



UNIVERSIDAD
POLITÉCNICA
DE MADRID

ESTUDIO DE TÉCNICAS DE STRUCTURAL HEALTH MONITORING MEDIANTE DEEP LEARNING

CASO DE ESTUDIO 2
MÁSTER UNIVERSITARIO EN SISTEMAS ESPACIALES

Autor: Daniel del Río Velilla

Tutor: Antonio Fernández López

MADRID, 25 DE ENERO, 2022

Resumen

Este trabajo recoge el desarrollo de una herramienta de localización y caracterización energética de impactos utilizando sensores piezoeléctricos y modelos de Deep Learning. El objetivo principal del trabajo ha sido la localización de impactos, coordenadas X e Y, y la obtención de la masa y velocidad del objeto impactado, además de la energía del impacto partiendo de la onda mecánica producida por este. Se ha prestado especial atención a la diferenciación de las diversas masas y velocidades del objeto impactado que producen una única energía de impacto.

Índice

Índice de figuras	I
Índice de tablas	III
Acrónimos	IV
1. Introducción y motivación	1
2. Impactador automático	3
2.1. Requisitos	3
2.2. Soluciones de diseño	4
2.3. Diseño mecánico	8
2.3.1. CAD	8
2.3.2. Impactador de control numérico fabricado	13
2.4. Software de control	17
2.4.1. Impactor_Class	17
3. Deep Learning	18
3.1. Objetivos del modelo	18
3.2. Número de muestras a obtener	19
3.3. Preparación de impactos	20
3.4. Arquitectura utilizada	21
3.5. Entrenamiento	23
4. Resultados	25
4.1. Localización	25
4.1.1. Modelo 1	26
4.1.2. Modelo 2	28
4.1.3. Modelo 3	29
4.1.4. Modelo 4	31
4.1.5. Resumen del modelo de localización	32

4.2. Caracterización energética	34
4.2.1. Modelo 1	34
4.2.2. Modelo 2	35
4.2.3. Modelo 3	36
4.2.4. Modelo 4	37
4.2.5. Resumen del modelo energético	37
5. Conclusiones	39
Referencias	40

Índice de figuras

1.	Perfiles de aluminio extruido y tornillería.	4
2.	Electroimán y cerramiento del OI.	5
3.	Motor paso a paso Nema 17.	5
4.	Ejemplo de lastre cilíndrico (50 g).	6
5.	Arduino Mega 2560 y Ramps 1.4 utilizados en el ICNC.	7
6.	Driver DRV8825.	7
7.	CAD del conjunto <i>Estructura principal</i>	8
8.	CAD del conjunto <i>Carro X</i>	9
9.	CAD del conjunto <i>Carro Y</i>	10
10.	CAD del conjunto <i>Carro Z</i>	11
11.	Objeto impactador.	11
12.	Sistema de reocogida del OI	12
13.	CAD completo del ICNC	13
14.	Vista isométrica del ICNC.	14
15.	Vista lateral del ICNC.	15
16.	Vista frontal del ICNC.	16
17.	Electroimán activado manteniendo el OI (rotada).	17
18.	Placa rigidizada utilizada para el estudio de impactos.	19
19.	Comparación de un impacto antes y después de ser procesado.	21
20.	Secuencia de capas del compresor.	22
21.	Secuencia de capas del regresor.	22
22.	Arquitectura del funcionamiento completo del modelo de detección de impactos.. . .	22
23.	Comparación de un impacto antes y después de ser procesado.	24
24.	Predicción de la red y distribución de errores.	26
25.	<i>Performance</i> de la red al predecir la coordenada X del impacto para la distribución 1. .	26
26.	<i>Performance</i> de la red al predecir la coordenada Y del impacto para la distribución 1. .	27
27.	<i>Performance</i> de la red al predecir la posición del impacto para la distribución 1. . . .	27
28.	<i>Performance</i> de la red al predecir la coordenada X del impacto para la distribución 2. .	28
29.	<i>Performance</i> de la red al predecir la coordenada Y del impacto para la distribución 2. .	28

Índice de tablas

1.	<i>Performance</i> de la red de localización de impactos.	33
2.	<i>Performance</i> de la red de caracterización energética.	38

Acrónimos

CAD Computer Aided Design. 4

DL Deep Learning. 3, 18, 19

IA Inteligencia Artificial. 3

ICNC Impactador de Control Numérico. 3, 4, 6–8, 12–16, 39

ML Machine Learning. 18

NN Neutal Networks. 18, 20, 23

OI Objeto impactador. 3–6, 10–13, 17, 18

1. Introducción y motivación

El 1 de febrero de 2003 ocurrió una de las mayores catástrofes a la que la comunidad aeroespacial se ha enfrentado, el *Columbia disaster*.

Este día, la misión número 28 del *Space Shuttle orbiter Columbia* comenzó el proceso de reentrada cerca a las 13:59 UTC. Sin embargo, pocos segundos después se apreció por las cámaras de televisión como el *Columbia* comenzó a arder e impactó con el terreno sin dejar ningún superviviente.

Casi instantáneamente, la NASA comenzó una investigación para encontrar la causa del accidente. Cuando revisaron los vídeos del lanzamiento descubrieron que, 81 segundos después de la ignición de los motores, parte del aislamiento de FOAM del tanque de combustible exterior se desprendió e impactó sobre el borde de ataque del ala izquierda.

El impacto fue evaluado por los ingenieros considerando que no era lo suficientemente energético como para producir daño significativo en el recubrimiento de carbono-carbono. Sin embargo, se produjo una rotura en el recubrimiento, lo cual fue demostrado en ensayos posteriores.

La misión 28 del *Columbia* no fue la primera vez que se detectó un desprendimiento de aislante que impactó sobre la protección térmica. En las anteriores ocasiones no se produjeron daños significativos, pero de haberlo sido, tampoco se podría haber cuantificado para detectar la presencia de daño.

La motivación de este trabajo nace del *Columbia disaster*. La ausencia de herramientas de caracterización de impactos en estructuras complejas hizo imposible la detección del daño producido por el desprendimiento de aislante. Posiblemente, si el *Space Shuttle* hubiera dispuesto de tecnología de *Structural Health Monitoring* (SHM) habría operado de forma más segura y durante más tiempo.

Por eso, este trabajo consiste en el desarrollo de una herramienta de caracterización de impactos en estructuras complejas, obteniendo la localización, energía, masa y velocidad del objeto impactado.

En diversos campos de la ingeniería se realizan modelos matemáticos que permiten determinar el comportamiento de un sistema o simularlo bajo diversas condiciones. En el caso de localización de impactos en estructuras complejas de fibra de carbono (CFRP) la construcción de modelos matemáticos se vuelve una tarea complicada dada la dependencia de la velocidad de propagación de la información en el material con la secuencia de apilado.

Por este motivo, se ha propuesto la construcción de un modelo de caracterización de impacto basado en Deep Learning. Esto hace que el propio proceso de entrenamiento de la red sea capaz de aprender la física del problema y ajustar sus parámetros de forma autónoma.

El lado negativo de utilizar modelos de Deep Learning es que es necesario una gran cantidad de datos (orden de 10^4) para que la confianza en los modelos entrenados sea aceptable. Traducido al problema objeto de este trabajo, es necesario generar una gran cantidad de impactos y etiquetarlos según su posición, masa, velocidad y energía.

Estas actividades repetitivas son susceptibles a la automatización, por lo que se ha desarrollado y construido un *Impactador automático* que obtendrá y pre-procesará todos los impactos utilizados en este trabajo.

2. Impactador automático

Los algoritmos de Inteligencia Artificial, IA, necesitan de una gran cantidad de información para su correcto aprendizaje y funcionamiento.

Como se ha dicho, el objetivo principal de este trabajo es el desarrollo de una herramienta de caracterización de impactos basada en Deep Learning, DL. Esta herramienta va a utilizar la onda de Lamb producida por el Objeto Impactado, OI, leída por sensores piezoelectricos y teniendo como salida las coordenadas X-Y, masa, velocidad y energía del impacto.

Dicho lo anterior, para poder localizar un impacto se va a requerir de una gran cantidad de impactos en múltiples coordenadas. Para predecir la energía de impacto se va a tener que impactar a varias velocidades y con diversas masas. Además, para que el modelo de DL sea capaz de aprender de datos, estos tienen que ser etiquetados con la respectiva coordenada, velocidad, masa y energía.

Todo esto hace inviable la generación de impactos de entrenamiento de forma manual, por lo que se ha propuesto el desarrollo de un Impactador de Control Numérico (ICNC). Los procesos repetitivos que requieren precisión son susceptibles a la automatización, ya que con ello se consiguen más y mejores resultados.

A continuación se detalla el desarrollo del Impactador CNC.

2.1. Requisitos

Los requisitos impuestos para el desarrollo del ICNC se presentan a continuación. Estos deben ser íntegramente cumplidos.

- R-1 El diseño debe ser paramétrico.
- R-2 Las dimensiones máximas de la pieza impactada ($X \times Y \times Z$) deben ser $1440 \times 990 \times 150$ mm³.
- R-3 Se deben poder aplicar condiciones de contorno de empotramiento sobre todos los bordes del objeto impactado.
- R-4 La masa máxima del objeto a impactar debe ser de 1 kg.
- R-5 Se debe poder variar la masa del objeto impactador.
- R-6 La altura máxima desde la que se pueda realizar un impacto debe ser 1750 mm.
- R-7 Se debe poder seleccionar la altura de impacto vía *software*.
- R-8 La precisión de posicionamiento en todos los ejes debe ser menor a 0.05 mm.
- R-9 El ICNC debe poder funcionar sin necesidad de supervisión durante tiempos prolongados.

R-10 El ICNC debe ser compatible con la tarjeta de adquisición de datos NI-USB-6356.

R-11 La realización de un ensayo debe ser configurable desde una interfaz.

R-12 Los impactos realizados deben ser verificados antes de su almacenamiento.

R-13 Los impactos erróneos deberán ser repetidos.

2.2. Soluciones de diseño

En esta sección se van a proponer las soluciones seleccionadas para cumplir los diferentes requisitos definidos en la Sección 2.1.

R-1

Para poder realizar un diseño paramétrico se ha utilizado el *software* de CAD Catia.

R-2 & R-3

La estructura del ICNC se ha construido mediante perfiles de aluminio extruido. Estos perfiles pueden ser comprados con diversas secciones (mayor o menor rigidez) y de longitudes entre 50 mm hasta 3000 mm [1].

La flexibilidad de uniones y configuraciones permiten un sencillo diseño paramétrico (R-1). Además, se han utilizado perfiles de aluminio como útil de aplicación de condiciones de contorno siendo unidos a la estructura principal.

En la Figura 1 se pueden varios perfiles junto a elementos de unión.



Figura 1: Perfiles de aluminio extruido y tornillería.

R-4

Para poder desplazar verticalmente la masa máxima del OI se ha utilizado un electroimán, [2], junto con una pieza metálica, que actuará de cerramiento roscándose el extremo del OI.

El electroimán elegido es capaz de levantar la masa máxima requerida y la pieza de cerramiento ha sido torneada en acero, ya que se necesitaba un material ferromagnético.

Se pueden ver ambos elementos en la Figura 2.

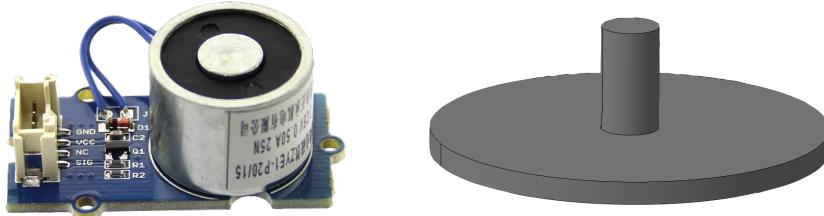


Figura 2: Electroimán y cerramiento del OI.

Además, esta masa va a tener que ser sostenida por un motor por lo que el par estático del motor ha de ser suficiente. Para cumplir el requisito se ha elegido un motor Nema 17, [3], que posee un par estático de 45 Ncm. Teniendo en cuenta que el eje del motor va a una polea de radio 2.5 cm el motor puede levantar una masa igual a

$$T = 9.81 \times m \times R \quad [\text{Ncm}] \quad \rightarrow \quad m = \frac{45}{9.81 \times 2.5} = 1.85 \quad [\text{kg}]. \quad (1)$$

Por lo que el Nema 17, cumple el requisito R-4

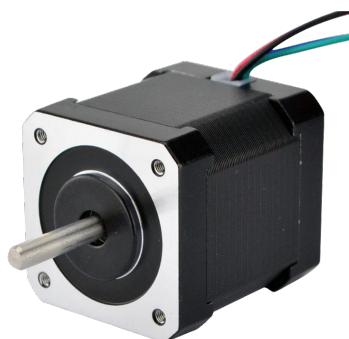


Figura 3: Motor paso a paso Nema 17.

R-5

La energía producida por el impacto de un objeto sigue la expresión de energía potencial,

$$E = \frac{1}{2}mv^2 = mgh \quad [\text{J}], \quad (2)$$

esto quiere decir que se puede variar la energía de un impacto variando la altura y la masa de OI.

Esto es interesante ya que se pueden obtener diversas configuraciones de altura-masa que den un mismo valor de energía.

Para poder variar la masa del objeto impactador se han fabricado unos lastres cilíndricos de acero con una masa unitaria igual a 50 g. Esto hace fácilmente configurable la masa del OI.

Un ejemplo de esta masa se puede ver en la Figura 4.



Figura 4: Ejemplo de lastre cilíndrico (50 g).

R-6

El tipo de impacto elegido ha sido por gravedad. Esto quiere decir que el OI va a efectuar una caída libre hasta impactar contra la superficie de la pieza impactada.

Para guiar al impactador en la caída se ha utilizado un tubo de PVC, [4], de 2 metros de longitud. De esta forma se consigue que todos los impactos tengan el mismo coeficiente aerodinámico ya que el área frontal del OI y la interfaz con el tubo será igual en todas las caídas.

Asumiendo que la altura de suelta es pequeña y que no se va a llegar nunca a la velocidad límite de caída se puede calcular la velocidad de impacto mediante la siguiente expresión,

$$v = \sqrt{2hg} \quad [\frac{\text{m}}{\text{s}}], \quad (3)$$

siendo h la altura de suelta del OI y g es la aceleración de la gravedad e igual a $9.81 \frac{\text{m}}{\text{s}^2}$. Es importante destacar que la velocidad de impacto es independiente de la masa a bajas alturas.

