

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
ESCUELA DE POSGRADO



**“Diseño e Implementación de un Sistema de Diagnóstico de Fallas
para la Inspección y Detección de Fallas en Componentes de
Procesos Industriales utilizando un Robot Móvil y Algoritmos de
Inteligencia Artificial”**

**TESIS PARA OPTAR EL GRADO ACADÉMICO DE MAGISTER EN
INGENIERÍA DE CONTROL Y AUTOMATIZACIÓN**

AUTOR

Alberto Hiroshi Inafuku Yoshida

ASESOR:

Dr. Carlos Gustavo Pérez Zúñiga

Septiembre, 2020

RESUMEN

El presente trabajo de tesis tiene como principal objetivo desarrollar el algoritmo de control de un robot móvil, el cual se desplazará por una planta industrial en busca de fallas en equipos críticos como son los motores y las bombas. Los motores y las bombas son componentes básicos y fundamentales para los procesos industriales, por lo tanto, es importante mantenerlos en óptimas condiciones con el fin de evitar paradas imprevistas, sobrecostos, y pérdidas de calidad y eficiencia del proceso. En muchos casos, los trabajos de mantenimiento implican medir la vibración de los equipos en lugares confinados o de difícil acceso y debido a que, en la actualidad, se debe velar por la seguridad y bienestar del personal, se propone utilizar un robot móvil para esta tarea. Por lo tanto, se propone diseñar e implementar un sistema de diagnóstico de fallas utilizando mediciones de vibración en conjunto con técnicas de Inteligencia Artificial para un robot móvil. Para lograr el objetivo propuesto, primero se estudian los componentes industriales de interés, resaltando las fallas más comunes y la normativa correspondiente para la clasificación de su condición. Seguidamente, se presentan las estrategias de mantenimiento existentes para su evaluación, resaltando las ventajas y desventajas de cada una, y se introducen los robots móviles como alternativa, teniendo como ejemplos casos reales en donde estos son importantes para realizar trabajos de búsqueda y rescate, en condiciones inaccesibles y peligrosas para las personas. Luego se presentan los algoritmos utilizados en el análisis vibracional, resaltando la Transformada de Hilbert Huang, la cual descompone las señales en Funciones de Modo Intrínseco, señales ortogonales entre sí, que describirán la vibración en función del tiempo y la frecuencia. Las Funciones de Modo Intrínseco serán las entradas a nuestra Red Neuronal Convolucional, una metodología de Aprendizaje Profundo e Inteligencia Artificial para entrenar un modelo que nos ayudará a diagnosticar el tipo de falla. Posteriormente, se proponen los algoritmos a implementar con Python, Keras y TensorFlow y se ponen a prueba con la base de datos de mediciones de vibración MAFAULDA. Finalmente se presentan los parámetros del diseño del robot móvil mencionando el hardware a utilizar y la metodología de medición, y se realiza una comparación de tiempos de procesamiento con una PC para pruebas y la NVIDIA Jetson Nano.

Dedicatoria

A Dios, por darme la fuerza para lograr todas mis metas. A mis padres Lucho e Irma, por haberme forjado como persona y profesional. A mi novia Ericka, por su apoyo y aliento en cada paso que doy. Y a mi asesor Gustavo Pérez, por la motivación, consejos y conocimiento brindado.



Contenido

Introducción	14
Capítulo 1. Estado del arte en metodologías de diagnóstico de fallas para componentes de los procesos industriales	16
1.1. Introducción	16
1.2. Fallas en componentes de los procesos industriales	16
1.2.1. Terminología	18
1.3. Componentes industriales a evaluar	18
1.3.1. Bombas	18
1.3.2. Motores	23
1.3.3. Normativa	26
1.4. Estrategias de Mantenimiento	29
1.4.1. Mantenimiento Correctivo	29
1.4.2. Mantenimiento Preventivo	29
1.4.3. Mantenimiento Predictivo	30
1.4.4. Mantenimiento predictivo mediante Análisis Vibracional	31
1.5. Robots para el diagnóstico de fallas	41
1.6.1. Basado en Modelos	43
1.6.2. Basado en Datos	44
1.7. Procesamiento de datos	44
1.7.1. Análisis Vibracional	44
1.8. Inteligencia Artificial	46
1.8.1. Definiciones de Inteligencia Artificial	46
1.8.2. Inteligencia artificial aplicado en el diagnóstico de fallas	47
1.9. Propuesta de Tesis y Objetivos	48
1.9.1. Propuesta de Tesis	48
1.9.2. Objetivo General y Objetivos Específicos	48
Capítulo 2. Procesamiento para el diagnóstico de fallas	49
2.1. Introducción	49
2.2. Adquisición de datos	49
2.2.1. Medición con acelerómetro	49
2.3. Acondicionamiento de la señal	51
2.4. Procesamiento de datos	51
2.4.1. La Transformada de Hilbert Huang	52

