

Modelado de un cuello robótico blando mediante aprendizaje automático

Nicole A. Continelli*, Luis F. Nagua, Concepción A. Monje, Carlos Balaguer

Departamento de Ingeniería de Sistemas y Automática, RoboticsLab, Universidad Carlos III de Madrid, C/ Madrid nº126, 28903, Madrid, España.

To cite this article: Continelli, N., Nagua, L., Monje, C.A., Balaguer, C. 2023. Modeling of a soft robotic neck using machine learning techniques. *Revista Iberoamericana de Automática e Informática Industrial* 20, 282-292. <https://doi.org/10.4995/riai.2023.18752>

Resumen

En este trabajo se aborda el problema del modelado de un cuello robótico blando mediante el uso de diferentes arquitecturas de redes neuronales, estudiando la influencia en los resultados del número de capas de cada red y de su correspondiente función de activación. Se emplearán las funciones de activación Tangente Hiperbólica (TANH) y Unidad Lineal Exponencial (ELU). Los modelos obtenidos se compararán con un modelo basado en Perceptrón Multicapa (MLP) de parámetros optimizados, así como con el modelo cinemático analítico del cuello. Los resultados experimentales obtenidos demostrarán la ventaja del empleo de las técnicas de aprendizaje automático para el modelado de sistemas altamente no lineales como el del cuello robótico blando, cuya característica elástica dificulta la formulación de un modelo analítico robusto.

Palabras clave: robótica blanda, curvatura constante (CC), aprendizaje automático, red neuronal, perceptrón multicapa (MLP), función de activación.

Modeling of a soft robotic neck using machine learning techniques

Abstract

In this paper we address the problem of modeling a soft robotic neck by using different neural network architectures, studying the influence on the results of the number of layers of each network and its corresponding activation function. The Tangent Hyperbolic Tangent (TANH) and Exponential Linear Unit (ELU) activation functions are used. The obtained models are compared with a Multi-Layer Perceptron (MLP) with optimized parameters, as well as with the kinematic model of the neck. The experimental results demonstrate the advantage of using machine learning techniques for modeling highly nonlinear systems such as this soft robotic neck, whose elastic characteristics make it difficult to formulate a robust analytical model.

Keywords: soft robotics, constant curvature (CC), machine learning, neural network, multilayer perceptron (MLP), activation function.

1. Introducción

Los modelos matemáticos son una de las herramientas más esenciales de la robótica. Un modelo, proporciona información sobre el comportamiento del sistema y su relación con el entorno. La obtención del modelo matemático de un mecanismo robótico no es exacta, ya que existen incertidumbres difíciles de identificar y abordar, como las relacionadas con la fricción o la holgura electromecánica, entre otras. En el caso de la robótica blanda, la naturaleza de los elementos blandos del robot hace aún más difícil modelar su cinemática, ya que la deformación

de estos elementos afecta a la pose final del robot. Este hecho limita el uso de modelos analíticos y abre la puerta al uso de técnicas de modelado basadas en datos.

Según Siciliano et al. (2008), el control de robots necesita de modelos precisos para lograr un alto grado de desempeño de los mismos, poniendo de lo contrario en peligro al robot y su entorno. Por otro lado, debido a la complejidad de los sistemas robóticos modernos, los modelos analíticos estándar son aproximaciones que no modelan con precisión las no linealidades de dichos sistemas, procedentes, entre otros, de sus sistemas inhe-

*Autor para correspondencia: nicolecontinelli@gmail.com

rentes de actuación, de los rozamientos mecánicos de su estructura u otras fuentes de fricción, o de la deformación de ciertos elementos constitutivos del sistema como cables o eslabones.

Es por ello que actualmente los métodos de aprendizaje de modelos basados en datos se han convertido en herramientas cada vez más interesantes para el modelado de robots. Por ejemplo, las redes neuronales se utilizan ampliamente para la identificación de la cinemática inversa tanto para robots manipuladores redundantes Köker et al. (2014); Becerra et al. (2019); Tran et al. (2011), como no redundantes Jin et al. (2018); Jio-kou Kouabon et al. (2020).

El cálculo de la cinemática inversa se agrava especialmente para el caso de los robots blandos, cuyo modelo es muy dependiente de las características propias de cada robot, de su morfología y del tipo de material blando que lo constituye. Por ejemplo, en Thuruthel et al. (2017) se aborda un enfoque basado en el aprendizaje automático para el desarrollo de modelos dinámicos para un manipulador robótico blando y un método de optimización de trayectorias para el control predictivo del manipulador en el espacio de tareas. En Šegota et al. (2021) se entrena una red neuronal artificial, más concretamente un perceptrón multicapa (MLP) de tipo feed-forward, para poder utilizarla en el cálculo de la cinemática inversa de un manipulador robótico. Y en Wang et al. (2021) se propone un método de optimización que resuelve la cinemática inversa de robots redundantes en base al método de mínimos cuadrados amortiguados (DLS). En cada caso, la cinemática del robot es muy diferente y la solución propuesta está muy enfocada a la problemática concreta de cada robot.

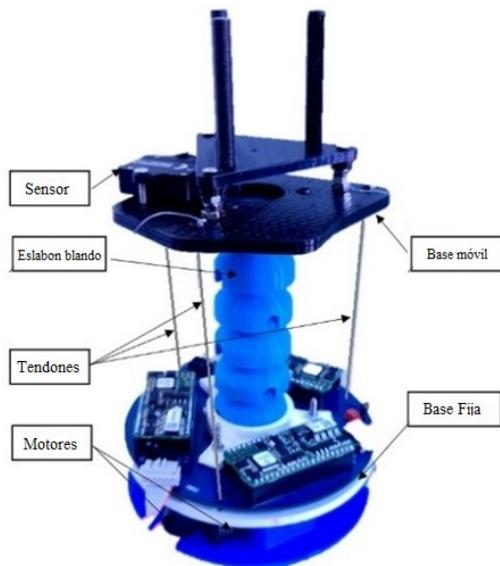


Figura 1: Cuello robótico blando accionado por tendones

Los trabajos de Mena et al. (2020); Nagua et al. (2018b) abordan el modelo de cinemática inversa de un cuello robótico blando, desarrollado por el grupo de investigación RoboticsLab de la Universidad Carlos III de Madrid. El modelo analítico obtenido, se basa en la asunción de la característica de curvatura constante del cuello al aplicar las fuerzas actuadoras que permiten el movimiento del mismo y en despreciar la influencia de la elasticidad del material blando que lo conforma. Estos supuestos merman la exactitud del modelo obtenido, que en determi-

nadas regiones del campo de trabajo del robot, no representa adecuadamente el comportamiento del mismo. Pese a que estos desajustes entre el modelo teórico y el modelo experimental puedan afrontarse desde el punto de vista de un control robusto Relaño et al. (2022); Mena et al. (2020), es conveniente abordar otras técnicas más eficientes de modelado para este tipo de sistemas, concretamente las técnicas basadas en datos, como haremos a continuación.

El resto del documento se organiza como sigue. En la Sección 2 se presenta la descripción de la cinemática inversa del cuello blando. La Sección 3 aborda el aprendizaje automático aplicado a dicho cuello. Los resultados experimentales obtenidos tras la validación de los modelos se discuten en la Sección 4. En la Sección 5 se exponen las principales conclusiones y los trabajos futuros.

2. Cuello Robótico Blando

En la Figura 1 se presenta el cuello robótico blando, que está compuesto por un eslabón blando central, que hace las veces de columna vertebral, y un mecanismo paralelo accionado por cables, que produce la flexión del eslabón central, de forma que la plataforma puede alcanzar la posición deseada tanto en inclinación como en orientación dentro de los límites de operación establecidos por la flexión máxima admisible y el recorrido de los cables. Las partes que forman el cuello son: Base, Plataforma móvil, Eslabón blando, Tendones, Motores, Sensor.