R-7

Como es necesario tener control de la altura a la que se va a soltar el OI, además del posicionamiento en los ejes X-Y-Z, hace obligatorio tener una electrónica y *software* capaz de controlar al menos 4 ejes.

La elección de un Arduino Mega 2560, [5], junto a la placa expansora de potencia Ramps 1.4, [6], permite el uso de hasta 5 ejes. Para completar la funcionalidad de el cerebro del ICNC se ha utilizado el *firmware* *grbl-Mega-5X*, [7], un control numérico de 5 ejes de *software* libre y fácilmente configurable.

En la Figura 5 se pueden ver las placas utilizadas.

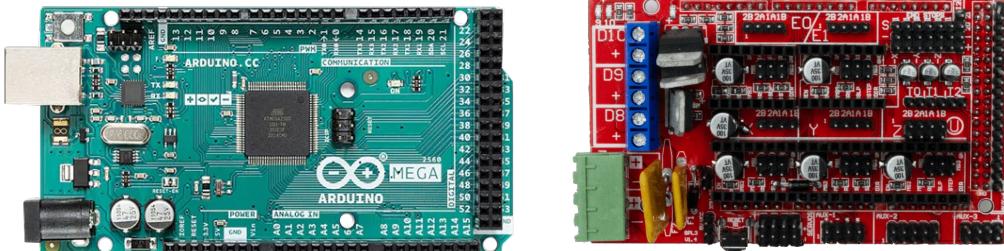


Figura 5: Arduino Mega 2560 y Ramps 1.4 utilizados en el ICNC.

R-8

La precisión del posicionamiento es fruto de la combinación de la resolución del paso del Nema 17, 1.8° , del elemento que transmite el movimiento y del número de veces que divide el paso del motor la placa de potencia del motor.

Para el ICNC se ha seleccionado como placa de potencia el Pololu DRV8825, [8], que divide el paso en 64.

El movimiento de los ejes X-Y se transmite con poleas de 10 mm de radio, el eje Z utiliza un husillo de paso 4 mm, por lo que el movimiento más restrictivo es el del eje A (cuarto eje, altura de impacto) ya que funciona con una polea de 25 mm. La resolución en el eje A se calcula mediante la siguiente expresión,

$$l = \pi \frac{1.8}{180 \cdot 64} \cdot 25 = 0.0123 \quad [\text{mm}], \quad (4)$$

cuyo resultado cumple la resolución mínima. Además, este driver es capaz de dar la corriente máxima que exige el motor Nema 17.

En la Figura 6 se puede ver el driver seleccionado.

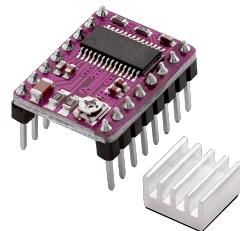


Figura 6: Driver DRV8825.

R-9 & R-10 & R-11 & R-12

Para cumplir estos requisitos se ha desarrollado un *software* en MATLAB basado en una clase (`Impactor_Class.m`) que contiene todos los procedimientos necesarios para realizar un ciclo de impactos.

Este código se comunica mediante puerto Serial con el Arduino, comandando las coordenadas del impacto, altura requerida, encender o pagar el electroimán, activar la tarjeta de adquisición de datos, procesar el impacto para comprobar si es válido y almacenarlo etiquetado.

En la Sección 2.4 se explicará con mayor profundidad el funcionamiento de este *software*.

2.3. Diseño mecánico

En esta sección se va a mostrar el diseño final realizado, se separará en los distintos subconjuntos principales y finalmente se podrá ver el ICNC montado.

2.3.1. CAD

A continuación se va a descomponer el diseño en los principales subconjuntos.

Estructura principal X

En la parte inferior de la estructura se integran los perfiles de aplicación de condición de contorno y en su parte superior se integran las guías del eje X, por las que se desplaza el carro X.



Figura 7: CAD del conjunto *Estructura principal*.

Carro X

El Carro X va montado sobre la guías de la estructura principal. En su parte inferior se encuentran los patines del eje X y en la parte superior se encuentra en las guías del eje Y.

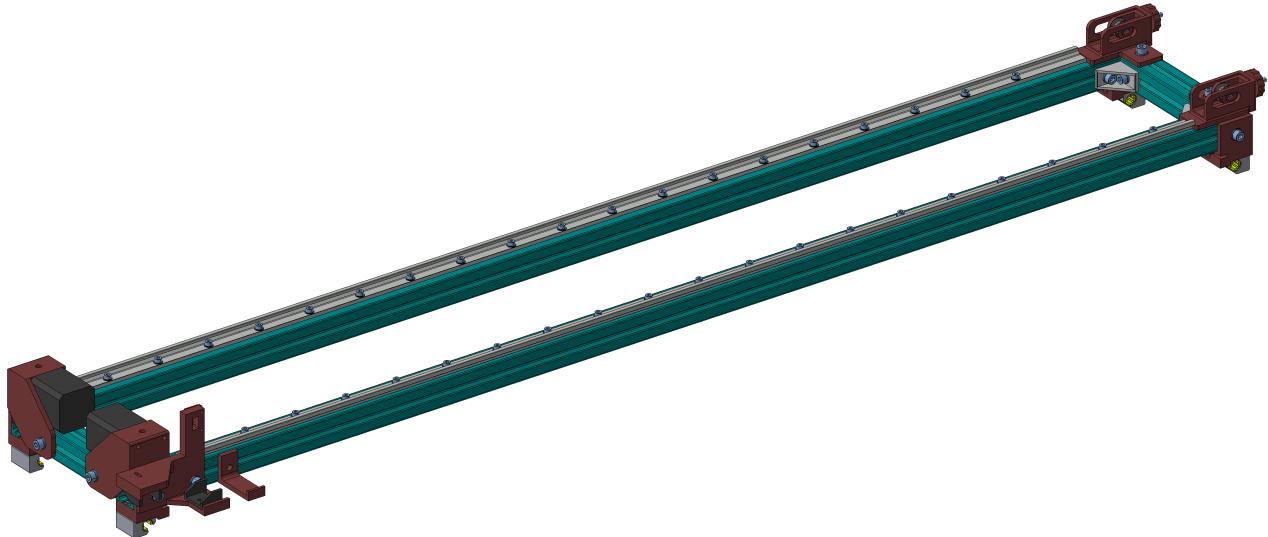


Figura 8: CAD del conjunto *Carro X*.

Carro Y

El carro Y consiste en la estructura en la que se mueve el Carro Z. Contiene los motores y baslas por las que se desplaza el Carro Z, además de los patines necesarios para moverse sobre las guías del Carro X.

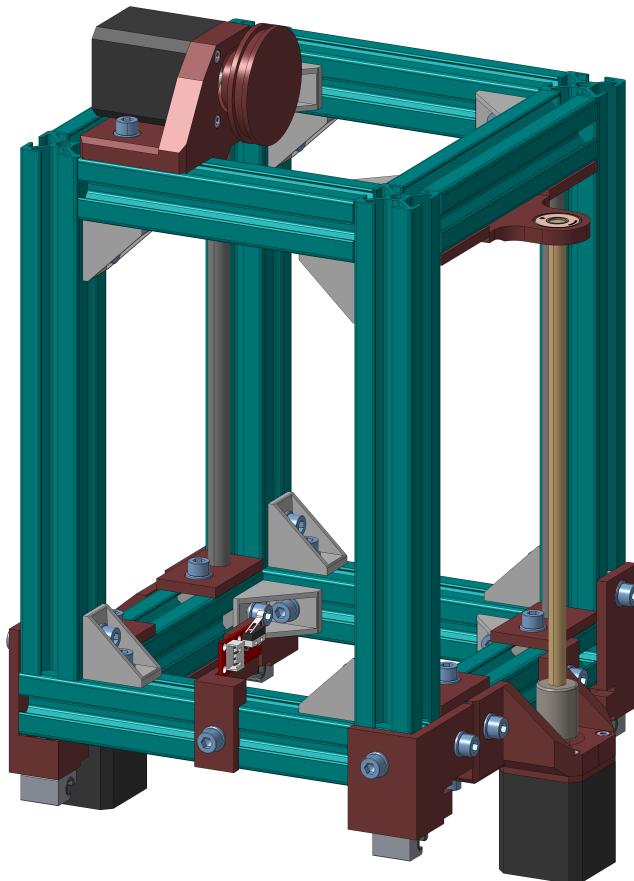


Figura 9: CAD del conjunto *Carro Y*.

Carro Z

El Carro Z contiene la mordaza que integra el tubo guía del OI. Con esto se puede subir y bajar el tubo para evitar las variaciones de altura de las piezas impactadas.

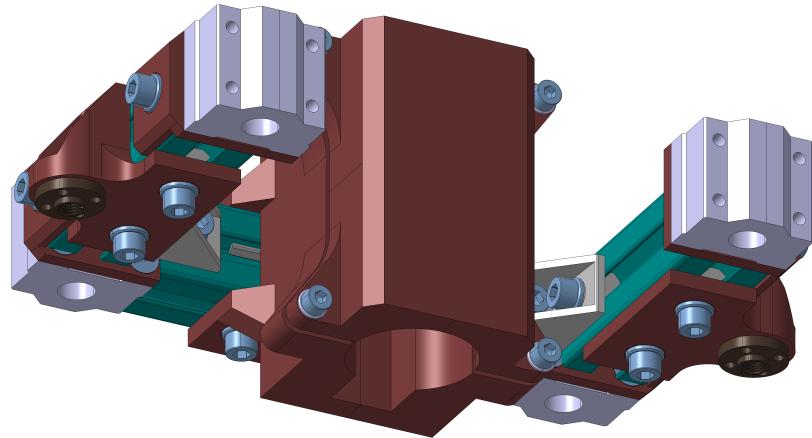


Figura 10: CAD del conjunto *Carro Z*.

Objeto impactador

El OI consiste en: punta impactadora, eje al que se rosca la punta y tapa ferromagnética.

Quitando la tapa se pueden introducir los discos de masa para variar la masa del OI.



Figura 11: Objeto impactador.

Sistema de de recogida de OI

Este subconjunto consiste en el motor del eje A (cuatro eje), que se une mediante un cable con la pieza a la que se integra el electroimán. Por medio de las diferentes poleas se consigue desplazar verticalmente el OI cuando rota el eje A.

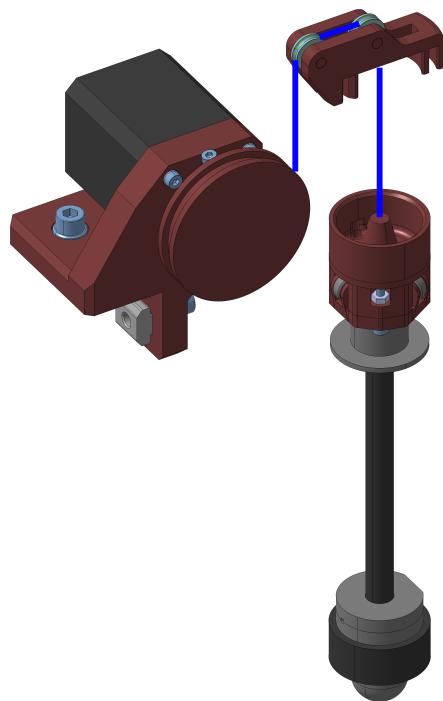


Figura 12: Sistema de reocogida del OI

CAD completo

En la Figura 13 se presenta el modelo final del ICNC. En esta imagen se puede ver una estructura compleja de ejemplo (Costilla del Airbus A380), además está el tubo guía por el que cae el OI.

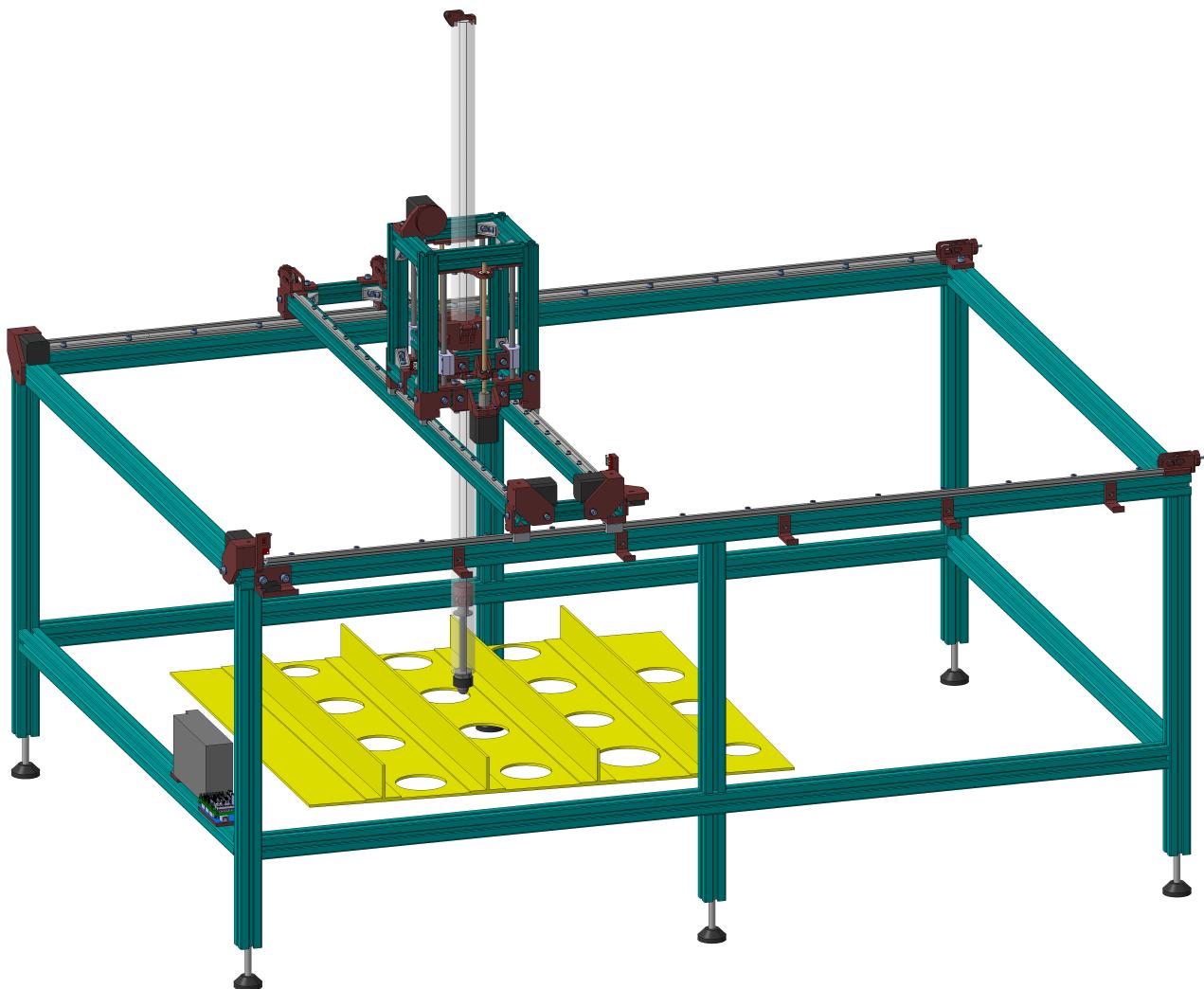


Figura 13: CAD completo del ICNC

2.3.2. Impactador de control numérico fabricado

Una vez completado el modelo CAD, se compraron todos los componentes, se realizó el montaje y posteriormente el cableado.

