ANEXO G

CÓDIGOS DE LOS PROGRAMAS DE MATLAB

```
function [Kp,tao,to]=ajuste_fit3(c,t,dm)
% Ajuste de una curva por el mètodo del FIT3
% A la función se le pasan los valores del tiempo, la salida y el cambio en
% el set point

span_C=100;            %Rango de la variable de salida proceso
span_m=100;            %Rango de la variable de entrada proceso(salida controlador)
dC=c(length(c))-c(1);   % Para calcular el cambio total de la señal se resta
                        % el primer valor con el ùltimo.

Ct1=c(1)+dC*0.283;      % Se calcula el valor de la señal cuando a llegado
Ct2=c(1)+dC*0.632;      % al 28.3%  y  63.2% de su de su cambio total.

it1=min(find(c==Ct1));  % se halla el indice en el cual se encuentra el
it2=min(find(c==Ct2));  % t1 y t2.

% En caso de que no se encuentre un valor igual en la tabla se procede a
% realizar una interpolacion lineal para hallar t1 y t2. Verifica si el
% camio en la variable fua ascendente o descendente de modo que se
% modifiquen las desigualdades para encontrar el tiempo anterior al tiempo
% no encontrado t1 o t2.


if(it1)
   t1=t(it1);
else
   if dC < 0
      itl=max(find(c>Ct1));
      ith=itl+1;
   else
      itl=max(find(c<Ct1));
      ith=itl+1;
   end
   t1=((Ct1-c(itl))*t(ith)+(c(ith)-Ct1)*t(itl))/(c(ith)-c(itl));
end

clear itl ith  %se limpian las variables para su reuso

if(it2)
```

```
t2=t(it2);
```

```
else
   if dC < 0
      itl=max(find(c>Ct2));
      ith=itl+1;
   else
      itl=max(find(c<Ct2));
      ith=itl+1;
   end

   t2=((Ct2-c(itl))*t(ith)+(c(ith)-Ct2)*t(itl))/(c(ith)-c(itl));;
end

%Se halla la constante de tiempo del sistema
tao=1.5*(t2-t1);

%Se halla el tiempo muerto del sistema
to=t2-tao;

%Se halla la ganancia del sistema en %T0/%CO
Kp=(dC/span_C)/(dm/span_m);
```

```
function [child,bestmember]=cruzamiento4(trait,fitness,CROSS_PROB,POP_SIZE,...
   NUM_TRAITS,ELITISM,bestmember,HIGHTRAIT,LOWTRAIT)
% Esta función realiza el crossover heurístico de la siguiente manera:
% hijo=random*(padre1 -padre2)+padre1, teniendo en cuenta que padre1 es
% mejor que dos y de que además es para real coded es decir tal y como son.

for pop_member=1:POP_SIZE
   if ELITISM==1 && pop_member==bestmember
      child(:,pop_member)=trait(:,pop_member);
      bestmember=pop_member;
   else
      if CROSS_PROB > rand
         for num_trait=1:NUM_TRAITS
            parent1=trait(num_trait,pop_member);
            pop_member2=ceil(rand*POP_SIZE);
            % se verifica que no sea el mismo miembro de la poblacion y
            % además que su fitness no sea menor que el fitness actual
               while pop_member2==pop_member && ...
                     fitness(pop_member2) < fitness(pop_member)
                  pop_member2=ceil(rand*POP_SIZE);
               end
            parent2=trait(num_trait,pop_member2);
            % se realiza el cruzamiento para real coded de acuerdo a la
            % regla.
            child(num_trait,pop_member)=rand*(parent1-parent2)+parent1;
            % se verifica si se paso del límite par no generar miembros de
            % la población no factibles.
            if child(num_trait,pop_member) > HIGHTRAIT(num_trait)
               child(num_trait,pop_member)=HIGHTRAIT(num_trait);
            elseif child(num_trait,pop_member) < LOWTRAIT(num_trait)
               child(num_trait,pop_member)=LOWTRAIT(num_trait);
            end
         end
      else
         child(:,pop_member)=trait(:,pop_member);
      end
   end
end
```

```
function [child,bestmember]=cruzamiento5(trait,CROSS_PROB,POP_SIZE,...
   NUM_TRAITS,ELITISM,bestmember,HIGHTRAIT,LOWTRAIT)
% Esta función realiza el cruzamiento atirmetico de la siguiente forma
% hijo=a*padre2*(1-a)*padre1, siendo "a" una constante o un número aleatorio
% que para nuestro caso fue un número aleatorio por haberse demostrado que
% tiene mejores resultados.

for pop_member=1:POP_SIZE
   if ELITISM==1 && pop_member==bestmember
      child(:,pop_member)=trait(:,pop_member);
      bestmember=pop_member;
   else
      if CROSS_PROB > rand
         for num_trait=1:NUM_TRAITS
            parent1=trait(num_trait,pop_member);
            pop_member2=ceil(rand*POP_SIZE);
            % se verifica que no sea el mismo miembro de la poblacion
               while pop_member2==pop_member
                  pop_member2=ceil(rand*POP_SIZE);
               end
            parent2=trait(num_trait,pop_member2);
            % se realiza el cruzamiento para real coded de acuerdo a la
            % regla.
            a=rand;
            child(num_trait,pop_member)=a*parent2+(1-a)*parent1;
            % se verifica si se paso del límite par no generar miembros de
            % la población no factibles.
            if child(num_trait,pop_member) > HIGHTRAIT(num_trait)
               child(num_trait,pop_member)=HIGHTRAIT(num_trait);
            elseif child(num_trait,pop_member) < LOWTRAIT(num_trait)
               child(num_trait,pop_member)=LOWTRAIT(num_trait);
            end
         end
      else
         child(:,pop_member)=trait(:,pop_member);
      end
   end
end
```

```
function  m =  dmc2(r,c2,time);

global m mbar predvect T lambda np nu G i gamma1

% inicializacion
if time==0
   load parametros_dmc;
   mbar=50;
   %lambda=0.2;
   m(1)=mbar;
   m(2)=0;
   predvect=c2*ones(np,1);
   i=time;
end
   if rem(time,T)==0 && time > i
      em=c2-predvect(1);
      predvect_u=[predvect(2:np);predvect(np)]+em;
      E=r*ones(np,1)-predvect_u;
      gamma2=gamma1*eye(np);
      Kc=inv(G'*gamma2'*gamma2*G+lambda*eye(nu))*G'*gamma2'*gamma2;
      delta_u=Kc*E;
      delta_u=delta_u(1);
      if (m(1) + delta_u) > 100,
         delta_u=100-m;
      elseif (m(1) + delta_u) < 0,
         delta_u = -m(1);
      end
      m(1)=m(1)+delta_u;
      predvect=G*[delta_u;zeros(nu-1,1)]+predvect_u;
      i=time;
      m(2)=em;
   else
         m=m;
   end
```

```
%Program for training a RAWN network
%This function finds the weights for the hidden and the output layer, that produce the
%minimum SSE (Y-Ypred).
%The inputs and outputs for training, and the number of hidden nodes must be provided.

%Weights are stored in a file called weights.

function
[Wh,Bh,Wo,Bo,Ymin,SSEtrain]=entrenamiento2(inputs,outputs,nodes,iter,inputs_2,outputs
_2);

ie=size(inputs,2);                          %identifies number of input elements
l=size(inputs,1);                                   %identifies training set dimension

%Include bias input
inputs(:,ie+1)=ones(l,1);          %ie+1 input elements

%Initializing variables
SSEflag=0;
Y=outputs;

%Finding weights for the minimum SSE of Y-Ypredicted
h = waitbar(0,'Entrenando RAWN...');
for j=1:iter                                                %iter is the number of trials per node
set
  W=zeros(ie+1,nodes);        %Initializing the weights of the hidden layer
  for q=1:nodes
    R=randn(ie+1,1);                    %generates a file of normally distributed random
numbers
    W(:,q)=sqrt(2/(max(sum(inputs.^2,2))))*R;
  end

  Z=inputs*W;   %net for each node in the hidden layer
  V=tansig(Z);  %output for each node in the hidden layer

  %Include bias input
  V(:,nodes+1)=ones(l,1);   %nodes+1 input elements
  Wout=inv(V'*V)*V'*Y;      %Least squares method  ojo!!!!!!!!!

  %validacion cruzada
  Wh_2=W(1:ie,:);
  Bh_2=W(ie+1,:);
  Wo_2=Wout(1:nodes,:);
  Bo_2=Wout(nodes+1,:);
  [Ypred]=nn_sim_val(inputs_2,Wh_2,Bh_2,Wo_2,Bo_2);
```

```
  %Ypred=V_2*Wout;
  SSE=sumsqr(outputs_2-Ypred);

  if SSEflag==0
    SSEtrain=SSE;              %Initializing SSEmin
    SSEflag=1;          %blocking
  end

  %Separating weights and biases
  if SSE<= SSEtrain
     SSEtrain=SSE;              %Picking the minimum SSE
     Ymin=Ypred;
     Wh=W(1:ie,:);
     Bh=W(ie+1,:);
     Wo=Wout(1:nodes,:);
     Bo=Wout(nodes+1,:);
      if SSEtrain<0.1
         break
      end
  end
     waitbar(j/iter)
  end
close(h)
fprintf('SSEtrain= %g\n',SSEtrain)
```

```
function nuevos=evr_selec(F,D,POP_SIZE2);
% Esta función realiza la seleccion de los individuos que se encontrarán en
% el archivo es decir hace la selección de Pt+1.

nuevos=find(F < 1);
nuevos=nuevos';
N=length(nuevos);
if N < POP_SIZE2
   [F2,nuevos2]=sort(F);
   nuevos=nuevos2(1:POP_SIZE2)';
else
   % se inicializa el contador para los miebros de la poblacion Pt+1
   % quienes son "nuevos"
   pop_member=1;
   % se inicializa la cuenta para ir desde la distancia minima hasta la
   % máxima en caso de que no se reduzcan los miembros con la primera.  se
   % empieza en tres porque la primera distancia mínima es la de el mismo y
   % siempre es cero
   dist_min=3;
   % Se hace un ciclo infinito que se rompe internamente solo cuando se
   % llega hasta N2
   while 1
      % Se busca si el miembro hace parte de la primera distancia y
      % debido a que se encuentran ordenadas ascendentemente esta es la
      % menor.  En caso de que se encuentre que ese miembro es el miembro
      % con distancia mínima a otro se le asigna al indice que se
      % encuentra en el vector "nuevos" de manera que se borre de la
      % nueva población.
      dist=find(D(:,dist_min)==nuevos(1,pop_member));
      if dist
         nuevos(pop_member)=0;
      end
      N2=nnz(nuevos);
      % Se pregunta  si ya se llegó al limite deseado, en caso que no sea
      % asi se pregunta si ya se recorrio toda la población para aumentar
      % el contardor dist_min y seguir con la distancia mínima siguiente
      if N2 ==POP_SIZE2
         break
      else
         if pop_member==N
            dist_min=dist_min+1;
            pop_member=1;
            %% si el puntero sobrepasa la longitud de la matriz D
            %% entonces sencillamente se hace lo mismo que arriba se
            %% ordenan y se cogen los primeros.
```

```
            if dist_min==size(D,2)
               [F2,nuevos2]=sort(F);
               nuevos=nuevos2(1:POP_SIZE2);
               break
            end
         else
            pop_member=pop_member+1;
         end
      end
   end
   % Una vez el ciclo termina se recuperar los indices de miembros que no
   % son ceros y se almacenan en el archivo.
   nuevos=nonzeros(nuevos);
   nuevos=nuevos';
end
```

```
function [fitness]=funciones_objetivos(trait,POP_SIZE,...
   y_act,y_ant,predvect_2,r_act,np_2,nu_2,G_2,m_2,Wh,Bh,Wo,Bo,delta_p,rph,mdu)


% se ejecuta el algoritmo de control DMC
em_2=y_act-predvect_2(1);
predvect_u_2=[predvect_2(2:np_2);predvect_2(np_2)]+em_2;
E_2=r_act-predvect_u_2;

for chrom_number = 1:POP_SIZE,       % Test fitness

   lambda_2=trait(1,chrom_number);
   gamma_2=trait(2,chrom_number);

   %E_2=r_act*ones(np_2,1)-predvect_u_2;
   gamma_2=gamma_2*eye(np_2);

Kc_2=inv(G_2'*gamma_2'*gamma_2*G_2+lambda_2*eye(nu_2))*G_2'*gamma_2'*gam
ma_2;
   delta_u_21=Kc_2*E_2;

   m_p=m_2;
   for cnt=1:length(delta_u_21)
       if (m_p(cnt) + delta_u_21(cnt)) > 100,
          delta_u_21(cnt)=100-m_p(cnt);
       elseif (m_p(cnt) + delta_u_21(cnt)) < 0,
          delta_u_21(cnt) = -m_p(cnt);
       end
       m_p(cnt+1)=delta_u_21(cnt)+m_p(cnt);
   end

   %se saturan las predicciones del DMC para los deltas de u

   predvect=G_2*[delta_u_21]+predvect_u_2;
   epred2=r_act-predvect;
   clear cnt

   % Se simula la red neuronal

   [y_pred]=nn_sim_auto([m_p(1:nu_2) y_act],...
      Wh,Bh,Wo,Bo,rph)+delta_p;

   % se calcula el error de prediccion
   e_pred=r_act(1:length(y_pred))-y_pred';
```

```
    if abs(delta_u_21(1)) <=mdu
        fitness(chrom_number,1)=sumsqr(e_pred);
        fitness(chrom_number,2)=sumsqr(delta_u_21);
        fitness(chrom_number,3)=sumsqr(epred2);
    else
        fitness(chrom_number,1)=1;
        fitness(chrom_number,2)=1;
        fitness(chrom_number,3)=1;
    end

end

no_ones=find(fitness(:,1)~=1);
max_fit=max(fitness(no_ones,1));
fitness(no_ones,1)=fitness(no_ones,1)./max_fit;
clear no_ones max_fit
no_ones=find(fitness(:,2)~=1);
max_fit=max(fitness(no_ones,2));
fitness(no_ones,2)=fitness(no_ones,2)./max_fit;
clear no_ones max_fit
no_ones=find(fitness(:,3)~=1);
max_fit=max(fitness(no_ones,3));
fitness(no_ones,3)=fitness(no_ones,3)./max_fit;
```

Entrenamiento inicial para la estrategia SISO sobre el tanque reactor con tanque de homogenización

```
function [Kp,tao,to,T,np,nu,lambda,gamma1,SSEtrain,Kc,ti,td,alpha1]=...
   initial_training1(dm11,T,np,nu,lambda,gamma1,at,z1,...
   fts,mo,nodes,iter,handles,at2,Kc,ti,td,alpha1,ftc)
% Identificación del proceso como FOPDT
h2 = waitbar(0,'FOPDT Identification');
waitbar(1/5,h2)
myopt = simset('DstWorkspace ', 'current','SrcWorkspace','current');
dm=dm11;
sim('model_ident_fopdt',3000,myopt)

%% ojo el algoritmo hay que revisarlo (interpolación) porque esta muy
%% dependiente del tiempo del step de entrada.
[Kp,tao,to]=ajuste_fit3(c,tout,dm);

% Se simula el modelo FOPDT obtenido a partir del fit3 para observar su
% comportamiento respecto al modelo real.
open('modelx.mdl')
set_param('modelx/FOPDT','m_bar',num2str(50))
set_param('modelx/FOPDT','c_bar',num2str(44.526))
set_param('modelx/FOPDT','Kp',num2str(Kp))
set_param('modelx/FOPDT','tao',num2str(tao))
set_param('modelx/FOPDT','to',num2str(to))
sim('modelx.mdl',2000,myopt)
axes(handles.axes34)
plot(tout,salida)
title('FOPDT system Vs Real system')
xlabel('Time (s)')
ylabel('%TO')
close_system('modelx',1)

waitbar(2/5,h2,'DMC tuning')
if at=='Yes'
% Sintonización del DMC
[T,np,nu,lambda]=sinto_dmc(Kp,tao,to);
% Factor de peso en la matriz dinámica
% Se escoge a priori debido a que las fórmulas de sintonizacion no lo
% arrojan.
gamma1=1;
end

% Con el período de muestreo obtenido anteriormente se obtiene la matriz
% dinámica
```

```
clear c tout  % se limpian las variables para que no haya incovenientes

% se simula teniendo en cuenta que le hace falta el parámetro T
% obtenido en el paso anterior
sim('model2.mdl',2000,myopt)

% Con los datos arrojados por el modelo se obtiene la matriz dinámica
% el único requisito es que las variables de entrada y salida se almacenen
% en el workspace como m y c
G=matrix_dinamica(c,m,np,nu);

% Se guardan los datos de sintonía en esa variable de modo que cuando se
% ejecute el DMC en simulink esten guardados los valores correctos alli.

save parametros_dmc G lambda np nu T gamma1

% Se escoge la respuesta requerida para el modelo de referencia de modo que
% se pueda obtener su vector igualmente.
waitbar(3/5,h2,'Reference Model')
z=z1;
Ts=fts*tao;
dm=10;
% es un modelo de primer orden debido a que el de segundo orden estab
% extraño
[num,den,polos]=polo_lazo_cerrado(z,Ts,T,mo);
open('dmc_adap_mod_ref.mdl')
set_param('dmc_adap_mod_ref/modelo_referencia/MR','Numerator',strcat('[',num2str(num)
,']'));
set_param('dmc_adap_mod_ref/modelo_referencia/MR','Denominator',strcat('[',num2str(de
n),']'));
close_system('dmc_adap_mod_ref',1)
% se limpian las variables para que no haya inconvenientes con la
% simulación
clear tout m
%open('modelo_referencia.mdl')
sim('modelo_referencia.mdl',2000,myopt)

% La variable de salida del modelo de referencia debe ser "mod_ref" y la de
% entrada "m". Además los valores iniciales del step y de entrada y salida
% de la función de transferencia deben coincidir con los del proceso.
% El horizonte de predicción es uno porque se asume desconocida
% la real referencia y el horizonte de predicción es el mismo que para la
% matriz del proceso.
```

```
R=matrix_dinamica(mod_ref,m,np,1);
axes(handles.axes36)
tiempo=1:T:length(R)*T;
plot(tiempo,R)
clear tiempo
title('Unit step response of the reference model')
ylabel('%TO')
xlabel('Time (s)')
% Se guardan el valor del vector de referencia de modo que cuando se
% ejecute en simulink  el DMC ésta se encuentre guardada propiamente.
save referencia R

% Se generan los conjuntos de entrenamiento y validación para las ANN a
% partir de una serie de pruebas escalón al proceso en donde se mira su
% comportamiento en todo el rango de entrada desde 0 a 100% en pasos
% definidos a priori. La salida del modelo de simulación debe llamarse "c"
% y la entrada que se va a usar para el entrenamiento se debe  llamar "md."
waitbar(4/5,h2,'SISO RAWNN Training')
% Conjunto de entrenamiento
clear m c tout
sim('data_train.mdl',5*tao*4,myopt)
m_train=md;
c_train=c;
save datos_planta c md

% Conjunto de validación
clear md c tout
sim('data_val.mdl',5*tao*4,myopt)
m_val=md;
c_val=c;
save datos_planta_val c md

% Entrenamiento de la red neuronal RAWN


[Wh,Bh,Wo,Bo,P,Ymin,SSEtrain]=train_rawn(m_train,c_train,m_val,c_val,nodes,iter,1,0);
save weights Wh Bh Wo Bo P;
clear Wh Bh Wo Bo P

waitbar(5/5,h2,'Digital PID Tuning')
if at2=='Yes'
%Sintonizacion del PID digital
[Kc,ti,td,T]=sintonizacion_PID(Kp,tao,to,0,0,1,tao/ftc,T);
% Valor de la constante para el filtro derivativo
end
```

```
save parametros_pid Kc ti td alpha1 T
close(h2)
```

