SIMULACIÓN DE IDENTIFICADORES A TRAVÉS DE REDES NEURONALES Y CONTROL INVERSO GENERALIZADO

•ING. JOSE GUILLERMO GUARNIZO MARIN

joguiguma@ieee.org

ESTUDIANTE MAESTRIA ATOMATIZACION INDUSTRIAL

UNIVERSIDAD NACIONAL

•ING. JUAN CARLOS LOPEZ RODRIGUEZ

wittolfr@yahoo.ca

INGENIEROS ELECTRONICOS

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

FACULTAD DE INGENIERÍA

LINEA DE INVESTIGACIÓN EN FUENTES ALTERNATIVAS DE ENERGÍA LIFAE

Agradecimientos por su colaboración a los profesores:

- Ing. José Jairo Soriano Méndez .

  Ingeniero Electrónico. Profesor Universidad Distrital Francisco José de Caldas. Miembro Laboratorio de Automática, Microelectrónica e Inteligencia Computacional LAMIC.

  josoriano@udistrital.edu.co

- Ing. Msc. Javier Antonio Guacaneme Moreno.

  jguacaneme@udistrital.edu.co

  Ingeniero Electrónico. Profesor Universidad Distrital Francisco José de Caldas. Miembro Línea de Investigación en Fuentes Alternativas de Energía LIFAE

- Ing. Msc. Cesar Leonardo Trujillo Rodriguez.

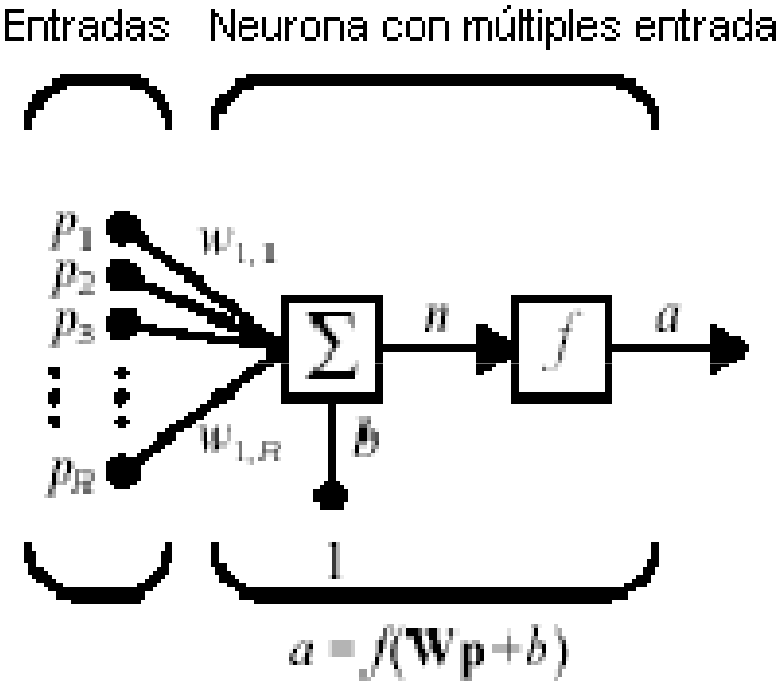
  cltrujillo@udistrital.edu.co

  Ingeniero Electrónico. Profesor Universidad Distrital Francisco José de Caldas. Miembro Línea de Investigación en Fuentes Alternativas de Energía LIFAE

REDES NEURONALES

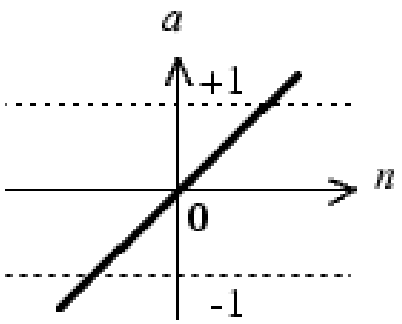
- Modelos computacionales basados parcialmente en emular el comportamiento del cerebro.
- Compuestas por unidades fundamentales llamadas neuronas, donde se interconectan las unas a las otras.
- Estas se conectan bajo ciertas reglas, cada conexión tiene un determinado peso, umbral y función de activación.
- Pueden ser entrenadas para generar respuestas específicas para una entrada deseada, imitar comportamiento de sistemas, identificación de patrones, entre muchas otras aplicaciones.
- Existen diversos tipos de topologías y entrenamientos de redes neuronales según las necesidades existentes

PERCEPTRON

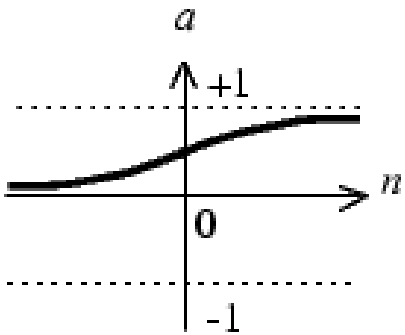


$$n = w_{1,1}p_1 + w_{1,2}p_2 + \ldots + w_{1,R}p_R + b$$

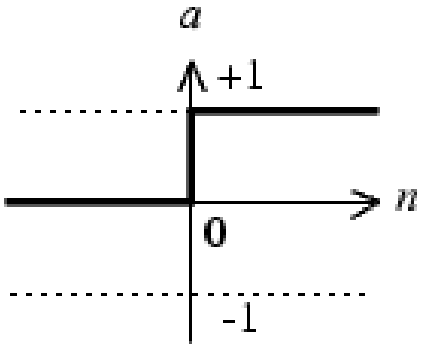
ALGUNAS FUNCIONES DE ACTIVACIÓN



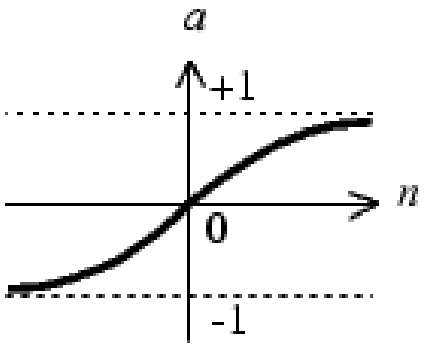
$a = purelin(n)$



$a = logsig(n)$



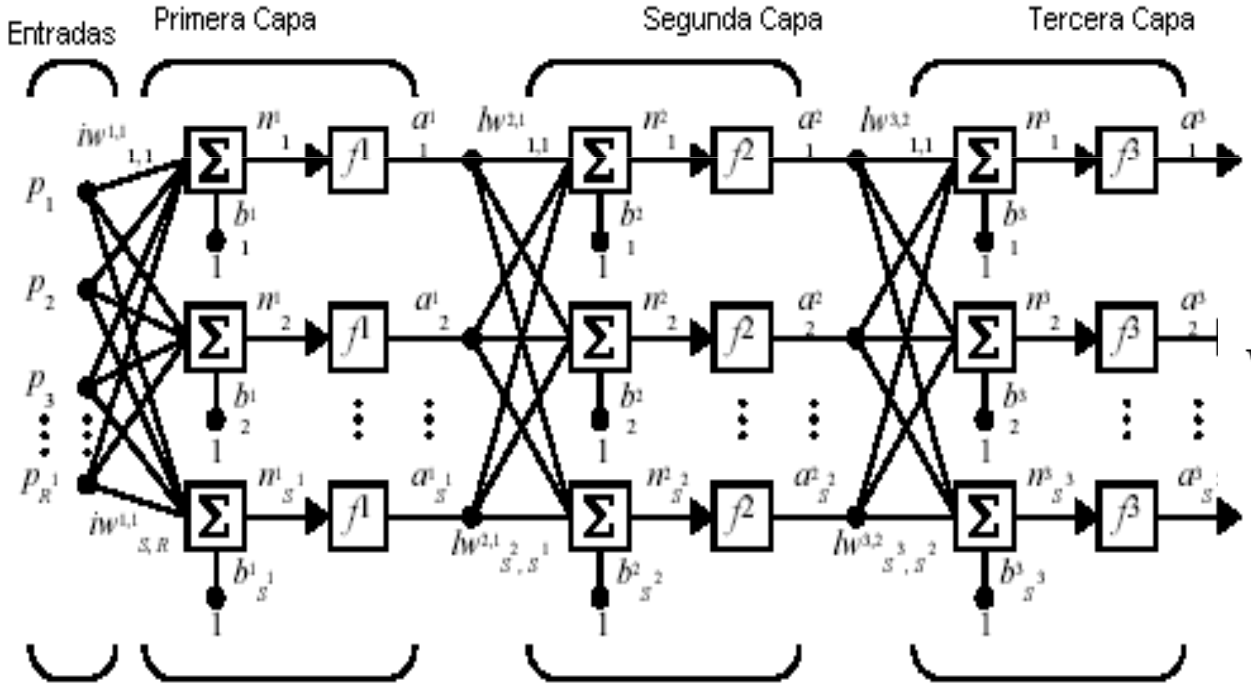
$a = hardlim(n)$



$a = tansig(n)$

PERCEPTRON MULTICAPA

1969: Minsky y Seymour: No separabilidad linear del perceptron.
1986: Rumelhart, Hinton y Williams: perceptron Multicapa. Back Propagation.



$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ & & & \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} b1 \\ b2 \\ \\ \\ bs \end{bmatrix}$$

$$\mathbf{a}^1 = \mathbf{f}^1(\mathbf{IW}^{1,1}\mathbf{p} + \mathbf{b}^1) \qquad \mathbf{a}^2 = \mathbf{f}^2(\mathbf{LW}^{2,1}\mathbf{a}^1 + \mathbf{b}^2) \qquad \mathbf{a}^3 = \mathbf{f}^3(\mathbf{LW}^{3,2}\mathbf{a}^2 + \mathbf{b}^3)$$

$$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{LW}^{3,2}\,\mathbf{f}^2(\mathbf{LW}^{2,1}\mathbf{f}^1(\mathbf{IW}^{1,1}\mathbf{p} + \mathbf{b}^1) + \mathbf{b}^2) + \mathbf{b}^3)$$

ENTRENAMIENTO

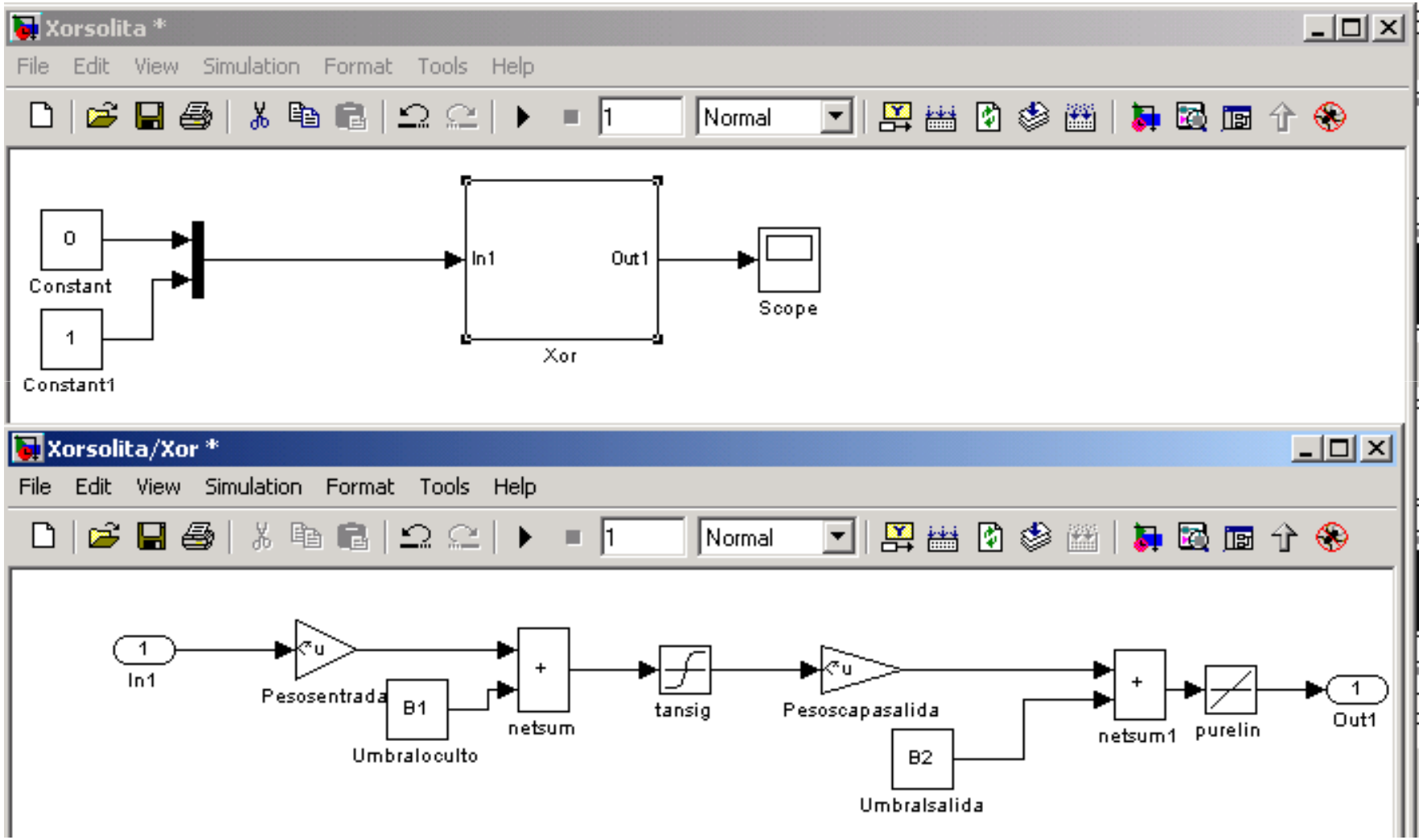
- A partir patrones Entrada-Salida
- Técnica del descenso por el gradiente.

$$E(k) = \frac{1}{2} * \sum_{j=1}^{N_E} (y_d - y_N)^2 \; = f(w_{kj})$$

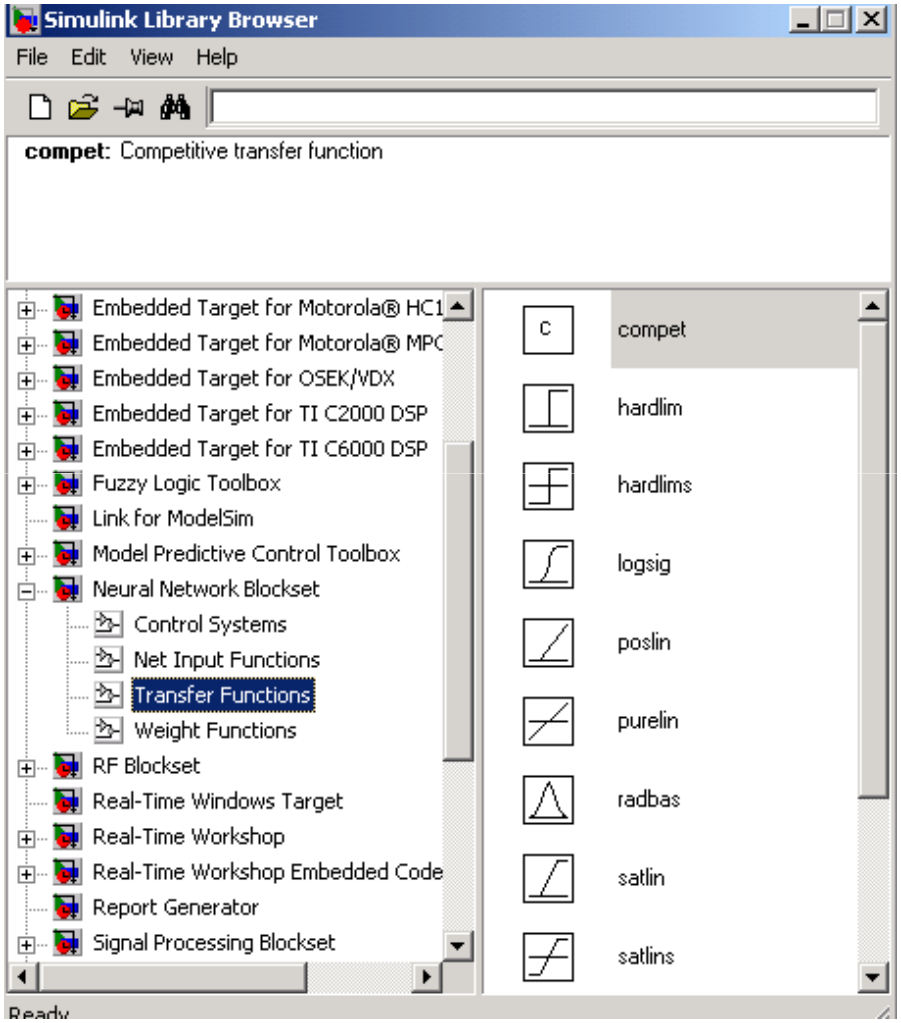
$$w^R(t+1) = w^R(t) - \alpha \Delta w^R$$

