



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA NAVAL Y OCEÁNICA

MÁSTER EN INGENIERÍA NAVAL Y OCEÁNICA

TRABAJO FIN DE MÁSTER

# **APLICACIÓN DE HERRAMIENTAS DE MACHINE LEARNING AL COMPORTAMIENTO EN LA MAR**

Autor: Pablo Romero Tello

Directores:

José Enrique Gutiérrez Romero

Borja Serván Camas

CARTAGENA

Octubre 2019



# Agradecimientos

En primer lugar, querría hacer una especial mención al Dr. José Enrique Gutiérrez Romero, director de este proyecto, ya que gracias a él he podido disfrutar de la beca de investigación que me ha permitido realizar este proyecto, además de por todo el tiempo, conocimientos y dedicación brindados. Gracias también al codirector de este proyecto el Dr. Borja Serván Camas, por sus consejos e ideas.

Dar las gracias al aula CIMNE por el soporte prestado, puesto que sin esta ayuda no se podría haber hecho.

Este proyecto también es el fruto de todos aquellos que han participado en todas mis etapas académicas, (tanto aquellos que fueron muy buenos como los que no lo fueron tanto, ya que siempre se puede aprender algo de quien tienes enfrente). Más en especial es fruto de todos los profesores de la Escuela de Navales de la UPCT.

Agradecerles a mis padres y mi hermano el apoyo recibido durante la realización de este proyecto, así como todos los sacrificios que han realizado para que yo haya podido disfrutar de esta fase de mi vida.

Muchas gracias a mi consejera, por toda la ayuda que me has dado, pos apuntes, las llamadas desesperadas cuando Matlab se negaba a funcionar, por las largas conversaciones de Machine Learning, y un largo etcétera, muchísimas gracias por todo, ya que sin tu ayuda y paciencia este proyecto no habría llegado a buen puerto.

Y como no, también gracias a todos mis amigos, tanto los de la UPCT cómo los de toda la vida, que han sabido hacerme desconectar en esas ocasiones en las que era imprescindible para poder continuar. Al resto de personas que no he mencionado (porque la lista sería infinita) pero que sé que siempre han estado ahí, gracias.



# Índice general

<b>Resumen</b>	<b>11</b>
<b>Abstract</b>	<b>13</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Machine Learning</b>	<b>3</b>
2.1. Neurona Artificial Básica . . . . .	6
2.2. Arquitectura de una red neuronal . . . . .	7
2.3. Funciones de activación . . . . .	9
2.4. Algoritmos de aprendizaje . . . . .	11
2.5. Conjuntos de datos . . . . .	15
<b>3. Estado del arte</b>	<b>17</b>
<b>4. Comportamiento en la mar</b>	<b>21</b>
4.1. Formulación Básica . . . . .	22
4.2. Metodologías . . . . .	28
<b>5. Motivaciones y objetivos</b>	<b>33</b>
<b>6. Metodología</b>	<b>35</b>
6.1. Generación de casos de entrenamiento . . . . .	37
6.2. Obtención de datos de entrenamiento . . . . .	39
6.3. Caracterización de los casos de entrenamiento . . . . .	56
6.4. Análisis de sensibilidad de la RNA . . . . .	58
<b>7. Resultados</b>	<b>61</b>
7.1. Discusión de resultados . . . . .	71
<b>8. Conclusiones</b>	<b>73</b>
<b>9. Futuros desarrollos</b>	<b>75</b>
<b>10. Bibliografía</b>	<b>77</b>



# Índice de figuras

Figura 2.1. Estructura básica del perceptron. . . . .	6
Figura 2.2. Estructura básica de Adaline. . . . .	7
Figura 2.3. Estructura básica del Perceptrón actual. . . . .	7
Figura 2.4. Esquema de una red con una sola capa. . . . .	8
Figura 2.5. Esquema de una red multicapa. . . . .	8
Figura 2.6. Esquema de una arquitectura <i>feedforwardnet</i> con 10 entradas, una capa oculta de 10 neuronas y 60 salidas. Fuente <i>Matlab</i> . . . . .	9
Figura 2.7. Esquema de una red recurrente. . . . .	9
Figura 2.8. Esquema de <i>cascadeforwardnet</i> . Fuente <i>Matlab</i> . . . . .	9
Figura 2.9. Función de activación Umbral. . . . .	10
Figura 2.10. Función de activación Rampa. . . . .	10
Figura 2.11. Función de activación Sigmoidea. . . . .	10
Figura 2.12. Función de activación tangente hiperbólica. . . . .	11
Figura 4.1. Grados de libertad del buque. . . . .	22
Figura 4.2. Esquema de ecuaciones y condiciones de contorno del problema lineal. . . . .	23
Figura 4.3. Esquema de descomposición de las fuerzas que actúan sobre un cuerpo en flotación. . . . .	24
Figura 4.4. Modelo de buque en ensayo de canal. Fuente: <i>Canal de Experiencias Hidrodinámicas del Pardo, CEHIPAR</i> . . . . .	31
Figura 6.1. Flujograma seguido para el entrenamiento de la RNA para el estudio del comportamiento en la mar. . . . .	36
Figura 6.2. Áreas seccionales unitarias de los distintos buques seleccionados para el entrenamiento de la RNA. . . . .	38
Figura 6.3. Flujograma seguido para la generación de casos de entrenamiento y simulación de estos. . . . .	39
Figura 6.4. Imágenes de distintas escalas del buque tanque usadas para el entrenamiento. . . . .	49
Figura 6.5. Imágenes de distintas escalas del buque portacontenedores usadas para el entrenamiento. . . . .	49
Figura 6.6. Imágenes de distintas escalas del buque crucero usadas para el entrenamiento. . . . .	50
Figura 6.7. Imágenes de distintas escalas del buque perforador usadas para el entrenamiento. . . . .	50
Figura 6.8. Imágenes de distintas escalas del buque FPSO usadas para el entrenamiento. . . . .	51
Figura 6.9. Imágenes de distintas escalas del buque fragata usadas para el entrenamiento. . . . .	51
Figura 6.10. Imágenes de distintas escalas del buque remolcador usadas para el entrenamiento. . . . .	52
Figura 6.11. Imágenes de distintas escalas del buque semisumergible usadas para el entrenamiento. . . . .	52

Figura 6.12. Imágenes de distintas escalas del buque patrullero usadas para el entrenamiento. . . . .	53
Figura 6.13. Imágenes de distintas escalas del buque RORO usadas para el entrenamiento. . . . .	53
Figura 6.14. Imágenes de distintas escalas del buque ship pro usadas para el entrenamiento. . . . .	54
Figura 6.15. Imágenes de distintas escalas del buque suministro usadas para el entrenamiento. . . . .	54
Figura 6.16. Imágenes de distintas escalas del buque X-bow usadas para el entrenamiento. . . . .	55
Figura 6.17. Imágenes con distintos volúmenes seccionales para distintos tipos de buques. . . . .	57
Figura 7.1. Comparativa de las curvas RAO para el buque perforador para las dos traslaciones. . . . .	62
Figura 7.2. Comparativa de las curvas RAO para el buque portacontenedores para las dos traslaciones. . . . .	62
Figura 7.3. Comparativa de las curvas RAO para el buque remolcador para las dos traslaciones. . . . .	63
Figura 7.4. Comparativa de las curvas de masas añadidas de traslaciones para el buque perforador para las tres traslaciones. . . . .	64
Figura 7.5. Comparativa de las curvas de masas añadidas de traslaciones para el buque portacontenedores para las tres traslaciones. . . . .	64
Figura 7.6. Comparativa de las curvas de masas añadidas de traslaciones para el buque remolcador para las tres traslaciones. . . . .	65
Figura 7.7. Comparativa de las curvas de masas añadidas de giros para el buque perforador para los dos giros. . . . .	65
Figura 7.8. Comparativa de las curvas de masas añadidas de giros para el buque portacontenedores para los dos giros. . . . .	66
Figura 7.9. Comparativa de las curvas de masas añadidas de giros para el buque remolcador para los dos giros. . . . .	66
Figura 7.10. Casco del buque tipo Nordstrom. . . . .	67
Figura 7.11. Casco del buque tipo Serie 64. . . . .	67
Figura 7.12. Casco del buque tipo SSPA. . . . .	67
Figura 7.13. Comparativa de las curvas de masas añadidas de traslaciones para el buque tipo Nordstrom para las dos traslaciones. . . . .	68
Figura 7.14. Comparativa de las curvas de masas añadidas de traslaciones para el buque tipo Serie 64 para las dos traslaciones. . . . .	68
Figura 7.15. Comparativa de las curvas de masas añadidas de traslaciones para el buque tipo SSPA para las dos traslaciones. . . . .	69
Figura 7.16. Comparativa de las curvas de masas añadidas de giros para el buque tipo Nordstrom para los dos giros. . . . .	69
Figura 7.17. Comparativa de las curvas de masas añadidas de giros para el buque tipo Serie 64 para los dos giros. . . . .	70
Figura 7.18. Comparativa de las curvas de masas añadidas de giros para el buque tipo SSPA para los dos giros. . . . .	70
Figura 8.1. Comparativa del tiempo de cálculo para el análisis del comportamiento en la mar de un buque entre Redes Neuronales Artificiales (RNA), programas CFD basados en teoría potencial (SP) y programas CFD basados en teoría viscosa (SV), para las distintas etapas del proceso de estimación: pre-procesado (pre), cálculo (calc) y post-procesado (post). . . . .	74



# Índice de tablas

Tabla 6.1. Características de los buques base empleados en el entrenamiento de la RNA. . . . .	37
Tabla 6.2. Valores tomados para las constantes empleadas en el código <i>GENERACIÓN DEL ARCHIVO DE APOYO</i> . . . . .	42
Tabla 6.3. Condiciones consideraras para la definición de las formas del casco. . . . .	58
Tabla 6.4. Análisis de sensibilidad de las distintas arquitecturas y optimizadores de la RNA. . . . .	58
Tabla 6.5. Análisis de sensibilidad de las arquitecturas <i>cascaforwardnet</i> y <i>feedforwardnet</i> . . . . .	59
Tabla 6.6. Análisis de sensibilidad de los distintos métodos de definición del buque. . . . .	59
Tabla 6.7. Análisis de sensibilidad de los distintos métodos de definición del buque. . . . .	60
Tabla 6.8. Análisis de sensibilidad del número de neuronas y del número de capas ocultas. . . . .	60



# Resumen

La Inteligencia Artificial (IA) es un campo que ha venido desarrollándose en las últimas décadas, con énfasis en sistemas inteligentes para reconocimiento de escritura, sistemas de voz, sistemas autónomos, etcétera. En el ámbito de la Arquitectura Naval, la aplicación de la IA puede tener una relevancia notable, sobre todo como herramientas predictivas para evaluar el grado de cumplimiento de las expectativas de diseño.

Este trabajo se centra en la aplicación de herramientas de Aprendizaje Máquina (*Machine Learning*) a un aspecto de la Ingeniería Naval, el comportamiento de buques en alta mar. La complejidad de los cálculos de respuesta del buque frente a olas hace habitual el uso de simplificaciones, por ejemplo, que la respuesta de la plataforma es de carácter lineal. El tipo de metodologías usualmente empleadas para la resolución de este tipo de problemas suelen ser los Métodos de Elementos Finitos o los Métodos de los Elementos de Contorno.

Se propone el desarrollo de un algoritmo de aprendizaje basado en redes neuronales artificiales (*Artificial Neural Network*, ANN) que permita una evaluación de los parámetros del comportamiento en la mar en búsqueda de un ahorro en el tiempo de cálculo con precisiones similares a las ofrecidas por metodologías alternativas. Así, se llevará a cabo un estudio del estado del arte la IA en el ámbito de la Ingeniería Naval. Se propondrá una metodología de trabajo y entrenamiento de la red. Se llevarán a cabo comparativas para distintos tipos de buques. Finalmente, se expondrán unas conclusiones con los resultados relevantes obtenidos.



# Abstract

The Artificial Intelligence (AI) has been developing in the last decades, focused on smart writing, voice recognition systems, autonomous systems, and others. The application of AI in Naval Architecture field might become a relevant tool, especially those applied to assess the expected degree in compliance of a design.

This work focuses on the application of Machine Learning to a specific field of Naval Architecture, the seakeeping. The complexity of ship-wave interaction makes common using mathematics simplifications, such as the platforms response is linear. Finite Element Methods or Boundary Element Methods are common methodologies used to solve these type of problems.

Learning methodology based on an Artificial Neural Network (ANN) to evaluate the seakeeping performance parameters is proposed. This methodology offers a time saving alternative when compared with other solutions. The state of art of IA applied to Naval Architecture and Marine Engineering is introduced first. Then, the methodology and network training is explained. Later, comparisons with different types of ships are performed. Finally, relevant conclusions about the results obtained are shown.

**Keywords:** Artificial Neural Network · Artificial Intelligence · Naval Architecture · Seakeeping



# 1

## Introducción

---

El primer Capítulo presenta las ventajas de la utilización de las RNA en Ingeniería Naval. Finalmente, hace un esbozo de los motivos por las que se ha decidido llevar a cabo este proyecto, además de los objetivos fundamentales del mismo.

---

Desde el punto de vista de la ingeniería, la utilidad de las herramientas de Aprendizaje Máquina (en terminología inglesa, *Machine Learning*), es innegable, sobre todo en análisis de patrones. Desde un punto de vista práctico, las empresas buscan sistemas fiables que permitan evaluar de forma rápida distintos aspectos de un sistema. Más concretamente, dentro de la Ingeniería Naval, el uso de las herramientas de *Machine Learning* puede suponer una ventaja muy significativa, ya que estas podrían facilitar gran parte de los cálculos que resultan esenciales para evaluar el grado de cumplimiento de las expectativas de diseño, estos cálculos envuelven distintas disciplinas que van desde la mecánica de fluidos hasta el cálculo estructural, que requieren de costosísimas y largas simulaciones computacionales. Además, los fenómenos que envuelven el comportamiento dinámico son en su mayoría de carácter altamente no lineal.

En el caso del comportamiento en la mar, está claro que consiste en un fenómeno no lineal, pero dada la complejidad de los cálculos, es habitual y necesario realizar simplificaciones, asumiendo que la respuesta del buque es de carácter lineal. Este tipo de hipótesis permite obtener funciones como las curvas de respuesta en amplitud (en terminología inglesa *Response Amplitude Operator*, RAO) que son muy útiles a la hora de identificar los modos propios de comportamiento de estructuras flotantes y evaluar en una primera instancia su comportamiento frente a un determinado estado de la mar. Este trabajo de investigación, propone una metodología alternativa a los métodos de cálculo utilizados hasta ahora en el comportamiento en la mar, basada en el desarrollo de un algoritmo de aprendizaje de redes neuronales artificiales (RNA) que permita la obtención de la curva de respuesta en amplitud, de una manera más rápida y sencilla, con precisiones similares a los ofrecidos por metodologías alternativas.

Fundamentalmente, se buscará el planteamiento de algoritmos RNA, con distintas arquitecturas, algoritmos de entrenamiento, funciones de transferencia para la evaluación de las curvas RAO en cualquier buque. El grado de precisión no solo depende de la conveniencia del algoritmo utilizado para captar el patrón de comportamiento, sino que

depende muy significativamente del nivel de entrenamiento al que se le someta; esto es, a mayor entrenamiento mejores predicciones se puedan alcanzar. Para llevar a cabo las labores de entrenamiento del algoritmo, se hará uso de programas de cálculo de comportamiento en la mar, que permitirán establecer las funciones objetivo del aprendizaje del algoritmo RNA.

El resultado del trabajo será un algoritmo RNA que permita la evaluación previa del comportamiento en la mar de un buque con tiempos de cálculo muy reducidos, tal y como apuntan diversos autores en la bibliografía relacionada



# 2

## Machine Learning

---

En el siguiente Capítulo se muestra una breve descripción de las distintas herramientas de la Inteligencia Artificial, centrandose en especial en las Redes Neuronales Artificiales, tratando de manera más pormenorizada aspectos como la neurona artificial, las funciones de activación y los algoritmos de entrenamiento.

---

La Inteligencia Artificial (IA) puede definirse como la combinación de un conjunto de algoritmos planteados con el propósito de crear máquinas que presenten las mismas capacidades que el ser humano. Dentro de la inteligencia artificial, se pueden destacar los sistemas de aprendizaje automático o más conocidos como *machine learning*, son una aplicación de la inteligencia artificial que proporciona a los sistemas la capacidad de aprender y mejorar automáticamente a partir de la experiencia sin estar programados explícitamente.

El proceso de aprendizaje, comienza con la obtención de datos, ejemplos, representativos del problema a solventar, con el fin de buscar patrones en los datos y tomar mejores decisiones en el futuro en base a dichos ejemplos de aprendizaje. El objetivo principal es permitir que los ordenadores aprendan automáticamente sin intervención o asistencia humana. Dentro de la rama de *machine learning*, cabe destacar las siguientes técnicas [1, 2]:

- **Árboles de decisión**, consisten en una técnica de aprendizaje automático que permiten identificar conceptos a partir de un conjunto de entrenamiento. Se construyen mediante la formulación de preguntas al conjunto de entrenamiento sobre características determinadas, y se clasifican en distintos conjuntos dependiendo de la respuesta. Su principal aplicación es la clasificación y la extracción de reglas de conocimiento.
- **Algoritmo de agrupamiento** o en terminología inglesa *clustering*, es un procedimiento de agrupación de una serie de vectores de acuerdo con un criterio. El conocimiento de los grupos puede permitir una descripción sintética de un conjunto de datos complejo, de ahí su uso en minería de datos. Esta descripción más sencilla se consigue con la sustitución de la descripción de todos los elementos de un conjunto de datos por la de un valor característico del mismo.

- **Sistemas expertos**, son programas informáticos que emulan el comportamiento de un experto humano en la solución de un problema. Los sistemas expertos funcionan de manera que almacenan conocimientos concretos para un campo determinado y solucionan los problemas, utilizando esos conocimientos, mediante deducción lógica de conclusiones. Con ellos se busca una mejora en calidad y rapidez de respuestas dando así lugar a una mejora de la productividad del experto.
- **Máquinas de soporte vectorial**, en terminología inglesa, *Support Vector Machines*, **SVM**, son algoritmos de clasificación binario que dado un conjunto de datos de dos tipos los algoritmos SVM genera un hiperplano que permite separar esos datos en dos grupos.
- **Redes neuronales artificiales**, RNA, en terminología inglesa, *Artificial Neural Network*, **ANN**, son modelos matemáticos inspirados en el comportamiento biológico de las neuronas y en cómo se organizan formando la estructura del cerebro. Estos algoritmos tienen la capacidad de predecir y adivinar patrones en base a datos o experiencias previas. Una RNA aprende mediante la modificación de una serie de valores o pesos asignados, así como la topología de cada función que compone el conjunto de algoritmos.
- **Algoritmos genéticos**, AG, es una técnica de resolución de problemas que imita a la evolución biológica como estrategia para la resolución de estos. Dado un problema específico a resolver, la entrada del AG es un conjunto de soluciones potenciales a ese problema, codificadas de alguna manera, y un sesgo llamado función de aptitud, que permite evaluar cuantitativamente a cada solución candidata. Estas candidatas pueden ser soluciones que ya se sabe que funcionan, con el objetivo de que el AG las mejore, pero se suelen generar aleatoriamente. A partir de ahí, AG evalúa cada candidata de acuerdo con la función de aptitud.

Dentro de técnicas de *machine learning* presentadas anteriormente, en este proyecto se va a llevar a cabo utilizando redes neuronales artificiales, puesto que en la actualidad es una de las técnicas más potentes utilizadas para predicción. Además, y puesto que el problema a resolver es un problema de regresión, y, por tanto, la utilización de las distintas técnicas presentadas anteriormente queda limitado. Algunas ventajas de las redes neuronales son:

- **Aprendizaje adaptativo**. Esta es una de las características más notoria de las redes neuronales, esta hace referencia a la capacidad de la red de encontrar y aprender a diferenciar patrones mediante entrenamiento con ejemplos ilustrativos, esto se consigue debido al autoajuste de las neuronas que componen el sistema. Al aplicar a la resolución de un problema redes neuronales, no es necesario la generación de un algoritmo de resolución, sino que ella genera su propia distribución de pesos en la fase de entrenamiento.
- **Sencillez**. Si se compara este sistema de resolución frente a la resolución numérica, destaca la sencillez, ya que una vez definida y entrenada la red su aplicación es muy sencilla y rápida.
- **Tolerancia a fallos**. Se encuentran dos aspectos diferenciados en cuanto a la tolerancia a fallos, por un lado, las redes pueden aprender a identificar patrones mediante ejemplos que tengan ruido, estén distorsionados o incompletos. Por otro lado, la red puede seguir realizando su función, aunque se destruya parte de la red, amentando el error. Esta última es posible ya que su información está distribuida en las conexiones entre neuronas, de tal modo que existe cierto grado de redundancia.

- **Capacidad de generalización.** Una red neuronal convenientemente entrenada, frente a un modelo que nuevo, es capaz de generar resultados coherentes con la naturaleza del problema para el cual han sido entrenadas.

La computación neuronal provee un acercamiento al reconocimiento y percepción humana que no consiguen los métodos tradicionales de cálculo. Las RNA presentan resultados razonables, y a continuación se enumeran algunas de las aplicaciones donde se están utilizando:

- ∨ Análisis y procesado de señales.
- ∨ Control de procesos.
- ∨ Robótica.
- ∨ Diagnósticos médicos.
- ∨ Reconocimiento de imágenes.
- ∨ Procesado del lenguaje.
- ∨ Predicción.
- ∨ Modelos económicos y financiero.

A continuación, se muestran de manera resumida los hitos de las RNA a través de la historia. El primer modelo de red neuronal fue propuesto por McCulloch y Pitts (1943), era un modelo de neuronas binarias, donde cada neurona podía adaptar dos estados. Las RNA se crearon para simular los sistemas nerviosos biológicos, constituidos por neuronas y nodos que interconectan las neuronas [3, 4].

En 1957, Rosenblatt publicó el mayor trabajo de investigación en computación neuronal realizado hasta esa fecha, en este desarrollo el elemento llamado *Perceptron*, como un sistema clasificador de patrones que podía identificar patrones geométricos y abstractos. El *Perceptron* fue originalmente diseñado para el reconocimiento óptico de patrones. La mayor limitación del *Perceptron* era su incapacidad para resolver la función lógica OR exclusivo.

En 1959, Widrow desarrollo un elemento adaptativo lineal llamado *Adaline* [5], y una versión de dos capas, *Madaline*, estos elementos se usaron en diversas aplicaciones como reconocimiento de voz y de caracteres, predicción del tiempo, etc.

A mediados de los años 60 se paralizó casi por completo la investigación en RNA tras la publicación del libro *Perceptrons* [6], publicado por Minsky y Papert pertenecientes al MIT. La conclusión de este trabajo es que el *Perceptron* y la computación neuronal no eran temas interesantes que estudiar y desarrollar. Sin embargo, en 1982, Hopfield con la publicación del artículo *Hopfield Model*, junto con la invención del algoritmo *Backpropagation* se consiguió devolver el interés y la confianza en la computación neuronal tras dos décadas de casi absoluta inactividad.

Hoy en día, muchos grupos investigadores de distintas universidades de todo el mundo que están trabajando en el área de las RNA. Cabe destacar a Rumelhart de la universidad de Stanford, que es uno de los principales impulsores de la red más utilizada en la mayoría de las aplicaciones actuales, la red *Backpropagation* [7].

El presente proyecto no sería completamente apropiado calificarlo como un proyecto de *Big Data* ya que, a pesar de hacer uso de un cierto volumen de datos, este no es tan

elevado como para que se deban emplear técnicas y herramientas de análisis diferentes a las convencionales. Sería más adecuado encajarlo dentro del ámbito del *Business Intelligence* o Inteligencia Empresarial, en el cual se persigue el objetivo de extraer conocimiento de los datos que se poseen para dar valor a un producto, siendo el volumen de estos de menor envergadura y variedad que los de proyectos *Big Data*.

Además, los motivos expuestos anteriormente sirven para justificar el uso del *software Matlab* con su paquete *Deep Learning Toolbox* para el análisis de los datos, puesto que, al poseer una cantidad de datos asumible, se hace innecesario emplear lenguajes como *PySpark* o *Scala*, destinados al tratamiento de cantidades ingentes de datos en entornos de desarrollo como *clusters* de servidores ejecutándose bajo el ecosistema de *Hadoop*, que es un *framework* de *software* que soporta aplicaciones con licencia libre, y permite a estas aplicaciones trabajar con miles de nodos y *petabytes* de datos.

## 2.1. Neurona Artificial Básica

Una neurona artificial es la unidad de procesamiento básica de una red neuronal artificial. En esencia, se aplica un conjunto de entradas a la neurona. Cada una de estas entradas ( $x_i$ ) se multiplica por su peso sináptico ( $w_i$ ), que no es más que la ponderación correspondiente al grado de conexión de la sinapsis. Una vez ponderadas todas las entradas se suman y se determina el nivel de excitación o activación de la neurona. El resultado del sumatorio de las entradas ponderadas suele ser procesada por una función de activación ( $f()$ ) para producir la salida. Como se verá más adelante esta función de activación puede tomar distintas formas. El primer modelo de neurona artificial fue el perceptrón, que en su primera versión únicamente permitía una salida binaria, es decir únicamente clasificaba si los datos de entrada pertenecen a una clase u a otra [8, 9].

