UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

Modelado dinámico de un pantógrafo ferroviario mediante el uso de redes neuronales artificiales.

Trabajo Fin de Máster

Máster Universitario en Ingeniería Aeronáutica

AUTOR/A: Gutiérrez Arnaldo, Sergio

Tutor/a: Gregori Verdú, Santiago

Cotutor/a: Gil Romero, Jaime

CURSO ACADÉMICO: 2022/2023

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

Modelado dinámico de un pantógrafo ferroviario mediante el uso de redes neuronales artificiales.

Trabajo Fin de Máster

Máster Universitario en Ingeniería Aeronáutica

AUTOR/A: Gutiérrez Arnaldo, Sergio

Tutor/a: Gregori Verdú, Santiago

Cotutor/a: Gil Romero, Jaime

CURSO ACADÉMICO: 2022/2023

Modelado de pantógrafo ferroviario mediante redes neuronales.

DOCUMENTO 1. MEMORIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Trabajo Final de Máster en Ingeniería Aeronáutica

Alumno: Sergio Gutiérrez Arnaldo

Director: Santiago Gregori Verdú

Codirector: Jaime Gil Romero

Curso 2022-2023

AGRADECIMIENTOS

Quiero dedicar este breve espacio para agradecer a toda la gente que me ha apoyado y ha estado conmigo durante estos dos largos años.

A mis padres, por apoyarme desde que con 17 años me fui de casa con el objetivo de un día ser ingeniero aeronáutico, y creer en mí para llegar a donde estoy ahora. A mi familia, hermano, tíos y sobre todo abuelos, que ya no están ninguno de ellos, pero sé que estarían muy orgullosos de lo que he conseguido. A mis amigos que me han acompañado, tanto a los que ya estaban como Judit, Trapote, Eva, Candela, Mara, Arana, Lucía e Ixhel; como los que he conocido en Valencia y han hecho más amenas las tardes de estudio y le han quitado importancia a los malos momentos y aportado valor y felicidad a los buenos; gracias, Iván, Mune, Sara, Diego, Ruth, Lucía y Sofía por haber estado en esta preciosa etapa, os llevo de por vida conmigo. También agradecer a mi tutor de TFM, Jaime Gil Romero por toda la dedicación que ha tenido conmigo durante este proyecto y por su completo apoyo cuando lo he necesitado.

Por último, agradecerle a Dios todas las alegrías que han ocurrido y por haberme cuidado y protegido durante todo este tiempo.

Resumen

La obtención de modelos dinámicos de sistemas suele realizarse mediante ajuste de los parámetros a partir de medidas experimentales. Aun así, cuando se trata de sistemas complejos con comportamientos no lineales, la propia formulación del modelo matemático no se ajusta perfectamente al comportamiento dinámico real del sistema. El objetivo principal de este trabajo es desarrollar una estrategia para obtener el modelo dinámico de un pantógrafo ferroviario mediante el uso de técnicas de inteligencia artificial, en concreto de redes neuronales. Dichas redes se entrenan con el objetivo de poder predecir ciertas variables de interés a partir de medidas experimentales sobre el pantógrafo, como, por ejemplo, la fuerza aplicada sobre el colector del pantógrafo a partir de la aceleración vertical del mismo o viceversa. Como en este trabajo no se disponen de medidas experimentales de estas magnitudes, los datos para entrenar las redes se obtendrán a partir de simulaciones numéricas en las que se utilizará un modelo de pantógrafo de masas concentradas. En el presente trabajo se exploran dos posibles aplicaciones de las redes neuronales entrenadas. La primera consiste en realizar medidas indirectas de ciertas magnitudes a partir de otros datos medidos (en este caso simulados). Esta aplicación puede resultar muy útil para predecir la fuerza de contacto entre pantógrafo y catenaria conociendo tan solo la aceleración del cabezal colector del pantógrafo, lo cual resulta mucho más simple y económico respecto del procedimiento actual de medida directa de dicha fuerza. La segunda aplicación consiste en utilizar la red neuronal entrenada como un modelo dinámico de pantógrafo capaz de calcular el movimiento del pantógrafo en función de la fuerza aplicada.

Palabras clave: Inteligencia Artificial, Redes neuronales, Modelado, Pantógrafo

Resum

L'obtenció de models dinàmics de sistemes sol realitzar-se mitjançant ajust dels paràmetres a partir de mesures experimentals. Així i tot, quan es tracta de sistemes complexos amb comportaments no lineals, la pròpia formulació del model matemàtic no s'ajusta perfectament al comportament dinàmic real del sistema. L'objectiu principal d'aquest treball és desenvolupar una estratègia per a obtindre el model dinàmic d'un pantògraf ferroviari mitjançant l'ús de tècniques d'intel·ligència artificial, en concret de xarxes neuronals. Aquestes xarxes s'entrenen amb l'objectiu de poder predir unes certes variables d'interés a partir de mesures experimentals sobre el pantògraf, com per exemple, la força aplicada sobre el col·lector del pantògraf a partir de l'acceleració vertical del mateix o viceversa. Com en aquest treball no es disposen de mesures experimentals d'aquestes magnituds, les dades per a entrenar les xarxes s'obtindran a partir de simulacions numèriques en les quals s'utilitzarà un model de pantògraf de masses concentrades. En el present treball s'exploren dues possibles aplicacions de les xarxes neuronals entrenades. La primera consisteix a realitzar mesures indirectes d'unes certes magnituds a partir d'altres dades mesurades (en aquest cas simulats). Aquesta aplicació pot resultar molt útil per a predir la força de contacte entre pantògraf i catenària coneixent tan sols l'acceleració del capçal col·lector del pantògraf, la qual cosa resulta molt més simple i econòmic respecte del procediment actual de mesura directa d'aquesta força. La segona aplicació consisteix a utilitzar la xarxa neuronal entrenada com un model dinàmic de pantògraf capaç de calcular el moviment del pantògraf en funció de les força aplicada.

Paraules clau: Intel·ligència Artificial, Xarxes neuronals, Modelatge, Pantògraf

Abstract

Obtaining dynamic models of systems is usually done by adjusting the parameters from experimental measurements. Even so, when it comes to complex systems with nonlinear behaviors, the formulation of the mathematical model itself does not fit perfectly with the actual dynamic behavior of the system. The main objective of this work is to develop a strategy to obtain the dynamic model of a railway pantograph through the use of artificial intelligence techniques, specifically neural networks. These networks are trained with the aim of being able to predict certain variables of interest from experimental measurements on the pantograph, such as the force applied on the pantograph manifold from the vertical acceleration of the pantograph or vice versa. As experimental measurements of these magnitudes are not available in this work, the data to train the networks will be obtained from numerical simulations in which a concentrated mass pantograph model will be used. This paper explores two possible applications of trained neural networks. The first consists of making indirect measurements of certain magnitudes from other measured data (in this case simulated). This application can be very useful to predict the contact force between pantograph and catenary knowing only the acceleration of the pantograph collector head, which is much simpler and cheaper compared to the current procedure of direct measurement of this force. The second application is to use the trained neural network as a dynamic pantograph model capable of calculating the pantograph movement as a function of the applied force.

Keywords: Artificial Intelligence, Neural networks, Modeling, Pantograph

Índice

Índice de ilustraciones

Índice de tablas

1. Introducción

El transporte es un elemento clave para el desarrollo socioeconómico de un país. El sector del ferrocarril es una parte esencial del transporte público y lo seguirá siendo en el futuro. En su funcionamiento, uno de los elementos clave y el que es el objeto de estudio del presente documento es la interacción pantógrafo-catenaria, que suministra la energía eléctrica necesaria para el funcionamiento del sistema. La capacidad de esta infraestructura para transportar energía eléctrica suficiente de manera precisa y segura es indispensable para el ferrocarril.

Un pantógrafo es un dispositivo montado en la parte superior de un tren cuya función principal es recolectar energía eléctrica desde una catenaria, para suministrarla a los motores del vehículo. El pantógrafo es una estructura articulada con un conjunto de barras que se deslizan sobre los cables de contacto de la catenaria.

1.1 Motivación

Es importante tener en cuenta que los trenes [5] están diseñados para moverse a altas velocidades y tienen que ser estudiados desde un punto de vista dinámico, en el que todas las fuerzas y desplazamientos en el pantógrafo cambian a lo largo del tiempo. El rendimiento del sistema pantógrafo-catenaria viene principalmente definido por la fuerza de contacto, que es la fuerza que es aplicada por la catenario sobre el pantógrafo y viceversa, siendo deseable tener un nivel de oscilación bajo o una fuerza de contacto más suave. En la ilustración 1 se muestra el sistema pantógrafo de un tren. Dicha fuerza es obtenida mediante simulaciones con modelos computacionales o mediante ensayos experimentales. Las normas europeas tratan tanto la validación de modelos de pantógrafo como la certificación de la medida de la fuerza de contacto en vía. Por lo tanto, es necesario disponer de modelos de pantógrafo que permitan realizar simulaciones fidedignas pero que también pueden ayudar para medir la fuerza de contacto de manera indirecta debido a la imposibilidad de ubicar una célula de carga (sensor de fuerza) entre el pantógrafo y la catenaria.

Se han desarrollado varias técnicas de medición de fuerza de contacto en el pantógrafo, incluyendo el uso de sensores de fuerza y acelerómetros, los cuales permiten inferir la fuerza corrigiendo la medida de las células de carga con el termino inercial de la masa del pantógrafo que está por encima de las células de carga y las fuerzas aerodinámicas. Sin embargo, la instrumentación de los pantógrafos requiere importantes modificaciones, donde instalar las células es complicado. Por tanto, parece muy conveniente obtener la fuerza de contacto con tan solo las medidas de aceleración, pero para ello se tiene que disponer de un buen modelo de pantógrafo. Las técnicas de inteligencia artificial (IA) puede ser una herramienta útil para relacionar la fuerza con la aceleración y dar solución por tanto a este problema.

Ilustración 1: Pantógrafo de un tren

En particular, las redes neuronales pueden ser muy apropiadas para este problema, permitiendo la generación de modelos sin necesidad de conocer ni modelar la física del sistema. El uso de estos algoritmos puede ayudar a analizar el comportamiento dinámico del sistema ante distintas condiciones, como variaciones de temperatura o viento, sin necesidad de realizar costosos ensayos en vía. El principal requerimiento [1] para generar modelos de IA es disponer de datos para entrenarlos, datos obtenidos en bancos de ensayos pueden ser utilizados para entrenar las redes neuronales para ser después utilizadas para hacer la tarea para la que han sido entrenadas.

El objetivo es desarrollar redes neuronales que sean capaz de relacionar secuencias temporales de variables físicas del pantógrafo, por ejemplo, fuerza de contacto y aceleración del punto de aplicación de la fuerza.

En este contexto, a lo largo de este trabajo se explora en detalle cómo las técnicas de inteligencia artificial mediante redes neuronales pueden ser aplicadas para generar sistemas dinámicos que se utilicen en el análisis y evaluación del pantógrafo, disminuyendo el margen de error que se presentan en la actualidad y permitiendo así un mayor control y seguridad de la operación ferroviaria. Estos hallazgos pueden ser aplicados tanto en el diseño de nuevas infraestructuras, como en la mejora y optimización del diseño y la gestión de trenes ya en funcionamiento.

1.2 Antecedentes

Las redes neuronales son un método de procesamiento de datos inspirado en la estructura y funcionamiento del cerebro humano. Una red neuronal artificial es una red compuesta por un gran número de unidades de procesamiento interconectadas entre sí por medio de enlaces.

Cada unidad de procesamiento en una red neuronal se llama neurona artificial. En ella se reciben una o varias señales de entrada, se procesan y se emite una señal de salida que es enviada a otras neuronas.

Las conexiones entre neuronas artificiales frecuentemente vienen definidas por pesos y sesgos, que controlan el nivel de activación entre neuronas. Estos valores son ajustados de manera tal que las predicciones de la red produzcan el menor error posible respecto a datos conocidos en un proceso denominado entrenamiento.

Tal y como se muestra en la ilustración 2, la estructura básica de una red neuronal artificial incluye tres capas principales: la capa de entrada (inputs), la capa oculta (hidden layer) y la capa de salida (output layer). En la capa de entrada se introducen las variables de entrada, mientras que en la capa de salida se produce la salida de datos. En la capa oculta se realizan las operaciones y cálculos necesarios para alcanzar la salida deseada.

A menudo, se pueden usar varias capas intermedias para procesar los datos. El tamaño de las capas intermedias, así como el número de las mismas determina la capacidad de la red para predecir fenómenos más complejos de una manera precisa.

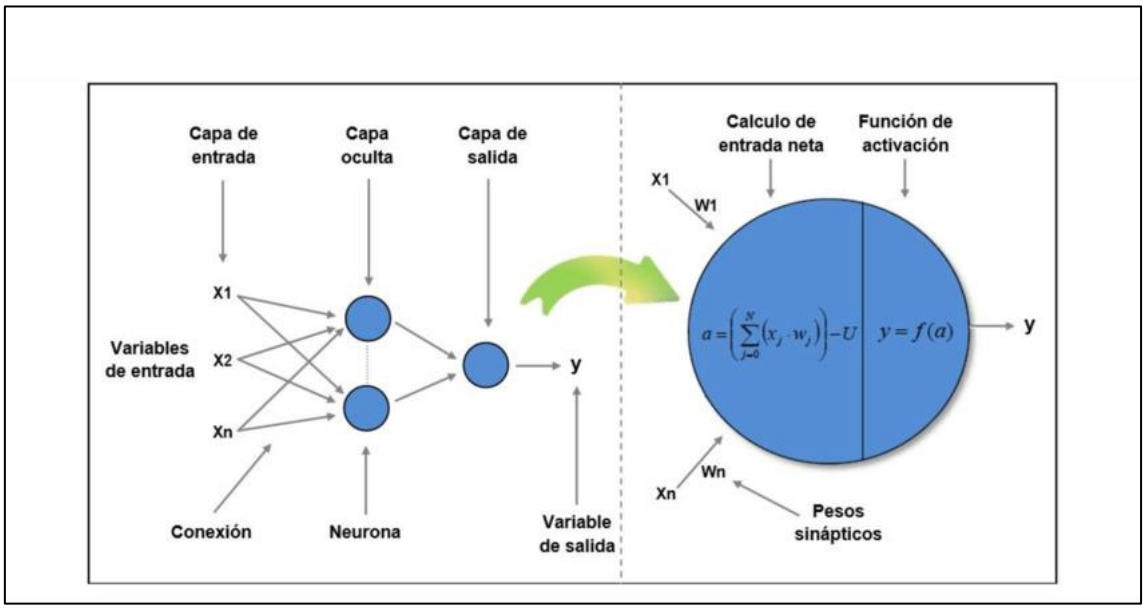

Ilustración 2: Estructura de una red neuronal

En general, la capacidad de las redes neuronales para analizar y categorizar datos complejos las hace útiles para una variedad de tareas. Estos incluyen procesamiento de imágenes, reconocimiento de voz, análisis de datos financieros e identificación de patrones de interés en big data. A medida que se realizan más y más investigaciones sobre redes neuronales, se vuelven cada vez más importantes en áreas como la automatización de procesos, la toma de decisiones complejas y la robótica.

Aunque las redes neuronales se han utilizado con éxito en una variedad de tareas, también tienen algunas limitaciones. En particular, las redes neuronales pueden tardar mucho en entrenarse correctamente, y un entrenamiento efectivo puede requerir una gran cantidad de datos. También pueden tener problemas para generalizar datos y sobreajustarlos (overfitting). Sin embargo, a pesar de estas limitaciones, las redes neuronales siguen siendo una poderosa herramienta para el análisis de datos.

Las redes neuronales pueden clasificarse según la topología de red, lo que determina la arquitectura específica para conectar las neuronas entre sí. A continuación, se detallan algunas de las topologías más comunes:

- **Redes neuronales *feedforward***: Las redes neuronales *feedforward* son una de las arquitecturas más simples y utilizadas en el procesamiento de datos. Su estructura consiste en una serie de neuronas dispuestas en capas contiguas en las que la información fluye en una única dirección, desde la capa de entrada. a través de las capas ocultas hasta la capa de salida. Cada neurona se conecta únicamente a las neuronas de la capa anterior y a la capa posterior
- **Redes neuronales recurrentes**: Las redes neuronales recurrentes son una arquitectura que permite a las neuronas tener conexiones hacia adelante y hacia atrás, lo que les permite tener memoria a corto plazo. Estas redes pueden utilizar información temporal y secuencial de las entradas actuales y anteriores para mejorar la precisión de la predicción. A diferencia de las redes feedforward, las redes recurrentes permiten que la información fluya en ambas direcciones, lo que es ideal para problemas de procesamiento de lenguaje natural.
- **Redes neuronales convolucionales**: Las redes neuronales convolucionales se utilizan comúnmente en el análisis de imágenes. En ellas, se aplican convoluciones a los datos de entrada para extraer características específicas de la imagen [2] [3]. En lugar de tener conexiones entre todas las neuronas, las convoluciones se aplican solo a ciertas regiones de la imagen. Las redes convolucionales también pueden incluir capas de agrupación, que reducen el tamaño de las características extraídas para reducir la carga computacional.
- **Redes neuronales Bayesianas**: Las redes neuronales Bayesianas son una forma de modelar la incertidumbre relacionada con los datos de entrada. En estas redes, cada entrada se asume incierta y se modela como una variable aleatoria. Las redes Bayesianas utilizan el teorema de Bayes para asignar probabilidades a las diferentes salidas potenciales.
- **Redes neuronales autoencoders**: Un autoencoder es una red neuronal que aprende a comprimir y descomprimir datos. La arquitectura de autoencoder típicamente consta de dos partes: el codificador y el decodificador. El codificador toma una entrada de datos y produce una representación comprimida (también conocida como *hidden layer*). Esta representación puede ser usada para reducir los datos de la entrada a una forma más manejable y para detectar patrones. El decodificador toma la representación comprimida y la convierte de nuevo en la forma original.

Estas arquitecturas básicas de redes neuronales se utilizan en diferentes áreas y aplicaciones y pueden ser combinadas de diferentes maneras para crear redes neuronales más complejas. Cada arquitectura tiene sus ventajas y desventajas, y es importante seleccionar la que mejor se adapte a la tarea en cuestión. Las redes neuronales forman una parte esencial de la inteligencia artificial y el aprendizaje

automático, y su importancia no deja de crecer a medida que se desarrollan nuevas arquitecturas y aplicaciones.

1.3 Problema de interés

En este trabajo, se pretende utilizar diferentes tipos de redes neuronales con la finalidad de que sirvan como modelo del pantógrafo. Estas redes tienen que ser entrenadas con datos de un pantógrafo de modo que aprenda a reproducir su dinámica. Dado de que en este trabajo no se disponen de medidas experimentales que puedan servir para entrenar las redes, se va a utilizar un modelo de pantógrafo computacional del que se obtendrán los datos que emularán o remplazarán las medidas del sistema real. Por tanto, se tendrá que crear un modelo computacional e integrarlo numéricamente con el fin de generar datos para entrenar las diferentes redes y posteriormente probar su eficacia.

La aplicación de las redes neuronales diseñadas consiste en utilizar las redes para modelar el pantógrafo para analizar y estudiar el sistema pantógrafo-catenaria. En este caso, las magnitudes de entrada y de salida de la red serán distintas, por ejemplo se puede crear un modelo que proporcione la aceleración del cabezal del pantógrafo en función de la fuerza aplicada.

El pantógrafo presenta una serie de condiciones, como puede ser la fricción seca, que producen un comportamiento no lineal, haciendo muy complicada la tarea de su modelado. En este documento se realizarán dos modelos del pantógrafo, uno lineal y otro no lineal al que se le añadirán fuerzas de fricción seca.

1.4 Estado del arte

Existen muchos estudios y documentación sobre soluciones de inteligencia artificial para problemas de ingeniería, en este caso, se valorarán estudios previos donde la inteligencia artificial sirva para la creación de modelos subrogados aplicados a respuestas dinámicas de estructuras que presenten ciertas similitudes con la temática que se trata en este trabajo.

El primer artículo valorado es “**Predicting the dynamic response of a structure using an artificial neural network”[8]** . El objetivo de esta investigación era utilizar el aprendizaje automático para hacer predicciones sobre la respuesta dinámica de un sistema estructural no lineal, en concreto una viga.

El material de la viga se describió utilizando un modelo constitutivo de material elástico lineal simple con un módulo de Young similar al del acero. Toda la respuesta de la viga se supone que tiene un comportamiento elástico lineal. La interacción por fricción entre el extremo de la viga y una pared fija es la causa del comportamiento no lineal descrito. La no linealidad se puede observar en dos escenarios de carga: uno con carga de baja presión y alto coeficiente de fricción y otro con carga de alta presión y bajo coeficiente

de fricción. En el primer caso, la respuesta sería más parecida a una respuesta fija-fija, mientras que en el segundo caso sería más parecida a una respuesta fija-libre. Por lo tanto, la respuesta puede producir modos completamente diferentes dependiendo de los escenarios de carga. Esto demuestra la no linealidad esperada en este modelo. Se muestran los estudios mencionados anteriormente en la ilustración 3.

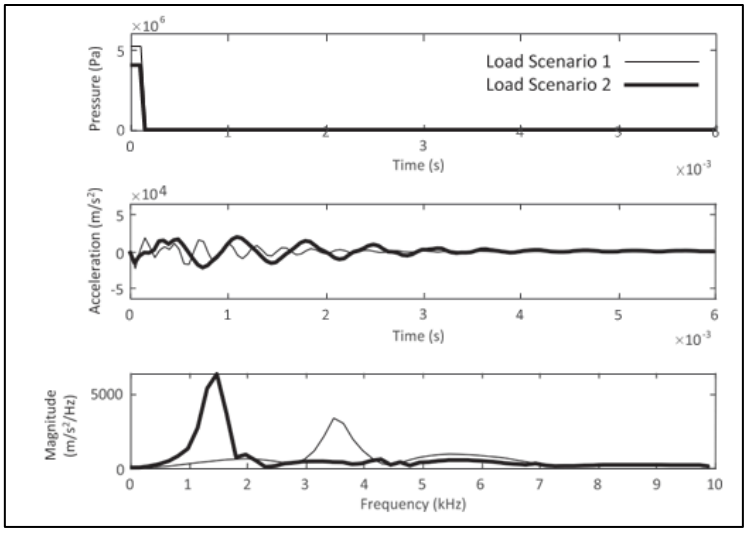

Ilustración 3: Casos de prueba del estudio

El objetivo es predecir la aceleración de varios puntos de la viga, pero para reducir la dimensión de estas variables se utilizan operadores según el método GSD (*granulometric size distribution*) que permiten representar la aceleración con muchos menos datos, denominados GSD output.

