



UNIVERSIDAD DEL AZUAY
FACULTAD DE CIENCIA Y TECNOLOGÍA
ESCUELA DE INGENIERÍA ELECTRÓNICA

**“Modelado y Simulación de un Robot Manipulador de 3
GDL para el Control de la Trayectoria Mediante
Inteligencia Artificial.”**

Trabajo de graduación previo a la obtención del título de:

INGENIERO ELECTRÓNICO

Autores:

PABLO AGUSTÍN CORDERO DELGADO

BRIAN TORRES ALVES

Director:

Ing. HUGO MARCELO TORRES SALAMEA Ph.D.

CUENCA, ECUADOR

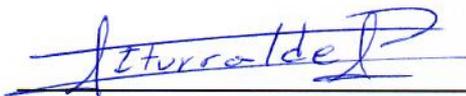
2019

Modelado y Simulación de un Robot Manipulador de 3 GDL para el Control de la Trayectoria Mediante Inteligencia Artificial.

RESUMEN

Debido al crecimiento industrial, la inclusión de robots en sus procesos y la complejidad de las tareas que estos deben realizar, se requieren controladores de alto rendimiento. Por esta razón, se modeló y simuló un robot manipulador de 3 grados de libertad (GDL) para el control de la trayectoria utilizando controladores basados en inteligencia artificial: PD+I Difuso y redes neuronales. Estos, incluyendo un PID convencional, fueron evaluados al seguir dos tipos de trayectorias, resultando el PD+I difuso como el más eficaz sin importar la presencia de cambios bruscos de dirección en las trayectorias planteadas.

Palabras clave: Robot manipulador, Controlador, Lógica Difusa, Red Neuronal, PID.



Ing. Daniel Iturralde Piedra. Ph.D.
Coordinador de Carrera.



Ing. Hugo Marcelo Torres Salamea Ph.D.
Director de Trabajo de Titulación



Pablo Agustín Cordero Delgado
Autor



Brian Torres Alves
Autor

Modeling and Simulation of a 3 GDL Manipulator Robot to Control Trajectories Through Artificial Intelligence

Abstract

Due to industrial growth, the presence of robots in their processes and the complexity of the tasks they must perform, high-performance controllers are required. For this reason, a 3-degree-of-freedom (GDL) manipulator robot was modeled and simulated to control the trajectory by using artificial intelligence-based controllers: PD + I Diffuse and neural networks. These, including a conventional PID, were evaluated by following two types of trajectories. It was concluded that the diffuse PD + I was the most effective trajectory, regardless the presence of sudden changes in the direction of the trajectories.

Keywords: Robot manipulator, Controller, Fuzzy Logic, Neural Network, PID.

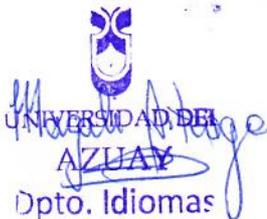
Eng. Daniel Iturralde Piedra, Ph.D.
Faculty School Director.

Eng. Hugo Marcelo Torres Salamea, Ph.D.
Thesis Director

Pablo Agustín Cordero Delgado

Brian Torres Alves

Autors



UNIVERSIDAD DEL
AZUAY
Dpto. Idiomas



Translated by
Mg. Magali Arteaga

Modelado y Simulación de un Robot Manipulador de 3 GDL para el Control de la Trayectoria Mediante Inteligencia Artificial

Pablo Agustín Cordero Delgado
Escuela de Ingeniería Electrónica
Universidad del Azuay
Cuenca, Ecuador
pablocor.11@gmail.com

Brian Torres Alves
Escuela de Ingeniería Electrónica
Universidad del Azuay
Cuenca, Ecuador
bri_toal@hotmail.com

Resumen: Debido al crecimiento industrial, la inclusión de robots en sus procesos y la complejidad de las tareas que estos deben realizar, se requieren controladores de alto rendimiento. Por esta razón, se modeló y simuló un robot manipulador de 3 grados de libertad (GDL) para el control de la trayectoria utilizando controladores basados en inteligencia artificial: PD+I Difuso y redes neuronales. Estos, incluyendo un PID convencional, fueron evaluados al seguir dos tipos de trayectorias, resultando el PD+I difuso como el más eficaz sin importar la presencia de cambios bruscos de dirección en las trayectorias planteadas.

PALABRAS CLAVE: Robot manipulador, Controlador, Lógica Difusa, Red Neuronal, PID.

I. INTRODUCCIÓN

Actualmente el seguimiento de trayectorias en robots manipuladores industriales es de vital importancia para la maximización del rendimiento de producción industrial, por lo que se han diseñado e implementado nuevos controladores, tales como PID (Proporcional Integrativo Derivativo), control adaptativo, entre otros, que minimicen errores en la trayectoria provocados por elementos externos tales como: perturbaciones, vibraciones, campos electromagnéticos, entre otros [1, 2, 3].

Debido a las exigencias industriales actuales, nuevos controladores deben ser desarrollados con el fin de mejorar las propiedades de los métodos convencionales tomando como base, algoritmos de inteligencia artificial, como lo son: redes neuronales, algoritmos genéticos, lógica difusa, sistemas expertos, etc.[4].

En el caso de las redes neuronales, se han desarrollado controladores que parten del proporcional-derivativo con una compensación mediante una red de dos capas, aplicada en varios manipuladores [5]. En una publicación de J. Kern et al. [6] hacen uso de una red neuronal *feed forward backpropagation*, que resulta superior comparada con un controlador basado en un par calculado clásico, al seguir determinadas trayectorias en un robot manipulador de 5 GDL. Así mismo, se hace uso de una red de tres capas que trabaja paralelamente con un controlador PD, con el fin de realizar el seguimiento de una trayectoria en un robot industrial AdeptOne, que se observa en la Fig. 1 [7].



Fig. 1 Robot AdeptOne [7].

Otro estudio, hace uso de una red neuronal *feed-forward backpropagation*, entrenada con información proveniente de la cinemática directa, aplicada a la cinemática inversa de un manipulador de 3 GDL [8].

En cuanto a los algoritmos genéticos, estos se han aplicado en robots manipuladores tales como: 2R, PUMA 560 y no redundantes para el seguimiento de trayectorias con presencia de obstáculos fijos, mediante su cinemática directa [9]. En un robot manipulador industrial se presenta el caso donde se debe realizar el planeamiento y control de una trayectoria punto a punto, con el fin de optimizar las posibles rutas en relación con su longitud y dificultad [10]. Del mismo modo, se ha realizado una investigación que propone el desarrollo y comparación de un controlador PID optimizado mediante un algoritmo genético para el seguimiento de una trayectoria en un robot 3R con un controlador PID clásico [11]. El resultado de esta comparación se puede visualizar en la Fig. 2.

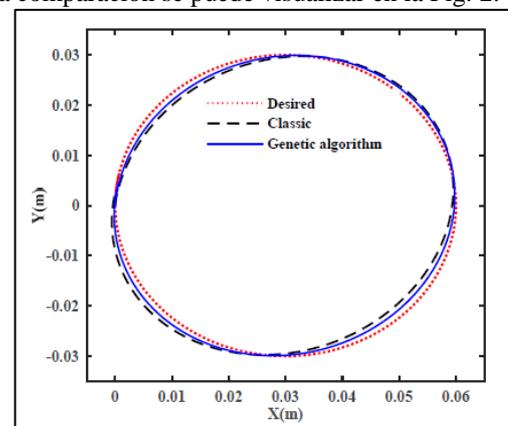


Fig. 2 Seguimiento de una trayectoria con distintos controladores [11].

Existen otros tipos de controladores, como muestra un estudio realizado en un robot de 3 GDL, en el cual se controla la trayectoria a través del uso de distintos métodos de optimización para la sintonización de un controlador PID,

siendo el basado en algoritmos genéticos el mejor de los propuestos [12]. V. Banga et al. [13] realiza un estudio en el cual optimiza las articulaciones de un robot manipulador, utilizando algoritmos genéticos, reemplazando así el proceso de prueba y error para conseguir la mejor combinación de ángulos en cada una de estas.

Una investigación realizada por F. Botsali et al. [14] propone el diseño de un controlador difuso para el seguimiento de trayectoria de un manipulador con el propósito de reducir las vibraciones producidas al realizar líneas rectas. De igual manera, A. Nawrocka et al. [15] implementan un controlador basado en lógica difusa en un robot de 2 GDL, haciendo énfasis en el tiempo de recuperación del robot frente a perturbaciones, obteniendo un controlador de mayor calidad comparado con un PID tradicional.

La lógica difusa a su vez puede ser utilizada para mejorar las características de controladores convencionales, tal como demuestra S. Al-khayyt [16] al implementar sistemas difusos dentro de un controlador PID. Consiguiendo así, una reducción de error en estado estático, al igual que una mejora en el comportamiento del robot.

Otro trabajo importante es el desarrollado por H. Chaudhary et al. [17], en el que desarrollan un controlador PD+I sintonizado por lógica difusa para el control de posición y fuerza de un robot manipulador industrial de 6 GDL. De esta manera, el controlador es capaz de tomar una decisión incluso al existir incertidumbre y ausencia de parámetros dinámicos del robot dentro de un ambiente restrictivo.

Por otro lado, se han desarrollado controladores híbridos, que combinan distintos métodos de computación flexible tales como: ANNFL1 (Controlador adaptativo de redes neuronales basado en lógica difusa), HANNFL2 (Controlador jerárquico adaptativo de redes neuronales basado en lógica difusa) y ANNIT2FL3 (Controlador adaptativo de redes neuronales basado en intervalos de lógica difusa tipo 2) [18]. Zhao et al. [19], presentan la optimización, por medio de un algoritmo genético, de un controlador difuso PID. Obteniéndose una mejora notable en los tiempos de respuesta y minimización de errores del brazo robótico. En otra publicación se utiliza un controlador de redes neuronales optimizado con algoritmos genéticos en un robot manipulador de 5 GDL [20].

Otro método utilizado son los sistemas expertos. L. Tudor y A. Moise [21] desarrollaron un controlador, para el seguimiento de trayectoria de un manipulador en donde se incluye un sistema de aprendizaje desarrollado con sistemas neuro-difusos para mejorar su rendimiento y eficiencia. La estructura de este controlador se puede apreciar en la Fig. 3.

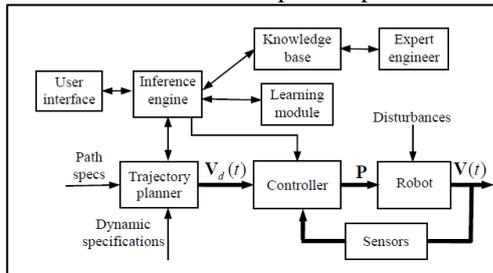


Fig. 3 Estructura de un controlador basado en un sistema experto automático [21].

Producto de un análisis profundo del estado del arte, esta investigación tiene como objetivo modelar y simular un robot manipulador de 3 GDL para el control de la trayectoria

mediante inteligencia artificial. Con el fin de obtener varias respuestas en el seguimiento trayectorias frente a diversos elementos externos, se seleccionan y realizan 3 controladores: dos con redes neuronales y uno mediante lógica difusa. Estos serán comparados con un controlador PID convencional y entre sí, utilizando el software *Simulink*, al igual que *Robotics Toolbox for MATLAB* desarrollado por Peter Corke y así determinar qué controlador tiene mayor eficacia para seguir distintas trayectorias.

II. FUNDAMENTACIÓN TEÓRICA

A. Matrices de Transformación Homogénea

Es una de las técnicas de la robótica más utilizadas. La representación de estas se realiza en coordenadas de $n + 1$ dimensiones [3]. El uso de esta herramienta se debe a sus múltiples aplicaciones, entre las cuales se pueden destacar las siguientes:

- Representación de la posición, al igual que la orientación de un sistema que rota y se traslada con respecto a uno de referencia.
- La transformación de un vector que es expresado en “coordenadas móviles” y su representación en un sistema de coordenadas fijo.

