



**UNIVERSIDAD MICHOACANA
DE SAN NICOLÁS DE HIDALGO**



FACULTAD DE INGENIERÍA MECÁNICA

**DIVISIÓN DE ESTUDIOS DE POSGRADO EN
INGENIERÍA MECÁNICA**

**APRENDIZAJE PROFUNDO APLICADO AL
DISEÑO ÓPTIMO MULTICRITERIO DE
ÁLABES DE TURBINA DE VIENTO**

TESIS

QUE PARA OBTENER EL GRADO DE:

DOCTOR EN CIENCIAS EN INGENIERÍA MECÁNICA

PRESENTA:

M.C. JOSUÉ JORDI ZAVALA MORALES

DIRECTOR DE TESIS:

DR. ERASMO CADENAS CALDERÓN

CO-DIRECTOR:

DR. RAFAEL CAMPOS AMEZCUA



Morelia Michoacán; Julio de 2024

AGRADECIMIENTOS

Primeramente, a mis asesores los Doctores Erasmo Cadenas Calderón y Rafael Campos Amezcua por la disposición, asesorías y ayuda brindada en este camino, por creer en mi persona desde el comienzo de este difícil recorrido y lograr una marca total en mi vida profesional.

Gracias Universidad Michoacana de San Nicolás de Hidalgo (UMSNH) por acogerme como mi segunda casa, agradezco a mis profesores de la División del Posgrado de la Facultad de Ingeniería Mecánica, por ser académicos con un gran compromiso por la educación. Las palabras para demostrar mi sentimiento no existen; por hacer mis metas y sueños posibles, forjarme como una profesional de bien. Gracias a esta gran Institución por su calidad y humanidad por ser mi hogar durante algunos años, siempre le deberé lo que soy profesionalmente. Agradezco al Consejo Nacional de Humanidades Ciencias y Tecnologías (CONAHCYT) por la beca (No 893842) para realizar mis estudios de Doctorado.

Agradezco a los miembros de mi comité sinodal, los Doctores Gilberto González Avalos, Miguel Villagómez y Gildardo Solorio Días, reconociendo su colaboración en mi trabajo de tesis para obtener el grado de Doctor, desde el inicio del proceso y hasta el día de hoy. Gracias por la ayuda en mi formación profesional y humana, por sus comentarios y revisiones, durante el periodo de estudios.

Finalmente, pero no por ello menos importante agradezco a Michelle, familiares, amigos y compañeros por el apoyo durante este difícil camino.

Resumen

En este trabajo se presenta el desarrollo de un modelo de aprendizaje profundo (Deep Learning, DL) usando el diseño óptimo multicriterio de las aspas de un aerogenerador, centrándose en las variables de la velocidad específica y el esfuerzo de Von Mises para predecir: masa del aspa, el coeficiente de potencia y la frecuencia natural. El modelo DL se entrenó con datos que fueron generados a partir de tres funciones matemáticas, las cuales fueron construidas con la técnica de la distancia inversa ponderada. El modelo permite generar múltiples diseños factibles que satisfacen las restricciones de diseño planteadas. La teoría BEM se utilizó en la generación del modelo aerodinámico usando un perfil aerodinámico NRELS818 y un aerogenerador con una potencia de 12.5 kW. Se realizaron simulaciones, con el objetivo de analizar el comportamiento estructural de las aspas, para esto se usaron tres casos de cargas simplificadas de diseño especificadas en la norma IEC 61400-2. Los resultados mostraron que en la fase de entrenamiento, las métricas de error MAE, MSE y MSR fueron una guía esencial en el desarrollo del modelo DL. Se observaron comportamientos interesantes debido a los diversos resultados obtenidos, que probablemente se deban a los ajustes multidimensionales difíciles de observar en las funciones de ajuste generadas. Sin embargo, el entrenamiento entre 200 y 250 épocas obtuvo mejores resultados, con errores que oscilaron entre 0.050 y 0.120 para el MAE. El modelo DL mostró una notable capacidad para predecir las variables de salida óptimas, con una precisión que osciló entre el 90% y el 98%.

Palabras clave: aprendizaje profundo, aerogeneradores, diseño óptimo, función objetivo, diseño multicriterio.

ABSTRACT

This work presents the development of a deep learning (DL) model using the multi-criteria optimal design of wind turbine blades, focusing on the variables of tip speed ratio and Von Mises stress to predict: blade mass, power coefficient and natural frequency. The DL model was trained with data that were generated from three mathematical functions, which were constructed with the weighted inverse distance technique. The model allows the generation of multiple feasible designs that satisfy the design constraints. The BEM theory was used in the generation of the aerodynamic model using an NRELS818 airfoil and a wind turbine with a power of 12.5 kW. Simulations were performed, with the objective of analyzing the structural behavior of the blades, using three cases of simplified design loads specified in the IEC 61400-2 standard. The results showed that in the training phase, the MAE, MSE and MSR error metrics were an essential guide in the development of the DL model. Interesting behaviors were observed due to the diverse results obtained, which are probably due to the difficult-to-observe multidimensional fits in the fit functions generated. However, training between 200 and 250 epochs obtained better results, with errors ranging from 0.050 to 0.120 for the MAE. The DL model showed a remarkable ability to predict the optimal output variables, with accuracy ranging from 90% to 98%.

Keywords: deep learning, wind turbines, optimal design, objective function, multi-criteria design.

Índice general

| | |
|--|-----------|
| AGRADECIMIENTOS..... | II |
| RESUMEN | III |
| ABSTRACT | IV |
| 1 INTRODUCCIÓN | 1 |
| 1.1 ANTECEDENTES | 2 |
| 1.2. OBJETIVOS..... | 5 |
| 1.3. JUSTIFICACIÓN | 5 |
| 1.4. HIPÓTESIS..... | 6 |
| 1.5. CONTRIBUCIÓN | 6 |
| 2 DISEÑO DE TURBINAS DE VIENTO | 7 |
| 2.1. LAS TURBINAS DE VIENTO A TRAVÉS DEL TIEMPO | 7 |
| 2.1.1. <i>El viento</i> | 7 |
| 2.2. LAS PRIMERAS MÁQUINAS DE VIENTO | 9 |
| 2.3. DISEÑO AERODINÁMICO DE LAS ASPAS DE UNA TURBINA DE VIENTO | 13 |
| 2.3.1. <i>Método de momento de elemento de pala (BEM)</i> | 14 |
| 2.4 ANÁLISIS FEM | 20 |
| 3 DISEÑO ÓPTIMO | 26 |
| 3.1 FORMULACIÓN DEL PROBLEMA DE DISEÑO ÓPTIMO..... | 26 |
| 3.2. DISEÑO ÓPTIMO CONTRA DISEÑO CONVENCIONAL | 27 |
| 3.3. DISEÑO ÓPTIMO DEL ASPA | 29 |
| 3.3.1. <i>Variables de diseño</i> | 30 |
| 3.3.2. <i>Restricciones</i> | 30 |
| 3.4. TÉCNICA DE LA DISTANCIA INVERSA PONDERADA | 31 |
| 3.5. FUNCIONES OBJETIVO | 33 |
| 4 INTELIGENCIA ARTIFICIAL..... | 36 |
| 4.1. APRENDIZAJE AUTOMÁTICO | 37 |
| | 38 |
| 4.2. REDES NEURONALES ARTIFICIALES..... | 39 |
| 4.3. APRENDIZAJE PROFUNDO (DEEP LEARNING) | 42 |
| 4.4. CARACTERÍSTICAS CLAVE DEL DEEP LEARNING | 42 |
| 4.4.1. <i>Limitaciones del deep learning</i> | 43 |
| 4.5. MÉTRICAS DE EVALUACIÓN | 43 |
| 4.5.1 <i>Error cuadrático medio</i> | 44 |
| 4.5.2 <i>Error Absoluto Medio</i> | 44 |
| 4.6 MODELO DE REDES NEURONALES ARTIFICIALES | 44 |
| 4.6.1. <i>Neuronas</i> | 45 |
| 4.7. FUNCIONES DE ACTIVACIÓN | 47 |
| 4.7.1 <i>Función de activación lineal</i> | 47 |
| | 48 |
| 4.7.2. <i>Función escalón</i> | 48 |
| 4.7.3. <i>Función sigmoideal</i> | 49 |
| 4.7.4. <i>Función de activación lineal rectificada</i> | 50 |
| 4.8. ARQUITECTURA DE LAS REDES NEURONALES | 51 |
| 4.8.1. <i>Redes simples, con una única capa</i> | 51 |

| | |
|--|-----------|
| | 51 |
| 4.8.2. <i>Redes multicapa, con una capa oculta</i> | 51 |
| 4.8.3. <i>Redes profundas, con varias capas ocultas</i> | 52 |
| 5 DEEP LEARNING APLICADO AL DISEÑO ÓPTIMO DE LAS ASPAS DE UNA TURBINA DE VIENTO | 53 |
| 5.1. BASE DE DATOS GENERADA A PARTIR DE FUNCIONES OBJETIVO | 53 |
| 5.2. DISEÑO ÓPTIMO DEL MODELO DL..... | 54 |
| 5.3. IMPLEMENTACIÓN DEL MODELO DE APRENDIZAJE PROFUNDO | 54 |
| 5.4 PROGRAMA DEL MODELO DE DEEP LEARNING. | 56 |
| 5.5. FUNCIONAMIENTO DEL MODELO DE DL..... | 60 |
| 6 ANÁLISIS DE RESULTADOS | 65 |
| 7 CONCLUSIONES | 68 |
| GLOSARIO | 70 |
| ANEXO: PUBLICACIONES Y APORTES | 71 |
| ANEXO: DOCUMENTOS PARA DIGITALIZACIÓN DGB | 73 |
| | 74 |
| | 75 |
| REFERENCIA BIBLIOGRÁFICA | 76 |

Índice de figuras

| | |
|--|----|
| FIG. 1. CAÍDA DE ICARO (FITZSIMONS, S. (2009) POMPEII WALL ART, POMPEII WALL ART. DISPONIBLE EN: HTTP://LOOKINGGLASSCARUS.BLOGSPOT.COM/2009/06/POMPEII-WALL-ART.HTML | 7 |
| FIG. 2. DISEÑO DE ORNITÓPTERO. BRITANNICA, T. EDITORS OF ENCYCLOPEDIA BRITANNICA ORNITHOPTER. ... | 8 |
| FIG. 3. LOS MOLINOS DE VIENTO DE NASHTIFAN (SURFIRAN EDITORIAL TEAM (ED.) (2017) 1000-YEAR-OLD WINDMILLS STILL IN USE TODAY IN IRAN, SURFIRAN MAGAZINE. | 9 |
| FIG. 4. MOLINOS DE VIENTO EN ESPAÑA. (BRITANNICA, T. EDITORS OF ENCYCLOPEDIA). | 10 |
| FIG. 6. EJE DEL ROTOR Y ASPA DE TURBINA DE VIENTO (CARTA, J. A., CALERO, R., COLMENAR, A., & CA. CENTRALES DE ENERGÍAS RENOVABLES GENERACIÓN ELÉCTRICA CON ENERGÍAS RENOVABLES). | 13 |
| FIG. 7. FUERZAS DE SUSTENTACIÓN Y ARRASTRE SOBRE UN PERFIL AERODINÁMICO.(CARTA, J. A., CALERO, R., COLMENAR, A., & CA. CENTRALES DE ENERGÍAS RENOVABLES GENERACIÓN ELÉCTRICA CON ENERGÍAS RENOVABLES). | 14 |
| FIG. 8. PERFIL AERODINÁMICO NRELS818. (NREL). | 16 |
| FIG. 9. DISTRIBUCIÓN DE LAS CUERDAS EN LOS 3 DISEÑOS DE ASPAS. | 16 |
| FIG. 10. MODELO CAD DEL ASPA DE LA TURBINA DE VIENTO. | 17 |
| FIG. 11. MODELO DEL ASPA 1 CON VELOCIDAD ESPECÍFICA DE 8 DISEÑADA EN SOLIDWORKS. | 18 |
| FIG. 12. MODELO DEL ASPA 2 CON VELOCIDAD ESPECÍFICA DE 9 DISEÑADA EN SOLIDWORKS. | 18 |
| FIG. 13. MODELO DEL ASPA 3 CON VELOCIDAD ESPECÍFICA DE 10 DISEÑADA EN SOLIDWORKS. | 19 |
| FIG. 14. CONSTRUCCIÓN DEL MODELO DE UN ASPA EN ANSYS. | 22 |
| FIG. 15. MODELO DE UN ASPA CONSTRUIDO EN ANSYS. | 22 |
| FIG. 16. CONFIGURACIÓN DE LA MALLA EN UN ASPA REALIZADA EN ANSYS. | 23 |
| FIG. 17. SIMULACIÓN FEM DE UNO DE LOS DISEÑOS DEL ASPA REALIZADA EN ANSYS. | 23 |
| FIG. 18. GRAFICA DE CONVERGENCIA DE LA MALLA DEL ANÁLISIS DE ELEMENTO FINITO. | 24 |
| FIG. 19. COMPARACIÓN ENTRE (A) EL DISEÑO CONVENCIONAL Y (B) EL DISEÑO ÓPTIMO. | 28 |
| FIG. 20. PUNTOS DE LOS DISEÑOS INICIALES. | 31 |
| FIG. 21. FUNCIÓN OBJETIVO DE LA MASA CON LOS 26 PUNTOS DE LA TABLA 3. | 33 |
| FIG. 22. FUNCIÓN OBJETIVO DE LA MASA. | 34 |
| FIG. 23. FUNCIÓN OBJETIVO DEL COEFICIENTE DE POTENCIA. | 35 |
| FIG. 24. FUNCIÓN OBJETIVO DE LA FRECUENCIA NATURAL. | 35 |
| FIG. 25. DEEP LEARNING COMO UN SUBCONJUNTO DE TÉCNICAS DE APRENDIZAJE AUTOMÁTICO. (BERZAL, F. (2018). REDES NEURONALES & DEEP LEARNING. GRANADA: FERNANDO BERZAL.). | 38 |
| FIG. 26. RED NEURONAL ARTIFICIAL SIMPLE. (BERZAL, F. (2018). REDES NEURONALES & DEEP LEARNING. GRANADA: FERNANDO BERZAL.). | 39 |
| FIG. 27. CÁLCULO DEL GRADIENTE CON PROPOGACIÓN DE ERRORES HACIA ATRÁS. (BERZAL, F. (2018). REDES NEURONALES & DEEP LEARNING. GRANADA: FERNANDO BERZAL.). | 41 |
| FIG. 28. MODELO ABSTRACTO DE UNA NEURONA ARTIFICIAL. (BERZAL, F. (2018). REDES NEURONALES & DEEP LEARNING. GRANADA: FERNANDO BERZAL.). | 45 |
| FIG.29. MODELO SIMPLIFICADO DE NEURONA CON FASE DE INTEGRACIÓN Y DISPARO. (BERZAL, F. (2018). REDES NEURONALES & DEEP LEARNING. GRANADA: FERNANDO BERZAL.). | 46 |
| FIG. 30. MODELO ESTÁNDAR DE NEURONA ARTIFICIAL CON FUNCIÓN DE ACTIVACIÓN. (BERZAL, F. (2018). REDES NEURONALES & DEEP LEARNING. GRANADA: FERNANDO BERZAL.). | 47 |
| FIG. 31. GRAFICA DE UNA FUNCIÓN DE ACTIVACIÓN LINEAL. (BERZAL, F. (2018). REDES NEURONALES & DEEP LEARNING. GRANADA: FERNANDO BERZAL.). | 48 |
| FIG. 32. GRAFICA DE UNA FUNCIÓN DE ACTIVACIÓN DE ESCALÓN. (BERZAL, F. (2018). REDES NEURONALES & DEEP LEARNING. GRANADA: FERNANDO BERZAL.). | 48 |
| FIG. 33. GRAFICA DE UNA FUNCIÓN DE ACTIVACIÓN DE SIGMOIDAL (AZUL). GRAFICA DE LA DERIVADA DE LA FUNCIÓN SIGMOIDAL (ROJO). (BERZAL, F. (2018). REDES NEURONALES & DEEP LEARNING. GRANADA: FERNANDO BERZAL.). | 50 |
| FIG. 34. GRAFICA DE UNA FUNCIÓN DE ACTIVACIÓN LIENAL RECTIFICADA. (BERZAL, F. (2018). REDES NEURONALES & DEEP LEARNING. GRANADA: FERNANDO BERZAL.). | 50 |

| | |
|---|----|
| FIG. 35. RED NEURONAL SIMPLE SIN CAPAS OCULTAS. (BERZAL, F. (2018). REDES NEURONALES & DEEP LEARNING. GRANADA: FERNANDO BERZAL.) | 51 |
| FIG. 37. RED NEURONAL PROFUNDA CON MÚLTIPLES CAPAS OCULTAS. (BERZAL, F. (2018). REDES NEURONALES & DEEP LEARNING. GRANADA: FERNANDO BERZAL.) | 52 |
| FIG. 38. ARQUITECTURA DEL MODELO DE DEEP LEARNING. | 56 |
| FIG. 39. GRÁFICA DE LA PRUEBA 1 DEL ENTRENAMIENTO DEL MODELO DE DL CON 100 ÉPOCAS, MAE:0.115, MSE: 0.030 y RMSE:0.174..... | 61 |
| FIG. 40. GRÁFICA DE LA PRUEBA 2 DEL ENTRENAMIENTO DEL MODELO DE DL CON 150 ÉPOCAS, MAE: 0.100, MSE: 0.024 y RMSE:0.153..... | 62 |
| FIG. 41. GRÁFICA DE LA PRUEBA 3 DEL ENTRENAMIENTO DEL MODELO DE DL CON 200 ÉPOCAS, MAE: 0.090, MSE: 0.023 y RMSE:0.152..... | 62 |
| FIG. 42. GRÁFICA DE LA PRUEBA 4 DEL ENTRENAMIENTO DEL MODELO DE DL CON 250 ÉPOCAS, MAE: 0.100, MSE: 0.028 y RMSE:0.168..... | 63 |
| FIG. 43. GRÁFICA DE LA PRUEBA 5 DEL ENTRENAMIENTO DEL MODELO DE DL CON 300 ÉPOCAS, MAE: 0.118, MSE: 0.032 y RMSE:0.180..... | 63 |
| FIG. 44. GRÁFICA DE LA PRUEBA 6 DEL ENTRENAMIENTO DEL MODELO DE DL CON 350 ÉPOCAS, MAE: 0.053, MSE: 0.013 y RMSE:0.112..... | 64 |

Índice de Tablas

| | |
|--|----|
| TABLA 1. CARACTERÍSTICAS GEOMÉTRICAS Y OPERACIONALES DE LAS ASPAS DISEÑADAS | 16 |
| TABLA 2. FRECUENCIA NATURAL DE LOS MODELOS DE ASPAS ANALIZADOS..... | 25 |
| TABLA 3. VALORES DE LA FUNCIÓN OBJETIVO DE LA MASA..... | 32 |
| TABLA 4. VALIDACIÓN DE LA FUNCIÓN OBJETIVO DE LA MASA USANDO QBLADE. | 34 |
| TABLA 5. RESULTADOS DE LOS EXPERIMENTOS DE ENTRENAMIENTO DE LOS MODELOS DE DL, VARIANDO EL NÚMERO DE ÉPOCAS POR PRUEBA. LAS PREDICIONES CONSIDERADAS COMO PRECISAS SON AQUELLAS CON AL MENOS UN 90% DE PRECISIÓN EN CADA PRUEBA. | 60 |
| TABLA 6. PREDICIONES DE LA MASA DEL ASPA EN DIFERENTES PRUEBAS, LOS RESULTADOS MUESTRAN UNA PRECISIÓN SUPERIOR AL 90% EN EL MODELO DE DL..... | 65 |
| TABLA 7. PREDICIONES DEL COEFICIENTE DE POTENCIA PARA DIFERENTES PRUEBAS, CON UNA PRECISIÓN SUPERIOR AL 90% EN EL MODELO DE DL..... | 66 |
| TABLA 8. PREDICIONES DE FRECUENCIAS NATURALES PARA DIFERENTES PRUEBAS, CON UNA PRECISIÓN SUPERIOR AL 90% EN EL MODELO DE DL..... | 66 |

1

INTRODUCCIÓN

Las fuentes de energías renovables han tenido una participación notable en los últimos 10 años, fuentes como la energía eólica empiezan a tener un papel más importante alrededor del mundo para la producción energía eléctrica. Las turbinas modernas buscan transformar de una mejor forma la energía cinética del viento en energía eléctrica, mediante la implementación de avances tecnológicos que han surgido de las investigaciones científicas, las cuales han impulsado a la energía eólica a nivel mundial reduciendo los costos y convirtiéndola en una forma rentable de producción de energética frente a otros tipos de generación de energía [1]. La implementación de parques eólicos a gran escala, ha resultado en la búsqueda de nuevas herramientas para potenciar la extracción de energía eólica y en la optimización de las turbinas de viento. Desde el año 2000, la energía eólica ha tenido una tasa de crecimiento de más del 21%. Desde el 2010 la instalación de energías renovables se ha acelerado, incrementando los niveles de adición de capacidad de energía anual. La capacidad instalada acumulada global de energía eólica onshore alcanzó 542 GW a finales de 2018. Se espera que para el 2030 el total de la capacidad instalada de la energía eólica onshore crecerá a 1787 GW y para el 2050 crecerá cerca de 5044 GW [2]. El aumento significativo de la demanda de la energía eólica, se ha traducido en el aumento de tamaño de los rotores, que se entiende en una mayor captación del viento, pero contar con aerogeneradores de dimensiones mucho más grandes, exige mayores medidas de durabilidad y rendimiento estructural más exigentes. Tomando en cuenta todas estas consideraciones, el diseñar una turbina de viento se vuelve un problema complejo, es por ello que se ha convertido en un problema de optimización en donde se tienen restricciones de diseño y una función objetivo. El enfoque de este trabajo es obtener la función objetivo del alabe de una turbina de viento de baja capacidad para un perfil aerodinámico NREL S818, usando las variables de la velocidad específica y los esfuerzos de Von Mises, con la finalidad de usar esta función para la generación de datos para la conformación de una base de datos, misma que se implementaría en el entrenamiento de una red neuronal con la capacidad de hacer predicciones de 3 variables de diseño. Otros estudios orientados en turbinas de viento [3-4], en donde han implementado el uso de las redes neuronales se enfrentan ante la escasez de datos que son necesarios en el entrenamiento de las redes neuronales. Ante esta situación el empleo de una función objetivo como generador de datos se presenta como una opción viable, dotando a la red de los datos necesarios para generar un modelo con la capacidad de hacer las predicciones necesarias.

Este trabajo se ha estructurado en siete capítulos:

Capítulo 1-Introducción. En este capítulo se aborda la revisión de antecedentes y el estado del arte, se exponen y justifican los objetivos de la presente investigación. Finalmente, se menciona la aportación de este estudio.

Capítulo 2-Diseño de turbinas de viento. Se realiza un recorrido sobre el viento y las primeras máquinas de viento en la historia. Se muestra la metodología del diseño aerodinámico de las aspas y el análisis de elemento finito de los diseños obtenidos.

Capítulo 3-Diseño óptimo. Se menciona la metodología de diseño óptimo, su aplicación al diseño de las aspas y la construcción de funciones matemáticas que sirven como criterios de optimización .

Capítulo 4- Inteligencia artificial. Este capítulo explica de forma general la inteligencia artificial, las redes neuronales artificiales y el deep learning.

Capítulo 5- Deep learning aplicado al diseño óptimo de las aspas de un aerogenerador. Este capítulo describe el procedimiento de la conformación del modelo de deep learning y como es aplicado al diseño óptimo de las aspas de una turbina de viento.

Capítulo 6-Análisis de resultados se muestran los resultados de las predicciones del modelo de deep learning obtenido.

Capítulo 7-Conclusiones. En este capítulo se enfatizan los principales hallazgos obtenidos en relación a los objetivos planteados.

1.1 Antecedentes

El diseño de aspas de aerogeneradores es un proceso en donde se involucran un gran número de variables; por esta razón es importante el uso de mejores herramientas para obtener un diseño aceptable. El diseño de las aspas se basa tanto en optimizaciones en la parte aerodinámica [1, 2] como en la parte estructural [3, 4].

Así, el diseño aerodinámico considera la geometría del aspa y su reacción a las cargas aerodinámicas. En la parte estructural, el armazón del aerogenerador debe tener la capacidad de soportar las condiciones naturales del emplazamiento donde se instalará. En la parte de la

optimización se han empleado diferentes técnicas y teorías en la mejora de aerogeneradores con diferentes propósitos. Por ejemplo, Karam y Hani [6] implementaron un modelo de optimización para minimizar el peso y las vibraciones, así como para maximizar la rigidez de las aspas. En este estudio, el modelo propuesto para hallar el óptimo global proporcionó un modelo matemático para calcular las frecuencias del sistema. El problema de optimización se formuló como una programación matemática no lineal resuelta mediante técnicas de búsqueda multidimensional. El procedimiento mostró resultados prometedores, pero resultaba complicado y extenso.

Dongran et al. [7] propusieron un método para minimizar el costo de la energía de los aerogeneradores optimizando el radio del rotor, la altura del hub y la potencia nominal. Realizaron estudios para comparar sus resultados entre tres algoritmos de optimización diferentes; en donde el algoritmo de optimización de pares Yin-Yang mejorado, mostró un mejor rendimiento en la resolución del problema de optimización. Del mismo modo, Jaehwan et al. [8] propusieron un método de optimización para encontrar la altura del hub; su proceso utilizó la producción anual de energía, como función objetivo, y utilizando funciones matemáticas con sus restricciones, encontraron la altura óptima. Estos trabajos muestran el uso de diferentes variables de optimización, que en algunos casos pueden verse limitadas debido a la complejidad del problema. Estos trabajos indican que abordar un caso de estudio como un problema de optimización con restricciones de diseño y una función objetivo arroja resultados favorables en comparación con la técnica de diseño clásica, que incluye un factor de seguridad para garantizar el correcto funcionamiento del sistema.

En años recientes los diferentes algoritmos de optimización se han implementado en mayor medida; estos se clasifican en métodos matemáticos y programación dinámica, y algoritmos metaheurísticos.

Algunos de ellos se mencionan en los siguientes trabajos: Kaveh y Sabati [9] utilizan un algoritmo llamado Colliding Bodies Optimization (CBO) para optimizar la estructura de un aerogenerador offshore. El algoritmo se implementa para encontrar las soluciones óptimas; como desventaja, el uso del algoritmo CBO sufre de baja velocidad de convergencia y convergencia prematura. Quang et al. [10] implementaron un algoritmo de Optimización por Enjambre de Partículas (PSO) para optimizar perfiles aerodinámicos. Algunas desventajas del uso de este algoritmo son que puede caer fácilmente en un óptimo local en un espacio de alta dimensión y tiene una baja velocidad de convergencia en el proceso iterativo.

