RED NEURONAL CONTROLADORA DE LA TRAYECTORIA DE UN MÓVIL

Ing. Bruno Elio Vargas Tamani
d2700400@unmsm.edu.pe

Facultad de Ingeniería Electrónica de la Universidad Nacional Mayor de San Marcos de Lima Perú

Resumen: Se presenta la simulación de un controlador basado en una red neuronal, que permite que un móvil desarrolle las trayectorias adecuadas desde cualquier punto inicial hasta un punto final deseado. El modelo no lineal del móvil se ha linealizado y usado para el diseño de un controlador lineal; a partir del cual, se generan los datos de entrenamiento de la red. La red neuronal reemplaza al controlador linealizado.

Abstract: The simulation of a neuronal network – based controller is presented, allowing a moving body to develop the suitable paths from any starting point up to a final desired point. The non – linear moving body model has been linearized and used for the design of a linear controller, from which system training data are generated. The neuronal network takes the place of the linearized controller.

Palabras claves: red neuronal, entrenamiento, funcionamiento, trayectoria, móvil, no lineal, lineal, simulación.

I. INTRODUCCIÓN

Las redes neuronales cuentan entre sus múltiples aplicaciones, las referidas al control automático. Es así que en muchos problemas la red neuronal hace el papel de controlador del sistema. Las propiedades de las redes han sido comprobadas con bastante certeza, y es de resaltar la etapa de entrenamiento en la cual se usan algoritmos iterativos resaltando para el uso de control el algoritmo de backpropagation. [Irwin, et al., 1995.[Kartalopowlos, V..]

La no linealidad de los sistemas puede ser aprendida por la red y de esta manera obtener un control satisfactorio es por este motivo el interés del estudio de los sistemas de control solucionados a partir de redes neuronales.

II. PLANTEAMIENTO DEL PROBLEMA

Se tiene un móvil el cual se puede ubicar en cualquier posición inicial de un plano (x,y), con una dirección determinada, se desea que este móvil llegue a una posición deseada (x des, y des), con posición final $\theta = \frac{\pi}{2}$ por la acción de una señal de control δ . que es la dirección de las ruedas del móvil. Variando la posición angular δ, se llevará el móvil a la posición final deseada, momento en el cual cesa la acción de control. Un esquema, que representa las variables de interés del sistema, se muestra en la figura 1.

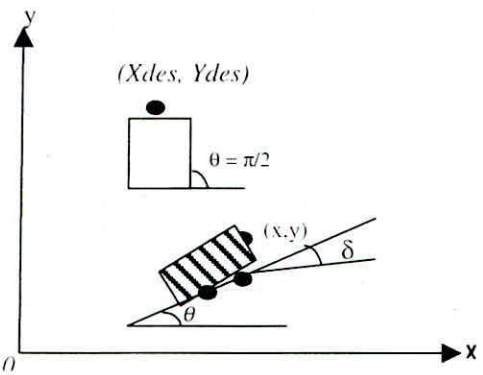


Figura 1. Variables del sistema

La forma en que los cambios instantáneos de δ, afecta la trayectoria del móvil son conocidas.

El modelo del móvil se representa por:

$$\dot{x} = v \cos\theta$$
$$\dot{y} = v \operatorname{sen}\theta$$
$$\dot{\theta} = 2 \operatorname{tg} \delta \qquad (1)$$

donde v es la velocidad lineal del móvil en la dirección θ.

A partir del modelo, se realiza el análisis para decidir sobre la acción de control a tomar en δ, para desarrollar la trayectoria que debe desarrollar el móvil, dada una posición cartesiana inicial y una dirección inicial, en todos los casos el recorrido del móvil lo debe llevar a la posición deseada con la dirección de π/2.

III. LINEALIZACIÓN DEL MODELO Y CONTROL LINEAL

El modelo como se puede ver de (1), es no lineal.

