EE 5323 Nonlinear Control Systems

Homework Pledge of Honor

On all homeworks in this class - YOU MUST WORK ALONE.

Any cheating or collusion will be severely punished.

It is very easy to compare your software code and determine if you worked together It does not matter if you change the variable names.

Please sign this form and include it as the first page of all of your submitted homeworks.

Typed Name: VRUSHABH SURESH DONGE

Pledge of honor:

"On my honor I have neither given nor received aid on this homework."

Signature: V.S. Donge.

EE 5323 Homework 1

Vrushabh S Donge (UTA ID: 1001914437)

Fall 2021 (Nonlinear Control Systems)

1 Vanderpol Oscillator

The Vanderpol Oscillator with following dynamics simulated for 100 seconds.

$$\ddot{y} + \alpha(y^2 - 1)\dot{y} + y = 0 \tag{1.1}$$

The second order differential equation in 1.1 can be written in state space variables with states $x_1 = y$ and $x_2 = \dot{y}$.

$$\dot{x}_1 = x_2 \tag{1.2}$$

$$\dot{x}_2 = -\alpha(x_1^2 - 1)x_2 - x_1 \tag{1.3}$$

1.1 MATLAB Code

```
function xdot = vanderpol( t, x)
 %alpha = 0.03 ;
 alpha = 0.95 ;
 xdot(1, 1) = x(2);
 xdot(2, 1) = - alpha * ( x(1)^2 - 1 ) * x(2)  - x(1);
end
```

Listing 1: Matlab function file for Vanderpol Oscillator

```
close all;
clc;
clear all;


tint = [0 100];
x0 = [0.1 0.2]';
[t, x] = ode23(@vanderpol,tint,x0);

plot(t, x(:,1));
legend('y(t)');
xlabel('t');
ylabel('y(t)');

figure
plot(x(:,1), x(:,2));
legend('y vs ydot');
xlabel('y(t)');
ylabel('dy(t)');
```

