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

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

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

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

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

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

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

(3) Analyze the stability of the equilibrium points using the linearized system

The nonlinear system: $\dot{x} = f(x)$

The Jacobian matrix: $A = \frac{\partial}{\partial x} f|_{x=x_{\mathrm{eq}}}$, and to analyze $\dot{x} = \mathrm{Ax}$.

J_eq is the Jacobian matrix evaluated at the equilibrium point $([0;0;0])$.

- If all eigenvalues have strictly negative real parts → the equilibrium is locally asymptotically stable.
- If any eigenvalue has a positive real part → the equilibrium is unstable.
- If any eigenvalue is purely imaginary → linearlization is inconclusive.

Since all eigenvalues have negative real parts, the system is locally asymptotically stable at the equilibrium point $[0;0;0]$.

```
eig(J_eq)
```

```
ans = 3×1 complex
   0.0473 + 1.1359i
   0.0473 - 1.1359i
  -3.0946 + 0.0000i
```

```
A = [0, 1, 0;
     -2, -3, -4;
      1, 0, 0];

eigenvalues = eig(A)
```

```
eigenvalues = 3×1 complex
  -0.1018 + 1.1917i
  -0.1018 - 1.1917i
  -2.7963 + 0.0000i
```

(4) Find a Lyapunov function.

$$\dot{x} = \mathrm{Ax} = \begin{bmatrix} 0 & 1 & 0 \\ -2 & -3 & -4 \\ 1 & 0 & 0 \end{bmatrix} x$$

(a) Krasovskii's method

$V(x) = \dot{x}^{\top}\dot{x} = (\mathrm{Ax})^{\top}(\mathrm{Ax}) = x^{\top}A^{\top}\mathrm{Ax}$, define $P = A^{\top}A \Longrightarrow V(x) = x^{\top}\mathrm{Px}$.

Try to solve P using the Lyapunov equation: $A^{\top}P + \mathrm{PA} = -Q$, choose $Q = I$.

This guarantees $P > 0, \dot{V} = -x^{\top}\mathrm{Qx} < 0$.

```
A = [0 1 0;
     -2 -3 -4;
      1 0 0];

Q = eye(3);
P = lyap(A', Q);

% Check positive definiteness
eigs_P = eig(P);
disp('Eigenvalues of P (should be > 0):');
```

```
Eigenvalues of P (should be > 0):
```

```
disp(eigs_P);
```

```
    0.1741
    5.4904
   13.4605
```

```
% Confirm V_dot = x'*(A'*P + P*A)*x < 0
```

```
eig_AP_PA = eig(A'*P + P*A);
disp('Eigenvalues of A''P + PA (should be < 0):');
```

```
Eigenvalues of A'P + PA (should be < 0):
```

```
disp(eig_AP_PA);  % Should be negative definite
```

```
   -1.0000
   -1.0000
   -1.0000
```

(b) The Variable Gradient Method

This method starts by guessing a gradient form for the Lyapunov function: $\nabla V(x) = g(x)$, then integratees to get $V(x)$, and

ensures $\dot{V}(x) < 0$.

I choose $\nabla V(x) = \begin{bmatrix} 2x_1 \\ 2x_2 \\ 2x_3 \end{bmatrix} \Rightarrow V(x) = {x_1}^2 + {x_2}^2 + {x_3}^2$, then $\dot{V}(x) = \nabla V^\top \dot{x} = 2x_1\dot{x_1} + 2x_2\dot{x_2} + 2x_3\dot{x_3}$.

Using the linearized system:

$\dot{x_1} = x_2$

$\dot{x_2} = -2x_1 - 3x_2 - 4x_3$

$\dot{x_3} = x_1$

$$
\begin{aligned}
\dot{V}(x) &= 2x_1x_2 + 2x_2(-2x_1 - 3x_2 - 4x_3) + 2x_3x_1 \\
&= 2x_1x_2 - 4x_1x_2 - 6{x_2}^2 - 8x_2x_3 + 2x_1x_3 \\
&= (-2x_1x_2) - 6{x_2}^2 - 8x_2x_3 + 2x_1x_3 \\
&< 0
\end{aligned}
$$