Los cables son conducidos mediante tres actuadores independientes situados en la base, compuestos por un conjunto de driver-motor-reductora con las siguientes características:

- Driver: Technosoft iPOS 3604 MX ; 400 W, 12-50 Volt, 8 Amp
- Motor: Maxon RE 16-118739; graphite brushes, 48Volt, 4 Watt
- Reductora: Maxon 134777 (24 : 1)
- Encoder: Maxon mr201937

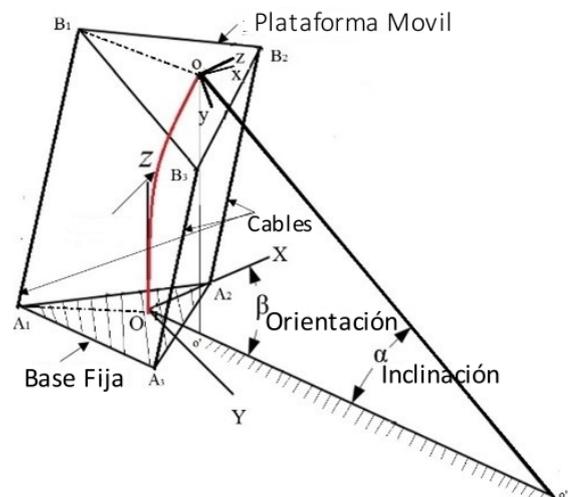


Figura 2: Diagrama esquemático del cuello robótico blando

La Figura 2 presenta un diagrama esquemático del cuello blando, donde $OXYZ$ es el sistema de coordenadas de la base fija, $oxyz$ es el sistema de coordenadas de la plataforma móvil y los tendones conectan los puntos $B_i = (i = 1, 2, 3)$ y $A_i = (i = 1, 2, 3)$ desde la plataforma móvil a la fija, respectivamente, y estos puntos a su vez están equidistantes 120° entre sí.

2.1. Cinemática Inversa

La pose del cuello blando tiene dos componentes independientes: el ángulo α (ángulo de inclinación) y el ángulo β (ángulo de orientación) (ver Figura 2), las cuales determinarán el valor de longitud de cada cable para llegar a dicha pose.

Tal y como se estudia en Nagua et al. (2018a), dichas longitudes vienen dadas por la siguiente ecuación:

$$L_i = \left\| {}^oT_{o'} \vec{oB}_i - \vec{OA}_i \right\| \quad (i = 1, 2, 3). \quad (1)$$

En dicha ecuación:

- ${}^oT_{o'}$ es la matriz de transformación homogénea (ecuación 2) que representa la proyección de $oxyz$ (base móvil) sobre $OXYZ$ (base fija).

$${}^oT_{o'} = \begin{bmatrix} {}^oR_{o'} & P_o \\ 0 & 1 \end{bmatrix}, \quad (2)$$

donde:

- $P_o = \begin{bmatrix} s_o \cos \beta & s_o \sin \beta & t_o \end{bmatrix}^T$ es el vector posición del punto o con respecto al sistema de coordenadas de la base fija y la matriz rotacional ${}^oR_{o'}$ que representa la orientación de la plataforma móvil usando ángulos de Euler ZYZ .

$${}^oR_{o'} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}.$$

- \vec{oB}_i representa los vectores desde el centro de la base móvil hacia los puntos de sujeción de los cables.
- \vec{OA}_i representa los vectores desde el centro de la base fija hacia los puntos de sujeción de los cables.

Los parámetros geométricos definidos para el cuello blando se muestran en la Tabla 1, siendo L_o la longitud del cuello.

Tabla 1: Parámetros geométricos del cuello robótico blando

L_o [m]	$ \vec{OA}_i $ [m]	$ \vec{oB}_i $ [m]
0.107	0.45	0.45

Los modelos para robots blandos y continuos están basados en la forma y estructura del material y el tipo de actuadores que se utilizan para su movimiento. Para los robots continuos se considera una curvatura constante aproximada (CC) en toda la longitud del eslabón blando Webster III and Jones (2010); Reinecke et al. (2016); Jones and Walker (2006); Copaci et al. (2020). En el caso del cuello blando presentado en este trabajo,

Se utiliza la CC que forma la flexión del eslabón blando (ángulo α) entre la base y la plataforma móvil, como se muestra en la Figura 3.

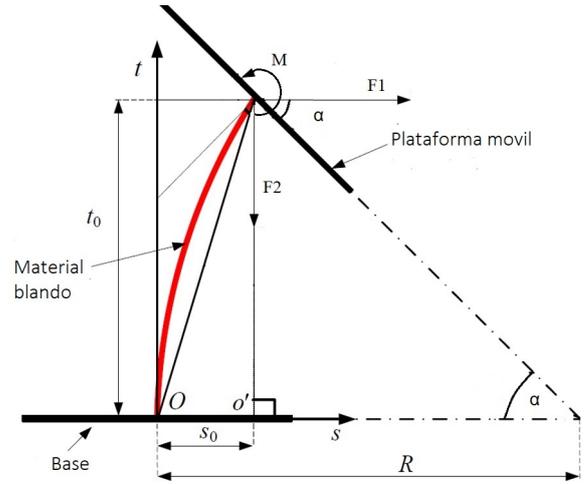


Figura 3: Fuerza y torque del sistema

Dado que s_o y t_o no pueden ser números arbitrarios, se resolverán geoméricamente considerando la flexión del eslabón blando como una CC:

$$R = \frac{L_o}{\alpha} \quad (3)$$

Basándose en geometrías triangulares, se tiene:

$$R = s_o + R \cos \alpha \quad (4)$$

Igualando las ecuaciones (3) y (4), el resultado es:

$$s_o = \frac{L_o(1 - \cos \alpha)}{\alpha} \quad (5)$$

y t_o viene dado por

$$t_o = \frac{L_o \sin \alpha}{\alpha} \quad (6)$$

Finalmente, al resolver la ecuación (1) se determinan las longitudes de cada uno de los tendones L_i que actúan sobre el cuello en función de los ángulos α y β (ecuaciones 7, 8, 9). Las longitudes cambian a su vez gracias a la variación de la posición angular θ_i de cada uno de los tres motores, cuya relación viene determinada por el radio r de la polea acoplada a cada motor: $\theta_i = \frac{L_o - L_i}{r}$.

$$L_1 = \sin\left(\frac{\alpha}{2}\right) \left(\frac{2L_o}{\alpha} - 2a \cos\left(\beta - \frac{\pi}{2}\right) \right) \quad (7)$$

$$L_2 = \sin\left(\frac{\alpha}{2}\right) \left(\frac{2L_o}{\alpha} - 2a \cos\left(\beta - \frac{7\pi}{6}\right) \right) \quad (8)$$

$$L_3 = \sin\left(\frac{\alpha}{2}\right) \left(\frac{2L_o}{\alpha} - 2a \cos\left(\beta - \frac{11\pi}{6}\right) \right) \quad (9)$$

Sin embargo, estos métodos basados en modelos CC tienden a fallar cuando el robot blando es altamente no lineal, no uniforme y está sujeto a incertidumbres externas.

Aunque este problema pueda tratarse mediante un control robusto, nuestro propósito es entrenar diferentes redes neuronales, buscar los hiper-parámetros más óptimos e identificar el modelo inverso desde un enfoque basado en datos reales y comparar los resultados entre sí y con el modelo de cinemática inversa presentado en esta sección Nagua et al. (2018a).