Interfaz grafica de usuario para la estrategia SISO sobre el tanque reactor con tanque de homogenización

```
function varargout = interfaz2(varargin)
% INTERFAZ2 M-file for interfaz2.fig
%      INTERFAZ2, by itself, creates a new INTERFAZ2 or raises the existing
%      singleton*.
%
%      H = INTERFAZ2 returns the handle to a new INTERFAZ2 or the handle to
%      the existing singleton*.
%
%      INTERFAZ2('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in INTERFAZ2.M with the given input arguments.
%
%      INTERFAZ2('Property','Value',...) creates a new INTERFAZ2 or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before interfaz2_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to interfaz2_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.

% Edit the above text to modify the response to help interfaz2

% Last Modified by GUIDE v2.5 08-Jun-2006 14:37:00

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @interfaz2_OpeningFcn, ...
                   'gui_OutputFcn',  @interfaz2_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
```

```
else
   gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before interfaz2 is made visible.
function interfaz2_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to interfaz2 (see VARARGIN)

% Choose default command line output for interfaz2
handles.output = hObject;
%%%%%%%%%Valores iniciales para sintonizacion dmc y
PID%%%%%%%%%%%%%
handles.tfinal=12000;
handles.T1=0;
handles.np=0;
handles.nu=0;
handles.lambda=0;
handles.gamma1=0;
handles.at='Yes';
handles.at2='Yes';
handles.Kc=0;
handles.ti=0;
handles.td=0;
handles.alpha1=0.1;
handles.ftc=3;
%%%%%%%%%%%%%Valores iniciales para identificacion%%%%%%%%%%%%
handles.dm=-20;
%%%%%%%%%%Valores iniciales para modelos de referencia%%%%%%%%%%%%
handles.z1=0.8;
handles.fts=2;
handles.mo=2;
%%%%%%%%%%Valores iniciales para entrenamiento de la RNA%%%%%%%%%%
handles.nodes=8;
handles.iter=100;
%%%%%%%%%%Valores Iniciales del SPEA
II%%%%%%%%%%%%%%%%%%%%%%%%%%
handles.pc=0.9;
handles.pm=0.3;
handles.p=50;
```

```
handles.p1=20;
handles.n=10;
handles.lmin=10;
handles.lmax=2;
handles.gmin=10;
handles.gmax=2;
handles.mdu=15;
handles.rph=7;

pc=handles.pc;
pm=handles.pm;
p=handles.p;
p1=handles.p1;
n=handles.n;
lmin=handles.lmin;
lmax=handles.lmax;
gmin=handles.gmin;
gmax=handles.gmax;
mdu=handles.mdu;
rph=handles.rph;

save parametros_spea pc pm n p p1 lmin lmax gmin gmax mdu rph

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes interfaz2 wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = interfaz2_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



function DM_Callback(hObject, eventdata, handles)
% hObject    handle to DM (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of DM as text
%        str2double(get(hObject,'String')) returns contents of DM as a double

handles.dm=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function DM_CreateFcn(hObject, eventdata, handles)
% hObject    handle to DM (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function KP_Callback(hObject, eventdata, handles)
% hObject    handle to KP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of KP as text
%        str2double(get(hObject,'String')) returns contents of KP as a double


% --- Executes during object creation, after setting all properties.
function KP_CreateFcn(hObject, eventdata, handles)
% hObject    handle to KP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
```

```
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function TAO_Callback(hObject, eventdata, handles)
% hObject    handle to TAO (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TAO as text
%        str2double(get(hObject,'String')) returns contents of TAO as a double


% --- Executes during object creation, after setting all properties.
function TAO_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TAO (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function TO_Callback(hObject, eventdata, handles)
% hObject    handle to TO (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TO as text
%        str2double(get(hObject,'String')) returns contents of TO as a double


% --- Executes during object creation, after setting all properties.
function TO_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TO (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
```

```
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function T_Callback(hObject, eventdata, handles)
% hObject    handle to T (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of T as text
%        str2double(get(hObject,'String')) returns contents of T as a double
handles.T=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function T_CreateFcn(hObject, eventdata, handles)
% hObject    handle to T (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function NP_Callback(hObject, eventdata, handles)
% hObject    handle to NP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of NP as text
%        str2double(get(hObject,'String')) returns contents of NP as a double
handles.np=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
```

```
function NP_CreateFcn(hObject, eventdata, handles)
% hObject    handle to NP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function NU_Callback(hObject, eventdata, handles)
% hObject    handle to NU (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of NU as text
%        str2double(get(hObject,'String')) returns contents of NU as a double
handles.nu=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function NU_CreateFcn(hObject, eventdata, handles)
% hObject    handle to NU (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function LAMBDA_Callback(hObject, eventdata, handles)
% hObject    handle to LAMBDA (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of LAMBDA as text
%        str2double(get(hObject,'String')) returns contents of LAMBDA as a double
handles.lambda=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function LAMBDA_CreateFcn(hObject, eventdata, handles)
% hObject    handle to LAMBDA (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function GAMMA_Callback(hObject, eventdata, handles)
% hObject    handle to GAMMA (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of GAMMA as text
%        str2double(get(hObject,'String')) returns contents of GAMMA as a double
handles.gamma1=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function GAMMA_CreateFcn(hObject, eventdata, handles)
% hObject    handle to GAMMA (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function Z1_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to Z1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Z1 as text
%        str2double(get(hObject,'String')) returns contents of Z1 as a double
handles.z1=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function Z1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Z1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function FTS_Callback(hObject, eventdata, handles)
% hObject    handle to FTS (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of FTS as text
%        str2double(get(hObject,'String')) returns contents of FTS as a double
handles.fts=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function FTS_CreateFcn(hObject, eventdata, handles)
% hObject    handle to FTS (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
```

```
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes on selection change in MO.
function MO_Callback(hObject, eventdata, handles)
% hObject    handle to MO (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns MO contents as cell array
%        contents{get(hObject,'Value')} returns selected item from MO
contents = get(hObject,'String');
handles.mo=str2num(contents{get(hObject,'Value')});
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function MO_CreateFcn(hObject, eventdata, handles)
% hObject    handle to MO (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes on selection change in AT.
function AT_Callback(hObject, eventdata, handles)
% hObject    handle to AT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns AT contents as cell array
%        contents{get(hObject,'Value')} returns selected item from AT
contents = get(hObject,'String');
handles.at=contents{get(hObject,'Value')};
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
```

```
function AT_CreateFcn(hObject, eventdata, handles)
% hObject    handle to AT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function NODES_Callback(hObject, eventdata, handles)
% hObject    handle to NODES (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of NODES as text
%        str2double(get(hObject,'String')) returns contents of NODES as a double
handles.nodes=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function NODES_CreateFcn(hObject, eventdata, handles)
% hObject    handle to NODES (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function ITER_Callback(hObject, eventdata, handles)
% hObject    handle to ITER (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ITER as text
```

```
%        str2double(get(hObject,'String')) returns contents of ITER as a double
handles.iter=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function ITER_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ITER (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function ERROR_Callback(hObject, eventdata, handles)
% hObject    handle to ERROR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ERROR as text
%        str2double(get(hObject,'String')) returns contents of ERROR as a double


% --- Executes during object creation, after setting all properties.
function ERROR_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ERROR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes on button press in initial.
```

```
function initial_Callback(hObject, eventdata, handles)
% hObject    handle to initial (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[handles.Kp,handles.tao,handles.to,handles.T1,handles.np,handles.nu,handles.lambda,...
   handles.gamma1,handles.SSEtrain,handles.Kc,handles.ti,handles.td,handles.alpha1]=...
   initial_training1(handles.dm,handles.T1,handles.np,handles.nu,...
   handles.lambda,handles.gamma1,handles.at,handles.z1,handles.fts,...
   handles.mo,handles.nodes,handles.iter,handles,handles.at2,handles.Kc,...
   handles.ti,handles.td,handles.alpha1,handles.ftc);
guidata(hObject, handles);

%colocacion de los números en la interfaz2%%%%%%%%%%

% identificaion FOPDT
set(handles.KP,'String',num2str(handles.Kp));
set(handles.TAO,'String',num2str(handles.tao));
set(handles.TO,'String',num2str(handles.to));

%Sintonizacion DMC
set(handles.T,'String',num2str(handles.T1));
set(handles.NP,'String',num2str(handles.np));
set(handles.NU,'String',num2str(handles.nu));
set(handles.LAMBDA,'String',num2str(handles.lambda));
set(handles.GAMMA,'String',num2str(handles.gamma1));

%Sintonizacion PID
set(handles.KC,'String',num2str(handles.Kc));
set(handles.TI,'String',num2str(handles.ti));
set(handles.TD,'String',num2str(handles.td));
set(handles.ALPHA1,'String',num2str(handles.alpha1));

%RANWNN
set(handles.ERROR,'String',num2str(handles.SSEtrain));
msgbox('Training Finish','Initial Training','help')



function P_Callback(hObject, eventdata, handles)
% hObject    handle to P (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of P as text
```

```
%        str2double(get(hObject,'String')) returns contents of P as a double
handles.p=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function P_CreateFcn(hObject, eventdata, handles)
% hObject    handle to P (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function P1_Callback(hObject, eventdata, handles)
% hObject    handle to P1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of P1 as text
%        str2double(get(hObject,'String')) returns contents of P1 as a double
handles.p1=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function P1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to P1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
function N_Callback(hObject, eventdata, handles)
% hObject    handle to N (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of N as text
%        str2double(get(hObject,'String')) returns contents of N as a double
handles.n=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function N_CreateFcn(hObject, eventdata, handles)
% hObject    handle to N (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function PM_Callback(hObject, eventdata, handles)
% hObject    handle to PM (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of PM as text
%        str2double(get(hObject,'String')) returns contents of PM as a double
handles.pm=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function PM_CreateFcn(hObject, eventdata, handles)
% hObject    handle to PM (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
```

```
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function LMIN_Callback(hObject, eventdata, handles)
% hObject    handle to LMIN (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of LMIN as text
%        str2double(get(hObject,'String')) returns contents of LMIN as a double
handles.lmin=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function LMIN_CreateFcn(hObject, eventdata, handles)
% hObject    handle to LMIN (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function LMAX_Callback(hObject, eventdata, handles)
% hObject    handle to LMAX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of LMAX as text
%        str2double(get(hObject,'String')) returns contents of LMAX as a double
handles.lmax=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function LMAX_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to LMAX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function GMIN_Callback(hObject, eventdata, handles)
% hObject    handle to GMIN (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of GMIN as text
%        str2double(get(hObject,'String')) returns contents of GMIN as a double
handles.gmin=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function GMIN_CreateFcn(hObject, eventdata, handles)
% hObject    handle to GMIN (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function GMAX_Callback(hObject, eventdata, handles)
% hObject    handle to GMAX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of GMAX as text
%        str2double(get(hObject,'String')) returns contents of GMAX as a double
handles.gmax=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function GMAX_CreateFcn(hObject, eventdata, handles)
% hObject    handle to GMAX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function MDU_Callback(hObject, eventdata, handles)
% hObject    handle to MDU (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of MDU as text
%        str2double(get(hObject,'String')) returns contents of MDU as a double
handles.mdu=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function MDU_CreateFcn(hObject, eventdata, handles)
% hObject    handle to MDU (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
function PC_Callback(hObject, eventdata, handles)
% hObject    handle to PC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of PC as text
%        str2double(get(hObject,'String')) returns contents of PC as a double
handles.pc=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function PC_CreateFcn(hObject, eventdata, handles)
% hObject    handle to PC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function RPH_Callback(hObject, eventdata, handles)
% hObject    handle to RPH (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of RPH as text
%        str2double(get(hObject,'String')) returns contents of RPH as a double
handles.rph=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function RPH_CreateFcn(hObject, eventdata, handles)
% hObject    handle to RPH (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes on button press in DMC_START.
function DMC_START_Callback(hObject, eventdata, handles)
% hObject    handle to DMC_START (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global G T gamma1 lambda np nu
myopts = simset('DstWorkspace ', 'current','SrcWorkspace','current');
sim('model_dmc',handles.tfinal,myopts)
save resultados_dmc cont_dmc salida_real_dmc em_dmc salida_dmc salida_ruido_dmc
iae_dmc
axes(handles.axes1)
plot(tout,salida_real_dmc)
title('Output concentration ca3')
xlabel('Time (s)')
ylabel('Kg/m^3')
axes(handles.axes9)
plot(tout,salida_dmc)
title('Controlled variable without noise')
xlabel('Time (s)')
ylabel('%TO')
axes(handles.axes8)
plot(tout,salida_ruido_dmc)
title('Controlled variable with noise')
xlabel('Time (s)')
ylabel('%TO')
axes(handles.axes10)
plot(tout,cont_dmc)
title('Controller output')
xlabel('Time (s)')
ylabel('%CO')
axes(handles.axes11)
plot(tout,em_dmc)
title('Model Error')
xlabel('Time (s)')
ylabel('%TO')
set(handles.IAE,'String',num2str(max(iae_dmc(:,1))))
msgbox('Simulation Terminated','DMC','help')
```

```
% --- Executes on button press in NEDMC_START.
function NEDMC_START_Callback(hObject, eventdata, handles)
% hObject    handle to NEDMC_START (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global G T gamma1 lambda np nu
pc=handles.pc;
pm=handles.pm;
p=handles.p;
p1=handles.p1;
n=handles.n;
lmin=handles.lmin;
lmax=handles.lmax;
gmin=handles.gmin;
gmax=handles.gmax;
mdu=handles.mdu;
rph=handles.rph;

save parametros_spea pc pm n p p1 lmin lmax gmin gmax mdu rph

load parametros_dmc
myopts = simset('DstWorkspace ', 'current','SrcWorkspace','current');
sim('dmc_adap_mod_ref',handles.tfinal,myopts)
save resultados_dmc_adap cont_dmc_adap salida_real_dmc_adap em_dmc_adap
salida_dmc_adap salida_dmc_adap_red salida_ruido_dmc_adap iae_dmc_adap
axes(handles.axes12)
plot(tout,salida_real_dmc_adap)
title('Output concentration ca3')
xlabel('Time (s)')
ylabel('Kg/m^3')
axes(handles.axes15)
plot(tout,salida_dmc_adap)
title('Controlled variable without noise')
xlabel('Time (s)')
ylabel('%TO')
axes(handles.axes14)
plot(tout,salida_ruido_dmc_adap)
title('Controlled variable with noise')
xlabel('Time (s)')
ylabel('%TO')
axes(handles.axes16)
plot(tout,cont_dmc_adap)
title('Controller output')
xlabel('Time (s)')
```

```
ylabel('%CO')
axes(handles.axes17)
plot(tout,em_dmc_adap(:,1))
title('Model error')
xlabel('Time (s)')
ylabel('%TO')
axes(handles.axes18)
plot(tout,salida_dmc_adap_red)
title('RNA output vs Process output')
xlabel('Time (s)')
ylabel('%TO')

algo=dlmread('factores.txt');
n2=length(algo(:,1));
tiempo=1:T:n2*T;
lambda1=algo(:,1);
gamma11=algo(:,2);
clear algo
axes(handles.axes19)
stem(tiempo,lambda1);
title('Lambda each sample time')
xlabel('Time (s)')
ylabel('Lambda')
axes(handles.axes21)
stem(tiempo,gamma11,'y');
title('Gamma each sample time')
xlabel('Time (s)')
ylabel('Gamma')

set(handles.IAE_NEDMC,'String',num2str(max(iae_dmc_adap)))
msgbox('Simulation Terminated','ADAPTIVE DMC','help')

% --- Executes on button press in PLOTS_DMC.
function PLOTS_DMC_Callback(hObject, eventdata, handles)
% hObject    handle to PLOTS_DMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.Panel1,'Visible','off')
set(handles.Panel2,'Visible','off')
set(handles.Panel3,'Visible','off')
set(handles.DMC_RES,'Visible','on')
set(handles.BACK,'Visible','on')


% --- Executes on button press in PLOTS_NEDMC.
```

```
function PLOTS_NEDMC_Callback(hObject, eventdata, handles)
% hObject    handle to PLOTS_NEDMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.Panel1,'Visible','off')
set(handles.Panel2,'Visible','off')
set(handles.Panel3,'Visible','off')
set(handles.NEDMC_RES,'Visible','on')
set(handles.BACK,'Visible','on')

% --- Executes on button press in BACK2.
function BACK_Callback(hObject, eventdata, handles)
% hObject    handle to BACK2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.Panel1,'Visible','on')
set(handles.Panel2,'Visible','on')
set(handles.Panel3,'Visible','on')
set(handles.DMC_RES,'Visible','off')
set(handles.NEDMC_RES,'Visible','off')
set(handles.PID_RES,'Visible','off')
set(handles.BACK,'Visible','off')

function IAE_Callback(hObject, eventdata, handles)
% hObject    handle to IAE (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of IAE as text
%        str2double(get(hObject,'String')) returns contents of IAE as a double


% --- Executes during object creation, after setting all properties.
function IAE_CreateFcn(hObject, eventdata, handles)
% hObject    handle to IAE (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
function IAE2_Callback(hObject, eventdata, handles)
% hObject    handle to IAE2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of IAE2 as text
%        str2double(get(hObject,'String')) returns contents of IAE2 as a double


% --- Executes during object creation, after setting all properties.
function IAE2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to IAE2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end




function IAE_NEDMC_Callback(hObject, eventdata, handles)
% hObject    handle to IAE_NEDMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of IAE_NEDMC as text
%        str2double(get(hObject,'String')) returns contents of IAE_NEDMC as a double


% --- Executes during object creation, after setting all properties.
function IAE_NEDMC_CreateFcn(hObject, eventdata, handles)
% hObject    handle to IAE_NEDMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function IAE2_NEDMC_Callback(hObject, eventdata, handles)
% hObject    handle to IAE2_NEDMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of IAE2_NEDMC as text
%        str2double(get(hObject,'String')) returns contents of IAE2_NEDMC as a double


% --- Executes during object creation, after setting all properties.
function IAE2_NEDMC_CreateFcn(hObject, eventdata, handles)
% hObject    handle to IAE2_NEDMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



% --- Executes on button press in NEXT2.
function NEXT_Callback(hObject, eventdata, handles)
% hObject    handle to NEXT2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.NEDMC_RES,'Visible','off')
set(handles.BACK,'Visible','off')
```

```
set(handles.NEDMC_RES2,'Visible','on')


% --- Executes on button press in BACK2.
function BACK2_Callback(hObject, eventdata, handles)
% hObject    handle to BACK2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

set(handles.NEDMC_RES,'Visible','on')
set(handles.BACK,'Visible','on')
set(handles.NEDMC_RES2,'Visible','off')


% --- Executes on button press in NEXT2.
function NEXT2_Callback(hObject, eventdata, handles)
% hObject    handle to NEXT2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

set(handles.NEDMC_RES2,'Visible','off')
set(handles.NEDMC_RES3,'Visible','on')


% --- Executes on button press in BACK3.
function BACK3_Callback(hObject, eventdata, handles)
% hObject    handle to BACK3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

set(handles.NEDMC_RES2,'Visible','on')
set(handles.NEDMC_RES3,'Visible','off')


% --- Executes on button press in PLOTS_TRAINING.
function PLOTS_TRAINING_Callback(hObject, eventdata, handles)
% hObject    handle to PLOTS_TRAINING (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.Panel1,'Visible','off')
set(handles.Panel2,'Visible','off')
set(handles.Panel3,'Visible','off')
set(handles.T_RES,'Visible','on')

% --- Executes on button press in BACKT.
```

```
function BACKT_Callback(hObject, eventdata, handles)
% hObject    handle to BACKT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.Panel1,'Visible','on')
set(handles.Panel2,'Visible','on')
set(handles.Panel3,'Visible','on')
set(handles.T_RES,'Visible','off')



function KC_Callback(hObject, eventdata, handles)
% hObject    handle to KC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of KC as text
%        str2double(get(hObject,'String')) returns contents of KC as a double
handles.Kc=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function KC_CreateFcn(hObject, eventdata, handles)
% hObject    handle to KC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function TI_Callback(hObject, eventdata, handles)
% hObject    handle to TI (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TI as text
%        str2double(get(hObject,'String')) returns contents of TI as a double
```

```
handles.ti=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function TI_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TI (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function TD_Callback(hObject, eventdata, handles)
% hObject    handle to TD (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TD as text
%        str2double(get(hObject,'String')) returns contents of TD as a double
handles.td=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function TD_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TD (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
function ALPHA1_Callback(hObject, eventdata, handles)
% hObject    handle to ALPHA1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ALPHA1 as text
%        str2double(get(hObject,'String')) returns contents of ALPHA1 as a double
handles.alpha1=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function ALPHA1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ALPHA1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end




% --- Executes on selection change in AT2.
function AT2_Callback(hObject, eventdata, handles)
% hObject    handle to AT2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns AT2 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from AT2
contents = get(hObject,'String');
handles.at2=contents{get(hObject,'Value')};
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function AT2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to AT2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end




% --- Executes on button press in PID_START.
function PID_START_Callback(hObject, eventdata, handles)
% hObject    handle to PID_START (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global T Kc ti td alpha1
load parametros_pid
myopts = simset('DstWorkspace ', 'current','SrcWorkspace','current');
sim('model_pid_digital',handles.tfinal,myopts)
save resultados_pi_d cont_pi_d salida_real_pi_d salida_pi_d salida_ruido_pi_d iae_pi_d
axes(handles.axes38)
