 0.1284i  -0.0000 + 0.4573i  0.1617 + 0.0000i  0.2326 + 0.0000i  0.8283 + 0.0000i |
| 0.0000 + 1.1042i | 0.0000 - 0.1040i  0.0000 - 0.5782i  -0.0000 + 0.3247i  0.1148 + 0.0000i  0.6385 + 0.0000i  -0.3585 + 0.0000i |
| 0.0000 - 1.1042i | 0.0000 + 0.1040i  0.0000 + 0.5782i  -0.0000 - 0.3247i  0.1148 - 0.0000i  0.6385 + 0.0000i  -0.3585 - 0.0000i |

Thus, the response is characterized by