



Resolución de la Elástica mediante una red neuronal ODE para cinemática de mecanismos planos flexibles

O. Altuzarra¹, J.L. Ruiz Erezuma¹, V. Petuya¹, M. Urizar¹

¹ Dpto. de Ingeniería Mecánica. Universidad del País Vasco UPV/EHU. Escuela de Ingeniería de Bilbao
oscar.altuzarra@ehu.eus, jose Luis.ruiz@idom.com, victor.petuya@ehu.eu, monica.urizar@ehu.eus

Los mecanismos planos formados de elementos flexibles o continuos, en configuraciones de cadena cerrada, permiten diseñar sistemas mecánicos para manipulación con un grado de conformidad que los hacen atractivos en aplicaciones en entornos con personas por su nivel de seguridad ante impactos imprevistos. Además, esta flexibilidad puede ser también útil en aplicaciones de montaje automático de precisión o incluso como sistemas de actuación y sensorización háptica.

Este tipo de mecanismo conlleva una mayor dificultad de análisis de la cinemática y condiciona en gran medida el desarrollo de los subsiguientes pasos de simulación y control. En concreto el reto está en la resolución numérica y la caracterización de los parámetros físicos reales, que forman parte de las ecuaciones diferenciales de los modelos matemáticos empleados. Las redes neuronales artificiales pueden ser una herramienta útil para la modelización de algunos de estos problemas.

El ámbito de aplicación de las redes neuronales es muy amplio, así como la naturaleza de estas. En concreto, cuando el problema a analizar tiene como base un sistema de ecuaciones diferenciales existe un tipo concreto de redes neuronales denominado genéricamente NODE. Este tipo de redes es aplicado en el modelizado de sistemas dinámicos como forma de mejorar las técnicas de estimación de parámetros dinámicos.

Este trabajo trata de probar el uso de redes neuronales NODE en otro tipo de problemas caracterizados por ecuaciones diferenciales cuya variable no es el tiempo sino una coordenada espacial. En concreto nos proponemos abordar algunos problemas de resolución de la clásica curva elástica de una barra esbelta bajo diferentes condiciones de contorno, como paso previo a su implementación en la resolución de problemas de posición en mecanismos planos paralelos formados de elementos flexibles. La variable espacial es la longitud de arco sobre la curva elástica, y la ecuación diferencial está definida en función del material y sección de la barra. El objetivo es resolver la forma deformada de la elástica de una forma más rápida que con los clásicos métodos de integración directa, o la aproximación mediante integrales elípticas.

1. Introducción

Los mecanismos paralelos flexibles considerados son sistemas mecánicos en los que un elemento terminal rígido es unido al elemento fijo a través de elementos (generalmente barras) muy flexibles que son actuadas en ese extremo mediante un cambio de su posición y/o orientación. Precisamente adquieren su movilidad gracias a grandes deformaciones de las barras esbeltas, que se producen en el rango de la no-linealidad y que generalmente requieren de materiales super-elásticos como el Nitinol [1]. Su cinemática deja de ser un problema geométrico de compatibilidad de restricciones que se explicita como la resolución de un sistema de ecuaciones algebraicas no lineales, y pasa a ser un problema que involucra condiciones geométricas, modelos de grandes deformaciones y de equilibrio de fuerzas. La modelización de las grandes deformaciones se puede plantear con diferentes métodos [2], entre los que destacan por su precisión los basados en modelos continuos que emplean la Teoría de barras de Cosserat [3]. En cualquier caso, esos modelos suponen la introducción de un sistema de ecuaciones diferenciales ordinarias (EDO) cuya resolución es normalmente numérica. Además, los modelos planteados tienen grandes problemas de identificación de parámetros reales y en consecuencia de precisión, porque en muchas ocasiones no son capaces de modelar de forma efectiva efectos dinámicos, perturbaciones de estabilidad o fricción [4].

Para paliar esas deficiencias, recientemente se ha iniciado una línea de investigación en la generación de modelos híbridos que complementan a los modelos analíticos con redes neuronales para producir modelos más precisos [5]. Las redes neuronales artificiales pueden ser una herramienta muy potente para generar modelos por regresión cuando existe abundancia de datos del problema que puedan ser usados para su entrenamiento [6]. Cuando esos datos provienen de un modelo matemático clásico se busca acelerar los tiempos de resolución sustituyendo el uso de esos modelos clásicos por una red neuronal que genere las soluciones por inferencia. Cuando se tienen datos experimentales es incluso posible crear modelos de fenómenos físicos cuyas leyes físicas no se conozcan, o si se conocen es posible obtener los parámetros físicos que las caracterizan.

En el análisis cinemático de los mecanismos paralelos flexibles este grupo desarrolló modelos de resolución del problema directo e inverso que requieren un cálculo numérico intensivo [7]. El reto ahora es conseguir métodos de resolución más eficientes, precisos, y adecuados para una identificación de parámetros físicos con experimentos. Para ello nos planteamos usar las redes neuronales artificiales en primer lugar con datos obtenidos de modelos clásicos buscando eficiencia de cálculo, y posteriormente pasar a redes entrenadas con datos experimentales. Con este doble objetivo se inicia la investigación, y en este trabajo se introducen los primeros pasos buscando el tipo de red neuronal más adecuado para la tarea. Para ello se aborda el problema clásico de la elástica, que es parte constitutiva del problema a resolver en mecanismos paralelos flexibles planos, que en el fondo son un conjunto de elásticas ensambladas bajo diferentes condiciones de contorno interrelacionadas. Hasta donde este grupo conoce, el único trabajo sobre la elástica abordado desde el empleo de redes neuronales artificiales [8] realiza una aproximación por regresión utilizando redes neuronales artificiales Multi Layer Perceptron (MLP). Este grupo se plantea la posibilidad de explotar la naturaleza del problema de la elástica expresado como integración de un sistema de ecuaciones diferenciales para resolverla usando redes neuronales Neural Ordinary Differential Equations (NODEs) [9]. Este enfoque es el generalmente usado en la resolución de problemas dinámicos y de control mediante redes neuronales y viene inspirado por la analogía clásica entre el problema dinámico del péndulo simple y el problema de la elástica.

2. Analogía clásica de la Elástica y la dinámica del Péndulo Simple

La analogía matemática entre el péndulo simple y la elástica es bien conocida [10]. Considérense los modelos físicos de la Figura 1 del péndulo simple y de la elástica de una barra empotrada-libre sometida a una fuerza R de inclinación ψ en su extremo.

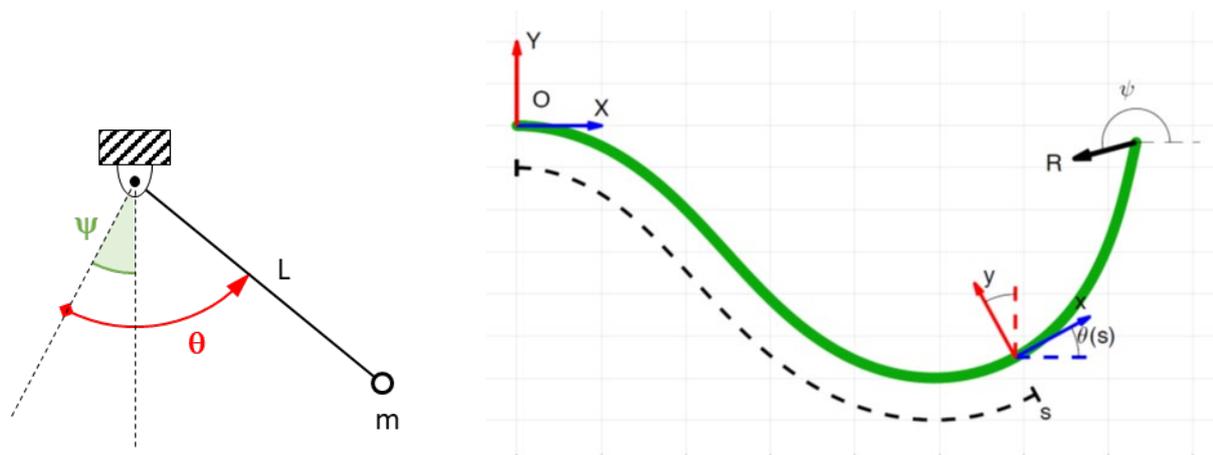


Figura 1: Modelo de Péndulo simple, y Elástica de una barra empotrada-libre.