Mohamed et al. [11] utilizaron un algoritmo evolutivo combinado con el software Ansys, buscando el mejor coeficiente de potencia y optimizando para encontrar la mejor forma aerodinámica. Ernesto y Andre [12] también utilizaron un algoritmo evolutivo orientado a un método de optimización multiobjetivo para maximizar la energía producida anualmente en un emplazamiento eólico y reducir el COE.

Los algoritmos evolutivos (AE) son excelentes para encontrar soluciones óptimas. Sin embargo, es esencial tener en cuenta que no necesariamente encuentran la solución óptima, es decir, un máximo o un mínimo global. Una desventaja adicional son los requisitos computacionales relativamente altos de los EA, que pueden estar relacionados con la complejidad del problema. Otra alternativa son los algoritmos genéticos, los cuales ofrecen flexibilidad y adaptabilidad, pero pueden llegar a ser complejos computacionalmente, presentar convergencia prematura y depender de la aleatoriedad. Mojtaba et al. [13] utilizaron un algoritmo genético en una turbina eólica de eje horizontal para optimizar la generación de energía como su función objetivo.

El uso de algoritmos en la optimización ofrece resultados suficientes, aunque en algunos casos, estos algoritmos pueden estar limitados en su capacidad para encontrar soluciones óptimas a problemas complejos; en estas situaciones, esto abre la puerta a otras herramientas, como el deep learning.

De acuerdo al gran número de variables que se manejan en el diseño de aerogeneradores y a su compleja teoría, la adopción de nuevas herramientas, como las Redes Neuronales Artificiales (RNA) se vuelve esencial. En los últimos años, el uso de estos modelos computacionales se ha orientado a la energía eólica [14, 15], por ejemplo, en la predicción del pronóstico de energía eólica [16], sistemas de control de aerogeneradores [17], diagnóstico de fallos [18], etc.

Sessarego et al. [19] optimizan el diseño de una curva de aspas aplicando una RNA y un simulador aerodinámico, consiguiendo un 1% de potencia adicional con un ligero incremento del empuje medio del rotor del 0.02%. En este estudio, una de las limitaciones fue la generación de un conjunto de datos de muestra que contenía veinte formas de aspas que era el máximo permitido debido al coste computacional y a los límites de tiempo.

Romanski et al. [3] mostraron una RNA para modelar los parámetros de funcionamiento de un aerogenerador de baja capacidad; el diseño de la turbina permite modificar el ángulo de redirección de las palas y la distancia entre los rotores. En este trabajo, los autores mencionan que el uso de redes neuronales para correlacionar las variables de funcionamiento con la potencia generada es esencial porque puede ayudar a construir una unidad de control inteligente para un aerogenerador.

Otros estudios sobre aerogeneradores, en los que han implementado RNA, se enfrentan a la escasez de datos necesarios para el entrenamiento de la RNA. Wen et al. [4] implementaron una curva de Bessel en el entrenamiento de una RNA con capacidad para predecir los coeficientes de sustentación y resistencia con una precisión del 90% para optimizar un perfil aerodinámico. Aún en estos casos, la cantidad de datos restringe el modelo, por lo que el

perfil aerodinámico optimizado se ve afectado por la falta de datos sobre su rendimiento. Por lo tanto, la falta de datos es un problema común cuando se utilizan modelos de aprendizaje profundo. Esta investigación utiliza funciones objetivo del diseño óptimo para desarrollar un modelo de aprendizaje profundo para predecir tres variables de diseño. Este trabajo contribuye a generar una base de datos a partir de las funciones objetivo, que se implementó en el entrenamiento de un modelo DL.

1.2. Objetivos

General: Desarrollar un procedimiento de optimización multicriterio del diseño aerodinámico y estructural de los álabes de una turbina de viento implementando un modelo deep learning.

Específicos

1. Revisión bibliográfica.
2. Diseño aerodinámico de los álabes de una turbina de viento.
3. Diseño estructural de los álabes de una turbina de viento.
4. Diseño óptimo general de elementos de máquinas.
5. Aplicación del diseño óptimo al proceso de diseño de un álabes de turbina de viento.
6. Exploración de técnicas de Inteligencia Artificial aplicables al diseño óptimo.

1.3. Justificación

El diseño de las turbinas de viento se ha vuelto un problema complejo, por la gran cantidad de variables que se toman en cuenta, su optimización en distintos enfoques ha generado la necesidad de incorporar nuevas herramientas, como es el caso de las redes neuronales artificiales o deep learning, las cuales dependen completamente de las bases de datos para poder desarrollar un modelo confiable. Ante la falta de los datos necesarios para desarrollar un modelo de deep learning, en este trabajo se intenta suplir una base de datos con una función objetivo como un generador de datos. Dicha función es construida empleando una metodología del diseño óptimo, con el cual se consigue un sistema que permite al diseñador manejar la cantidad de variables necesarias y sus respectivas restricciones, otorgando flexibilidad y factibilidad. La generación de la base de datos usando la función objetivo, otorgó el conjunto de datos suficientes para el entrenamiento, prueba y validación del modelo

de deep learning. Logrando predicciones acertadas bajo un porcentaje aceptable de precisión, ofreciendo una alternativa diferente en las aplicaciones de modelos de deep learning orientado al diseño óptimo de los álabes de una turbina de viento.

1.4. Hipótesis

Es posible generar bases de datos para entrenar un modelo DL utilizando funciones objetivo, construido con la metodología de diseño óptimo y la distancia ponderada inversa (IDW), que proporciona un sistema que establece un diseño multicriterio y sus respectivas restricciones, otorgando flexibilidad y factibilidad al diseño.

1.5. Contribución

Los métodos convencionales que son usados en el diseño de turbinas viento presentan un manejo de las variables necesarias cumpliendo con las especificaciones adecuadas, sin embargo, estos métodos no contemplan restricciones de diseño que ayudan a plantear un criterio de optimización en el diseño y se ven limitados en el número de variables que pueden usarse en un mismo estudio de manera simultánea. Ante esta falta de capacidad, surge la necesidad de mejores herramientas que ayuden a lograr la optimización en el diseño.

Esta investigación aporta:

- Las funciones objetivo de las aspas de una turbina de viento usando IDW, las cuales se encuentran en el dominio de la velocidad específica y el esfuerzo de Von mises, que tienen como variables independientes: la masa, el coeficiente de potencia y la frecuencia natural.
- La conformación de bases de datos, a partir de las funciones objetivo para el entrenamiento del modelo de deep learning.
- El diseño multicriterio de las aspas de una turbina de viento utilizando un modelo de deep learning.

2

DISEÑO DE TURBINAS DE VIENTO

2.1. Las turbinas de viento a través del tiempo

2.1.1. El viento

El viento ha sido un elemento que ha estado presente en el planeta desde tiempos inmemorables. Su manifestación en las antiguas civilizaciones no tardó en llamar la atención de los grandes pensadores de diferentes épocas.

Icaro

Uno de los clásicos ejemplos que existen en la historia, sobre como el hombre se ha fascinado por el viento, se encuentra en la mitología griega.

Las alas de Ícaro, un mito clásico, cuenta la historia del hijo de Dédalo, inventor y arquitecto, que ideó una forma de escapar de la isla de Creta, donde fueron encerrados por mandato del rey Minos. Dédalo construyó unas alas utilizando plumas de aves y cera derretida para unir los materiales, le advirtió a Icaro que no volara cerca del sol para que la cera no se derritiera y que no volara cerca del mar para que las alas no se mojaran.



Fig. 1. Caída de Icaro (Fitzsimons, S. (2009) Pompeii wall art, Pompeii Wall Art. Disponible en: <http://lookingglasscarus.blogspot.com/2009/06/pompeii-wall-art.html>.

Da Vinci

Leonardo da Vinci fue otro gran pensador que busco aplicar sus conocimientos en el dominio del viento con sus máquinas voladoras. Da Vinci mostraba una gran comprensión de los principios aerodinámicos para su época, dentro de los diseños más sobresalientes se encuentra el ornitóptero. El funcionamiento de esta máquina se inspira en el vuelo de las aves, el cual buscaba imitar el movimiento de las alas de un pájaro cuando se eleva en el aire. El funcionamiento de esta máquina se origina del estudio que da Vinci realizó en la anatomía de las aves, descubrió que para el vuelo era importante la forma de las alas y en como podían moverse. El diseño también contemplaba una pequeña cabina en donde el piloto podía usar sus brazos para emular el movimiento que hace un ave.

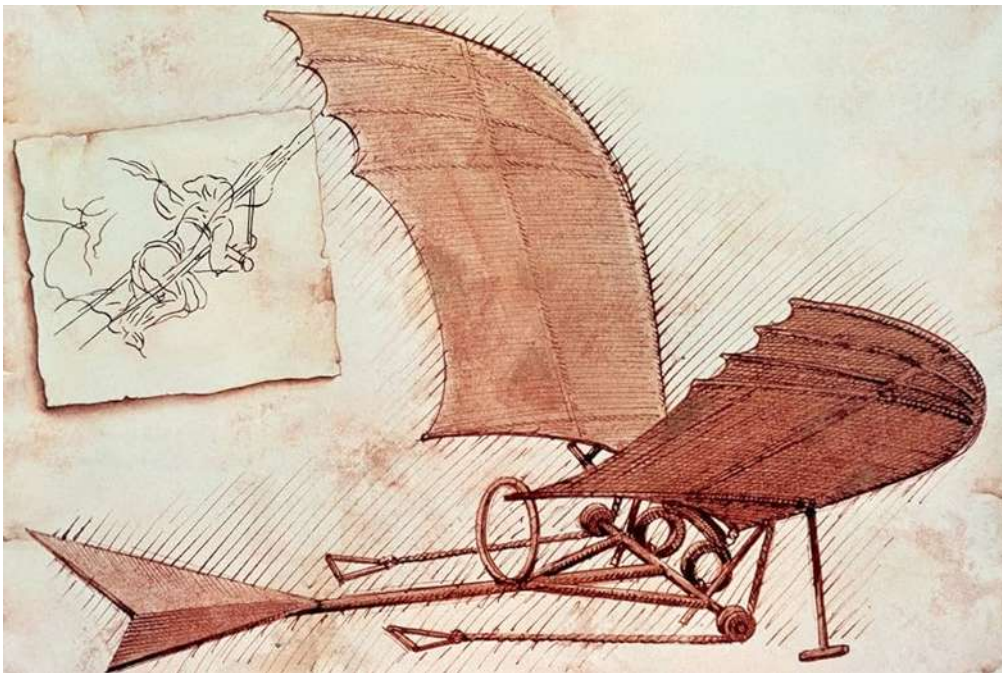


Fig. 2. Diseño de ornitóptero. Britannica, T. Editors of Encyclopedia Britannica ornithopter.

A pesar de que las máquinas voladoras de Leonardo dan Vinci no llegaron a construirse en su época, su visión y comprensión basada en, los principios aerodinámicos han dejado una marca indeleble en la historia de la aviación. Sus diseños, aunque a menudo considerados utópicos en su tiempo, han sido apreciados como obras maestras de la ingeniería y la creatividad, sirviendo como inspiración para generaciones de inventores y aviadores que finalmente hicieron realidad el sueño ancestral de volar.

2.2. Las primeras máquinas de viento

Desde las primeras civilizaciones de la humanidad, o incluso antes, el viento ha influenciado en la mitología y empujado a los grandes pensadores a imaginar novedosas máquinas con el objetivo de dominar a este elemento. El viento ha estado presente en las diferentes actividades del hombre, como la navegación, la agricultura, el transporte, etc. Los molinos de viento fueron las primeras máquinas en utilizar el viento como fuente de energía, existieron en la antigüedad en lugares como Egipto, Persia, Mesopotamia, y China, sus principales usos fueron la extracción de agua del subsuelo y la molienda de granos.



Fig. 3. Los molinos de viento de Nashtifan (SURFIRAN Editorial Team (ed.) (2017) 1000-Year-Old Windmills Still in Use Today in Iran, SURFIRAN Magazine.

Como una de las principales desventajas que tiene el viento como recurso es su intermitencia, poca fiabilidad, inestabilidad y que es impredecible. Como ventajas el viento es un recurso renovable que depende en gran medida de la fuente inagotable de la energía del sol que se queda en la atmósfera.

En Europa el uso de los molinos aparece en la Edad Media en lugares como Inglaterra, su implementación se propagó por todo el continente por sus importantes aplicaciones en la molienda de granos, bombear agua, fabricación de papel, aceite de semillas y aserrado de madera. Pero fue en Holanda donde los molinos de viento tuvieron una participación importante, ya que se usaban para drenar el agua del río Rin. Los diseños se adaptaban a las

características del lugar donde se deseaban instalar. Entre los diseños se tienen los de eje horizontal y vertical, siendo los de eje horizontal los preferidos por las personas dada su alta eficiencia. Surgieron 3 tipos de molinos de vientos europeos: con diseño de poste, diseño de torre y diseño tipo bata.



Fig. 4. Molinos de viento en España. (Britannica, T. Editors of Encyclopedia).

El gran auge de los molinos de viento en Europa no tardo en llamar la atención del continente americano. Estados Unidos desarrollo su propia versión con una estructura vertical de acero con una hélice de múltiples aspas que aprovecha el viento para bombear agua y almacenarla en un tanque de agua.

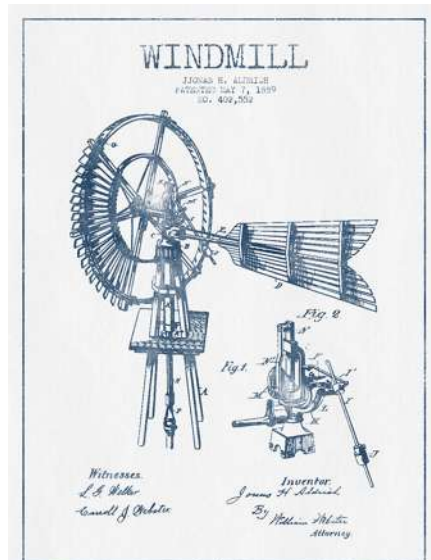


Fig. 5. Patente de molino de viento de Aldrich. (Aldrich windmill patent drawing from 1889, fine art america).

Los primeros molinos de viento modernos eran aquellos con la capacidad de accionar generadores eléctricos, fueron introducidos en Francia en el siglo XX. La aparición de la máquina de vapor y el motor de combustión interna sustituyeron a varias fuentes de energía, al ser más eficientes su popularidad creció de manera notable, que incluso en la actualidad estas máquinas se siguen implementando para un sinnúmero de aplicaciones.

La transición de los molinos a las turbinas de viento fue un proceso marcado por un gran número de cambios en el diseño, pero aun faltaba la incorporación de las ciencias para llegar a los mejores diseños. Fue hasta el siglo XVIII, que se implementaron la física y las matemáticas en el desarrollo de estas máquinas, nuevos diseños en las velas o aspas de las turbinas de viento fueron elaboradas basadas en las leyes de mecánica de fluidos.

En 1890, el danés Poul La Cour realizó una investigación científica sobresaliente en aerodinámica y diseño de molinos de viento. Poul La Cour fue animado por el gobierno Danés que buscaba una forma de suministrar electricidad a las zonas rurales del país. Con ese objetivo La Cour construyó un modelo experimental de una turbina que usaba un dinamo, su invento producía energía de corriente directa por electrólisis y almacenaba el gas de hidrógeno producido[27]. En años posteriores La Cour expandió sus actividades en Askov en donde estableció una estación de pruebas para las turbinas de viento, usando un túnel de

viento que el mismo construyó, después instaló una estación de pruebas mucho más grandes y fundo una asociación danesa de ingenieros de energía eólica.

Albert Betz fue quien hizo el mayor de los aportes con fórmulas aerodinámicas que dieron base a los principios modernos de conversión de energía del viento, permitiendo el desarrollo de hojas aluminio, basado en perfiles de aviones, con una aleta auxiliar ajustable que permitía la regulación de potencia y velocidad del molino de viento, que incrementarían en mayor medida la potencia de salida.

A lo largo de los años la energía eólica ha sufrido numerosos cambios hasta llegar a ser una posible alternativa para suplantar a los combustibles fósiles. La innovación de la energía eólica ha sido debido a la innovación en nuevos materiales, controles electrónicos y avances tecnológicos.

En 1980 los experimentos con las turbinas de gran escala empezaban a tomar un protagonismo importante entre las potencias mundiales, que buscaban sacar más provecho de la energía eólica y dar un salto en el rango de potencia a Megawatt. Los proyectos de las grandes turbinas de viento fueron tomados por empresas reconocidas a nivel global, ya que solo estas corporaciones tenían los recursos para desarrollar y construir proyectos de esta magnitud. Entre las grandes empresas estaban : Boeing, General Electric y Westinghouse en E.U, MAN, MBB, Dornier, Voith en Alemania y Kvaerner en Suecia [28].

En un principio el desarrollo de estos proyectos empezaron en Estados Unidos, primero con las series de turbinas de viento de gran capacidad MOD-0 a MOD-5 fueron instaladas y puestas a prueba. En Canadá las instituciones del gobierno estaban interesadas en realizar investigaciones en tecnología para producir energía eléctrica con turbinas de viento de eje vertical tipo Darrieus. En Europa alcanzar grandes dimensiones en sus turbinas de viento no era su objetivo principal, dado que sus turbinas experimentales eran de menor tamaño comparadas con las de Estados Unidos. El objetivo principal para todos estos países que participaron en las pruebas de muchos modelos de turbinas de gran potencia era darle un sustento científico a la tecnología desarrollada y sobre todo que se tuviera un estudio estratégico con la viabilidad económica. Las investigaciones, estudios y pruebas en las que participaron numerosas organizaciones gubernamentales y privadas generó nuevos modelos con fines comerciales, además de que la Agencia Internacional de Energía (AIE) fomentó el intercambio internacional de resultados de pruebas [28].

Los primeros prototipos de primera generación fueron sometidos a pruebas intensivas y luego fueron operados durante otros diez años, interrumpidos por períodos relativamente largos de descanso. Pero los resultados basados en el número de horas de operación, las turbinas más exitosas alcanzaron un número de unos pocos miles de horas de operación, mientras que las

menos exitosas solo lograron unas pocas horas, lo cual fue una decepción para las altas expectativas políticas, sin embargo, todas estas pruebas y recursos fueron necesarios para sentar las bases de la energía eólica. Las grandes turbinas experimentales sentaron las bases tecnológicas de la tecnología moderna de energía eólica proporcionando una amplia base técnica científica, conformada por años de pruebas, diseños y estudios tuvieron lugar para llevar la tecnología de energía eólica a su nivel actual [28].

2.3. Diseño aerodinámico de las aspas de una turbina de viento

Las turbinas viento se pueden clasificar en máquinas de eje horizontal o eje vertical, los aerogeneradores de eje horizontal, son los más usados debido a su alta eficiencia, de acuerdo a las bases teóricas propuestas por Albert Betz, la eficiencia máxima posible que se puede alcanzar es de un 59.3 % [29]. En la búsqueda por mejorar la eficiencia que se tienen en este tipo de dispositivos, nuevas técnicas se han implementado en la optimización de sus partes, como lo son las aspas. Un aerogenerador se encuentra conformado por un gran número de componentes, entre los cuales destacan el rotor, en este se encuentra uno de los elementos más importantes, el aspa, la cual se encarga de transformar la energía cinética del viento a energía mecánica.

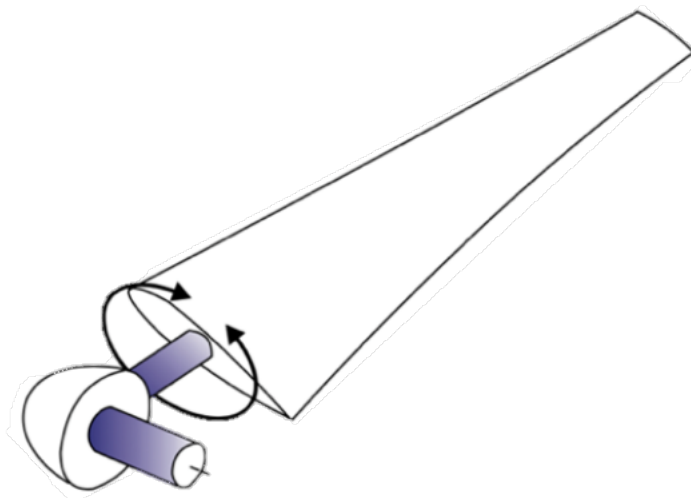


Fig. 6. Eje del rotor y aspa de turbina de viento (Carta, J. A., Calero, R., Colmenar, A., & Ca. Centrales de energías renovables generación eléctrica con energías renovables).

El diseño de un aspa de una turbina de viento, comienza con la selección del perfil aerodinámico, estos perfiles tienen una cuerda, la cual une el borde de ataque y el borde de fuga, se forma un ángulo de inclinación o ángulo de ataque entre la cuerda y la dirección del viento, el cual tiene una gran influencia sobre las fuerzas de sustentación y arrastre.

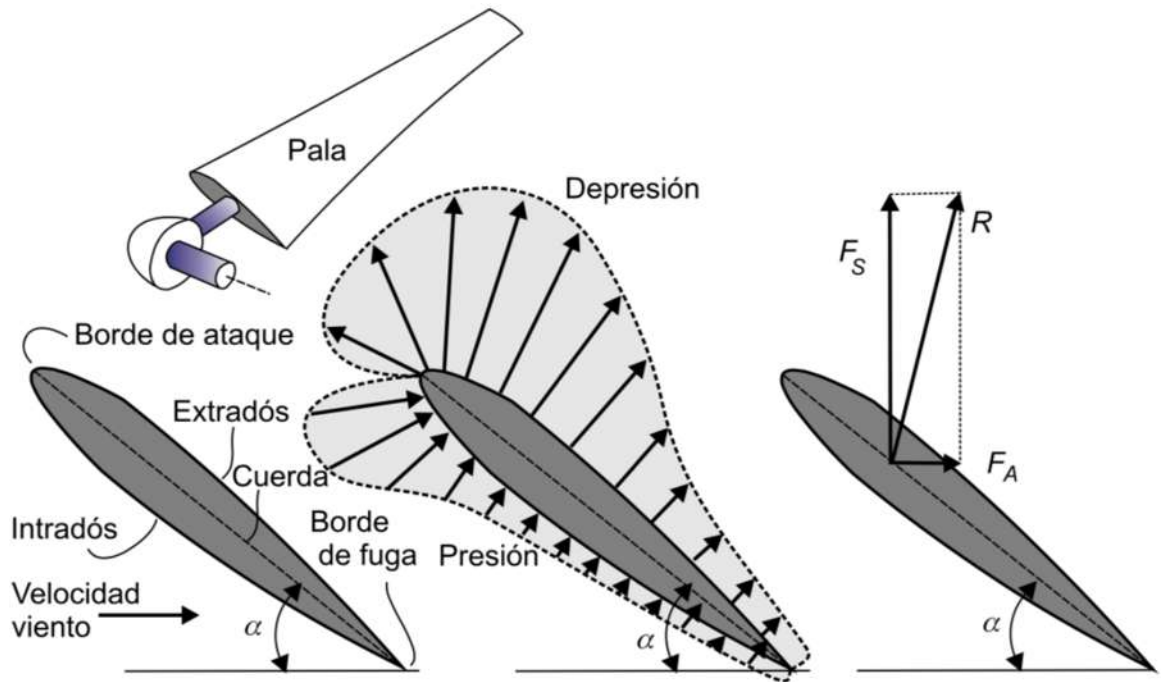


Fig. 7. Fuerzas de sustentación y arrastre sobre un perfil aerodinámico. (Carta, J. A., Calero, R., Colmenar, A., & Ca. Centrales de energías renovables generación eléctrica con energías renovables). En el proceso de diseño de las aspas de un aerogenerador intervienen muchas variables, parámetros y teorías. Los parámetros usados en el diseño de las aspas fueron los siguientes: una velocidad del viento de 12 m/s [36], una temperatura de 25°C, densidad del aire de 1.28 kg/m^3 , y una viscosidad dinámica de $1.8490 \times 10^{-5} \text{ kg/m} \cdot \text{s}$. El rango de la velocidad específica propuesta para las aspas fue de $5 \leq \lambda \leq 10$ [29].

2.3.1. Método de momento de elemento de pala (BEM)

El método BEM incluye la teoría del momento en 1D, en la que el análisis supone un disco actuador dentro de un tubo de corriente. Las características aerodinámicas del rotor (disco actuador), como la potencia, P , la fuerza de empuje, T , y el par, Q , pueden calcularse a partir de la velocidad del viento aguas arriba, U , el radio del rotor, R , y la velocidad angular del rotor.

$$dT = \rho U^2 4a(1-a)\pi r dr \quad (1)$$

$$dQ = 4a'(1-a)\rho U \pi r^3 \Omega dr \quad (2)$$

donde ρ es la densidad del aire, a es el factor de inducción axial, a' es el factor de inducción angular, r es el radio local del elemento y dr es la longitud radial del elemento.

El método BEM también incluye el método del elemento de pala, en el que las fuerzas sobre el aspa vienen determinadas por las características de sustentación y resistencia del perfil aerodinámico, de forma que:

$$dT = B(1/2)\rho U_r^2 (C_l \cos\phi + C_d \sin\phi) c dr \quad (3)$$

$$dQ = B(1/2)\rho U_r^2 (C_l \sin\phi - C_d \cos\phi) c dr \quad (4)$$

donde B es el número de aspas del rotor, U_r es la velocidad relativa del viento, C_l es el coeficiente de sustentación, C_d es el coeficiente de arrastre, ϕ es el ángulo del viento relativo y c es la longitud de cuerda del perfil aerodinámico.

$$C_P = \frac{P}{P_{wind}} = \frac{\int_{r_h}^R \Omega dQ}{\frac{1}{2} \rho \pi R^2 U^3} = \frac{8}{\lambda^2} \int_{\lambda_h}^{\lambda} \lambda_r^3 a' (1-a) [1 - (C_d/C_l) \cot\phi] d\lambda_r \quad (7)$$

Combinando las ecuaciones de momento y de elemento de pala, se pueden derivar expresiones para obtener los factores de inducción axial y radial:

$$a = \frac{1}{[1 + 4 \sin^2 \phi / (\sigma' C_l \cos\phi)]} \quad (5)$$

$$a' = \frac{1}{[(4 \cos \phi / (\sigma' C_l)) - 1]} \quad (6)$$

donde $\sigma = Bc/2\pi r$ σ es la solidez local.