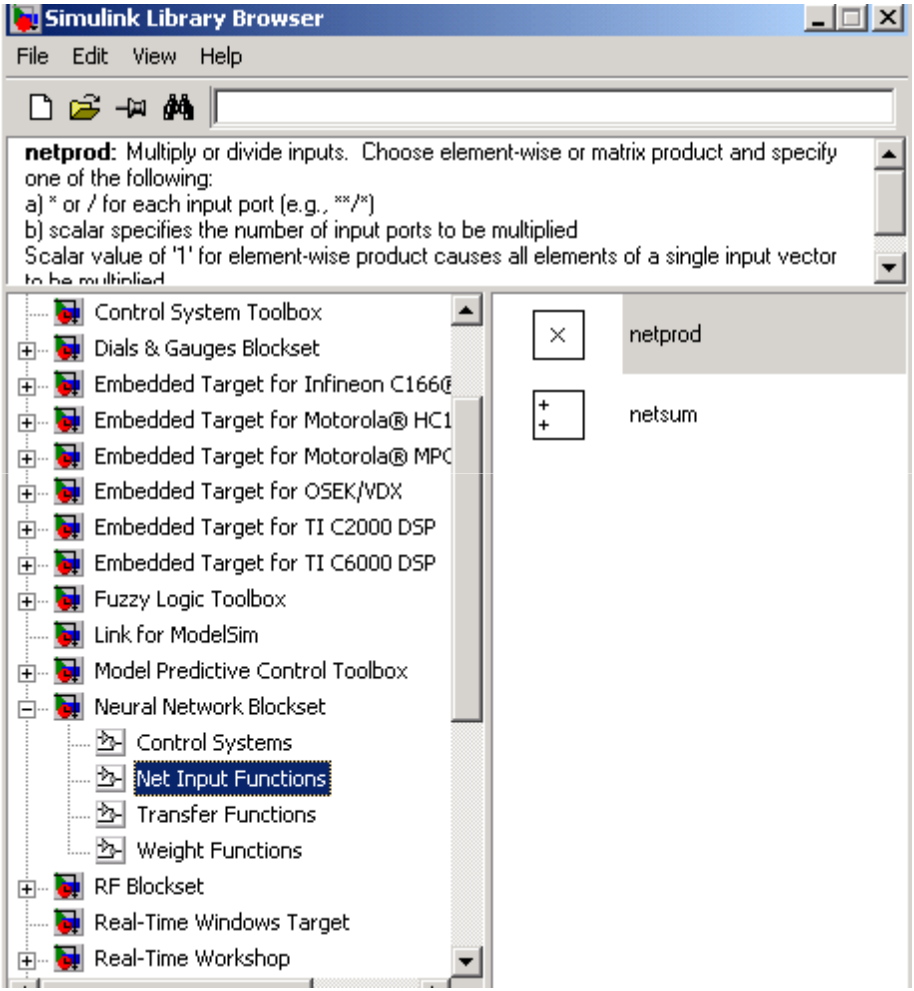
$$w^R(t+1) = w^R(t) - \alpha (y_d - y_N) * f'(n) * p_R$$