De la 14 a 15 se puede ver el ICNC completamente montado y cableado. Además, en la Figura 17 se aprecia como el electroimán es capaz de mantener el OI en el aire cuando está activado.

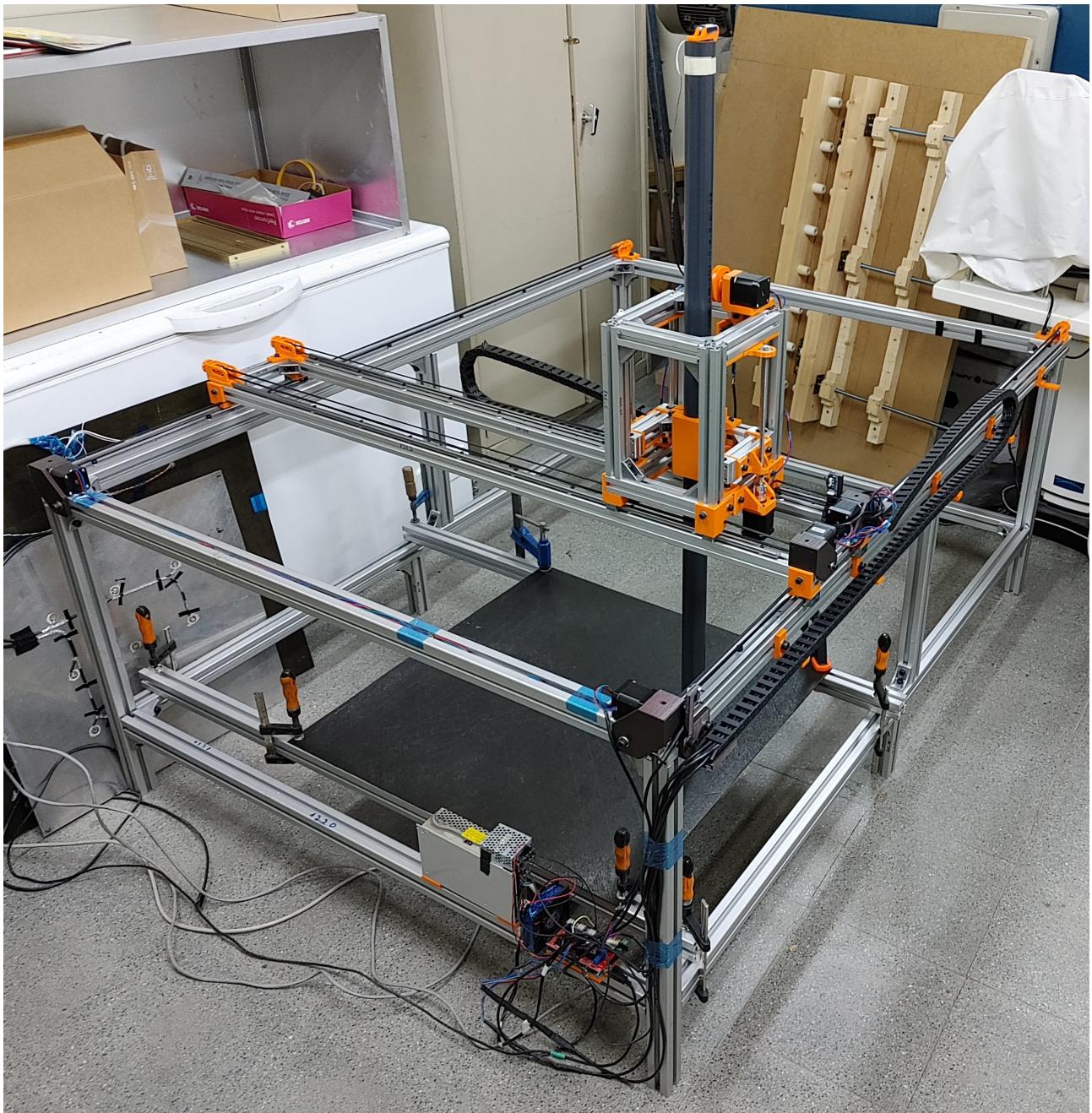


Figura 14: Vista isométrica del ICNC.

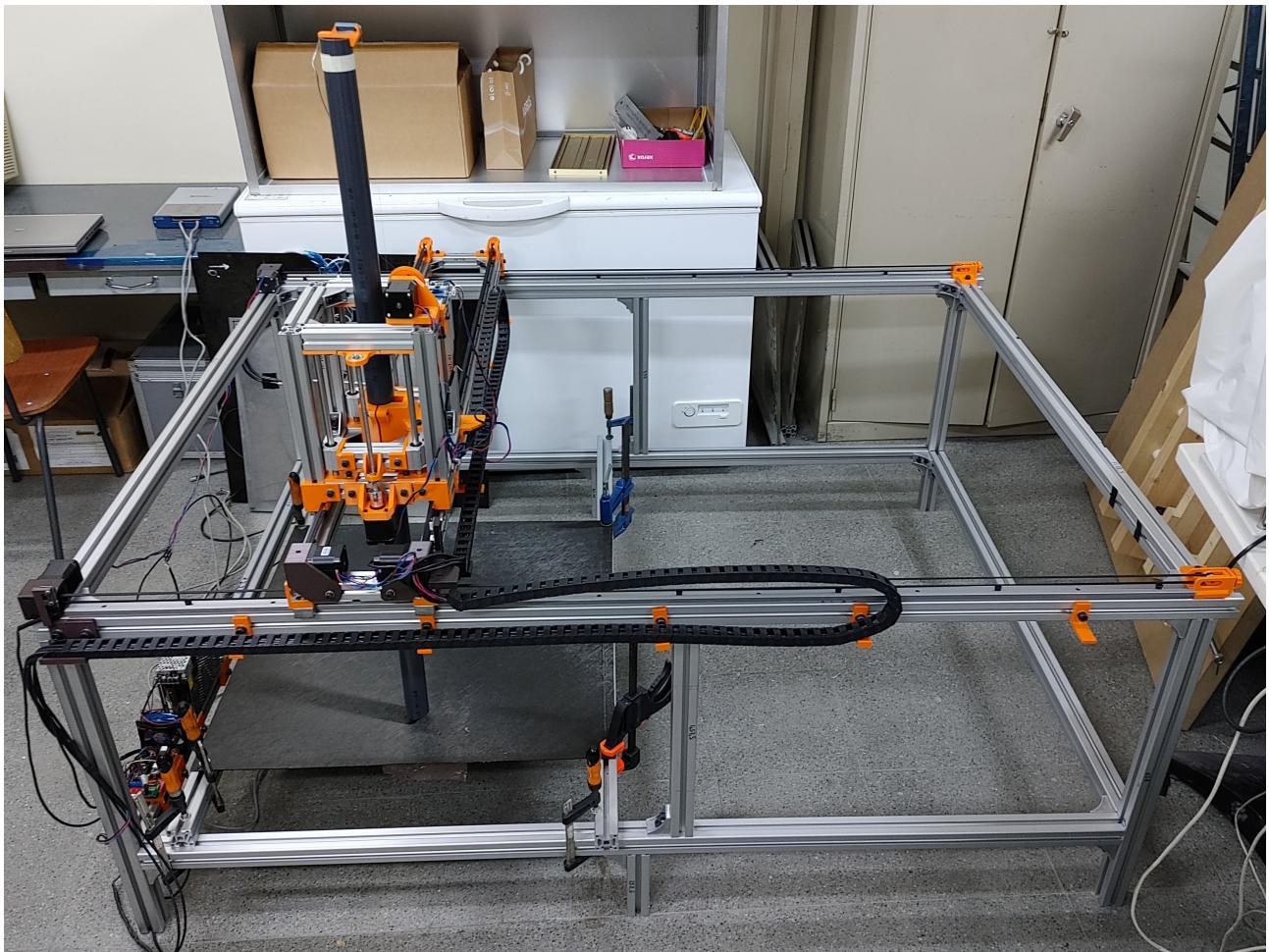


Figura 15: Vista lateral del ICNC.

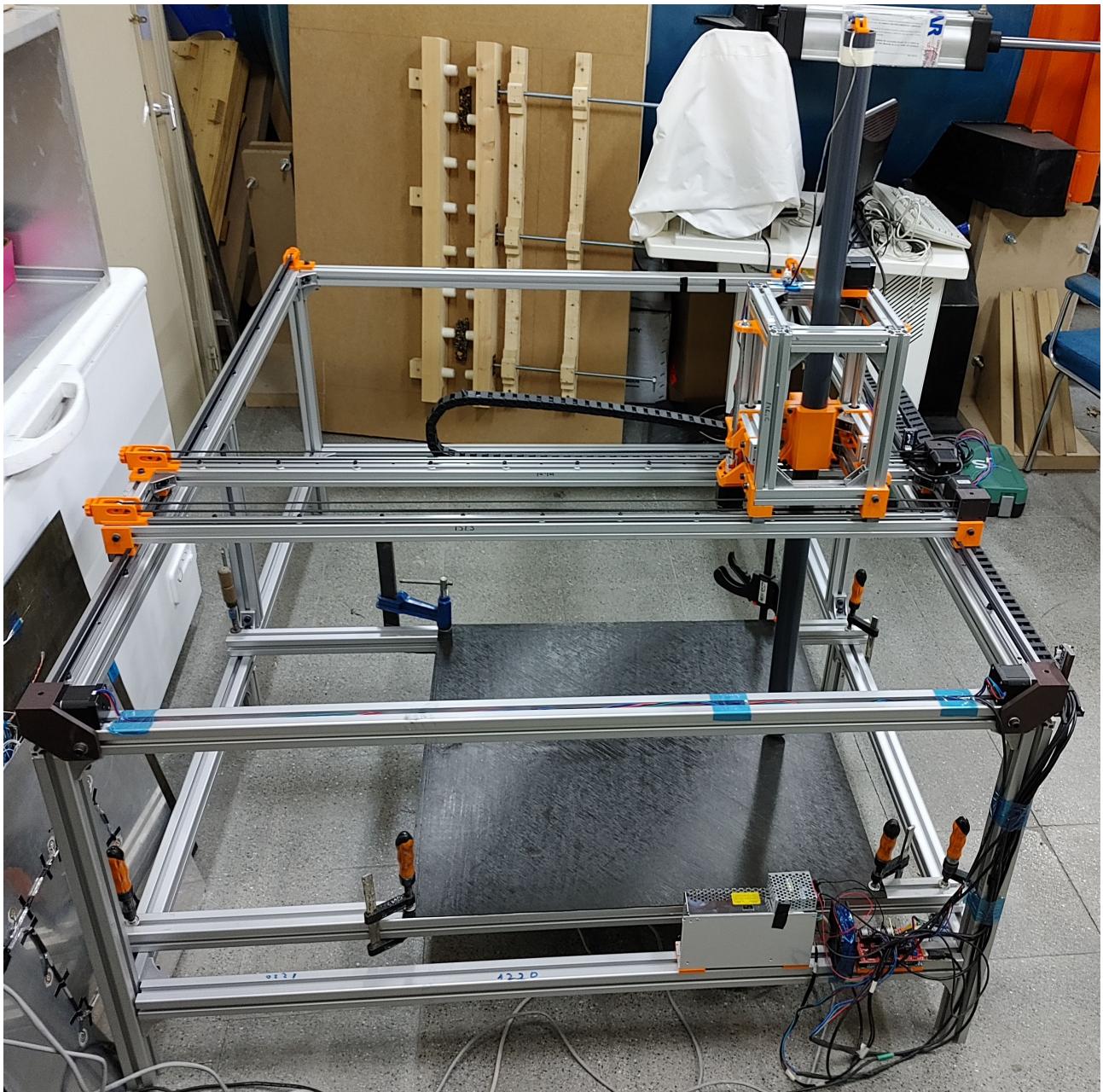


Figura 16: Vista frontal del ICNC.

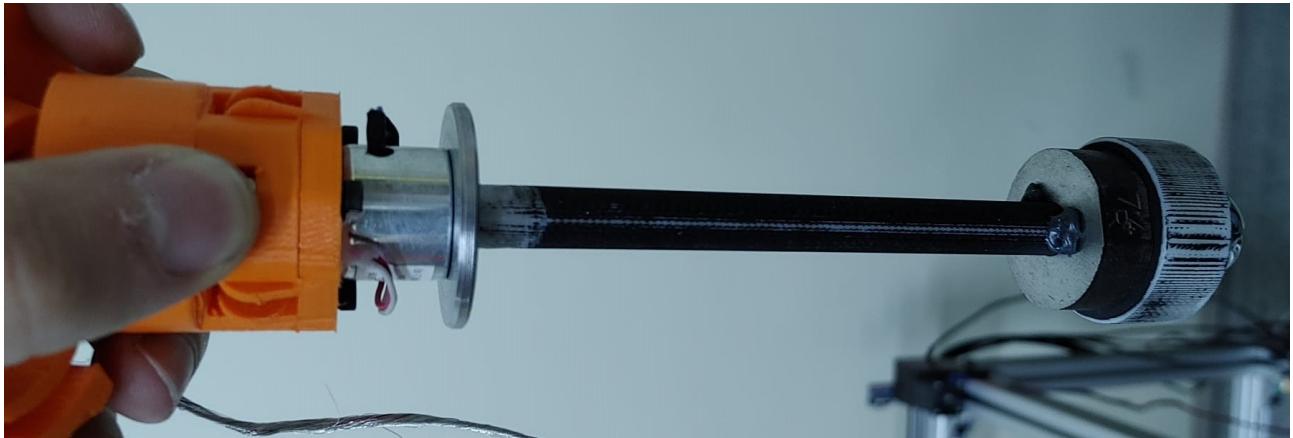


Figura 17: Electoimán activado manteniendo el OI (rotada).