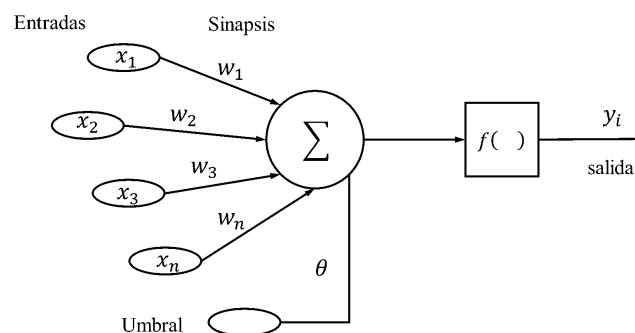


Figura 2.1. Estructura básica del perceptrón.

Otra neurona artificial es la *Adaline*, es una neurona muy similar al perceptrón, la característica más notoria es que su manera de aprender está basada en la regla Delta, en esta regla se evalúa el error a la salida de la neurona con el error cuadrático medio, tratándose en esta regla de minimizar ese error global del conjunto de ejemplos mediante un proceso iterativo en el que se modifican los pesos sinápticos de la neurona mediante el método del descenso del gradiente.

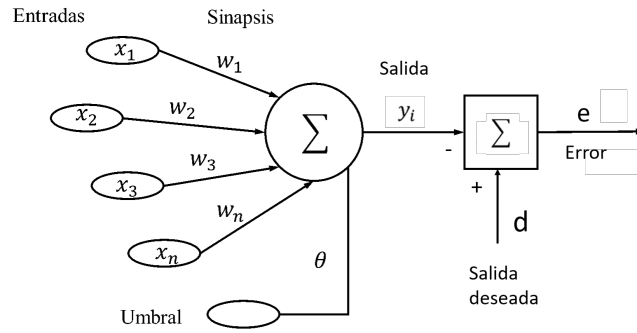


Figura 2.2. Estructura básica de Adaline.

Comparando el primer modelo de Perceptrón con la *Adaline*, se puede concluir que la salida de la *Adaline* es real mientras que la del perceptrón es binaria, en la *Adaline* siempre se modifican los pesos sinápticos mediante la diferencia real entre la salida real y la esperada mientras que en el perceptrón únicamente se modifican si la salida real y la esperada no son de la misma clase, además, cabe destacar que en la *Adaline* se conoce el error cometido por la neurona mientras que en el perceptrón únicamente se sabe si ha acertado o no.

Se puede observar que si al *Adaline* se le incluye la función de activación se convierte en un perceptrón, pudiendo así conocer el error cometido por el perceptrón e introducir como método de aprendizaje del perceptrón la regla Delta.

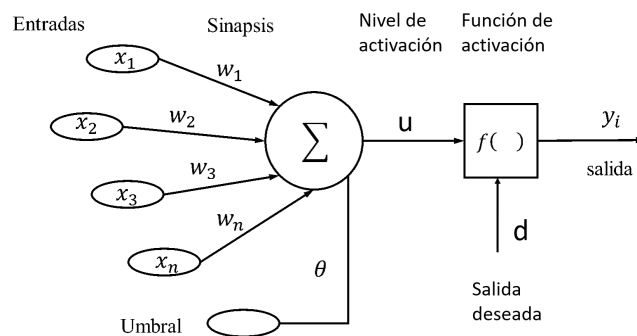


Figura 2.3. Estructura básica del Perceptrón actual.

A día de hoy, el perceptrón es una de las neuronas más utilizadas, ya que al introducir distintas funciones de activación ya no tiene una salida binaria, y al combinar varios perceptrones se consigue una salida muy aceptable

## 2.2. Arquitectura de una red neuronal

La potencia de la computación neuronal proviene de las conexiones de las neuronas constituyendo así una RNA. La configuración más simple es un único grupo de neuronas ordenadas en una sola capa. Por tanto, en las redes de una capa, se tiene la capa de entrada y la de salida [8].

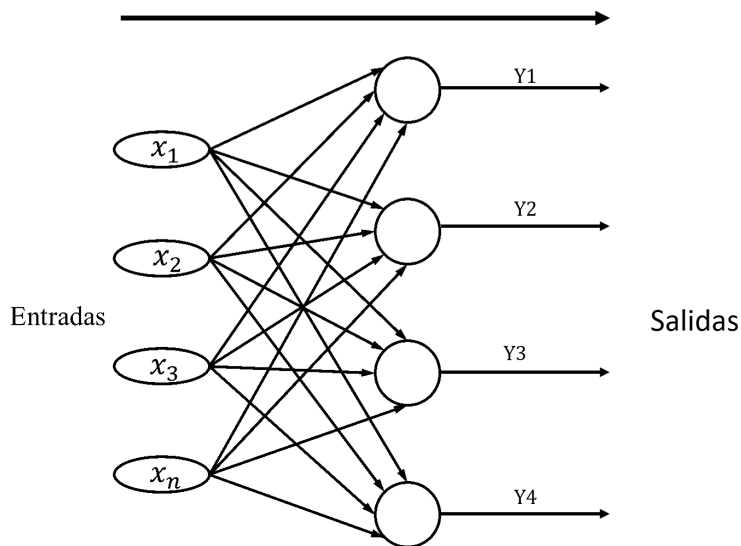


Figura 2.4. Esquema de una red con una sola capa.

Las redes con mayores prestaciones son aquellas más complejas, es decir, con mayor número de capas, conocidas como redes multicapa. Estas redes tienen la capa de entrada, la capa de salida y capas ocultas interconectadas.

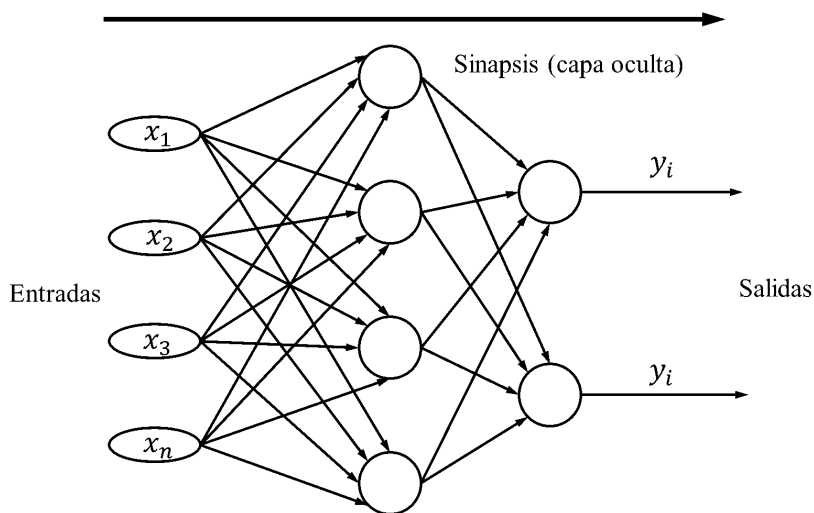
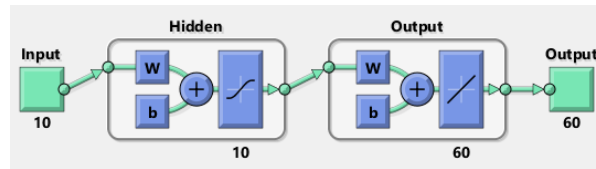


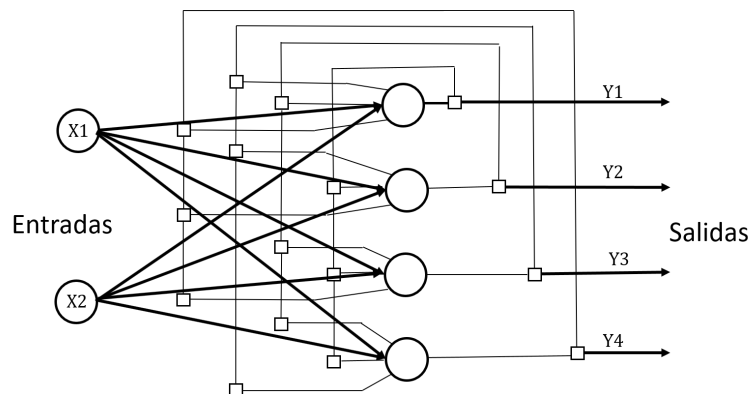
Figura 2.5. Esquema de una red multicapa.

Como ya se ha indicado anteriormente en la realización de este proyecto se utilizará el software *Matlab*, dentro de este software, la arquitectura multicapa prealimentada, la información va hacia adelante, se encuentran la arquitectura *fitnet* y la arquitectura *feedforwardnet*, (véase Figura 2.6), siendo ambas básicamente un perceptrón multicapa, en el caso de *fitnet* está diseñada únicamente para la predicción, y en el caso de *feedforwardnet* se utiliza tanto para predicción (regresión) como para clasificación [10].



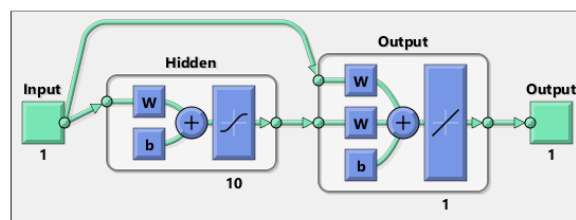
**Figura 2.6.** Esquema de una arquitectura *feedforwardnet* con 10 entradas, una capa oculta de 10 neuronas y 60 salidas. Fuente *Matlab*.

Dentro de todas las configuraciones de redes que se pueden encontrar hay que destacar aquellas arquitecturas en las que la información puede ir desde la capa de entrada a las salidas directamente o en la que la información sea recurrente, esto quiere decir que la salida de una neurona es la entrada de otra neurona en la misma capa, tal y como se ve en la Figura 2.7.



**Figura 2.7.** Esquema de una red recurrente.

En *Matlab* aparece la arquitectura *cascaforwardnet* (véase Figura 2.8) que básicamente es idéntica a la *feedforwardnet*, solo que incluyen una conexión entre las neuronas de entrada y cada una de las capas siguientes.



**Figura 2.8.** Esquema de *cascaforwardnet*. Fuente *Matlab*.

La labor del diseñador de la red consiste en encontrar la arquitectura que mejor se adapte al problema a resolver, dentro de esto aparece, encontrar el tipo de red, si es un perceptrón multicapa, o una red recurrente, así como definir el número de capas y de neuronas por capa que mejor se adapte, y la función de activación a utilizar en cada capa, estas labores descritas anteriormente se desarrollarán en capítulos posteriores del trabajo.

## 2.3. Funciones de activación

Como se ha indicado anteriormente, la salida de la neurona es modificada por la función de activación, los tipos de función de activación más utilizados se muestran a continuación [11, 12]:

- **Umbral**, esta función tiene una salida binaria, siendo 1 (se activa) siempre que el sumatorio de las entradas ponderadas sea mayor que 0 y 0 (no se activa) siempre que el sumatorio sea menor o igual a 0.

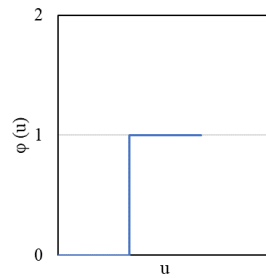


Figura 2.9. Función de activación Umbral.

- **Rampa**, es una función lineal limitada en un rango, siendo 0 cuando está por debajo de ese rango y 1 cuando se encuentra por encima. Y dentro del rango da una respuesta lineal entre 0 y 1.

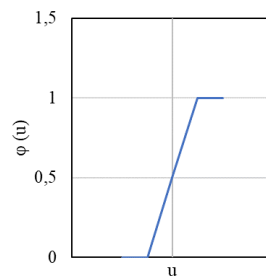


Figura 2.10. Función de activación Rampa.

- **Sigmoidal**, es una de las funciones de activación más usada, da una salida no lineal y se rige por la ecuación 2.1.

$$\varphi(u) = \frac{1}{1 + e^{-au}}, \quad (2.1)$$

donde a es la pendiente de la sigmoide.

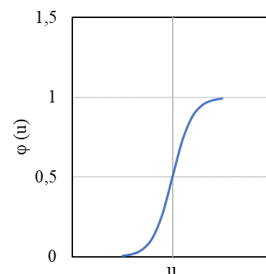


Figura 2.11. Función de activación Sigmoidal.

- **Tangente hiperbólica**, junto con la función sigmoide es una de las más utilizadas, da una salida no lineal entre -1 y 1, se rige por la ecuación 2.2.



$$f(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}} \quad (2.2)$$

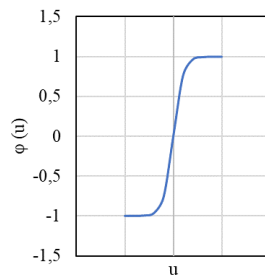


Figura 2.12. Función de activación tangente hiperbólica.

De todas las funciones de activación expuestas, la función sigmoïdal, es la que tiene el uso más extendido y es por tanto la que se ha usado en este trabajo de investigación.

## 2.4. Algoritmos de aprendizaje

Dentro de los paradigmas de aprendizaje se puede diferenciar entre el aprendizaje supervisado y no supervisado, en el supervisado un agente externo controla el aprendizaje y permite la modificación de los pesos en caso de error en la salida, este aprendizaje puede ser por corrección de error, por refuerzo o estocástico. En el no supervisado, la corrección de pesos no depende de un agente externo, puede ser hebbiano o competitivo. En este proyecto se utilizará un aprendizaje supervisado, ya que se conoce la salida deseada, y en la fase de entrenamiento esta se le suministra a la red para que modifique los pesos, de tal modo que se consiga un error mínimo [2, 13, 14].

Para llevar a cabo el proceso de aprendizaje supervisado es necesario emplear funciones de optimización. Entre ellas se pueden encontrar: el método del descenso del gradiente con y sin tasa de aprendizaje adaptativo, descenso del gradiente con variación del *momentum* o *Levenberg-Marquardt*.

- **Descenso del gradiente.**

Este método tiene un uso muy extendido debido a su sencillez, es un algoritmo iterativo de optimización que permite encontrar valores mínimos de funciones convexas y derivables en todo su dominio.

La función que evalúa como se ajusta la salida de la red al conjunto de datos disponibles se denomina término de error, y en este caso se notará como  $f$ , el valor del error depende de los parámetros de la red, los pesos de las neuronas y los bias, estos se agrupan en un vector de peso, que se notará como  $w$ . Por tanto, el error que comete la red con unos pesos  $w$  será  $f(w)$ . Por ende, el objetivo es encontrar el valor  $w^*$  para el que se obtenga un mínimo global de la función  $f$ . Para encontrar el mínimo de esta función se hace de una manera iterada, llegando a los parámetros adecuados.

Se comienza con un vector de parámetros inicial, normalmente elegido al azar, a continuación, se genera un nuevo vector de parámetros esperando que la función de error se reduzca, este proceso se repite hasta que se reduce el error por debajo de lo tolerable.

Si la función error es derivable, se puede calcular la primera derivada, obteniendo el vector gradiente:

$$\frac{\delta f}{\delta w_i}, \text{ para } i = 1, 2, \dots, n^I \quad (2.3)$$

Y la matriz Hessiana, calculada a partir de las derivadas parciales del vector gradiente, quedaría como:

$$H_{ij}f(w) = \frac{\delta^2 f}{\delta w_i \delta w_j}, \text{ para } i, j = 1, 2, \dots, n^I \quad (2.4)$$

A continuación, se muestra un resumen de la nomenclatura empleada:

- $w$ , vector de parámetros, notando  $w_0, w_1, \dots, w_i$  la sucesión de vectores de parámetros.
- $f(w_i)$ , el valor del error en el paso  $i$  de la iteración.
- $g_i$ , el valor del gradiente de la función de error en el paso  $i$ .

Para generar el siguiente vector de parámetros, se hace en función del gradiente, del modo que se indica en la Ecuación 2.5.

$$w_{i+1} = w_i - \nu_i g_i, \quad (2.5)$$

siendo  $\nu_i$  la tasa de aprendizaje, esta se selecciona para maximizar la cantidad a la que decrece la función de error en cada iteración. Se puede fijar a priori o calcularse mediante un proceso de optimización en cada iteración, normalmente para simplificar el proceso se suele seleccionar un valor fijo.

Algunos de los problemas que se plantean al aplicar este método, es que puede quedar atrapado en mínimos locales, sin llegar por tanto al mínimo de la función de error. Si el tamaño de paso es pequeño y si el valor óptimo está lejos la convergencia puede ser muy lenta.

• **Descenso del gradiente con tasa de aprendizaje adaptativo.**

Como se ve en el método del descenso del gradiente el rendimiento de dicho algoritmo es muy sensible a la tasa de aprendizaje,  $\nu_i$ . Por tanto, se puede mejorar este rendimiento si

<sup>I</sup>El vector gradiente queda como:  $\nabla f = \begin{pmatrix} \frac{\delta f}{\delta x_1} \\ \frac{\delta f}{\delta x_2} \\ \vdots \\ \frac{\delta f}{\delta x_n} \end{pmatrix}$

<sup>II</sup>La matriz Hessiana queda como:  $\nabla^2 f = \begin{pmatrix} \frac{\delta^2 f}{\delta x_1^2} & \frac{\delta^2 f}{\delta x_1 \delta x_2} & \cdots & \frac{\delta^2 f}{\delta x_1 \delta x_n} \\ \frac{\delta^2 f}{\delta x_2 \delta x_1} & \frac{\delta^2 f}{\delta x_2^2} & \cdots & \frac{\delta^2 f}{\delta x_2 \delta x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\delta^2 f}{\delta x_n \delta x_1} & \frac{\delta^2 f}{\delta x_n \delta x_2} & \cdots & \frac{\delta^2 f}{\delta x_n^2} \end{pmatrix}$

se permite cambiar a la tasa de aprendizaje durante el entrenamiento. En este método, se intenta mantener la tasa de aprendizaje lo mayor posible, siempre que sea estable.

Para realizar el entrenamiento, en primer lugar, se calcula la salida inicial de la red y el error, en cada iteración los nuevos pesos se calculan con la tasa actual de aprendizaje, y a continuación se calcula la nueva salida y el error. Si el nuevo error es mayor que el anterior, esos pesos se descartan y, además se disminuye la tasa de aprendizaje, multiplicándola por un factor reductor, normalmente 0,7. En caso contrario, es decir, que el nuevo error sea menor, los pesos se mantienen y se aumenta la tasa de aprendizaje mediante un factor incremental, normalmente 1,05.

De este modo se aumenta la velocidad de aprendizaje siempre que esto no conlleve un aumento del error, obteniéndose así una tasa de aprendizaje casi óptima. Mientras el aprendizaje se mantenga estable se incrementará la tasa de aprendizaje. Pero si la tasa de aprendizaje es demasiado grande para garantizar una disminución del error, esta se disminuye hasta conseguir estabilizar el aprendizaje.

- **Descenso del gradiente con variación del *momentum*.**

Este método lleva el descenso del gradiente en la dirección correcta, esto se hace añadiendo un factor, llamado *momentum*, que es el resultado de añadir el producto de un coeficiente ( $\gamma$ ) que varía entre 0 y 1 y el vector de pesos anterior al vector de pesos actual, quedando, por tanto:

$$w_{i+1} = w_i - M_t, \quad (2.6)$$

donde  $M_t$  es el momentum y se formula como se muestra en la Ecuación 2.7.

$$M_t = \gamma w_{i-1} + \nu_i g_i. \quad (2.7)$$

El *momentum* se incrementará para los gradientes que vayan hacia una dirección de forma consistente, mientras que, si no es así se reducirá, de tal modo que con este método se reducen las oscilaciones y se acelera la convergencia.

- **Levenberg-Marquardt backpropagation.**

Este método también es conocido como el método de mínimos cuadrados amortiguado, para su aplicación es necesario calcular el vector gradiente y la matriz Jacobiana. Si la función de error se puede expresar como:

$$f(w) = \sum_{i=0}^m e_i^2(w), \quad (2.8)$$

siendo  $m$  el tamaño del conjunto de datos de entrenamiento y  $e$  el vector en el que se acumulan los errores cometidos.

La matriz Jacobiana se puede calcular como la formada por las derivadas de los errores con respecto a los parámetros, del modo que se observa en la Ecuación 2.9.

$$J_{ij}(w) = \frac{\delta e_i}{\delta w_j}, \text{ donde } i = 1, \dots, m \text{ y } j = 1, \dots, n. \text{III} \quad (2.9)$$

siendo  $m$  el tamaño del conjunto de datos de entrenamiento y  $n$  el número de parámetros de la red neuronal.

El vector gradiente se puede calcular a partir de la transpuesta de la matriz Jacobiana y el vector de todos los términos de error:

$$g_i = \nabla f = 2J^T e. \quad (2.10)$$

El Hessiano se puede aproximar mediante la matriz Jacobiana y un factor de amortiguamiento,  $\lambda$ , que asegura la positividad del Hessiano.

$$H \approx 2J^T J + \lambda I, \quad (2.11)$$

siendo  $I$  la matriz identidad.

El proceso de mejora del vector de pesos mediante este algoritmo queda definido conforme a la Ecuación 2.12.

$$w_{i+1} = w_i - (J_i^T J_i + \lambda_i I)^{-1} (2J_i^T e_i). \quad (2.12)$$

Si  $\lambda$  se iguala a 0 se obtiene el método de Newton, mientras que si es muy grande se convierte en el algoritmo del descenso del gradiente, con una tasa de aprendizaje pequeña. Por ello,  $\lambda$  se inicializa con un valor grande, de modo que las primeras iteraciones vayan en la dirección de descenso del gradiente. En el caso de que alguna iteración genere un fallo entonces se incrementa  $\lambda$ , en caso contrario, a medida que disminuye el error,  $\lambda$  disminuye, de manera que el algoritmo de *Levenberg-Marquardt* se aproxima al método de Newton. Mediante este proceso normalmente se consigue acelerar la convergencia.

Si los conjuntos de datos y la red neuronal son muy grande la matriz Jacobiana se hace enorme, y por tanto, se requiere de mucha memoria de computadora, haciendo que los tiempos de cálculo sean muy grandes y dificultando el entrenamiento. Por este motivo, en muchas ocasiones conviene llevar el proceso de entrenamiento a clusters de computación para incrementar las capacidades de cálculo y reducir los tiempos de entrenamiento.

---

III La matriz Jacobiana queda como:  $J = \nabla(f) = \begin{pmatrix} \frac{\delta f_1}{\delta x_1} & \frac{\delta f_1}{\delta x_2} & \dots & \frac{\delta f_1}{\delta x_n} \\ \frac{\delta f_2}{\delta x_1} & \frac{\delta f_2}{\delta x_2} & \dots & \frac{\delta f_2}{\delta x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\delta f_m}{\delta x_1} & \frac{\delta f_m}{\delta x_2} & \dots & \frac{\delta f_m}{\delta x_n} \end{pmatrix}$

siendo  $m$  el tamaño del conjunto de datos de entrenamiento y  $n$  el número de parámetros de la red neuronal.

## 2.5. Conjuntos de datos

Como ya se comentó anteriormente, para la aplicación de las metodologías de IA, se necesitan ejemplos representativos del problema a solventar, para que se encuentren patrones. Estos ejemplos componen el conjunto de datos.

El total de los datos que se tiene, se debe dividir en tres conjuntos, de entrenamiento, de test y de validación. El conjunto de entrenamiento es el conjunto mayoritario y es el que se utiliza para entrenar la red. Los conjuntos de validación y test, normalmente son del mismo tamaño, el de validación es utilizado para validar el modelo, es decir, prevenir el sobreajuste o el infrajuste. Y el conjunto de test, es una porción de los datos que se mantiene aparte y sobre la cual se evalúa el modelo, normalmente la eficacia del modelo se calcula a partir de los resultados en este conjunto.

Normalmente el conjunto de entrenamiento suele ser entre el 50% y el 80% del total de los datos, y los conjuntos de validación y test en partes iguales.

Se emplean estos conjuntos para comprobar que la red está generalizando bien, es decir, que el error no se está reduciendo porque simplemente haya memorizado las muestras de entrenamiento.

El sobreajuste se da en los modelos generados con datos muy similares entre ellos, este modelo puede no detectar los patrones reales del proceso original y sin embargo presentar un error pequeño. Para prevenir este problema, como se ha indicado anteriormente, se divide el conjunto total de datos en tres conjuntos. Otro método de prevención es el conocido como *dropout*, este método consiste en ignorar temporal y aleatoriamente del cálculo algunas neuronas, este número depende de un factor de probabilidad, *dropout rate*.



# 3

## Estado del arte

---

En este tercer Capítulo se realiza una revisión de la literatura encontrada en cuanto a la aplicación de la inteligencia artificial en el campo de la ingeniería naval, centrandose más concretamente en los artículos de investigación focalizados en el campo de la hidrodinámica.

---

La Inteligencia Artificial tiene un enorme potencial en el ámbito de la Ingeniería Naval. Este potencial ha sido destacado en distintos trabajos que pueden ser encontrados en la literatura especializada. Por ejemplo, Gougoulidis [15] presenta una revisión de las aplicaciones de las RNA a la ingeniería naval. Este autor destaca, por ejemplo, que las RNA pueden ser muy útiles en la obtención de las dimensiones principales de un buque, así como la disposición general de buques de tipo portacontenedores. Más detalles sobre la aplicación de estas herramientas en las primeras fases del diseño de buque se pueden encontrar en Clausen et al. [16]. Gougoulidis también destaca que las RNA pueden ser aplicadas a otros ámbitos como el análisis estructural del casco, la propulsión o la hidrodinámica. Por ejemplo, en aplicaciones en la determinación de los coeficientes de resistencia residuo, de resistencia de generación de formas o para la búsqueda de patrones que relacionen características del buque con la menor resistencia al avance en aguas tranquilas.