3. Modelos basados en aprendizaje automático

El aprendizaje automático es un campo de búsqueda en la intersección de la estadística, la inteligencia artificial y la informática. También es denominado adicionalmente como el proceso de la analítica de pronóstico o aprendizaje matemático. Según Nori et al. (2007), este aprendizaje es el diseño y el estudio de artefactos de software que utilizan la experiencia pasada para tomar decisiones futuras; es el estudio de programas que aprenden de los datos. Su objetivo fundamental es inducir una regla desconocida a partir de ejemplos de aplicación de la regla. En este trabajo se aplica el aprendizaje supervisado, ya que los algoritmos utilizados se entrenan con una base de datos en la que las entradas y salidas ya están estipuladas, tal y como se describirá en la Sección 4.

3.1. Red neuronal artificial - Perceptrón Multicapa

Uno de los modelos más empleados en el aprendizaje automático es el modelo basado en redes neuronales artificiales, o simplemente redes neuronales. Según Zou et al. (2009), una red neuronal artificial es un tipo de modelo de aprendizaje automático que se inspira en la estructura y el funcionamiento de las redes neuronales biológicas. Se compone de capas de nodos interconectados, llamadas neuronas artificiales, que se activan o desactivan según un conjunto de reglas matemáticas (funciones de activación). Las redes neuronales se caracterizan por el tipo de función de activación utilizada para generar una salida y la arquitectura de la red en términos de cómo están interconectados los nodos. Según Perlich (2010), un perceptrón multicapa (MLP por sus siglas en inglés) es una red neuronal que tiene distintas capas de neuronas. La capa de entrada consiste en las entradas de la red neuronal, y la capa de salida consiste en las salidas finales del MLP. Una capa oculta es cualquier capa intermedia, y un MLP puede tener una o más capas ocultas.

Dos de las tareas del aprendizaje automático supervisado más comunes son la clasificación y la regresión. En el aprendizaje supervisado, Nori et al. (2007) establece que estos modelos deben aprender a predecir valores discretos para las variables "respuesta" a partir de una o más variables explicativas en tareas de clasificación. Es decir, para las nuevas observaciones, el programa debe predecir la categoría, clase o etiqueta más probable. Por otro lado, el modelo debe predecir el valor de una variable continua en problemas de regresión. Esto significa que la solución a este problema es representada por una variable continua, que puede ser flexiblemente determinada por las entradas a nuestro modelo, en lugar de estar restringida a un grupo posible de valores, determinados por clases, como es en el caso de la clasificación. Básicamente, en este trabajo se aborda la tarea de regresión, cuyo objetivo es predecir los valores de la posición angular de los tres motores que actúan sobre el cuello (valores en radianes) para alcanzar los valores de inclinación y orientación dados.

A continuación, vamos a explorar la eficacia del uso de redes neuronales para el modelado del cuello robótico blando descrito en la Sección 2. Estudiaremos la influencia del número de capas de la red y de las distintas funciones de activación elegidas, las cuales se detallan en la siguiente sección.

3.2. Funciones de activación

Las funciones de activación, según Sharma et al. (2017), se utilizan especialmente en las redes neuronales artificiales para transformar una señal de entrada en una señal de salida que, a su vez, sirve de entrada a la siguiente capa de neuronas. En una red neuronal artificial se calcula la suma de productos de las entradas y sus correspondientes pesos y finalmente aplicamos una función de activación para obtener la salida de esa capa en particular, y la suministramos como entrada a la siguiente capa.

Para obtener un mejor resultado con el menor error posible, hay que tener en cuenta una serie de factores, como el número de capas ocultas de una red, los métodos de entrenamiento, el ajuste de los hiper-parámetros y la función de activación. Éste último es uno de los parámetros más importantes a tener en cuenta.

La elección de la función de activación adecuada para un sistema concreto es un proceso minucioso Sharma et al. (2017), ya que depende del caso de los datos y su aplicación. Asimismo, la elección de la función de activación en la capa de la salida tendrá más peso que la que se aplique en las ocultas. Una capa oculta en una red neuronal, es una capa que recibe entradas de otra capa (como otra capa oculta o una capa de entrada) y proporciona salidas a otra capa (como otra capa oculta o una capa de salida). Una capa oculta no contacta directamente los datos de entrada, ni produce salidas para un modelo, al menos en general. Según Brownlee and Mastery (2017), normalmente se utiliza una función de activación no lineal diferenciable en las capas ocultas de una red neuronal. Esto permite que el modelo aprenda funciones más complejas que una red entrenada mediante una función de activación lineal. Para el caso del cuello robótico blando, al tener datos tanto positivos como negativos (giros en sentido horario y antihorario) en las posiciones angulares de los tres motores, se utilizarán funciones de activación cuyos valores incluyan positivos y negativos, para obtener resultados óptimos en las predicciones. Por ello, emplearemos las funciones de activación Tangente Hiperbólica (TANH) y Unidad Lineal Exponencial (ELU), que por otro lado son las más utilizadas en las capas ocultas de las redes neuronales. Éstas se definen a continuación.

3.2.1. Tangente Hiperbólica (TANH)

La tangente hiperbólica, definida en la Ecuación (10), es similar a la función sigmoidea (otra función de activación). Sin embargo, ésta es simétrica alrededor del origen. Esto genera diferentes signos de las salidas de capas anteriores que serán alimentadas como entradas a la siguiente capa, según se describe en Zaki and Meira (2020).

$$f(x) = 2 * \text{sigmoid}(2x) - 1. \quad (10)$$

La función TANH (Figura 4) es continua y diferenciable, y sus valores se encuentran entre -1 y 1. En comparación con la función sigmoidea, el gradiente de la función TANH es más

pronunciado. Es preferible utilizar esta función frente a la función sigmoidea porque tiene gradientes que no están restringidos a variar en una dirección determinada y asimismo, está centrada en 0. Es decir, la función toma cualquier valor real como valores de entrada y de salida en el rango de -1 a 1. Cuanto mayor sea la entrada (más positiva), más cerca estará el valor de salida a 1, mientras que cuanto menor sea la entrada (más negativa), más cerca estará la salida al valor -1. Dado lo anterior, al utilizar esta función de activación se ajustarán mejor a los datos, ya que las posiciones angulares de los motores se encuentran en el mismo rango de valores que dicha función.

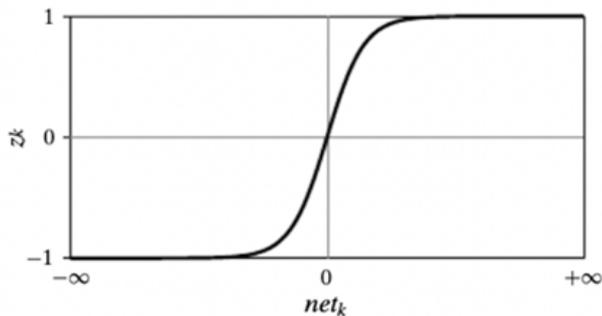


Figura 4: Función TANH

3.2.2. Unidad Lineal Exponencial (ELU)

La Unidad Lineal Exponencial o ELU, definida en la Ecuación (11) (Figura 5), es una variante de la Unidad Lineal Rectificada (RELU). A diferencia de las RELU, las ELU tienen valores negativos, lo que les permite acercar las activaciones de unidades medias a cero, como la normalización por lotes, pero con una menor complejidad computacional. Los cambios medios hacia cero aceleran el aprendizaje, al acercar el gradiente normal al gradiente natural de la unidad, debido a un efecto de cambio de sesgo reducido Clevert et al. (2015). Esta función introduce un parámetro pendiente para los valores negativos de x . Utiliza una curva logarítmica para definir los valores negativos, según se describe en Sharma et al. (2017). Es decir, al utilizar esta función de activación se pueden obtener resultados óptimos en la predicción de los datos, ya que las posiciones angulares de los motores se encuentran en el mismo rango de valores que dicha función y tiende a converger a cero más rápido, produciendo resultados más exactos.

$$\begin{aligned} f(x) &= x \quad (x \geq 0), \\ f(x) &= \alpha(e^x - 1) \quad (x < 0). \end{aligned} \quad (11)$$

A continuación se presentan los resultados obtenidos al modelar el cuello robótico blando mediante distintas configuraciones de MLP y se comparan con el modelo analítico de su cinemática inversa descrito en la Sección 2.1.