En el caso del péndulo simple, de longitud L y masa en su extremo m , si expresamos la inclinación variable del péndulo con la variable θ que tiene por origen una posición angular desde la vertical dada por ψ (Fig. 1), su ecuación dinámica es:

$$mL^2\ddot{\theta} + mgL\text{sen}(\theta - \psi) = 0 \quad (1)$$

que para su integración en el tiempo se puede replantear como un sistema de ecuaciones diferenciales de primer orden:

$$\frac{d\theta}{dt} = \omega; \quad \frac{d\omega}{dt} = -\frac{g}{L}\text{sen}(\theta - \psi) \quad (2)$$

Su resolución numérica es un problema de valor inicial. Por ejemplo, si el péndulo se libera a la acción gravitatoria sin velocidad angular inicial $\omega=0$ y desde $\theta=\pi$ considerando el ángulo de referencia $\psi=0$ su movimiento en el tiempo de medio ciclo y su diagrama de fase serían los mostrados en la Figura 2.

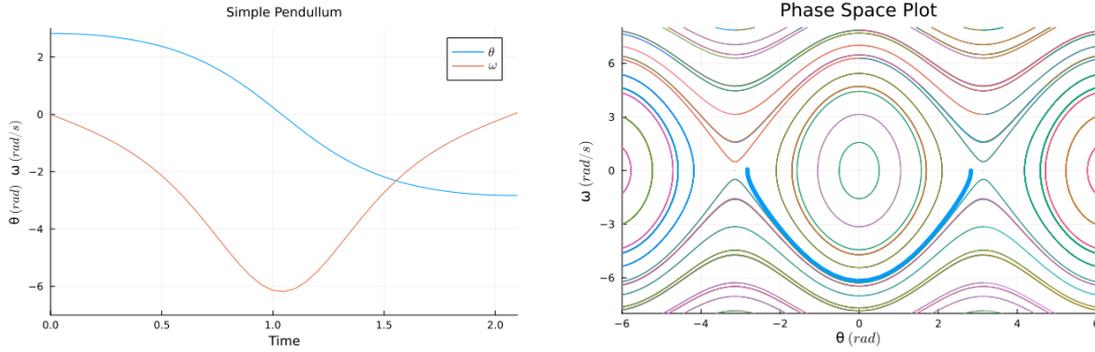


Figura 2: Movimiento del Péndulo simple, y Diagrama de fase.

En el caso de la elástica, se tiene una barra deformable de longitud L , sección A , momento de área de la sección transversal I , y módulo elástico E . Los puntos de la elástica vienen dados por coordenadas x e y en el plano de deformación, a lo largo de la variable de longitud de arco s , de tal forma que la sección transversal en cada posición sobre la curva deformada tiene una pendiente θ y una curvatura κ . Las fuerzas que actúan sobre la barra generan unas fuerzas internas de componentes n_x y n_y , y un momento flector m_z que varían a lo largo de la barra. Aplicando la teoría de Cosserat, en ausencia de cargas distribuidas sobre la barra, la deformación responde a este sistema de ecuaciones diferenciales:

$$\begin{aligned} \frac{dx}{ds} &= \cos\theta \\ \frac{dy}{ds} &= \text{sen}\theta \\ \frac{d\theta}{ds} &= \frac{m_z}{EI} \\ \frac{dn_x}{ds} &= 0 \\ \frac{dn_y}{ds} &= 0 \\ \frac{dm_z}{ds} &= n_x \text{sen}\theta - n_y \cos\theta \end{aligned} \quad (3)$$

de donde se obtienen:

$$\frac{d^2\theta}{ds^2} = \frac{R}{EI}\text{sen}(\theta - \psi); \quad \frac{dx}{ds} = \cos\theta; \quad \frac{dy}{ds} = \text{sen}\theta \quad (4)$$

La primera de las ecuaciones diferenciales es análoga a la ecuación de la dinámica del péndulo simple, y de ahí la analogía clásica de ambos problemas. De hecho, la conversión de esa ecuación diferencial de segundo orden en un sistema de ecuaciones diferenciales de primer orden introduciendo la curvatura genera las ecuaciones análogas a (2):

$$\frac{d\theta}{ds} = \kappa; \quad \frac{d\kappa}{ds} = \frac{R}{EI} \text{sen}(\theta - \psi) \quad (5)$$

Por ejemplo, ante una carga vertical en el extremo de la barra empotrada libre se obtendría la deformación de la elástica y la variación de pendiente y curvatura a lo largo de la longitud de la barra como se muestra en la Figura 3.

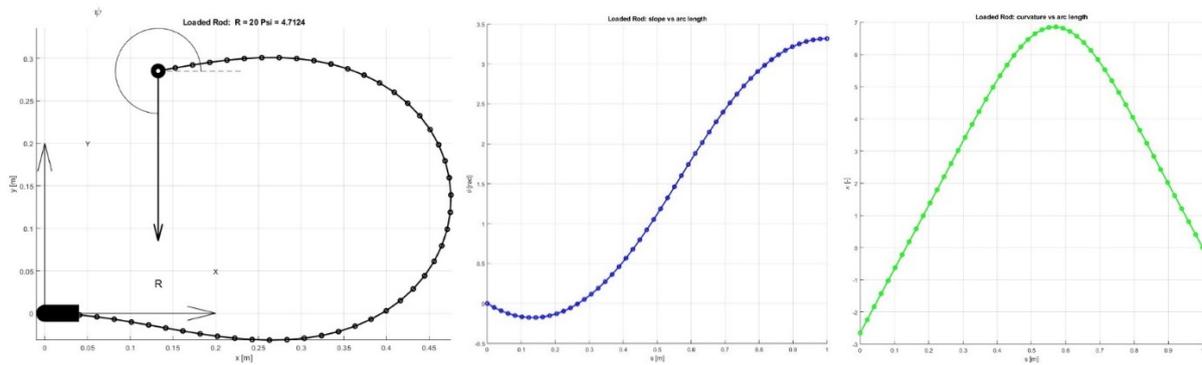


Figura 3: Modelo de barra deformada, su pendiente y curvatura.

Y esta información se puede mostrar en un diagrama de fase como en la Figura 4.

