**Artículo Original**

**Análisis de Cinemática para Manipulador Robótico 2R no-planar.**

**Kinematic Analysis for Non-Planar 2R Robotic Manipulator.**

**Adriano Alberto Spinola Olivares1,\***

1Facultad de Ingeniería Automática y Biomédica. Universidad Tecnológica de La Habana “José Antonio Echeverría” (Cujae). Calle 114 No 11901 entre 119 y 127, Marianao, La Habana, Cuba.

\*Correspondencia:[adrianosp@automatica.cujae.edu.cu](mailto:adrianosp@automatica.cujae.edu.cu)

Estedocumentoposeeuna[licenciaCreativeCommonsReconocimiento/NoComercial4.0Internacional](http://creativecommons.org/licenses/by-nc/4.0/)

**Resumen**

La investigación considera el control en el movimiento de un robot no-plano 2R consistente de dos articulaciones conectadas a través de dos uniones revolutivas. El objetivo del estudio es realizar un análisis exhaustivo a partir de las dimensiones del manipulador robótico para obtener primeramente las matrices de transformación características de la cinemática directa con apoyo de los parámetros de Denavit–Hartenberg (DH) para posteriormente lograr el control del posicionamiento del efector final del robot entro del espacio de trabajo del mismo dada una trayectoria definida (cinemática inversa) obtenido a través de métodos algebraicos y de Machine Learning (Red Neuronal Artificial, ANN).

**Palabras clave**:manipulador robótico 2R, Red Neuronal, Cinemática inversa, Cinemática directa.

**Abstract**

The research considers the control of the movement of a non-planar 2R robot consisting of two joints connected through two revolute joints. The objective of the study is to conduct a thorough analysis based on the dimensions of the robotic manipulator to first obtain the characteristic transformation matrices of forward kinematics using Denavit–Hartenberg (DH) parameters and subsequently achieve control of the positioning of the robot's end-effector within its workspace along a defined trajectory (inverse kinematics) obtained through algebraic methods and Machine Learning (Artificial Neural Network, ANN).

**Keywords**: 2R robotic manipulator, Neural Network, Inverse kinematics, Direct kinematics.

**1. Introducción**

La robótica ha sido, desde sus inicios, una disciplina en constante evolución que ha transformado diversas facetas de nuestra vida y la industria. Desde los primeros autómatas mecánicos de la antigüedad hasta los sofisticados robots de hoy en día, la robótica ha recorrido un largo camino, impulsada por avances tecnológicos y la creatividad humana.

En el pasado, la robótica se centraba principalmente en el diseño y construcción de máquinas que pudieran realizar tareas repetitivas y peligrosas, con una precisión y eficiencia inalcanzables para los humanos. Los primeros manipuladores robóticos, utilizados en líneas de montaje y manufactura, representaron un hito en la automatización industrial, permitiendo una producción en masa más rápida y segura. Los manipuladores robóticos modernos no solo realizan tareas mecánicas, sino que también interactúan con su entorno de manera inteligente, gracias a la integración de sensores avanzados y algoritmos de control sofisticados.

La robótica es un campo en desarrollo activo, pero es probable que aún no se haya alcanzado su máximo potencial. En primer lugar, no existe una solución única (marco) que se ajuste a todos los modelos de robots. Es necesario utilizar herramientas altamente especializadas para resolver problemas específicos. En segundo lugar, cada robot es un mecanismo con características únicas que requieren un enfoque individual. Esto se relaciona con la descripción de la cinemática, la dinámica del robot, cómo se conectan los enlaces y su orden, si un cuerpo sólido forma la base del sistema mecánico y otras particularidades. Por lo tanto, aún quedan muchos descubrimientos por hacer en robótica, y a menudo hoy en día se refieren a modelos de robots específicos diseñados para resolver un rango estrecho de problemas. En [[15](#quince)], se describen tendencias en el sector de la robótica industrial y se muestra gráficamente el crecimiento del número de publicaciones en el campo de la robótica industrial. El artículo [[16](#dieciseis)]muestra las tendencias en la evolución de los robots de servicio y enumera algunos problemas no resueltos en robótica, como "la falta de generalización y formalismo en las clasificaciones y la taxonomía, el valor utilitario percibido actualmente, la modelización y estimación de la batería y la autonomía, la ética e incluso problemas de diseño relacionados con sesgos de género basados en la ocupación del robot".

Encontrar una solución para la Cinemática inversa de un manipulador robótico puede ser desafiante, sobre todo cuando aumentan los grados de libertad del mismo, por tanto, puede volverse engorroso encontrar el sistema de ecuaciones que definen los ángulos de las uniones (en el caso de estudio uniones revolutivas) para cada una de las posiciones y rotaciones del efector final dentro del espacio de trabajo del manipulador.

Los métodos de aprendizaje automático se utilizan ampliamente en robótica, no solo para resolver la tarea cinemática directa (DKT) y la tarea cinemática inversa (IKT). El aprendizaje automático permite resolver problemas en los que es difícil derivar dependencias mediante métodos clásicos. La revisión de Laith Alzubaidi y sus coautores [[17](#diecisiete)] afirma que "recientemente, el aprendizaje automático (ML) se ha vuelto muy extendido en la investigación y se ha incorporado en una variedad de aplicaciones, incluyendo minería de texto, detección de spam, recomendación de videos, clasificación de imágenes y recuperación de conceptos multimedia" y que "la efectividad de un algoritmo depende en gran medida de la integridad de la representación de los datos de entrada". Por ejemplo, los sistemas holonómicos en la resolución de DKT e IKT tienen el problema de la inexactitud geométrica del resultado obtenido. El artículo [[14](#catorce)] discute una solución a este problema utilizando la Red Neuronal Artificial (ANN) y "se entrena para predecir el error de posición en el espacio de trabajo para las coordenadas de entrada dadas de la geometría plana".

