



8 CÓDIGO FUENTE

REDNEURONAL

```
%redneuronal

inicia_red;

% salida correcta y salida red
%subplot(2,2,1)
%gr_eje_bp0

% varias pasadas de entrenamiento

for pas=1:npas,
    %for pas=1101:1300,
    bucle;
    sumaerrores(pas)=sum;

    %entrenamiento;
end

%subplot(2,2,2)
%gr_eje_bp0
```

BUCLE

```
inicia;
sumerr=0;

for k=1:nf,
    controlador_pd;
    %entrena_mra_momentos;
    entrena_mra;
    % salida_red;
    %fin salida de la red
```

Autora :Carmen García Olloqui	Departamento de Sistemas y Automática	
Tutor : Manuel Ruiz Arahal	Universidad de Sevilla	Página 115



```
vU1(pas,k)=U1;
%sumaU1(pas)=sumaU1(pas)+U1;
vU2(pas,k)=U2;
Uc=U1+U2;%/salida del controlador si no se saturara.
%
%Uc=U2;
sistema;
%entrenamiento;
sumerr=sumerr+U1*U1;
%actualización de X
sum=sum+ek*ek;
suma_ek=suma_ek+ek;
Us(k)=U1;
U2s(k)=U2;
vU(k)=U;
    for j=nument:-1:3,
        X(j)=X(j-1);
    end
    X(2)=yk;
    if k>nf-anticipacion_referencia, X(5)=vr(k-
nf+anticipacion_referencia);
        else X(5)=vr(k+anticipacion_referencia);end
    X(12)=U;

    X(17)=suma_ek; %prueba para tener el efecto de un PID
end
```

BUCLE STOP

```
inicia;
sumerr=0;

for k=1:nf,
    controlador_pd;
    %entrena_mra_momentos;

    if pas>2,

        if sumaerrores(pas-1)< sumaerrores(pas-2),
            entrena_mra;
            pas_entrenamiento=pas;
            sumaerrores(pas-1);
            sumaerrores(pas-2);
        else
            aaa=1;
        end

    else
        entrena_mra;
        pas_entrenamiento=pas;
```

Autora :Carmen García Olloqui	Departamento de Sistemas y Automática	
Tutor : Manuel Ruiz Arahal	Universidad de Sevilla	Página 116



end

```
% salida_red;
%fin salida de la red
vU1(pas,k)=U1;

%sumaU1(pas)=sumaU1(pas)+U1;
vU2(pas,k)=U2;

Uc=U1+vU2(pas_entrenamiento,k)    ;%/salida del controlador si no se
saturara.

%Uc=U1;
sistema;

%entrenamiento;
sumerr=sumerr+U1*U1;
%actualización de X
sum=sum+ek*ek;
Us(k)=U1;
U2s(k)=U2;
vU(k)=U;
    for j=nument:-1:3,
        X(j)=X(j-1);
    end
    X(2)=yk;
    if k>nf-anticipacion_referencia, X(5)=vr(k-
nf+anticipacion_referencia);
        else X(5)=vr(k+anticipacion_referencia);end
    X(11)=U;

end
```

ENTRENA

```
fp(1,1)=1;
for in=1:nn,
    net(in)=X'*W(:,in);
    ss = net(in);
    es = exp(-ss);

    % n(in) = 1/(1+es) ;% hasta ahora estaba asi
    n(in) = (1-es)/(1+es);

    sp(1,in) = 2*es/((1+es)*(1+es)); % hasta ahora estaba asi
```

Autora :Carmen García Olloqui		Departamento de Sistemas y Automática	
Tutor : Manuel Ruiz Arahal		Universidad de Sevilla	Página 117



```
%sp(1,in) = 1*es/((1+es)*(1+es)); %sigma prima (para adaptacion)
%fp(1,in+1)=sp(1,in);
end

%utiliza BP para modificar pesos de la red

err =U1;

% modifica pesos redn

% pesos wos

Wos=Wos+alfa*err*n';%*n';prueba=prueba+1;
Wts=Wts+alfa*err;
% pesos weo
%W = W + alfa*err* X*Ws*sp;
for ent=1:nument,
    for nodo=1:nn,
        W(ent,nodo)=W(ent,nodo)+alfa*err*X(ent)*sp(nodo)*Wos(nodo);
    end
end
```

CALCULA SALIDA

```
%salida de la red neuronal
U2=Wos*n+Wts;
```

```
yk = a*yk + (1-a)*Ur;
```

INICIA

```
%estado inicial nulo
a=0.6;
d=3;
U=0;Uc=0;U1=0;ek=0; yk=0; uk=0; vur = zeros(1,d); sumek=0;
vy=0;% vu=0; vk=0;
sumaU=0;
uma=1;
uzm=0.1;
sum=0;

suma_ek=0;
X=zeros(nument,1);
X(1)=1;
Us=zeros(nf,1);
```

Autora :Carmen García Olloqui	Departamento de Sistemas y Automática	
Tutor : Manuel Ruiz Arahal	Universidad de Sevilla	Página 118

