Table of Contents

HW 9 P1

a

```
clear;clc;close all

% A = [-1 1;0 -1];
% eig(A)

% A = [-1 1;-1 -1];
% eig(A)

A = [-1 2001;-1 0];
[V, D] = eig(A)

rankV = rank(V);
sizeA = size(A, 1);  % Number of rows/columns of A
if rankV < sizeA
    disp('The matrix A is defective.');
else
    disp('The matrix A is not defective.');
end
```

```
V =

   0.9998 + 0.0000i   0.9998 + 0.0000i
   0.0002 + 0.0223i   0.0002 - 0.0223i


D =

  -0.5000 +44.7297i   0.0000 + 0.0000i
   0.0000 + 0.0000i  -0.5000 -44.7297i

The matrix A is not defective.
```

b

```
clear;clc;close all

A = [-1 0; 0 1];
[V, D] = eig(A)

rankV = rank(V);
sizeA = size(A, 1);  % Number of rows/columns of A
if rankV < sizeA
    disp('The matrix A is defective.');
else
    disp('The matrix A is not defective.');
end
```

```
V =

     1     0
     0     1


D =

    -1     0
     0     1

The matrix A is not defective.
```

c

```
clear;clc;close all

A = [1i 1; 0 1i];
[V, D] = eig(A)

rankV = rank(V);
sizeA = size(A, 1);  % Number of rows/columns of A
if rankV < sizeA
    disp('The matrix A is defective.');
else
    disp('The matrix A is not defective.');
end
```

```
V =

    1.0000    1.0000
         0   -0.0000


D =
```

```
   0.0000 + 1.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 1.0000i

The matrix A is defective.
```

d

```
clear;clc;close all

A = [-2 0;0 0.5];
[V, D] = eig(A)

rankV = rank(V);
sizeA = size(A, 1);  % Number of rows/columns of A
if rankV < sizeA
    disp('The matrix A is defective.');
else
    disp('The matrix A is not defective.');
end


V =

     1     0
     0     1


D =

   -2.0000         0
         0    0.5000

The matrix A is not defective.
```

P2

```
clear;clc;close all

A = [0.5 1 -0.5 0; -1 0.5 0 -0.5; 0.5 0 -0.5 1;0 0.5 -1 -0.5];
[V, D] = eig(A)

rankV = rank(V);
sizeA = size(A, 1);  % Number of rows/columns of A
if rankV < sizeA
    disp('The matrix A is defective.');
else
    disp('The matrix A is not defective.');
end


V =

   0.5000 + 0.0000i   0.5000 + 0.0000i  -0.5000 - 0.0000i  -0.5000 + 0.0000i
```

```
  -0.0000 + 0.5000i  -0.0000 - 0.5000i   0.0000 - 0.5000i   0.0000 + 0.5000i
   0.5000 + 0.0000i   0.5000 - 0.0000i  -0.5000 + 0.0000i  -0.5000 + 0.0000i
  -0.0000 + 0.5000i  -0.0000 - 0.5000i   0.0000 - 0.5000i   0.0000 + 0.5000i


D =

   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  -0.0000 + 1.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 + 0.0000i   0.0000 + 0.0000i  -0.0000 - 1.0000i

The matrix A is not defective.
```

P3.a.

```
clear;clc;close all

% A1 = [0 1;-1.732 2];
A1 = [0 1;-pi/3 2];
[V, D] = eig(A1)

rankV = rank(V);
sizeA1 = size(A1, 1);  % Number of rows/columns of A
if rankV < sizeA1
    disp('The matrix A1 is defective.');
else
    disp('The matrix A1 is not defective.');
end

A2 = [0 1;-1.732 -2];
[V, D] = eig(A2)

rankV = rank(V);
sizeA2 = size(A2, 1);  % Number of rows/columns of A
if rankV < sizeA2
    disp('The matrix A2 is defective.');
else
    disp('The matrix A2 is not defective.');
end


V =

   0.6830 - 0.1484i   0.6830 + 0.1484i
   0.7152 + 0.0000i   0.7152 + 0.0000i


D =

   1.0000 + 0.2172i   0.0000 + 0.0000i
   0.0000 + 0.0000i   1.0000 - 0.2172i
```

```
The matrix A1 is not defective.

V =

  -0.4597 - 0.3933i  -0.4597 + 0.3933i
   0.7962 + 0.0000i   0.7962 + 0.0000i


D =

  -1.0000 + 0.8556i   0.0000 + 0.0000i
   0.0000 + 0.0000i  -1.0000 - 0.8556i

The matrix A2 is not defective.
```

