

Diseño de un control neuronal aplicado a un brazo robótico de alta precisión

Design of a neural control applied to a high-precision robotic arm

Projeto de um controle neural aplicado a um braço robótico de alta precisão

Leonela Del Roció de la A Salinas

ldelaa@institutos.gob.ec

<https://orcid.org/0000-0002-0320-4397>

Centro Tecnológico Naval Santa Elena, Ecuador

Rolando Edison Carrera Fernández

r.carrera@hotmail.com

<https://orcid.org/0000-0002-1965-4915>

Universidad Católica de Santiago de Guayaquil, Ecuador

Mario Fernando Martínez Marca

mmartine@espol.edu.ec

<https://orcid.org/0000-0003-2916-2305>

Escuela Superior Politécnica del Litoral Guayaquil, Ecuador

RESUMEN

El objetivo principal de este trabajo es presentar una propuesta de control neuronal basado en un modelo de seguimiento de referencia aplicado a un brazo robótico de alta precisión, donde se comenzará explicando y deduciendo el modelo matemático del brazo robótico el cual se lo representará mediante un modelo de espacios, donde la variable de entrada será el torque mecánico y la variable de salida será el control del ángulo y la posición de robot en cada instante de tiempo, lo que se busca mediante el control neuronal es mediante los diferentes cambios en las variables de entrada y de salida encontrar y calibrar un sistema de control que garantice el seguimiento de referencia, es decir dada una trayectoria o posición deseada queremos que el brazo robótico siga con exactitud y presión la posición o trayectoria deseada. El sistema de control de seguimiento de referencia está basado en método de inteligencia artificial de machine learning especialmente en el campo de redes neuronales inteligentes.

Palabras claves: Diseño, control neuronal, brazo robótico.

ABSTRACT

The main objective of this work is to present a neural control proposal based on a reference tracking model applied to a high-precision robotic arm, where it will begin by explaining and deducing the mathematical model of the robotic arm which will be represented by a space model, where the input variable will be the mechanical torque and the output variable will be the angle and position control of the robot at each instant of time, what is sought through neural control is through the different changes in the variables input and output find and calibrate a control system that guarantees reference tracking, that is, given a desired path or position, we want the robotic arm to accurately follow the desired position or path with pressure. The reference tracking control system is based on artificial intelligence method of machine learning especially in the field of intelligent neural networks.

Keywords: Design, neural control, robotic arm.

RESUMO

O objetivo principal deste trabalho é apresentar uma proposta de controle neural baseada em um modelo de rastreamento de referência aplicado a um braço robótico de alta precisão, onde começará explicando e deduzindo o modelo matemático do braço robótico que será representado por um espaço modelo, onde a variável de entrada será o torque mecânico e a variável de saída será o controle de ângulo e posição do robô em cada instante de tempo, o que se busca através do controle neural é através das diferentes mudanças nas variáveis de entrada e saída encontrar e calibrar um sistema de controle que garanta o rastreamento de referência, ou seja, dado um caminho ou posição desejada, queremos que o braço robótico siga com precisão a posição ou caminho desejado com pressão. O sistema de controle de rastreamento de referência é baseado no método de inteligência artificial de aprendizado de máquina, especialmente no campo de redes neurais inteligentes.

Palavras-chave: Design, controle neural, braço robótico.

Introducción

Los métodos utilizados para el diseño basado en técnicas de control suelen incluir la construcción de modelos matemáticos que describen la dinámica del sistema controlado y la aplicación de técnicas analíticas para inferir reglas de control.

Técnicas clásicas suelen fallar cuando es difícil obtener un modelo representativo del sistema real. Esta situación ocurre cuando hay incertidumbre, el sistema real es demasiado complejo o se violan los supuestos basados en las técnicas de síntesis de la estrategia de control.

En los últimos años, existe un creciente interés por los sistemas basados en redes neuronales aplicados a la resolución de problemas de control. Tal aumento se debe principalmente a su capacidad para aplicarse con éxito, ya que las estrategias tradicionales aún no han demostrado su eficacia. La diferencia entre la forma en que las redes neuronales procesan la información y la forma en que funcionan los sistemas de control tradicionales es la capacidad de procesamiento colectivo que les da la capacidad de responder rápidamente a entradas sensoriales complejas.

Por otro lado, los algoritmos de control complejos son muy limitados en el tiempo que lleva operarlos, la tercera diferencia, y quizás la más importante, es que el control que realiza el sistema nervioso se obtiene a través del aprendizaje, mientras que en los controladores clásicos es necesario definir un algoritmo. Por lo tanto, para implementar un controlador aritmético efectivo, se requiere un conocimiento completo de la planta controlada, lo cual es difícil de lograr en la práctica.

