James Watt School of Engineering

University of Glasgow

Simulation of Engineering Systems 3 & Simulation of Aerospace Systems

Lab Exercise 2: Introduction to Matlab Results

Introduction

The following highlights the matlab code and results that should have been obtained in the second lab exercise.

2.1: Before using the computer

Choosing $x_1 = y$ and $x_2 = dy/dt$ gives the following State space model for Van der Polo's Equation:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -\mu\left(x_1^{\,2} - 1\right)x_2 - x_1$$

It can't be put into standard state space form because it has nonlinear terms that can't be represented in the linear matrix equation.

2.2: Van der Pol's Equation

Euler Integration

The figure below shows the plot from the original code.

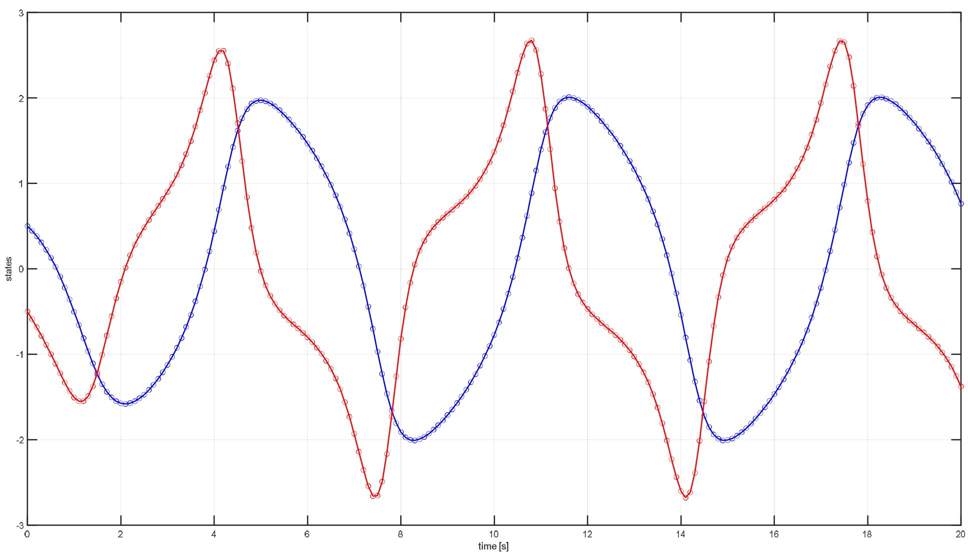

Figure 1: Original Response from vanpool matlab code (μ = 1)

The response shown in Figure 2 is obtained when the following commands are executed

```
clf
subplot(2,1,1)
plot(tout,xout(:,1))
```

```
ylabel(“x1”); xlabel(“time [s]”)
subplot(2,1,2)
plot(tout,xout(:,2))
ylabel(“x2”); xlabel(“time [s]”)
```

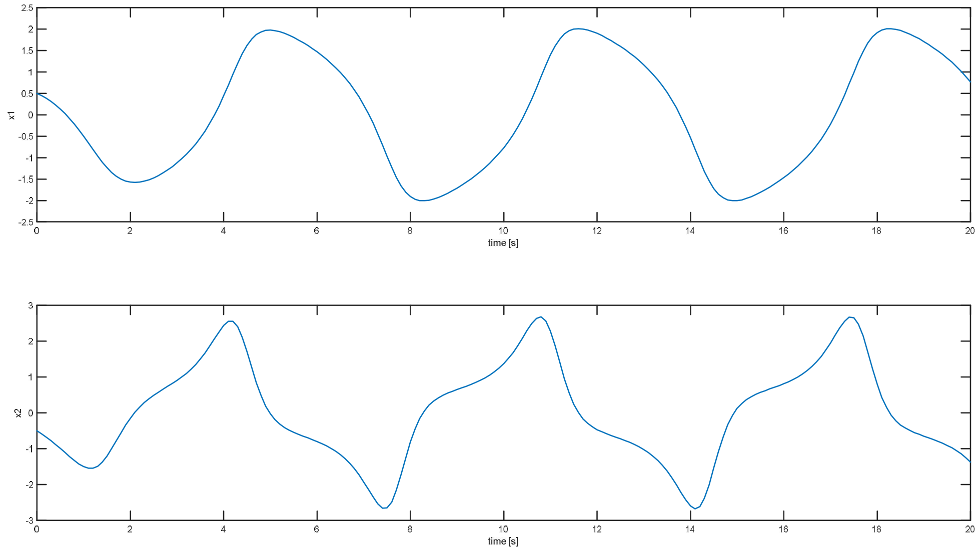

Figure 2: Van der Pol states plotted separately.