P3.b.

```
clear;clc;close all

A = [0 1;-1 0];
[V, D] = eig(A)

rankV = rank(V);
sizeA = size(A, 1);  % Number of rows/columns of A
if rankV < sizeA
    disp('The matrix A is defective.');
else
    disp('The matrix A is not defective.');
end


V =

   0.7071 + 0.0000i   0.7071 + 0.0000i
   0.0000 + 0.7071i   0.0000 - 0.7071i


D =

   0.0000 + 1.0000i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 - 1.0000i

The matrix A is not defective.
```

P3.c.

```
clear;clc;close all

A = [0 1 0 0; 0 0 1 0; 0 0 0 1; 1 0 0 0];
[V, D] = eig(A)

rankV = rank(V);
sizeA = size(A, 1);  % Number of rows/columns of A
```

```
if rankV < sizeA
    disp('The matrix A is defective.');
else
    disp('The matrix A is not defective.');
end
```

```
V =

   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
  -0.5000 + 0.0000i   0.0000 - 0.5000i   0.0000 + 0.5000i   0.5000 + 0.0000i


D =

  -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

The matrix A is not defective.
```

P4

```
clear;clc;close all

A = [0 1; -0.5 0];
[V, D] = eig(A)

rankV = rank(V);
sizeA = size(A, 1);  % Number of rows/columns of A
if rankV < sizeA
    disp('The matrix A is defective.');
else
    disp('The matrix A is not defective.');
end
```

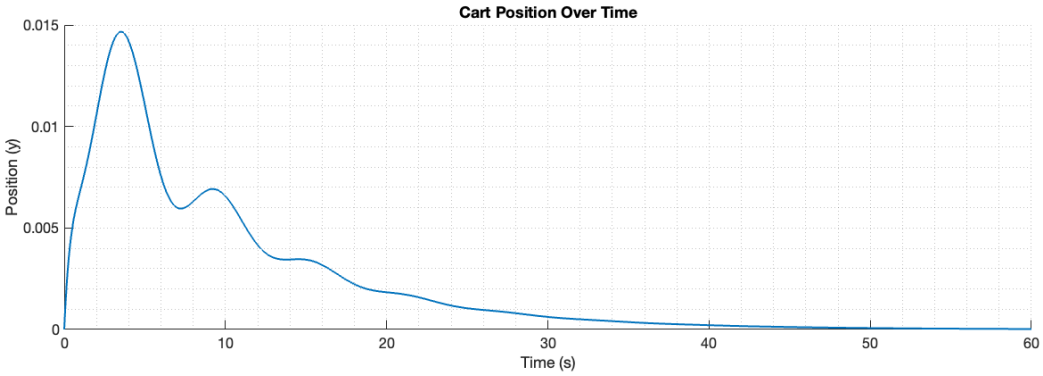
```
V =

   0.8165 + 0.0000i   0.8165 + 0.0000i
   0.0000 + 0.5774i   0.0000 - 0.5774i


D =

   0.0000 + 0.7071i   0.0000 + 0.0000i
   0.0000 + 0.0000i   0.0000 - 0.7071i

The matrix A is not defective.
```