Una vez obtenidos los factores de inducción de cada sección, el coeficiente de potencia del rotor puede calcularse a partir de la siguiente ecuación:

Donde $\lambda = \Omega R/U$ es la velocidad específica, y $\lambda_r = \lambda \cdot (r/R)$ es la velocidad específica local. Manwell et al. [29] describen con más detalle el método BEM.

En base a la teoría BEM, se diseñaron tres aspas utilizando el perfil aerodinámico NRELS818 [30]; véase la Fig. 8. Éste forma parte de una familia de perfiles aerodinámicos desarrollados por el Laboratorio Nacional de Energías Renovables. Se caracteriza por su buen rendimiento en un amplio rango de operación y su baja sensibilidad a los efectos de la rugosidad.

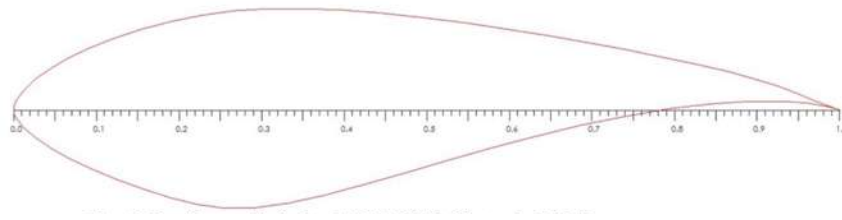


Fig. 8. Perfil aerodinámico NRELS818. (NREL).

En la Tabla 1 se resumen las principales características geométricas de las tres aspas diseñadas, así como el coeficiente de potencia del rotor obtenido a partir del análisis BEM.

Tabla 1. Características geométricas y operacionales de las aspas diseñadas.

| Álabe | Perfil | Velocidad específica | Peso(kg) | Longitud (m) | Cp |
|-------|----------|----------------------|----------|--------------|------|
| 1 | NRELS818 | 8 | 8.20 | 3 | 0.28 |
| 2 | NRELS818 | 9 | 6.32 | 3 | 0.27 |
| 3 | NRELS818 | 10 | 4.38 | 3 | 0.26 |

La Fig. 9 se muestra la distribución de la cuerda a lo largo de las aspas, se observa que la longitud de la cuerda es proporcional a la masa en los 3 casos propuestos.

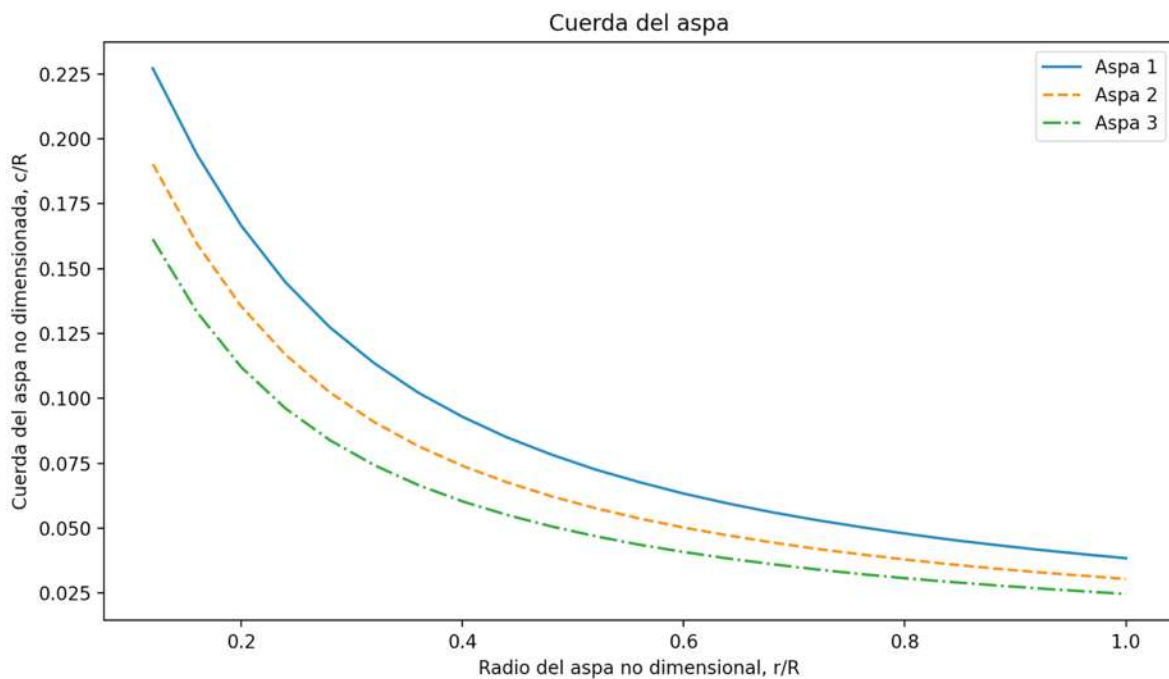


Fig. 9. Distribución de las cuerdas en los 3 diseños de aspas.

La cuerda de los perfiles de las aspas se calcula con Ec.(8), con una velocidad específica alta, las aspas se vuelven esbeltas con cuerdas más pequeñas, pero con una velocidad específica baja sucede lo contrario. Aunque las aspas conservan la misma longitud, la masa de las aspas se ve afectada por el tamaño de la cuerda, que depende por la relación que se tiene con velocidad específica. El tamaño de la cuerda de los 3 diseños de aspa puede verse en la Fig. 9.

$$\frac{N c}{2\pi R} = \frac{4\lambda\mu^2 a'}{\frac{W}{U_\infty} C_l} \quad (8)$$

En este trabajo el modelado de las aspas en 3D, se realizó en los programas de Solidworks y ANSYS. En la simulación CAD, como se muestra en la Fig. 10, se insertaron los planos correspondientes de los perfiles aerodinámicos que conforman a los diseños de aspas de manera consecutiva hasta configurar la forma completa del aspa. En la Fig. 11-13, se muestran los diseños finales realizados en el programa SolidWorks.

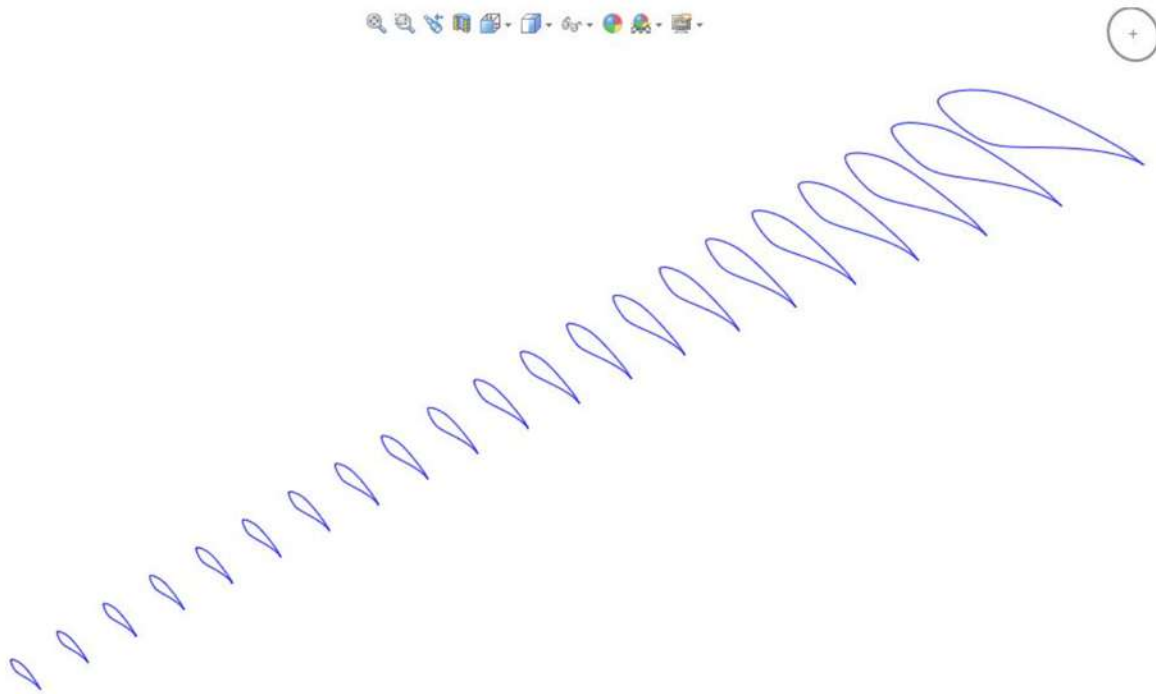


Fig. 10. Modelo CAD del aspa de la turbina de viento.

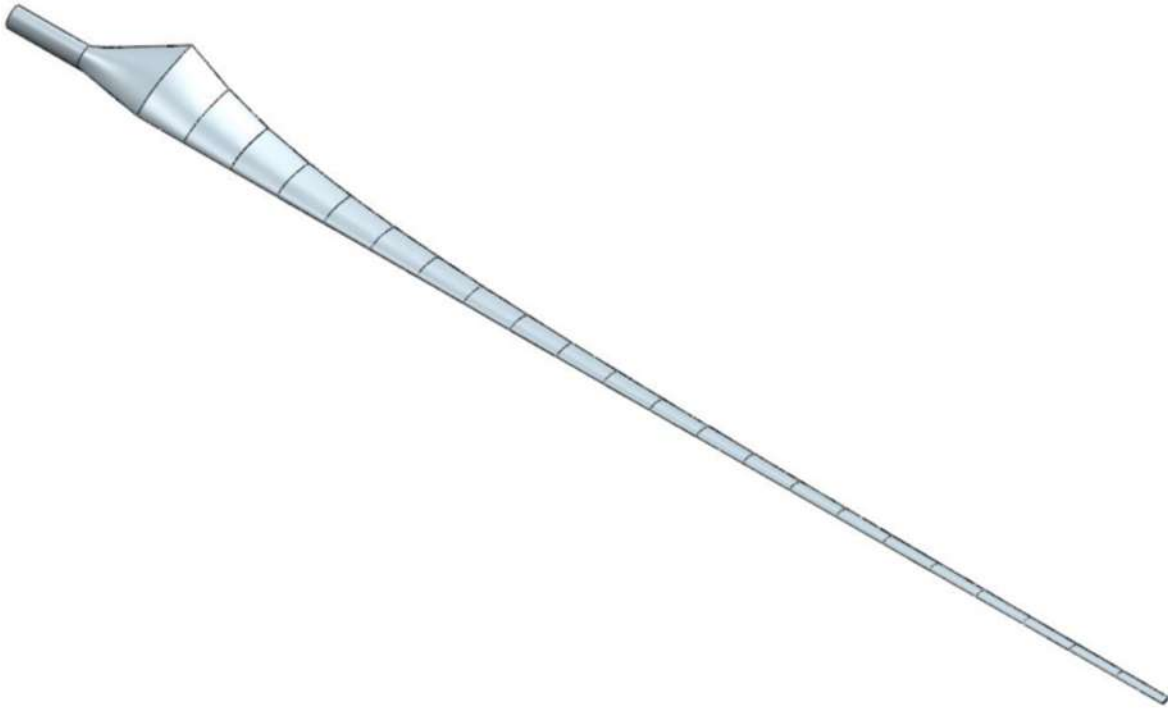


Fig. 11. Modelo del aspa 1 con velocidad específica de 8 diseñada en SOLIDWORKS.

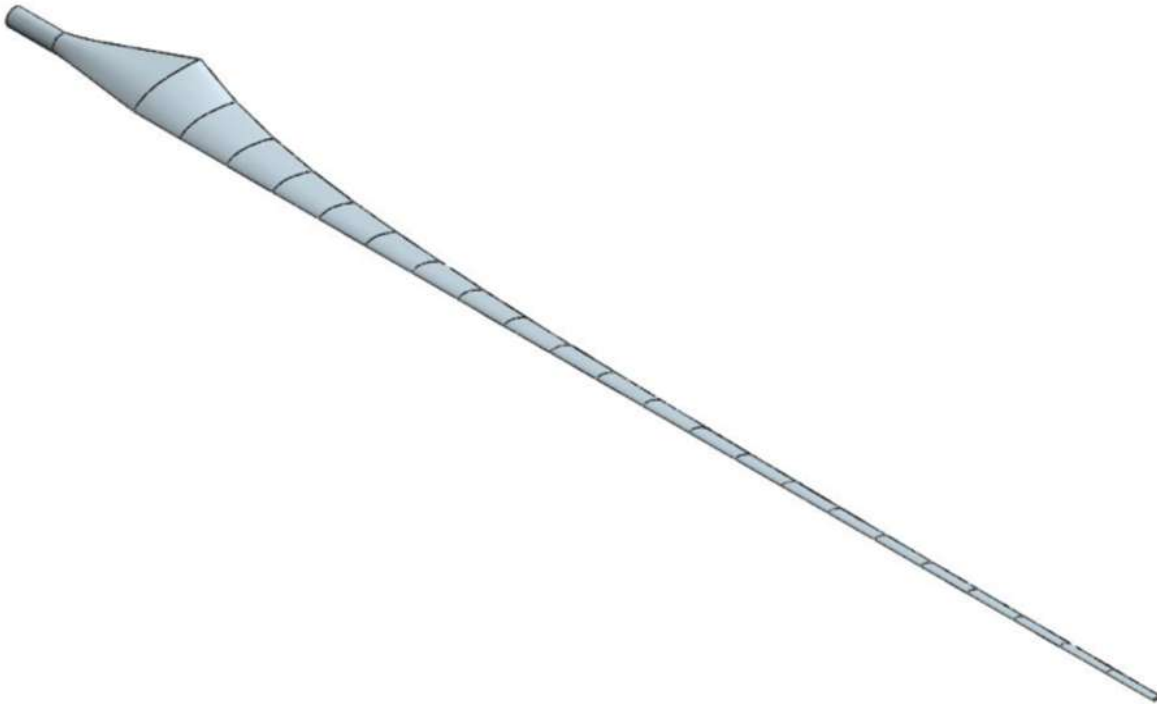


Fig. 12. Modelo del aspa 2 con velocidad específica de 9 diseñada en SOLIDWORKS.

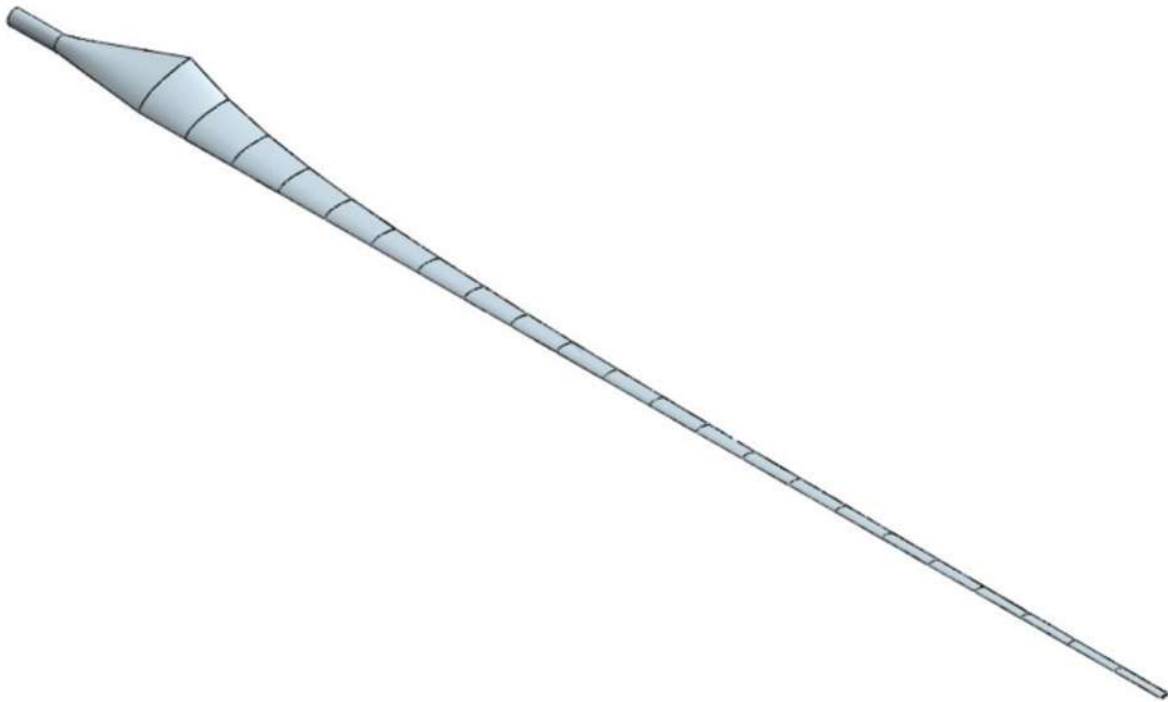


Fig. 13. Modelo del aspa 3 con velocidad específica de 10 diseñada en SOLIDWORKS.

Los 3 modelos de aspas propuestos, Fig.11-13, fueron diseñados con el perfil aerodinámico NRELS818, tienen una longitud de 2.93 m, se conforman por 23 secciones y se les asignó el material GLASS-E. Realizando una comparación entre los diseños, el aspa 1 que tiene un peso de 8.23 kg, y el aspa 2 con un peso de 6.32 kg, se tiene una diferencia 1.91 kg, lo cual nos indica que el aspa 2 es 23% más ligera que el aspa 1. El aspa 3 tiene un peso de 4.38 kg, si la comparamos con el peso que tiene el aspa 1, se tiene una diferencia 3.82 kg, lo cual nos indica que el aspa 3 es un 53% más ligera. En base a estas comparaciones entre los pesos de los 3 diseños, se observa como la velocidad específica tiene una relación significativa con la masa de las aspas.

El análisis aerodinámico se realizó en el software QBlade [31], obteniendo los coeficientes de potencia, C_p , de las 3 aspas analizadas. En QBlade, las fuerzas aerodinámicas que actúan

sobre un rotor se pueden modelar utilizando el método de momento y el elemento de aspa, incluyendo mejoras con varios modelos de corrección. El método vincula la teoría del disco actuador y la teoría del elemento de aspa y fue presentado por primera vez por Glauert [32]. A pesar de su simplicidad, el método BEM permite representar con precisión las cargas aerodinámicas constantes que actúan sobre el rotor de un aerogenerador, siempre que no se incumplan determinados supuestos del modelo. El método BEM aún es empleado para el diseño aerodinámico y la evaluación de turbinas de viento [33]. Actualmente, existe la forma de simular los fenómenos aerodinámicos que se presentan en las turbinas en el campo bajo condiciones diferentes tipos de condiciones, tales como turbulencia atmosférica, cizalladura del viento, desprendimiento de flujo, interacciones con turbinas vecinas, estela, etc., utilizando técnicas de Dinámica Computacional de Fluidos (CFD) [34]. Sin embargo, para el caso de estudio principal de este trabajo, es la incorporación de un modelo de deep learning en la metodología para el diseño óptimo de las aspas de un aerogenerador.

El material destinado a la fabricación de las aspas de aerogeneradores suele ser un material compuesto que puede contener distintos tipos de fibras, entre ellas fibras de vidrio. Las razones por las que se utilizan estos materiales residen en sus propiedades mecánicas. Las fibras por sí solas no son del todo útiles para fines estructurales, pero sus ventajas pueden aprovecharse en forma de materiales compuestos, que ofrecen ventajas en rigidez y masa. Los materiales compuestos pueden ser más difíciles de modelar que un material isótropo. Hay que tener cuidado al definir las propiedades y la orientación de las capas, ya que cada capa puede tener propiedades de material ortotrópico diferentes.

En el diseño de las aspas se utilizó una fibra de vidrio denominada Glass-E, que tiene las siguientes propiedades mecánicas Rigidez $K = 72 \text{ GPa}$, Resistencia a la tracción $\sigma = 3500 \text{ MPa}$, y Densidad $\rho = 2.54 \text{ g/cm}^3$.

Aunque los materiales compuestos presentan propiedades anisótropas, el vidrio que generalmente está compuesto por dióxido de silicio, óxido de aluminio y pequeñas cantidades de otros óxidos, sí presenta propiedades isótropas, debido a que los átomos de sílice y oxígeno forman una red sin orden cristalográfico, por lo que las fibras de vidrio son amorfas con propiedades isótropas, por ejemplo, rigidez y dilatación térmica [35].

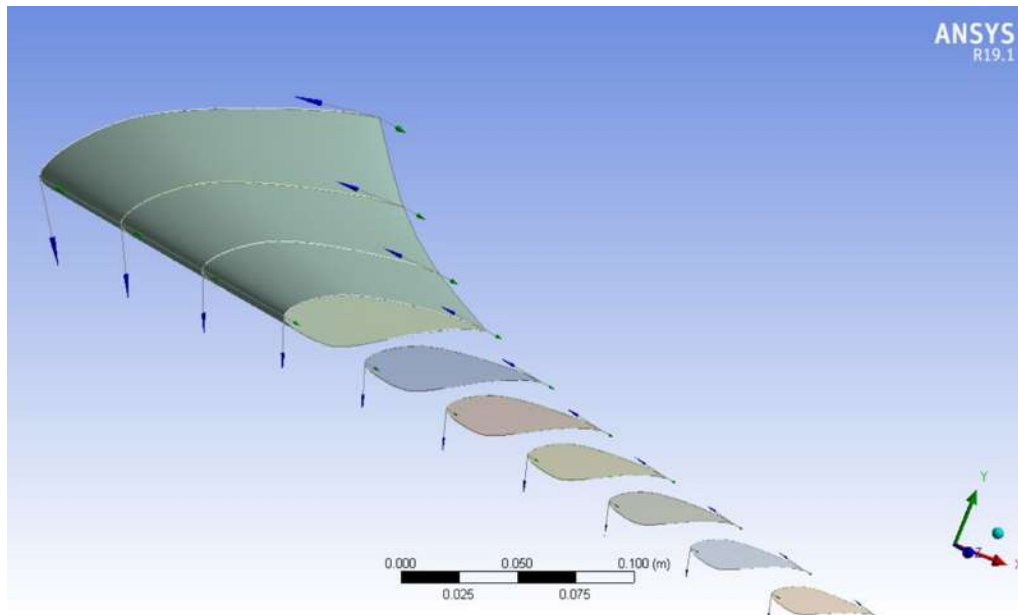
2.4 Análisis FEM

El análisis de elementos finitos es un método numérico que permite analizar el comportamiento mecánico de cada aspa en diferentes condiciones de carga [36, 37]. Las fuerzas aerodinámicas obtenidas a partir de la teoría BEM (fuerzas de empuje y tangenciales)

se utilizaron para analizar las aspas en el software ANSYS [38]. La simulación FEM se conformó de los siguientes pasos:

- **Modelo de la geometría:** es la representación por medio de diferentes tipos de elementos como: figuras geométricas, líneas, curvas, superficies, etc. El objetivo es tener una representación detallada del objeto de estudio para su posterior simulación.
- **Mallado:** es una tarea muy importante del pre-proceso, se trata de la discretización de la geometría en pequeños elementos o celdas, con la finalidad de encontrar una solución por medio de aproximaciones usando funciones simples como polinomios.
- **Material o medio:** es la definición de las propiedades del material del elemento(s) del objeto de estudio. Es importante definir un material para la correcta simulación de diferentes fenómenos. Entre algunas de las propiedades más usadas se encuentra el módulo de Young, y el módulo de corte que son necesarios para el análisis de tensiones de sólidos y estructuras, etc.
- **Condiciones de frontera:** Las condiciones de límite, inicial y de carga son importantes en la resolución de la simulación. El uso de estas condiciones generalmente se realiza fácilmente mediante pre-procesadores comerciales, y a menudo se interconecta con gráficos. Los usuarios pueden especificar estas condiciones ya sea a la geometría (puntos, líneas o curvas, superficies y sólidos) o con los elementos o cuadrículas.
- **Simulación:** Una vez que se tiene la malla de la geometría, la simulación FEM usa un conjunto de sistemas de ecuaciones simultáneas que son formuladas usando las aproximaciones existentes que se basan en principios de energía, el principio de energía potencial mínima, etc.
- **Solución de ecuaciones:** el modelo computacional que fue creado, es alimentado con un solucionador para resolver el sistema discretizado y las ecuaciones simultáneas para las variables de campo en los nodos de la malla.
- **Visualización:** El resultado generado después de resolver las ecuaciones del sistema suelen ser un gran volumen de datos digitales. Los resultados deben visualizarse de tal manera que sea fácil de interpolar, analizar y presentar. La visualización se realiza a través de un denominado post-procesador, la mayoría de estos procesadores permiten al usuario mostrar objetos 3D de muchas maneras prácticas y coloridas en la pantalla [45].

Se construyeron los modelos en 3-D de las aspas en el programa ANSYS, por medio del acomodo de los perfiles aerodinámicos hasta formar el aspa completa, con los modelos finalizados se procedió a llevar a cabo la simulación estructural.



La Fig. 15 muestra el modelo CAD de una de las aspas construida en ANSYS.

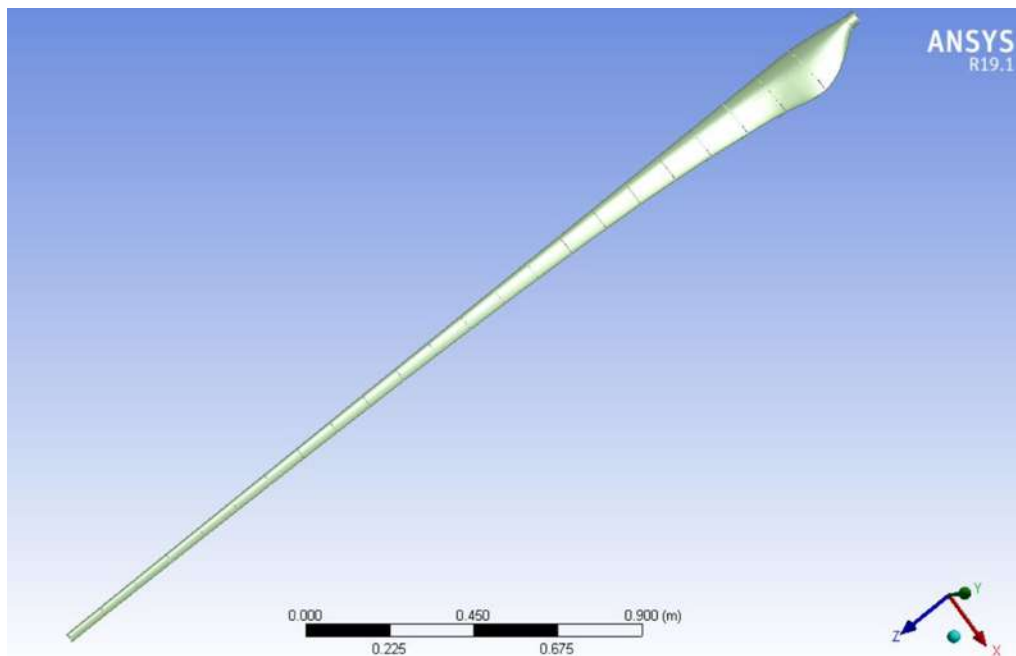


Fig. 15. Modelo de un aspa construido en ANSYS.