When rk4 is used the following function call is used in the INTEG Section of the simulation programme:

x = RK4int(‘model’, stepsize, x, u);

The following responses are obtained.

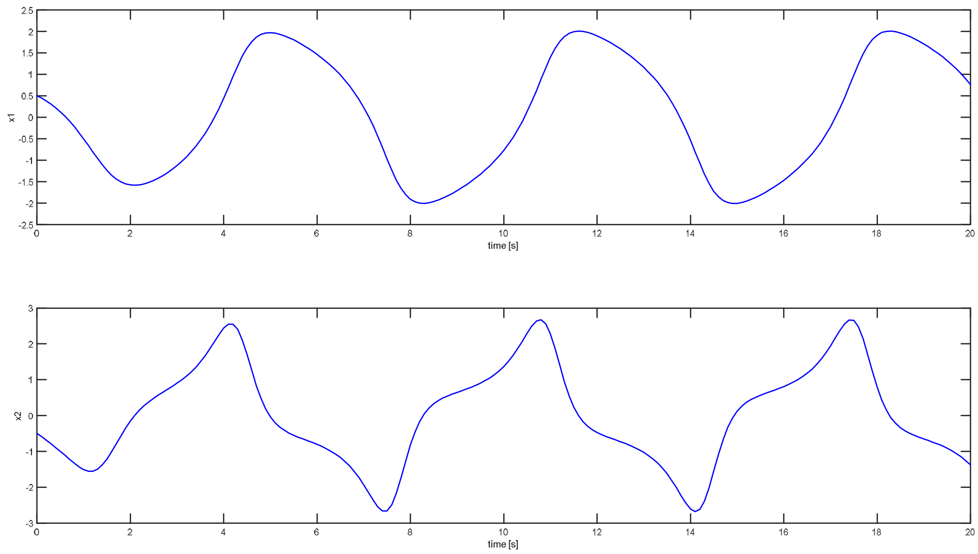

Figure 3: Van der Pol states with RK4 (Euler represented by dash line)

For the chosen step-size the Euler and RK4 are identical.

The following plot shows states plotted against one another using plot(xout(:1),xout(:,2))

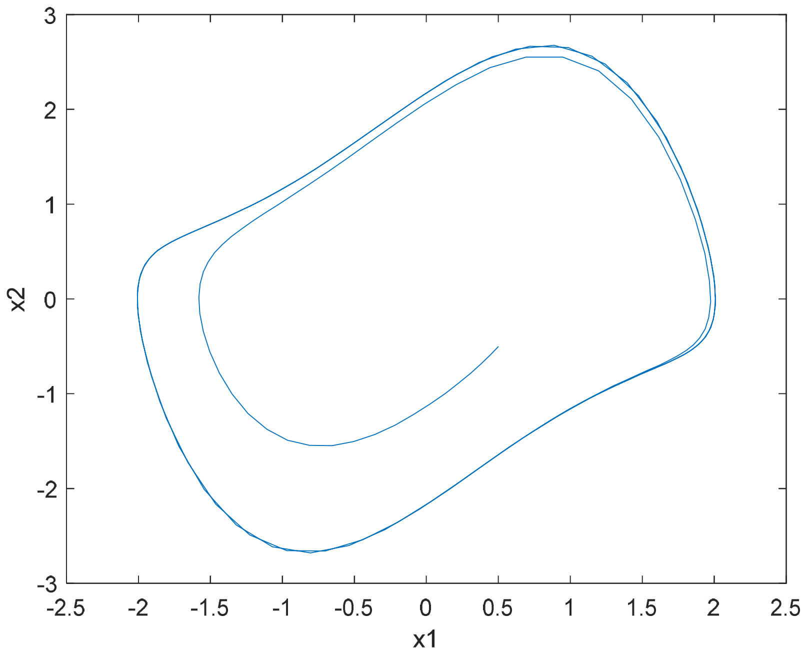

Figure 4: State X2 against state X1

Plot shows states a cyclic and looks similar to the response found in Lab 1.

Changing the value for the gain mu varies the amplitude of the state x2 (rate of change) and the frequency of the oscillation.