La representación de una matriz de transformación homogénea viene estructurada tal y como se puede visualizar en (1), es decir como una matriz \mathbf{T} de 4×4 , donde intervienen sub-matrices correspondientes a: rotación, traslación, perspectiva y escalado [3].

$$\mathbf{T} = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{p}_{3 \times 1} \\ \mathbf{f}_{1 \times 3} & \mathbf{w}_{1 \times 1} \end{bmatrix} = \begin{bmatrix} \text{Rotación} & \text{Traslación} \\ \text{Perspectiva} & \text{Escalado} \end{bmatrix} \quad (1)$$

En donde:

- \mathbf{R} corresponde a la matriz de rotación.
- \mathbf{f} representa la transformación de perspectiva.
- \mathbf{w} es el escalado global.
- \mathbf{p} corresponde a la matriz de traslación.

Normalmente en el campo de la robótica, la matriz \mathbf{f} se considera nula o cero y a la matriz \mathbf{w} se la considera 1. Algo que se puede acotar es que la matriz de transformación homogénea, dependiendo de su aplicación, puede ser escrita de distintas formas, una de ellas viene expresada como la matriz \mathbf{T} que se muestra a continuación en (2) [3].

$$\mathbf{T} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{n} & \mathbf{o} & \mathbf{a} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Analizando (2), se puede acotar que \mathbf{n} , \mathbf{o} y \mathbf{a} es un conjunto de vectores que representan la orientación del robot, mientras que \mathbf{p} es un vector que describe la posición.

B. Lógica Difusa

Método basado en inteligencia artificial que permite analizar información con poco grado de precisión en conjuntos borrosos que se asocian a un valor lingüístico en específico. Estos conjuntos definen una función de pertenencia ($u_a(t)$). Así mismo, se utilizan reglas que definen las acciones que el programa realizará. Entre las partes que componen este método, se encuentran las siguientes: entradas, fuzzificación, base de reglas, inferencia, defuzzificación y sus salidas [22]. Esto se puede observar en la Fig. 4.

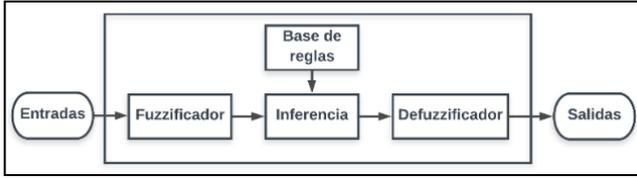


Fig. 4 Estructura de un sistema de lógica de difusa.

La fuzzificación es aquel proceso encargado de transformar los valores de entradas en variables lingüísticas y clasificarlas según su valor de pertenencia. Existen varios métodos de fuzzificación, entre los cuales destacan:

- Singleton: Uno de los métodos más utilizados en sistemas de control, que considera a todos los valores de entrada discretos como difusos y una función de pertenencia en específico.
- No singleton: Es una función con forma de campana, misma que se centra en un valor de entrada x . Este método se define por (3) donde: a hace referencia a la amplitud y σ al ancho de la función de pertenencia $\mu_{A'}(x')$ [23].

$$\mu_{A'}(x') = a \cdot \exp \left[- \left(\frac{x' - x}{\sigma} \right)^2 \right] \quad (3)$$

Las reglas establecen las condiciones para que cada valor de entrada se clasifique en un valor difuso o función de pertenencia. Estas se definen mediante la estructura si-y-entonces (*if/and/then*). Las funciones de pertenencia son aquellas a las cuales son clasificadas cada una de las entradas que conforman el sistema [23, 24, 25].

El resultado difuso es calculado por la inferencia, el cual es obtenido de acuerdo a las reglas propuestas anteriormente. Estos datos difusos se envían al proceso defuzzificación que los transforma en variables interpretables en la salida del sistema. Existen varios métodos de defuzzificación tales como [23]:

- Defuzzificador por máximo.
- Defuzzificador por media de centros.
- Defuzzificador por centro de área.

C. Redes Neuronales (ANN)

Las redes neuronales en inteligencia artificial intentan emular las neuronas biológicas con la intención de crear sistemas que procesen información de manera paralela, distribuida, al igual que adaptativa. Estas se componen por unidades que se conectan entre sí con su respectivo peso, determinando el signo y fuerza de la conexión. Para obtener el potencial h_i de la neurona se usa (4) a partir de las entradas y los pesos [22].

$$h_i = \sum w_{ij} x_j \quad (4)$$

En donde:

- w_{ij} es el peso de la conexión.
- x_j es el conjunto de entradas.

La función de activación es aquella que arroja un resultado cuando el conjunto de entradas contiene los valores correctos, y no lo hace cuando tienen valores erróneos. Esta no debe ser lineal debido a que, al analizar una función lineal, esta podría

colapsar. La activación a_i está regida por la expresión mostrada en (5) [23].

$$a_i = g(\sum w_{ij} x_j) \quad (5)$$

En donde:

- g es la función de activación.

Las redes *feed-forward* están organizadas por capas, donde cada una de estas ingresa los valores que su capa anterior ha enviado. Este tipo de redes normalmente se desglosan según el número de capas: multicapa y de una sola capa o perceptrones [22].

La estructura más simple de las redes multicapa se basa en una capa oculta que posee múltiples neuronas, las cuales, al trabajar en conjunto, tienen la capacidad de representar todo tipo de función continua con precisión arbitraria. Así mismo, existe una capa de salida y una de entrada que están definidas en base al dimensionamiento de la red según sea el problema a resolver. Algunas de las redes multicapa que se usan actualmente son [26, 27]:

- *Feed-Forward Backpropagation*.
- *Cascade-Forward Backpropagation*.
- *Radial Basis* (Base Radial).
- *Madaline*.
- *Adaline*.
- Red Kohonen (Modelos Competitivos).

D. Feed-Forward Backpropagation (FFBP)

Se caracterizan por utilizar un algoritmo de retro-propagación basado en la regla de aprendizaje Widrow-Hoff para redes multicapa. Este se encarga de actualizar los pesos y bias de la red en la misma dirección en la que el rendimiento se reduce más rápido (gradiente negativo). Este algoritmo viene descrito por (6) [28].

$$x_{k+1} = x_k - \alpha_k g_k \quad (6)$$

En donde:

- x_{k+1} es la siguiente iteración
- x_k es el vector de los pesos y "bias" actuales.
- α_k es la tasa de aprendizaje.
- g_k es el gradiente actual.

En este tipo de redes también se utilizan algoritmos de aprendizaje que se seleccionan en base a la aplicación y cantidad de recursos disponibles; entre los principales están: Levenberg-Marquadt, gradiente negativo, gradiente conjugado, entre otros [6]. Su estructura se muestra en la Fig. 5.

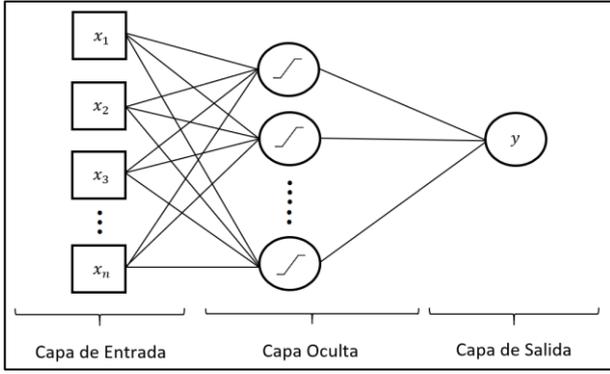


Fig. 5 Estructura de una red neuronal *feed-forward backpropagation*.

La ecuación (7) describe la salida y de la red neuronal de la figura anterior [29].

$$y = f^o(w^b + \sum_{j=1}^k w^o f^h(w_j^b + \sum_{i=1}^n w_{ji}^h x_i)) \quad (7)$$

En donde:

- f^o es la función de activación de la capa de salida.
- f^h es la función de activación de la capa oculta.
- w^b es el valor de los “bias” en la capa de salida.
- w_j^b es el valor de los “bias” en la capa oculta.
- w^o es el valor de los pesos en la capa de salida.
- w_{ji}^h es el valor de los “bias” en la capa oculta.
- x_i son las entradas de la red neuronal.

E. Cascade-Forward Backpropagation (CFBP)

Este tipo de redes son similares a las *Feed-Forward Backpropagation*. Utilizan el algoritmo de retropropagación, al igual que los demás métodos de aprendizaje. La diferencia radica en que estas incluyen una conexión desde cada una de las entradas hacia cada una de las capas subsiguientes [30]. La estructura de esta red se muestra en la Fig. 6.

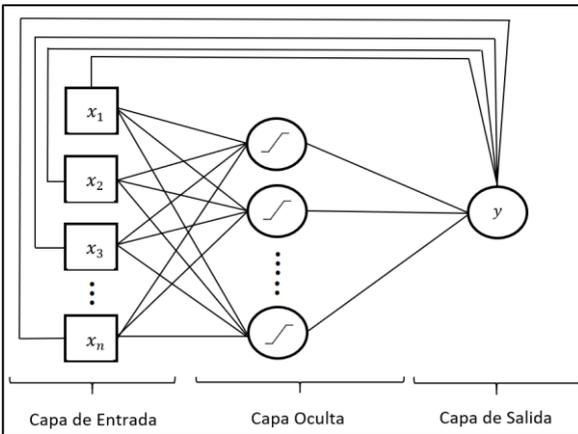


Fig. 6 Estructura de una red neuronal *cascade-forward backpropagation*.

La ecuación (8) describe la salida y de la red neuronal de la figura anterior [29].

$$y = \sum_{i=1}^n f^i w_i^i x_i + f^o(w^b + \sum_{j=1}^k w_j^o f^h(w_j^b + \sum_{i=1}^n w_{ji}^h x_i)) \quad (8)$$

En donde:

- f^o es la función de activación de la capa de salida.

- f^h es la función de activación de la capa oculta.
- f^i es la función de activación entre la capa de entrada y la capa de salida.
- w^b es el valor de los “bias” en la capa de salida.
- w_j^b es el valor de los “bias” en la capa oculta.
- w^o es el valor de los pesos en la capa de salida.
- w_i^i es el valor de los pesos entre la capa de entrada y la capa de salida.
- w_{ji}^h es el valor de los “bias” en la capa oculta.
- x_i son las entradas de la red neuronal.

III. ROBOT REDUNDANTE PLANAR 3R

El robot en el cual se va a trabajar es uno de tipo planar 3R. Debido al diseño de este, cada una de las articulaciones podrá moverse únicamente en un rango de 0 a 180°, cumpliéndose con el movimiento permitido de la mayoría de los servomotores que existen actualmente. A su vez se implementarán varios controladores para el seguimiento de determinadas trayectorias, utilizando distintos métodos basados en inteligencia artificial al igual que convencionales. A continuación, se adjunta el modelo en tres dimensiones del robot en cuestión, en la Fig. 7.



Fig. 7 Robot redundante planar 3R.

IV. CINEMÁTICA

La cinemática de un robot estudia su movimiento sin considerar las fuerzas que intervienen en este, referenciado a un sistema de coordenadas. Dentro del estudio de la cinemática existen dos problemas: el directo e inverso. Ambos tienen como objetivo relacionar las velocidades y posiciones de los extremos del robot con sus articulaciones, tal como se puede observar en la Fig. 8 [3].

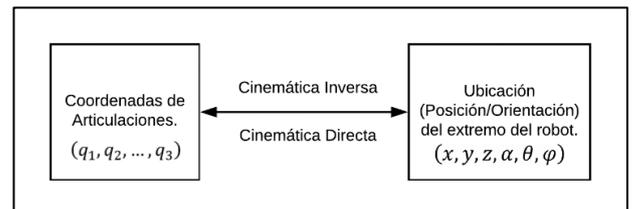


Fig. 8 Relación cinemática directa vs inversa.