Red Neuronal En Simulink

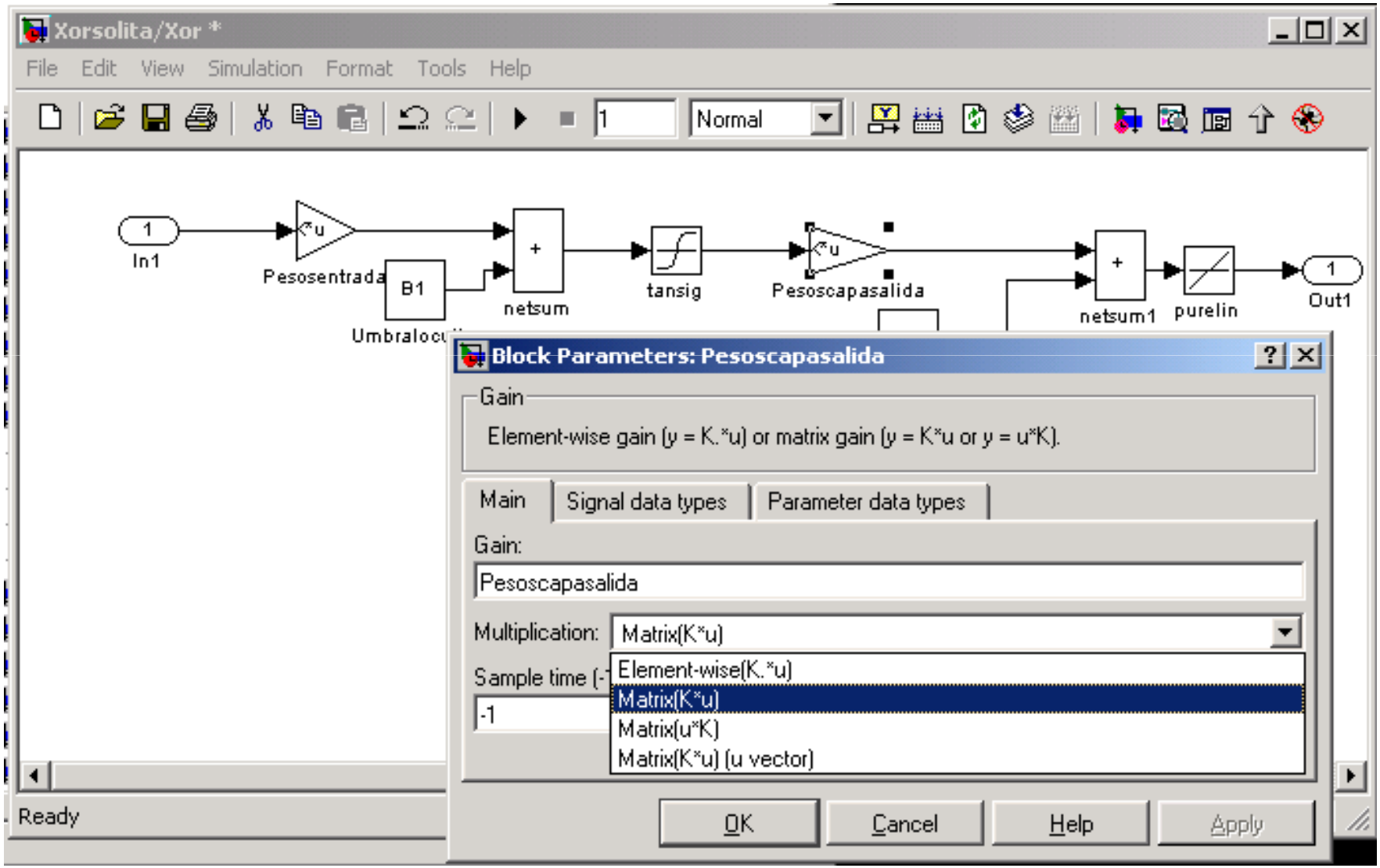


Selección Componentes

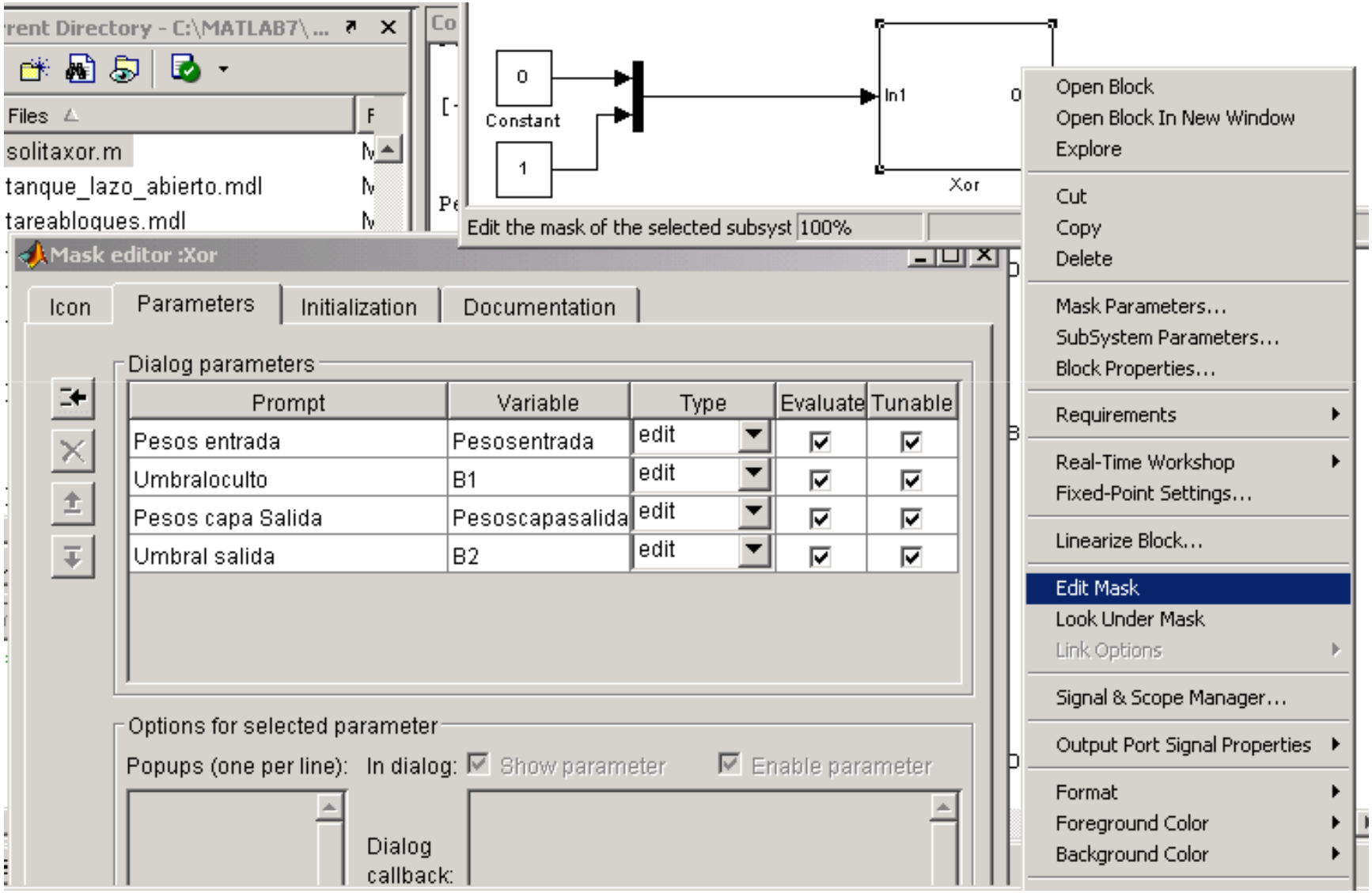




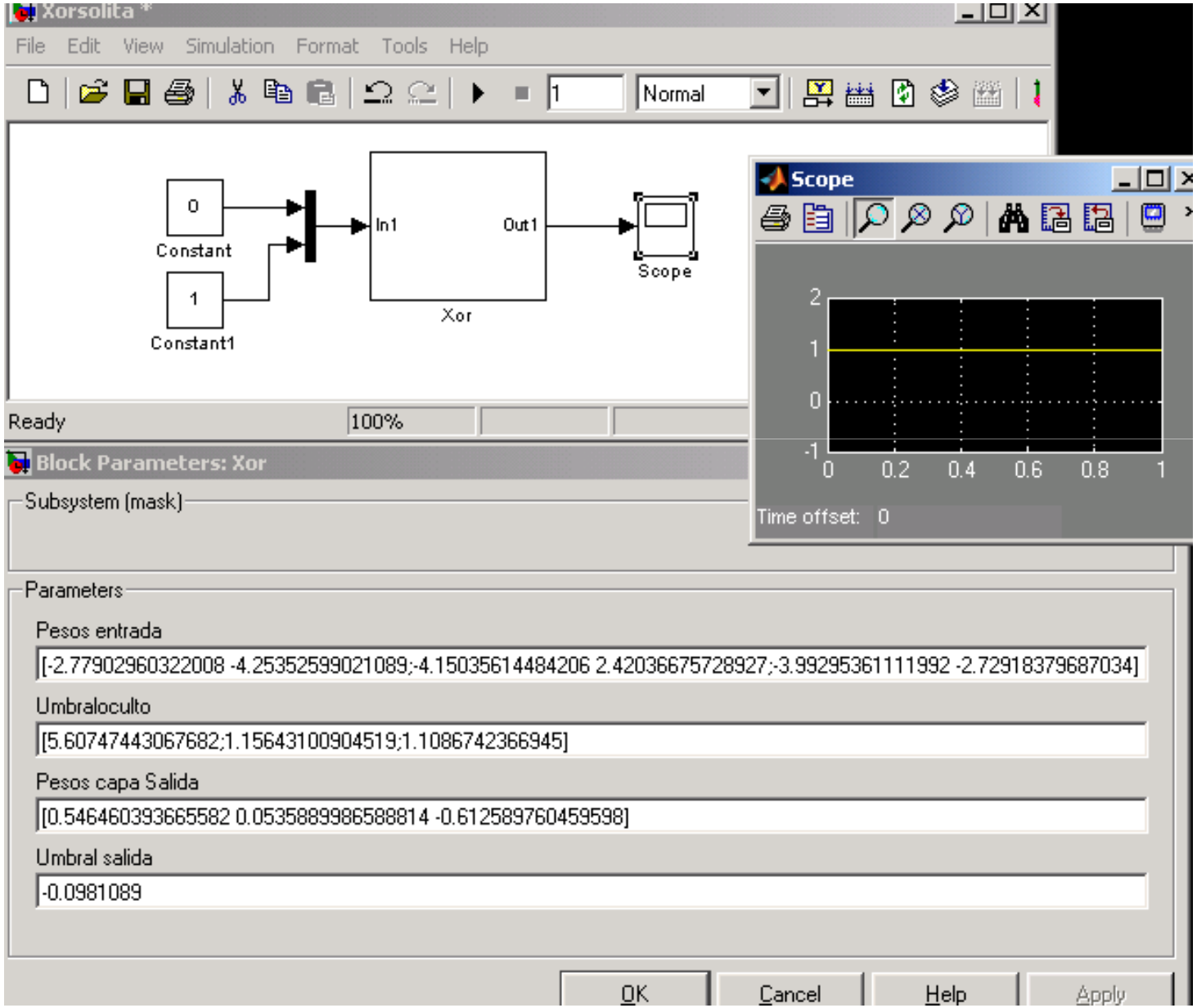
Edición etiquetas



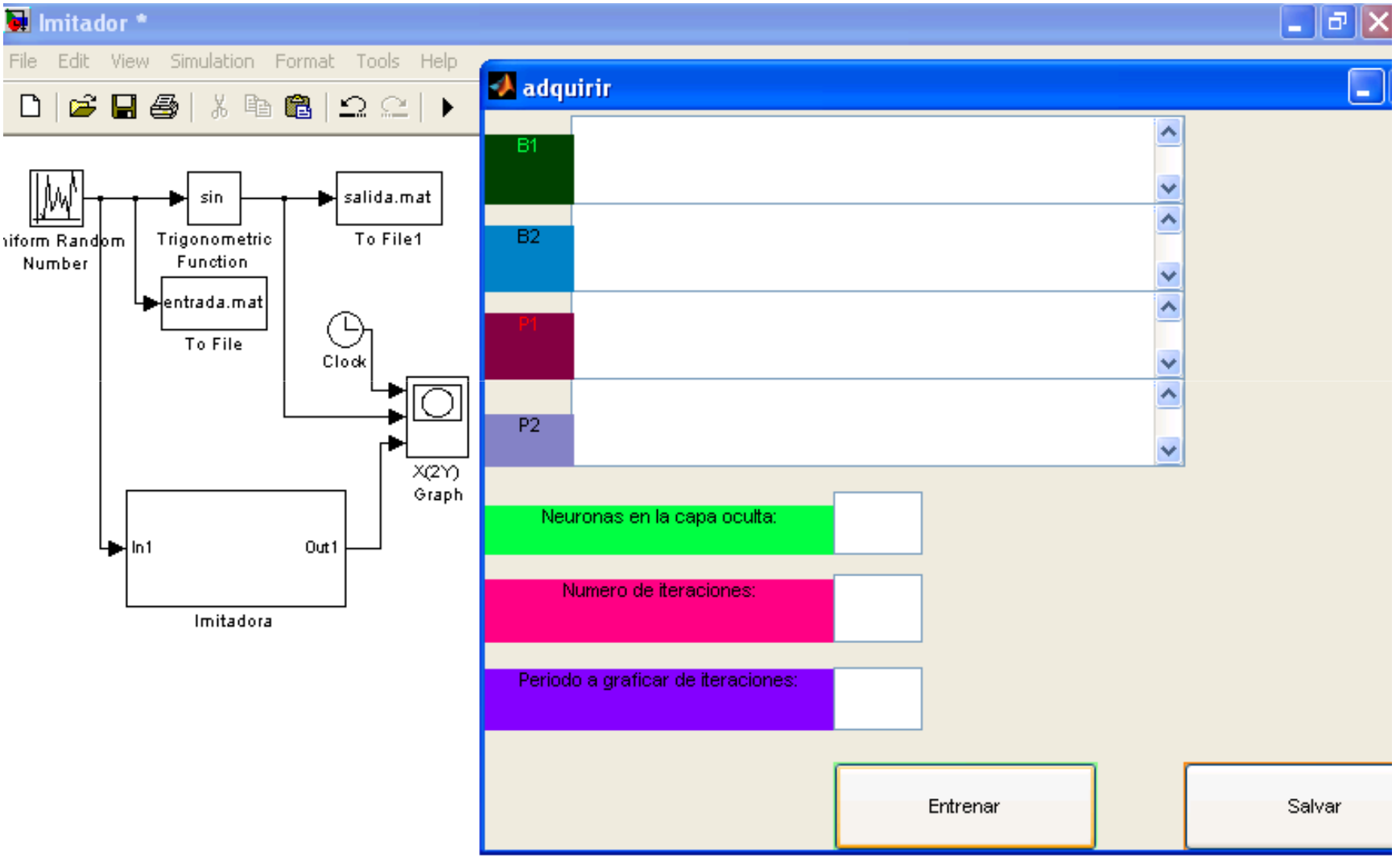
Edición etiquetas



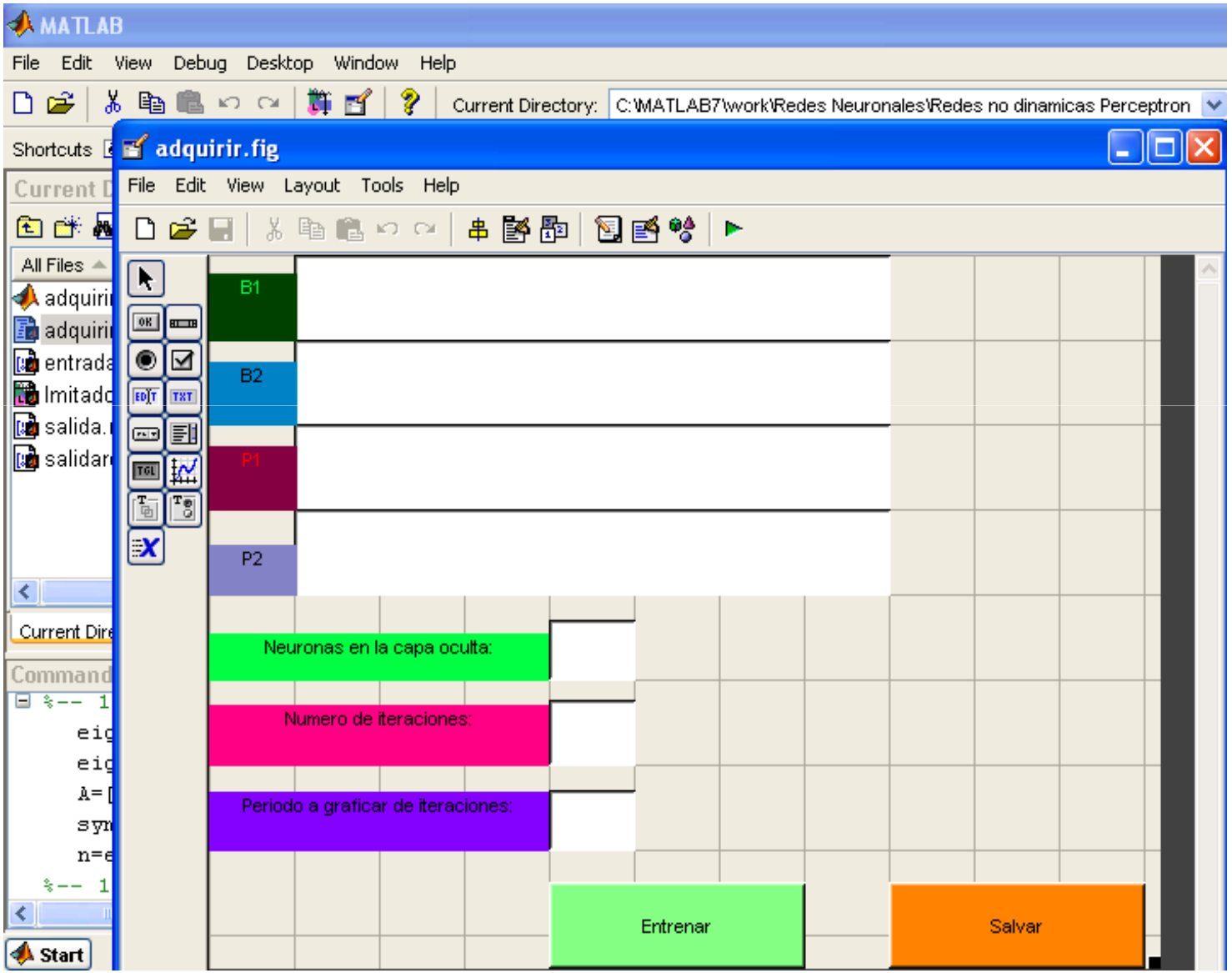
Agregar Parámetros



Imitador de Funciones



GUIDE


MATLAB
File Edit View Debug Desktop Window Help
Current Directory: C:\MATLAB7\work\Redes Neuronales\Redes no dinamicas Perceptron
Shortcuts
adquirir.fig
File Edit View Layout Tools Help
B1
B2
P1
P2
Neuronas en la capa oculta:
Numero de iteraciones:
Periodo a graficar de iteraciones:
Entrenar
Salvar
Start

```
73
74    % Get default command line output from handles structure
75 -  varargout{1} = handles.output;
76    function model_open(handles)%
77    % Make sure the diagram is still open
78 -  if  isempty(find_system('Name','Imitador')),%Las 2 siguientes líneas son usadas para...
79 -      open_system('Imitador');    %abrir el archivo de simulink.
80 -  end
81    % --- Executes on button press in pushbutton1.
82    function pushbutton1_Callback(hObject, eventdata, handles)
83    % hObject    handle to pushbutton1 (see GCBO)
84    % eventdata  reserved - to be defined in a future version of MATLAB
85    % handles    structure with handles and user data (see GUIDATA)
86 -  model_open(handles)
87 -  NewStrVal = get(hObject, 'String');
88 -  NewVal = str2double(NewStrVal);
89 -  nl = str2double(get(handles.edit5,'String'));%Captura numero:Neuronas en la capa oculta
90 -  ep = str2double(get(handles.edit6,'String'));%Captura numero:Numero de iteraciones
91 -  sh = str2double(get(handles.edit7,'String'));%Captura numero:Periodo a graficar...
92                                                  %de iteraciones
93 -  load entrada %carga entrada.mat, con set de datos de entrada de la red neuronal
94 -  load salida%carga salida.mat.
95 -  p=p(2,:);%Carga fila 2 y todas las columnas de la variable de entrada
96 -  t=t(2,:);%Carga fila 2 y todas las columnas de la variable de salida
97
```

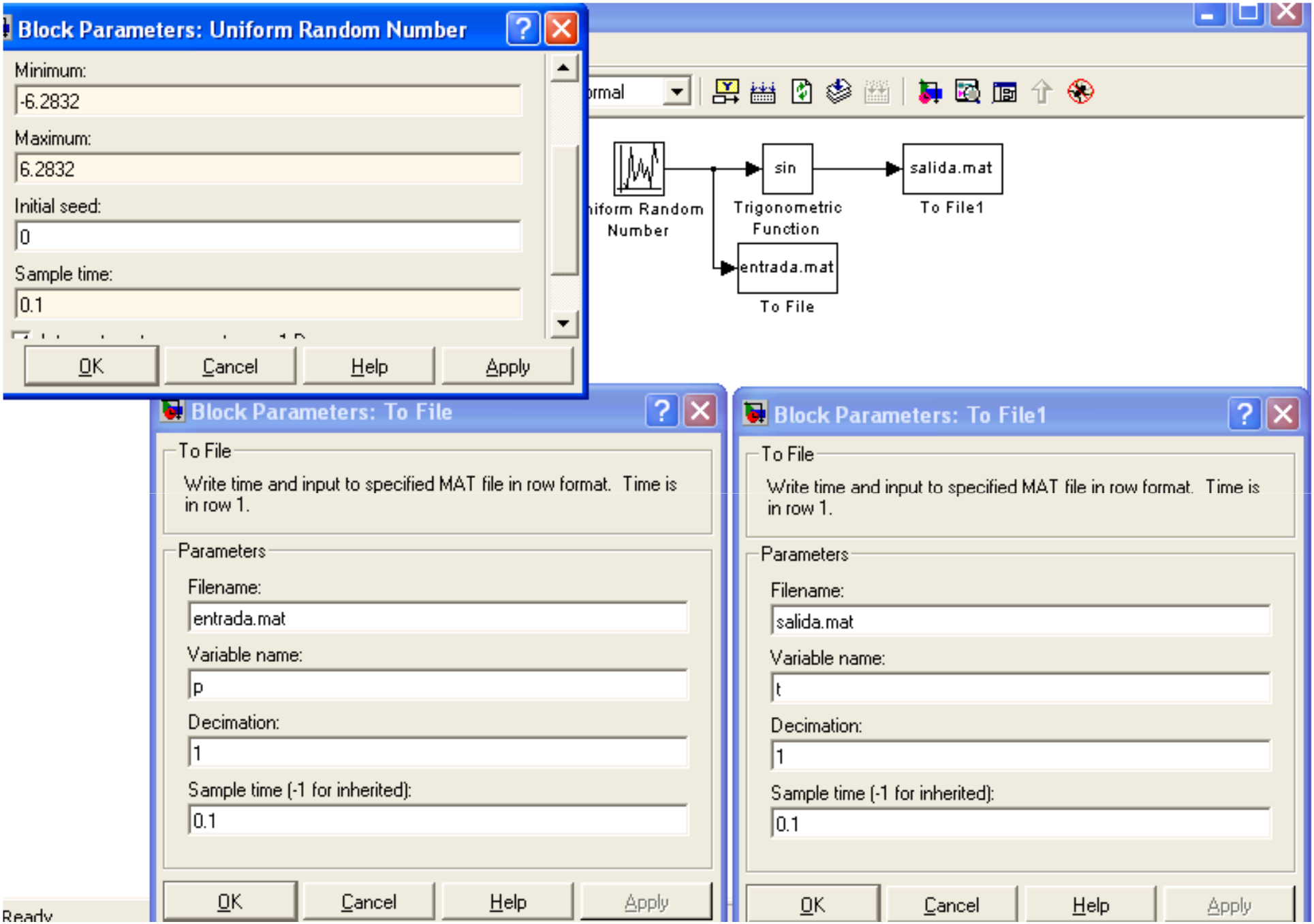
Editor - C:\MATLAB7\work\Redes Neuronales\Redes no dinamicas Perceptron\adquirir.m*

File Edit Text Cell Tools Debug Desktop Window Help

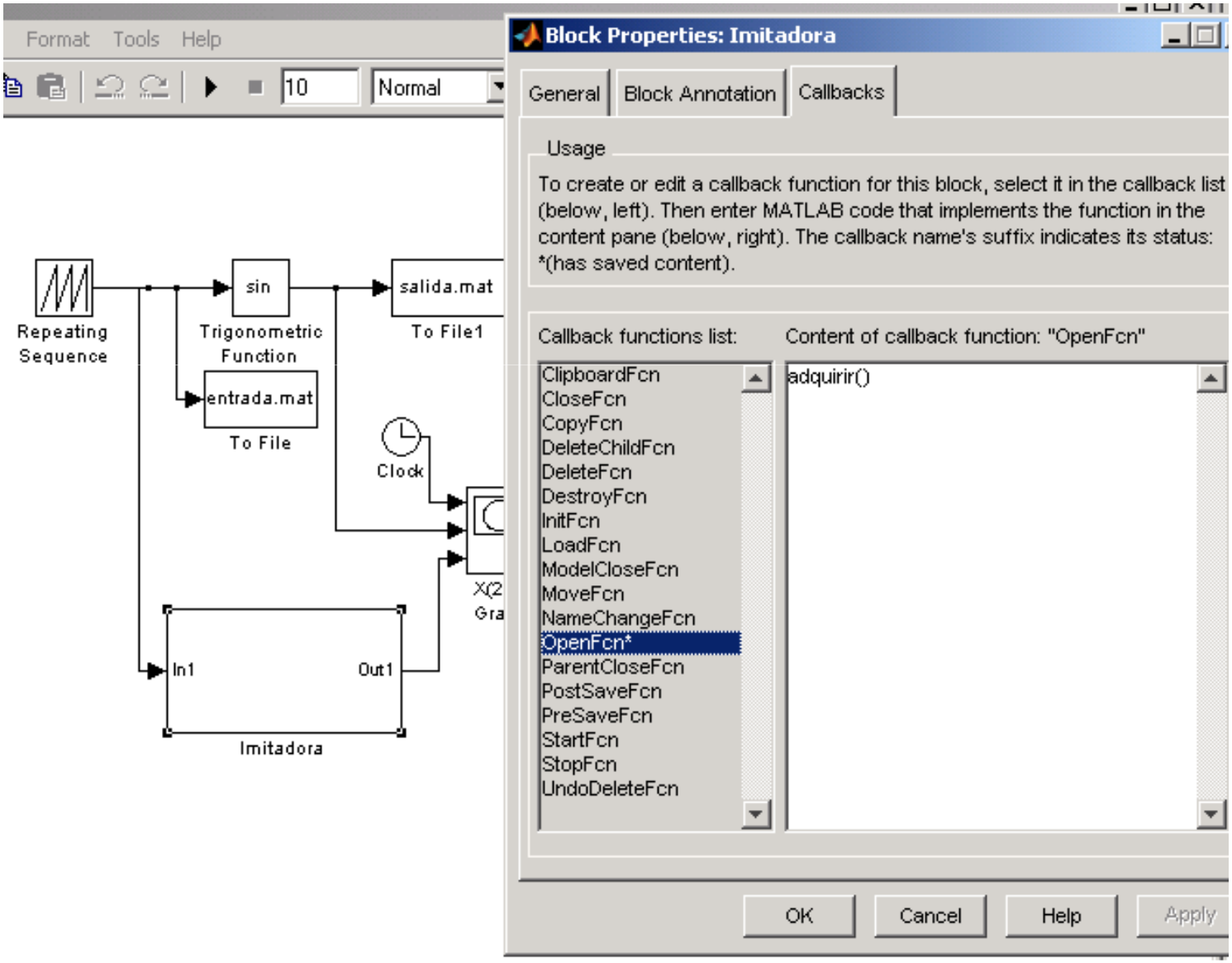
Stack: Base

```
97 -  net=newff(minmax(p),[nl,1],{'tansig','purelin'},'trainlm');%crea red neuronal
98 -  net.trainParam.show = sh;%periodo de parametros a mostrar en grafica de entrenamiento
99 -  net.trainParam.epochs = ep;%epocas de entrenamiento
100 -  net = init(net);%inicializa la red neuronal
101 -  [net,tr]=train(net,p,t);%entrena la red neuronal, con p como entrada y t como salida
102 -  a = sim(net,p)%simula la red entrenada en el command window
103 -  B1 = mat2str(net.b{1});%convierte en matriz un string
104 -  B2 = mat2str(net.b{2});
105 -  P1=mat2str(net.IW{1,1});
106 -  P2=mat2str(net.LW{2,1});
107 -  handles.bias1 =B1;%variable global
108 -  handles.bias2 =B2;
109 -  handles.pesos1=P1;
110 -  handles.pesos2=P2;
111 -  guidata(hObject,handles);%carga la anterior estructura
112 -  set(handles.edit1,'String',B1)%fija los datos en la ventana del guide
113 -  set(handles.edit2,'String',B2)
114 -  set(handles.edit3,'String',P1)
115 -  set(handles.edit4,'String',P2)
```