En el artículo se utiliza una red neuronal artificial (ANN) para relaccionar los imputs físicos del problema con los GSD output que representan la aceleración. Se opera del siguiente modo; el resultado de este proceso se refleja en la ilustración 4:

- Se crea una curva de aceleración para cada conjunto de parámetros de características físicas. Sea Ui el conjunto de características físicas del sistema utilizado para generar una curva de aceleración Ai.
- Se creó la GSDi correspondiente a cada Ai a partir de los tamaños de tamiz S1,i an S14,i.
- Todas las GSDi se agruparon de manera jerárquica.
- Se capacitó a una RNA para usar Ui como entrada y GSDi como salida utilizando conglomerados.
- Tras el entrenamiento, la RNA puede predecir cada GSDk utilizando un nuevo conjunto de parámetros físicos Uk.
- Una vez que se descubre una nueva GSDk, se utiliza un enfoque de vecino más cercano para determinar a qué clúster predeterminado pertenece la nueva GSDk y a qué curva de aceleración conocida Ai está más próxima dentro del clúster.

Por lo tanto, se predice variables del GSD (que nos permiten conocer la aceleración) a partir de la RNA a partir del conjunto de características físicas del sistema. **Esto completa el proceso de realizar predicciones sobre la respuesta dinámica de una estructura utilizando una RNA en lugar de un modelo basado en la física.**

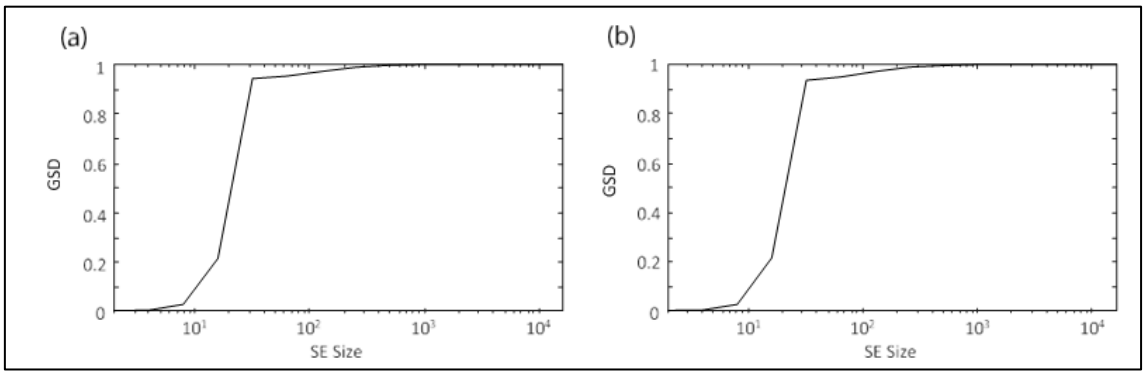

Ilustración 4 : (a) GSD producida por RNA. (b) GSD existente más cercana.

El segundo estudio analizado previo es ***Artificial neural networks in structural dynamics: A new modular radial basis function approach vs. convolutional and feedforward topologies***, [9] cuyo objetivo es desarrollar una variedad de redes neuronales artificiales (RNA) y determinar qué tipo de red neuronal puede predecir con mayor precisión las deformaciones estructurales medidas mediante la comparación de experimentos. Para lograrlo, se proponen tres RNA distintas. En primer lugar, la forma clásica de una RNA en forma de red neuronal feedforward (FFNN). El segundo método sugiere una nueva red neuronal modular de función de base radial (RBFNN), mientras que el tercer método sugiere una red neuronal convolucional profunda (DCNN). Se estudia la aplicabilidad de cada tipo de red mediante cálculos comparativos entre mediciones y predicciones numéricas mejoradas con redes neuronales.

Los experimentos que se realizan en el presente estudio tienen como objetivo cubrir una amplia gama de velocidades de deformación en la respuesta dinámica de las estructuras. Esto permite medir y proporcionar evoluciones de las deformaciones estructurales dependientes de la velocidad de deformación, incluidas las no linealidades geométricas y físicas, a las redes neuronales desarrolladas.

En el caso de la FFNN permite el uso de múltiples capas ocultas para aumentar la complejidad de la aproximación de funciones. La red neuronal aproxima los valores de salida a sistemas algebraicos de ecuaciones que incluyen funciones de activación no lineales en lugar de ecuaciones diferenciales en códigos de elementos finitos, como las ecuaciones de evolución en las leyes de materiales.

Este estudio espera aplicar las RBFNN a la dinámica estructural porque, en comparación con la FFNN, el comportamiento de deformación oscilante puede aproximarse mejor con un número menor de neuronas. Las áreas de activación de las neuronas ocultas están limitadas a regiones capturadas, a pesar de que las RBFNN muestran ventajas en la aproximación de funciones complejas. El usuario decide los centros, y la mayoría de las veces solo se utiliza la suma ponderada entre la capa oculta y la de salida en lugar de retropropagación de errores con los centros. Como resultado, el estudio actual sugiere

una RBFNN modular para superar los campos de activación demasiado pequeños y permitir áreas de activación más amplias. Para tareas específicas, las redes neuronales modulares ofrecen subredes independientes y agregan sus resultados a la salida final completa.

Para una red convolucional profunda, los datos de entrada se correlacionan con los datos de salida en forma de deflexión de la placa. La DCNN (red neuronal convolucional profunda) desarrollada en este estudio genera pares ordenados de valores. Esto también se hizo para la FFNN y la RBFNN, pero en la DCNN se lee la evolución completa de las variables de entrada a lo largo de todo el periodo de carga y se correlaciona con todo el historial de deformación. Esta es una distinción significativa entre DCNN y FFNN que se utiliza en la investigación actual. Siguiendo este método, los conjuntos de datos de entrada y salida se pueden ver como representaciones de evoluciones de carga y deformación, similar a las ilustraciones de reconocimiento de patrones, que también son números. Los autores son conscientes del hecho de que los valores de entrada en este caso se derivan de datos de medición secuenciales. Sin embargo, todas las deformaciones dependientes de la trayectoria se incluyen en los mapas de características de la red de la manera sugerida. Una vez presentadas las tres redes entrenadas, en la ilustración 5 se muestran los resultados.

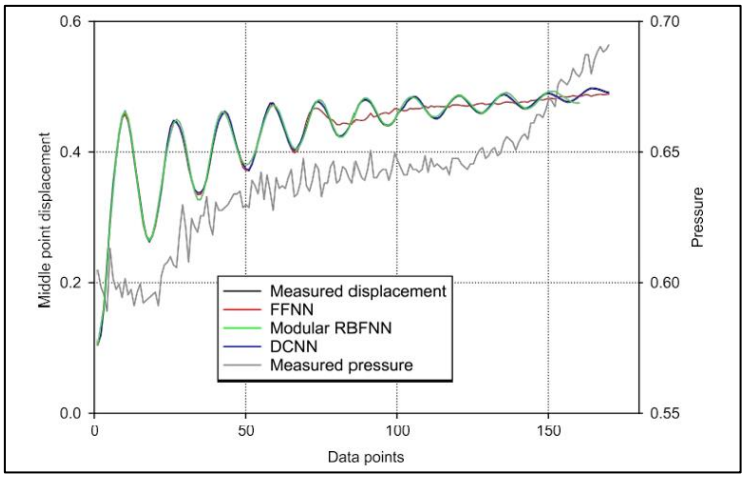

Ilustración 5: Mediciones y simulaciones con FFNN, RBFNN modular y DCNN para una placa de acero

La deformación deseada no presenta oscilaciones, sino deformaciones mayores que en el caso de las placas de acero. El FFNN y el DCNN son capaces de calcular estas deformaciones. La DCNN resultó ser la más precisa, incluso con un número menor de pesos y sesgos en comparación con el caso FFNN. La FFNN muestra elerror más alto en comparación con la medición, pero se mantiene dentro de un rango aceptable.La RBFNN no es capaz de predecir estas desviaciones monótonas del punto medio. La razón puede ser que la ventaja de la RBFNN es simular señales compuestas por funciones de tipo gaussiano, que no se dan en este ejemplo. Concluyendo, en el presente estudio se propusieron tres tipos de redes neuronales artificiales para una aplicación en dinámica estructural. La FFNN clásica tenía sus límites en el caso de deformaciones estructurales viscoplásticas oscilantes complejas bajo cargas repetidas. Incluso para cargas únicas se observaron divergencias tras un número especial de puntos de datos utilizados. En cada ejemplo de comportamiento estructural oscilante,

la RBFNN fue capaz de predecir los valores de salida deseados. Sin embargo, fue necesario extender esta teoría a una red modular ampliando el rango de activación de la red oculta y, por tanto, de la neurona de salida. La DCNN resultó ser el tipo de red más potente, ya que genera un conjunto de capas filtrando los mapas de características de entrada. Sin embargo, en el caso de la RBFNN fueron necesarios menos parámetros para obtener resultados muy similares a los de la DCNN. Una ventaja adicional de la DCNN es que utiliza toda la evolución de los valores de entrada y salida durante el entrenamiento, lo que resulta esencial para las deformaciones dependientes de la trayectoria. En una validación con datos de prueba adicionales, las simulaciones que utilizaron los tres tipos de redes se comportaron de forma estable, si se producían cambios en los datos de entrada.

El tercer estudio es ***Predictions of vertical train-bridge response using artificial neural network-based surrogate model*** **[10]**. En este estudio se presenta un método de modelado sustitutivo basado en un modelo de red neuronal artificial autorregresivo no lineal con entrada exógena y una muestra importante. Este método puede pronosticar las respuestas de sistemas dinámicos, como los sistemas vehículo-puente, sometidos a excitaciones estocásticas, y mejorar la eficacia de los cálculos de fiabilidad de los sistemas vehículo-puente. Además, se propone un método para el análisis del método. Se utiliza un modelo de un cuarto de vehículo para verificar la precisión del método propuesto. Además, se utiliza un modelo de red neuronal artificial autorregresivo no lineal con entrada exógena para predecir las respuestas de los sistemas vehículo-puente verticales. Los hallazgos muestran que, cuando la muestra con la respuesta máxima se considera una muestra importante y se utiliza para entrenar el modelo de red neuronal artificial autorregresivo no lineal con entrada exógena y solo requiere una simulación numérica de dos veces como máximo (o simulación Monte Carrasco), el modelo de red neuronal artificial autorregresivo no lineal con entrada exógena tiene mejor precisión de predicción.

Los procedimientos de entrenamiento y predicción del modelo NARX-ANN se describen a continuación:

- Paso 1: el modelo AR determina la muestra aleatoria inicial de entrenamiento y el modelo numérico real calcula la respuesta.
- Paso 2: Se describen las opciones generales de NARX-ANN. Estos incluyen los desfases máximos calculados de entrada y salida, el número máximo de épocas, el valor mínimo del gradiente y el algoritmo de entrenamiento. A continuación, el solver MATLAB se utiliza para entrenar y validar el modelo NARX-ANN.
- Paso 3. El modelo NARX-ANN entrenado puede calcular los valores de predicción de los sistemas dinámicos y el modelo AR puede generar m excitaciones aleatorias. El modelo numérico real busca y modifica simultáneamente una muestra importante con una respuesta máxima en todas las excitaciones aleatorias y la selecciona como muestra de entrenamiento.
- Paso 4, reinicio del proceso de entrenamiento y predicción basado en muestra importante. Los valores de predicción se actualizan simultáneamente basándose en m excitaciones aleatorias y el nuevo modelo NARX-ANN se entrena basándose

en la muestra importante. El postprocesamiento de la respuesta de predicción ahora está disponible. Tenga en cuenta que si el modelo NARX-ANN basado en la muestra de entrenamiento inicial parece lo suficientemente preciso, se debe ignorar el proceso de ciclo e implementar directamente el postprocesamiento.

Una vez desarrollado el modelo, y por ser esta la gráfica de más interés por las características del presente documento se muestra en la ilustración 6 los valores reales frente a la predicción y el error del modelo.

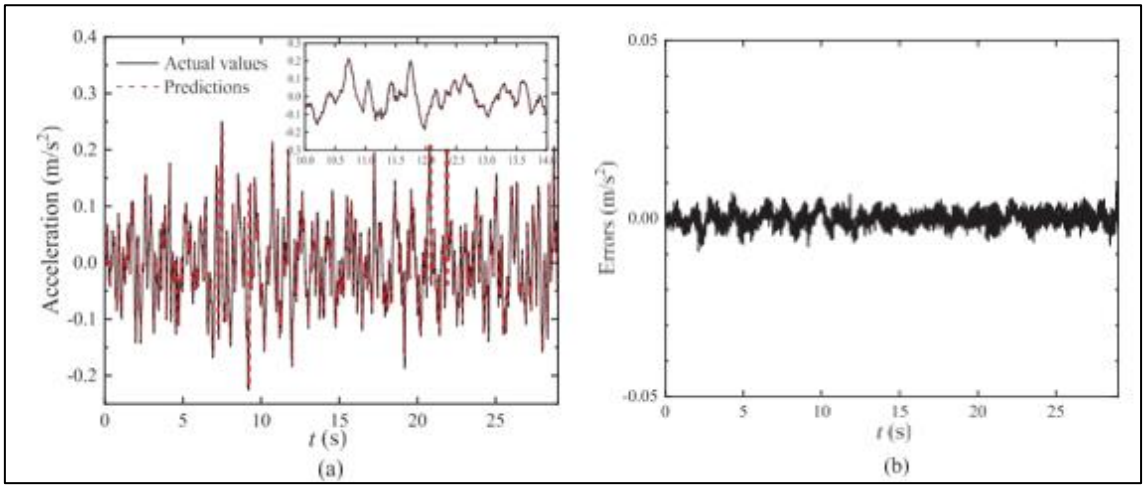

Ilustración 6: Historial temporal de la aceleración y sus errores: (a) valores reales y de predicción y (b) errores.

Las siguientes son **algunas conclusiones**: La precisión de la predicción del modelo NARX-ANN está significativamente influenciada por la selección de muestras de entrenamiento. El modelo NARX-ANN entrenado por muestra importante (muestra MaxM) tiene una precisión excelente en la predicción de respuestas dinámicas, mientras que el modelo NARX-ANN entrenado por muestra MinM tiene una precisión baja. El modelo sustituto solo requiere una simulación numérica (o MCS) de dos veces como máximo para el entrenamiento del modelo NARX-ANN y tiene una precisión de predicción notable incluso para sistemas complejos, como los sistemas vehículo-puente.

1.5 Objetivos

El objetivo principal de este trabajo es desarrollar redes neuronales que sirvan como modelos dinámicos subrrogados de un pantógrafo ferroviario. Para tal fin se tiene que desarrollar **un modelo del pantógrafo computacional que permita evaluar la dinámica de este para generar datos con los que entrenar las redes neuronales**. Los objetivos específicos para alcanzar este objetivo principal son los siguientes:

- Identificar los principales factores que afectan la dinámica del pantógrafo y generar computacionalmente un modelo lineal y no lineal de pantógrafo.

- Diseñar y programar una función en Matlab para la generación de valores aleatorios de fuerza para obtener diferentes sets de valores para el entrenamiento y la validación del modelo.
- Estudiar los distintos tipos de redes neuronales existentes en la literatura científica y técnica para determinar cuál es más adecuada para la resolución del problema planteado. Se valorarán diferentes tipos de redes neuronales en Matlab para comprobar su funcionamiento.
- Entrenar las redes neuronales seleccionadas y probar su capacidad para predecir el comportamiento dinámico del pantógrafo.
- Utilizar las redes neuronales para distintas aplicaciones, predecir la aceleración en función de la fuerza o realizar medidas indirectas en las que la fuerza es obtenido con la información de la aceleración.

Además de los objetivos específicos, este proyecto y su realización está alineado con los objetivos de desarrollo sostenible (ODS) con el objetivo de apostar por la innovación para mejorar la calidad de vida de las personas a la par que reducir las desigualdades y cuidar el medioambiente. A continuación, en la ilustración 7 se muestran todos los ODS:

Ilustración 7 : Objetivos de desarrollo sostenible

1.6 Organización del trabajo

Previamente, se ha explicado de manera breves cuál es la motivación de la que emana este trabajo de fin de máster, así como una breve introducción a las redes neuronales y los modelos subrogados que existen y se pueden emplear en este problema. Además, se has enumerado de manera concisa los objetivos técnicos y el objetivo principal que se alcanzará una vez concluido este documento.

El esquema propuesto para proseguir este estudio tendrá la siguiente estructura:

- Un segundo capítulo explicando de manera más detallada las redes neuronales que se van a utilizar en este trabajo

- Un tercer capítulo dedicado al modelo computacional del pantógrafo, en el que se explicará, los dos modelos (lineal y no lineal).
- Un cuarto capítulo de generación de valores aleatorios de fuerza para obtener un dataset para entrenar y validar las redes, además de la configuración de las redes neuronales optimizadas.
- Un quinto capítulo de resultados, analizando como se han comportado las redes comparadas y concluyendo cual ha funcionado mejor. En este capítulo se mostrarán las diferencias entre el pantógrafo lineal y no lineal.
- Un sexto apartado de conclusiones con una tabla comparativa del error.
- Cuatro anexos, que son los siguientes:
  - Presupuesto del proyecto.
  - Código de Matlab desarrollado.
  - Pliegue de condiciones.
  - Alineación del proyecto con los objetivos de desarrollo sostenible ya presentados en los objetivos.

2. Redes Neuronales

Las redes neuronales que más interesantes resultan para el objetivo de este proyecto son las *feedforward* y las recurrentes.

Las redes neuronales *feedforward* son una de las arquitecturas más simples utilizadas en el procesamiento de datos. También se les conoce como redes neuronales de propagación hacia adelante. La estructura básica de una red *feedforward* consiste en una serie de capas de neuronas, dispuestas en una secuencia desde la capa de entrada hasta la capa de salida, y en las que las señales fluyen en una única dirección, es decir, desde las capas de entrada hacia las capas de salida.

Las neuronas en la capa de entrada reciben los datos de entrada y las neuronas de la capa de salida producen las salidas. Las capas ocultas, actúan como procesadores de la información, utilizando los pesos que conectan las señales de entrada y salida de las neuronas para generar señales de salida más complejas, en la ilustración 8 se puede ver su arquitectura.

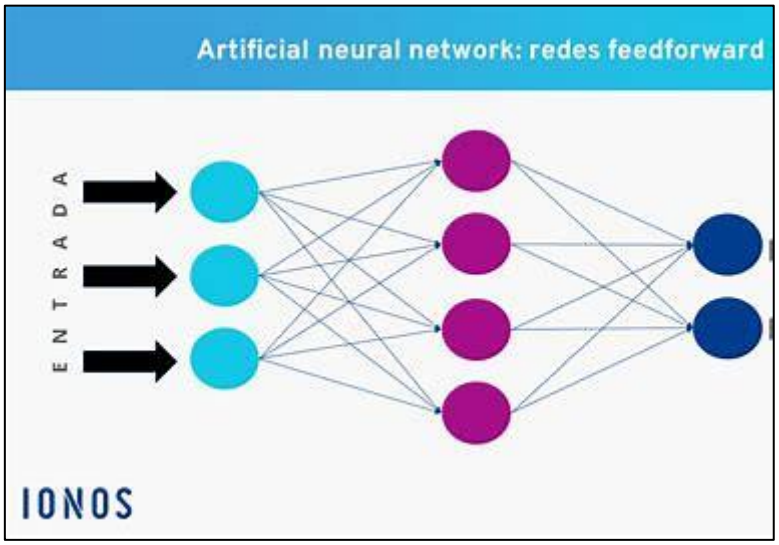

Ilustración 8: : Arquitectura feedforward

En general, una red *feedforward* puede tener desde una sola capa oculta hasta varias capas ocultas. El uso de múltiples capas ocultas permite que la red pueda aprendizaje características complejas y no lineales de la entrada.

Este tipo de arquitectura de redes utiliza la propagación hacia delante, se utiliza la función de coste, la entrada y los parámetros a modificar para obtener el MSE. La expresión matemática del cálculo de cada neurona es la siguiente: $Z = W_1 \cdot X_1 + W_2 \cdot X_2 + b$

A continuación, se describen algunos tipos de arquitectura de redes neuronales *feedforward*:

- **Redes neuronales de una sola capa (SLFN)**: Las redes neuronales de una sola capa, también conocidas como perceptrones multicapa, son las redes *feedforward* más simples. Una sola capa oculta es suficiente para la mayoría de las aplicaciones, pero no para todas. Este tipo de redes son óptimas para aplicaciones de clasificación de patrones y pronósticos sencillos, tales como la predicción de precios, la clasificación de imágenes y reconocimiento de voces.
- **Redes neuronales de múltiples capas (MLFN):** Las redes multilayer, también conocidas como redes neuronales de múltiples capas, tienen una capa oculta más grande que una sola capa. La idea detrás de un MLFN es construir una arquitectura de red neuronal que pueda resolver problemas no lineales. Los resultados en este tipo de redes son más precisos y adaptables a cualquier tipo de problema, pero llevan asociado mayor coste computacional.
- **Redes neuronales con funciones de activación especiales**: En una red neuronal *feedforward* convencional, las neuronas suelen tener una función de activación lineal de forma predeterminada. Sin embargo, también existen otras funciones de activación especiales que se explicaron previamente en el estado del arte, estas son las funciones ReLU o sigmoide entre otros.

Las redes neuronales recurrentes son una arquitectura de redes neuronales que se especializan en la manipulación de datos secuenciales. A diferencia de las redes *feedforward*, las redes neuronales recurrentes tienen conexiones hacia atrás, es decir, las señales tienen la capacidad de retroceder y ser utilizadas nuevamente en la red. Esto les permite tener memoria a corto plazo y aprender patrones en datos secuenciales.

La idea central detrás de una red neuronal recurrente es la construcción de un loop, en el cual las salidas previas de la red se utilicen como entradas futuras en esta. Las neuronas en una capa oculta en una red neuronal recurrente actúan no solo como procesadores de información, sino también como "almacenamiento temporal" que almacena información del pasado.

Existen varias arquitecturas de redes neuronales recurrentes, pero una de las más utilizadas es la arquitectura de red neuronal de memoria de corto plazo (LSTM, por sus siglas en inglés). Las LSTM se utilizan a menudo en el procesamiento del habla natural, en el análisis de texto y en el modelado de secuencias. En la ilustración 9 se comparan estas arquitecturas.