plot(tout,salida_real_pi_d)
title('Output concentration ca3')
xlabel('Time (s)')
ylabel('Kg/m^3')
axes(handles.axes41)
plot(tout,cont_pi_d)
title('Controller output')
xlabel('Time (s)')
ylabel('%CO')
axes(handles.axes39)
plot(tout,salida_ruido_pi_d)
title('Controlled variable with noise')
xlabel('Time (s)')
ylabel('%TO')
axes(handles.axes40)
plot(tout,salida_pi_d)
title('Controlled variable without noise')
xlabel('Time (s)')
ylabel('%TO')
set(handles.IAE_PID,'String',num2str(max(iae_pi_d)))
msgbox('Simulation Terminated','PID','help')

% --- Executes on button press in PLOTS_PID.
```

```
function PLOTS_PID_Callback(hObject, eventdata, handles)
% hObject    handle to PLOTS_PID (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.Panel1,'Visible','off')
set(handles.Panel2,'Visible','off')
set(handles.Panel3,'Visible','off')
set(handles.PID_RES,'Visible','on')
set(handles.BACK,'Visible','on')


function IAE_PID_Callback(hObject, eventdata, handles)
% hObject    handle to IAE_PID (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of IAE_PID as text
%        str2double(get(hObject,'String')) returns contents of IAE_PID as a double


% --- Executes during object creation, after setting all properties.
function IAE_PID_CreateFcn(hObject, eventdata, handles)
% hObject    handle to IAE_PID (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end




% --- Executes on button press in OPEN_PID.
function OPEN_PID_Callback(hObject, eventdata, handles)
% hObject    handle to OPEN_PID (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
open('model_pid_digital.mdl')

% --- Executes on button press in OPEN_DMC.
```

```
function OPEN_DMC_Callback(hObject, eventdata, handles)
% hObject    handle to OPEN_DMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
open('model_dmc.mdl')

% --- Executes on button press in OPEN_NEDMC.
function OPEN_NEDMC_Callback(hObject, eventdata, handles)
% hObject    handle to OPEN_NEDMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
open('dmc_adap_mod_ref.mdl')
```

```
function G=matrix_dinamica(c,m,np,nu)
% Funcion para obtener la matriz dinámica del DMC basandose en la respuesta
% a un paso escalón muestreada con período T. Se deben especificar
% parámetros como el horizonte de prediccion y control de modo que se pueda
% crear la matriz adecuadamente.
% G=matrix_dinamica(c,m,np,nu)

Su=c;
Su=(Su-Su(1))./(m(length(m))-m(1));

G = [];
for i = 1:nu
G = [G, [zeros(i-1,1);Su(1:np-(i-1))]];
end
```

```
function
[trait,bestmember]=mutacion2(trait,MUTAT_PROB,POP_SIZE,NUM_TRAITS,HIGHTR
AIT,LOWTRAIT,...
   popcount,MAX_GENERATION,ELITISM,bestmember)
% Realiza la mutacion no uniforme sobre real coded genes

b=2; % este valor es escogido a priori se escogio el mismo del paper.
for pop_member=1:POP_SIZE
   if ELITISM==1 && pop_member == bestmember  % verifica si hay elitismo
      trait(:,pop_member)=trait(:,pop_member);
      bestmember=pop_member;
   else
      if MUTAT_PROB > rand
         for num_trait=1:NUM_TRAITS
            if round(rand)==0
               delta=(HIGHTRAIT(num_trait)-trait(num_trait,pop_member))*...
                  rand*(1-popcount/MAX_GENERATION)^b;
               trait(num_trait,pop_member)=trait(num_trait,pop_member)+ delta;
            else
               delta=(trait(num_trait,pop_member)-LOWTRAIT(num_trait))*...
                  rand*(1-popcount/MAX_GENERATION)^b;
               trait(num_trait,pop_member)=trait(num_trait,pop_member)- delta;
            end
         end
      else
         trait(:,pop_member)=trait(:,pop_member);
      end
   end
end
```

```
function [m]=nedmcs(r,c2,time);

global m mbar predvect T lambda gamma1 lambda_inicial np nu G i r_inicial
global R delta_r r_ant mod_ref Wh Bh Wo Bo
global c_ant cp_ant delta_p delta_u_1 y_ant P
global pc pm n p p1 lmin lmax gmin gmax mdu rph
% inicializacion
if time==0
   load parametros_dmc;
   load referencia
   load weights
   load parametros_spea
   lambda_inicial=lambda;
   dlmwrite('factores.txt',[lambda gamma1]);
   mbar=50;
   r_inicial=r;
   m(1)=mbar;
   m(2)=0;
   % Se guardan los valores anteriores de predicciones al igual que de la
   % entrada para sumarle el delta a las nuevas predicciones hechas por la
   % red en el SPEA II
   cp_ant=nn_sim_val([m(1) c2],Wh,Bh,Wo,Bo);
   m(3)=cp_ant;
   delta_p=cp_ant(1)-c2;
   % Valor incial de la predicción
   predvect=c2*ones(np,1);
   % Valor anterior para actualizacion del modelo de referencia
   r_ant=r;
   % Valor incial del delta_r porque no hay cambio al principio empieza
   % con ese delta.
   delta_r=r-44.524;
   % Valor inicial del modelo de referencia
   mod_ref=R*delta_r+44.524*ones(np,1);
   i=time;
   y_ant=c2;
end
   if rem(time,T)==0 && time > i

      % Cambio de la referencia
      delta_r=r-r_ant;
      % se actualiza el vector de modelo de referencia
      mod_ref=R*delta_r+[mod_ref(2:np);mod_ref(np)];

      %%%%% Actualizacion pesos RAWNN %%%%%%%%%%%%%%%%%%%%
      [Wh,Bh,Wo,Bo,P]=rls_rawnn2([m(1) y_ant],c2,Wh,Bh,Wo,Bo,P);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%

    %%%%%%%% Cálculo del mejor lambda a través del SPEA II
%%%%%%%%%%%%%
    best=SPEAIIS([lambda gamma1],r,delta_r,mod_ref,c2,y_ant,...
       predvect,G,np,nu,m,Wh,Bh,Wo,Bo,delta_p,...
       pc,pm,n,p,p1,lmin,lmax,gmin,gmax,mdu,rph);
    lambda=best(1);
    gamma1=best(2);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%

    em=c2-predvect(1);
    predvect_u=[predvect(2:np);predvect(np)]+em;
    E=mod_ref-predvect_u;
    gamma2=gamma1*eye(np);
    Kc=inv(G'*gamma2'*gamma2*G+lambda*eye(nu))*G'*gamma2'*gamma2;
    delta_u_t=Kc*E;
    delta_u=delta_u_t(1);

    % Saturación de los delta_u
    if delta_u > mdu
       delta_u=mdu;
    elseif delta_u < -mdu
       delta_u=-mdu;
    end

    % Saturación de la señal de control final
    if (m(1) + delta_u) > 100,
       delta_u=100-m(1);
    elseif (m(1) + delta_u) < 0,
       delta_u = -m(1);
    end
    m(1)=m(1)+delta_u;

    % Cálculo de la nueva predicción
    predvect=G*[delta_u;zeros(nu-1,1)]+predvect_u;

    i=time;
    % Cálculo de las acciones de control futuras basadas en los
    % m+delta_m para el calculo de la predicci{on de la red neuronal
```

```
    m_p=m;
    for cnt=1:length(delta_u_t)
       if (m_p(cnt) + delta_u_t(cnt)) > 100,
          delta_u_t(cnt)=100-m_p(cnt);
       elseif (m_p(cnt) + delta_u_t(cnt)) < 0,
          delta_u_t(cnt) = -m_p(cnt);
       end
       m_p(cnt+1)=delta_u_t(cnt)+m_p(cnt);
    end
    % Se actualiza el r anterior
    r_ant=r;

    % Error de modelaje de la red neuronal
    delta_p=cp_ant(1)-c2;
    cp_ant=nn_sim_val([m_p(1) c2],Wh,Bh,Wo,Bo);
    m(2)=em;
    m(3)=cp_ant;
    y_ant=c2;
    return
 else
       m=[m];
       return
end
```

```
function y=nn_sim_auto(inputs,Wh,Bh,Wo,Bo,ph)
%Esta función simula una red neuronal que predice hasta ph basándose en sus
%propias predicciones

y(1)=nn_sim_val([inputs(1) inputs(ph+1)],Wh,Bh,Wo,Bo);

for i=2:ph
   y(i)=nn_sim_val([inputs(i) y(i-1)],Wh,Bh,Wo,Bo);
end
```

```
%This function runs the RAWN net (one hidden layer, one output layer)
%This function is used in the file testing.mdl in the Matlab function.

function [c_pred]=nn_sim_val(inputs,Wh,Bh,Wo,Bo)


                                                        %adapting the input
vector (matrix 1 x #input elements)
ie=size(Wh,1);          %number of input elements
nodes=size(Wh,2);                       %number of nodes in the hidden layer

inputs(:,ie+1)=1;                               %including bias
Wh(ie+1,:)=Bh;                                  %Complete weight matrix for
hidden layer
net=inputs*Wh;                                  %net of each node
outs=tansig(net);                               %output of each node

outs(:,nodes+1)=1;                      %including bias
Wo(nodes+1,:)=Bo;                               %complete weight matrix for output
layer
c_pred=outs*Wo;                         %Predicted output
```

```
function [frente,S,R,D,F]=pareto(fitness,POP_SIZE,NUM_TRAITS,OBJETIVOS)
% fitness(pop_member,num_obj)
i=1;
S=zeros(POP_SIZE,1);% se inicializan los rangos de cada punto solucion.
S_fente=0;%se almacena el rango del los miembros del frente asi como de todos los demás
R=zeros(POP_SIZE,1);% se almacenan el valor de raw fitness
% se calcula el valor del miembro a tomar para la distancia segun la regla
% del paper, teniendo en cuenta que este POP_SIZE es fruto de la suma de
% ambos tamaños.
kth=ceil(sqrt(POP_SIZE));
% Se iniciliza el contador de elementos que dominan a una solución
cont=zeros(POP_SIZE,1);
% Se inicializa la matriz que alamcena las posciones en donde se encuentran
% los miemebros que dominan a cada solucion
r=zeros(POP_SIZE,1);
% se inicializan las distancias k-esimas
D=zeros(POP_SIZE,POP_SIZE+1);
frente=1;
for pop_member=1:POP_SIZE
   % Se inicializa la variable para asignar las distancias que se
   % almacenaran temporalmente
   d_temp=zeros(POP_SIZE,1);
   % se calcula un fitness temporal que es equivalente a la suma de todos
   % los fitness de manera que se penalice al que la suma de sus ftness
   % sea mayor, asegurando que sea menor que uno debido a que esta
   % cantidad se le va asumar a F.
   %fit_temp(pop_member,1)=1/(sum(fitness(pop_member,:))+4);
   for a=1:POP_SIZE
      k=0;% cuenta cuantos objetivos tiene por debajo del resto
      % se calcula la distancia del miembro pop_member a los otros
      % miembros "a".
      d_temp(a)=sqrt(sum((fitness(pop_member)-fitness(a)).^2));
      for num_obj=1:OBJETIVOS
         % Dominancia pareto fuerte
         if fitness(pop_member,num_obj) < fitness(a,num_obj)
            k=k+1; % se aumenta en uno si encuentra alguno
         end
      end
      if k==OBJETIVOS
         S(pop_member)=S(pop_member)+1;
         % Si la solucion pop_member domina a la solucion a entonces la
         % solución a es dominada por el.(sounds stupid but..)
         cont(a)=cont(a)+1;
         r(a,cont(a))=pop_member;
      end
```

```
    end
    % Se organizan las distancias en orden ascendente
    [d_temp,I]=sort(d_temp);
    % Se escoge la distancia k-esima
    sigmak=d_temp(kth);
    % Se asigna la distancia e acuerdo a la fórmula del paper
    D(pop_member,1)=1/(sigmak + 2);
    % se penaliza si la distancia mínima es cero.
    if sigmak==0
       D(pop_member,1)=1;
    end
    % Se guardan los indices de los miembros con las distancias desde la
    % menor hasta la mayor ya ordenadas.  Esto es para uso posterior en el
    % algoritmo de enviromental selection.
    D(pop_member,2:size(I,1)+1)=I';
 end
% se suman todos los S(j) de los elementos no ceros que dominan a i
for pop_member=1:POP_SIZE
    if r(pop_member,1)~=0
       R(pop_member)=sum(S(nonzeros(r(pop_member,:))'));
    else
       R(pop_member)=0;
       frente(i,1)=pop_member;
       i=i+1;
    end
end

F=R+D(:,1);%+fit_temp;
```

```
function m=pid_digital(r,c2,c_ant,m_ant,time);

global m mbar T Kc ti td alpha1 e e_ant i a

% inicializacion
if time==0
   load parametros_pid;
   a=(alpha1*td)/(alpha1*td+T);
   e_ant=0;
   m=50;
   i=time;
end
   if rem(time,T)==0 && time > i
      cn=a*c_ant+(1-a)*c2+(td/(alpha1*td+T))*(c2-c_ant);
      e=r-cn;
      delta_m=Kc*(e-e_ant+(T/ti)*e);
      e_ant=e;
      m=delta_m+m_ant;
      i=time;
      return
   else
         m=[m];
         return
   end
```

```
function [num,den,polos]=polo_lazo_cerrado(z,Ts,T,orden)
% Funcion que obtiene el numerador y el denominador de la función de
% transferencia del modelo de referencia en tiempo discreto a partir de
% las especificaciones de respuesta dinámica como lo son: el zita, el
% tiempo de estabilización y el tiempo de muestreo. La función arroja demás
% los polos en lazo cerrado.
% [num,den,polos]=polo_lazo_cerrado(z,Ts,T)

wn=5/(z*Ts);


if nargin ~=4
   orden=1; % primer orden por defecto
end

if orden==2
   b=exp(-T*z*wn);
   a=T*wn*sqrt(1-z^2);
   num=[1-2*b*cos(a)+b^2];
   den=[1 -2*b*cos(a) b^2];
   polos=roots(den);
else
   tao=Ts/5;
   a=exp(-T/tao);
   b=1-a;
   num=[b];
   den=[1 -a];
   polos=roots(den);
end
```

```
function [Wh,Bh,Wo,Bo,P]=rls_rawnn2(X,Y,Wh,Bh,Wo,Bo,P)


    %sig=0.01;
    ie=size(Wh,1);
    nodes=size(Wh,2);
    Wh2=Wh;
    Wh2(ie+1,:)=Bh;
    Wo2=Wo;
    Wo2(nodes+1,:)=Bo;
    %X=[a b];
    X(:,ie+1)=1;
    Z=X*Wh2;
    V=tansig(Z);
    V(:,nodes+1)=1;
    P=P-P*V'*inv(1+V*P*V')*V*P;
    %P=(1/sig)*(eye(size(P,1))-P*V'*inv(sig+V*P*V')*V)*P;
    Wo2=Wo2+P*V'*(Y-V*Wo2);
    Wo=Wo2(1:nodes,:);
    Bo=Wo2(nodes+1,:);
```

```
function trait=saturacion(trait,POP_SIZE,NUM_TRAITS,HIGHTRAIT,LOWTRAIT)

for pop_member = 1:POP_SIZE

   for current_trait = 1:NUM_TRAITS
      if trait(current_trait,pop_member)>HIGHTRAIT(current_trait)
      % The trait has went higher than the upper bound so let the trait equal to the
      % HIGHTRAIT bound.

         trait(current_trait,pop_member)=HIGHTRAIT(current_trait);

% Now consider the other case:

      elseif trait(current_trait,pop_member)<LOWTRAIT(current_trait)
         % The trait has went lower than the lower bound so let the trait equal to the
         % LOWTRAIT bound

           trait(current_trait,pop_member)=LOWTRAIT(current_trait);
      end
   end
end
```

```
function P_sel=seleccion4(F,nuevos)
% Esta funcion realiza la seleccion mediante binary tournament
% primero se encuentra la longitud del vector nuevos para saber cual es el
% tamañao del archivo
N=length(nuevos);
% Se itera hasta N para asegurar dew que haya N miembros en el mating pool
for pop_member=1:N
   % Se selecciona aleatoriamente dos padres y el que tenga mayor fitness
   % de los dos gana. Se aproxima siempre hacia arriba de modo que no
   % tengamos un indice cero.
   parent1=ceil(rand*N);
   parent2=ceil(rand*N);
   if F(parent1) <= F(parent2)
      % se captura el indice real en el vector "nuevos" debido a que cuando
      % entra a esta funcion el vector F pierde los indices reales y
      % además los F que necesitamos son solo los que quedaron despues
      % del enviromental selection
      P_sel(pop_member)=nuevos(parent1);
   else
      P_sel(pop_member)=nuevos(parent2);
   end
end
```

```
function [T,np,nu,lambda]=sinto_dmc(Kp,tao,to)
% Función para sintonizar los parámetros de un DMC SISO
% basándose en la identificación de un proceso como FOPDT
% [T,np,nu,lambda]=sinto_dmc(Kp,tao,to)

T=floor(max(0.1*tao,0.5*to));
k=(to/T)+1;
np=round((5*tao)/T+k);
nu=round((tao/T)+k);
%lambda=(np/10)*((3.5*tao)/T+2-(np-1)/2)*Kp^2;
lambda=(nu/500)*(Kp^2)*(np-k-1.5*(tao/T)+2-(nu-1)/2);
```

```
function [Kc,ti,td,T]=sintonizacion_PID(Kp,tao,to,metodo,modificado,digital,tc,T)
% A esta funcion se le pasan los parametro Kp,tao,to, del proceso
% y el metodo de sintonizacion que se requiera a saber:
% metodo=1  IAE
% metodo=2  ITAE
% si no se le da un metodo entonce seleccion ISE
% Se da la opción de modificado en caso de que se implemente con el filtro
% derivativo y las modificaciones propuestas en el libro de Smith. 1 si es
% modificado y 0 si no.
% Iguelmente se da la opción en caso de que la implementación del
% controlador sea digital, por tanto las fórmulas de sintonización varían
% También se requiere que se especifique tc el cual es la constante de
% tiempo en lazo cerrrado en caso de implementación digital.

if digital==0
   if metodo==1
      a1=1.435;
      b1=-0.921;
      a2=0.878;
      b2=0.749;
      a3=0.482;
      b3=1.137;
   elseif metodo==2
      a1=1.357;
      b1=-0.947;
      a2=0.842;
      b2=0.738;
      a3=0.381;
      b3=0.995;
   else
      a1=1.495;
      b1=-0.945;
      a2=1.101;
      b2=0.771;
      a3=0.560;
      b3=1.006;
   end

   Kc=(a1/Kp)*(to/tao)^b1;
   ti=(tao/a2)*(to/tao)^b2;
   td=(a3*tao)*(to/tao)^b3;

   if modificado==1
      b=(0.5+sqrt(-0.25+(td/ti)));
      Kc=Kc*b;
```

```
        ti=ti*b;
        td=td/b;
    end
    T=0;
else
    % las siguientesw fórmulas difieren de las originales en el sentido que
    % estan pensadas para fopdt y no para sopdt por tanto se suprimio b2 de
    % as formulas el cual esta relacionado con la segunda constante de
    % tiempo del sistema sopdt.
    if nargin == 8
        T=T;
    else
        T=floor(0.1*tao);
    end
    b1=exp(-T/tao);
    N=to/T;
    q=exp(-T/tc);
    Kc=((1-q)*b1)/(Kp*(1-b1)*(1+N*(1-q)));
    ti=(T*b1)/(1-b1);
    % td es cero debido a que depende de b2 la cual siempre es cero si t2=0.
    td=0;
end
```

```
function best3=SPEAIIS(params,r,delta_r,r_act,y_act,y_ant,...
   predvect_2,G_2,np_2,nu_2,m_2,Wh,Bh,Wo,Bo,delta_p,...
   pc,pm,n,p,p1,lmin,lmax,gmin,gmax,mdu,rph)


  if abs(delta_r) > 0
     global G lambda np nu T gamma1
       load parametros_dmc
       params=[lambda gamma1];
     clear G lambda np nu T gamma1
  end
  NUM_TRAITS=2;            % Number of traits in each individual

  HIGHTRAIT=[params(1)+lmax params(2)+gmax];            % Upper limit of a trait

  if params(1)-lmin < 0.2
     params(1)=lmin+0.2;
  end

  if params(2)-gmin < 1
     params(2)=gmin+1;
  end

  LOWTRAIT =[params(1)-lmin params(2)-gmin];
  MUTAT_PROB=pm;
  CROSS_PROB=pc;
  POP_SIZE=p;
  POP_SIZE2=p1;
  ELITISM=0;
  % se desactiva el elitismo y se hace el mejor miembro cero debido a que
  % las funciones de variacion lo necesitan(mutacion y cruzamiento)
  bestmember=0;

  MAX_GENERATION=n;        % Number of times the loop will run before
                                                % giving up on the EPSILON-DELTA
termination cond.

   OBJETIVOS=3;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

popcount=1;              % Initialize the generation count,
                                                % set it to one, the first population
```

```
% Se inicializa la poblacion de controladores con un controlador diseñado
% previamente con formulas de sintonizacion basadas en un modelo fopdt de
% la planta y se le agrega un ruido de media 0 y varianza uno.


for i=1:NUM_TRAITS
   for j=1:POP_SIZE
      trait(i,j,:)=(HIGHTRAIT(i) -LOWTRAIT(i) + 1)*rand + LOWTRAIT(i);
   end
end

for i=1:NUM_TRAITS
   for j=1:POP_SIZE2
      trait2(i,j,:)=(HIGHTRAIT(i)-LOWTRAIT(i) + 1)*rand+ LOWTRAIT(i);
   end
end

% Se calcula el tamaño de la población unión del archivo y de la población
POP_SIZE3=POP_SIZE+POP_SIZE2;

while popcount <= MAX_GENERATION

% First, fix bad traits (i.e., ones that are out of the range
% specified by HIGHTRAIT and LOWTRAIT) by saturation at the extremes

trait(:,:,popcount)=saturacion(trait(:,:,popcount),POP_SIZE,NUM_TRAITS,...
   HIGHTRAIT,LOWTRAIT);

trait2(:,:,popcount)=saturacion(trait2(:,:,popcount),POP_SIZE2,NUM_TRAITS,...
   HIGHTRAIT,LOWTRAIT);

% se genera el conjunto union concatenando los dos conjuntos
trait_t=[trait(:,:,popcount) trait2(:,:,popcount)];

[fitness(:,:,popcount)]=funciones_objetivos(trait_t,POP_SIZE3,y_act,y_ant,predvect_2,r_ac
t,np_2,...
   nu_2,G_2,m_2,Wh,Bh,Wo,Bo,delta_p,rph,mdu);

%fitness(:,:,popcount)=evaluacion(trait_t,POP_SIZE3);

% se obtiene todas las soluciones no dominadas
[frente_temp,S(:,popcount),R(:,popcount),D(:,:,popcount),F(:,popcount)]=...
   pareto(fitness(:,:,popcount),POP_SIZE3,NUM_TRAITS,OBJETIVOS);
```

```
% Se almacena en un frente temporal debido a que la longitud de este varía
% y por tanto puede haber un erro en la asignación cuando la longitud sea
% menor.
long_frente=length(frente_temp);
frente(1:long_frente,popcount)=frente_temp;
clear frente_temp

nuevos=evr_selec(F(:,popcount),D(:,:,popcount),POP_SIZE2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
% Create the next generation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% First, form the mating pool.
% To do this we select as parents the
% chromosomes that are most fit.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

P_sel=seleccion4(F(nuevos,popcount),nuevos);

popcount=popcount+1;                    % Increment to the next generation

% Se copian los individuos del nuevo frente en el cual se han puesto a
% concursar a los individuos modificados y a los resultantes de la
% seleccion anterior en trait

trait(:,1:POP_SIZE2,popcount)=trait_t(:,nuevos);

trait2(:,:,popcount)=cruzamiento4(trait_t(:,P_sel),F(nuevos,popcount-1),...
    CROSS_PROB,POP_SIZE2,NUM_TRAITS,ELITISM,bestmember,...
    HIGHTRAIT,LOWTRAIT);

trait2(:,:,popcount)=cruzamiento5(trait2(:,:,popcount),...

CROSS_PROB,POP_SIZE2,NUM_TRAITS,ELITISM,bestmember,HIGHTRAIT,LOWT
RAIT);

trait2(:,:,popcount)=mutacion2(trait2(:,:,popcount),MUTAT_PROB,POP_SIZE2,NUM_TR
AITS,...
   HIGHTRAIT,LOWTRAIT,popcount,MAX_GENERATION,ELITISM,bestmember);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%
end  % End "for pop_count=..." loop - the main loop.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%


fitness_temp=fitness(nuevos,1,MAX_GENERATION)+...
   fitness(nuevos,2,MAX_GENERATION)+...
   fitness(nuevos,3,MAX_GENERATION);

[best_fitness,best2]=min(fitness_temp);
best2=nuevos(best2);
best3=trait_t(:,best2);

dlmwrite('factores.txt',best3','-append');
```

```
function s = sumsqr(a)
%SUMSQR Sum squared elements of a matrix.
%
%  Synopsis
%
%    sumsqr(m)
%
%  Description
%
%    SUMSQR(M) returns the sum of the squared elements in M.
%
%  Examples
%
%    s = sumsqr([1 2;3 4])

% Mark Beale, 1-31-92
% Copyright 1992-2002 The MathWorks, Inc.
% $Revision: 1.10 $  $Date: 2002/04/14 21:36:05 $

if nargin < 1,error('Not enough input arguments.');end

s = sum(sum(a.*a));
```

```
function a = tansig(n,b)
%TANSIG Hyperbolic tangent sigmoid transfer function.
%
%  Syntax
%
%    A = tansig(N)
%    info = tansig(code)
%
%  Description
%
%    TANSIG is a transfer function.  Transfer functions
%    calculate a layer's output from its net input.
%
%    TANSIG(N) takes one input,
%      N - SxQ matrix of net input (column) vectors.
%    and returns each element of N squashed between -1 and 1.
%
%    TANSIG(CODE) returns useful information for each CODE string:
%      'deriv'  - Returns name of derivative function.
%      'name'   - Returns full name.
%      'output' - Returns output range.
%      'active' - Returns active input range.
%
%    TANSIG is named after the hyperbolic tangent which has the
%    same shape.  However, TANH may be more accurate and is
%    recommended for applications that require the hyperbolic tangent.
%
%  Examples
%
%    Here the code to create a plot of the TANSIG transfer function.
%
%      n = -5:0.1:5;
%      a = tansig(n);
%      plot(n,a)
%
%  Network Use
%
%    You can create a standard network that uses TANSIG
%    by calling NEWFF or NEWCF.
%
%    To change a network so a layer uses TANSIG set
%    NET.layers{i,j}.transferFcn to 'tansig'.
%
%    In either case, call SIM to simulate the network with TANSIG.
%    See NEWFF or NEWCF for simulation examples.
```

```
%
%  Algorithm
%
%    TANSIG(N) calculates its output according to:
%
%      n = 2/(1+exp(-2*n))-1
%
%    This is mathematically equivalent to TANH(N).  It differs
%    in that it runs faster than the MATLAB implementation of TANH,
%    but the results can have very small numerical differences.  This
%    function is a good trade off for neural networks, where speed is
%    important and the exact shape of the transfer function is not.
%
%  See also SIM, DTANSIG, LOGSIG.

% Mark Beale, 1-31-92
% Revised 12-15-93, MB
% Revised 11-31-97, MB
% Copyright 1992-2002 The MathWorks, Inc.
% $Revision: 1.10 $  $Date: 2002/04/14 21:31:55 $

if nargin < 1, error('Not enough arguments.'); end

% FUNCTION INFO
if isstr(n)
  switch (n)
    case 'deriv'
    a = 'dtansig';
    case 'name'
    a = 'Tan Sigmoid';
    case 'output'
    a = [-1 1];
    case 'active'
    a = [-2 2];
    case 'type'
    a = 1;

    % **[ NNT2 Support ]**
    case 'delta'
    a = 'deltatan';
    nntobsu('tansig','Use TANSIG("deriv") instead of TANSIG("delta").')
    case 'init'
    a = 'nwtan';
    nntobsu('tansig','Use network propreties to obtain initialization info.')
```

```
    otherwise, error('Unrecognized code.')
  end
  return
end

% CALCULATION

% **[ NNT2 Support ]**
if nargin == 2
  nntobsu('tansig','Use TANSIG(NETSUM(Z,B)) instead of TANSIG(Z,B).')
  n=n+b(:,ones(1,size(n,2)));
end

a = 2 ./ (1 + exp(-2*n)) - 1;
i = find(~finite(a));
a(i) = sign(n(i));
```

```
function
[Wh,Bh,Wo,Bo,P,Ymin,SSEtrain]=train_rawn(m_train,c_train,m_val,c_val,nodes,iter,ph,c
h)

md=m_train;
c=c_train;
md_2=m_val;
c_2=c_val;

clear m_train c_train m_val c_val

if ch==0  % se verifica el horizonte de entradas sea distinto de cero.
   m=0;
else
   % se hacen los vectores para el entrenamiento desde m(t) hasta m(t+ch)
   for i=1:ch-1
   m(:,i)=[md(i+1:length(md));md(length(md))*ones(i,1)];
   end
end

% se hacen los vectores de entrenamiento desde y(t) hasta y(t+ph)
for j=1:ph
y(:,j)=[c(j+1:length(c));c(length(c))*ones(j,1)];
end

if ch==0
   inputs=[md c];
else
   inputs=[md m c];
end

outputs=y;

% Lo siguiente es igual a lo anterior pero para la validacion cruzada


if ch==0  % se verifica el horizonte de entradas sea distinto de cero.
   m_2=0;
else
   % se hacen los vectores para el entrenamiento desde m(t) hasta m(t+ch)
   for i=1:ch-1
   m_2(:,i)=[md_2(i+1:length(md_2));md_2(length(md_2))*ones(i,1)];
   end
end
```

```
% se hacen los vectores de entrenamiento desde y(t) hasta y(t+ph)
for j=1:ph
y(:,j)=[c_2(j+1:length(c_2));c_2(length(c_2))*ones(j,1)];
end

if ch==0
   inputs_2=[md_2 c_2];
else
   inputs_2=[md_2 m_2 c_2];
end

outputs_2=y;


[Wh,Bh,Wo,Bo,Ymin,SSEtrain]=entrenamiento2(inputs,outputs(:,1),nodes,iter,inputs_2,out
puts_2(:,1));

ie=size(inputs,2);                    %identifies number of input elements
l=size(inputs,1);                     %identifies training set dimension
%Include bias input
inputs(:,ie+1)=ones(l,1);
W=[Wh;Bh];
Wout=[Wo;Bo];

clear Wo Bo

% se calculan los pesos de salida para cada una de las neuronas por
% separado utilizando minimos cuadrados igualmente.

for k=2:ph

Z=inputs*W;    %net for each node in the hidden layer
V=tansig(Z);       %output for each node in the hidden layer

%Include bias input
V(:,nodes+1)=ones(l,1);        %nodes+1 input elements
Wout(:,k)=inv(V'*V)*V'*outputs(:,k);       %Least squares method  ojo!!!!!!!!!

end

% para el caso SISO unicamente por el momento
Z=inputs*W;
V=tansig(Z);
V(:,nodes+1)=ones(l,1);
P=inv(V'*V);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Wo=Wout(1:nodes,:);
Bo=Wout(nodes+1,:);
```