Es bien sabido que en robótica existen dos tareas cinemáticas: directa e inversa (DKT e IKT). La solución de la DKT nos da una respuesta inequívoca, mientras que la IKT no siempre proporciona una solución única o inequívoca. Debemos basarnos en el objetivo que se persigue al resolver la IKT, que puede ser encontrar múltiples soluciones y procesar los resultados, o derivar una única solución que cumpla los criterios. La IKT es un paso importante en la resolución de muchos problemas cinemáticos en robótica, por ejemplo, resolver la IKT es un paso importante para calcular el espacio de trabajo de un robot. Cabe señalar que la aplicación de la Inteligencia Artificial (IA) cada vez se superpone más con los problemas de robótica y proporciona muchas formas novedosas de resolver diferentes problemas. La ubicación exacta no solo del efector final, sino también de otros enlaces en el trayecto, puede ser crítica, dependiendo de las tareas que el robot está diseñado para realizar. Por ejemplo, al enseñar a un robot a repetir movimientos para entrenar los músculos flexores-extensores para quienes necesitan rehabilitación, no es suficiente que el robot siga el movimiento del efector final a lo largo de una trayectoria dada; también necesitamos el movimiento más preciso de todos los enlaces debido al grado limitado de libertad de la mano humana. Hay muchas tareas donde este criterio es importante según la condición o formulación del problema. Además, la solución de la IKT es importante en el cálculo del espacio de trabajo del robot. Conocer el espacio de trabajo del robot ayuda a garantizar que los mecanismos y partes del robot funcionen de manera menos traumática para el mismo. Este estudio propone resolver el problema de la cinemática directa e inversa para un brazo robótico no-plano 2R apoyándose en el empleo de redes neuronales artificiales (ANN) para esta última. A partir de los resultados obtenidos será posible caracterizar la posición del efector final en el espacio para cualquier par de ángulos en las uniones y viceversa. Las características y ventajas de la solución propuesta se presentan en las secciones posteriores del documento.

**2. Materiales y Métodos**

**Metodología de trabajo**

Los módulos desarrollados para implementar el objetivo de la investigación fueron escritos en Python. Se implementaron diferentes modelos de una ANN totalmente acoplada con diferentes topologías utilizando los frameworks TensorFlow y Scikit-learn. Los gráficos se construyeron con la ayuda de la biblioteca Matplotlib.

Los pasos de investigación implementados para trabajar en el simulador y el prototipo del modelo de robot y para encontrar una topología ANN adecuada son los siguientes:

1. Diseño del circuito del futuro prototipo del robot, descripción de la cinemática del robot, implementación del modelo en el entorno MatLab utilizando la librería Robotics Toolbox implementada por Peter Corke [[13](#PC)].
2. Desarrollo de un módulo para construir el espacio de trabajo del robot y sus límites.
3. Desarrollo de un módulo para generar una trayectoria pseudoaleatoria dentro del espacio de trabajo.
4. Montaje de la construcción del prototipo del robot con empleo de la placa ESP-WROOM-32.
5. Desarrollo de un módulo para generar muestras de entrenamiento y prueba basadas en los movimientos del modelo de robot en el entorno MatLab.
6. Diseño de las topologías ANN, verificación, entrenamiento, prueba.
7. Visualización de resultados, evaluación del rendimiento de la ANN y selección de los mejores modelos.
8. Desarrollo de un módulo para probar, verificar los resultados del entrenamiento y recrear los movimientos del robot en el modelo de robot en el entorno MatLab.
9. Prueba del modelo implementado en la placa ESP-WROOM-32.

**Descripción del robot manipulador**

En el presente trabajo se operará con un brazo robot no-planar para probar la hipótesis y realizar la investigación. Se considerará toda la metodología para obtención de los parámetros de posición y rotación del efector final con respecto a tierra para a partir de estos obtener los parámetros de IKT del brazo robótico. El estudio trabaja con un robot no-planar 2R debido a sus facilidades constructivas el cual consiste de dos articulaciones conectadas en paralelo a través de dos uniones revolutivas.

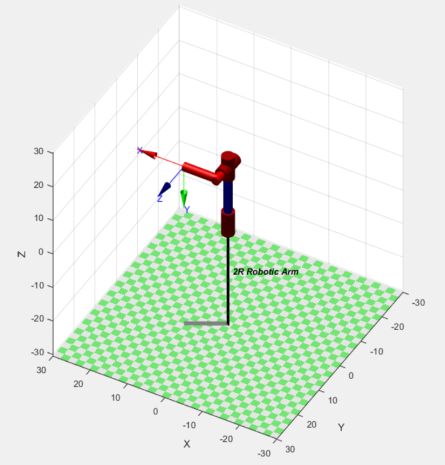
El prototipo del robot se presenta en la Figura 1:



**Figura 1:** Robot 2R no-planar

El prototipo de robot ha sido ensamblado a partir de materiales básicos como cartón y utilizando la placa ESP-WROOM-32 con el empleo de dos servomotores para realizar la rotación y control. Los dos servomotores (modelo SG 90) juegan el rol de actuadores en la implementación física. La modelación del robot ha sido realizada en MatLab con el empleo de la librería Robotics Toolbox implementado por Peter Corke.