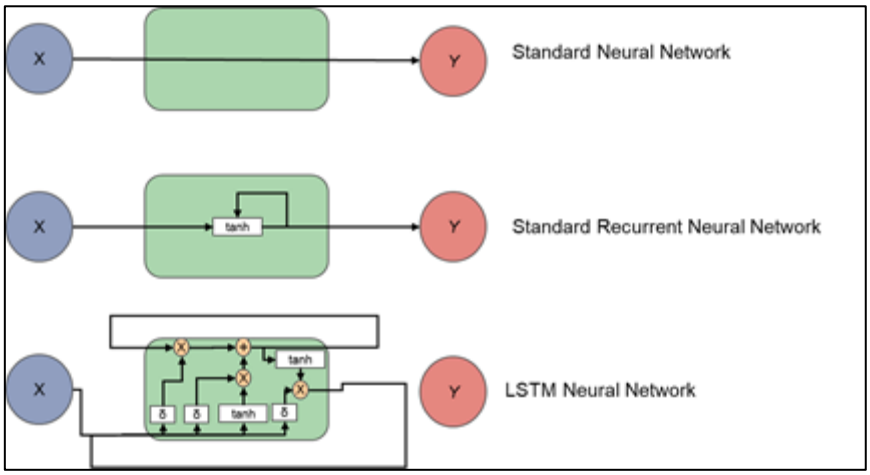

Ilustración 9 Comparación entre redes neuronales recurrentes y redes neuronales estándar

Además, es importante destacar que, debido a la naturaleza temporal de los datos, las redes neuronales recurrentes tienen una capacidad única para modelar patrones de datos que cambian con el tiempo. Debido a esto, se utilizará una red neuronal recurrente bidireccional (BRNN) en este proyecto, prácticamente igual a la LSTM solo que las neuronas dan información hacia delante y hacia atrás. Estas serán las tres redes que se utilizarán.

Sin embargo, y pese a que las redes recurrentes sobre el papel parezcan claramente superiores, también es importante tener en cuenta que las redes neuronales recurrentes pueden ser más difíciles de entrenar que las *feedforward* debido a la complejidad de sus conexiones temporales. También tienen un mayor costo computacional y pueden requerir más datos para su entrenamiento.

3. Modelo computacional del pantógrafo

Debido a la situación de no existir un conjunto de datos iniciales para el entrenamiento y validación de las redes neuronales, se programará en software MATLAB modelos de pantógrafo que serán excitados por señales de fuerzas pseudoaleatorias para general el necesario dataset.

En este capítulo se van a generar los dos modelos computacionales del pantógrafo, el lineal y el no lineal. Estos se emplearán posteriormente para entrenar las redes neuronales que constituyes el modelo subrogado.

El proceso de generación del modelo computacional se realizará mediante integración numérica, con el fin de obtener un conjunto de valores de desplazamiento y aceleración para realizar las predicciones una vez programadas las redes.

Se ha simplificado el sistema para facilitar su manejo como se ve en la ilustración 10, ya que un pantógrafo es un sistema bastante complejo. Para ello se ha descompuesto el pantógrafo en tres bloques, cada uno de ellos con un valor de masa, rigidez y amortiguamiento. Cabe destacar que este es el modelo lineal, una vez se haya obtenido se añadirá la fricción de Coulomb.

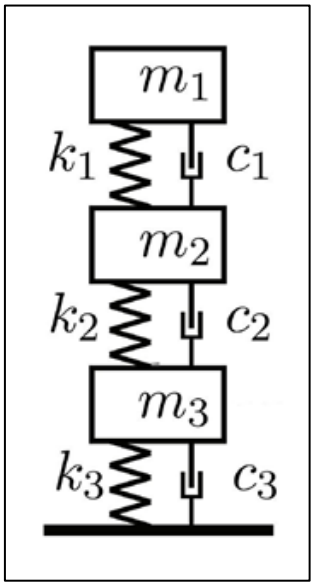

Ilustración 10: Descomposición del problema en bloques

Estos bloques tienen asociados unos valores que son los siguientes expuestos en la tabla 2

Tabla 1: Variables de los bloques

Variable	*valor*
M1	8,5
M2	9

M3	1,31
K1	7000
K2	2200
K3	0,1
C1	30
C2	0
C3	13,10

La masa está en kg, la rigidez en N/m y el amortiguamiento en Ns/N. Una vez tenemos estos valores se introducen en forma de matriz en el software para proceder a la integración. Se ha valorado el empleo de diferentes modos de integración, pero se ha terminado por elegir **Newmark [7].**

El método Newmark-beta es un método numérico de integración utilizado para la resolución de ecuaciones diferenciales. Su aplicación principal y más extendida es en el campo de respuestas dinámicas de estructuras y sólidos, especialmente en análisis de elementos finitos para modelar sistemas dinámicos. Este método fue desarrollado por Nathan Newmark en 1959. Se parte de la ecuación de equilibrio dinámico que es la siguiente:

$$MU"(t + \Delta t) + CU'(t + \Delta t) + KU(t + \Delta t) = f(t + \Delta t)$$

Donde $KU(t + \Delta t)$ son las fuerzas internas por unidad de desplazamiento y $f(t + \Delta t)$ las fuerzas externas por unidad de desplazamiento. Utilizando el teorema del valor medio extendido, el método de Newmark-β establece que la primera derivada temporal (velocidad en la ecuación de movimiento) se puede resolver de la siguiente manera:

$$u'n + 1 = u´n + \Delta t * u´´y$$

Donde:

$$u´´y = (1 - y)u´´n + y * u´´n + 1 \;\; 0 \leq y \leq 1$$

por tanto:

$$u´n + 1 = u´n + (1 - y)\Delta t * u´´n + y\Delta t * u´´n + 1$$

Extendiendo a la segunda derivada ya que la aceleración también depende sustancialmente del tiempo, para obtener apropiadamente el desplazamiento:

$$u´´\beta = (1 - 2\beta)u´´n + 2\beta * u´´n + 1 \; 0 \leq 2\beta \leq 1$$

La ecuación estructural discretizada tiene la siguiente forma:

$$Mu´´n + 1 + Cu´n + 1 + f_{int}(un + 1) = f_{n+1ext}$$

En el método de Newmark, la respuesta temporal se obtiene mediante la resolución iterativa de las ecuaciones de equilibrio para cada paso de tiempo. A continuación, y después de explicar el tratamiento de las ecuaciones anteriores, se presenta un esquema general de la formulación de diferencia central utilizada en el método de Newmark:

- Se discretiza el tiempo en intervalos de paso de tiempo Δt.
- Se establecen las condiciones iniciales para la posición, velocidad y aceleración.
- Se calculan las fuerzas externas y las fuerzas internas del sistema en el paso de tiempo actual.
- Se actualizan las aceleraciones y velocidades de manera iterativa.
- Se repiten los pasos 3 y 4 para cada paso de tiempo hasta alcanzar el tiempo final deseado.

Cabe destacar que los parámetros β y γ deben ser seleccionados de manera apropiada respecto al ámbito de aplicación para garantizar la estabilidad y precisión del método de Newmark. Los valores típicos son β = 0.25 y γ = 0.5, que corresponden al método Newmark lineal y son los que se emplearán en esta solución.

Una vez explicado en qué consiste el método, se pasa a mostrar como se ha realizado la programación de los scripts para la integración.

Inicialmente se ha creado un primer script al que nos referiremos como "Simular_pantografo.m", en él se han introducido los valores iniciales como se ha explicado previamente, y se ha procedido a la integración de la siguiente forma:

```
%% Integracion del panto
dt = 0.001; % Incremento de tiempo
t = 0:dt:10; % Vector de tiempo
[A, U] = int_panto(M, C, K, F, dt); % Simulación del modelo con la señal de fuerza aleatoria


end

function [A, U] = int_panto(M, C, K, F, dt)
%Integra un modelo de pantografo con fuerza en el gdl 1 y condiciones iniciales nulas

Ngdl = size(M,1);
Nstp = length(F);
```

```
%% PARAMETROS INTEGRACION TEMPORAL----------------------------------------------------------
------------
%- Parámetros de integración (Newmark)
bet = 0.25;
gam = 0.5;

%- Constates de Newmark lineal
b1 = 1/(bet*dt^2);
b2 = -1/(bet*dt);
b3 = 1-1/(2*bet);
b4 = gam*dt*b1;
b5 = 1+gam*dt*b2;
b6 = dt*(1+gam*b3-gam);

%- Inicialización de variables
U = zeros(Ngdl,Nstp);A = zeros(Ngdl,Nstp);
u = zeros(Ngdl,1);
v = zeros(Ngdl,1);
a = zeros(Ngdl,1);

%- Matriz de integración temporal
K_it = K + b4*C + b1*M;

%% bucle principal de integracion
for stp = 1:Nstp

   %- Variables del instante anterior
   Uant = u;
   Vant = v;
   Aant = a;
   %- Vector de fuerzas de las condiciones iniciales
   F_it = M*(b1*Uant-b2*Vant-b3*Aant) + C*(b4*Uant-b5*Vant-b6*Aant);
   %- Vector de fuerzas externas
   F_ext=[F(stp);0;0];

   %- Solución del paso temporal
   u = K_it\(F_it+F_ext);
   U(:,stp) = u;

   %- Actualización de velocidad y aceleración
   v = b4*(u-Uant) + b5*Vant + b6*Aant;
   a = b1*(u-Uant) + b2*Vant + b3*Aant;
   A(:,stp) = a;
end
end
```

Este script tiene como objetivo principal contener el método de integración para que mediante la aplicación de una fuerza externa desde otro script se obtengan resultados de manera clara y ordenada.

El segundo script y último empleado en la generación aleatoria de sistemas dinámicos, responde al nombre de "FuncionPrincipal.m". En este script, se introduce la función que emula como son las fuerzas existentes en la catenaria de un tren. Esta tiene la siguiente forma:

```
function [f] = pseudo_modulated(T, f_max, dt, Amp_f)
% Creates a pseudo ramdon funcion modulated
Narm = round(f_max * T);
t = 0:dt:T; %numero impar
Nstp = length(t);

Fmod = rand(1, Narm) .* linspace(rand, 0.1 * rand, Narm);
Fmod(1) = 0;
Farg = rand(1, Narm) * 2 * pi;
Farg(1) = 0;

F = Fmod .* exp(1i * Farg);
F((Nstp + 1) / 2) = 0;
f = Nstp * ifft([F flip(conj(F(2:end)))]);

f = f .* linspace(rand, rand, Nstp) .* abs(rand + abs(sin(rand * t)));
f = f * Amp_f / max(max(f), -min(f));
end
```

La función F depende de una serie de valores, estos son el periodo, la frecuencia máxima, el paso temporal y la amplitud de la frecuencia. Se han dado respectivamente estos valores, aunque se podrían utilizar los deseados para predecir cualquier tipo de catenaria en caso de necesitar otros.

- Periodo: 10 segundos
- Frecuencia máxima: 50 Hz
- Paso temporal: 0.001 segundos
- Amplitud: 1

Una vez aportados los valores, se realizará la integración llamando al archivo simular_pantografo.m y se guardará un archivo que recibirá el nombre de "datos_pantografo.mat" que contendrá los vectores de fuerza, desplazamiento y aceleración del sistema. Por simplicidad y comodidad, vamos a realizar y mostrar los resultados para la aceleración únicamente. La gráfica queda de la siguiente forma como se muestra en la ilustración 11:

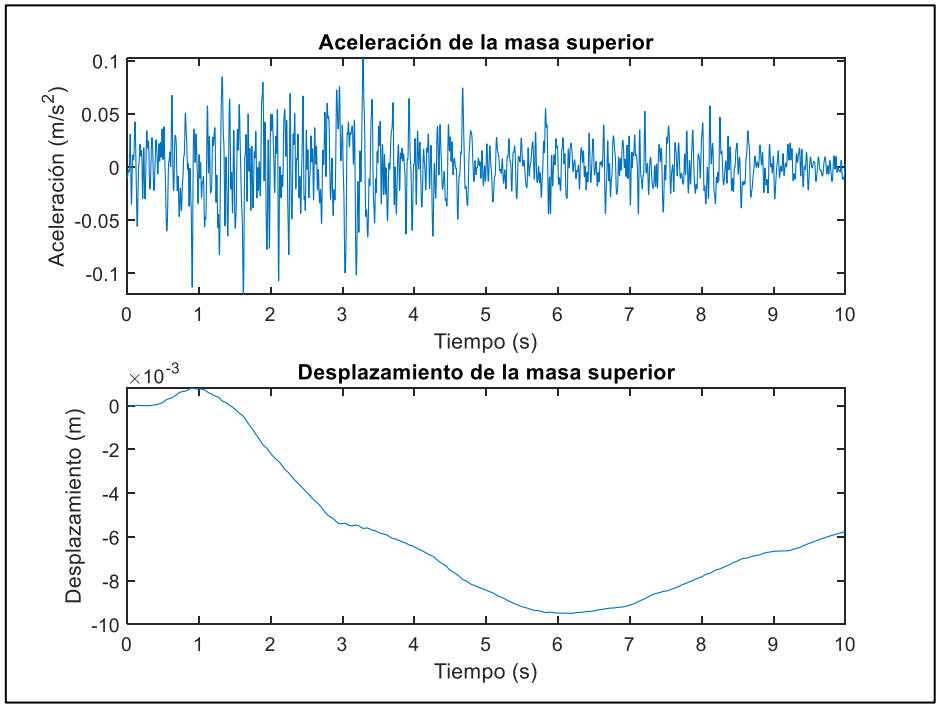

Ilustración 11 : Datos generados aleatoriamente 1

Para demostrar que efectivamente estos sistemas son completamente aleatorios, se vuelve a ejecutar la función y se obtiene la siguiente gráfica, en la ilustración 12, observablemente distinta a la anterior:

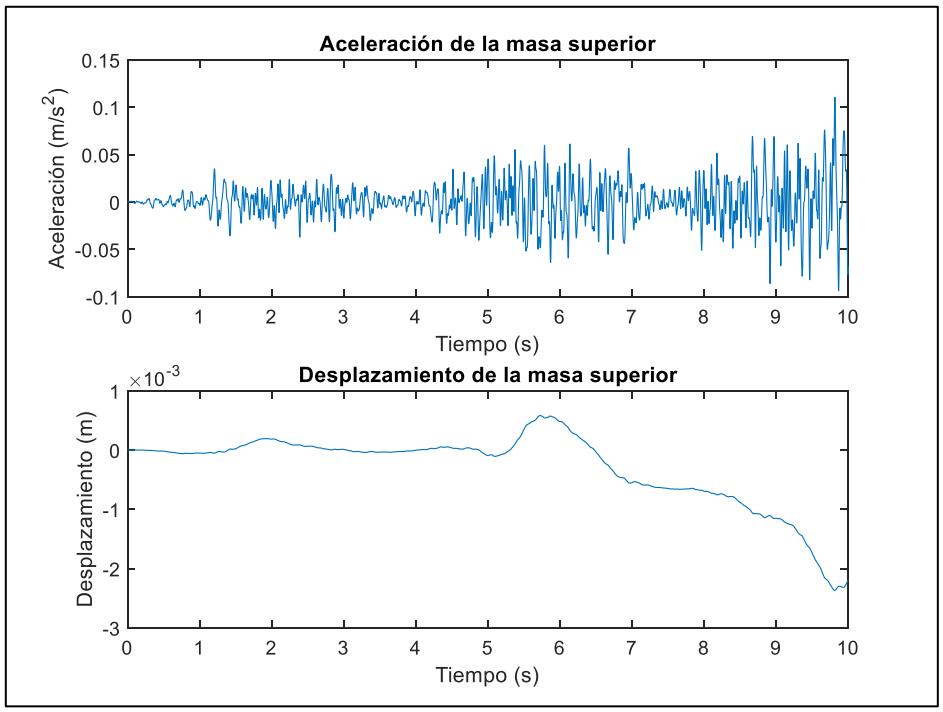

Ilustración 12: Datos generados aleatoriamente 2

Por otro lado, se ha realizado el modelo computacional del pantógrafo añadiendo no linealidades, en este caso mediante el efecto de la fricción seca o ley de Coulomb. Esta es una fuerza que aparece en oposición al movimiento relativo entre dos superficies, en este caso pantógrafo y catenaria. La fuerza de fricción es de la forma:

$$F_{fricción} = \mu \cdot F_{normal}$$

Esta fuerza, se añadió al código anterior de la siguiente forma:

```
%% Parámetros de fricción de Coulomb
    F0 = 2; % Valor máximo de fricción de Coulomb
    v_rel_prev = 0; % Variable auxiliar para regularización

% Fricción de Coulomb
        v_rel = Vant(1) - Vant(2); % Velocidad relativa entre los grados de
libertad 1 y 2
        signo_v = sign(v_rel); % Signo de la velocidad relativa

        % Regularización
        if signo_v == 0
            signo_v = sign(v_rel_prev);
        end
```

Esto produce un cambio en las gráficas de aceleración y desplazamiento que tiene la siguiente forma, como se muestra en la ilustración 13; donde se compara la generación aleatoria del caso lineal y el no lineal:

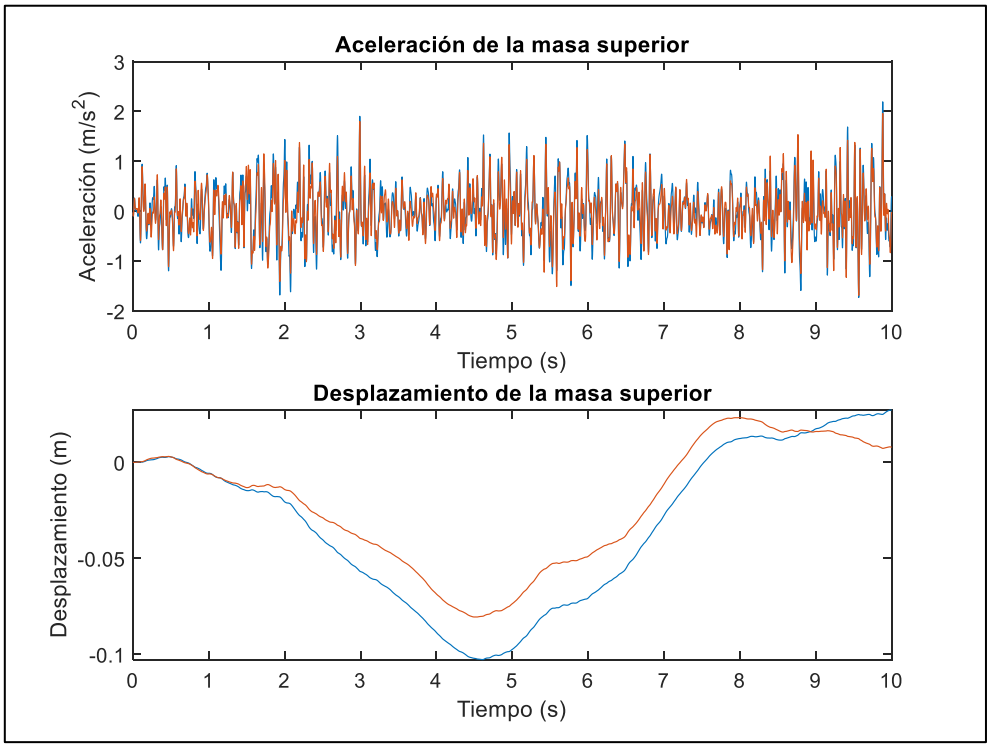

Ilustración 13: Modelo computacional no lineal

4. Entrenamiento de redes neuronales

En este apartado se entrenará, validará y optimizará tres tipos de redes neuronales, estas son:

- Red *feedforward*
- Red LSTM
- Red BRNN

Se ha valorado también emplear una red neuronal convolucional, pero se ha descartado posteriormente la idea ya que este tipo de redes tiene buen funcionamiento en el procesamiento de imágenes, y aquí se está tratando únicamente con datos generados aleatorios. Este tipo de red hubiera tenido más interés en otras aplicaciones como el mantenimiento de la propia catenaria en caso de tener imágenes de defectos para entrenar la red y así poder detectar imperfecciones.

La primera red que hemos entrenado es la Feed Forward, y es de lejos la más simple y la que obtiene resultados menos precisos. Se realizó una primera programación de la red sin tener en cuenta detalles de optimización. La arquitectura inicial tiene la forma de la ilustración 14:

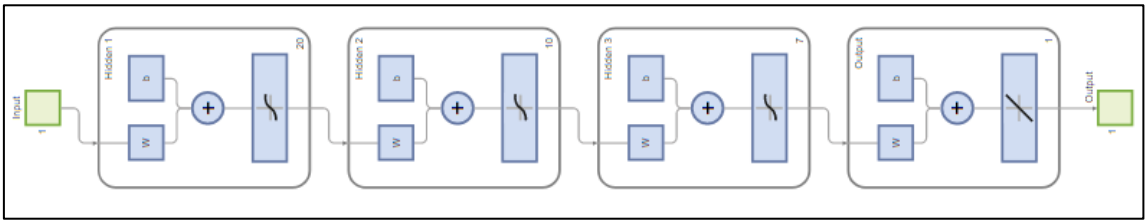

Ilustración 14: : Arquitectura inicial de la feedforward

Como se puede apreciar, la red consta de tres capas ocultas de 20, 10 y 7 neuronas respectivamente. De manera tradicional y siguiendo otros ejemplos que se han encontrado en la bibliografía, se suele dividir el número de datos para el entrenamiento y la validación. Esto suele ser del orden del 70% de los datos aleatoriamente generados para el entrenamiento y el 30% restante para la validación. Sin embargo, en el presente trabajo se ha optado por utilizar todos los datos para el entrenamiento y una vez iniciado el entrenamiento, reintroducir la función de fuerza aleatoria para generar otros 5 nuevos sets de datos para validar la red mediante la media de los 5 errores. Esto se ha realizado del siguiente modo:

```
% Número de conjuntos de validación a generar
num_val_sets = 5;
rmse_sum = 0;

for i = 1:num_val_sets
    % Generar nuevos datos de fuerza y aceleración usando la función de fuerza aleatoria
    F_new = pseudo_modulated(10, 50, 0.001, 1);
```

```
    [A_new, U_new] = simular_pantografo(F_new);

    % Preparar los nuevos datos para la validación
    X_val = F_new';
    Y_val = A_new(1, :)';

    % Probar la red neuronal con los nuevos datos de fuerza
    predictions = net(X_val');

    % Calcular el error cuadrático medio (RMSE)
    rmse = sqrt(mean((predictions - Y_val').^2));
    rmse_sum = rmse_sum + rmse;
    fprintf('RMSE for validation set %d: %.4f\n', i, rmse);
end

% Calcular el RMSE promedio
rmse_avg = rmse_sum / num_val_sets;
```

Los resultados de este primer entrenamiento se muestran en la ilustración 15:

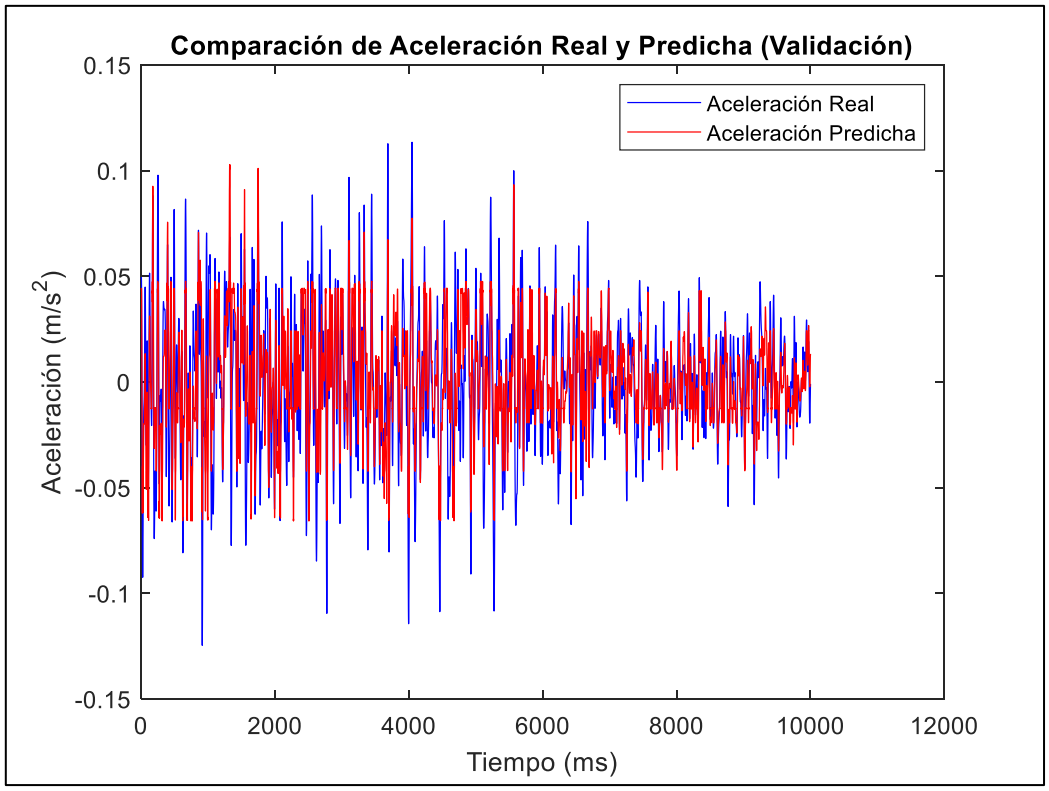

Ilustración 15: Primera iteración de la feedforward

Como se puede observar, los resultados no son muy precisos sobre todo cuando las aceleraciones presentan picos. Por ello, se procede a optimizar la red modificando su estructura. Cabe destacar que los errores variarán al volver a ejecutar el código debido

a como se generan nuevos datos cada vez; esto también ocurría en el estado del arte cuando se explicaron las soluciones similares. El RMSE de esta red es de un valor de X.