A negative value for mu gives a controller system which doesn’t oscillate:

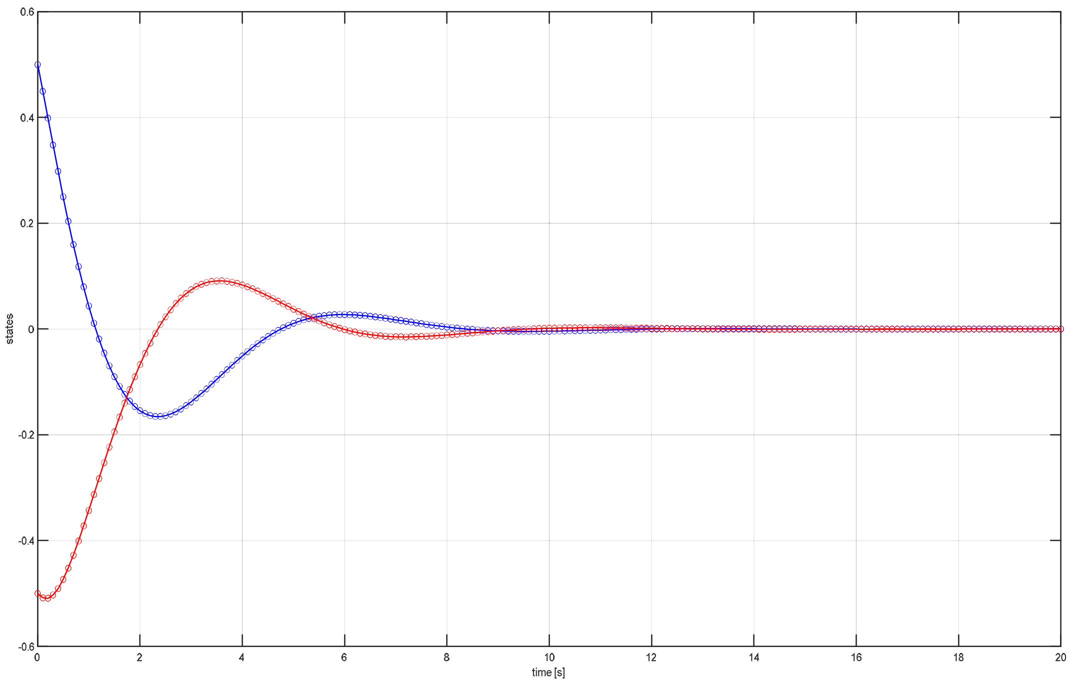

Figure 5: Van der Pol Oscillatory response with mu=-1

2.3: Modified Oscillator

The modified oscillatory should be represented in the model function as follows:

```
function xdot = modmodel(x,u)

global mu                   % global parameter transferred from main
program

xdot(1,1) = x(2);
xdot(2,1) = -mu*(x(1)^2 - 0.3*x(1)^3 - 1)*x(2)-x(1);
```

The resulting response of this modified system is shown in Figure 6.

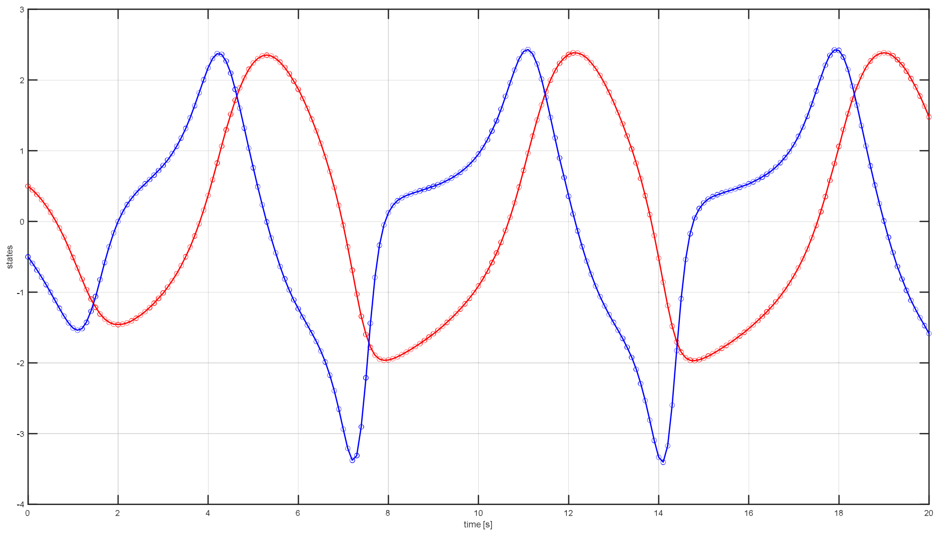

Figure 6: Modified Oscillatory with mu = 1

The output from the modified oscillatory shows that the positive section of the waveform is no longer the same shape, amplitude and duration as the negative part. Due to the cubic term in the model, the output for x1 has a shorter duration but larger amplitude on the positive side. On the negative side the duration of x1 is longer but the amplitude is smaller.