Linealizamos el sistema, usando las variables auxiliares u y z

$$\dot{x} = u$$
$$\dot{\theta} = z \qquad (2)$$

Consideramos a las variables u y z como señales de control, con lo cual el sistema es lineal [Hassan, K., 1996.]. Queremos llevar el error entre x y xdes a cero, antes que se llegue al ydes; de esa manera si hacemos θ igual a π/2, podemos llegar a ydes siguiendo una línea recta con x=xdes. En (2) se puede ver que ahora tendríamos un sistema desacoplado.

Por lo tanto en el sistema lineal, realimentamos el error entre x y xdes con una ganancia adecuada k1, es decir:

$$\dot{x} = u = -k1\ (x - xdes) \tag{3}$$

Según esto último al igualar con (1), vamos a tener que el valor de θdes sería:

$$\theta des = \arccos\left(\frac{-k1}{v}\ (x - xdes)\right) \tag{4}$$

Para lograr que el ángulo real θ tienda a θdes, realimentamos el error entre estos dos ángulos, con una ganancia k2, es decir:

$$\dot{\theta} = z = -k2(\theta - \theta des) \tag{5}$$

Con lo cual al reemplazar (5) en (1) se obtendrá:

$$\delta = \operatorname{arctg}\left(-\frac{k2}{2}\left(\theta - \arccos\left(-\frac{k1}{v}(x - xdes)\right)\right)\right) \tag{6}$$

Esto quiere decir que la acción de control para el móvil, se puede calcular del modelo lineal de acuerdo a (6).

Para ello previamente se debe estabilizar (3) y (5) que es un sistema lineal desacoplado. Podemos notar que la única condición sería tener ganancias k1 y k2 positivas. Se elige k1 y k2 de tal manera que conocida la variación de x, la solución de (4) sea posible, y además se obtenga una señal de control δ en el rango de entre -π/6 a π/6 que es una restricción práctica del sistema. Se ajustará también las ganancias para obtener una respuesta adecuada en la trayectoria del móvil.

El diagrama de bloques del sistema de control linealizado, se muestra en la figura 2. Allí se puede identificar que las señales a realimentar, son el error de posición del eje x y la dirección del móvil. El controlador neuronal a ser diseñado, reemplazará al controlador linealizado diseñado anteriormente.

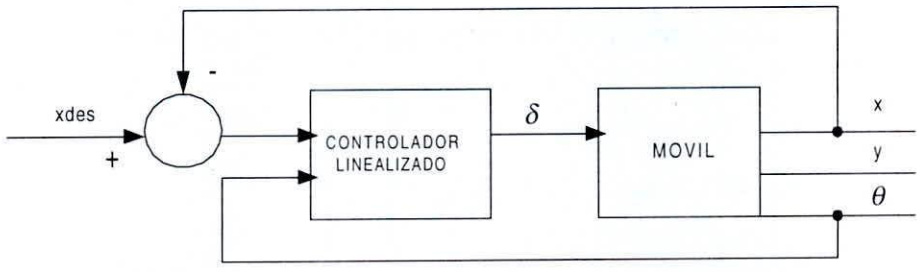


Figura 2. Control Linealizado

IV. RED NEURONAL CONTROLADORA

El controlador linealizado, se simula con un programa de Matlab. Cuando este controlador está bien sintonizado, se que este programa muestran que el control elegido genera trayectorias adecuadas para la posición del móvil a partir de unos puntos equidistantes de prueba. Las trayectorias desarrolladas sirven para guardar en un archivo, los datos de entrada y salidad del controlador, los cuales se usarán para el entrenamiento de la red neuronal controladora que reemplazará después al controlador linealizado.