Esta se muestra en la Figura 2:



**Figura 2:** Modelo MatLab de Robot 2R no-planar.

La longitud de las articulaciones del robot es de L1 = 16.5 cm, L3 = 9cm. Por otro lado, se ha considerado incluir el offset presente entre la primera y segunda articulación al tratarse de un robot no-planar el cual es L2 = 5cm ya que esto afecta sustancialmente los resultados finales del efector. El diseño del robot ha sido seleccionado en concordancia con las características del modelo de servomotor empleado y el peso de los elementos utilizados para la construcción. Las características del servomotor SG 90 pueden ser tomadas del sitio web oficial del fabricante.

**Cinemática Directa. Obtención de parámetros de Denavit-Hartenberg**

A partir de las especificaciones anteriormente mencionadas se procede a la obtención de los parámetros de Denavit–Hartenberg (DH) del robot. Los parámetros de Denavit-Hartenberg son los cuatro parámetros asociados con la convención DH para adjuntar marcos de referencia a los eslabones de una cadena cinemática espacial o manipulador robot. Los cuatro parámetros se presentan a continuación:

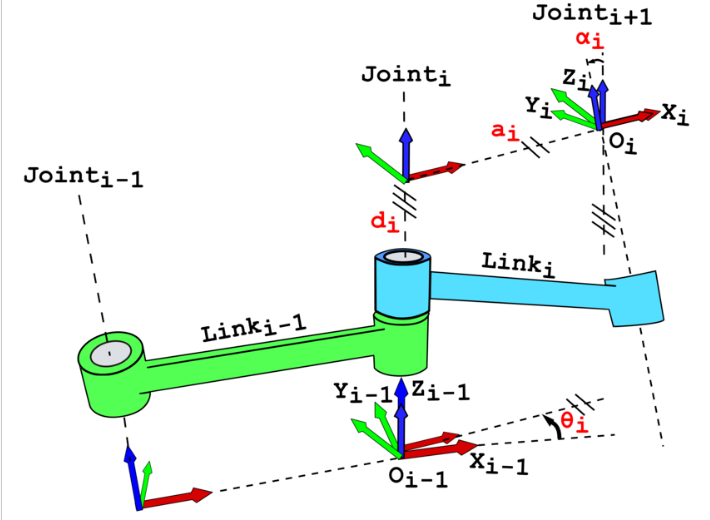
di: Desplazamiento a lo largo de Z anterior a la normal común.

Θi: Ángulo sobre la Z anterior desde la Xi-1 hasta la Xi nueva.

ai: Longitud de la normal común. Suponiendo una unión revoluta, este es el radio alrededor de Zi-1.

αi: Ángulo sobre la normal común, desde el eje Zi-1 hasta el eje Zi.

La Figura 3 describe la relación de los parámetros para una unión revoluta.



**Figura 3:** Los cuatro parámetros de la convención DH clásica se muestran en texto rojo, que son θi, di, ai, αi. Con esos cuatro parámetros, podemos traducir las coordenadas de Oi–1, Xi–1, Yi–1, Zi-1 a Oi, Xi, Yi, Zi.La tabla 1 recoge los resultados obtenidos para nuestro manipulador robótico:

**Tabla 1**

Parámetros DH.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Unión | ai-1 | αi-1 | di | θi |
| **1** | L1 | 0 | 0 | Θ1 |
| **2** | 0 | 90° | 0 | 0 |
| **3** | L3 | 0 | L2 | Θ2 |

A través de la obtención de estos parámetros es posibleconfeccionar las matrices de transformación correspondientes para concatenar cada una de las uniones y de esta forma obtener la posición y orientación del efector final con respecto a tierra para cualquier combinación de ángulos de Θ1 y Θ2.