(5) Plot a Lyapunov function using MATLAB

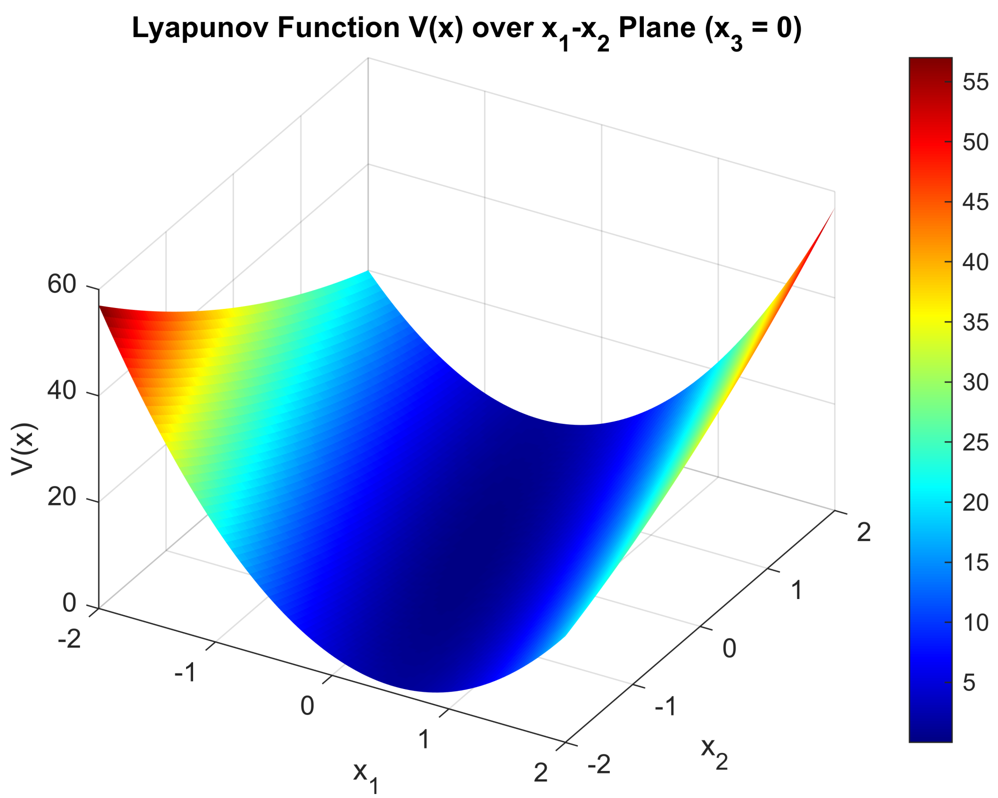
The Lyapunov function: $V(x) = x^\top \mathrm{P}x$, over a 2D slice ($x_3 = 0$) for visualization. Using the Lyapunov equation solution to get a positive

define P, then visualize $V(x)$as a surface and contour plot.

- A bowl-shaped surface for $V(x)$ if $P > 0$.
- The contours are elliptic, showing level sets of constant energy.
- This visually confirms $V(x) > 0$ and shows how it increases as you move away from the origin.

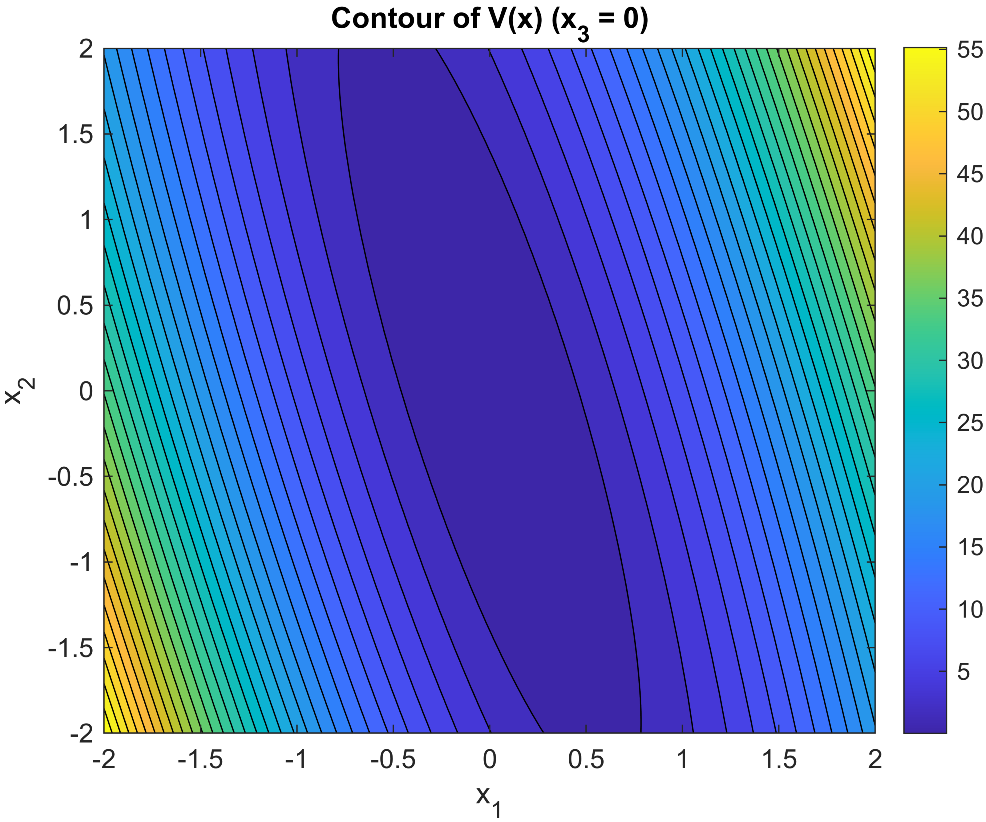
```
% System matrix
A = [0 1 0;
    -2 -3 -4;
```

```
     1 0 0];

% Solve Lyapunov equation: A'P + P*A = -Q
Q = eye(3);
P = lyap(A', Q);

% Generate grid over x1 and x2 (fix x3 = 0)
[x1, x2] = meshgrid(linspace(-2, 2, 100), linspace(-2, 2, 100));
x3 = 0;

V = zeros(size(x1));
for i = 1:numel(x1)
    x_vec = [x1(i); x2(i); x3];
    V(i) = x_vec' * P * x_vec;
end

% Plot Lyapunov function
figure;
surf(x1, x2, V, 'EdgeColor', 'none');
xlabel('x_1'); ylabel('x_2'); zlabel('V(x)');
title('Lyapunov Function V(x) over x_1-x_2 Plane (x_3 = 0)');
colormap jet; colorbar;
view(30, 40); % 3D view angle
```




```
% Optional: Contour plot
figure;
```

```
contourf(x1, x2, V, 30);
xlabel('x_1'); ylabel('x_2');
title('Contour of V(x) (x_3 = 0)');
colorbar;
```