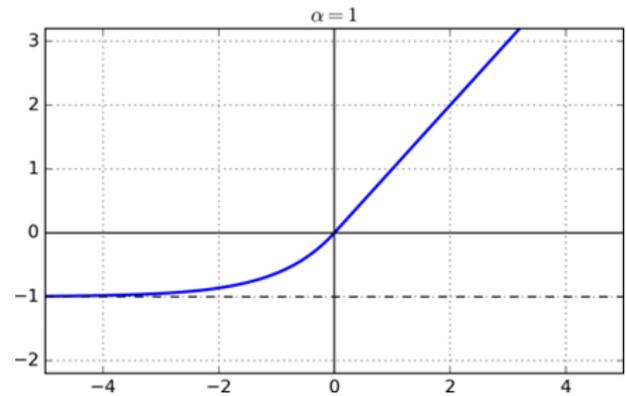


Figura 5: Función ELU

4. Resultados del modelado del cuello robótico blando

En esta sección se describen los datos empleados para el entrenamiento de los distintos MLP, el tipo de entrenamiento llevado a cabo y los resultados experimentales obtenidos en cada caso.

4.1. Extracción de datos

Para la identificación del modelo del cuello robótico blando se utilizan los siguientes datos:

- **Datos de pose:** se utiliza un sensor de Unidad de Medición Inercial (IMU) 3DM-GX5-10 para capturar la inclinación y orientación (poses) del cuello en tiempo real.
- **Tiempo de muestreo:** durante la recolección de datos, el sensor requiere un tiempo de estabilización de unos dos segundos para poder medir con precisión la inclinación y orientación del cuello, como se evidencia en la Figura 7. Por lo tanto, este valor se establece como el tiempo de muestreo durante la toma de datos.
- **Traectoria de pose:** la trayectoria prevista para la adquisición de datos es la siguiente: la inclinación varía de 5° a 35° en pasos de 5°; para cada una de las inclinaciones, la orientación varía de 0° a 360° en pasos de 10°. En total se obtienen 28.800 datos. Para cada par de inclinación y orientación se tiene establecido un tiempo de muestreo de dos segundos, por la estabilización del sensor. Es decir, obtenemos 100 datos por cada pose en el tiempo de muestreo. Esta trayectoria es comandada en la plataforma robótica real, empleando el modelo analítico de cinemática inversa descrito en la sección 2.1 para el cálculo de las posiciones angulares de referencia correspondientes a cada uno de los tres motores.
- **Datos de los encoders:** las posiciones angulares reales alcanzadas por cada uno de los tres motores se miden mediante su respectivo encoder (sensor de posición) en radianes, y se representan en la Figura 6.

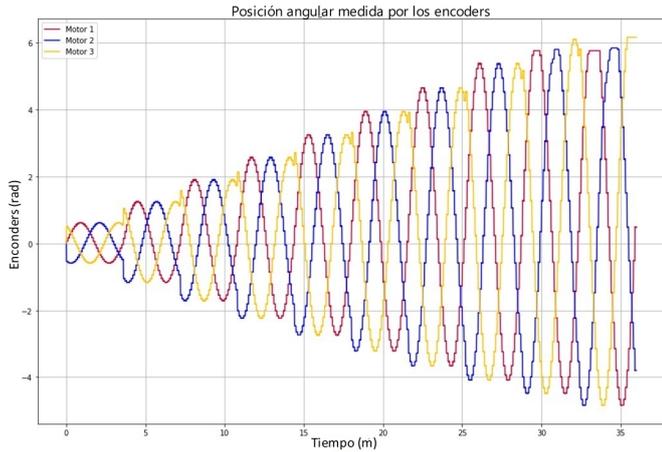


Figura 6: Posiciones angulares de los tres motores del cuello medidas por los encoders (en radianes) para la trayectoria dada

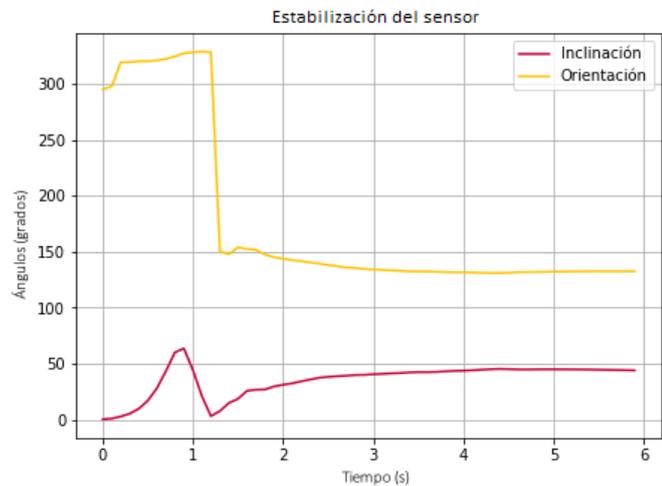


Figura 7: Estabilización del sensor IMU

4.2. Entrenamiento de los modelos

En este trabajo se aborda la tarea de regresión, cuyo objetivo es predecir los valores de las posiciones angulares de los tres motores (que son valores continuos) para los valores de inclinación y orientación dados (en grados). Las entradas y salidas de la red se definen de la siguiente manera:

- **Entrada:** valores de inclinación y orientación extraídos del sensor IMU, dado el movimiento que producen las posiciones angulares de los motores calculadas según el modelo analítico de cinemática inversa del cuello robótico blando.
- **Salida:** posiciones angulares de los tres motores, sensadas por los encoders de dichos motores.
- **División de datos:** la mayoría de los procesos de modelado por aprendizaje automático utilizan una relación 70/30 u 80/20 entre los conjuntos de datos de entrenamiento y de prueba, según Pedregosa et al. (2011); Gholamy et al. (2018). En nuestro caso, el 70 % de los datos empleados se utilizan para el entrenamiento y el 30 % para la validación.

4.3. Resultados experimentales

Para realizar los experimentos se utilizaron tres arquitecturas, las cuales permitirán ver cuál es la óptima a utilizar en el proceso de consolidación de la red neuronal:

- **Simétrica:** la primera capa tendrá cinco neuronas y su número aumentará de cinco en cinco en cada capa hasta llegar a la mitad de las capas ocultas, y a partir de ahí disminuirá de cinco en cinco, tal y como se representa en la Figura 8.

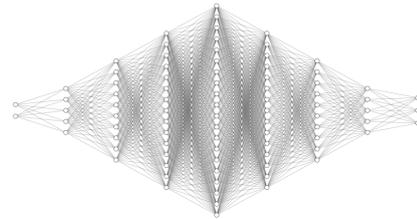


Figura 8: Arquitectura simétrica

- **Constante:** en esta arquitectura todas las capas ocultas tendrán 25 neuronas, como se visualiza en la Figura 9.