El diagrama de bloques del sistema de control considerando el control con red neuronal, se muestra en la fig. 3:

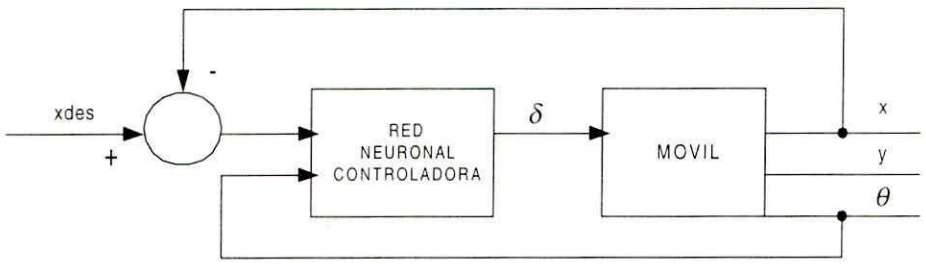


Figura 3. Control neuronal

La red neuronal controlada, es una red multicapa; contiene tres capas, la primera es una capa lineal de dos neuronas para las entradas, la capa oculta consta de siete neuronas, todas con función de salida sigmoidal, la última capa es lineal y consta sólo de una neurona.

El algoritmo de entrenamiento ha sido el de backpropagation; el cual se ha ejecutado con entradas y salidas escaladas, por lo tanto cuando se use este controlador se requiere escalar las entradas y desescalar la salida.

V. SISTEMA DE CONTROL NEURONAL

Conforme se entrenaba a la red paralelamente se simulaba su efecto en el sistema. Es decir se obtuvieron trayectorias que se comparaban con las deseadas y cuyos resultados se muestran en los gráficos que se acompañan en el anexo N° 2. También se presenta la edición del programa en Matlab que simula el sistema con el controlador neuronal y grafica las trayectorias que desarrolla el móvil, en el anexo N° 3. En este caso la prueba del sistema de lazo cerrado, se realiza con los pesoss de la red obtenidos al finalizar la etapa de entrenamiento.

VI. RESULTADOS

Se puede comprobar de los resultados de la simulación que las trayectorias del control neuronal se van aproximando a las deseadas conforme va disminuyendo el error de entrenamiento. (anexo N° 2).

Finalmente, se presenta los resultados de las trayectorias que se producen cuando está funcionando la red neuronal como controlador con pesos fijos; cuando se inicializa con puntos diferentes a los usados para el entrenamiento, con lo cual se confirma la capacidad de generalización de las redes neuronales. El anexo N° 4 presenta la variación temporal de las variables x, y, θ de salida y de la señal de control δ. El anexo N° 5 presenta las trayectorias desarrolladas por el móvil a partir de puntos iniciales escogidos aleatoriamente.

En todos los casos, se superponen las respuestas para todos los puntos de prueba escogidos. Se puede observar que las trayectorias desarrolladas son las más lógicas, esto depende de la elección de las ganancias k_1 y k_2.

Se aprecia que luego que se llega a la posición x deseada, el movimiento sólo se produce en ele eje y, a velocidad constante y con el móvil direccionado con $\theta=\pi/2$, como debe ser. Cuando se llega a valor de y deseado, se debe anular la acción de control.

VII. REFERENCIAS BIBLIOGRAFIAS

Irwin, Warwick, Hunt *"Neural Netwrk Application in Control"*, IEEE Control Engineering Series, 1995.
Hassan, K., Knadil *"Non linear systems"*, Prentice Hall, 1996.
Kartalopowlos, V., "Understanding neural networks and fuzzy logic" stamatios, IEEE Press, 1996.