En el análisis estructural de las aspas se realizó un análisis de elemento finito con la finalidad de estimar el desempeño mecánico de cada diseño de aspa bajo diferentes condiciones de carga. Como parte de este procedimiento, se configuró una malla de Hexaedros, con diferentes tamaños de elementos en las distintas simulaciones y una gráfica de convergencia de la malla buscando encontrar los mejores resultados.

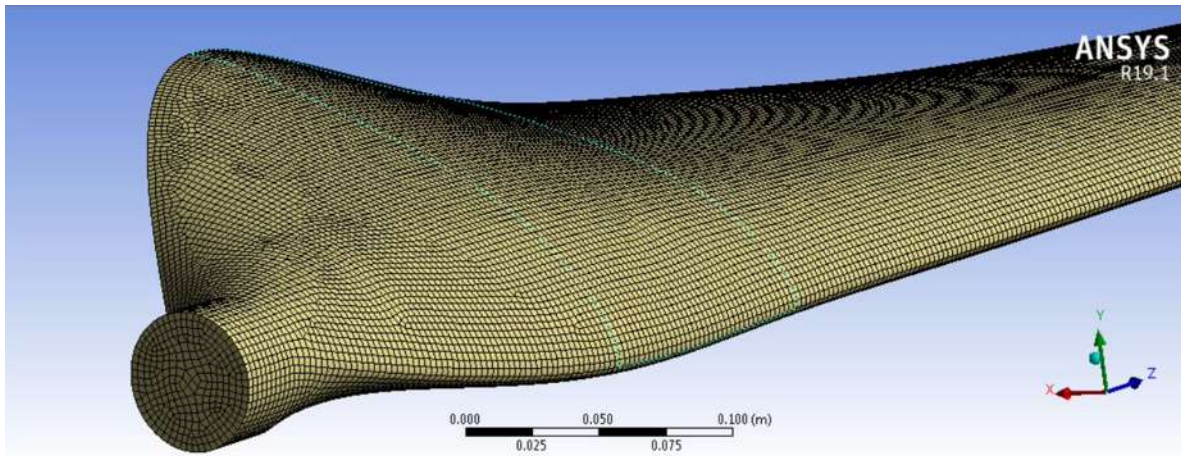


Fig. 16. Configuración de la malla en un aspa realizada en ANSYS.

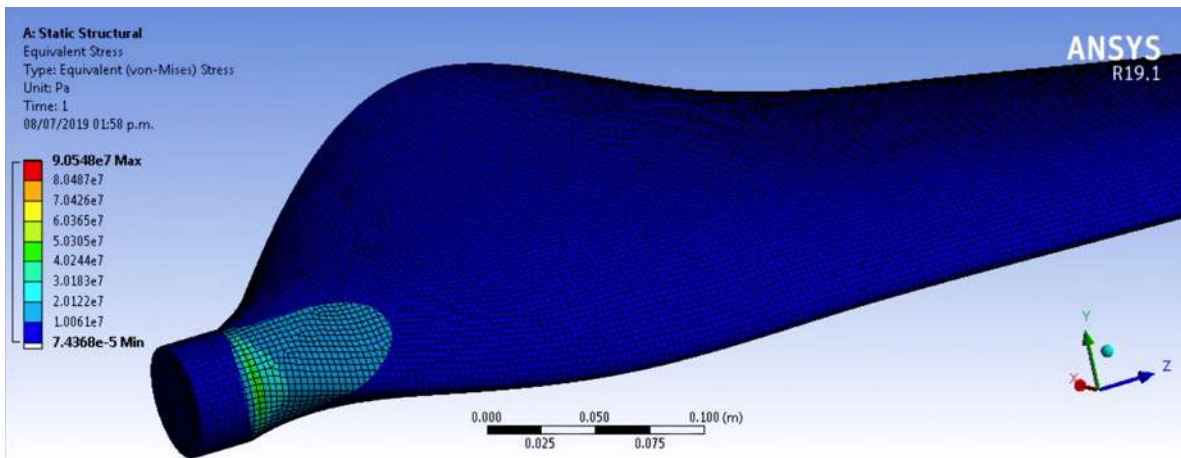


Fig. 17. Simulación FEM de uno de los diseños del aspa realizada en ANSYS.

El esfuerzo de Von Mises indica cuándo falla un elemento estructural por exceder su energía de distorsión. Esta variable se eligió en los resultados del análisis estático [39]. La Fig. 17 muestra los resultados del esfuerzo de Von Mises a partir del análisis de elementos finitos del aspa 1.

De acuerdo con la escala de colores, el esfuerzo de Von Mises tiene un valor de 90.5 MPa en la región de la raíz del aspa 1. La validación de la simulación se puede hacer con un gráfico de la convergencia de la malla con respecto al número de elementos y el esfuerzo de Von Mises equivalente, donde se alcanza un error relativo del 2% como se muestra en la Fig. 18.

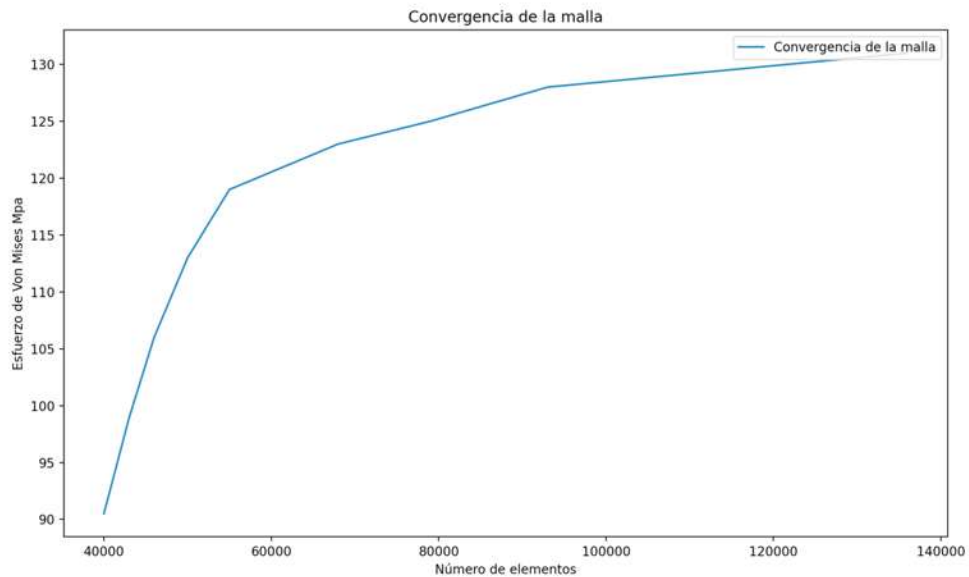


Fig. 18. Grafica de convergencia de la malla del análisis de elemento finito.

Por otro lado, las aspas son estructuras, experimentan cierto grado de vibración, estas vibraciones pueden clasificarse en: libre y forzada. Un sistema bajo vibración libre vibrará en una o más de sus frecuencias naturales. La vibración forzada tiene lugar bajo la excitación de fuerzas externas, cuando la excitación es oscilatoria, el sistema se ve obligado a vibrar a la frecuencia de excitación. Si la frecuencia de excitación coincide con una de las frecuencias naturales del sistema, se encuentra una condición de resonancia y pueden producirse oscilaciones peligrosas. El aspa está sometido a un proceso de fuerzas acopladas, por lo que cuando tenemos la misma frecuencia natural para el aspa y las fuerzas excitantes, se produce la resonancia. El análisis modal se llevó a cabo en el software ANSYS y en la Tabla 2 se muestran los resultados de este estudio. En la Ec.9, se muestra una expresión para calcular la frecuencia natural, la cual relaciona las propiedades de un sistema dinámico establecido por su distribución de masa y rigidez.

$$F_n = \sqrt{\frac{k}{m}} \quad (9)$$

Tabla 2. Frecuencia natural de los modelos de aspas analizados.

| Aspa | Modo | ω_n (Hz) | k (m/kN) |
|-------------|-------------|-----------------|------------|
| 1 | 1 | 4.25 | 148.3 |
| 2 | 1 | 3.41 | 73.4 |
| 3 | 1 | 2.79 | 34.1 |

En nuestro caso, para la simulación y cálculo de las frecuencias naturales, no se tienen en cuenta las cargas externas, ya que el medio debe vibrar libremente para conocer los modos de vibración en los que fallará y conocer su frecuencia antes de aplicar una fuerza externa, así como también se tomó el 1er modo de vibración. Los siguientes trabajos fueron consultados para conocer el comportamiento y análisis con el mismo material y software. Jureczko et al. [40] realizaron un modelo FEM de un aspa con el material GLASS-E en ANSYS con 124,042 elementos, 55044 nodos y 327 áreas de malla, posteriormente realizaron un análisis modal donde se obtuvieron los resultados de la frecuencia natural. Darshil et al. [41] realizaron ensayos de flexión de un aspa con material Glass E de acuerdo con las normas de certificación, en condiciones normales de funcionamiento y en el peor de los casos de carga.

3

DISEÑO ÓPTIMO

El diseño de un sistema puede ser formulado como un problema de optimización, en donde el rendimiento es optimizado mientras todos los demás requisitos son satisfechos. Diferentes técnicas y métodos de optimización han sido desarrollados y usados para diseñar mejores sistemas. Cualquier problema en donde ciertos parámetros necesitan ser determinados para satisfacer las restricciones puede ser formulado como un problema de optimización, en base al criterio de optimización establecido se resuelve el problema.

El diseño es un proceso iterativo, en donde varios diseños son revisados hasta obtener un diseño aceptable o el mejor diseño. En este proceso, el diseñador propone un diseño de prueba basado en la experiencia, la intuición y en análisis matemáticos. En el proceso de optimización, el diseño de prueba se analiza para determinar si es el mejor. Dependiendo de las especificaciones, "el mejor diseño" puede tener diferentes connotaciones para diferentes sistemas, por lo general, implica que un sistema es rentable, eficiente, confiable y duradero.

3.1 Formulación del problema de diseño óptimo

La correcta formulación de un problema de optimización juega un papel importante en el trabajo necesario para resolver el problema, ya que la solución será tan buena como la formulación. Las restricciones son una parte fundamental dentro de la formulación del problema, ya que hace factible una solución dentro del criterio de optimización del problema.

El desarrollo de una formulación adecuada para el diseño óptimo de problemas prácticos es un proceso iterativo, por esta razón es crítico seguir un procedimiento bien definido para la correcta formulación. Los siguientes cinco pasos son usados para formulación del problema de optimización:

1. **Descripción del Proyecto/Problema:** el proceso de formulación comienza desarrollando una declaración descriptiva para el proyecto / problema. La declaración describe los objetivos generales del proyecto y los requisitos que deben cumplirse.

2. **Recopilación de datos e información:** la formulación matemática correcta de un problema, necesita de la recopilación de información como: las propiedades del material, los requisitos de rendimiento, los límites de recursos, el costo de las materias primas, etc.
3. **Definición de variables de diseño:** es la identificación de un conjunto de variables que describen el sistema, denominadas variables de diseño o variables de optimización:
 - Estas se consideran libres para asignarles cualquier valor.
 - El número de variables de diseño independientes otorga al diseño grados de libertad para el problema.
 - Si no se seleccionan las variables de diseño adecuadas para un problema, la formulación será incorrecta o imposible.
 - Las variables de diseño deben ser independientes entre sí en la medida de lo posible. Si no lo son, debe haber algunas restricciones de igualdad entre ellas.
 - Debe existir un número mínimo de variables de diseño para formular correctamente un problema de optimización del diseño.
4. **Criterio de Optimización:** el criterio debe ser una función escalar cuyo valor numérico se puede obtener una vez que el diseño es establecido. Tal criterio se suele denominar función objetivo para el problema de diseño óptimo, y puede maximizarse o minimizarse según los requisitos del problema.
5. **Formulación de restricciones:** las restricciones, deben depender de las variables de diseño, ya que solo entonces cambian sus valores con diferentes diseños de prueba.

3.2. Diseño óptimo contra diseño convencional

En la ingeniería existen dos procesos de diseño: el diseño convencional y el diseño óptimo. En la Fig. 19a se presenta el método convencional de diseño y en la Fig. 19b el método de diseño óptimo. Ambos métodos son iterativos, algunos bloques que requieren de cálculos similares y otros que necesitan diferentes tipos de operaciones. A continuación, se describe los bloques para el proceso de Diseño Óptimo y Diseño Convencional.

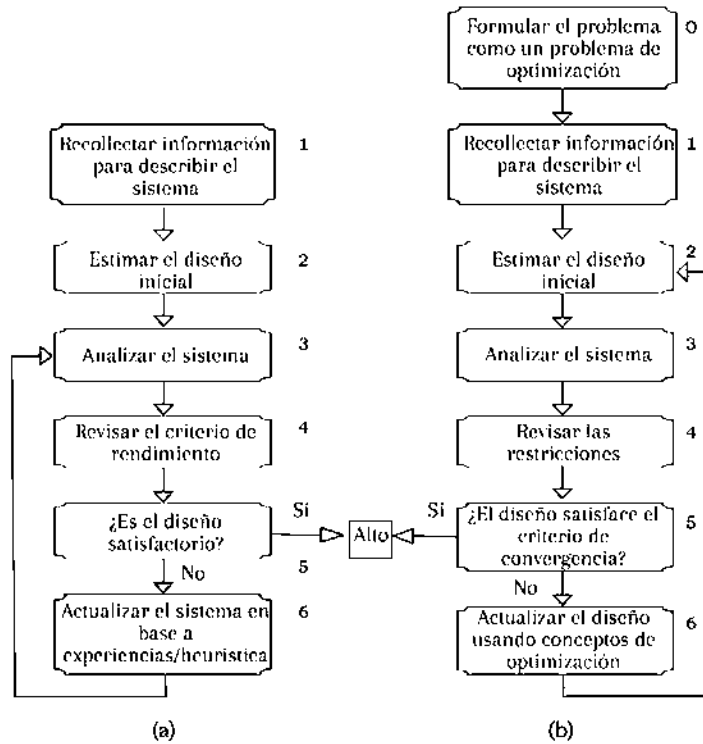


Fig. 19. Comparación entre (a) el diseño convencional y (b) el diseño óptimo.

La Fig.19 muestra las diferencias entre los dos enfoques de diseño, indicando que el proceso de diseño convencional es menos formal, porque no identifica una función objetivo que pueda medir el rendimiento de un diseño. El método de diseño tradicional, actualiza el diseño basándose en la experiencia del diseñador, mientras que el método de diseño óptimo utiliza conceptos y procedimientos de optimización para actualizar el diseño actual [42].

En el diseño óptimo es necesario analizar varios diseños hasta obtener la mejor propuesta, para lograr este análisis, se necesita de un modelo matemático general para describir los conceptos y métodos de optimización. Este modelo se define como la minimización de una función costo en la que se satisfacen todas sus restricciones de igualdad y desigualdad.

En los problemas de diseño se debe tener un criterio de optimización, el cual es usado para comparar varios diseños y determinar cual es el mejor. Se puede definir un modelo matemático general para un diseño óptimo que contemple todas estas posibilidades.

Un vector $n \times x = (x_1, x_2, \dots, x_n)$ de variables de diseño para minimizar la función costo:

$$f(x) = f(x_1, x_2, \dots, x_n) \tag{10}$$

Que este sujeto a las restricciones de igualdad r :

$$h_j(x) = h_j(x_1, x_2, \dots, x_n) = 0; \quad j = 1, \dots, r \quad (11)$$

y las restricciones de desigualdad w :

$$g_i(x) = g_i(x_1, \dots, x_n) \leq 0; \quad i = 1, \dots, w \quad (12)$$

La mayoría de los problemas de diseño también deben satisfacer las restricciones de rendimiento y otras limitaciones. En algunos problemas de diseño solo tienen restricciones de desigualdad, otros solo tienen restricciones de igualdad y algunos tienen ambas. Un conjunto factible para el problema del diseño es una colección de todos los diseños factibles. Matemáticamente, el conjunto R es una colección de puntos de diseño que satisfacen todas las restricciones:

$$R = \{x | h_j(x) = 0, j = 1 \text{ a } r; g_i(x) \leq 0, i = 1, \dots, w\} \quad (13)$$

La región factible se reduce cuando se agregan más restricciones al problema de diseño y se expande cuando se eliminan algunas restricciones. Cuando la región factible se reduce, se reduce el número de diseños posibles que pueden optimizar la función de costo; es decir, hay menos diseños factibles.

3.3. Diseño óptimo del aspa

El diseño óptimo de las aspas de un aerogenerador se logra por medio de la asignación de un criterio de optimización; este criterio debe ser una función escalar, en donde se obtiene un valor numérico cuando el diseño satisface todas las restricciones. En este estudio se han utilizado las siguientes variables de diseño: la velocidad específica, que es una variable operativa, y el esfuerzo de Von Mises, que es una variable estructural; ambas tienen una relación significativa con la masa de los diseños de aspas de aerogeneradores. La velocidad específica es fundamental en los cálculos, ya que es la relación entre la velocidad en la punta del aspa y la velocidad del viento, y para un aerogenerador rápido está se encuentra idealmente en el rango entre 8 y 10 [29], lo que se transforma en una restricción de diseño. Es importante realizar un análisis estructural del aspa, ya que normalmente este tipo de elementos se ve sometido a distintos tipos de fuerzas en su instalación.

El esfuerzo de Von Mises es el producto de dicho análisis del aspa, a cada relación de velocidad de específica le corresponde un valor de esfuerzo, es decir, están relacionadas y ambas tienen restricciones, por lo que son dos variables que pueden formar la capa de entrada de la red neuronal y además constituyen el dominio de las funciones objetivo.

Las variables de salida fueron elegidas por la importancia que representan en el proceso de diseño óptimo, es decir, es importante conocer la masa óptima del aspa, su frecuencia natural y su eficiencia aerodinámica. Es importante mencionar que al ser un diseño óptimo multicriterio, se puede generar una red neuronal con más variables de salida, las elegidas para este caso de estudio. Podemos optimizar cada variable por separado, pensando en una función tridimensional, Fig. 20, es decir, $m = f(\lambda, \sigma)$, $\omega_n = f(\lambda, \sigma)$ y $C_p = f(\lambda, \sigma)$, sin embargo, dado que las redes neuronales aprenden con patrones de datos, y a cada valor del dominio o variables de entrada le corresponde un valor de cada variable de salida (m , ω_n y C_p), se generó una base de datos para el entrenamiento, una vez entrenada la red, es posible calcular los tres parámetros al mismo tiempo.

3.3.1. Variables de diseño

En el aspa, la masa, el coeficiente de potencia y la frecuencia natural son factores importantes que pueden relacionarse con distintas variables. En este estudio, se decidió utilizar las variables de diseño de la velocidad específica y el esfuerzo de Von Mises, ambas conservan una relación significativa con la masa de diseño. En el diseño, una velocidad específica alta da como resultado un aspa más delgada, mientras que una velocidad específica baja da como resultado aspas más pesadas, lo que está estrechamente relacionado con su comportamiento ondulatorio, es decir, la frecuencia natural y con la potencia que puede generar la turbina.

3.3.2. Restricciones

La mayoría de los problemas de diseño están sujetos a restricciones y tienen un conjunto de diseños factibles dentro de una región. Un conjunto factible de soluciones para un diseño óptimo es una colección de propuestas que satisfacen todas las restricciones.

Por lo general, la región factible se reduce cuando se añaden más restricciones al modelo y se amplía cuando se eliminan algunas. Cuando la región factible se reduce, el número de diseños posibles que pueden optimizar la función objetivo también se reduce; es decir, hay menos diseños factibles.

En el proceso de diseño de este trabajo se tuvieron en cuenta las siguientes restricciones:

- Velocidad específica: $5 \leq \lambda \leq 10$.
- Esfuerzo de Von mises: $90.5Mpa \leq \sigma \leq 130Mpa$.

3.4. Técnica de la distancia inversa ponderada

La técnica de la distancia inversa ponderada permite generar más puntos en la función objetivo, por medio del cálculo de alturas en áreas específicas; el objetivo es obtener valores adicionales en las funciones objetivo, para ello se comienza con 3 valores conocidos dentro de los límites y los puntos desconocidos dentro de la función son estimados por medio de la interpolación [29]. Contando con 3 puntos de los diseños iniciales, se pueden interpolar más puntos en el área delimitada por las restricciones de diseño. Este método matemático usa las distancias que hay del punto que se desea conocer, hacia los puntos con valores conocidos, algunos investigadores han empleado también esta técnica en otro tipo de proyectos [43- 44]. Entre más cercanos sean los puntos conocidos de los puntos que se desea encontrar, la aproximación será más exacta. La fórmula para la interpolación inversa ponderada es la siguiente:

$$Z_p = \frac{\sum_{i=1}^n \left(\frac{z_i}{d_i^p} \right)}{\sum_{i=1}^n \left(\frac{1}{d_i^p} \right)} \quad (10)$$

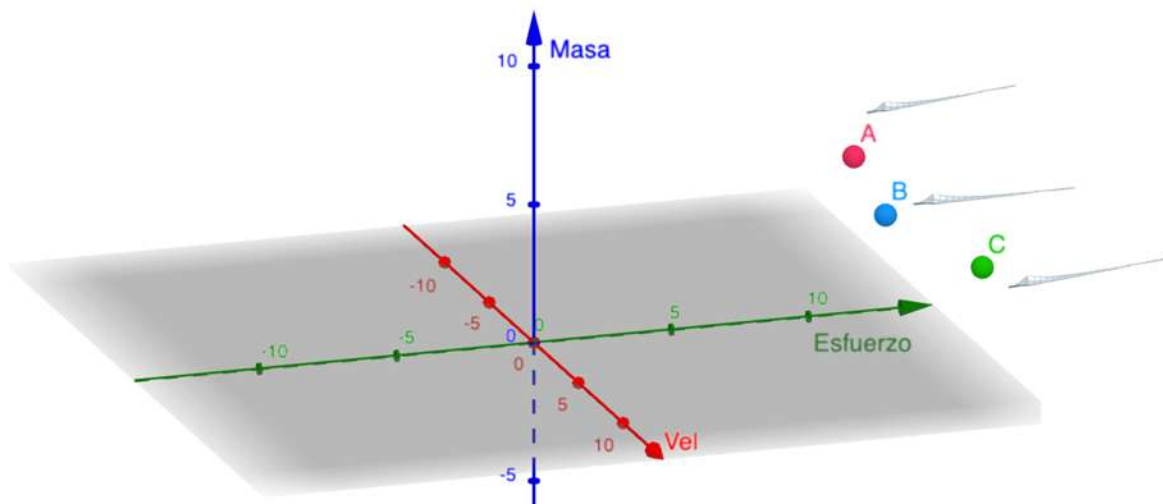


Fig. 20. Puntos de los diseños iniciales.

En la Ec. (10), z_p , es la nueva altura que se desea calcular. En el numerador se tiene una sumatoria de fracciones, de los puntos que son conocidos entre sus distancias al nuevo punto que se desea conocer. En el denominador se tiene otra sumatoria, del cociente de 1 entre las distancias de los 3 puntos conocidos. Usando la técnica de interpolación de la distancia inversa ponderada se obtienen 23 puntos adicionales, añadiendo los primeros 3 puntos que surgen de los diseños de álabes, se tienen en total 26 puntos, los cuales se grafican nuevamente para obtener una nueva función objetivo como muestra en la Fig.21.

Tabla 3. Valores de la función objetivo de la masa.

| Point | x | y | z | Point | x | y | z |
|--------------|-------------|--------------|------------|--------------|----------|----------|----------|
| A | 8.0 | 90.5 | 8.2 | 11 | 9.5 | 130.8 | 4.5 |
| B | 9.0 | 98.8 | 6.3 | 12 | 9.0 | 130.8 | 4.6 |
| C | 10.0 | 130.8 | 4.4 | 13 | 8.5 | 120.7 | 5.7 |
| 1 | 9.0 | 110.6 | 6.3 | 14 | 8.5 | 130.8 | 4.7 |
| 2 | 8.5 | 100.5 | 6.7 | 15 | 8.0 | 120.7 | 5.8 |
| 3 | 8.5 | 110.6 | 6.3 | 16 | 8.0 | 130.8 | 4.8 |
| 4 | 8.0 | 100.5 | 6.7 | 17 | 9.0 | 100.5 | 6.6 |
| 5 | 8.0 | 110.6 | 6.3 | 18 | 10.0 | 120.7 | 5.5 |
| 6 | 8.5 | 90.5 | 8.1 | 19 | 10.0 | 110.6 | 6.2 |
| 7 | 9.0 | 90.5 | 7.9 | 20 | 9.5 | 100.5 | 6.6 |
| 8 | 9.5 | 110.6 | 6.3 | 21 | 10.0 | 100.5 | 6.5 |
| 9 | 9.5 | 120.7 | 5.6 | 22 | 9.5 | 90.5 | 7.8 |
| 10 | 9.0 | 120.7 | 5.7 | 23 | 10.0 | 110.6 | 6.3 |

En la Tabla 3, los tres primeros puntos (A, B, C) son los tres diseños iniciales de álabes, el resto de los puntos fueron calculados utilizando la técnica IDW, Ec. (10).

