HW 02 - Nonlinear Systems Simulation

Table of Contents

Document Information:

- Author: Bardia Mojra
- Date: 09/14/2021
- Title: HW 02 - Nonlinear Systems Simulation
- Term: Fall 2021
- Class: EE 5323 - Nonlinear Systems
- Dr. Lewis

HW 01 - Nonlinear Systems Simulation:

1. Duffing's Equation
2. Lozenz Attractor Chaotic System
3. Voltera Predator-Prey System

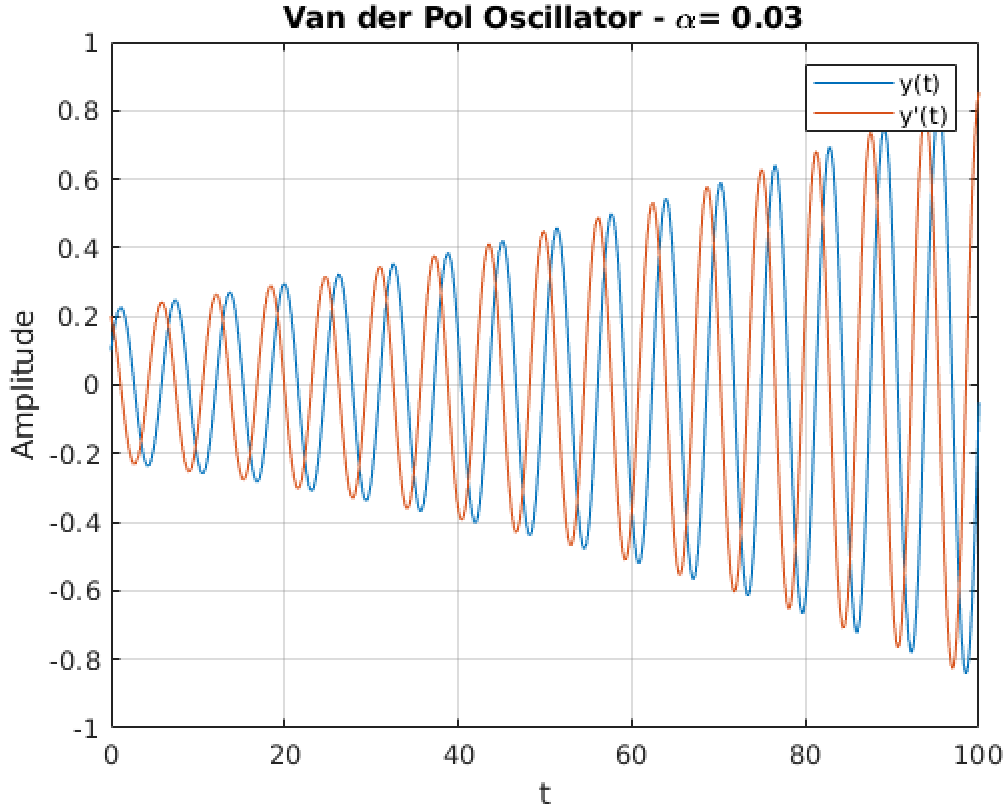
Van der Pol Oscillator:

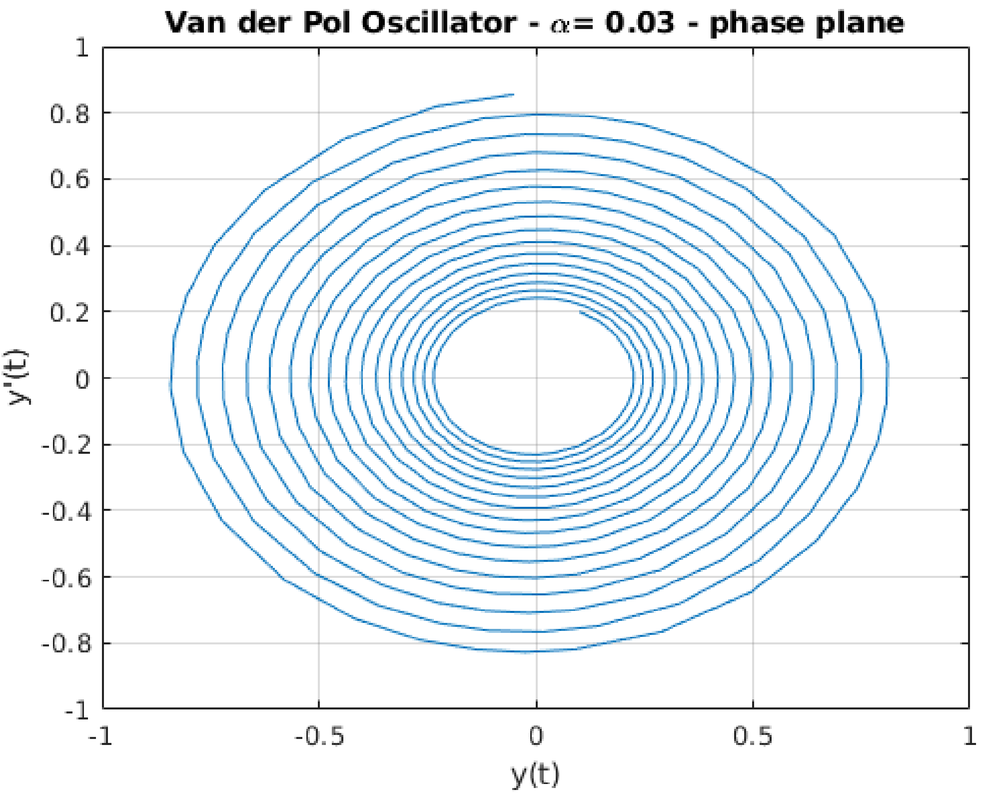
Duffing’s equation is interesting in that it exhibits bifurcation, or dependence of stability properties and number of equilibrium points on a parameter. The undamped Duffing equation is * $$ \dd{x} + \alpha x + x^3 = 0 $$. a.0)=0.2 $$ as initial conditions. # ITEM1 # ITEM2 * Plot $$ y(t) vs. t . $*Plotthephaseplaneplot$ y'(t) $vs.$ y(t) $$.

```
Error updating Text.

 String scalar or character vector must have valid interpreter syntax:
$$ as initial conditions. # ITEM1 # ITEM2 * Plot $$

Error updating Text.
```

```
String scalar or character vector must have valid interpreter syntax:
$$ \dd{x} + \alpha x + x^3 = 0 $$

clc
close all


disp('P01-A Van der Pol - \alpha=0.3')
disp('Set initial conditions for y(0) and ydot(0):')
disp('Plot y(t) vs t for \alpha=0.03')
x0 = [0.1, 0.2]';
t_interval= [0 100];

figure
[t,x]= ode23('VanDerPolA', t_interval, x0);
plot(t,x)
ylabel('Amplitude');
xlabel('t');
grid on;
title('Van der Pol Oscillator - \alpha= 0.03');
legend('y(t)', "y'(t)");

disp("Plot y(t) vs y'(t) for \alpha=0.03")
figure
plot(x(:,1),x(:,2))
xlabel('y(t)');
ylabel("y'(t)");
grid on;
title('Van der Pol Oscillator - \alpha= 0.03 - phase plane');
```

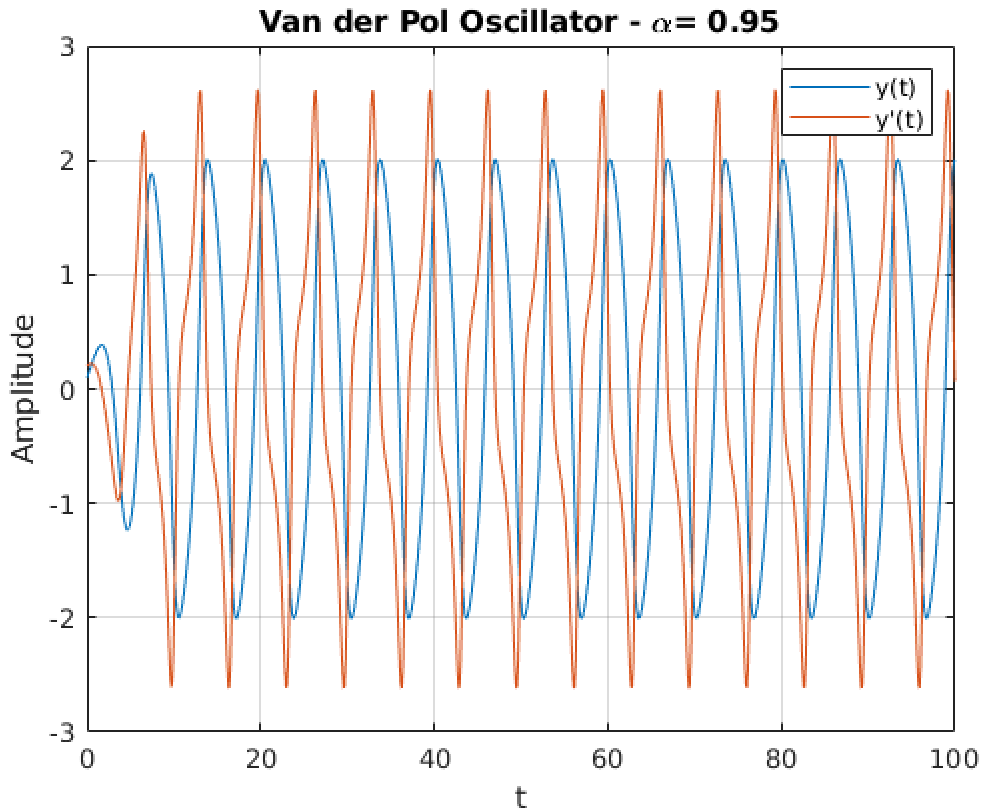
```
P01-A Van der Pol - \alpha=0.3
Set initial conditions for y(0) and ydot(0):
Plot y(t) vs t for \alpha=0.03
Plot y(t) vs y'(t) for \alpha=0.03
```