**INICIA RED**

```
%inicia red neuronal
% numero de nodos
%nn=10;
%anticipacion_referencia=10;
%nument=17;

%alfa = 0.25;

kp=0.7;
ki=0;
kd=0.8;

% los valores de x usados para entrenar
%vxce = [-.8 -.7 -.6 -.55 -.3 -.1 .05 .2 .5 .66 .7 .72 .85 .9];
%npatce = length(vxce);
%npas=300;
% rand('state',0);
% estado inicial aleatorio
%wos = (rand(nn,1)*2-ones(nn,1))*0.1;
%weo = (rand(1,nn)*2-ones(1,nn))*0.1;

%W=(rand(nument,nn)-ones(nument,nn))*0.1;
%Ws=(rand(1,nn+1)-ones(1,nn+1))*0.1;
%n=zeros(nn+1,1);
%n(1)=1;
% vector de referencias
%baseg = ones(1, 40);
%basep = ones(1, 25);
baseg=ones(1,40);
basep=ones(1,50);
%vr = [ 0.5*baseg -0.25*baseg 0.7*baseg 0*basep -0.14*baseg -
0.2*basep];
vr =[ 0.5*baseg -0.25*baseg 0.7*baseg 0*basep -0.14*baseg -
0.2*basep];

k=1;
d=3;

%inicia_mra_momentos;
inicia_mra
nf=length(vr);

Us1=zeros(nf,1);
Us10=zeros(nf,1);
U2s=zeros(nf,1);
sumaerrores=zeros(1,npas);
```



INICIA ENTRENA

```
nn=10;  
anticipacion_referencia=5;  
nument=17;  
%alfa = 0.001;%%la buena  
  
alfa=0.00001; %para pruebas de pesos inicilaes
```

```
npas=500;  
W=(rand(nument,nn)-ones(nument,nn)).*0.1;  
Wos=(rand(1,nn)-ones(1,nn)).*0.1;  
Wts=rand(1,1)*0.1;  
inicio_pesos_pd;  
n=zeros(nn,1);  
X(1)=1;
```

INICIA PESOS PD

```
W=(rand(nument,nn)-ones(nument,nn))*0;  
Wos=(rand(1,nn)-ones(1,nn))*0;  
Wts=rand(1,1)*0;
```

```
kp_pesos=0.2;  
kd_pesos=0.1;  
ki_pesos=0.3;
```

```
%W(10,1)=1;  
%W(11,2)=1;  
%W(2,3)=1;  
%W(3,4)=1;
```

```
%Wos(1)=2*(kd_pesos+kp_pesos);  
%Wos(2)=2*(-kd_pesos);  
%Wos(3)=-2*(kd_pesos+kp_pesos);  
%Wos(4)=2*(kd_pesos);
```

```
%W(15,5)=1*1;  
%Wos(5)=1*2*ki_pesos;
```

```
%repartiendo pesos
```

```
W(10,1)=0.5;  
W(10,6)=0.5;
```

```
W(11,2)=0.5;  
W(11,7)=0.5;
```

Autora :Carmen García Olloqui	Departamento de Sistemas y Automática	
Tutor : Manuel Ruiz Arahal	Universidad de Sevilla	Página 120



```
W(2,3)=0.5;
```

```
W(2,8)=0.5;
```

```
W(3,4)=0.5;
```

```
W(3,9)=0.5;
```

```
W(17,5)=0.5; %para PID
```

```
W(17,10)=0.5; %para PID
```

```
Wos(1)=2*(kd_pesos+kp_pesos);
```

```
Wos(6)=2*(kd_pesos+kp_pesos);
```

```
Wos(2)=2*(-kd_pesos);
```

```
Wos(7)=2*(-kd_pesos);
```

```
Wos(3)=-2*(kd_pesos+kp_pesos);
```

```
Wos(8)=-2*(kd_pesos+kp_pesos);
```

```
Wos(4)=2*(kd_pesos);
```

```
Wos(9)=2*(kd_pesos);
```

```
Wos(5)=2*(ki_pesos);
```

```
Wos(10)=2*(ki_pesos);
```

INICIA PESOS PD COMPLETO

```
W=(rand(nument,nn)-ones(nument,nn))*0;
```

```
Wos=(rand(1,nn)-ones(1,nn))*0;
```

```
Wts=rand(1,1)*0;
```

```
kp_pesos=0.2;
```

```
kd_pesos=0.1;
```

```
ki_pesos=0.3;
```

```
%repartiendo pesos
```

```
for a=1:10
```

```
W(10,a)=0.4;
```

Autora :Carmen García Olloqui	Departamento de Sistemas y Automática	
Tutor : Manuel Ruiz Arahal	Universidad de Sevilla	Página 121



```
end

for a=6:10

W(11,a)=-0.4;
end

for a=1:10

W(2,a)=-0.4;
end

for a=6:10
    W(3,a)=0.4;

    end

for a=1:5
Wos(a)=(kp_pesos);
end

for a=6:10
Wos(a)=(kd_pesos);
end
```