Interfaz gráfica de usuario para la estrategia sobre el reactor neutralizador de pH

```
function varargout = interfaz2(varargin)
% INTERFAZ2 M-file for interfaz2.fig
%      INTERFAZ2, by itself, creates a new INTERFAZ2 or raises the existing
%      singleton*.
%
%      H = INTERFAZ2 returns the handle to a new INTERFAZ2 or the handle to
%      the existing singleton*.
%
%      INTERFAZ2('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in INTERFAZ2.M with the given input
arguments.
%
%      INTERFAZ2('Property','Value',...) creates a new INTERFAZ2 or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before interfaz2_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to interfaz2_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.

% Edit the above text to modify the response to help interfaz2

% Last Modified by GUIDE v2.5 08-Jun-2006 14:37:00

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @interfaz2_OpeningFcn, ...
                   'gui_OutputFcn',  @interfaz2_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
```

```
else
   gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before interfaz2 is made visible.
function interfaz2_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to interfaz2 (see VARARGIN)

% Choose default command line output for interfaz2
handles.output = hObject;
%%%%%%%%%Valores iniciales para sintonizacion dmc y
PID%%%%%%%%%%%%%%
handles.tiempo=10500;
handles.T1=0;
handles.np=0;
handles.nu=0;
handles.lambda=0;
handles.gamma1=0;
handles.at='Yes';
handles.at2='Yes';
handles.Kc=0;
handles.ti=0;
handles.td=0;
handles.alpha1=0.1;
handles.ftc=3;
%%%%%%%%%%%%%%Valores iniciales para
identificacion%%%%%%%%%%%
handles.dm=-10;
%%%%%%%%%%Valores iniciales para modelos de
referencia%%%%%%%%%%%%
handles.z1=0.9;
handles.fts=2;
handles.mo=2;
%%%%%%%%%%%Valores iniciales para entrenamiento de la
RNA%%%%%%%%%%
handles.nodes=7;
handles.iter=100;
%%%%%%%%%%Valores Iniciales del SPEA
II%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
handles.pc=0.9;
handles.pm=0.2;
handles.p=40;
handles.p1=20;
handles.n=7;
handles.lmin=10;
handles.lmax=10;
handles.gmin=10;
handles.gmax=2;
handles.mdu=7;
handles.rph=10;

pc=handles.pc;
pm=handles.pm;
p=handles.p;
p1=handles.p1;
n=handles.n;
lmin=handles.lmin;
lmax=handles.lmax;
gmin=handles.gmin;
gmax=handles.gmax;
mdu=handles.mdu;
rph=handles.rph;

save parametros_spea pc pm n p p1 lmin lmax gmin gmax mdu rph

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes interfaz2 wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = interfaz2_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
```

```
function DM_Callback(hObject, eventdata, handles)
% hObject    handle to DM (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of DM as text
%        str2double(get(hObject,'String')) returns contents of DM as a double

handles.dm=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function DM_CreateFcn(hObject, eventdata, handles)
% hObject    handle to DM (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end




function KP_Callback(hObject, eventdata, handles)
% hObject    handle to KP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of KP as text
%        str2double(get(hObject,'String')) returns contents of KP as a double


% --- Executes during object creation, after setting all properties.
function KP_CreateFcn(hObject, eventdata, handles)
% hObject    handle to KP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function TAO_Callback(hObject, eventdata, handles)
% hObject    handle to TAO (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TAO as text
%        str2double(get(hObject,'String')) returns contents of TAO as a double


% --- Executes during object creation, after setting all properties.
function TAO_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TAO (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function TO_Callback(hObject, eventdata, handles)
% hObject    handle to TO (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TO as text
%        str2double(get(hObject,'String')) returns contents of TO as a double


% --- Executes during object creation, after setting all properties.
function TO_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TO (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function T_Callback(hObject, eventdata, handles)
% hObject    handle to T (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of T as text
%        str2double(get(hObject,'String')) returns contents of T as a double
handles.T=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function T_CreateFcn(hObject, eventdata, handles)
% hObject    handle to T (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function NP_Callback(hObject, eventdata, handles)
% hObject    handle to NP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of NP as text
%        str2double(get(hObject,'String')) returns contents of NP as a double
handles.np=str2double(get(hObject,'String'));
```

```
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function NP_CreateFcn(hObject, eventdata, handles)
% hObject    handle to NP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function NU_Callback(hObject, eventdata, handles)
% hObject    handle to NU (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of NU as text
%        str2double(get(hObject,'String')) returns contents of NU as a double
handles.nu=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function NU_CreateFcn(hObject, eventdata, handles)
% hObject    handle to NU (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function LAMBDA_Callback(hObject, eventdata, handles)
% hObject    handle to LAMBDA (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of LAMBDA as text
%        str2double(get(hObject,'String')) returns contents of LAMBDA as a double
handles.lambda=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function LAMBDA_CreateFcn(hObject, eventdata, handles)
% hObject    handle to LAMBDA (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function GAMMA_Callback(hObject, eventdata, handles)
% hObject    handle to GAMMA (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of GAMMA as text
%        str2double(get(hObject,'String')) returns contents of GAMMA as a double
handles.gamma1=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function GAMMA_CreateFcn(hObject, eventdata, handles)
% hObject    handle to GAMMA (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
```

```
end

function Z1_Callback(hObject, eventdata, handles)
% hObject    handle to Z1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Z1 as text
%        str2double(get(hObject,'String')) returns contents of Z1 as a double
handles.z1=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function Z1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Z1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function FTS_Callback(hObject, eventdata, handles)
% hObject    handle to FTS (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of FTS as text
%        str2double(get(hObject,'String')) returns contents of FTS as a double
handles.fts=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function FTS_CreateFcn(hObject, eventdata, handles)
% hObject    handle to FTS (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
```

```
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes on selection change in MO.
function MO_Callback(hObject, eventdata, handles)
% hObject    handle to MO (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns MO contents as cell array
%        contents{get(hObject,'Value')} returns selected item from MO
contents = get(hObject,'String');
handles.mo=str2num(contents{get(hObject,'Value')});
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function MO_CreateFcn(hObject, eventdata, handles)
% hObject    handle to MO (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes on selection change in AT.
function AT_Callback(hObject, eventdata, handles)
% hObject    handle to AT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns AT contents as cell array
%        contents{get(hObject,'Value')} returns selected item from AT
contents = get(hObject,'String');
handles.at=contents{get(hObject,'Value')};
```

```
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function AT_CreateFcn(hObject, eventdata, handles)
% hObject    handle to AT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function NODES_Callback(hObject, eventdata, handles)
% hObject    handle to NODES (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of NODES as text
%        str2double(get(hObject,'String')) returns contents of NODES as a double
handles.nodes=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function NODES_CreateFcn(hObject, eventdata, handles)
% hObject    handle to NODES (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function ITER_Callback(hObject, eventdata, handles)
% hObject    handle to ITER (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ITER as text
%        str2double(get(hObject,'String')) returns contents of ITER as a double
handles.iter=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function ITER_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ITER (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function ERROR_Callback(hObject, eventdata, handles)
% hObject    handle to ERROR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ERROR as text
%        str2double(get(hObject,'String')) returns contents of ERROR as a double


% --- Executes during object creation, after setting all properties.
function ERROR_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ERROR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
% --- Executes on button press in initial.
function initial_Callback(hObject, eventdata, handles)
% hObject    handle to initial (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[handles.Kp,handles.tao,handles.to,handles.T1,handles.np,handles.nu,handles.lambda
,...

handles.gamma1,handles.SSEtrain,handles.Kc,handles.ti,handles.td,handles.alpha1]=.
..
   initial_training1(handles.dm,handles.T1,handles.np,handles.nu,...
   handles.lambda,handles.gamma1,handles.at,handles.z1,handles.fts,...
   handles.mo,handles.nodes,handles.iter,handles,handles.at2,handles.Kc,...
   handles.ti,handles.td,handles.alpha1,handles.ftc);
guidata(hObject, handles);

%colocacion de los números en la interfaz2%%%%%%%%%%

% identificaion FOPDT
set(handles.KP,'String',num2str(handles.Kp));
set(handles.TAO,'String',num2str(handles.tao));
set(handles.TO,'String',num2str(handles.to));

%Sintonizacion DMC
set(handles.T,'String',num2str(handles.T1));
set(handles.NP,'String',num2str(handles.np));
set(handles.NU,'String',num2str(handles.nu));
set(handles.LAMBDA,'String',num2str(handles.lambda));
set(handles.GAMMA,'String',num2str(handles.gamma1));

%Sintonizacion PID
set(handles.KC,'String',num2str(handles.Kc));
set(handles.TI,'String',num2str(handles.ti));
set(handles.TD,'String',num2str(handles.td));
set(handles.ALPHA1,'String',num2str(handles.alpha1));

%RANWNN
set(handles.ERROR,'String',num2str(handles.SSEtrain));
msgbox('Training Finish','Initial Training','help')
```

```
function P_Callback(hObject, eventdata, handles)
% hObject    handle to P (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of P as text
%        str2double(get(hObject,'String')) returns contents of P as a double
handles.p=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function P_CreateFcn(hObject, eventdata, handles)
% hObject    handle to P (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function P1_Callback(hObject, eventdata, handles)
% hObject    handle to P1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of P1 as text
%        str2double(get(hObject,'String')) returns contents of P1 as a double
handles.p1=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function P1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to P1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
```

```
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function N_Callback(hObject, eventdata, handles)
% hObject    handle to N (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of N as text
%        str2double(get(hObject,'String')) returns contents of N as a double
handles.n=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function N_CreateFcn(hObject, eventdata, handles)
% hObject    handle to N (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function PM_Callback(hObject, eventdata, handles)
% hObject    handle to PM (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of PM as text
%        str2double(get(hObject,'String')) returns contents of PM as a double
handles.pm=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function PM_CreateFcn(hObject, eventdata, handles)
% hObject    handle to PM (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function LMIN_Callback(hObject, eventdata, handles)
% hObject    handle to LMIN (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of LMIN as text
%        str2double(get(hObject,'String')) returns contents of LMIN as a double
handles.lmin=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function LMIN_CreateFcn(hObject, eventdata, handles)
% hObject    handle to LMIN (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function LMAX_Callback(hObject, eventdata, handles)
% hObject    handle to LMAX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of LMAX as text
```

```
%        str2double(get(hObject,'String')) returns contents of LMAX as a double
handles.lmax=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function LMAX_CreateFcn(hObject, eventdata, handles)
% hObject    handle to LMAX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function GMIN_Callback(hObject, eventdata, handles)
% hObject    handle to GMIN (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of GMIN as text
%        str2double(get(hObject,'String')) returns contents of GMIN as a double
handles.gmin=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function GMIN_CreateFcn(hObject, eventdata, handles)
% hObject    handle to GMIN (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
function GMAX_Callback(hObject, eventdata, handles)
% hObject    handle to GMAX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of GMAX as text
%        str2double(get(hObject,'String')) returns contents of GMAX as a double
handles.gmax=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function GMAX_CreateFcn(hObject, eventdata, handles)
% hObject    handle to GMAX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function MDU_Callback(hObject, eventdata, handles)
% hObject    handle to MDU (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of MDU as text
%        str2double(get(hObject,'String')) returns contents of MDU as a double
handles.mdu=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function MDU_CreateFcn(hObject, eventdata, handles)
% hObject    handle to MDU (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function PC_Callback(hObject, eventdata, handles)
% hObject    handle to PC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of PC as text
%        str2double(get(hObject,'String')) returns contents of PC as a double
handles.pc=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function PC_CreateFcn(hObject, eventdata, handles)
% hObject    handle to PC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function RPH_Callback(hObject, eventdata, handles)
% hObject    handle to RPH (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of RPH as text
%        str2double(get(hObject,'String')) returns contents of RPH as a double
handles.rph=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function RPH_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to RPH (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes on button press in DMC_START.
function DMC_START_Callback(hObject, eventdata, handles)
% hObject    handle to DMC_START (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global G T gamma1 lambda np nu
myopts = simset('DstWorkspace ', 'current','SrcWorkspace','current');
sim('model_dmc',handles.tiempo,myopts)
save resultados_dmc cont_dmc salida_real_dmc em_dmc salida_dmc
salida_ruido_dmc iae_dmc
axes(handles.axes1)
plot(tout,salida_real_dmc)
title('Output pH')
xlabel('Time (s)')
ylabel('pH')
axes(handles.axes9)
plot(tout,salida_dmc)
title('Controlled variable without noise')
xlabel('Time (s)')
ylabel('%TO')
axes(handles.axes8)
plot(tout,salida_ruido_dmc)
title('Controlled variable with noise')
xlabel('Time (s)')
ylabel('%TO')
axes(handles.axes10)
plot(tout,cont_dmc)
title('Controller output')
xlabel('Time (s)')
ylabel('%CO')
axes(handles.axes11)
plot(tout,em_dmc)
```

```
title('Model error')
xlabel('Time (s)')
ylabel('%TO')
set(handles.IAE,'String',num2str(max(iae_dmc(:,1))))
msgbox('Simulation Terminated','DMC','help')


% --- Executes on button press in NEDMC_START.
function NEDMC_START_Callback(hObject, eventdata, handles)
% hObject    handle to NEDMC_START (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global G T gamma1 lambda np nu
pc=handles.pc;
pm=handles.pm;
p=handles.p;
p1=handles.p1;
n=handles.n;
lmin=handles.lmin;
lmax=handles.lmax;
gmin=handles.gmin;
gmax=handles.gmax;
mdu=handles.mdu;
rph=handles.rph;

save parametros_spea pc pm n p p1 lmin lmax gmin gmax mdu rph

load parametros_dmc
myopts = simset('DstWorkspace ', 'current','SrcWorkspace','current');
sim('dmc_adap_mod_ref',handles.tiempo,myopts)
save resultados_dmc_adap cont_dmc_adap salida_real_dmc_adap em_dmc_adap
salida_dmc_adap salida_dmc_adap_red salida_ruido_dmc_adap iae_dmc_adap
axes(handles.axes12)
plot(tout,salida_real_dmc_adap)
title('Output pH')
xlabel('Time (s)')
ylabel('pH')
axes(handles.axes15)
plot(tout,salida_dmc_adap)
title('Controlled variable without noise')
xlabel('Time (s)')
ylabel('%TO')
axes(handles.axes14)
plot(tout,salida_ruido_dmc_adap)
title('Controlled variable with noise')
```

```
xlabel('Time (s)')
ylabel('%TO')
axes(handles.axes16)
plot(tout,cont_dmc_adap)
title('Controller output')
xlabel('Time (s)')
ylabel('%CO')
axes(handles.axes17)
plot(tout,em_dmc_adap(:,1))
title('Model error')
xlabel('Time (s)')
ylabel('%TO')
axes(handles.axes18)
plot(tout,salida_dmc_adap_red)
title('RNA output Vs Process output')
xlabel('Time (s)')
ylabel('%TO')

algo=dlmread('factores.txt');
n2=length(algo(:,1));
tiempo=1:T:n2*T;
lambda1=algo(:,1);
gamma11=algo(:,2);
clear algo
axes(handles.axes19)
stem(tiempo,lambda1);
title('Lambda each sample time')
xlabel('Time (s)')
ylabel('Lambda')
axes(handles.axes21)
stem(tiempo,gamma11,'y');
title('Gamma each sample time')
xlabel('Time (s)')
ylabel('Gamma')

set(handles.IAE_NEDMC,'String',num2str(max(iae_dmc_adap)))
msgbox('Simulation Terminated','NEDMC','help')

% --- Executes on button press in PLOTS_DMC.
function PLOTS_DMC_Callback(hObject, eventdata, handles)
% hObject    handle to PLOTS_DMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.Panel1,'Visible','off')
set(handles.Panel2,'Visible','off')
```

```
set(handles.Panel3,'Visible','off')
set(handles.DMC_RES,'Visible','on')
set(handles.BACK,'Visible','on')


% --- Executes on button press in PLOTS_NEDMC.
function PLOTS_NEDMC_Callback(hObject, eventdata, handles)
% hObject    handle to PLOTS_NEDMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.Panel1,'Visible','off')
set(handles.Panel2,'Visible','off')
set(handles.Panel3,'Visible','off')
set(handles.NEDMC_RES,'Visible','on')
set(handles.BACK,'Visible','on')

% --- Executes on button press in BACK2.
function BACK_Callback(hObject, eventdata, handles)
% hObject    handle to BACK2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.Panel1,'Visible','on')
set(handles.Panel2,'Visible','on')
set(handles.Panel3,'Visible','on')
set(handles.DMC_RES,'Visible','off')
set(handles.NEDMC_RES,'Visible','off')
set(handles.PID_RES,'Visible','off')
set(handles.BACK,'Visible','off')

function IAE_Callback(hObject, eventdata, handles)
% hObject    handle to IAE (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of IAE as text
%        str2double(get(hObject,'String')) returns contents of IAE as a double


% --- Executes during object creation, after setting all properties.
function IAE_CreateFcn(hObject, eventdata, handles)
% hObject    handle to IAE (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
```

```
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end




function IAE2_Callback(hObject, eventdata, handles)
% hObject    handle to IAE2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of IAE2 as text
%        str2double(get(hObject,'String')) returns contents of IAE2 as a double


% --- Executes during object creation, after setting all properties.
function IAE2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to IAE2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end




function IAE_NEDMC_Callback(hObject, eventdata, handles)
% hObject    handle to IAE_NEDMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of IAE_NEDMC as text
%        str2double(get(hObject,'String')) returns contents of IAE_NEDMC as a double
```

```
% --- Executes during object creation, after setting all properties.
function IAE_NEDMC_CreateFcn(hObject, eventdata, handles)
% hObject    handle to IAE_NEDMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function IAE2_NEDMC_Callback(hObject, eventdata, handles)
% hObject    handle to IAE2_NEDMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of IAE2_NEDMC as text
%        str2double(get(hObject,'String')) returns contents of IAE2_NEDMC as a
double


% --- Executes during object creation, after setting all properties.
function IAE2_NEDMC_CreateFcn(hObject, eventdata, handles)
% hObject    handle to IAE2_NEDMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
% --- Executes on button press in NEXT2.
function NEXT_Callback(hObject, eventdata, handles)
% hObject    handle to NEXT2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.NEDMC_RES,'Visible','off')
set(handles.BACK,'Visible','off')
set(handles.NEDMC_RES2,'Visible','on')


% --- Executes on button press in BACK2.
function BACK2_Callback(hObject, eventdata, handles)
% hObject    handle to BACK2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

set(handles.NEDMC_RES,'Visible','on')
set(handles.BACK,'Visible','on')
set(handles.NEDMC_RES2,'Visible','off')


% --- Executes on button press in NEXT2.
function NEXT2_Callback(hObject, eventdata, handles)
% hObject    handle to NEXT2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

set(handles.NEDMC_RES2,'Visible','off')
set(handles.NEDMC_RES3,'Visible','on')


% --- Executes on button press in BACK3.
function BACK3_Callback(hObject, eventdata, handles)
% hObject    handle to BACK3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

set(handles.NEDMC_RES2,'Visible','on')
set(handles.NEDMC_RES3,'Visible','off')


% --- Executes on button press in PLOTS_TRAINING.
function PLOTS_TRAINING_Callback(hObject, eventdata, handles)
% hObject    handle to PLOTS_TRAINING (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
set(handles.Panel1,'Visible','off')
set(handles.Panel2,'Visible','off')
set(handles.Panel3,'Visible','off')
set(handles.T_RES,'Visible','on')

% --- Executes on button press in BACKT.
function BACKT_Callback(hObject, eventdata, handles)
% hObject    handle to BACKT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.Panel1,'Visible','on')
set(handles.Panel2,'Visible','on')
set(handles.Panel3,'Visible','on')
set(handles.T_RES,'Visible','off')



function KC_Callback(hObject, eventdata, handles)
% hObject    handle to KC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of KC as text
%        str2double(get(hObject,'String')) returns contents of KC as a double
handles.Kc=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function KC_CreateFcn(hObject, eventdata, handles)
% hObject    handle to KC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
function TI_Callback(hObject, eventdata, handles)
% hObject    handle to TI (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TI as text
%        str2double(get(hObject,'String')) returns contents of TI as a double
handles.ti=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function TI_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TI (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function TD_Callback(hObject, eventdata, handles)
% hObject    handle to TD (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TD as text
%        str2double(get(hObject,'String')) returns contents of TD as a double
handles.td=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function TD_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TD (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
```

```
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function ALPHA1_Callback(hObject, eventdata, handles)
% hObject    handle to ALPHA1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ALPHA1 as text
%        str2double(get(hObject,'String')) returns contents of ALPHA1 as a double
handles.alpha1=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function ALPHA1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ALPHA1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



% --- Executes on selection change in AT2.
function AT2_Callback(hObject, eventdata, handles)
% hObject    handle to AT2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns AT2 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from AT2
contents = get(hObject,'String');
handles.at2=contents{get(hObject,'Value')};
guidata(hObject, handles);
```

```
% --- Executes during object creation, after setting all properties.
function AT2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to AT2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end




% --- Executes on button press in PID_START.
function PID_START_Callback(hObject, eventdata, handles)
% hObject    handle to PID_START (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global T Kc ti td alpha1
load parametros_pid
myopts = simset('DstWorkspace ', 'current','SrcWorkspace','current');
sim('model_pid_digital',handles.tiempo,myopts)
save resultados_pi_d cont_pi_d salida_real_pi_d salida_pi_d salida_ruido_pi_d
iae_pi_d
axes(handles.axes38)
plot(tout,salida_real_pi_d)
title('Output pH')
xlabel('Time (s)')
ylabel('pH')
axes(handles.axes41)
plot(tout,cont_pi_d)
title('Controller output')
xlabel('Time (s)')
ylabel('%CO')
axes(handles.axes39)
plot(tout,salida_ruido_pi_d)
title('Controlled variable with noise')
xlabel('Time (s)')
ylabel('%TO')
axes(handles.axes40)
```

```
plot(tout,salida_pi_d)
title('Controlled variable without noise')
xlabel('Time (s)')
ylabel('%TO')
set(handles.IAE_PID,'String',num2str(max(iae_pi_d)))
msgbox('Simulation Terminated','PID','help')

% --- Executes on button press in PLOTS_PID.
function PLOTS_PID_Callback(hObject, eventdata, handles)
% hObject    handle to PLOTS_PID (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.Panel1,'Visible','off')
set(handles.Panel2,'Visible','off')
set(handles.Panel3,'Visible','off')
set(handles.PID_RES,'Visible','on')
set(handles.BACK,'Visible','on')


function IAE_PID_Callback(hObject, eventdata, handles)
% hObject    handle to IAE_PID (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of IAE_PID as text
%        str2double(get(hObject,'String')) returns contents of IAE_PID as a double


% --- Executes during object creation, after setting all properties.
function IAE_PID_CreateFcn(hObject, eventdata, handles)
% hObject    handle to IAE_PID (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
% --- Executes on button press in OPEN_PID.
function OPEN_PID_Callback(hObject, eventdata, handles)
% hObject    handle to OPEN_PID (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
open('model_pid_digital.mdl')

% --- Executes on button press in OPEN_DMC.
function OPEN_DMC_Callback(hObject, eventdata, handles)
% hObject    handle to OPEN_DMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
open('model_dmc.mdl')

% --- Executes on button press in OPEN_NEDMC.
function OPEN_NEDMC_Callback(hObject, eventdata, handles)
% hObject    handle to OPEN_NEDMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
open('dmc_adap_mod_ref.mdl')
```

```
function  m =  dmc_mult(R,C,time);

global m mbar predvect T lambda np nu G i gamma1

% inicializacion
if time==0
   load parametros_dmc_mult;
   mbar=52;
   m=zeros(1,4);
   m(1)=mbar;
   m(2)=mbar;
   m(3:4)=0;
   predvect=zeros(np,2);
   predvect(:,1)=C(1)*ones(np,1);
   predvect(:,2)=C(2)*ones(np,1);
   i=time;
end
   if rem(time,T)==0 && time > i
      em(1)=C(1)-predvect(1,1);
      em(2)=C(2)-predvect(1,2);

      predvect_u(:,1)=[predvect(2:np,1);predvect(np,1)]+em(1);
      predvect_u(:,2)=[predvect(2:np,2);predvect(np,2)]+em(2);

      E(:,1)=R(1)*ones(np,1)-predvect_u(:,1);
      E(:,2)=R(2)*ones(np,1)-predvect_u(:,2);
      Et=[E(:,1);E(:,2)];

      gamma2=gamma1(1)*eye(np);
      gamma2(np+1:2*np,np+1:2*np)=gamma1(2)*eye(np);

      lambda2=lambda(1)*eye(nu);
      lambda2(nu+1:2*nu,nu+1:2*nu)=lambda(2)*eye(nu);

      Kc=inv(G'*gamma2*G+lambda2)*G'*gamma2;

      delta_u=Kc*Et;
      delta_u2(1)=delta_u(1);
      delta_u2(2)=delta_u(nu+1);

      if (m(1) + delta_u2(1)) > 100,
         delta_u2(1)=100-m(1);
      elseif (m(1) + delta_u2(1)) < 0,
         delta_u2(1) = -m(1);
      end
```

```
    if (m(2) + delta_u2(2)) > 100,
       delta_u2(2)=100-m(2);
    elseif (m(2) + delta_u2(2)) < 0,
       delta_u2(2) = -m(2);
    end
    m(1)=m(1)+delta_u2(1);
    m(2)=m(2)+delta_u2(2);
    m(3:4)=em;

    predvect(:,1)=G(1:np,1:nu)*[delta_u2(1);zeros(nu-1,1)]+...
       G(1:np,nu+1:2*nu)*[delta_u2(2);zeros(nu-1,1)]+predvect_u(:,1);

    predvect(:,2)=G(np+1:2*np,1:nu)*[delta_u2(1);zeros(nu-1,1)]+...
       G(np+1:2*np,nu+1:2*nu)*[delta_u2(2);zeros(nu-1,1)]+predvect_u(:,2);

    i=time;
else
       m=m;
end
```