2.4.2.	Evaluación de la Transformada de Hilbert Huang	58
2.4.3.	Entradas para la Red Neuronal	65
2.5.	Conclusiones del Capítulo	67
Capítulo 3.	Diseño de la Red Neuronal Convolutiva	68
3.1.	Introducción	68
3.2.	Redes Neuronales	68
3.3.	Aprendizaje Automático	70
3.4.	Aprendizaje Profundo	70
3.5.	Redes Neuronales Convolucionales (CNN)	71
3.5.1.	Aprendizaje de Aspectos	74
3.5.2.	Clasificación	80
3.5.3.	Diseño de una Red Neuronal Convolutiva para el diagnóstico de fallas en componentes industriales	86
3.5.4.	Programación de la red neuronal diseñada.	90
3.6.	Pruebas del algoritmo.....	92
3.6.1.	Información de la Base de Datos	92
3.7.	Evaluación de parámetros	95
3.9.	Conclusiones del Capítulo	110
Capítulo 4.	Propuesta de Implementación	111
4.1.	Introducción.....	111
4.2.	Implementación del Hardware	111
4.2.1.	Diagrama general del sistema.....	111
4.2.2.	Sensores.....	113
4.2.2.	CPU	113
4.3.	Procedimiento para la configuración e instalación del Hardware y Software de la NVIDIA Jetson Nano	116
4.3.1.	Instalación del Hardware.....	116
4.3.2.	Instalación y configuración del Software	117
4.4.	Evaluación de tiempos de procesamiento en NVIDIA Jetson Nano	118
4.5.	Procedimiento para la configuración e instalación del Hardware y Software de la STM32 Nucleo-64 431RB y el acelerómetro triaxial MPU 6050.....	119
4.5.1.	Instalación del Hardware.....	119
4.5.1.	Software utilizado para la tarjeta STM32 Nucleo64 G431RB	120
4.6.	Procedimiento para la medición	123
4.6.1.	Para la adquisición de datos en motores y bombas:	123

4.7. Conclusiones.....	126
Conclusiones	127
Recomendaciones	129
Bibliografía.....	130



ÍNDICE DE FIGURAS

- Figura 1. Isermann, 2011, Esquema de proceso automatizado, pág. 19
- Figura 2. Bomba centrífuga, 0.5 HP, 1"x1",
<https://www.maestro.com.pe/productos/gasfiteria/bombas/bomba-centrifuga-05hp-1-x-1>
- Figura 3. Partes de una bomba centrífuga, <http://www.cindex.com.mx/bombas/conoces-el-funcionamiento-de-una-bomba-centrifuga/>
- Figura 4. Isermann, 2011, Diagrama seccional de una bomba centrífuga, pág. 158
- Figura 5. Diagrama de pastel con la distribución de las frecuencias, elaboración propia.
- Figura 6. Bomba reciprocante de desplazamiento positivo
<https://www.miltonroy.com/es/productos/bombas-de-proceso/bomba-reciprocante-de-desplazamiento-positivo-poweroyal/>
- Figura 7. Isermann, 2011, Esquema de una bomba de diafragma, pág. 184
- Figura 8. Motor AC Asíncrono Trifásico, <https://es.aliexpress.com/item/ac-three-phase-asynchronous-motor/1212888134.html>
- Figura 9. Partes de un motor de inducción,
<https://maquinaselectricasblog.wordpress.com/motor-de-induccion/>
- Figura 10. Siddiqui y Sahay, 2014, Clasificación de fallas en motores de inducción, pág. 3
- Figura 11. Siddiqui y Sahay. 2014, Diagrama de bloques del proceso de monitoreo en-línea, pág. 13
- Figura 12. Diagrama de pastel con la distribución de las frecuencias, elaboración propia
- Figura 13. Inspección Manual, <https://www.dmc.pt/es/analise-de-vibracoes-em-motores-eletricos/>
- Figura 14. White, 2010, Movimiento Armónico Simple, pág 17
- Figura 15. Ejemplo de desfase entre 2 señales, <https://www.fiscalab.com/ejercicio/752>
- Figura 16. Sarmiento, 2020, Comparación del Desplazamiento, Velocidad y Aceleración de Vibración.
- Figura 17-20 Sarmiento, 2020, Síntomas de fallas en rodamientos
- Figura 21. Procedimiento para el Análisis Vibracional, elaboración propia
- Figura 22. Tabla de Charlotte, <https://www.technicalassociates.net/>
- Figura 23. Tabla de vibración RMS, <https://rms-reliability.com/training/>
- Figura 24. Tadokoro, 2015, Robot Quince, pág. 1
- Figura 25. Abhishek, et. al, 2017, Robot DRASB, pág. 6

Figura 26. Robot para inspección de tuberías, Departamento de Robótica y Mecatrónica de la Facultad de Ingeniería Mecánica de la Universidad AGH de Ciencia y Tecnología de Cracovia, 2017.