Uno de los primeros trabajos que pueden encontrarse en el ámbito de la IA, en el ámbito de la Ingeniería Naval, fue el presentado por Mazaheri et al. [17] donde mediante el uso de RNA (Redes Neuronales Artificiales), se propone la predicción de la respuesta de un buque tanque FPSO de 200.000 toneladas de peso muerto sometido a viento, olas y cargas de corriente de forma aleatoria con el fin de reducir el tiempo computacional requerido. Posteriormente, estos autores compararon los resultados obtenidos mediante software numérico con los obtenidos por la RNA, comprobando la validez de la metodología empleada en la predicción del comportamiento de estructuras marinas frente a cargas ambientales aleatorias. Para alcanzar un grado de precisión elevado en el pronóstico del comportamiento de la estructura estos autores propusieron una metodología similar a la empleada en el presente trabajo. En primer lugar, mediante un modelo matemático determinaron la respuesta del buque tanque sometido a cargas ambientales. Con los resultados del análisis matemático generaron un conjunto de datos de entrenamiento, que posteriormente se usó para el algoritmo de aprendizaje. Ha de tenerse en

cuenta, que la mayor parte de estos trabajos son de aplicación limitada, pues únicamente se centran en un determinado tipo de estructura o buque. Se deduce de ello, que la aplicabilidad de los algoritmos presentados en estas investigaciones es de carácter limitado y en su mayoría presentan el fenómeno del sobreajuste o conocido en inglés como *overfitting*; esto es un alto grado de ajuste para casos similares al conjunto de entrenamiento, pero se comete un error considerable si se trata de aplicar a casos distintos de los usados en el entrenamiento, esto es, que la red no generaliza bien.

La mayor parte de los trabajos encontrados en la actualidad relacionados con la Inteligencia Artificial en el ámbito de la Ingeniería Naval están relacionados con la optimización de formas base o con el control dinámico de la embarcación. Por ejemplo, Yu y Wang [18] plantean el diseño y la modificación de formas de cascos complejas para un rendimiento óptimo del barco mediante el análisis de componentes principales para la obtención de los parámetros relevantes que afectan de forma significativa a la resistencia al avance. Concluyen que los coeficientes de bloque y prismático no representa la complejidad de las formas del casco. Posteriormente, con los resultados obtenidos que relacionan los parámetros del casco con la resistencia al avance plantean el entrenamiento de una red neuronal para tratar de establecer correlaciones entre estos parámetros y la resistencia al avance de forma que permite una generación, evaluación y optimización de forma sistemática.

Otro ejemplo de uso de Redes Neuronales Artificiales (RNA) es el trabajo de Magari et al. [19] en el que la IA se usa para la predicción de resistencia al avance en aguas tranquilas de las formas del casco diseñadas de acuerdo con la Serie Sistemática MARAD. Los datos experimentales del coeficiente de resistencia residuo de los casos de la serie MARAD son usados para entrenar y evaluar RNA con el objetivo de estimar el coeficiente de resistencia residual de los buques diseñados de acuerdo con la serie MARAD.

Otro ejemplo de aplicación es el usado por Im y Nguyen [20] para el atraque automático de barcos. En este trabajo se presenta un sistema de inteligencia artificial donde es posible aplicar el atraque automático en cualquier muelle sin necesidad de llevar a cabo el entrenamiento de la red para cada uno de los puertos.

Recientes ejemplos como los presentados por Sclavounos y Ma [21] usan herramientas de Machine Learning en el campo de la hidrodinámica marina para el estudio de ensayos de extinción para la determinación de la restauración a balance, el momento de inercia y el amortiguamiento viscoso, junto con el pronóstico de las elevaciones del mar. Para conseguir la predicción de estas variables hidrodinámicas estos autores usaron algoritmos de tipo SVM (*Support Vector Machine*) para llevar a cabo el entrenamiento de los algoritmos con unos resultados prometedores. Otro estudio relacionado con la predicción de los movimientos de balance puede ser encontrado en [22]. En este estudio mediante SVM combinado con Mínimos Cuadrados se propone el pronóstico de las componentes lineales y no lineales de balance de un casco trimarán.

Otro ejemplo, que se puede encontrar de la aplicación de RNA y uso de SVM es el trabajo presentado por Li et al. [23] donde pretende avanzar en uno de los aspectos claves en el proceso de diseño de un nuevo buque; el tiempo empleado en la optimización y obtención de una solución a la resistencia al avance mediante el análisis con Mecánica de Fluidos Computacional (*Computational Fluid Dynamics*, CFD). Estos autores mediante el control del error en algoritmos SVM proponen sustituir la CFD por modelos RNA y SVM para la optimización y determinación de la resistencia al avance en aguas tranquilas en buques. Mediante experimentación y comparación



son programas CFD potenciales y viscosos realizan comparaciones en donde se demuestra la habilidad de estos algoritmos para predecir patrones y variables de diseño para distintos tipos de buques.

El trabajo se estructura de la siguiente forma, una vez expuestas las bases de los algoritmos de aprendizaje máquina, se expondrá brevemente el problema matemático del comportamiento en la mar. Seguidamente, se expondrá la metodología empleada para llevar a cabo la generación de datos de entrenamiento para la red. A continuación, se expondrá las distintas alternativas usadas para poder definir las formas del buque. Se procederá, a exponer un análisis de sensibilidad de las distintas combinaciones de algoritmos y formas definición del buque, de las que se seleccionará aquellas que permitan alcanzar los mejores resultados. Finalmente, se expondrá el entrenamiento de la red y se expondrán los resultados. Además, se llevarán a cabo validaciones con resultados obtenidos mediante un programa de computación del comportamiento en la mar en el dominio de la frecuencia.



# 4

## Comportamiento en la mar

---

El presente Capítulo describe de manera detallada el problema del comportamiento en la mar, es decir, el campo en el que se quieren aplicar las herramientas de *Machine Learning*. Realizándose un breve resumen de en que consiste dicho campo de la Ingeniería Naval, la formulación que engloba el problema, así como las distintas metodologías que existen actualmente para atacar dicho problema.

---

Se define el estudio del comportamiento del buque en la mar como una metodología que tiene como objeto el estudio de los movimientos y de los esfuerzos producidos por las olas en los sistemas marinos. La respuesta de estructuras offshore a las cargas de oleaje resulta fundamental en su diseño y análisis. El oleaje causa la mayor parte de las cargas en las estructuras marítimas, sus efectos serán dinámicos y afectarán directamente a los esfuerzos de la estructura. Los movimientos del artefacto en los seis grados de libertad son los siguientes (véase Figura 4.1):

Traslaciones:

- ∨ Movimiento longitudinal: Largada (*Surge*).
- ∨ Movimiento transversal: Deriva (*Sway*).
- ∨ Movimiento vertical: Arfada (*Heave*).

Rotaciones:

- ∨ Según el eje longitudinal: Balance (*Roll*).
- ∨ Según el eje transversal: Cabeceo (*Pitch*).
- ∨ Según el eje vertical: Guiñada (*Yaw*).

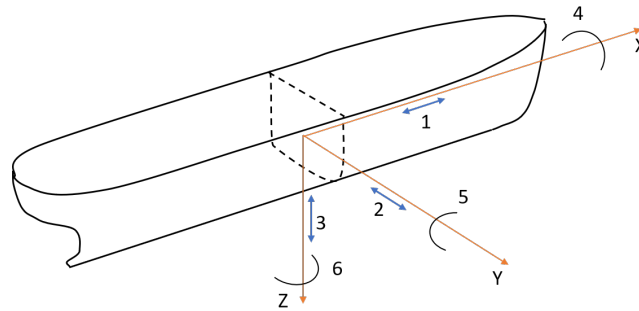


Figura 4.1. Grados de libertad del buque.

Parámetros tan importantes como la operatividad, la comodidad del pasaje, la propulsión, la maniobrabilidad, la operatividad de los equipos y sistemas, así como las importantes cargas estructurales a las que se ve sometido el buque que dependen directamente de él. También es importante conocer la dinámica del buque puesto que esto será imprescindible para evitar efectos de resonancia, que pueden causar la pérdida del buque.

Dependiendo del fin para el que ha sido diseñado el buque será más o menos relevante el estudio dinámico, un claro ejemplo de la necesidad de un buen estudio dinámico son todos los buques que tengan un helipuerto, ya que para las operaciones de despegue y aterrizaje sobre cubierta tienen unos requisitos de estabilidad del buque muy altos, también es muy importante en todos los buques militares, en la maniobra de efectuar el disparo, ya que esta no se puede realizar exitosamente si el buque no se encuentra estabilizado, se debe destacar la importancia del comportamiento en la mar para los buques de pasaje, ya que es importantísimo para que el pasaje disfrute del viaje.

Un buen estudio dinámico del buque es esencial para determinar la operatividad del mismo en las distintas zonas marítimas. Puesto que dependiendo de la zona a operar puede cambiar significativamente su diseño.

## 4.1. Formulación Básica

El problema del comportamiento en la mar se basa en la resolución de las ecuaciones de Navier-Stokes y por tanto la resolución de las fuerzas y presiones sobre el casco, y a partir de ello resolver el problema dinámico, el gasto computacional para la resolución de Navier-Stokes es altísimo, y por tanto se plantea una simplificación del problema, el error cometido en la simplificación en la mayoría de casos es asumible.

Las hipótesis de la simplificación son [24]:

- El flujo fluido es derivado de un potencial de velocidad, el fluido se supone homogéneo, incompresible, sin viscosidad e irrotacional.
- La amplitud de la ola es pequeña comparada con la longitud y la profundidad, por tanto, los movimientos del buque son pequeños. El buque oscila armónicamente con una frecuencia igual a la de encuentro con las olas.
- Se supone una relación lineal entre la amplitud de las olas y las amplitudes de los distintos movimientos del buque.
- Se considera el buque como un sólido rígido.

- ◊ La profundidad es constante.
- ◊ El lecho marino se considera impermeable.

El problema matemático linealizado se puede formular mediante el conjunto de Ecuaciones 4.1. En el que aparece la ecuación a resolver y las condiciones de contorno (C.C.) del problema.

$$\begin{aligned} \nabla^2 \phi &= 0, \text{ en } \Omega, \text{ Ecuación de Laplace} \\ \frac{\delta \phi}{\delta t} + g\xi &= 0, \text{ en } z = 0, \text{ C.C. dinámica en superficie libre} \\ \frac{\delta \xi}{\delta t} &= \frac{\phi}{\delta z}, \text{ en } z = 0, \text{ C.C. cinemática en superficie libre} \\ \frac{\delta \phi}{\delta z} &= 0, \text{ en } z = -h, \text{ C.C. cinemática en el fondo} \end{aligned} \quad (4.1)$$

donde  $\phi$  y  $\xi$  son el potencial de velocidades y la elevación de la superficie libre de las olas,  $\Omega$  es el dominio fluido,  $h$  la profundidad. En la Figura 4.2 se puede observar un resumen del problema matemático del problema de ondas por la teoría lineal.

El diagrama muestra un dominio fluido en un canal de profundidad  $h$ . El eje vertical es  $z$ , con  $z=0$  en la superficie libre y  $z=-h$  en el fondo. El eje horizontal es  $x$ , con  $x=0$  y  $x=\lambda$ . Las ecuaciones y condiciones de contorno son:

- En la superficie libre ( $z=0$ ):  $\frac{\delta \phi}{\delta z} + \frac{1}{g} \frac{\partial^2 \phi}{\partial t^2} = 0$
- En el fondo ( $z=-h$ ):  $\frac{\delta \phi}{\delta z} = 0$
- En las paredes laterales ( $x=0$  y  $x=\lambda$ ):  $\frac{\delta \phi}{\delta x} = 0$
- En el interior del dominio ( $\Omega$ ):  $\frac{\delta^2 \phi}{\delta z^2} + \frac{\partial^2 \phi}{\partial x^2} = 0$
- Condición de contorno en la superficie libre:  $\frac{\delta \phi}{\delta t} = (0, \lambda, t)$
- Condición de contorno en el fondo:  $\frac{\delta \phi}{\delta z} = (0, \lambda, t) = \frac{\delta \phi}{\delta z} = (0, \lambda, t + T)$

**Figura 4.2.** Esquema de ecuaciones y condiciones de contorno del problema lineal. Fuente [24].

Si se incluye la condición de periodicidad se pueden sustituir las variables  $x$  y  $t$  por el ángulo de fase, para poder reescribir y resolver de forma más sencilla la ecuación de Laplace. Tras realizar el cambio de coordenadas de  $(x, y, t)$  a teta, siendo el ángulo de fase,  $\theta = kx + ky - \omega t$ . Quedando el sistema de ecuaciones como se ve en el conjunto de ecuaciones 4.2 [25].

$$\begin{aligned} k^2 \frac{\delta^2 \phi}{\delta \theta^2} + \frac{\delta^2 \phi}{\delta z^2} &= 0, \text{ en } \Omega, \\ \frac{\delta \phi}{\delta z} &= 0, \text{ en } z = -h, \\ \frac{\delta \phi}{\delta z} + \frac{\omega^2 \delta^2 \phi}{g \delta \theta^2} &= 0, \text{ en } z = \xi, \\ -k \frac{\delta \phi}{\delta \theta} (0, z) &= -k \frac{\delta \phi}{\delta \theta} (2\pi, z). \end{aligned} \quad (4.2)$$

Resolviendo este sistema de ecuaciones se llega a:

$$\phi(\theta, z) = AC \frac{\cosh(k(z+h))}{\sinh(kh)} \sin(\theta) \quad (4.3)$$

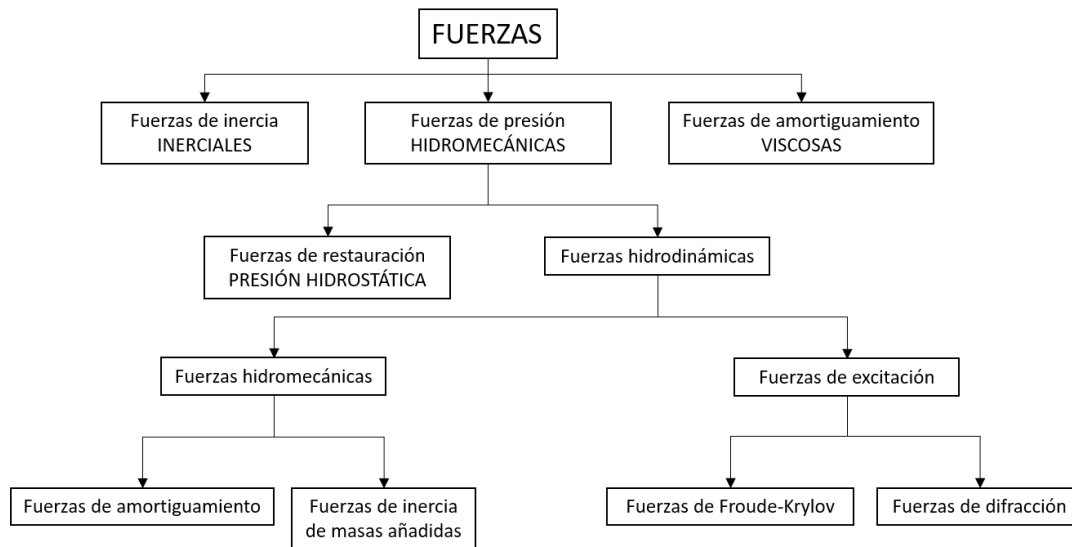
Quedando por determinar el valor de las constantes AC, esto se puede obtener a partir de la condición de contorno dinámica de la superficie libre, resultando,

$$\phi(\theta, z) = AC \frac{\cosh(k(z+h))}{\sinh(kh)} \sin(\theta) \quad (4.4)$$

Por tanto, se puede escribir el potencial de velocidades de acuerdo con la teoría lineal,

$$\phi = \frac{g\xi}{\omega} \frac{\cosh(k(z+h))}{\sinh(kh)} \sin(kx + ky - \omega t + \delta) \quad (4.5)$$

La excitación de la ola ejerce una serie de fuerzas oscilatorias sobre el cuerpo que hacen que este se mueva. Estas fuerzas pueden dividirse en tres grupos: fuerzas de inercia, fuerzas hidromecánicas y fuerzas de amortiguamiento viscoso (véase la Figura 4.3).



**Figura 4.3.** Esquema de descomposición de las fuerzas que actúan sobre un cuerpo en flotación. Fuente [24].

Se puede describir el oleaje como la composición de las contribuciones de distintas fuentes que se pueden clasificar como [24]:

- potencial de olas incidentes,  $\phi_I$ .
- potencial de olas generadas por el movimiento del artefacto flotante en cada uno de los seis grados de libertad,  $\sum_j^6 \phi_j^g$ .
- potencial de olas difractado-refractado, el tren de olas difractado refractado por la presencia de la estructura,  $\phi_{DR}$ . Debido a la impermeabilidad del objeto en flotación, la presencia del mismo provoca un flujo, y por eso se dan las ondas llamadas ondas de difracción, asociadas con  $\phi_{DR}$ . Este potencial es desconocido y debe satisfacer un problema donde el cuerpo oscila de manera opuesta con respecto a las ondas incidentes, esto es,  $-\delta\phi_{DR}/\delta\xi = 0$ .

El potencial total queda tal y como se formula en la Ecuación 4.6.

$$\phi = \phi_I + \phi_{DR} + \sum_j^6 \phi_j^g \quad (4.6)$$

Como ya se ha mencionado, la teoría lineal no contempla las fuerzas viscosas, que se suelen incluir mediante coeficientes de amortiguamiento calculados mediante ensayos [24, 26].

- **Fuerzas inerciales.**

Son aquellas provocadas por la propia aceleración del cuerpo.

$$[F_I] = -[M]^I [\ddot{\eta}], \quad (4.7)$$

siendo  $[\ddot{\eta}]$  el vector aceleración del cuerpo.

- **Fuerzas de restauración hidrostática.**

Son aquellas provocadas por el cambio de volumen sumergido al considerar el movimiento del buque.

$$[F_{RH}] = -[C]^II [\eta], \quad (4.8)$$

siendo  $[\eta]$  el vector desplazamiento del cuerpo.

- **Fuerzas de amortiguamiento.**

Las fuerzas de amortiguamiento se pueden conseguir mediante la linealización de las fuerzas de arrastre, estas están generadas por la componente normal de la velocidad del flujo. Su magnitud depende totalmente de la forma geométrica del cuerpo.

$$[F_B] = -[B] [\dot{\eta}_{relativa}], \quad (4.9)$$

siendo  $[\dot{\eta}_{relativa}]$  el vector velocidad relativa del cuerpo, pudiéndose escribir el vector velocidad relativa del cuerpo como se ve en la Ecuación 4.10.

$$[\dot{\eta}_{relativa}] = [\dot{\eta}_{cuerpo}] - [\dot{s}_{fluido}], \quad (4.10)$$

siendo  $[\dot{s}_{fluido}]$  el vector velocidad del fluido y  $[\dot{\eta}_{cuerpo}]$  el vector velocidad del cuerpo.

- **Fuerzas de masa añadida.**

Es la fuerza de inercia que ejerce el fluido sobre un sólido, debida a la aceleración del sólido.

---

<sup>I</sup>Quedando la matriz de masa como:  $[M] = \begin{pmatrix} m & 0 & 0 & 0 & mZG & -mYG \\ 0 & m & 0 & -mZG & 0 & -mXG \\ 0 & 0 & m & mYG & -mXG & 0 \\ 0 & -mZG & mYG & m_{44} & 0 & m_{46} \\ mZG & 0 & -mXG & 0 & m_{55} & 0 \\ -mYG & mXG & 0 & m_{64} & 0 & m_{66} \end{pmatrix}$

<sup>II</sup>Quedando la matriz de restauración como:  $[C] = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & C_{33} & 0 & C_{35} & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & C_{53} & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

$$[F_A] = -[AM]^{III} [\ddot{\eta}_{relativa}], \quad (4.11)$$

siendo  $[\ddot{\eta}_{relativa}]$  el vector aceleración relativa del cuerpo con respecto al fluido, pudiendose escribir el vector aceleración relativa del cuerpo como se ve en la Ecuación 4.12.

$$[\ddot{\eta}_{relativa}] = [\ddot{\eta}_{cuerpo}] - [\ddot{s}_{fluido}], \quad (4.12)$$

siendo  $[\ddot{s}_{fluido}]$  el vector aceleración del fluido y  $[\ddot{\eta}_{cuerpo}]$  el vector aceleración del cuerpo.

- **Fuerzas de excitación.**

Como se ha visto en la figura 4.3 las fuerzas de excitación se componen de las fuerzas de difracción y de las fuerzas de Froude-Krylov.

- **Fuerzas de Froude-Krylov.**

Hacen referencia a las fuerzas de presión debida al movimiento de la ola sobre el cuerpo.

$$[F_{FK}] = - \int_S p \vec{n} dS = -\rho \int_S \frac{\delta\phi}{\delta t} \vec{n} dS, \quad (4.13)$$

siendo  $p$  la presión ejercida por la ola sobre el cuerpo,  $\vec{n}$  el vector normal a la superficie del casco,  $\rho$  la densidad del agua del mar.

- **Fuerzas de difracción.**

Estas fuerzas provienen del potencial de olas difractadas-refractadas. Para cuerpos de pequeño tamaño sometidos a ondas de gran longitud y baja pendiente se puede formular como se ve en la Ecuación 4.14

$$[F_{DR}] = [\ddot{s}_{fluido}] [AM] + [\dot{s}_{fluido}] [B], \quad (4.14)$$

siendo  $[\ddot{s}_{fluido}]$  y  $[\dot{s}_{fluido}]$  la aceleración y la velocidad del fluido respectivamente.

Se puede observar que las fuerzas de difracción ya se tienen en cuenta en la parte fluida de las fuerzas de masa añadida y en las fuerzas de amortiguamiento.

Planteando el equilibrio de fuerzas resulta que,

$$[F_{FK}] = [F_{RH}] + [F_I] + [F_{AM}] + [F_B], \quad (4.15)$$

De manera extendida quedaría,

---

<sup>III</sup>Quedando la matriz de masa añadida como:  $[AM] = \begin{pmatrix} a_{11} & 0 & a_{13} & 0 & a_{15} & 0 \\ 0 & a_{22} & 0 & a_{24} & 0 & a_{26} \\ a_{31} & 0 & a_{33} & 0 & a_{35} & 0 \\ 0 & a_{42} & 0 & a_{44} & 0 & a_{46} \\ a_{51} & 0 & a_{53} & 0 & a_{55} & 0 \\ 0 & a_{62} & 0 & a_{64} & 0 & a_{66} \end{pmatrix}$



$$\rho \int_S \frac{\delta \phi}{\delta t} \vec{n} dS - [C] [\eta] - [M] [\ddot{\eta}] - [AM] ([\ddot{\eta}_{cuerpo}] - [\ddot{s}_{fluido}]) - [B] ([\dot{\eta}_{cuerpo}] - [\dot{s}_{fluido}]) = 0, \quad (4.16)$$

Si se considera que el cuerpo tiene un movimiento armónico, se puede escribir el movimiento del cuerpo del modo que se ve en la Ecuación 4.17.

$$\eta = H e^{-i\omega t}, \quad (4.17)$$

siendo  $H$  la amplitud del movimiento del cuerpo

Y por ende la velocidad y la aceleración del cuerpo será,

$$\dot{\eta} = -H i \omega e^{-i\omega t} \quad (4.18)$$

$$\ddot{\eta} = -H \omega^2 e^{-i\omega t} \quad (4.19)$$

Pudiendo formular la fuerza de excitación del modo indicado en la Ecuación 4.20.

$$[F_{ex}] e^{-i\omega t} = [F_{FK}] [AM] [\ddot{s}_{fluido}] + [B] [\dot{s}_{fluido}], \quad (4.20)$$

De acuerdo a todo lo expuesto en este apartado, se puede escribir la ecuación fundamental de la dinámica como se muestra en la ecuación 4.21.

$$[F_{ex}] e^{-i\omega t} = ([AM] + [M]) [\ddot{\eta}] + [B] [\dot{\eta}] + [C] [\eta], \quad (4.21)$$

Como ya se ha ido indicando a lo largo del documento, dado que el problema es muy complejo, normalmente se trata como un problema lineal; es decir, que la respuesta del cuerpo ante una ola de altura "h" será "H", Y si por ejemplo, la ola es de 2 h de altura, la respuesta ha de ser 2 H. Con esta premisa se puede definir el Operador de Respuesta en Amplitud, o en inglés *Response Amplitude Operator*, RAO, del modo que se muestra en la Ecuación 4.22.

$$RAO = \frac{H}{\xi}, \quad (4.22)$$

siendo  $H$  la amplitud del movimiento del cuerpo, y  $\xi$  la altura de la ola.

Se puede obtener la amplitud del movimiento del cuerpo,  $H$ , de la ecuación fundamental de la dinámica 4.21.

$$[F_{ex}] e^{-i\omega t} = -([AM] + [M]) H \omega^2 e^{-i\omega t} - [B] H i \omega e^{-i\omega t} + [C] H e^{-i\omega t}, \quad (4.23)$$

$$[F_{ex}] = -([AM] + [M]) H \omega^2 - [B] H i \omega + [C] H, \quad (4.24)$$

Quedando finalmente la amplitud movimiento del cuerpo del modo indicado en la Ecuación 4.25, y la fase del movimiento como se indica en la Ecuación 4.26.

$$H_i = \frac{F_{ex\ i}}{\sqrt{((AM_{i,j} + M_{i,j})\omega^2 + C_{i,j})^2 - (B_{i,j}\omega)^2}}, \quad (4.25)$$

con  $i \in [1, 6]$  y  $j \in [1, 6]$ , siendo  $i = j$ ,

$$\theta_i = \arctan\left(\frac{B_{i,j}\omega}{(AM_{i,j} + M_{i,j})\omega^2 + C_{i,j}}\right), \quad (4.26)$$

con  $i \in [1, 6]$  y  $j \in [1, 6]$ , siendo  $i = j$ ,

La RAO se calcula como se indicó en la Ecuación 4.22. Y si se considera una ola de altura un metro, la RAO quedaría finalmente como se ve en la Ecuación 4.27

$$RAO_i = H_i = \frac{F_{ex\ i}}{\sqrt{((AM_{i,j} + M_{i,j})\omega^2 + C_{i,j})^2 - (B_{i,j}\omega)^2}}, \quad (4.27)$$

con  $i \in [1, 6]$  y  $j \in [1, 6]$ , siendo  $i = j$ .