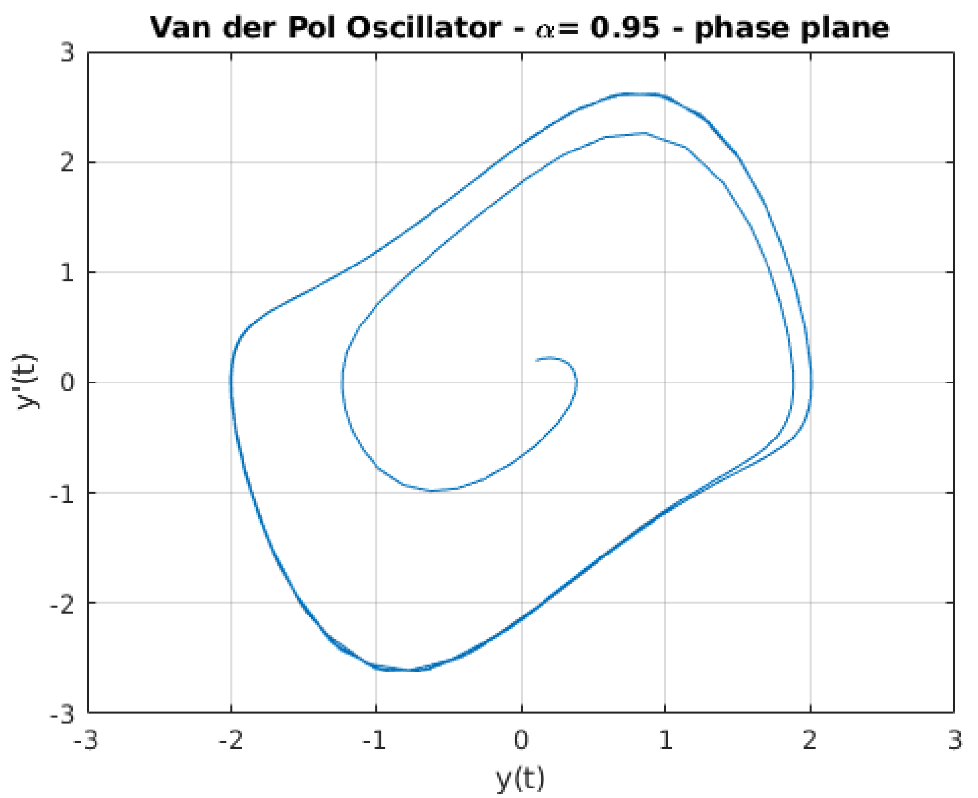

Van der Pol Oscillator - α= 0.03
y(t)
y'(t)
Amplitude
t
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
20
40
60
80
100