El controlador neuronal implementa una forma de control adaptativo donde el controlador es una red no lineal de neuronas dispuestas en múltiples capas y los parámetros adaptativos son los pesos de las conexiones entre neuronas.

Este tipo de microcontrolador tiene las siguientes características básicas: capacidad de procesamiento colectivo, adaptabilidad e inmunidad a interferencias, lo que conduce a un control más efectivo en presencia de incertidumbre, complejidad y no linealidades del sistema. Cuando los parámetros del objeto de control son desconocidos, puede usar diferentes métodos de evaluación.

Las redes neuronales también se pueden utilizar para esta tarea y es particularmente apropiado cuando se pretende usar otra red neuronal como controlador. En este trabajo nos proponemos: Simulación de control de robots basada en métodos de redes neuronales y compararlos con los métodos clásicos aplicados al mismo robot.

Desarrollo

Redes neuronales

RNA (Red Neural Artificial) es un modelo matemático basado en el comportamiento biológico de las neuronas y la estructura del cerebro. También se puede ver como un sistema inteligente que realiza tareas diferentes a las de las computadoras modernas.

El proceso de procesamiento posterior de la información es muy rápido, existen tareas muy complejas como el reconocimiento de patrones y la clasificación requiere demasiado tiempo y esfuerzo incluso para las computadoras más poderosas de la actualidad, pero el cerebro humano está mejor equipado para manejarlos, a menudo sin un esfuerzo visible (por ejemplo, considere a un lector que reconoce una cara familiar entre muchas otras). El cerebro puede ser visto como un sistema muy complejo. Su unidad básica, la neurona, tiene una distribución muy grande de

conexiones entre ellas (se estima que 10 mil millones de neuronas y 60 mil millones de conexiones neuronales en la corteza cerebral).

Aunque hay muchos tipos diferentes de neuronas biológicas, en la Figura 1 se muestra un diagrama simple de una de las más comunes. Vemos que incluye:

1. El cuerpo central, llamado soma, que contiene el núcleo celular.
2. Una prolongación del soma, el axón.
3. Una ramificación terminal, las dendritas.
4. Una zona de conexión entre una neurona y otra, conocida como

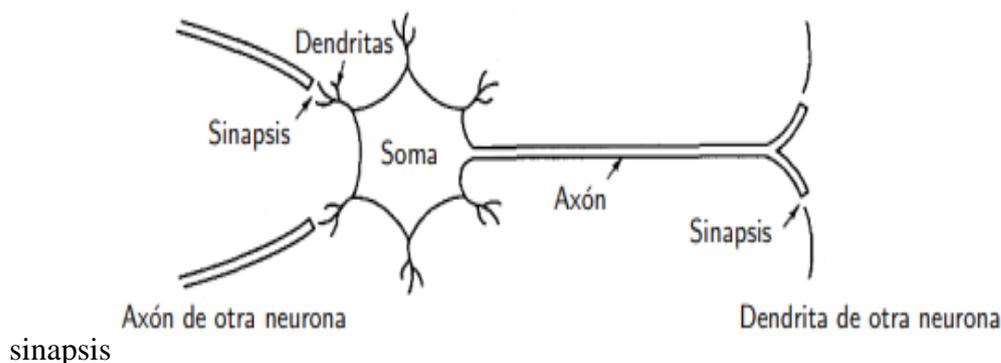


Figura1. Modelo simplificado de una neurona biológica

La función principal de las neuronas es transmitir los impulsos nerviosos, las cuales viajan a través de las neuronas comenzando en las dendritas hasta llegar al axón, donde se transmiten a otra neurona a través de una conexión sináptica.

La forma en que respondemos a los estímulos del mundo exterior y cómo aprendemos sobre ellos está directamente relacionada con las conexiones neuronales en el cerebro, y las redes neuronales artificiales son un intento de emular este hecho.

Modelo matemático de una red neuronal

Las redes neuronales artificiales (RNA) son información estructuras que subyacen al funcionamiento de alguna computadora programas, tales redes consisten en elementos simples de del mismo tipo (neuronas), interconectadas de cierta manera, cuya funcionalidad es similar a la mayoría de los Funciones elementales de una neurona biológica. La comunicación puede ser fuerte (significativa) y débil (insignificante hasta que esté ausente), que en neural artificiales redes se caracteriza por un coeficiente de ponderación de acoplamiento (w). Cada neurona se caracteriza por su corriente estado (excitado o inhibido), que está determinado por la función de activación.

