ISSN: 1692-7257 - Volumen 2 – Número 46 - 2025

RCTA
Revista Colombiana de Tecnologías de Avanzada
UNIPAMPLONA

Digital Object Identifier: 10.24054/rcta.v2i46.3583

Cinemática inversa para un robot planar 2R basada en una red neuronal con datos sintéticos a través de un enfoque de ensamble de modelos

Inverse kinematics for a 2R planar robot based on a neural network with synthetic data through a model ensemble approach

Angie Paola Mancipe García[1], Nicolas David Barrera Fonseca[1], MSc. Jorge Eduardo Cote Ballesteros[1], MSc. Jhon Edisson Rodríguez Castellanos[1], MSc. Fabián Barrera Prieto[1]

[1] ***Universidad ECCI,*** *Facultad de Ingeniería, Ingeniería Mecatrónica, Bogotá, Cundinamarca, Colombia.*

Correspondencia: *fbarrerap@ecci.edu.co*

Recibido: *20 febrero 2025.* ***Aceptado:*** *25 junio 2025.* ***Publicado:*** *01 julio 2025.*

Cómo citar: *A. P. Mancipe García, N. D. Barrera Fonseca, J. E. Cote Ballesteros, J. E. Rodríguez Castellanos, y F. Barrera Prieto, «Cinemática inversa para un robot planar 2R basada en una red neuronal con datos sintéticos a través de un enfoque de ensamble de modelos», RCTA, vol. 2, n.° 46, pp. 40–48, jul. 2025.*
Recuperado de https://ojs.unipamplona.edu.co/index.php/rcta/article/view/3583

Esta obra está bajo una licencia internacional
Creative Commons Atribución-NoComercial 4.0.

Resumen: Este artículo presenta el desarrollo de un modelo basado en redes neuronales, entrenado con datos sintéticos, para reemplazar la cinemática inversa geométrica de un robot planar 2R. Este enfoque busca simplificar la implementación de cinemáticas, reduciendo el tiempo de desarrollo y el uso de recursos computacionales. El modelo se creó en Google Colab (Python) utilizando TensorFlow/Keras, facilitando su creación y entrenamiento. Además, el sistema integra procesamiento de imágenes en tiempo real para reconocer y seguir contornos, los cuales el robot traza posteriormente. Se implementó un movimiento lineal y vertical mediante un mecanismo piñón-cremallera, permitiendo el trazo discontinuo entre los contornos de una imagen. Los resultados, promediando tres modelos de red neuronal, muestran alta precisión en la predicción de los ángulos de las dos primeras articulaciones del robot 2R, con un RMSE de 0.2293 y 0.0739 respecto a la cinemática inversa geométrica.

Palabras clave: robot 2R, raspberry pi, visión por computador, redes neuronales, robótica.

Abstract: This article presents the development of a neural network-based model, trained with synthetic data, to replace the geometric inverse kinematics of a 2R planar robot. This approach aims to simplify the implementation of kinematics, reducing development time and computational resource usage. The model was created in Google Colab (Python) using TensorFlow/Keras, which facilitated its creation and training. Furthermore, the system integrates real-time image processing to recognize and follow contours, which the robot subsequently traces. A linear and vertical motion was implemented using a rack-and-pinion mechanism, enabling discontinuous tracing between contours of an image. The results, averaging three neural network models, show high accuracy in predicting the angles of the robot's first two joints, with an RMSE of 0.2293 and 0.0739 compared to the geometric inverse kinematics.

Keywords: 2R robot, raspberry pi, inverse kinematics, neural networks, robotics.

1. INTRODUCCIÓN

En los últimos años, las redes neuronales han potencializado diversos campos de la ingeniería, incluida la robótica, al proporcionar soluciones innovadoras para problemas complejos [1]. Su capacidad para modelar sistemas altamente no lineales y adaptarse a entornos dinámicos las convierte en herramientas importantes en una amplia variedad de casos [2]. Una de estas aplicaciones es la Cinemática Inversa (CI) en robots de configuraciones seriales, donde las redes neuronales ofrecen una alternativa prometedora a los métodos geométricos y matemáticos tradicionales.

Históricamente, la cinemática inversa ha sido abordada mediante modelos matemáticos que requieren ecuaciones complejas y parámetros específicos del robot [3], [4], [5], [6]. Aunque estos métodos son efectivos en situaciones estándar, presentan limitaciones importantes al tratar de adaptarse a configuraciones personalizadas o entornos altamente dinámicos, donde las variaciones en la geometría o las condiciones de operación requieren ajustes constantes. En estos casos, los métodos tradicionales pueden resultar ineficaces o lentos. Investigaciones recientes han demostrado que los modelos de aprendizaje automático, y específicamente las redes neuronales, pueden ser entrenados para manejar de manera eficiente estas variabilidades, proporcionando soluciones rápidas y precisas sin necesidad de recalcular manualmente los parámetros para cada configuración o condición del robot, lo que justifica su uso en aplicaciones robóticas que requieren alta adaptabilidad y autonomía [7], [8].