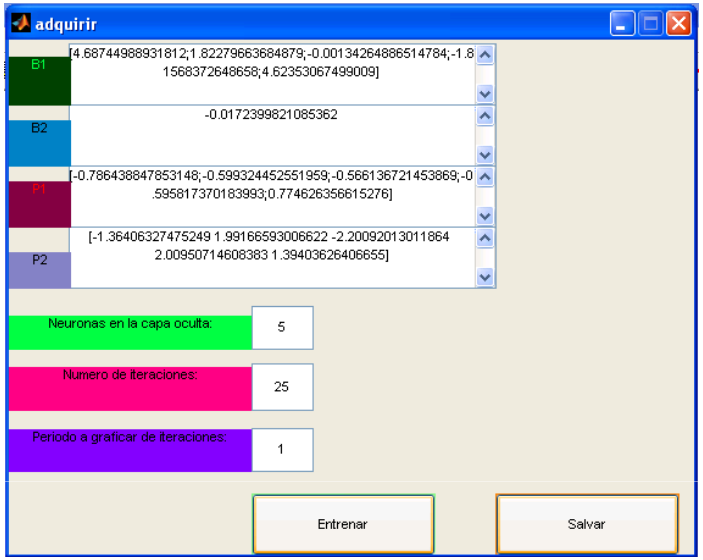
```
116     % --- Executes on button press in pushbutton2.
117     function pushbutton2_Callback(hObject, eventdata, handles)
118     % hObject    handle to pushbutton2 (see GCBO)
119     % eventdata  reserved - to be defined in a future version of MATLAB
120     % handles    structure with handles and user data (see GUIDATA)
121 -   model_open(handles)
122     % Get the new value for the
123 -   set(handles.edit1,'String',handles.bias1)%carga la variable global
124 -   set_param('Imitador/Imitadora','B1',get(handles.edit1,'String' ))%ubica los valores...
125                                                 %de la variable en la posicion indicada.
126 -   set(handles.edit2,'String',handles.bias2)
127 -   set_param('Imitador/Imitadora','B2',get(handles.edit2,'String' ))
128 -   set(handles.edit3,'String',handles.pesos1)
129 -   set_param('Imitador/Imitadora','P1',get(handles.edit3,'String' ))
130 -   set(handles.edit4,'String',handles.pesos2)
131 -   set_param('Imitador/Imitadora','P2',get(handles.edit4,'String' ))
132
```

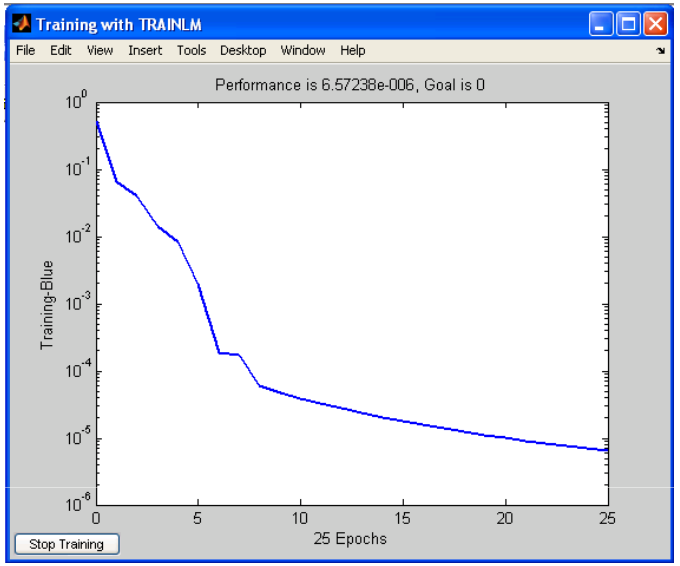

Block Parameters: Uniform Random Number
Minimum:
-6.2832
Maximum:
6.2832
Initial seed:
0
Sample time:
0.1
OK
Cancel
Help
Apply
sin
Trigonometric Function
salida.mat
To File1
entrada.mat
To File
Block Parameters: To File
To File
Write time and input to specified MAT file in row format. Time is in row 1.
Parameters
Filename:
entrada.mat
Variable name:
p
Decimation:
1
Sample time (-1 for inherited):
0.1
OK
Cancel
Help
Apply
Block Parameters: To File1
To File
Write time and input to specified MAT file in row format. Time is in row 1.
Parameters
Filename:
salida.mat
Variable name:
t
Decimation:
1
Sample time (-1 for inherited):
0.1
OK
Cancel
Help
Apply
Ready

Llamada de Guide



Entrenamiento

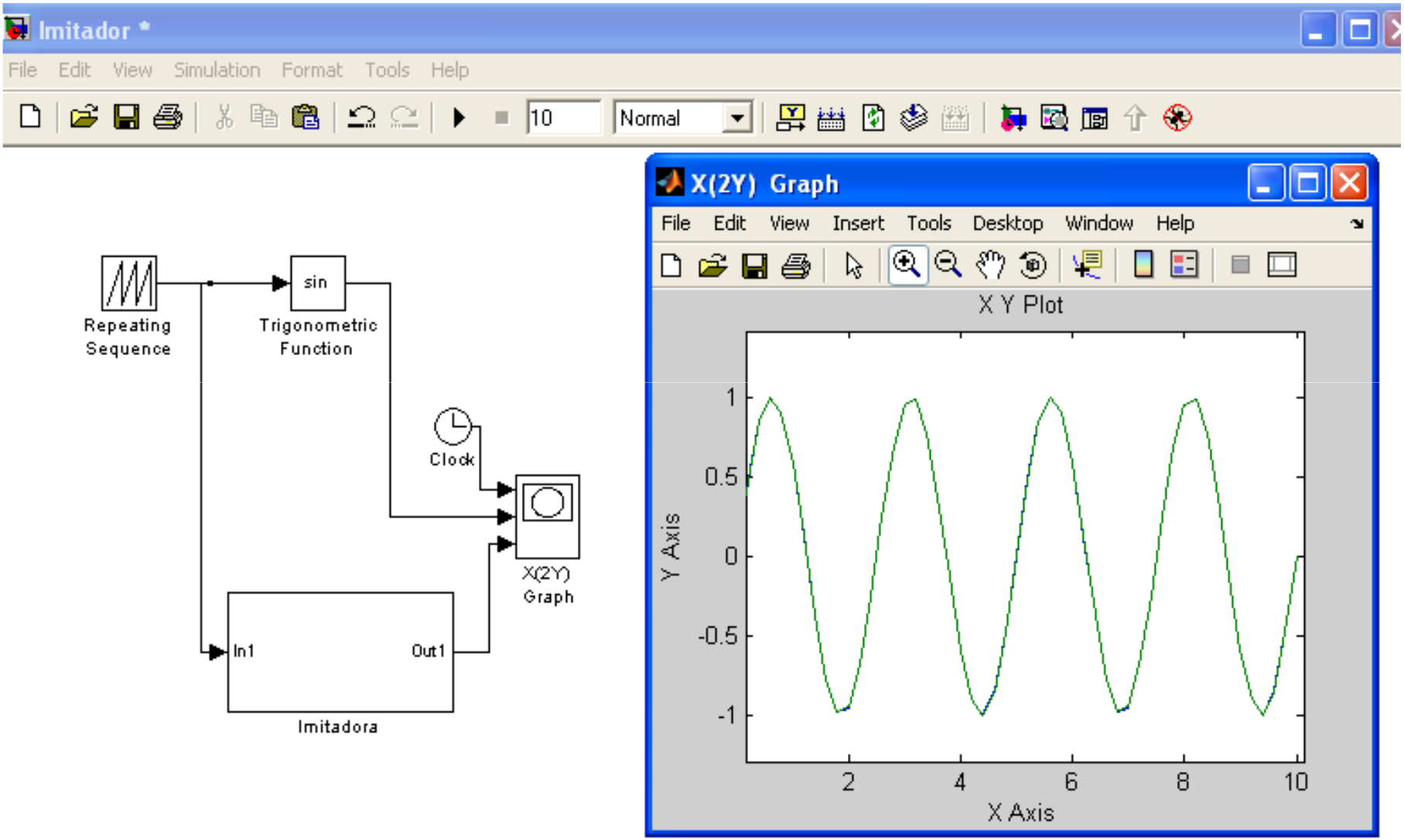





```
>> B1

B1 =

    4.8435
   -1.8564
   -0.0020
   -1.8455
    4.7493

>> B2

B2 =

   -0.0229
```

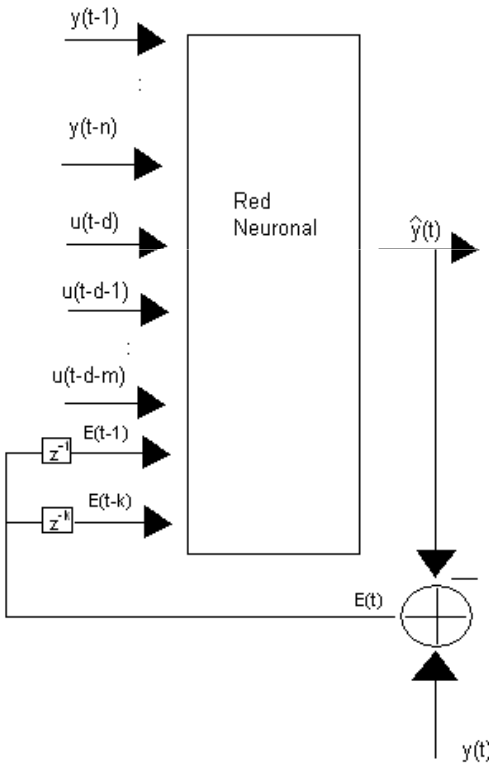
```
>> P1

P1 =

   -0.8144
    0.6108
   -0.5750
   -0.6058
    0.7970

>> P2

P2 =

   -1.3044   -1.9317   -2.1400    1.9554    1.3438
```

Imitación de la función.



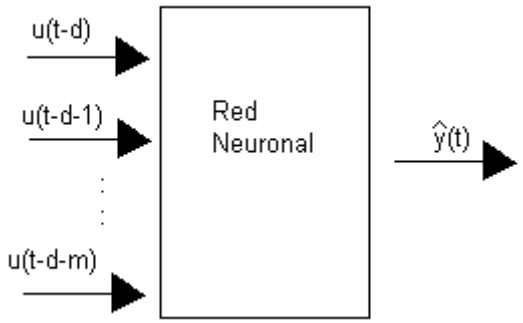
MODELOS DE APROXIMACION

$$\varphi(t) = [y(t-1)...y(t-n), u(t-d)...$$
$$u(t-d-m), E(t-1)...E(t-1-k)]^T$$



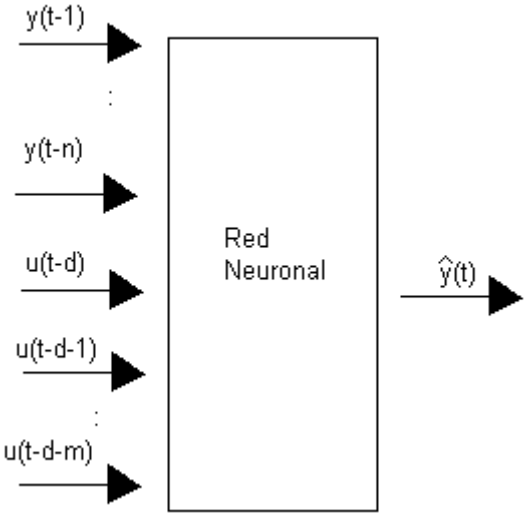
- *NNARMAX*

$$\hat{y}(t) = f(\varphi(t) * w_{S,R} + B_S)$$



•*NNFIR*

$$\varphi(t) = [u(t-d)...u(t-d-m)]^T$$

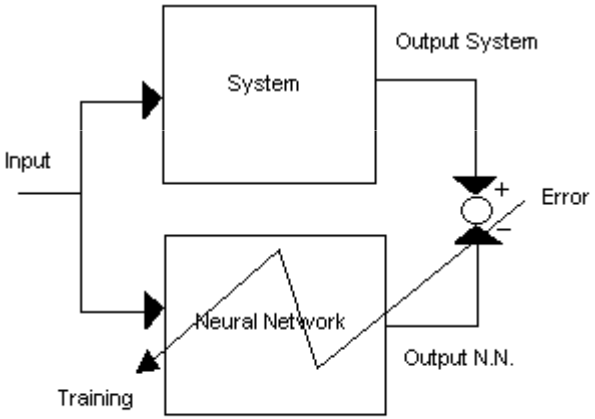


NNARX

$$\varphi(t) = [y(t-1)...y(t-n), u(t-d)...u(t-d-m)]^T$$

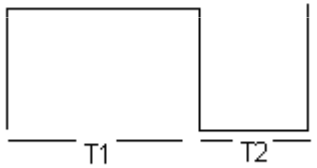
IDENTIFICACION DE SISTEMAS

- Identifica la dinámica de un sistema sin conocimiento previo del mismo.
- Identificación neuronal a partir de patrones de Entrada-Salida del sistema a identificar.
- Necesita vectores de regresión en las señales de entrada de la red neuronal identificadora.
- Redes multicapa para la identificación (perceptron 2 capas).



CONVERSOR DC-DC

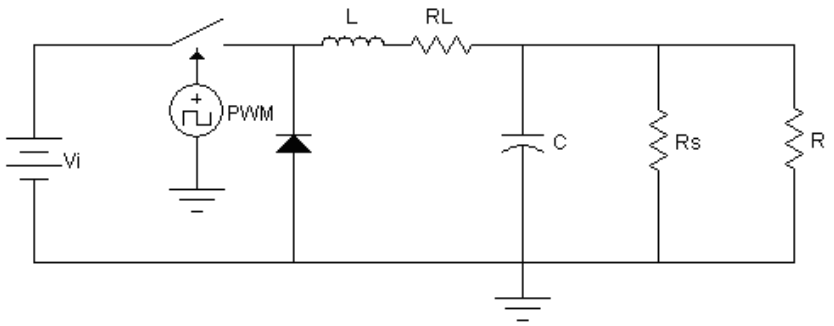
- Transforman una señal de tensión continua en otra señal continua.
- Trabajan bajo el principio de regulación conmutada.
- Principio el cual la tensión es administrada por una señal PWM.