2.4: Chaotic System

The chaotic system can be represented by a single state space model where x_1=x, x_2=y and x_3=z, thus:

$$\dot{x}_1 = 10(-x_1 + x_2) + u$$
$$\dot{x}_2 = 28x_1 - x_2 - x_1x_3$$
$$\dot{x}_3 = -\frac{8}{3}x_3 + x_1x_2$$

The resulting matlab code for the model becomes:

```
function xdot = chaotic(x,u)

xdot(1,1) = 10*(-x(1)+x(2)) + u;
xdot(2,1) = 28*x(1) - x(2) - x(1)*x(3);
xdot(3,1) = -8/3*x(3) + x(1)*x(2);
```

Also, the initial segment requires 3 states and 3 state derivatives to be defined. When the input is set to zero we get the following responses:

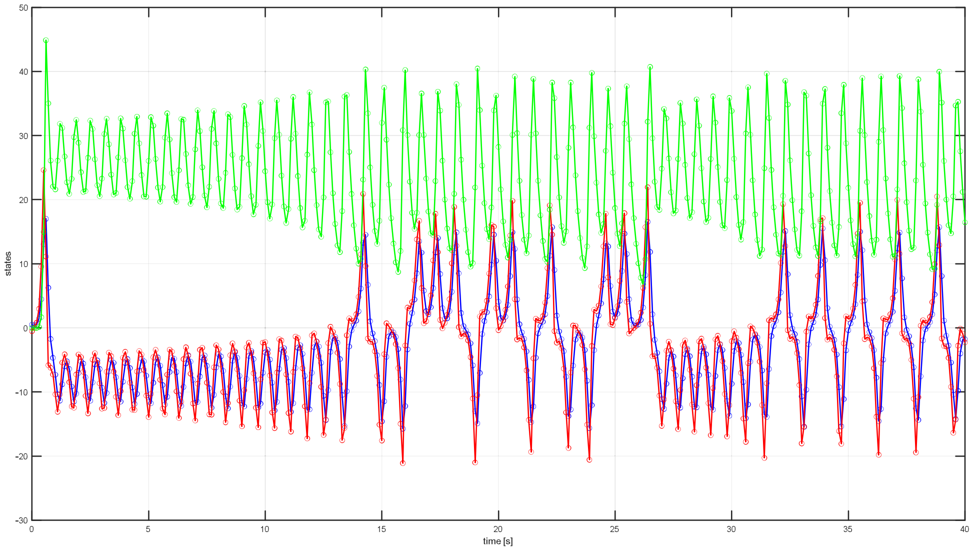

Figure 7: Output of the chaotic system without control

The response shows that the three states change in a chaotic manner after the initial wind-up period (i.e. up to time = 12s).

The controller equation is implemented in the Control Section of the Dynamic Segment in the simulation i.e.

```
u = -(24*x(1)+18.64*x(2));
```

This gives the following output response from the simulation of this system:

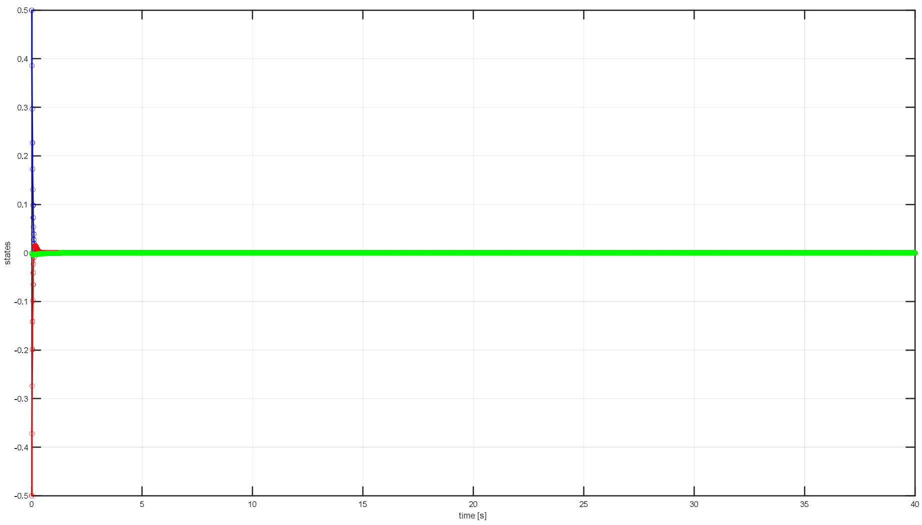

Figure 8: Output of the chaotic system with control

This shows that with a suitably designed control system you can regulate the chaotic behaviour for this system.