A. Cinemática Directa

La cinemática directa da a conocer la ubicación del robot (posición y orientación) siempre y cuando se conozcan los parámetros que adoptarán las articulaciones. Para la obtención del modelo cinemático se utilizan métodos específicos, entre los cuales se pueden destacar dos: geométricos y basados en cambio de sistema de referencia (Denavit-Hartenberg).

El algoritmo de Denavit-Hartenberg (D.H.) es un método matricial que propone la ubicación que debe adoptar cada sistema de coordenadas (S_i) asociadas a cada eslabón dentro de una cadena de articulaciones para obtener las ecuaciones que rigen a la cinemática de un robot. Para este método, es necesario tener en cuenta tres cosas: la correcta elección del sistema de coordenadas que rige a cada eslabón, la realización de cuatro transformaciones a ese sistema y el orden de estas. Estas cuatro transformaciones son rotaciones y traslaciones que relacionan al elemento ($i - 1$) con el elemento (i) del sistema completo a ser analizado con respecto a un sistema de referencia móvil. Generalmente, estas deben seguir el orden denotado a continuación [3, 31]:

- Rotación alrededor del eje de coordenadas z_{i-1} , desplazándose así un ángulo θ_i .
- Traslación en el eje de coordenadas z_{i-1} , desplazándose de esta manera una distancia d .
- Traslación en el eje de coordenadas x_i , desplazándose de esta manera una distancia d .
- Rotación alrededor del eje de coordenadas x_i , desplazándose así un ángulo α_i .

De este modo, la expresión que rige este método se adjunta en (9).

$$A_i = Rotz(\theta_i)T(0,0,d_i)T(a_i,0,0)Rotx(\alpha_i) \quad (9)$$

Aplicando el método de D.H. en el robot mostrado en la Fig. 9, se obtiene la siguiente tabla de valores, correspondiente a sus parámetros cinemáticos. Esta puede observarse en la Tabla 1.

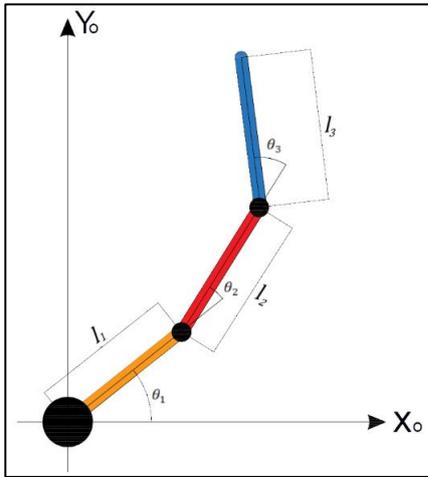


Fig. 9 Vista superior del robot planar 3R con sus respectivos parámetros cinemáticos.

En donde:

- $\theta_1, \theta_2, \theta_3$ son los desplazamientos angulares de cada articulación.

- l_1, l_2, l_3 son las longitudes de cada eslabón.

TABLA 1 PARÁMETROS CINEMÁTICOS DEL ROBOT PLANAR 3R.

# Articulación	θ_i	d_i	a_i	α_i
1	θ_1	l_0	l_1	0
2	θ_2	0	l_2	0
3	θ_3	0	l_3	0

Con estos parámetros, la orientación y posición del robot se registrarán por la matriz T , que se adjunta en (10).

$$T = {}^0A_3 = {}^0A_1 {}^1A_2 {}^2A_3 \quad (10)$$

En las expresiones (11), (12) y (13), pueden observarse las matrices de transformación homogéneas correspondientes a cada articulación.

$${}^0A_1 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & l_1 \cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & l_1 \sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

$${}^1A_2 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & l_2 \cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & l_2 \sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

$${}^2A_3 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & l_3 \cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & l_3 \sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

Resolviendo (10), se obtiene (14).

$$T = \begin{bmatrix} \cos(\theta_1 + \theta_2 + \theta_3) & -\sin(\theta_1 + \theta_2 + \theta_3) & 0 & x \\ \sin(\theta_1 + \theta_2 + \theta_3) & \cos(\theta_1 + \theta_2 + \theta_3) & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

En donde los valores de x e y se muestran en (15) y (16), respectivamente.

$$x = l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \quad (15)$$

$$y = l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \quad (16)$$

B. Cinemática Inversa

El método inverso tiene como finalidad obtener las coordenadas articulares del robot (q_1, q_2, \dots, q_n) para que el efector final tenga una determinada posición (p) y orientación ($[n, o, a]$) en el espacio. Para la resolución de la cinemática inversa se utilizan dos métodos [3]:

- **Método Geométrico.** - Utiliza relaciones trigonométricas y geométricas para resolver triángulos construidos al considerar ciertos elementos del robot. Obteniendo así, valores de las variables articulares.
- **Ecuaciones de la Cinemática Directa.** - Usa la representación inversa de la matriz de transformación homogénea, mostrada en (17), para

obtener las variables articulares deseadas. Está basada en la cinemática directa del robot.

$$\mathbf{T}^{-1} = \begin{bmatrix} n_x & n_y & n_z & -n^T p \\ o_x & o_y & o_z & -o^T p \\ a_x & a_y & a_z & -a^T p \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (17)$$

Aplicando (17) en las matrices denotadas en (11), (12) y (13), se obtiene la representación inversa mostrada en (18), (19) y (20).

$$({}^0 A_1)^{-1} = \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 & -l_1 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & -l_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (18)$$

$$({}^1 A_2)^{-1} = \begin{bmatrix} \cos(\theta_2) & \sin(\theta_2) & 0 & -l_2 \\ -\sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (19)$$

$$({}^2 A_3)^{-1} = \begin{bmatrix} \cos(\theta_3) & \sin(\theta_3) & 0 & -l_3 \\ -\sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (20)$$

Resolviendo (21) y (22), al considerar las expresiones mostradas en (23) y (24) [32], se obtiene la cinemática inversa en (25) y (26) para el robot mostrado en la Fig. 10.

$$({}^0 A_1)^{-1} \mathbf{T} = {}^1 A_2 {}^2 A_3 \quad (21)$$

$$({}^1 A_2)^{-1} ({}^0 A_1)^{-1} \mathbf{T} = {}^2 A_3 \quad (22)$$

$$\theta_2 = \theta_3 \quad (23)$$

$$l_1 = l_2 = l_3 = l \quad (24)$$

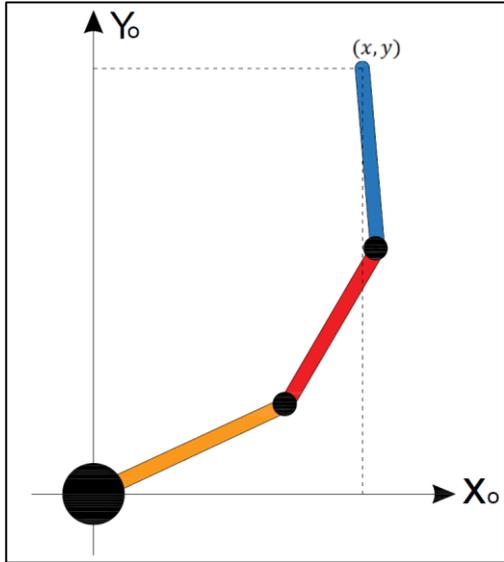


Fig. 10 Robot correspondiente a las consideraciones tomadas.

$$\theta_2 = \pm \cos^{-1} \left(\pm \frac{\sqrt{x^2 + y^2}}{2l} - \frac{1}{2} \right) \quad (25)$$

$$\theta_1 = \text{atan2}(y, x) - \theta_2 \quad (26)$$

V. DINÁMICA ROBOT 3R

Analiza el movimiento del robot al igual que su rendimiento. Se deben considerar ciertos elementos que afectan el movimiento del robot, tales como: la inercia, fuerza centrífuga, gravitacional y efecto Coriolis. De no ser considerados, se puede producir inestabilidad en el control. Los parámetros a considerar para obtener el torque ($\boldsymbol{\tau}$) se aprecian en (27) [33].

$$\boldsymbol{\tau} = \mathbf{A}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) \quad (27)$$

En donde:

- $\mathbf{A}(\mathbf{q})$ se refiere a la matriz de energía cinética.
- $\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}})$ representa el vector de fuerza centrífuga y efecto Coriolis.
- $\mathbf{g}(\mathbf{q})$ hace referencia al vector de fuerza gravitacional.

A. Matriz de Energía Cinética

En cuanto a la matriz de energía cinética, esta debe ser simétrica de dimensión n . Se puede apreciar en (28) la regla que rige a esta [33].

$$\mathbf{A}(\mathbf{q}) = \sum_{i=1}^n (\mathbf{m}_i \mathbf{J}_{vi}^T \mathbf{J}_{vi} + \mathbf{J}_{wi}^T \mathbf{I}_i \mathbf{J}_{wi}) \quad (28)$$

Así mismo, se puede representar por (29).

$$\mathbf{A}(\mathbf{q}) = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \mathbf{A}_{13} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \mathbf{A}_{23} \\ \mathbf{A}_{31} & \mathbf{A}_{32} & \mathbf{A}_{33} \end{bmatrix} \quad (29)$$

En donde sus respectivos elementos son los siguientes:

$$\begin{aligned} \mathbf{A}_{11} = & I_1 + I_2 + I_3 + l_{c1}^2 m_1 + (l_1^2 + 2l_1 l_{c2} C_2 + l_{c2}^2) m_2 \\ & + (l_1^2 + l_2^2 + l_{c3}^2 + 2l_1 l_{c3} C_{23} + 2l_1 l_2 C_2 \\ & + 2l_2 l_{c3} C_3) m_3 \end{aligned}$$

$$\begin{aligned} \mathbf{A}_{12} = & I_2 + I_3 + (l_{c2}^2 + l_1 l_{c2} C_2) m_2 \\ & + (l_2^2 + l_{c3}^2 + 2l_2 l_{c3} C_3 + l_1 l_2 C_2 \\ & + l_1 l_{c3} C_{23}) m_3 \end{aligned}$$

$$\mathbf{A}_{13} = I_3 + (l_{c3}^2 + l_2 l_{c3} C_3 + l_1 l_{c3} C_{23}) m_3$$

$$\begin{aligned} \mathbf{A}_{21} = & I_2 + I_3 + (l_1 l_{c2} C_2 + l_{c2}^2) m_2 \\ & + (l_2^2 + l_{c3}^2 + 2l_2 l_{c3} C_3 + l_1 l_2 C_2 \\ & + l_1 l_{c3} C_{23}) m_3 \end{aligned}$$

$$\begin{aligned} \mathbf{A}_{22} = & I_2 + I_3 + l_{c2}^2 m_2 + (l_2^2 + l_{c3}^2 + 2l_2 l_{c3} C_3) m_3 \\ \mathbf{A}_{23} = & I_3 + (l_{c3}^2 + l_2 l_{c3} C_3) m_3 \end{aligned}$$

$$\mathbf{A}_{31} = I_3 + (l_{c3}^2 + l_2 l_{c3} C_3 + l_1 l_{c3} C_{23}) m_3$$

$$\mathbf{A}_{32} = I_3 + (l_{c3}^2 + l_2 l_{c3} C_3) m_3$$

$$\mathbf{A}_{33} = I_3 + l_{c3}^2 m_3$$

Sabiendo que, I_i es el momento de inercia, l_{ci} la distancia hacia los centroides, C_i y S_i las funciones trigonométricas coseno y seno de los desplazamientos angulares. Considerando que i es el número de articulación.