La RNA se puede visualizar en la forma de un gráfico dirigido, cuyos vértices serán corresponden a las neuronas, y los arcos que conectan los vértices corresponderán a conexiones sinápticas o pesos.

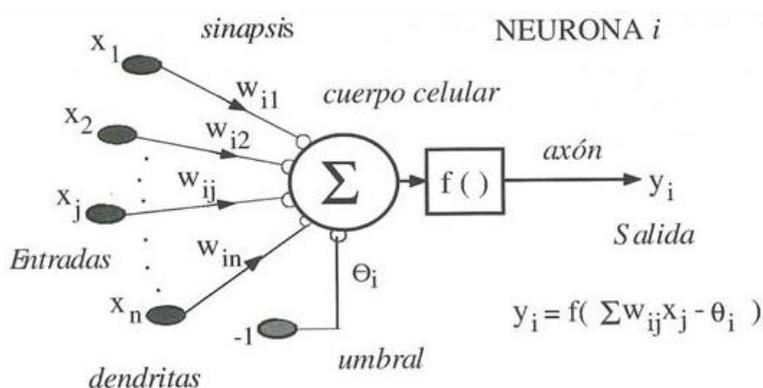


Figura 2. Esquema principal de red neuronal inteligente

Donde;

Las señales de entrada, la información recibida por las neuronas artificiales, son variables independientes de N valores de entrada y se multiplican por sus respectivos pesos, es decir, en resumen, el vector de entrada se multiplica por el vector de peso, produciendo un conjunto lineal de valores de entrada y pesos que llamamos función de ponderación:

$$X * W^t = (x_1, x_2, \dots, x_n) * \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix} = \sum_{i=1} x_i * w_i$$

A continuación, se aplicará una función activación:

$$\phi\left(\sum_{i=1} x_i * w_i\right)$$

Y finalmente, el resultado especificado se expande a la salida. Este valor puede ser la entrada de la nueva neurona, creando así la red neuronal, o puede ser el resultado final de nuestra variable de retroalimentación

Función de activación.

La función de activación es una función de transferencia generada por una combinación lineal de pesos y entradas. Es decir, cómo pasar información entre conexiones de salida. La información se puede enviar sin cambios.

Describe la función de identidad. o no envíe la información.

Las funciones de activación generalmente hacen que el modelo no sea lineal porque queremos poder resolver problemas que hacen que la red sea más compleja. Las funciones de activación más conocidas o utilizadas son:

1. Función Escalón, (similar a la función binaria.)

$$\phi(x) = \begin{cases} 0 & \text{si } x < 0. \\ 1 & \text{si } x \geq 0. \end{cases}$$

2. Función Sigmoidal.

$$\phi(x) = \frac{1}{1 + e^{-x}}$$

3. Función Rectificadora (ReLU).

$$\phi(x) = \max\{0, x\}, \text{ siendo } x \geq 0.$$

4. Función Tangente Hiperbólica.

$$\phi(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

Podemos concluir diciendo que una red neuronal es como un gráfico, con una capa de entrada que recibe una señal de entrada, la envía a través de estímulos a la siguiente capa oculta, que se encarga de procesar la información y pasarla a la siguiente capa. A continuar, hasta la próxima clase de clase. El último nivel, el nivel de salida, el nivel que nos da la respuesta.