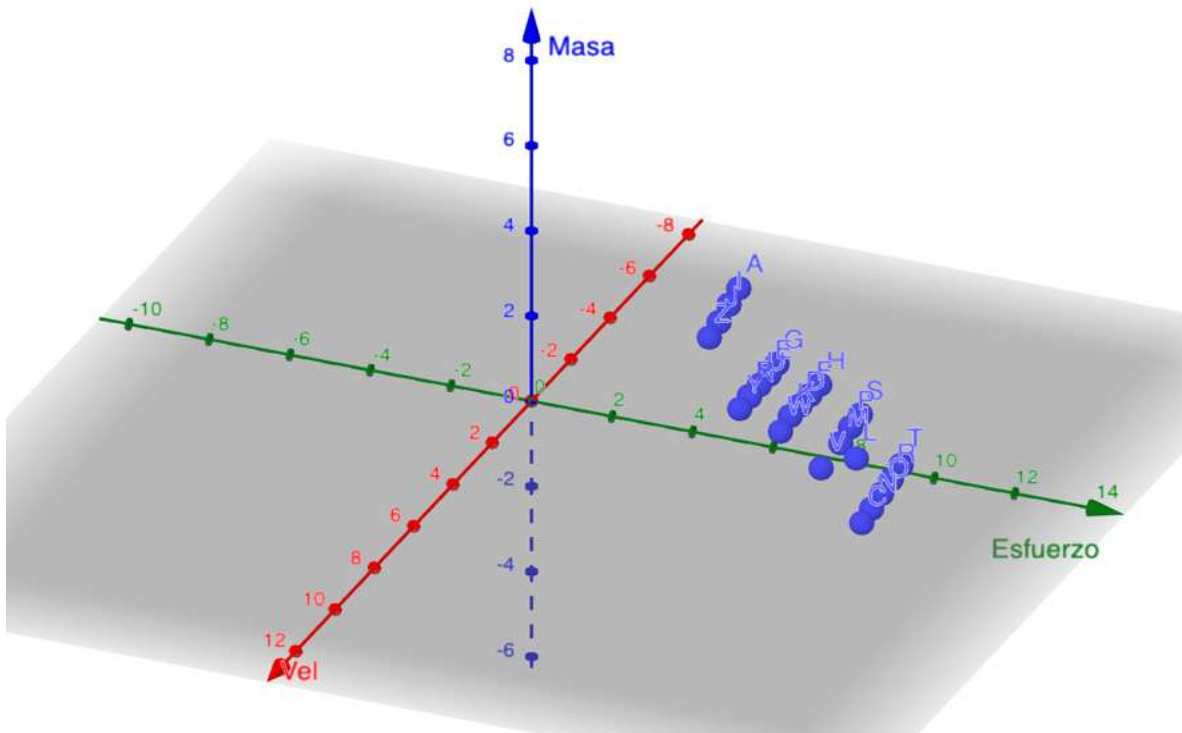


Fig. 21. Función objetivo de la masa con los 26 puntos de la Tabla 3.

3.5. Funciones objetivo

Usando la técnica de la Distancia Inversa Ponderada (IDW) se generaron 3 funciones objetivo; esta técnica permite generar múltiples puntos dentro de las restricciones de diseño a partir de tres diseños iniciales. De esta forma, fueron construidas tres funciones objetivo, todas ellas en el dominio de la velocidad específica y el esfuerzo de Von Mises. Las funciones objetivo, surgen del desarrollo de tres diseños de aspa que representan los puntos iniciales en la función objetivo. Mediante el uso de IDW fue posible generar el resto de los puntos que se ajustan a las funciones objetivo. Se realizó una validación de la precisión de la técnica IDW, se elaboró la Tabla 4 que consta de 3 columnas: en la primera columna están los valores de velocidad específica, en la segunda el rango de valores de esfuerzo obtenidos con IDW y en la tercera los valores obtenidos a partir de la simulación FEM de Qblade. La Tabla 4, muestra cómo la mayoría de los valores obtenidos en la simulación entran dentro del rango de la función objetivo obtenida con IDW. La simulación con una velocidad específica de 10, presenta un comportamiento diferente, con un error del 11,6%, esto es debido a que la simulación FEM en Qblade presenta un margen de error de aproximadamente el 18% [45].

Tabla 4. Validación de la función objetivo de la masa usando Qblade.

| TSR | IDW (Mpa) | QFEM (Mpa) |
|------------|------------------|-------------------|
| 5 | 90-120 | 98.46 |
| 6 | 90-130.8 | 95.1 |
| 7 | 90.5-130.8 | 92.34 |
| 8 | 95-130 | 125.83 |
| 9 | 95-130 | 111.76 |
| 10 | 90.5-130.8 | 145.97 |

Para este caso de estudio, se generaron tres ejercicios de diseño óptimo. En el primero, la variable a optimizar fue la masa; en el segundo, el coeficiente de potencia; y en la tercera, la frecuencia natural. Las variables de diseño fueron el coeficiente de la velocidad específica (λ) y el esfuerzo de Von Mises (σ), que representan el dominio de las funciones objetivo. Puede observarse en la Fig. 22 la función objetivo del coeficiente de la masa, en la Fig. 23 la función objetivo del coeficiente de potencia, y en la Fig. 24 la función objetivo de la frecuencia natural.

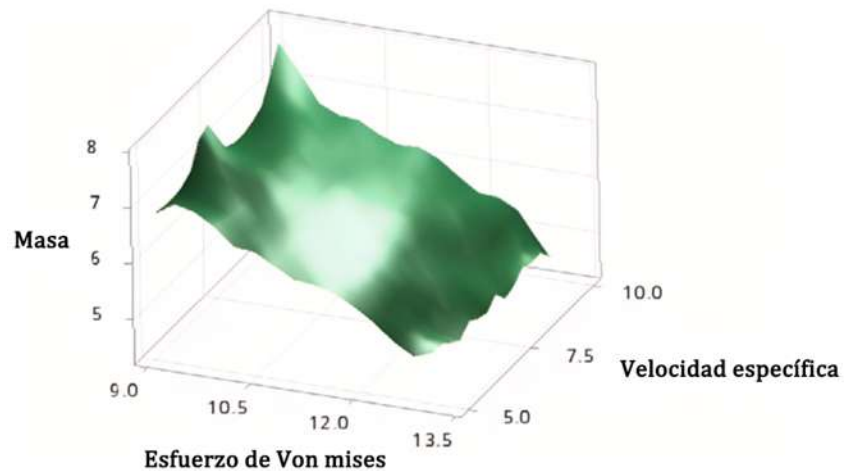


Fig. 22. Función objetivo de la masa.

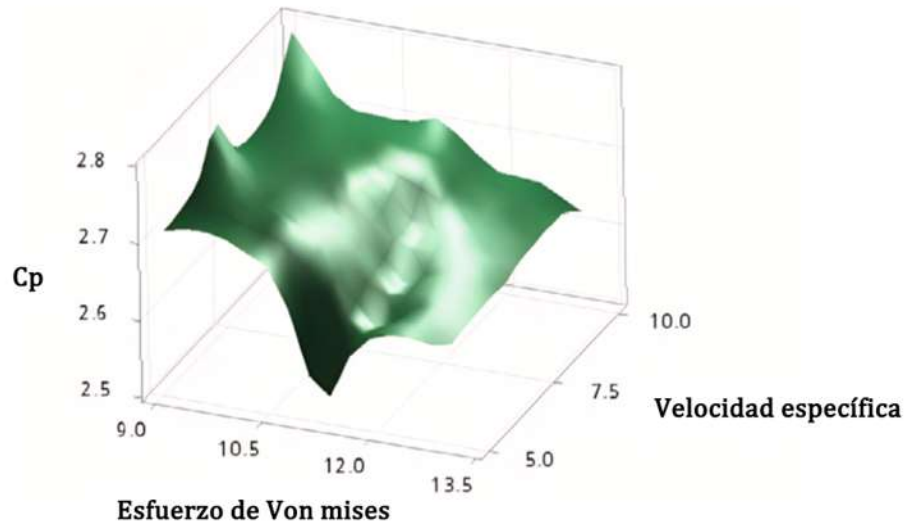


Fig. 23. Función objetivo del coeficiente de potencia.

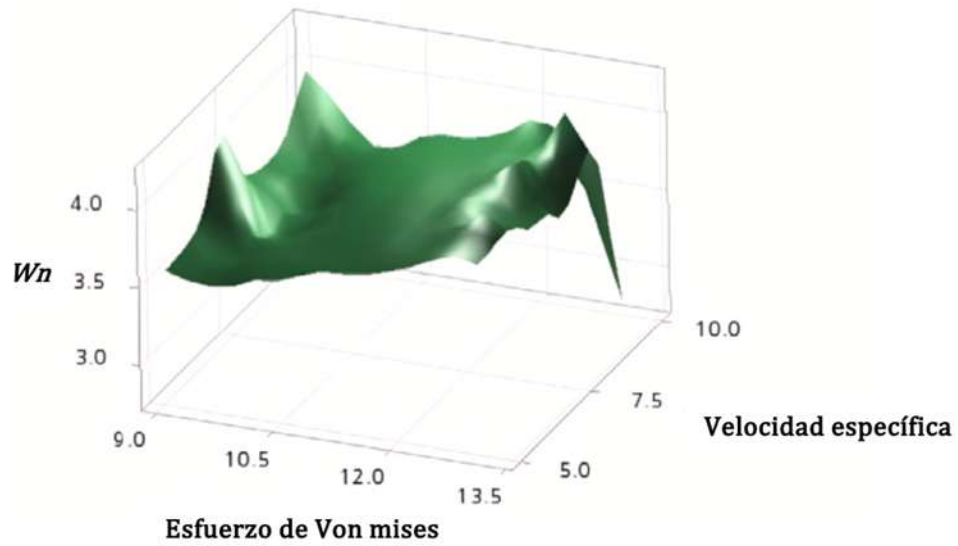


Fig. 24. Función objetivo de la frecuencia natural.

4

INTELIGENCIA ARTIFICIAL

El término Inteligencia Artificial (I.A.) tiene muchas definiciones, algunos autores relacionan el término con los sistemas robóticos, algunos otros lo definen como sistemas capaces de razonar o pensar. Hay personas que definen a la Inteligencia Artificial comparándola con los seres humanos, asumiendo que estos sistemas cuentan con inteligencia propia, también se asocia con la racionalidad y la capacidad de tomar decisiones propias. Una de las mejores definiciones para la Inteligencia Artificial es: el estudio de cómo hacer que los ordenadores hagan cosas que, por ahora, los humanos hacen mejor. El origen de la I.A., tuvo lugar en la universidad privada de Dartmouth College en Estados Unidos, entre una reunión de investigadores. La idea de la I.A., fue tomada con mucho optimismo entre los investigadores que participaron en la reunión de Dartmouth. El objetivo final de la I.A., es evaluado con un test, el cual fue propuesto por Alan Turing en 1950, el cual sirve examinar si un sistema artificial exhibe una conducta inteligente. De acuerdo al test de Turing, una máquina puede pasar la prueba si es capaz de lograr la eficiencia de un ser humano en todas las actividades de tipo cognitivo, incluso con la capacidad de engañar a un evaluador humano, con la incapacidad de determinar si se encuentra hablando con una máquina o con un humano.

En 1955 Allen Newell, Herbert Simon y Cliff Shaw escribieron el programa “Logic Theorist”, el primer programa con Inteligencia Artificial, el cual tenía la capacidad de imitar las habilidades de resolución de problemas de un ser humano para demostrar teoremas. Años después la I.A., reorientaría su camino en la solución de problemas concretos de interés práctico, como, por ejemplo, la búsqueda de caminos de coste mínimo en diferentes aplicaciones, en trayectorias para vehículos autónomos. Peter Hart, Nils Nilsson y Bertram Raphael, del Stanford Research Institute en Menlo Park, California, crearon el algoritmo A*; este algoritmo es el gran exponente de las técnicas de búsqueda heurística [46].

En la actualidad la sociedad ya cuenta con sistemas que usan la I.A., entre los ejemplos se tienen a los juegos de ajedrez, sistemas de reconocimiento de voz, asistentes de digitales, etc., Michelle Zhou divide la evolución de la I.A., en tres etapas. La primera, inteligencia de reconocimiento, en donde las máquinas son capaces de identificar patrones y extraer información de bloques de texto y posiblemente de entender el significado de un documento. En la segunda etapa, la inteligencia cognitiva, las máquinas tienen la capacidad de realizar

inferencias a partir de los datos, es decir, aprender de ellos. La tercera etapa se alcanzaría cuando se tenga la capacidad de crear seres humanos virtuales, que piensen, actúen y se comporten como lo hacemos nosotros. En la actualidad los avances en la I.A., se encuentra en la primera fase, con técnicas de Deep Learning más eficientes y mejores para detectar patrones.

Los primeros desarrollos de la I.A., se enfocaron en la manipulación de símbolos siguiendo reglas lógicas, permitiendo realizar tareas como demostraciones de teoremas matemáticos, rompecabezas, optimizar la distribución de los componentes electrónicos en un circuito integrado. Con la incorporación de las técnicas de aprendizaje automático y el Deep Learning se han conseguido la construcción de sistemas con resultados comparables a los de un ser humano que resuelve los mismos problemas [46].

4.1. Aprendizaje Automático

El aprendizaje automático, también conocido como “*machine learning*”, es una técnica por medio de la cual, es posible que un ordenador aprenda de manera independiente a resolver un problema. Se diseña un algoritmo para el problema, pero es el ordenador que se encarga de resolver el problema, usando los datos disponibles y de las técnicas de aprendizaje impuestas en el algoritmo de aprendizaje.

Un elemento muy importante de la inteligencia artificial es el aprendizaje automático, que se basa principalmente en el refinamiento de la habilidad para resolver problemas, y la adquisición de conocimiento por medio de la experiencia. Dicha experiencia es adquirida a partir de los datos que recibe el modelo, y mediante un entrenamiento, que le permitirá al sistema hacer predicciones de datos futuros. Dentro de los problemas en los que se puede emplear el uso del aprendizaje automático, se encuentra el reconocimiento de patrones, como los sistemas de voz, identificación de objetos en imágenes, etc. Problemas como los mencionados anteriormente, brinda una gran cantidad de variables para analizar, teniendo al final un número de datos enorme, lo cual sería una tarea imposible de llevar a cabo con los métodos convencionales. Pero con el aprendizaje automático y las técnicas de aprendizaje es

posible diseñar un algoritmo que aprenda de los datos disponibles y así que crea un programa para resolver el problema [46].

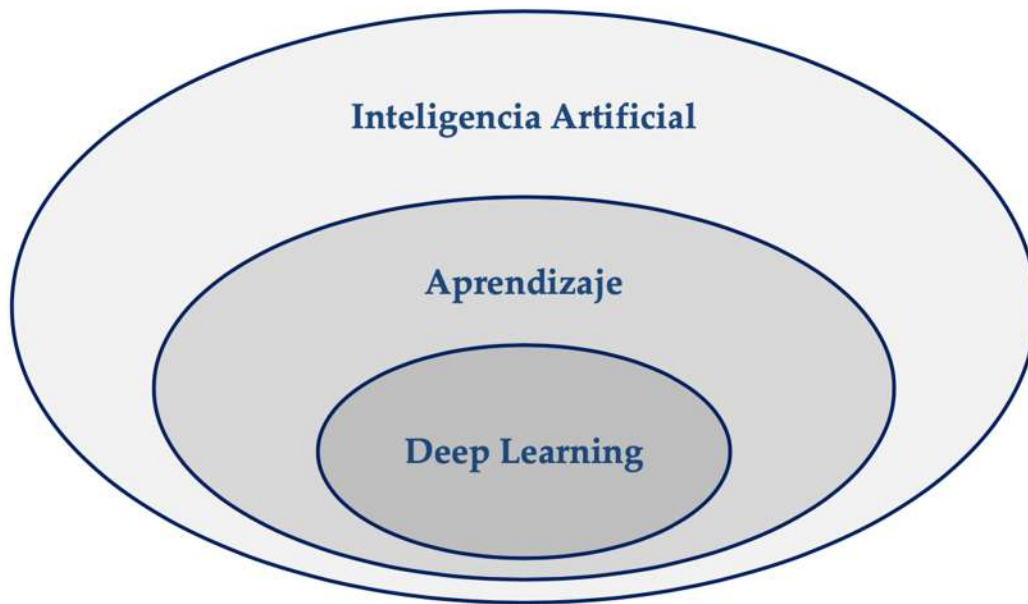


Fig. 25. Deep learning como un subconjunto de técnicas de aprendizaje automático. (Berzal, F. (2018). Redes neuronales & deep learning. Granada: Fernando Berzal.).

4.2. Redes Neuronales Artificiales

Dentro de las técnicas de aprendizaje automático, existen una variedad de diferentes ramas, pero entre ellas las redes neuronales son las que en los últimos años han marcado un progreso interesante en los problemas de aprendizaje automático, hasta la fecha, han sido las únicas capaces de alcanzar una eficacia comparable con la del ser humano en diferentes problemas.

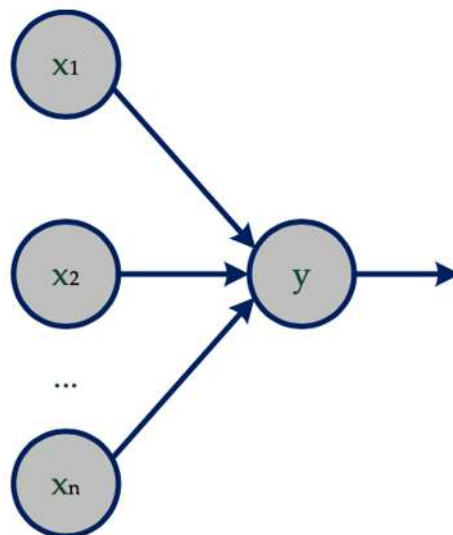


Fig. 26. Red neuronal artificial simple. (Berzal, F. (2018). Redes neuronales & deep learning. Granada: Fernando Berzal.).

Las redes se basan en la anatomía del cerebro humano como modelo, actualmente a las redes neuronales artificiales se les asocia con el término de deep learning. Una red neuronal multicapa, también se le asocia con un aproximador de funciones, que tiene la capacidad de aprender sobre cualquier base de datos. El entrenamiento de una red neuronal usa un algoritmo denominado *backpropagation*, el cual propaga los errores hacia atrás, usa el error observado en la salida de la red, para ajustar los parámetros que regulan la salida de la red, reduciendo el error en la red neuronal. Las redes neuronales tienen la capacidad de

generalizar a partir de ejemplos o bases de datos, sin la necesidad de formalizar el conocimiento adquirido, es por ello que se les asocia con cajas negras, por la gran dificultad de interpretar el funcionamiento interno de estas.

Para diseñar una red neuronal, primero se establece su estructura, luego se entrena usando el algoritmo de *backpropagation*, posteriormente se establecen los parámetros de la red. En una red neuronal pequeña, que contiene una cantidad de parámetro menor, su entrenamiento resulta más simple, pero un número de parámetros bajo en el modelo, lo hace propenso al infraaprendizaje, es decir que la red no cuenta con el número de parámetros suficientes para modelar la complejidad del problema. Por el otro lado, una red neuronal grande, incluye un número elevado de parámetros, haciendo su entrenamiento más prolongado, si el número parámetros es muy grande, la red es propensa al sobreaprendizaje, lo cual se puede evitar con ciertos mecanismos de regularización [46].

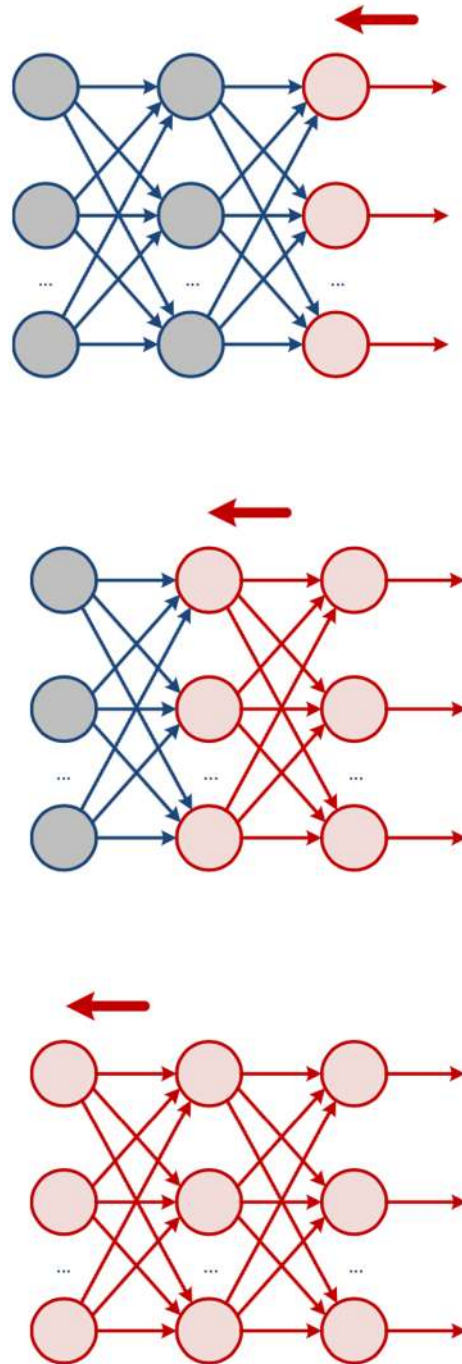


Fig. 27. Cálculo del gradiente con propogación de errores hacia atrás. (Berzal, F. (2018). Redes neuronales & deep learning. Granada: Fernando Berzal.).

4.3. Aprendizaje Profundo (Deep Learning)

Las redes neuronales, también conocidas con el término Deep Learning, es uno de los mecanismos con el que se puede conseguir que una máquina aprenda. Este modelo informático está inspirado en el cerebro humano, el modelo esta formado por múltiples unidades denominadas neuronas. Estas neuronas, se conectan entre sí para formar redes neuronales artificiales, las cuales son manipuladas a través de sus conexiones para conseguir el comportamiento deseado.

Las redes neuronales representan conceptos mediante patrones de actividad en una red de neuronas. Comparando los modelos simbólicos, que representan la forma explícita el conocimiento mediante reglas lógicas, las redes neuronales representan el conocimiento de forma implícita, a través del peso con el que son modeladas las conexiones entre las neuronas. El procesamiento subsimbólico de datos típico de las redes neuronales, dota a las redes de una capacidad automática de generalización, el cual es muy útil para la resolución de problemas complejos de aprendizaje.

Al momento de entrenar una red neuronal, el ajuste de los pesos de las conexiones para aprender un nuevo concepto, afecta la representación distribuida de otros conceptos que tienen patrones de actividad similares. Al usar una red neuronal, ante patrones con actividades similares, pero no iguales a los vistos por la red durante su fase de entrenamiento, la red tiene la capacidad de generalizar con éxito a diferencia de otras técnicas de aprendizaje automático que fracasan ante estos casos. Las técnicas de Deep Learning son un subconjunto dentro de un subconjunto. Las cuales forman parte del conjunto de técnicas de aprendizaje automático, que a su vez son un subconjunto de las técnicas utilizadas en Inteligencia Artificial.

Las técnicas de Deep Learning tienen la capacidad de descubrir y crear nuevas características a partir de los datos. Esta técnica también tiene la capacidad de aumentar los niveles de abstracción con los que se trabaja, con la creación de jerarquías con varios niveles de características y descubrir nuevas características de manera automática a partir de las características del nivel anterior [46].

4.4. Características clave del deep learning

Las redes neuronales poseen la gran capacidad de resolver problemas que ningún programador ha podido resolver con la ayuda de una computadora. La mayoría de los algoritmos utilizados en el deep learning, constan de una serie de pasos para resolver algún problema de aprendizaje automático, los pasos son:

- Tener una base de datos relacionados al problema.

- Construir una función de coste de acuerdo al problema.
- Escoger el modelo de red neuronal y establecer sus parámetros.
- Aplicar un algoritmo de optimización para minimizar la función de coste modificando los parámetros de la red.

El deep learning ha tomado una presencia más fuerte en el aprendizaje automático, existen 3 causas principales que han logrado que el deep learning sea más popular en los últimos años. Los factores son: la disponibilidad de datos, que en la actualidad el conjunto de datos disponibles ha aumentado de manera exponencial con el paso de los años. Disponer de una base de datos en la actualidad es posible gracias a los avances tecnológicos. La potencia de cálculo, el avance tecnológico en los procesadores de computadores ha tenido grandes resultados mejorando notablemente el rendimiento de los ordenadores. Por último, están los algoritmos, que son las técnicas encargadas de entrenar a las redes neuronales profundas [46].

4.4.1. Limitaciones del deep learning

Las primeras limitaciones que tuvo el *deep learning* eran problemas prácticos, los ordenadores no eran lo suficientemente potentes y los conjuntos de datos no era los suficientes para alimentar a este tipo de redes neuronales. Pero en la actualidad estos problemas prácticos han quedado atrás, pero aun en la actualidad siguen persistiendo un par de problemas. El sobre aprendizaje es uno de los problemas más recurrentes en los modelos de aprendizaje automático, el segundo problema es su semejanza a cajas negras, ya que no se sabe con certeza que ocurre dentro de las redes neuronales y que es un aspecto que puede tener implicaciones con respecto a la seguridad de los sistemas que pueden llegar a usar esta clase de sistemas.

4.5. Métricas de evaluación

El empleo de las redes neuronales en la resolución de problemas, es un proceso que usualmente este sujeto a errores, esto debido a que en las redes neuronales artificiales se usan modelos, los cuales son una simplificación de la realidad. Es posible evaluar la calidad del modelo y verificar su credibilidad y determinar si es útil en la práctica.

Con las redes neuronales se pueden resolver problemas de clasificación y problemas de regresión. El enfoque de este trabajo es el de resolver problemas de regresión y hacer predicciones. Para medir la exactitud de estas predicciones se ejecutan sobre el modelo medidas de error.

4.5.1 Error cuadrático medio

El error cuadrático medio (MSE) es usado para conocer la calidad de un modelo de predicción, su fórmula es la siguiente:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y})^2 \quad (1)$$

Esta medida de error nos permite saber la precisión del modelo, conocer los errores cometidos por el modelo en las predicciones. El problema que se presenta al usar una medida de error al cuadrado, es que crea sensibilidad en datos con anomalías, por lo que los errores serán más grandes.

4.5.2 Error Absoluto Medio

El error absoluto medio (MAE) es más simple que el error cuadrático medio (MSE), siendo solamente la media de las diferencias, en valor absoluto, entre las predicciones y las observaciones. El error absoluto medio se define de la siguiente forma:

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}| \quad (2)$$

El error absoluto medio es más robusto con datos que contienen ruido y valores con anomalías, ya que cada uno de los errores es proporcional al valor absoluto del error.

4.6 Modelo de redes neuronales artificiales

En la inteligencia artificial y el *deep learning*, es de interés construir modelos computacionales que sean lo más eficientes posibles.

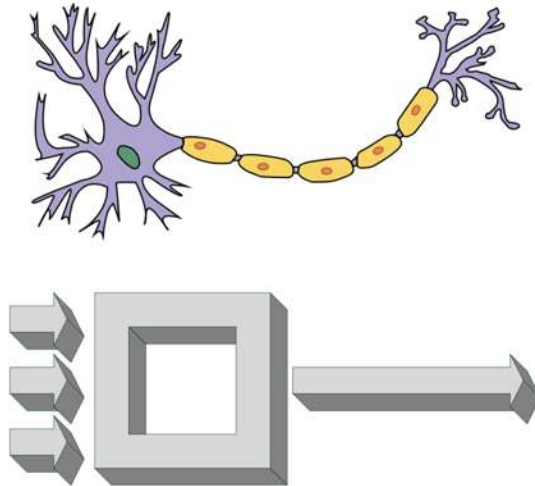


Fig. 28. Modelo abstracto de una neurona artificial. (Berzal, F. (2018). Redes neuronales & deep learning. Granada: Fernando Berzal.).