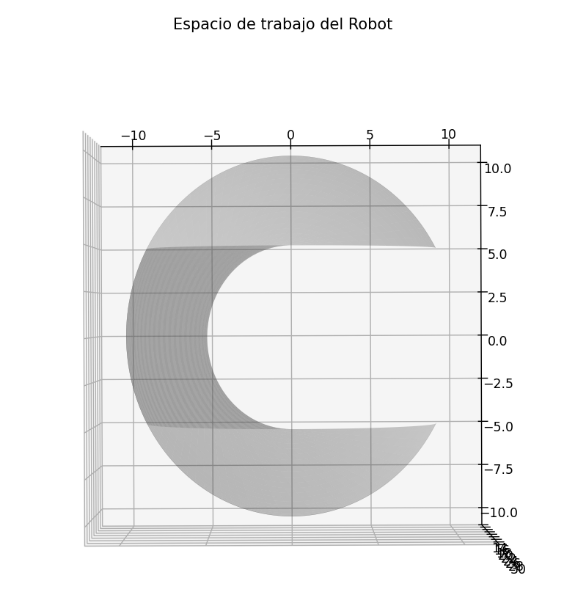
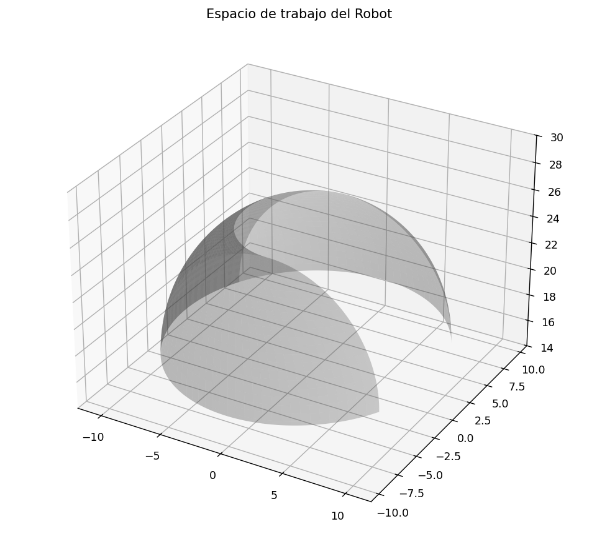
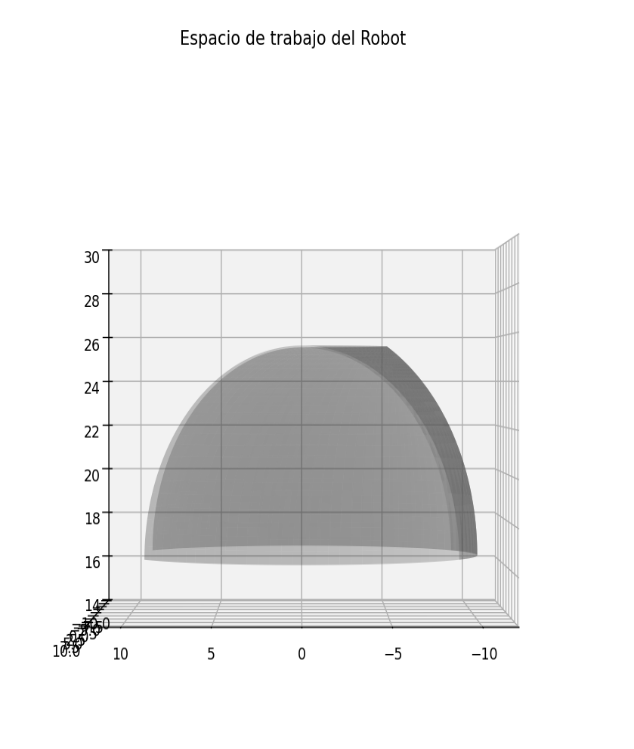
A partir de ello se obtiene la siguiente matriz de transformación del efector final con respecto a tierra:

(1)

A través de la matriz de transformación es posible obtener las ecuaciones que describen la cinemática directa del efector final en el espacio las cuales corresponden a los valores a14, a24 y a34 de la matriz de transformación:

(2)

Realizando cálculos según la fórmula (1), teniendo en cuenta las restricciones de los servomotores, obtenemos el conjunto de puntos del espacio de trabajo del robot-manipulador. El paso de cálculo es igual a 1 grado, con ángulos desde 0 hasta 180 grados. Luego, se sustraen del conjunto los puntos que son puntos de límite. Con base en esto, y considerando las longitudes dadas de los eslabones, el espacio de trabajo tiene la superficie mostrada en la Figura 4.



**Figura 4:** Espacio de Trabajo del Robot No-Planar 2R con restricciones. De izquierda a derecha: Plano XZ, Perspectiva, Plano XY.

Los límites de alcance del brazo robótico en el espacio con respecto a tierra (en cm) es:

**X:** máximo = 9.0, mínimo = -10.295

**Y:** máximo = 9.294, mínimo = -10.295

**Z:** máximo = 25.498, mínimo = 16.5

**Propuesta de método para cinemática inversa.**

Una vez conocidos los parámetros de Cinemática Directa y el espacio de trabajo del manipulador robótico es posible obtener los parámetros de cinemática inversa.

A través de la cinemática directa obtenemos los valores de la posición del efector final en términos de las variables espaciales X, Y, Z para cualquier par de valores de los ángulos y . Para el caso de la cinematica inversa lo que se busca es que, dados los valores de X, Y, Z para cualquier posición del efector final dentro del espacio de trabajo, sea posible obtener los valores en radianes de los ángulos de las uniones para lograr la correcta colocación del gripper. En el caso de estudio no se contempla la orientación del mismo ya que no es posible obtener una posición en el brazo robótico a través de más de una trayectoria, por lo que para una posición P1 solo habrá un Angulo de rotación R1.

Existen varias formas de obtener la cinemática inversa a partir de la cinemática directa. Algunas de las principales son el método algebraico, el método geométrico y el uso de redes neuronales.

El método algebraico involucra el uso de ecuaciones matemáticas para resolver la cinemática inversa. Este método se basa en expresar las coordenadas del efector final en términos de los ángulos de las articulaciones y luego resolver las ecuaciones resultantes para obtener estos ángulos. Este método puede ser complejo y poco práctico para robots con muchas articulaciones o configuraciones geométricas complicadas. Además, puede haber múltiples soluciones o incluso ninguna solución válida en ciertos casos.

Proponemos el siguiente método para resolver la cinemática inversa (IKT) para un robot de múltiples eslabones 2R:

* La cinemática inversa del Ángulo de la unión K2 () se resuelve empleando métodos algebraicos.
* La cinemática inversa del Ángulo de la unión K1() se resuelve utilizando Redes Neuronales Artificiales (ANN). Se dan los valores de la posición del efector final en el espacio y el valor del ángulo de la unión K2 ()
* En la entrada de la ANN se encuentran las coordenadas del del efector final y el valor del ángulo de la unión K2 ().
* El modelo ANN propuesto da una solución única en la salida.

Así, la ANN proporciona un resultado único y no ambiguo. Los datos de entrada, que contienen el valor del ángulo de la unión K2, permiten evitar la ambigüedad de la solución.