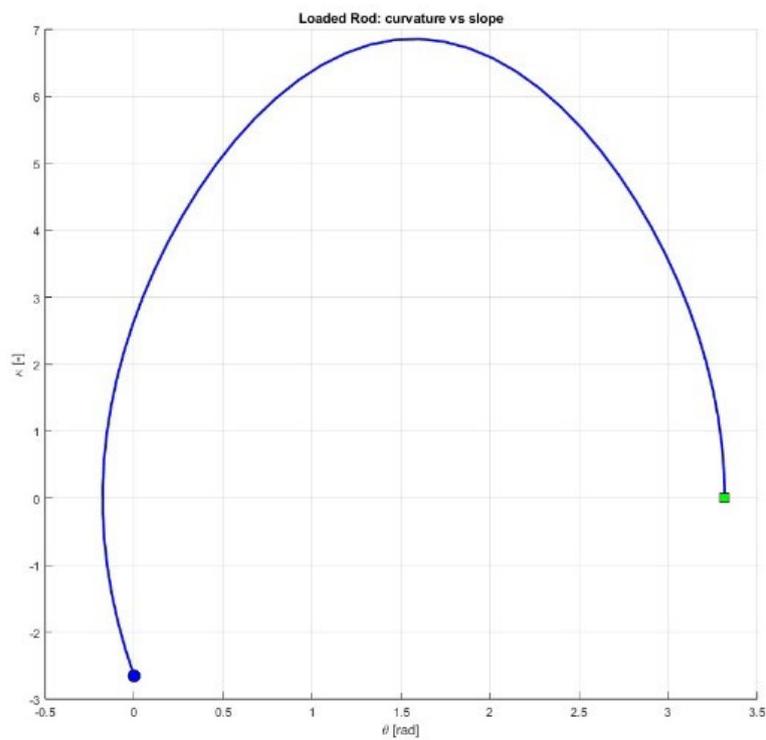


Figura 4: Diagrama de Fase de la deformada elástica.

Aunque existe la analogía matemática en las ecuaciones que caracterizan ambos problemas, hay que tener en cuenta que el péndulo simple representa un Problema de Valor Inicial, puesto que la solución única queda determinada por sus condiciones iniciales, posición angular θ y velocidad angular ω ; mientras que la elástica es un Problema de Valor de Contorno. Así, el ejemplo mostrado de la elástica requiere tantear valores iniciales de la curvatura $\kappa(s=0)$ y escoger aquél que hace que la curvatura en el extremo libre sea nula $\kappa(s=L)=0$ (ver Figura 5), lo cual hace que el problema a resolver sea computacionalmente más costoso.

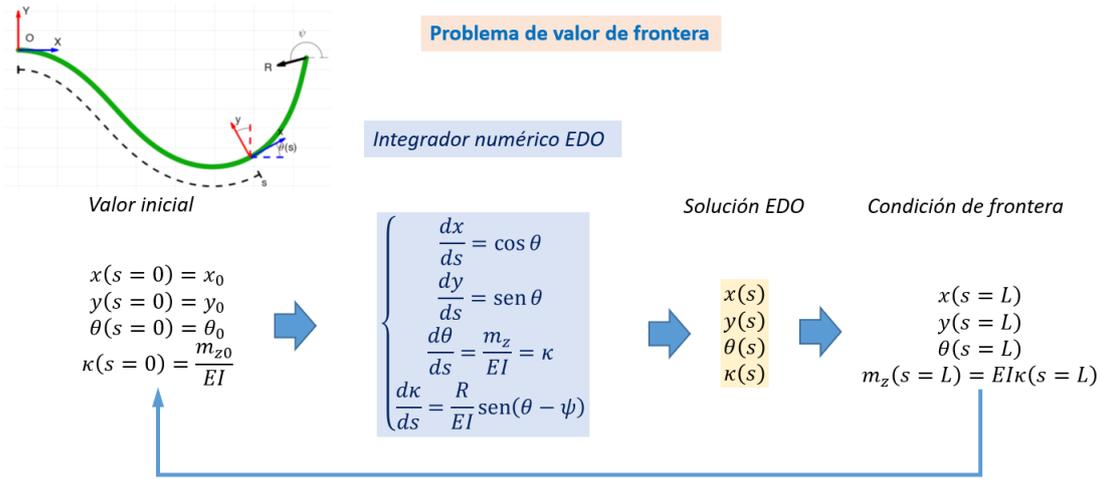


Figura 5: Modelo de barra de Kirchhoff, extremos fijo-rótula, resuelto empleando integración numérica de valores de frontera.

Esa es la motivación para el uso de redes neuronales artificiales, y la que nos lleva a explorar las técnicas usadas en la aplicación de esa herramienta a la resolución del problema dinámico del péndulo [11].

3. Obtención de datos de entrenamiento para la Elástica

Como se expuesto anteriormente, en este trabajo se van a explorar alternativas de redes neuronales para el cálculo de la elástica. Esto requiere obtener datos de entrenamiento para esas redes, es decir soluciones de la elástica precalculadas que sirvan para luego inferir nuevas soluciones con condiciones diferentes. Se van a tomar datos obtenidos mediante el cálculo clásico mediante integración directa y resolución del problema de contorno expuesto en el apartado anterior para evitar introducir la variabilidad estadística y las incertidumbres de medida que acompañan los datos experimentales, y poder así valorar mejor los errores inherentes a la red neuronal analizada. Además, se va a elegir un caso de elástica especialmente útil para las aplicaciones posteriores a mecanismos flexibles. Se va a considerar una barra empotrada en el primer extremo y articulada en el segundo extremo, y se van a mantener ambos puntos fijos, variando la orientación del empotramiento. Así se analizará la capacidad de inferir la elástica en esas situaciones. El esquema de resolución empleado es el de la figura 6, donde en el extremo inicial x_0 , y_0 y θ_0 son conocidos y se deben inicializar valores de tanteo para n_x , n_y y m_z ; tras integrar mediante Runge-Kutta [12] se obtienen esas variables a lo largo de la longitud de arco s ; y debe verificarse que en el extremo final las coordenadas x_L , y_L se corresponden con la posición de articulación, y el momento flector m_z es nulo. De no cumplirse, se debe ejecutar un proceso de minimización del residuo de esas condiciones hasta encontrar los valores adecuados de n_x , n_y y m_z en $s=0$.

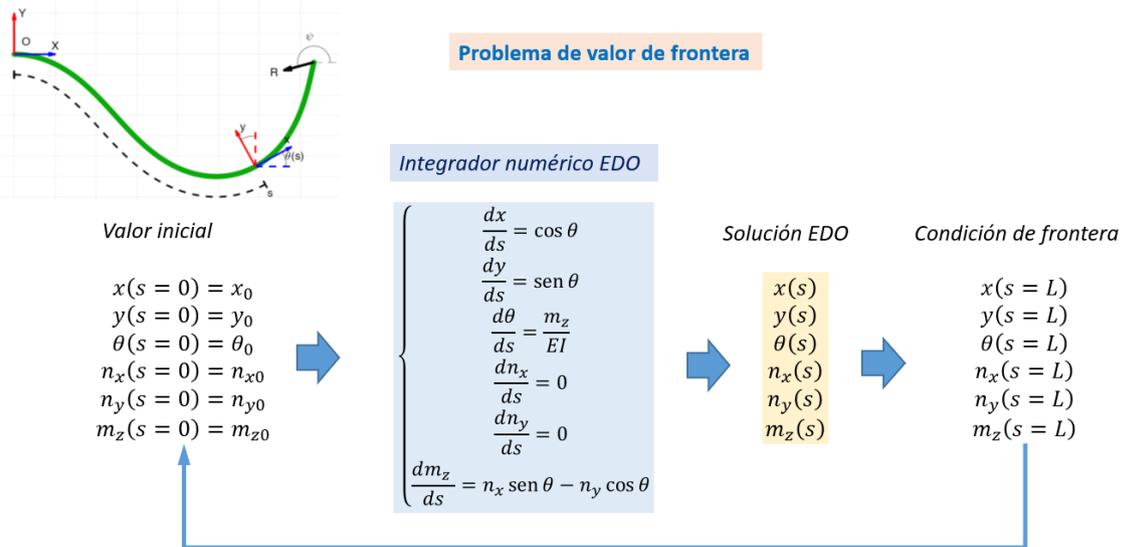


Figura 6: Esquema de resolución de la elástica empleando integración numérica de valores de frontera.