Con los avances en inteligencia artificial, las redes neuronales se han consolidado como una alternativa eficaz para enfrentar los desafíos en robótica. Diversos estudios han mostrado su impacto en la mejora de la precisión, adaptabilidad y flexibilidad en aplicaciones robóticas. Por ejemplo, el artículo [9] presenta un método de alta precisión en tareas de paletizado con un robot SCARA de 3 GDL utilizando redes neuronales recurrentes. Así mismo, en [10] se demuestra cómo estas redes pueden interpretar señales neurológicas para controlar dispositivos robóticos.

De manera similar, en [11] se utilizó una red neuronal *feedforward* para resolver la cinemática inversa de un manipulador planar de tres eslabones, logrando trayectorias precisas incluso en configuraciones complejas donde los métodos analíticos tradicionales no son viables.

Además, en [12] se desarrolló un controlador basado en redes neuronales que mejora el seguimiento de trayectorias repetitivas, compensando incertidumbres en el modelo del robot. Este enfoque, combinado con técnicas de estabilidad basadas en el criterio de Lyapunov, garantiza un control preciso y estable sin necesidad de ajustes constantes.

En el caso de los robots planar 2R, destacan por su velocidad, precisión y simplicidad, lo que los convierte en herramientas ideales para tareas como el trazado de patrones en superficies planas, el corte por láser en 2D, la experimentación educativa en robótica y manipulaciones sencillas que no requieren movimientos fuera de su área de trabajo [13]. De igual manera, enfoques como el ensamblaje de modelos permiten combinar las fortalezas de diversas redes neuronales para mejorar el rendimiento, reduciendo errores y aumentando la robustez en la predicción ante datos no entrenados [14].

En el contexto del aprendizaje automático uno de los inconvenientes más frecuentes es la consecución de los datos para el entrenamiento y validación del algoritmo. En consecuencia, en los últimos años ha venido creciendo el uso de datos sintéticos, obtenidos a partir de fuentes diferentes a medidas físicas en el robot. Este trabajo propone el uso de datos sintéticos para el entrenamiento de una red neuronal que calcule los ángulos articulares de un robot 2R para tareas de delineado de contornos.

El documento se estructura en tres secciones: la Sección 2 abarca la metodología, incluyendo el desarrollo de la red neuronal, las cinemáticas (inversa y directa) y el procesamiento de imágenes; la Sección 3 presenta los análisis de los resultados del funcionamiento de la red en los diferentes modos de trabajo y la Sección 4 expone las conclusiones y posibles trabajos futuros.

2. METODOLOGÍA

2.1. Componentes del Sistema y su Integración

La Fig. 1 presenta la interconexión de los componentes físicos que se integran en este trabajo, donde se observa un robot 2R diseñado en SolidWorks e impreso en 3D. Además, una Raspberry Pi 4B (8 GB de RAM) [15] actúa como

controladora del robot 2R, coordinando la interacción entre los componentes del sistema. La placa PCA9685 cuya señal de control se adquiere a través de I2C [16], es la encargada de variar la posición angular de los tres servomotores a través de canales PWM independientes con 12 bits de resolución cada uno. Finalmente, una cámara de 5 MP captura imágenes a través de vídeo en tiempo real, para su posterior procesamiento.

En la Raspberry Pi se ejecuta la interfaz gráfica de usuario (GUI) desarrollada en Qt designer, la cual permite ejecutar los dos modos de operación: trazar un nombre de máximo nueve caracteres y delinear los contornos de una imagen capturada a través de la cámara. El modelo de red neuronal de la cinemática inversa permite llevar a cabo la trayectoria deseada. La manipulación y supervisión del robot son realizadas de forma remota, y el procesamiento de imágenes se basa en investigaciones previas en el área [17], [18], [19].

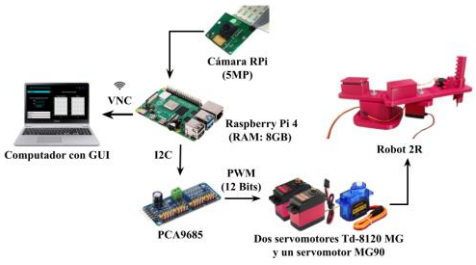


***Fig. 1**. Interconexión de componentes.*
***Fuente:** elaboración propia.*

2.2. Red Neuronal Perceptrón Multicapa (MLP)

La red neuronal Perceptrón Multicapa (MLP) es un tipo de red neuronal compuesta por varias capas (ver Fig. 2), una capa de entrada, una o más capas ocultas y una capa de salida; cada neurona está conectada a las de la siguiente capa, y la información se propaga hacia adelante utilizando funciones de activación no lineales.