2.4. Software de control

A continuación se va a comentar la clase desarrollada y el proceso de realización de impactos. Este software ha sido íntegramente desarrollado en MATLAB.

2.4.1. Impactor_Class

La clase Impactor_Class se va a dividir en: Atributos y Métodos. Los atributos se pueden entender como las propiedades de la clase y los métodos son las funciones que puede realizar.

Las tareas de esta clase han sido las siguientes:

1. Comunicación con el programa *grbl-Mega-X5*. Contiene implementado todas las funciones necesarios para poder comandar, en código G, las posiciones y encendido y apagado del electoimán.
2. Comunicación con la tarjeta de adquisición de datos. Estos métodos permiten leer durante el tiempo elegido los piezoelectrinos. Así mismo, se procesa el impacto para evitar almacenar un falso impacto o un impacto erróneo.
3. Proceso de ronda de impactos. Con una configuración especificidad de coordenadas, alturas, masa y número de impactos por condición; realiza todo el proceso de forma automática moviendo los motores, adquiriendo el impacto, procesándolo y repitiéndolo de haber sido erróneo.

3. Deep Learning

Deep Learning, DL, es un conjunto de algoritmos de aprendizaje automático (machine learning, ML) que intenta modelar abstracciones de alto nivel en datos usando arquitecturas computacionales que admiten transformaciones no lineales múltiples e iterativas de datos expresados en forma matricial o tensorial, [9].

En este trabajo se ha desarrollado un modelo de DL basado en Redes Neuronales, NNs, el cual ha sido utilizado como herramienta para lograr los objetivos que de exponen a continuación.

3.1. Objetivos del modelo

La importancia de caracterizar un impacto no reside únicamente en su localización, sino también en conocer la masa y velocidad del OI y por lo tanto la energía de dicho impacto.

Por un lado, pudiendo localizar el impacto es más sencillo realizar una inspección detallada de la zona afectada. Además conociendo los daños que puede sufrir el material utilizado en función de las condiciones del impacto; energía, masa y velocidad; es posible conocer en todo momento si una estructura se ha dañado por causa de un impacto y en donde.

En la literatura ya se han realizado estudios de localización de impactos mediante modelos de DL, [10]. Sin embargo, estos estudios se han realizado en piezas de pequeño tamaño (210×210 mm) y utilizando una gran cantidad de coordenadas impactadas (441) y a bajos niveles de energía de impacto (0.44, 1.33 y 2.22 mJ). Hay pocos artículos que traten de obtener la energía del impacto y menos que separen masa y velocidad para un mismo nivel de energía.

Por ello, en este trabajo va a buscar caracterizar impactos partiendo de las siguientes combinaciones:

- Placa rigidizada de fibra de carbono con dimensiones de 630×710 mm. Las condiciones de contorno aplicadas son de empotramiento en el $X = 0$ mm y $X = 630$ mm.
- Se ha impactado en una malla equispaciada de $N_X = 15$ y $N_X = 17$, 255 coordenadas en total. Esta estructura se puede ver en la Figura 18.
- La masa del OI comienza en 60 g y se aumenta de forma progresiva con saltos de 50 g hasta llegar a un máximo de 260 g (5 masas).
- Las velocidades de impacto van desde una altura de suelta de 35 mm, siendo incrementada en 25 mm hasta llegar a un máximo de 260 mm (9 velocidades).

Con estas combinaciones de masa y altura se asegura que se realicen varios impactos a una misma energía con diferentes combinaciones de masa y altura. Esta discriminación para iguales energías es uno de los objetivos principales del trabajo.

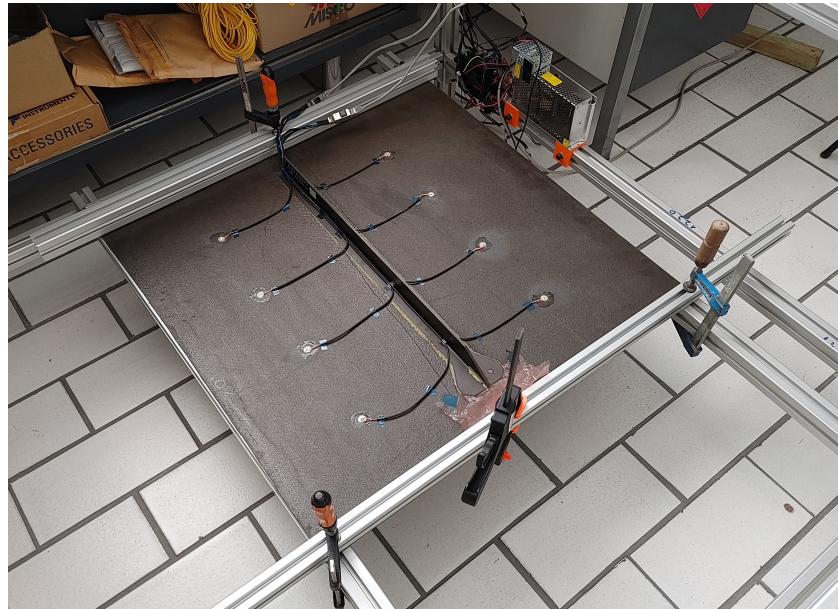


Figura 18: Placa rigidizada utilizada para el estudio de impactos.

3.2. Número de muestras a obtener

Uno de los mayores interrogantes del uso de DL es saber cuantas muestras son necesarias para entrenar un modelo. En la literatura se proponen diversas formas de afrontar este problema, [11] [12]. Sin embargo, como los problemas a resolver son altamente dependientes de la propia naturaleza de los datos a utilizar hace que sea difícil utilizar estudios anteriores como referencia.

Para este trabajo se ha afrontado el problema de obtención de muestras de la siguiente forma.

Se va a multiplicar el número de combinaciones de la propiedad más densa por un factor. De esta forma se tendrá un primer número mínimo de impactos a realizar (combinando el resto de variables) con el que poder entrenar a la red. Una vez que se hayan realizado los impactos, se procederá a entrenar la red.

Si se considera que el funcionamiento de la red no es lo suficientemente bueno, se aumentará este factor y se obtendrán más impactos.

De este modo, se puede optimizar el tiempo de impactación y evaluar de forma simultánea el nivel de *overfit* del modelo.

Se comenzó con un número inicial de impactos igual a $255 \times 100 = 25500$ y finalmente el *dataset* de impactos terminó siendo formado por un total de 44375 impactos válidos.

3.3. Preparación de impactos

Una vez obtenidos todos los impactos es necesario juntarlos todos en una única matriz, ídem con sus etiquetas (posición X, Y, masa y velocidad del impacto).

En este apartado se va a detallar la metodología utilizada para preparar y preprocesar los impactos que posteriormente se utilizarán para alimentar a la NN.

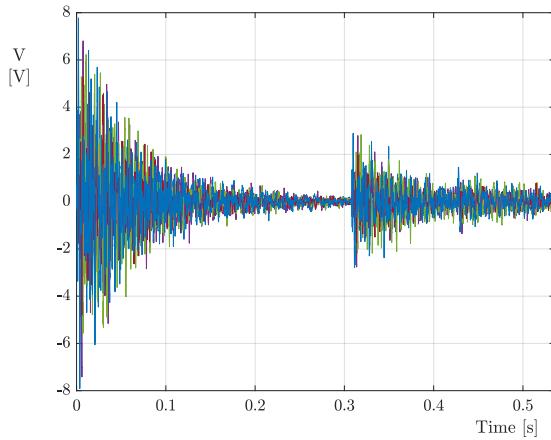
Los pasos seguidos son los siguientes:

1. Analizar los impactos, uno a uno, para comprobar que no es erróneo. Si lo es, se descartará.
2. Almacenar los impactos en una matriz y las etiquetas en una tabla.
3. Generar los índices de los impactos que se van a utilizar para entrenamiento, validación y test.
Se han generado 4 combinaciones diferentes ya que se va a evaluar el funcionamiento de la red mediante la técnica de *k-Fold Cross Validation*.
4. Normalizar (0-1) la matriz de datos y las etiquetas utilizando únicamente el subconjunto de entrenamiento.
5. Guardar la matriz de datos como archivo **.mat** de versión '**-7.3**' ya que los datos ocupan alrededor de 5 Gb. La tabla de etiquetas normalizada se guarda como archivo **.csv**.
6. Subir las matrices a Google Drive.

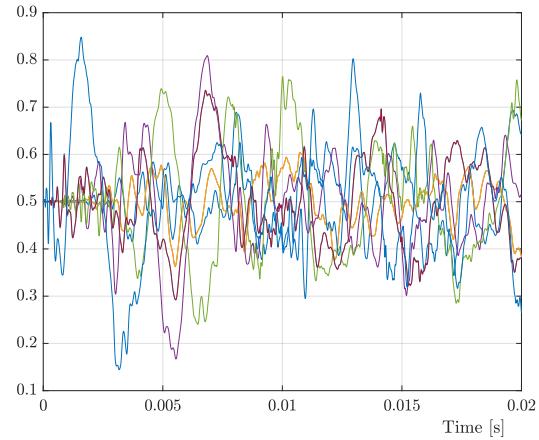
Los datos se han preprocesado siguiendo las indicaciones del libro *Data Preparation for Machine Learning*, [13].

Una vez se tienen los impactos procesados en Google Drive se puede comenzar el desarrollo de la NN. Para ello se ha utilizado Google Colab y la librería PyTorch.

La comparación de un impacto antes y después de ser procesado se puede ver en la Figura 19.



(a) Impacto antes de ser procesado.



(b) Impacto después de ser procesado.

Figura 19: Comparación de un impacto antes y después de ser procesado.

3.4. Arquitectura utilizada

Este trabajo no busca construir un clasificador. No se quiere obtener unos resultados discretos, no que quiere poder obtener únicamente las etiquetas con las que se ha entrenado a la red, sino que se pretende desarrollar un modelo regresor.

Un impacto no tiene por qué coincidir con uno de los puntos donde se han recogido datos, las estructuras son continuas y los datos solo se pueden obtener en zonas discretas, dónde se hayan instalado los sensores. De este modo, se busca entrenar una red que de una salida continua, en definitiva, se busca que la red sea capaz de aprender la física de la estructura impactada para que ante impactos nuevos, sea capaz de predecir sus características con el menor error posible.

No hay ningún método estandarizado para construir una red. Generalmente se parte de un modelo sencillo y se va aumentando su complejidad hasta lograr los resultados deseados.

A pesar de ello, en este trabajo se han seguido las recomendaciones del libro *Better Deep Learning. Train Faster, Reduce Overfitting, and Make Better Predictions*, [13].

La red desarrollada está compuesta por dos partes principales, compresor y regresor, a continuación se va a detallar la función de cada una de ellas.

- **Compresor**

El compresor es una serie de capas que procesan la señal de cada sensor de forma individual, comprimen su información y la preparan para que pase a la siguiente etapa.

En la Figura 20 se puede ver la secuencia de capas de esta subred.

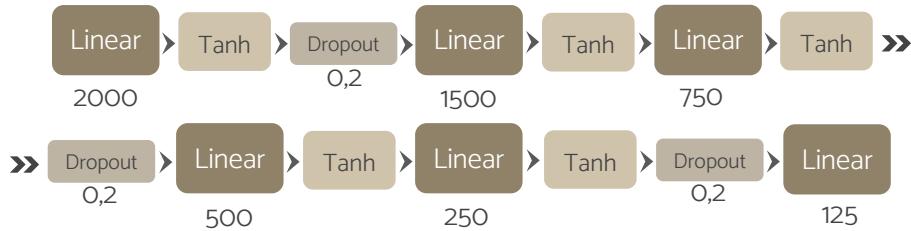


Figura 20: Secuencia de capas del compresor.

• Regresor

El regresor coge la información comprimida de todos los sensores de la estructura y devuelve la propiedad deseada. Como se ha dicho anteriormente, la red devuelve un valor continuo, no una clase.

Habrá tantos regresores como propiedades se quieran obtener por modelo.

En la Figura 21 se puede ver la secuencia de capas de esta subred.

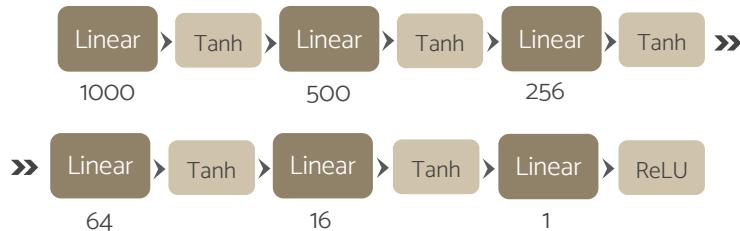


Figura 21: Secuencia de capas del regresor.

La arquitectura completa se muestra en la Figura 22.

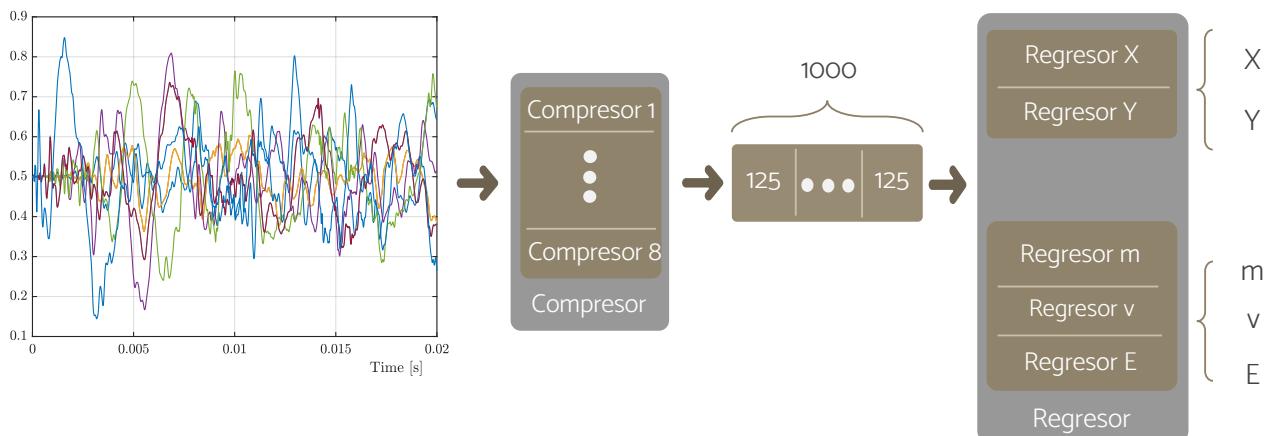


Figura 22: Arquitectura del funcionamiento completo del modelo de detección de impactos..