Para la orientación del empotramiento horizontal los resultados obtenidos a lo largo de 50 puntos de la barra son los mostrados en la Figura 7.

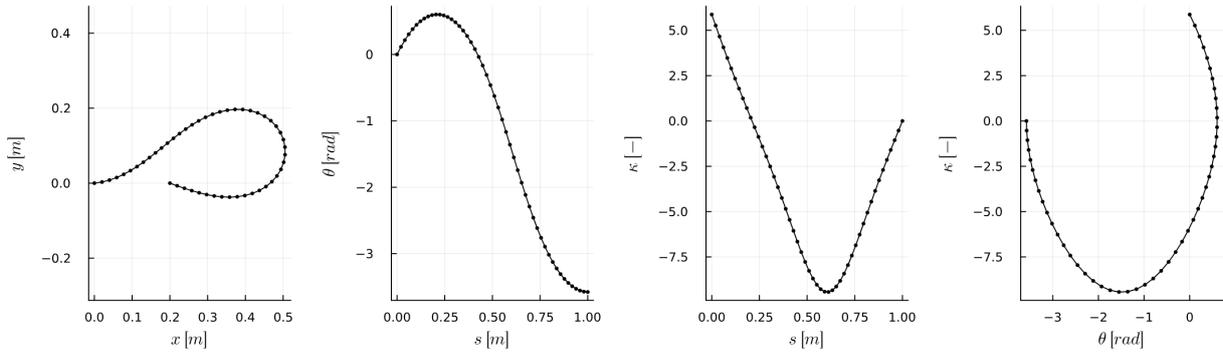


Figura 7: Solución barra empotrada-articulada.

Ese cálculo se puede extender a una serie de orientaciones del empotramiento en una rotación completa, de tal forma que sirvan de conjunto de datos de entrenamiento. En la Figura 8 se muestra el set de entrenamiento utilizado consistente en 72 curvas equiespaciadas en 360° del empotramiento y 50 nodos equidistantes por cada curva elástica.

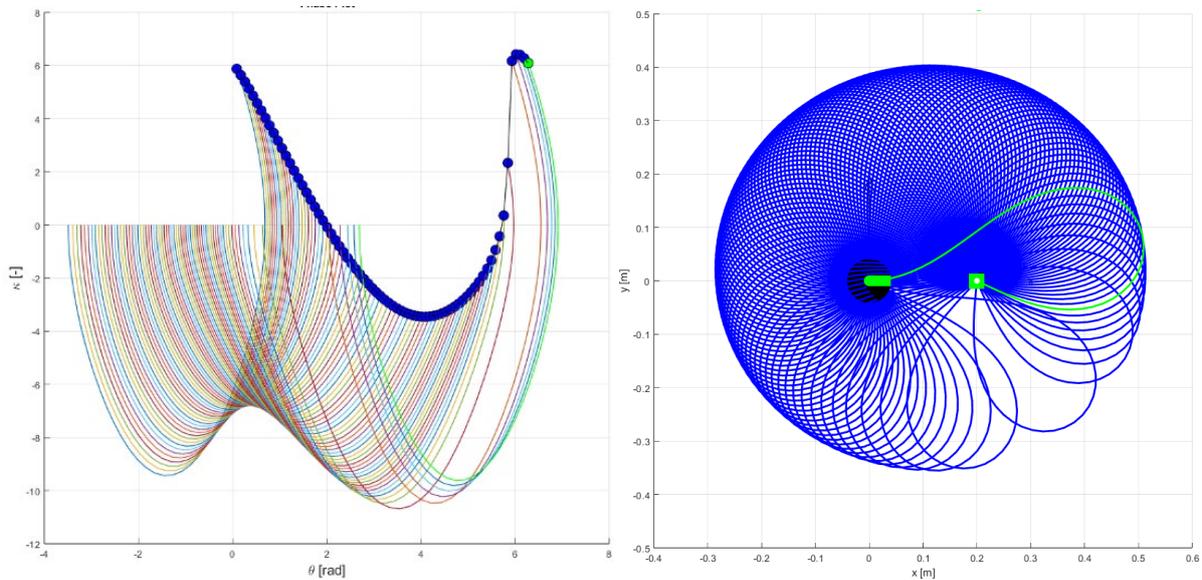


Figura 8: Trayectorias equiespaciadas.

4. Redes neuronales artificiales para resolver la Elástica

A continuación, se analizan las diferentes alternativas de uso de redes neuronales consideradas en este trabajo.

4.1. Modelo híbrido de red *MultiLayer Perceptron* e integrador DOE clásico

La primera alternativa planteada es la simplificación del problema de valor de contorno a resolver sobre el sistema de ecuaciones diferenciales de la elástica. La hipótesis es usar redes neuronales para inferir el valor inicial necesario de curvatura y fuerza interna que hace cumplirse la condición de contorno del extremo articulado sin necesidad de realizar un proceso iterativo de minimización del residuo de la condición de contorno. Así es posible integrar una sola vez el sistema de ecuaciones diferenciales (5) para obtener la solución correcta.

Para poder integrar el sistema de ecuaciones diferenciales (5) es preciso conocer el valor adecuado de la fuerza R y su orientación ψ y la curvatura en el empotramiento $\kappa(s=0)$ correspondiente a la orientación de empotramiento deseada θ_0 , para ello se van a entrenar tres redes neuronales de tipo *MultiLayer Perceptron* MLP con los datos de las 72 orientaciones para poder inferir esos valores en cualquier otra orientación. En definitiva, se trata de resolver un problema de regresión simple sobre las curvas de la Figura 9. Se emplean tres redes MLP que permiten obtener los valores precisos en el eje y (κ_0 , R , ψ respectivamente) para el valor de empotramiento θ_0 indicado en el eje x .

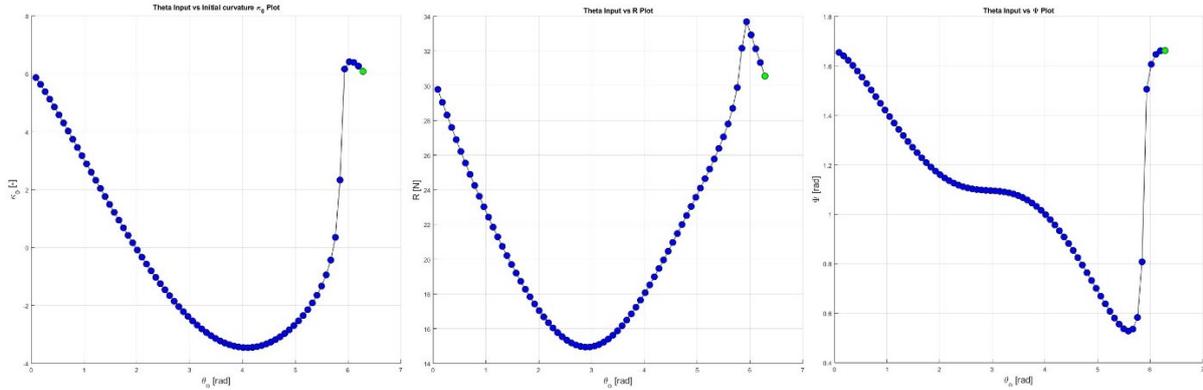


Figura 9: Datos de entrenamiento y curvas inferidas en las MLPs.