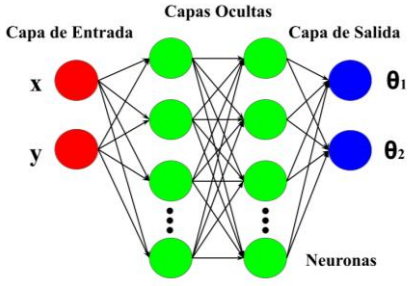


***Fig. 2**. Red Neuronal Perceptrón multicapa.*
***Fuente:** elaboración propia.*

El entrenamiento de la red neuronal MLP se realiza mediante el algoritmo de retropropagación, que ajusta los pesos de las conexiones para minimizar el error entre las salidas y los valores reales. La salida de cada neurona en un perceptrón se calcula utilizando (1).

$$y = f\left(\sum_{i=1}^{n} x_i w_i + b\right) \quad (1)$$

Donde:

- y: Salida de la neurona.
- x_i: Entradas de la neurona (con $i = 1, 2, 3, \ldots, n$).
- w_i: Pesos asociados a cada entrada x_i
- b: Sesgo (bias) de la neurona.
- f: Función de activación aplicada a la suma ponderada de las entradas.

El desarrollo de las redes neuronales se realizó en Google Colab utilizando el lenguaje Python, donde se creó un conjunto de 9.872.164 datos, de los cuales el 70% de dichos datos fue utilizado para el entrenamiento y el 30% para la validación del modelo. Este conjunto de datos (ver Fig. 3) se obtuvo a partir de la simulación del movimiento combinado de los ángulos θ_1 y θ_2 cada 0.001 radianes dentro del espacio de trabajo, para así obtener las posiciones X y Y (entradas de la red neuronal) con la MTH en (2).

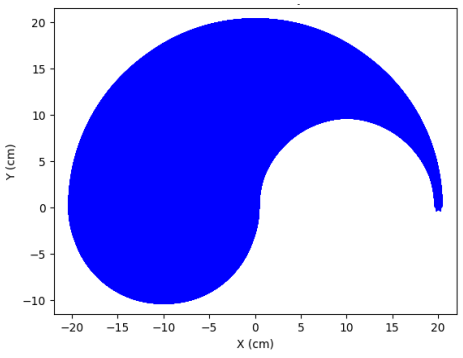


***Fig. 3**. Datos sintéticos en el espacio de trabajo del robot 2R.*
***Fuente:** elaboración propia.*

Para la implementación de la red neuronal, se utilizaron las bibliotecas TensorFlow, SciPy y scikit-learn. En la Tabla 1 se evidencian los datos más relevantes utilizados en el desarrollo de la red neuronal.

Tabla 1: *Parámetros de entrenamiento de la red neuronal.*

Parámetro	Valor
Variables de entrada	X y Y
Variables de salida	θ_1 y θ_2
Datos de entrenamiento	70%
Datos de validación	30%
Optimizador	Adam
Función de activación	Sigmoide
Función de pérdida	MSE
Métrica de error	MAE

Fuente: *elaboración propia*

2.3. Cinemáticas

2.2.1. Cinemática Directa

La Cinemática Directa (CD) a través de una Matriz de Transformación Homogénea (MTH) permite determinar la posición (traslación) y la orientación (rotación) del Punto Central de la Herramienta (PCH), con respecto al sistema de referencia (SC{0}), a partir de los movimientos de las articulaciones del robot. Es decir, dado un conjunto de ángulos, se obtienen las coordenadas y los ángulos de Euler (Roll, Pitch y Yaw) del PCH.

La Fig. 4 muestra el diagrama cinemático del robot SCARA, con los Sistemas Coordenados (SC) en cada articulación, teniendo en cuenta que las constantes del sistema son las longitudes de los eslabones l_1 y l_2 son 10 cm, mientras que l_3 es 2.5 cm; mientras que θ_1 y θ_2 son variables de 0° a 180°, de igual manera d_3 varía de 0 cm a 2.5 cm. Las dos primeras articulaciones son rotacionales y la tercera articulación es prismática, la cual varía la altura del efector final mediante un mecanismo de piñón-cremallera.

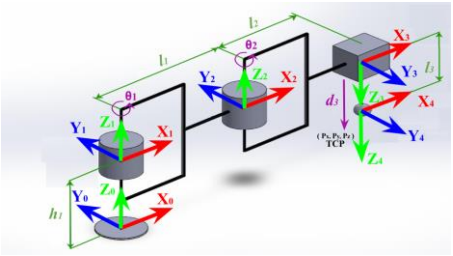


Fig. 4*. Esquema del robot SCARA y sus sistemas coordenados.*
Fuente: *elaboración propia.*

Se emplea el método de Denavit-Hartenberg (DH), el cual tiene un enfoque estándar y sistemático para describir la cinemática directa de robots de configuración serial, permitiendo modelar los movimientos de las articulaciones a través de la asignación de sistemas coordenados en cada articulación y en el PCH [20]. De esta manera, se determinan los parámetros DH (ver Tabla 2).