En esta Figura se representa de forma esquemática como funciona la red diseñada. La medida de cada sensor entra a la fase de compresión, se reduce la dimensionalidad de forma individual quedando

solo la información más relevante. Antes de entrar al regresor se concatenan todos los sensores para que en la etapa del regresor se obtenga la variable requerida.

Es importante destacar que se ha desarrollado un modelo para la localización de impactos y otro para las propiedades energéticas, pero la arquitectura es idéntica.

3.5. Entrenamiento

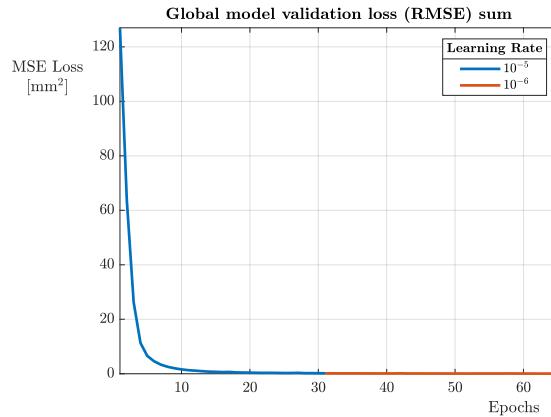
Las NN son algoritmos de aprendizaje supervisado que necesitan de un proceso de entrenamiento para alcanzar un funcionamiento óptimo.

En este trabajo se ha decidido dividir el total de datos en 50 % para entrenamiento, 25 % para validación y 25 % para test.

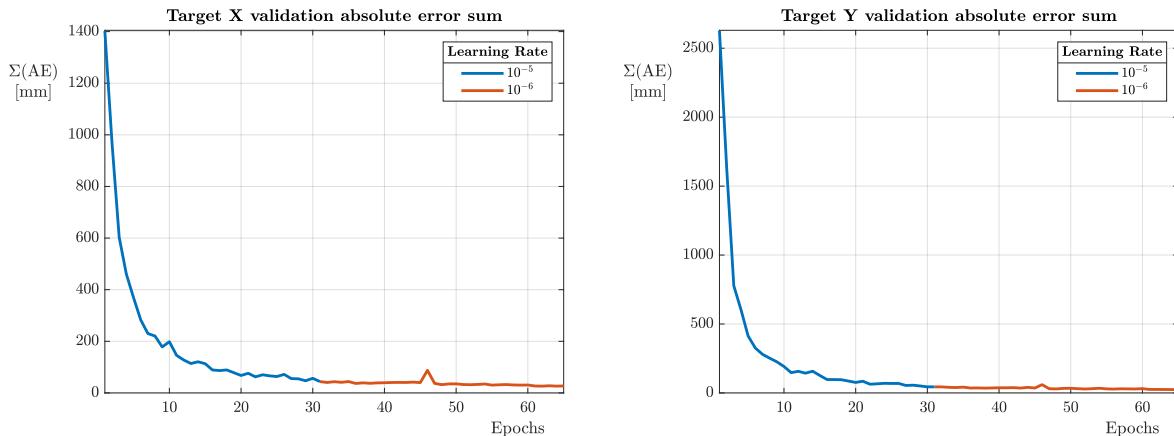
La necesidad de un subconjunto de validación reside en que el modelo debe ser evaluado con impactos diferentes a los de entrenamiento durante el proceso de aprendizaje para asegurar que está aprendiendo correctamente.

La evolución del entrenamiento para un modelo de localización se puede observar en la Figura 23. Aquí se aprecia como, a medida que pasan épocas de entrenamiento (número de evaluaciones completas de los datos), los indicadores utilizados para evaluar el funcionamiento de la red bajan. De haber algún problema durante el entrenamiento volverían a subir.

Por otro lado, los datos de test son necesarios para obtener un *performance* final del modelo con un grupo de impactos que nunca haya sido utilizado previamente por el modelo durante la fase de entrenamiento.



(a) *Loss* acumulado del modelo completo durante validación.



(b) Error absoluto acumulado en validación del *target* X.
(c) Error absoluto acumulado en validación del *target* Y.

Figura 23: Comparación de un impacto antes y después de ser procesado.

En la Figura anterior se puede observar que para este trabajo se ha decidido utilizar un entrenamiento con dos *Learning Rates* diferentes, uno inicial de 10^{-5} seguido por 10^{-6} .

El proceso de entrenamiento era detenido cuando no se apreciaba una disminución significativa de los indicadores utilizados.

4. Resultados

Una vez completadas todas las rondas de impacto se obtuvieron un total de 44375 impactos válidos tras un total de 12 días completos.

Como se ha comentado anteriormente, se hicieron 4 distribuciones diferentes de impactos en los grupos de entrenamiento, test y validación y con ellos se entrenaron 8 redes (4 de localización y 4 de caracterización energética) con el fin de evaluar el funcionamiento del modelo mediante validación cruzada.

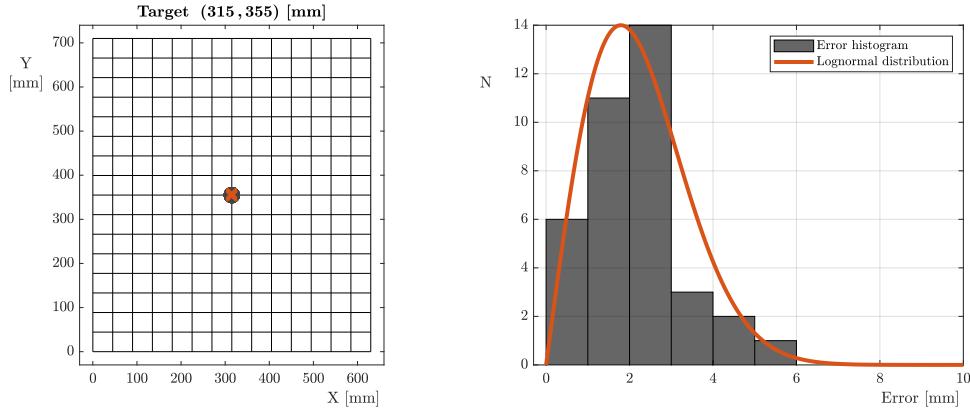
Durante el proceso de entrenamiento la red solo utilizó los subconjuntos de entrenamiento y validación. El subconjunto de test es el utilizado para obtener los resultados del funcionamiento del modelo ya que estos datos nunca han sido introducidos a la red con anterioridad, por lo que el modelo no ha ajustado sus parámetros con estos datos.

A continuación se presentan los resultados obtenidos tras evaluar los diferentes modelos con los datos de test para el modelo de localización y para el modelo de caracterización energética.

4.1. Localización

Cuando se introduce en el modelo de localización un impacto, éste devuelve una coordenada X y una coordenada Y. Al ser el modelo una regresión, su predicción no es exactamente la coordenada objetivo, sino que hay un error con el objetivo.

En la Figura 24a se puede observar como hay una nueve de predicciones que están cerca del punto objetivo, pero siempre se quedan a una cierta distancia, siempre se tiene un error de predicción. La distribución de error de predicción se presenta en la Figura 24b. Se ve como el error sigue una distribución Lognormal, por lo tanto, se puede evaluar el modelo de forma estadística mediante la media (μ) y la desviación típica (σ) de la distribución de errores.



(a) Predicciones de impacto para un único objetivo.
(b) Distribución de error de distancia entre la predicción y objetivo.

Figura 24: Predicción de la red y distribución de errores.

Por lo tanto, a continuación se van a mostrar los valores medios de error en predicción de coordenada X, coordenada Y y del error entre distancia predicha y objetivo, así mismo también se presenta la desviación típica asociada a la distribución para cada coordenada impactada.

Estos resultados se han representado con mapas de calor a continuación.

4.1.1. Modelo 1

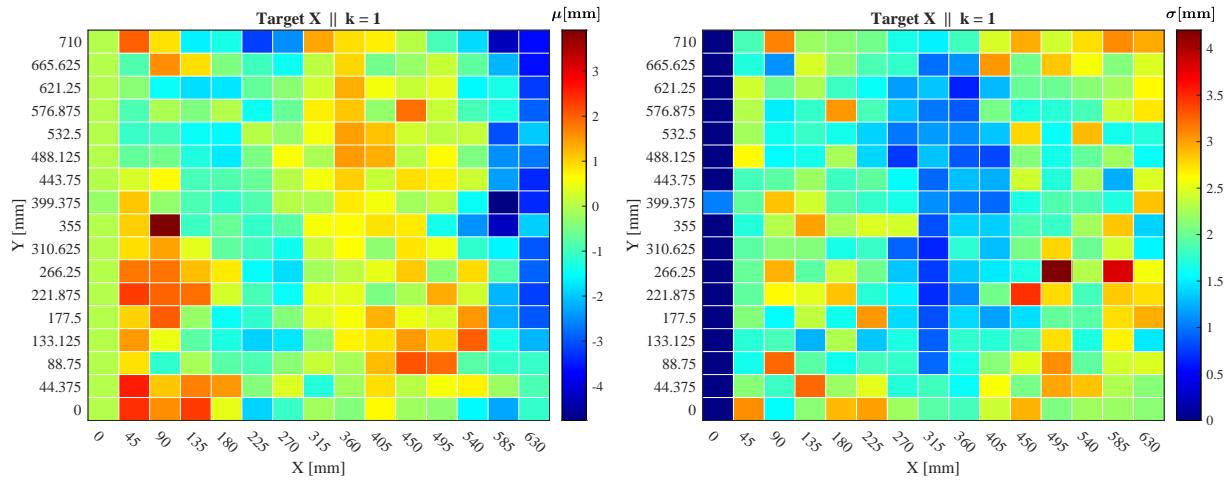


Figura 25: *Performance* de la red al predecir la coordenada X del impacto para la distribución 1.

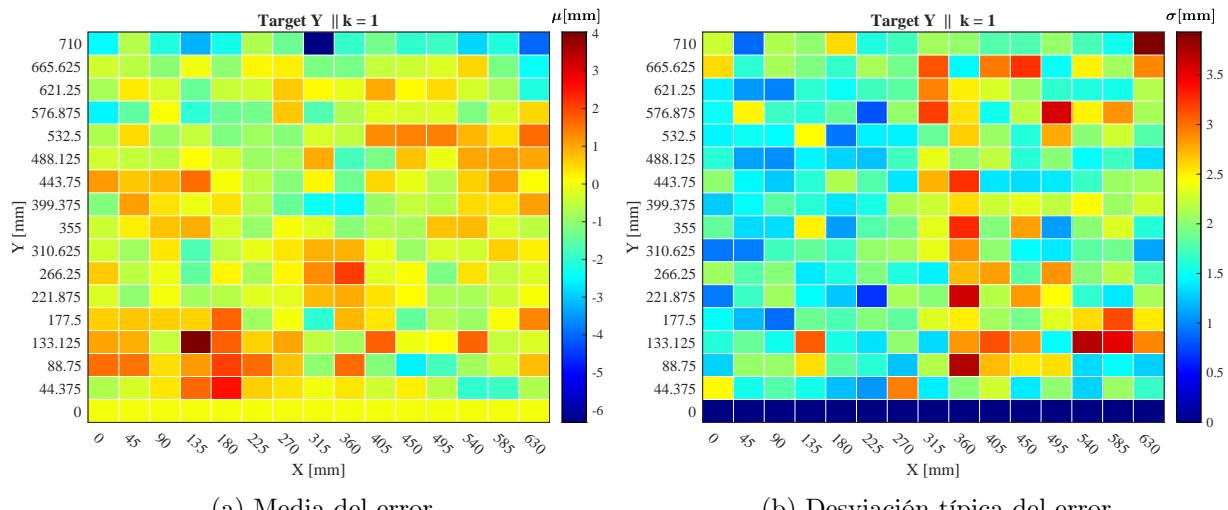


Figura 26: *Performance* de la red al predecir la coordenada Y del impacto para la distribución 1.

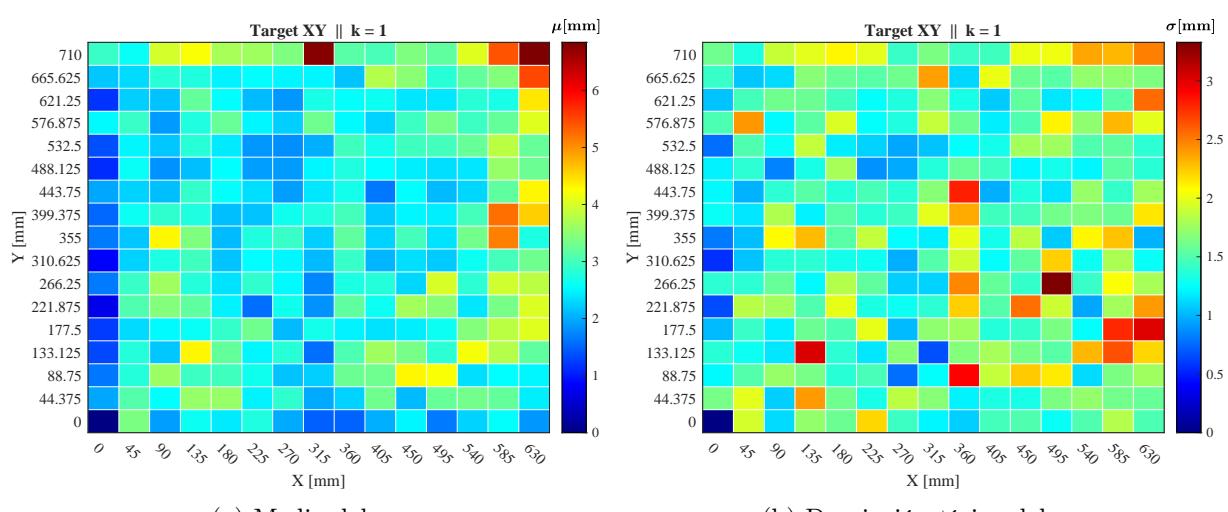


Figura 27: *Performance* de la red al predecir la posición del impacto para la distribución 1.