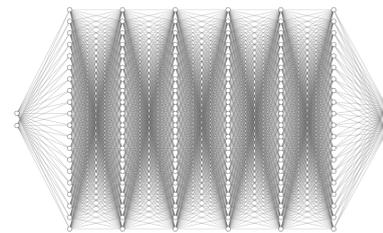


Figura 9: Arquitectura constante

- **Alternada:** para esta arquitectura las capas ocultas tendrán en modo alterno 10 y 20 neuronas, tal y como se representa en la Figura 10.

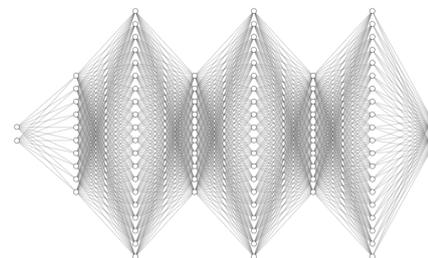


Figura 10: Arquitectura alternada

En la Figura 11 se presentan los nueve modelos obtenidos utilizando los tres tipos de arquitectura expuestos, alternando solamente el número de capas (10, 20 y 30) por cada uno de ellos y seleccionando la función de activación TANH. Todos los modelos presentados están entrenados con 100 épocas (ciclo

completo de los datos entrenados) para minimizar la diferencia entre el resultado real y el teórico.

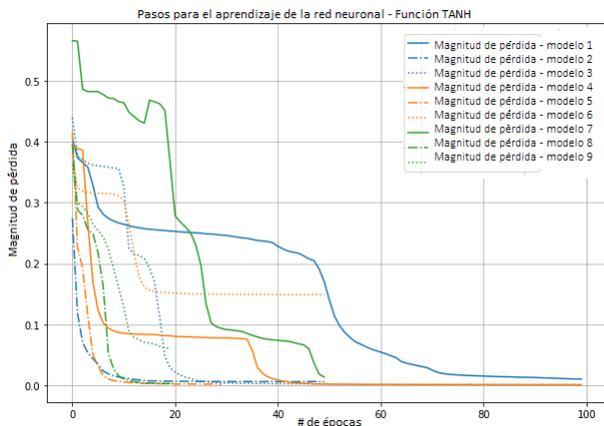


Figura 11: Magnitud de pérdida utilizando la función de activación TANH

En la Figura 12 se presentan los nueve modelos obtenidos en las mismas condiciones que el caso anterior, pero seleccionando la función de activación ELU.

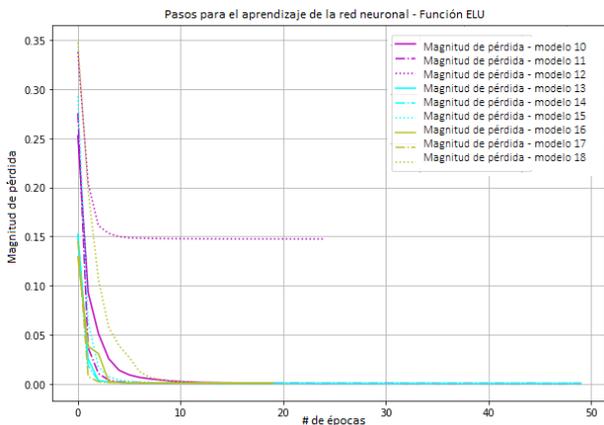


Figura 12: Magnitud de pérdida utilizando la función de activación ELU.

Los resultados obtenidos muestran aspectos diferenciales. Por ejemplo, las arquitecturas tienen una adaptabilidad considerable con la función de activación ELU, ya que ésta necesita menos épocas para obtener una magnitud de pérdida aceptable, mientras que para la función TANH se necesitan más épocas para tener la misma precisión. En otras palabras, podemos decir que en las arquitecturas donde se aplica la función TANH se realiza un mayor esfuerzo al tratar de generalizar la predicción. Se observa también que, cuanto mayor es el número de capas ocultas, menor número de épocas se necesita. Por último, el modelo de arquitectura constante es el que mejor desempeño tiene en comparación con los otros dos modelos planteados.

A raíz de estos resultados de simulación y de las conclusiones alcanzadas Continelli et al. (2022), se han seleccionado dos redes neuronales diferentes para la caracterización del modelo cinemático del cuello blando, cuyas características se definen en la Tabla 2.

Tabla 2: Redes neuronales elegidas

Arquitectura	# de Capas	# de Neuronas	Función de activación
Constante	10	25	Tanh
Constante	10	25	Elu

Para llevar a cabo el estudio comparativo de los modelos cinemáticos identificados con las redes neuronales, se tomará como referencia el modelo de cinemática inversa analítico descrito en la sección 2. Aplicando dicho modelo, se obtiene la terna de posiciones angulares de los tres motores que permite posicionar el cuello de manera que la pose del mismo alcance inclinaciones que varían de 5° a 35° en pasos de 5°, y para cada una de estas inclinaciones, la orientación del cuello varía de 0° a 360° en pasos de 10°, tal y como se describe en la sección 4.1. El espacio de trabajo real resultante, medido a través de la IMU del cuello, se representa en la Figura 13.

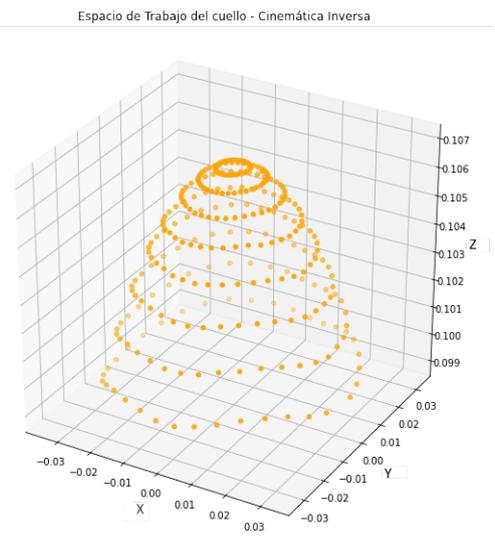


Figura 13: Espacio de trabajo del cuello - Modelo de cinemática inversa analítico