ANEXO – 1

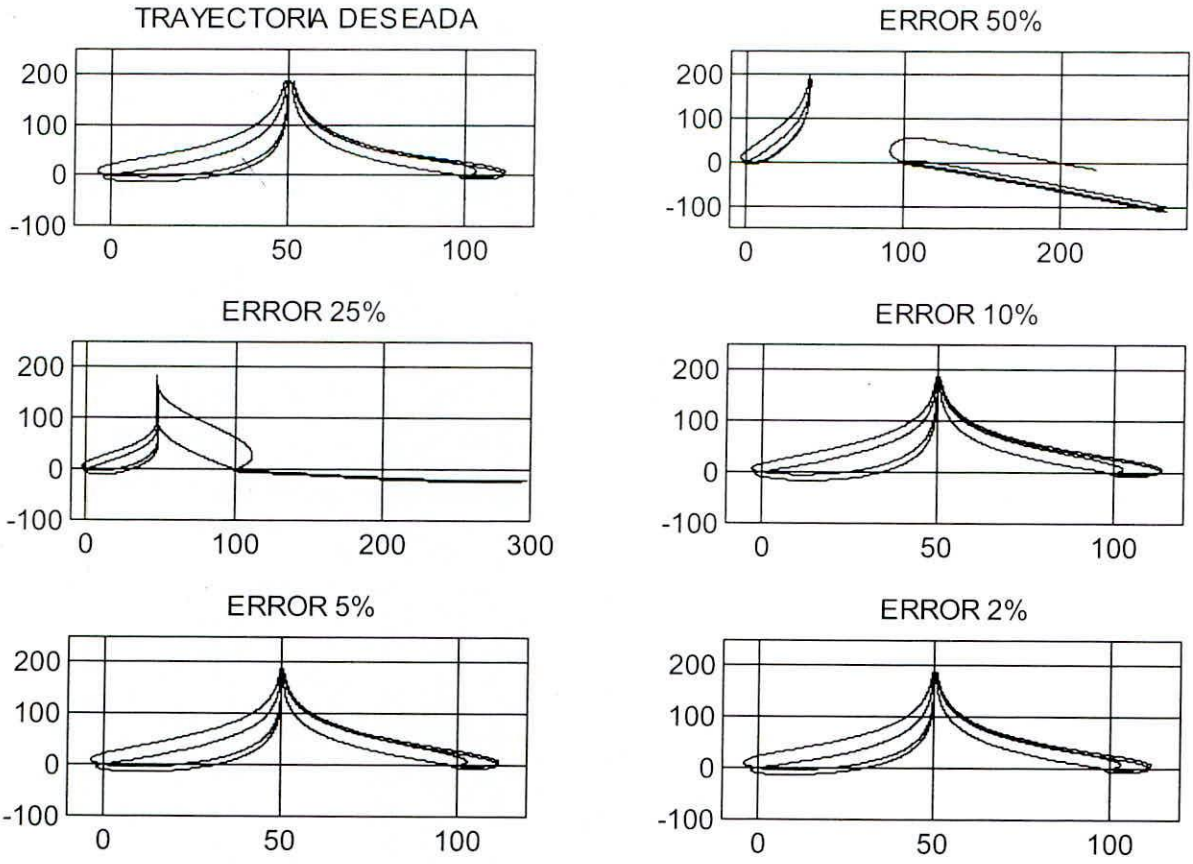
PROGRAMA DE SIMULACION DE LA RED NEURONAL CONTROLADORA PARA DIFERENTES ERRORES

```
%Archivo REVTOTAL.M
%Controlador neurolineal.
clc;
close all;
revorig
ne = 2;
ns=1;
nt = ns + ne;
ne = ne + 1;
nt = ne + ns;
for nro=1:5
if (nro==1)
  pesofile ='pes50';
end
if (nro==2)
  pesofile ='pes25';
end
if (nro==3)
  pesofile ='pes10';
end
if (nro==4)
  pesofile ='pes5';
end
if (nro==5)
  pesofile ='pesli';
end
fid = -1;
while(fid == -1)
 fid = fopen(pesofile);
end
peso = fscanf(fid,'%g');
fclose(fid);
pesz = size(peso);
kn = pesz(1,1);
nm = kn/(ne+ns) - 1;
nm = nm + 1;
kn = kn / nm;
for k = 1:kn
  ki = nm*(k-1) + 1;
  kf = ki + (nm-1);
  pesos(k,1:nm) = (peso(ki:kf,1))';
end
vpeso = pesos(1:ne,:);
vpeso(:,nm) = zeros(ne,1);
wpeso = pesos(ne+1:nt,:);
wpeso = wpeso';
bias = 1;
fid = -1;
while(fid == -1)
  escafile = 'escli';
  fid = fopen(escafile);
end
esca = fscanf(fid,'%g');
fclose(fid);
escasz = size(esca);
kn = escasz(1,1);
kn2 = kn/2;
ascala = esca(1:kn2,1);
ascala = ascala';
bscala = esca(kn2+1:kn,1);
bscala = bscala';
xd=50;
phid=pi/2;
dt=0.5;
ti=0;
tf=100;
r=1;
v=2;
k=1;
for q=1:8
if(q==1)
datos_sal=[0 0 pi/4];
end
if(q==2)
datos_sal=[0 0 -pi/4];
end
if(q==3)
datos_sal=[0 0 (3*pi/4)];
end
if(q==4)
datos_sal=[0 0 -(3*pi/4)];
end
if(q==5)
datos_sal=[100 0 pi/4];
end
if(q==6)
datos_sal=[100 0 -pi/4];
end
if(q==7)
datos_sal=[100 0 (3*pi/4)];
end
if(q==8)
datos_sal=[100 0 -(3*pi/4)];
end
for t=ti:dt:tf
data=[(datos_sal(1,1)-xd)  (datos_sal(1,3))];
```

```
datasca = ascala(1,1:2).* data(1,1:2) + bscala(1,1:2);