La primera modificación que se ha realizado para mejorar la precisión en la predicción de la red ha sido modificar su arquitectura. Después de probar diferentes combinaciones se ha aumentado el número de capas y neuronas a una combinación de 100, 50, 30 y 20 neuronas respectivamente. Mediante esto, se ha reducido el RMSE a un valor de 0,0169.

Como el error sigue siendo grande, se ha optado como siguiente paso realizar una variación en la tasa de aprendizaje. Al probar a variar la tasa de aprendizaje se observa que el error no varía significativamente y en algún caso aumenta; con lo cual el siguiente paso será probar una regularización [4].

Se ha optado por una regularización L2, también llamada dropout. Se ha agregado una capa de dropout a la primera capa oculta con una tasa de desactivación del 20%. Esto ayudará a regularizar la red y potencialmente mejorar la precisión. Esto ha conseguido una ligera reducción de los valores de RMSE.

Finalmente, se ha optado por variar los parámetros de entrenamiento, el número de capas en aquellos que converjan a una solución más rápido y el mínimo RMSE que se es capaz de obtener es de 0,016. Este valor se da para la siguiente arquitectura, mostrada en la ilustración 16:

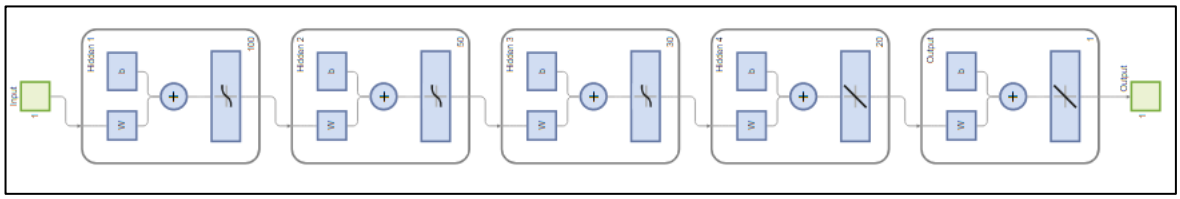

Ilustración 16: Arquitectura óptima de la red

El método de entrenamiento óptimo es el Levenberg-Marquardt, pese a necesitar más tiempo de simulación, proporciona un resultado de 0,016 como se ha dicho antes, frente a los 0,02 de media que dan otros resultados. Así predice la red neuronal *feedforward* optimizada, la gráfica se muestra en la ilustración 17.

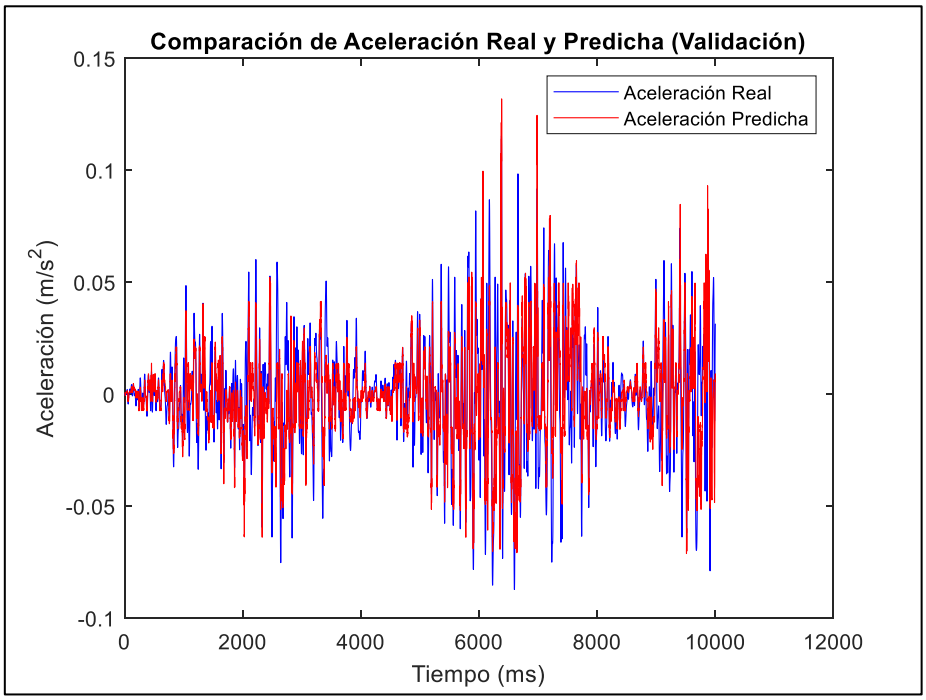

Ilustración 17: Predicción de la red FeedForward optimizada

Como alternativa, proponemos utilizar solo un número reducido de puntos de la fuerza para predecir la aceleración de un instante dado. Se ha programado este tipo de red con otro código que entrena los valores de la red de manera diferente. La principal diferencia es que se utiliza la variable **N** para representar la dimensión de la función de entrada y se utiliza para determinar cuántos pasos hacia atrás se utilizarán para predecir el valor actual. El código empleado es el siguiente:

```
% Cargar los datos del archivo .mat
load('datos_pantografo.mat', 'F', 'A', 'U');

%Dimension de la funcion de entrada (numero de stps necesarios para predecir
el actual)
N=20;

% Preparar los datos para el entrenamiento
M=size(F,2);
%matriz de fuerzas
ind=(1:M-N+1)+(0:N-1)';
X=F(ind);
Y=A(1, N:end);


% Crear y configurar la red neuronal
hiddenLayerSize = [50, 30, 20];
net = fitnet(hiddenLayerSize);

% Entrenar la red neuronal
[net, tr] = train(net, X, Y);
view(net);

%Evaluacion del entrenamiento
```

```
figure
plot(Y);hold on;plot(net(X))

%% Validation
% Número de conjuntos de validación a generar
num_val_sets = 1;
rmse_sum = 0;

for i = 1:num_val_sets
     % Generar nuevos datos de fuerza y aceleración usando la función de
fuerza aleatoria
     F_new = pseudo_modulated(10, 50, 0.001, 1);
     [A_new, U_new] = simular_pantografo(F_new);


     M=size(F_new,2);
     %mariz de fuerzas
     ind=(1:M-N+1)+(0:N-1)';
     X_val=F_new(ind);
     Y_val=A_new(1, N:end);



     % Probar la red neuronal con los nuevos datos de fuerza
     predictions = net(X_val);

     % Calcular el error cuadrático medio (RMSE)
     rmse = sqrt(mean((predictions - Y_val).^2));
     rmse_sum = rmse_sum + rmse;
     fprintf('RMSE for validation set %d: %.4f\n', i, rmse);
end

% Calcular el RMSE promedio
rmse_avg = rmse_sum / num_val_sets;

% Mostrar el RMSE promedio
fprintf('Average RMSE: %.4f\n', rmse_avg);

% Graficar resultados
figure;
plot(Y_val, 'b');
hold on;
plot(predictions, 'r');
legend('Aceleración Real', 'Aceleración Predicha');
xlabel('Tiempo (ms)');
ylabel('Aceleración (m/s^2)');
title('Comparación de Aceleración Real y Predicha (Validación)');

function [f] = pseudo_modulated(T, f_max, dt, Amp_f)
   % Creates a pseudo random function modulated
   Narm = round(f_max * T);
   t = 0:dt:T; %numero impar necesario
   Nstp = length(t);

   Fmod = rand(1, Narm) .* linspace(rand, 0.1 * rand, Narm);
   Fmod(1) = 0;
   Farg = rand(1, Narm) * 2*pi;
   Farg(1) = 0;
```

```
    F = Fmod .* exp(1i * Farg);
    F((Nstp + 1) / 2) = 0;
    f = Nstp * ifft([F flip(conj(F(2:end)))]);

    f = f .* linspace(rand, rand, Nstp) .* abs(rand + abs(sin(rand * t)));
    f = f * Amp_f / max(max(f), -min(f));
end
```

Al realizar diferentes simulaciones buscando su optimización, el mejor resultado que se ha encontrado se muestra en la ilustración 18:

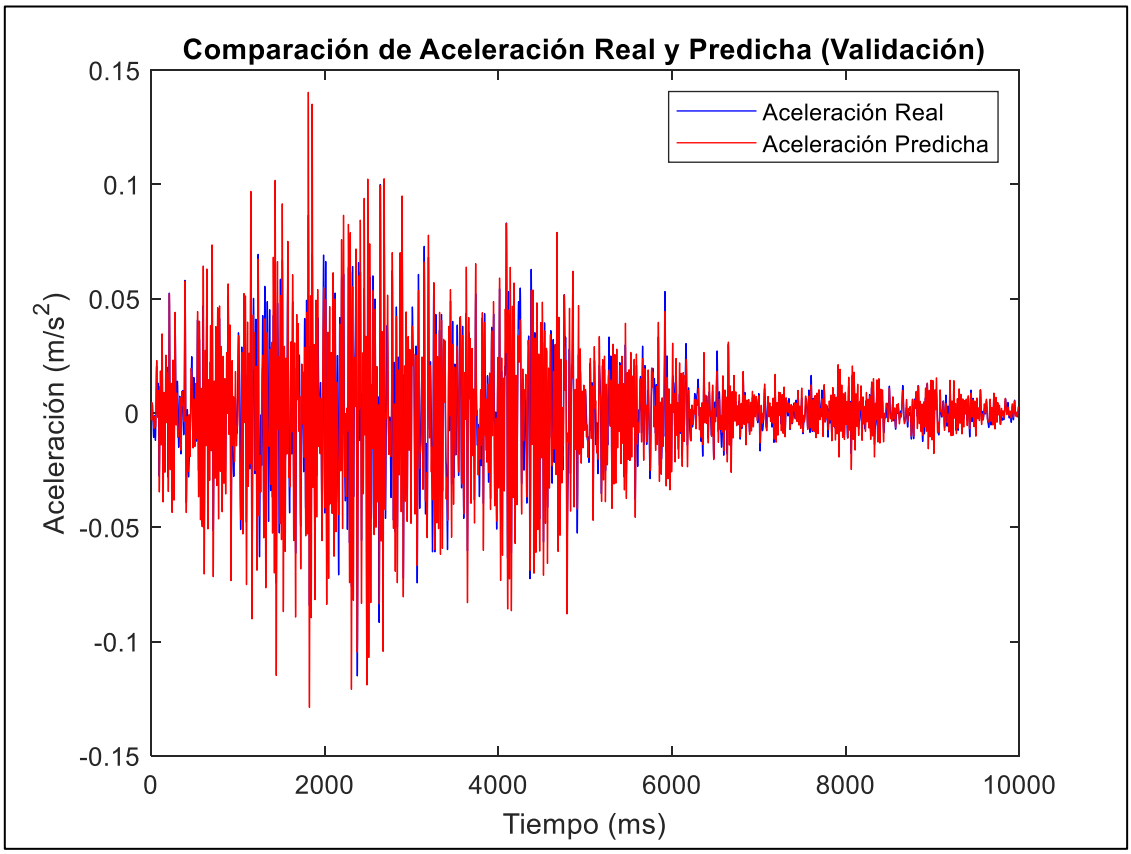

Ilustración 18: Predicción feed forward con el nuevo código (optimización final)

Como se puede comprobar, la red funciona mucho mejor que la anterior *feedforward* para la arquitectura óptima; reduciendo su RMSE de 0,016 a 0,009. Con lo cual, la red *feedforward* final que se usará de referencia en este trabajo es la segunda presentada.

Una vez comprobada la eficiencia de la red para un modelo lineal, se realiza el modelo subrogado del modelo con el efecto de la fricción seca, y este es el resultado que se ha obtenido, mostrado en la ilustración 19:

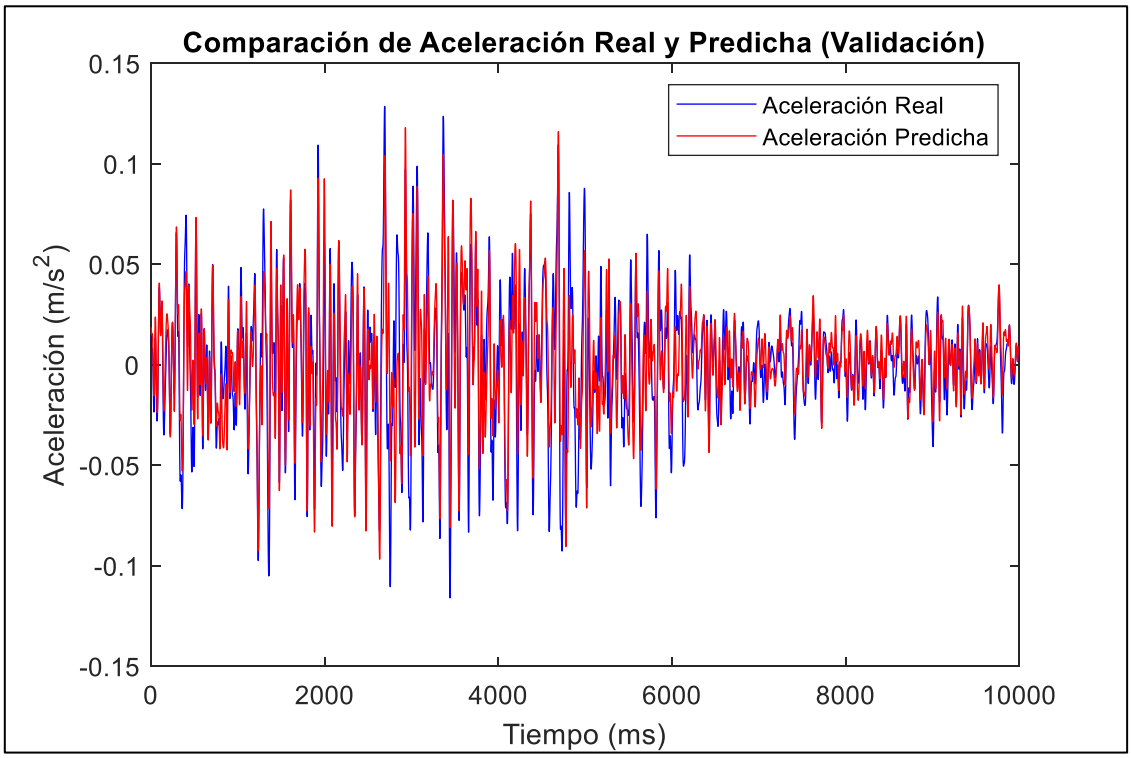

Ilustración 19: Predicción feedforward del modelo no lineal

El RMSE es de 0,013; un valor ligeramente superior al del modelo lineal pero aun así mucho mejor que el obtenido en la primera configuración de red.

La segunda y tercera red seleccionada para su entrenamiento son redes neuronales recurrentes. Esto se debe a que, observando el estado del arte, son las más adecuadas para tratar este tipo de problemas con fuerte dependencia en el tiempo. La primera es una red LSTM, es decir, una red que funciona con memoria a corto plazo que a diferencia de la anterior utiliza retroalimentación entre sus conexiones para aproximar de manera más precisa la solución.

La arquitectura inicial de la red LSTM que se ha empleado para una primera aproximación se muestra en la ilustración 20:

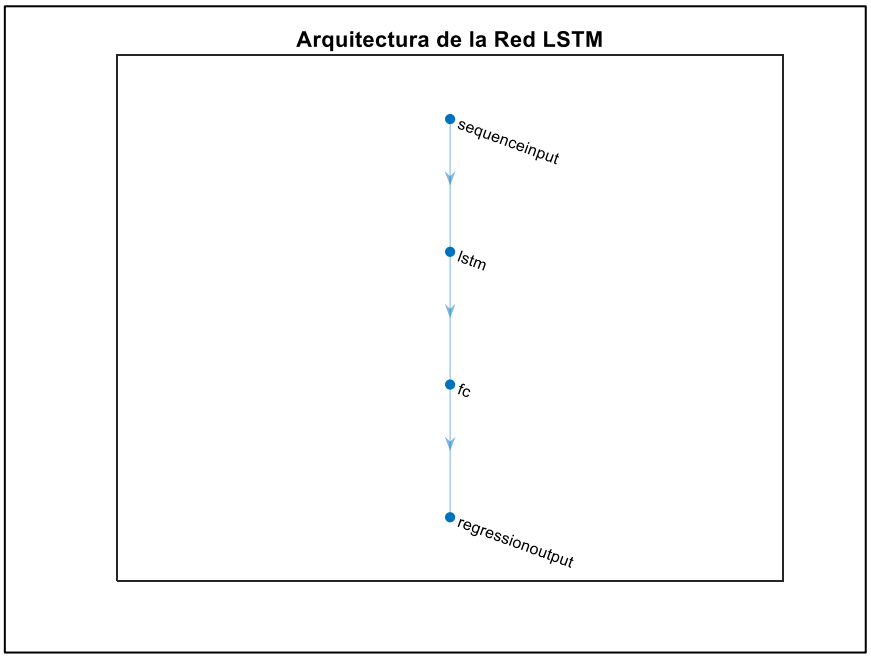

Ilustración 20: Arquitectura inicial de la LSTM

Para programar la red, se han ajustado los datos de entrada y salida para que tengan el mismo tamaño de la siguiente forma:

```
% Cargar los datos del archivo .mat
load('datos_pantografo.mat', 'F', 'A', 'U');

% Preparar los datos para el entrenamiento
X = F';
Y = A(1, :)';

% Cambiar la forma de los datos para cumplir con los requisitos de entrada de una RNN
numFeatures = 1;
X = reshape(X, [numFeatures, size(X, 1), size(X, 2)]);
Y = reshape(Y, [numFeatures, size(Y, 1), size(Y, 2)]);
```

Inicialmente como se puede apreciar, solo se tiene una capa oculta con 25 neuronas a la cual se le aplican las siguientes condiciones de entrenamiento:

```
options = trainingOptions('adam', ...
    'ExecutionEnvironment', 'cpu', ...
    'MaxEpochs', 200, ...
    'MiniBatchSize', 128, ...
    'GradientThreshold',1, ...
    'InitialLearnRate',0.001, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropPeriod',125, ...
    'LearnRateDropFactor',0.2, ...
```

```
'Verbose',0, ...
'Plots','training-progress');
```

Las anteriores líneas de código corresponden respectivamente a lo siguiente:

- **'Adam'** es un algoritmo de entrenamiento que se emplea en este caso para la red al ser muy popular debido a sus buenas condiciones de optimización.
- Las **'MaxEpochs'** son el número de iteraciones completadas para el conjunto de datos durante su entrenamiento. Se ha fijado en 200 porque en una primera aproximación se observaba que con 100 epochs la red no había terminado de converger a un valor, pero se modificará según la necesidad posteriormente.
- El **'MiniBatchSize'** hace referencia al tamaño del minilote que se ha utilizado en el presente entrenamiento. Un minilote consiste en un subconjunto de menor tamaño de muestras de entrenamiento que se procesan en conjunto antes de realizar una actualización de los pesos de la red. Inicialmente se ha escogido un tamaño de minilote de 128.
- El **'GradientTreshold'** se establece como umbral para el recorte del gradiente en caso de que el gradiente de cualquier parámetro de esta red supera el valor de la unidad. Su función consiste en evitar gradientes demasiado altos que provoquen la explosión de este.
- **'InitialLearnRate'** es la tasa de aprendizaje inicial. Este factor se multiplica por los gradientes para la posterior actualización de los pesos de la red durante la etapa de entrenamiento. Es uno de los factores más importante a tener en cuenta, ya que optimizar este valor permitiría tener un valor adecuado, ni demasiado permisivo ni demasiado restrictivo que termine con un sobreajuste de la red.
- **'LearnRateSchedule'** es el tipo de tasa de aprendizaje que se utilizará. En este caso se ha elegido el Piecewise, que es una tasa de aprendizaje por pasos que se ajustará en diferentes puntos durante el entrenamiento, reduciendo la tasa de aprendizaje en momentos determinados.
- **'LearnRateDropPreiod'** es un indicador de la frecuencia con la que se hará una reducción de la tasa de aprendizaje durante el entrenamiento respecto al número de epochs. Se escoge el valor 125, que quiere decir que la tasa de aprendizaje se reducirá a partir de las 125 epochs.
- **'LearnRateDropFactor'** Es un indicador del factor de reducción de la tasa de aprendizaje (En este caso ocurre a partir de las 125 epochs). El valor 0,2 indica que a partir de 125 epochs la tasa de aprendizaje se reduce en un 20%.
- **'Verbose'** se utiliza para el control de la información durante el entrenamiento, en este caso 0, que significa que hay desactivación de información detallada.