La arquitectura de cada una de las tres redes neuronal MLP empleadas está formada por seis capas: la primera con una única neurona que recibe el valor de θ_0 , cuatro capas ocultas con 15 neuronas cada una y función de activación tipo *sigmoid* y la capa de salida con una única neurona que genera la salida deseada (κ_0 , R o ψ) con función de activación identidad. Esta red neuronal ha sido entrenada con los datos obtenidos de solucionar el problema de valores en frontera del modelo de Cosserat, empleando el optimizador Adam con una tasa de aprendizaje de 0.015, 100.000 iteraciones y una función de pérdida de error cuadrático medio. Estos hiperparámetros han sido ajustados manualmente. La MLP ha sido creada utilizando el entorno de programación SciML de Julia: DifferentialEquations.jl [13], Optimization.jl [14] y Lux.jl [15].

El esquema de resolución completo quedaría como se ve en la Figura 10. Para un valor del empotramiento introducido θ_0 las tres MLPs infieren los valores de (κ_0 , R y ψ) que introducidos en un ODE solver tipo Runge-Kutta sobre el sistema (5) permiten obtener la variación de θ y κ con la longitud de arco s , y a continuación integrar el segundo sistema de ecuaciones diferenciales desde el valor inicial conocido de x_0 , y_0 para obtener a su vez la forma deformada de la barra en $x(s)$ e $y(s)$.

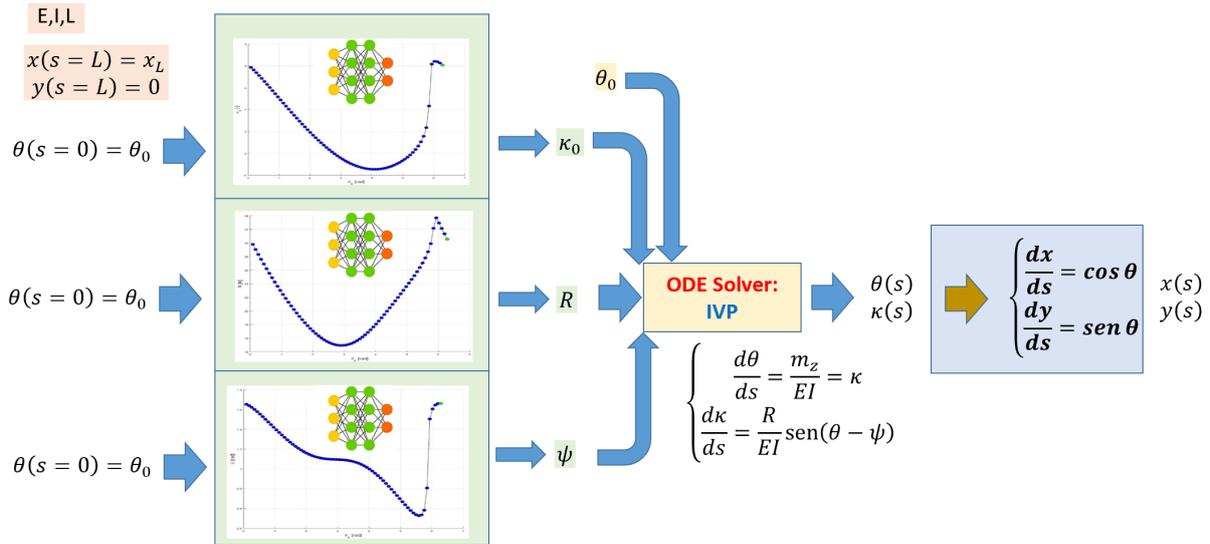


Figura 10: Vista esquemática 3MLPs + ODE Solver

A continuación, se muestran en la Figura 11 y Figura 12 las variables de estado de la primera posición entrenada $\theta_0 = 0$, $\kappa_0 = 0.5872$, los valores inferidos por el modelo híbrido en los 50 nodos de la barra, así como del error L2 correspondiente.

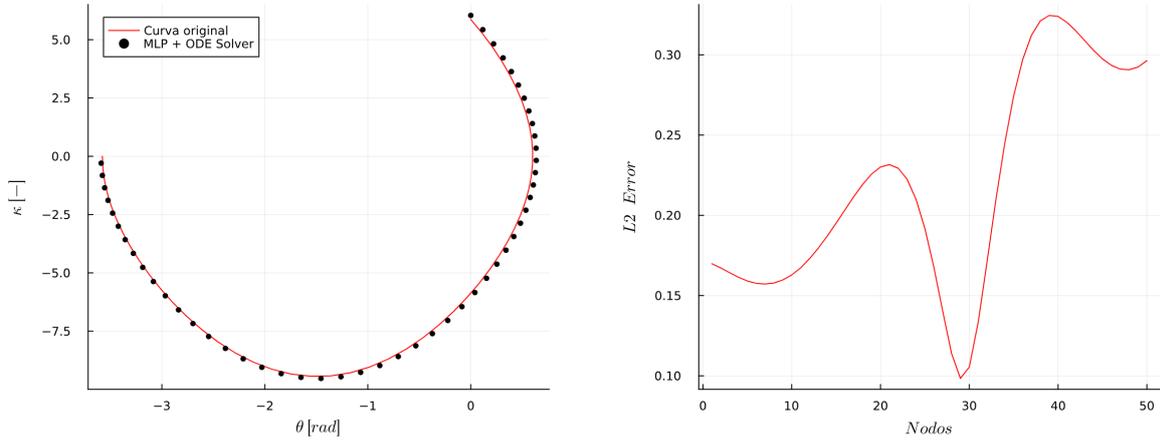


Figura 11: Diagrama de fase de las dos variables de estado, θ y κ , y error L2 en los nodos.

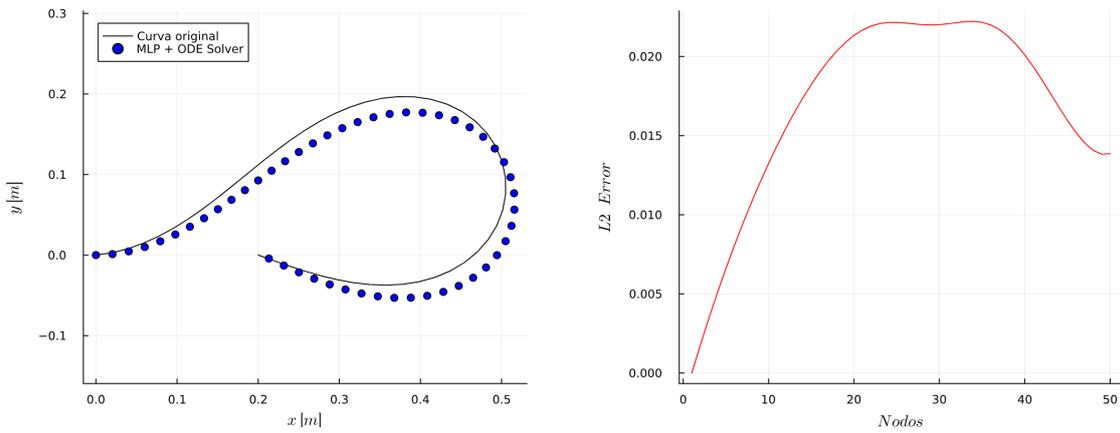


Figura 12: Posición espacial de los nodos en la deformada y error L2.