B. Efecto Coriolis y Fuerzas Centrífugas

La matriz $\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}})$ mostrada en (30), relaciona el Efecto Coriolis y la fuerza centrífuga con la matriz de Energía Cinética. Esta relación se puede observar en (31) [33].

$$\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) = - \begin{bmatrix} \mathbf{b}_{11} \\ \mathbf{b}_{21} \\ \mathbf{b}_{31} \end{bmatrix} \quad (30)$$

$$\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) = \dot{\mathbf{A}}(\mathbf{q})\dot{\mathbf{q}} - \frac{1}{2} \begin{bmatrix} \dot{\mathbf{q}}^T \mathbf{A}_{q1} \dot{\mathbf{q}} \\ \dot{\mathbf{q}}^T \mathbf{A}_{q2} \dot{\mathbf{q}} \\ \vdots \\ \dot{\mathbf{q}}^T \mathbf{A}_{qn} \dot{\mathbf{q}} \end{bmatrix} \quad (31)$$

Los valores correspondientes a la matriz $\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}})$ son los siguientes:

$$\begin{aligned} \mathbf{b}_{11} = & -m_2 l_1 l_{c2} S_2 \dot{\theta}_2^2 - 2m_2 l_1 l_{c2} S_2 \dot{\theta}_1 \dot{\theta}_2 \\ & + m_3 \dot{\theta}_2^2 (-l_1 l_{c3} S_{23} - l_1 l_2 S_2) \\ & + m_3 \dot{\theta}_3^2 (-l_1 l_{c3} S_{23} - l_2 l_{c3} S_3) \\ & + m_3 [\dot{\theta}_1 \dot{\theta}_2 (-2l_1 l_2 S_2 - 2l_1 l_{c3} S_{23}) \\ & + \dot{\theta}_1 \dot{\theta}_3 (-2l_2 l_{c3} S_3 - 2l_1 l_{c3} S_{23}) \\ & + \dot{\theta}_2 \dot{\theta}_3 (-2l_2 l_{c3} S_3 - 2l_1 l_{c3} S_{23})] \end{aligned}$$

$$\begin{aligned} \mathbf{b}_{21} = & m_2 l_1 l_{c2} S_2 \dot{\theta}_1^2 + m_3 \dot{\theta}_1^2 (l_1 l_2 S_2 + l_1 l_{c3} S_{23}) \\ & - m_3 l_2 l_{c3} S_3 \dot{\theta}_3^2 \\ & + m_3 (-2l_2 l_{c3} S_3 \dot{\theta}_1 \dot{\theta}_3 - 2l_2 l_{c3} S_3 \dot{\theta}_2 \dot{\theta}_3) \end{aligned}$$

$$\begin{aligned} \mathbf{b}_{31} = & m_3 \dot{\theta}_1^2 (l_2 l_{c3} S_3 + l_1 l_{c3} S_{23}) + 2m_3 l_2 l_{c3} S_3 \dot{\theta}_1 \dot{\theta}_2 \\ & + m_3 l_2 l_{c3} S_3 \dot{\theta}_2^2 \end{aligned}$$

En donde:

$$S_{23} = \text{sen}(\theta_2 + \theta_3) \quad C_{23} = \text{cos}(\theta_2 + \theta_3)$$

C. Gravedad

Es el gradiente de la energía potencial, que incrementa según la elevación de su centro de masa. Los valores gravitatorios de los elementos en el sistema pueden ser observados en (32) [33].

$$\mathbf{g}(\mathbf{q}) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (32)$$

Se hace esta consideración debido a que no existe ningún movimiento del robot en el eje Z.

VI. MODELO DINÁMICO ACTUADORES

Para modelar un motor de corriente continua (CC), se parte de un sistema conectado a una carga mediante engranajes. Existen ciertas variables a tomar en cuenta, tales como: el voltaje de armadura (V_{ar}), la corriente de armadura (i_{ar}), resistencia de armadura (R_{ar}), inductancia de la armadura (L_{ar}), el par del motor sin carga (t_m), la fuerza electromotriz inversa del motor (V_b), momento de inercia del motor (J_m), la fricción viscosa (B_m), entre otras [34]. Esto se puede observar en la Fig. 11.

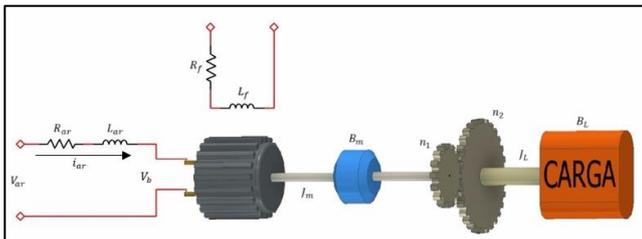


Fig. 11 Estructura electromecánica de un motor con carga acoplada por engranajes.

Analizando eléctricamente al sistema se tiene que el par total del motor (t_o) corresponde a la ecuación mostrada en (33).

$$t_o = i_{ar} k_a \quad (33)$$

Sabiendo que en (33) la constante del par del motor (k_a), regida por (34), es el producto de la constante de proporcionalidad del motor (k_1) y el campo magnético (k_2).

$$k_a = k_1 k_2 \quad (34)$$

De la Fig. 11, se obtiene que el voltaje de armadura (V_{ar}), mostrado en (35), depende de la fuerza contraelectromotriz (V_b) que se define por (36).

$$V_{ar} = L_{ar} \frac{\partial i_{ar}}{\partial t} + R_{ar} i_{ar} + V_b \quad (35)$$

$$V_b = k_b \dot{\theta}_m \quad (36)$$

Donde:

- k_b es la constante de fuerza contra electromotriz.
- $\dot{\theta}_m$ es la velocidad angular.

En la parte mecánica, el par total del motor se define por (37) mientras que, el par sin carga es equivalente a (38). Este último incluye la aceleración angular ($\ddot{\theta}_m$).

$$t_o = t_m + n t_L \quad (37)$$

$$t_m = J_m \ddot{\theta}_m + B_m \dot{\theta}_m \quad (38)$$

Sabiendo que el modelo dinámico de un servomotor, parte de un motor CC conectado a otros sistemas de conversión, se puede redibujar el sistema en el diagrama de la Fig. 12 [35].

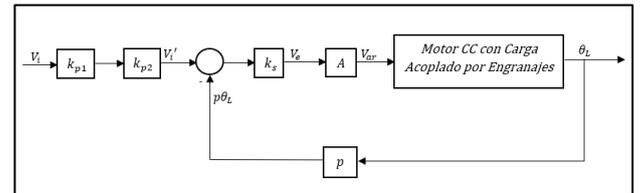


Fig. 12 Diagrama de bloques que representa la estructura de un servomotor.

En donde:

- V_i : voltaje de entrada.
- K_{p1} : constante de proporcionalidad del convertidor de tensión a PWM.
- K_{p2} : constante de proporcionalidad del convertidor de PWM a tensión.
- K_s : constante de proporcionalidad del comparador de posición.
- A : constante de amplificación del puente H que controla el giro del motor.
- p : constante asociada al potenciómetro que sirve de transductor de la posición.

De la Fig. 12 se puede obtener (39), en base a las constantes de proporcionalidad mencionadas anteriormente.

$$V_{ar} = K_s A (V_i K_{p1} K_{p2} - p \theta_L) \quad (39)$$

Sustituyendo la expresión (37) en (35), se obtiene (40).

$$V_{ar} = \frac{L_{ar}}{k_a} (J_m \ddot{\theta}_m + B_m \dot{\theta}_m + n \dot{t}_L) + \frac{R_{ar}}{k_a} (J_m \ddot{\theta}_m + B_m \dot{\theta}_m + n \dot{t}_L) + k_b \dot{\theta}_m \quad (40)$$

Reemplazando (41) y (39) en (40) resulta la ecuación (42), misma que describe el par con carga del servomotor (t_L) al despreciar la inductancia de la armadura (L_{ar}) [35].

$$\theta_m = \frac{1}{n} \theta_L \quad (41)$$

$$t_L = \frac{1}{n} \left[\frac{k_a}{R_{ar}} V_i K_s A K_{p1} K_{p2} - J_m \frac{1}{n} \ddot{\theta}_L - \left(\frac{k_a k_b}{R_{ar}} + B_m \right) \frac{1}{n} \dot{\theta}_L - \frac{k_a}{R_{ar}} K_s A p \theta_L \right] \quad (42)$$

En donde:

- θ_m es el desplazamiento angular sin carga.
- θ_L es el desplazamiento angular con carga.
- $\dot{\theta}_L$ es la velocidad angular con carga.
- $\ddot{\theta}_L$ es la aceleración angular con carga.

VII. SISTEMA DE CONTROL

Un sistema de tipo MIMO (*Multiple Input Multiple Output*) es propuesto para el control del manipulador. Observando la Fig. 13, el bloque Controlador tiene como variables de entrada, los desplazamientos angulares deseados ($\theta_1 d, \theta_2 d, \theta_3 d$), obtenidos del bloque Generador de Trayectorias. Por otro lado, en la salida se tienen los voltajes de ingreso (V_{i1}, V_{i2}, V_{i3}) que ingresan a los Actuadores. Finalmente, se considera una realimentación de la posición angular generada ($\theta_1, \theta_2, \theta_3$) proveniente de Robot 3R que representa el modelo dinámico del robot.

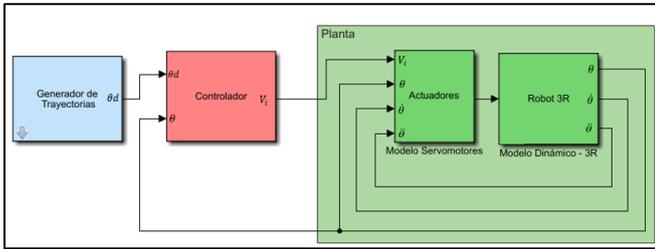


Fig. 13 Diagrama de control para el seguimiento de trayectorias.

A. Controlador PID

Con el fin de tener una referencia respecto a los controladores con inteligencia artificial, se utiliza un controlador clásico PID, desarrollado con *Simulink* mediante la herramienta *Tune* del complemento *Simulink Control Design*. Este permite no depender de ningún cálculo de la función de transferencia de la planta. En la Fig. 14 se muestra el diagrama de este controlador.

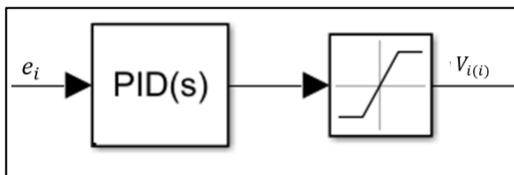


Fig. 14 Diagrama de bloques del controlador PID en *Simulink*.

El error (e_i) que se observa en la Fig. 14, resulta de la operación mostrada en (43).

$$e_i = \theta_i d - \theta_i \quad (43)$$

En donde:

- i representa el número de articulación del robot.

A. Controlador Difuso

Para este controlador, se propone uno que simule un PD convencional, agregando la constante I. En este las variables lingüísticas se desglosan en: cambio de error (CE), error (E) y el voltaje de ingreso (V_i). En cuanto al universo discurso, este viene representado por siete estados:

- PB viene de Positivo Alto.
- PM significa Positivo Medio.
- PS es Positivo Bajo.
- ZE significa cero.
- NS proviene de Negativo Pequeño.
- NM significa Negativo Medio.
- NB viene de Negativo Alto.

En este estudio, la base de reglas utilizadas para el controlador PD+I difuso, son adjuntadas en la Tabla 2 que se muestra a continuación.

TABLA 2 REGLAS DEL CONTROLADOR PD+I DIFUSO.

E \ CE	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZE
NM	NB	NB	NM	NM	NS	ZE	PS
NZ	NB	NM	NS	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PS	PM	PB
PM	NS	ZE	PS	PM	PM	PB	PB
PB	ZE	PS	PM	PB	PB	PB	PB

Las funciones de entrada pueden ser observadas en la Fig. 15 y 16, mientras que la salida se muestra en la Fig. 17.

