Nonlinear System Midterm Exam

電機碩一 陳昕佑 61375017H

I designed a third-order nonlinear system inspired by “single-joint manipulator + integral action” ideas, and its construction guarantees local asymptotic stability about the origin. It features:

- A linear part chosen to be asymptotically stable at the origin.
- Six distinct nonlinear terms, all of which vanish at $(0, 0, 0)$.
- A total of three states (x_1, x_2, x_3), making it third-order.

Porposed System

Define the states:

- x_1 :as the joint angle error $q - q_{\text{desired}}$.
- x_2 :the joint angular velocity (or rate of change of x_1).
- x_3 :an integral-of-error state, often introduced in control to remove steady-state offset.

The dynamics are:

$$\dot{x_1} = x_2,$$

$$\dot{x_2} = -2x_1 - 3x_2 - 4x_3 + x_1^2 x_2 + x_1^3 + \sin(x_1) + x_2^2 x_3 + x_1^2 \sin(x_3) + x_1 x_2 x_3,$$

$$\dot{x_3} = x_1,$$

counting the six nonlinear terms, all appear in the $\dot{x_2}$ equation, and each of these is zero at $(x_1, x_2, x_3) = (0, 0, 0)$.

(1) Find the equilibrium points of the system.

```
function dx = my_system(x)
    % x = [x1; x2; x3]
    x1 = x(1);
    x2 = x(2);
    x3 = x(3);

    dx1 = x2;
    dx2 = -2*x1 - 3*x2 - 4*x3 ...
          + x1^2 * x2 ...
          + x1^3 ...
          + sin(x1) ...
          + x2^2 * x3 ...
          + x1^2 * sin(x3) ...
          + x1 * x2 * x3;
    dx3 = x1;

    dx = [dx1; dx2; dx3];
end
```

```
% Initial guess near origin
x0 = [0; 0; 0];

% Solve for equilibrium point (dx = 0)
equilibrium = fsolve(@my_system, x0);
```

```
Equation solved at initial point.

fsolve completed because the vector of function values at the initial point
is near zero as measured by the value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>
```

```
disp('Equilibrium point:');
```

```
Equilibrium point:
```

```
disp(equilibrium);
```

```
     0
     0
     0
```

```
x0_list = [0 0 0; 1 1 1; -1 -1 -1; 0.5 -0.5 0.5];
for i = 1:size(x0_list, 1)
    x0 = x0_list(i, :)';
    eq = fsolve(@my_system, x0);
    disp(['Guess ' num2str(i) ': ' mat2str(eq)]);
end
```

```
Equation solved at initial point.

fsolve completed because the vector of function values at the initial point
is near zero as measured by the value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>
Guess 1: [0;0;0]
Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>
Guess 2: [0;0;0]
Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>
Guess 3: [0;0;0]
Equation solved.

fsolve completed because the vector of function values is near zero
```

```
as measured by the value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>
Guess 4: [0;9.25185853854298e-18;-6.93889390390723e-18]
```

(2) Linearize the system around the equilibrium points.

```
syms x1 x2 x3

dx1 = x2;
dx2 = -2*x1 - 3*x2 - 4*x3 ...
      + x1^2 * x2 ...
      + x1^3 ...
```

```
      + sin(x1) ...
      + x2^2 * x3 ...
      + x1^2 * sin(x3) ...
      + x1 * x2 * x3;
dx3 = x1;

f = [dx1; dx2; dx3];
x = [x1; x2; x3];

J = jacobian(f, x);

J_eq = double(subs(J, x, [0; 0; 0]));
disp('Jacobian at the origin:');
```

```
Jacobian at the origin:
```

```
disp(J_eq);
```

```
     0     1     0
    -1    -3    -4
     1     0     0
```

```
eig(J)  % Get eigenvalues
```

```
ans =
```

$$\begin{pmatrix} \frac{x_1\,x_3}{3}+\sigma_1+\frac{\sigma_4}{\sigma_3}+\sigma_3+\frac{{x_1}^2}{3}-1 \\ \frac{x_1\,x_3}{3}+\sigma_1-\frac{\sigma_4}{2\,\sigma_3}-\frac{\sigma_3}{2}+\frac{{x_1}^2}{3}-1-\sigma_2 \\ \frac{x_1\,x_3}{3}+\sigma_1-\frac{\sigma_4}{2\,\sigma_3}-\frac{\sigma_3}{2}+\frac{{x_1}^2}{3}-1+\sigma_2 \end{pmatrix}$$

where

$$\sigma_1=\frac{2\,x_2\,x_3}{3}$$

$$\sigma_2=\frac{\sqrt{3}\,\left(\frac{\sigma_4}{\sigma_3}-\sigma_3\right)\,\mathrm{i}}{2}$$

$$\sigma_3=\left(\sigma_5+\frac{{x_1}^2\cos(x_3)}{2}+\frac{{\sigma_6}^3}{27}+\frac{x_1\,x_2}{2}+\frac{{x_2}^2}{2}+\sqrt{\left(\sigma_5+\frac{{x_1}^2\cos(x_3)}{2}+\frac{{\sigma_6}^3}{27}+\frac{x_1\,x_2}{2}+\frac{{x_2}^2}{2}-2\right)^2-{\sigma_4}^3}-2\right)^{1/3}$$

$$\sigma_4=\frac{\cos(x_1)}{3}+\frac{{\sigma_6}^2}{9}+\frac{2\,x_1\,x_2}{3}+\frac{x_2\,x_3}{3}+\frac{2\,x_1\sin(x_3)}{3}+{x_1}^2-\frac{2}{3}$$

$$\sigma_5=\frac{\sigma_6\,\left(\cos(x_1)+2\,x_1\,x_2+x_2\,x_3+2\,x_1\sin(x_3)+3\,{x_1}^2-2\right)}{6}$$

$$\sigma_6={x_1}^2+x_3\,x_1+2\,x_2\,x_3-3$$