Listing 2: Matlab main file for Vanderpol Oscillator

1.2 Simulation results

1.2.1 Case 1:

For initial condition $x(0) = [0.1 \quad 0.2]^T$ and $\alpha = 0.03$

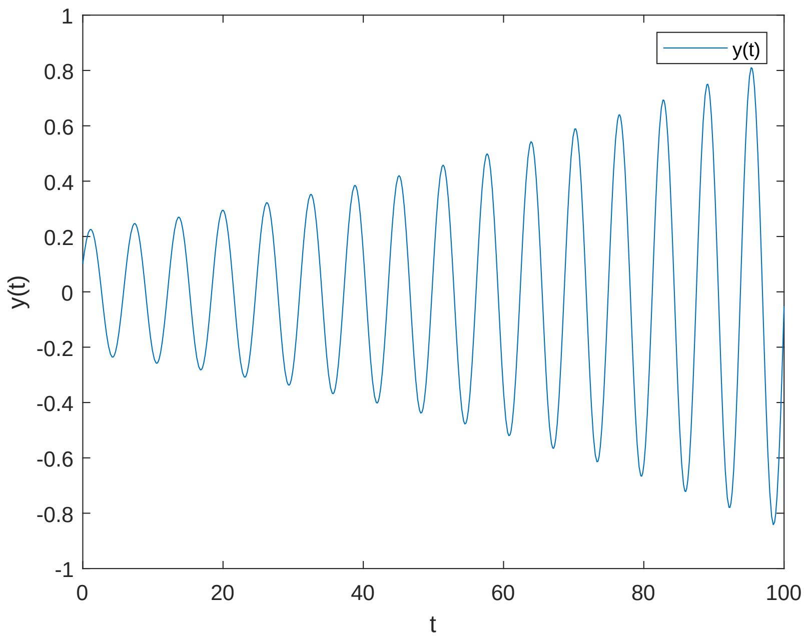

Figure 1: Case 1- $y(t)$ vs t

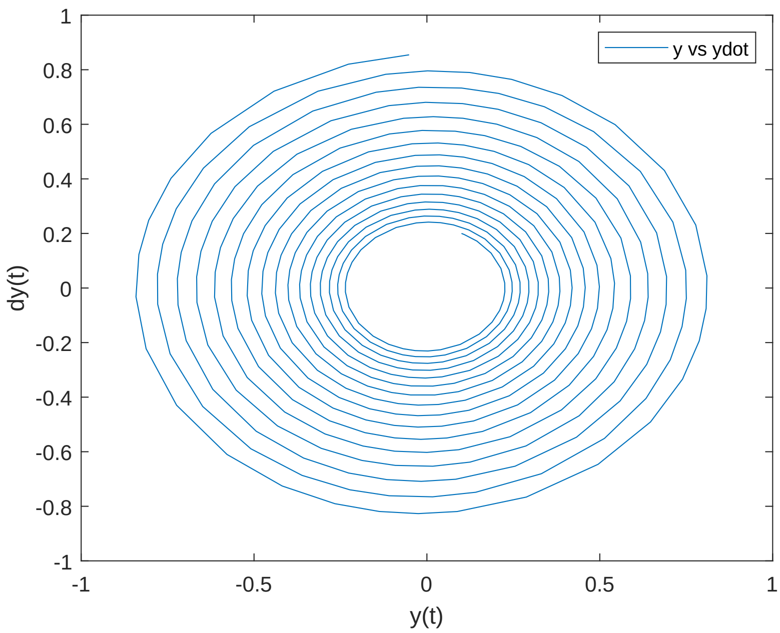

Figure 2: Case 1 Phase plane plot- $dy(t)$ vs $y(t)$

1.2.2 Case 2:

For initial condition $x(0) = [0.1 \quad 0.2]^T$ and $\alpha = 0.95$

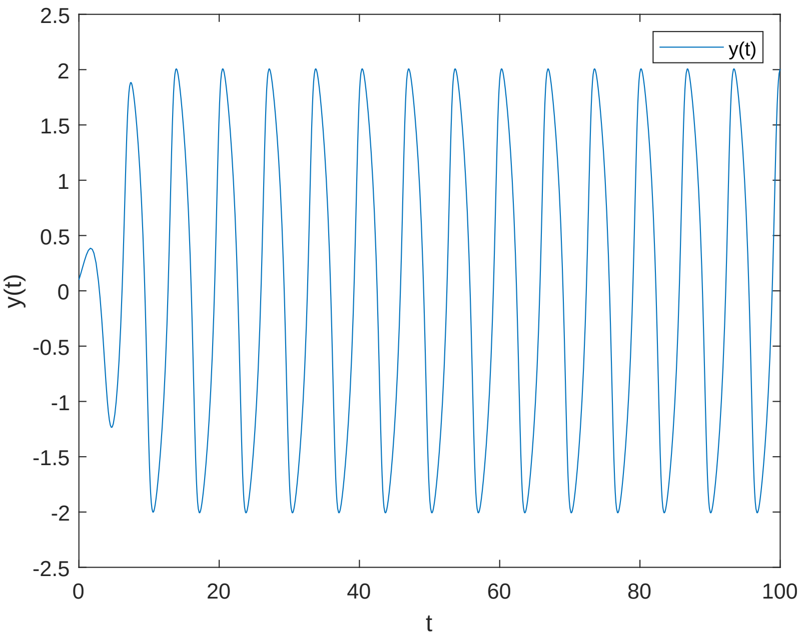

Figure 3: Case 2- $y(t)$ vs t

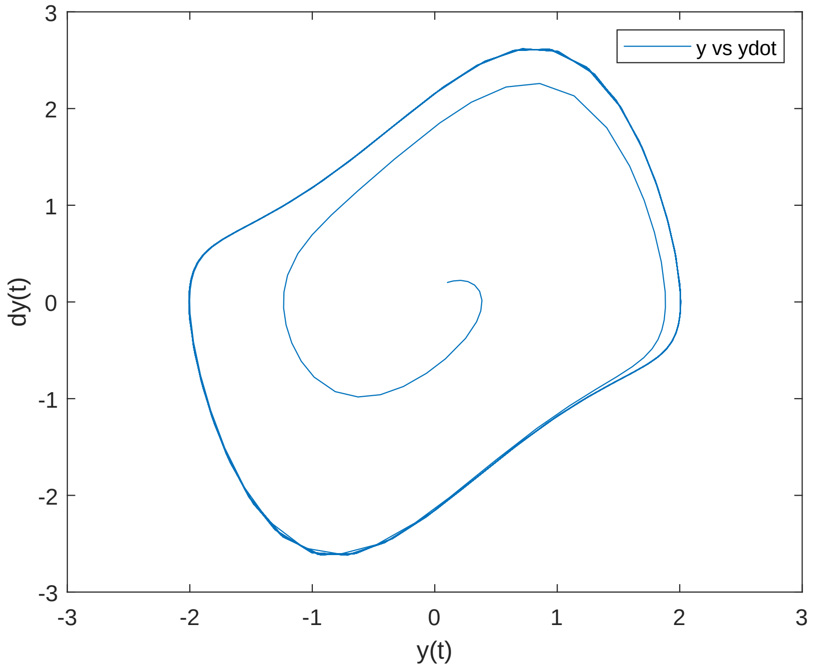

Figure 4: Case 2 Phase plane plot- $y(t)'$ vs $y(t)$

2 Lorenz Curve

The Lorenz attractor chaotic system simulated for 150 seconds.

$$\dot{x}_1 = \sigma(x_1 - x_2) \tag{2.1}$$

$$\dot{x}_2 = rx_1 - x_2 - x_1x_3 \tag{2.2}$$

$$\dot{x}_3 = -bx_3 + x_1x_2 \tag{2.3}$$

2.1 MATLAB Code

```
clc;
clear all;
close all;

a=10; %sigma
b=8/3;%beta
c=28; %rho

f=@(t,x)[-a*x(1)+a*x(2);...
    c*x(1)-x(2)-x(1)*x(3);...
    -b*x(3)+x(1)*x(2)]; %x(1)=x;x(2)=y;x(3)=z

% [1 150]is time interval
% [0.5 0.5 0.5]is initial condition
[t,x]=ode23(f,[0 150],[0.5 0.5 0.5]);%solving ode

plot(t,x)
xlabel('t');ylabel('x');
legend('x1','x2','x3')

figure
plot3(x(:,1),x(:,2),x(:,3));grid on;
xlabel('x');ylabel('y');zlabel('z');title('(a)lorenz curve')
```

Listing 3: Matlab code for Lorenz attractor

2.2 Simulation results

Case: For initial condition, $x(0) = [0.5 \quad 0.5 \quad 0.5]^T$ and Use $\sigma = 10$, $r = 28$ and $b = \frac{8}{3}$