4.1.2. Modelo 2

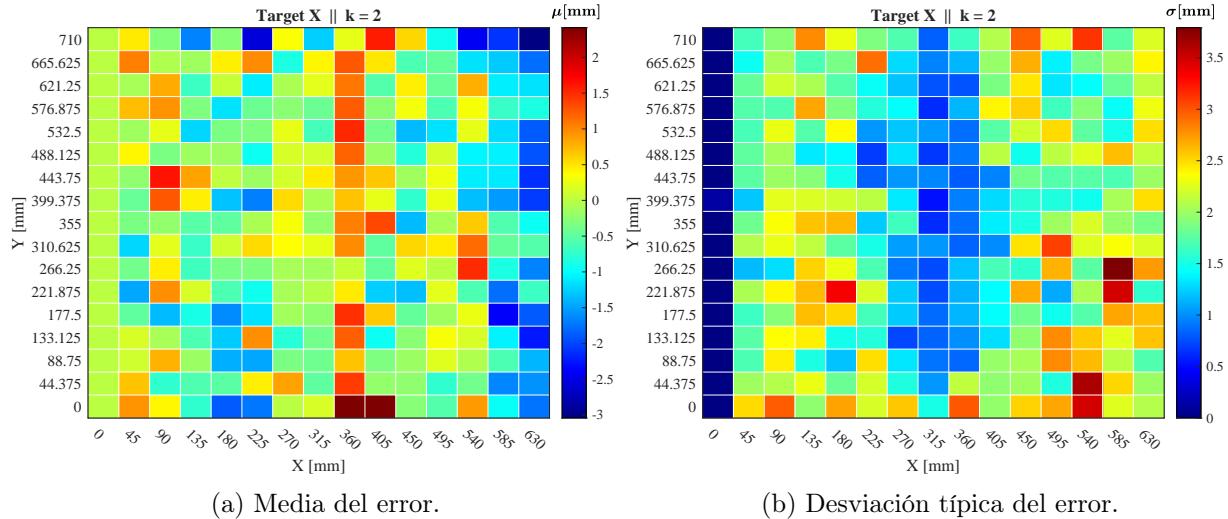


Figura 28: *Performance* de la red al predecir la coordenada X del impacto para la distribución 2.

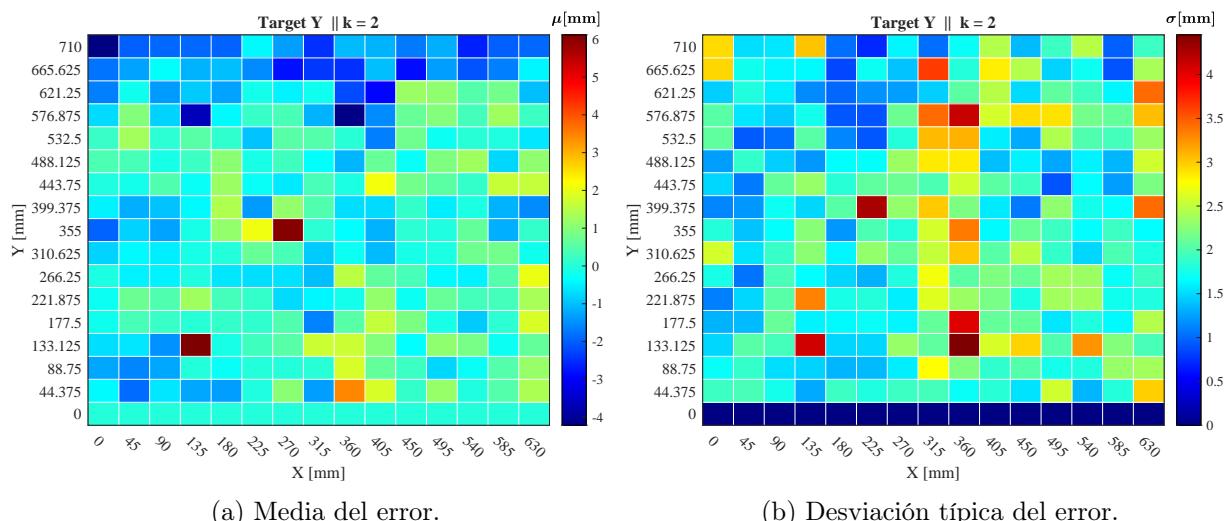


Figura 29: *Performance* de la red al predecir la coordenada Y del impacto para la distribución 2.

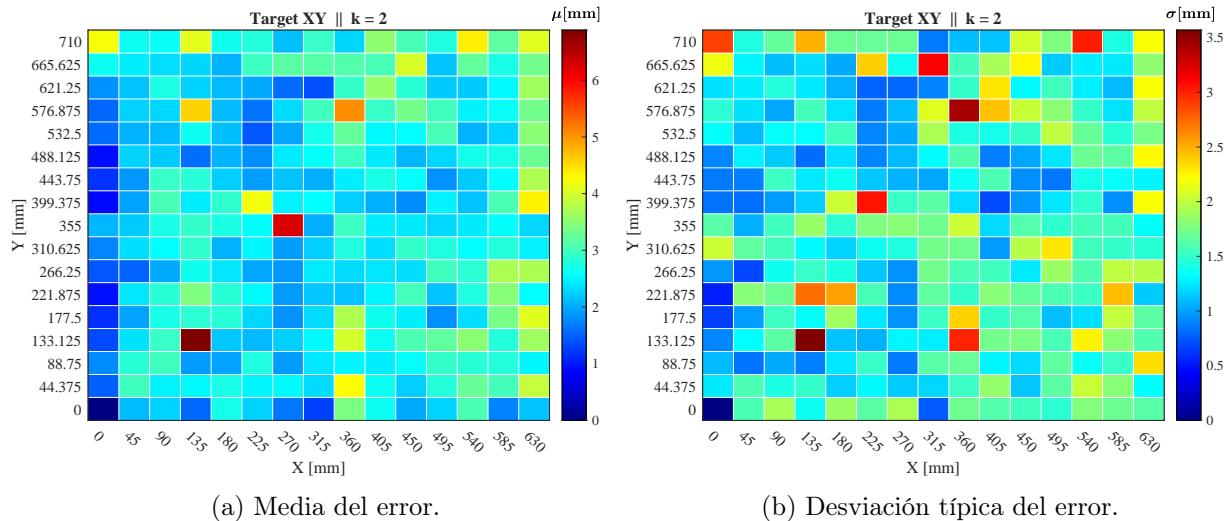


Figura 30: *Performance* de la red al predecir la posición del impacto para la distribución 2.

4.1.3. Modelo 3

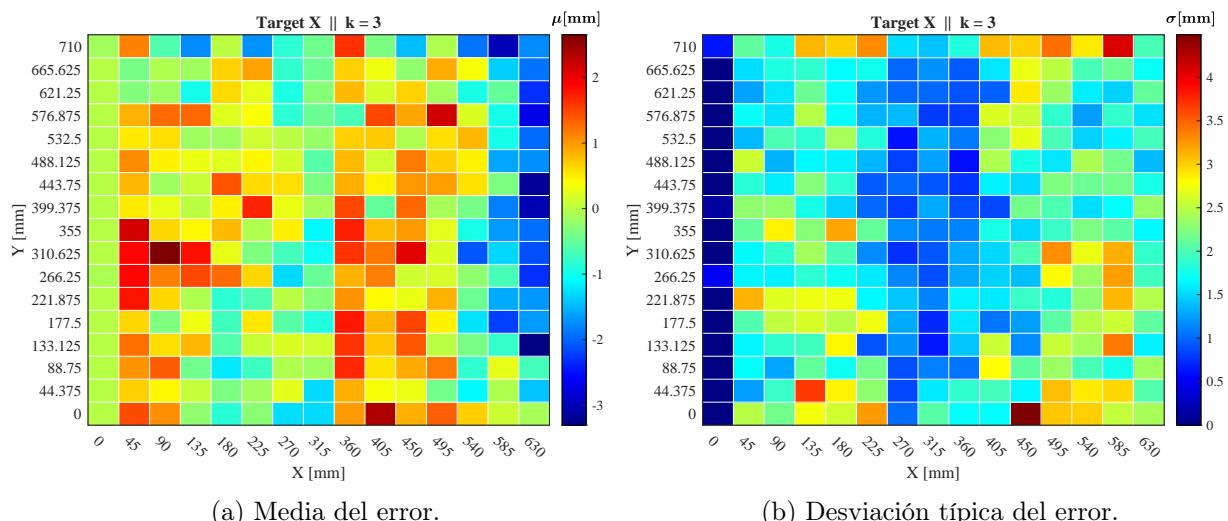


Figura 31: *Performance* de la red al predecir la coordenada X del impacto para la distribución 3.

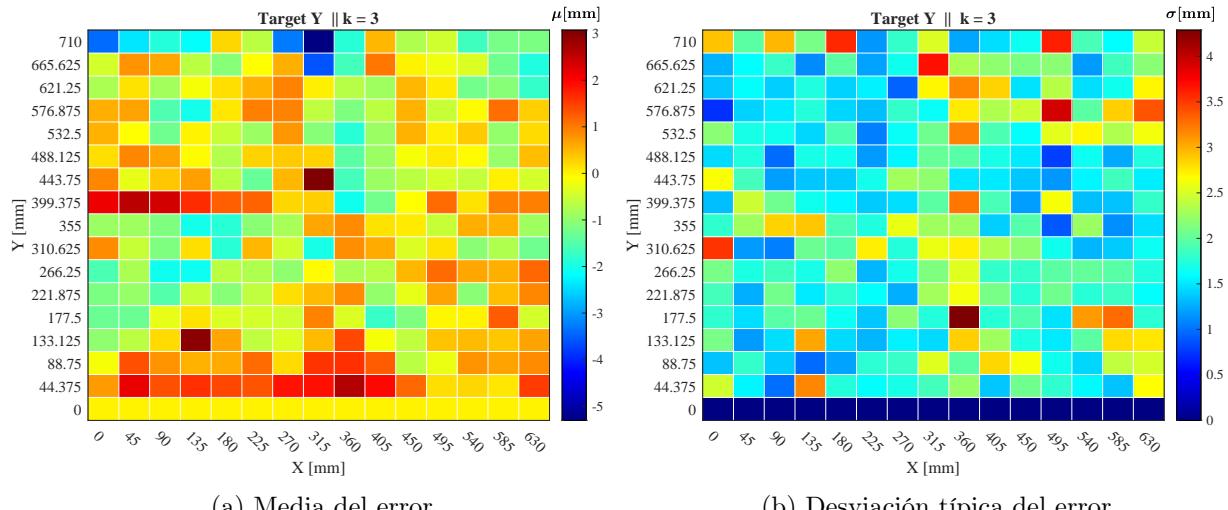


Figura 32: *Performance* de la red al predecir la coordenada Y del impacto para la distribución 3.

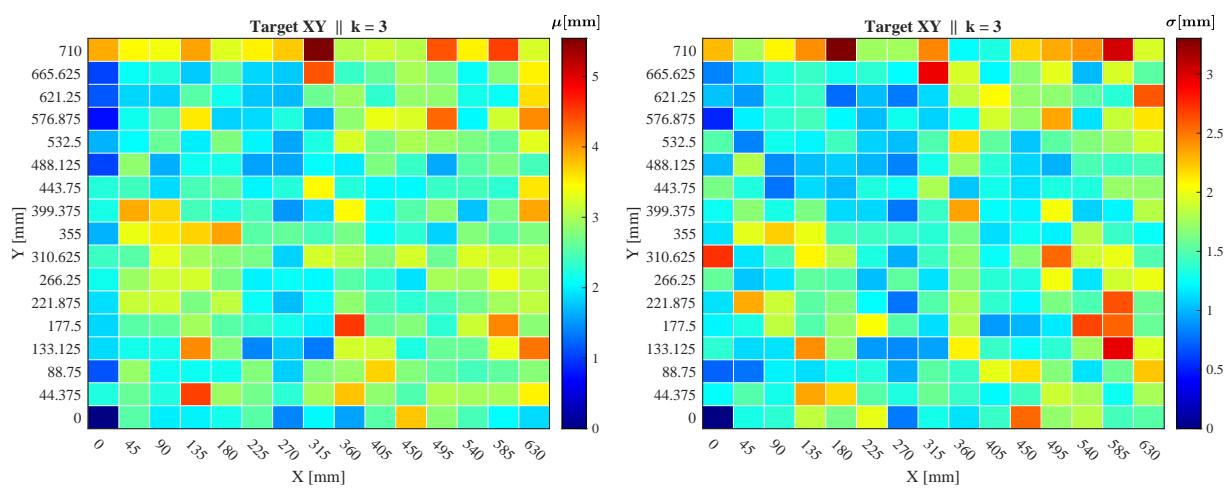


Figura 33: *Performance* de la red al predecir la posición del impacto para la distribución 3.

4.1.4. Modelo 4

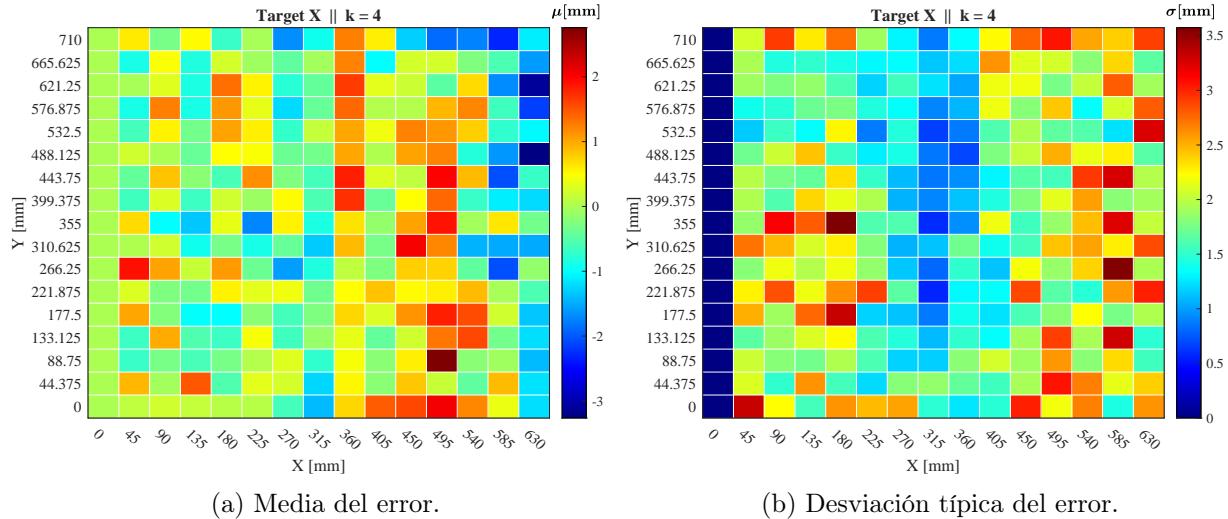


Figura 34: *Performance* de la red al predecir la coordenada X del impacto para la distribución 4.