En las redes neuronales artificiales, en los modelos computacionales, se enfoca en como las entradas afectan el comportamiento de las neuronas. En la entrada de cada neurona se pueden tener varias señales que se combinan y dependiendo de estas se puede tener una activación de la neurona, y por consecuencia un valor determinado en la salida de la neurona, por ello la manipulación de los pesos asociados a las conexiones de la neurona se vuelve una tarea crucial en el entrenamiento de las redes neuronales artificiales [46].

4.6.1. Neuronas

Cada neurona es considerada como un nodo, entre las neuronas existen conexiones o enlaces por donde las señales son transmitidas. La activación de una neurona humana puede depender de diferentes factores. Pero en el modelado de redes neuronales artificiales, estos factores pueden modelarse mediante pesos, los cuales tienen un valor positivo, si la sinapsis es positiva se considera excitatoria y negativa si la sinapsis es inhibitoria. Los modelos de neuronas que son usados en las redes neuronales artificiales no contemplan todos los aspectos biológicos de sus semejantes biológicas, pero existe un modelo clásico que es tomado como el punto de partida, el de modelo de Hodgkin y Huxley, este modelo, describe a detalle el funcionamiento del axón de un calamar gigante. Inspirando nuevas formas de representar el funcionamiento del axón, se construyeron modelos de circuitos eléctricos que trataban de emular el mismo funcionamiento. Pero en la inteligencia artificial, se puede trabajar con modelos que tienen un nivel de abstracción más grande, omitiendo los canales de iones, de esta forma se puede obtener un modelo de neurona estándar, Ec. (11), también conocido como el “modelo de integración y disparo con pérdidas”.

$$C_M \frac{dV_j}{dt} = -g_{perdida}(V_j - E_{perdida}) + g_{int} \sum_i^n x_i w_{ij} \quad (11)$$

En el modelo de arriba, V_j , es el potencial de la membrana para la neurona j -ésima, $g_{perdida}$ es el factor de pérdida, g_{int} es la constante de integración, x_i que es la salida actual de la neurona i -ésima y w_{ij} , es el peso de las conexiones de cada neurona. Se puede asumir que $E_{leak} = 0$ y anulando los factores de pérdida y de integración, se tiene la siguiente expresión:

$$C_M \frac{dV_j}{dt} = -V_j + \sum_{i=1}^n x_i w_{ij} \quad (12)$$

Un modelo más simplificado se logra al quitar la corriente de pérdida ($-V_j$), el modelo de integración y disparo. En este modelo las neuronas pueden saturarse, ya que el modelo acumula la carga sin perderla, por lo que las neuronas siempre generan la misma salida. Sin la corriente de pérdida, se tiene un modelo de neurona sin memoria, lo que significa que su salida solo dependerá exclusivamente de la suma ponderada de los pesos y las entradas de la neurona.

$$V_j = \sum_{i=1}^n x_i w_{ij} \quad (13)$$

Este es el modelo que es usado con más frecuencia en Inteligencia Artificial para la construcción de redes neuronales artificiales.

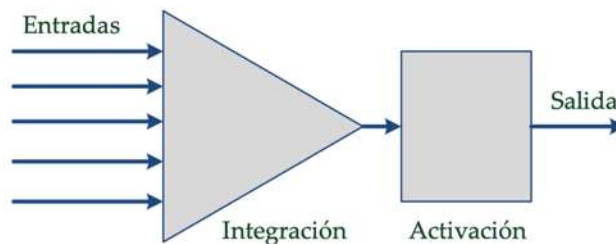


Fig.29. Modelo simplificado de neurona con fase de integración y disparo. (Berzal, F. (2018). Redes neuronales & deep learning. Granada: Fernando Berzal.).

El modelo simplificado de la Fig. 29, se conforma de dos secciones, en la primera sección, las entradas provenientes de otras neuronas son combinadas respetando los pesos correspondientes de cada una de ellas. Como resultado de la primera sección es la entrada neta de la neurona. En la segunda sección, la entrada neta es usada en la salida de la neurona,

la cual se propagará a otras neuronas. La generación de la salida de la neurona depende en gran parte de una función de activación, el cual se logran con una transformación no lineal en la entrada neta. La transformación tiene que ser no lineal, ya que en redes neuronales de múltiples capas se buscan realizar cálculos más complejos, para llegar a un modelo más complejo, se tiene el modelo estándar de una neurona artificial. Los pesos tienen valores reales positivos que sirven para modelar las conexiones excitatorias y negativos para modelar las conexiones inhibitorias.

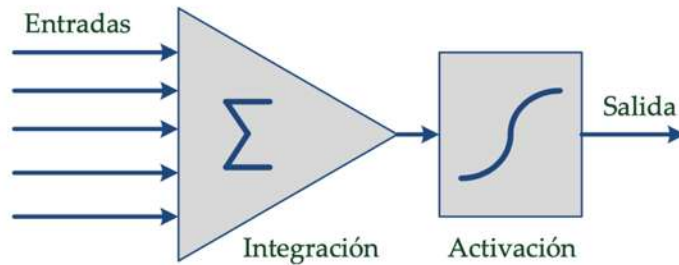


Fig. 30. Modelo estándar de neurona artificial con función de activación. (Berzal, F. (2018). Redes neuronales & deep learning. Granada: Fernando Berzal.).

En algunos modelos de neuronas, el sesgo sirve para conocer el umbral de activación de la neurona, es decir, el punto del cual parte la neurona para activar su salida. Si el nivel de excitación de la neurona se encuentra por debajo del umbral, su salida se mantiene inactiva. Si el nivel de excitación supera el umbral de activación, se activa la salida de la neurona [46].

4.7. Funciones de activación

Las neuronas dentro de una red artificial, tienen diferentes conexiones, las cuales permiten la interconexión entre las neuronas. En la entrada de las neuronas de la capa intermedia, se encuentran la combinación de entradas o la entrada neta, a la cual se le aplica una función de transferencia o de activación. Esta función de activación es la que se encarga de determinar cual es la salida de la neurona.

4.7.1 Función de activación lineal

La función de activación lineal es la función más sencilla, se expresa de la siguiente forma:

$$y = f_{lin}(z) = z \quad (14)$$

Al aplicar la función de transformación lineal a la entrada neta se tendrá como resultado una neurona lineal, las neuronas lineales se comportan como simples clasificadoras lineales, ya

que en serie son el equivalente a una sola capa de neuronas lineales, por lo que no resultan especialmente útiles para resolver problemas más complejos.

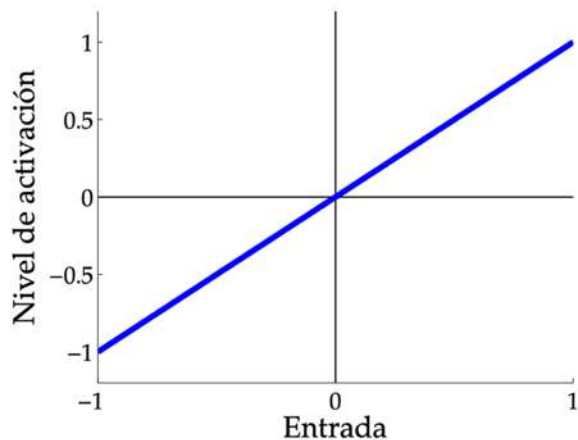


Fig. 31. Grafica de una función de activación lineal. (Berzal, F. (2018). Redes neuronales & deep learning. Granada: Fernando Berzal.).

4.7.2. Función escalón

La función escalón fue una de las primeras funciones no lineales usada para modelar redes neuronales artificiales. Es una función de activación binaria en su implementación en redes neuronales.

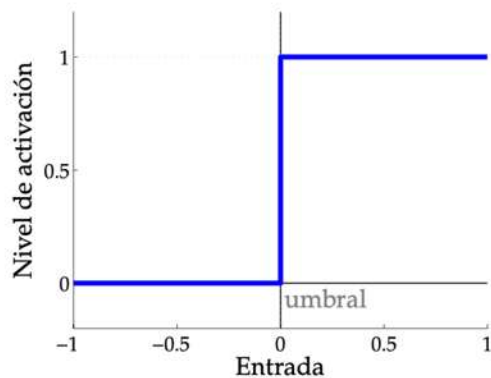


Fig. 32. Grafica de una función de activación de escalón. (Berzal, F. (2018). Redes neuronales & deep learning. Granada: Fernando Berzal.).

4.7.3. Función sigmoïdal

La función de activación sigmoïdal es una función no lineal, creciente, continua y derivable. Un requisito que es importante en las funciones de activación de una neurona, y que son entrenadas usando el algoritmo *backpropagation*. Este algoritmo propaga los errores observados en la capa de salida hacia atrás en la red, para modificar los parámetros internos de la red. La función sigmoïdal logística:

$$y = F_{log}(z) = \sigma(z) = \frac{1}{1 + e^{-z}} \quad (15)$$

La función logística es simétrica:

$$\sigma(-z) = 1 - \sigma(z) \quad (16)$$

La derivada de la función logística es:

$$\frac{d\sigma(z)}{dz} = \frac{d}{dz} \left[\frac{1}{1 + e^{-z}} \right] \quad (17)$$

$$= \frac{e^{-z}}{(1 + e^{-z})^2} = \frac{1}{1 + e^{-z}} \frac{e^{-z}}{1 + e^{-z}} \quad (18)$$

Se tiene:

$$\frac{d\sigma(z)}{dz} = \sigma(z)(1 - \sigma(z)) \quad (19)$$

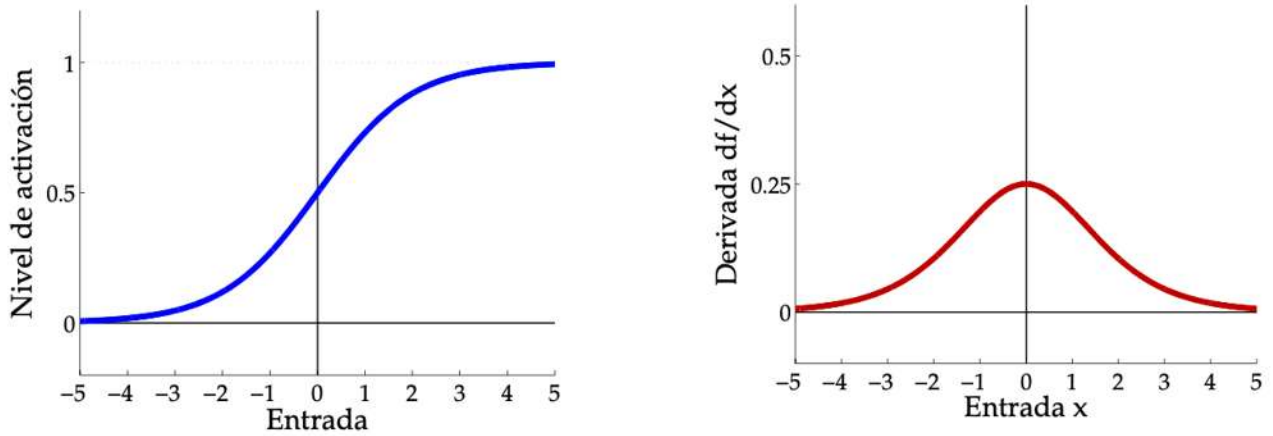


Fig. 33. Grafica de una función de activación de sigmoidal (azul). Grafica de la derivada de la función sigmoidal (rojo). (Berzal, F. (2018). Redes neuronales & deep learning. Granada: Fernando Berzal.).

4.7.4. Función de activación lineal rectificada

Las funciones lineales rectificadas ofrecen una ventaja sobre las funciones sigmoidales en modelos entrenados con backpropagation; debido a que las funciones sigmoidales tienden a saturarse a partir de ciertos valores, provocan que su derivada sea nula, lo cual provoca que el cálculo del gradiente del error se vea afectado, ese gradiente es el que le dice al modelo como modificar los pesos de la red, es decir que el entrenamiento de la red será muy lento.

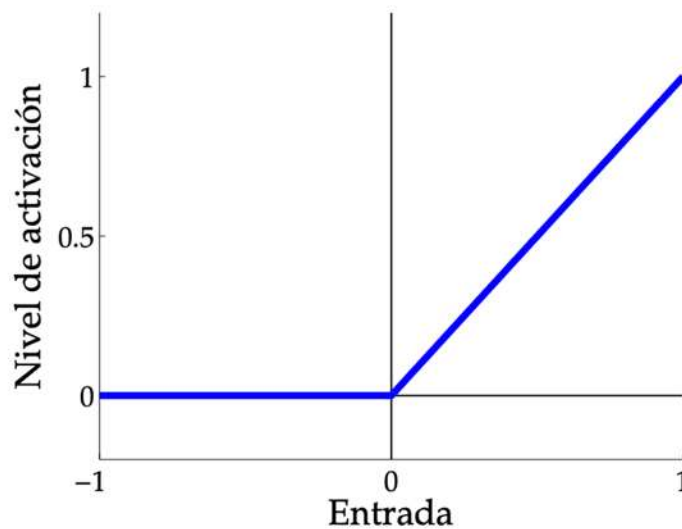


Fig. 34. Grafica de una función de activación lineal rectificada. (Berzal, F. (2018). Redes neuronales & deep learning. Granada: Fernando Berzal.).

4.8. Arquitectura de las redes neuronales

El uso de redes neuronales artificiales con múltiples capas se ha vuelto más común por su gran capacidad para resolver problemas complejos. Por lo general las capas de una red neuronal se pueden dividir en dos: capas visibles y capas ocultas. Las capas visibles normalmente son las capas de entrada y de salida. Las capas ocultas se encuentran entre las capas de entrada y salida de la red.

4.8.1. Redes simples, con una única capa

Son las redes neuronales más simples, solo se tienen a las neuronas de la capa de entrada que se encargan de recibir las señales externas, para después redistribuir la información a las neuronas de la capa de salida.

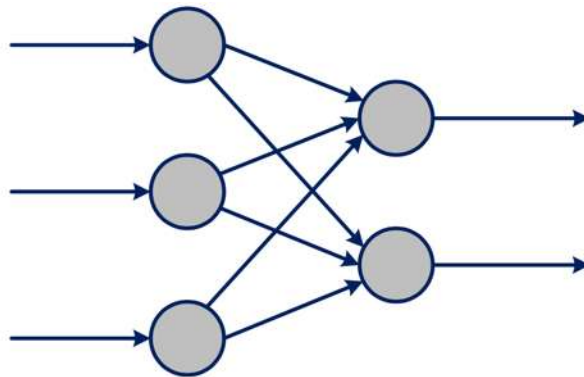


Fig. 35. Red neuronal simple sin capas ocultas. (Berzal, F. (2018). *Redes neuronales & deep learning*. Granada: Fernando Berzal.).

4.8.2. Redes multicapa, con una capa oculta

Las redes multicapa simples cuentan con una capa de entrada y de salida visibles. Pero cuenta con una capa oculta intermedia. Cuando una red neuronal artificial, posee al menos una capa oculta, esta ya posee la capacidad de ser un aproximador de funciones.

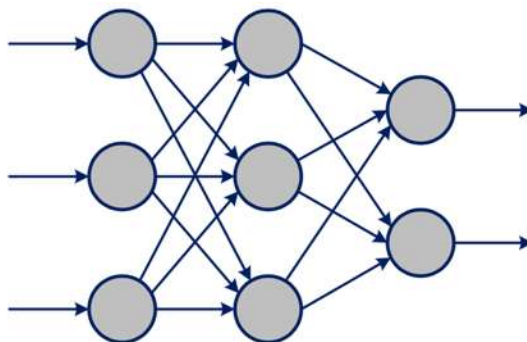


Fig. 36. Red neuronal multicapa con una capa oculta. (Berzal, F. (2018). *Redes neuronales & deep learning*. Granada: Fernando Berzal.).

4.8.3. Redes profundas, con varias capas ocultas

Solucionar problemas muy complejos con las redes neuronales artificiales requiere de emplear las técnicas de *deep learning*, para ello en la arquitectura de la red es necesario incluir múltiples capas ocultas.

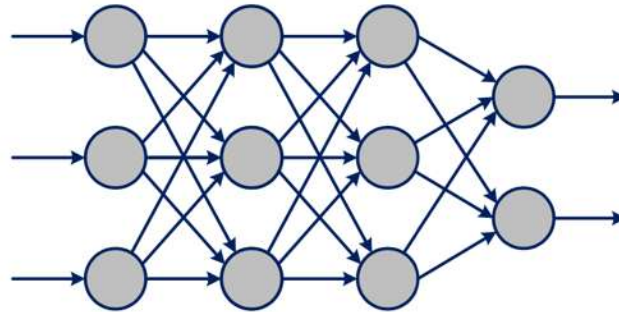


Fig. 37. Red neuronal profunda con múltiples capas ocultas. (Berzal, F. (2018). *Redes neuronales & deep learning*. Granada: Fernando Berzal.).

5

DEEP LEARNING APLICADO AL DISEÑO ÓPTIMO DE LAS ASPAS DE UNA TURBINA DE VIENTO

Se implementó un modelo DL con tres capas ocultas. El modelo realiza las siguientes etapas antes de realizar una estimación: entrenamiento, compilación, ajuste y evaluación. El vector de entrada de la red estaba compuesto por la velocidad específica, λ , y el esfuerzo de Von Mises, σ ; en el vector de salida se tiene: la masa del aspa, m , la frecuencia natural, ω_n ; y el coeficiente de potencia del rotor, C_p . Las RNA más utilizadas son las CNN, las RNN y las MLP. Las CNN ayudan a identificar patrones de imágenes para reconocer objetos, clases y categorías. Las RNN se utilizan en problemas de series temporales, traducción de idiomas y reconocimiento de voz. Los MLP resuelven problemas de asociación de patrones, segmentación de imágenes, compresión de datos, etc. [47]. Para este caso de estudio presentado, se aborda desde la perspectiva de la asociación de patrones, ya que las bases de datos utilizadas para entrenar la red se generan a partir de las funciones objetivo creadas con el proceso de diseño óptimo, análisis estructural y distancia media ponderada, que asocian las variables de entrada (tensión y relación de velocidad de punta) con las variables de salida (frecuencia natural, coeficiente de potencia y masa).

Por estas razones, se eligieron los MLP, que son relativamente fáciles de implementar. El entrenamiento de un MLP se realiza mediante los algoritmos de retropropagación y descenso de gradiente, que permiten calcular el gradiente de error relativo a los distintos parámetros de la red y realizar ajustes en ellos. El gradiente se calcula para una función de error, también conocida como función de pérdida, y el método de optimización ajusta los pesos de la red para minimizar esa función de pérdida [46].

5.1. Base de datos generada a partir de funciones objetivo

Los datos se generaron con ayuda de funciones objetivo. Los valores seleccionados están dentro de los límites de las restricciones de diseño. Como resultado, se generaron 3000 valores de masa, frecuencia natural y coeficiente de potencia. Los datos disponibles para un modelo de aprendizaje profundo pueden ser considerables; idealmente, cuantos más datos

estén disponibles, mejor se podrá entrenar el modelo [46]. Las funciones objetivo crean un conjunto de datos para el entrenamiento, la validación y la prueba del modelo. Las condiciones que predice la red son las condiciones de las funciones objetivo y sus restricciones, por lo tanto, para conocer la suficiencia de los algoritmos utilizados en la red neuronal, la muestra que es utilizada para el entrenamiento del modelo de DL, se divide en un conjunto de datos de entrenamiento, un conjunto de datos de validación y un conjunto de datos de prueba. En estos datos de prueba, la adecuación de la predicción se mide a través del error de ajuste, en este caso el error medio absoluto (MAE), véanse las Tablas 5-8. Este problema puede evitarse dividiendo adecuadamente los datos en un conjunto de entrenamiento, validación y prueba [46]. En este estudio, en el que se generaron 3000 datos a partir de las funciones objetivo, el 67% se utilizó para el ajuste del modelo, y el 33% restante se utilizó para la evaluación.

5.2. Diseño óptimo del modelo DL

El modelo DL cumple los criterios de optimización al estimar la masa, el coeficiente de potencia y la frecuencia natural dentro de las restricciones de diseño óptimas. Las estimaciones tienen una precisión de entre el 90% y el 98%. TensorFlow, desarrollado por Google, se utilizó como plataforma para crear un modelo de aprendizaje profundo en el que se usó Python como lenguaje de programación [48]. TensorFlow puede manejar grandes conjuntos de datos en forma de matrices multidimensionales o tensores, que se mueven entre las diferentes capas de la red. En este estudio se utilizó el algoritmo de optimización ADAM, una extensión del descenso de gradiente estocástico, en lugar del procedimiento clásico de descenso de gradiente para actualizar los pesos de la red [49]. Para simular el modelo de aprendizaje profundo se utilizaron los siguientes programas y librerías: Pycharm versión 2020.2.5, Python 3.7, Keras 2.3.1, y Tensorflow 2.0.3.

5.3. Implementación del modelo de aprendizaje profundo

A continuación se describe la implementación del modelo de deep learning:

1. **Definición del modelo.** En el presente estudio se implementó un modelo secuencial para definir el número de entradas y salidas de la red, el número de capas intermedias, así como el número de neuronas en cada capa. La arquitectura definió el número de entradas y salidas que tendrá el modelo, a partir de los datos de los que se componen las funciones objetivo: el ratio de velocidad de punta (λ), las tensiones de Von Mises (σ), la masa de la pala (m), la frecuencia natural (ω_n) y el coeficiente de potencia del rotor (CP). Se probaron varias configuraciones de arquitectura de red; en estas pruebas se modificó el número de capas y nodos. La Fig. 38 muestra la arquitectura final del modelo de aprendizaje profundo. Se aplicó una función de transferencia o

activación a la entrada de las neuronas de la capa oculta o a la entrada de la red. En este estudio, la función de activación seleccionada fue la función lineal rectificadora porque ofrece una ventaja sobre las funciones sigmoideas en modelos entrenados con retropropagación. Dado que las funciones sigmoideas tienden a saturarse a partir de ciertos valores, lo que afecta al cálculo del gradiente de error, el entrenamiento de la red será más lento [29].

2. **Compilación del modelo.** En la compilación del modelo se eligió la función de pérdida MAE, ya que en este estudio se aborda un problema de regresión porque el objetivo es hacer predicciones. Este error es más robusto con datos que contienen ruido y valores que pueden presentar anomalías. La función de pérdida se optimizó con el algoritmo de optimización ADAM. Esta extensión del descenso de gradiente estocástico puede utilizarse en lugar del procedimiento clásico de descenso de gradiente para actualizar los pesos de la red [46, 50]. El uso de un DL en la resolución de problemas es un proceso que suele estar sujeto a errores. Por esta razón, la métrica del error cuadrático medio (MSE) se integró en el proceso de compilación. Esta métrica ayuda a guiar el proceso de aprendizaje, evaluar la calidad del modelo, verificar su credibilidad y determinar si puede ser útil en la práctica.
3. **Ajuste de modelos.** En el ajuste de modelos, el número de épocas, el tamaño del lote y el número de muestras utilizadas en cada época se definen para evaluar el modelo. El entrenamiento utiliza el algoritmo de optimización para minimizar la función de pérdida y actualizar el modelo con el error de descenso de gradiente. Deben realizarse varias ejecuciones de entrenamiento para realizar ajustes que reduzcan el error.
4. **Estimación del modelo.** Por último, se realizaron estimaciones para hacer una evaluación final del comportamiento del modelo de aprendizaje profundo. Esta etapa también requirió un nuevo conjunto de datos, cuyo resultado era conocido. Por lo tanto, se proporcionaron nuevos valores de la velocidad específica y del esfuerzo de Von Mises para realizar nuevas estimaciones.

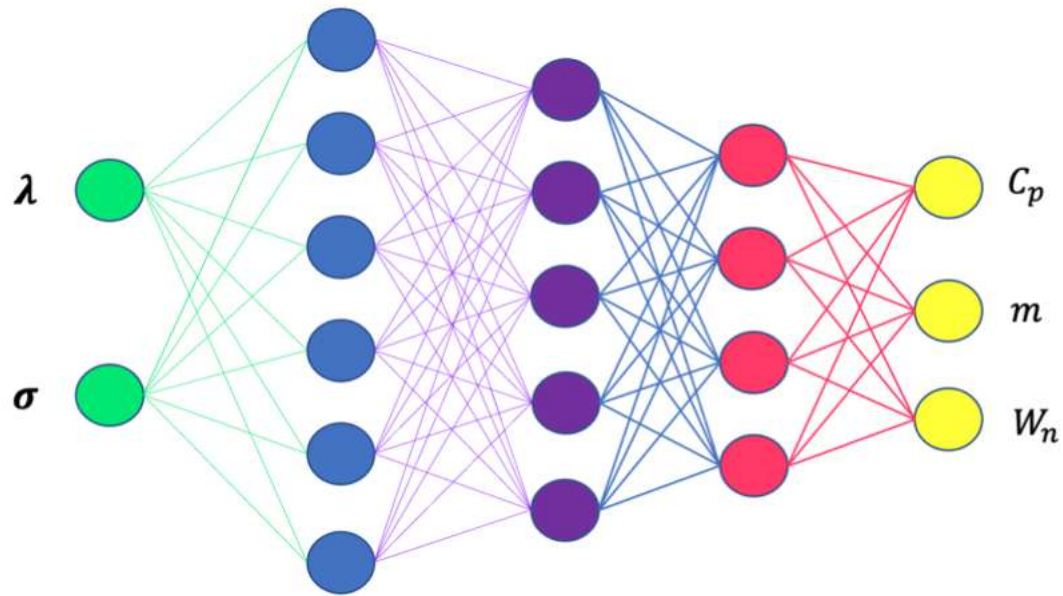


Fig. 38. Arquitectura del modelo de deep learning.