Cálculo de las funciones objetivo para la estrategia MMO

```
function [fitness]=funciones_objetivo(trait,POP_SIZE,...
   y_act,y_ant,predvect_2,r_act,Rr,np_2,nu_2,G_2,m_2,Wh,Bh,Wo,Bo,delta_p,...
   mdu1,mdu2,rph)

%Se ejecuta el algoritmo de control DMC
em_2(1)=y_act(1)-predvect_2(1,1);
em_2(2)=y_act(2)-predvect_2(1,2);

predvect_u_2(:,1)=[predvect_2(2:np_2,1);predvect_2(np_2,1)]+em_2(1);
predvect_u_2(:,2)=[predvect_2(2:np_2,2);predvect_2(np_2,2)]+em_2(2);

E_2(:,1)=Rr(:,1)-predvect_u_2(:,1);
E_2(:,2)=Rr(:,2)-predvect_u_2(:,2);

Et_2=[E_2(:,1);E_2(:,2)];

for chrom_number = 1:POP_SIZE,

lambda_2=[trait(1:2,chrom_number)];
gamma_2=trait(3:4,chrom_number);

gamma_3=gamma_2(1)*eye(np_2);
gamma_3(np_2+1:2*np_2,np_2+1:2*np_2)=gamma_2(2)*eye(np_2);

lambda_3=lambda_2(1)*eye(nu_2);
lambda_3(nu_2+1:2*nu_2,nu_2+1:2*nu_2)=lambda_2(2)*eye(nu_2);

Kc_2=inv(G_2'*gamma_3*G_2+lambda_3)*G_2'*gamma_3;
delta_u_21=Kc_2*Et_2;

m_p=m_2;

%se saturan las predicciones del DMC para los deltas de u de modo  que
%cuando se prediga con base en ellos no arroje informaciones erróneas o se
%vaya a infinito la predicción
cnt2=1;
for cnt=1:length(delta_u_21)
   if cnt<=nu_2
      if (m_p(cnt,1) + delta_u_21(cnt)) > 100,
        delta_u_21(cnt)=100-m_p(cnt,1);
        elseif (m_p(cnt,1) + delta_u_21(cnt)) < 0,
        delta_u_21(cnt) = -m_p(cnt,1);
      end
```

```
        m_p(cnt+1,1)=delta_u_21(cnt)+m_p(cnt,1);
    else
        if (m_p(cnt2,2) + delta_u_21(cnt)) > 100,
          delta_u_21(cnt)=100-m_p(cnt2,2);
          elseif (m_p(cnt2,2) + delta_u_21(cnt)) < 0,
          delta_u_21(cnt) = -m_p(cnt2,2);
        end
       m_p(cnt2+1,2)=delta_u_21(cnt)+m_p(cnt2,2);
       cnt2=cnt2+1;
    end
end

predvect(:,1)=G_2(1:np_2,1:nu_2)*[delta_u_21(1);zeros(nu_2-1,1)]+...
    G_2(1:np_2,nu_2+1:2*nu_2)*[delta_u_21(2);zeros(nu_2-1,1)]+predvect_u_2(:,1);

predvect(:,2)=G_2(np_2+1:2*np_2,1:nu_2)*[delta_u_21(1);zeros(nu_2-1,1)]+...
    G_2(np_2+1:2*np_2,nu_2+1:2*nu_2)*[delta_u_21(2);zeros(nu_2-
1,1)]+predvect_u_2(:,2);

epred2(:,1)=Rr(:,1)-predvect(:,1);
epred2(:,2)=Rr(:,2)-predvect(:,2);

clear cnt


%Se simula la red neuronal
inputs3=[];
lim=rph;
for i=1:lim
    temp=[m_p(i,1) m_p(i,2)];
    inputs3=[inputs3 temp];
end

[y_pred]=nn_sim_auto_mult4([inputs3 y_act(1) y_act(2)],...
     Wh,Bh,Wo,Bo,lim);

y_pred(:,1)=y_pred(:,1)+delta_p(1);
y_pred(:,2)=y_pred(:,2)+delta_p(2);

% se calcula el error de prediccion
e_pred(:,1)=Rr(1:lim,1)-y_pred(:,1);
e_pred(:,2)=Rr(1:lim,2)-y_pred(:,2);


    if (abs(delta_u_21(1)) <=mdu1) && (abs(delta_u_21(nu_2+1)) <= mdu2)
```

```
        fitness(chrom_number,1)=sumsqr(e_pred(:,1));
        fitness(chrom_number,2)=sumsqr(e_pred(:,2));
        fitness(chrom_number,3)=sumsqr(delta_u_21(1:nu_2));
        fitness(chrom_number,4)=sumsqr(delta_u_21(nu_2+1:2*nu_2));
        fitness(chrom_number,5)=sumsqr(epred2(:,1));
        fitness(chrom_number,6)=sumsqr(epred2(:,2));


    else

        fitness(chrom_number,1)=1;
        fitness(chrom_number,2)=1;
        fitness(chrom_number,3)=1;
        fitness(chrom_number,4)=1;
        fitness(chrom_number,5)=1;
        fitness(chrom_number,6)=1;

    end


end

% Normalizacion de los fitness
no_ones=find(fitness(:,1)~=1);
max_fit=max(fitness(no_ones,1));
fitness(no_ones,1)=fitness(no_ones,1)./max_fit;
clear no_ones max_fit

no_ones=find(fitness(:,2)~=1);
max_fit=max(fitness(no_ones,2));
fitness(no_ones,2)=fitness(no_ones,2)./max_fit;
clear no_ones max_fit

no_ones=find(fitness(:,3)~=1);
max_fit=max(fitness(no_ones,3));
fitness(no_ones,3)=fitness(no_ones,3)./max_fit;
clear no_ones max_fit

no_ones=find(fitness(:,4)~=1);
max_fit=max(fitness(no_ones,4));
fitness(no_ones,4)=fitness(no_ones,4)./max_fit;
clear no_ones max_fit

no_ones=find(fitness(:,5)~=1);
```

```
max_fit=max(fitness(no_ones,5));
fitness(no_ones,5)=fitness(no_ones,5)./max_fit;
clear no_ones max_fit

no_ones=find(fitness(:,6)~=1);
max_fit=max(fitness(no_ones,6));
fitness(no_ones,6)=fitness(no_ones,6)./max_fit;
```