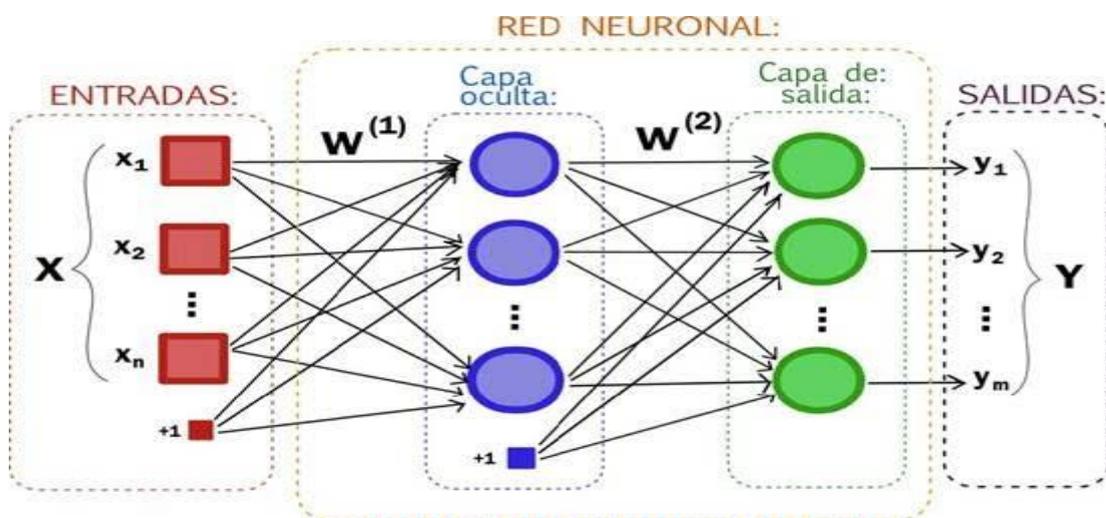


Figura 3. Arquitectura de red neuronal multicapa.

Sistema de estructura variable

Los Sistemas de Estructura Variable están conformados por subsistemas o estructuras continuas que son conmutadas en forma alternativa la cual consisten en un conjunto de subsistemas continuos con una lógica de conmutación adecuada y, como resultado, las acciones de control son funciones discontinuas de las perturbaciones del estado del sistema (si son accesibles para la medición) y las entradas de referencia. Se mostrará que el papel dominante en la teoría VSS lo juegan los modos deslizantes y la idea central de diseñar algoritmos de control VSS consiste en imponer este tipo de movimiento en algunas variedades en los espacios de estado del sistema.

Considere los sistemas de estructura variable gobernados por

$$\begin{aligned} \dot{x}(t) &= f(x, u, t), \\ u &= \begin{cases} u^+(x, t) & \text{if } s(x) > 0 \\ u^-(x, t) & \text{if } s(x) < 0, \end{cases} \\ s(x) &= [s_1(x), \dots, s_m(x)]^t, \end{aligned}$$

Donde $x \in \mathbb{R}^N$ es el estado del sistema y u pertenece \mathbb{R}^M y son entradas del sistema con funciones continuas:

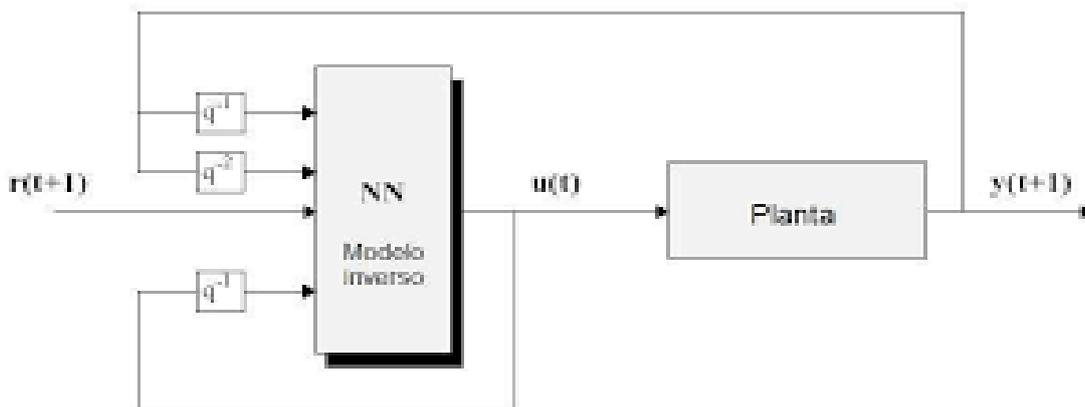


Figura 4. Esquema de control por estructura de variable

Modelo matemático de un brazo robótico

Para la obtención del modelo matemático de la Figura 5, se simplificó el sistema por medio de una masa ubicada en el centro de cada eslabón y una masa puntual al final del eslabón dos que representa la carga del manipulador