Van der Pol Oscillator - α= 0.03 - phase plane
y'(t)
y(t)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1

```
function xdot = VanDerPolA(t,x)
  alpha= 0.03;
  xdot = [x(2); -alpha*((x(1)^2)-1)*x(2) - x(1)];
end

clc
disp('P01-B Van der Pol - \alpha=0.95')
disp('Set initial conditions for y(0) and ydot(0):')
disp('Plot y(t) vs t for \alpha=0.95')

figure
[t,x]= ode23('VanDerPolB', t_interval, x0);
plot(t,x)
ylabel('Amplitude');
xlabel('t');
grid on;
title('Van der Pol Oscillator - \alpha= 0.95');
legend('y(t)', "y'(t)");

disp("Plot y(t) vs y'(t) for \alpha=0.95")
figure
plot(x(:,1),x(:,2))
xlabel('y(t)');
ylabel("y'(t)");
grid on;
title('Van der Pol Oscillator - \alpha= 0.95 - phase plane');
```

```
P01-B Van der Pol - \alpha=0.95
Set initial conditions for y(0) and ydot(0):
Plot y(t) vs t for \alpha=0.95
Plot y(t) vs y'(t) for \alpha=0.95
```


Van der Pol Oscillator - α= 0.95
y(t)
y'(t)
Amplitude
t
3
2
1
0
-1
-2
-3
0
20
40
60
80
100


Van der Pol Oscillator - α= 0.95 - phase plane
y'(t)
y(t)
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3

```
function xdot = VanDerPolB(t,x)
  alpha= 0.95;
  xdot = [x(2); -alpha*((x(1)^2)-1)*x(2) - x(1)];
end
```

Lorenz Attractor Chaotic System:

- $\dot{x}_1 = -\sigma(x_1 - x_2)$
- $\dot{x}_2 = rx_1 - x_2 - x_1x_3$
- $\dot{x}_3 = -bx_3 + x_1x_2$
- Time Interval 150 sec.
- All inital condition equal to 0.5.
- Plot state versus time and 3D plot of x1, x2, x3.

```
clc

t_intv= [0 150];
x_0= [0.5 0.5 0.5]'; % initial conditions for x(t)
[t,x]= ode23('Lorenz', t_intv, x_0);

figure
plot(t,x)
grid on;
title('Lorenz Attractor Chaotic System');
ylabel('Magnitude');
xlabel('t (sec)');
legend("x_1'", "x_2'", "x_3'");
hold on;

figure
plot3(x(:,1),x(:,2),x(:,3))
grid on;
title('Lorenz Attractor Chaotic System');
ylabel('x_2');
xlabel('x_1');
zlabel('x_3');
hold on;
```

```
Error using alpha (line 34)
Not enough input arguments.

Error in Lorenz (line 11)
    alpha;

Error in odearguments (line 90)
f0 = feval(ode,t0,y0,args{:});   % ODE15I sets args{1} to yp0.

Error in ode23 (line 114)
```

```
   odearguments(FcnHandlesUsed, solver_name, ode, tspan, y0, options,
 varargin);

Error in Copy_of_main_HW02_BM (line 110)
[t,x]= ode23('Lorenz', t_intv, x_0);
```

```
function xdot = Lorenz(t,x)
  sigma= 10; r=28; b=8/3;
  xdot = [-sigma*(x(1)-x(2)); r*x(1)-x(2)-x(1)*x(3); -
b*x(3)+x(1)*x(2)];
end
```

Voltera Predator-Prey System:

- $\dot{x}_1 = -x_1 + x_1 x_2$
- $\dot{x}_2 = x_2 - x_1 x_2$
- Initial conditions to be evenly spaced for $x_1 = [-2, 2], x_2 = [-2, 2]$.
- Plot phase plane on $[-5, 5]$ by $[-5, 5]$

```
clc

x0_set = -2:.5:2;
t_intv= [0 100];
x_0= [4.5, 9.7]'; % initial conditions for x(t)

figure
[t,x]= ode23('Voltera', t_intv, x_0);
plot(t,x)
hold on;
grid on;
title('Voltera Predator-Prey System');
ylabel('x');
xlabel('t (sec)');
legend('Predator', 'Prey');


t_intv= [0 10];
figure
for i=x0_set
  for j=x0_set
    x0 = [i; j];
    [t,x]= ode45('Voltera', t_intv, x0);
    plot(x(:,1),x(:,2))
    hold on;
  end
end
title('Voltera Predator-Prey System - Phase Plane');
ylabel('x_2 - Predator');
xlabel('x_1 - Prey');
axis([-5 5 -5 5]);
```

```
grid on;

function xdot = Voltera(t,x)
     xdot = [-x(1)+x(1)*x(2); x(2)-x(1)*x(2)];
end
```

Published with MATLAB® R2021a