**Cinemática Inversa. Empleo de Red Neuronal Artificial (ANN).**

El valor de para cualquier posición del efector final ha sido posible obtenerla a través del método algebraico, ya que contamos con las variables necesarias para su obtención planteando a partir de (2) que:

(3)

(4)

(5)

El uso de redes neuronales para resolver la cinemática inversa es un enfoque basado en el aprendizaje automático. Las redes neuronales pueden aprender a mapear las posiciones del efector final a los ángulos de las articulaciones mediante el entrenamiento con datos de ejemplos.

No es posible obtener el valor de a través del método algebraico ni del método geométrico sin realizar algún tipo de aproximación a través de métodos numéricos, por lo que se ha propuesto el empleo de redes neuronales para la obtención de este ángulo.

Se propone el uso de un perceptrón multicapa el cual será expuesto a profundidad en secciones posteriores.

Las muestras de entrenamiento se generaron en incrementos de 0.01 grado, ya que un incremento mayor no tuvo un efecto suficiente en el proceso de aprendizaje de la ANN, mientras que un incremento menor no produjo mejores resultados de aprendizaje.

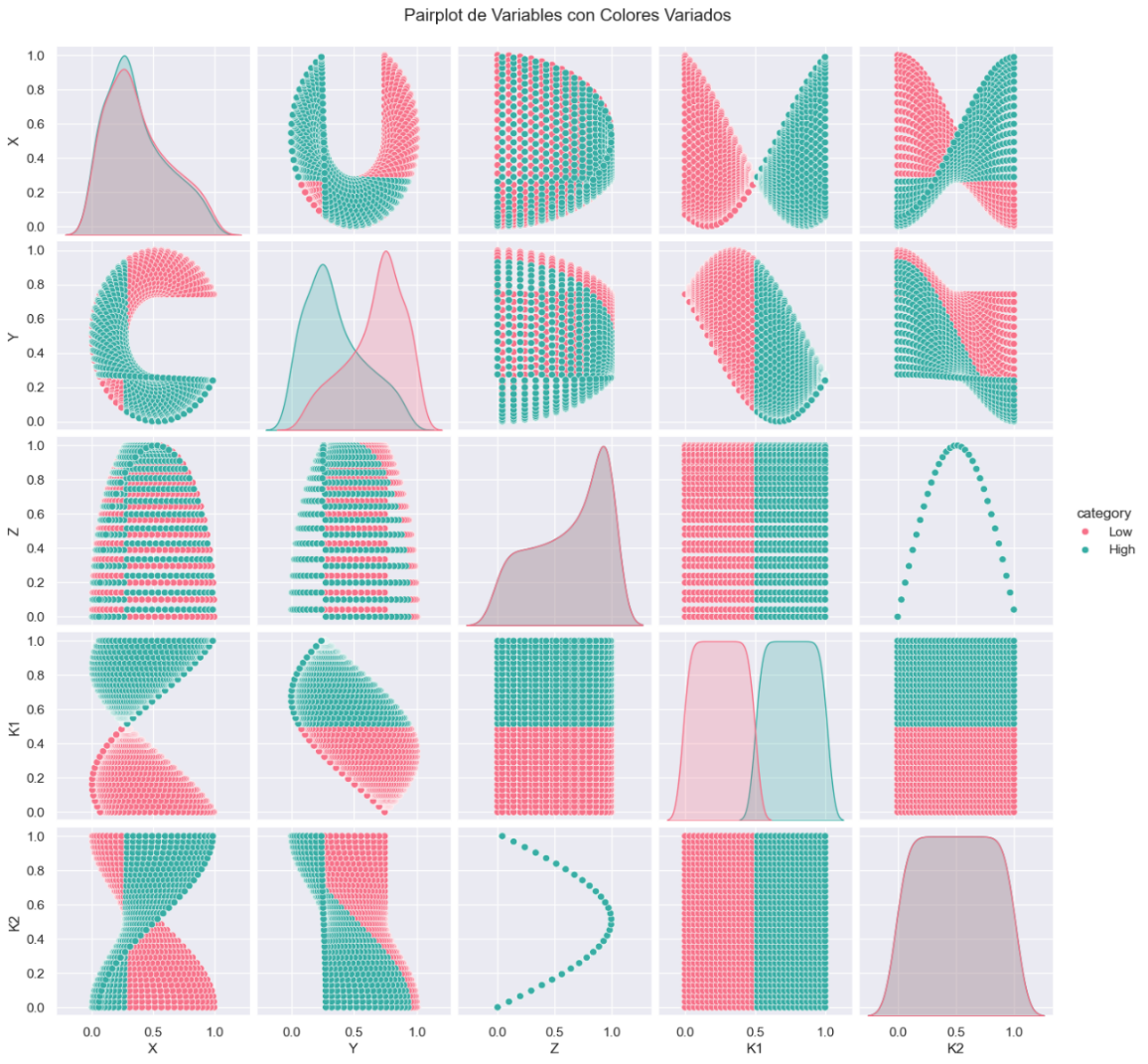
**Preparación de las muestras de entrenamiento y prueba.**

El conjunto de puntos de trayectoria dispuestos consecutivamente del área de trabajo, calculado en el rango de los actuadores angulares [0°, 180°] con un paso de 0.01° y el conjunto correspondiente de puntos intermedios constituye una muestra de entrenamiento para la Red Neuronal Artificial (ANN). La muestra de entrenamiento con este paso se elige debido a los buenos resultados obtenidos durante el trabajo. El tamaño de la muestra de entrenamiento es 99225 datos.