A continuación, en la ilustración 21, se muestra como transcurre este primer entrenamiento por épocas:

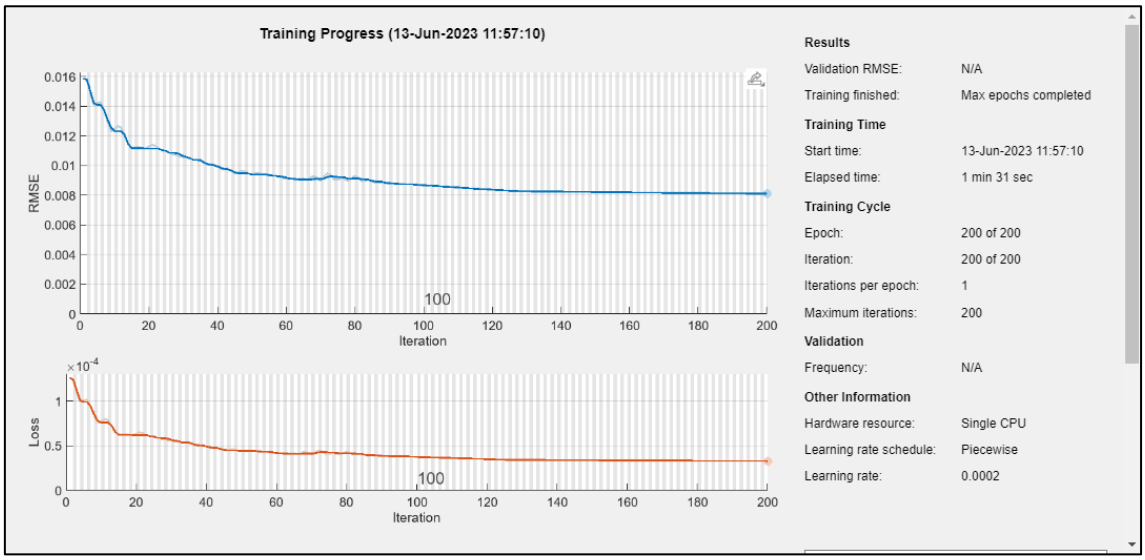

Ilustración 21: Evolución del RMSE durante el primer entrenamiento LSTM

Como se puede observar, el valor del RMSE es muchísimo menor al que se obtiene en las redes *FeedForward*; En este caso es de 0,01. Eso es exactamente La mitad de error que el que se producía al principio de la red *FeedForward* sin optimizar. A continuación, la gráfica que contrasta la aceleración real con la predicha por la red se muestra en la ilustración 22:

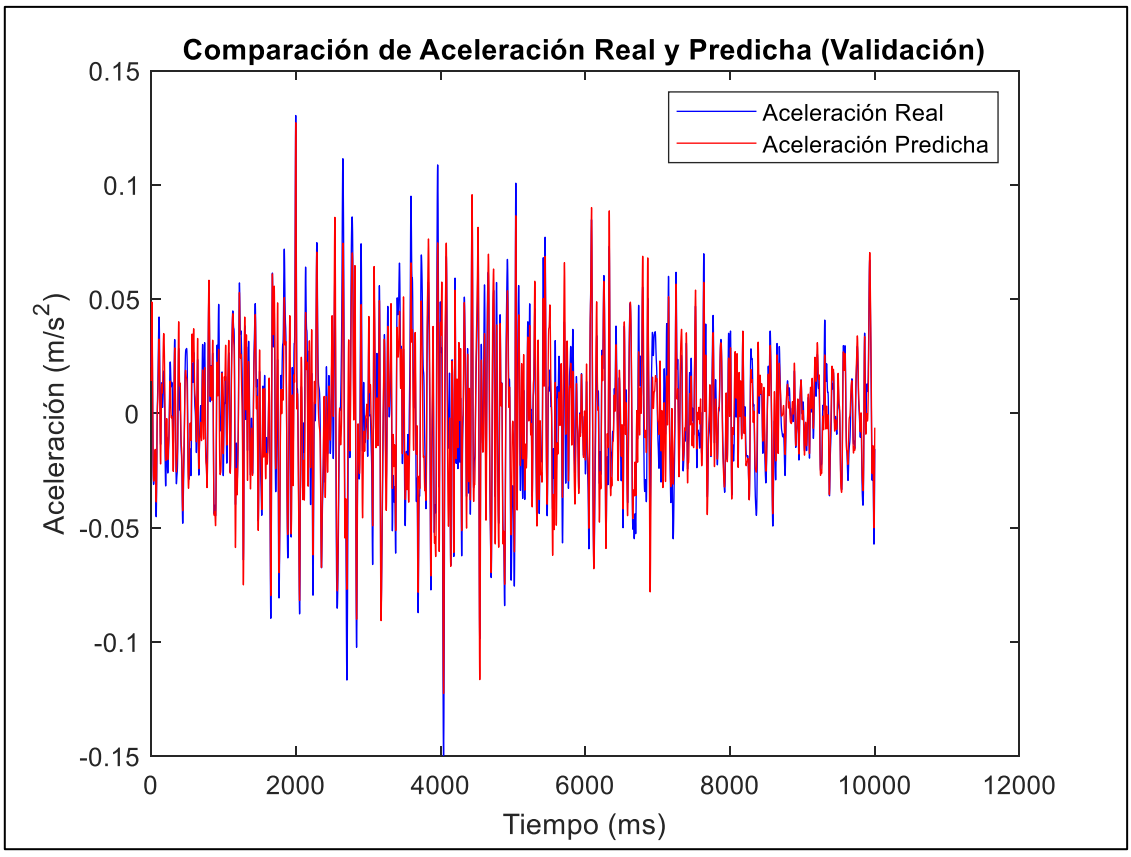

Ilustración 22: Primera iteración de la red LSTM

Para optimizar esta red se ha procedido a variar los parámetros de la red. Lo primero que se ha hecho es aumentar el número de capas modificando así la arquitectura de la red; esto ha conseguido aumentar el error hasta duplicarlo, lo cual hace obvio que con una capa se obtienen buenos resultados.

Se ha probado a realizar el entrenamiento con 10 combinaciones de neuronas y se ha observado que desde 25 hasta 100 neuronas los resultados empiezan a mejorar sustancialmente a partir de 50 neuronas, con lo cual la decisión es tomar 50 ya que la variación de precisión respecto a usar 50, 75, 80 0 100 neuronas es completamente inapreciable. En estos momentos del proceso iterativo el valor del RMSE es de 0,004; que es un valor muy bueno en contraposición con los que se obtuvieron hasta ahora en las redes *FeedForward*. A continuación, en la ilustración 23, se muestra el funcionamiento de la red frente a los valores reales:

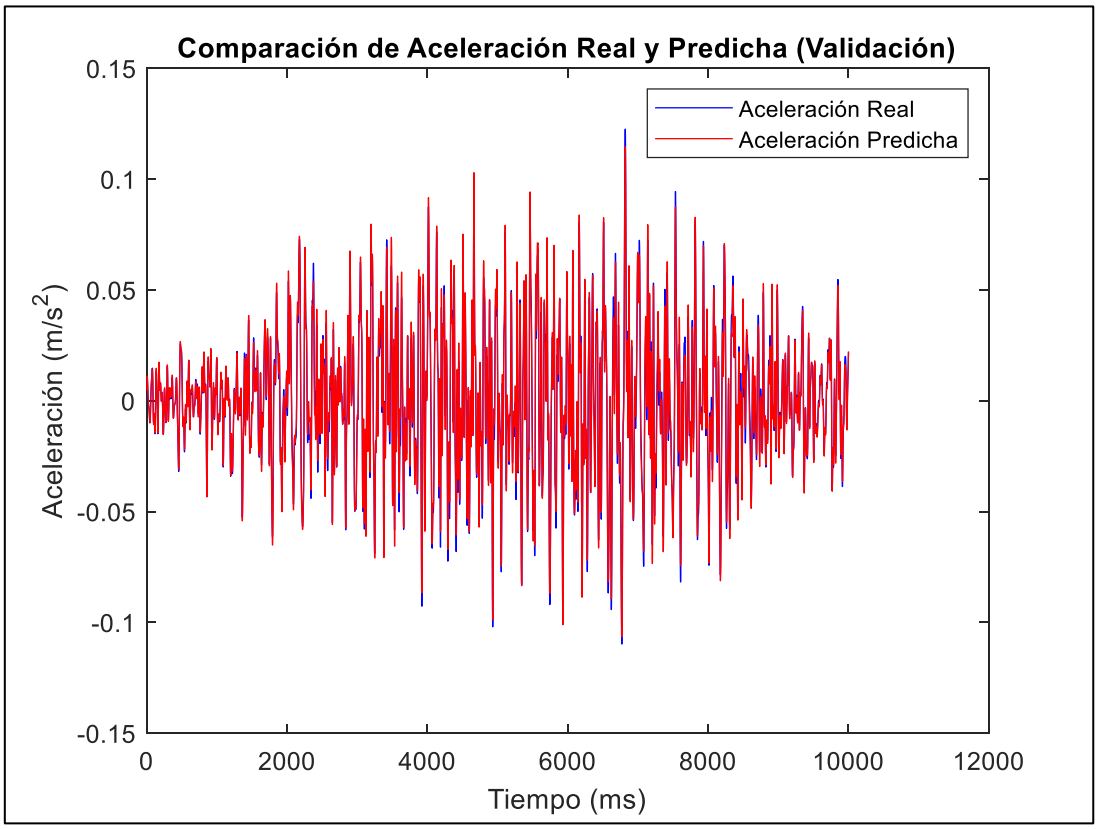

Ilustración 23: Primera iteración en el proceso de optimización de la red LSTM

El siguiente parámetro modificado fue el MiniBatchSize, a un valor de 32 se encontraron los mejores resultados reduciendo a 0,0036 el RMSE. Prosiguiendo, se han estudiado diferentes algoritmos de optimización, estos han sido:

- Adam
- Rmsprop
- Sgdm

El que mejores resultados ha proporcionado ha sido el Adam, seguido del Rmsprop; con lo cual se ha decidido continuar la optimización con este primero. A partir de aquí, mejorar el resultado de 0,0036 se está complicando bastante, se ha probado a usar un dropout del 20%, a regular los pesos (Esto no ha sido útil ya que en Matlab la optimización mediante Adam no admite regularlos directamente, y el descenso de gradiente estocástico con momento se comporta mucho peor en este tipo de problema) y finalmente se prueba con el aumento de los epochs hasta un número de 1000.

Con esto obtenemos el mejor resultado posible, tenemos un RMSE de 0,003 que es el mínimo que se ha logrado en este estudio, en la imagen que se muestra a continuación se puede observar claramente que la red predice de manera muy precisa el comportamiento del modelo. En la ilustración 24 se muestra el modelo LSTM optimizado.

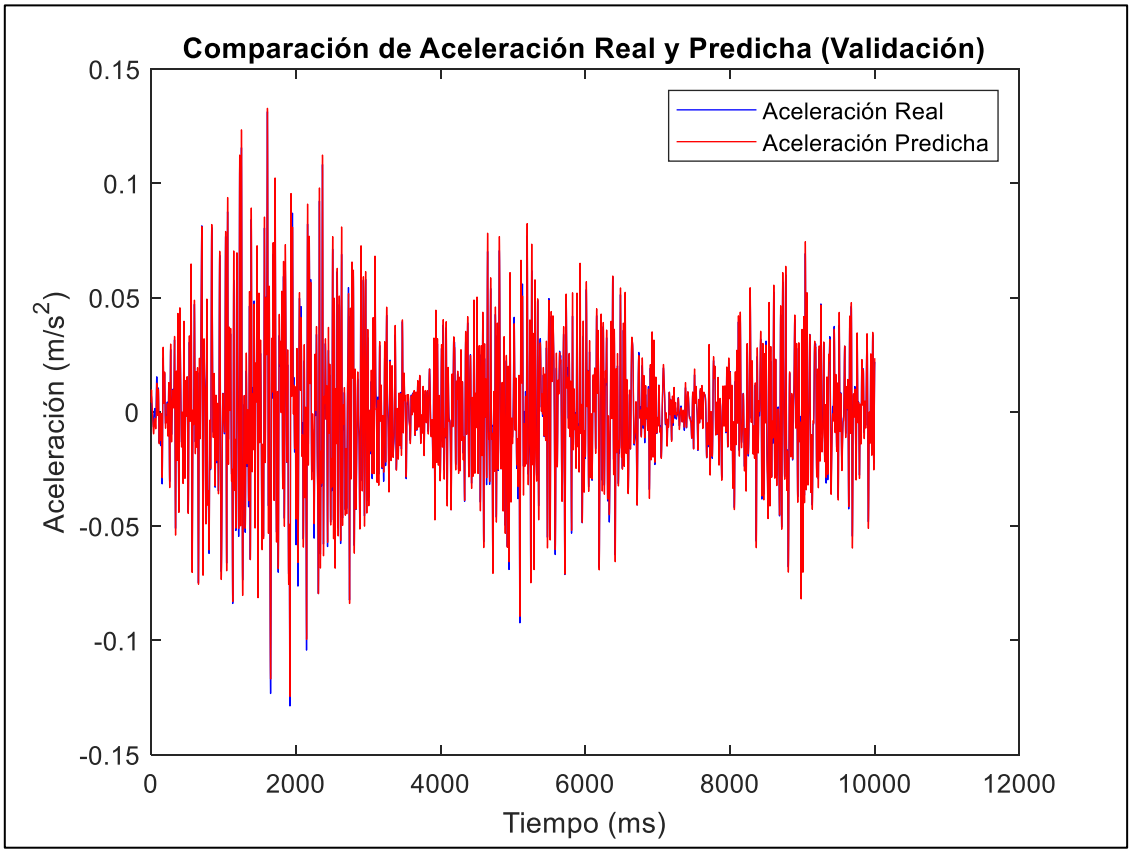

Ilustración 24: Modelo LSTM optimizado

De manera análoga a la red anterior, se prueba la red neuronal para el modelo con el efecto de la fricción seca, mostrado en la ilustración 25. El mismo tipo de arquitectura y de parámetros de entrenamiento se utilizó para el modelo, sin embargo, los valores de predicción obtenidos fueron mucho peores que los del modelo lineal, aumentando el error notablemente hasta 0,0095.

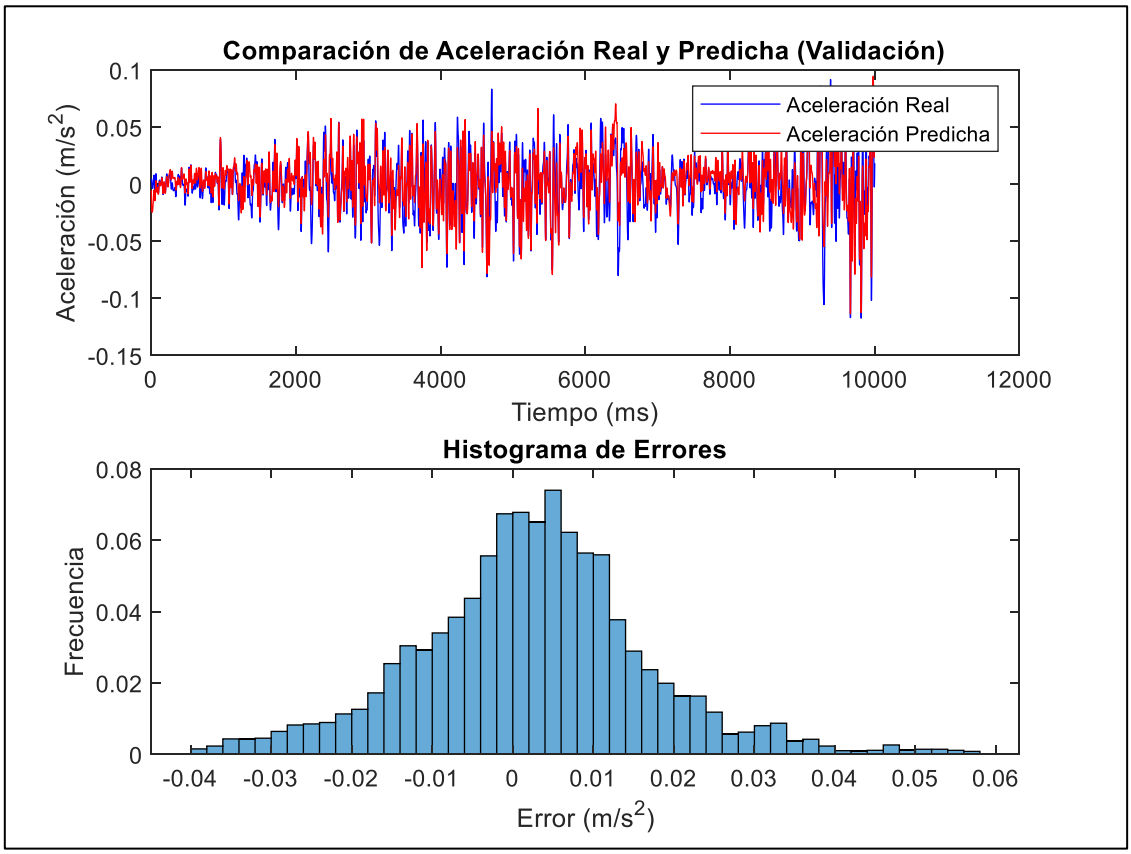

Ilustración 25: Predicción LSTM del modelo no lineal

En la gráfica del histograma, la frecuencia se encuentra en el eje y. La frecuencia representa el número de ocurrencias de cada rango de errores. Cada barra en el histograma muestra la cantidad de veces que un error específico aparece en el conjunto de datos de validación.

La última red utilizada es una red BRNN, que es una variante de las redes LSTM. El procedimiento de programación inicial de la red ha sido parecido al de la red LSTM por ser altamente similar. La diferencia entre una red LSTM y una BRNN es que esta última utiliza paralelamente dos redes LSTM, una en orden directo y otra en orden inverso. Esto significa que se utiliza la recurrencia tanto adelante como para atrás, permitiendo contextualizar los puntos de la secuencia de manera precisa.

Inicialmente se han cargado los datos de entrenamiento y se han acomodado para su correcto procesamiento de la siguiente manera:

```
% Cargar los datos del archivo .mat
load('datos_pantografo.mat', 'F', 'A', 'U');

% Preparar los datos para el entrenamiento
X = F';
Y = A(1, 1:length(X))';

trainLength = round(0.6 * length(Y));
```

```
valLength = round(0.2 * length(Y));

% Permutar los datos de manera aleatoria antes de dividirlos
perm = randperm(length(Y));
X_perm = X(perm, :);
Y_perm = Y(perm, :);

X_train = X_perm(1:trainLength, :);
Y_train = Y_perm(1:trainLength);

X_val = X_perm((trainLength + 1):(trainLength + valLength), :);
Y_val = Y_perm((trainLength + 1):(trainLength + valLength));

X_test = X_perm((trainLength + valLength + 1):end, :);
Y_test = Y_perm((trainLength + valLength + 1):end, :);

% Cambiar la forma de los datos para cumplir con los requisitos de entrada de una red
LSTM
X_train = reshape(X_train, [1, size(X_train, 1), size(X_train, 2)]);
X_val   = reshape(X_val, [1, size(X_val, 1), size(X_val, 2)]);
X_test  = reshape(X_test, [1, size(X_test, 1), size(X_test, 2)]);

X_train = permute(X_train, [3 2 1]);
X_val = permute(X_val, [3 2 1]);
X_test = permute(X_test, [3 2 1]);

% Cambiar la forma de los datos y convertirlos en cell arrays de secuencias individuales
X_train = num2cell(X_train, 1);
X_val   = num2cell(X_val, 1);
X_test  = num2cell(X_test, 1);

% Convertir las etiquetas de respuesta en cell arrays de secuencias individuales
Y_train = num2cell(Y_train);
Y_val = num2cell(Y_val);
Y_test = num2cell(Y_test);
```

Una vez ordenados los datos y tratados para que MATLAB los procese adecuadamente, se inicia el entrenamiento definiendo las propiedades de la red para una primera aproximación, que son las siguientes:

```
% Crear y conilustraciónr la red neuronal recurrente bidireccional
numHiddenUnits = 100;

layers = [
   sequenceInputLayer(1)
```

```
  bilstmLayer(numHiddenUnits, 'OutputMode', 'sequence') %Barajar meter otra capa aquí o meter un dropuot para regularizar
  fullyConnectedLayer(1)
  regressionLayer];

options = trainingOptions('adam', ... %Jugar en un futuro con estos valores hasta ver si puedo reducir algo más el error
  'ExecutionEnvironment', 'cpu', ...
  'MaxEpochs', 100, ...
  'MiniBatchSize', 128, ...
  'GradientThreshold', 1, ...
  'InitialLearnRate', 0.005, ...
  'LearnRateSchedule', 'piecewise', ...
  'LearnRateDropPeriod', 125, ...
  'LearnRateDropFactor', 0.2, ...
  'Verbose', 0, ...
  'Plots', 'training-progress', ...
  'ValidationData', {X_val, Y_val}, ...
  'ValidationFrequency', 30, ...
  'Shuffle', 'every-epoch');
```

Se han empleado 100 neuronal inicialmente, se variarán en la optimización para no incurrir en sobreajuste y para evitar perder eficiencia computacional.

Respecto a las capas utilizadas, se ha probado inicialmente una capa oculta; debido a los malos resultados obtenidos con esta arquitectura, se añadirán capas y se valorará meter un dropout para regularizar, esto se plasmará en la optimización.

A continuación, en la ilustración 26, se muestra el entrenamiento por etapas de esta red:

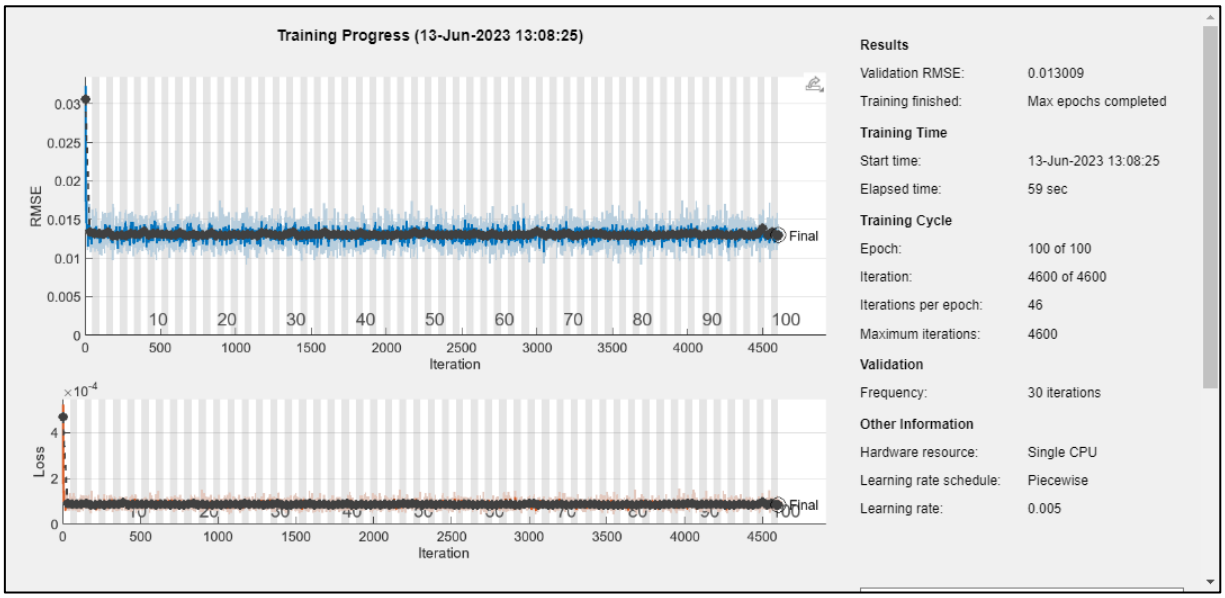

Ilustración 26: Entrenamiento preliminar red BRNN

Los resultados como se ha dicho no son muy positivos y se muestran a continuación en la ilustración 27.