## 4.2. Metodologías

Para la resolución del problema del comportamiento en la mar se pueden aplicar distintas metodologías. La metodología de análisis utilizada en la resolución del problema, define la confianza del proceso de diseño, se diferencian cuatro metodologías [24, 27]:

- **Analítica**, aplicando la formulación anteriormente descrita, se resuelve el problema de manera lineal, para obtener la masa añadida, se puede aplicar la teoría de las rebanadas, y para obtener la matriz de amortiguamiento es necesaria la realización de un ensayo de extinción. Se obtendrían las fuerzas de Fruode-Krylov mediante la integración de la presión de la ola en el casco. Resolviendo esto para distintas frecuencias en los distintos movimientos se obtendría la RAO. A partir de esta y con un espectro de oleaje se obtendría el espectro de amplitud de los movimientos del buque en un determinado mar. El hecho de no poder obtener el amortiguamiento hidrodinámico sin un ensayo, así como la necesidad de discretizar el casco para obtener las fuerzas de presión y la problemática existente en la resolución del problema de difracción-refracción hacen que o bien se utilicen técnicas de computación numérica o se obtenga directamente de la experimentación.
- **Metodología numérica**, aquí se distinguen distintas metodologías como se ha indicado anteriormente, entre ellas destaca el método de los elementos finitos y el método de los elementos de contorno.
  - **Método de los elementos finitos, MEF**, en terminología inglesa *Finite Element Method*, **FEM**, es un método muy común para la resolución de problemas de interacción fluido estructura, usando tanto teoría potencial como teoría viscosa. El uso de la teoría viscosa requiere un gran gasto computacional. Para la resolución por este método se han desarrollado numerosas metodologías que se sitúan entre la teoría potencial y la teoría viscosa, estas metodologías tratan de incluir la mayor

parte de los efectos no lineales que tienen efecto, en este trabajo únicamente se tratará con la resolución lineal del problema. En este método se malla el volumen fluido y se calcula el campo de presiones y velocidades en todo el volumen. En este trabajo se ha utilizado dentro de este método de resolución el *solver* comercial *SeaFEM*. Usa una malla no estructurada que hace que tenga flexibilidad geométrica y que sea más rápida la inicialización [28]. Se basa en la resolución discreta del problema variacional:

$$\int_{\Omega} \nabla \nu_h \nabla \phi_h d\Omega = \int_{\Gamma^B} \nu_h \widehat{\phi}_n^B d\Gamma + \int_{\Gamma^R} \nu_h \widehat{\phi}_n^R d\Gamma + \int_{\Gamma^{Z_0}} \nu_h \widehat{\phi}_n^{Z_0} d\Gamma + \int_{\Gamma^{Z-h}} \nu_h \widehat{\phi}_n^{Z-h} d\Gamma, \quad (4.28)$$

$$\forall \nu_h \in Q_h$$

donde  $\widehat{\phi}_n^B$ ,  $\widehat{\phi}_n^R$ ,  $\widehat{\phi}_n^{Z_0}$  y  $\widehat{\phi}_n^{Z-h}$  son los gradientes de los potenciales correspondientes a las condiciones de contorno de Neumann, en el cuerpo, de radiación, superficie libre y en el fondo, respectivamente.

En el calculo de la RAO mediante *SeaFEM* se realiza analizando la respuesta de las series temporales del buque, usando un espectro de ruido blanco discretizado. Este espectro se define por un número N de ondas de igual amplitud y períodos que varían entre el valor máximo y el mínimo definido por el usuario. Los valores de estos N períodos se seleccionan para que coincidan con la transformada de Fourier de la señal de salida, que viene dada por,

$$X = \sum_{n=0}^{N^w-1} x_n e^{-i2\pi\Delta f * n}, \quad (4.29)$$

donde X es el vector que contiene los movimientos en los 6 grados de libertad del cuerpo, x el vector desplazamiento del cuerpo. Y  $T_{min}$  y  $T_{max}$ , son el periodo mínimo y máximo para el analisis, definiendo el incremento de frecuencia,  $\Delta f$ , conforme a la Ecuación 4.30.

$$\Delta f = \frac{1}{T_{min}} - \frac{1}{T_{max}} \cdot \frac{1}{N^w - 1}. \quad (4.30)$$

- **Método de los elementos de contorno**, o como es más conocido, en terminología inglesa **Boundary Element Method, BEM**, Este método resulta de uso común en la mayoría de los problemas de comportamiento en la mar, sobre todo para la resolución en problemas lineales o de segundo orden, basados en teoría potencial. En este método se debe dividir la estructura en paneles planos o curvos, dependiendo de si se realiza un calculo de bajo orden o de alto orden, en estos paneles, se determinan los potenciales de velocidad, con ello la distribución de presiones y, por tanto, las fuerzas en cada uno de los grados de libertad. Para aplicación de este método, se ha utilizado el *software* comercial *SeaBEM*, basado en *NEMOH* [29].

Para resolver el problema mediante BEM, se usa la función de Green que permite tener en cuenta las condiciones de contorno en el cuerpo, fondo y superficie libre. Se

discretiza la integración de los coeficientes en paneles, en el caso general se puede escribir,

$$\phi(\vec{X}) = \frac{1}{4\pi} \int_{S_b} G(\vec{X}, \vec{X}_S) \sigma(\vec{X}_S) dS, \quad (4.31)$$

donde  $\vec{X}$  es la posición del origen en la superficie del buque,  $S_b$  es la media de la superficie mojada,  $G(\vec{X}, \vec{X}_S)$  es la función de Green que describe el flujo en una localización debido a una fuerza en el origen  $\vec{X}_S$  y  $\sigma(\vec{X}_S)$  describe la fuerza en el origen  $\vec{X}_S$ .

El problema dinámico se puede resolver con un análisis en el dominio del tiempo o en el dominio de la frecuencia. El análisis en el dominio de la frecuencia únicamente capturará efectos lineales, mientras que si se realiza en el dominio del tiempo se pueden obtener los efectos no lineales y estados transitorios. Si bien es cierto, que el dominio de la frecuencia permite estimar los comportamientos críticos de la estructura frente a espectros de oleaje irregulares, así como la frecuencia de resonancia de la estructura. Siempre que se esté frente a un análisis lineal se podrá pasar del dominio del tiempo al de la frecuencia usando la transformada de Fourier.

- **Análisis experimental**, el análisis experimental se utiliza con distintas finalidades:
  - Investigar un problema que aun no tiene solución analítica.
  - Obtener coeficientes que se requiere para la aplicación de técnicas de predicción analítica, como ocurre con el amortiguamiento hidrodinámico.
  - Validar una técnica analítica de predicción del comportamiento del modelo.
  - Evaluar los efectos de alto orden que las simplificaciones analíticas no predicen, obteniendo un factor de correlación entre la predicción del comportamiento del modelo y el verdadero comportamiento.

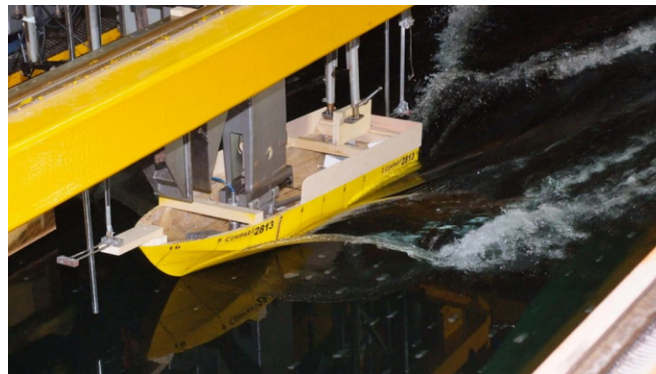
Para realizar este tipo de análisis es necesaria la generación de un modelo, se pueden distinguir tres tipos:

- Replica a escala del prototipo, el comportamiento del modelo será idéntico al del prototipo, pero la respuesta del modelo diferirá de la del prototipo según el factor de escala.
- Modelos distorsionados, en general existe semejanza entre el comportamiento del modelo y el del prototipo, en estos modelos se distorsionan algunos factores, como la eslora, etc.
- Modelos análogos, consisten en un sistema que en apariencia no se parece físicamente con el prototipo, pero son gobernados por la misma clase de ecuaciones, como la de Poisson o Laplace.

Para la experimentación de problemas hidrodinámicos y en los de flujo en superficie libre, normalmente los modelos son generalmente replicas a escala del prototipo. Aplicando el teorema de Pi Buckingham se sabe que las fuerzas viscosas son dependientes del número de Reynolds, y que las fuerzas gravitatorias son dependientes del número de Froude. Por la imposibilidad de realizar un ensayo a igualdad del número de Froude y de Reynolds, el número de Froude se considera como el criterio crítico de escala en los problemas de olas.

La importancia del número de Reynolds disminuye cuando este aumenta, por este motivo a menudo se induce artificialmente la turbulencia del flujo en la proa del modelo, de tal modo que los efectos que se obtienen son debidos a un alto número de Reynolds.

El modelo a construir ha de tener una alta precisión además de ser capaz de resistir el tiempo que sea necesario en el ambiente acuático, esto es difícil y requiere tiempo. Los materiales de fabricación del modelo más comunes son: madera y materiales compuestos. El factor de escala debe ser tal que permita realizar el ensayo, esto es, que sea apropiado para el canal donde se vaya a realizar, que permita introducir la instrumentación necesaria, otro aspecto muy importante a tener en cuenta al seleccionar las dimensiones del modelo es el coste de fabricación, a priori cuanto mayor sea el modelo más caro resultará, sin embargo si se realiza un modelo excesivamente pequeño puede complicarse la fabricación del mismo y será mucho más sensible.



**Figura 4.4.** Modelo de buque en ensayo de canal. Fuente: Canal de Experiencias Hidrodinámicas del Pardo, CEHIPAR.

El método de construcción del modelo depende del material utilizado, en el caso de utilizar madera, se parte de un bloque de dimensiones similares a las del modelo a fabricar y mediante una fresadora por control numérico se elimina el exceso de materia quedando el modelo, este método es muy similar cuando el material a utilizar es algún tipo de espuma. En el caso de utilizar materiales compuestos, con fibra de vidrio, se construye a partir de un molde sobre el cual se lamina [30].



# 5

## Motivaciones y objetivos

---

En este Capítulo, se presenta la idea principal del proyecto, así como lo que se pretende con él. Además, se explica como surge la idea a partir de la problemática encontrada para el diseño preliminar considerando el comportamiento en la mar del buque.

---

Como se ha comentado en la introducción, las herramientas de Inteligencia Artificial prometen una revolución en todos los campos de la ciencia, incluida la Ingeniería Naval. En este sentido, este trabajo pretende ahondar un poco más en la aplicación de estas herramientas en un campo tan específico como es el del comportamiento en la mar, más concretamente se pretende aplicar el uso de redes neuronales en dicho campo, es decir, desarrollar y entrenar una red neuronal para predecir el comportamiento de cualquier tipología de tipologías típicas de buques monocasco. Esta solución permite determinar parámetros clave en el estudio del comportamiento del buque, como son las fuerzas de interacción con el oleaje, las masas añadidas y el amortiguamiento por difracción y radiación del oleaje.

Como se ha visto en el **Capítulo 4**, *Comportamiento en la mar*, para la resolución de este problema en la actualidad se aplican distintas metodologías numéricas, ya sea en el dominio del tiempo o de la frecuencia para calcular el comportamiento en la mar del artefacto en última instancia, se suele comprobar con un ensayo de canal. Esto impide que el diseño se base en el comportamiento en la mar, si no que más bien es una comprobación al final del diseño y en caso de que no se comporte todo lo bien que se espera se realiza alguna modificación con todos los problemas que esto conlleva en un proyecto avanzado.

Por un lado, la aplicación de los métodos numéricos implica la preparación del modelo y el cálculo. El mallado de la geometría es un proceso lento y complicado, ya que la precisión de los resultados depende de la calidad de la malla, y que, por tanto, esta debe ser lo suficientemente robusta. Por otro lado, la obtención del comportamiento en la mar mediante experimentación es lento y caro, ya que se tiene que fabricar el modelo y realizar el ensayo. Por estos motivos es casi imposible diseñar en base a su comportamiento en el ámbito marino.

De este problema surge la idea de aplicar la inteligencia artificial a la resolución del problema del comportamiento en la mar. De tal modo que sirva para obtener un resultado inicial, con un

error asumible, en tiempo muy reducido y sin un elevado coste computacional, así se podrá resolver el problema en fase de diseño y por tanto diseñar teniendo en cuenta el comportamiento en la mar. Si bien es cierto que al final del proceso de diseño y si se precisa unos resultados completamente fidedignos se deberá aplicar o bien un método numérico o bien realizar un ensayo.



# 6

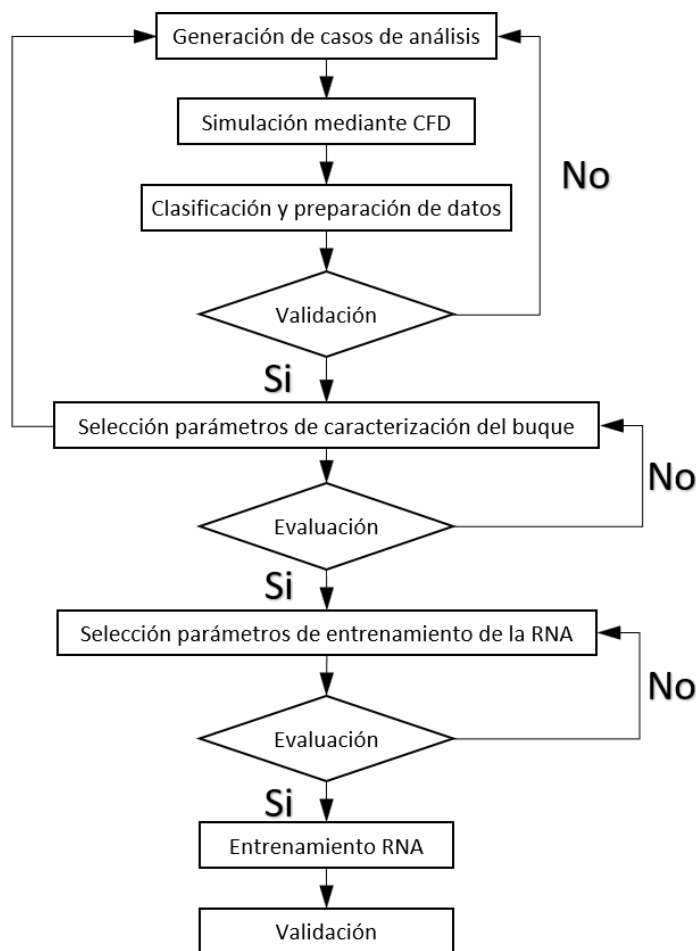
## Metodología

---

Este Capítulo recoge los resultados de las pruebas realizadas sobre el algoritmo definitivo con el fin de analizar el funcionamiento del mismo. Se ha medido tanto el tiempo de ejecución del algoritmo como el número de códigos que es capaz de leer correctamente con distintas orientaciones y condiciones de iluminación.

---

Para conseguir el fin último del presente proyecto, es decir, la generación de un algoritmo basado en redes neuronales capaz de estimar el comportamiento en la mar de un buque monocasco cualquiera, se ha empleado una metodología para la generación y el entrenamiento del algoritmo. Como ya se indicó en anteriores capítulos, el comportamiento en la mar de la embarcación se estimará mediante las curvas del Operador de Respuesta en Amplitud (*Response Amplitude Operator*, RAO). Se sabe que el comportamiento en la mar está gobernado por distintos parámetros, que son resultado de distintas hipótesis relativas al fluido, el oleaje y a la interacción del cuerpo con la mar. La metodología planteada se puede resumir en la Figura 6.1, donde se muestra el flujograma seguido hasta llegar a la generación un RNA válida para la predicción de masas añadidas, amortiguamiento, fuerzas de radiación-difracción y curvas RAO.



**Figura 6.1.** Flujograma seguido para el entrenamiento de la RNA para el estudio del comportamiento en la mar.

Cómo se observa en la Figura 6.1, la secuencia de trabajo seguida tiene varias partes bien diferenciadas, partiendo de la generación de casos de análisis, es decir los buques que se utilizarán para entrenar la red, siendo necesario obtener los resultados de cada uno de ellos mediante simulación mediante CFD. Se ve la necesidad de definir un modo de caracterización del buque asumible por la red, es por ello que un apartado dentro de la metodología consiste en seleccionar los parámetros capaces de caracterizar el buque. Del modo en que se vio en el **Capítulo 2, Machine Learning**, en el diseño de la red neuronal es tarea del diseñador probar las distintas arquitecturas, así como los distintos algoritmos de entrenamiento para obtener el que mejor resultados arroja, es por ello que se incluye una tarea de selección de parámetros de la RNA. Finalmente, una vez se tienen todos los buques generados, con sus resultados correspondientes, se conoce el método de caracterización del buque, y se ha definido la RNA óptima, solo queda entrenar las distintas RNA para obtener los resultados deseados, es decir, matriz de masas añadidas, matriz de amortiguamiento, fuerzas de radiación-difracción y las curvas RAO. A modo de validación se realizará una comparativa entre resultados obtenidos mediante CFD y los obtenidos mediante la RNA.

## 6.1. Generación de casos de entrenamiento

Para generar un algoritmo de RNA que permita estimar con una precisión aceptable los aspectos clave del comportamiento en la mar de cualquier tipo de buque, se requiere fijar un conjunto de cascos que cubran la mayor parte de las tipologías que se puedan encontrar navegando. El estudio aquí presentado se centra en buques de tipo monocasco, que van desde buques de apoyo a plataforma hasta buques tanque. La Tabla 6.1 se presentan las distintas tipologías de buques base empleadas para el entrenamiento de la RNA.

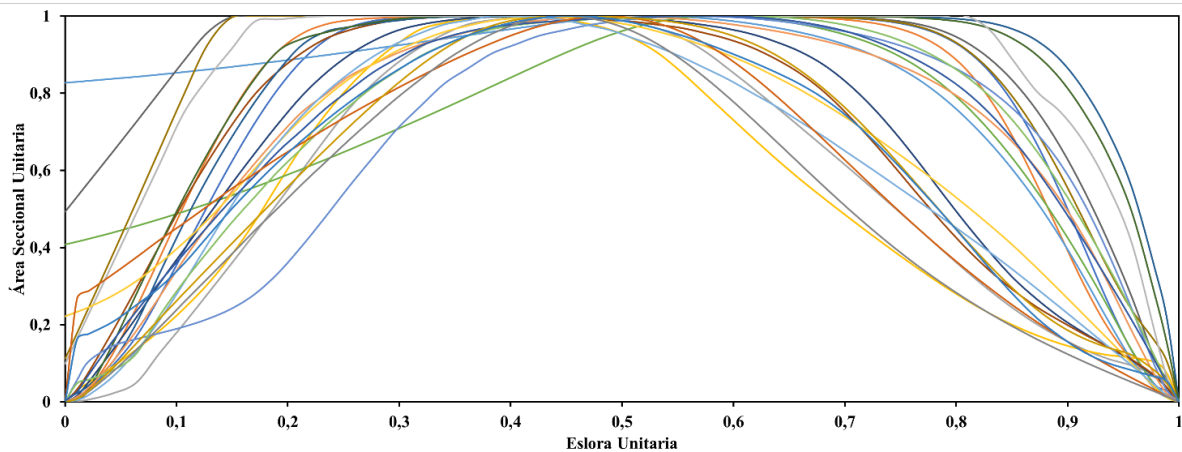
**Tabla 6.1.** Características de los buques base empleados en el entrenamiento de la RNA.

Tipo	Eslora Total (m)	Manga máxima (m)	Calado (m)	Desplazamiento (Tm)	Coef. Bloque ( $C_b$ )
Tanque	156,1	23,1	10,0	27369	0,74
Fragata	80,2	12,6	4,0	2138	0,51
Portacontenedores	109,6	16,5	6,5	6656	0,55
Crucero	325,5	38,6	8,8	67556	0,60
Perforador ( <i>drillship</i> )	226,1	38,0	14,0	107587	0,87
Remolcador de puerto	22,9	7,8	3,0	341	0,62
FPSO	317,8	54,3	12,0	175705	0,83
Ro-Ro	160,3	22,7	5,5	12479	0,61
Patrullero	75,8	12,8	4,5	2201	0,49
<i>Ship pro</i>	79,1	13,5	6,0	4636	0,71
X-Bow	63,4	17,4	6,8	4765	0,62
Suministro	28,5	7,1	2,5	276	0,53
Semisumergible ( <i>Heave Lift</i> )	182,9	50,0	9,0	71340	0,84

Puesto que se pretende que la RNA prediga el comportamiento en la mar de cualquier buque monocasco, es decir generalice bien, es necesario que el conjunto de buques que se le suministran tengan formas bien diferenciadas. Con los trece buques base que se muestran en la Tabla 6.1 se ha intentado cubrir el mayor espectro de formas posibles, además cabe destacar que para cada tipología de buque, salvo en el caso del FPSO, X-Bow, Ro-Ro y *Heave Lift*, se han fijado dos condiciones de carga con calados totalmente distintos.

Se decide, para facilitar la labor de generar los buques de entrenamiento, adimensionalizar los buques con respecto a la eslora, es decir, todos los buques base se escalan de tal modo que tengan una eslora total igual a un metro.

Se puede apreciar en la Figura 6.2, en la que se muestran las curvas de áreas seccionales de los distintos buques base incluyendo ambas condiciones de carga, como se cubre casi todo el espectro de posibilidades en cuanto a formas, yendo desde buques con un gran cuerpo cilíndrico y hombros pronunciados hasta buques con formas muy hidrodinámicas.



**Figura 6.2.** Áreas seccionales unitarias de los distintos buques seleccionados para el entrenamiento de la RNA.

Un aspecto fundamental para establecer un algoritmo RNA que permita hacer un pronóstico del comportamiento en la mar es disponer de un número suficiente de datos de entrenamiento. A priori, disponer de una cantidad suficiente (miles o millones) de casos para entrenar la RNA, es decir, buques con sus datos correspondientes en cuanto al comportamiento en la mar resulta muy complicado, y a veces imposible, ya que no son datos de libre disposición y generarlos supone un coste tanto económico como temporal muy grande.

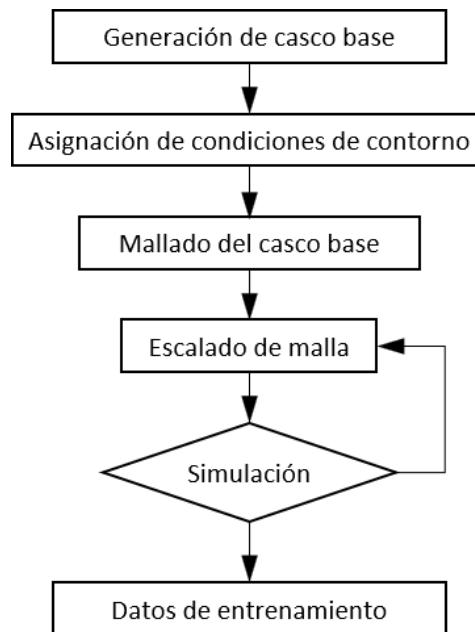
Como se puede apreciar, los casos presentados con anterioridad, tanto en los correspondientes a las áreas presentadas en la Figura 6.2, como los expuestos en la Tabla 6.1, no constituyen un conjunto de datos suficientes para aplicar metodologías de *Machine Learning*, ni mucho menos para conseguir una RNA que generalice bien. Para conseguir un conjunto de datos más o menos suficiente para trabajar con estas técnicas, se propone una variación sistemática de las formas de los buques base. Como ya se ha indicado primeramente se adimensionalizan los buques en eslora, esto quiere decir que se aplica un factor de escala  $\lambda$  consiguiendo una eslora unitaria. Para conseguir modificar la manga y el calado, se realizarán variaciones en la relación eslora-manga,  $L/B$ , variando esta relación entre 3 y 7, ambos límites incluidos, y además se modificará la relación manga-calado,  $B/T$ , esta relación se modificará entre 1 y 3, incluyendo también los dos límites. Aplicando estos factores de escala de un modo combinado se consigue un total de 576 casos de entrenamiento por cada tipo de buque base.

Como resultado de todo este proceso de generación de casos de entrenamiento se llega a disponer de 12.672 casos, que se consideran suficientes como para llevar a cabo el entrenamiento de la RNA. Sin embargo, el siguiente paso será disponer de los datos correspondientes al comportamiento en la mar de los distintos tipos de buques generados, esto es, matrices de masas añadidas y amortiguamiento por radiación, y fuerzas de difracción en cada una de las direcciones consideradas, así como de las curvas RAO para cada uno de los casos de entrenamiento.

## 6.2. Obtención de datos de entrenamiento

Cabe recordar, cómo se indicó en el **Capítulo 5, Motivaciones y objetivos**, que el trabajo se centra en la determinación de las curvas de respuesta en amplitud de buques monocasco para distintas direcciones de incidencia del oleaje, por tanto y para entrenar la RNA será necesario obtener estos datos, es decir, matrices de masas añadidas, matrices de amortiguamiento y fuerzas de difracción para cada dirección del oleaje, con estos resultados una vez clasificados se procederá al entrenamiento de la red.

En la Figura 6.3 se muestra un flujograma con la secuencia de trabajo seguida, tanto para la obtención de casos de entrenamiento, como para la simulación y obtención de los datos correspondientes al comportamiento en la mar.