Tabla 2: *Parámetros del método DH*

i	θ_i	d_i	α_i	a_i
1	0	h_1	0	0
2	θ_1	0	0	l_1
3	θ_2	0	π	l_2
4	0	$l_3 + d_3$	0	0

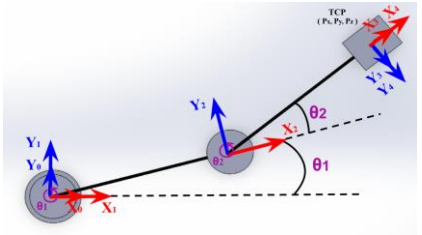
Fuente: *elaboración propia*

La MTH simbólica del SC{0} al SC{4} en (2) representa la posición y la orientación del PCH del robot de la Fig. 3, para cualquier valor de l_1, l_2, l_3, h_1, θ_1, θ_2 y d_3.

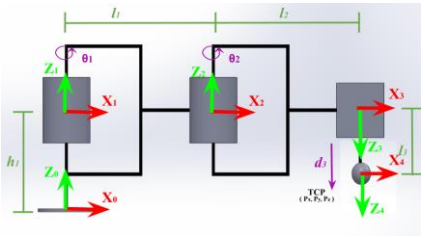
$$T_4^0 = \begin{bmatrix} c(\theta_1+\theta_2) & s(\theta_1+\theta_2) & 0 & l_2\cdot c(\theta_1+\theta_2)+l_1\cdot c(\theta_1) \\ s(\theta_1+\theta_2) & -c(\theta_1+\theta_2) & 0 & l_2\cdot s(\theta_1+\theta_2)+l_1\cdot s(\theta_1) \\ 0 & 0 & -1 & h_1-l_3-d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

2.2.2. Cinemática Inversa

La Cinemática Inversa (CI) calcula los movimientos de las articulaciones de un robot para una posición y orientación deseada del PCH. En la Fig. 5, se presentan las vistas del robot SCARA, en la cual se observa la proyección de la vista superior (Ver Fig. 5a) de las articulaciones rotacionales, las cuales representan los ángulos θ_1 y θ_2 en el plano X, Y, además, la proyección de la vista lateral (Ver Fig. 5b) muestra el movimiento lineal d_3 de la articulación prismática.



(a) Vista Superior.



(b) Vista Lateral.
Fig. 5*. Vistas del robot SCARA.*
Fuente: *elaboración propia.*

Mediante un análisis geométrico de la vista superior del robot 2R de la Fig. 5a se definieron (3) y (4), las cuales permiten calcular los ángulos θ_1 y θ_2 [21]. De igual manera, analizando la vista lateral del robot de la Fig. 5b se obtuvo (5) que representa la

distancia vertical d_3, correspondiente a la articulación prismática del robot SCARA.

$$\theta_1 = \tan^{-1}\frac{P_y}{P_x} - \tan^{-1}\frac{l_2 \cdot \pm\sqrt{1-\left(\frac{b^2-l_2^2-l_1^2}{2\cdot l_1\cdot l_2}\right)^2}}{l_1 + l_2 \cdot \frac{b^2-l_2^2-l_1^2}{2\cdot l_1\cdot l_2}} \quad (3)$$

$$\theta_2 = \tan^{-1}\frac{\pm\sqrt{1-\left(\frac{b^2-l_2^2-l_1^2}{2\cdot l_1\cdot l_2}\right)^2}}{\frac{b^2-l_2^2-l_1^2}{2\cdot l_1\cdot l_2}} \quad (4)$$

$$d_3 = h_1 - l_3 - P_z \quad (5)$$

2.4. Procesamiento de imágenes

Para realizar el procesamiento de imágenes, se delimitaron dos secciones dentro del espacio de trabajo del robot, como se presenta en la Fig. 6. La sección de color azul está destinada a la escritura de caracteres alfanuméricos (máximo 9), mientras que la sección de color rojo se utiliza para delimitar las dimensiones máximas de alto y ancho de la imagen a dibujar.

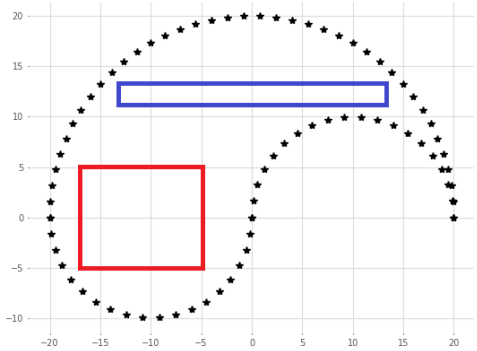
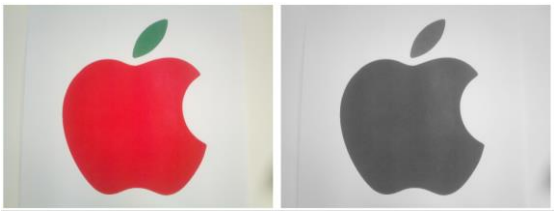