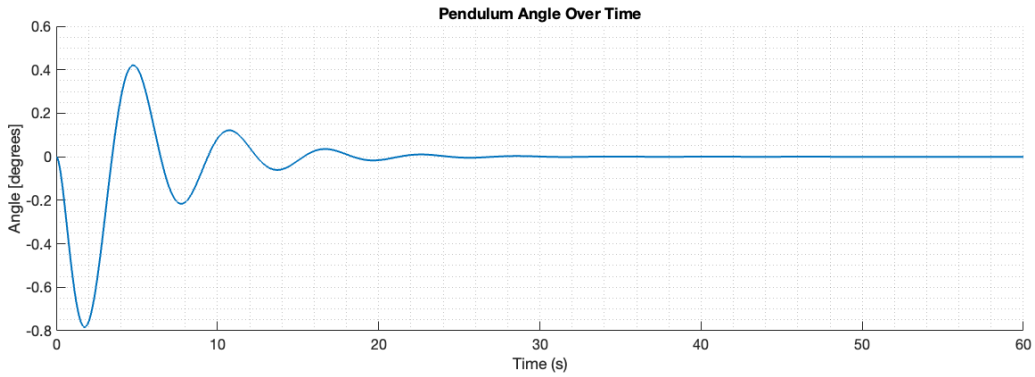
P5

```
clear;clc;close all

c = 5;
k = 0.5;

% Define symbolic variables
syms m0 m1 m2 l1 l2 theta1_e theta2_e g real
% Define the matrix M
M = [m0 + m1 + m2, -m1*l1*cos(theta1_e), -m2*l2*cos(theta2_e);
     -m1*l1*cos(theta1_e), m1*l1^2, 0;
     -m2*l2*cos(theta2_e), 0, m2*l2^2];

% Invert the matrix
K = [-k 0 0;
      0 -m1*l1*g*cos(theta1_e) 0;
      0 0 -m2*l2*g*cos(theta2_e)];

C = [-c 0 0; 0 0 0; 0 0 0];

Phi = [1; 0; 0];

A_top = [zeros(3), eye(3)];
A_bottom = [M\K, M\C];
A = [A_top; A_bottom];

B = [0; 0; 0; M\Phi];

A = double( subs(A, {m0, m1, m2, l1, l2, g, theta1_e, theta2_e}, {2, 1, 1, 1,
1, 1, 0, 0}))
B = double( subs(B, {m0, m1, m2, l1, l2, g, theta1_e, theta2_e}, {2, 1, 1, 1,
1, 1, 0, 0}));
disp('L1')
disp(eig(A))
% A = double( subs(A, {m0, m1, m2, l1, l2, g, theta1_e, theta2_e}, {2, 1, 1,
1, 1, 1, pi, pi}))
% B = double( subs(B, {m0, m1, m2, l1, l2, g, theta1_e, theta2_e}, {2, 1, 1,
1, 1, 1, pi, pi}));
% disp('L2')
% disp(eig(A))


% Simulate Linearized system

% Initial conditions
y0 = 0;               % Initial position of the cart
y_dot0 = 0;           % Initial velocity of the cart
theta0 = 1 * pi/180;      % Initial angle of the pendulum (45 degrees)
theta_dot0 = 0;       % Initial angular velocity of the pendulum
initial_conditions = [y0; y_dot0; 0; theta0; theta_dot0; 0];

% Time span
```

```
t_span = [0 60]; % Simulate for 10 seconds

options = odeset('RelTol', 1e-10,'AbsTol',1e-15);
[t, X] = ode45(@(t, x, A, B) linearizedTwoPendulumTwoCart(t, x, A, B),
t_span, initial_conditions,options, A, B);

a = figure();
subplot(2,1,1);
hold on
grid minor
plot(t, X(:,1),'LineWidth',2); % Cart position
title('Cart Position Over Time');
xlabel('Time (s)');
ylabel('Position (y)');
ax = gca;
ax.FontSize = 16;
subplot(2,1,2);
hold on
grid minor
plot(t, X(:,3)*180/pi,'LineWidth',2); % Pendulum angle
title('Pendulum Angle Over Time');
xlabel('Time (s)');
ylabel('Angle [degrees]');
a.Position = [100 100 1400 1000];
ax = gca;
ax.FontSize = 16;
```

```
A =

         0         0         0    1.0000         0         0
         0         0         0         0    1.0000         0
         0         0         0         0         0    1.0000
   -0.2500   -0.5000   -0.5000   -2.5000         0         0
   -0.2500   -1.5000   -0.5000   -2.5000         0         0
   -0.2500   -0.5000   -1.5000   -2.5000         0         0

L1
  -1.9694 + 0.0000i
  -0.1095 + 0.0000i
  -0.2105 + 1.0557i
  -0.2105 - 1.0557i
  -0.0000 + 1.0000i
  -0.0000 - 1.0000i
```





Published with MATLAB® R2024b