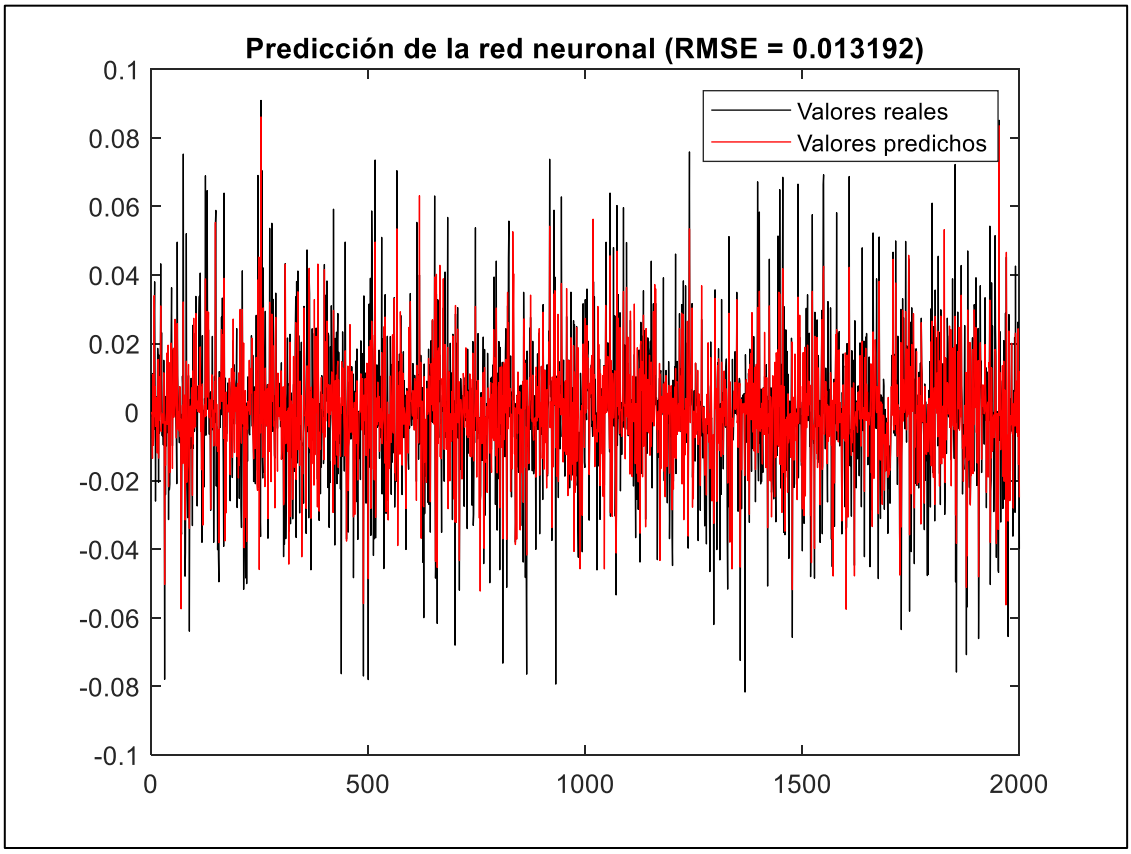

Ilustración 27: Primera predicción de la red BRNN

Como se puede apreciar, el resultado no es demasiado preciso, con lo que se procederá a su optimización para reducir el error de RMSE. El orden de actuación para ello es el que se muestra a continuación:

- Aumentar el número de unidades neuronales en la capa. Se probará con valores más altos, como 200 o 300, para permitir que la red capture patrones más complejos en los datos.
- Agregación de capas LSTM adicionales. Se experimentará agregando capas LSTM adicionales para aumentar la capacidad de la red y capturar relaciones temporales más profundas.
- Agregar regularización mediante técnicas de Dropout, al igual que se ha hecho anteriormente
- Ajustar los parámetros de entrenamiento, que ha sido donde mayores efectos se han producido en las soluciones anteriores, con lo cual se tendrá muy en cuenta para esta solución.

Al empezar a probarse combinaciones de parámetros, se puede denotar que la solución converge a un rango de resultados fijos tras la primera epoch y esta no varía hasta finalizar la simulación.

Debido a este problema, se ha procedido a hacer la red mucho más profunda, añadiendo dos capas completamente conectadas con función de activación ReLu. Esto consigue una mayor capacidad de representación.

Sin embargo, los valores siguen estancándose en 0,013 de RMSE. Ya que no se es capaz de encontrar una solución, a continuación, se exponen una serie de razones por las que puede estar ocurriendo esto:

- Los datos son insuficientes. Se necesitaría proporcionar una serie de datos mucho mayor para entrenar la red.
- No existe un preprocesamiento de datos. Al incluir alguna técnica como el escalamiento o la eliminación de ruido se podría quizás mejorar el comportamiento de la red.
- Error personal en la asignación de los parámetros. Cabe la posibilidad de que no se esté dando con la combinación adecuada para lograr un buen resultado.

Se ha probado también a aumentar el número de datos de entrenamiento y aún así se sigue estancando el error en 0,013.

Algunas de las razones por las que la BRNN no es tan eficiente como la LSTM para el problema concreto de la catenaria son las siguientes:

- **Dependencias en un tiempo largo**: El diseño recurrente bidireccional de las BRNN les permite capturar dependencias en ambos sentidos de la secuencia temporal. Sin embargo, las dependencias a largo plazo son esenciales para comprender completamente el comportamiento en algunas circunstancias,

como en el caso de una catenaria. La estructura interna de las celdas de memoria de las LSTM permite la captura y el mantenimiento de datos relevantes a largo plazo. Esto puede ser crucial para modelar patrones complejos y capturar las dependencias a largo plazo que existen en la catenaria.

- **Secuencias de longitud variable**: La longitud de una secuencia puede cambiar a lo largo del tiempo en algunos casos, como en el caso de una catenaria donde la longitud de la cadena puede cambiar. Dado que las redes LSTM pueden adaptarse automáticamente a diferentes longitudes de secuencia durante el entrenamiento y la inferencia, tienen inherentemente la capacidad de manejar secuencias de longitud variable. Sin embargo, la estructura fija de las BRNN y su necesidad de una longitud de secuencia predeterminada pueden ser limitantes en situaciones donde las secuencias varían en longitud.
- **Modelado de dependencias temporales complejas**: Los comportamientos complejos y no lineales de las catenarias pueden depender de eventos pasados en la secuencia temporal. El objetivo de las LSTM es modelar y capturar dependencias temporales complejas en los datos. Su estructura de memoria permite hacer predicciones precisas recordando y utilizando información relevante de eventos pasados. Las BRNN también pueden modelar dependencias temporales, pero en comparación con las LSTM, no pueden capturar dependencias a largo plazo.
- **Mejor capacidad de generalización**: Los patrones aprendidos en los datos de entrenamiento se pueden generalizar y aplicar a nuevas secuencias para hacer predicciones precisas. Esto significa que, durante el entrenamiento, pueden capturar las características y comportamientos generales de una catenaria y usar esa información para hacer predicciones en situaciones no vistas previamente. Las BRNN también pueden generalizar, pero debido a su estructura bidireccional, pueden ser más propensas a sobre ajustar los datos de entrenamiento y pueden ser más sensibles a patrones particulares en los datos.

Para el caso del modelo no lineal, la situación es exactamente la misma, solo que el error aumenta hasta una cifra de 0,03; valor totalmente inaceptable para una predicción.

Al final de este documento en el Anexo 1, se reflejan todos los códigos de Matlab finales utilizados para obtener las ilustraciones de la solución optimizada. Solo se exponen los del caso lineal ya que el caso no lineal son modificaciones muy ligeras que ya se han presentado a lo largo del trabajo y resultaría redundante.

5. Resultados

Dicho lo anterior, la primera red que se utilizó fue la *feedforward*. Inicialmente tuvo malas prestaciones, pero al modificar el código se redujo el error considerablemente. El mejor resultado obtenido mediante esta red fue de 0,009 para el caso lineal. A continuación, en las ilustraciones 28 y 29 se presentan dos gráficas. La primera es un histograma de errores de la simulación y la segunda es una regresión de la red neuronal.

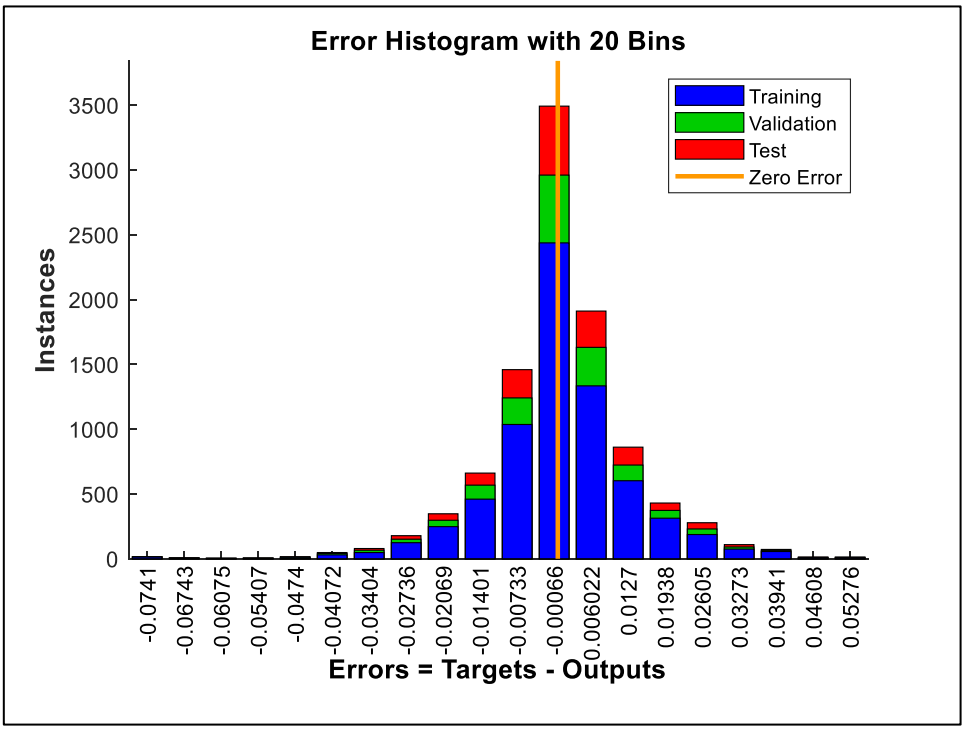

Ilustración 28: Histograma de errores de la red Feed Forward

La validación se utiliza durante el entrenamiento para ajustar el modelo y tomar decisiones sobre su arquitectura, evitando el sobreajuste. El conjunto de test es independiente y se usa solo una vez para obtener una medida objetiva del rendimiento del modelo en datos completamente nuevos, no vistos antes durante el entrenamiento o la validación.

En la gráfica de error histograma con 20 bins, "bins" se refiere a la cantidad de divisiones (intervalos) en los que se agrupan los errores para construir el histograma. Cada barra representa la frecuencia de errores que caen en un rango específico de valores. Si se utilizan más bins, el histograma mostrará más detalles sobre la distribución de errores, mientras que con menos bins, la información se agrupará en menos intervalos, lo que proporcionará una visión más general del rendimiento del modelo.

Es importante tener en cuenta que el conjunto de prueba solo debe utilizarse una vez, después de que se hayan realizado todas las decisiones de modelado y ajuste. No se debe utilizar para tomar decisiones sobre el modelo, ya que esto podría llevar a un sesgo en la evaluación del rendimiento del modelo.

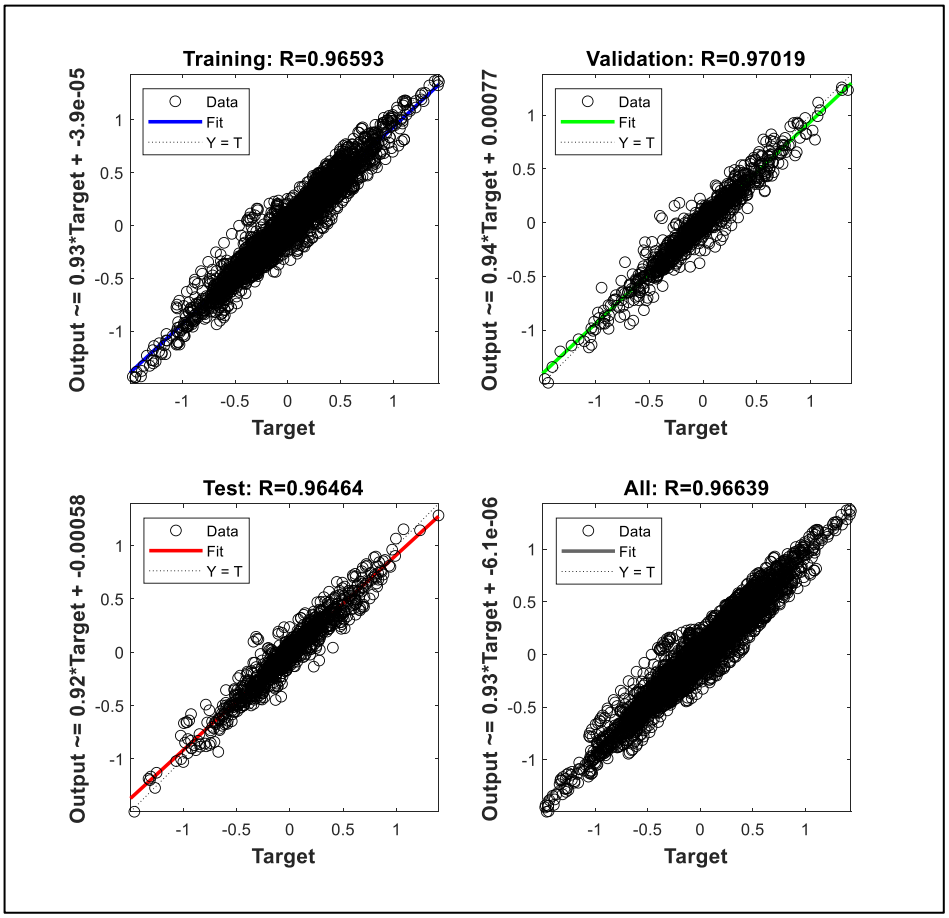

Ilustración 29: Regresión de la red Feed Forward

Se puede apreciar una correlación positiva, siendo el objetivo una R de valor 1. Los valores oscilan el 0,965 que no es un mal resultado y aproxima de manera bastante cercana el comportamiento del modelo computacional.

Las 4 gráficas abordan las siguientes funcionalidades:

- Loss (Pérdida): Muestra cómo evoluciona la función de pérdida (loss) durante el entrenamiento.
- Mean Squared Error (MSE): Es una métrica de evaluación que muestra cómo cambia el error cuadrático medio en el conjunto de entrenamiento a medida que avanza el entrenamiento. Cuanto menor sea el MSE, mejor será el rendimiento del modelo.
- Learning Rate (Tasa de aprendizaje): Esta gráfica muestra cómo cambia la tasa de aprendizaje durante el entrenamiento. Algunos algoritmos de optimización ajustan la tasa de aprendizaje automáticamente para mejorar la convergencia del modelo (Como es este caso).
- Gradient L2-Norm (Norma L2 del gradiente): Muestra cómo cambia la norma L2 del gradiente durante el entrenamiento. Esta métrica puede ser útil para verificar

si el gradiente está explotando o desvaneciendo, lo que podría afectar la estabilidad del entrenamiento.

Las otras dos redes, de carácter recurrente, han obtenido buenas predicciones respecto a los valores reales en el caso lineal. Ambas podrían ser utilizadas, pero debido a que no se ha conseguido reducir el error más de 0,013 en el caso de la BRNN y que para el caso no lineal no ofrece un buen comportamiento, se ha seleccionado la red LSTM como la solución propuesta debido a que presenta mejores resultados con un coste computacional muy aceptable. Contextualizando, la red LSTM es 4 veces más precisa que la BRNN y 3 veces más precisa que la *FeedForward*.

A continuación, en la ilustración 30 se presenta el histograma de errores de la solución LSTM, como se puede observar, la mayoría de los casos se encuentra muy cerca del error 0 lo cual hacer concluir que la solución es bastante precisa.

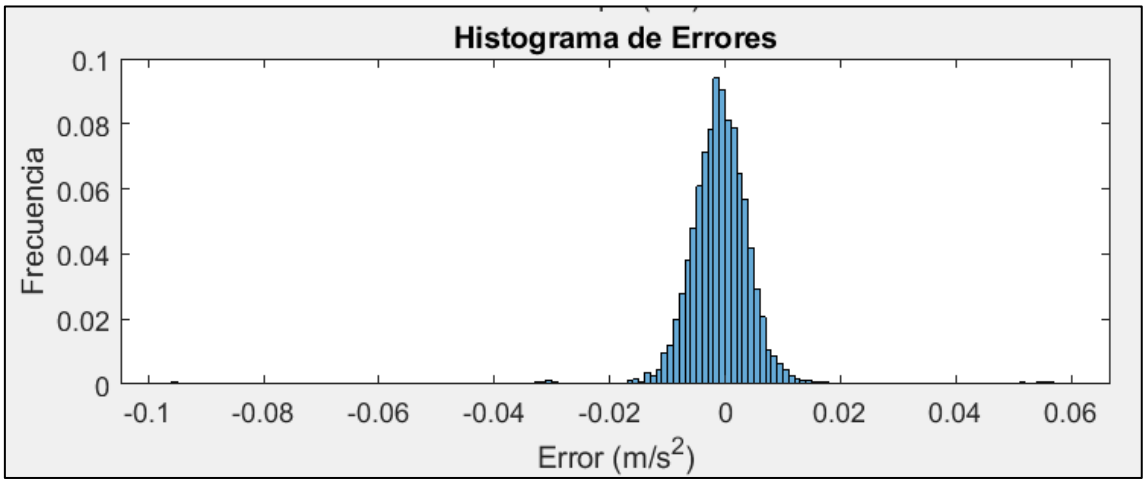

Ilustración 30: Histograma de errores de la red LSTM

Por otro lado, se muestra en la ilustración 31 el histograma de la mejor solución de red BRNN que se ha logrado encontrar:

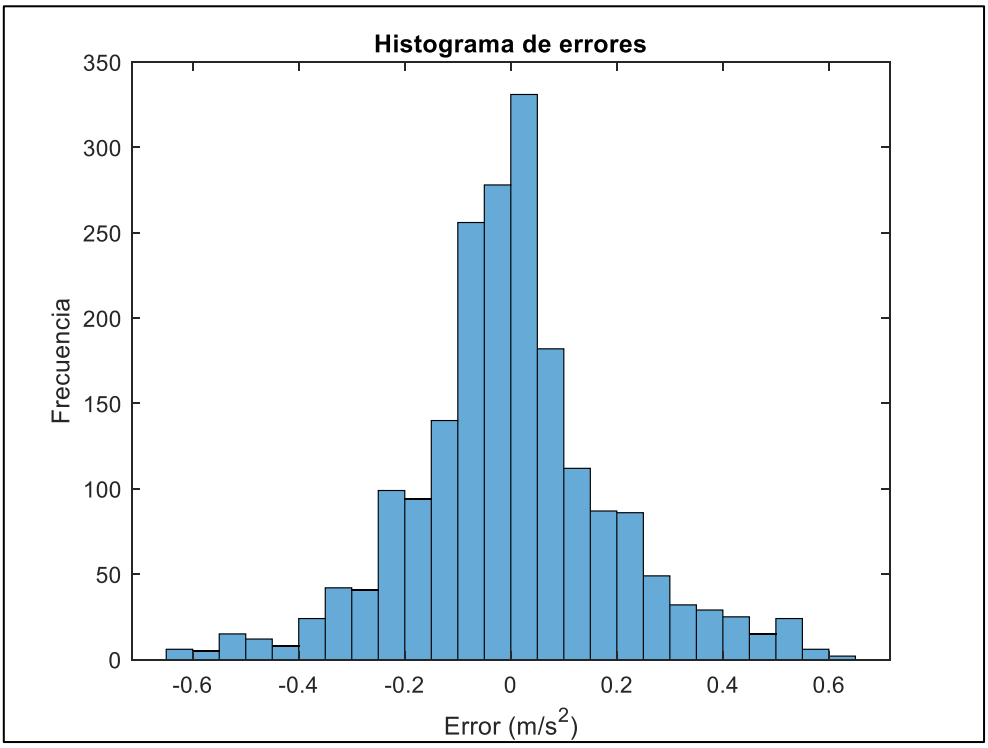

Ilustración 31: Histograma de error de la red BRNN

A modo de resumen, se presenta una tabla comparando los errores RMSE de las tres redes entrenadas para contrastar lo comentado:

Tabla 2: Comparativa entre las 3 redes neuronales

Red	***RMSE (modelo lineal)***	***RMSE modelo no lineal***
Feed Forward	0.009	0.013
LSTM	0.003	0.009
BRNN	0.013	0.03

6. Conclusiones

A lo largo de este proyecto, se ha demostrado la gran utilidad que tienen los modelos subrogados para problemas dinámicos como ocurre con la interacción pantógrafo-catenaria. Las redes neuronales son una herramienta con altas prestaciones y un futuro prometedor, a la par que mucho más económicas que otros modelos existentes.

Una vez entrenadas y presentadas todas las redes, se debe aclarar un aspecto fundamental de este estudio. Esto es que se debe tener muy presente que las soluciones han sido programadas y ejecutadas en un portátil de uso universitario; esto implica que muchas soluciones que se han programado no se han expuesto en este documento debido a que la potencia computacional del ordenador no ha permitido ejecutar el código o por su defecto, empleaba tiempos larguísimos y la idea de este proyecto de fin de máster es exponer una solución muy eficiente que no dependa de ordenadores extremadamente potentes y que permita obtener un conocimiento didáctico de cómo funcionan estos sistemas.

