

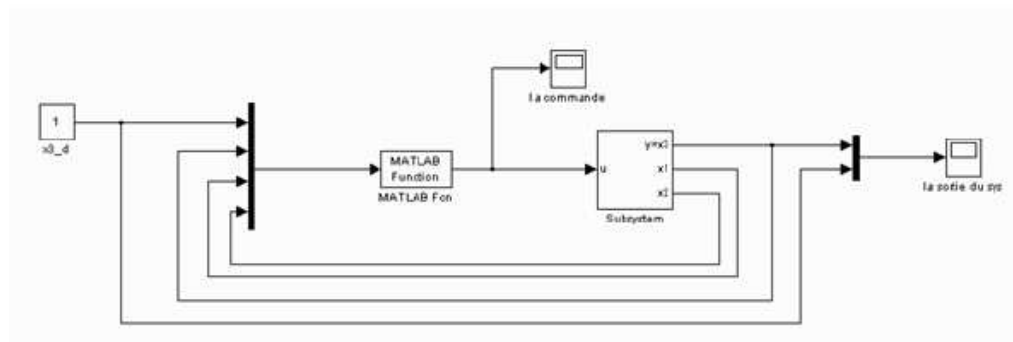
Lab 4 : Feedback linearization

Ex 1 : Consider the nonlinear system

$$\dot{x} = \begin{pmatrix} 0 \\ x_1 - 0.5x_2^2 \\ x_1 - x_2 \end{pmatrix} + \begin{pmatrix} e^{x_2} \\ -e^{x_2} \\ 0 \end{pmatrix} u$$

$$y = h(x) = x_3$$

Closed loop simulation :



```
function out=commande(in)

x_3_d=in(1);
x_3=in(2);
x_1=in(3);
x_2=in(4);

U=(8*(x_3-x_3_d)+17*(x_1-x_2)+0.1*(-x_1+0.5*x_2^2)-x_2*(-.5*x_2^2+x_1))/(exp(x_2)*(-x_2-1))

out=U
```

Ex 2 : For the nonlinear system :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = a_1 x_1 x_2 + a_2 x_2^2 + a_3 x_2 \end{cases}$$

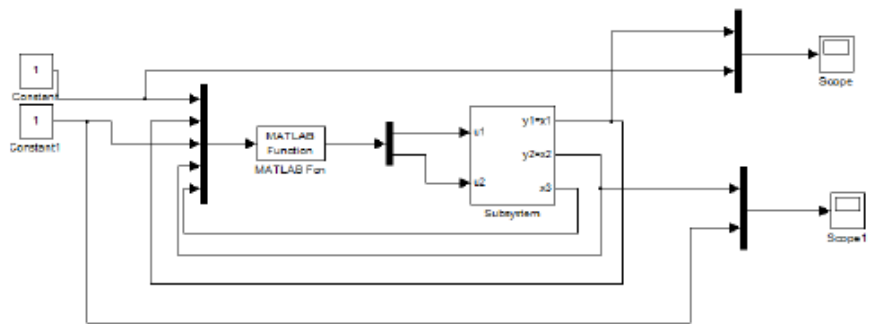
$$a_1 = -0.8636, \quad a_2 = -0.1454, \quad a_3 = -0.0074,$$

Design the feedback linearization controller.

Ex 3 : Design the feedback linearization controller for the nonlinear system :

$$\dot{x} = \begin{pmatrix} x_1 \\ x_3 \\ x_3 x_1 \end{pmatrix} + \begin{pmatrix} \cos(x_3) \\ 0 \\ 1 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u_2$$

$$y = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$



```
function out=commande3(in)
x_1_d=in(1);
y_1=in(2);
x_2_d=in(3);
y_2=in(4);
x_3=in(5);
u_1=-1/cos(x_3)*(2*y_1-x_1_d)
u_2=1/cos(x_3)*(2*y_1-x_1_d)-20*(y_2-x_2_d)-6*x_3

out(1)=u_1
out(2)=u_2
```