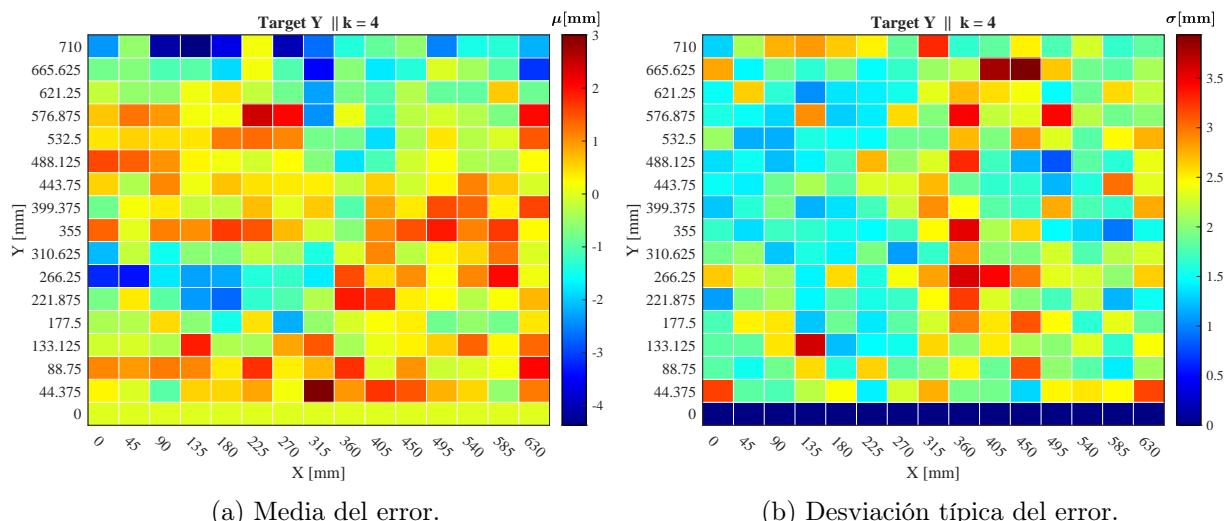


Figura 35: *Performance* de la red al predecir la coordenada Y del impacto para la distribución 4.

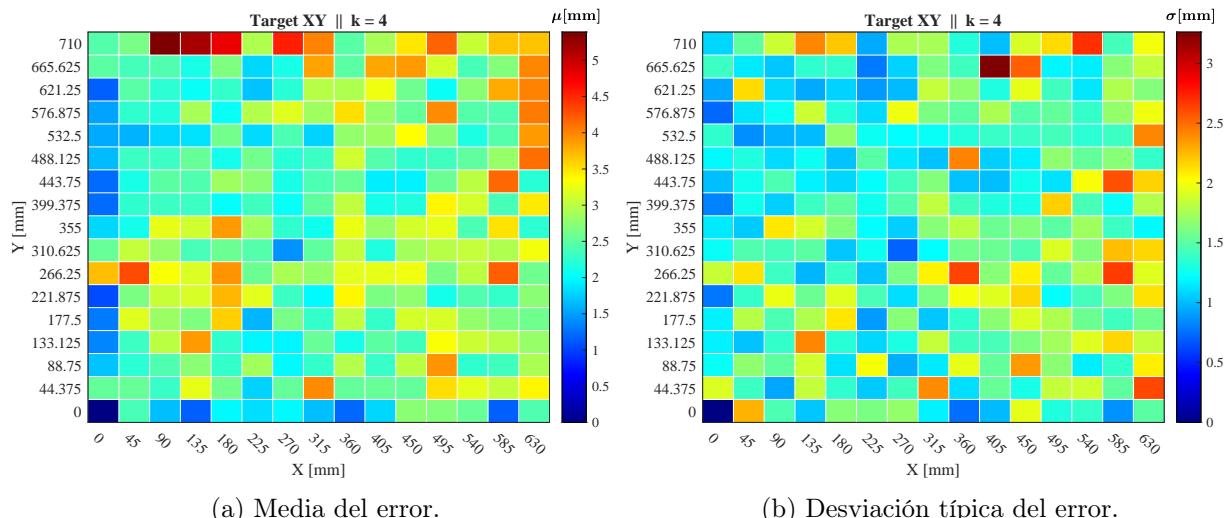


Figura 36: *Performance* de la red al predecir la posición del impacto para la distribución 4.

4.1.5. Resumen del modelo de localización

Una vez presentados los resultados estadísticos de las coordenadas individuales se va a mostrar el error global del modelo calculando su error medio y desviación típica total.

En la Figura 37 se han representado las distribuciones de error obtenidas con el primer modelo.

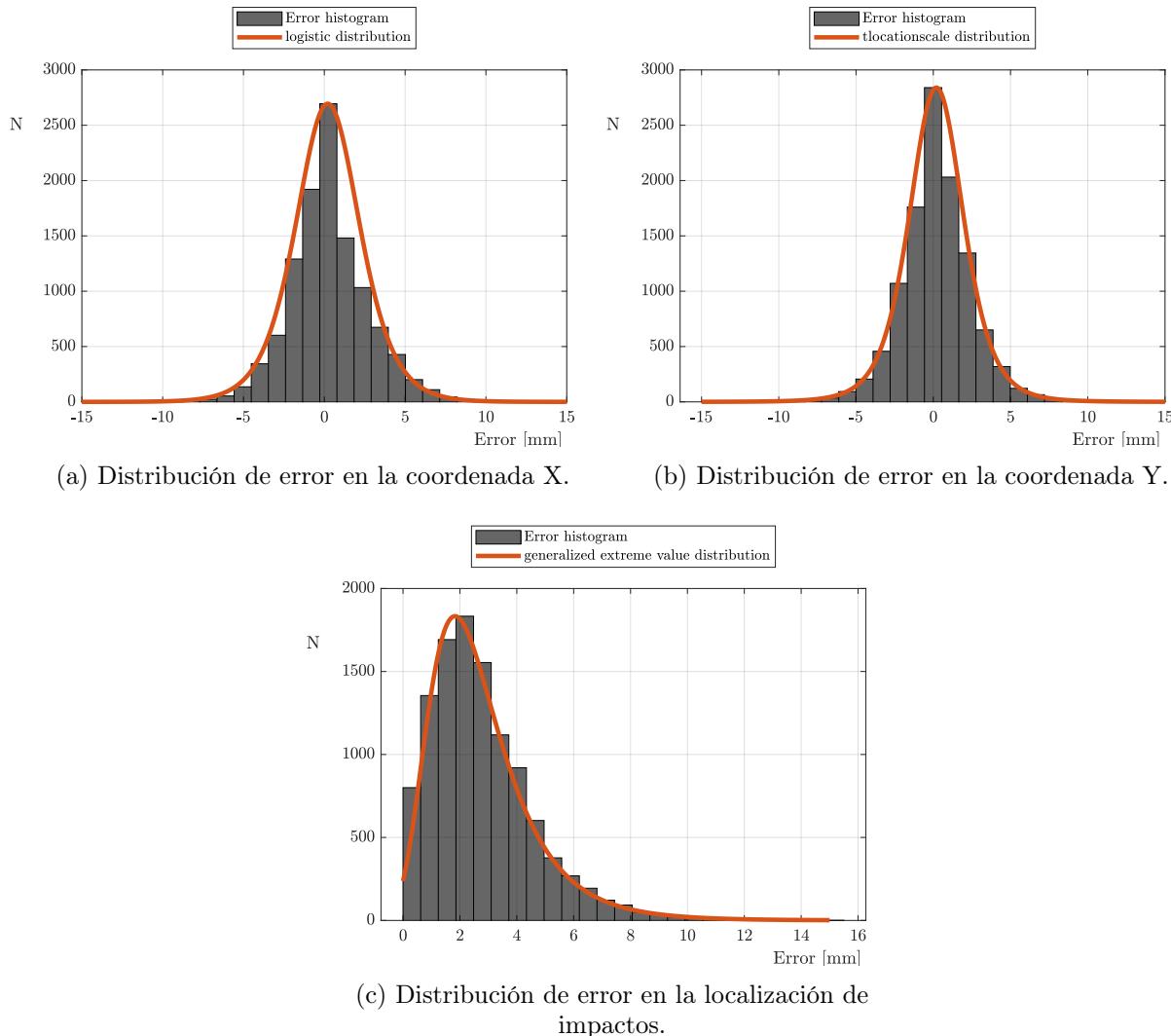


Figura 37: Error del modelo entrenado con la distribución 1.

Y los resultados del *performance* completo de todos los modelos entrenados se han recogido en la Tabla 1.

Tabla 1: *Performance* de la red de localización de impactos.

Modelo	μ X [mm]	σ X [mm]	μ Y [mm]	σ Y [mm]	μ loc. [mm]	σ loc. [mm]
1	0.29	2.41	0.20	2.26	2.78	1.83
2	0.25	2.06	0.14	2.34	2.60	1.75
3	-0.04	2.21	0.13	2.23	2.62	1.73
4	0.02	2.12	0.09	2.37	2.68	1.72

4.2. Caracterización energética

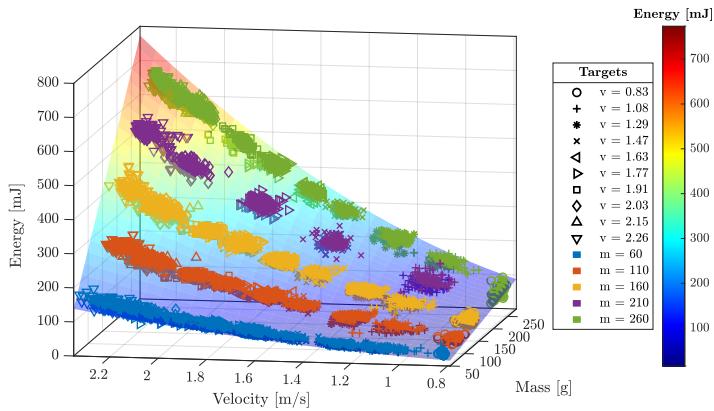
Para visualizar el funcionamiento del modelo que predice la masa, velocidad y energía de un impacto, se ha pintado sobre la superficie obtenida al evaluar la energía producida por un impacto con combinación de masa y altura, Ec (2), las predicciones de la red.

En las Figuras que se mostrarán a continuación se ha representado los datos de la siguiente forma:

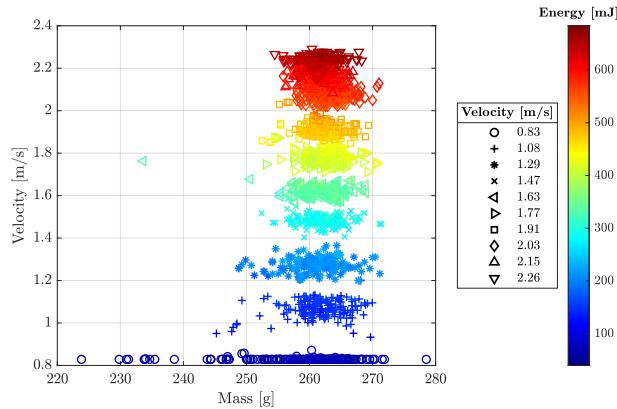
- Color: el color de cada marcador representa la masa objetivo del impacto.
- Marcador: el icono del marcador representa la velocidad objetivo de cada impacto.
- Superficie: representa la energía objetivo para combinación de masa y velocidad de impacto.

A continuación se muestran los resultados de los cuatro modelos entrenados.

4.2.1. Modelo 1



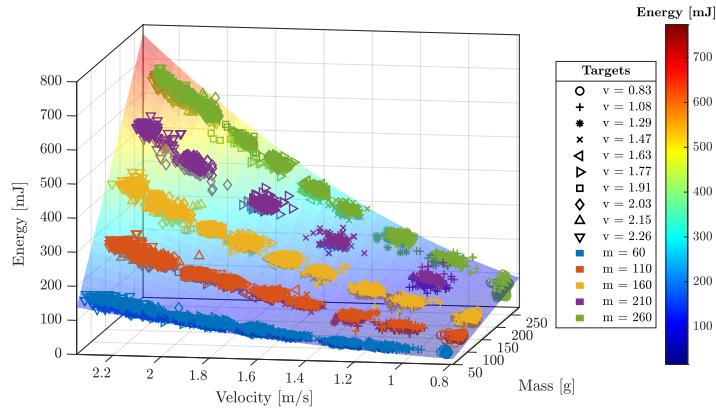
(a) Vista tridimensional de las predicciones.



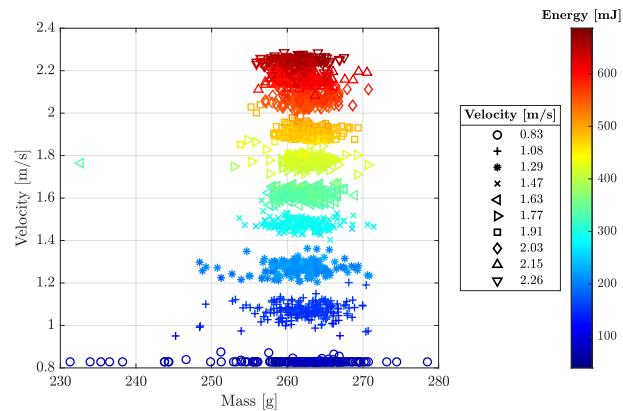
(b) Detalle de $m = 260$ g.

Figura 38: *Performance* de la red al predecir las propiedades energéticas de los impactos para la distribución 1.