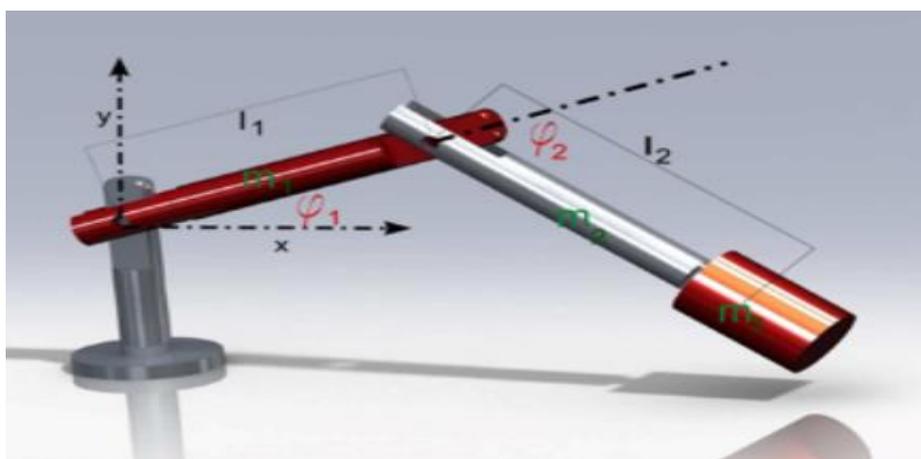


Figura 5. Esquema básico de un brazo robótico.

El primer paso para la obtención del modelo dinámico es conocer la ubicación de las masas, con respecto al origen R_i , que componen el sistema; como se puede observar en las ecuaciones:

$$r_1 = \begin{bmatrix} l_1 \cos(\varphi_1) / 2 \\ l_1 \sin(\varphi_1) / 2 \end{bmatrix}$$

$$r_2 = \begin{bmatrix} l_1 \cos(\varphi_1) + l_2 \cos(\varphi_1 + \varphi_2) / 2 \\ l_1 \sin(\varphi_1) + l_2 \sin(\varphi_1 + \varphi_2) / 2 \end{bmatrix}$$

$$r_3 = \begin{bmatrix} l_1 \cos(\varphi_1) + l_2 \cos(\varphi_1 + \varphi_2) \\ l_1 \sin(\varphi_1) + l_2 \sin(\varphi_1 + \varphi_2) \end{bmatrix}$$

Donde;

L_1 y m_1 son las longitudes en metros y las masas en kilogramos del eslabón respectivo i φ_1 es la posición angular en radianes del eslabón uno con respecto a otro a la horizontal φ_2 es la posición angular en radianes del eslabón 2 con res respecto al otro, a partir de las posiciones se encuentran la energía potencial del sistema

$$U = m_1 g r_{1y} + m_2 g r_{2y} + m_3 g r_{3y}$$

Donde g es la gravedad m/s^2 , a la energía cinética del sistema se calculó implementado la siguiente ecuación

$$K = \frac{1}{2}(m_1(\dot{r}_1x^2 + \dot{r}_1y^2)^2 + m_2(\dot{r}_2x^2 + \dot{r}_2y^2)^2 + m_3(\dot{r}_3x^2 + \dot{r}_3y^2)^2 + I_1\varphi_1^2 + I_2(\varphi_1 + \varphi_2)^2)$$

Y finalmente definido el Lagrangiano del sistema del brazo robótico como:

$$L(\varphi_1, \varphi_2) = K - U$$

Una vez determinada el Lagrangiano del sistema aplicamos las ecuaciones movimiento de Euler Lagrange y obtenemos las ecuaciones diferenciales que describen el funcionamiento del brazo robótico.

Metodología

Diseño y simulación de brazo robótico en MATLAB

En el desarrollo y simulación de brazo robótico en Matlab utilizamos las ecuaciones de movimiento de deducidas del modelo matemático