***Fig. 6**. Secciones para caracteres alfanuméricos y trazo de imágenes.*
***Fuente:** elaboración propia.*

A partir de la imagen capturada (ej: logo de Apple (ver Fig. 7a)), se realiza la binarización de la imagen (ver Fig. 7b), es decir, se convierte a una escala de blanco y negro para simplificar el análisis, ya que de esta manera solamente se considera la intensidad de los píxeles en lugar de trabajar con múltiples canales de color. Para este proceso, el umbral binario se ajusta con un valor mínimo de 117 y un valor máximo de 255. Dado que la imagen se tomará en tiempo real con una cámara de 5MP, es necesario ajustar la resolución a 1200 pixeles de ancho para delimitarla a la sección destinada al trazo de contornos, manteniendo la altura en una escala proporcional para evitar distorsionar la imagen.



(a) Imagen original. *(b) Imagen Apple binarizada.*
***Fig. 7**. Logo Apple.*
***Fuente:** elaboración propia.*

Posteriormente, se realiza la identificación de contornos de la imagen, tanto internos como externos, y se extraen las coordenadas de cada uno de los contornos obtenidos. En el caso de la imagen de Apple se identificaron dos contornos. Finalmente, cada una de las coordenadas obtenidas en el paso anterior se divide entre 100 para escalizar la imagen y por tanto, se ajusta para la sección de color rojo de la Fig. 6, donde se trazarán los contornos, los cuales se representan en la Fig. 8.

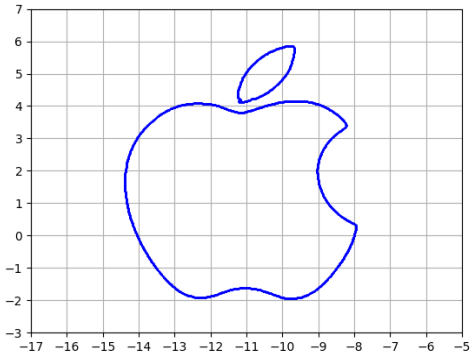


***Fig. 8**. Imagen escalada dentro de la sección del área de trabajo.*
***Fuente:** elaboración propia.*

3. RESULTADOS

La Fig. 9 presenta el montaje físico del robot SCARA, destacando la integración de los diferentes componentes de la Fig. 1, como la cámara, la Raspberry Pi 4, el computador con la GUI y el módulo PCA9685. Cabe destacar que, para este último, se diseñó una PCB para simplificar las conexiones entre dispositivos. El usuario selecciona uno de los dos modos de operación mediante una GUI, la cual también presenta la simulación del robot SCARA, ambos modos utilizan el enfoque de red neuronal de ensamblaje de modelos que reemplaza la cinemática inversa geométrica.

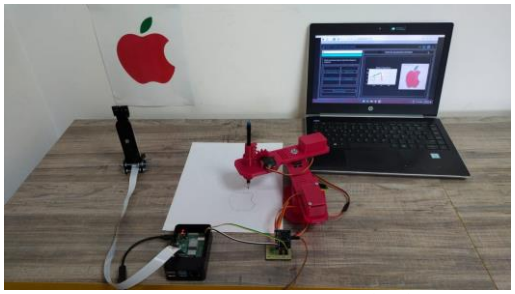


***Fig. 9**. Interacción entre componentes y el Robot SCARA.*
***Fuente:** elaboración propia.*

Posteriormente al entrenamiento de los modelos de las redes neuronales, se implementó un método iterativo que consistió en realizar varias pruebas para determinar la mejor combinación de parámetros teniendo en cuenta el desempeño de los modelos. En la Tabla 3 se presentan los errores RMSE de modelos de redes neuronales para diferentes números de neuronas, número de capas ocultas y épocas, obteniendo como mejor resultado el modelo de ensamble.

***Tabla 3:** Parámetros del modelo final de la Red Neuronal.*

No.	Modelo	Neur.	Capas	Ép.	Error (RMSE)
1	Modelo 1	150	1	90	$\theta_1 = 1.6086$
					$\theta_2 = 0.3717$
2	Modelo 2	200	2	77	$\theta_1 = 0.4204$
					$\theta_2 = 0.3072$
3	Ensamble Modelo 3.1	200	2	77	$\theta_1 = 0.2293$ $\theta_2 = 0.0739$
	Ensamble Modelo 3.2	210	2	33	
	Ensamble Modelo 3.3	250	2	80	

***Fuente:** elaboración propia.*

La Fig. 10 muestra las posiciones X y Y del logo de Apple generadas por cada uno de los modelos de redes neuronales en comparación con la cinemática inversa geométrica, con lo cual se demuestra de manera gráfica que el modelo 3 es el que se aproxima más a la cinemática inversa geométrica.