Figura 27. Robot SPOT desarrollado por Energy Robotics y Boston Robotics. Fuente: Energy Robotics, 2020.

Figuras 28-32. ISO 10816-1. Posicionamiento del sensor en las máquinas.

Figura 33. Diagrama de flujo del algoritmo Transformada de Hilbert Huang, elaboración propia.

Figura 34. Diagrama de flujo del algoritmo Descomposición Empírica de Modos, elaboración propia.

Figura 35. Señal de entrada con 5 tonos, elaboración propia.

Figura 36. Espectro de la señal de entrada utilizando la Transformada de Fourier, elaboración propia.

Figura 37. IMF obtenidas en la etapa EMD, elaboración propia.

Figura 38. Señal original X098_FE_1772, elaboración propia.

Figura 39. Espectro de la señal de entrada utilizando la Transformada de Fourier, elaboración propia

Figura 40. Espectro de las IMF encontradas, elaboración propia.

Figura 41. Señal original de un motor con falla en rodamientos. Elaboración propia

Figura 42. Espectro de la señal de un motor con falla en rodamientos. Elaboración propia

Figura 43. Espectro de las IMF encontradas. Elaboración propia

Figura 44. Ejemplo de Red Neuronal, elaboración propia.

Figura 45. Diagrama de Ventt representando Inteligencia Artificial, Aprendizaje Automático y Aprendizaje Profundo, elaboración propia.

Figura 46. Saha, 2018, Red Neuronal Convolutacional.

Figura 47. Saha, 2018, Ejemplo de implementación de una Red Neuronal Convolutacional para reconocimiento de dígitos escritos a mano.

Figura 48. Imagen de una casa, en escala de grises, 8-bit, 231 x 218 pixeles, elaboración propia.

<https://shodhganga.inflibnet.ac.in/bitstream/10603/171122/2/11.%20chapter%203.pdf>

Figura 49. Representación matricial de la imagen con valores de intensidad, elaboración propia.

Figura 50. Representación matricial de la imagen con valores de intensidad normalizados (I_n), elaboración propia.

Figura 51. Convolución de la imagen normalizada I_n y el filtro F_v , elaboración propia.

Figura 52. Imagen Resultado de la convolución, elaboración propia.

Figura 53. Resultado de la convolución con el filtro F_h , elaboración propia.

Figura 54. Función de Activación tipo ReLU, elaboración propia.

Figura 55. Función de Activación de tipo Sigmoidea, elaboración propia.

Figura 56. Saha, 2018, Extracción o Pooling.

Figura 57. Esquema de implementación de Red Neuronal Convolutiva 1D, elaboración propia.

Figura 58. Diagrama de Flujo de algoritmo CNN 1D, elaboración propia

Figura 59. Distribución de Datos de Entrenamiento, elaboración propia

Figura 60. Distribución de Datos de Validación, elaboración propia.

Figura 61. Condiciones Iniciales del Modelo, elaboración propia.

Figura 62-75. Pruebas variando diferentes parámetros, elaboración propia.

Figura 76. Robot Móvil propuesto, elaboración propia.

Figura 77. Acelerómetro Triaxial MPU6050. <https://www.osepp.com/electronic-modules/sensor-modules/116-accelerometer-gyro-breakout-balancing-module>

Figuras 78-79. NVIDIA Jetson Nano y Jetson Nano Developer Kit. <https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit#write>

Figura 80. Tarjeta de acondicionamiento STM32 Nucleo
<https://www.st.com/en/evaluation-tools/stm32-nucleo-boards.html>

Figura 81. Instalación del Hardware en Jetson Nano Developer Kit. Elaboración propia.

Figura 82. Diagrama de conexiones de la Tarjeta STM32 Nucleo64 G431RB y el acelerómetro triaxial MPU 6050. https://github.com/candle-usb/candleLight_fw/issues/21

Figura 83. Instalación del Hardware.

Figura 84. Ventana de inicio del Software STM32 CubeMX:
<https://www.st.com/en/development-tools/stm32cubemx.html>

Figura 85. Elección de la tarjeta STM32.

Figura 86. Configuración de puertos, comunicación, reloj y proyecto.

Figura 87. Ventana de inicio del Software 4 STM32. <https://www.st.com/en/development-tools/sw>

Figuras 88-91. Instrucciones para la medición de la vibración en bombas y motores, elaboración propia.



ÍNDICE DE TABLAS

Tabla 1. Isermann, 2011, pág. 157

Tabla 2. Saddiqui, 2016, pág. 2

Tabla 3. ISO 10816-1. Límites en velocidad de vibración para máquinas

Tabla 4. Corrección de las envolventes utilizando la Interpolación Lineal Local, elaboración propia.

Tabla 5. Tonos encontrados en la señal, elaboración propia.

Tabla 6-7. IMF encontradas, elaboración propia.