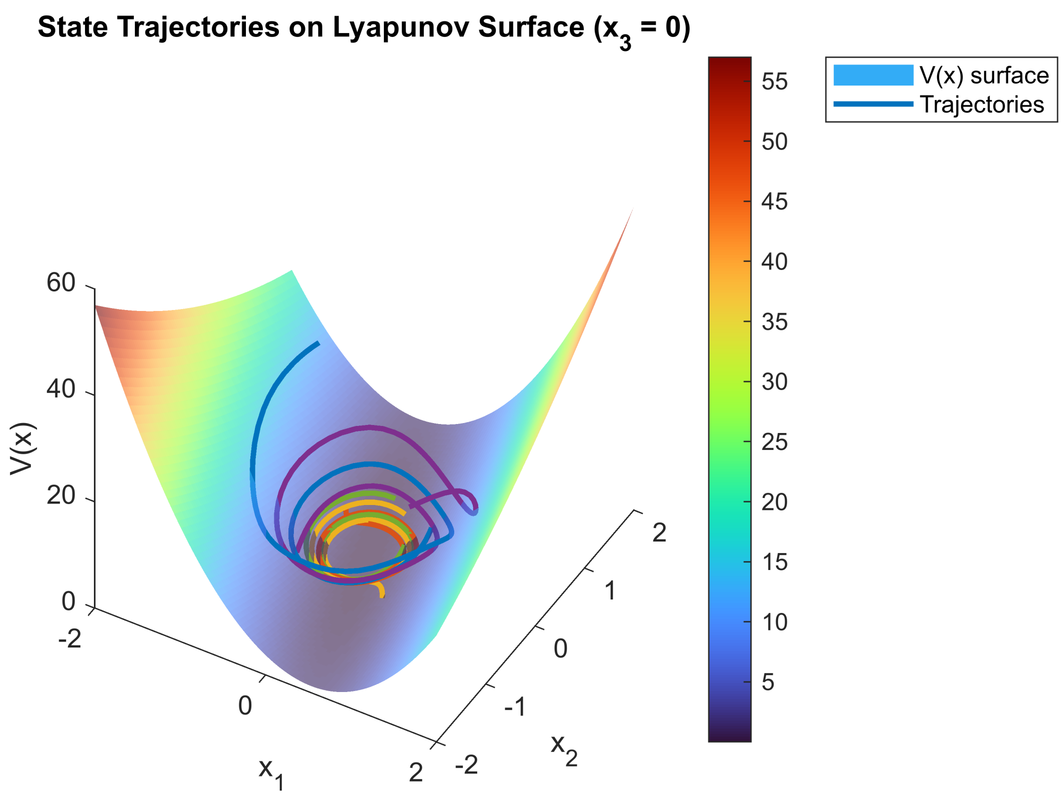


(6) Plot state trajectories of the system with Lyapunov function near the origin using MATLAB.