OPTIMIZA SOLOPI

```
kp=0;

for ipi=1:10
    kd=0;
    for jpi=1:10
        ki=0;
        for kpi=1:10
            solopi;
            errores_solopi(ipi,jpi,kpi)=sumpi;
            ki=ki+0.1;
        end
        kd=kd+0.1;
    end
    kp=kp+0.1
end

%halla el error minimo
e_minimo=errores_solopi(1,1,1);
ipi_min=1;
jpi_min=1;
kpi_min=1;
```

Autora :Carmen García Olloqui	Departamento de Sistemas y Automática	
Tutor : Manuel Ruiz Arahal	Universidad de Sevilla	Página 122



```
for ipi=1:10
    for jpi=1:10
        for kpi=1:10
            if errores_solopi(ipi,jpi,kpi)< e_minimo
                e_minimo=errores_solopi(ipi,jpi,kpi);
                ipi_min=ipi;
                jpi_min=jpi;
                kpi_min=kpi;
            end
        end
    end
end

kp=0.1*(ipi_min-1)
kd=0.1*(jpi_min-1)
ki=0.1*(kpi_min-1)

e_minimo
```

SOLOPI

```
%solo PI
%inicia;
a=0.6;
d=3;
uma=1;
uzm=0.1;
vur = zeros(1,d);
baseg=ones(1,40);
basep=ones(1,50);
vr =[ 0.5*baseg -0.25*baseg 0.7*baseg 0*basep -0.14*baseg -
0.2*basep];
nf=length(vr);
ek=0;
yk=0;
sumek=0;
derek=0;

%kp=0.9;
%kd=0.3;
%ki=0.3;

%kp=0.9;pid optimo
%ki=0.3;pid optimo
%kd=0.4;pid optimo
%kp =0.9000 %REF DOBLE
```

Autora :Carmen García Olloqui		Departamento de Sistemas y Automática	
Tutor : Manuel Ruiz Arahal		Universidad de Sevilla	Página 123



```
%kd =0.4000%REF DOBLE
%ki =0.1000%REF DOBLE
%kp =0.8000 %REF mitad
%kd =0.9000%REF mitad
%ki =0.3000%REF mitad

sumpi=0;
for k=1:nf,
    controlador_pd;
    %controlador_pi;
    %salida de la red
    %fin salida de la red
U=U1;
% función satura;
% vectores histórico
vypi(k)=yk;   vk(k)=k;   vupi(k) = U;

if U<uzm & U>-uzm, U=0; else U=U-uzm*sign(U); end;
% saturacion de u
if U>uma, U=uma; end
if U<-uma, U=-uma; end
% zona muerta de u

% retraso de u
Ur = vur(1,d);
for j=d:-1:2,
    vur(1,j) = vur(1,j-1);
end
vur(1,1)=U;
%fin de funcion satura
vupi_sat(k)=Ur;
calcula_salida;
%sumaU=sumaU+U1;
%actualización de X
vepi(k)=ek;
sumpi=sumpi+ek*ek;
end
```

OPTIMIZA ALFA

```
%alfa = 0.001;

alfa=0;
%alfa=0.01;
for indice=1:20
    alfa=alfa+0.0005;
    redneuronas;
    varia_errores(indice,:)=sumaerrores;
    indice
end
```

Autora :Carmen García Olloqui		Departamento de Sistemas y Automática	
Tutor : Manuel Ruiz Arahal		Universidad de Sevilla	Página 124



CONTROLADOR PID

```
% calcula la salida del pi  
%
```

```
ekant=ek;  
ek = vr(k) - yk;  
% integral del error  
sumek = sumek + ek;  
derek=ek-ekant;  
% señal de mando  
U1 = kp*ek + ki*sumek+kd*derek;
```

SISTEMA

```
% vectores histórico  
vy(k)=yk; vk(k)=k; vu(k) = Uc; ve(k)=ek;  
U=Uc;  
if Uc<uzm & Uc>-uzm, U=0; else U=Uc-uzm*sign(Uc); end;
```

```
% saturacion de u  
if Uc>uma, U=uma; end  
if Uc<-uma, U=-uma; end  
% zona muerta de u
```

```
% retraso de u  
% Ur=U;  
Ur = vur(1,d);
```

```
for j=d:-1:2,
```

```
vur(1,j) = vur(1,j-1);  
end
```

```
vur(1,1)=U;
```

```
calcula_salida;
```

TRATA OPTIMIZA ALFA

```
for i=1:indice  
valfa(i)=varia_errores(i,npas)  
end
```

Autora :Carmen García Olloqui	Departamento de Sistemas y Automática	
Tutor : Manuel Ruiz Arahal	Universidad de Sevilla	Página 125



```
for i=1:indice
    min=varia_errores(i,1);
    min_ind=1;

    for j=1:npas
        if varia_errores(i,j)<min
            min=varia_errores(i,j);
            min_ind=j;
        end
        indvalfa(i)=min_ind;
        min_err(i)=min;
    end
end
plot(indvalfa)
figure
plot( min_err)
```