La tabla de datos cuenta de cinco parámetros:

* X, Y, Z (input): Determinan la posición deseada para el efector final en la iteración
* Determina el ángulo obtenido algebraicamente para la unión K2.
* Determina el ángulo deseado para la unión K1.

Se presentan en la Figura 5 las características esenciales de los datos, las cuales han sido obtenidas empezando la librería seaborn. Los datos son pretratados e incluidos dentro de un dataframe empleando la librería pandas para un tratamiento más eficiente de los mismos.



**Figura 5:**PairPlot de Variables empleando la librería seaborn

1. **Red Neuronal Artificial.**

La ANN que brindó los mejores resultados consiste en una capa de entrada de dimensión 4, dos capas ocultas de 16 neuronas de dimensión y una capa de salida de dos neuronas (Ver Tabla 2). La topología de la ANN totalmente conectada se muestra en la Figura 6.

**Tabla 2**

Modelo: “secuencial”

|  |  |  |
| --- | --- | --- |
| Capa (Tipo) | Formato de salida | Numero de Parámetros |
| Entrada | 4 | 4 |
| Densa | 16 | 80 |
| Densa\_1  Densa\_2  Parámetros Totales  Parámetros de entrenamiento | 16  1  369  369 | 272  17 |

* La función de activación ReLU (Rectified Linear Unit) permite introducir no linealidades en el modelo, permitiendo que aprenda representaciones complejas de los datos.
* La función de activación lineal simplemente pasa la entrada a través de la función sin modificaciones.

Estas funciones de activación se utilizaron en el modelo final de la ANN porque proporcionaron los mejores resultados de aprendizaje y son adecuadas para resolver problemas similares, ya que son las más comúnmente utilizadas y bien probadas. Las funciones de activación personalizadas no fueron necesarias y su desarrollo no fue el propósito del estudio.

Para medir la precisión, Scikit-learn no presenta una métrica embebida para su evaluación por lo que se ha implementado la métrica R2 calculada empleando la formula (8). La métrica R², conocida como coeficiente de determinación, es una medida estadística que evalúa qué tan bien un modelo de regresión puede predecir la variable dependiente basada en las variables independientes. Indica la proporción de la varianza en los datos de la variable dependiente que es explicada por el modelo. Esta métrica se eligió para la tarea debido a su simplicidad y claridad.

Donde:

* yi: Valores reales de la variable dependiente.
* i: Valores predichos por el modelo.
* n: es el número de observaciones.

Las pérdidas integradas se eligieron como función de pérdida MSE (error cuadrático medio) calculado utilizando la ecuación (9) a continuación. El MSE mide el promedio de los cuadrados de los errores, es decir, la diferencia entre los valores predichos por el modelo y los valores reales.

Donde:

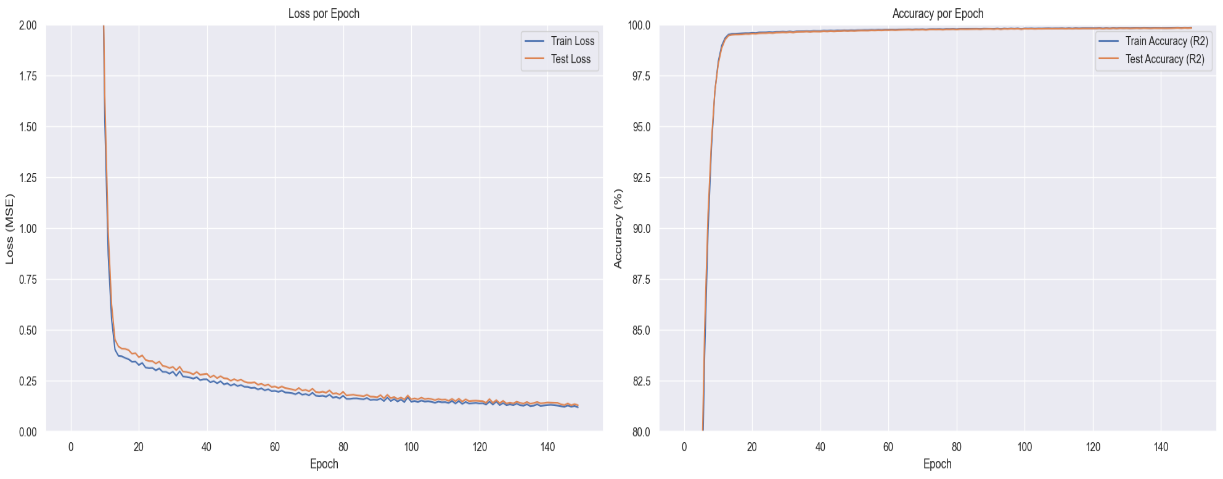
* yi: Valores reales de la variable dependiente.
* i: Valores predichos por el modelo.
* n: es el número de observaciones

El algoritmo de propagación inversa del error se utilizó para el entrenamiento. La función Adam se utilizó como optimizador y el parámetro de tasa de aprendizaje para la ANN fue igual a 0.01. Se pueden encontrar más detalles sobre métricas, funciones de pérdida y optimizadores en la documentación de Scikit-learn.