4.2.2. Modelo 2



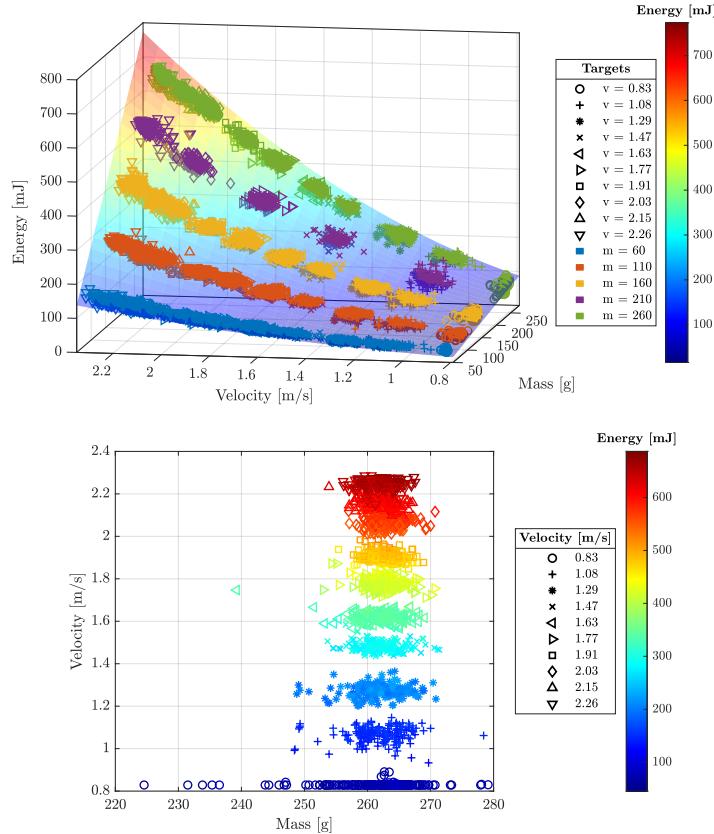
(a) Vista tridimensional de las predicciones.



(b) Detalle de $m = 260$ g.

Figura 39: *Performance* de la red al predecir las propiedades energéticas de los impactos para la distribución 2.

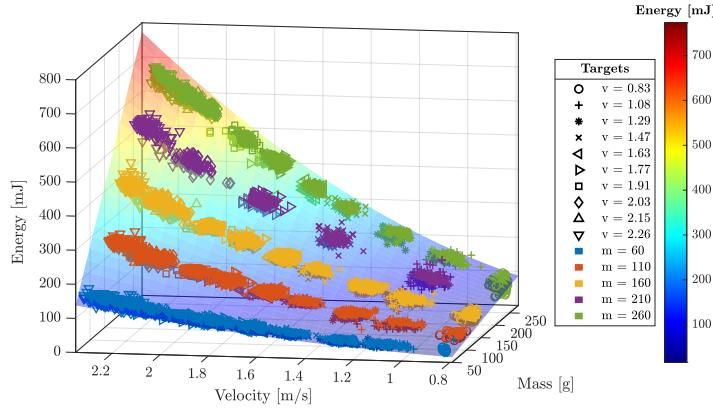
4.2.3. Modelo 3



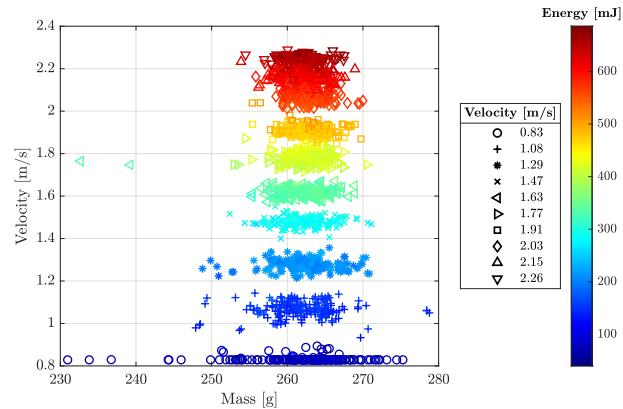
(b) Detalle de $m = 260$ g.

Figura 40: *Performance* de la red al predecir las propiedades energéticas de los impactos para la distribución 3.

4.2.4. Modelo 4



(a) Vista tridimensional de las predicciones.



(b) Detalle de $m = 260$ g.

Figura 41: *Performance* de la red al predecir las propiedades energéticas de los impactos para la distribución 4.

4.2.5. Resumen del modelo energético

Una vez visualizados los resultados de las predicciones de la caracterización energética de los impactos, se va a evaluar el *performance* del modelo de la misma forma que en el modelo de localización, con la media y desviación típica de los errores.

En la Figura 42 se han representado las distribuciones de error obtenidas con el primer modelo.

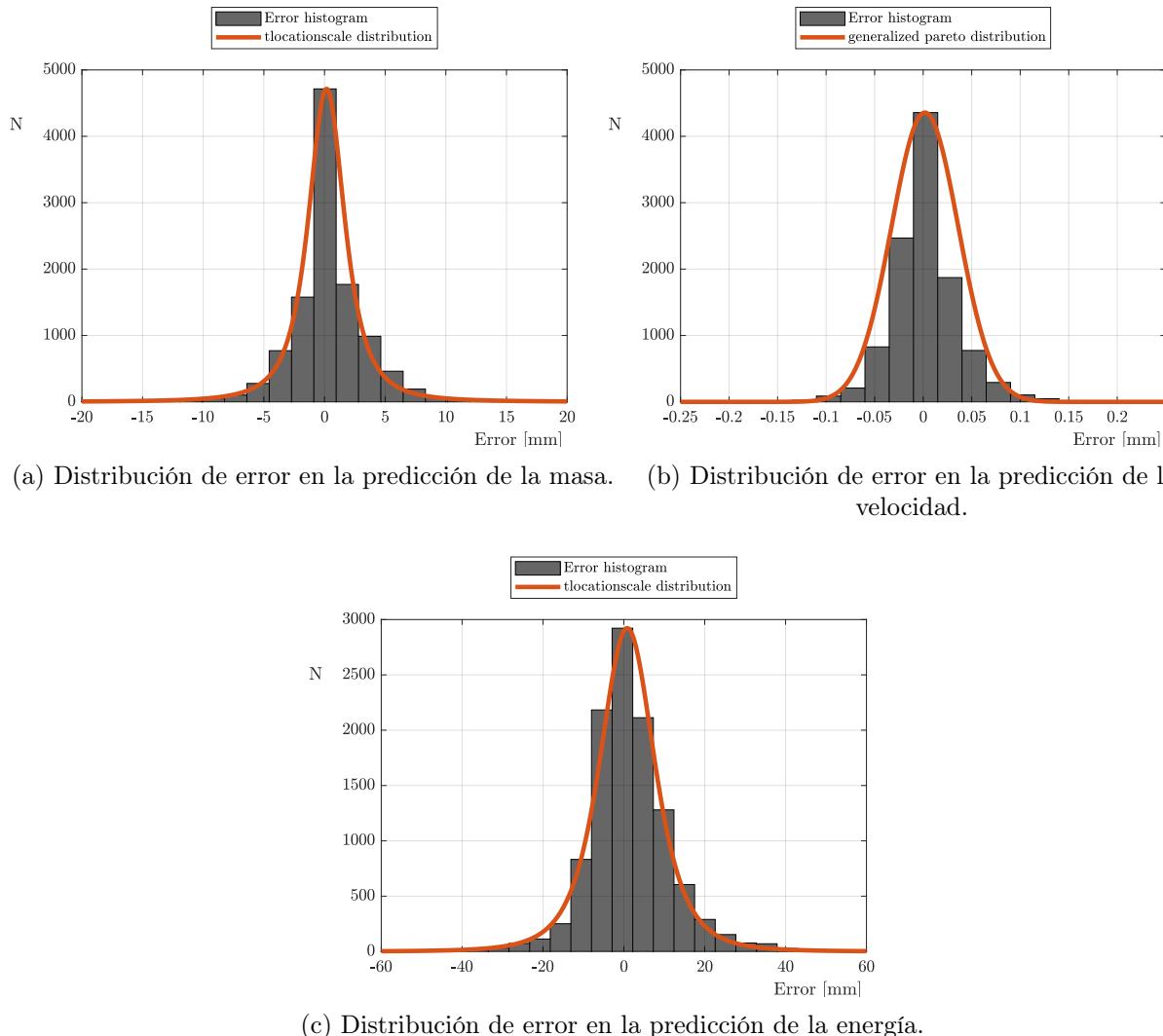


Figura 42: Error del modelo entrenado con la distribución 1.

Y los resultados del *performance* completo de todos los modelos entrenados se recogen en la Tabla 2.

Tabla 2: *Performance* de la red de caracterización energética.

Modelo	μ_m [g]	σ_m [g]	μ_v [$\frac{m}{s}$]	σ_v [$\frac{m}{s}$]	μ_E [mJ]	σ_E [mJ]
1	0.25	3.68	0.0015	0.040	1.34	11.76
2	0.31	3.11	0.0016	0.034	1.41	10.34
3	0.25	3.20	0.0013	0.034	1.39	10.48
4	0.32	3.19	0.0017	0.034	1.44	10.43

5. Conclusiones

Tras el trabajo desarrollado y los resultados obtenidos se ha llegado a las conclusiones que se exponen a continuación.

El objetivo principal de desarrollar una herramienta para localizar y caracterizar impactos se ha conseguido satisfactoriamente.

Los modelos construidos han sido capaces de obtener un error medio y desviación típica medias pequeños en comparación a las dimensiones y valores máximos de las variables estudiadas.

Además, se ha conseguido discriminar las masas y velocidades que generan una única energía, lo cual en un descubrimiento novedoso, ya que posiciona este trabajo en el estado del arte de la caracterización de impactos en SHM puesto que no se había conseguido previamente en dimensiones tan grandes y con un rango de masas, velocidades y energías tan amplio.

Todos estos resultados han sido obtenidos con una arquitectura única y relativamente sencilla, lo cual conlleva a que los modelos son susceptibles a una amplia mejora de su precisión mediante la construcción de modelos más complejos y con arquitecturas modernas.

Así mismo, se ha comprobado que no es necesario desarrollar modelos excesivamente complejos para obtener buenos resultados.

Para concluir, este trabajo no hubiera sido posible sin el ICNC. Para obtener una gran cantidad de datos es imprescindible automatizar el proceso y eliminar la necesidad de un operario que realice de forma manual los impactos.

Se puede afirmar que el ICNC ha funcionado correctamente, al igual que el *software* desarrollado para su control, adquisición y procesado de impactos.

Referencias

- [1] Aluminum Extrusion and Brackets | MISUMI online shop - Select, configure, order.
URL <https://uk.misumi-ec.com/vona2/mech/M1500000000/M1501000000/>
- [2] Placa de evaluación Seeed Studio Grove-Electromagnet - 101020073 | RS Components.
- [3] Nema 17 Bipolar 1.8deg 45Ncm (63.7oz.in) 1.68A 2.8V 42x42x48mm 4 cables.
- [4] 20014B148 | Tubo PVC Georg Fischer, 2m, PVC-U, diámetro externo: 40mm, Grosor: 3.0mm | RS Components.
URL <https://es.rs-online.com/web/p/tubos-pvc/2874493>
- [5] Arduino - ArduinoBoardMega2560.
URL <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>
- [6] RAMPS 1.4 - RepRap.
URL https://reprap.org/wiki/RAMPS_1.4
- [7] fra589/grbl-Mega-5X: 5/6 Axis version of Grbl, the open source, embedded, high performance g-code-parser and CNC milling controller written in optimized C that will run on an Arduino Mega2560.
URL <https://github.com/fra589/grbl-Mega-5X>
- [8] Pololu - DRV8825 Stepper Motor Driver Carrier, High Current.
URL <https://www.pololu.com/product/2133>
- [9] Y. Bengio, A. Courville, P. Vincent, Representation Learning: A Review and New Perspectives
doi:10.1109/TPAMI.2013.50.
URL <http://ml.nec-labs.com/senna/>.
- [10] A. M. Damm, C. Spitzmüller, A. T. S. Raichle, A. Bühler, P. Weißgraeber, P. Middendorf, Deep learning for impact detection in composite plates with sparsely integrated sensors, *Smart Materials and Structures* 29 (12) (2020) 125014. *doi:10.1088/1361-665x/abb644*.
URL <https://doi.org/10.1088/1361-665x/abb644>

- [11] C. Beleites, U. Neugebauer, T. Bocklitz, C. Krafft, J. Popp, Sample size planning for classification models, *Analytica Chimica Acta* 760 (2013) 25–33. [arXiv:1211.1323](https://arxiv.org/abs/1211.1323), doi:10.1016/j.aca.2012.11.007.
- [12] J. Brownlee, How Much Training Data is Required for Machine Learning? (2017).
URL <https://machinelearningmastery.com/much-training-data-required-machine-learning/>
- [13] J. Brownlee, Better Deep Learning. Train Faster, Reduce Overfitting, and Make Better Predictions, Vol. 1, 2018.
- [14] (7) A TRANSFORMER-BASED FRAMEWORK FOR MULTIVARIATE TIME SERIES REPRESENTATION LEARNING - YouTube.
URL https://www.youtube.com/watch?v=aXKv7uFNZLY&ab_channel=Fellowship
- [15] A. M. Tripathi, R. Dutta Baruah, Multivariate Time Series Classification With An Attention-Based Multivariate Convolutional Neural Network.
- [16] Attention for time series forecasting and classification | by Isaac Godfried | Towards Data Science.
- [17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention Is All You Need.
- [18] Attention For Time Series Forecasting And Classification.
URL <https://www.topbots.com/attention-for-time-series-forecasting-and-classification/>
- [19] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, C. Eickhoff, A Transformer-based Framework for Multivariate Time Series Representation Learning [arXiv:2010.02803v2](https://arxiv.org/abs/2010.02803v2).
URL <http://arxiv.org/abs/2010.02803>
- [20] A TRANSFORMER-BASED FRAMEWORK FOR MULTI-VARIATE TIME SERIES REPRESENTATION LEARNING.
- [21] L. Van Der Maaten, G. Hinton, Visualizing Data using t-SNE, *Journal of Machine Learning Research* 9 (2008) 2579–2605.
- [22] H. Song, D. Rajan, J. J. Thiagarajan, A. Spanias, Attend and Diagnose: Clinical Time Series Analysis using Attention Models [arXiv:1711.03905v2](https://arxiv.org/abs/1711.03905v2).

URL www.aaai.org

- [23] J. Arm, Z. Bradac, S. Misik, J. Streit, CNC Motion Controller Testing Methods, Vol. 51, Elsevier B.V., 2018, pp. 244–249. doi:10.1016/j.ifacol.2018.07.161.

- [24] Y. Hao, H. Cao, A New Attention Mechanism to Classify Multivariate Time Series.

URL <http://www.auslan.org.au/dictionary/words/arithmetic-1.html>

- [25] RS-15-5 | Fuente de alimentación de modo conmutado (SMPS) integrada Mean Well, 5V dc, 3A, 15W, 1 salida, Montaje en panel | RS Components.

URL <https://es.rs-online.com/web/p/fuentes-de-alimentacion-de-modo-conmutado-smps-integradas/0413673>

- [26] Polea GT2 5mm Mecánica Productos-Createc 3D Shop.

URL <https://createc3d.com/es/correas-y-poleas/80-comprar-polea-gt2-5mm-prusa-precio-oferta.html>