**Figura 6.3.** Flujograma seguido para la generación de casos de entrenamiento y simulación de estos.

Para disponer de datos de entrenamiento, es decir los datos sobre el comportamiento en la mar de cada casos diseñados anteriormente, se debe realizar una experimentación. Esto resulta inviable técnica, económica y temporalmente. Es por ello que la solución que se plantea a esta problemática es la simulación numérica de forma secuencial de cada uno de los miles de cascos generados. Para llevar a cabo estas simulaciones se realizan mediante el método de los elementos de contorno en el dominio de la frecuencia, a través del módulo *SeaBEM* del programa comercial *SeaFEM* (<http://www.compassis.com/compass/es/Productos/SeaFEM>), calculando los casos de simulación para cada buque de entrenamiento, estas simulaciones se realizan para obtener los valores numéricos de las curvas RAO, matrices de masas añadidas, de amortiguamiento y vectores de fuerzas de difracción.

En el proceso de mallado del casco base, cabe destacar que para cada caso, será necesario generar una malla de paneles adaptada al cuerpo, que ha de generarse minuciosamente y ha de ser perfectamente simétrica para conservar determinadas propiedades hidrodinámicas. Para generar el número total de casos de entrenamiento se aplicarán las escalas comentadas en el apartado **6.1 Generación de casos de entrenamiento** a los nodos correspondientes a la malla de los casos base.

Una vez generadas las mallas del total de los casos de entrenamiento, es necesario indicar las condiciones de contorno y condiciones de cálculo de los casos a simular. Estas condiciones se generan y se indican en un archivo ".CAL". El número de direcciones elegidas es de 7 en el rango entre 0 y 180°, puesto que se van a generar las curvas RAO y estas, según se vió en el **Capítulo 4, Comportamiento en la mar**, se generan a partir de adimensionalizar el movimiento del buque con la amplitud de la ola, se realizarán los calculos con amplitudes de onda de 1 m. Se calcularán para 20 periodos equiespaciados entre 0,18 y 0,80 segundos, siendo estos periodos los correspondientes a una ola de 0,05 m y de 1 m de longitud de onda, es decir, que se obtienen resultados para una ola mínima del 5 % de la eslora del modelo y una ola máxima igual a la eslora del mismo, todos los cálculos se realizan para los seis grados de libertad. Consiguiendo mediante estas condiciones que los resultados de todos los casos de entrenamiento se generen para los mismos periodos y las mismas direcciones, esto es absolutamente imprescindible para alimentar la RNA.

Resumiendo el proceso seguido para generar los casos a simular y llevar a cabo las simulaciones, los pasos que se han seguido son: crear un archivo de apoyo, en el que se incluye todo lo necesario para generar los casos a simular, a partir de este se generan dichos casos y se simulan. A continuación, se muestran unos pseudocódigos en los que se explica como se ha realizado este proceso.

<b>GENERACIÓN DEL ARCHIVO DE APOYO</b>
--

<i>El archivo de apoyo, consiste en un documento de texto en el que se incluye información necesaria para la generación de los casos a correr a través del programa de computación numérica.</i>
--

\*\*\*\*Se genera el directorio de trabajo donde se creará el archivo de apoyo.

\*\*\*\*A continuación, se definen la manga, el calado y la masa de cada uno de los buques base, la eslora no hace falta puesto que se adimensionalizó, por ende, es 1 m para todos los buques.

set Bi "Manga del buque tipo i"  
 set Ti "Calado del buque tipo i"  
 set Mi "Masa del buque tipo i"

\*\*\*\* Se haría con todos los tipos de buques que se tengan.

for {set Tp 1} {\$Tp <= n} {incr Tp} {

    for {set ll VminL} {\$ll <= VmaxL} {set lb [expr ((VmaxLB-VminLB)/nLB) + \$lb]} {

        for {set bt VminBT} {\$bt <=VmaxBT} {set bt [expr ((VmaxBT-VminBT)/nBT)+ \$bt]} {

\*\*\*\* Mediante el comando *if*, se cargará el valor de la manga, el calado y la masa del buque base que se esté escalando en cada momento.

```
if { $Tp == i } {

    set Bo $Bi
    set To $Ti
    set Mo $Mi

}
```

\*\*\*\* Se calculan los valores de eslora, manga y calado que debería tener el buque aplicando los factores de escala y relaciones expuestas anteriormente.

```
set L [expr $Lo*$ll]
set B [expr $L/$lb]
set T [expr $B/$bt]
```

\*\*\*\* Se calculan los factores de escala a partir de los valores que tiene el buque con relación a los del buque base.

```
set Sx [expr $L/$Lo]
set Sy [expr $B/$Bo]
set Sz [expr $T/$To]
```

\*\*\*\* Puesto que la eslora de todos los barcos es adimensional, valor 1, las escalas se adimensionalizan mediante la escala en X, de tal modo que el valor X es invariable e igual a 1 m. Además, se calcula el valor de la masa del buque a partir de las escalas.

```
set SX [expr $Sx/$Sx]
set SY [expr $Sy/$Sx]
set SZ [expr $Sz/$Sx]
set M [expr $Mo*$SX*$SY*$SZ]
```

\*\*\*\* Se escribe en el archivo de texto creado al principio, en el que se incluye el tipo de buque, la escala X, Y, Z y la masa del buque correspondiente, además, en la primera columna se indica a que número de buque corresponde.

```
puts "en el archivo de apoyo" "$cont $Tp $SX $SY $SZ $M"

}

}

}
```

siendo  $T_p$  es el tipo de buque base,  $l$  es el factor de escala en eslora,  $l_b$  es la relación  $L/B$ ,  $b_t$  es la relación  $B/T$ .

En el código empleado para la realización del proyecto, las constantes tomaban los valores que aparecen en la Tabla 6.2.

**Tabla 6.2.** Valores tomados para las constantes empleadas en el código *GENERACIÓN DEL ARCHIVO DE APOYO*.

Constante	Valor
n	22
VminL	0,5
VmaxL	3,5
nL	15
VminLB	3
VmaxLB	7
nLB	5
VminBT	1
VmaxBT	3
nBT	5

**ESCALAR BUQUES Y GENERAR EL ARCHIVO DE CÁLCULO.**

*Este código consta de dos partes, en primer lugar, a partir de la malla base y de las escalas guardadas en el archivo de apoyo antes creado, se genera un archivo de malla con las coordenadas de los nodos de la malla base convenientemente escaladas. En segundo lugar, se genera el archivo (.CAL), en el que se incluye toda la información necesaria para llevar a cabo la simulación, como son las condiciones de contorno.*

```

****Se generan los siguientes directorios:

*   Donde se encuentra el archivo de apoyo.
*   Donde se encuentran los archivos de mallas de los buques base.
*   Donde se generarán los buques escalados.

**** Se carga el archivo de apoyo.

set j 0

while { el archivo de apoyo no se haya leído completamente } {

    set Tp(1,$j)  [dato posición 1]
    set Sx(1,$j)  [dato posición 2]
    set Sy(1,$j)  [dato posición 3]
    set Sz(1,$j)  [dato posición 4]
    set M(1,$j)   [dato posición 5]

```



\*\*\*\* Se carga la información de la malla del buque base correspondiente.

\*\*\*\* Se genera la carpeta donde se acumulará la información de cada buque a simular.

[Buque-\$Tp(1,\$j)-\$j.gid]

\*\*\*\* Se crean dentro de la carpeta “.gid”, las carpetas necesarias para llevar a cabo las simulaciones.

[kochin]

[mesh]

[results]

\*\*\*\* Se crea dentro de esta misma carpeta un archivo de texto donde se incluye la licencia del programa de simulación.

\*\*\*\* Se genera el archivo donde se escribirá la malla del caso a simular.

[Body\_data\_Auto1\_fd.msh]

\*\*\*\* A continuación, se obtiene el número de nodos y de elementos que tiene la malla base correspondiente.

```
set k 0
```

```
while {mientras no se haya leído completamente el archivo de malla del buque base} {
```

```
  if {Si se encuentra escrito “End Coordinates” } {
```

```
    set a $k
```

```
    set NoN [expr $a - 2]
```

```
  }
```

```
  if {Si se encuentra escrito “End Elements” } {
```

```
    set b $k
```

```
  }
```

```
  incr k
```

```
}
```

```
close $archivo de malla base
```

```
set i 0
```

\*\*\*\* Se vuelve a abrir el archivo de malla del buque base correspondiente.

```
while { mientras no se haya leído completamente el archivo de malla del buque base } {
    if { $i < 1 } {
        puts $archivo de malla escalada "MESH dimension 3 ElemType Quadrilateral Nnode 4"
        puts $ archivo de malla escalada "Coordinates"
    }

    if { $i > 1 && $i < [ expr $NoN + 2 ] } {
```

\*\*\*\* Se cargan los valores de las coordenadas de los nodos del archivo de malla original y después se multiplican por la escala generada en el archivo de apoyo.

```
    set num(1,$i)    [dato posición 0]
    set xi(1,$i)     [dato posición 1]
    set yi(1,$i)     [dato posición 2]
    set zi(1,$i)     [dato posición 3]

    set x_(1,$i) [expr $xi(1,$i) * $Sx(1,$j)]
    set y_(1,$i) [expr $yi(1,$i) * $Sy(1,$j)]
    set z_(1,$i) [expr $zi(1,$i) * $Sz(1,$j)]

    puts $archivo de malla del caso a simular "$num(1,$i) $x_(1,$i) $y_(1,$i) $z_(1,$i)"
}

if { $i == [expr $NoN + 2] } {
    puts $archivo de malla del caso a simular "End Coordinates "
    puts $archivo de malla del caso a simular "Elements"
}

if { $i > [expr $NoN + 4] && $i < $b } {

    set num_e(1,$i)    [dato posición 0]
    set xe(1,$i)       [dato posición 1]
    set ye(1,$i)       [dato posición 2]
    set ze(1,$i)       [dato posición 3]
```

```

set we(1,$i)      [dato posición 4]

puts $archivo de malla del caso a simular "$num_e(1,$i) $xe(1,$i) $ye(1,$i) $ze(1,$i)
$we(1,$i)"

}

if { $i == $b } {

puts $archivo de malla del caso a simular "End Elements"

}

incr i

}

set NoE [expr $i - $NoN -6]

close $archivo de malla del caso a simular

##### Generación del Fichero de Cálculo #####
#####

**** Se genera el archivo donde se escribirá la información correspondiente al cálculo.

[Buque-$Tp(1,$j)-$j.cal]

puts $archivo Cal. del caso a simular "- - - Environment - - - - -"

puts $archivo Cal. del caso a simular "1025 ! RHO ! KG/M**3 !
Fluid specific volume
9.8066 ! G ! M/S**2 ! Gravity
0 ! DEPTH ! M ! Water depth
0.0 0.0 ! XEFF YEFF ! M ! Wave measurement point"

puts $archivo Cal. del caso a simular "- - - Description of floating bodies - - - - -"

puts $archivo Cal. del caso a simular "1 ! Number of bodies"

puts $archivo Cal. del caso a simular "- - - Body data Auto1 - - - - -"

puts $archivo Cal. del caso a simular "Body_data_Auto1_fd.msh
! Name of mesh file
$NoN $NoE ! Number of nodes and number of elements
6 ! Number of degrees of freedom

```

```

1 1 0 0 0 0 0 ! Surge
1 0 1 0 0 0 0 ! Sway
1 0 0 1 0 0 0 ! Heave
2 1 0 0 0.0 0.0 0.0 ! Roll about CdG
2 0 1 0 0.0 0.0 0.0 ! Pitch about CdG
2 0 0 1 0.0 0.0 0.0 ! Yaw about CdG
6 ! Number of resulting generalised forces
1 1 0 0 0 0 0 ! Force in x direction
1 0 1 0 0 0 0 ! Force in y direction
1 0 0 1 0 0 0 ! Force in z direction
2 1 0 0 0.0 0.0 0.0 ! Moment force in x direction about CdG
2 0 1 0 0.0 0.0 0.0 ! Moment force in y direction about CdG
2 0 0 1 0.0 0.0 0.0 ! Moment force in z direction about CdG
24 ! Number of lines of additional information
1 0.0 0.0 0.0 ! Center Of Gravity of the body
2 1.0 0 0 ! Radii of gyration of the body (matrix): Rxx Rxy Rxz
3 0 1.0 0 ! Rxy Ryy Ryz
4 0 0 1.0 ! Rxz Ryz Rzz
5 $M(1,$j) ! Mass of body
6 0.0 0.0 0.0 ! Reference point
7 0.0 0.0 0.0 0.0 0.0 0.0 ! Additional Inertia matrix (row 1)
8 0.0 0.0 0.0 0.0 0.0 0.0 ! Additional Inertia matrix (row 2)
9 0.0 0.0 0.0 0.0 0.0 0.0 ! Additional Inertia matrix (row 3)
10 0.0 0.0 0.0 0.0 0.0 0.0 ! Additional Inertia matrix (row 4)
11 0.0 0.0 0.0 0.0 0.0 0.0 ! Additional Inertia matrix (row 5)
12 0.0 0.0 0.0 0.0 0.0 0.0 ! Additional Inertia matrix (row 6)
13 0.0 0.0 0.0 0.0 0.0 0.0 ! Additional Restoration matrix (row 1)
14 0.0 0.0 0.0 0.0 0.0 0.0 ! Additional Restoration matrix (row 2)
15 0.0 0.0 0.0 0.0 0.0 0.0 ! Additional Restoration matrix (row 3)
16 0.0 0.0 0.0 0.0 0.0 0.0 ! Additional Restoration matrix (row 4)
17 0.0 0.0 0.0 0.0 0.0 0.0 ! Additional Restoration matrix (row 5)
18 0.0 0.0 0.0 0.0 0.0 0.0 ! Additional Restoration matrix (row 6)
19 0.0 0.0 0.0 0.0 0.0 0.0 ! Additional Damping matrix (row 1)
20 0.0 0.0 0.0 0.0 0.0 0.0 ! Additional Damping matrix (row 2)
21 0.0 0.0 0.0 0.0 0.0 0.0 ! Additional Damping matrix (row 3)
22 0.0 0.0 0.0 0.0 0.0 0.0 ! Additional Damping matrix (row 4)
23 0.0 0.0 0.0 0.0 0.0 0.0 ! Additional Damping matrix (row 5)
24 0.0 0.0 0.0 0.0 0.0 0.0 ! Additional Damping matrix (row 6)”

puts $archivo Cal. del caso a simular “- - - Load cases to be solved - - - - -”

puts $archivo Cal. del caso a simular “20 35.11069577 7.850990247 ! Number of wave
frequencies, Min, and Max (rad/s)
7 0.0 180.0 ! Number of wave directions, Min and Max (degrees)”

```

```

puts $archivo Cal. del caso a simular "- - - Post processing - - - - -"

puts $archivo Cal. del caso a simular "1 0.1 10. ! IRF ! IRF calculation (0 for no
calculation), time step and duration
1 ! Show pressure
721 0. 360. ! Kochin function ! Number of directions of calculation (0 for no calculations),
Min and Max (degrees)
none 0 0 ! Free surface elevation ! Name of free surface file ('none' for no free surface),
number of nodes and number of panels."

puts $archivo Cal. del caso a simular "- - - GiD Inputs - - - - -"

puts $archivo Cal. del caso a simular "0.0 0.0 ! Speed (m/s) ! Speed direction (degrees)
5 ! Results file
0 ! Critical Damping Coefficient
8 ! Number of CPUs"

incr j

}

```

siendo “j” un contador que indica en que línea del archivo de apoyo se está, “k” e “i” indican en que línea del archivo de malla se está. “NoN” es el número de nodos que tiene la malla base correspondiente. “b” es la variable donde se guarda el número de filas total que hay en el archivo de malla base y “NoE” es el número de elementos que tiene la malla base correspondiente.

### SIMULACIONES SECUENCIALES.

*Este código consigue, utilizando el archivo de apoyo para saber en que caso se está, simular mediante SeaBEM los casos generados anteriormente de manera secuencial.*

```
set exec_file [file join {C:/Program Files/Compass/Tdyn15.1.0
x64/problemtypes/compassfem.gid/exec} seabem.exe]

****Se carga el directorio donde esta el archivo de apoyo.

**** Se abre el archive de apoyo.

set e 0

while { el archivo de apoyo no se haya leído completamente } {

    set Tp(1,$e)    [dato posición 1]

    **** Se genera el directorio de la carpeta donde se incluye todo lo necesario para la
    simulación de cada caso.

    cd $Directory

    catch {

        exec $exec_file -r $Directory Buque-$Tp(1,$e)-$e

    }

    incr e

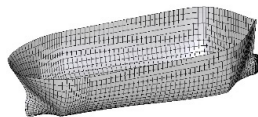
}
```

donde “e” es un contador que indica en que línea del archivo de apoyo se está.

En las siguientes figuras se muestra un ejemplo de la malla empleada para cada caso base y unos ejemplos de la resultante tras el proceso de escalado.



(a) Buque tanque original.

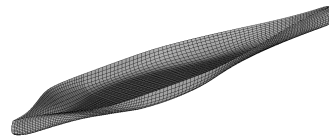


(b) Buque tanque con variación de calado.

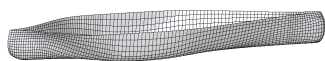


(c) Buque tanque con variación de manga.

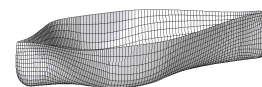
**Figura 6.4.** Imágenes de distintas escalas del buque tanque usadas para el entrenamiento.



(a) Buque portacontenedores original.



(b) Buque portacontenedores con variación de calado.



(c) Buque portacontenedores con variación de manga.

**Figura 6.5.** Imágenes de distintas escalas del buque portacontenedores usadas para el entrenamiento.

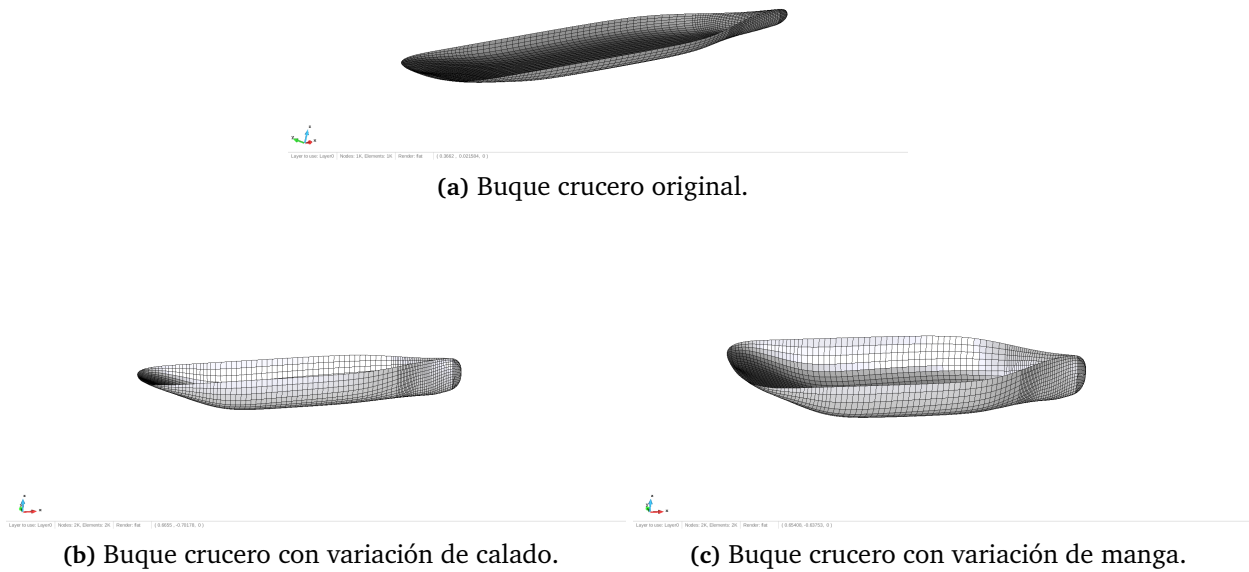


Figura 6.6. Imágenes de distintas escalas del buque crucero usadas para el entrenamiento.

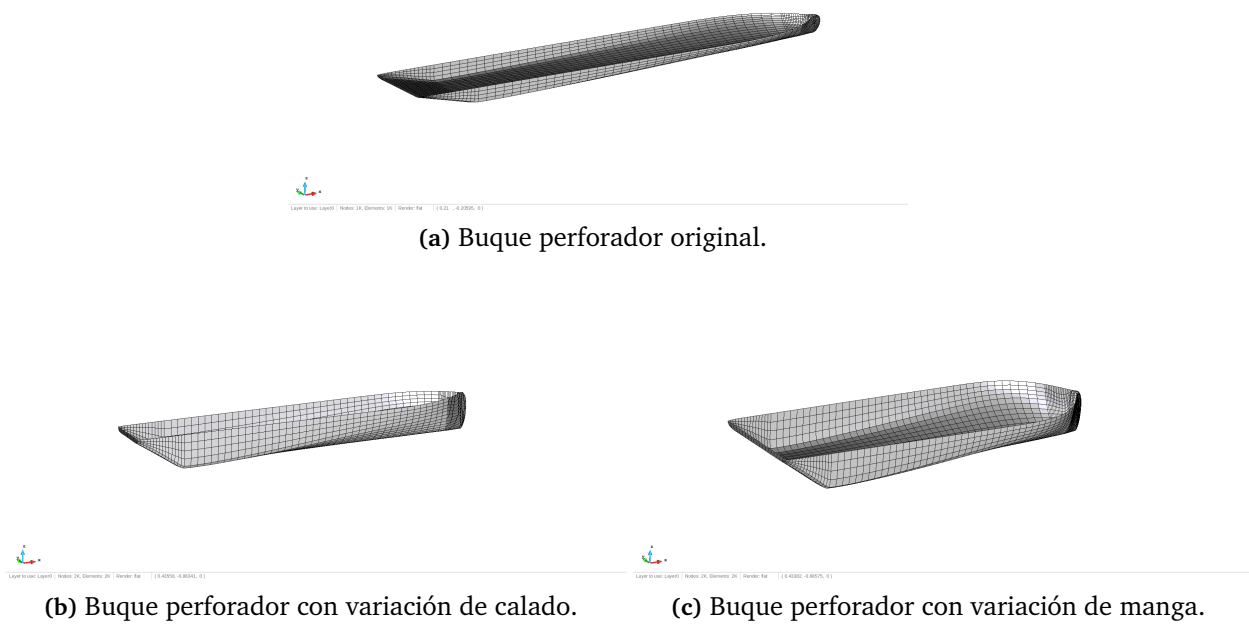
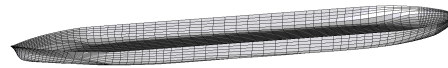
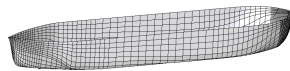


Figura 6.7. Imágenes de distintas escalas del buque perforador usadas para el entrenamiento.





(a) Buque FPSO original.



(b) Buque FPSO con variación de calado.



(c) Buque FPSO con variación de manga.

**Figura 6.8.** Imágenes de distintas escalas del buque FPSO usadas para el entrenamiento.



(a) Buque fragata original.

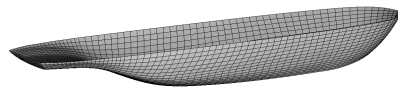


(b) Buque fragata con variación de calado.



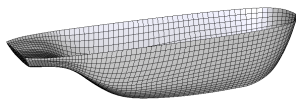
(c) Buque fragata con variación de manga.

**Figura 6.9.** Imágenes de distintas escalas del buque fragata usadas para el entrenamiento.



Leon Franco Lopez / Pablo H. Romero Tello / Javier Sáez | (1.2023) 4.4026 (3)

(a) Buque remolcador original.



Leon Franco Lopez / Pablo H. Romero Tello / Javier Sáez | (1.4023) 3.2024 (3)

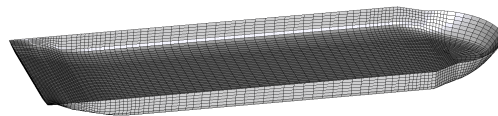
(b) Buque remolcador con variación de calado.



Leon Franco Lopez / Pablo H. Romero Tello / Javier Sáez | (1.2024) 3.2027 (3)

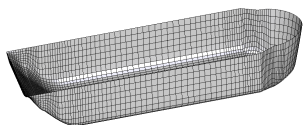
(c) Buque remolcador con variación de manga.

Figura 6.10. Imágenes de distintas escalas del buque remolcador usadas para el entrenamiento.



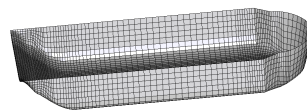
Leon Franco Lopez / Pablo H. Romero Tello / Javier Sáez | (1.2021) 4.2762 (3)

(a) Buque semisumergible original.



Leon Franco Lopez / Pablo H. Romero Tello / Javier Sáez | (1.2022) 4.3700 (3)

(b) Buque semisumergible con variación de calado.



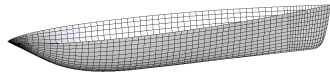
Leon Franco Lopez / Pablo H. Romero Tello / Javier Sáez | (1.2022) 4.3224 (3)

(c) Buque semisumergible con variación de manga.

Figura 6.11. Imágenes de distintas escalas del buque semisumergible usadas para el entrenamiento.



(a) Buque patrullero original.

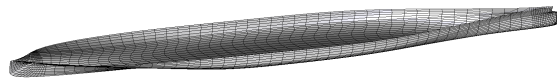


(b) Buque patrullero con variación de calado.

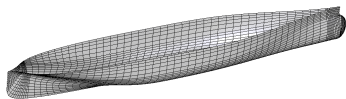


(c) Buque patrullero con variación de manga.

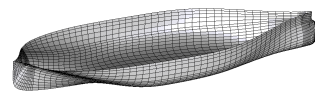
**Figura 6.12.** Imágenes de distintas escalas del buque patrullero usadas para el entrenamiento.