**3. Resultados y Discusión**

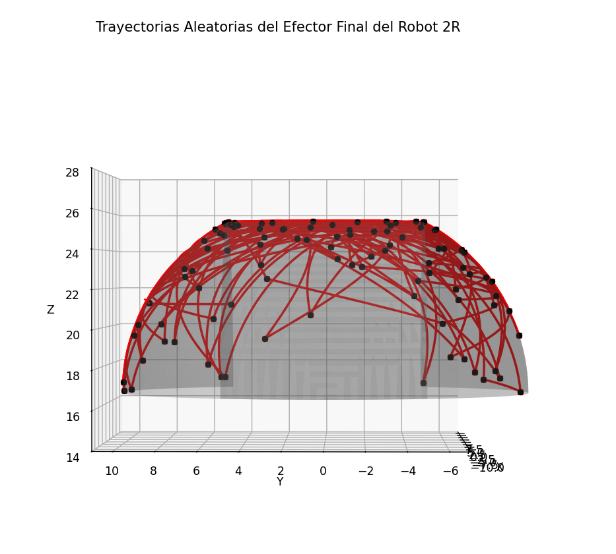
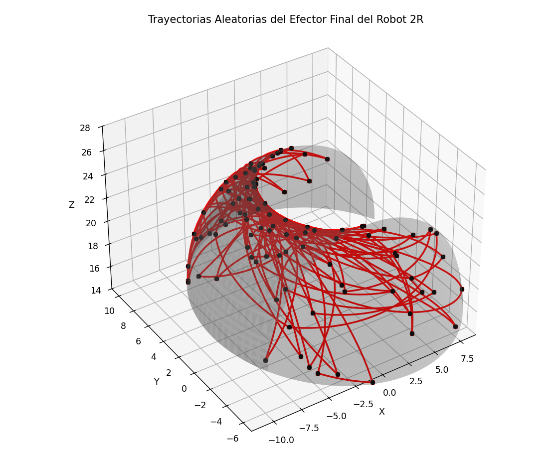
Los resultados del entrenamiento y prueba de la ANN se presentan en esta sección. La evaluación de la ANN entrenada durante la prueba con muestras de prueba (model.predict ()) se realizó con las métricas creadas anteriormente junto con el set de pruebas de test creado a partir de la función df.train\_test\_split().

La red neuronal diseñada fue entrenada con una precisión final igual al 99.73% (ver Figura 7b). Un total de 150 épocas fueron necesarias durante el entrenamiento para lograr la precisión especificada. El valor final de la función de perdida es de 10.3% (ver Figura 7a). El optimizador Adam fue empleado con una ratio de aprendizaje de 0.01.