entrared = [ datasca bias ];
mentra = entrared * vpeso;
msale = 2.0 ./( 1 + exp(-mentra)) - 1.0;
msale(:,nm) = bias;
sal_delta = msale * wpeso;
sal_delta=(sal_delta-bscala(1,3))./ascala(1,3);
if (nro==1)
x1(k,q)=datos_sal(1,1);
y1(k,q)=datos_sal(1,2);
end
if (nro==2)
x2(k,q)=datos_sal(1,1);
y2(k,q)=datos_sal(1,2);
end
if (nro==3)
x3(k,q)=datos_sal(1,1);
y3(k,q)=datos_sal(1,2);
end
if (nro==4)
x4(k,q)=datos_sal(1,1);
y4(k,q)=datos_sal(1,2);
end
if (nro==5)
x5(k,q)=datos_sal(1,1);
y5(k,q)=datos_sal(1,2);
end
ang(k,1)=datos_sal(1,3);
tiempo(k,1)=t;
phi_p=2*tan(sal_delta);
x_p=v*cos(datos_sal(1,3));
y_p=v*sin(datos_sal(1,3));
datos_sal=datos_sal + dt.*[x_p y_p phi_p];
k=k+1;
end
k=1;
end
end
clc;
figure(1);
subplot(321)
plot(x,y,'w');
hold on;
title('TRAYECTORIA DESEADA');
axis([-10 120 -100 250])
grid
disp('!!!!....PULSAR UNA TECLA PARA CONTINUAR.....')
pause
subplot(322)
plot(x1,y1,'w');
hold on;
title('ERROR 50%');
axis([-10 280 -150 250])
grid
disp('!!!!....PULSAR UNA TECLA PARA CONTINUAR.....')
pause
subplot(323)
plot(x2,y2,'w');
hold on;
title('ERROR 25%');
axis([-10 300 -100 250])
grid
disp('!!!!....PULSAR UNA TECLA PARA CONTINUAR.....')
pause
subplot(324)
plot(x3,y3,'w');
hold on;
title('ERROR 10%');
axis([-10 120 -100 250])
grid
disp('!!!!....PULSAR UNA TECLA PARA CONTINUAR.....')
pause
subplot(325)
plot(x4,y4,'w');
hold on;
title('ERROR 5%');
axis([-10 120 -100 250])
grid
disp('!!!!....PULSAR UNA TECLA PARA CONTINUAR.....')
pause
subplot(326)
plot(x5,y5,'w');
hold on;
title('ERROR 2%');
axis([-10 120 -100 250])
grid
```