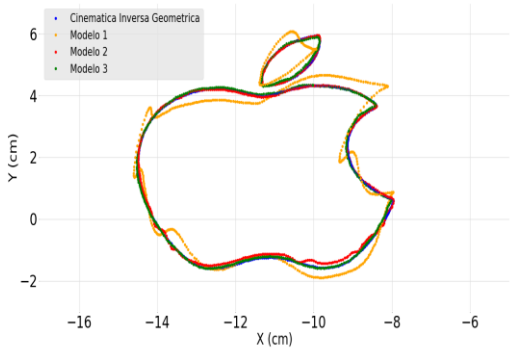


***Fig. 10**. Comparación de las trayectorias de la cinemática inversa geométrica vs modelos de redes neuronales para el logo de Apple.*
***Fuente:** elaboración propia.*

La Fig. 11 presenta la comparación de θ_1 y θ_2 calculados por la cinemática inversa geométrica [22] con los ángulos (θ_1 y θ_2) obtenidos con los tres modelos de redes neuronales, para cada uno de los 410 puntos (posiciones en X y Y) que conforman los contornos del logo de Apple. Así mismo, en la Fig. 11 se observa el comportamiento de cada uno de los tres modelos de redes neuronales con respecto a la cinemática inversa geométrica, donde se evidencia que los ángulos (θ_1 y θ_2) del modelo 3 son los que mejor siguen el comportamiento de θ_1 y θ_2 de la cinemática inversa geométrica.

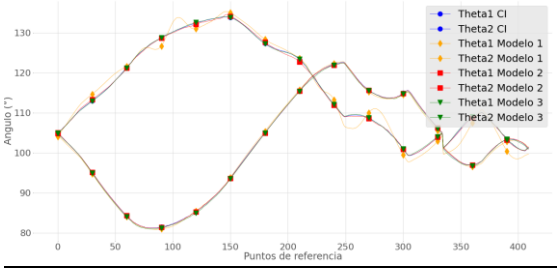


***Fig. 11**. Comparación cinemática inversa geométrica vs modelos de redes neuronales para el logo de Apple.*
***Fuente:** elaboración propia.*

Teniendo en cuenta la Tabla 3 y la Fig. 11, se determinó que el mejor modelo de la red neuronal es el modelo 3, un modelo de ensamble que promedia tres modelos. Este enfoque otorga una mayor capacidad de generalización, superando así a los modelos individuales. También se evidenció que incrementar el número de capas ocultas, en lugar de simplemente aumentar la cantidad de neuronas, contribuyó de manera significativa a mejorar el desempeño de la red, ya que así se consiguió disminuir el error RMSE. Por esta razón, se seleccionó este modelo para implementarlo en la estimación de θ_1 y θ_2 en el robot 2R.

3.1. Modo 1: Escritura de nombre

La Fig. 12 presenta la GUI en el modo de escritura de un *string* hasta de nueve caracteres, en la cual se observa que el nombre ingresado es "NICOLAS", mostrando también la simulación del robot para el trazo de dicho nombre.

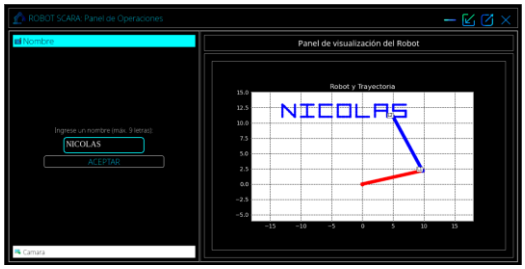


***Fig. 12**. Ingreso del nombre y la simulación en la GUI.*
***Fuente:** elaboración propia.*

La Fig. 13 muestra un detalle significativo para realizar una escritura más limpia y ordenada de un *string* y es el trazo discontinuo del nombre, es decir, los espacios entre las letras, lo cual es resultado del movimiento lineal y vertical de la articulación prismática que realiza las tareas de elevación y descenso del lápiz en las transiciones entre caracteres. Este comportamiento es evidencia del funcionamiento del mecanismo piñón-cremallera.

***Fig. 13**. Escritura del nombre NICOLAS con el robot SCARA físico.*
***Fuente:** elaboración propia.*

3.2. Modo 2: Captura de imagen para el dibujo

La Fig. 14 presenta la GUI en el modo de captura de la imagen con la cámara, donde se visualizan la imagen capturada y la identificación de los contornos del logotipo de Apple. Además, se observa la simulación del movimiento del robot en tiempo real.

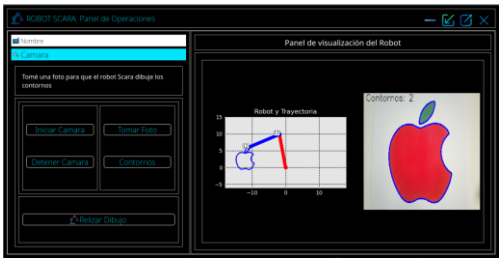


***Fig. 14**. Captura de imagen y simulación en la GUI.*
***Fuente:** elaboración propia.*

En la Fig. 15 se evidencia el trazo final de los contornos del logo de Apple, realizado físicamente por el robot. Por tanto, se demostró el trazo de contornos de imágenes procesadas en tiempo real a través de una cámara, lo que muestra flexibilidad para adaptarse a nuevas posiciones dentro del espacio de trabajo. Sin embargo, es importante resaltar que los servomotores tienen un rol importante en el trazo de los contornos ya que la resolución de estos actuadores no es el esperado, generando así un error en los ángulos θ_1 y θ_2 físicos y por este motivo el logo dibujado tiene algunas imperfecciones en el trazo.