Entrenamiento inicial para la estrategia MIMO

```
function [Kp,tao,to,T,np,nu,lambda,gamma1,SSEtrain]=...
initial_training(dm11,dm22,T,np,nu,lambda,gamma1,at,z1,z2,fts1,fts2,mo1,mo2,nodes,iter,
handles)
% Identificación del proceso como FOPDT
h2 = waitbar(0,'FOPDT Identification');
waitbar(1/4,h2)
myopts = simset('DstWorkspace ', 'current','SrcWorkspace','current');
dm1=dm11;
dm2=0.0;
%open('model_ident_fopdt_mult')
sim('model_ident_fopdt_mult.mdl',2000,myopts)

%% ojo el algoritmo hay que revisarlo (interpolación) porque esta muy
%% dependiente del tiempo del step de entrada.
%Kp(1)= Kp11
%Kp(3)= Kp21
%Ojo en el futuro hay que incluir el cálculo de la matriz de ganancias
%relativas de manera que se escojan las variables manipuladas y controladas
%correctamente
[Kp(1,1),tao(1,1),to(1,1)]=ajuste_fit3(c1,tout,dm1);
[Kp(2,1),tao(2,1),to(2,1)]=ajuste_fit3(c2,tout,dm1);

dm1=0.0;
dm2=dm22;
sim('model_ident_fopdt_mult.mdl',2000,myopts)

[Kp(1,2),tao(1,2),to(1,2)]=ajuste_fit3(c1,tout,dm2);
[Kp(2,2),tao(2,2),to(2,2)]=ajuste_fit3(c2,tout,dm2);

open('modelx_mult.mdl')

set_param('modelx_mult/FOPDT-system/FOPDT-G11','m_bar',num2str(52))
set_param('modelx_mult/FOPDT-system/FOPDT-G11','c_bar',num2str(50.1786))
set_param('modelx_mult/FOPDT-system/FOPDT-G11','Kp',num2str(Kp(1,1)))
set_param('modelx_mult/FOPDT-system/FOPDT-G11','tao',num2str(tao(1,1)))
set_param('modelx_mult/FOPDT-system/FOPDT-G11','to',num2str(to(1,1)))

% ojo el valor inicial de salida de las funciones de transferencias
% cruzadas debe ser cero debido a que el valor inicial lo incluyen  la
% funcion de transferencia directa
set_param('modelx_mult/FOPDT-system/FOPDT-G12','m_bar',num2str(52))
set_param('modelx_mult/FOPDT-system/FOPDT-G12','c_bar',num2str(0))
set_param('modelx_mult/FOPDT-system/FOPDT-G12','Kp',num2str(Kp(1,2)))
```

```
set_param('modelx_mult/FOPDT-system/FOPDT-G12','tao',num2str(tao(1,2)))
set_param('modelx_mult/FOPDT-system/FOPDT-G12','to',num2str(to(1,2)))

set_param('modelx_mult/FOPDT-system/FOPDT-G21','m_bar',num2str(52))
set_param('modelx_mult/FOPDT-system/FOPDT-G21','c_bar',num2str(0))
set_param('modelx_mult/FOPDT-system/FOPDT-G21','Kp',num2str(Kp(2,1)))
set_param('modelx_mult/FOPDT-system/FOPDT-G21','tao',num2str(tao(2,1)))
set_param('modelx_mult/FOPDT-system/FOPDT-G21','to',num2str(to(2,1)))

set_param('modelx_mult/FOPDT-system/FOPDT-G22','m_bar',num2str(52))
set_param('modelx_mult/FOPDT-system/FOPDT-G22','c_bar',num2str(51.25))
set_param('modelx_mult/FOPDT-system/FOPDT-G22','Kp',num2str(Kp(2,2)))
set_param('modelx_mult/FOPDT-system/FOPDT-G22','tao',num2str(tao(2,2)))
set_param('modelx_mult/FOPDT-system/FOPDT-G22','to',num2str(to(2,2)))

dm1=dm11;
dm2=0.0;
sim('modelx_mult.mdl',2000,myopts)
axes(handles.axes34)
plot(tout,salida1)
title('FOPDT system output 1 Vs Real system output 1')
xlabel('Time (s)')
ylabel('%TO')
axes(handles.axes35)
plot(tout,salida2)
title('FOPDT system output 2 Vs Real system output 2')
xlabel('Time (s)')
ylabel('%TO')
close_system('modelx_mult',1)



waitbar(2/4,h2,'DMC tuning')
if at=='Yes'
[T,np,nu,lambda]=sinto_dmc_mult(Kp,tao,to);
gamma1=[1;1];
end
% Con el período de muestreo obtenido anteriormente se obtiene la matriz
% dinámica

clear c tout  % se limpian las variables para que no haya incovenientes

% se simula teniendo en cuenta que le hace falta el parámetro T
% obtenido en el paso anterior
dm1=dm11;
```

```
dm2=0.0;
sim('model2_mult.mdl',2000,myopts)
G11=matrix_dinamica(c1,m1,np,nu);
G21=matrix_dinamica(c2,m1,np,nu);

dm1=0.0;
dm2=dm22;
sim('model2_mult.mdl',2000,myopts)
G12=matrix_dinamica(c1,m2,np,nu);
G22=matrix_dinamica(c2,m2,np,nu);
% se concatena para generar la matriz grande
G=[G11 G12;G21 G22];

save parametros_dmc_mult G lambda np nu T gamma1

% Se escoge la respuesta requerida para el modelo de referencia de modo que
% se pueda obtener su vector igualmente.
waitbar(3/4,h2,'Reference Models')

Ts1=tao(1,1)*fts1;
Ts2=tao(2,2)*fts2;
dm1=10;
dm2=10;

[num,den,polos]=polo_lazo_cerrado(z1,Ts1,T,mo1);

% se limpian las variables para que no haya inconvenientes con la
% simulación
clear tout m
%open('modelo_referencia.mdl')
sim('modelo_referencia.mdl',2000,myopts)
R1=matrix_dinamica(mod_ref,m,np,1);
axes(handles.axes36)
tiempo=1:T:length(R1)*T;
plot(tiempo,R1)
clear tiempo
title('Unit step response of the model reference 1')
xlabel('Time (s)')
ylabel('%TO')
open('DmcAdaptMult_mod_ref.mdl')
set_param('DmcAdaptMult_mod_ref/modelo_referencia/MR','Numerator',strcat('[',num2str
(num),']'));
set_param('DmcAdaptMult_mod_ref/modelo_referencia/MR','Denominator',strcat('[',num2
str(den),']'));
clear tout m mod_ref
```

```
[num,den,polos]=polo_lazo_cerrado(z2,Ts2,T,mo2);
sim('modelo_referencia2.mdl',2000,myopts)
set_param('DmcAdaptMult_mod_ref/modelo_referencia1/MR','Numerator',strcat('[',num2st
r(num),']'));
set_param('DmcAdaptMult_mod_ref/modelo_referencia1/MR','Denominator',strcat('[',num
2str(den),']'));
R2=matrix_dinamica(mod_ref,m,np,1);
axes(handles.axes37)
tiempo=1:T:length(R2)*T;
plot(tiempo,R2)
clear tiempo
title('Unit step response of the model reference 2')
xlabel('Time (s)')
ylabel('%TO')
close_system('DmcAdaptMult_mod_ref',1)
% La variable de salida del modelo de referencia debe ser "mod_ref" y la de
% entrada "m". Además los valores iniciales del step y de entrada y salida
% de la función de transferencia deben coincidir con los del proceso.
% El horizonte de predicción es uno porque se asume desconocida
% la real referencia y el horizonte de predicción es el mismo que para la
% matriz del proceso.



% Se guardan el valor del vector de referencia de modo que cuando se
% ejecute en simulink  el DMC ésta se encuentre guardada propiamente.
save referencia R1 R2

% Se generan los conjuntos de entrenamiento y validación para las ANN a
% partir de una serie de pruebas escalón al proceso en donde se mira su
% comportamiento en todo el rango de entrada desde 0 a 100% en pasos
% definidos a priori. La salida del modelo de simulación debe llamarse "c"
% y la entrada que se va a usar para el entrenamiento se debe  llamar "md."

% Conjunto de entrenamiento
waitbar(4/4,h2,'MIMO RAWNN Training')
clear m1 m2 c1 c2 tout
sim('data_train_mult.mdl',5*tao(1,1)*4,myopts)
m1_temp=m1;
c1_temp=c1;
m2_temp=m2;
c2_temp=c2;
clear m1 m2 c1 c2 tout
sim('data_train_mult2.mdl',5*tao(2,2)*4,myopts)
```

```
m1_train=[m1_temp;m1];
c1_train=[c1_temp;c1];
m2_train=[m2_temp;m2];
c2_train=[c2_temp;c2];
clear m1_temp m2_temp c1_temp c2_temp tout

% Conjunto de validación
clear m1 m2 c1 c2 tout
sim('data_val_mult.mdl',5*tao(1,1)*4,myopts)
m1_temp=m1;
c1_temp=c1;
m2_temp=m2;
c2_temp=c2;
clear m1 m2 c1 c2 tout
sim('data_val_mult2.mdl',5*tao(2,2)*4,myopts)
m1_val=[m1_temp;m1];
c1_val=[c1_temp;c1];
m2_val=[m2_temp;m2];
c2_val=[c2_temp;c2];
clear m1_temp m2_temp c1_temp c2_temp tout


% red para lazo1
[Wh,Bh,Wo,Bo,Ymin,SSEtrain,P]=train_rawn_mult3(m1_train,m2_train,c1_train,...
    c2_train,m1_val,m2_val,c1_val,c2_val,nodes,iter);

save weights_mult1 Wh Bh Wo Bo P;
clear Wh Bh Wo Bo P
close(h2)
```