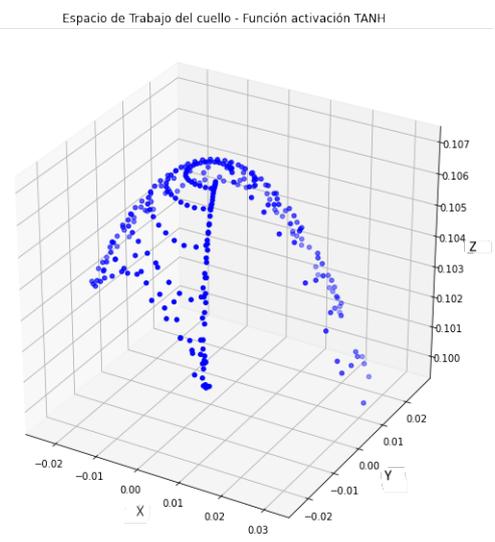


Figura 14: Espacio de trabajo del cuello - Red Constante Tanh

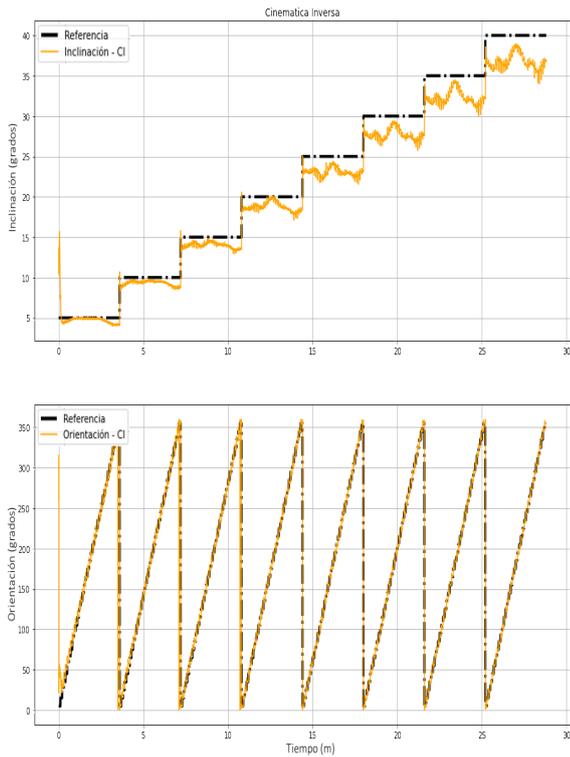


Figura 15: Consignas de posición - Modelo de cinemática inverso

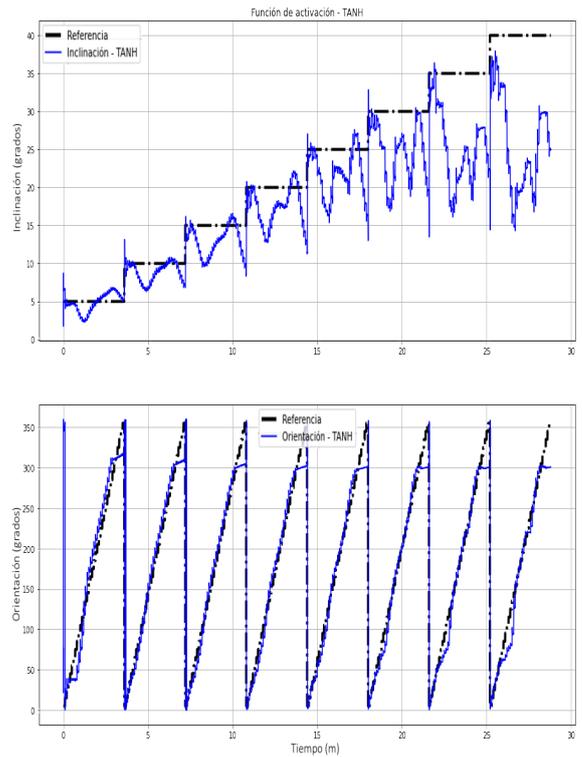


Figura 16: Consignas de posición - Modelo de activación TANH

Para un mejor análisis de los resultados, en la Figura 15 se presenta el comportamiento de la inclinación y orientación reales del cuello (recogidos por la IMU) frente a la referencia deseada. En cuanto a la orientación, se observa un buen ajuste del modelo cinemático analítico. Sin embargo, se observa que la inclinación real alcanzada difiere de la comandada, y la diferencia aumenta conforme aumenta el valor de la inclinación deseada. Analizaremos a continuación el ajuste de los modelos MLP.

Los espacios de trabajo alcanzados por cada una de las dos redes neuronales seleccionadas se representan en la Figura 14, para la red con arquitectura constante y función de activación TANH, y la Figura 18, para la red de arquitectura constante con función de activación ELU. Dichas poses se han medido con la IMU del cuello y se han alcanzado aplicando la terna de posiciones angulares de los tres motores arrojadas por las propias redes. Se han aplicado técnicas de regularización como Drop-Out, que significa desactivar cierto porcentaje de neuronas aleatoriamente (tanto ocultas como visibles) durante el proceso de entrenamiento de una red neuronal, y Early Stopping, que se basa en detener el entrenamiento en el momento que se observe un incremento en el valor del error de validación. Todas estas técnicas basadas en Goodfellow et al. (2016) se aplican para prevenir el sobreajuste de las redes durante el entrenamiento y así poder obtener resultados más óptimos en la predicción de las poses reales del cuello, evitando la memorización.

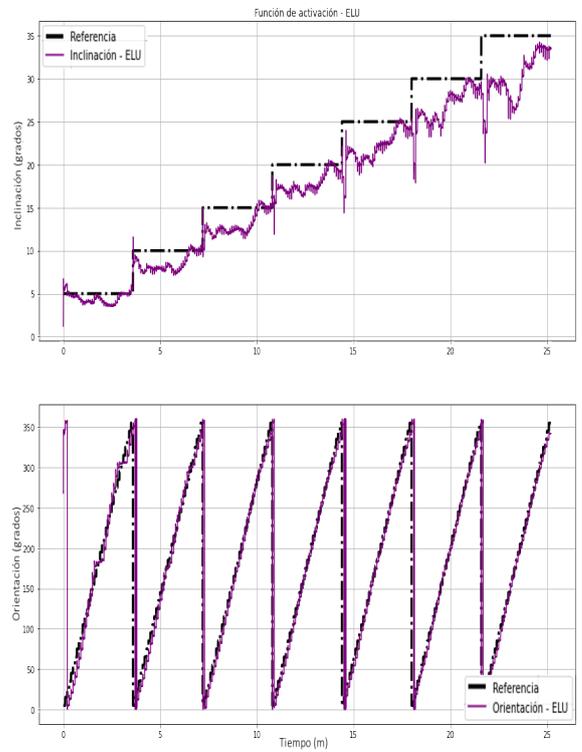


Figura 17: Consignas de posición - Modelo de activación ELU

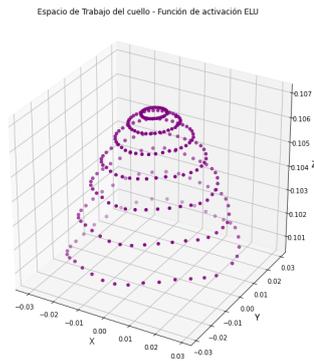


Figura 18: Espacio de trabajo del cuello - Red Constante ELU

Del análisis de los resultados mostrados en la Figura 14, se concluye que la red constante TANH no estima correctamente los valores de las posiciones angulares de los motores, dando lugar a un espacio de trabajo muy alejado del deseado y con errores significativos tanto en la orientación como en la inclinación, como se observa en la Figura 16. Puede que esta red haya tenido problemas en predecir ciertos valores, ya que la función de activación TANH es propensa al desvanecimiento del gradiente Goodfellow et al. (2016). Esto ocurre cuando el gradiente es demasiado pequeño y tiende a 0. A medida que nos movemos hacia atrás durante la retropropagación (manera en la que se entrenan las redes neuronales), el gradiente continúa haciéndose más pequeño, lo que hace que las capas anteriores de la red aprendan más lento que las capas posteriores, llevando esto al sobreajuste.

Los resultados representados en la Figura 18 para la red constante ELU muestran una mayor aproximación al espacio de trabajo de la Figura 13, evidenciando que este modelo estima con mayor precisión los valores de las posiciones angulares de los motores. Esto también se ve reflejado en las trayectorias mostradas en la Figura 17 de ángulos de inclinación y orientación, respectivamente, donde se aprecia una desviación menor que para el caso de la red TANH.

Cabe aclarar que en ninguna de las capas de salida de las redes escogidas se ha aplicado una función de activación. Esto se debe a que, si los resultados de la predicción se encuentran dentro de la escala de los valores reales, no es necesario aplicar dicha función, tal y como se recoge en Goodfellow et al. (2016).