Las soluciones de inteligencia artificial han ido aumentando en número durante los últimos años, no solo en este tipo de problemas, sino también en otros existentes en el campo como puede ser el análisis de defectos geométricos. Por todo ello, se considera que en un futuro aparecerán muchas más soluciones de redes neuronales obteniendo un ahorro significativo de costes.

Como conclusión final, se ha obtenido una muy buena solución al problema planteado inicialmente. Respecto a las posibles líneas de futuro de este proyecto, si la investigación pudiese continuar, el siguiente paso a seguir sería utilizar datos experimentales para entrenar las redes; después se experimentaría con más tipos de red, ordenadores con mayor computación y con redes y sets de datos mucho más amplios y profundos que los que se han presentado. Hasta donde ha alcanzado este estudio, las redes LSTM son las que han demostrado mejor comportamiento para predecir el modelo del pantógrafo.

7. Bibliografía

[1] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.
[2] Bishop, C. M. (1995). Neural networks for pattern recognition. Oxford university press.
[3] Obukhov, A., & Ustyuzhanin, A. (2019). Recurrent neural networks and LSTM networks for text classification: A review. Proceedings of the Institute for System Programming, 31(3), 39-51. Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). A comparative study of learning algorithms for deep convolutional neural networks.
[4] Sharma, P. (2018). An introduction to regularization in neural networks. Towards Data Science.
[5] Ahn, S., Kang, Y. S., Choi, S., & Cho, J. R. (2021). The Recent Applications of Machine Learning in Rail Track Maintenance: A Survey. Applied Sciences, 11(2), 719. doi:10.3390/app11020719
[6] Papadrakakis, M., Manolis, G., & Fragiadakis, M. (2017). Computational Structural Dynamics and Earthquake Engineering. Wiley.
[7] Wikipedia contributors. (2022). Newmark-beta method. Wikipedia. https://en.wikipedia.org/wiki/Newmark-beta_method
[8] Birky, D., Ladd, J., Guardiola, I., & Young, A. (2022). Predicting the dynamic response of a structure using an artificial neural network. Journal of Low Frequency Noise, Vibration and Active Control, 41(1), 182-195.
[9] Stoffel, M., Gulakala, R., Bamer, F., & Markert, B. (2020). Artificial neural networks in structural dynamics: A new modular radial basis function approach vs. convolutional and feedforward topologies. Computer Methods in Applied Mechanics and Engineering, 364, 112989.
[10] Han, X., Xiang, H., Li, Y., & Wang, Y. (2019). Predictions of vertical train-bridge response using artificial neural network-based surrogate model. *Advances in Structural Engineering*, *22*(12), 2712-2723.

ANEXO I: Códigos de Matlab utilizados

Script simular_pantografo.m

```
function [A, U] = simular_pantografo(F)
if nargin < 1
    error('Se requiere al menos un argumento de entrada.');
end

%- Parámetros de rigidez del pantógrafo:
k1 = 7000;
k2 = 2200;
k3 = 0.1;
%- Valores de amortiguamiento:
c1 = 30;
c2 = 0;
c3 = 13.10;
%- Valores de las masas:
m1 = 8.5;
m2 = 9;
m3 = 1.31;

%MATRICES
M=[ m1 0 0; 0 m2 0; 0 0 m3];
C=[ c1 -c1 0; -c1 c1+c2 -c2; 0 -c2 c2+c3];
K=[ k1  -k1  0; -k1 k1+k2 -k2; 0 -k2 k2+k3];

%% Integracion del panto
dt = 0.001; % Incremento de tiempo
t = 0:dt:10; % Vector de tiempo
[A, U] = int_panto(M, C, K, F, dt); % Simulación del modelo con la señal de
fuerza aleatoria


end

function [A, U] = int_panto(M, C, K, F, dt)
%Integra un modelo de pantografo con fuerza en el gdl 1 y condiciones
iniciales nulas

Ngdl = size(M,1);
Nstp = length(F);

%% PARAMETROS INTEGRACION TEMPORAL-----------------------------------------
-------------------------
%- Parámetros de integración (Newmark)
bet = 0.25;
gam = 0.5;

%- Constates de Newmark lineal
b1 = 1/(bet*dt^2);
b2 = -1/(bet*dt);
b3 = 1-1/(2*bet);
b4 = gam*dt*b1;
b5 = 1+gam*dt*b2;
b6 = dt*(1+gam*b3-gam);
```

```
%- Inicialización de variables
U = zeros(Ngdl,Nstp);A = zeros(Ngdl,Nstp);
u = zeros(Ngdl,1);
v = zeros(Ngdl,1);
a = zeros(Ngdl,1);

%- Matriz de integración temporal
K_it = K + b4*C + b1*M;

%% bucle principal de integracion
for stp = 1:Nstp

    %- Variables del instante anterior
    Uant = u;
    Vant = v;
    Aant = a;
    %- Vector de fuerzas de las condiciones iniciales
    F_it = M*(b1*Uant-b2*Vant-b3*Aant) + C*(b4*Uant-b5*Vant-b6*Aant);
    %- Vector de fuerzas externas
    F_ext=[F(stp);0;0];

    %- Solución del paso temporal
    u = K_it\(F_it+F_ext);
    U(:,stp) = u;

    %- Actualización de velocidad y aceleración
    v = b4*(u-Uant) + b5*Vant + b6*Aant;
    a = b1*(u-Uant) + b2*Vant + b3*Aant;
    A(:,stp) = a;
end
end
```

Script FuncionPrincipal.m

```
% Parte principal del script
t = 0:0.001:10; % Vector de tiempo

% Llama a la función pseudo_modulated con los parámetros deseados
F = pseudo_modulated(10, 50, 0.001, 1);

[A, U] = simular_pantografo(F);

% Guardar los datos en un archivo .mat
save('datos_pantografo.mat', 'F', 'A', 'U')

% Graficar resultados
figure;
subplot(2,1,1);
plot(t, A(1,:));
xlabel('Tiempo (s)');
ylabel('Aceleración (m/s^2)');
title('Aceleración de la masa superior');

subplot(2,1,2);
plot(t, U(1,:));
xlabel('Tiempo (s)');
```

```
ylabel('Desplazamiento (m)');
title('Desplazamiento de la masa superior');

% Añade la función pseudo_modulated al final del archivo del script que me
% está dando error
function [f] = pseudo_modulated(T, f_max, dt, Amp_f)
% Creates a pseudo ramdon funcion modulated
Narm = round(f_max * T);
t = 0:dt:T; %numero impar
Nstp = length(t);

Fmod = rand(1, Narm) .* linspace(rand, 0.1 * rand, Narm);
Fmod(1) = 0;
Farg = rand(1, Narm) * 2 * pi;
Farg(1) = 0;

F = Fmod .* exp(1i * Farg);
F((Nstp + 1) / 2) = 0;
f = Nstp * ifft([F flip(conj(F(2:end)))]);

f = f .* linspace(rand, rand, Nstp) .* abs(rand + abs(sin(rand * t)));
f = f * Amp_f / max(max(f), -min(f));
end
```

Script EntrenamientoFeedForward

```
% Cargar los datos del archivo .mat
load('datos_pantografo.mat', 'F', 'A', 'U');

%Dimension de la funcion de entrada (numero de stps necesarios para predecir
el actual)
N=15;

% Preparar los datos para el entrenamiento
M=size(F,2);
%matriz de fuerzas
ind=(1:M-N+1)+(0:N-1)';
X=F(ind);
Y=A(1, N:end);


% Crear y configurar la red neuronal
hiddenLayerSize = [50, 30, 20, 10];
net = fitnet(hiddenLayerSize);

% Entrenar la red neuronal
[net, tr] = train(net, X, Y);
view(net);

%Evaluacion del entrenamiento
figure
plot(Y);hold on;plot(net(X))

%% Validation
% Número de conjuntos de validación a generar
num_val_sets = 1;
rmse_sum = 0;
```

```
for i = 1:num_val_sets
     % Generar nuevos datos de fuerza y aceleración usando la función de
fuerza aleatoria
     F_new = pseudo_modulated(10, 50, 0.001, 1);
     [A_new, U_new] = simular_pantografo(F_new);


     M=size(F_new,2);
     %mariz de fuerzas
     ind=(1:M-N+1)+(0:N-1)';
     X_val=F_new(ind);
     Y_val=A_new(1, N:end);



     % Probar la red neuronal con los nuevos datos de fuerza
     predictions = net(X_val);

     % Calcular el error cuadrático medio (RMSE)
     rmse = sqrt(mean((predictions - Y_val).^2));
     rmse_sum = rmse_sum + rmse;
     fprintf('RMSE for validation set %d: %.4f\n', i, rmse);
end

% Calcular el RMSE promedio
rmse_avg = rmse_sum / num_val_sets;

% Mostrar el RMSE promedio
fprintf('Average RMSE: %.4f\n', rmse_avg);

% Graficar resultados
figure;
plot(Y_val, 'b');
hold on;
plot(predictions, 'r');
legend('Aceleración Real', 'Aceleración Predicha');
xlabel('Tiempo (ms)');
ylabel('Aceleración (m/s^2)');
title('Comparación de Aceleración Real y Predicha (Validación)');

function [f] = pseudo_modulated(T, f_max, dt, Amp_f)
   % Creates a pseudo random function modulated
   Narm = round(f_max * T);
   t = 0:dt:T; %numero impar necesario
   Nstp = length(t);

   Fmod = rand(1, Narm) .* linspace(rand, 0.1 * rand, Narm);
   Fmod(1) = 0;
   Farg = rand(1, Narm) * 2*pi;
   Farg(1) = 0;

   F = Fmod .* exp(1i * Farg);
   F((Nstp + 1) / 2) = 0;
   f = Nstp * ifft([F flip(conj(F(2:end)))]);

   f = f .* linspace(rand, rand, Nstp) .* abs(rand + abs(sin(rand * t)));
   f = f * Amp_f / max(max(f), -min(f));
```

```
end
```

Script EntrenamientoLSTM.m

```
% Cargar los datos del archivo .mat
load('datos_pantografo.mat', 'F', 'A', 'U');

% Preparar los datos para el entrenamiento
X = F';
Y = A(1, :)';

% Cambiar la forma de los datos para cumplir con los requisitos de entrada de
una RNN
numFeatures = 1;
X = reshape(X, [numFeatures, size(X, 1), size(X, 2)]);
Y = reshape(Y, [numFeatures, size(Y, 1), size(Y, 2)]);

% Crear y conilustraciónr la red neuronal LSTM
numHiddenUnits = 50;

layers = [ ...
    sequenceInputLayer(numFeatures)
    lstmLayer(numHiddenUnits, 'OutputMode', 'sequence')
    fullyConnectedLayer(1)
    regressionLayer];

options = trainingOptions('adam', ...
    'ExecutionEnvironment', 'cpu', ...
    'MaxEpochs', 1000, ...
    'MiniBatchSize', 32, ...
    'GradientThreshold',1, ...
    'InitialLearnRate',0.001, ...
    'LearnRateSchedule','piecewise', ...
    'LearnRateDropPeriod',125, ...
    'LearnRateDropFactor',0.2, ...
    'Verbose',0, ...
    'Plots','training-progress');

% Entrenar la red neuronal LSTM
net = trainNetwork(X, Y, layers, options);

% Generar nuevos datos de fuerza y aceleración usando la función de fuerza
aleatoria
F_new = pseudo_modulated(10, 50, 0.001, 1);
[A_new, U_new] = simular_pantografo(F_new);

% Preparar los nuevos datos para la validación
X_val = F_new';
Y_val = A_new(1, :)';

% Cambiar la forma de los datos para cumplir con los requisitos de entrada de
una RNN
X_val = reshape(X_val, [numFeatures, size(X_val, 1), size(X_val, 2)]);
Y_val = reshape(Y_val, [numFeatures, size(Y_val, 1), size(Y_val, 2)]);

% Probar la red neuronal con los nuevos datos de fuerza
YPreds = predict(net, X_val);
```

```
predictions = squeeze(YPreds)';

% Calcular el error cuadrático medio (RMSE)
rmse = sqrt(mean((predictions - Y_val').^2));
fprintf('RMSE for the validation set: %.4f\n', rmse);

% Graficar resultados
figure;
subplot(2, 1, 1);
plot(Y_val', 'b');
hold on;
plot(predictions, 'r');
legend('Aceleración Real', 'Aceleración Predicha');
xlabel('Tiempo (ms)');
ylabel('Aceleración (m/s^2)');
title('Comparación de Aceleración Real y Predicha (Validación)');

subplot(2, 1, 2);
errors = predictions - Y_val';
histogram(errors, 'Normalization', 'probability');
xlabel('Error');
ylabel('Frecuencia');
title('Histograma de Errores');

function [f] = pseudo_modulated(T, f_max, dt, Amp_f)
  % Creates a pseudo random function modulated
  Narm = round(f_max * T);
  t = 0:dt:T; %numero impar necesario
  Nstp = length(t);

  Fmod = rand(1, Narm) .* linspace(rand, 0.1 * rand, Narm);
  Fmod(1) = 0;
  Farg = rand(1, Narm) * 2*pi;
  Farg(1) = 0;

  F = Fmod .* exp(1i * Farg);
  F((Nstp + 1) / 2) = 0;
  f = Nstp * ifft([F flip(conj(F(2:end)))]);

  f = f .* linspace(rand, rand, Nstp) .* abs(rand + abs(sin(rand * t)));
  f = f * Amp_f / max(max(f), -min(f));
end
```

Script EntrenamientoBRNN

```
% Cargar los datos del archivo .mat
load('datos_pantografo.mat', 'F', 'A', 'U');

% Preparar los datos para el entrenamiento
X = F';
Y = A(1, 1:length(X))';

trainLength = round(0.6 * length(Y));
valLength = round(0.2 * length(Y));

% Permutar los datos de manera aleatoria antes de dividirlos
perm = randperm(length(Y));
X_perm = X(perm, :);
```

```
Y_perm = Y(perm, :);

X_train = X_perm(1:trainLength, :);
Y_train = Y_perm(1:trainLength);

X_val = X_perm((trainLength + 1):(trainLength + valLength), :);
Y_val = Y_perm((trainLength + 1):(trainLength + valLength));

X_test = X_perm((trainLength + valLength + 1):end, :);
Y_test = Y_perm((trainLength + valLength + 1):end, :);

% Cambiar la forma de los datos para cumplir con los requisitos de entrada de
una red LSTM
X_train = reshape(X_train, [1, size(X_train, 1), size(X_train, 2)]);
X_val   = reshape(X_val, [1, size(X_val, 1), size(X_val, 2)]);
X_test  = reshape(X_test, [1, size(X_test, 1), size(X_test, 2)]);

X_train = permute(X_train, [3 2 1]);
X_val = permute(X_val, [3 2 1]);
X_test = permute(X_test, [3 2 1]);

% Cambiar la forma de los datos y convertirlos en cell arrays de secuencias
individuales
X_train = num2cell(X_train, 1);
X_val   = num2cell(X_val, 1);
X_test  = num2cell(X_test, 1);

% Convertir las etiquetas de respuesta en cell arrays de secuencias
individuales
Y_train = num2cell(Y_train);
Y_val = num2cell(Y_val);
Y_test = num2cell(Y_test);

% Crear y conilustraciónr la red neuronal recurrente bidireccional
numHiddenUnits = 100;

layers = [
    sequenceInputLayer(1)
    bilstmLayer(numHiddenUnits, 'OutputMode', 'sequence')
    fullyConnectedLayer(64)
    reluLayer()
    fullyConnectedLayer(32)
    reluLayer()
    fullyConnectedLayer(1)
    regressionLayer];

options = trainingOptions('adam', ...
    'ExecutionEnvironment', 'cpu', ...
    'MaxEpochs', 100, ...
    'MiniBatchSize', 128, ...
    'GradientThreshold', 1, ...
    'InitialLearnRate', 0.0001, ...
    'LearnRateSchedule', 'piecewise', ...
    'LearnRateDropPeriod', 125, ...
    'LearnRateDropFactor', 0.2, ...
    'Verbose', 0, ...
    'Plots', 'training-progress', ...
    'ValidationData', {X_val, Y_val}, ...
```

```
    'ValidationFrequency', 30, ...
    'Shuffle', 'every-epoch');

% Entrenar la red neuronal recurrente bidireccional
net = trainNetwork(X_train, Y_train, layers, options);

% Probar la red neuronal con los datos de prueba
Y_pred = predict(net, X_test);

% Convertir la salida en un cell array en forma de matriz
Y_pred = cell2mat(Y_pred);

% Calcular el error cuadrático medio (RMSE)
RMSE = sqrt(mean((Y_pred - cell2mat(Y_test)).^2));

% Graficar resultados
figure;
plot(cell2mat(Y_test), 'k');
hold on;
plot(Y_pred, 'r');
title(['Predicción de la red neuronal (RMSE = ' num2str(RMSE) ')']);
legend('Valores reales', 'Valores predichos');

% Histograma de errores
errors = cell2mat(Y_test) - Y_pred;
figure;
histogram(errors);
title('Histograma de errores');
xlabel('Error');
ylabel('Frecuencia');
```

Anexo II. Pliegue de condiciones

En esta sección se detalla la normativa a seguir al elaborar el proyecto de fin de máster en seguridad, salud e higiene. Se recurre al Real Decreto 488/1997 Se recurre al Real Decreto 488/1997 del 14 de abril y el Real Decreto 486/1997.

Real decreto 488/1997 del 14 de abril

Para lograr los objetivos del Real Decreto, se aplicarán las obligaciones establecidas en el anexo siempre que los elementos mencionados existan en el puesto de trabajo y las exigencias o características intrínsecas de la tarea no se opongan.

Equipo

La utilización en sí misma del equipo no debería poner en peligro a los empleados.

- **Pantalla**
  - Los caracteres de Los caracteres de la pantalla deben estar bien definidos y conilustracióndos de forma clara, así como tener dimensiones adecuadas y espacio suficiente entre los caracteres y los renglones.
  - La imagen de la pantalla debe ser estable, sin destellos ni centelleos.
  - La pantalla debe ser inclinable y orientable de manera fácil para adaptarse a las necesidades del usuario.
  - Para la pantalla, se puede usar un pedestal independiente o una mesa regulable.
  - El usuario de dispositivos con pantalla debe poder ajustar fácilmente la luminosidad y el contraste entre los caracteres y el fondo de la pantalla, así como ajustarlos a las condiciones ambientales.
  - No debe haber reflejos ni reverberaciones en la pantalla que puedan molestar al usuario.
- **Teclado**
  - El teclado debe ser inclinable e independiente de la pantalla para que el trabajador pueda trabajar con comodidad sin cansarse los brazos o las manos.
  - Debe haber suficiente espacio delante del teclado para que el usuario pueda sostener sus manos y brazos.
  - Para evitar reflejos, la superficie del teclado debe ser mate.
  - La disposición del teclado y las funciones de las teclas deberían tender a ser más fáciles de usar.

- Los símbolos de las teclas deben ser visibles y legibles desde la posición de trabajo habitual.

- **Mesa o superficie de trabajo**
  - La mesa o superficie de trabajo debe tener dimensiones suficientes, poco reflejante y permitir la colocación flexible de la pantalla, el teclado, los documentos y otros materiales.
  - El soporte de documentos debe ser estable y regulable y colocado de manera que minimice los movimientos incómodos de la cabeza y los ojos.
  - El espacio deberá ser suficiente para que los empleados puedan sentarse cómodamente.
- **Asiento de trabajo**
  - El asiento de trabajo debe ser estable para que el trabajador pueda moverse libremente y tener una postura confortable.
  - Es necesario controlar la altura del mismo.
  - El respaldo debe ser reclinable y ajustable en altura.
  - Se habilitará un reposapiés para aquellos que lo soliciten.

Entorno

- **Espacio**
  - El puesto de trabajo debe tener una dimensión suficiente y un espacio suficiente para permitir cambios de postura y movimientos de trabajo.
- **Iluminación**
  - Dependiendo del tipo de trabajo, las necesidades visuales del usuario y el tipo de pantalla utilizado, la iluminación general y especial (lámparas de trabajo) cuando sea necesaria deberán garantizar niveles adecuados de iluminación y una relación adecuada de luminancias entre la pantalla y su entorno.
  - Para evitar deslumbramientos y reflejos molestos en la pantalla u otras partes del equipo, el acondicionamiento del lugar de trabajo y del puesto de trabajo, así como la situación y las especificaciones técnicas de las fuentes de luz artificial, deberán coordinarse de tal manera que se eviten los deslumbramientos y reflejos molestos.
- **Reflejos y deslumbramientos**
  - Los puestos de trabajo deben instalarse de tal manera que las fuentes de luz, como ventanas y otras aberturas, los tabiques transparentes o translúcidos y los equipos o tabiques de color claro no produzcan deslumbramiento directo ni produzcan reflejos molestos en la pantalla.

- Para reducir la luz del día que ilumina el puesto de trabajo, las ventanas deben estar equipadas con un dispositivo de cobertura adecuado y regulable.

- **Ruido**
  - Al diseñar el puesto de trabajo, debe tenerse en cuenta el ruido producido por los equipos instalados, especialmente para evitar perturbar la atención o la palabra.
- **Calor**
  - Los equipos instalados en el lugar de trabajo no deberían generar calor adicional que moleste a los empleados.
- **Emisiones**
  - Para proteger la seguridad y la salud de los trabajadores, toda radiación, excepto la parte visible del espectro electromagnético debe reducirse a niveles insignificantes.
- **Humedad**
  - Debe establecerse y mantenerse un nivel de humedad aceptable.

Interconexión ordenador/persona

El empresario tendrá en cuenta los siguientes factores al crear, elegir, comprar y modificar programas, así como definir las tareas que requieran pantallas de visualización.

- El programa debe ajustarse a la tarea a realizar.
- El programa debe ser fácil de usar y adaptarse al nivel de conocimientos y experiencia del usuario. No se debe utilizar ningún dispositivo de control cuantitativo o cualitativo sin previa información y consulta con los representantes de los empleados.
- Los sistemas deberán informar a los trabajadores sobre su progreso.
- Los sistemas deberán mostrar los datos en un formato y a un ritmo que sean adecuados para los operadores.
- La ergonomía debe aplicarse especialmente al tratamiento de la información personal.

Real Decreto 486/1997

El Real Decreto en cuestión establece las normas básicas de seguridad y salud para los lugares de trabajo, incluyendo los siguientes anexos significativos.

ANEXO I. Condiciones generales de seguridad en los lugares de trabajo

Seguridad estructural

- Los edificios y locales de trabajo deben tener la estructura y solidez adecuadas para su tipo de uso. Todos sus componentes, estructurales o de servicio, incluidas las plataformas de trabajo, las escaleras y las escalas deberán:
  - Tener la solidez y la resistencia necesarias para soportar las cargas o esfuerzos a que sean sometidos para las condiciones de uso previstas.
  - Se debe disponer de un sistema de armado, sujeción o apoyo para garantizar su estabilidad.
- Es ilegal cargar demasiado los elementos mencionados en el apartado anterior. Solo cuando se proporcionen los equipos necesarios para que el trabajo pueda realizarse de forma segura se permitirá el acceso a techos o cubiertas que no ofrezcan suficientes garantías de resistencia.