Tabla 8. Señales IMF de motor en estado normal, elaboración propia.

Tabla 9. Señales IMF de motor con fallas en rodamientos, elaboración propia.

Tabla 10. Tabla Comparativa Redes Neuronales y Redes neuronales Convolucionales

Tabla 11. Parámetros de la función Convolución 1D, elaboración propia.

Tabla 12. MAFAULDA, 2020, Base de Datos MAFAULDA

Tabla 13-26. Resultados de las pruebas modificando los parámetros del modelo, elaboración propia.

Tabla 27. Especificaciones del acelerómetro triaxial MPU 6050, elaboración propia.

Tabla 28. Especificaciones de la computadora NVIDIA Jetson nano, elaboración propia.

Tabla 29. Especificaciones de la tarjeta de acondicionamiento de señal, elaboración propia.

Tabla 30. Comparación de tiempos de procesamiento

ÍNDICE DE ACRÓNIMOS

ABVT	Alineamiento-Balanceo-Vibración
AC	Corriente Alterna
ADADELTA	Método de Ratio de Aprendizaje Adaptativo
ADAM	Estimador de Modo Adaptativo
BPFI	Frecuencia de deterioro de pista interior
BPFO	Frecuencia de deterioro de pista exterior
BSF	Frecuencia de deterioro de elementos rodantes
CNN	Red Neuronal Convolutiva
CPM	Ciclos por minuto
CPU	Unidad de Procesamiento Central
CV	Caballos de Vapor
DC	Corriente Directa
ECG	Electrocardiograma
EMD	Deconvolución Empírica de Modos
FTF	Frecuencia de deterioro de jaula
GB	Giga Byte
GPU	Unidad de Procesamiento Gráfico
HHT	Transformada de Hilbert Huang
HMI	Interfaz Humano Máquina
HP	Caballos de Fuerza
I2C	Circuitos inter-integrados
IEEE	Instituto de Ingenieros de Electricidad y Electrónica
IEPE	Electrónica Integrada Piezo Eléctrica
IMF	Función de Modo Intrínseco
IMI	Instrumento Industrial de Monitoreo
ISO	Organización Internacional de Estandarización
MAFAULDA	Base de datos de Fallas en Máquinas
MAS	Movimiento Armónico Simple

MFS	Simulador de Fallas en Máquinas
MSE	Error de Media Cuadrada
MW	Mega Vattios
NN	Red Neuronal
PC	Computador Personal
RMS	Valor eficaz, Raíz Media Cuadrática
RPM	Revoluciones por minuto
RPM	Revoluciones por minuto
USB	Universal Serial Bus



Introducción

En la actualidad, las industrias buscan aumentar la productividad y la eficiencia de sus procesos, mediante la reducción de costos y tiempos de parada, asegurando la calidad del producto y la seguridad del personal, los equipos y el medio ambiente. Las fallas que ocurren en los procesos industriales pueden ser determinantes en su rendimiento, principalmente cuando se presentan en equipos y componentes críticos como las bombas y motores. Estas fallas pueden ser prevenidas a través de un adecuado plan de mantenimiento predictivo, evitando sobrecostos por paradas imprevistas de los equipos y garantizando la calidad del producto final. Lamentablemente cuando las fallas se presentan en lugares inaccesibles o que ponen en riesgo la salud de los trabajadores, la tarea del mantenimiento se hace más complicada. Por esta razón, se resalta la importancia del uso de los robots para la adquisición de datos, se describen las metodologías de procesamiento de datos mediante análisis vibracional, se presentan las metodologías de diagnóstico de fallas existentes como el basado en modelos y el basado en datos, y se introducen los conceptos de Inteligencia Artificial aplicada a la detección de fallas.

En el Capítulo 1, se presenta la problemática de las fallas en bombas y motores industriales, la normativa utilizada para la clasificación de su condición, las estrategias de mantenimiento y metodologías de solución tradicionales, y el estado del arte en metodologías de análisis vibracional. Además, se presentan ejemplos de cómo los robots móviles son utilizados para tareas que ponen en peligro la vida de las personas. El capítulo termina con una breve introducción a la Inteligencia Artificial.

En el Capítulo 2, se describen los algoritmos de análisis vibracional como la Transformada Rápida de Fourier, Wavelets y la Transformada de Hilbert Huang, mencionando las ventajas y desventajas de cada una. Se presenta el procedimiento planteado para el preprocesamiento de datos, utilizando como base, el algoritmo de Hilbert-Huang y las Funciones de Modo Intrínseco, las cuales serán las entradas de la Red Neuronal Convolutiva.

En el Capítulo 3, se hace una introducción al Aprendizaje Asistido, Aprendizaje Profundo y Red Neuronal Convolutiva. Luego se describen las etapas de la metodología y su implementación utilizando Python, Keras y TensorFlow. Además, se presentan los resultados de la evaluación del algoritmo utilizando la Base de Datos MAFAULDA.