(a) Buque RORO original.



(b) Buque RORO con variación de calado.



(c) Buque RORO con variación de manga.

**Figura 6.13.** Imágenes de distintas escalas del buque RORO usadas para el entrenamiento.

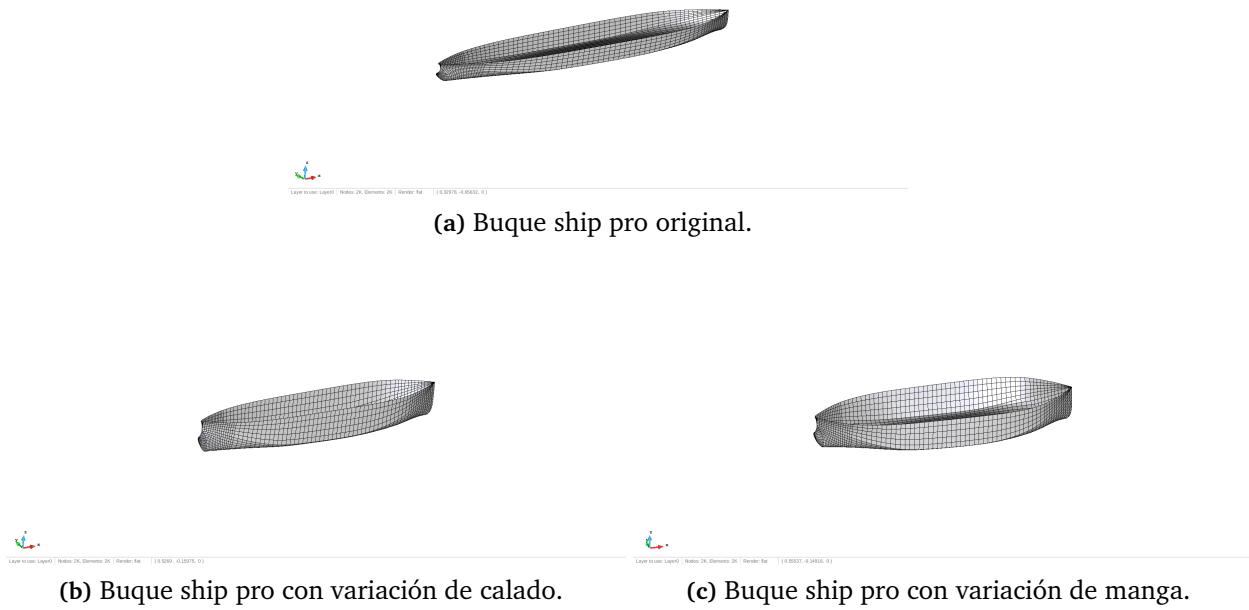


Figura 6.14. Imágenes de distintas escalas del buque ship pro usadas para el entrenamiento.

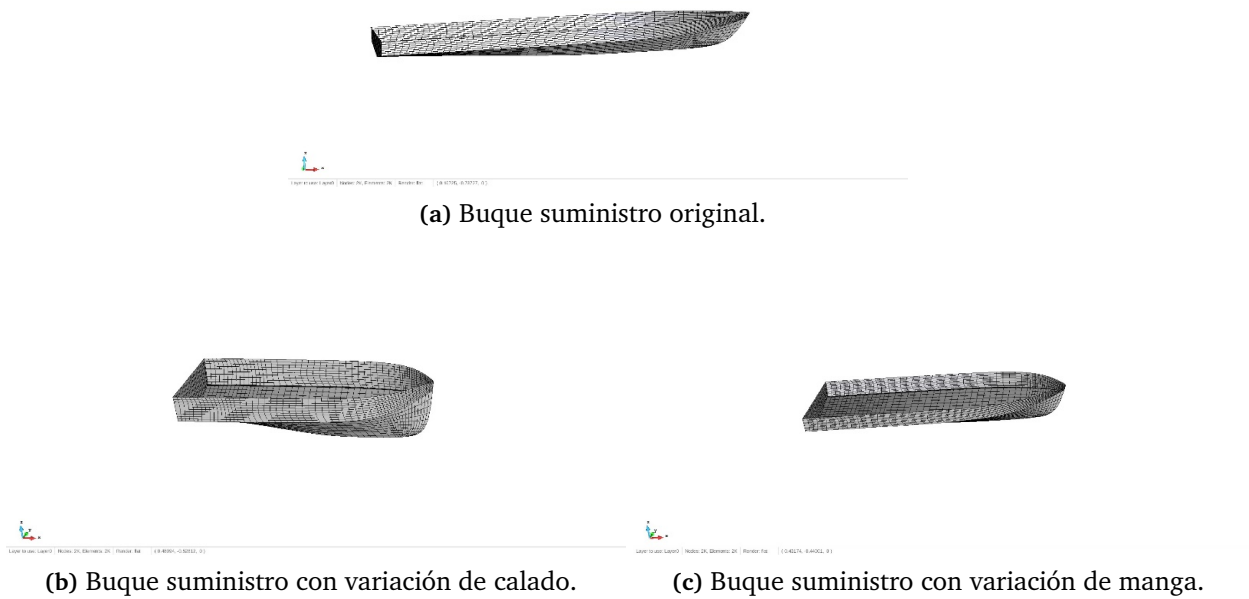
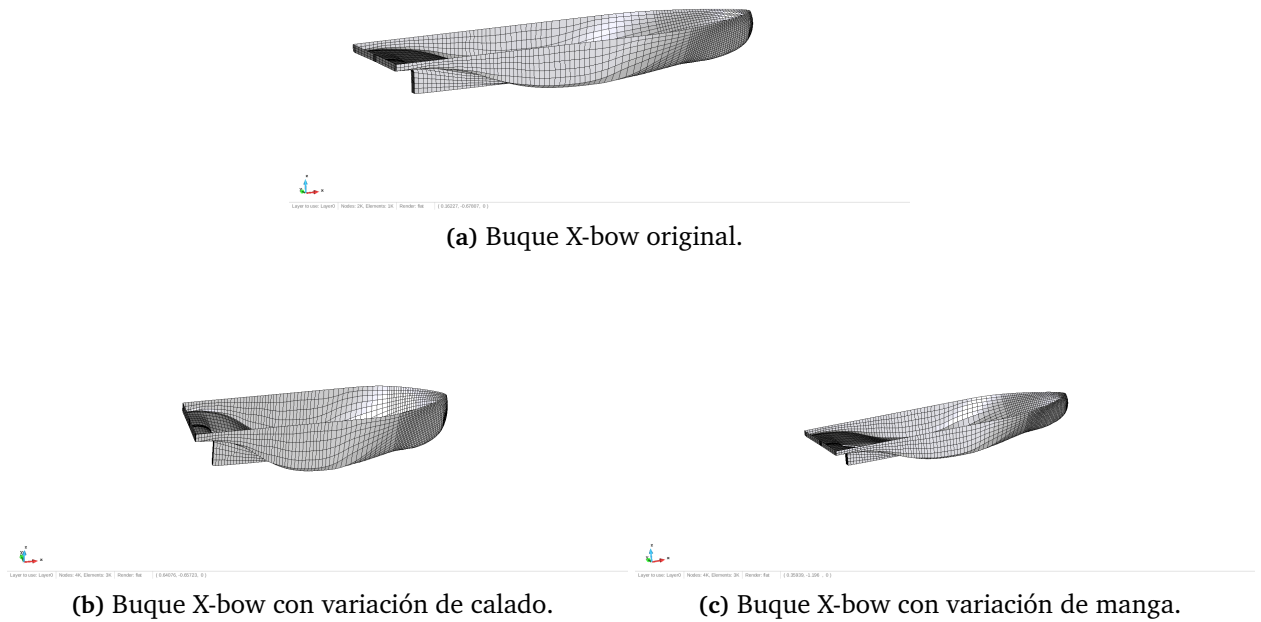


Figura 6.15. Imágenes de distintas escalas del buque suministro usadas para el entrenamiento.



**Figura 6.16.** Imágenes de distintas escalas del buque X-bow usadas para el entrenamiento.

Tras la generación del conjunto de buques a simular y realizar dicho cálculo, hay que procesar el conjunto de datos obtenidos, ya que siempre que se desea emplear herramientas de *Machine Learning*, se debe procesar la información que se le va a suministrar a las redes neuronales que se generarán para llevar a cabo el presente proyecto. De las simulaciones se obtienen matrices de masas añadidas, de amortiguamiento, fuerzas de difracción y RAOs.

En el caso de las matrices de masas añadidas y de amortiguamiento se obtiene una matriz de  $6 \times 6$  para cada periodo, es decir, que por cada buque de entrenamiento se obtienen 720 datos de masa añadida, y los mismos de amortiguamiento. En cuanto a los datos de fuerzas y RAOs, se genera una matriz de datos de  $20 \times 6$  por dirección, lo que supone un total de 840 datos tanto para las fuerzas como para las RAOs.

Dentro de las matrices de amortiguamiento y de masas añadidas, se han separado cuatro grupos de datos: el primero, los correspondientes a las traslaciones dentro de la diagonal principal, se puede ver en color verde; el segundo, los giros, que se encuentran en la diagonal principal, color azul; el tercero, consiste en los datos significativos relativos a los combinados, rojo; y el cuarto, en este se recogen todos los datos no significativos, identificados con un 0. En la siguiente matriz se muestran estos conjuntos de datos de forma esquemática de la matriz de masas añadidas, en el caso de la de amortiguamiento quedaría de manera idéntica.

$$\begin{pmatrix} A_{11} & 0 & 0 & 0 & A_{15} & 0 \\ & A_{22} & 0 & A_{24} & 0 & A_{26} \\ & & A_{33} & 0 & A_{35} & 0 \\ & & & A_{44} & 0 & 0 \\ & & & & A_{55} & 0 \\ & & & & & A_{66} \end{pmatrix}$$

En el caso de las fuerzas y las RAOs, se separan los valores por dirección, y también se separan los valores de traslación de los de giro, resultando por tanto 7 grupos de direcciones y por cada uno de ellos otros 2 grupos, es decir un total de 14 grupos de entrenamiento.

Este procesado de los datos resultantes de las simulaciones es el fruto de muchas horas de pruebas y entrenamientos, ya que se han probado muchas configuraciones hasta encontrar la que mejor resultados obtiene, que es la presentada con anterioridad.

Dentro del proceso de aprendizaje máquina que se ha seguido, una vez generado el conjunto de casos de entrenamiento, se procede a realizar el procesamiento y disposición de las entradas de la RNA, y una vez se consigan dichas entradas se generará la RNA y se podrá llevar a cabo el entrenamiento de dicha red.

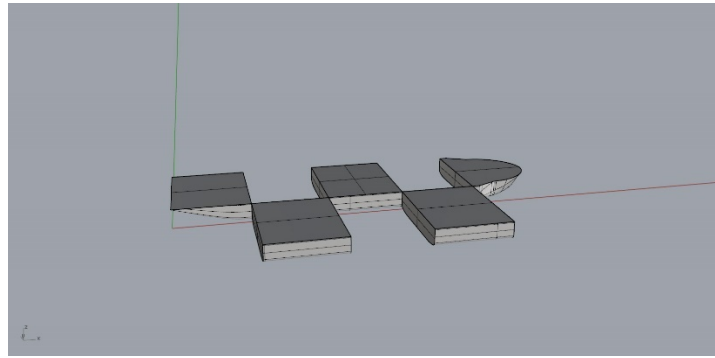
### 6.3. Caracterización de los casos de entrenamiento

Una vez realizadas las simulaciones mediante el anterior pseudocódigo ya se dispone de las salidas esperadas del conjunto de entrenamiento, más conocidos en terminología inglesa como *targets*, se tratará de buscar los parámetros que permitan a la RNA identificar las variaciones de formas de los distintos tipos de buques, y asociarles un valor determinado de los parámetros de comportamiento en la mar, es decir, se realizará la búsqueda de los parámetros de entrada para la RNA. Quizá este sea el paso más delicado del trabajo, ya que si los datos que se le suministrarán a la red no consiguen identificar las formas del buque, resultará imposible que la red consiga predecir nada con un mínimo de precisión.

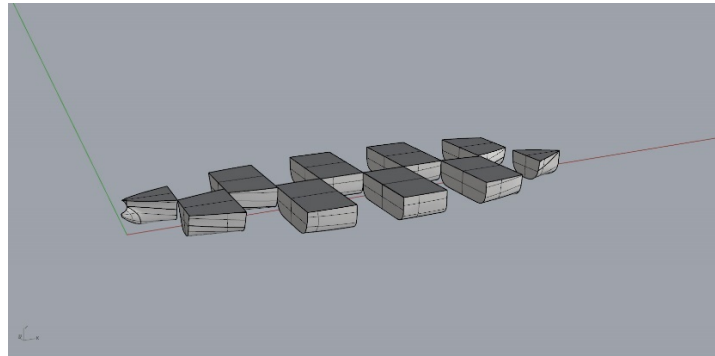
Como es sabido que una de las formas más empleadas a lo largo de la historia para definir las formas del casco es mediante la cartilla de trazado, pues define de forma exacta las coordenadas tridimensionales del mismo, sin embargo, si se intenta explorar esta vía de definición se observa que se necesita un número muy alto de variables, por lo que se desestima la idea de utilizar la cartilla de trazado. También se ha venido definiendo las formas del casco mediante el análisis de componentes principales de los distintos coeficientes adimensionales del buque, puede establecerse, a priori, como una solución plausible a la búsqueda de aquellos parámetros que mejor definan las formas del casco. Estos coeficientes son: el coeficiente de bloque, el prismático, o el de la flotación entre otros, que aportan alguna información sobre los volúmenes del casco o las características hidrodinámicas, pero resultan insuficientes para caracterizar las formas.

En este proyecto se han planteado dos métodos de caracterización de las formas totalmente diferenciados, el primero es mediante volúmenes seccionales y el segundo es mediante momentos de la curva de áreas seccionales.

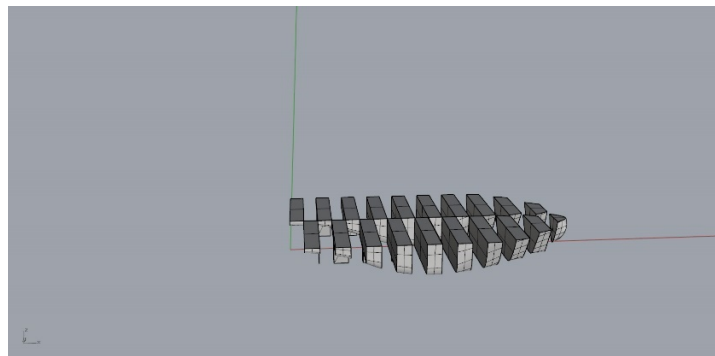
Para definir el casco mediante volúmenes seccionales, se han tomado 5, 10 y 20 volúmenes. Como se ha ido indicando a lo largo del presente proyecto, los buques están adimensionalizados en eslora, lo que quiere decir que todos los buques tienen una eslora de 1 m, por tanto, en el caso de 5 volúmenes seccionales, el buque se divide en bloques de 0,2 m de eslora y se calcula el volumen desplazado de cada bloque, haciendo lo mismo para las casuísticas de 10 y de 20 volúmenes. En la Figura 6.17 se muestran a modo de ejemplo gráfico como se dividiría cada buque para calcular los distintos volúmenes seccionales y generar las entradas de la red.



(a) Heave lift con 5 volúmenes seccionales.



(b) Ship pro con 10 volúmenes longitudinales.



(c) Xbow 20 con volúmenes longitudinales.

**Figura 6.17.** Imágenes con distintos volúmenes seccionales para distintos tipos de buques.

En cuanto al método de definición de las formas del casco mediante momentos de la curva de áreas seccionales, se toman momentos desde orden 0, es decir el volumen de carena total, hasta momentos de orden 5 en primer lugar y hasta momentos de orden 10 en segundo lugar. Los momentos se toman con respecto a popa, y básicamente para obtenerlos se multiplica el área de cada sección por la separación de dicha sección con respecto a la popa (0,0), elevada al orden del momento, puesto que la curva de areas es continua, se aproxima por trapecios.

A modo de resumen se presenta la Tabla 6.3, en la que se muestran los distintos métodos de caracterización de formas que se han estudiado durante la realización del proyecto.

**Tabla 6.3.** Condiciones consideraras para la definición de las formas del casco.

<b>Volúmenes</b>	5 volúmenes seccionales
	10 volúmenes seccionales
	20 volúmenes seccionales
<b>Momentos</b>	Orden 5
	Orden 10

## 6.4. Análisis de sensibilidad de la RNA

La secuencia de este análisis de sensibilidad consistirá, en seleccionar la combinación de arquitectura y algoritmo de entrenamiento óptima, junto con un análisis del número de capas y de neuronas de cálculo además del estudio del método de definición de las formas del casco.

Para la generación y el entrenamiento de la RNA se ha utilizado el software comercial Matlab (www.mathworks.com), con su *toolbox, neural network tool*. Se han seleccionado para la realización de las pruebas que atañen a la arquitectura y al algoritmo de entrenamiento, las arquitecturas: *feedforwardnet*, *cascaforwardnet* y *fitnet*, junto con los algoritmos de entrenamiento: descenso del gradiente (*gd*), descenso del gradiente con tasa de aprendizaje adaptativo (*gda*), descenso del gradiente con variación del momentum (*gdm*) y *Levenberg-Marquardt (lm)*. Para realizar este primer análisis, se utilizará como datos de entrada los 5 volúmenes seccionales, teniendo como objetivos la parte correspondiente a las traslaciones de la diagonal principal de la matriz de masas añadidas, es decir, las masas añadidas en largada, deriva y arfada.

**Tabla 6.4.** Análisis de sensibilidad de las distintas arquitecturas y optimizadores de la RNA.

Arquitectura	E	ITER	TR	Máx. ITER	Máx. F	Fin E	TE	ECM
<i>feedforwardnet</i>	gd	150	0,002	10 <sup>4</sup>	150	número de fallos	2"	56073,00
	gda	2583	0,002	10 <sup>4</sup>	100	número de fallos	43"	23,43
	gdm	10000	0,002	10 <sup>4</sup>	20	máx. iteraciones	2' 48"	22,23
	lm	340	0,002	10 <sup>4</sup>	20	número de fallos	3h 34' 04"	7,36
<i>cascaforwardnet</i>	gd	150	0,002	10 <sup>4</sup>	150	número de fallos	02"	6184,90
	gda	487	0,002	10 <sup>4</sup>	20	número de fallos	8' 00"	37,29
	gdm	10000	0,002	10 <sup>4</sup>	20	máx. iteraciones	3' 01"	18,25
	lm	213	0,002	10 <sup>4</sup>	20	número de fallos	3h 49' 48"	7,37
<i>fitnet</i>	gd	151	0,002	10 <sup>4</sup>	150	número de fallos	02"	2687,40
	gda	3314	0,002	10 <sup>4</sup>	100	número de fallos	53"	21,38
	gdm	10000	0,002	10 <sup>4</sup>	20	máx. iteraciones	2' 48"	18,94
	lm	243	0,002	10 <sup>4</sup>	20	número de fallos	3h 40' 34"	7,37

(\*) **E** = entrenamiento, **ITER** = número de iteraciones, **TR** = Ratio de aprendizaje, **Max. F** = número máximo de fallos, **Fin E** = Motivo de finalización del entrenamiento, **TE** = Tiempo de entrenamiento, **ECM** = Error Cuadrático Medio.

Observando la Tabla 6.4 se puede concluir que es mucho más influyente el algoritmo de entrenamiento usado que la arquitectura. Era esperable que pasara al comparar la arquitectura *fitnet* y la *feedforwardnet*, ya que recordando lo que se comentó en el **Capítulo 2, Machine**



**Learning**, básicamente son iguales, salvo que la arquitectura *feedforwardnet* también se puede utilizar para clasificación. El algoritmo *Levenberg-Marquardt (lm)*, es el que menor error consigue, del orden de 7,4. El siguiente mejor es el gradiente con variación del *momentum (gdm)* con un ECM algo menor de 20. Cabe comentar que este último algoritmo de entrenamiento finaliza por agotar el número total de iteraciones, por tanto no se sabe hasta donde bajaría el error si se dejara entrenar durante más iteraciones. Para comprobar la variación del error al aumentar el número de iteraciones a realizar, se decide comparar la combinatoria de *feedforwardnet - lm* y *cascaforwardnet - gdm*.

**Tabla 6.5.** Análisis de sensibilidad de las arquitecturas *cascaforwardnet* y *feedforwardnet*.

Arquitectura	E	ITER	TR	Máx. ITER	Máx. F	Fin E	TE	ECM
<i>feedforwardnet</i>	lm	255	0,002	10 <sup>4</sup>	50	número de fallos	2h 36' 22"	6,95
<i>cascaforwardnet</i>	gdm	10 <sup>6</sup>	0,002	10 <sup>6</sup>	20	máx. iteraciones	11h 19' 16"	15,69

(\*)E = entrenamiento, **ITER** = número de iteraciones, **TR** = Ratio de aprendizaje, **Max. F** = número máximo de fallos, **Fin E** = Motivo de finalización del entrenamiento, **TE** = Tiempo de entrenamiento, **ECM** = Error Cuadrático Medio.

De la comparativa realizada en la Tabla 6.5, se puede concluir con que para reducir el ECM utilizando el algoritmo de gradiente decreciente con variación del *momentum* se han de aumentar mucho los tiempos. También se debe comentar que pese a repetir el experimento de la utilización de la arquitectura y el algoritmo *feedforwardnet-lm*, cambian los resultados obtenidos debido a que los conjuntos de entrenamiento, test y validación se cogen de manera aleatoria del total de los datos. Aun así, y en base a los resultados obtenidos se decide seleccionar la arquitectura *feedforwardnet* con el algoritmo optimizador *Levenberg-Marquardt* para continuar con el análisis de sensibilidad. Se comprueba cómo afecta la ratio de aprendizaje al error, sin embargo, se observa que la ratio de aprendizaje no marca una tendencia relevante, ya que depende de otros factores como son la caracterización de las formas, el número de neuronas y de capas. Por tanto, se decide dejar el valor por defecto, 0,002.

A continuación, se decide analizar la combinatoria de arquitectura y algoritmo de entrenamiento seleccionados con los distintos métodos de identificación del buque, se continúa usando una red de una única capa oculta con 20 neuronas y se sigue con el objetivo de obtener las masas añadidas correspondientes a las traslaciones (véase la Tabla 6.6).

**Tabla 6.6.** Análisis de sensibilidad de los distintos métodos de definición del buque.

Método	ITER	Máx. F	Fin E	TE	ECM
5 Vol.	260	50	número de fallos	2h 19' 23"	6,35
10 Vol.	845	50	número de fallos	8h 16' 16"	6,58
20 Vol.	89	50	número de fallos	1h 02' 42"	7,71
5 M. sec.	56	50	número de fallos	30' 22"	17,42

(\*)**ITER** = número de iteraciones, **Max. F** = número máximo de fallos, **Fin E** = Motivo de finalización del entrenamiento, **TE** = Tiempo de entrenamiento, **ECM** = Error Cuadrático Medio.

De la Tabla 6.6, se pueden sacar dos conclusiones principales, el método de definición de formas del casco mediante momentos de la curva de áreas seccionales la RNA no lo identifica

bien, es por ello que no se realizó el experimento para 10 momentos. La segunda conclusión que se obtiene es que a mayor número de entradas, la RNA necesita mayor número de neuronas de cálculo y de capas, es decir, que para los métodos de 10 y 20 volúmenes se requiere un mayor número de neuronas. Por este último motivo, se decide comprobar los tres métodos de volúmenes con 3 capas ocultas con 20 neuronas cada una, siendo el objetivo en este caso predecir la RAO para la situación de largada, de deriva y de arfada. Los resultados pueden observarse en la Tabla 6.7.

**Tabla 6.7.** Análisis de sensibilidad de los distintos métodos de definición del buque.

Método	ITER	Máx. F	Fin E	TE	ECM
5 Vol.	1268	50	número de fallos	39h 25' 22"	$6,48 \cdot 10^{-5}$
10 Vol.	2619	50	número de fallos	78h 40' 42"	$4,11 \cdot 10^{-5}$
20 Vol.	453	50	número de fallos	21h 01' 36"	$8,67 \cdot 10^{-5}$

(\*)**ITER** = número de iteraciones, **Max. F** = número máximo de fallos, **Fin E** = Motivo de finalización del entrenamiento, **TE** = Tiempo de entrenamiento, **ECM** = Error Cuadrático Medio.

Como se observa tanto en la Tabla 6.6, como en la Tabla 6.7, el método de 20 volúmenes seccionales no mejora los resultados. Una explicación plausible es que al aumentar el número de entradas, se requieran más neuronas o capas para mejorar el error, o simplemente, que el aprendizaje resulte más difícil con este aumento de entradas.

Tras este análisis se decide trabajar con el método de 10 volúmenes longitudinales, ya que es el que menor ECM presenta. Una vez seleccionado el método de caracterización del buque, se procede a seleccionar el número de capas ocultas, así como el de neuronas por capa. Para ello se analizan redes con 2 y 3 capas ocultas con 20 y 30 neuronas en cada capa, es decir, se generan RNA con dos capas ocultas con 20 neuronas de cálculo por capa y con tres capas ocultas con 20 neuronas, y del mismo modo se procede para 30 neuronas por capa.