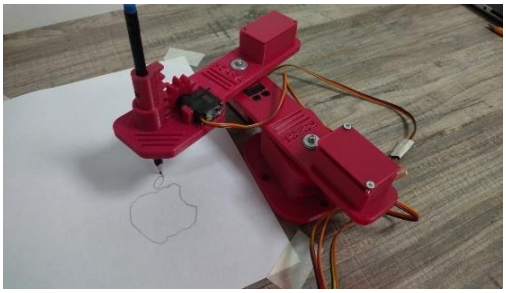


***Fig. 15**. Trazo de los contornos del logo de Apple con el robot SCARA físico.*
***Fuente:** elaboración propia.*

4. CONCLUSIONES

Los resultados obtenidos en este proyecto evidenciaron que el uso de una red neuronal para la predicción de la cinemática inversa de un robot 2R es una alternativa eficaz al tener errores RMSE bajos. Este método tiene la ventaja para aplicaciones con robots de mayor complejidad y grados de libertad, donde el procesamiento computacional de la cinemática inversa incrementa considerablemente, así como las posiciones singulares de dichos robots.

Al implementar todo el sistema en una Raspberry Pi, se logró una operación más rápida y eficiente del robot, la cámara y la GUI, ya que no se tienen tiempos de latencia asociados a las comunicaciones seriales. Además, la Raspberry Pi no solo gestiona

la GUI y el hardware, sino que también ejecuta el modelo de red neuronal, centralizando todas las operaciones en un único dispositivo. Es importante destacar que la red neuronal fue entrenada con datos sintéticos generados para abarcar diferentes puntos dentro del área de trabajo del robot, lo que permitió entrenar el modelo de manera efectiva para resolver tareas de cinemática inversa geométrica con gran fiabilidad.

La cámara integrada en el sistema mostró un rendimiento satisfactorio al capturar imágenes a partir de vídeo en tiempo real, con lo cual se consiguió identificar los contornos. Sin embargo, es fundamental considerar factores de iluminación y filtros al momento de la captura de la imagen, ya que estos afectan la calidad de identificación de los contornos. Además, es fundamental ajustar los parámetros del umbral binario para convertir la imagen en blanco y negro de manera correcta; lo cual facilita la detección de contornos al resaltar las áreas de interés. Es importante calibrar estos valores de umbral en función de las condiciones de iluminación, para garantizar una interpretación precisa de todos los detalles de la imagen.

Con respecto al trazo del robot físico, se observaron algunas limitaciones en la precisión de los movimientos, en parte debido al rendimiento de los servomotores utilizados. Se destacó que los movimientos lineales son trazados de mejor manera en el robot físico en comparación con los movimientos semicirculares, los cuales presentan mayor dificultad para mantener la precisión y fluidez deseadas. Para mejorar la calidad y fluidez en el trazo es recomendable considerar la sustitución de servomotores por motores de Corriente Continua (CC) con control de posición, lo cual permitiría una mayor precisión y suavidad en los movimientos del robot, para así obtener un dibujo final igual que en la simulación.

REFERENCIAS

[1] J. L. Sarmiento-Ramos, “Aplicaciones de las redes neuronales y el deep learning a la ingeniería biomédica,” *Revista UIS Ingenierías*, vol. 19, no. 4, 2020, doi: 10.18273/revuin.v19n4-2020001.

[2] J. F. Cortes Zarta, Y. A. Giraldo Tique, and C. F. Vergara Ramirez, “Red neuronal convolucional para la percepción espacial del robot InMoov a través de visión estereoscópica como tecnología de asistencia,” *Enfoque UTE*, vol. 12, no. 4, 2021, doi: 10.29019/enfoqueute.776.

[3] Y. He *et al.*, “Dynamic Modeling, Simulation, and Experimental Verification of a Wafer Handling SCARA Robot with Decoupling Servo Control,” *IEEE Access*, vol. 7, pp. 47143–47153, 2019, doi: 10.1109/ACCESS.2019.2909657.

[4] A. Visioli and G. Legnani, “On the trajectory tracking control of industrial SCARA robot manipulators,” *IEEE Transactions on Industrial Electronics*, vol. 49, no. 1, 2002, doi: 10.1109/41.982266.