5.4 Programa del modelo de deep learning.

A continuación se describe la programación del modelo de DL en python:

- **Librerías:** se muestran las librerías usadas en el código, las cuales permiten usar una gran variedad de funciones y operaciones.

```
# Librerías
from numpy import loadtxt
from numpy import sqrt
from sklearn.model_selection import train_test_split
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense
from matplotlib import pyplot
```

- **Base de datos:** A continuación, se define la base de datos, se indica en el programa que lea el archivo, se usa un archivo con extensión “csv”.

```
# Base de datos
dataset = loadtxt('dv.csv', delimiter=',')
print(dataset.shape)
"df = read_csv(dataset, header=None)"
```


- **División de los datos:** se hace una división en los datos que serán usados en la entrada de la RNA y los datos que serán usados en la salida.

```
#División los datos, para x son las primeras 3 columnas,  
# para y, son los datos de la cuarta a sexta columna  
X, y = dataset[:, :-3], dataset[:, -3:]
```

- **Conjunto de entrenamiento y prueba:** la base de datos es dividida, el 67% de los datos serán usados para el entrenamiento y el 33% para la validación.

```
# Divide los datos en datos de entrenamiento y de prueba  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33)
```

- **Vector de entrada:** en esta parte del programa se define la variable de los datos que serán usados en la entrada de la RNA.

```
# Determinar los valores de entrada del vector  
n_features = X_train.shape[1]
```

- **Arquitectura de la red neuronal:** en esta sección se definen la cantidad de neuronas por capa, las capas de la red, las funciones de activación y los inicializadores de la RNA.

```
# Arquitectura de la red neuronal  
model = Sequential()  
model.add(Dense(16, activation='relu', kernel_initializer='he_normal', input_shape=(n_features,)))  
model.add(Dense(12, activation='relu', kernel_initializer='he_normal'))  
model.add(Dense(10, activation='relu', kernel_initializer='he_normal'))  
model.add(Dense(8, activation='relu', kernel_initializer='he_normal'))  
model.add(Dense(3))
```

- **Compilación y ajuste del modelo:** se lleva a cabo la compilación usando las métricas de error seleccionadas, después se realiza el ajuste con el conjunto de datos en base a las épocas designadas y el lote o cantidad de datos usados en cada iteración.

```
# Compilación y ajuste del modelo,  
# history es un registro de los valores de las funciones de perdida en las épocas  
model.compile(optimizer='adam', loss='mae', metrics=['mse'])  
history = model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=150, batch_size=32)
```

- **Evaluación del modelo:** en base a las métricas de error, el modelo realiza una evaluación y estima el error que se tiene a la salida.

```
# Evaluación del modelo
mae, mse = model.evaluate(X_test, y_test, verbose=0)
"print('MSE: %.3f, RMSE: %.3f, MAE: %.3f' % (mse, sqrt(mse), mae))"
print('MAE: %.3f, MSE: %.3f, RMSE: %.3f' % (mae, mse, sqrt(mse)))
```

- **Predicción del modelo:** en esta parte del programa en base a datos nuevos, se le pide al modelo de DL hacer una predicción.

```
# Estimación del modelo
row = [10.00, 12.80]
newX = asarray([row])
yhat = model.predict([newX])
print('Predicted: %s' % yhat)
```

- **Gráfica de aprendizaje del modelo de DL:** en la parte final del programa, se realiza una gráfica de la curva de aprendizaje que tiene el modelo de DL en los entrenamientos realizados.

```
# Gráfica de la función de aprendizaje y el conjunto de prueba durante el entrenamiento
pyplot.subplot(211)
pyplot.title('Curva de aprendizaje')
pyplot.plot(history.history['loss'], label='Entrenamiento')
pyplot.plot(history.history['val_loss'], label='Prueba')
pyplot.xlabel('Épocas')
pyplot.ylabel('Error')
pyplot.legend()
pyplot.show()
```

A continuación se muestra el programa completo:

```
# Librerías
from numpy import loadtxt
from numpy import sqrt
from numpy import asarray
from sklearn.model_selection import train_test_split
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense
from matplotlib import pyplot

# Base de datos
dataset = loadtxt('valores2.csv', delimiter=',')
print(dataset.shape)
"df = read_csv(dataset, header=None)"

#División los datos, para x son las primeras 3 columnas,
# para y, son los datos de la cuarta a sexta columna
X, y = dataset[:, :-3], dataset[:, -3:]
```

```

# Divide los datos en datos de entrenamiento y de prueba,
# el modelo se ajustara al 67% de los datos, y el 33 restante se utiliza
para la evaluación

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33)

print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
print(y_train)

# Determinar los valores de entrada del vector
n_features = X_train.shape[1]

print(X_train.shape)

# Arquitectura de la red neuronal
model = Sequential()
model.add(Dense(16, activation='relu', kernel_initializer='he_normal',
input_shape=(n_features,)))
model.add(Dense(12, activation='relu', kernel_initializer='he_normal'))
model.add(Dense(10, activation='relu', kernel_initializer='he_normal'))
model.add(Dense(8, activation='relu', kernel_initializer='he_normal'))
model.add(Dense(3))

# Compilación y ajuste del modelo,
# history es un registro de los valores de las funciones de perdida en
las épocas
model.compile(optimizer='adam', loss='mae', metrics=['mse'])
history = model.fit(X_train, y_train, validation_data=(X_test, y_test),
epochs=150, batch_size=32)

# Evaluación del modelo
mae, mse = model.evaluate(X_test, y_test, verbose=0)
"print('MSE: %.3f, RMSE: %.3f, MAE: %.3f' % (mse, sqrt(mse), mae))"
print('MAE: %.3f, MSE: %.3f, RMSE: %.3f' % (mae, mse, sqrt(mse)))

# Estimación del modelo
row = [10.00, 12.80]
newX = asarray([row])
yhat = model.predict([newX])
print('Predicted: %s' % yhat)

# Gráfica de la función de aprendizaje y el conjunto de prueba durante el
entrenamiento
pyplot.subplot(211)
pyplot.title('Curva de aprendizaje')
pyplot.plot(history.history['loss'], label='Entrenamiento')
pyplot.plot(history.history['val_loss'], label='Prueba')
pyplot.xlabel('Épocas')
pyplot.ylabel('Error')
pyplot.legend()
pyplot.show()

```

5.5. Funcionamiento del modelo de DL

El modelo realiza el entrenamiento, la compilación, el ajuste y la evaluación. Puede estimar las tres variables de diseño al mismo tiempo. La estimación ayuda a conocer la precisión del modelo mediante una comparación entre el valor esperado y la salida de la red, y las métricas ayudan a evaluar el modelo de aprendizaje profundo. La precisión del modelo DL se evalúa en función de las tres variables de salida. La Tabla 5 muestra las pruebas realizadas al modelo DL. Esta tabla se compone de la siguiente manera: pruebas, arquitectura de red, épocas, lote, métrica de error y número de predicciones exitosas por prueba. Resultados de los experimentos de entrenamiento de las redes neuronales multicapa, variando el número de épocas en cada prueba. Las predicciones precisas son el número de predicciones con al menos un 90% de precisión en cada prueba.

Tabla 5. Resultados de los experimentos de entrenamiento de los modelos de DL, variando el número de épocas por prueba. Las predicciones consideradas como precisas son aquellas con al menos un 90% de precisión en cada prueba.

| Prueba | Arquitectura de la red | Épocas | Lote | MAE | MSE | RMSE | Predicciones Precisas |
|--------|------------------------|--------|------|-------|-------|-------|-----------------------|
| 1 | 16-12-10-8-3 | 100 | 32 | 0.144 | 0.070 | 0.224 | 8 |
| 2 | 16-12-10-8-3 | 150 | 32 | 0.130 | 0.062 | 0.210 | 9 |
| 3 | 16-12-10-8-3 | 200 | 32 | 0.09 | 0.022 | 0.150 | 10 |
| 4 | 16-12-10-8-3 | 250 | 32 | 0.080 | 0.020 | 0.140 | 10 |
| 5 | 16-12-10-8-3 | 300 | 32 | 0.100 | 0.027 | 0.160 | 10 |
| 6 | 16-12-10-8-3 | 350 | 32 | 0.080 | 0.020 | 0.138 | 8 |

La Tabla 5 muestra un resumen de los experimentos realizados con el número de pruebas correspondiente, así como los datos más importantes obtenidos en cada ejecución. Como puede observarse en la Tabla 5, las pruebas realizadas con 100 épocas sólo arrojaron 8 casos de predicciones acertadas. En las pruebas con 200 a 300 epochs, el número de predicciones acertadas fue de 10, mientras que en las pruebas con 350 epochs, los casos acertados fueron 8, lo que sugiere que debe mantenerse el rango de 200 a 300 epochs en el entrenamiento.

En la Tabla 5, se puede observar que en los entrenamientos de 100 y 150 epochs, el MAE presenta los mayores errores, entre 0,130 y 0,144, con un total de 17 predicciones acertadas entre estos dos. Sin embargo, en los entrenamientos de 200 y 250 épocas, el MAE presenta un error que oscila entre 0,08 y 0,09, donde se obtuvieron un total de 20 predicciones acertadas. Si se comparan las métricas MAE de los entrenamientos de 100 y 200 épocas, se obtiene una diferencia de 0.054, aproximadamente un 60%. Mientras tanto, la comparación del MAE de 100 y 250 épocas arroja una diferencia aún mayor de 0.064, es decir, aproximadamente un 80%.

En la Fig. 39 se muestra el gráfico de entrenamiento de una de las pruebas realizadas durante el entrenamiento de las redes neuronales artificiales con la configuración 16-12-10-8-3.

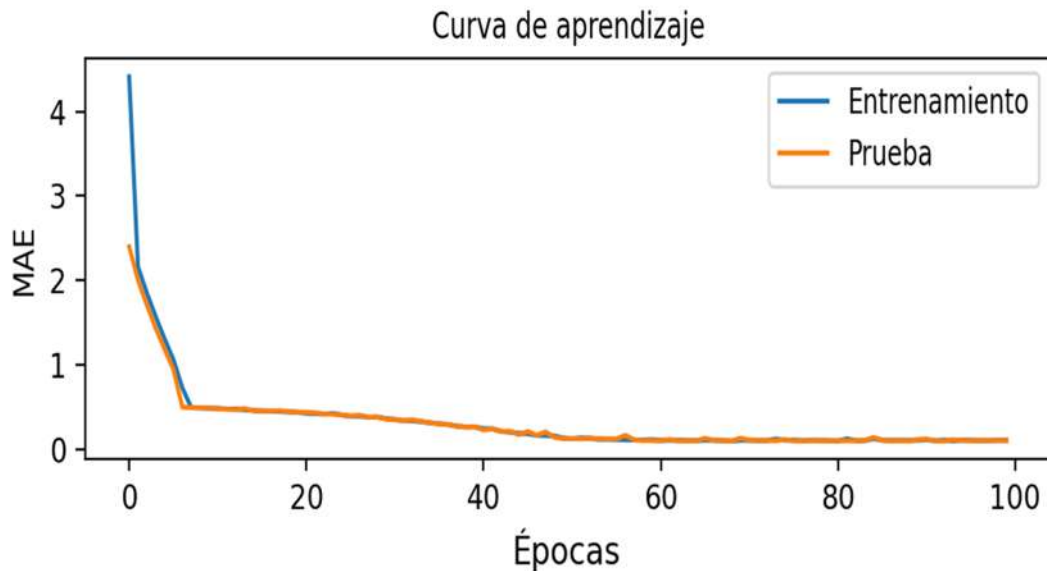


Fig. 39. Gráfica de la prueba 1 del entrenamiento del modelo de DL con 100 épocas, MAE:0.115, MSE: 0.030 y RMSE:0.174.

En la Fig. 40-44 se muestran los gráficos de las curvas de aprendizaje de algunos de los entrenamientos del modelo DL, los gráficos se presentan para 100 a 350 épocas. Las curvas de aprendizaje son gráficos que se utilizan para estimar el rendimiento de la función de aprendizaje o de pérdida utilizada en el modelo DL. Estos gráficos se generan cuando el modelo se evalúa con el conjunto de datos de entrenamiento y validación durante la fase de entrenamiento.

Las curvas de aprendizaje se utilizan para diagnosticar modelos infra o sobreajustados y también ayudan a saber si los conjuntos de datos de entrenamiento y validación son adecuados. Las curvas de aprendizaje nos dicen si el modelo se ajusta a los datos. Esto ocurre cuando, en el gráfico, las curvas de entrenamiento y validación decrecen hasta un punto de estabilidad con una diferencia mínima entre ambas curvas, y esta diferencia se conoce como retardo de generalización [48, 49]. La función de pérdida (MAE) para el entrenamiento (color azul) y la misma función en el conjunto de prueba (color naranja).

Curva de aprendizaje

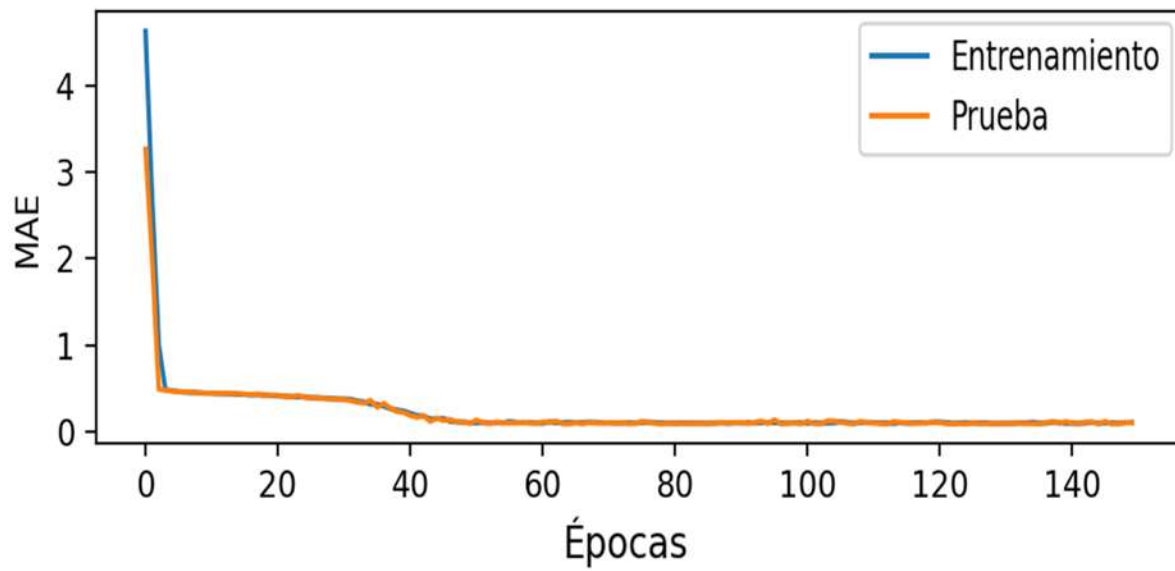


Fig. 40. Gráfica de la prueba 2 del entrenamiento del modelo de DL con 150 épocas, MAE: 0.100, MSE: 0.024 y RMSE:0.153.

Curva de aprendizaje

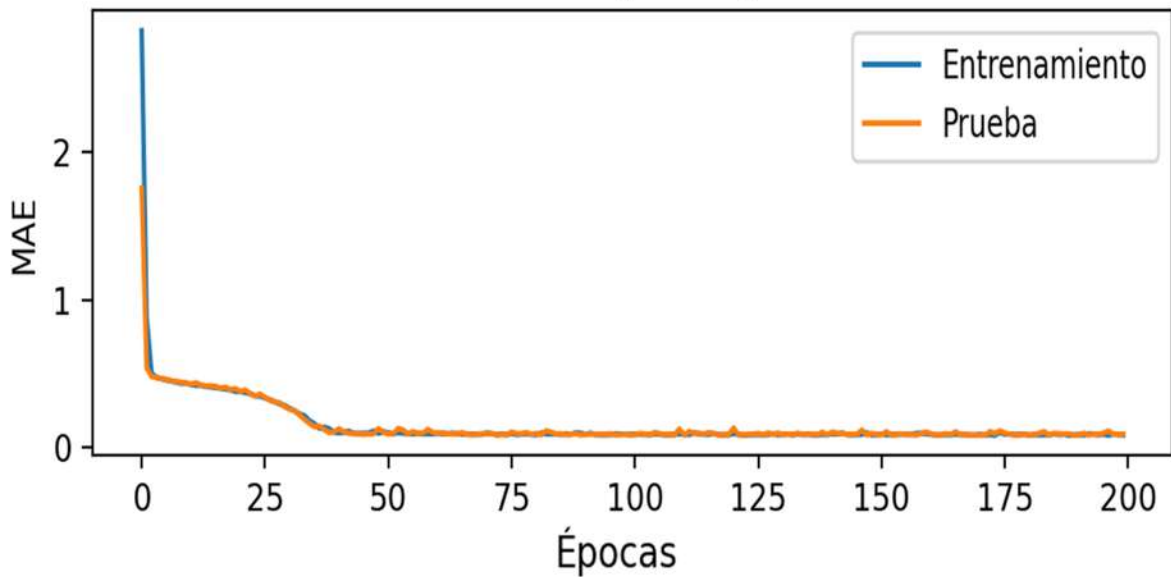


Fig. 41. Gráfica de la prueba 3 del entrenamiento del modelo de DL con 200 épocas, MAE: 0.090, MSE: 0.023 y RMSE:0.152.

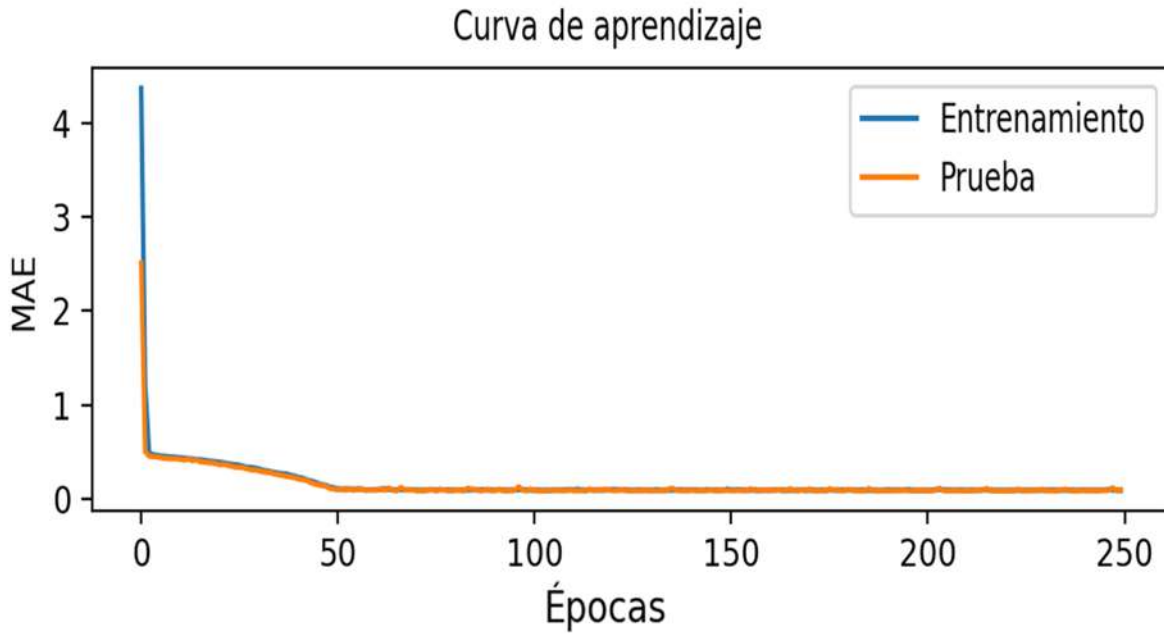


Fig. 42. Gráfica de la prueba 4 del entrenamiento del modelo de DL con 250 épocas, MAE: 0.100, MSE: 0.028 y RMSE:0.168.

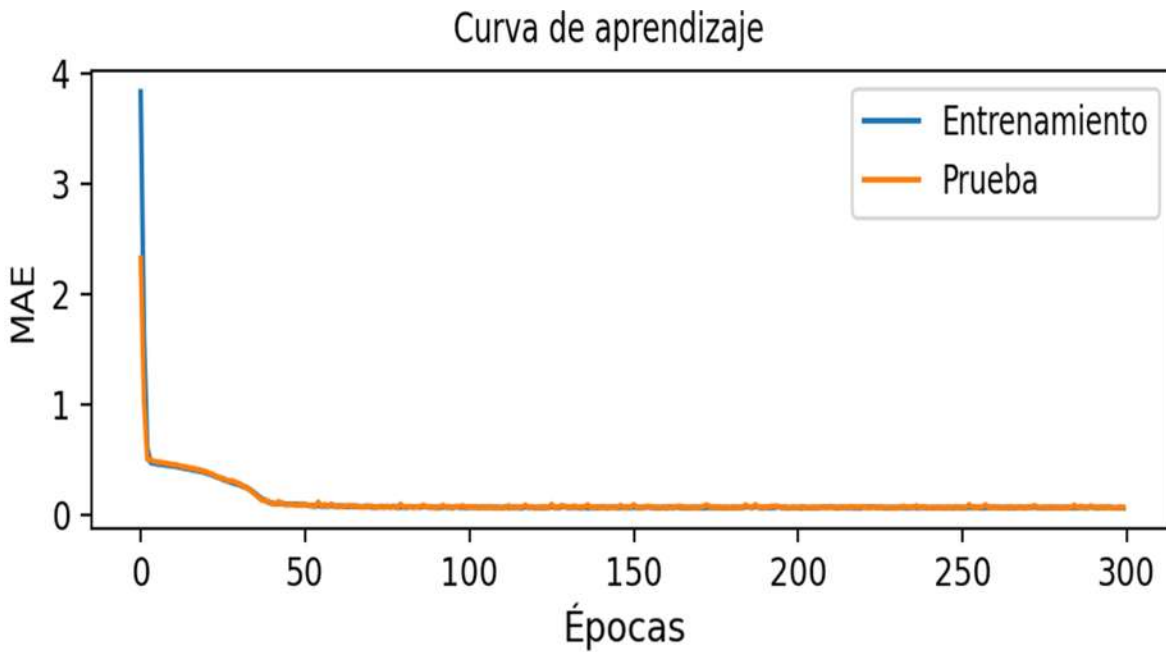


Fig. 43. Gráfica de la prueba 5 del entrenamiento del modelo de DL con 300 épocas, MAE: 0.118, MSE: 0.032 y RMSE:0.180.

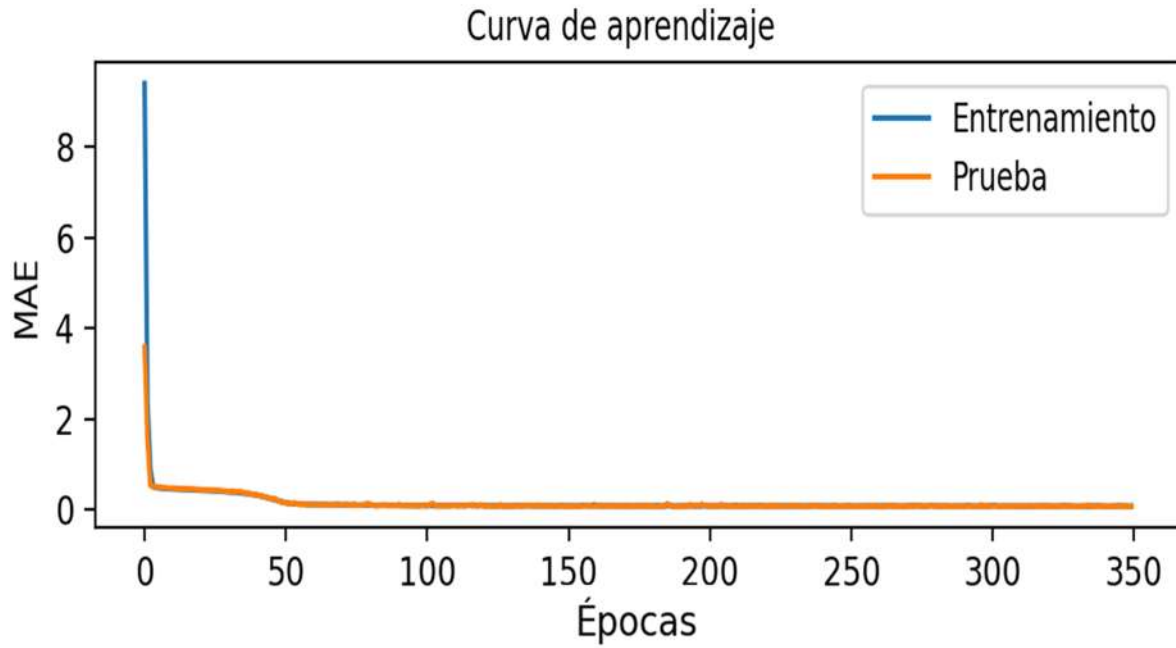


Fig. 44. Gráfica de la prueba 6 del entrenamiento del modelo de DL con 350 épocas, MAE: 0.053, MSE: 0.013 y RMSE:0.112.

6

ANÁLISIS DE RESULTADOS

Se realizaron diferentes números de pruebas con el modelo de DL, los resultados obtenidos se organizaron de la Tabla 6 a la Tabla 8, en cada una de las tablas se muestran las estimaciones realizadas por el modelo, se tienen valores de la masa, el coeficiente de potencia y la frecuencia natural, respectivamente. Las tablas se encuentran conformadas de la siguiente forma: el número de prueba, las épocas de cada prueba, el error absoluto medio, el error cuadrático medio, la raíz del error cuadrático medio, la predicción del modelo, el valor esperado y la precisión de la predicción.

Tabla 6. Predicciones de la masa del aspa en diferentes pruebas, los resultados muestran una precisión superior al 90% en el modelo de DL.

| Prueba | Épocas | MAE | MSE | RMSE | Predicción | Valor esperado | Exactitud |
|--------|--------|-------|-------|-------|------------|----------------|-----------|
| 1 | 100 | 0.115 | 0.030 | 0.174 | 4.83 | 4.76 | 98.53 |
| 5 | 300 | 0.118 | 0.032 | 0.18 | 4.55 | 4.76 | 95.59 |
| 4 | 200 | 0.120 | 0.030 | 0.172 | 4.53 | 4.76 | 95.17 |
| 1 | 100 | 0.130 | 0.030 | 0.172 | 4.52 | 4.76 | 94.96 |
| 1 | 100 | 0.180 | 0.031 | 0.175 | 4.51 | 4.76 | 94.75 |
| 2 | 150 | 0.100 | 0.024 | 0.153 | 4.50 | 4.76 | 94.54 |

La Tabla 6 muestra los resultados de las estimaciones de masa del modelo de DL. En las pruebas se obtuvieron estimaciones con una precisión superior al 94%.

La Tabla 7 muestra los resultados de las estimaciones del coeficiente de potencia (C_p), con estimaciones con una precisión superior al 95%. Se observa que el entrenamiento con 200 épocas tiene el error más bajo de 0.090 para el MAE y una precisión del 99.62%. Mientras que el entrenamiento con 350 épocas tiene un error de 0.095 para el MAE y una precisión ligeramente inferior del 96.97%. También se observa que el entrenamiento con 100 épocas presenta un error de 0.120 para el MAE y una precisión del 99.24%, casi igual que el entrenamiento con 200 épocas, sin embargo, las diferencias son mayores en las métricas de

error MSE y RMSE entre estos 2 entrenamientos, bajo estas observaciones se puede determinar que el entrenamiento con 200 épocas exhibe los mejores resultados.