Finalmente, en el Capítulo 4, se describen los sensores que se utilizarán, la tarjeta de acondicionamiento y el controlador, además de su implementación en el Robot Móvil. Se presentan los pasos para la instalación del sistema operativo en el controlador NVIDIA Jetson Nano y un análisis del tiempo de procesamiento.



Capítulo 1. Estado del arte en metodologías de diagnóstico de fallas para componentes de los procesos industriales

1.1. Introducción

En este capítulo se presenta una introducción a los conceptos de falla y diagnóstico de fallas, se describen las fallas que se presentan en los equipos de los procesos industriales, tales como: bombas y motores, así como, sus frecuencias de ocurrencia, causas y consecuencias, y se realiza un estudio del estado del arte en el procesamiento de datos para un diagnóstico de fallas con el uso de Inteligencia Artificial mediante el análisis vibracional. Además, se describe la necesidad y la importancia del uso de robots como medio para medir las señales de vibración. Finalmente, se propone el objetivo principal a alcanzar y los objetivos específicos que servirán de hitos para lograrlo.

1.2. Fallas en componentes de los procesos industriales

En todo proceso industrial se busca maximizar el rendimiento y la eficiencia de máquinas y equipos para cumplir con los estándares de calidad del producto final, garantizando la seguridad de los trabajadores y el cuidado del medio ambiente (Isermann, 2006, 2011). Para ello, es necesario asegurar el correcto funcionamiento de los componentes que participan en los procesos, como válvulas, tanques, calderas, motores y otros, mediante un plan de mantenimiento y una inspección eficiente que permita identificar y reportar fallas, con el fin de asegurar la calidad del producto y evitar paradas imprevistas, tiempos muertos y peligros que pongan en riesgo la vida de las personas y el medio ambiente.

Un proceso industrial automatizado, mostrado en la Figura 1, presenta diferentes niveles de control (Isermann, 2011). Generalmente, el primer nivel presenta controladores secuenciales y de realimentación para asegurar el correcto funcionamiento del proceso. El siguiente nivel contiene funciones de supervisión que garantizan la confiabilidad y seguridad del primer nivel. A medida que los controladores del primer nivel se hacen más sofisticados, la capa de supervisión debe mejorar y ser más eficiente, dado que aún se pueden presentar fallas.

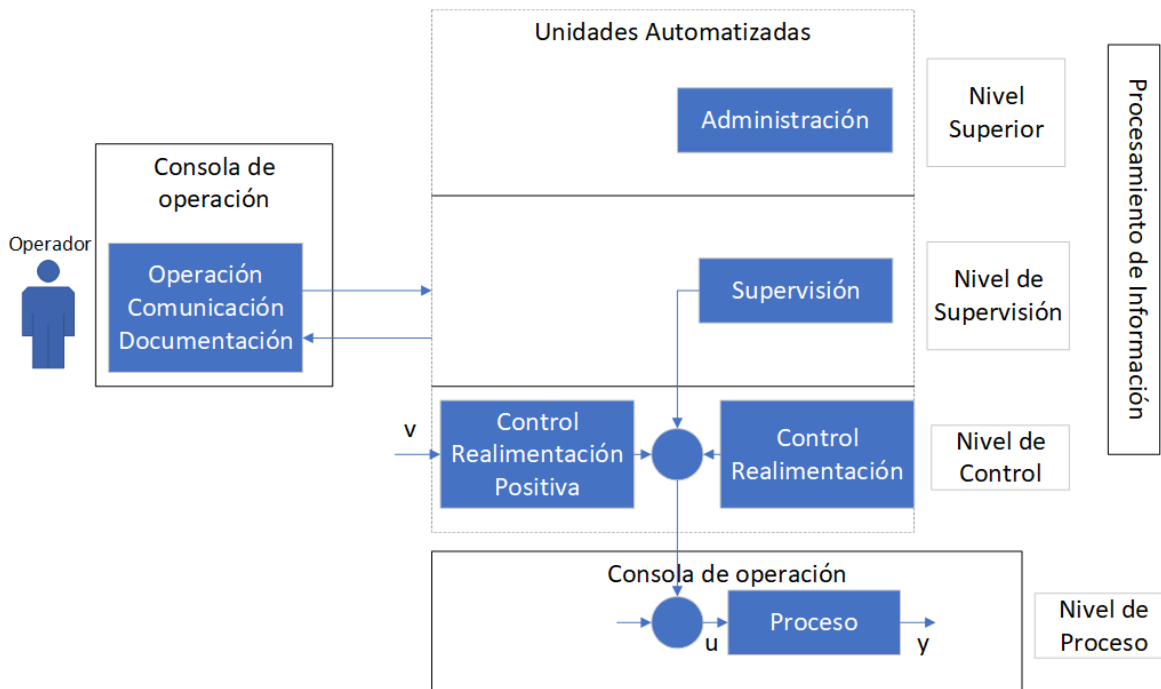


Figura 1. Esquema simplificado de un proceso de automatización. Fuente: Isemann 2011, pág. 19

En este contexto, las fallas son comportamientos inusuales de los procesos, equipos, máquinas y sus componentes, que se presentan por errores de operación, errores de diseño, deterioro debido al uso, el paso del tiempo, perturbaciones, etc. Muchas veces, una falla es detectada por el operador cuando esta ya es grave, el riesgo de accidentes es alto o la corrección implica un gasto considerable. Por estas razones, su detección temprana es importante; sin embargo, esta es complicada dado que pueden ser imperceptibles para un operador o presentarse en lugares de difícil acceso (Tadokoro, 2013).