4.2. Modelo NODE general

Como se indicó en la introducción existe un tipo de redes neuronales que se enfocan a la integración de sistemas de ecuaciones diferenciales Neural Ordinary Differential Equations (NODEs) [9]. En esencia este tipo de redes aproxima mediante una red neuronal MLP la función integral que luego emplea en un integrador numérico más o menos clásico sujeto a algunas variaciones para adaptarse al entrenamiento de la red neuronal. Así, ni siquiera es preciso conocer el modelo del fenómeno físico. En nuestro caso eso significa que no usaremos las ecuaciones de las funciones integrales de (5), y en consecuencia no es necesario conocer R y ψ correspondientes a cada situación θ_0 a ser resuelta. No obstante, aún es preciso inferir el valor inicial de la curvatura κ_0 .

El modelo esquemático de esta alternativa es el que se puede ver en la Figura 13. Una vez introducido el valor de θ_0 que se desea resolver, una primera MLP infiere el valor de κ_0 (la misma empleada en el método anterior) que también es necesario en la integración de las ecuaciones (5), y a continuación se utiliza una NODE que sustituye las ecuaciones (5). Esta NODE infiere la función diferencial completa y empleando un solver de EDO permite obtener $\theta(s)$ y $\kappa(s)$. La NODE emplea una MLP formada por cuatro capas: la primera capa con dos neuronas; otras dos capas ocultas con 20 neuronas cada una y función de activación tipo *tangente hiperbólica*; y la capa de salida con dos neuronas y función de activación identidad. Esta red neuronal ha sido entrenada con los datos obtenidos de solucionar el problema de valores en frontera del modelo de Cosserat, empleando el optimizador Adam con una tasa de aprendizaje de 0.005, 5.000 iteraciones y una función de pérdida de error absoluto medio. Estos hiperparámetros han sido ajustados manualmente. Esta NODE ha sido creada utilizando el entorno de programación SciML de Julia y el método UDE [16].

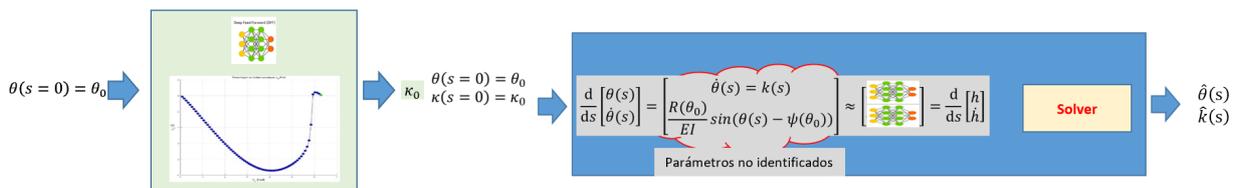


Figura 13: Vista esquemática de la NODE general

La Figura 14 muestra el resultado obtenido por la Node para la primera curva entrenada, $\theta_0 = 0$, $\kappa_0 = 0.5872$, así como el error L2.

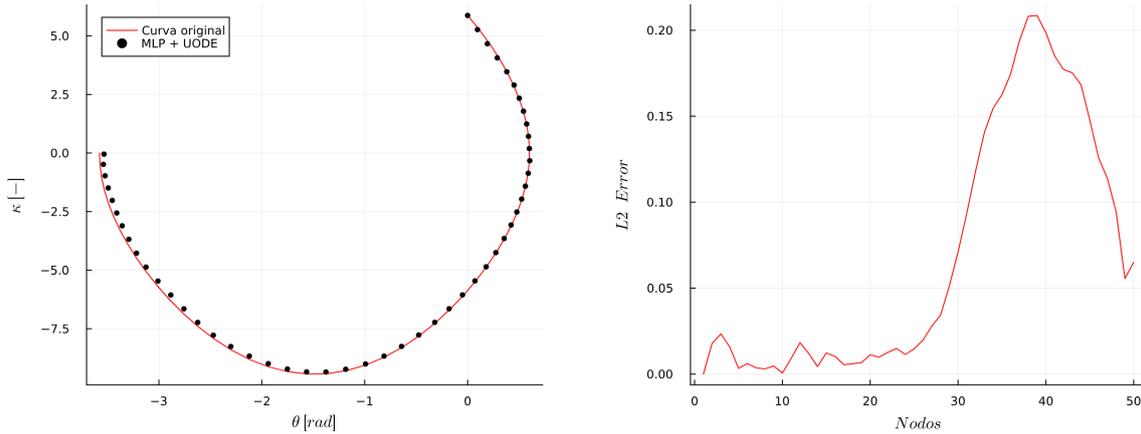


Figura 14: Diagrama de fase de las dos variables de estado θ y κ , y error L2 entre nodos.

4.3. Modelo NODE simplificada

La ventaja inicial de las redes NODE es que, como en el caso anterior, no es preciso conocer a priori un modelo de la función diferencial que representa el fenómeno físico a modelizar. Es en definitiva un procedimiento libre de modelo que en base a datos conocidos permite inferir nuevos resultados o incluso generar un modelo a partir de lo inferido. Otra de las ventajas de las redes neuronales NODE es que se puede incluir la parte del modelo que se conozca y completarlo con una red que incluya los efectos no modelizados. En nuestro caso, si observamos la ecuación (5) vemos que una de las ecuaciones diferenciales es simplemente el cambio de variable que precisamos para reducir la ecuación diferencial de segundo orden a un sistema de ecuaciones diferenciales de primer orden. Por tanto, es plausible aceptar esa parte de la función diferencial como cierta e inferir el resto. Es la única variación sobre el modelo de la Figura 13, y que se muestra en el esquema de la Figura 15.

Esta alternativa emplea una primera MLP para configurar los valores iniciales de la trayectoria inferiendo el valor de κ_0 a partir del valor deseado de θ_0 , por ejemplo ($\theta_0 = 0$, $\kappa_0 = 0.5872$), y una NODE para aproximar mediante una red neuronal los parámetros no identificados de las ecuaciones EDO del modelo de barra de Cosserat. La primera expresión EDO conocida es aportada al integrador numérico durante el entrenamiento, mientras que la segunda ecuación EDO desconocida es inferida íntegramente por la MLP de la NODE.

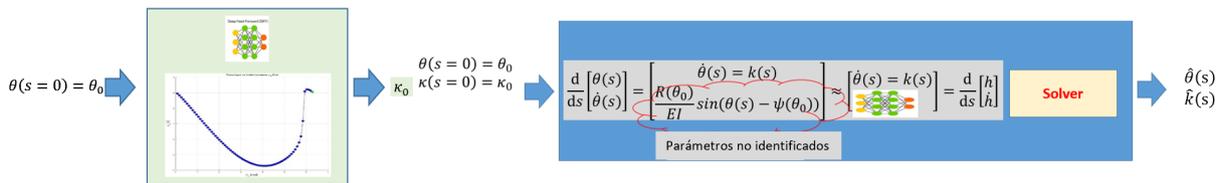


Figura 15: Vista esquemática del entrenamiento de la NODE 2

A continuación, la Figura 16 muestra las variables de estado obtenidas por la Node2 de la configuración de entrenamiento $\theta_0 = 0$, $\kappa_0 = 0.5872$, así como del error L2.