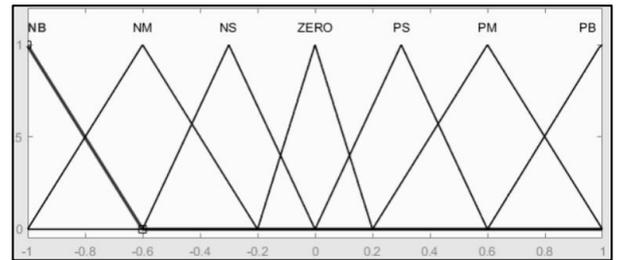


Fig. 15 Funciones de pertenencia para la variable cambio de error (CE).

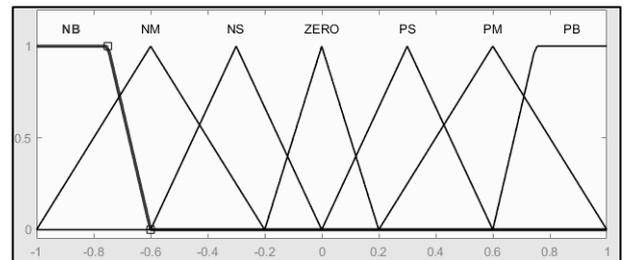


Fig. 16 Funciones de pertenencia para la variable error (E).

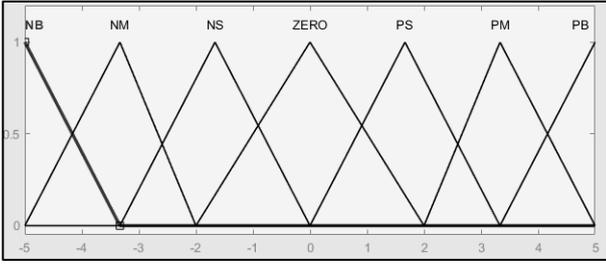


Fig. 17 Funciones para la variable voltaje de ingreso (V_i).

El Defuzzificador aplicado en este estudio, utiliza el método por centro de área (COA), que establece las relaciones entre las entradas (E y CE) y la salida (V_o). Mediante este, se obtiene la Fig. 18 en donde se observa la asignación de cada una de sus entradas.

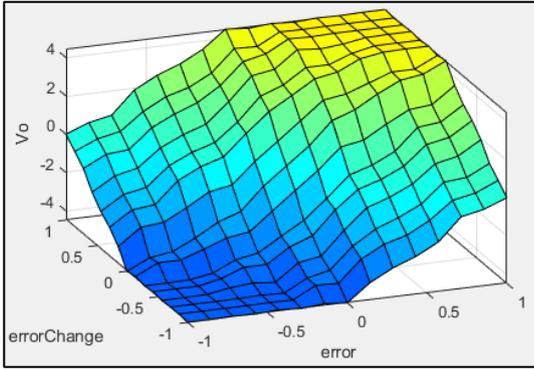


Fig. 18 Superficie de clasificación usando las reglas planteadas.

Para el controlador PD+I difuso, se debe partir de (44), que se observa a continuación [24]. El código desarrollado en *Simulink* se muestra en la Fig. 19

$$u'(t) = K_p e(t) + K_D \Delta e(t) + K_i \int e(t) \quad (44)$$

En donde:

- K_p es la constante proporcional del controlador.
- K_D es el factor de ganancia diferencial el controlador.
- K_i es el factor de ganancia integral del controlador.
- $e(t)$ y $\Delta e(t)$ es el error y diferencia de error obtenidos respectivamente.

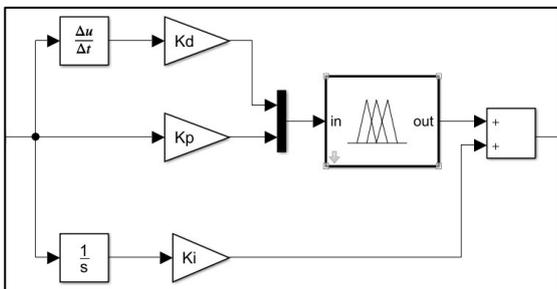


Fig. 19 Diagrama de bloques del controlador PD+I difuso en *Simulink*.

B. Controlador Feed-Forward Backpropagation

Este controlador se construyó basado en una red de cuatro capas: dos ocultas, una de entrada y otra de salida; usando un algoritmo de aprendizaje Levenberg-Marquadt. Se consideró el uso de funciones de activación de tipo sigmoide y lineal para las capas ocultas y, de salida. Cada capa oculta está

dimensionada con 5 neuronas para el procesamiento y aprendizaje de 3334 datos, provenientes de los resultados obtenidos del controlador difuso PD+I. La matriz de entrada corresponde a las posiciones angulares deseadas (θ_d) y reales (θ) de cada articulación, mientras que la salida es el voltaje de ingreso (V_i). A continuación, se presenta la estructura de la red neuronal, en la Fig. 20 obtenida al usar el complemento de *MATLAB Neural Network Toolbox*.

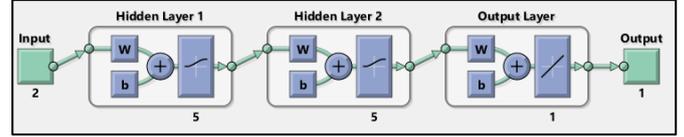


Fig. 20 Estructura de la red *feed-forward backpropagation* utilizada.

Al ser tres articulaciones, se tendrán tres controladores distintos que, exportados a *Simulink*, serán codificados (véase la Fig. 21) en base a la estructura de la red adjuntada anteriormente en la Fig. 20.

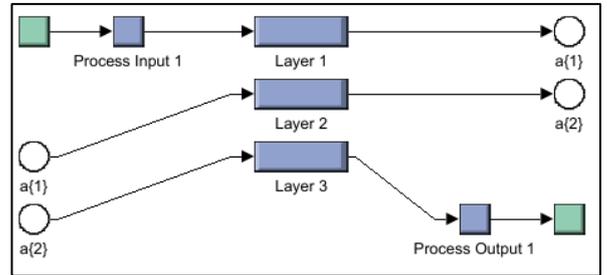


Fig. 21 Diagrama de bloques de la red *feed-forward backpropagation* en *Simulink*.

C. Controlador Cascade-Forward Backpropagation

El controlador desarrollado con esta red utiliza los mismos parámetros y valores de entrenamiento que la red *feed-forward backpropagation*. En la Fig. 22 y 23, se muestra la estructura del controlador diseñado con *Neural Network Toolbox* y su respectivo código en *Simulink*.

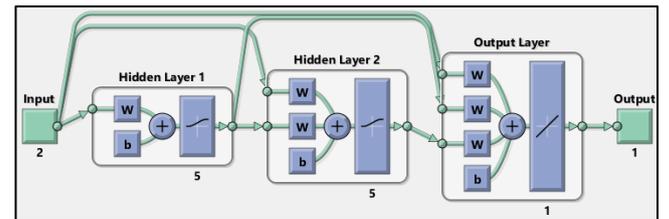


Fig. 22 Estructura de la red *cascade-forward backpropagation* utilizada.

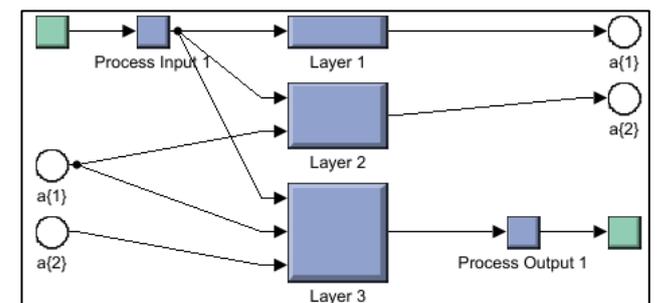


Fig. 23 Diagrama de bloques de la red *cascade-forward backpropagation* en *Simulink*.

VIII. RESULTADOS

Con el fin de obtener un comportamiento muy aproximado al del mundo real, los controladores son aplicados en el modelo dinámico de los actuadores y del manipulador. Las constantes dinámicas de los actuadores se obtienen de [35] y se muestran en la Tabla 3. Los parámetros dinámicos del robot se obtuvieron mediante un software de modelamiento mecánico. Teniendo como resultado los valores que se adjuntan en la Tabla 4.

TABLA 3 PARÁMETROS DINÁMICOS DE LOS ACTUADORES.

Parámetros Servomotores		
	Servomotor 1-2-3	Unidad
k_a	0,35	$(N * m/A)$
k_b	0,04	$(V * s/rad)$
R_{ar}	1,6	(Ω)
J_m	0,007	$(kg * m^2)$
B_m	0,01313	$(N * m * s/rad)$
ren	561,6	
A	15	
k_s	10	
k_p	1	
p	1	

TABLA 4 TABLA PARÁMETROS DINÁMICOS DEL ROBOT.

Parámetros Robot		
	Eslabón 1-2-3	Unidad
l	0,19	(m)
l_c	0,124	(m)
m	0,334	(kg)
I_{zz}	0,001425267	$(kg * m^2)$

A. Espacio de Trabajo

Para generar las trayectorias de prueba, se restringieron las soluciones de la cinemática inversa a un solo signo, para obtener una única posición del robot (brazo hacia arriba). Con esto, se obtiene el espacio de trabajo del robot en el plano (x,y) apreciable en la Fig. 24, siendo 57 “cm” el valor límite en el eje x e y . Simulando mediante el complemento de *MATLAB Robotics Toolbox – Peter Corke* se obtiene el gráfico mostrado en la Fig. 25.



Fig. 24 Espacio de trabajo del robot 3R.

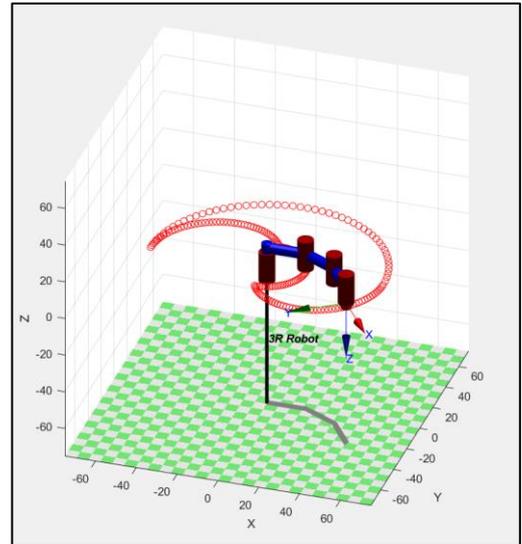


Fig. 25 Espacio de trabajo del robot graficado mediante la interfaz desarrollada por Peter Corke.

B. Simulaciones

En este apartado se plantean dos trayectorias, mismas que se diferencian por la cantidad de vértices, con el fin de observar el comportamiento de los controladores frente a cambios bruscos de dirección. Cada una de estas se aprecian en la Fig. 26 y Fig. 27.

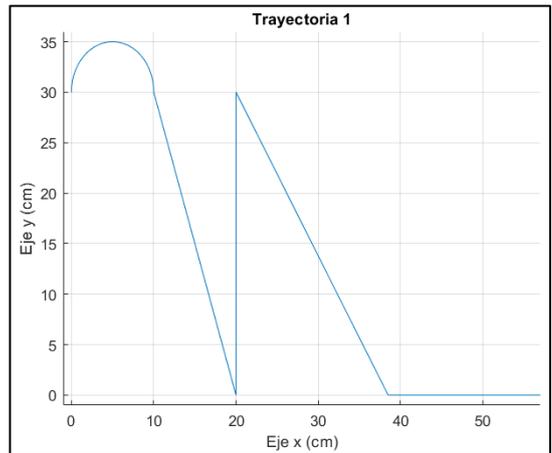


Fig. 26 Primera trayectoria planteada.