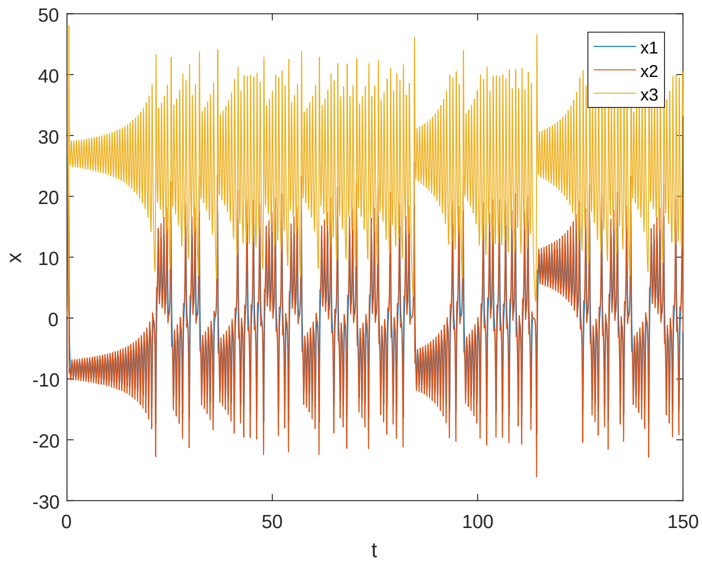

Figure 5: Evolution of states of Lorenz attractor with respect to time

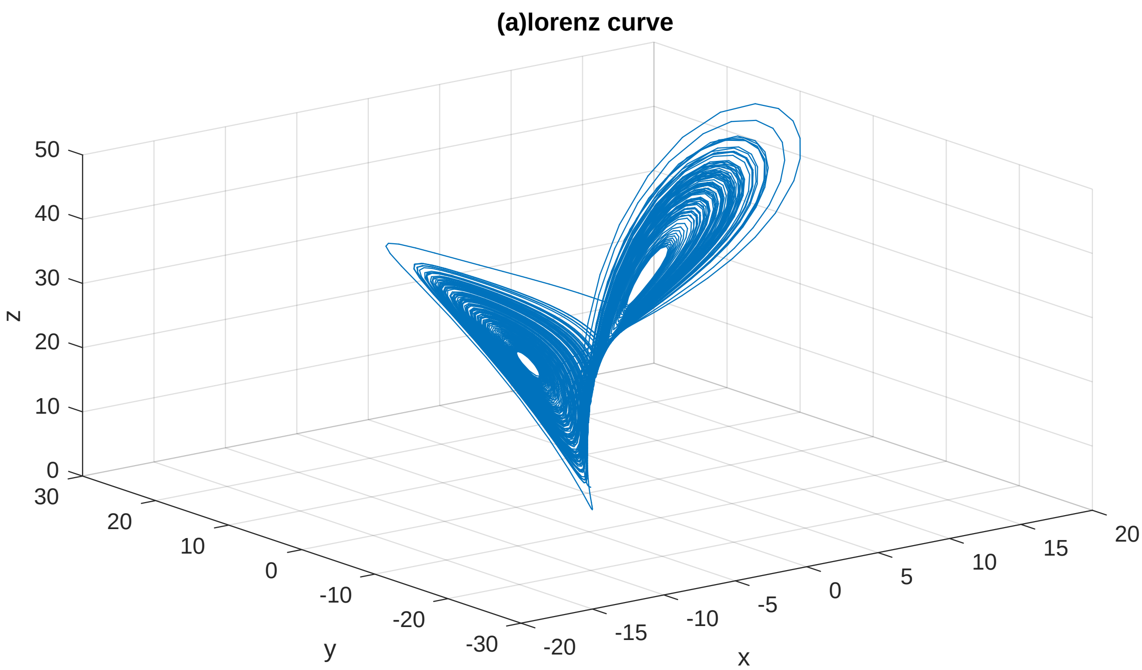

Figure 6: Lorenz curve

3 Voltera predator-prey system

The Voltera predator-prey system simulated for — seconds.

$$\dot{x}_1 = -x_1 + x_1 x_2 \quad [3.1]$$

$$\dot{x}_2 = x_2 - x_1 x_2 \quad [3.2]$$

3.1 MATLAB Code

```
function xdot=predatorprey(t, x)

 xdot(1, 1) = -x(1)+x(1)*x(2);
 xdot(2, 1) = x(2)-x(1)*x(2);
end
```

Listing 4: Matlab function file for Voltera predator-prey system

```
close all;
clc;
clear all;

tspan=[0 10];

for i=-2:0.25:2
    for j=-2:0.25:2

        x0=[i;j];
        [t,x]=ode23('predatorprey',tspan,x0);

        plot(x(:,1),x(:,2))
        grid on;hold on;
    end
end

xlabel('x1');
ylabel('x2');
axis([-5 5 -5 5])
```

Listing 5: Matlab main file for Voltera predator-prey system

3.2 Simulation results

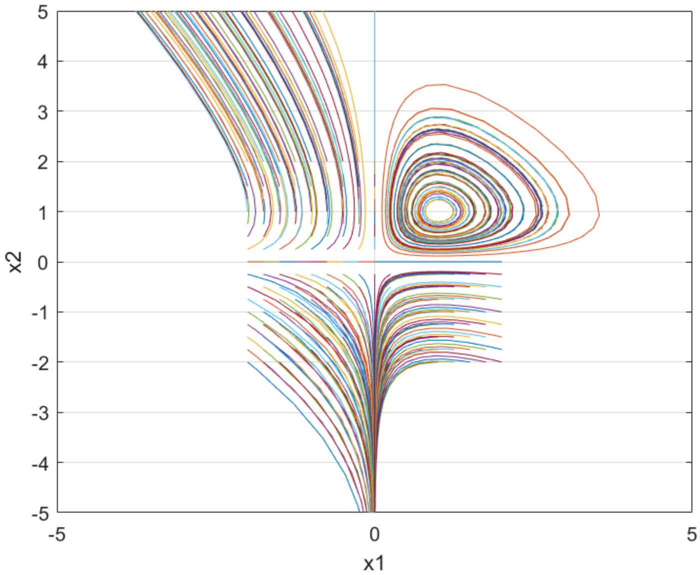

Figure 7: Phase plane plot