Guidelines

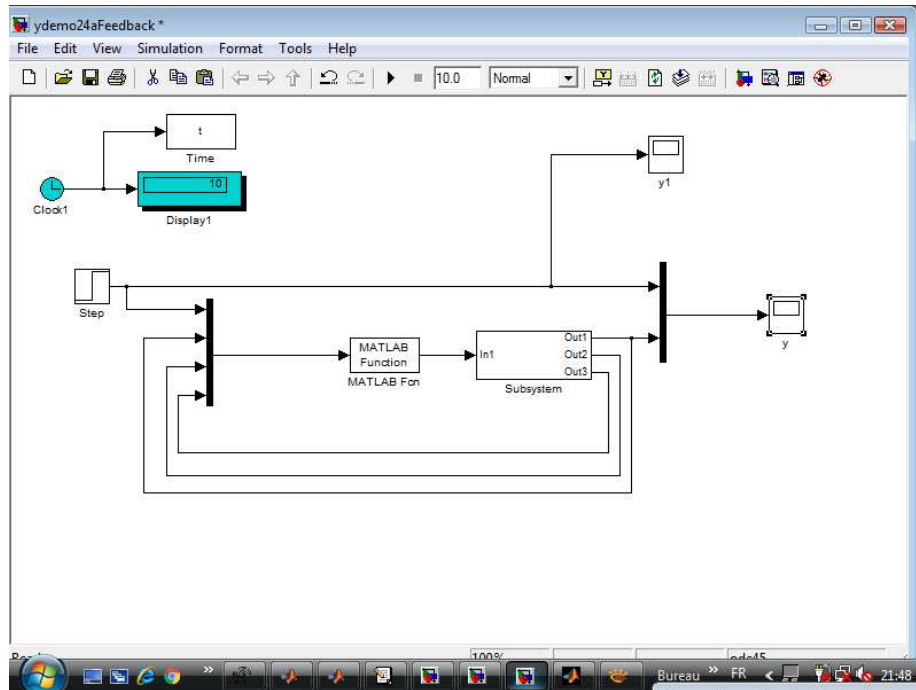


FIGURE 1 – The overall simulation model : ydemo24aFeedback.mdl

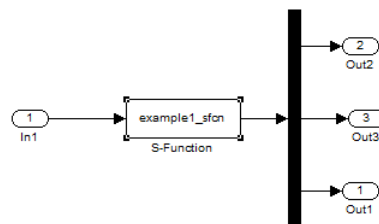


FIGURE 2 – The subsystem content with one input port and three output ports

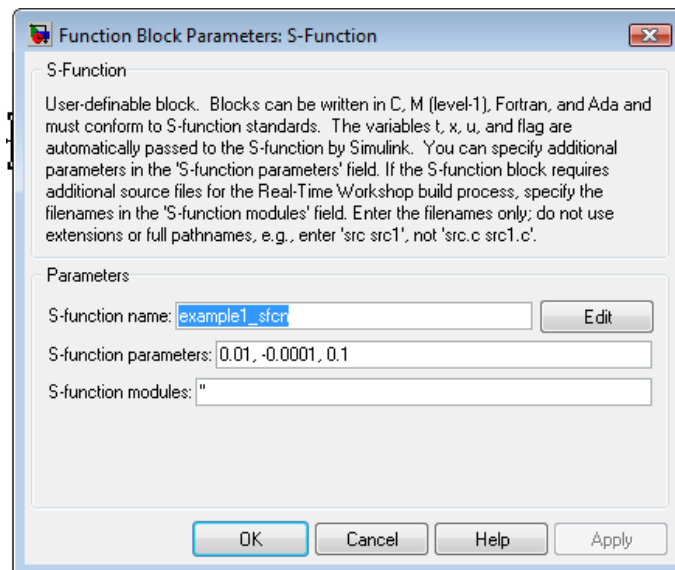


FIGURE 3 – The s-function block

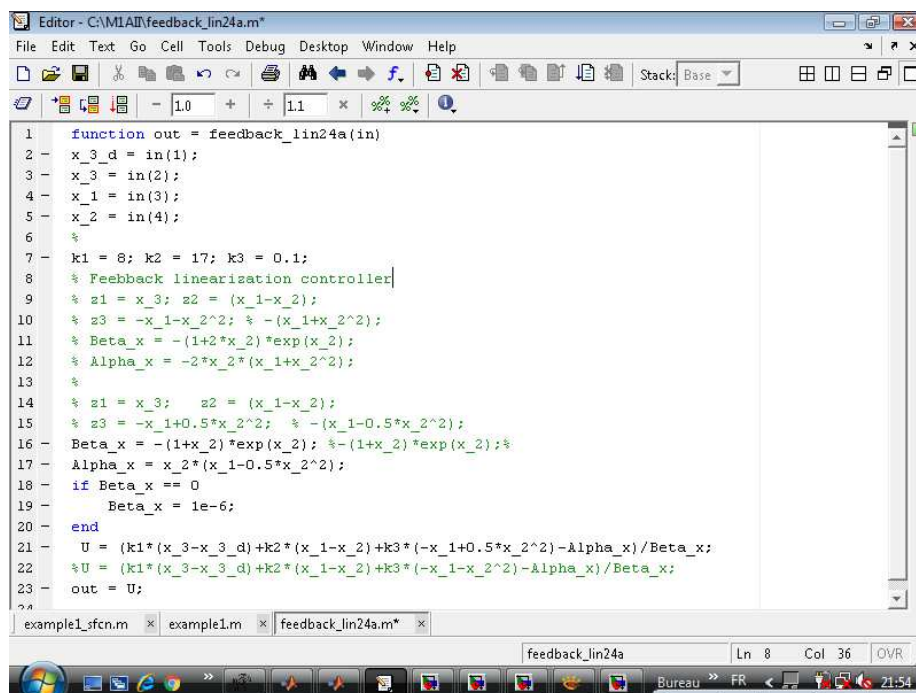
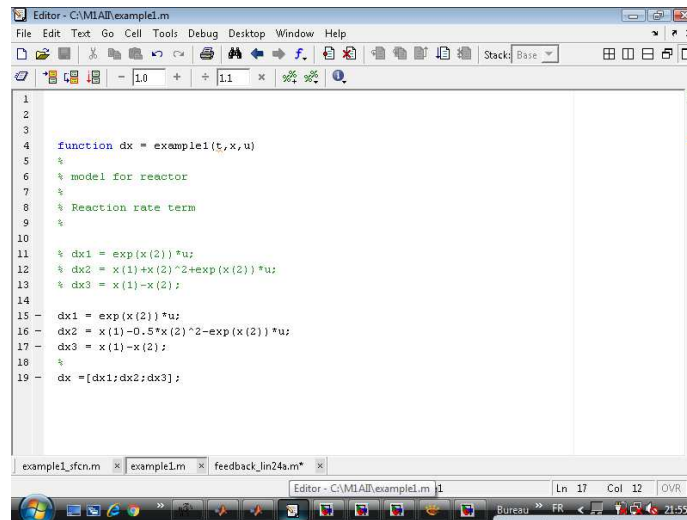