- Donde: $\Delta = \frac{T_1}{T}$

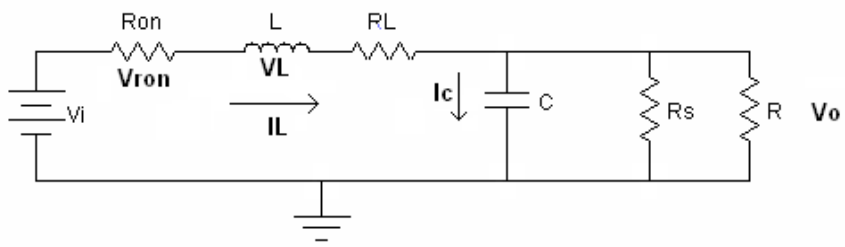
- Usados en la industria por su efectividad frente a perturbaciones eléctricas y bajo nivel de pérdidas

REDUCTOR DE TENSIÓN

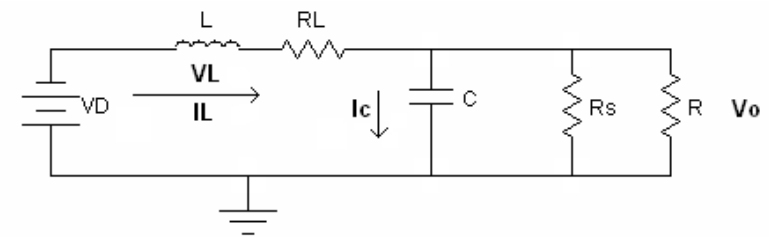


$$V_o = \frac{V_i * T_1 + 0 * T_2}{T_1 + T_2} = V_i * \frac{T_1}{T}$$

- T1



T2



- Promediando se obtiene:

$$L * \frac{di_L}{dt} = \delta * v_i - \delta * i_L * (R_L + R_{ron}) - v_0 - (1 - \delta) * v_D$$

$$\frac{dv_0}{dt} = i_L - \frac{v_0}{R_{ol}}$$

VALORES OBTENIDOS

De acuerdo con: $L_{\min} = V_o * \frac{1-\delta}{2 * I_s * f} = R_s * \frac{1-\delta}{2 * f}$ $C = \frac{L * I_1^{\,2}}{(V_{oMax}^2 - V_o^2)}$

se obtiene los valores de inductancia y capacitancia

Donde:

$L = 1 * 10^{-3} H.$

$C = 1 * 10^{-3} f.$

$R_{ON} = 0.5\Omega.$

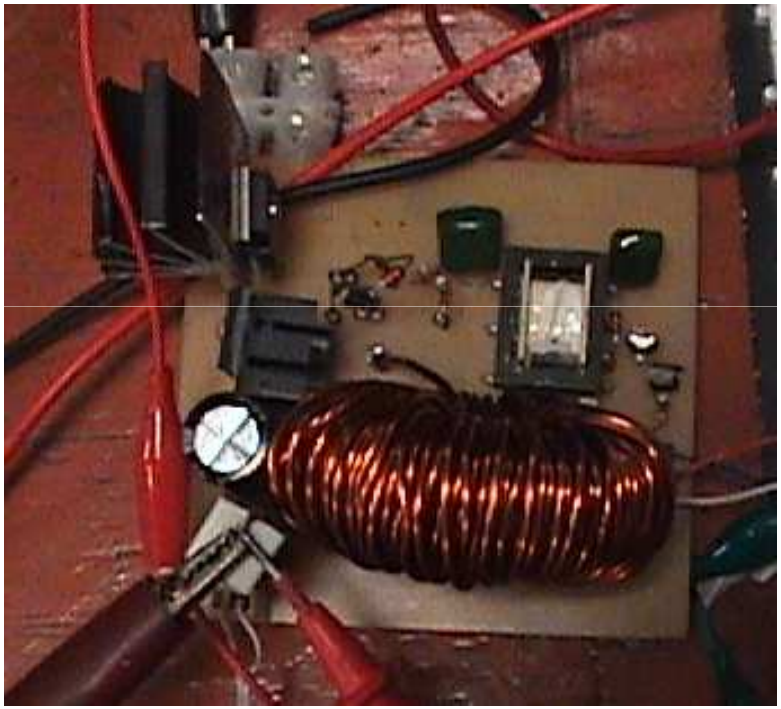
$V_i = 24V.$

$V_d = 0.7V.$

$R_s = 24\Omega.$

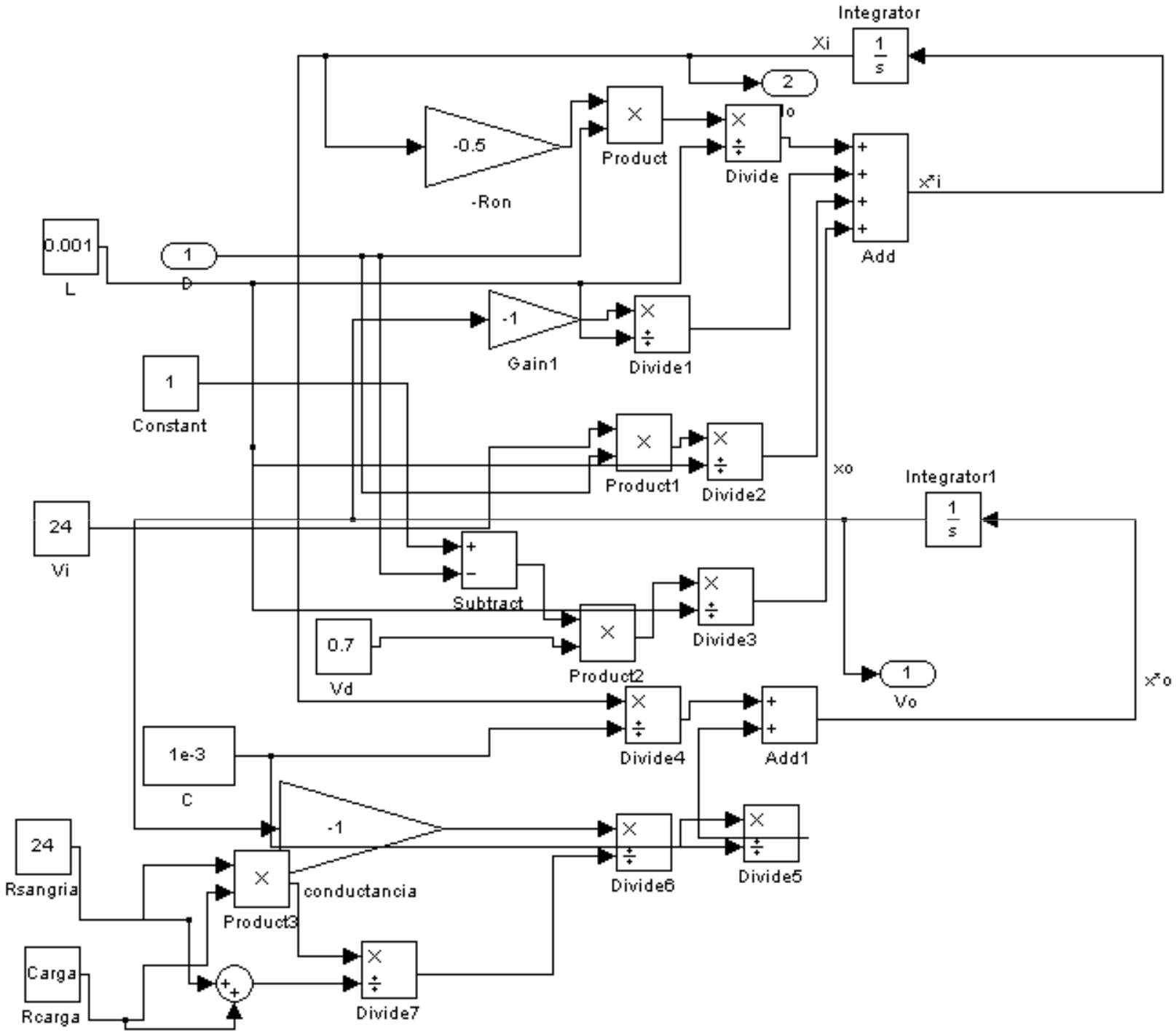
$V_o = 12V.$

$I_{l\_nom} = 8A.$

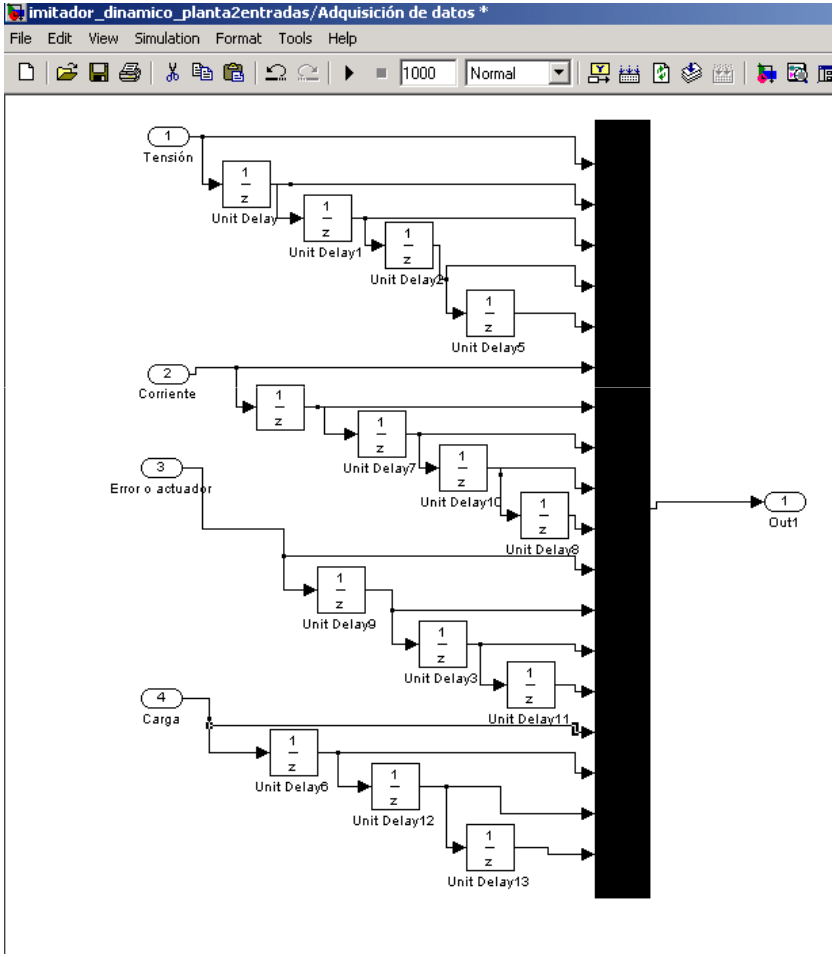


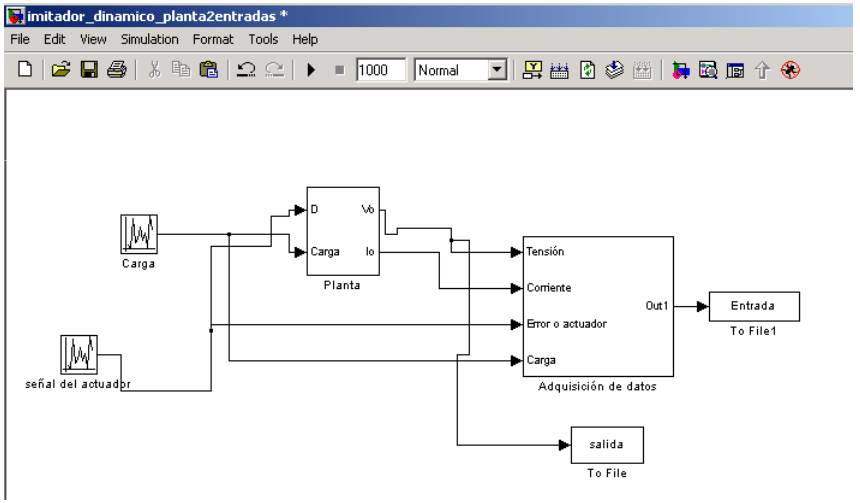
Dispositivo conmutación:FGA40N60UFD

Diodo: RURG3020C

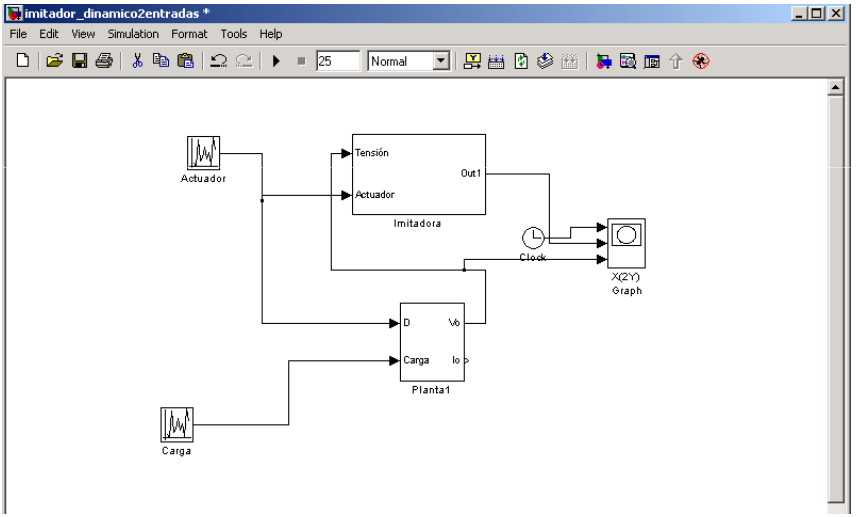

Integrator
Xi
1
s
2
-0.5
-Ron
Product
Divide
o
ẋi
Add
0.001
L
1
D
-1
Gain1
Divide1
1
Constant
Product1
Divide2
xo
Integrator1
24
Vi
Subtract
Divide3
0.7
Vd
Product2
1
Vo
ẋo
Divide4
Add1
1e-3
C
-1
24
Rsangria
Product3
conductancia
Divide6
Divide5
Carga
Rcarga
Divide7

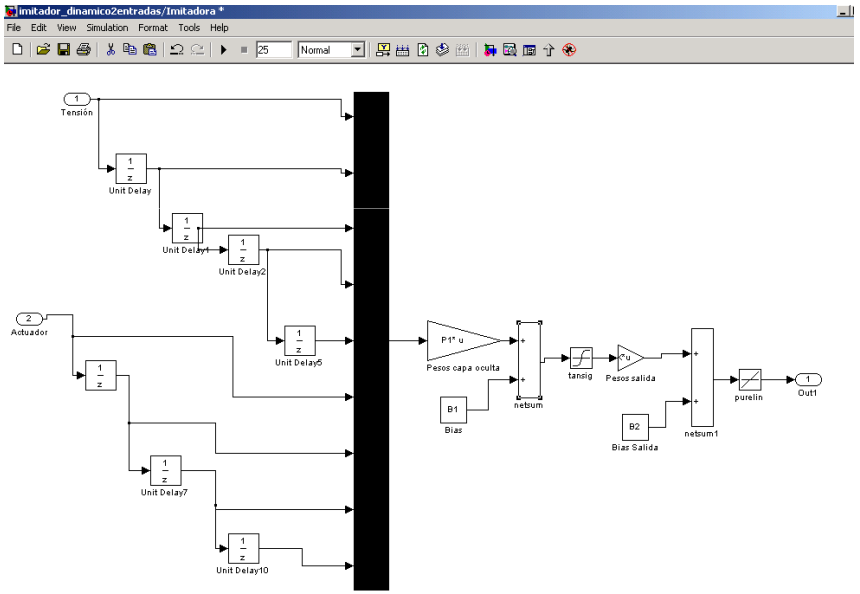
REDES NEURONALES DINAMICAS
ADQUISICION DE DATOS