ANEXO - 2

TRAYECTORIAS OBTENIDAS CONFORME EL ERROR DE ENTRENAMIENTO VA DISMINUYENDO



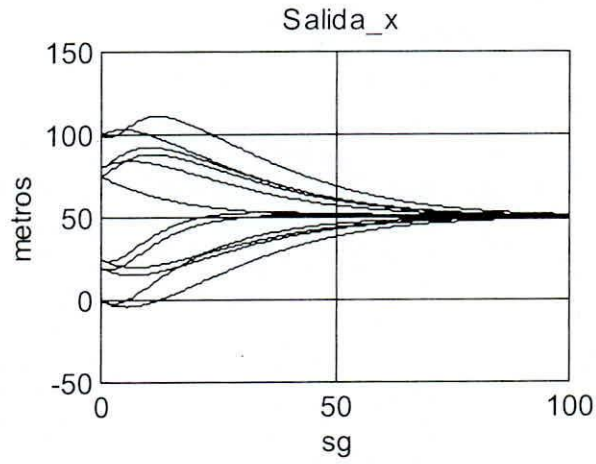
ANEXO – 3
PROGRAMA DE PRUEBA DE CONTROL NEURONAL CON PUNTOS INICIALES DIFERENTES A LOS USADOS EN EL ENTRENAMIENTO

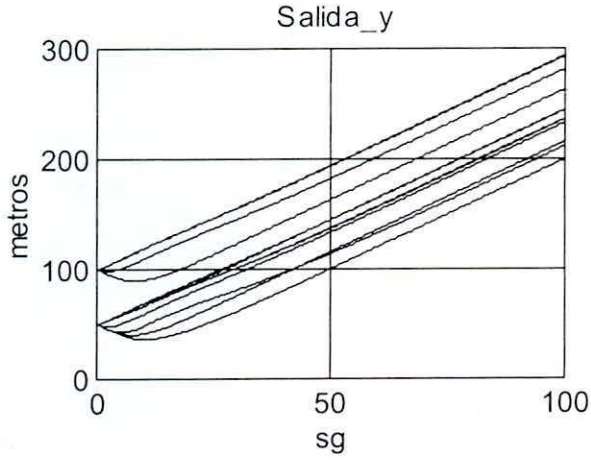
```
%Archivo REVCONT.M
%Controlador neurolineal.
clear;
clc;
close all;
ne = 2;
ns=1;
nt = ns + ne;
ne = ne + 1;
nt = ne + ns;

fid = -1;
while(fid == -1)
  pesofile ='pesli';
  fid = fopen(pesofile);
end
peso = fscanf(fid,'%g');
fclose(fid);
pesz = size(peso);
kn = pesz(1,1);
nm = kn/(ne+ns) - 1;
nm = nm + 1;
kn = kn / nm;
for k = 1:kn
  ki = nm*(k-1) + 1;
  kf = ki + (nm-1);
  pesos(k,1:nm) = (peso(ki:kf,1))';
end

vpeso = pesos(1:ne,:);
vpeso(:,nm) = zeros(ne,1);
wpeso = pesos(ne+1:nt,:);
wpeso = wpeso';
bias = 1;

fid = -1;
while(fid == -1)
  escafile = 'escli';
  fid = fopen(escafile);
end
esca = fscanf(fid,'%g');
fclose(fid);
escasz = size(esca);
kn = escasz(1,1);
kn2 = kn/2;
ascala = esca(1:kn2,1);
ascala = ascala';
bscala = esca(kn2+1:kn,1);

xd=50;
phid=pi/2;
dt=0.5;
ti=0;
tf=100;
r=1;
v=2;
k=1;

for q=1:12

if(q==1)
datos_sal=[0 50 3*pi/4];
end
if(q==2)
datos_sal=[25 50 -3*pi/4];
end
if(q==3)
datos_sal=[0 50 -(3*pi/4)];
end
if(q==4)
datos_sal=[25 50 (3*pi/4)];
end
if(q==5)
datos_sal=[100 50 pi/4];
end
if(q==6)
datos_sal=[75 50 -pi/4];
end
if(q==7)
datos_sal=[100 50 -(3*pi/4)];
end
if(q==8)
datos_sal=[75 50 3*pi/4];
end
if(q==9)
datos_sal=[20 100 3*pi/4];
end
if(q==10)
datos_sal=[20 100 -3*pi/4];
end
if(q==11)
datos_sal=[80 100 pi/4];
end
if(q==12)
datos_sal=[80 100 -pi/4];
end
```