Adicionalmente, se ha implementado otra MLP con optimización de hiper-parámetros como modelo de la cinemática del cuello blando. Según el trabajo de Goodfellow et al. (2014) (creador de la red generativa adversarial), el aumento del número de capas de las redes neuronales y de las neuronas de cada capa tiende a mejorar la precisión general del conjunto de pruebas. Para explorar cómo afectan estos parámetros al proceso de entrenamiento, se ha realizado la optimización de hiper-parámetros utilizando la función GridSearchCV de la biblioteca Scikit-learn Pedregosa et al. (2011), la cual nos permite analizar el rendimiento para diferentes combinaciones de parámetros, como se describe en la Tabla 3.

Tabla 3: Parámetros de optimización de la red MLP

Parámetros	Valores	Descripción
Tamaños de capas ocultas	(5,5), (10,10), (100), (50,50,50), (50,100,50)	Número de capas ocultas y número de neuronas en cada capa.
Función de activación	logística	Se refiere a cómo la suma ponderada de la entrada se transforma en una salida de un nodo en una capa.
Solver	sgd, adam	- 'sgd' se refiere al descenso de gradiente estocástico. - 'adam' se refiere a un optimizador basado en el gradiente estocástico.
Alpha	0.0001, 0.05	- parámetro de penalización (término de regularización).
Tasa de aprendizaje	constante, adaptive	- 'constant' se refiere a un ritmo de aprendizaje constante. - 'adaptive' mantiene la tasa de aprendizaje constante a 'learning_rate_init' mientras la pérdida de entrenamiento siga decreciendo.

Después de utilizar esta función de optimización de parámetros, el mejor rendimiento ha resultado ser el correspondiente a una red neuronal con tres capas ocultas de 50 neuronas cada una y los parámetros mostrados en la Tabla 4.

Tabla 4: Mejor desempeño de la optimización de hiper-parámetros

Tamaño de las capas ocultas	Fun. activación	Solver	Alpha	Tasa de aprendizaje
(50,50,50)	logística	adam	0.0001	constante

La Figura 19 muestra el espacio de trabajo resultante de aplicar las posiciones angulares arrojadas por esta red MLP optimizada. Se observa que esta red, en comparación con los resultados mostrados en las Figuras 14 y 18, es más exacta en la predicción del modelo cinemático inverso que la red constante TANH y la red constante ELU. Asimismo, en la Figura 20 se evidencia una aproximación más precisa que en el caso de los otros modelos en cuanto a los valores de inclinación y orientación deseados, a pesar de que el ajuste para algunos valores de inclinación no es del todo fino.

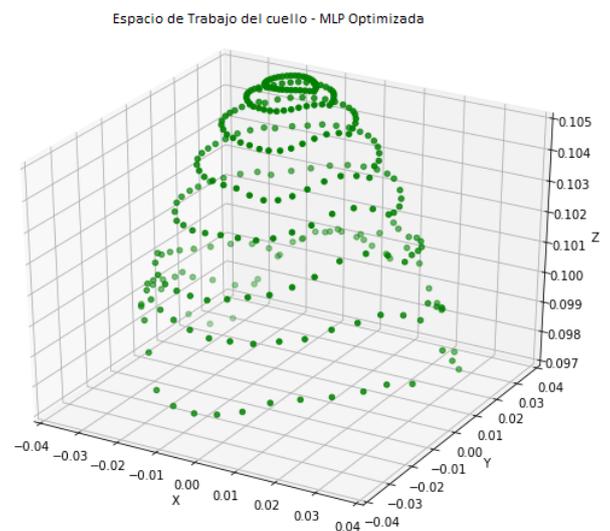


Figura 19: Espacio de trabajo del cuello con MLP optimizada

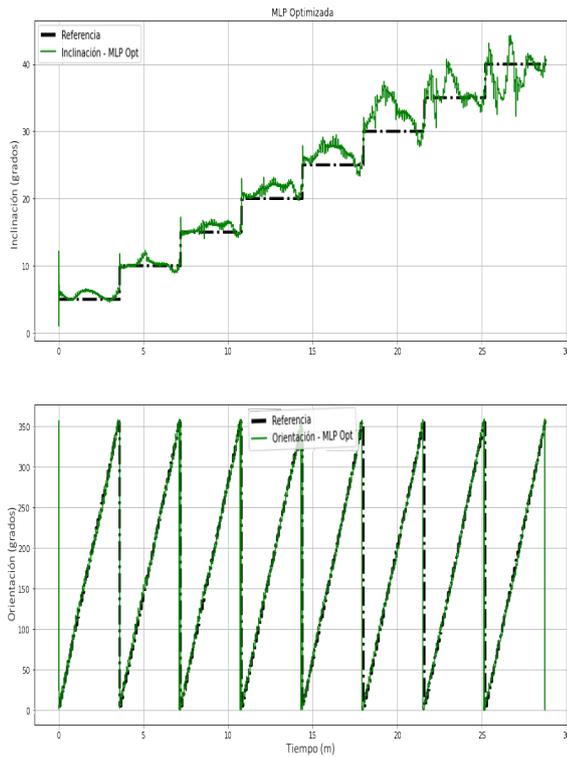


Figura 20: Consignas de posición - Modelo MLP Optimizado

La Figura 21 muestra los errores (en porcentaje) obtenidos al comparar las inclinaciones de cada una de las poses alcanzadas en los cuatro modelos con la trayectoria planteada en el experimento. En cuanto a los errores de orientación, se ha podido evidenciar a lo largo de la discusión de resultados que son despreciables, ya que la diferencia entre los valores teóricos y los reales son casi nulos. Se puede observar que el error menos significativo es el de la MLP optimizada, dado que su valor tiende a cero en la mayoría de los valores de inclinación.

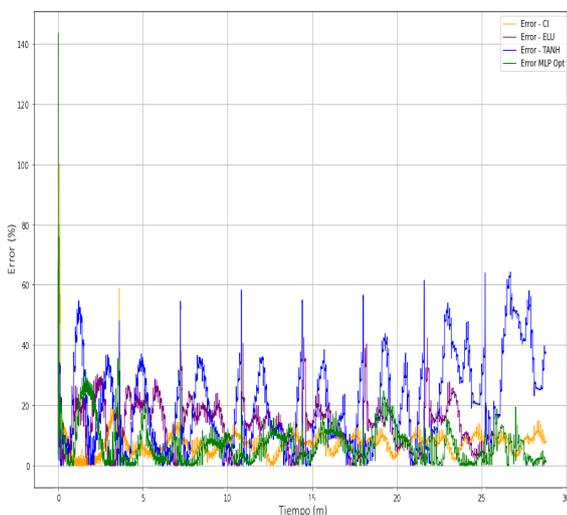


Figura 21: Porcentaje de error en Inclinación

5. Conclusiones

En este trabajo se ha abordado el problema de modelado de un cuello robótico blando mediante técnicas de aprendizaje automático. Concretamente, se han aplicado distintos tipos de redes neuronales y se ha estudiado la influencia de sus diversos parámetros de diseño en la precisión del modelo.

Se ha llevado a cabo un estudio comparativo entre tres tipos de redes: una red de arquitectura constante con función de activación TANH, una red de arquitectura constante con función de activación ELU y una red MLP de parámetros optimizados. Además, se han comparado los resultados de estas redes con los obtenidos del modelo analítico de cinemática inversa del cuello.

Las pruebas reales realizadas sobre el cuello robótico blando, midiendo con la IMU las poses resultantes de aplicar la terna de posiciones angulares obtenidas por las citadas redes, han arrojado resultados claros. Por un lado, se deduce que la función de activación TANH para la identificación del modelo del cuello robótico es deficiente, ya que solo se identifican ciertas regiones de su espacio de trabajo en concordancia con la cinemática inversa. Por otro lado, con la función de activación ELU se obtiene un mejor rendimiento de la red, al identificar con mayor precisión dicho espacio de trabajo. Además de la influencia de la función de activación elegida, se han optimizado otros parámetros como el número de capas ocultas y el número de neuronas de cada capa en una red MLP de hiper-parámetros optimizados mediante la función GridSearchCV de la biblioteca Scikit-learn, siendo esta red la que mejor se ajusta al modelo real de cinemática inversa del cuello.