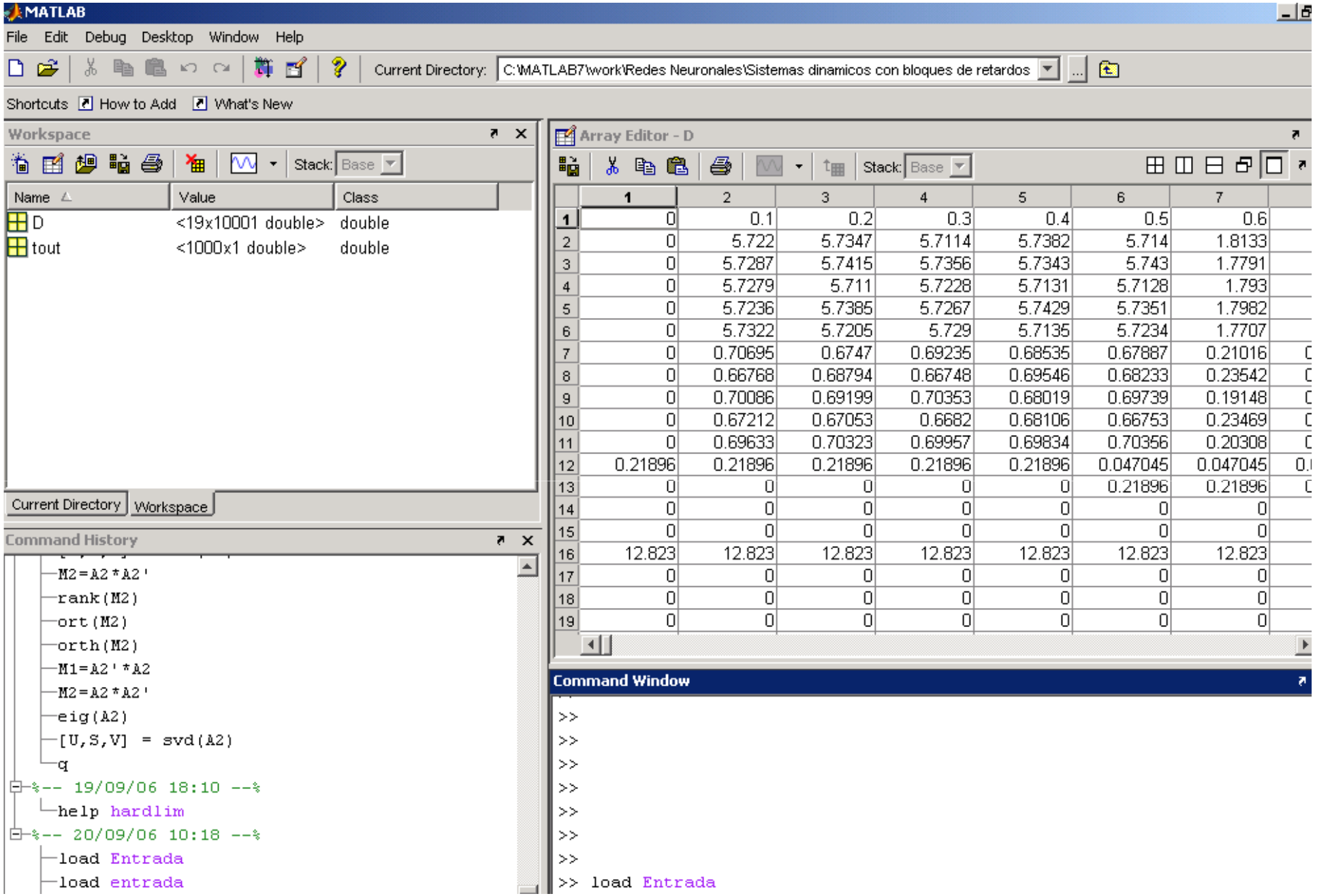




ENTRENAMIENTO Y SIMULACIÓN



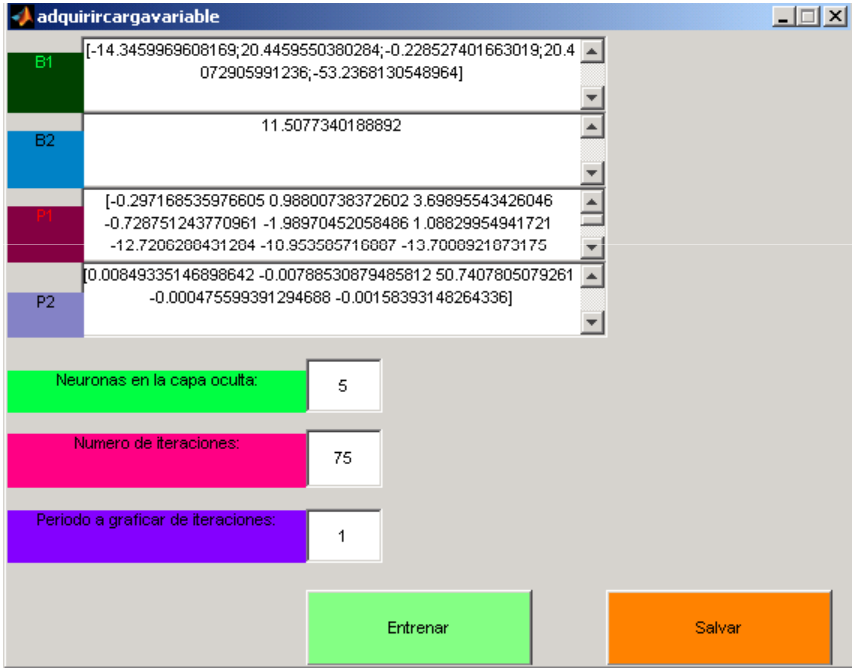


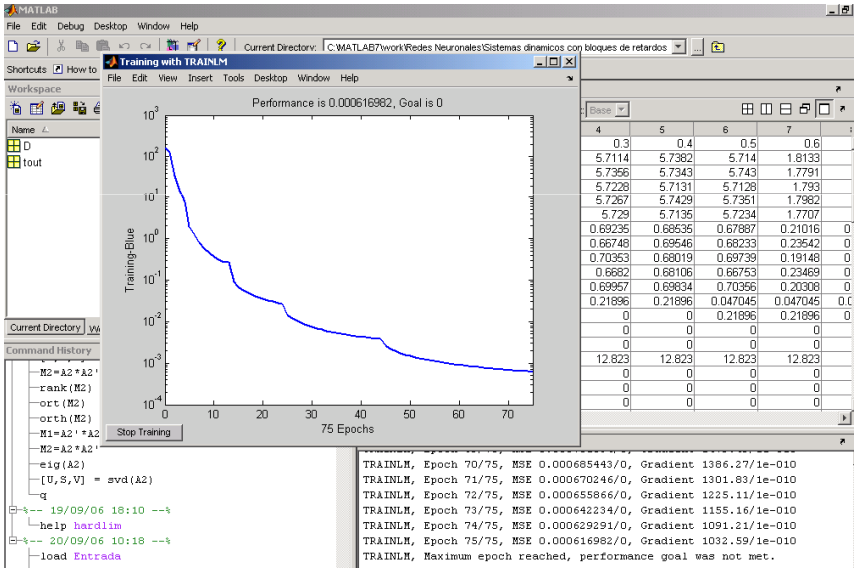

MATLAB
File Edit Debug Desktop Window Help
Current Directory: C:\MATLAB7\work\Redes Neuronales\Sistemas dinamicos con bloques de retardos
Shortcuts How to Add What's New
Workspace
Stack: Base
Name Value Class
D <19x10001 double> double
tout <1000x1 double> double
Current Directory Workspace
Command History
M2=A2*A2'
rank(M2)
ort(M2)
orth(M2)
M1=A2'*A2
M2=A2*A2'
eig(A2)
[U,S,V] = svd(A2)
q
%-- 19/09/06 18:10 --%
help hardlim
%-- 20/09/06 10:18 --%
load Entrada
load entrada
Array Editor - D
Stack: Base
1 2 3 4 5 6 7
1 0 0.1 0.2 0.3 0.4 0.5 0.6
2 0 5.722 5.7347 5.7114 5.7382 5.714 1.8133
3 0 5.7287 5.7415 5.7356 5.7343 5.743 1.7791
4 0 5.7279 5.711 5.7228 5.7131 5.7128 1.793
5 0 5.7236 5.7385 5.7267 5.7429 5.7351 1.7982
6 0 5.7322 5.7205 5.729 5.7135 5.7234 1.7707
7 0 0.70695 0.6747 0.69235 0.68535 0.67887 0.21016
8 0 0.66768 0.68794 0.66748 0.69546 0.68233 0.23542
9 0 0.70086 0.69199 0.70353 0.68019 0.69739 0.19148
10 0 0.67212 0.67053 0.6682 0.68106 0.66753 0.23469
11 0 0.69633 0.70323 0.69957 0.69834 0.70356 0.20308
12 0.21896 0.21896 0.21896 0.21896 0.21896 0.047045 0.047045
13 0 0 0 0 0 0.21896 0.21896
14 0 0 0 0 0 0 0
15 0 0 0 0 0 0 0
16 12.823 12.823 12.823 12.823 12.823 12.823 12.823
17 0 0 0 0 0 0 0
18 0 0 0 0 0 0 0
19 0 0 0 0 0 0 0
Command Window
>>
>>
>>
>>
>>
>>
>>
>> load Entrada

```
File  Edit  Text  Cell  Tools  Debug  Desktop  Window  Help
```

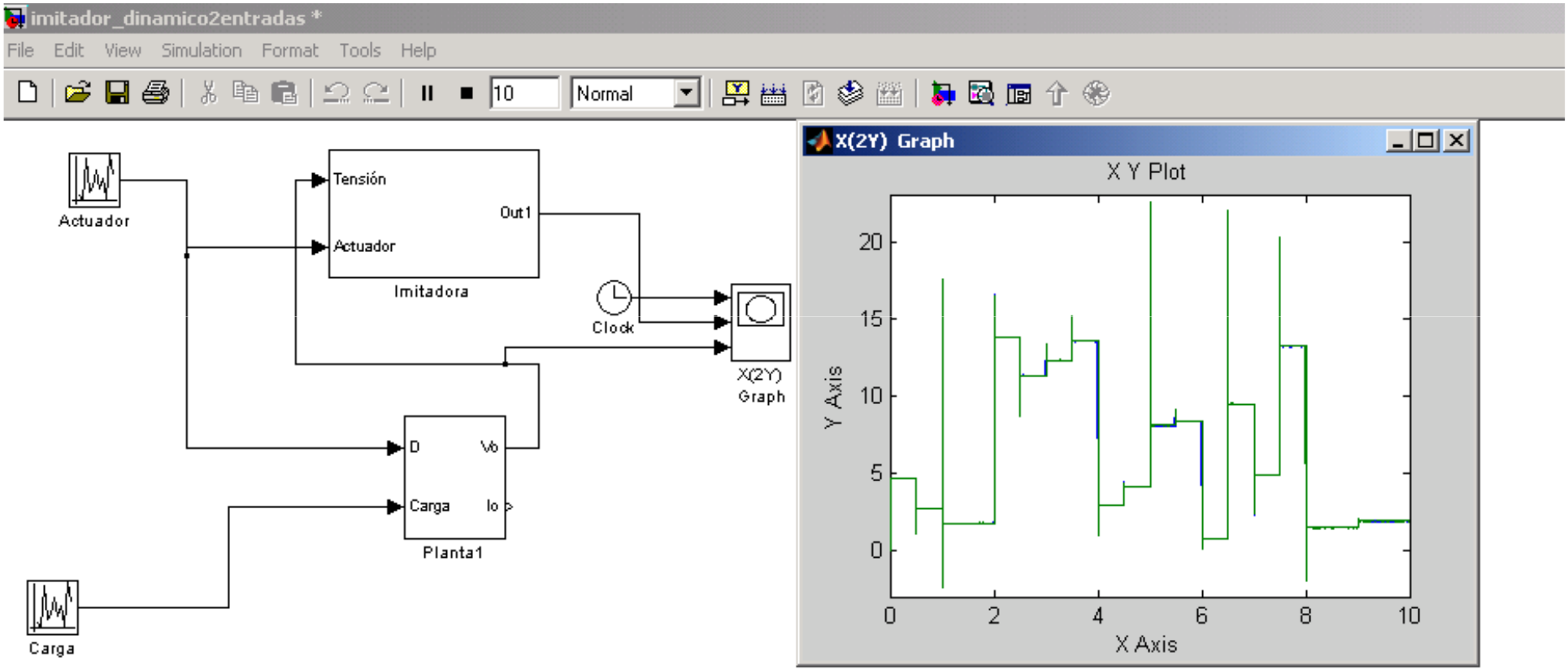
```
73
74    % Get default command line output from handles structure
75 -  varargout{1} = handles.output;
76    function model_open(handles)
77    % Make sure the diagram is still open
78 -  if  isempty(find_system('Name','imitador_dinamico2entradas')),
79 -      open_system('imitador_dinamico2entradas');  %Carga Simulink de adquisición de datos
80 -  end
81
82 -  if  isempty(find_system('Name','imitador_dinamico_planta2entradas')),
83 -      open_system('imitador_dinamico_planta2entradas');   %Carga Simulink de entrenamiento y sumulación
84 -  end
85    % --- Executes on button press in pushbutton1.
86    function pushbutton1_Callback(hObject, eventdata, handles)
87    % hObject    handle to pushbutton1 (see GCBO)
88    % eventdata  reserved - to be defined in a future version of MATLAB
89    % handles    structure with handles and user data (see GUIDATA)
90 -  model_open(handles)
91 -  NewStrVal = get(hObject, 'String');
92 -  NewVal = str2double(NewStrVal);
93 -  nl = str2double(get(handles.edit5,'String'));
94 -  ep = str2double(get(handles.edit6,'String'));
95 -  sh = str2double(get(handles.edit7,'String'));
96 -  load Entrada
97 -  load salida
98 -  D=[D(2,:);D(3,:);D(4,:); D(5,:);D(6,:);D(12,:);D(13,:);D(14,:);D(15,:);];%Entrada tensión y actuador
99                                                                              %con sus respectivos retardos
100 -  t=p(2,:);%Salida con señal de tensión
101 -  net=newff(minmax(D),[nl,1],{'tansig','purelin'},'trainlm');
102 -  net.trainParam.show = sh;
103 -  net.trainParam.epochs = ep;
104 -  net = init(net);
105 -  [net,tr]=train(net,D,t);
106 -  B1 = mat2str(net.b{1});
```