**Tabla 6.8.** Análisis de sensibilidad del número de neuronas y del número de capas ocultas.

Neuronas	ITER	Máx. F	Fin E	TE	ECM
20-20	1437	50	número de fallos	34h 16' 31"	$9,34 \cdot 10^{-5}$
30-30	2855	50	número de fallos	187h 12' 34"	$3,61 \cdot 10^{-5}$
20-20-20	2619	50	número de fallos	78h 40' 42"	$4,11 \cdot 10^{-5}$
30-30-30	2883	50	número de fallos	189h 25' 14"	$3,05 \cdot 10^{-5}$

(\*)**ITER** = número de iteraciones, **Max. F** = número máximo de fallos, **Fin E** = Motivo de finalización del entrenamiento, **TE** = Tiempo de entrenamiento, **ECM** = Error Cuadrático Medio.

Finalmente, y tras la realización de este análisis de sensibilidad, se decide, seleccionar la red Feedforwarnet con el optimizador Levenberg-Marquardt, junto con 3 capas ocultas de 30 neuronas por capa. La caracterización de las formas del buque se hará mediante 10 volúmenes longitudinales, ya que es la combinatoria con la que se obtiene un menor error.

# 7

## Resultados

---

El presente Capítulo expone los resultados obtenidos tras realizar todo lo expuesto en los anteriores capítulos, tanto de buques del grupo de validación como otros totalmente distintos, así como una discusión sobre los mismos.

---

En el transcurso de este Capítulo, se expondrán resultados pertenecientes a la parte de validación y test del entrenamiento, es decir, que se obtendrán los resultados de barcos que la red no ha visto nunca, pero serán buques de tipologías parecidas a las entrenadas. Para observar la precisión de estos resultados se realizará una comparación de los resultados obtenidos al aplicar la red con los botenidos con un resolovedor del comportamiento en la mar en el dominio de la frecuencia, basado en el método de paneles, *SeaBEM*. Los resultados que se muestran en este apartado son los pertenecientes a un portacontenedores, un perforador y un buque remolcador que no han sido usados para la fase de entrenamiento. Además, estos buques tienen formas totalmente diferentes, con lo que se puede comprobar la capacidad de generalización del algoritmo desarrollado basado en Inteligencia Artificial para la predicción del comportamiento en la mar. Se mostrarán resultados de RAO de traslaciones, y masas añadidas tanto de traslaciones como de giros.

En las Figuras 7.1, 7.2 y 7.3 se muestran de manera gráfica los resultados de las curvas RAOs de los movimientos de translación: largada (rojo) y arfada (azul), no se incluyen los resultados del movimiento de deriva por ser despreciable frente a los otros dos movimientos. Se comparan además los valores calculados mediante el Método de los Elementos de Contorno (*SeaBEM*) señalados con el símbolo del cuadrado y mediante redes neuronales artificiales (RNA) señalados con el símbolo de la x. Cabe destacar que estas curvas han sido obtenidas para un centro de gravedad de (0,0,0), es decir que es coincidente con el centro de referencia.

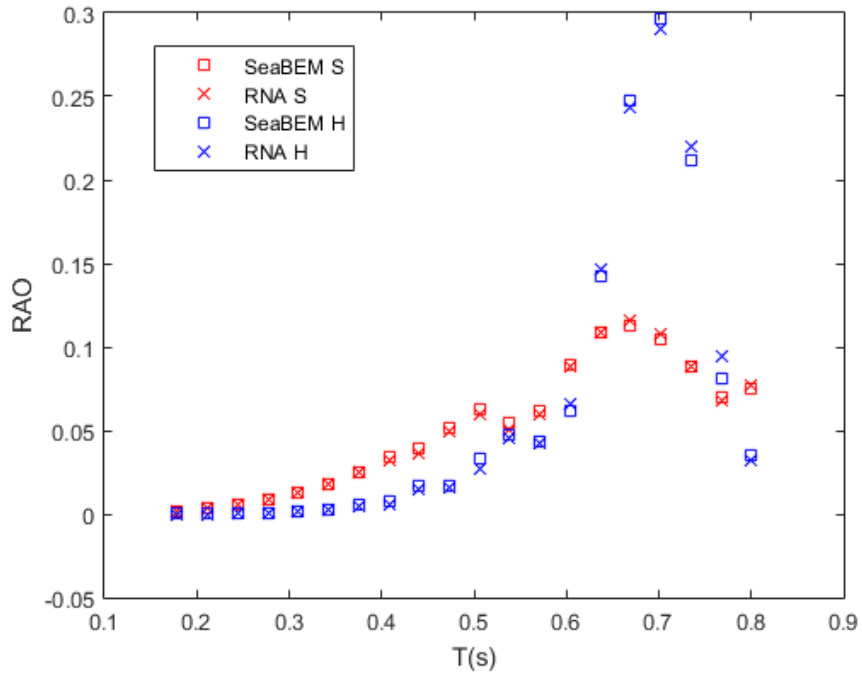


Figura 7.1. Comparativa de las curvas RAO para el buque perforador para las dos traslaciones.

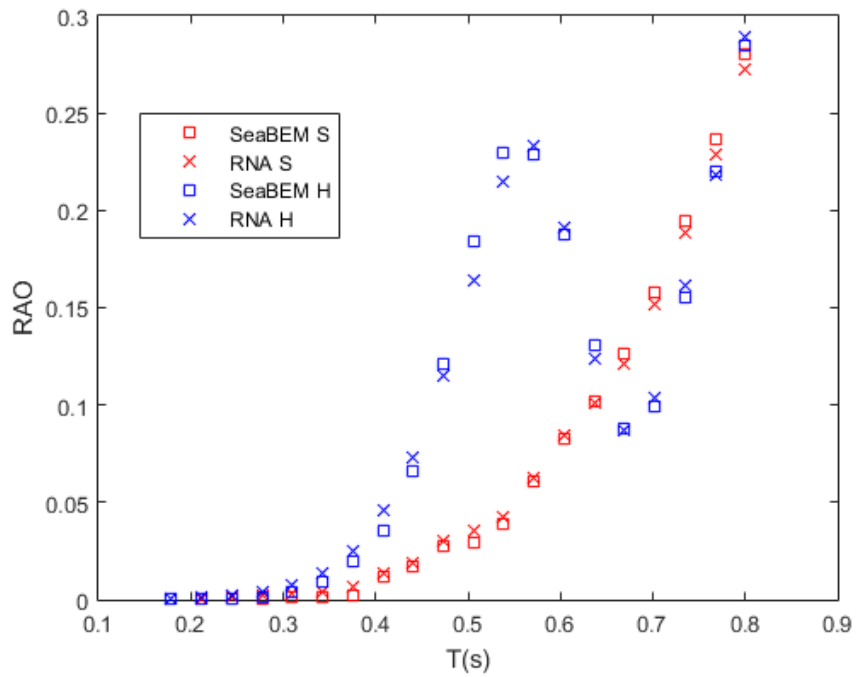
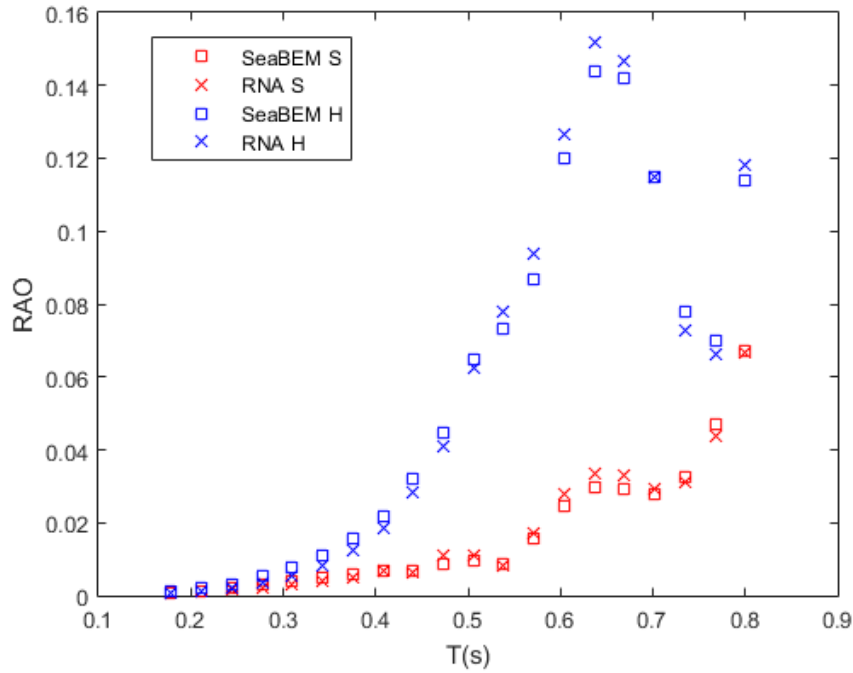
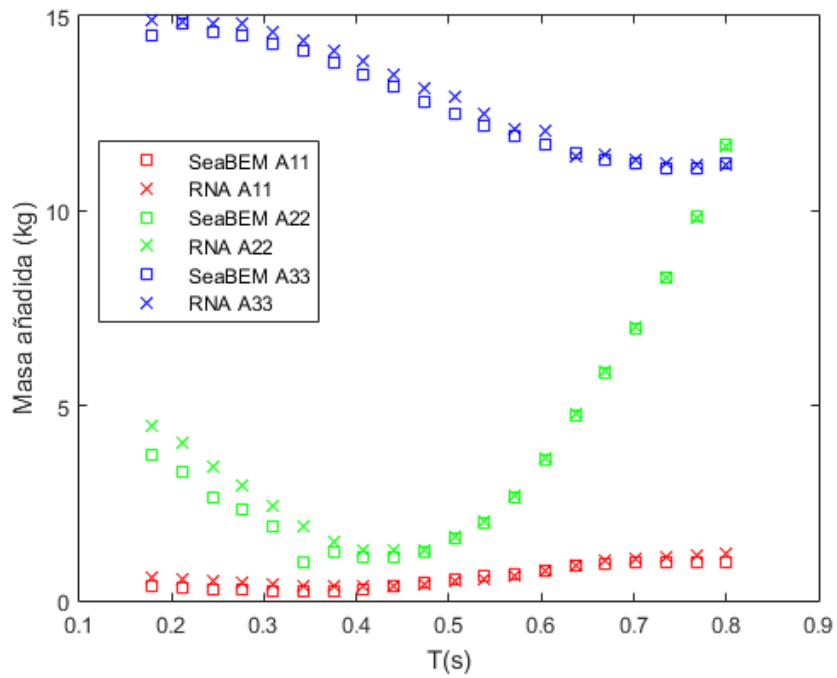


Figura 7.2. Comparativa de las curvas RAO para el buque portacontenedores para las dos traslaciones.

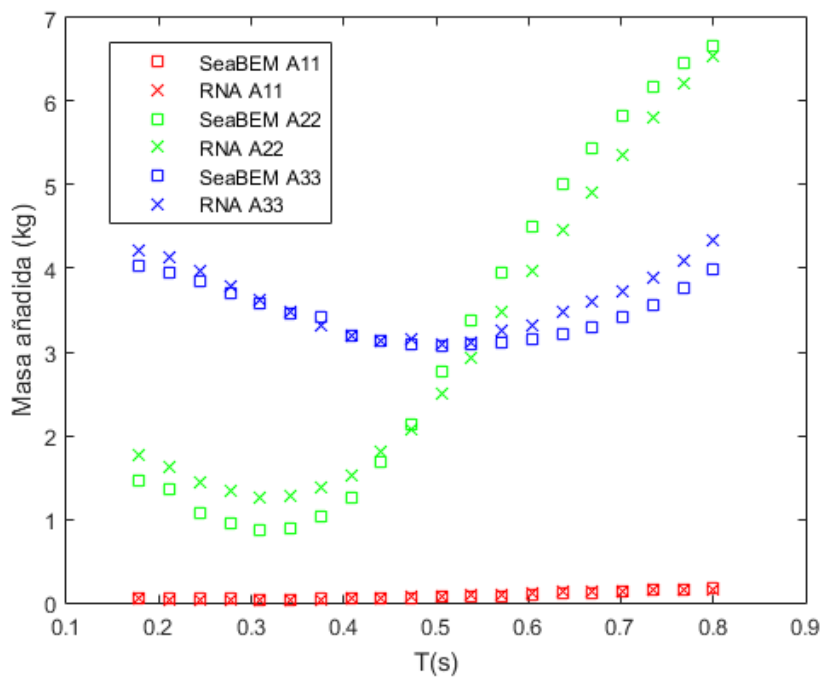


**Figura 7.3.** Comparativa de las curvas RAO para el buque remolcador para las dos traslaciones.

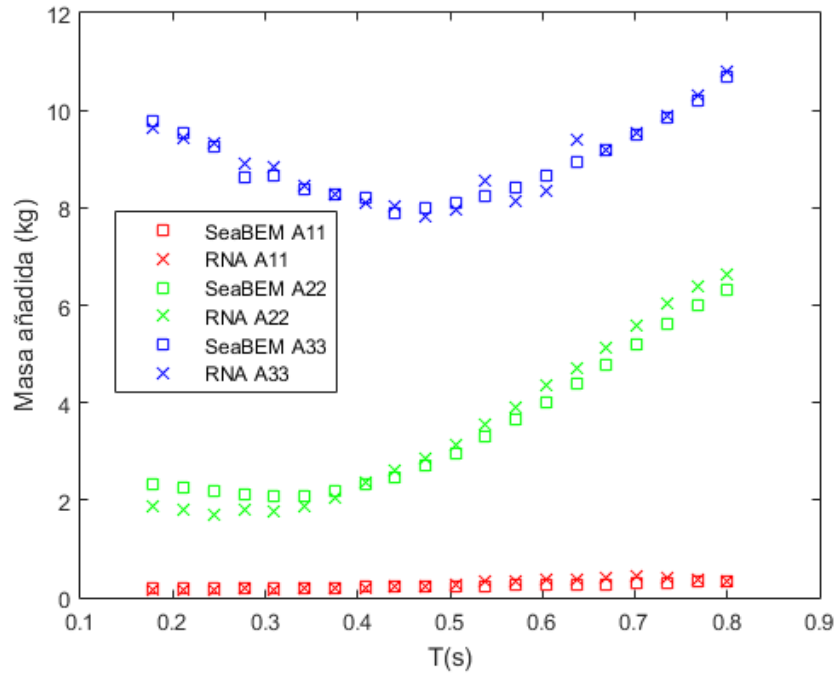
Además de las curvas RAOs, también se han representados de forma gráfica los valores de las masas añadidas, tanto de traslaciones como de giros, las de traslaciones se muestran en las Figuras 7.4, 7.5 y 7.6 (mostrando los datos de largada (rojo), deriva (verde) y arfada (azul)), y las masas añadidas de giro en las Figuras 7.7, 7.8 y 7.9, (en las que se representan (mostrando los datos de cabeceo (verde) y guiñada (azul), los datos de balance no se representan por no ser significativos). Del mismo modo que se hizo con las de RAO, se comparan los valores calculados con el Método de los Elementos de Contorno señalados con el símbolo del cuadrado y mediante redes neuronales artificiales (RNA) señalados con el símbolo de la x.



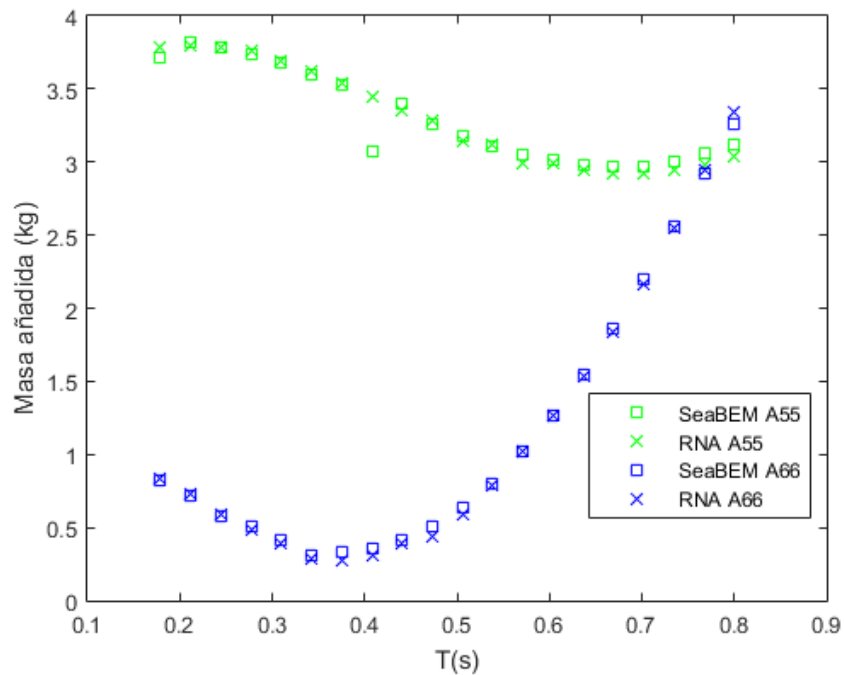
**Figura 7.4.** Comparativa de las curvas de masas añadidas de traslaciones para el buque perforador para las tres traslaciones.



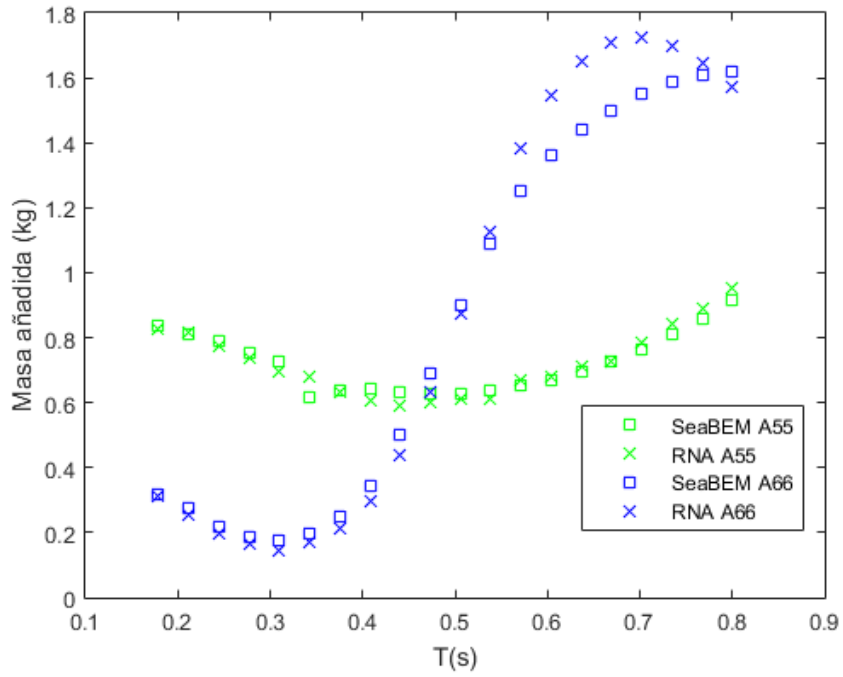
**Figura 7.5.** Comparativa de las curvas de masas añadidas de traslaciones para el buque portacontenedores para las tres traslaciones.



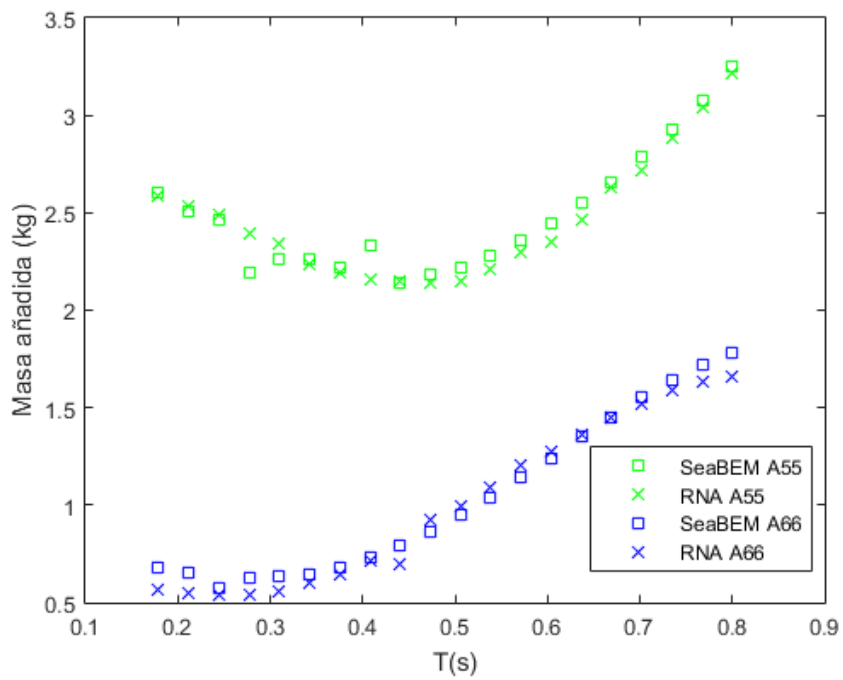
**Figura 7.6.** Comparativa de las curvas de masas añadidas de traslaciones para el buque remolcador para las tres traslaciones.



**Figura 7.7.** Comparativa de las curvas de masas añadidas de giros para el buque perforador para los dos giros.



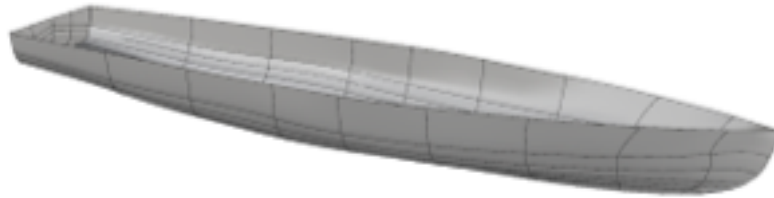
**Figura 7.8.** Comparativa de las curvas de masas añadidas de giros para el buque portacontenedores para los dos giros.



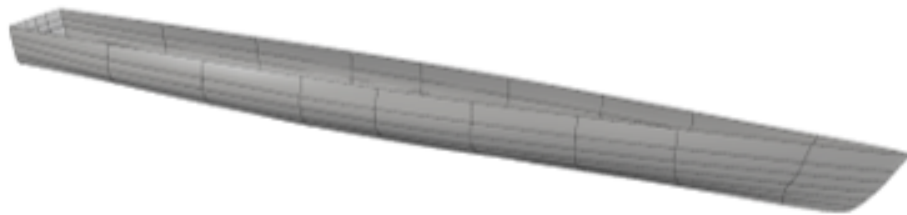
**Figura 7.9.** Comparativa de las curvas de masas añadidas de giros para el buque remolcador para los dos giros.



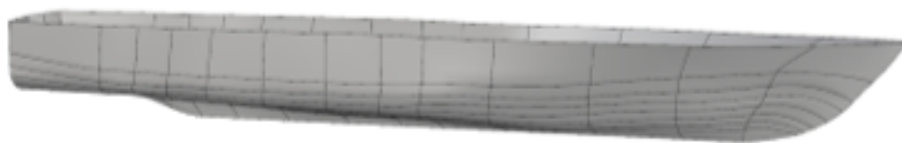
Aunque la red no haya sido entrenada con estos buques, cabe destacar que son del mismo tipo que con los que se ha entrenado, por ello a continuación se evaluará la red con buques totalmente distintos a los del conjunto de entrenamiento, más concretamente, se han seleccionado buques pertenecientes a las series sistemáticas *Nordstrom*, *Serie 64* y *SSPA*, veanse las Figuras 7.10, 7.11 y 7.12.