Código del programa

```
clc
clear all
close all

ts=0.1;
t=0:ts:60;

% 1. Condiciones iniciales del manipulador
l1=0.5;
l2=0.65;
h=0.25;

q1(1) = 0*(pi/180);
q2(1) = 30*(pi/180);
q3(1) = 30*(pi/180);

xr(1)=l1*cos(q2(1))*cos(q1(1))+l2*cos(q2(1)+q3(1))*cos(q1(1));
yr(1)=l1*cos(q2(1))*sin(q1(1))+l2*cos(q2(1)+q3(1))*sin(q1(1));
zr(1)=h+l1*sin(q2(1))+l2*sin(q2(1)+q3(1));

% 3) Referencias deseadas
% xrd = 0.5;
% yrd = -0.8;
% zrd = 0.2;

xrd = .6*cos(0.2*t);
yrd = .6*sin(0.2*t);
zrd = 0.7*ones(1,length(t));
% zrd =0.7+0.01*t;

xrd_p= -.6*0.2*sin(0.2*t);
```

```
    yrd_p= .6*0.2*cos(0.2*t);
%       zrd_p = 0.01*ones(1,length(t));

    zrd_p= 0*ones(1,length(t));

for k=1:length(t)

    %a) Errores de control
    xre(k) = xrd(k) - xr(k);
    yre(k) = yrd(k) - yr(k);
    zre(k) = zrd(k) - zr(k);

    e = [xre(k);yre(k);zre(k)];

    %b) Matriz Jacobiana
    J=[ -11*cos(q2(k))*sin(q1(k))-12*cos(q2(k)+q3(k))*sin(q1(k)) -
11*cos(q1(k))*sin(q2(k))-12*cos(q1(k))*sin(q2(k)+q3(k)) -
12*cos(q1(k))*sin(q2(k)+q3(k));...
    11*cos(q2(k))*cos(q1(k))+12*cos(q2(k)+q3(k))*cos(q1(k)) -
11*sin(q1(k))*sin(q2(k))-12*sin(q1(k))*sin(q2(k)+q3(k)) -
12*sin(q1(k))*sin(q2(k)+q3(k));...
    0 11*cos(q2(k))+12*cos(q2(k)+q3(k)) 12*cos(q2(k)+q3(k))];

    %c) Matriz de ganancia
    K = [0.2 0 0;...
        0 0.2 0;...
        0 0 0.2];

    %d) Ley de control
%       v = inv(J)*K*e;

    hd_p=[xrd_p(k) yrd_p(k) zrd_p(k)]';

    %d) Ley de control
    v = inv(J)*(hd_p+K*tanh(e));

    q1p_ref(k)=v(1);
    q2p_ref(k)=v(2);
    q3p_ref(k)=v(3);

    q1(k+1)=q1(k)+q1p_ref(k)*ts;
    q2(k+1)=q2(k)+q2p_ref(k)*ts;
    q3(k+1)=q3(k)+q3p_ref(k)*ts;

    xr(k+1)=11*cos(q2(k+1))*cos(q1(k+1))+12*cos(q2(k+1)+q3(k+1))*cos(q1(k+1));
    yr(k+1)=11*cos(q2(k+1))*sin(q1(k+1))+12*cos(q2(k+1)+q3(k+1))*sin(q1(k+1));
    zr(k+1)=h+11*sin(q2(k+1))+12*sin(q2(k+1)+q3(k+1));
end

pasos=5; fig=figure;
set(fig,'position',[10 60 980 600]);
axis square; cameratoolbar
grid on;
axis([-3 3 -3 3 0 1.5]);

Arm_Parameters;
```

```
M1=Arm_Plot3D(0,0,0,0,0,q1(1),q2(1),q3(1),0);  
  
for i=1:pasos:length(t)  
  
    delete (M1)  
    M1=Arm_Plot3D(0,0,0,0,0,q1(i),q2(i),q3(i),0);hold on;  
    plot3(xr(1:i),yr(1:i),zr(1:i),'r','linewidth',3);  
    pause(0.1)  
  
end  
%% Graficas  
figure('Name','Errores')  
subplot(311)  
plot(t,xre,'linewidth',2), grid on  
legend('Error en x')  
xlabel('Tiempo'), ylabel('Error [m]')  
subplot(312)  
plot(t,yre,'g','linewidth',2), grid on  
legend('Error en y')  
xlabel('Tiempo'), ylabel('Error [m]')  
subplot(313)  
plot(t,zre,'g','linewidth',2), grid on  
legend('Error en z')  
xlabel('Tiempo'), ylabel('Error [m]')  
  
figure('Name','Acciones de control')  
subplot(311)  
plot(t,q1p_ref,'linewidth',2), grid on  
legend('Velocidad q1')  
xlabel('Tiempo'), ylabel('Velocidad [rad/s]')  
subplot(312)  
plot(t,q2p_ref,'g','linewidth',2), grid on  
legend('Velocidad q2')  
xlabel('Tiempo'), ylabel('Velocidad [rad/s]')  
subplot(313)  
plot(t,q3p_ref,'g','linewidth',2), grid on  
legend('Velocidad q3')  
xlabel('Tiempo'), ylabel('Velocidad [rad/s]')
```

Cabe indicar que se crearon entornos gráficos con funciones especiales para graficas las cuales no abordaremos en este artículo:

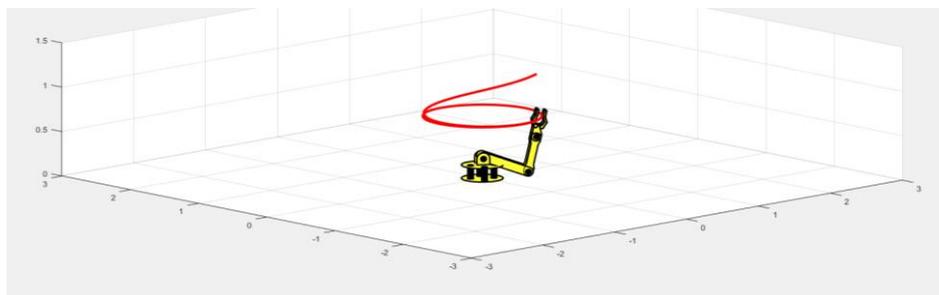


Figura 6 Simulación del brazo robótico

Diseño del controlador neuronal

Para el diseño del control neuronal se utilizó la herramienta de Simulink donde se definieron las variables de entrada el torque mecánico y de salida o variable de control la posición angular del brazo robótico.

Para la variable de entrada es decir el torque mecánico tiene un rango de límite entre $[0, 200]$ Nm, para simular este torque mecánico usaremos un impulso aleatorio definida entre ese rango, y para la variable de control o de salida su intervalo de estará entre -0.5 , a 0.5 metros, y lo que buscamos con el control neuronal es que el todo momento el brazo robótico siga una trayectoria de referencia con el mayor grado de precisión, en la figura 7 se muestra el circuito de control del sistema.

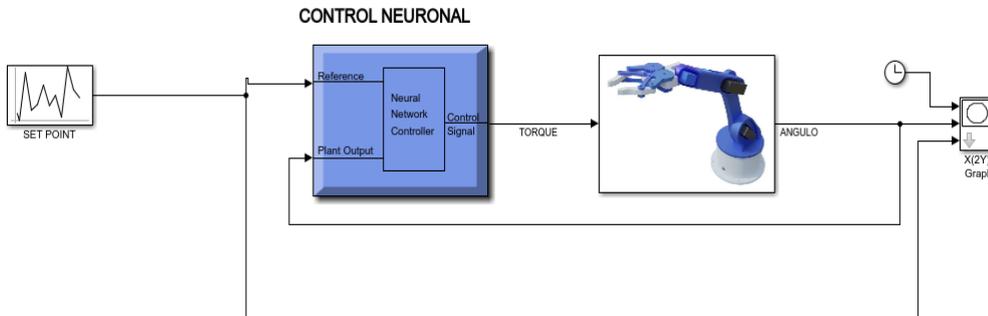


Figura 7. Circuito de control neuronal del brazo robótico.

Análisis y resultados obtenidos

Antes de aplicar el control neuronal inteligentes necesitamos entrenar al controlador para lo cual utilizamos, realizamos pruebas inyectando diferentes valores de torque mecánico de manera aleatoria y viendo como respondía el sistema.

Tabla1. Valores experimentales.

Variable de entrada Torque mecánico T_m	Variable de salida o de control Posición angular φ
63.7048	0.10
17.9901	-0.15
22.3411	-0.24
27.2585	0.45
135.7305	0.29
157.9926	0.34
99.0012	0.19

Una vez realizados estas mediciones se obtuvieron los siguientes resultados:

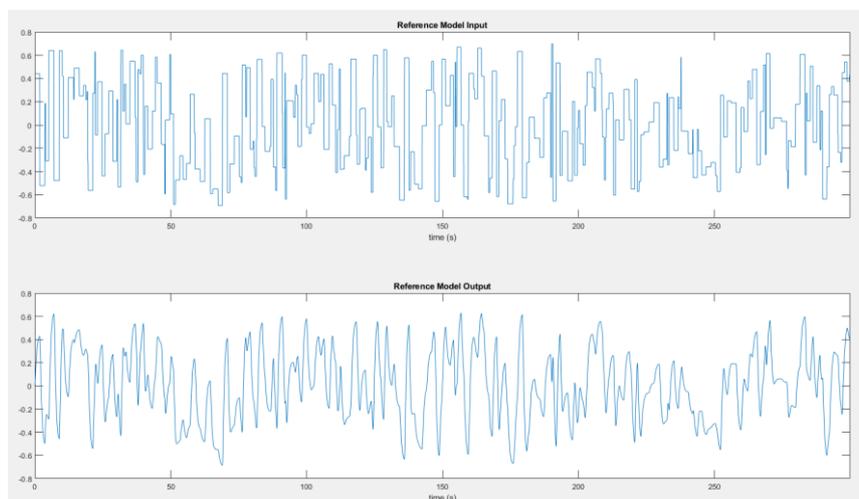


Figura 8. Resultados en proceso de aprendizaje del controlador neuronal.