Espacios de trabajo y zonas peligrosas

- Las dimensiones de los espacios de trabajo deben permitir que los empleados realicen su trabajo en condiciones ergonómicas aceptables y sin riesgos para su seguridad y salud. Las siguientes son sus dimensiones mínimas:
  - Tres metros de altura entre el techo y el piso. Sin embargo, en espacios comerciales, de servicios, oficinas y despachos, se podrá disminuir la altura a 2,5 metros.
  - Dos metros cuadrados de espacio disponible para cada trabajador.
  - Diez metros cúbicos de espacio libre por trabajador.
- Para que los trabajadores puedan realizar su trabajo en condiciones de seguridad, salud y bienestar, será suficiente separar los elementos materiales existentes en el puesto de trabajo. Cuando el espacio libre disponible no permite que el trabajador se mueva libremente por razones inherentes al trabajo, deberá disponer de más espacio cerca del puesto de trabajo.
- Las medidas adecuadas deben tomarse para proteger a los trabajadores autorizados an acceder a las áreas de los lugares de trabajo donde pueda haber riesgos de caída, caída de objetos y contacto o exposición an elementos agresivos. Deberá implementarse, siempre que sea posible, un sistema que impida la entrada de personas no autorizadas an esas áreas.
- Las áreas del lugar de trabajo en las que existe riesgo de caída, caída de objetos o contacto o exposición an elementos agresivos deben estar claramente señalizadas.

Suelos, aberturas y desniveles, y barandillas

- Los suelos de los espacios de trabajo deben ser estables, fijos y sin irregularidades ni pendientes peligrosas.
- Las barandillas u otros sistemas de seguridad similares se utilizarán para proteger aberturas o desniveles que puedan poner en peligro la caída de personas. Cuando sea necesario, estos sistemas tendrán componentes móviles. En particular, deben protegerse:
  - las grietas en el suelo.
  - Las aberturas en paredes o tabiques, siempre que su ubicación y dimensiones impliquen un riesgo de caída de personas, así como plataformas, muelles o estructuras similares. Sin embargo, si la altura de caída es menos de 2 metros, la protección no será necesaria.
  - Las escaleras y rampas tienen lados abiertos de más de 60 centímetros de altura. Si la anchura de la escalera es mayor de 1,2 metros, los lados cerrados tendrán pasamanos an una altura mínima de 90 centímetros. Si la anchura de la escalera es menor, al menos uno de los dos lados tendrá pasamanos.
- Las barandillas estarán hechas de materiales rígidos y tendrán una altura de al menos 90 centímetros. También tendrán una protección para evitar el paso o deslizamiento por debajo de ellas o la caída de objetos sobre las personas.

Vanos, ventanas y tabiques.

- Los tabiques transparentes o translúcidos, especialmente los acristalados, que se encuentran en las proximidades de los puestos de trabajo y las vías de circulación, deben estar claramente señalizados y fabricados con materiales seguros, o deberán estar separados de dichos puestos y vías de circulación para evitar que los trabajadores los golpean o se lesionen en caso de rotura.
- Los trabajadores deben poder abrir, cerrar, ajustar o fijar ventanas, vanos de iluminación cenital y dispositivos de ventilación de forma segura. No deberían colocarse de tal manera que puedan poner en peligro a los trabajadores cuando estén abiertos.
- Las ventanas y vanos de iluminación cenital deben poder limpiarse sin riesgo para los trabajadores que realicen esta tarea o para las personas que se encuentren en el edificio y sus alrededores. Para lograrlo, deberán tener los equipos necesarios o haber planeado incorporar sistemas de limpieza.

Vías de circulación

- Las puertas, pasillos, escaleras, escalas fijas, rampas y muelles de carga, tanto en el exterior como en el interior de los edificios y locales, deben ser fáciles de usar

y seguras para los peatones y vehículos que circulen por ellas, así como para el personal que trabaja en sus alrededores.

- El número, la situación, las dimensiones y las condiciones constructivas de las vías de circulación de personas o de materiales deberán adecuarse al número potencial de usuarios y a las características de la actividad y del lugar de trabajo, como se especifica en el apartado anterior.
- Las puertas exteriores y los pasillos tendrán una anchura mínima de 80 centímetros y 1 metro, respectivamente.
- La longitud de las carreteras que permitan el paso de vehículos y peatones deberá ser adecuada para permitir el paso simultáneo con una distancia de seguridad adecuada.
- Las carreteras para automóviles deben estar an una distancia adecuada de las puertas, portones, áreas para peatones, pasillos y escaleras.
- Si los muelles de carga tienen una gran longitud y son técnicamente factibles, deberán tener al menos una salida, o una en cada extremo.
- El trazado de las vías de circulación deberá estar claramente señalizado siempre que sea necesario para garantizar la seguridad de los trabajadores.

Puertas y portones

- Se debe colocar una señalización a la altura de la vista en las puertas transparentes.
- Cuando las superficies transparentes o translúcidas de las puertas y portones no sean de material de seguridad, deben protegerse contra la rotura.
- Las puertas y portones de vaivén deben ser transparentes o tener partes que permitan la visibilidad del área a la que se accede.
- Es necesario que las puertas correderas estén equipadas con un sistema de seguridad para evitar que se desvíen de los carriles y caigan.
- Se instalará un sistema de seguridad para evitar que las puertas y portones que se abran hacia arriba se caigan.
- Es necesario garantizar que los trabajadores puedan operar las puertas y portones mecánicos sin riesgo. Si no se abren automáticamente en caso de avería del sistema de emergencia, tendrán dispositivos de parada de emergencia fáciles de reconocer y acceder.
- Las puertas de acceso a las escaleras no se abrirán directamente sobre las escaleras, sino que se abrirán sobre descansos que tengan una anchura al menos igual a la de las escaleras.
- Los portones destinados principalmente a la circulación de vehículos deben permitir que los peatones los usen sin riesgos para su seguridad, o debe haber puertas expeditas y claramente señalizadas en su proximidad.

Rampas, escaleras fijas y de servicio

- Los materiales utilizados en las rampas, escaleras y plataformas de trabajo no deberían resbalarse y deberían tener elementos antideslizantes.
- La abertura máxima de los intersticios en escaleras o plataformas con pavimentos perforados es de 8 milímetros.
- Las rampas tendrán una pendiente máxima del 12 % si tienen menos de 3 metros, del 10 % si tienen menos de 10 metros o del 8 % en los demás casos.
- Las escaleras, excepto las de servicio, tendrán una anchura mínima de 1 metro.
- Los peldaños de una escalera deberían ser idénticos en tamaño. Se prohíben las escaleras de caracol a menos que estén en uso.
- Las huellas en las escaleras que no están en uso tendrán una longitud de 23 a 36 centímetros y una longitud de contrahuella de 13 a 20 centímetros. Las huellas en los escalones de las escaleras de servicio deben tener una longitud mínima de 15 centímetros y una longitud máxima de 25 centímetros.
- Entre los descansos de las escaleras, la altura máxima será de 3,7 metros. La profundidad de los descansos intermedios, medida en dirección a la escalera, no debe ser inferior an un metro, ni menos que la mitad de su anchura. La distancia vertical entre los peldaños no debe ser inferior a 2,2 metros.
- Para garantizar la seguridad de los trabajadores que las utilicen, las escaleras mecánicas y las cintas rodantes deben tener las condiciones de funcionamiento y los dispositivos necesarios. Sus dispositivos de parada de emergencia serán fáciles de reconocer y encontrar.

Escalas fijas

- Las escalas fijas tendrán una anchura mínima de cuarenta centímetros y una distancia máxima de treinta centímetros entre los peldaños.
- En las escalas fijas, debe haber al menos 75 centímetros de distancia entre el frente de los escalones y las paredes más cercanas al lado del ascenso. Los escalones y el objeto fijo más cercano deben estar a 16 centímetros de distancia. Si no hay jaulas u otros dispositivos equivalentes, habrá un espacio de 40 centímetros an ambos lados del eje de la escala.
- La barandilla o el borde de la escala se extenderá al menos 1 metro por encima del último peldaño o se tomarán medidas alternativas que proporcionen una seguridad equivalente cuando el paso desde el tramo final de una escala fija hasta la superficie a la que se desea acceder suponga un riesgo de caída por falta de apoyos.
- Las escalas fijas de más de 4 metros tendrán protección alrededor, al menos desde esa altura. En conductos, pozos angostos y otras instalaciones que ya tienen dicha protección por su estructura, esta medida no será necesaria.

- Se instalarán plataformas de descanso cada 9 metros o fracción si se utilizan escalas fijas para alturas superiores a 9 metros.

Las **escaleras de mano** de los lugares de trabajo deberán ajustarse a lo establecido en su normativa específica

Vías y salidas de evacuación

- La normativa específica se aplicará a las vías y salidas de evacuación, así como a las vías de circulación y las puertas que den acceso a ellas.
- Es importante que las vías y salidas de evacuación estén operativas y que desemboquen lo más directamente posible en el exterior o en una zona segura.
- Los empleados deben tener la capacidad de evacuar todos los lugares de trabajo en condiciones de máxima seguridad en caso de peligro.
- El número, la distribución y las dimensiones de las vías de evacuación y las salidas de evacuación dependerán del uso, del equipo, de las dimensiones de los lugares de trabajo y del número máximo de personas que puedan estar allí.
- Las puertas de emergencia deben estar abiertas y no cerradas para que cualquier persona que las necesite en caso de urgencia las pueda abrir fácil e inmediatamente. Estarán prohibidas las puertas que sean correderas o giratorias para situaciones de emergencia.
- Es importante que las puertas en los recorridos de las vías de evacuación estén bien señalizadas. Deberían poder abrirse desde el interior en cualquier momento sin ningún tipo de ayuda externa. Las puertas deberían poder abrirse cuando los lugares de trabajo estén ocupados.
- Según lo establecido en el Real Decreto 485/1997, de 14 de abril, sobre disposiciones mínimas de señalización de seguridad y salud en el trabajo, se deben señalar las vías y salidas específicas de evacuación. Esta señalización debe instalarse correctamente y permanecer en su lugar.
- No debe haber obstáculos en las entradas y salidas de evacuación, ni en las vías de circulación que permitan su uso sin obstáculos en cualquier momento. No cerrar las puertas de emergencia con llave.
- Las vías y salidas de evacuación que requieran iluminación deben estar equipadas con iluminación de seguridad de suficiente intensidad en caso de avería de la iluminación.

Condiciones de protección contra incendios

- Los lugares de trabajo deberán cumplir con lo establecido en la normativa que resulte de aplicación sobre condiciones de protección contra incendios.

- Los lugares de trabajo deberán estar equipados con dispositivos adecuados para combatir los incendios, así como detectores contra incendios y sistemas de alarma, si fuere necesario, en función de las dimensiones y el uso de los edificios, los equipos, las características físicas y químicas de las sustancias existentes.
- Los dispositivos no automáticos para combatir incendios deben ser fáciles de usar y manipular. Será necesario identificar estos dispositivos de acuerdo con lo establecido en el Real Decreto 485/1997, emitido el 14 de abril, que establece normas mínimas para la señalización de seguridad y salud en el lugar de trabajo. Dicha señalización debe instalarse correctamente y mantenerse por mucho tiempo.

Instalación eléctrica

- La instalación eléctrica de los lugares de trabajo debe cumplir con lo establecido en la normativa específica correspondiente.
- La instalación eléctrica no debería presentar peligro de incendio o explosión. Los empleados deben estar debidamente protegidos contra los riesgos de accidentes causados por contactos directos o indirectos.
- La instalación eléctrica y los dispositivos de protección deberán considerar el voltaje, los factores externos condicionantes y la capacidad de acceso de las personas.

Minusválidos

- Las puertas, las vías de circulación, las escaleras, los servicios higiénicos y los puestos de trabajo que son utilizados u ocupados por trabajadores minusválidos deben estar acondicionados para que puedan utilizarlos.

ANEXO II. Orden, limpieza y mantenimiento

Las áreas de paso, las salidas y las vías de circulación de los lugares de trabajo, y especialmente las salidas y vías de circulación previstas para la evacuación en casos de emergencia, deberán estar libres de obstáculos para garantizar que se puedan usar sin problemas en todo momento.

Para mantener en todo momento las condiciones higiénicas adecuadas, los lugares de trabajo, incluidos los locales de servicio, y sus respectivos equipos e instalaciones, se limpiarán periódicamente y siempre que sea necesario. Para lograr este objetivo, las características de los suelos, techos y paredes deberán ser tales que permitan la limpieza y el mantenimiento mencionados.

Los desperdicios, las manchas de grasa, los desechos de sustancias peligrosas y otros desechos que puedan causar accidentes o contaminar el lugar de trabajo se eliminarán con rapidez.

Las tareas de limpieza no deben poner en peligro a los empleados o a terceros, debiendo llevarse a cabo en los momentos, la forma y los medios apropiados para evitar riesgos. Los lugares de trabajo y, en particular, sus instalaciones, deberán ser mantenidos con regularidad para garantizar que sus condiciones de funcionamiento cumplan siempre con las especificaciones del proyecto y que las deficiencias que puedan afectar la seguridad y salud de los trabajadores sean subsanadas con rapidez.

Si se emplea una instalación de ventilación, debe estar en buen estado de funcionamiento y un sistema de control debe informar sobre cualquier avería cuando sea necesario para la salud de los trabajadores.

El mantenimiento de las instalaciones de protección debe incluir el monitoreo de su funcionamiento.

ANEXO III. Condiciones ambientales de los lugares de trabajo

Debe evitarse la exposición a temperaturas y humedades altas, cambios bruscos de temperatura, olores desagradables y corrientes de aire molestas, entre otras condiciones ambientales del lugar de trabajo.

El aislamiento térmico del lugar de trabajo debe adaptarse al clima local.

ANEXO IV. Iluminación de los lugares de trabajo

La iluminación en los lugares de trabajo debe adaptarse a las actividades que se realizan allí. Siempre que sea posible, se combinará iluminación natural con iluminación artificial para lograr una distribución de niveles de iluminación lo más uniforme posible, evitando deslumbramientos directos causados por la luz solar o artificial.

Un problema con el sistema de iluminación no debe poner en peligro la seguridad de los empleados ni causar problemas eléctricos, incendios o explosiones.

ANEXO V. SERVICIOS HIGIÉNICOS Y LOCALES DE DESCANSO

Los lugares de trabajo tendrán acceso fácil y suficiente agua potable. Se evitará cualquier situación que pueda causar la contaminación del agua potable. Si hay dudas sobre la calidad del agua, se indicará en las fuentes de agua.

Cuando los empleados deben usar ropa específica para su trabajo y no se puede pedir que se cambie por motivos de salud o decoro en otras dependencias, el lugar de trabajo proporcionará vestuarios.

Los vestuarios tendrán asientos y armarios o taquillas individuales con llave para guardar ropa y calzado. Cuando sea necesario debido a la contaminación, la suciedad o la humedad de la ropa de trabajo, se separarán los armarios o taquillas para la ropa de calle de la de trabajo. Los trabajadores deberán disponer de colgadores o armarios para colocar su ropa cuando no sea necesario.

Los lugares de trabajo tendrán áreas de aseo con espejos, lavabos con agua corriente y caliente si es necesario, jabón y toallas individuales y otros sistemas de secado con garantías higiénicas en las proximidades de los puestos de trabajo y vestuarios. Aparte de usar duchas de agua corriente, caliente y fría, cuando se lleven a cabo tareas sucias, contaminantes o que provoquen una gran sudoración, estarán disponibles. En estos casos, se proporcionarán a los empleados los medios de limpieza especiales necesarios. Cuando no estén integrados en estos últimos, los lugares de trabajo tendrán retretes con lavabos cerca de los puestos de trabajo, los espacios de descanso, los vestuarios y los aseos.

Se requiere un local de descanso accesible cuando la seguridad o salud de los trabajadores lo demande, a menos que el personal trabaje en despachos u otros lugares que ofrezcan condiciones equivalentes de descanso. Los locales de descanso deben tener dimensiones adecuadas y suficientes mesas y asientos para el número de trabajadores. Las trabajadoras embarazadas y madres lactantes deben tener la posibilidad de descansar en condiciones adecuadas. Si el trabajo se interrumpe regularmente y no hay locales de descanso, se deben proporcionar espacios para que los trabajadores puedan descansar sin riesgos. Tanto en los locales de descanso como en los espacios de interrupción, se deben tomar medidas para proteger a los no fumadores del humo del tabaco. Los dormitorios, si existen, deben cumplir con las normas de seguridad y salud y permitir el descanso adecuado de los trabajadores.

En los trabajos al aire libre, cuando sea necesario por motivos de seguridad o salud de los trabajadores, se deberá proporcionar un local de descanso de fácil acceso. Además, en aquellos casos en los que los trabajadores no puedan regresar a diario a su lugar de residencia debido a la distancia, se deberán proporcionar locales adecuados como dormitorios y comedores. Estos dormitorios y comedores deben cumplir con los requisitos de seguridad y salud necesarios, y permitir que los trabajadores descansen y se alimenten en condiciones adecuadas.

ANEXO VI. Material y locales de primeros auxilios

Los lugares de trabajo deben contar con material de primeros auxilios adecuado en cantidad y características, teniendo en cuenta el número de empleados, los riesgos presentes y la accesibilidad al centro médico más cercano. La ubicación y disposición del material debe permitir una respuesta rápida y efectiva acorde al tipo de daño previsible. Todo lugar de trabajo debe tener al menos un botiquín portátil con artículos básicos como desinfectantes, gasas, vendas y tijeras. Este material debe ser revisado y reemplazado con frecuencia. Los lugares de trabajo con más de 50 empleados deben tener un área dedicada a la atención médica, con un botiquín, una camilla y una fuente

de agua potable. Los centros de atención médica deben estar cerca de los lugares de trabajo.

ANEXO III. Relación del trabajo con los Objetivos de Desarrollo Sostenible de la agenda 2020

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. **Fin de la pobreza.**				X
ODS 2. **Hambre cero.**				X
ODS 3. **Salud y bienestar.**		X		
ODS 4. **Educación de calidad.**	X			
ODS 5. **Igualdad de género.**				X
ODS 6. **Agua limpia y saneamiento.**		X		
ODS 7. **Energía asequible y no contaminante.**		X		
ODS 8. **Trabajo decente y crecimiento económico.**		X		
ODS 9. **Industria, innovación e infraestructuras.**		X		
ODS 10. **Reducción de las desigualdades**.		X		
ODS 11. **Ciudades y comunidades sostenibles**.		X		
ODS 12. **Producción y consumo responsables.**		X		
ODS 13. **Acción por el clima.**	X			
ODS 14. **Vida submarina.**				X
ODS 15. **Vida de ecosistemas terrestres.**		X		
ODS 16. **Paz, justicia e instituciones sólidas.**				X
ODS 17. **Alianzas para lograr objetivos.**		X		

Descripción de la alineación del TFG/M con los ODS con un grado de relación más alto.

Varios Objetivos de Desarrollo Sostenible (ODS) buscan promover un desarrollo sostenible a nivel global. El proyecto de entrenamiento de redes neuronales simula el comportamiento de la catenaria de un tren. La innovación en el sector de la infraestructura ferroviaria (ODS 9) se fomenta al utilizar tecnologías avanzadas como la empleada en el presente documento. La simulación del comportamiento de la catenaria del tren ayuda a desarrollar soluciones más eficientes y seguras, lo que mejora la calidad y la eficacia de las infraestructuras de transporte.

En segundo lugar, el proyecto se vincula al ODS 11: Ciudades y comunidades sostenibles. Se puede mejorar la planificación y el diseño de la infraestructura ferroviaria en las ciudades al comprender y simular el comportamiento de la catenaria del tren. Esto permite la creación de sistemas de transporte más eficientes, seguros y sostenibles, lo que conduce al desarrollo de comunidades más sostenibles en términos de movilidad y acceso a servicios.

Además, el proyecto cumple con el ODS 13: Acción por el clima. Las redes neuronales pueden reducir las emisiones de gases de efecto invernadero simulando precisamente la catenaria del tren. El uso de energía en el sistema de transporte se optimiza al mejorar la eficiencia energética y la gestión de la infraestructura ferroviaria. Esto reduce el impacto ambiental y promueve prácticas más sostenibles para el clima.

Por otro lado, el ODS 12 sobre producción y consumo responsables está vinculado al proyecto. La comprensión y la simulación del comportamiento de la catenaria del tren ayudan an optimizar la producción y el consumo de recursos en la industria ferroviaria. La toma de decisiones informadas sobre el diseño, el mantenimiento y el uso de los sistemas ferroviarios puede reducir el desperdicio y fomentar una gestión más responsable de los recursos, lo que promueve prácticas de producción y consumo más sostenibles.

Anexo IV. Presupuesto

Los costes asociados al presente proyecto corresponden a 3 ámbitos, que son el coste del personal, el coste de software y el coste material.

Coste del personal

El coste del personal incluye a todas las partes involucradas para la realización de este trabajo de fin de máster. En este caso, solo hay dos miembros que lo integren que son el tutor de prácticas y el alumno. Para estimar el coste hora del alumno, se ha escogido el sueldo que está percibiendo este en su lugar de trabajo durante la realización del proyecto. A continuación, se muestra una tabla con los valores correspondientes:

Tabla 3: Costes referentes al personal

	Tiempo (horas)	*Sueldo (€/h)*	*Subtotal (€)*
Tutor de prácticas	30	30	900
Alumno	300	12,5	3750
		Total	**4650**

Coste de software

Los softwares empleados han sido gratuitos ya que la propia universidad facilita licencias para estudiantes. De todos modos, se realizarán los cálculos con el precio real de licencia para estimar cuanto costaría realizar este proyecto por un equipo no vinculado a la universidad.

Tabla 3: Costes referentes a las licencias

	Precio (€)	*Tiempo de uso (mes)*	*Subtotal (€)*
Microsoft Office	84	4	28
MATLAB	860	4	215
		Total	**243**

Coste del material

En los gastos de material, se ha incluido únicamente el precio del ordenador empleado por el alumno. Este es un Asus ZenBook 14, que se valorará al precio de compra por parte del estudiante:

Tabla 4: Costes referentes al material

	Precio (€)	*Total (€)*
Ordenador	980	980

Sumando todo el desglose anterior de presupuestos y añadiendo el IVA (Impuesto de valor añadido) de 21%, El presupuesto final es de 7106,33 euros.

Tabla 5: Presupuesto total del proyecto

Personal		*4650*
Software		243
Material		980
	Subtotal	5873
	IVA	1233,33
	Total	**7106,33**