EVOLUCIÓN ENTRENAMIENTO

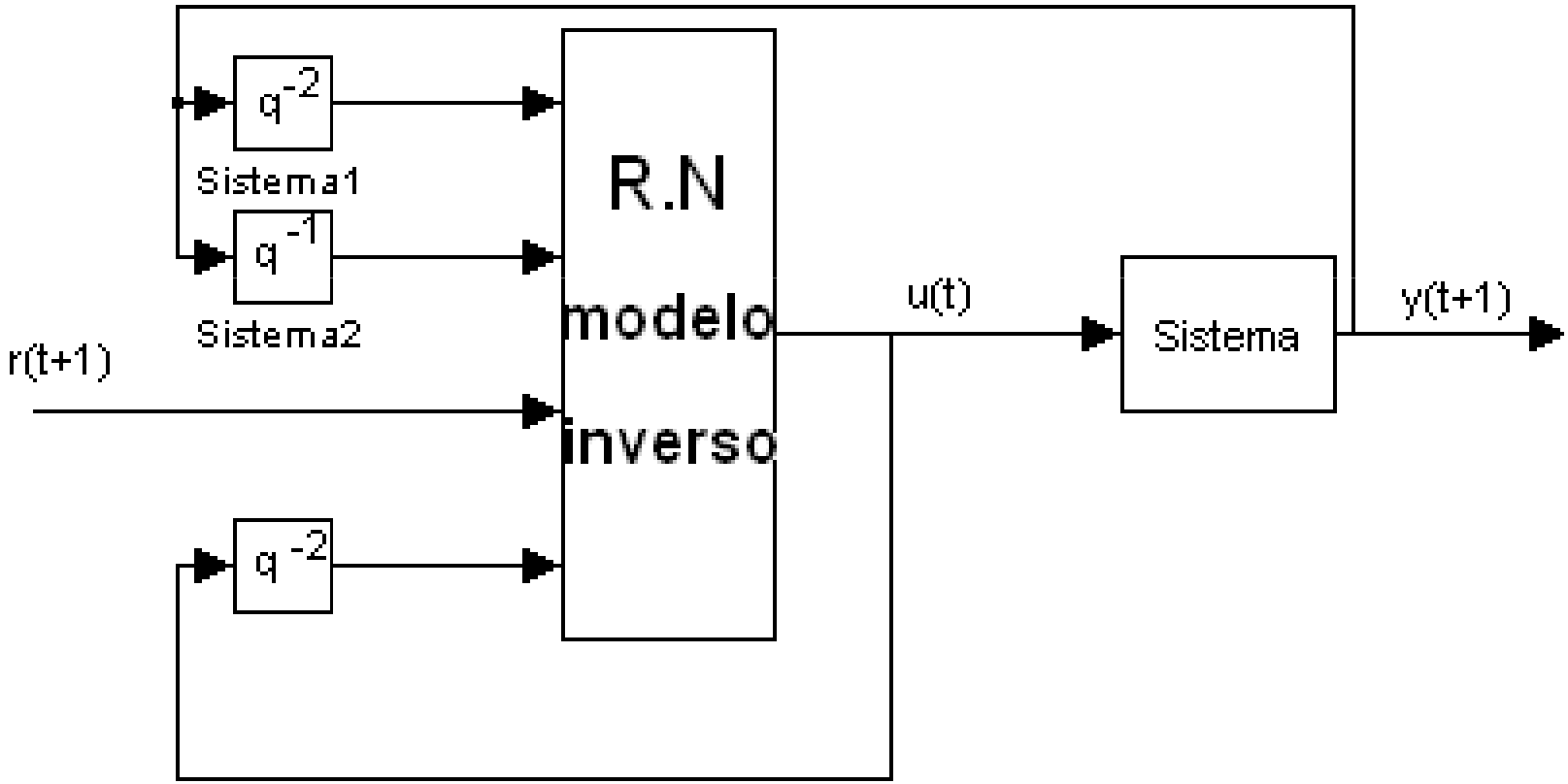


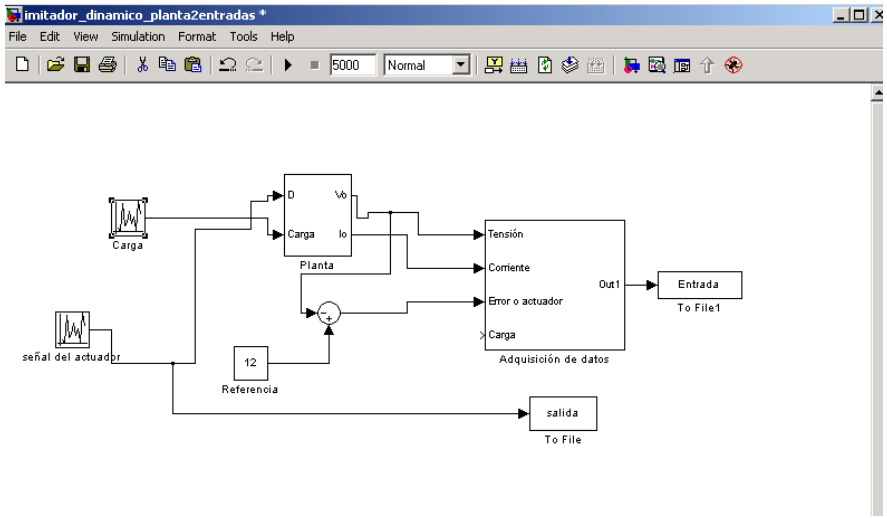


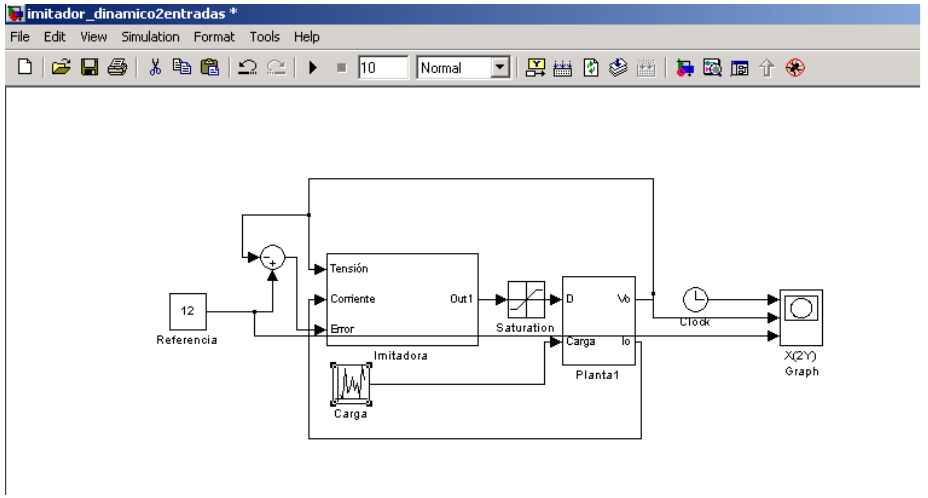
RESULTADOS

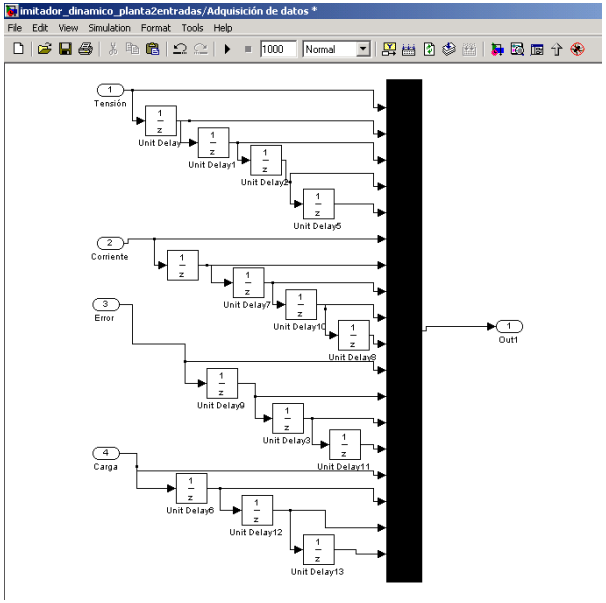

imitador_dinamico2entradas *
File Edit View Simulation Format Tools Help
10
Normal
Actuador
Tensión
Actuador
Out1
Imitadora
Clock
X(2Y) Graph
D
Vo
Carga
Io
Planta1
Carga
X(2Y) Graph
X Y Plot
Y Axis
X Axis
0
5
10
15
20
0
2
4
6
8
10

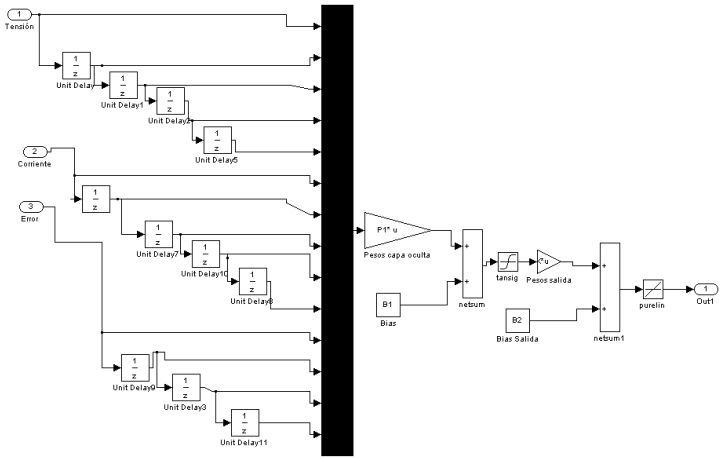
CONTROL INVERSO GENERALIZADO




imitador_dinamico_planta2entradas *
File Edit View Simulation Format Tools Help
5000
Normal
Carga
D
Vo
Carga
Io
Planta
Tensión
Corriente
Error o actuador
Carga
Out1
Adquisición de datos
Entrada
To File1
señal del actuador
12
Referencia
salida
To File


imitador_dinamico2entradas *
File Edit View Simulation Format Tools Help
10
Normal
12
Referencia
Tensión
Corriente
Error
Out1
Imitadora
Carga
Saturation
D
Vo
Carga
Io
Planta1
Clock
X(2Y)
Graph


imitador_dinamico_planta2entradas/Adquisición de datos *
File Edit View Simulation Format Tools Help
1000
Normal
1
Tensión
2
Corriente
3
Error
4
Carga
Unit Delay
Unit Delay1
Unit Delay2
Unit Delay5
Unit Delay7
Unit Delay10
Unit Delay8
Unit Delay9
Unit Delay3
Unit Delay11
Unit Delay6
Unit Delay12
Unit Delay13
1
Out1

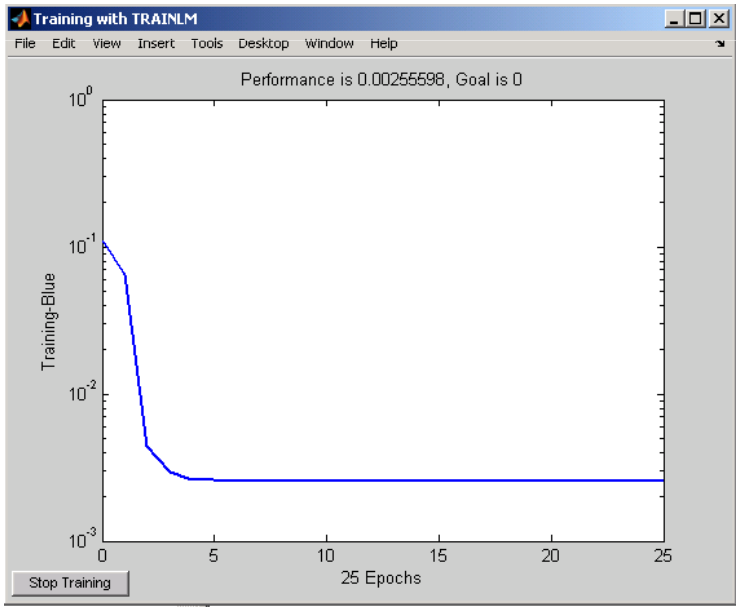

1
Tensión
2
Corriente
3
Error
Unit Delay
Unit Delay1
Unit Delay2
Unit Delay5
Unit Delay7
Unit Delay10
Unit Delay8
Unit Delay9
Unit Delay3
Unit Delay11
P1* u
Pesos capa oculta
B1
Bias
netsum
tansig
Pesos salida
B2
Bias Salida
netsum1
purelin
1
Out1

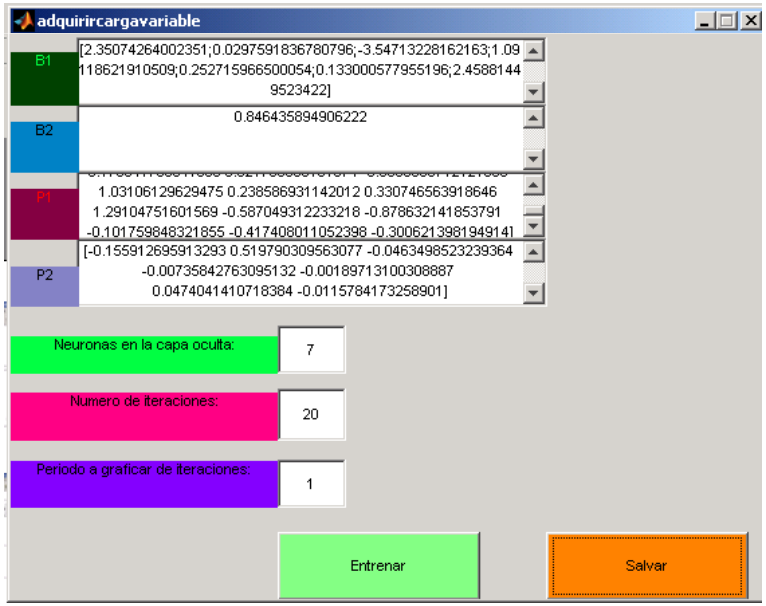
Array Editor - D

	1	2	3	4	5	6	7
1	0	0.1	0.2	0.3	0.4	0.5	0.6
2	0	11.246	11.231	11.237	11.253	11.266	10.112
3	0	11.258	11.248	11.243	11.236	11.252	10.09
4	0	11.234	11.249	11.254	11.26	11.244	10.123
5	0	11.266	11.247	11.242	11.236	11.253	10.093
6	0	11.231	11.249	11.255	11.26	11.243	10.12
7	0	1.2875	1.2955	1.3101	1.3052	1.2987	1.1826
8	0	1.3098	1.2765	1.2776	1.2999	1.3199	1.1541
9	0	1.2883	1.3206	1.3196	1.2971	1.2774	1.1785
10	0	1.3074	1.2766	1.2777	1.3003	1.3197	1.1562
11	0	1.2897	1.3205	1.3194	1.2968	1.2775	1.1766
12	12	0.75366	0.76889	0.763	0.7466	0.73444	1.8882
13	0	0.7418	0.7521	0.75727	0.76415	0.74767	1.9102
14	0	0.7661	0.75132	0.74607	0.73955	0.75634	1.8769
15	0	0.73377	0.75264	0.75798	0.76412	0.74703	1.9066
16	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0

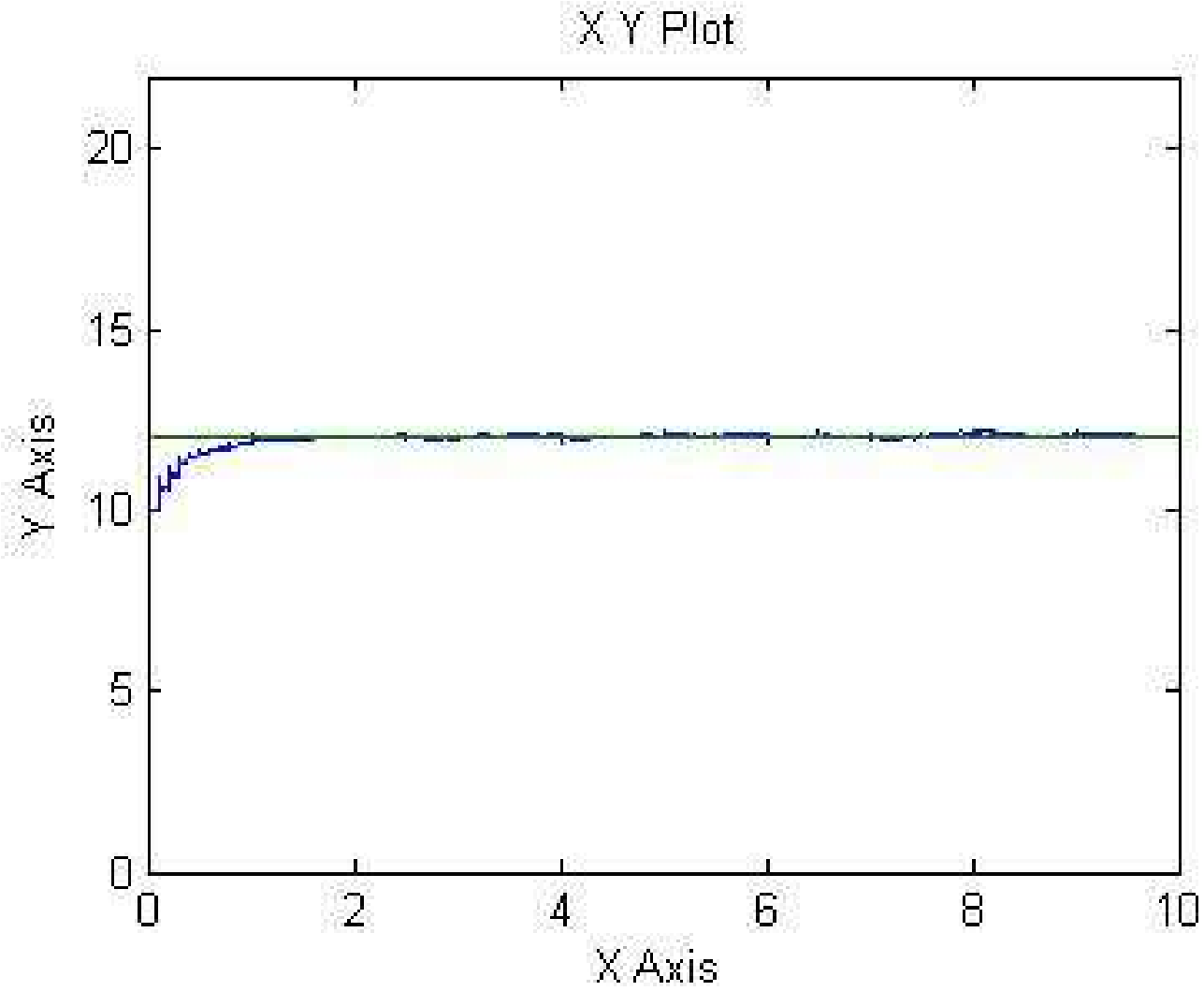
Editor - C:\MATLAB7\work\Redes Neuronales\Sistemas dinamicos con bloques de retardos\adquirircargavariable.m

```
96 -  load Entrada
97 -  load salida
98 -  D=[D(2,:);D(3,:);D(4,:);D(5,:);D(6,:);D(7,:);D(8,:);D(9,:);D(10,:);D(11,:);D(12,:);D(13,:);D(14,:);D(15,:);];
99 -  t=p(2,:);%Salida con señal de tensión
100 - net=newff(minmax(D),[n1,1],('tansig','purelin'),'trainlm');
```