**Figura 7.10.** Casco del buque tipo Nordstrom.

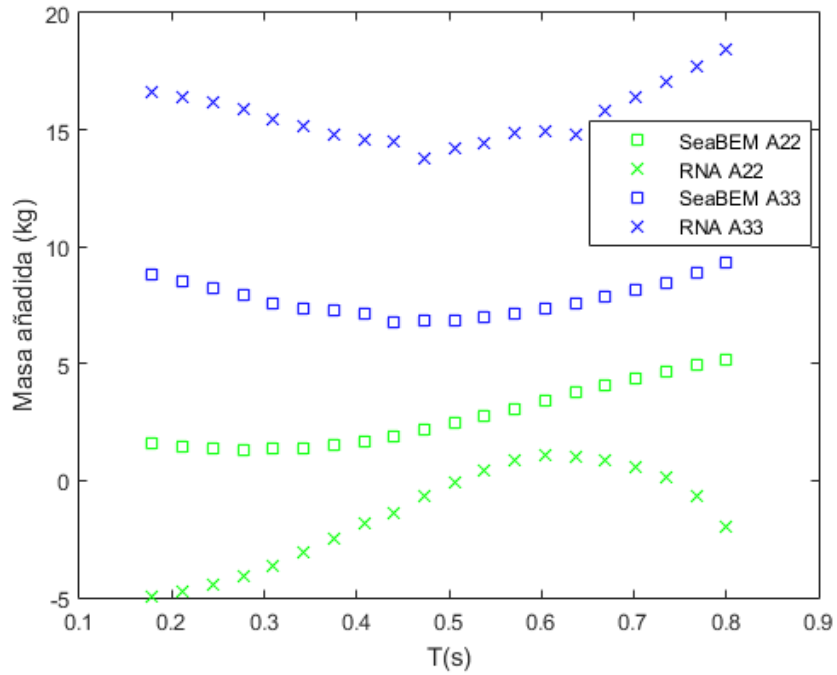


**Figura 7.11.** Casco del buque tipo Serie 64.

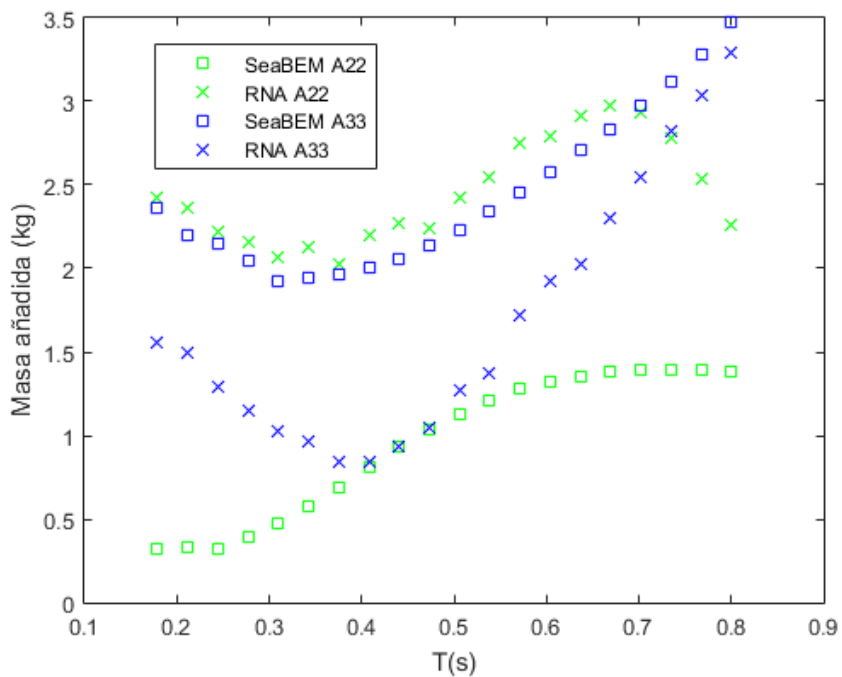


**Figura 7.12.** Casco del buque tipo SSPA.

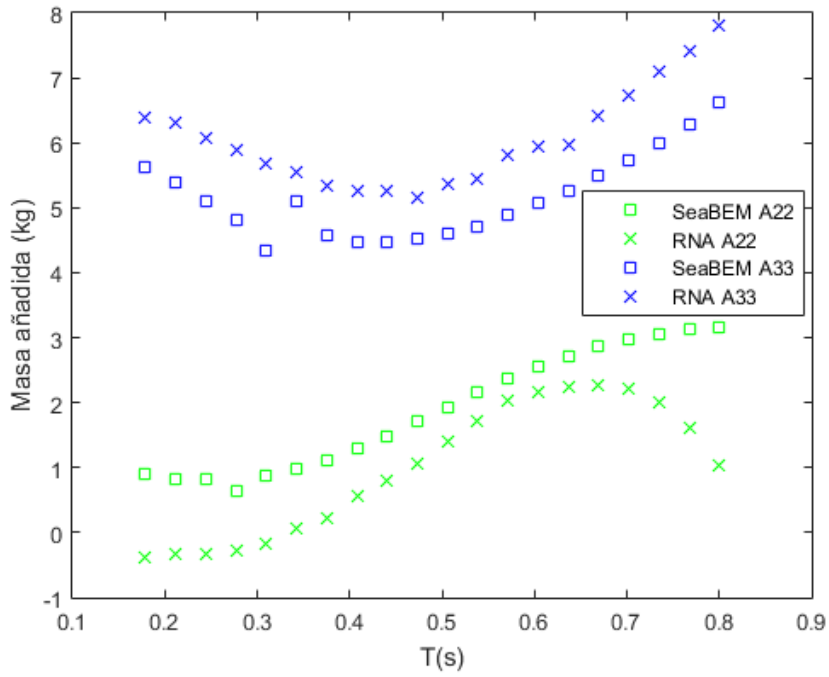
En las Figuras 7.13, 7.14 y 7.15, se muestran los resultados de masas añadidas de traslaciones (siendo los datos de deriva (verde) y arfada (azul)), no se muestran datos de largada por no ser significativos, y las masas añadidas de giro en las Figuras 7.16, 7.17 y 7.18, (en las que se representan (mostrando los datos de cabeceo (verde) y guiñada (azul), los datos de balance no se representan por no ser significativos).



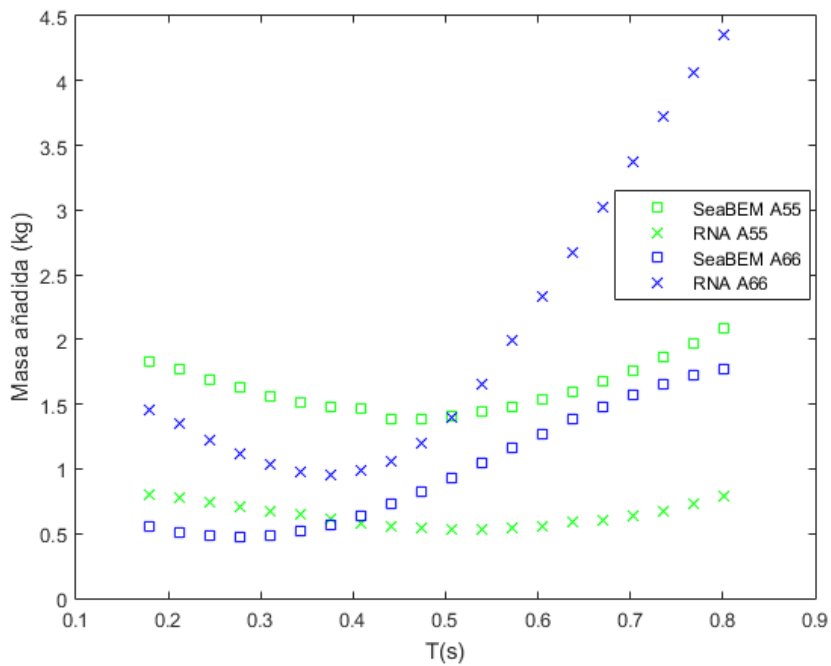
**Figura 7.13.** Comparativa de las curvas de masas añadidas de traslaciones para el buque tipo Nordstrom para las dos traslaciones.



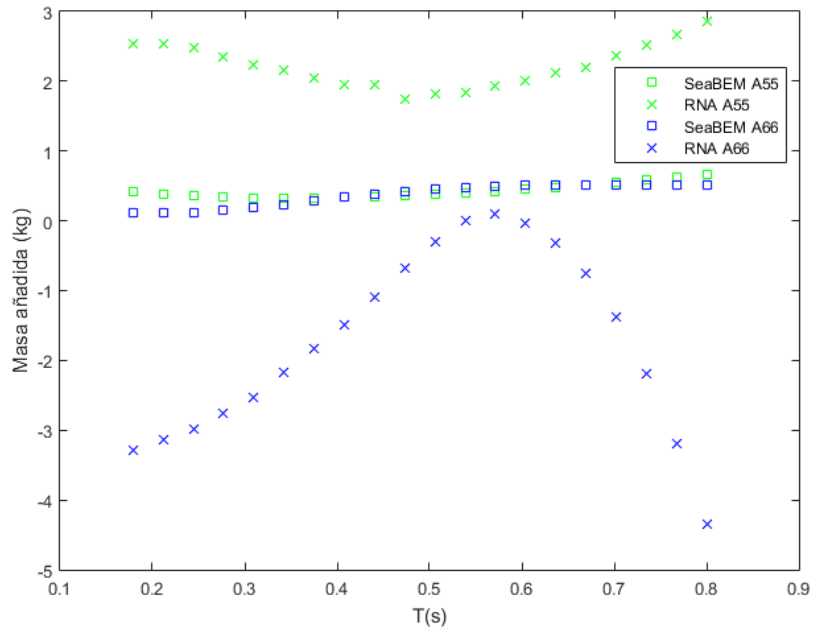
**Figura 7.14.** Comparativa de las curvas de masas añadidas de traslaciones para el buque tipo Serie 64 para las dos traslaciones.



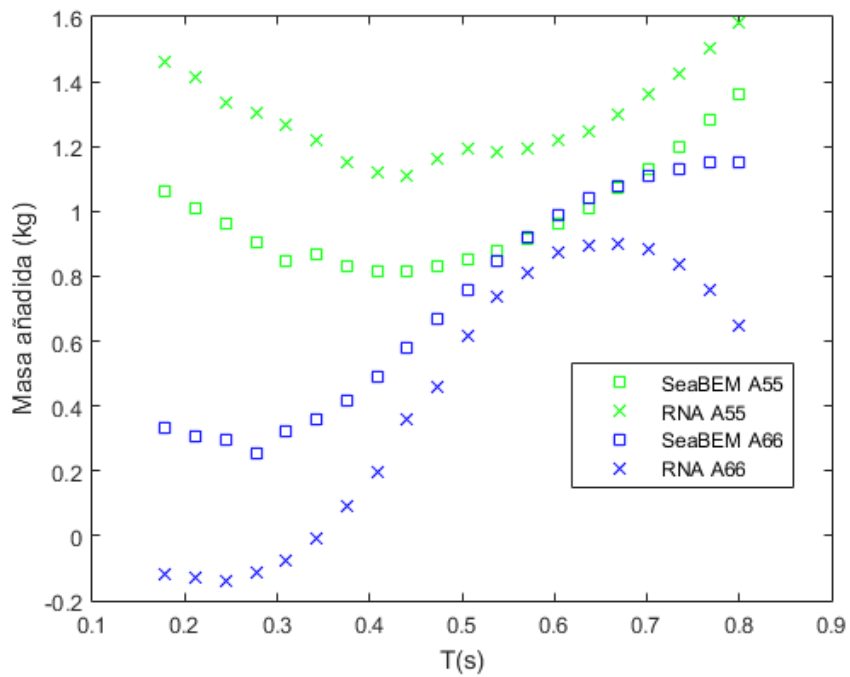
**Figura 7.15.** Comparativa de las curvas de masas añadidas de traslaciones para el buque tipo SSPA para las dos traslaciones.



**Figura 7.16.** Comparativa de las curvas de masas añadidas de giros para el buque tipo Nordstrom para los dos giros.



**Figura 7.17.** Comparativa de las curvas de masas añadidas de giros para el buque tipo Serie 64 para los dos giros.



**Figura 7.18.** Comparativa de las curvas de masas añadidas de giros para el buque tipo SSPA para los dos giros.

## 7.1. Discusión de resultados

Antes de comentar los resultados obtenidos, hay que destacar la dificultad añadida que supone el hecho de no tener una base de datos con resultados reales. Por tanto se tuvo que generar un conjunto de buques, y como ya se ha señalado a lo largo del trabajo, estos buques es el fruto de variaciones sistemáticas de escala, de este conjunto de buques se calculó su comportamiento mediante el *software* comercial *SeaBEM*. Para conocer el tiempo de procesamiento para la obtención de los buques con sus datos relativos al comportamiento, se ha de sumar el necesario para la obtención del conjunto, aproximadamente 2 semanas de computación, y el tiempo de cálculo, para esta parte se utilizaron 3 computadoras durante 2 meses. Una vez que se generó todo el conjunto de buques con sus resultados, se pasó a realizar las pruebas para la obtención de la red óptima, como ya se ha indicado, la combinatoria de arquitectura y algoritmo optimizador, método de definición de formas, y número de capas y de neuronas de cálculo por capa. Esta parte del trabajo ha sido la que más tiempo ha requerido, ya que, además de los tiempos mostrados en el **Capítulo 6, Metodología**, se han requerido multitud de pruebas previas. Aproximadamente, en esta fase del trabajo se requirió de 3 computadoras durante 5 meses, se está hablando de tiempos de entrenamientos superiores a las 300 horas, con la problemática que supone el tener computaciones tan extensas, ya que en todo ese tiempo puede haber complicaciones de toda índole, desde actualizaciones del sistema operativo a cortes de la corriente eléctrica.

Como ya se ha indicado, los resultados de los buques tipo perforador, remolcador y portacontenedores, pertenecen a buques que no se han utilizado dentro del conjunto de entrenamiento, se ve como se ajusta con un error bastante reducido lo predecido por la red a lo calculado mediante el resolvidor numérico. Si se analizan los resultados obtenidos de los buques tipo Nordstrom, Serie 64 y SSPA, se ve como el error es mucho más elevado, y en algunos casos la red no es capaz ni de al menos seguir la forma de la curva.

Quizá se ha producido el fenómeno del sobreajuste, este fenómeno ocurre cuando los datos de entrenamiento no son heterogéneos y la IA sólo es capaz de predecir datos similares a los suministrados, ya que los buques del mismo tipo que los pertenecientes al conjunto de entrenamiento si que la red es capaz de predecir bien, frente a los que son totalmente distintos, en los que no es capaz.

Cabe destacar que el tamaño de los buques pertenecientes a las series sistemáticas es muy diferente al de los buques pertenecientes al conjunto de entrenamiento, a modo de ejemplo, el desplazamiento es del 30 % que los de los buques de entrenamiento.

Pese a los resultados obtenidos en los buques de series sistemáticas, se puede afirmar que, la IA, más concretamente la RNA es capaz de predecir de forma bastante fidedigna los valores de las curvas RAO de distintos tipos de buque monocasco, pero habría que ampliar el conjunto de entrenamiento, incluyendo muchas más tipologías de cascos.



# 8

## Conclusiones

---

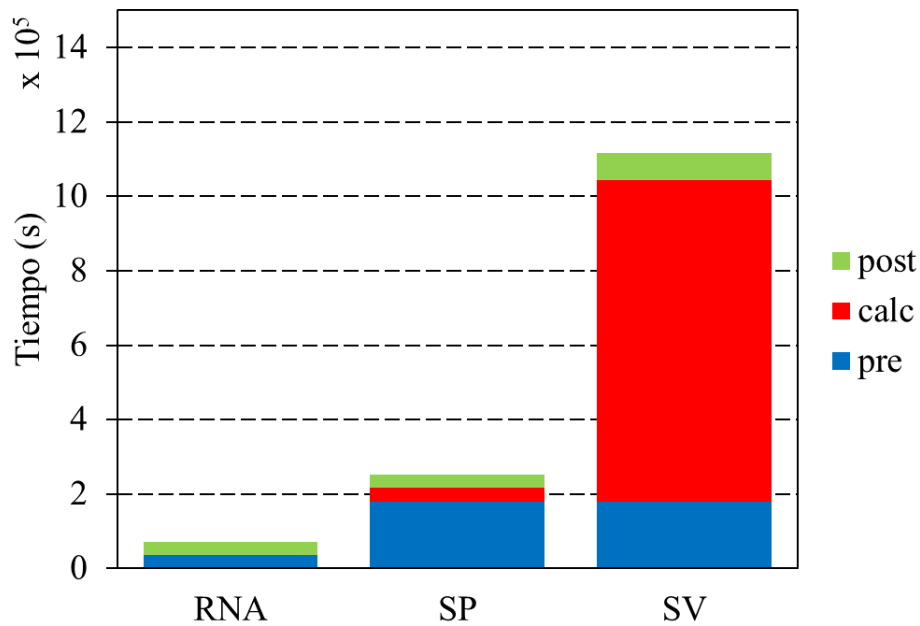
En el presente Capítulo se expone todo lo obtenido de este proyecto. Además, se ven las ventajas y las dificultades que supone enfrentarse a este nuevo método de resolución.

---

Es destacable el potencial que tienen las herramientas de Machine Learning, y más en concreto las RNA, en la ingeniería. Dentro de la Ingeniería Naval se están dando los primeros pasos en la aplicación de estas herramientas, pero es un camino complicado, ya que se tiene la dificultad añadida de lo costoso y a veces imposible que resulta conseguir un número de buques suficiente para aplicar estas técnicas, y que se tenga la información necesaria de dichos buques.

Se ha demostrado la habilidad de estos algoritmos para realizar tareas hasta ahora vedadas a metodologías clásicas con resultados similares. Ha de tenerse en cuenta el grado de incertidumbre ya existente en los datos de partida o entrenamiento, pues tanto los resultados numéricos como experimentales tiene una incertidumbre inherente a los mismos.

Una vez superado el obstáculo del procesamiento de los datos de entrenamiento, su clasificación y el entrenamiento de la IA, se advierte una enorme ventaja competitiva: el tiempo de cálculo requerido para llevar a cabo una estimación del comportamiento en la mar de un buque. En pocos segundos se pueden llevar a cabo cientos de predicciones con la consecuente ventaja competitiva en el proceso de diseño del buque. A modo comparativo la Figura 8.1 muestra una gráfica en el que se comparan los tiempos de cálculo de distintos resolvedores para un error relativo similar con respecto a valores experimentales.



**Figura 8.1.** Comparativa del tiempo de cálculo para el análisis del comportamiento en la mar de un buque entre Redes Neuronales Artificiales (RNA), programas CFD basados en teoría potencial (SP) y programas CFD basados en teoría viscosa (SV), para las distintas etapas del proceso de estimación: pre-procesado (pre), cálculo (calc) y post-procesado (post).

En este estudio se ha llevado a cabo la simulación de más de 12.000 casos de cálculo de comportamiento en la mar para proveer a una inteligencia artificial de datos suficientes para su entrenamiento. Se ha realizado el estudio de distintas arquitecturas de RNA con optimizadores de cálculo. También se han estudiado distintas formas de definición del buque para su reconocimiento por parte de la RNA. El tipo de arquitectura de red neuronal seleccionado es el Feedforwarnet con el optimizador Levenberg-Marquardt, junto con 3 capas ocultas de 30 neuronas que junto con el método de identificación de las formas del buque (10 volúmenes), tras un entrenamiento superior a 2000 horas, se ha demostrado que es capaz de predecir de forma adecuada y con un error ajustado el comportamiento en la mar, tanto en la predicción de las curvas RAO, como en las de masa añadida.

Con la realización proyecto se ha demostrado, por tanto, la validez de las RNA para predecir el comportamiento en la mar de buques monocasco, aun así, para conseguir una herramienta realmente útil, se ve la necesidad de generar un conjunto de datos muy superior, del orden de 10 veces superior al conjunto de datos usados en este estudio, incluyendo una amalgama muy superior de tipologías de buques, es decir, no se trata de aumentar el número de casos de entrenamiento mediante el aumento de escalas. Quizá sería conveniente disminuir el número de escalas de cada tipo de buque, y aumentar el número de tipologías, si bien es cierto que como ya se ha comentado, es muy complicado obtener datos relativos a esa cantidad de barcos.



# 9

## Futuros desarrollos

---

En el último Capítulo se expone el trabajo que podría realizarse en el futuro partiendo del ya desarrollado en el presente proyecto. Se hace un análisis tanto de las posibles mejoras y funcionalidades que se añadirían.

---

Tras la realización del presente proyecto, y continuando en la rama de predicción del comportamiento en la mar sin velocidad de avance, es decir, siguiendo estrictamente con la línea de investigación tratada en este documento. Cabe destacar que para conseguir un algoritmo de aplicación real se debe generar una cantidad de buques muy superior a la tratada en el presente proyecto, así como sus resultados relativos al comportamiento en la mar. Además, se ha de generar un algoritmo que sea capaz de extraer de una cartilla de trazado los volúmenes seccionales para conseguir aplicar directamente a la cartilla de trazado los algoritmos de IA entrenados. Es conveniente que, los resultados fácilmente obtenibles, como son: fuerzas de Froude-Krilov, matriz de restauración hidrostática y matriz de masas, se obtengan directamente de dicha cartilla de trazado sin necesidad de utilizar la IA.

También, se puede continuar esta línea de investigación aplicando herramientas de *Machine Learning* o *Deep Learning* al comportamiento en la mar con velocidad de avance, es decir, intentar obtener mediante estos algoritmos la resistencia añadida en olas, que hasta día de hoy se requieren costosísimas simulaciones para obtenerla. Se consigue así reducir sustancialmente los tiempos de cálculo, y por tanto se generaría la posibilidad diseñar basándose en el comportamiento en la mar. Este es un tema muy importante, ya que una parte de la energía generada por el motor propulsor se consume en vencer la resistencia al avance adicional que sufre el buque al navegar en un mar con un determinado oleaje.

Como ya se ha comentado, hoy en día no se diseña teniendo en cuenta algo tan importante como el comportamiento en la mar, ya sea con o sin velocidad de avance, lo que se viene haciendo es, tras terminar el diseño de un buque, comprobar que el comportamiento en la mar se encuentra dentro de lo asumible, y en el caso de no ser así modificar el diseño hasta que se consigue lo deseado. En esta línea se pretende conseguir algoritmos que ayuden al diseño original de la embarcación teniendo en cuenta desde un principio el comportamiento en la mar.



# 10

## Bibliografía

- [1] Raona. Los 10 Algoritmos esenciales en Machine Learning. (<http://www.raona.com>).
- [2] Sanz-Bobbi, M.A Data Mining. Comillas ICAI. Universidad Pontificia Comillas. 2019.
- [3] Muñoz J. L. Estudio de la predictibilidad del rendimiento académico de alumnos en asignaturas de segundo curso. Trabajo Final de Grado. Universidad Politécnica de Cartagena. 2015. (<http://hdl.handle.net/10317/4465>).
- [4] Ramos Soler, L. Uso de redes neuronales artificiales para la modelación del estado trófico de las masas de agua, aplicado al parque natural de la albufera de valencia. Trabajo Fin de Máster. Universidad Politécnica de Valencia. 2017. (<http://hdl.handle.net/10251/85329>).
- [5] Arbib, M.A. Handbook of Brain Theory and Neural Networks. MIT Press, 1995.
- [6] Minsky, M, Papert, S. Perceptrons. MIT Press. 1969.
- [7] Rumelhart, D.E. Hinton, G.E. Williams, R.J. Learning representations by back-propagating errors. Nature 1986, 323:533–536.
- [8] Larrañaga, P. Inza, I. Moujahid, A. Apuntes de curso. Tema 8. Redes Neuronales. Universidad del País Vasco. 2010.
- [9] Sanz-Bobbi, M.A. Redes Neuronales. Conceptos. Comillas ICAI. Universidad Pontificia Comillas. 2019.
- [10] Omaima, N. Ahmad, A.L. Cascade-Forward vs. Function Fitting Neural Network for Improving Image Quality and Learning Time in Image Compression System. Proceedings of the World Congress on Engineering 2012 Vol II WCE 2012, July 4 - 6, 2012, London, U.K.
- [11] Moctezuma J.C. Neural network toolbox de Matlab. Ciencias computacionales INAOE. Septiembre 2016.

- [12] Hinton, G. Srivastava, N. Swersky, K. Neural Networks for Machine Learning. Coursera. 2018. (<http://www.coursera.org>).
- [13] Sancho Peña, M. Diseño de Comportamientos Robóticos Mediante Deep Learning. Trabajo Final de Grado. Universidad de Alicante. 2017. (<http://hdl.handle.net/10045/71527>).
- [14] Sanz-Bobbi, M.A. Redes Neuronales. El Perceptrón Multicapa. Comillas ICAI. Universidad Pontificia Comillas. 2019.
- [15] Gougoudis, G. The Utilization of Artificial Neural Networks in Marine Applications: An Overview. American Society of Naval. Naval Engineers Journal. Vol. 120, N.3, pp. 19-26. 2008. (<https://doi.org/10.1111/j.1559-3584.2008.00150.x>)
- [16] Clausen, H.B., M. Lutzen, A. Friis-Hansen, and N Bjrneboe, Bayesian and neural networks for preliminar ship design. Marine Technology, Vol. 38, No. 4, pp. 268–277, 2001.
- [17] S. Mazaheri E. Mesbahi M. J. Downie A. Incecik Seakeeping Analysis of a turret - moored FPSO by using Artificial Neural Networks. ASME. 22nd International Conference on Offshore Mechanics and Arctic Engineering. 2003.
- [18] Yu D., Wangy L. Hull Form Optimization with Principal Component Analysis and Deep Neural Network. ArXiv, abs/1810.11701. 2018.
- [19] Margari V., Kanellopoulou A., Zaraphonitis G. On the use of Artificial Neural Networks for the calm water resistance prediction of MARAD Systematic Series' hullforms. Ocean Engineering 165:528-537.
- [20] Im, NK., Nguyen VS. Artificial neural network controller for automatic ship berthing using head-up coordinate system. International Journal of Naval Architecture and Ocean Engineering. Vol. 10, No. 3, pp. 235-249. 2018. (<https://doi.org/10.1016/j.ijnaoe.2017.08.003>)
- [21] Sclavounos P., Ma Y. Artificial Intelligence Machine Learning in Marine Hydrodynamics. ASME. International Conference on Offshore Mechanics and Arctic Engineering. OMAE 2018-77599. Madrid, España. 2018.
- [22] Li J., Yang, S., Pang, S. Free Roll Decay Modeling of a Trimaran Based on Least Square Support Vector Machine. OCEANS 2012 MTS/IEEE: Harnessing the Power of the Ocean. 2012. (<https://doi.org/10.1109/OCEANS.2012.6404855>).
- [23] Li D. Wilson P.A. Guan Y. Zhao X. An effective approximation modelling method for ship resistance in multidisciplinary ship design optimization. ASME. 33rd International Conference on Ocean, Offshore and Arctic Engineering. San Francisco, California, USA. 10.1115/OMAE2014-23407. 2014.
- [24] Gutiérrez-Romero, J.E. Apuntes de curso. Tema 3. Dinámica lineal de plataformas y artefactos. Universidad Politécnica de Cartagena. 2019.

- [25] Seván-Camas, B. A time-domain finite element method for seakeeping and wave resistance problems. Tesis Doctoral. Universidad Politécnica de Madrid, 2016. (<https://doi.org/10.20868/UPM.thesis.39794>).
- [26] Günther, C., Eike, L. y Carsten, Ö. Offshore Structures. Volume 1. Conceptual Design and Hydromechanics. 2012.
- [27] Gutiérrez-Romero, J.E. Apuntes de curso. Tema 4. Cargas en estructuras offshore. Universidad Politécnica de Cartagena. 2019.
- [28] SeaFEM Theory Manual. Compass. (<https://www.compassis.com>).
- [29] Delhommeau, G. Numerical Simulation of Hydrodynamics: Ships and Offshore Structures. Seakeeping codes AQUADYN and AQUAPLUS. Ecole Centrale de Nantes.
- [30] Kumar-Chakrabarti, S. Offshore structure modeling. Advanced series on Ocean Engineering. 1994.