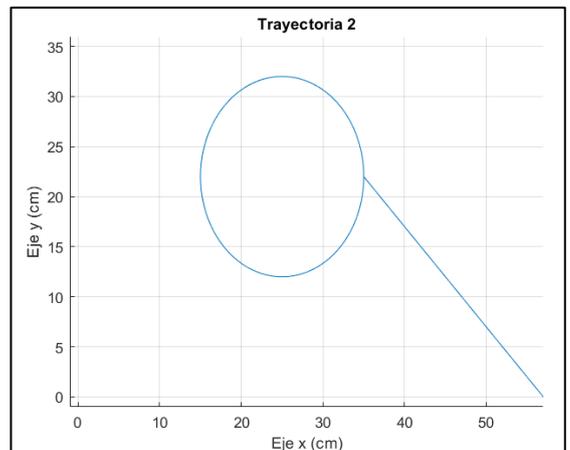


Fig. 27: Segunda trayectoria planteada.

Cada una de estas trayectorias han sido probadas con los controladores desarrollados, obteniéndose dos resultados. El primero corresponde al gráfico de la posición cartesiana deseada y generada por cada uno de los controladores, misma que se muestra en la Fig. 28 y 29.

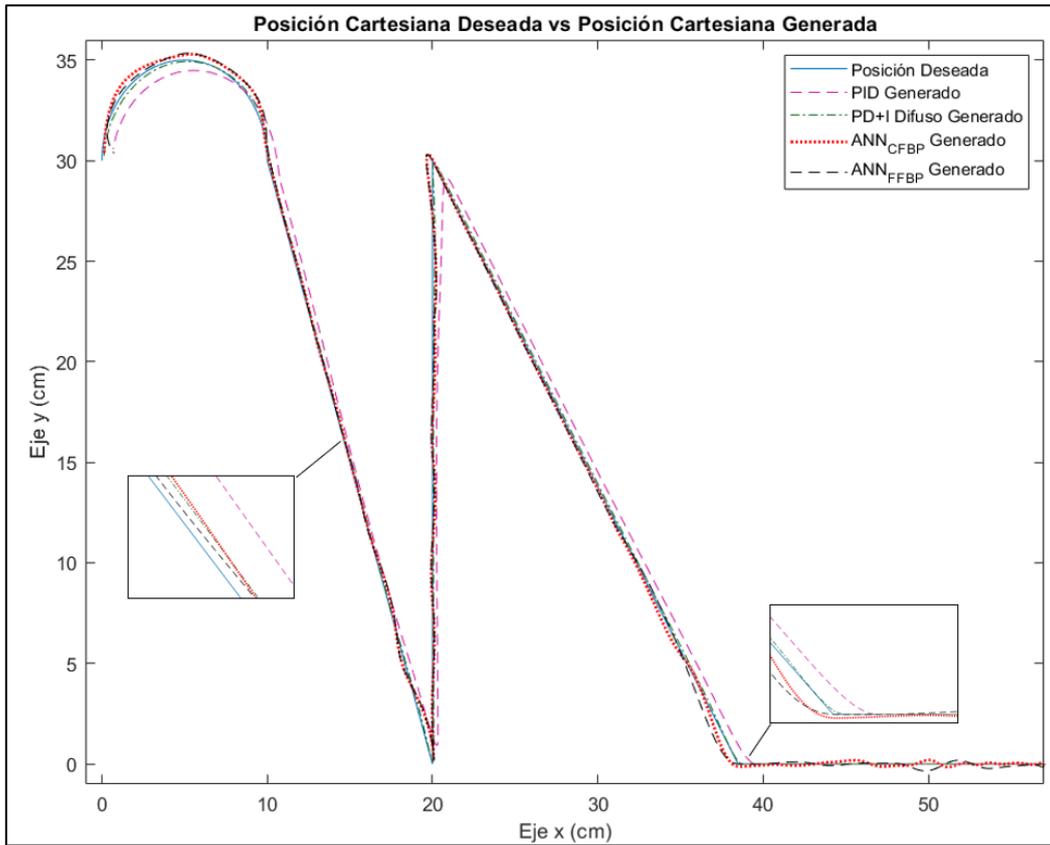


Fig. 28 Respuesta cartesiana de los controladores al seguir la trayectoria 1.

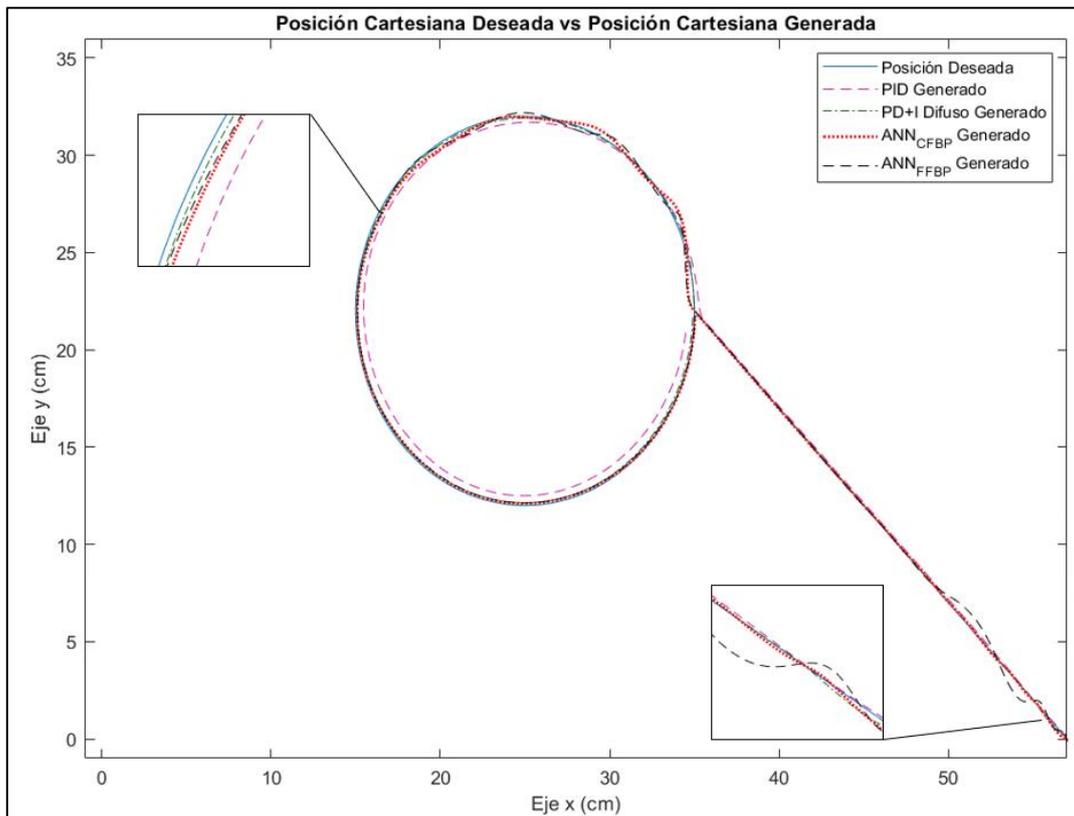


Fig. 29 Respuesta cartesiana de los controladores al seguir la trayectoria 2.

La Fig. 28 y 29 evidencian que el controlador PID presenta el mayor error entre todos los analizados. De igual manera, se observa que los controladores mediante redes neuronales se demoran en estabilizar al existir cambios de dirección en ambas trayectorias.

El segundo resultado, adjuntado en las Fig. 30-37, se refiere a las posiciones angulares con respecto al tiempo de cada una de las articulaciones.

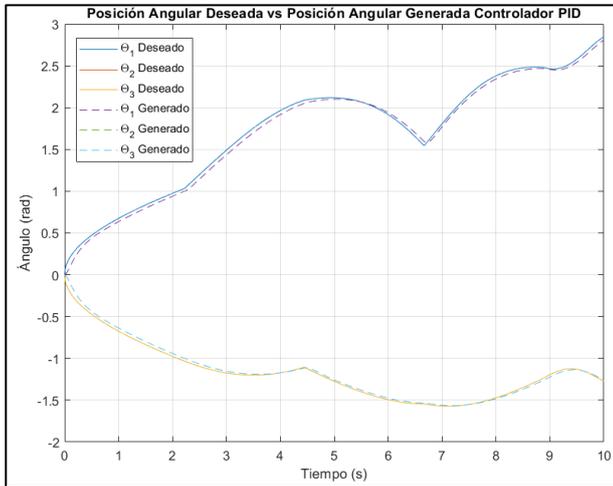


Fig. 30 Gráfico de comparación de posiciones angulares deseadas y generadas por el controlador PID, para la trayectoria 1.

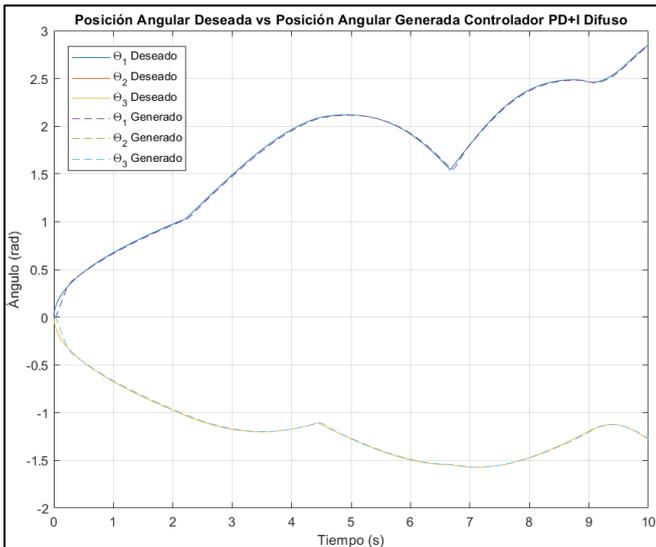


Fig. 31 Gráfico de comparación de posiciones angulares deseadas y generadas por el controlador PD+I Difuso, para la trayectoria 1.

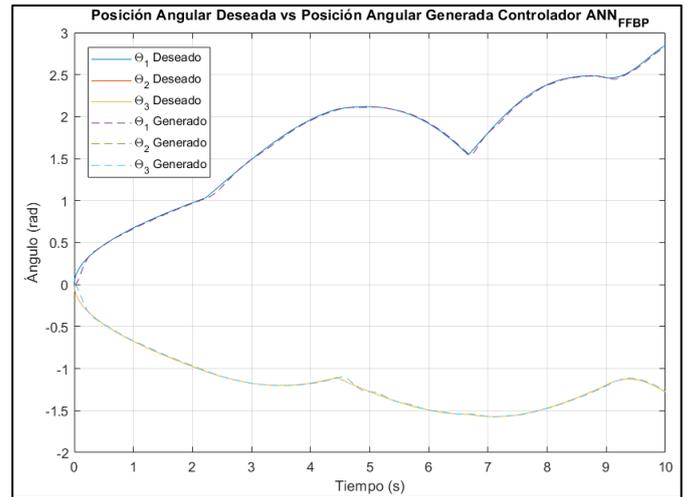


Fig. 32 Gráfico de comparación de posiciones angulares deseadas y generadas por el controlador de redes neuronales FFBP, para la trayectoria 1.

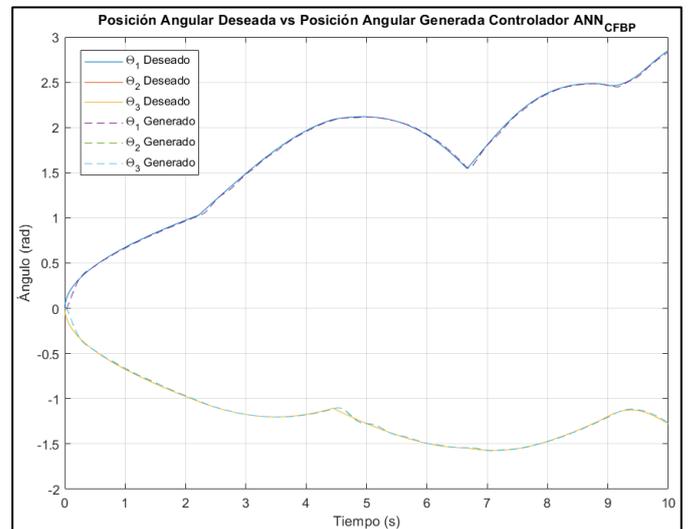


Fig. 33 Gráfico de comparación de posiciones angulares deseadas y generadas por el controlador de redes neuronales CFBP, para la trayectoria 1.