FIGURE 4 – The Matlab function named in our case Feedback_24lin.m

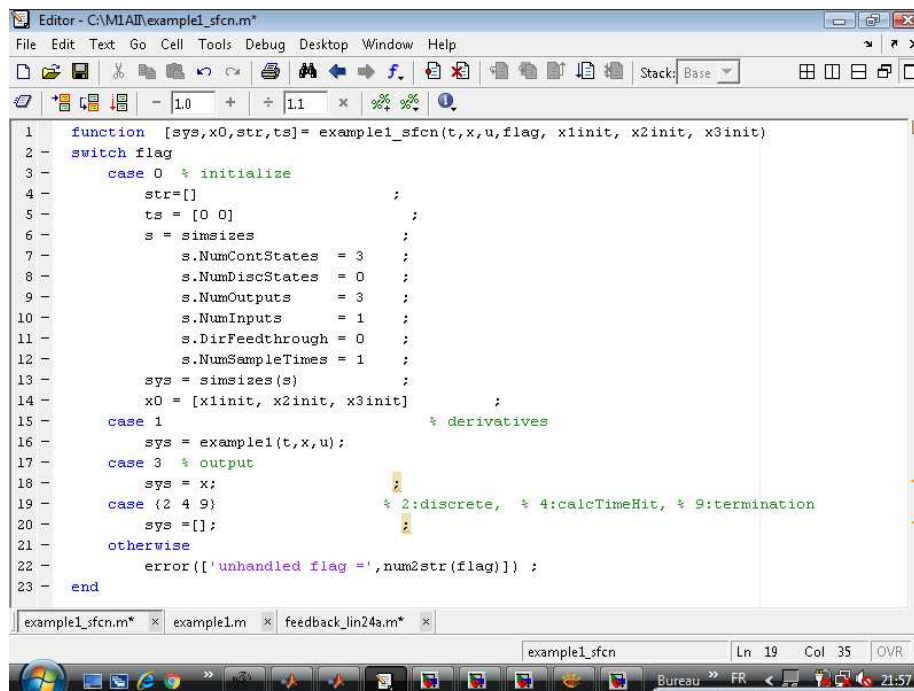


```

1
2
3
4 function dx = example1(t,x,u)
5 %
6 % model for reactor
7 %
8 % Reaction rate term
9 %
10
11 % dx1 = exp(x(2))*u;
12 % dx2 = x(1)+x(2)^2+exp(x(2))*u;
13 % dx3 = x(1)-x(2);
14
15 - dx1 = exp(x(2))*u;
16 - dx2 = x(1)-0.5*x(2)^2-exp(x(2))*u;
17 - dx3 = x(1)-x(2);
18 %
19 - dx =[dx1;dx2;dx3];

```

FIGURE 5 – The Matlab function named example1.m of the nonlinear model



```

1 function [sys,x0,str,ts]= example1_sfcn(t,x,u,flag, x1init, x2init, x3init)
2 switch flag
3 case 0 % initialize
4     str=[] ;
5     ts = [0 0] ;
6     s = simsizes ;
7     s.NumContStates = 3 ;
8     s.NumDiscStates = 0 ;
9     s.NumOutputs = 3 ;
10    s.NumInputs = 1 ;
11    s.DirFeedthrough = 0 ;
12    s.NumSampleTimes = 1 ;
13    sys = simsizes(s) ;
14    x0 = [x1init, x2init, x3init] ;
15 case 1
16     sys = example1(t,x,u); % derivatives
17 case 3 % output
18     sys = x;
19 case {2 4 9} % 2:discrete, % 4:calcTimeHit, % 9:termination
20     sys = [];
21 otherwise
22     error(['unhandled flag =',num2str(flag)]);
23 end

```

FIGURE 6 – The Matlab s-function named in our case example1_2sfcn.m

FIGURE 7 –