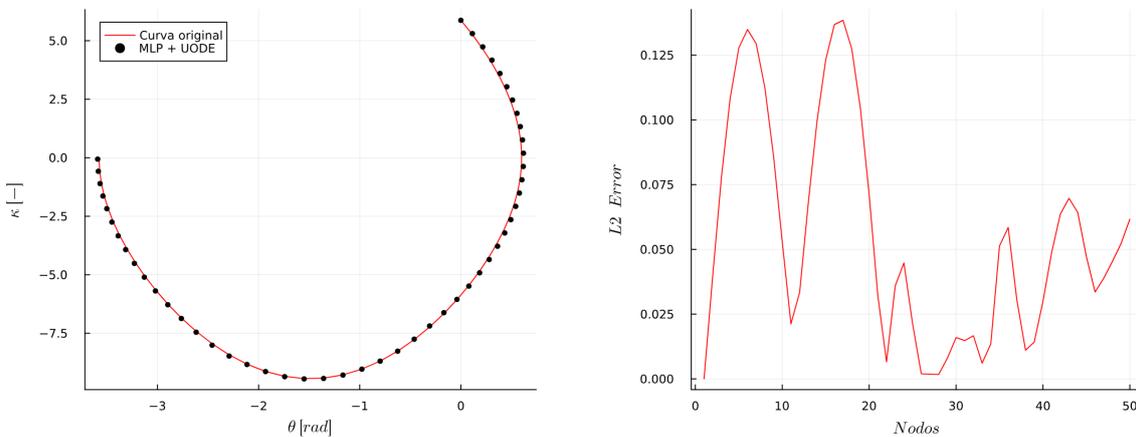


Figura 16: Diagrama de fase de las dos variables de estado θ y κ , y error L2 entre nodos.

4.4. Modelo NODE del sistema DOE completo

En las tres alternativas precedentes se ha resuelto la variación de la pendiente y la curvatura a lo largo de la longitud de arco s , $\theta(s)$ y $\kappa(s)$; para a continuación proceder a realizar una segunda integración sobre las segundas ecuaciones en (4) y resolver así la forma deformada de la barra $x(s)$ e $y(s)$. Si se observa la Figura 12 se puede apreciar que el error acumulado en dos integraciones directas numéricas de valor inicial se acumula y compromete la precisión del resultado. La última alternativa planteada afronta la modelización NODE del sistema de ecuaciones diferenciales que combina las ecuaciones en (4) y (5). Por tanto, esta NODE resuelve de una vez las cuatro variables de estado del modelo de barra de Cosserat, esto es $\theta(s)$, $\kappa(s)$, $x(s)$ e $y(s)$. Su esquema puede verse en la Figura 17, los datos iniciales son la posición del empotramiento x_0 , y_0 que es fijo, la orientación del empotramiento que se desea resolver θ_0 y la curvatura inicial κ_0 que es de nuevo inferida por la primera MLP ya mencionada anteriormente. Esos datos son introducidos en la NODE que es entrenada para inferir los resultados deseados.

La red neuronal MLP que infiere la derivada de la variable de estado en la NODE cuenta con cuatro neuronas de entrada, una para cada variable de estado y cuatro neuronas de salida, siendo el resto de los parámetros de red análogos al caso anterior. De estas cuatro variables de estado inferidas por la MLP, tres de ellas son utilizadas en la expresión de la parte derecha de las ecuaciones EDO, ver Figura 17, mientras que la cuarta $\kappa(s)$ es conocida y es aportada al ODE Solver.

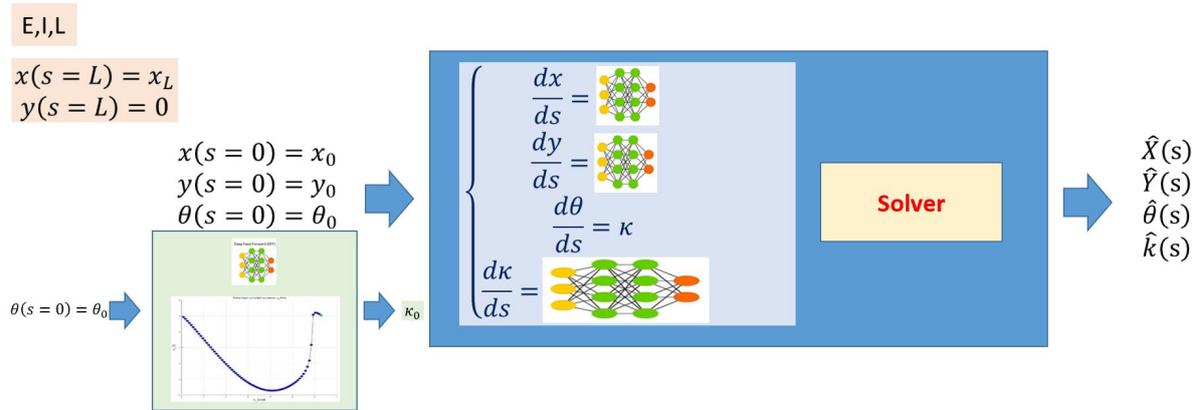


Figura 17: Vista esquemática NODE 3

En las Figuras 18 y Figura 19 se observan los resultados de nuevo para la primera de las elásticas entrenadas $\theta_0 = 0$, $\kappa_0 = 0.5872$, así como el error L2.

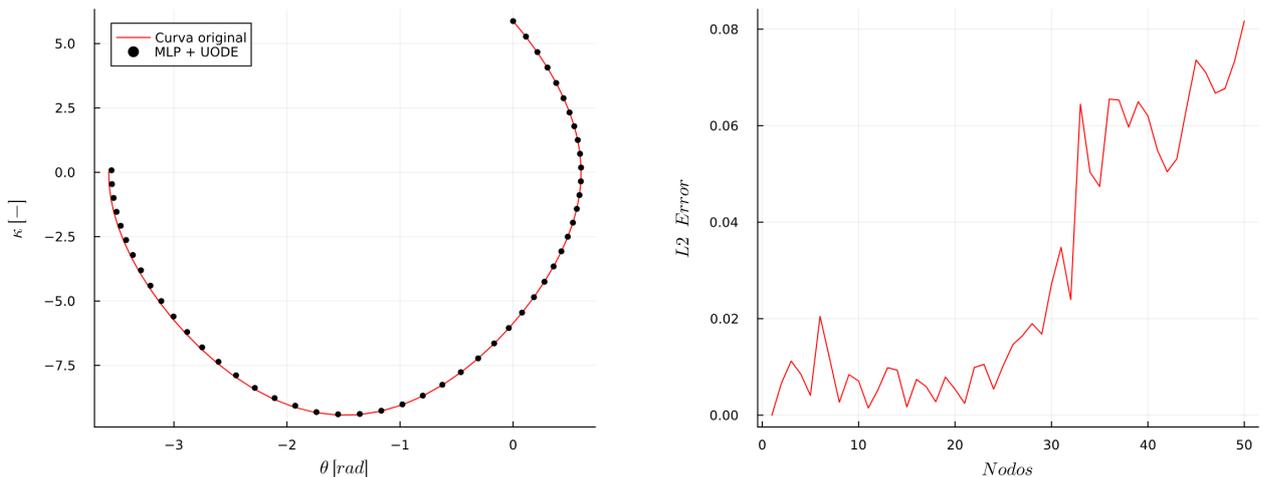


Figura 18: Diagrama de fase de las dos variables de estado θ y κ , y error L2 entre nodos.