Interfaz gráfica de usuario para la estrategia MIMO

```
function varargout = interfaz(varargin)
% INTERFAZ M-file for interfaz.fig
%      INTERFAZ, by itself, creates a new INTERFAZ or raises the existing
%      singleton*.
%
%      H = INTERFAZ returns the handle to a new INTERFAZ or the handle to
%      the existing singleton*.
%
%      INTERFAZ('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in INTERFAZ.M with the given input arguments.
%
%      INTERFAZ('Property','Value',...) creates a new INTERFAZ or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before interfaz_OpeningFunction gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to interfaz_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Copyright 2002-2003 The MathWorks, Inc.

% Edit the above text to modify the response to help interfaz

% Last Modified by GUIDE v2.5 17-Jun-2006 08:06:06

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @interfaz_OpeningFcn, ...
                   'gui_OutputFcn',  @interfaz_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
```

```
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before interfaz is made visible.
function interfaz_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to interfaz (see VARARGIN)

% Choose default command line output for interfaz
handles.output = hObject;
%%%%%%%%%Valores iniciales para sintonizacion dmc%%%%%%%%%%%%%
handles.tfinal=5500;
handles.T1=0;
handles.np=0;
handles.nu=0;
handles.lambda=zeros(1,2);
handles.gamma1=zeros(1,2);
handles.at='Yes';
%%%%%%%%%%%%%Valores iniciales para identificacion%%%%%%%%%%%%
handles.dm1=-10;
handles.dm22=10;
%%%%%%%%%Valores iniciales para modelos de referencia%%%%%%%%%%%
handles.z1=0.9;
handles.z2=0.9;
handles.fts1=2;
handles.fts2=2;
handles.mo1=2;
handles.mo2=2;
%%%%%%%%%%Valores iniciales para entrenamiento de la RNA%%%%%%%%%%
handles.nodes=10;
handles.iter=100;
%%%%%%%%%Valores Iniciales del SPEA
II%%%%%%%%%%%%%%%%%%%%%%%%%
handles.pc=0.9;
handles.pm=0.3;
handles.p=20;
handles.p1=10;
handles.n=7;
handles.lmin=5;
handles.lmax=5;
```

```
handles.gmin=5;
handles.gmax=2;
handles.mdu1=7;
handles.mdu2=7;
handles.rph=10;
handles.GT=[10 10 5 5];

pc=handles.pc;
pm=handles.pm;
p=handles.p;
p1=handles.p1;
n=handles.n;
lmin=handles.lmin;
lmax=handles.lmax;
gmin=handles.gmin;
gmax=handles.gmax;
mdu1=handles.mdu1;
mdu2=handles.mdu2;
rph=handles.rph;
GT=handles.GT;

save parametros_spea pc pm n p p1 lmin lmax gmin gmax mdu1 mdu2 rph GT

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes interfaz wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = interfaz_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



function DM1_Callback(hObject, eventdata, handles)
% hObject    handle to DM1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of DM1 as text
%        str2double(get(hObject,'String')) returns contents of DM1 as a double

handles.dm11=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function DM1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to DM1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function DM2_Callback(hObject, eventdata, handles)
% hObject    handle to DM2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of DM2 as text
%        str2double(get(hObject,'String')) returns contents of DM2 as a double
handles.dm22=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function DM2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to DM2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
```

```
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function KP11_Callback(hObject, eventdata, handles)
% hObject    handle to KP11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of KP11 as text
%        str2double(get(hObject,'String')) returns contents of KP11 as a double


% --- Executes during object creation, after setting all properties.
function KP11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to KP11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function KP12_Callback(hObject, eventdata, handles)
% hObject    handle to KP12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of KP12 as text
%        str2double(get(hObject,'String')) returns contents of KP12 as a double


% --- Executes during object creation, after setting all properties.
function KP12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to KP12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function KP21_Callback(hObject, eventdata, handles)
% hObject    handle to KP21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of KP21 as text
%        str2double(get(hObject,'String')) returns contents of KP21 as a double


% --- Executes during object creation, after setting all properties.
function KP21_CreateFcn(hObject, eventdata, handles)
% hObject    handle to KP21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function KP22_Callback(hObject, eventdata, handles)
% hObject    handle to KP22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of KP22 as text
%        str2double(get(hObject,'String')) returns contents of KP22 as a double


% --- Executes during object creation, after setting all properties.
```

```
function KP22_CreateFcn(hObject, eventdata, handles)
% hObject    handle to KP22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function TAO11_Callback(hObject, eventdata, handles)
% hObject    handle to TAO11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TAO11 as text
%        str2double(get(hObject,'String')) returns contents of TAO11 as a double


% --- Executes during object creation, after setting all properties.
function TAO11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TAO11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function TAO12_Callback(hObject, eventdata, handles)
% hObject    handle to TAO12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of TAO12 as text
%        str2double(get(hObject,'String')) returns contents of TAO12 as a double


% --- Executes during object creation, after setting all properties.
function TAO12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TAO12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function TAO21_Callback(hObject, eventdata, handles)
% hObject    handle to TAO21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TAO21 as text
%        str2double(get(hObject,'String')) returns contents of TAO21 as a double


% --- Executes during object creation, after setting all properties.
function TAO21_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TAO21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
function TAO22_Callback(hObject, eventdata, handles)
% hObject    handle to TAO22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TAO22 as text
%        str2double(get(hObject,'String')) returns contents of TAO22 as a double


% --- Executes during object creation, after setting all properties.
function TAO22_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TAO22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function TO11_Callback(hObject, eventdata, handles)
% hObject    handle to TO11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TO11 as text
%        str2double(get(hObject,'String')) returns contents of TO11 as a double


% --- Executes during object creation, after setting all properties.
function TO11_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TO11 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
```

```
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function TO12_Callback(hObject, eventdata, handles)
% hObject    handle to TO12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TO12 as text
%        str2double(get(hObject,'String')) returns contents of TO12 as a double


% --- Executes during object creation, after setting all properties.
function TO12_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TO12 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function TO21_Callback(hObject, eventdata, handles)
% hObject    handle to TO21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TO21 as text
%        str2double(get(hObject,'String')) returns contents of TO21 as a double


% --- Executes during object creation, after setting all properties.
function TO21_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TO21 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function TO22_Callback(hObject, eventdata, handles)
% hObject    handle to TO22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of TO22 as text
%        str2double(get(hObject,'String')) returns contents of TO22 as a double


% --- Executes during object creation, after setting all properties.
function TO22_CreateFcn(hObject, eventdata, handles)
% hObject    handle to TO22 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function T_Callback(hObject, eventdata, handles)
% hObject    handle to T (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of T as text
%        str2double(get(hObject,'String')) returns contents of T as a double
handles.T=str2double(get(hObject,'String'));
guidata(hObject, handles);
```

```
% --- Executes during object creation, after setting all properties.
function T_CreateFcn(hObject, eventdata, handles)
% hObject    handle to T (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function NP_Callback(hObject, eventdata, handles)
% hObject    handle to NP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of NP as text
%        str2double(get(hObject,'String')) returns contents of NP as a double
handles.np=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function NP_CreateFcn(hObject, eventdata, handles)
% hObject    handle to NP (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function NU_Callback(hObject, eventdata, handles)
% hObject    handle to NU (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of NU as text
%        str2double(get(hObject,'String')) returns contents of NU as a double
handles.nu=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function NU_CreateFcn(hObject, eventdata, handles)
% hObject    handle to NU (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function LAMBDA1_Callback(hObject, eventdata, handles)
% hObject    handle to LAMBDA1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of LAMBDA1 as text
%        str2double(get(hObject,'String')) returns contents of LAMBDA1 as a double
handles.lambda(1)=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function LAMBDA1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to LAMBDA1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
```

```
end



function LAMBDA2_Callback(hObject, eventdata, handles)
% hObject    handle to LAMBDA2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of LAMBDA2 as text
%        str2double(get(hObject,'String')) returns contents of LAMBDA2 as a double
handles.lambda(2)=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function LAMBDA2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to LAMBDA2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function GAMMA1_Callback(hObject, eventdata, handles)
% hObject    handle to GAMMA1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of GAMMA1 as text
%        str2double(get(hObject,'String')) returns contents of GAMMA1 as a double
handles.gamma1(1)=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function GAMMA1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to GAMMA1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function GAMMA2_Callback(hObject, eventdata, handles)
% hObject    handle to GAMMA2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of GAMMA2 as text
%        str2double(get(hObject,'String')) returns contents of GAMMA2 as a double
handles.gamma1(2)=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function GAMMA2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to GAMMA2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function Z1_Callback(hObject, eventdata, handles)
% hObject    handle to Z1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Z1 as text
%        str2double(get(hObject,'String')) returns contents of Z1 as a double
handles.z1=str2double(get(hObject,'String'));
```

```
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function Z1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Z1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function FTS1_Callback(hObject, eventdata, handles)
% hObject    handle to FTS1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of FTS1 as text
%        str2double(get(hObject,'String')) returns contents of FTS1 as a double
handles.fts1=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function FTS1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to FTS1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function FTS2_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to FTS2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of FTS2 as text
%        str2double(get(hObject,'String')) returns contents of FTS2 as a double
handles.fts2=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function FTS2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to FTS2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



% --- Executes on selection change in MO1.
function MO1_Callback(hObject, eventdata, handles)
% hObject    handle to MO1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns MO1 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from MO1
contents = get(hObject,'String');
handles.mo1=str2num(contents{get(hObject,'Value')});
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function MO1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to MO1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
```

```
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes on selection change in MO2.
function MO2_Callback(hObject, eventdata, handles)
% hObject    handle to MO2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns MO2 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from MO2
contents = get(hObject,'String');
handles.mo2=str2num(contents{get(hObject,'Value')});
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function MO2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to MO2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function Z2_Callback(hObject, eventdata, handles)
% hObject    handle to Z2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Z2 as text
%        str2double(get(hObject,'String')) returns contents of Z2 as a double
handles.z2=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
```

```
function Z2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Z2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes on selection change in AT.
function AT_Callback(hObject, eventdata, handles)
% hObject    handle to AT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns AT contents as cell array
%        contents{get(hObject,'Value')} returns selected item from AT
contents = get(hObject,'String');
handles.at=contents{get(hObject,'Value')};
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function AT_CreateFcn(hObject, eventdata, handles)
% hObject    handle to AT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function NODES_Callback(hObject, eventdata, handles)
% hObject    handle to NODES (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of NODES as text
%        str2double(get(hObject,'String')) returns contents of NODES as a double
handles.nodes=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function NODES_CreateFcn(hObject, eventdata, handles)
% hObject    handle to NODES (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function ITER_Callback(hObject, eventdata, handles)
% hObject    handle to ITER (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ITER as text
%        str2double(get(hObject,'String')) returns contents of ITER as a double
handles.iter=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function ITER_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ITER (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
```

```
end



function ERROR_Callback(hObject, eventdata, handles)
% hObject    handle to ERROR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ERROR as text
%        str2double(get(hObject,'String')) returns contents of ERROR as a double


% --- Executes during object creation, after setting all properties.
function ERROR_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ERROR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



% --- Executes on button press in initial.
function initial_Callback(hObject, eventdata, handles)
% hObject    handle to initial (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

[handles.Kp,handles.tao,handles.to,handles.T1,handles.np,handles.nu,handles.lambda,...
   handles.gamma1,handles.SSEtrain]=initial_training(handles.dm1,handles.dm22,...
   handles.T1,handles.np,handles.nu,handles.lambda,handles.gamma1,handles.at,...
   handles.z1,handles.z2,handles.fts1,handles.fts2,handles.mo1,handles.mo2,...
   handles.nodes,handles.iter,handles);
guidata(hObject, handles);

%colocacion de los numeros en la interfaz%%%%%%%%%%

% identificaion FOPDT
set(handles.KP11,'String',num2str(handles.Kp(1,1)));
```

```
set(handles.KP12,'String',num2str(handles.Kp(1,2)));
set(handles.KP21,'String',num2str(handles.Kp(2,1)));
set(handles.KP22,'String',num2str(handles.Kp(2,2)));
set(handles.TAO11,'String',num2str(handles.tao(1,1)));
set(handles.TAO12,'String',num2str(handles.tao(1,2)));
set(handles.TAO21,'String',num2str(handles.tao(2,1)));
set(handles.TAO22,'String',num2str(handles.tao(2,2)));
set(handles.TO11,'String',num2str(handles.to(1,1)));
set(handles.TO12,'String',num2str(handles.to(1,2)));
set(handles.TO21,'String',num2str(handles.to(2,1)));
set(handles.TO22,'String',num2str(handles.to(2,2)));

%Sintonizacion DMC
set(handles.T,'String',num2str(handles.T1));
set(handles.NP,'String',num2str(handles.np));
set(handles.NU,'String',num2str(handles.nu));
set(handles.LAMBDA1,'String',num2str(handles.lambda(1)));
set(handles.LAMBDA2,'String',num2str(handles.lambda(2)));
set(handles.GAMMA1,'String',num2str(handles.gamma1(1)));
set(handles.GAMMA2,'String',num2str(handles.gamma1(2)));

%RANWNN
set(handles.ERROR,'String',num2str(handles.SSEtrain));
msgbox('Training Finish','Initial Training','help')



function P_Callback(hObject, eventdata, handles)
% hObject    handle to P (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of P as text
%        str2double(get(hObject,'String')) returns contents of P as a double
handles.p=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function P_CreateFcn(hObject, eventdata, handles)
% hObject    handle to P (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function P1_Callback(hObject, eventdata, handles)
% hObject    handle to P1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of P1 as text
%        str2double(get(hObject,'String')) returns contents of P1 as a double
handles.p1=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function P1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to P1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function N_Callback(hObject, eventdata, handles)
% hObject    handle to N (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of N as text
%        str2double(get(hObject,'String')) returns contents of N as a double
handles.n=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
```

```
function N_CreateFcn(hObject, eventdata, handles)
% hObject    handle to N (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function PM_Callback(hObject, eventdata, handles)
% hObject    handle to PM (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of PM as text
%        str2double(get(hObject,'String')) returns contents of PM as a double
handles.pm=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function PM_CreateFcn(hObject, eventdata, handles)
% hObject    handle to PM (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


function LMIN_Callback(hObject, eventdata, handles)
% hObject    handle to LMIN (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of LMIN as text
%        str2double(get(hObject,'String')) returns contents of LMIN as a double
handles.lmin=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function LMIN_CreateFcn(hObject, eventdata, handles)
% hObject    handle to LMIN (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function LMAX_Callback(hObject, eventdata, handles)
% hObject    handle to LMAX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of LMAX as text
%        str2double(get(hObject,'String')) returns contents of LMAX as a double
handles.lmax=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function LMAX_CreateFcn(hObject, eventdata, handles)
% hObject    handle to LMAX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
function GMIN_Callback(hObject, eventdata, handles)
% hObject    handle to GMIN (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of GMIN as text
%        str2double(get(hObject,'String')) returns contents of GMIN as a double
handles.gmin=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function GMIN_CreateFcn(hObject, eventdata, handles)
% hObject    handle to GMIN (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function GMAX_Callback(hObject, eventdata, handles)
% hObject    handle to GMAX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of GMAX as text
%        str2double(get(hObject,'String')) returns contents of GMAX as a double
handles.gmax=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function GMAX_CreateFcn(hObject, eventdata, handles)
% hObject    handle to GMAX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
```

```
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function MDU1_Callback(hObject, eventdata, handles)
% hObject    handle to MDU1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of MDU1 as text
%        str2double(get(hObject,'String')) returns contents of MDU1 as a double
handles.mdu1=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function MDU1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to MDU1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function MDU2_Callback(hObject, eventdata, handles)
% hObject    handle to MDU2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of MDU2 as text
%        str2double(get(hObject,'String')) returns contents of MDU2 as a double
handles.mdu2=str2double(get(hObject,'String'));
guidata(hObject, handles);
```

```
% --- Executes during object creation, after setting all properties.
function MDU2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to MDU2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function PC_Callback(hObject, eventdata, handles)
% hObject    handle to PC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of PC as text
%        str2double(get(hObject,'String')) returns contents of PC as a double
handles.pc=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function PC_CreateFcn(hObject, eventdata, handles)
% hObject    handle to PC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function RPH_Callback(hObject, eventdata, handles)
% hObject    handle to RPH (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of RPH as text
%        str2double(get(hObject,'String')) returns contents of RPH as a double
handles.rph=str2double(get(hObject,'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function RPH_CreateFcn(hObject, eventdata, handles)
% hObject    handle to RPH (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end


% --- Executes on button press in DMC_START.
function DMC_START_Callback(hObject, eventdata, handles)
% hObject    handle to DMC_START (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global G T gamma1 lambda np nu
myopts = simset('DstWorkspace ', 'current','SrcWorkspace','current');
sim('model_dmc_mult',handles.tfinal,myopts)
save resultados_dmc cont_dmc salida_real_dmc em_dmc salida_dmc salida_ruido_dmc
iae_dmc
axes(handles.axes1)
plot(tout,salida_real_dmc(:,1))
title('Output pH')
xlabel('Time (s)')
ylabel('pH')
axes(handles.axes7)
plot(tout,salida_real_dmc(:,2))
title('Output height')
xlabel('Time (s)')
ylabel('m')
axes(handles.axes9)
plot(tout,salida_dmc(:,:))
title('Controlled variables without noise')
```

```
xlabel('Time (s)')
ylabel('%TO')
axes(handles.axes8)
plot(tout,salida_ruido_dmc(:,1),tout,salida_ruido_dmc(:,2))
title('Controlled variables with noise')
xlabel('Time (s)')
ylabel('%TO')
axes(handles.axes10)
plot(tout,cont_dmc(:,1),tout,cont_dmc(:,2))
title('Controller Output')
xlabel('Time (s)')
ylabel('%CO')
axes(handles.axes11)
plot(tout,em_dmc(:,1),tout,em_dmc(:,2))
title('Model error')
xlabel('Time (s)')
ylabel('%TO')
set(handles.IAE,'String',num2str(max(iae_dmc(:,1))))
set(handles.IAE2,'String',num2str(max(iae_dmc(:,2))))
msgbox('Simulation Terminated','DMC','help')


% --- Executes on button press in NEDMC_START.
function NEDMC_START_Callback(hObject, eventdata, handles)
% hObject    handle to NEDMC_START (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global G T gamma1 lambda np nu
pc=handles.pc;
pm=handles.pm;
p=handles.p;
p1=handles.p1;
n=handles.n;
lmin=handles.lmin;
lmax=handles.lmax;
gmin=handles.gmin;
gmax=handles.gmax;
mdu1=handles.mdu1;
mdu2=handles.mdu2;
rph=handles.rph;
GT=handles.GT;

save parametros_spea pc pm n p p1 lmin lmax gmin gmax mdu1 mdu2 rph GT

load parametros_dmc_mult
```

```
myopts = simset('DstWorkspace ', 'current','SrcWorkspace','current');
sim('DmcAdaptMult_mod_ref',handles.tfinal,myopts)
save resultados_dmc_adap cont_dmc_adap salida_real_dmc_adap em_dmc_adap
salida_dmc_adap salida_dmc_adap_red salida_ruido_dmc_adap iae_dmc_adap
axes(handles.axes12)
plot(tout,salida_real_dmc_adap(:,1))
title('Output pH')
xlabel('Time (s)')
ylabel('pH')
axes(handles.axes13)
plot(tout,salida_real_dmc_adap(:,2))
title('Output height')
xlabel('Time (s)')
ylabel('m')
axes(handles.axes15)
plot(tout,salida_dmc_adap(:,:))
title('Controlled variables without noise')
xlabel('Time (s)')
ylabel('%TO')
axes(handles.axes14)
plot(tout,salida_ruido_dmc_adap(:,1),tout,salida_ruido_dmc_adap(:,2))
title('Controlled variables with noise')
xlabel('Time (s)')
ylabel('%TO')
axes(handles.axes16)
plot(tout,cont_dmc_adap(:,1),tout,cont_dmc_adap(:,2))
title('Controller Output')
xlabel('Time (s)')
ylabel('%CO')
axes(handles.axes17)
plot(tout,em_dmc_adap(:,1),tout,em_dmc_adap(:,2))
title('Model error')
xlabel('Time (s)')
ylabel('%TO')
axes(handles.axes18)
plot(tout,salida_dmc_adap_red(:,:))
title('RNA output Vs Real output')
xlabel('Time (s)')
ylabel('%TO')

algo=dlmread('factores.txt');
n2=length(algo(:,1));
tiempo=1:T:n2*T;
lambda1=algo(:,1);
lambda2=algo(:,2);
```

```
gamma11=algo(:,3);
gamma22=algo(:,4);
W=algo(:,5:8);
clear algo
axes(handles.axes19)
stem(tiempo,lambda1);
title('Lambda 1 each sample time')
xlabel('Time (s)')
ylabel('Lambda 1')
axes(handles.axes20)
stem(tiempo,lambda2,'r');
title('Lambda 2 each sample time')
xlabel('Time (s)')
ylabel('Lambda 2')
axes(handles.axes21)
stem(tiempo,gamma11,'r');
title('Gamma 1 each sample time')
xlabel('Time (s)')
ylabel('Gamma 1')
axes(handles.axes22)
stem(tiempo,gamma22,'g');
title('Gamma 2 each sample time')
xlabel('Time (s)')
ylabel('Gamma 2')
axes(handles.axes30)
stem(tiempo,W(:,1),'b');
title('Weight 1 each sample time')
xlabel('Time (s)')
ylabel('W1')
axes(handles.axes31)
stem(tiempo,W(:,2),'r');
title('Weight 2 each sample time')
xlabel('Time (s)')
ylabel('W2')
axes(handles.axes32)
stem(tiempo,W(:,3),'g');
title('Weight 3 each sample time')
xlabel('Time (s)')
ylabel('W3')
axes(handles.axes33)
stem(tiempo,W(:,4),'r');
title('Weight 4 each sample time')
xlabel('Time (s)')
ylabel('W4')
```

```
set(handles.IAE_NEDMC,'String',num2str(max(iae_dmc_adap(:,1))))
set(handles.IAE2_NEDMC,'String',num2str(max(iae_dmc_adap(:,2))))
msgbox('Simulation Terminated','NEDMC','help')

% --- Executes on button press in PLOTS_DMC.
function PLOTS_DMC_Callback(hObject, eventdata, handles)
% hObject    handle to PLOTS_DMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.Panel1,'Visible','off')
set(handles.Panel2,'Visible','off')
set(handles.Panel3,'Visible','off')
set(handles.DMC_RES,'Visible','on')
set(handles.BACK,'Visible','on')


% --- Executes on button press in PLOTS_NEDMC.
function PLOTS_NEDMC_Callback(hObject, eventdata, handles)
% hObject    handle to PLOTS_NEDMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.Panel1,'Visible','off')
set(handles.Panel2,'Visible','off')
set(handles.Panel3,'Visible','off')
set(handles.NEDMC_RES,'Visible','on')
set(handles.BACK,'Visible','on')

% --- Executes on button press in BACK2.
function BACK_Callback(hObject, eventdata, handles)
% hObject    handle to BACK2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.Panel1,'Visible','on')
set(handles.Panel2,'Visible','on')
set(handles.Panel3,'Visible','on')
set(handles.DMC_RES,'Visible','off')
set(handles.NEDMC_RES,'Visible','off')
set(handles.BACK,'Visible','off')

function IAE_Callback(hObject, eventdata, handles)
% hObject    handle to IAE (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of IAE as text
%        str2double(get(hObject,'String')) returns contents of IAE as a double


% --- Executes during object creation, after setting all properties.
function IAE_CreateFcn(hObject, eventdata, handles)
% hObject    handle to IAE (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end




function IAE2_Callback(hObject, eventdata, handles)
% hObject    handle to IAE2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of IAE2 as text
%        str2double(get(hObject,'String')) returns contents of IAE2 as a double


% --- Executes during object creation, after setting all properties.
function IAE2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to IAE2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
function IAE_NEDMC_Callback(hObject, eventdata, handles)
% hObject    handle to IAE_NEDMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of IAE_NEDMC as text
%        str2double(get(hObject,'String')) returns contents of IAE_NEDMC as a double


% --- Executes during object creation, after setting all properties.
function IAE_NEDMC_CreateFcn(hObject, eventdata, handles)
% hObject    handle to IAE_NEDMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function IAE2_NEDMC_Callback(hObject, eventdata, handles)
% hObject    handle to IAE2_NEDMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of IAE2_NEDMC as text
%        str2double(get(hObject,'String')) returns contents of IAE2_NEDMC as a double


% --- Executes during object creation, after setting all properties.
function IAE2_NEDMC_CreateFcn(hObject, eventdata, handles)
% hObject    handle to IAE2_NEDMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
```

```
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



% --- Executes on button press in NEXT2.
function NEXT_Callback(hObject, eventdata, handles)
% hObject    handle to NEXT2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.NEDMC_RES,'Visible','off')
set(handles.BACK,'Visible','off')
set(handles.NEDMC_RES2,'Visible','on')


% --- Executes on button press in BACK2.
function BACK2_Callback(hObject, eventdata, handles)
% hObject    handle to BACK2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

set(handles.NEDMC_RES,'Visible','on')
set(handles.BACK,'Visible','on')
set(handles.NEDMC_RES2,'Visible','off')


% --- Executes on button press in NEXT2.
function NEXT2_Callback(hObject, eventdata, handles)
% hObject    handle to NEXT2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

set(handles.NEDMC_RES2,'Visible','off')
set(handles.NEDMC_RES3,'Visible','on')


% --- Executes on button press in BACK3.
function BACK3_Callback(hObject, eventdata, handles)
% hObject    handle to BACK3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)

set(handles.NEDMC_RES2,'Visible','on')
set(handles.NEDMC_RES3,'Visible','off')


% --- Executes on button press in PLOTS_TRAINING.
function PLOTS_TRAINING_Callback(hObject, eventdata, handles)
% hObject    handle to PLOTS_TRAINING (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.Panel1,'Visible','off')
set(handles.Panel2,'Visible','off')
set(handles.Panel3,'Visible','off')
set(handles.T_RES,'Visible','on')

% --- Executes on button press in BACKT.
function BACKT_Callback(hObject, eventdata, handles)
% hObject    handle to BACKT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.Panel1,'Visible','on')
set(handles.Panel2,'Visible','on')
set(handles.Panel3,'Visible','on')
set(handles.T_RES,'Visible','off')



% --- Executes on button press in OPEN_DMC.
function OPEN_DMC_Callback(hObject, eventdata, handles)
% hObject    handle to OPEN_DMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
open('model_dmc_mult.mdl')

% --- Executes on button press in OPEN_NEDMC.
function OPEN_NEDMC_Callback(hObject, eventdata, handles)
% hObject    handle to OPEN_NEDMC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
open('DmcAdaptMult_mod_ref.mdl')


function MDR1_Callback(hObject, eventdata, handles)
% hObject    handle to MDR1 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of MDR1 as text
%        str2double(get(hObject,'String')) returns contents of MDR1 as a double
handles.GT(1,1)=str2double(get(hObject,'String'));
guidata(hObject, handles)

% --- Executes during object creation, after setting all properties.
function MDR1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to MDR1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function MDR2_Callback(hObject, eventdata, handles)
% hObject    handle to MDR2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of MDR2 as text
%        str2double(get(hObject,'String')) returns contents of MDR2 as a double
handles.GT(1,2)=str2double(get(hObject,'String'));
guidata(hObject, handles)

% --- Executes during object creation, after setting all properties.
function MDR2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to MDR2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
```

```
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function ME1_Callback(hObject, eventdata, handles)
% hObject    handle to ME1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ME1 as text
%        str2double(get(hObject,'String')) returns contents of ME1 as a double
handles.GT(1,3)=str2double(get(hObject,'String'));
guidata(hObject, handles)

% --- Executes during object creation, after setting all properties.
function ME1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ME1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end



function ME2_Callback(hObject, eventdata, handles)
% hObject    handle to ME2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ME2 as text
%        str2double(get(hObject,'String')) returns contents of ME2 as a double
handles.GT(1,4)=str2double(get(hObject,'String'));
guidata(hObject, handles)

% --- Executes during object creation, after setting all properties.
function ME2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ME2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

Controlador DMC adaptativo para la estrategia MIMO