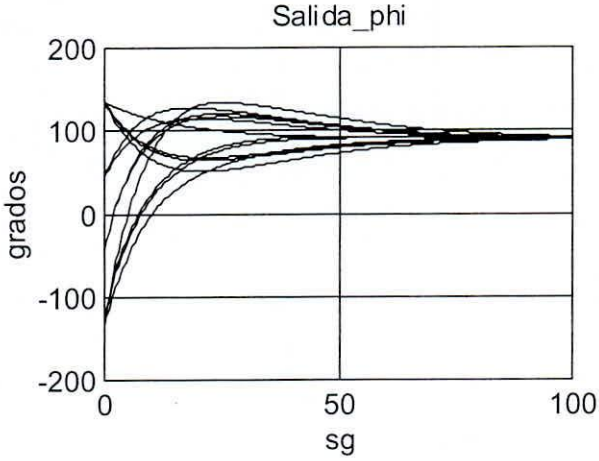
```
bscala = bscala';
data=[(datos_sal(1,1)-xd)  (datos_sal(1,3))];
datasca = ascala(1,1:2).* data(1,1:2) +
bscala(1,1:2);
entrared = [ datasca  bias ];
mentra = entrared * vpeso;
msale = 2.0 ./( 1 + exp(-mentra))  - 1.0;
msale(:,nm) = bias;
sal_delta = msale * wpeso;

sal_delta=(sal_delta-bscala(1,3))./ascala(1,3);

x(k,q)=datos_sal(1,1);
y(k,q)=datos_sal(1,2);
ang(k,q)=datos_sal(1,3);
sal_con(k,q)=sal_delta;
tiempo(k,q)=t;
phi_p=2*tan(sal_delta);
x_p=v*cos(datos_sal(1,3));
y_p=v*sin(datos_sal(1,3));
datos_sal=datos_sal + dt.*[x_p y_p phi_p];

k=k+1;
end
k=1;
end
tiempo=ti:dt:tf;

clc;
figure(1);
subplot(221)
plot(tiempo,x,'w');
grid
for t=ti:dt:tf
title('Salida_x');
xlabel('sg');
ylabel('metros');
grid
subplot(222)
plot(tiempo,y,'w');
title('Salida_y');
xlabel('sg');
ylabel('metros');
grid
subplot(223)
plot(tiempo,ang*180/pi,'w');
hold on;
title('Salida_phi');
xlabel('sg');
ylabel('grados');
grid
subplot(224)
plot(tiempo,sal_con*180/pi,'w');
hold on;
title('Salida del controlador');
xlabel('sg');
ylabel('grados');
grid

figure(2)
plot(x,y,'w');
hold on;
title('TRAYECTORIAS');
xlabel('metros');
ylabel('metros');
```

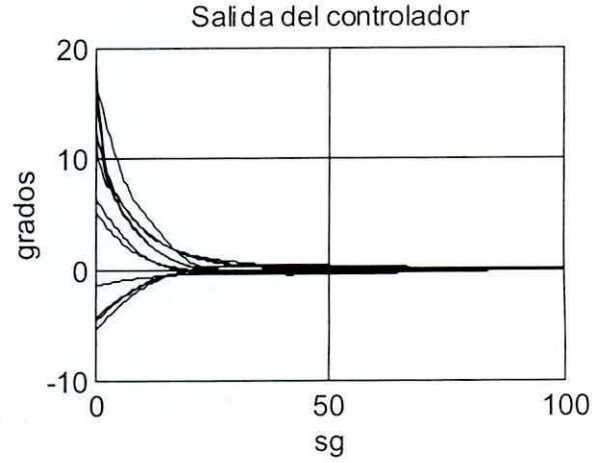
ANEXO - 4

RESULTADO DEL CONTROL NEURONAL PARA PUNTOS INICIALES DIFERENTES A LOS DE ENTRENAMIENTO









ANEXO - 5

TRAYECTORIAS OBTENIDAS CON EL CONTROL NEURONAL