Tabla 7. Predicciones del coeficiente de potencia para diferentes pruebas, con una precisión superior al 90% en el modelo de DL.

| Test | Épocas | MAE | MSE | RMSE | Predicción | Valor esperado | Exactitud |
|------|--------|-------|-------|-------|------------|----------------|-----------|
| 3 | 200 | 0.090 | 0.023 | 0.152 | 2.63 | 2.64 | 99.62 |
| 1 | 100 | 0.120 | 0.032 | 0.179 | 2.62 | 2.64 | 99.24 |
| 5 | 300 | 0.120 | 0.036 | 0.190 | 2.61 | 2.64 | 98.86 |
| 2 | 150 | 0.110 | 0.027 | 0.163 | 2.68 | 2.64 | 98.48 |
| 4 | 250 | 0.100 | 0.028 | 0.168 | 2.57 | 2.64 | 97.35 |
| 6 | 350 | 0.095 | 0.023 | 0.152 | 2.56 | 2.64 | 96.97 |

La Tabla 8 muestra los resultados de las estimaciones por encima del 96% de la frecuencia natural. Se puede observar que los valores de las métricas de error son prácticamente similares en los diferentes entrenamientos. En el entrenamiento con 350 épocas, tiene un error de 0.053 para el MAE y también tiene una predicción con una precisión del 99.52%. El entrenamiento con 250 y 300 épocas tiene el mismo error de 0.050 para el MAE y una diferencia de precisión del 0.72%. Otra diferencia se encuentra en el RMSE, donde el entrenamiento con 300 épocas tiene el valor más pequeño de 0.107.

Basándonos en las métricas de evaluación y en la precisión, el entrenamiento con 250 épocas se encuentra dentro de los valores adecuados, como se muestra en la Tabla 5, donde se menciona el rango de épocas en el que se deben mantener los entrenamientos del modelo DL.

Tabla 8. Predicciones de frecuencias naturales para diferentes pruebas, con una precisión superior al 90% en el modelo de DL.

| Prueba | Épocas | MAE | MSE | RMSE | Predicción | Valor esperado | Exactitud |
|--------|--------|-------|-------|-------|------------|----------------|-----------|
| 6 | 350 | 0.053 | 0.013 | 0.112 | 4.17 | 4.19 | 99.52 |
| 4 | 250 | 0.050 | 0.012 | 0.109 | 4.13 | 4.19 | 98.57 |
| 1 | 100 | 0.060 | 0.014 | 0.120 | 4.12 | 4.19 | 98.33 |
| 5 | 300 | 0.05 | 0.012 | 0.107 | 4.10 | 4.19 | 97.85 |
| 4 | 250 | 0.060 | 0.012 | 0.111 | 4.06 | 4.19 | 96.90 |

Se puede observar que en el as diferentes pruebas se usaron más de una métrica de error, esto se debe a que se está trabajando con múltiples variables y no existe una métrica de error óptima, ya que hay que considerar que cada variable depende de múltiples valores. Sin

embargo, es fundamental elegir y justificar las adecuadas para cada estudio. En este estudio, se eligieron dos métricas consideradas de escala porque se expresan en las unidades originales de los datos (MAE y RMSE). Además, se eligió el MSE que penaliza los errores de predicción grandes, ya que se expresan al cuadrado.

Las métricas de error son una guía esencial para decidir el modelo DL final; sin embargo, debido a los diversos resultados obtenidos, se utilizó la precisión de predicción para definir el modelo final. En la Tabla 6, la última fila, el entrenamiento con 150 épocas presenta las métricas de error más bajas, pero también la precisión más insuficiente, lo que lo convierte en un candidato menos competitivo en comparación con otros presentados en la tabla.

En las pruebas, las filas 3 y 4 presentan una diferencia de 0.002 en el RMSE en comparación con la prueba con 100 épocas; sin embargo, la precisión es menor. Aunque uno de los entrenamientos con 100 épocas presenta un error para el MAE de 0.115, otros entrenamientos con las mismas épocas presentan errores mayores de 0.130 y 0.180, la variabilidad de los errores es más evidente en los entrenamientos con 100 épocas, lo cual nos indica que es necesario aumentar el número de épocas en esos entrenamientos. El entrenamiento con 200 épocas presenta un error bajo para MAE de 0.120 y una precisión de 95.17%, que son valores aceptables, sugiriendo nuevamente los entrenamientos con 200 épocas, como se observó en la Tabla 5.

En términos generales, todas las medidas de error oscilan en un cierto rango, pero están dentro de límites aceptables; sin embargo, la precisión fue el criterio para definir las épocas de entrenamiento. Los resultados de las Tablas 6-8 muestran que algunas pruebas con entrenamiento de 100 épocas presentan métricas de error más bajas y estimaciones con mayor precisión porque se generan en un único modelo de DL. Este comportamiento se debe probablemente a los ajustes multidimensionales, que son difíciles de observar en las funciones de ajuste generadas.

7

CONCLUSIONES

El diseño de turbinas de viento se ha convertido en un problema complejo, debido al gran número de variables que se tienen en cuenta y el diseño de sus aspas no es la excepción. Su optimización desde diferentes enfoques ha generado la necesidad de incorporar nuevas herramientas como lo es la inteligencia artificial, en específico, los modelos de deep learning, que su funcionamiento emula al del cerebro humano. Estos modelos informáticos presentan grandes ventajas, sin embargo, dependen enteramente de las bases de datos para su entrenamiento. En este estudio se muestra la posibilidad de generar bases de datos para entrenar un modelo DL utilizando funciones objetivo construidas con la metodología de diseño óptimo y la distancia inversa ponderada proporcionando un sistema que permite realizar un diseño multicriterio, el cual ofrece flexibilidad y viabilidad.

La base de datos generada de las funciones objetivo proporcionó un conjunto de información suficiente para el entrenamiento, prueba y validación del modelo de deep learning. En el entrenamiento, las métricas de error fueron una guía esencial para definir el modelo final de DL; sin embargo, debido a los diversos resultados obtenidos en las pruebas, se utilizó la precisión de las predicciones para definir el modelo final.

Los resultados obtenidos de las diferentes pruebas ayudaron en el modelo final, en la Tabla 6, las filas 3 y 4 presentan una diferencia de 0.002 en el RMSE respecto a la prueba con 100 épocas; sin embargo, la precisión es menor. Mientras que en el entrenamiento con 200 épocas, con errores que oscilan entre 0.09 y 0.120 para el MAE y en el entrenamiento con 250, con errores que oscilan entre 0.05 y 0.100 para el MAE, se muestran las estimaciones con mayor precisión y menores errores en las métricas. En general, todas las métricas de error oscilan en un cierto rango, pero están dentro de límites aceptables; Sin embargo, la precisión fue el criterio para definir las épocas de entrenamiento.

Los resultados de la Tabla 6 a la Tabla 8 muestran que algunas pruebas con 100 épocas de entrenamiento tienen métricas de error más bajas y estimaciones con mayor precisión porque se generan en un único modelo DL. Este comportamiento se debe probablemente a ajustes multidimensionales que son difíciles de observar en las funciones de ajuste generadas. El procedimiento mostró que el modelo DL se incorporó con éxito al diseño multicriterio óptimo. Este modelo DL permite estimar valores que pueden ser difíciles de medir, como la

frecuencia natural. Los resultados alcanzan predicciones bajo un rango de precisión entre el 90% y el 98%. La aplicación del modelo de DL junto con las funciones objetivo para la generación de datos simplificó el entrenamiento del modelo, ofreciendo una alternativa diferente en el diseño óptimo de aspas de aerogeneradores.

Glosario

Onshore. Energía eólica generada por aerogeneradores situados en tierra.

I.A. Inteligencia artificial.

Deep learning. Modelo de redes neuronales artificiales con varias capas intermedias.

BEM. Teoría de momento de elemento de pala.

IDW. Interpolación inversa ponderada.

Darrieus. Es un tipo de aerogenerador de eje vertical usado para generar energía eléctrica con la corriente del viento.

Hub. Parte del aerogenerador donde se instalan las aspas.

CAD. Diseño asistido por computadora (por sus siglas en inglés).

ANSYS. Programa de computadora para la simulación y el diseño en 3-D.

QBlade. Es un software de simulación de código abierto para el diseño de aspas de turbinas de viento y la simulación aerodinámica.

FEM. Modelo de elemento finito.

Anexo: Publicaciones y Aportes

| | |
|--------------------|---|
| Nombre del trabajo | Multi-criteria optimal design of small wind turbine blades based on deep learning methods |
| Revista | Energy |
| Editorial | ELSEVIER |
| Factor de impacto | 9 <i>Q1</i> |
| Autores | Josué Jordi Zavala Morales, Erasmo Cadenas Calderón, Rafael Campos Amezcua |
| Año de publicación | 5 Febrero 2024 |
| Doi | https://doi.org/10.1016/j.energy.2024.130625 |



Multi-criteria optimal design of small wind turbine blades based on deep learning methods

Zavala J. Jordi ^a, Cadenas Erasmo ^{a,*}, Campos-Amezcuca Rafael ^b

^a Faculty of Mechanical Engineering, Universidad Michoacana de San Nicolas de Hidalgo, Santiago Tapia 403, 58000, Morelia, Mexico

^b The National Technological Institute of Mexico/CENIDET, Interior Internado Palmira S/N, Col. Palmira, 62490, Cuernavaca, Mexico

ARTICLE INFO

Keywords:

Deep learning
Wind energy
SHAWT
Multi-criteria optimum design

ABSTRACT

The development of a Deep Learning (DL) model using multi-criteria optimal design of wind turbine blades is presented, focusing on the key variables TSR and Von Mises stress to predict: blade mass, power coefficient and natural frequency. The DL model was trained with data from three mathematical functions generated using distance-weighted inverse interpolation. The model allowed the generation of multiple feasible designs, which satisfy the design constraints. The BEM theory was used in the generation of the aerodynamic model of the 12.5 kW wind turbine using the NREL-S818 airfoil. Subsequently, the structural behavior of the blades was analyzed under three design load cases specified in IEC 61400-2. The results showed that in the training phase, the MAE, MSE and MSR error metrics were an essential guide in the development of the DL model. Interesting behaviors were observed due to the diverse results obtained, which are probably due to the multidimensional fits that are difficult to observe in the fit functions generated. However, training between 200 and 250 epochs performed better, with errors ranging from 0.05 to 0.120 for the MAE. The DL model exhibited the remarkable ability to predict the optimal output variables, with accuracy ranging from 90% to 98%.

1. Introduction

The design of wind turbine blades involves the choice and calculation of several variables, some involving the solution of specific energy supply problems, such as power; others involving aerodynamics such as geometry, tip speed ratio, power coefficient, number of blades, etc., and others of a structural nature such as the stresses generated by aerodynamic forces and the behavior of the chosen material. Another discipline that derives from the design of machine elements is the optimal design of mechanical elements. In the process of optimal design, a variable to be optimized is chosen and other variables that are affected are involved in the process. The success of the optimization generally consists of minimizing or maximizing the dependent variable (variable to be optimized) without considerably affecting the other variables. There is also the multi-criteria optimal design, in this process two or more variables to be optimized and two or more independent variables may be involved.

In the optimal design process, the relationship between dependent and independent variables can be shown through two-dimensional or three-dimensional graphs, these graphs are called objective function. However, when the formulated problem has more than three variables, it is not possible to observe the behavior of such function with traditional graphs. An indispensable issue in the optimal design process

are the constraints, which must be identified when approaching the problem and are defined as the limitations that the variables have within the problem formulation.

For the issue at hand, the optimal design of wind turbine blades, there are cases where it is approached only from an aerodynamic approach [1,2], there are also only structural approaches as in [3,4].

In their study Gun and Rosen [5], propose the annual energy production as a variable to be optimized. The most important results are obtained when they involve the rotational speed in the optimal design process. Karam and Hani [6], presented a process in which they proposed to minimize blade mass and vibrations while maximizing stiffness. The results were successful and the article highlights the multidimensional model used to find the global optimum in the objective function and the formulation of the optimization problem using nonlinear mathematical techniques. The study showed promising results, however, it is complicated and extensive.

Dogan et al. [7] proposed a method to minimize the cost of energy generated by wind turbines by optimizing rotor radius, hub height and rated power. The work compares three different optimization algorithms, highlighting the so-called improved Yin-Yang pair optimization, which achieved better results. Jaechwan et al. [8] proposed an optimization method to find the best hub height of a wind turbine;

* Corresponding author.

E-mail address: ecadenas@umich.mx (C. Erasmo).

<https://doi.org/10.1016/j.energy.2024.130625>

Received 1 May 2023; Received in revised form 2 January 2024; Accepted 4 February 2024

Available online 5 February 2024

0360-5442/© 2024 Elsevier Ltd. All rights reserved.

Anexo: Documentos para digitalización

DGB



Identificación de reporte de similitud: oid:3117:362339436

NOMBRE DEL TRABAJO

**APRENDIZAJE PROFUNDO APLICADO A
L DISEÑO ÓPTIMO MULTICRITERIO DE
ÁLABES DE TURBINA DE VIENTO**

AUTOR

Josué Jordi Zavala Morales

RECUENTO DE PALABRAS

20197 Words

RECUENTO DE CARACTERES

113080 Characters

RECUENTO DE PÁGINAS

86 Pages

TAMAÑO DEL ARCHIVO

11.3MB

FECHA DE ENTREGA

Jun 21, 2024 1:55 PM CST

FECHA DEL INFORME

Jun 21, 2024 1:57 PM CST

● 22% de similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos.

- 20% Base de datos de Internet
- Base de datos de Crossref
- 0% Base de datos de trabajos entregados
- 12% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossref

Formato de Declaración de Originalidad y Uso de Inteligencia Artificial

Coordinación General de Estudios de Posgrado
Universidad Michoacana de San Nicolás de Hidalgo



A quien corresponda,

Por este medio, quien abajo firma, bajo protesta de decir verdad, declara lo siguiente:

- Que presenta para revisión de originalidad el manuscrito cuyos detalles se especifican abajo.
- Que todas las fuentes consultadas para la elaboración del manuscrito están debidamente identificadas dentro del cuerpo del texto, e incluidas en la lista de referencias.
- Que, en caso de haber usado un sistema de inteligencia artificial, en cualquier etapa del desarrollo de su trabajo, lo ha especificado en la tabla que se encuentra en este documento.
- Que conoce la normativa de la Universidad Michoacana de San Nicolás de Hidalgo, en particular los Incisos IX y XII del artículo 85, y los artículos 88 y 101 del Estatuto Universitario de la UMSNH, además del transitorio tercero del Reglamento General para los Estudios de Posgrado de la UMSNH.

| Datos del manuscrito que se presenta a revisión | | |
|---|---|----------------------------|
| Programa educativo | Doctorado en ciencias ingeniería mecánica | |
| Título del trabajo | APRENDIZAJE PROFUNDO APLICADO AL DISEÑO ÓPTIMO MULTICRITERIO DE ÁLABES DE TURBINA DE VIENTO | |
| | Nombre | Correo electrónico |
| Autor/es | Josué Jordi Zavala Morales | 1731329j@umich.mx |
| Director | Dr. Erasmo Cadenas Calderón | ecadenas@umich.mx |
| Codirector | Dr. Rafael Campos Amezcua | rafael.ca@cenidet.tecnm.mx |
| Coordinador del programa | Dra. Georgina Carbajal de la Torre | georgina.carbajal@umich.mx |

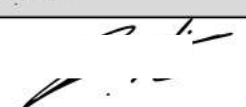
| Uso de Inteligencia Artificial | | |
|--------------------------------|-------------|-------------|
| Rubro | Uso (sí/no) | Descripción |
| Asistencia en la redacción | No | |

Formato de Declaración de Originalidad y Uso de Inteligencia Artificial

Coordinación General de Estudios de Posgrado
Universidad Michoacana de San Nicolás de Hidalgo



| Uso de Inteligencia Artificial | | |
|--|-------------|-------------|
| Rubro | Uso (sí/no) | Descripción |
| Traducción al español | No | |
| Traducción a otra lengua | No | |
| Revisión y corrección de estilo | No | |
| Análisis de datos | No | |
| Búsqueda y organización de información | No | |
| Formateo de las referencias bibliográficas | No | |
| Generación de contenido multimedia | No | |
| Otro | No | |

| Datos del solicitante | |
|-----------------------|---|
| Nombre y firma | Josué Jordi Zavala Morales  |
| Lugar y fecha | 21 de junio de 2024 |

Referencia Bibliográfica

1. IRENA (2016), Wind Power, Technology Brief, International Renewable Energy Agency, Abu Dhabi.
2. IRENA (2019), Future of wind: Deployment, investment, technology, grid integration and socio-economic aspects (A Global Energy Transformation paper), International Renewable Energy Agency, Abu Dhabi.
3. Romanski L, Bieniek J, Komarnicki P, Debowski M, Detyna J. Estimation of operational parameters of the counter-rotating wind turbine with artificial neural networks. *Arch Civ Mech Eng* 2017;17:1019–28.
4. Wen H, Sang S, Qiu C, Du X, Zhu X, Shi Q. A new optimization method of wind turbine airfoil performance based on Bessel equation and GABP artificial neural network. *Energy* 2019;187:116106.
5. Fatehi M, Nili-Ahmadabadi M, Nematollahi O, Minaeian A, Kim KC. Aerodynamic performance improvement of wind turbine blade by cavity shape optimization. *Renew Energy* 2019;132:773–85.
6. Zhu J, Zhou Z, Cai X. Multi-objective aerodynamic and structural integrated optimization design of wind turbines at the system level through a coupled blade-tower model. *Renew Energy* 2020;150:523–37.
7. Sjolund JH, Lund E. Structural gradient based sizing optimization of wind turbine blades with fixed outer geometry. *Compos Struct* 2018;203:725–39.
8. Yang H, Chen J, Pang X, Chen G. A new aero-structural optimization method for wind turbine blades used in low wind speed areas. *Compos Struct* 2018;207:446–59.
9. Karam Y, Hani M. Optimal frequency design of wind turbine blades. *J Wind Eng Ind Aerodyn* 2017;90:961–86.
10. Dongran S, Jumbo L, Nematollahi O, Jian Y, Mei S, Yun M, Xuebing Y, Lingxiang H, Young H. Optimal design of wind turbines on high-altitude sites based on improved Yin-Yang pair optimization. *Energy* 2020;193:116794.
11. Jaehwan L, Dong R, Kwan-Soo L. Optimum hub height of a wind turbine for maximizing annual net profit. *Energy Convers Manage* 2015;100:90–6.

12. Kaveth A, Sabeti S. Optimal of jacket supporting structures for offshore wind turbines using CBO and ECBO algorithms. *Period Polytech Civ Eng* 2017.
13. Quan W, Jun W, Jinfeng S, Jun R, Qiong W. Optimal design of wind turbine airfoils based on functional integral and curvature smooth continuous theory. *Aerpace Sci Technol* 2016;11:34–42.
14. Mohamed M, Janiga E, Pap E, Thévenin D. Optimal blade shape of a modified Savonius turbine using an obstacle shielding the returning blade. *Energy Convers Manage* 2010;52:236–42.
15. Ernesto B, Andre T. Optimal design of horizontal-axis wind turbines using blade- element theory and evolutionary computation. *J Solar Energy Eng* 2002;124:357–63.
16. Mojtaba T, Tahmine S, Kiana R, Pouria A. Aerodynamic optimal design of wind turbine blades using geneti algorithm. *Energy Equip Syst* 2014;2:185–93.
17. Wang H, Xue W, Liu Y, Peng J, Jiang H. Probabilistic wind power forecasting based on spiking neural network. *Energy* 2020;196:117072.
18. Mi X, Zhao S. Wind speed prediction based on singular spectrum analysis and neural network structural learning. *Energy Convers Manage* 2020;216:11956.
19. Zi L, Xiaolei L. Wind power forecasting of an offshore wind turbine based on high-frequency SCADA data and deep learning neural network. *Energy* 2020;201:117693.
20. Kreutz M, Ait-Alla A, Varasteh K, Oelker S, Greulich A, Freitag M, Thoben K-D. Machine learning-based icing prediction on wind turbines. *Proc CIRP* 2019;81:423–8.
21. Bedoud K, rachedi Mahieddine A, Bahi T, Lakel R, Grid A. Robust control of doubly fed induction generator for wind turbine under sub-synchronous operation mode. *Energy Procedia* 2015;74:886–99.
22. Sessarego M, Feng J, Ramos-García N, Horcas SG. Design optimization of a curved wind turbine blade using neural networks and an aero-elastic vortex method under turbulent inflow. *Renew Energy* 2019;146:1524–35.
23. Fitzsimons, S. (2009) Pompeii wall art, Pompeii Wall Art. *Disponibile en:* <http://lookingglasscarus.blogspot.com/2009/06/pompeii-wall-art.html> (*Consultado:* 30 Abril 2024).

24. Britannica, T. Editors of Encyclopedia (2024, March 5). ornithopter. Encyclopedia Britannica. <https://www.britannica.com/technology/ornithopter>
25. Britannica, T. Editors of Encyclopaedia (2024, January 14). windmill. Encyclopedia Britannica. <https://www.britannica.com/technology/windmill>.
26. Patente de un Molino viento elaborado por Aldrich, recuperado de: <https://fineartamerica.com/art/aldrich+windmill>
27. Nissen, P -O, Quistgaard, T, Thorndahl, J, Christensen, Benny, Maegaard, P, Madsen, Birger T, and Hvidtfelt Nielsen, K. Wind power the Danish way. From Poul la Cour to modern wind turbines. Denmark: N. p., 2009. Web.
28. Hau, E., (2006), Wind turbines: fundamentals, technologies, application, economics. Springer Verlag.
29. Manwell, J.F. Wind energy explained : theory, design, and application / James Manwell, Jon McGowan, Anthony Rogers. – 2nd ed.
30. Somers DM. The S816, S817, and S818 airfoils. 2004.
31. Marten D, Wendler J, Pechlivanoglou G, Nayeri C, Paschereit C. Qblade: An open source tool for design and simulation of horizontal and vertical axis wind turbines. *Int J Emerg Technol Adv Eng* 2013;3:264–9.
32. Glauert H. In: Durand WF, editor. Airplane propellers, in division I of aerodynamic theory. Springer-Verlag; 1976.
33. Wang A. A brief review on wind turbine aerodynamics. *Theor Appl Mech Lett* 2012; 2.
34. Zhao D, Han N, Goh E, Cater J, Reinecke A. Aerodynamics of horizontal axis wind turbines and wind farms, *Wind turbines and aerodynamics energy harvesters*. Academi Press; 2019, p. 431–61.
35. Brøndsted P, Lillholt H, Lystrup A. Composite materials for wind power turbine
36. blades. *Annu Rev Mater Res* 2005;35:505–38.
37. Chakrabarty A, Mannan S, Cagin T. In: Arnab Chakrabarty TC, editor. Multiscale
38. modeling for process safety applications. Butterworth-Heinemann; 2016.

39. Molholt Jensen F, Branner K. Introduction to wind turbine blade design. *Des Mater* 2013;3–28.
40. Thompson M, Thompson J. Introduction to ANSYS and finite element modeling. Kidlington, United Kingdom: Butterworth-Heinemann; 2017.
41. De la Fuente E. Von Mises stresses in random vibration of linear structures. *Comput Struct* 2009;87:1253–62.
42. Jureczko M, Pawlak M, Goh E, Mezyk A. Optimisation of wind turbine blades. *Mater Process Technol* 2005;167:463–71.
43. Darshil S, Peter S, Mike C. Can flax replace E-glass in structural composites? A small wind turbine blade case study. *Composites B* 2013;52:172–81.
44. Arora JS. Introduction to optimum design. Iowa, United States: Elsevier; 2012.
45. Shukla, K., Kumar, P., Mann, G. S., & Khare, M. (2019). Mapping spatial distribution of particulate matter using Kriging and Inverse Distance Weighting at supersites of megacity Delhi. *Sustainable Cities and Society*, 101997. doi:10.1016/j.scs.2019.101997.
46. Lu, G. Y., & Wong, D. W. (2008). An adaptive inverse-distance weighting spatial interpolation technique. *Computers & Geosciences*, 34(9), 1044–1055. doi:10.1016/j.cageo.2007.07.010.
47. Lennie M, Marten D, Pechlivanoglou G, Nayeri C, Paschereit C. Development and validation of a modal analysis code for wind turbine blades. In: *Proceedings of ASME turbo expo 2014: Turbine technical conference and exposition GT2014*. vol. 3B, 2014.
48. Berzal F. *Redes neuronales y deep learning*. Granda, Spain: Fernando Berzal; 2018.
49. Brownlee J. When to use MLP, CNN, and RNN neural networks. 2022, <https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural-networks/>.
50. Irving G, Isard M, Jozefowicz R, Jia Y, Kaiser L, Kudlur M, Levenberg J, Mané D, Schuster M, Monga R, Moore S, Murray D, Olah C, Shlens J, Steiner B, Sutskever I, Talwar K, Tucker P, Vanhoucke V, Vasudevan V, Viégas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X. TensorFlow: Large-scale machine learning on heterogeneous systems. 2015, 2015. Software available from tensorflow.org., <https://www.tensorflow.org/?hl=es>- 419.

51. Kingma DP, Ba J. Adam: A method for stochastic optimization. 2017, arXiv: 1412.6980.
52. Brownlee J. Tutorial: Get started in deep learning with tf.keras. 2019, <https://machinelearningmastery.com/tensorflow-tutorial-deep-learning-with-tf-keras/>.
53. GISGeography. Inverse Distance Weighting (IDW) interpolation. 2021, URL: <https://gisgeography.com/inverse-distance-weighting-idw-interpolation/>.
54. IEC 61400-2-2015 wind turbines design requirements for small wind turbines 2015.
55. Johnson GJ. Wind energy systems. Englewood Cliffs, NJ: Prentice Hall; 1985.
56. Betz A. The maximum of the theoretically possible exploitation of wind by means of a wind motor. Wind Eng 2013;37(4):4416. Translation of: Das Maximum der theoretisch möglichen Ausnutzung des Windes durch Windmotoren, Zeitschrift für das gesamte Turbinenwesen, Heft 26, 1920.
57. Van Kuik GAM. The Betz-Joukovsky limit: on the contributions to rotor aerodynamics by the British, German and Russian scientific schools. Wind Energy 2012;15:335-44.
58. van Kuik GAM. The Lanchester-Betz-Joukovsky limit. Wind Energy 2007;10:289-91.
59. Hansen MOL. Aerodynamics of wind turbines. 3rd ed. London and New York, NY: Routledge; 2015. ISBN 978-1-138-77507-7.
60. Burton T, Jenkins N, Sharpe D, Bossanyi E. Wind energy handbook. 2nd ed. Wiley; 2011. ISBN 9781119993926.
61. Schaffarczyk AP. Introduction to wind turbine aerodynamics. Dordrecht: Springer; 2011. ISBN-13: 978-3642364082.