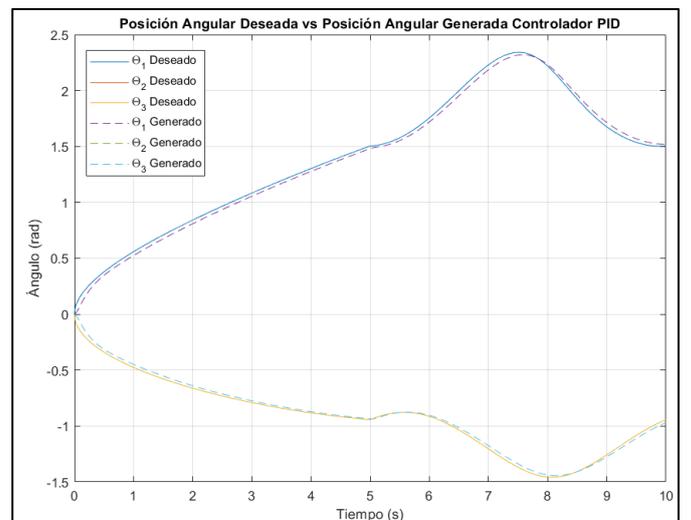


Fig. 34 Gráfico de comparación de posiciones angulares deseadas y generadas por el controlador PID, para la trayectoria 2.

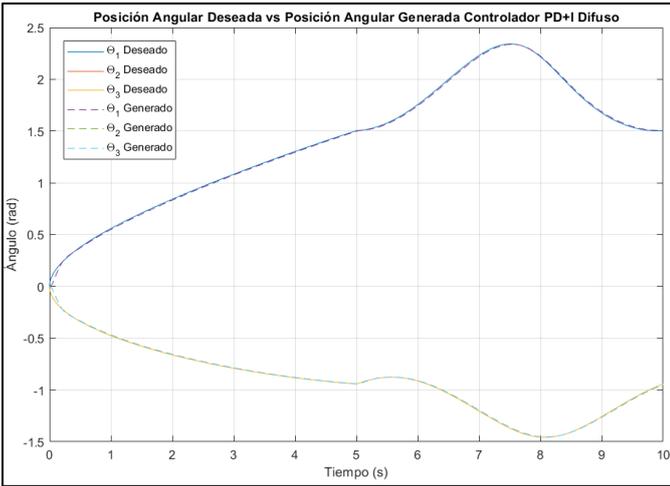


Fig. 35 Gráfico de comparación de posiciones angulares deseadas y generadas por el controlador PD+I Difuso, para la trayectoria 2.

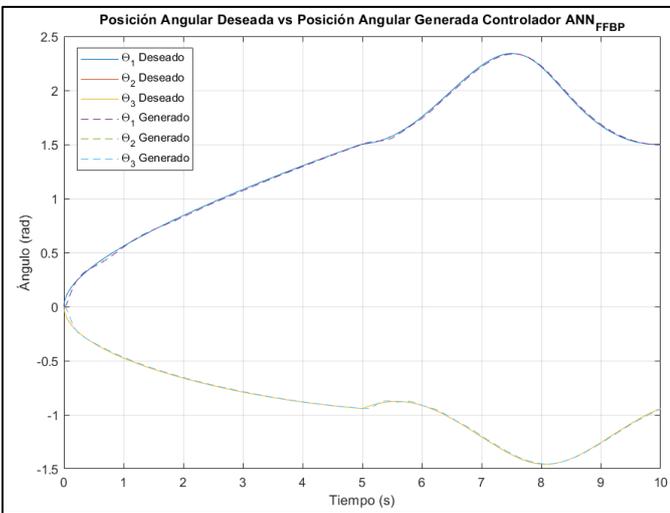


Fig. 36 Gráfico de comparación de posiciones angulares deseadas y generadas por el controlador de redes neuronales FFBP, para la trayectoria 2.

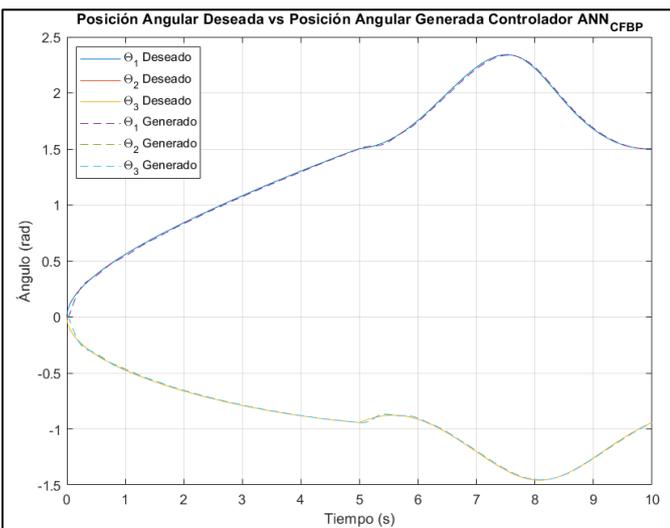


Fig. 37 Gráfico de comparación de posiciones angulares deseadas y generadas por el controlador de redes neuronales CFBP, para la trayectoria 2.

Mediante un análisis de las Fig. 30-37, se llega a determinar visualmente que el controlador PID presenta mayor error en la posición angular. Al ser los errores cartesianos y angulares muy pequeños se propone una cuantificación de estos a través de la utilización del error medio cuadrático relativo (RMSE). Esta será desarrollada posteriormente en la misma sección.

Al simular el robot al seguir ambas trayectorias, se obtienen las siguientes imágenes mostradas en la Fig. 38 y Fig. 39.

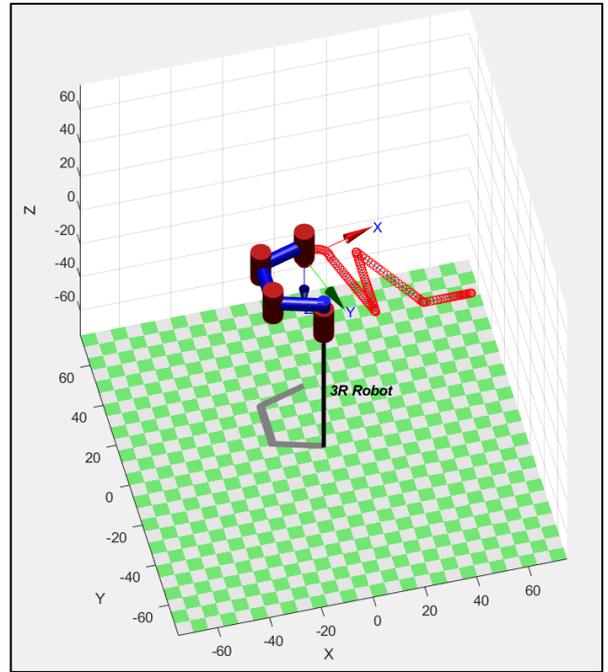


Fig. 38 Simulación del robot al seguir la trayectoria 1.

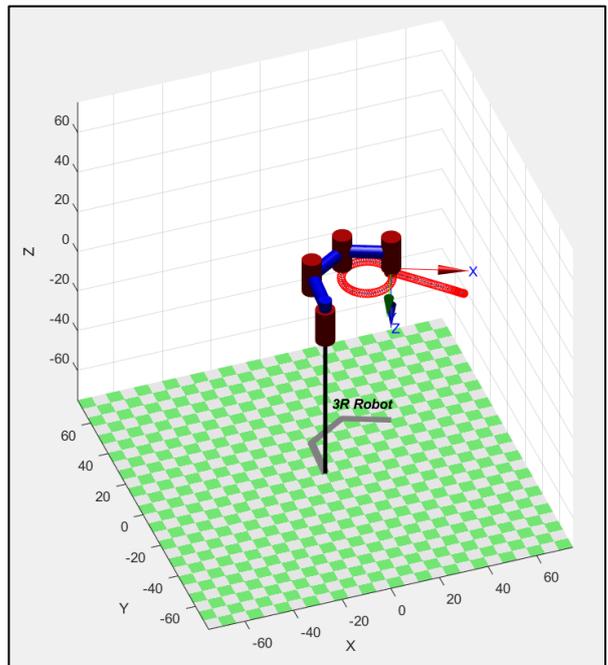


Fig. 39 Simulación del robot al seguir la trayectoria 2.

Mediante la obtención de las gráficas mostradas en la Fig. 25, Fig. 38 y Fig. 39, se corrobora la utilidad y funcionalidad del complemento *Robotics Toolbox for MATLAB* desarrollado por Peter Corke al momento de simular el movimiento del robot.

Finalmente, se obtienen los índices de error medio cuadrático relativo (RMSE), a partir de (45). Los resultados se adjuntan en la Fig. 40 y 41 para la primera trayectoria, mientras que para la segunda en la Fig. 42 y Fig. 43.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2} \quad (45)$$

En donde (n) representa el número de datos y (e) es el error, ya sea cartesiano o angular.

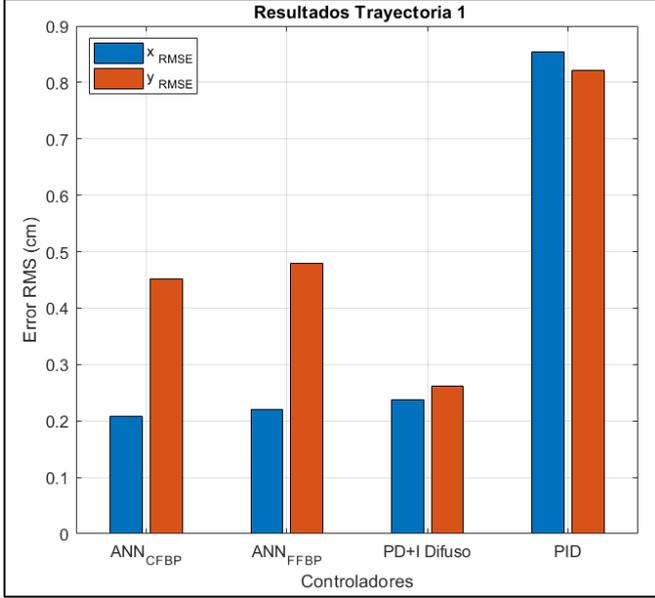


Fig. 40 Error RMS cartesiano para cada uno de los controladores analizados en la trayectoria 1.

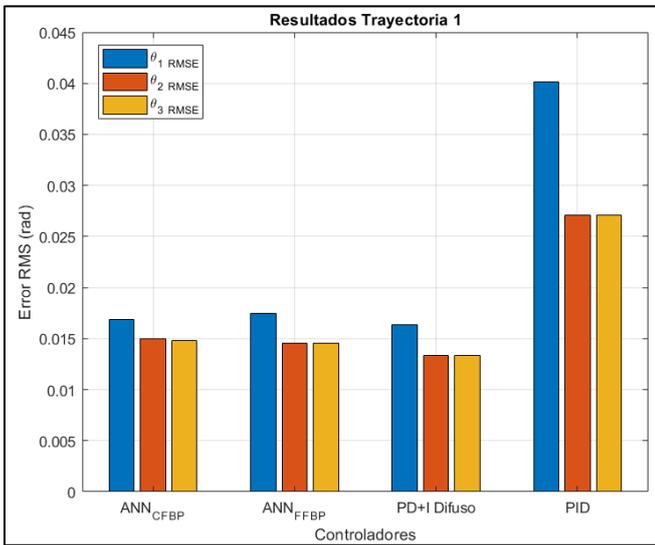


Fig. 41 Error RMS angular para cada uno de los controladores analizados en la trayectoria 1.