Para lograr esto, los sistemas de control y supervisión se hacen cada vez más sofisticados y complejos, y el personal necesita mayor entrenamiento y conocimiento de las diversas herramientas. La complejidad de estos sistemas introduce fallas que afectan el rendimiento del proceso (Gharahbagheri, 2016), por lo cual, es indispensable un sistema eficiente, robusto y confiable de diagnóstico de fallas para su detección y respuesta inmediata.

1.2.1. Terminología

Según la Real Academia Española (Real Academia Española, 2018), una falla es un defecto o una falta. En (Isermann, 2011), el autor describe una falla como la desviación de al menos una propiedad del sistema para alcanzar una condición aceptable. Una falla puede provocar una avería o un malfuncionamiento. Una avería es una interrupción permanente, mientras que un malfuncionamiento es una irregularidad intermitente del sistema. Presenta también, los conceptos: detección de fallas y diagnóstico de fallas. La detección de fallas es la acción de reconocer la ocurrencia de una falla. El diagnóstico de fallas incluye la detección de la falla, además de un análisis de los síntomas del sistema para obtener, mediante métodos de diagnóstico, información sobre el tipo, la ubicación, la severidad y el tiempo de la falla. Las fallas pueden categorizarse en función al tiempo; éstas pueden ser fallas abruptas, incipientes o intermitentes. Considerando el modelo del proceso, las fallas pueden ser clasificadas como aditivas cuando la falla se suma con salida del sistema o multiplicativas cuando la falla multiplica a la salida.

1.3. Componentes industriales a evaluar

1.3.1. Bombas

En la mayoría de los casos, las bombas utilizan un motor eléctrico o mecanismo de engranes para succionar el fluido por una boquilla de entrada y expulsarlo por una boquilla de salida (Isermann, 2011). Comúnmente en las industrias químicas, petroquímicas, mineras, energéticas, manufacturas, etc., se utilizan las bombas para el transporte de fluidos (Isermann, 2011). Debido a su amplio uso en la industria, se puede categorizar a estos componentes como críticos dentro de un correcto funcionamiento de las plantas industriales y sus procesos, según la definición dada por Scheffer (Scheffer, 2004).

Las bombas se clasifican en dos categorías:

1. **Bombas centrífugas (o cinéticas)**

Tienen un impulsor giratorio que agrega energía centrífuga al fluido expulsándolo por la boquilla de salida. Se caracterizan por trabajar con bajas presiones y altos flujos. Las Figuras 2, 3 y 4 muestran la bomba centrífuga y sus partes.



Figura 2. Bomba centrífuga, 0.5 HP, 1"x1",
Fuente:
<https://www.maestro.com.pe/productos/gasfiteria/bombas/bomba-centrifuga-05hp-1-x-1>

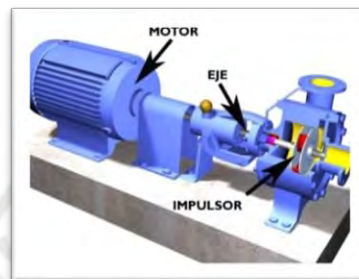


Figura 3. Partes de una bomba centrífuga, Fuente:
<http://www.cindex.com.mx/bombas-ces-el-funcionamiento-de-una-bomba-centrifuga/>

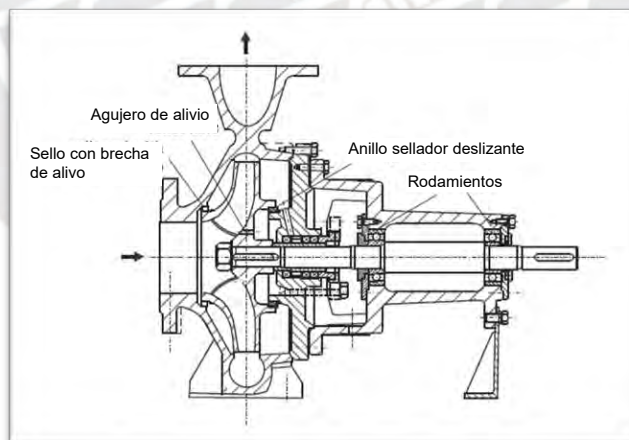


Figura 4. Diagrama seccional de una bomba centrífuga. Fuente: Isermann 2006, pág 158

En (Isermann, 2011) el autor presenta una tabla con las fallas más frecuentes en las bombas centrífugas.