```
function [m]=dmc_mod_ref(R,C,time);

global m mbar predvect T lambda gamma1 lambda_inicial np nu G i Wh Bh Wo Bo
global c_ant cp_ant delta_p delta_u_1 y_ant Wh2 Bh2 Wo2 Bo2 R_ant R1 R2
global mod_ref1 mod_ref2 delta_r1 delta_r2 decisor P
global pc pm n p p1 lmin lmax gmin gmax mdu1 mdu2 rph GT

% inicializacion
if time==0
   load parametros_spea
   load parametros_dmc_mult;
   load referencia
   decisor=readfis('decisor2.fis');
   load weights_mult1
   lambda_inicial=lambda;
   dlmwrite('factores.txt',[lambda(1) lambda(2) gamma1' 0.16 0.16 0.24 0.24]);
   mbar=52;
   m(1)=mbar;
   m(2)=mbar;
   m(3:4)=0;
   predvect=zeros(np,2);
   predvect(:,1)=C(1)*ones(np,1);
   predvect(:,2)=C(2)*ones(np,1);

   % se guardan los valores anteriores de predicciones al igual que de la
   % entrada para sumarle el delta a las nuevas predicciones hechas por la
   % red en el spea2
   cp_ant=nn_sim_val([m(1) m(2) C(1) C(2)],Wh,Bh,Wo,Bo);
   m(5:6)=cp_ant;
   delta_p(1)=cp_ant(1)-C(1);
   delta_p(2)=cp_ant(2)-C(2);

   i=time;
   y_ant=C;
   R_ant=R;

   delta_r1=R(1)-50.1786; % valor incial del delta_r porque no hay cambio al principio
empieza con ese delta.
   mod_ref1=R1*delta_r1+50.1786*ones(np,1);% valor inicial del modelo de referencia

   delta_r2=R(2)-51.25; % valor incial del delta_r porque no hay cambio al principio
empieza con ese delta.
   mod_ref2=R2*delta_r2+51.25*ones(np,1);% valor inicial del modelo de referencia
```

```
end
   if time >= T+i

        delta_r1=R(1)-R_ant(1);   % Cambio de la referencia
        mod_ref1=R1*delta_r1+[mod_ref1(2:np);mod_ref1(np)];% se actualiza el vector de
modelo de referencia

        delta_r2=R(2)-R_ant(2);   % Cambio de la referencia
        mod_ref2=R2*delta_r2+[mod_ref2(2:np);mod_ref2(np)];% se actualiza el vector de
modelo de referencia

        %%%%% Actualizacion pesos RAWNN %%%%%%%%%%%%%%%%%%
        [Wh,Bh,Wo,Bo,P]=rls_rawnn2([m(1) m(2) y_ant(1) y_ant(2)],[C(1)
C(2)],Wh,Bh,Wo,Bo,P);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%

        %%%%%%%%Cálculo del mejor lambda a través del
SPEA%%%%%%%%%%%%%%%

        best=SPEAII([lambda gamma1'],R,R_ant,mod_ref1,mod_ref2,C,...
           y_ant,predvect,G,np,nu,m,Wh,Bh,Wo,Bo,delta_p,decisor,...
           pc,pm,n,p,p1,lmin,lmax,gmin,gmax,mdu1,mdu2,rph,GT);

        lambda=best(1:2)';
        gamma1=best(3:4);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%

        em(1)=C(1)-predvect(1,1);
        em(2)=C(2)-predvect(1,2);

        predvect_u(:,1)=[predvect(2:np,1);predvect(np,1)]+em(1);
        predvect_u(:,2)=[predvect(2:np,2);predvect(np,2)]+em(2);

        E(:,1)=mod_ref1-predvect_u(:,1);
        E(:,2)=mod_ref2-predvect_u(:,2);
        Et=[E(:,1);E(:,2)];

        gamma2=gamma1(1)*eye(np);
        gamma2(np+1:2*np,np+1:2*np)=gamma1(2)*eye(np);
```

```
lambda2=lambda(1)*eye(nu);
lambda2(nu+1:2*nu,nu+1:2*nu)=lambda(2)*eye(nu);

Kc=inv(G'*gamma2*G+lambda2)*G'*gamma2;

delta_u=Kc*Et;
delta_u2(1)=delta_u(1);
delta_u2(2)=delta_u(nu+1);
%%%%%%%%Saturación de los delta_u
if delta_u2(1) > mdu1
   delta_u2(1)=mdu1;
elseif delta_u2(1) < -mdu1
   delta_u2(1)=-mdu1;
end

if delta_u2(2) > mdu2
   delta_u2(2)=mdu2;
elseif delta_u2(2) < -mdu2
   delta_u2(2)=-mdu2;
end

%%%%%%%%Saturación de la señal de control m
if (m(1) + delta_u2(1)) > 100,
   delta_u2(1)=100-m(1);
elseif (m(1) + delta_u2(1)) < 0,
   delta_u2(1) = -m(1);
end

if (m(2) + delta_u2(2)) > 100,
   delta_u2(2)=100-m(2);
elseif (m(2) + delta_u2(2)) < 0,
   delta_u2(2) = -m(2);
end

m(1)=m(1)+delta_u2(1);
m(2)=m(2)+delta_u2(2);
m(3:4)=em;

predvect(:,1)=G(1:np,1:nu)*[delta_u2(1);zeros(nu-1,1)]+...
   G(1:np,nu+1:2*nu)*[delta_u2(2);zeros(nu-1,1)]+predvect_u(:,1);

predvect(:,2)=G(np+1:2*np,1:nu)*[delta_u2(1);zeros(nu-1,1)]+...
   G(np+1:2*np,nu+1:2*nu)*[delta_u2(2);zeros(nu-1,1)]+predvect_u(:,2);
```

```
    delta_p(1)=cp_ant(1)-C(1);
    delta_p(2)=cp_ant(2)-C(2);

    cp_ant=nn_sim_val([m(1) m(2) C(1) C(2)],Wh,Bh,Wo,Bo);
    m(5:6)=cp_ant;
    i=time;
    y_ant=C;
    R_ant=R;
else
        m=[m];
end
```

```
function [y]=nn_sim_auto_mult4(inputs,Wh,Bh,Wo,Bo,ph)
%Esta función simula una red neuronal que predice hasta ph basándose en sus
%propias predicciones de un sistema 2 X 2

y(1,:)=nn_sim_val([inputs(1:2) inputs(ph*2+1) inputs(ph*2+2)],Wh,Bh,Wo,Bo);
k=2;
for i=3:2:ph*2
   y(k,:)=nn_sim_val([inputs(i:i+1) y(k-1,:)],Wh,Bh,Wo,Bo);
   k=k+1;
end
```

```
function [T,np,nu,lambda]=sinto_dmc_mult(Kp,tao,to)
% Función para sintonizar los parámetros de un DMC SISO
% basándose en la identificación de un proceso como FOPDT
% [T,np,nu,lambda]=sinto_dmc(Kp,tao,to)

T=min(min(ceil(max(0.1.*tao,0.5.*to))));
k=(to./T)+1;
np=max(max(round((5.*tao)./T+k)));
nu=max(max(round((tao./T)+k)));
[a,b]=size(Kp);
for i=1:b
   suma=0;
   for j=1:a
     suma=(Kp(j,i)^2)*(np-k(j,i)-(3/2)*(tao(j,i)/T)+2-(nu-1)/2)+suma;
   end
   lambda(i)=(nu/500)*suma;
end
```

Algoritmo evolutivo para la estrategia MIMO

```
function best3=SPEAII(params,r,r_ant,mod_ref1,mod_ref2,y_act,y_ant,...
   predvect_2,G_2,np_2,nu_2,m_2,Wh,Bh,Wo,Bo,delta_p,decisor,...
   pc,pm,n,p,p1,lmin,lmax,gmin,gmax,mdu1,mdu2,rph,GT)


 NUM_TRAITS=4;            % Number of traits in each individual

 if r(1)~=r_ant(1) || r(2)~=r_ant(2)
    global G T gamma1 lambda np nu
    load parametros_dmc_mult
    params=[lambda gamma1'];
    clear G T gamma1 lambda np nu
 end

 HIGHTRAIT=[params(1)+lmax params(2)+lmax params(3)+gmax params(4)+gmax];
 for para=1:4
  if params(para) < lmin+0.2
     params(para)=lmin+0.2;
  end
 end
 LOWTRAIT=[params(1)-lmin params(2)-lmin params(3)-gmin+0.8 params(4)-
gmin+0.8];

 MUTAT_PROB=pm;
 CROSS_PROB=pc;
 POP_SIZE=p;
 POP_SIZE2=p1;
 ELITISM=0;
 % se desactiva el elitismo y se hace el mejor miembro cero debido a que
 % las funciones de variacion lo necesitan(mutacion y cruzamiento)
 bestmember=0;

 MAX_GENERATION=n;       % Number of times the loop will run before
                                                % giving up on the EPSILON-DELTA
termination cond.

  OBJETIVOS=6;

popcount=1;              % Initialize the generation count,
                                                % set it to one, the first population
```

```
% Se inicializa la poblacion de controladores con un controlador diseñado
% previamente con formulas de sintonizacion basadas en un modelo fopdt de
% la planta y se le agrega un ruido de media 0 y varianza uno.


for i=1:NUM_TRAITS
   for j=1:POP_SIZE
      %trait(i,j,:)=params(i)+rand;
      trait(i,j,:)=(HIGHTRAIT(i)-LOWTRAIT(i)-1)*rand+LOWTRAIT(i);
   end
end

for i=1:NUM_TRAITS
   for j=1:POP_SIZE2
      %trait2(i,j,:)=params(i)+rand;
      trait2(i,j,:)=(HIGHTRAIT(i)-LOWTRAIT(i)-1)*rand+LOWTRAIT(i);
   end
end

% Se calcula el tamaño de la población unión del archivo y de la población
POP_SIZE3=POP_SIZE+POP_SIZE2;

while popcount <= MAX_GENERATION

% First, fix bad traits (i.e., ones that are out of the range
% specified by HIGHTRAIT and LOWTRAIT) by saturation at the extremes

trait(:,:,popcount)=saturacion(trait(:,:,popcount),POP_SIZE,NUM_TRAITS,...
   HIGHTRAIT,LOWTRAIT);

trait2(:,:,popcount)=saturacion(trait2(:,:,popcount),POP_SIZE2,NUM_TRAITS,...
   HIGHTRAIT,LOWTRAIT);

% se genera el conjunto union concatenando los dos conjuntos
trait_t=[trait(:,:,popcount) trait2(:,:,popcount)];


[fitness(:,:,popcount)]=funciones_objetivo(trait_t,POP_SIZE3,y_act,y_ant,...
   predvect_2,r,[mod_ref1 mod_ref2],np_2,nu_2,G_2,m_2,Wh,Bh,Wo,Bo,delta_p,...
   mdu1,mdu2,rph);

% se obtiene todas las soluciones no dominadas
[frente_temp,S(:,popcount),R(:,popcount),D(:,:,popcount),F(:,popcount)]=...
   pareto(fitness(:,:,popcount),POP_SIZE3,NUM_TRAITS,OBJETIVOS);
% Se almacena en un frente temporal debido a que la longitud de este varía
```

```
% y por tanto puede haber un erro en la asignación cuando la longitud sea
% menor.
long_frente=length(frente_temp);
frente(1:long_frente,popcount)=frente_temp;
clear frente_temp


nuevos=evr_selec(F(:,popcount),D(:,:,popcount),POP_SIZE2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Create the next generation
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% First, form the mating pool.
% To do this we select as parents the
% chromosomes that are most fit.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

P_sel=seleccion4(F(nuevos,popcount),nuevos);

popcount=popcount+1;                    % Increment to the next generation


% Se copian los individuos del nuevo frente en el cual se han puesto a
% concursar a los individuos modificados y a los resultantes de la
% seleccion anterior en trait

trait(:,1:POP_SIZE2,popcount)=trait_t(:,nuevos);

trait2(:,:,popcount)=cruzamiento4(trait_t(:,P_sel),F(nuevos,popcount-1),...
   CROSS_PROB,POP_SIZE2,NUM_TRAITS,ELITISM,bestmember,...
   HIGHTRAIT,LOWTRAIT);

trait2(:,:,popcount)=cruzamiento5(trait2(:,:,popcount),...

CROSS_PROB,POP_SIZE2,NUM_TRAITS,ELITISM,bestmember,HIGHTRAIT,LOWTRAIT);


trait2(:,:,popcount)=mutacion2(trait2(:,:,popcount),MUTAT_PROB,POP_SIZE2,NUM_TRAITS,...
  HIGHTRAIT,LOWTRAIT,popcount,MAX_GENERATION,ELITISM,bestmember);
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%
end  % End "for pop_count=..." loop - the main loop.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%

dR=abs(r-r_ant);
E=abs(r-y_act);

entradas=[dR(1) dR(2) E(1) E(2)];
entradas=entradas./GT;

for i=1:4
    if entradas(i) > 1
        entradas(i)=1;
    end
end

w=evalfis(entradas,decisor);
%w=[0.16 0.16 0.16 0.16];
fitness_temp=w(1)*fitness(nuevos,1,MAX_GENERATION)+...
    w(2)*fitness(nuevos,2,MAX_GENERATION)+...
    w(3)*fitness(nuevos,3,MAX_GENERATION)+...
    w(4)*fitness(nuevos,4,MAX_GENERATION)+...
    w(1)*fitness(nuevos,5,MAX_GENERATION)+...
    w(2)*fitness(nuevos,6,MAX_GENERATION);

[best_fitness,best2]=min(fitness_temp);
best2=nuevos(best2);
best3=trait_t(:,best2);

dlmwrite('factores.txt',[best3' w],'-append');
```

```
function [Wh,Bh,Wo,Bo,Ymin,SSEtrain,P]=train_rawn_mult3(m1t,m2t,c1t,c2t,...
   m1v,m2v,c1v,c2v,nodes,iter)


% se hacen los vectores de entrenamiento

y1(:,1)=[c1t(2:length(c1t));c1t(length(c1t))*ones(1,1)];
y2(:,1)=[c2t(2:length(c2t));c1t(length(c2t))*ones(1,1)];

inputst=[m1t m2t c1t c2t];
outputst=[y1 y2];

% Lo siguiente es igual a lo anterior pero para la validacion cruzada
clear m1t m2t c1t c2t y1 y2
% se hacen los vectores de entrenamiento desde y(t) hasta y(t+ph)

y1(:,1)=[c1v(2:length(c1v));c1v(length(c1v))*ones(1,1)];
y2(:,1)=[c2v(2:length(c2v));c2v(length(c2v))*ones(1,1)];

inputsv=[m1v m2v c1v c2v];
outputsv=[y1 y2];
clear m1v m2v c1v y1 y2

[Wh,Bh,Wo,Bo,Ymin,SSEtrain]=entrenamiento2(inputst,outputst,nodes,iter,inputsv,output
sv);

ie=size(inputst,2);                    %identifies number of input elements
l=size(inputst,1);                          %identifies training set dimension
%Include bias input
inputst(:,ie+1)=ones(l,1);
W=[Wh;Bh];
Wout=[Wo;Bo];

%%%%%%% para el caso MIMO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Z=inputst*W;
V=tansig(Z);
V(:,nodes+1)=ones(l,1);
P=inv(V'*V);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
clear Wo Bo
Wo=Wout(1:nodes,:);
Bo=Wout(nodes+1,:);
```