1. Define the nonlinear system as $\dot{x} = f(x)$.
2. Simulate trajectories from nearby initial conditions using $\mathrm{ode45}$.
3. Plot the Lyapunov function surface $(V(x_1, x_2) \text{ at } x_3 = 0)$.
4. Overlay the trajectories onto the surface or contour plot.

```
% System definition: nonlinear system
function dx = nonlinear_sys(t, x)
    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

% Solve Lyapunov equation for V(x) = x' P x
A = [0 1 0;
```

```
    -2 -3 -4;
     1 0 0];
Q = eye(3);
P = lyap(A', Q);

% Grid for V(x1, x2), fix x3 = 0
[x1g, x2g] = meshgrid(linspace(-2,2,100), linspace(-2,2,100));
x3 = 0;
V = zeros(size(x1g));
for i = 1:numel(x1g)
    x = [x1g(i); x2g(i); x3];
    V(i) = x' * P * x;
end

% Simulate trajectories from different initial points near origin
init_conditions = [
     0.5  0.5  0;
    -0.5  0.5  0;
     0.5 -0.5  0;
    -0.5 -0.5  0;
     0.3  0.3  0
];

tspan = [0 10];

figure;
hold on;
surf(x1g, x2g, V, 'EdgeColor', 'none', 'FaceAlpha', 0.6);
xlabel('x_1'); ylabel('x_2'); zlabel('V(x)');
title('State Trajectories on Lyapunov Surface (x_3 = 0)');
colormap turbo;
colorbar;
view(30,40);

for i = 1:size(init_conditions,1)
    [t, x] = ode45(@nonlinear_sys, tspan, init_conditions(i,:)');

    % Project to x1-x2 plane at x3 = 0
    V_traj = zeros(length(t),1);
    for k = 1:length(t)
        xk = x(k,:)';
        V_traj(k) = xk' * P * xk;
    end

    plot3(x(:,1), x(:,2), V_traj, 'LineWidth', 2);
end

legend('V(x) surface','Trajectories');
```



(7) Show the origin is stable using Lyapunov's direct method.

My system:

$\dot{x_1} = x_2,$

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

$\dot{x_3} = x_1,$

denote it as: $\dot{x} = f(x)$ with $x = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^{\top}$.

Using a positive definite function: $V(x) = x^{\top}\mathrm{Px}$ with $P > 0$, and previously solved $A^{\top}P + \mathrm{PA} = -Q,\ Q = I$, where A is the linearlization

around origin, so $V(x) = x^{\top}\mathrm{Px}$ is positive definite.

To check $\dot{V}(x) = \dfrac{\mathrm{d}}{\mathrm{d}t}(x^{\top}\mathrm{Px}) = \dot{x}^{\top}\mathrm{Px} + x^{\top}P\dot{x} = x^{\top}(\mathrm{Pf}(x) + f(x)^{\top}P)$ is negative definiteness:

$f(x) \approx \mathrm{Ax} \Rightarrow \dot{V}(x) = x^{\top}(\mathrm{A}^{\top}P + \mathrm{PA})x = -x^{\top}\mathrm{Qx}$ (negative definite).

Since $V(x)$ is positive definite, $\dot{V}(x)$ is negative definite near the origin, the origin is locally asymptotically stable.

(8) Show the origin is asymptotically stable using Invariant set theorem.

With the nonlinear srstem and the Lyapunov candidate, $\dot{V}(x) = \dot{x}^{\top}\mathrm{Px} + x^{\top}P\dot{x} = 2x^{\top}P\dot{x}$.

Near the origin, we can write $\dot{V}(x) = -x^{\top}\mathrm{Qx} +$ higher-order nonlinear terms. So, $\dot{V}(x) \leq 0$ and the equality $\dot{V}(x) = 0$ only occurs when

$x = 0$, because $x^{\top}\mathrm{Qx} > 0$ for all $x \neq 0$. Thus, $\varepsilon = \{x | x^{\top}\mathrm{Qx} = 0\} = \{0\} \Rightarrow$ only invariant solution in ε is $x(t) = 0$.

Since $V(x)$ is positive definite, $\dot{V}(x) \leq 0$, the largest invariant set in $\left\{x | \dot{V}(x) = 0\right\}$ is the origin, by LaSalle's Invariance Principle, all

trajectories near the origin approach the origin as $t \to \infty$. So the origin is locally asymptotically stable.

(9) Performance analysis of the system.

We've already shown local asymptotic stability via Linearization & eigenvalues, Lyapunov direct method, Invariant set theorem, so the

system returns to equilibrium for small perturbations.