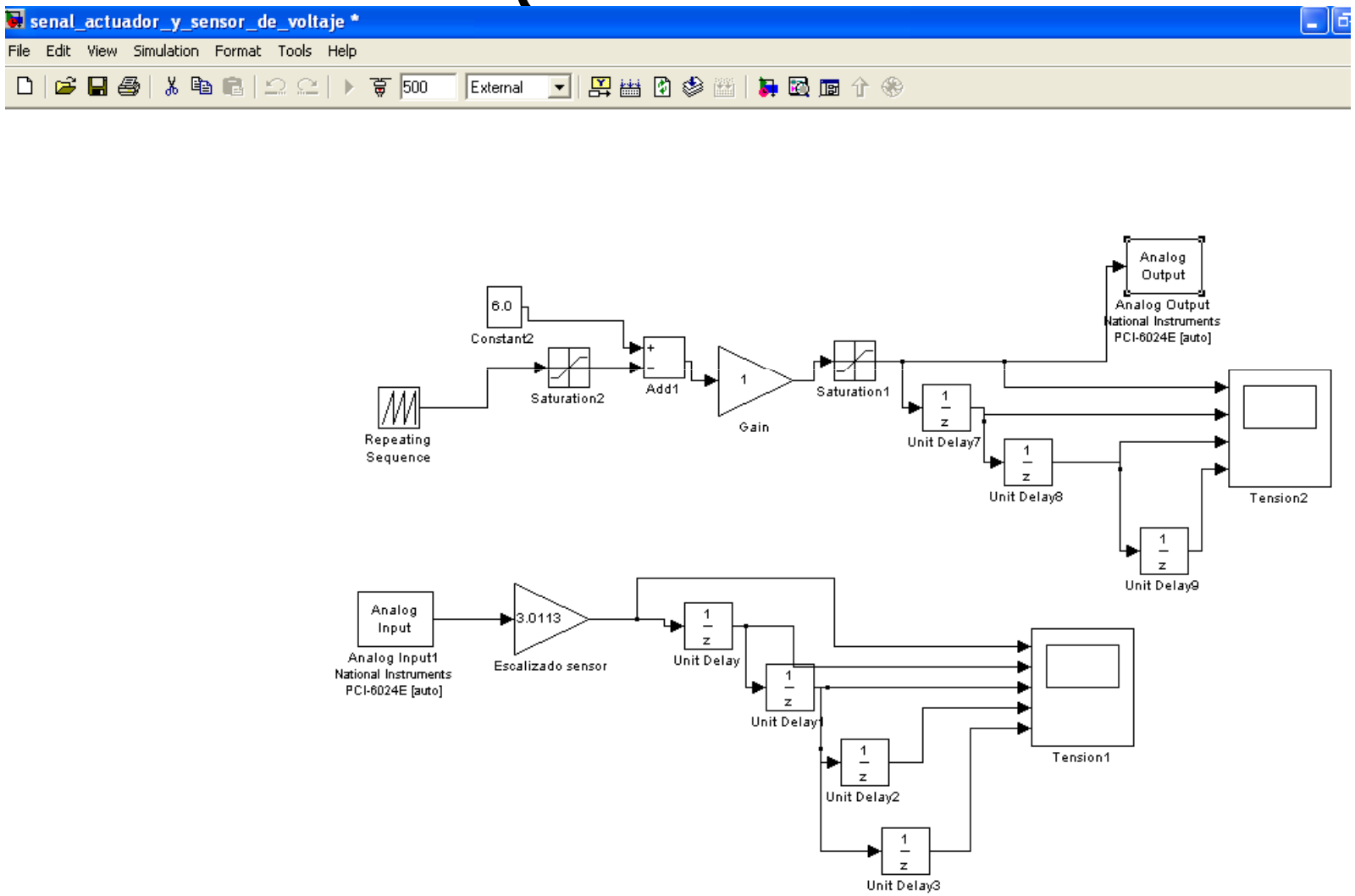




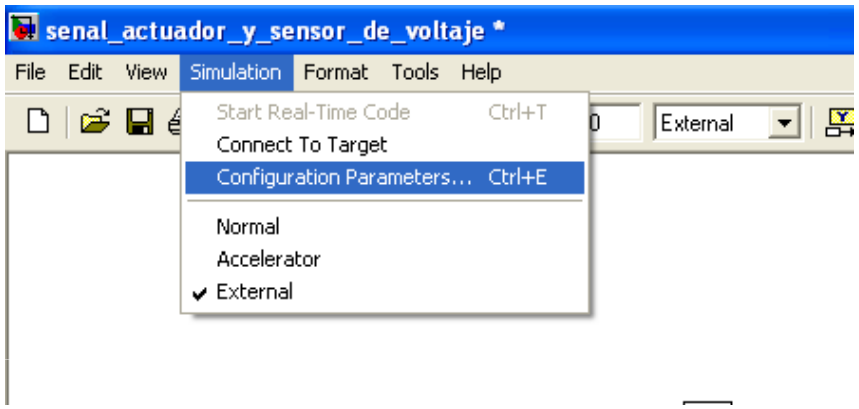
RESULTADO

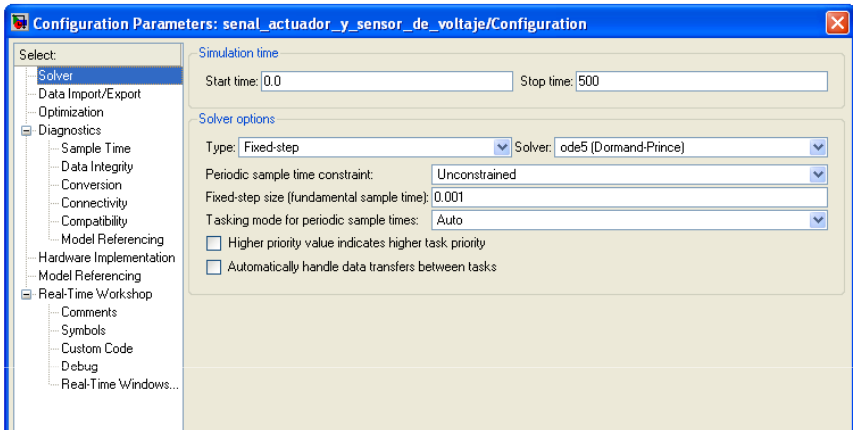


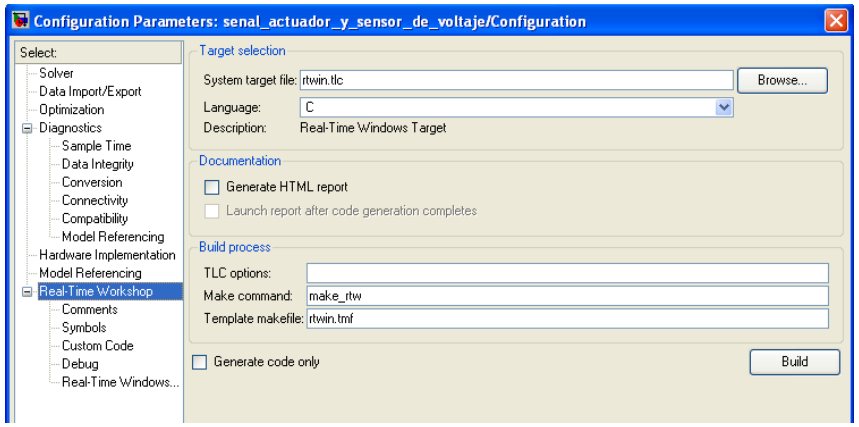
DAQ 6024-E Y REAL TIME ADQUISICIÓN

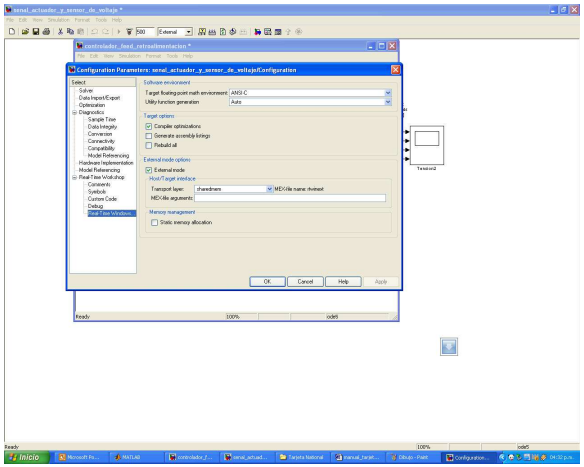


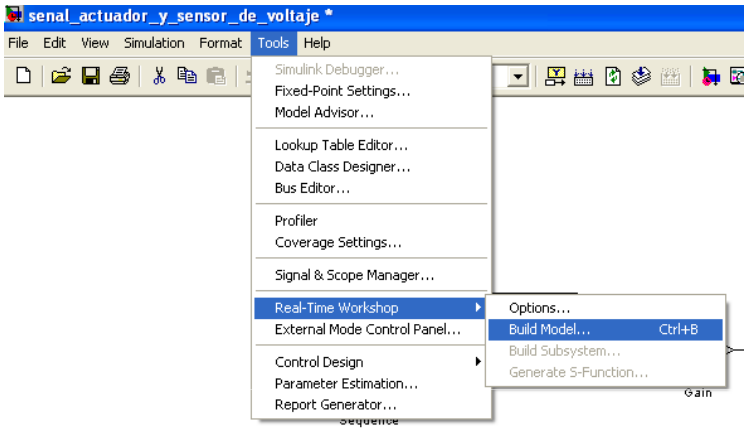
CONFIGURACIÓN REAL TIME

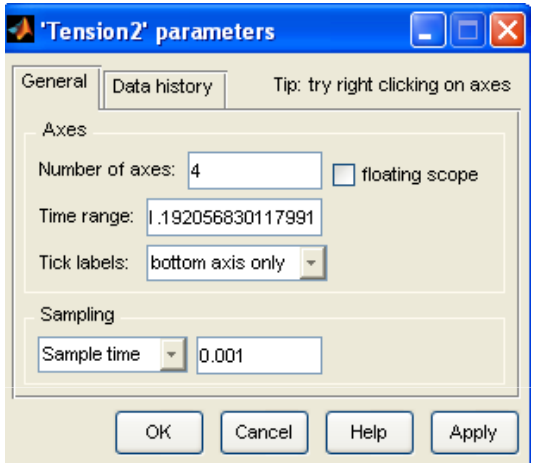


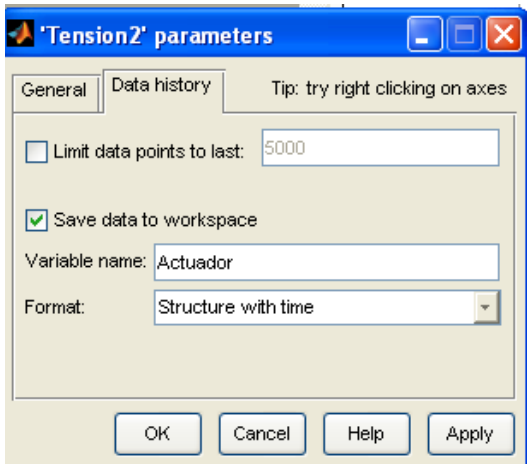


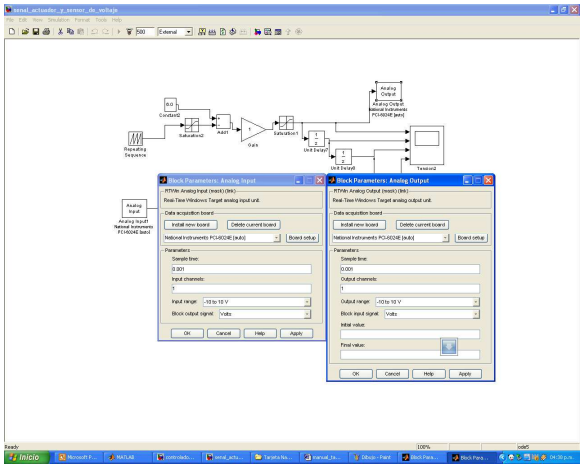




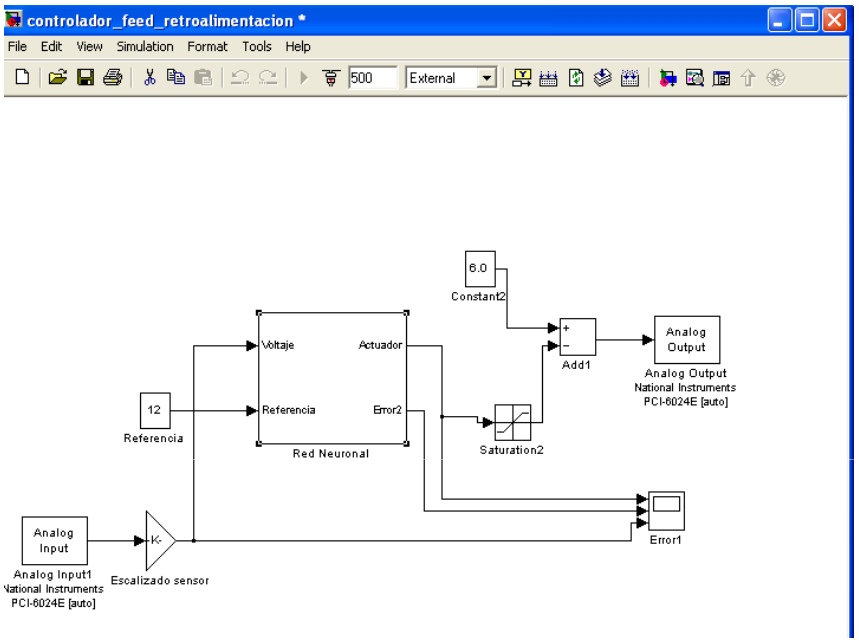

senal_actuador_y_sensor_de_voltaje *
File Edit View Simulation Format Tools Help
Simulink Debugger...
Fixed-Point Settings...
Model Advisor...
Lookup Table Editor...
Data Class Designer...
Bus Editor...
Profiler
Coverage Settings...
Signal & Scope Manager...
Real-Time Workshop
External Mode Control Panel...
Control Design
Parameter Estimation...
Report Generator...
Options...
Build Model... Ctrl+B
Build Subsystem...
Generate S-Function...
Gain

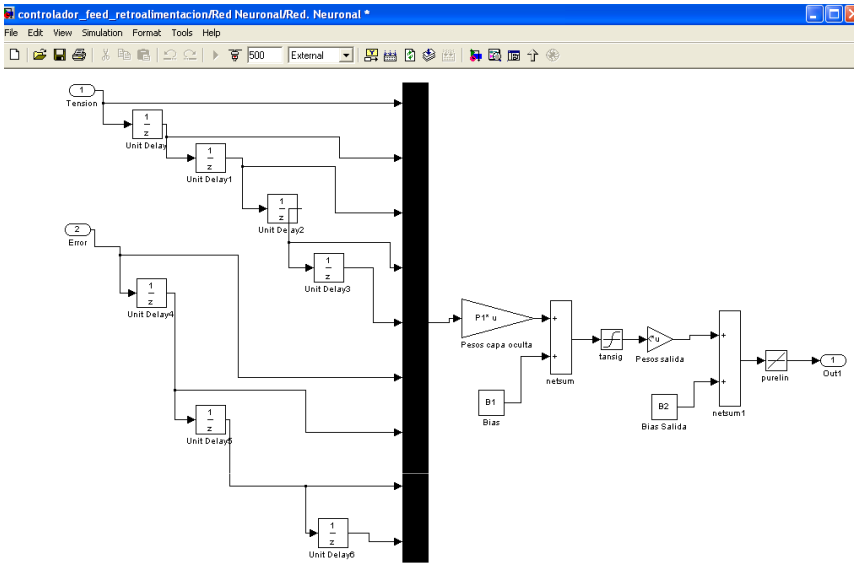

'Tension2' parameters
General
Data history
Tip: try right clicking on axes
Axes
Number of axes: 4
floating scope
Time range: 1.192056830117991
Tick labels: bottom axis only
Sampling
Sample time
0.001
OK
Cancel
Help
Apply


'Tension2' parameters
General
Data history
Tip: try right clicking on axes
Limit data points to last: 5000
Save data to workspace
Variable name: Actuador
Format: Structure with time
OK
Cancel
Help
Apply



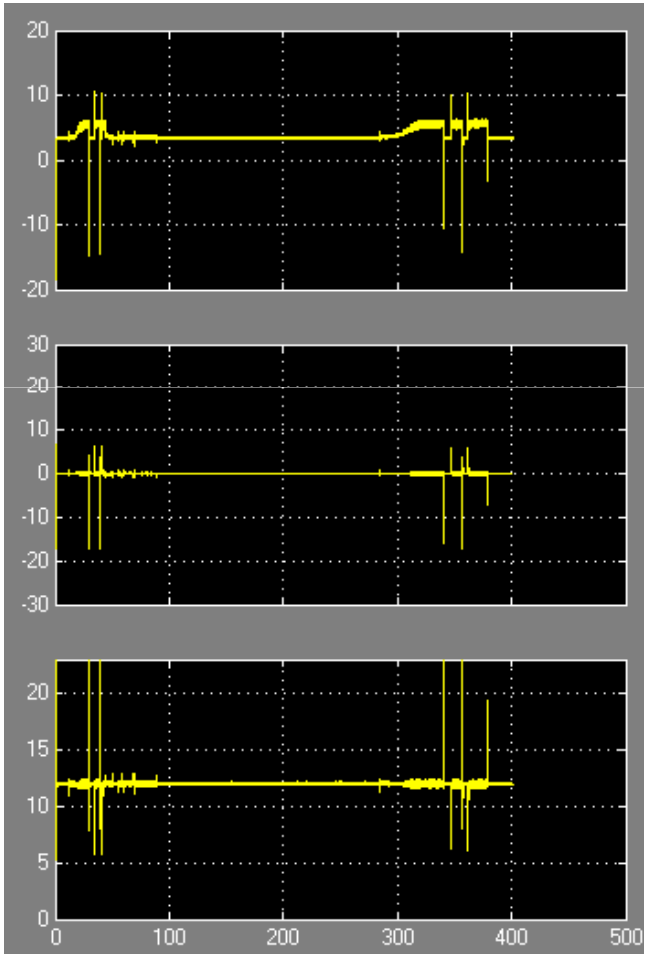
CONTROLADOR





RESULTADOS

- REFERENCIA 12v.
- Carga variable a través de reóstato.
- Corriente entre 0.28A hasta 8.2 A.



PREGUNTAS