Componentes con falla	Frecuencia Registrada (%)
Sello del anillo deslizante	31
Rodamientos	22
Fugas	10
Motor de accionamiento	10
Rotor	9
Cojinetes deslizantes	8
Embrague	4
Tubo dividido	3
Chasis	3

Tabla 1. Componentes con fallas y frecuencias de ocurrencia. Fuente: Isemann 2011, pág. 157

En la Figura 5, se presentan las fallas mencionadas en una distribución de tipo pastel.

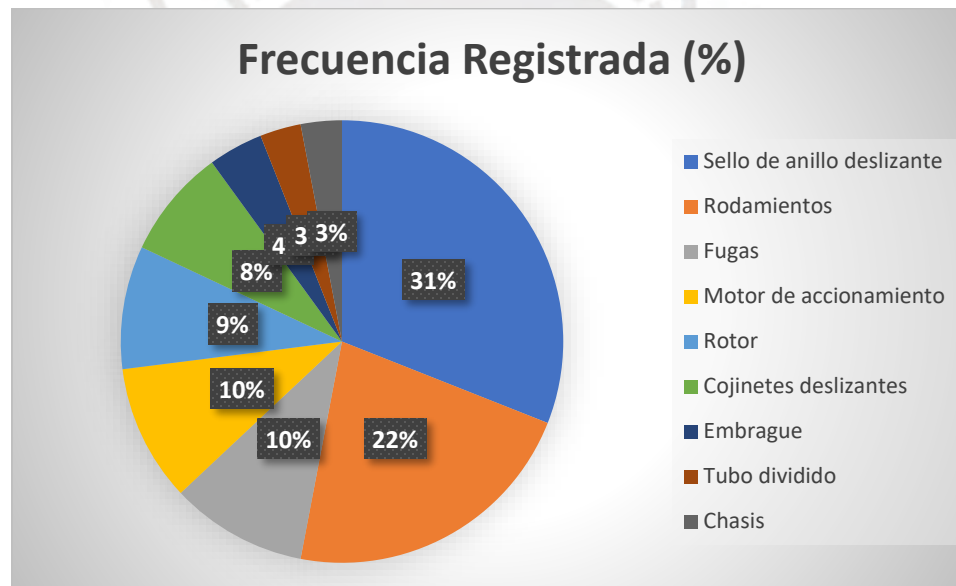


Figura 5. Diagrama de pastel con la distribución de las frecuencias. Elaboración propia.

Algunos tipos de malos funcionamientos de las bombas son:

Cavitación: cuando aparecen burbujas de vapor dentro del fluido debido a una caída de la presión. Estas burbujas pueden perjudicar el rendimiento de la bomba debido a que daña las aspas del impulsor.

Gas en fluidos: cuando ocurre una caída en la presión, pueden aparecer gases indeseados en los líquidos transportados.

Corrida seca: la falta de enfriamiento y el sobrecalentamiento de los cojinetes pueden ocasionar este problema.

Desgastes: debido a la erosión, corrosión, instalación.

Depósitos: bajo rendimiento debido a materiales orgánicos.

Oscilaciones: desbalance en el rotor debido al daño o depósitos en el rotor, ocasiona daños en los cojinetes.

Para conseguir un monitoreo adecuado del correcto funcionamiento de las bombas es necesario medir las presiones de entrada y salida, flujos, velocidades y temperaturas de los cojinetes (Isermann, 2011). Con estas medidas se pueden establecer límites para encontrar problemas de malfuncionamientos como corrida seca, gas indeseado, depósitos o cojinetes. Pero no es posible establecer un diagnóstico eficiente de fallas.

Mediante la medición con sensores de vibración se pueden evaluar fallas bajo condiciones de operación definidas. También se puede realizar un control mediante metodologías basadas en modelos.

Bombas de desplazamiento positivo (o hidrostáticas)

1.1. Bombas recíprocas:

Las bombas recíprocas se caracterizan por trabajar con altas presiones y bajos flujos. Son utilizadas en industrias químicas y farmacéuticas (Isermann, 2011). La presión puede llegar hasta los 3000 bar y el flujo volumétrico está dentro de 0.1 ml/h y 1000 m³/h y su potencia puede llegar a alcanzar 1MW. Las Figuras 6 y 7 muestran ejemplos de bombas de desplazamiento positivo.



Figura 6. Bomba recíprocante de desplazamiento positivo, Fuente: <https://www.miltonroy.com/es/productos/bombas-de-proceso/bomba-reciprocante-de-desplazamiento-positivo-poweroyal/>

Dentro de las bombas recíprocas se pueden clasificar los siguientes tipos:

- de pistón,
- de diafragma, y
- de émbolo.