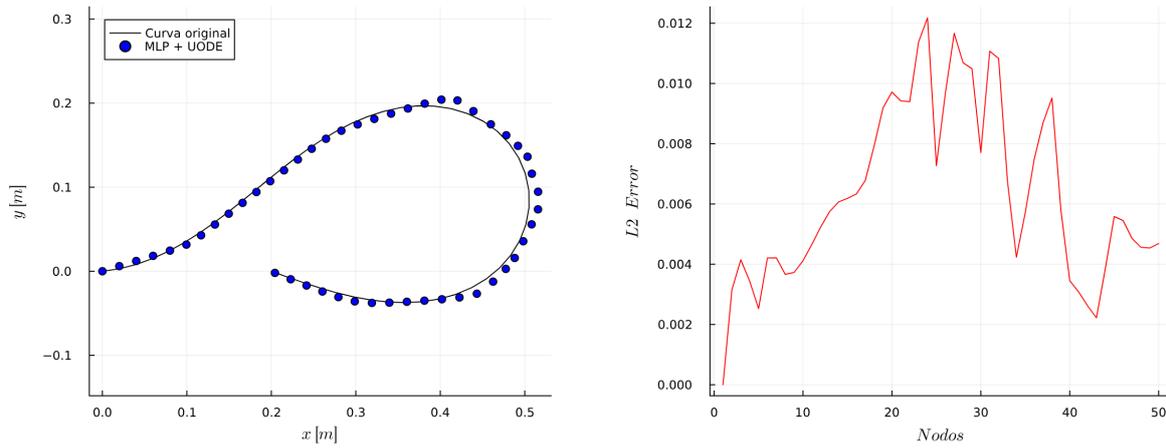


Figura 19: Posición espacial de los nodos en la deformada y error L2.

5. Discusión

Para realizar una comparativa de las diferentes alternativas es preciso añadir a los resultados anteriores unos datos relativos a tiempos de ejecución y costo computacional. En la Tabla 1 se comparan: el tiempo de ejecución de las operaciones de integración del ODE Solver y el de las operaciones de predicción de las NODE; el número de asignaciones o de reserva de memoria, utilizada para almacenar la estructura de datos y las variables; y el uso de memoria por el código, durante la ejecución de la operación analizada.

Tabla 1: Comparativa de rendimiento

	Tiempo (s)	Asignaciones (M)	Memoria (KiB)
Modelo híbrido	0,539834	0,000821	29.062
NODE general	0,004046	18,04	1.118.000
NODE simplificada	0,002073	15,22	966.984
NODE completa	0,002197	16,00	1.043.000

En relación al primer modelo híbrido de solución planteado, el empleo de redes neuronales en problemas de regresión es muy eficiente tanto a nivel de velocidad de inferencia como de precisión en la aproximación de la solución siempre que las funciones aproximadas sean continuas y unívocas. Así las MLP para aproximar valores iniciales para su posterior integración correspondientes a la Figura 9 tienen un muy buen rendimiento, no obstante, al encadenarse posteriormente dos integraciones numéricas los pequeños errores iniciales se ven aumentados en las soluciones finales de las Figuras 11 y 12. Además, como se puede observar en la Tabla 1 es el modelo más costoso en tiempo de ejecución.

Los modelos NODE son mucho más eficientes en tiempo de ejecución, y además tienen errores menores como se ha podido observar en las figuras de la sección precedente. El conocimiento de parte de la función diferencial supone una reducción de tiempos del 50%, y ampliar la función diferencial para resolver las cuatro variables de estado no ha supuesto una penalización en tiempos de ejecución. Todas las pruebas se han realizado con el mismo set de entrenamiento, y el mismo tipo de red neuronal, resultando que la definición de la función de error utilizada en el entrenamiento tiene una gran importancia en la consecución de buenos resultados.

El siguiente paso es depurar las fases de entrenamiento, analizando la cantidad de datos precisos para obtener buenos resultados y la influencia de ruido numérico en los datos. Finalmente se procederá a realizar ese entrenamiento con datos experimentales.

6. Conclusiones

En este trabajo se ha demostrado que las redes neuronales NODE pueden ser empleadas en la integración de las ecuaciones diferenciales de la deformación en el dominio de la variable espacial de longitud de arco, con la misma eficiencia que se emplean en la integración de las ecuaciones diferenciales de la dinámica en el dominio de la variable temporal.

Además, se han planteado diversas alternativas de redes para generar modelos sobre sistemas de ecuaciones de tamaño y nivel de conocimiento variable. Se ha observado que el procedimiento se adapta bien a problemas diferentes, pero que es clave el proceso de entrenamiento seguido, tanto en volumen de datos como en funciones de error y parámetros de los optimizadores.

7. Referencias

- [1] Bryson, C., Rucker, D., “Toward parallel continuum manipulators.”, *IEEE International Conference on Robotics and Automation (ICRA)*, 778-785, Hong Kong, China, (2014)
- [2] Chen, G., et al., “Design, modeling, and evaluation of parallel continuum robots: A survey.”, *Sci. China Technol. Sci.* **67**, 673-695, (2024)
- [3] Antman S.S., *Nonlinear Problems of Elasticity*, Springer, New York, NY, USA, (2005)
- [4] Kurtz, A., “Learning the Complete Shape of Concentric Tube Robots.”, *IEEE Transactions on Medical Robotics and Bionics* **2**, (2020)
- [5] Bhattacharjee, S., et al., “Analytical and Deep Neural Network Based Hybrid Modelling of Single-Segment Continuum Robot.”, *Acta del International Conference on Industrial Instrumentation and Control (ICI2C)* 815, Kolkata, India, (2021)
- [6] A. Zubizarreta, M. Larrea, E. Irigoyen, I. Cabanes, E. Portillo, Real time direct kinematic problem computation of the 3PRS robot using neural networks, *Neurocomputing*, vol. 271, pp. 104-114, 2018. Doi: 10.1016/j.neucom.2017.02.098.
- [7] O. Altuzarra, M. Urizar, M. Cichella, V. Petuya, Kinematic Analysis of three degrees of freedom planar parallel continuum mechanisms, *Mechanism and Machine Theory*, vol. 185, 2023, p. 105311. Doi: 10.1016/j.mechmachtheory.2023.105311.
- [8] Celledoni, E., et al., “Neural networks for the approximation of Euler’s elastica.”, *Computer Methods in Applied Mechanics an Engineering* **435**, 117584 (2025)
- [9] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, D. Duvenaud, Neural Ordinary Differential Equations, 32nd Conf. on Neural Information Processing Systems (NeurIPS 2018), Montréal (Canada), 2018.
- [10] Goss, V.G.A., “The History of the Planar Elastica: Insights into Mechanics and Scientific Method.”, *Science & Education* **18**, 1057–1082 (2009)
- [11] Legaard, C., et al., “Constructing Neural Network Based Models for Simulating Dynamical Systems.”, *ACM Computing Surveys* **55**, (2023)
- [12] Verner, JH., “Numerically optimal Runge-Kutta pairs with interpolants.”, *Numerical Algorithms* **53**, (2010)
- [13] Rackauckas, C., et al., “DifferentialEquations.jl – A Performant and Feature-Rich Ecosystem for Solving Equations in Julia.”, *Journal of open Research Software* **5**, (2017)
- [14] Vaibhav, K., Rackauckas, C., “Optimization.jl: A Unified Optimization Package (v3.12.1).”, *Zenodo* **7738525**, (2023)
- [15] Pal, A., “Lux: Explicit Parametrization of Deep Neural Networks in Julia (LuxLib-v1.6.1).”, *Zenodo* **14736728**, (2025)
- [16] Rackauckas, C., et al., “Universal Differential Equations for Scientific Machine Learning.”, *Cornell University arXiv:2001.04385v4*, (2020)