Luego de entrenar al controlador neuronal con los datos antes mencionados el controlador neuronal encontró un buen modelo matemático que seguirá la referencia o trayectoria desea en la Figura 9, podemos apreciar como el controlador neuronal una vez entrenado con los datos de prueba, encuentra una acción de control la cual sigue la referencia con alta precisión.

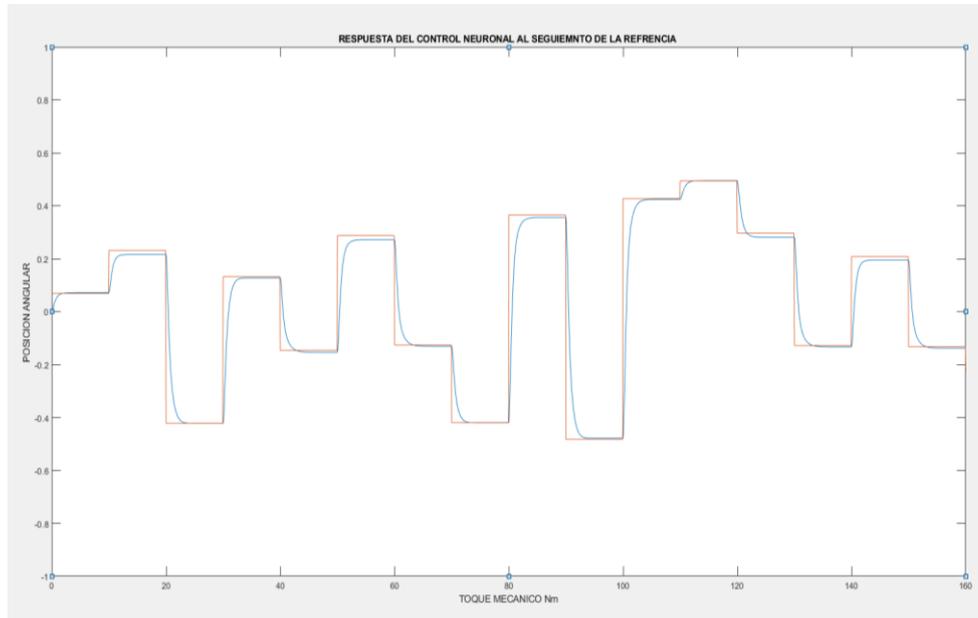


Figura 9. Acción de control del controlador neuronal.

Conclusiones

El controlador neuronal inteligente empleado para el desarrollo de un sistema de control de referencia en un brazo robótico responde de manera eficiente a los diferentes cambios de referencia teniendo un error casi despreciable, por lo cual podemos concluir diciendo que el modelo de control neuronal es una excelente opción para diseñar un sistema de control cuando los sistemas son altamente no lineales y cuando no se puede modelar matemáticamente el problema, estos tipos de controladores tienen la capacidad de que a partir de datos experimentales y mediciones que hagamos entre las variables de entrada y salida podemos encontrar una ley de control que siga la referencia.

Referencias Bibliográficas

- Cano Mendoza, D. F. (2005). Diseño de controladores predictivos con redes neuronales.
- Maldonado, A. C., Guerrero, O. G., & Sandoval, J. O. (2018). Procesamiento de señales EMG en un sistema embebido para el control neuronal de un brazo robótico. *REVISTA COLOMBIANA DE TECNOLOGIAS DE AVANZADA (RCTA)*, 2(32), 139-147.
- Quintero, C. M., Rodríguez, J. L. D., & García, A. P. (2013). Aplicación de redes neuronales al control de velocidad en motores de corriente alterna. *Revista Colombiana de Tecnologías De Avanzada (RCTA)*, 2(20).
- Ramírez, C. A. G., García, D. F. B., & Gómez, J. O. M. (2009). Diseño e implementación de un sistema de control basado en redes neuronales artificiales para el brazo robótico Move Master. *Ingeniería solidaria*, 5(9), 68-74.
- Tosini, M. A. (2009). Proyecto NNGen: Aplicación de controladores neuronales a la navegación autónoma. In *XI Workshop de Investigadores en Ciencias de la Computación*.
- Utili, L., Barriga, B. E. N. J. A. M. I. N., & Domínguez, D. (2007). Diseño y fabricación de un brazo robot de cinco grados de libertad articulado verticalmente: diseño y selección. In *Octavo congreso iberoamericano de ingeniería mecánica*. Disponible en: <https://congreso.pucp.edu.pe/cibim8/pdf/19/19-18.pdf>