[5] K. S. Mohamed Sahari, K. H. Weng, Y. W. Han, A. Anuar, M. Z. Baharuddin, and S. S. K. Mohideen, “Design and development of A 4-dof SCARA robot for educational purposes,” *Jurnal Teknologi (Sciences and Engineering)*, vol. 54, pp. 193–215, 2011, doi: 10.11113/jt.v54.810.

[6] A. Kotani and S. Tellex, “Teaching robots to draw,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2019. doi: 10.1109/ICRA.2019.8793484.

[7] L. H. Ríos González, M. Bueno López, and S. Sánchez A., “Generación de trayectorias para un robot móvil empleando redes neuronal,” *Scientia et Technica*, vol. 2, pp. 94–99, 2008. [Online]. Available: https://dialnet.unirioja.es/servlet/articulo?codigo=4742539

[8] J. G. C. Yufra Torres and K. De La Torre Muña, “Diseño de un robot móvil recolector y compactador de botellas de plástico utilizando redes neuronales en playas con arena fina,” *Universidad Ricardo Palma - URP*, 2020, [Online]. Available: https://hdl.handle.net/20.500.14138/3529

[9] J. Hernández Díaz, M. Iparraguirre Monzón, C. Montenegro Castro, J. Alva Alcántara, and E. Manzano Ramos, “MODELADO NO PARAMÉTRICO DE ROBOT SCARA 3-GDL UTILIZANDO REDES NEURONALES RECURRENTES,” *Revista de Investigaciones*, vol. 10, no. 4, pp. 379–390, Dec. 2021, doi: 10.26788/riepg.v10i4.3502.

[10] M. X. González, A. M. Brugiati, D. H. Cornejo, and C. I. Pinzón, “Prototipo de mano robótica controlada mediante el procesamiento de señales cerebrales utilizando redes neuronales recurrentes,” *Revista de Iniciación Científica*, vol. 6, 2021, doi: 10.33412/rev-ric.v6.0.3154.

[11] A.-V. Duka, “Neural Network based Inverse Kinematics Solution for Trajectory Tracking of a Robotic Arm,” *Procedia*

Technology, vol. 12, 2014, doi: 10.1016/j.protcy.2013.12.451.

[12] N. Cobanoglu, E. Tatlicioglu, and E. Zergeroglu, "Neural network based repetitive learning control of robot manipulators," in *Proceedings of the American Control Conference*, 2017. doi: 10.23919/ACC.2017.7963781.

[13] A. De Luca, R. Mattone, and G. Oriolo, "Control of underactuated mechanical systems: Application to the planar 2R robot," in *Proceedings of the IEEE Conference on Decision and Control*, 1996. doi: 10.1109/cdc.1996.572718.

[14] J. J. Sprockel, A. Fandiño, W. G. Chaves, C. O. Benavides, and J. J. Diaztagle, "Ensamble de redes neuronales artificiales ponderado mediante características operativas para el pronóstico de la insuficiencia cardiaca aguda," *Revista Colombiana de Cardiología*, vol. 30, no. 5, 2023, doi: 10.24875/rccar.22000065.

[15] Raspberry Pi Ltd, "Raspberry Pi 4 Model B Datasheet," 2024

[16] NXP Semiconductors, "PCA9685 16-channel, 12-bit PWM Fm+ I2C-bus LED controller," *NXP Semiconductors N.V*, no. April, 2015.

[17] M. J. Delgado-Gutiérrez, D. F. Herrera-Guillén, L. M. Medina-Barragán, and J. P. Corredor–Gómez, "Implementación de un sistema de procesamiento de imágenes integrado con Raspberry PI 2B para reconocimiento y recolección de fresas maduras," *Revista Politécnica*, vol. 13, no. 25, 2017, doi: 10.33571/rpolitec.v13n25a6.

[18] G. I. Viera Maza, "Procesamiento de imágenes usando OpenCV aplicado en Raspberry Pi para la clasificación del cacao," *Thesis*, 2017.

[19] G. Viera, "Procesamiento de imágenes usando OpenCV aplicado en Raspberry Pi para la clasificación del cacao," *Thesis*, 2017.

[20] C. R. Arroyo Flores and F. J. Triveño Vargas, "Modelamiento y simulación de un robot SCARA de tres grados de libertad empleando SolidWorks y Matlab/Simulink®," *SciELO Analytics*, vol. 10, Mar. 2022, [Online]. Available: http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S1683-07892022000100318

[21] S. Kucuk and Z. Bingul, "Robot Kinematics: Forward and Inverse Kinematics," in *Industrial Robotics: Theory, Modelling and Control*, 2006. doi: 10.5772/5015.

[22] A. ; B. N. ; R. D. B. F. Mancipe, "Desarrollo de un brazo robótico de 2 DOF impreso en 3d a través de un pic 18f46k22 y MATLAB para el trazo de contornos de imágenes," *I + T + C Investigación, Tecnología y Ciencia. Vol 1. Num. 17. 2023.*, 2023, [Online]. Available: https://revistas.unicomfacauca.edu.co/ojs/index.php/itc/article/view/406/330