En futuros trabajos se abordará el problema de la estimación de cargas del cuello robótico blando mediante estas técnicas de aprendizaje automático. Dicha estimación es necesaria desde el punto de vista del control, puesto que permitirá garantizar un rendimiento robusto del cuello durante la ejecución de sus diferentes tareas. Un trabajo previo de los autores Muñoz et al. (2020) propone un método de control robusto basado en controladores de orden fraccionario para hacer frente a las variaciones de la carga. Se espera que la estimación previa de dicha carga mejore significativamente la efectividad del sistema de control. Adicionalmente, se aplicarán estas técnicas para el modelado y control de un nuevo brazo robótico blando que se encuentra actualmente en desarrollo por el grupo de investigación RoboticsLab y que se integrará en el robot humanoide TEO para la mejora de sus tareas de manipulación Hernández-Vicen et al. (2021).

Agradecimientos

Esta investigación ha recibido financiación del proyecto SOFIA: Articulación blanda inteligente con capacidades de reconfiguración y modularidad para plataformas robóticas, con referencia PID2020-13194GB-I00, financiado por el Ministerio de Economía, Industria y Competitividad.

Referencias

Becerra, Y., Arbulu, M., Soto, S., Martínez, F., 2019. A comparison among the denavit-hartenberg, the screw theory, and the iterative methods to solve inverse kinematics for assistant robot arm. In: International Conference on Swarm Intelligence. Springer, pp. 447–457.

- Brownlee, J., Mastery, M. L., 2017. Deep Learning with Python: Develop Deep Learning Models on Theano and TensorFlow Using Keras. Machine Learning Mastery.
URL: <https://books.google.es/books?id=eJw2nQAACAAJ>
- Clevert, D.-A., Unterthiner, T., Hochreiter, S., 2015. Fast and accurate deep network learning by exponential linear units (elus).
URL: <https://arxiv.org/abs/1511.07289>
DOI: 10.48550/ARXIV.1511.07289
- Continelli, N., Nagua, L., Monje, C. A., Balaguer, C., 2022. Identificación de un cuello robótico blando mediante aprendizaje automático. In: Jornadas de Robótica, Educación y Bioingeniería. pp. 124–130.
- Copaci, D., Muñoz, J., González, I., Monje, C. A., Moreno, L., 2020. SMA-driven soft robotic neck: Design, control and validation. IEEE Access 8, 199492–199502.
- Gholamy, A., Kreinovich, V., Kosheleva, O., 2018. Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation. International Journal of Intelligent Technologies and Applied Statistics 11 (2), 105–111.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. Advances in neural information processing systems 27.
- Goodfellow, I. J., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press, Cambridge, MA, USA.
- Hernández-Vicen, J., Martínez, S., Balaguer, C., 2021. Principios básicos para el desarrollo de una aplicación de bi-manipulación de cajas por un robot humanoide. Revista Iberoamericana de Automática e Informática Industrial 18 (2), 129–137.
- Jin, L., Li, S., Yu, J., He, J., 2018. Robot manipulator control using neural networks: A survey. Neurocomputing 285, 23–34.
- Jiokou Kouabon, K., Melingui, A., Lakhal, O., Kom, M., Merzouki, R., 2020. A learning framework to inverse kinematics of redundant manipulators. IFAC-PapersOnLine 53 (2), 9912–9917.
- Jones, B. A., Walker, I. D., 2006. Practical kinematics for real-time implementation of continuum robots. IEEE Transactions on Robotics 22 (6), 1087–1099.
- Köker, R., Çakar, T., Sari, Y., 2014. A neural-network committee machine approach to the inverse kinematics problem solution of robotic manipulators. Engineering with Computers 30 (4), 641–649.
- Mena, L., Monje, C. A., Nagua, L., Muñoz, J., Balaguer, C., 2020. Test bench for evaluation of a soft robotic link. Frontiers in Robotics and AI 7, 27.
- Muñoz, J., Monje, C. A., Nagua, L. F., Balaguer, C., 2020. A graphical tuning method for fractional order controllers based on iso-slope phase curves. ISA transactions 105, 296–307.
- Nagua, L., Monje, C. A., Muñoz, J., Balaguer, C., 2018a. Design and performance validation of a cable-driven soft robotic neck. In: Proc. Actas de las Jornadas Nacionales de Robótica. pp. 1–5.
URL: <http://hdl.handle.net/10016/30567>
- Nagua, L., Muñoz, J., Monje, C. A., Balaguer, C., 2018b. A first approach to a proposal of a soft robotic link acting as a neck. Actas de las XXXIX Jornadas de Automática, Badajoz, 5-7 de Septiembre de 2018.
DOI: 10.17979/spudc.9788497497565.0522
- Nori, F., Jamone, L., Sandini, G., Metta, G., 2007. Accurate control of a human-like tendon-driven neck. In: 2007 7th IEEE-RAS International Conference on Humanoid Robots. IEEE, pp. 371–378.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.
- Perlich, C., 2010. Learning curves in machine learning. In: Encyclopedia of Machine Learning. p. 5.
- Reinecke, J., Deutschmann, B., Fehrenbach, D., 2016. A structurally flexible humanoid spine based on a tendon-driven elastic continuum. In: 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 4714–4721.
- Relaño, C., Muñoz, J., Monje, C. A., Martínez, S., González, D., 2022. Modeling and control of a soft robotic arm based on a fractional order control approach. Fractal and Fractional 7 (1), 8.
- Šegota, S. B., Anđelić, N., Mrzljak, V., Lorencin, I., Kuric, I., Car, Z., 2021. Utilization of multilayer perceptron for determining the inverse kinematics of an industrial robotic manipulator. International Journal of Advanced Robotic Systems 18 (4), 1729881420925283.
- Sharma, S., Sharma, S., Athaiya, A., 2017. Activation functions in neural networks. towards data science 6 (12), 310–316.
- Siciliano, B., Khatib, O., Kröger, T., 2008. Springer handbook of robotics. Vol. 200. Springer.
- Thuruthel, T. G., Falotico, E., Renda, F., Laschi, C., 2017. Learning dynamic models for open loop predictive control of soft robotic manipulators. Bioinspiration & biomimetics 12 (6), 066003.
- Tran, L., Zhang, Z., Yeo, S., Sun, Y., Yang, G., 2011. Control of a cable-driven 2-dof joint module with a flexible backbone. In: 2011 IEEE Conference on Sustainable Utilization and Development in Engineering and Technology (Student). IEEE, pp. 150–155.
- Wang, X., Liu, X., Chen, L., Hu, H., 2021. Deep-learning damped least squares method for inverse kinematics of redundant robots. Measurement 171, 108821.
- Webster III, R. J., Jones, B. A., 2010. Design and kinematic modeling of constant curvature continuum robots: A review. The International Journal of Robotics Research 29 (13), 1661–1683.
- Zaki, M. J., Meira, Jr, W., 2020. Data Mining and Machine Learning: Fundamental Concepts and Algorithms, 2nd Edition. Cambridge University Press. DOI: 10.1017/9781108564175
- Zou, J., Han, Y., So, S.-S., 2009. Overview of artificial neural networks. Artificial neural networks: methods and applications, 14–22.