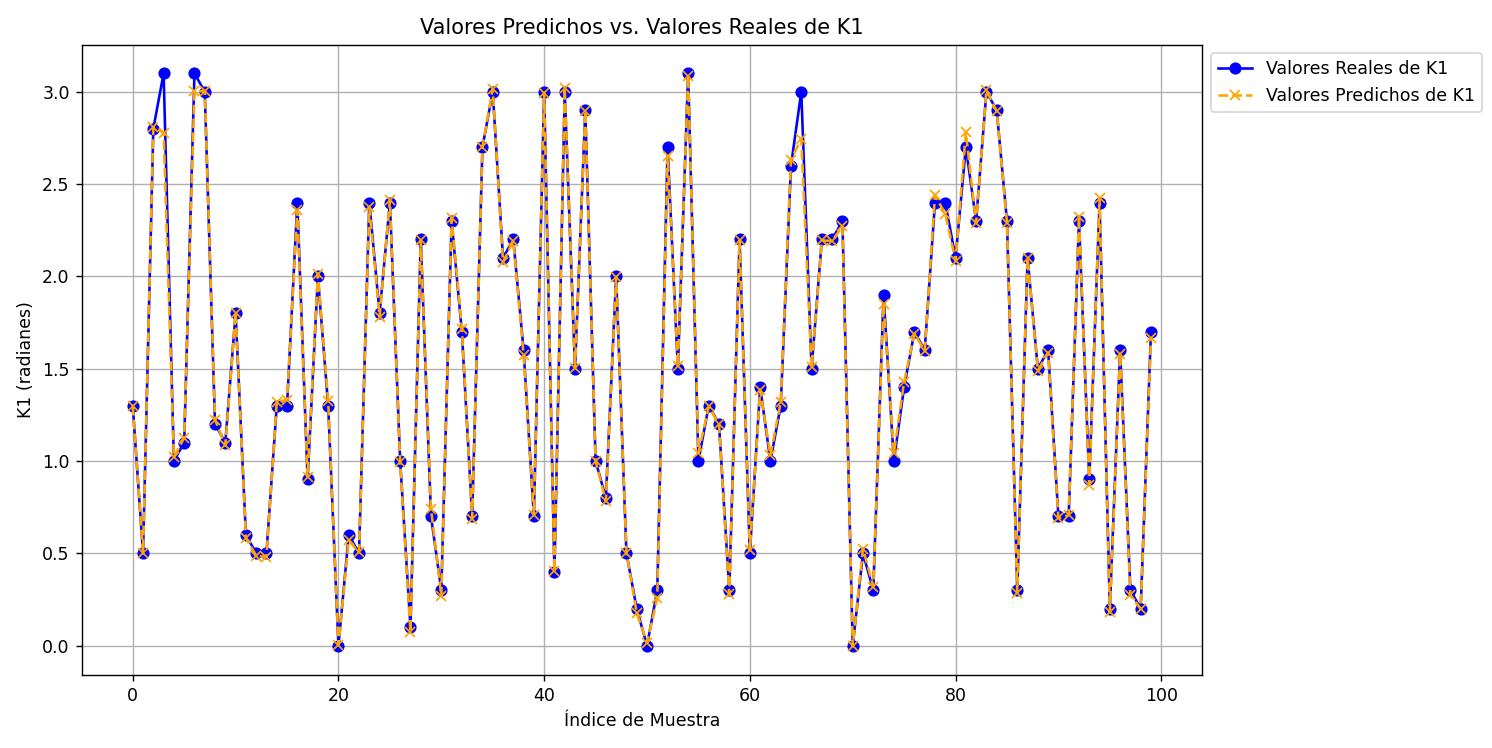
**Figura 7:** a) Pérdida b) Precisión

A través del módulo de Generación de Trayectoria Aleatoriaimplementado en Python con el uso de la cinemática inversa es posible simular las trayectorias del efector final en el espacio y observar si se corresponden con el espacio del trabajo del robot. Las trayectorias generadas pueden ser observadas en la Figura 8.



**Figura 8:** Generación de trayectorias aleatorias en el robot 2R. De izquierda a derecha: Plano XY, perspectiva, Plano YZ.

La ANN tuvo un buen rendimiento en las pruebas y mostró una precisión del 99%. El resultado gráfico de los parámetros de salida se muestra en la Figura 9. La línea azul indica los datos de prueba, mientras que la línea naranja son los valores predichos como resultado del funcionamiento de la ANN. Como se muestra en los gráficos, las líneas casi se repiten entre sí.



**Figura 9:** Resultados de la red neuronal.

Otra propiedad de la ANN propuesta es que siempre produce un resultado que está dentro del espacio de trabajo del robot, incluso cuando los datos de entrada tienen valores fuera del espacio de trabajo. Esto permite proteger los mecanismos del robot y permite incorporar las restricciones asociadas con el diseño del robot en el proceso de aprendizaje y entrenamiento de la ANN.

**4. Conclusiones**

El principal resultado de la investigación es la obtención de un modelo que represente correctamente el posicionamiento del efector final en el espacio de trabajo de un manipulador robótico no-planar 2R contemplando dentro del mismo la cinemática directa e inversa, esta última con uso de redes neuronales artificiales como respuesta a las no-linealidades del sistema. El sistema obtuvo una precisión del 99% en las pruebas realizadas entregando resultados confiables y dentro del espacio de trabajo del robot.Los resultados de este estudio pueden ser aplicados a robots manipuladores con configuraciones similares teniendo en cuenta las dimensiones del mismo y como estas pueden afectar el espacio de trabajo del robot. Estos resultados pueden ser adaptados a robots de líneas de envasado, ensamblaje y manipulación de materiales en la industria.Los resultados del trabajo pueden desarrollarse para otros modelos de robots, así como mejorar las características mecánicas del movimiento, continuar la investigación en la dirección de calcular la trayectoria óptima, las leyes de cambio de los momentos de control en los actuadores, mejorar la comodidad del movimiento y minimizar los costos energéticos.

**Referencias**

1. Tanyrbergenova, K., Mirgalikyzy, T., Mukanova, B., & Posypkin, M. *Using Artificial Neural Network for Solving Inverse Kinematic Task of 2R Planar Robot-Manipulator*. Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, Moscow, Russia*,* 2025.
2. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*, 2016*.* MIT Press. ISBN: 978-026-253-755-1
3. Burkov, A. *The Hundred-Page Machine Learning Book,* 2019*.* ISBN: 978-177-700-547-4
4. James, G., Witten, D., Hastie, T., & Tibshirani, R. *An Introduction to Statistical Learning: with Applications in R*, 2013. Springer. ISBN: 978-107-161-420-4
5. Nielsen, M. *Neural Networks and Deep Learning*, 2025. ISBN:978-331-994-462-3
6. Matloff, N. (2011). *The Art of R Programming: A Tour of Statistical Software Design*. No Starch Press.
7. Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems***.** O'Reilly Media.
8. Russell, S., & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach (3rd ed.).* Prentice Hall.
9. Chollet, F. (2018). *Deep Learning with Python*. Manning Publications.
10. Raschka, S., & Mirjalili, V. (2017). *Python Machine Learning.* Packt Publishing.
11. Corke, Peter (2020). *Robotics Toolbox for MATLAB Release 10.*
12. J. Craig, Jhon.*Introduction to Robotics Mechanics and Control 3rd edition*, 2005. Pearson Education, Inc.
13. Corke, Peter. *Robotics, Vision and Control. Fundamental Algorithms in MATLAB*, 2020. Springer.
14. M. Praveen Kumar, S. D. Ashok, *Artificial neural network based geometric error correction model for enhancing positioning accuracy of a robotic sewing manipulator*, Procedia Computer Science 133 (2018).
15. A. Dzedzickis, J. Subačiūtė-žemaitienė, E. Šutinys, U. Samukaitė-Bubnienė, and V. Bučinskas, *Advanced applications of industrial robotics: New trends and possibilities, Applied Sciences* (Switzerland) 12.1 (2022).
16. J. A. Gonzalez-Aguirre, R. Osorio-Oliveros, K. L. Rodríguez-Hernández, J. Lizárraga-Iturralde, R. Morales Menendez, R. A. Ramírez-Mendoza, M. A. Ramírez-Moreno, J. D. J. Lozoya-Santos, *Service Robots:Trends and Technology*,Appl.Sci.11(2021)10702.
17. L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, L. Farhan, *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*, Journal of Big Data 8.1 (2021). doi:10.1186/s40537-021-00444-8.