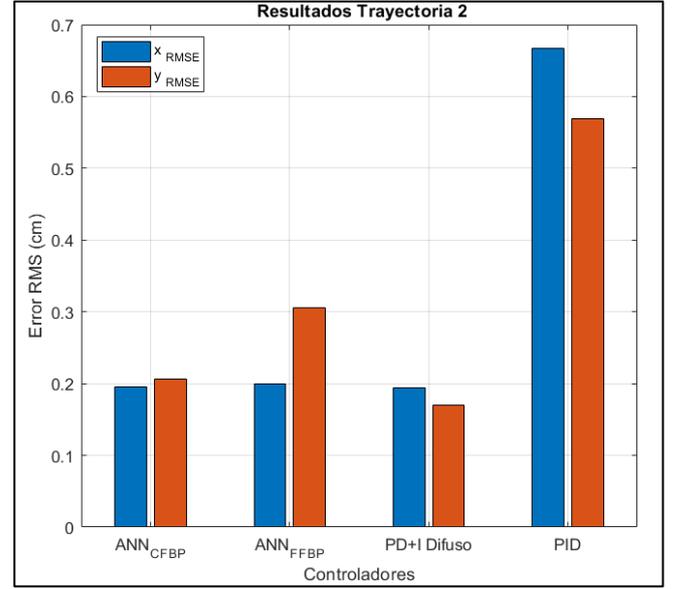


Fig. 42 Error RMS cartesiano para cada uno de los controladores analizados para la trayectoria 2.

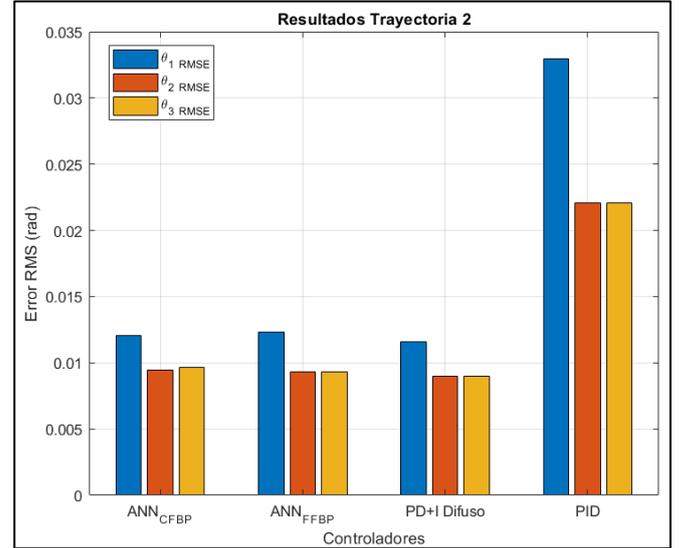


Fig. 43 Error RMS angular para cada uno de los controladores analizados para la trayectoria 2.

IX. CONCLUSIONES

Los resultados obtenidos reflejan que, los controladores basados en inteligencia artificial permiten realizar el seguimiento de distintas trayectorias en el robot propuesto, siendo estos más eficaces comparados con un PID. Sin embargo, se observa que los de redes neuronales son más lentos frente a cambios bruscos de dirección causados por la inercia que existe en cada articulación. Por lo que se genera un mayor error RMS cartesiano y angular (Fig. 40-43) en la trayectoria 1 (Fig. 27).

Por otro lado, al ser utilizado el espacio de trabajo de la Fig. 24, se consiguió que no surgieran problemas de posicionamiento del robot al trazar trayectorias en donde se presenten singularidades en las soluciones de la cinemática inversa. No obstante, a partir de esta restricción se generó una disminución del rango angular al que pueden moverse las articulaciones 2 y 3, reduciéndose a tan solo un intervalo de $[-\frac{\pi}{2}, 0]$ "rad", mientras que, la articulación 1 mantiene el intervalo $[-\frac{\pi}{2}, \frac{\pi}{2}]$ "rad".

Por último, se tiene que el controlador PD+I difuso resulta ser superior respecto a los demás, al analizar cada uno de los índices de error RMS en ambas trayectorias. A pesar de ello, los basados en redes neuronales representan una buena opción siempre que se apliquen en trayectorias sin muchos cambios bruscos.

AGRADECIMIENTOS

Esta investigación pudo ser posible gracias al apoyo del Vicerrectorado de Investigaciones de la Universidad del Azuay a través del Proyecto INV-PELE-E06.

REFERENCIAS

- [1] INEC, «Directorio de Empresas,» 2017. [En línea]. Available: http://produccion.ecuadorencifras.gob.ec/geoolik/proxy/QvAJAXZfc/opensdoc.htm?document=empresas_test.qvw&host=QVS%40virtualqv&anonymous=true.
- [2] International Federation of Robotics, «Survey: 1.3 million industrial robots to enter service by 2018,» 25 Febrero 2016. [En línea]. Available: <https://ifr.org/news/-survey-13-million-industrial-robots-to-enter-service-by-2018-/>.
- [3] A. Barrientos, L. F. Peñin, C. Balaguer y R. Aracil, Fundamentos de Robótica, Madrid: McGraw Hill, 2007.
- [4] J. C. Chahuara Quispe, «Sistemas neuro-difusos - UNMSM,» 2011. [En línea]. Available: http://sisbib.unmsm.edu.pe/bibvirtualdata/Tesis/Ingenie/chahuara_qj/Cap2.pdf.
- [5] S. Puga Guzmán, J. Moreno Valenzuela y V. Santibáñez, «Adaptive Neural Network Motion Control of Manipulators with Experimental Evaluations,» *The Scientific World Journal*, pp. 1-13, 2014.
- [6] J. Kern, M. Jamett, C. Urrea y H. Torres, «Development of a neural controller applied in a 5 DOF robot redundant,» *IEEE LATIN AMERICA TRANSACTIONS*, pp. 98-106, 2014.
- [7] Z.-H. Jiang, «Trajectory Control of Robot Manipulators Using a Neural Network Controller,» *Robot Manipulators Trends and Development*, pp. 361-376, 1 Marzo 2010.
- [8] A.-V. Duka, «Neural network based inverse kinematics solution for trajectory tracking of a robotic arm,» *Procedia Technology*, pp. 21-27, 2014.
- [9] Z. S. Abo-Hammour, O. Alsmadi, S. Bataineh, M. A. Al-Omari y N. Affach, «Continuous Genetic Algorithms for Collision-Free Cartesian Path Planning of Robot Manipulators,» *International Journal of Advanced Robotic Systems*, pp. 14-36, 2011.
- [10] S. Khemaissia, Multi-Objective Genetic Algorithm-Based Time Optimal Trajectory Planning and Robot Control, Riyadh: Riyadh College of Technology, 2011.
- [11] L. Sheng y W. Li, «Optimization Design by Genetic Algorithm Controller for Trajectory Control of a 3-RRR Parallel Robot,» *Algorithms*, pp. 1-13, 2018.
- [12] T. Kathuria, A. Gupta, J. Kumar, V. Kumar y K. Rana, «Study of Optimization Methods for Tuning of Gains for Three Link Manipulator,» *IEEE*, pp. 99-104, 2017.
- [13] V. K. Banga, R. Kumar y Y. Singh, «Fuzzy-genetic optimal control for robotic systems,» *International Journal of the Physical Sciences*, vol. 6, p. 9, 2011.
- [14] F. M. Botsali, M. Tinkir, M. Kalyoncu and Ü. Önen, "Fuzzy Logic Trajectory Control of Flexible Robot Manipulator With Rotating Prismatic Joint," *IEEE*, vol. III, pp. 35-39, 2010.
- [15] A. Nawrocka, A. Kot y M. Nawrocki, «Type - 2 Fuzzy Logic Controller for Nonlinear Object Control,» *IEEE 16th International Carpathian Control Conference*, pp. 334-337, 2015.
- [16] S. Al-khayyt, «Comparison between Fuzzy Logic Based Controllers for Robot Manipulator Trajectory Tracking,» *FNCS'12*, 2012.
- [17] H. Chaudhary, V. Panwar, N. Sukavanum y R. Prasad, «Fuzzy PD+I based Hybrid force/ position control of an Industrial Robot Manipulator,» *Third International Conference on Advances in Control and Optimization of Dynamical Systems*, pp. 429-436, 2014.
- [18] M. Tinkir, M. Kalyoncu, Ü. Önen and F. M. Botsali, "Competition of Fuzzy Logic Controllers Applied on Flexible Manipulator," *IEEE*, pp. 99-103, 2011.
- [19] J. Zhao, L. Han, L. Wang y Z. Yu, «The Fuzzy PID Control Optimized by Genetic Algorithm for Trajectory Tracking of Robot Arm,» *IEEE*, pp. 556-559, 2016.
- [20] A. Pajaziti y H. Cana, «Robotic Arm Control with Neural Networks Using Genetic Algorithm Optimization Approach,» *World Academy of Science, Engineering and Technology International Journal of Mechanical and Mechatronics Engineering*, vol. 8, n° 8, pp. 1431-1435, 2014.
- [21] L. Tudor y A. Moise, «Automatic Expert System for Fuzzy Control of Robot Trajectory in Joint Space,» *IEEE*, pp. 1057-1062, 2013.
- [22] S. Russell y P. Norvig, Inteligencia Artificial: Un Enfoque Moderno, Madrid: Pearson Prentice Hall, 2004.
- [23] B. Martín del Brío y A. Sanz Molina, Redes Neuronales y Sistemas Borrosos, Zaragoza, 2007.
- [24] L. Reznik, Fuzzy Controllers, Melbourne: BH Newnes, 1997.
- [25] B. Hamed y M. Almobaied, «Fuzzy PID Controllers Using FPGA Technique for Real Time DC Motor Speed Control,» *Intelligent Control and Automation*, vol. 2, pp. 233-240, 2011.
- [26] F. Casacuberta Nolla y E. Vidal Ruíz, «Redes Neuronales Multicapa,» Universidad Politécnica de Valencia, Valencia, 2017.
- [27] H. Franco, «Redes Neuronales Multicapa,» Universidad Central, Bogotá, 2010.
- [28] H. Demuth y M. Beale, Neural Network Toolbox User's Guide, Natick: The MathWorks, Inc., 2009.
- [29] B. Warsito, R. Santoso, Suparti y H. Yasin, «Cascade Forward Neural Network for Time Series Prediction,» *Journal of Physics*, pp. 1-8, 2018.
- [30] S. Goyal y G. Kumar Goyal, «Cascade and Feedforward Backpropagation Artificial Neural Network Models For Prediction of Sensory Quality of Instant Coffee Flavoured Sterilized Drink,» *Canadian Journal on Artificial Intelligence, Machine Learning and Pattern Recognition*, vol. 2, n° 6, pp. 78-82, 2011.
- [31] M. W. Spong, S. Hutchinson y M. Vidyasagar, Robot modeling and control, New York: Wiley, 2006.
- [32] J. A. Kern Molina, Análisis e Implementación de Sistemas de Control Tolerantes a Fallas Aplicados en un Robot de Tipo

Manipulador Industrial Real, Santiago de Chile: Universidad de Santiago de Chile, 2010.

- [33] O. Khatib, *Advanced Robotic Manipulation*, Standford, 2005.
- [34] L. Zaccarian, *DC motors: dynamic model and control techniques*, Roma: Lecture Notes, 2012.
- [35] C. Urrea y J. Kern, «A New Model for Analog Servo Motors. Simulations and Experimental Results,» *Canadian Journal on Automation, Control and Intelligent Systems.*, vol. 2, n° 2, pp. 29-38, 2011.