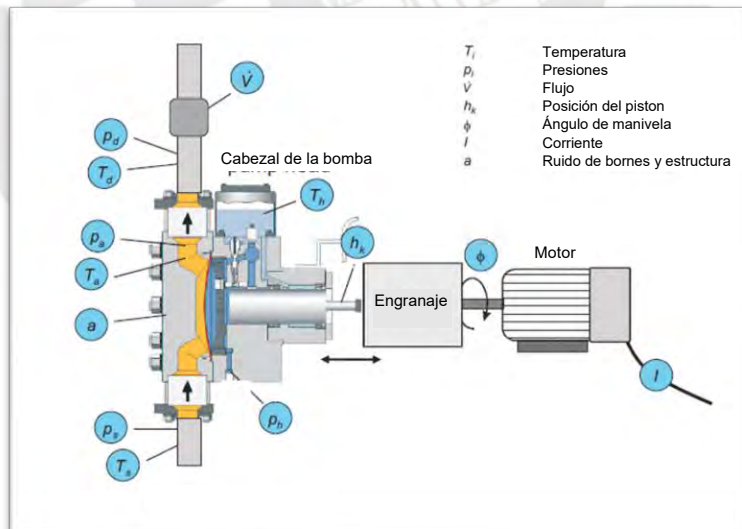


Figura 7. Esquema de una bomba de diafragma. Fuente: Isermann 2011, pág 184

1.3.2. Motores

Los motores son máquinas de amplio uso en la industria. Su funcionamiento básico consta de dos elementos: estator y rotor. El estator es la parte fija que recibe energía eléctrica y la convierte en campo magnético y el rotor es la parte giratoria. El motor es un componente importante para el funcionamiento de las bombas.

Existen varios tipos de motores entre los que encontramos tres categorías:

- i. Motores DC
 - En serie
 - En paralelo
 - De campo permanente
- ii. Motores AC Trifásicos
 - De inducción (asíncronos)
 - Síncronos
- iii. Motores AC Monofásicos
 - Conmutador (Motor universal)
 - Jaula de ardilla

Nos enfocaremos en los motores AC Trifásicos de inducción debido a su amplio uso en la industria. Los motores de Inducción AC Trifásicos consisten en tres bobinados en el estator interconectados con las 3 fases del voltaje de alimentación y con conexión tipo Delta o Estrella (Y). La velocidad angular del rotor se genera debido a el campo magnético producido por los bobinados. El motor de inducción se muestra en las Figuras 8 y 9.



Figura 8. Motor AC Asíncrono Trifásico, Fuente: <https://es.aliexpress.com/item/ac-three-phase-asynchronous-motor/1212888134.html>

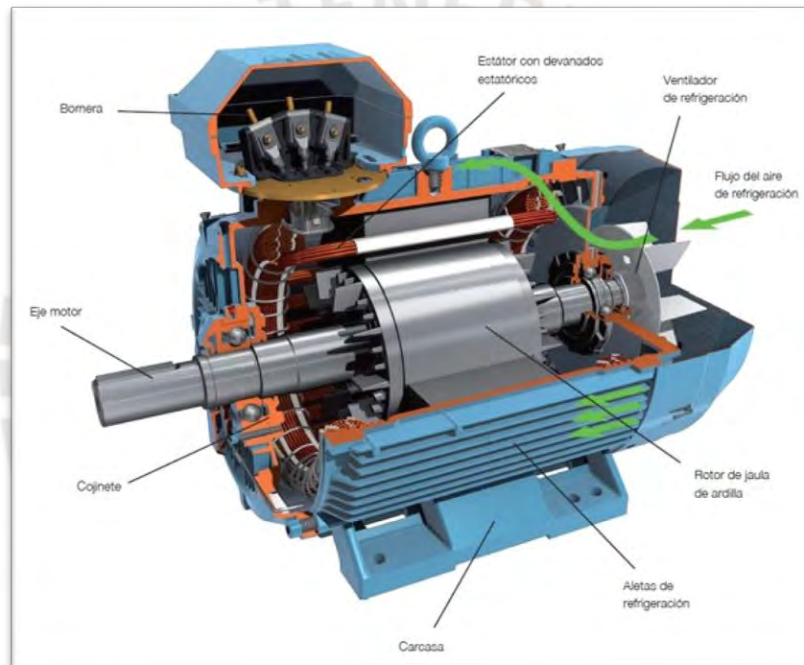


Figura 9. Partes de un motor de inducción, Fuente: <https://maquinaselectricasblog.wordpress.com/motor-de-induccion/>

En (Siddiqui y Sahay, 2014), los autores clasifican las fallas en los motores de inducción como se muestra en la Figura 10. Se describe también el procedimiento utilizado para la evaluación de fallas se muestra en la Figura 11. Finalmente, presenta las fallas más comunes en motores de inducción y sus frecuencias.

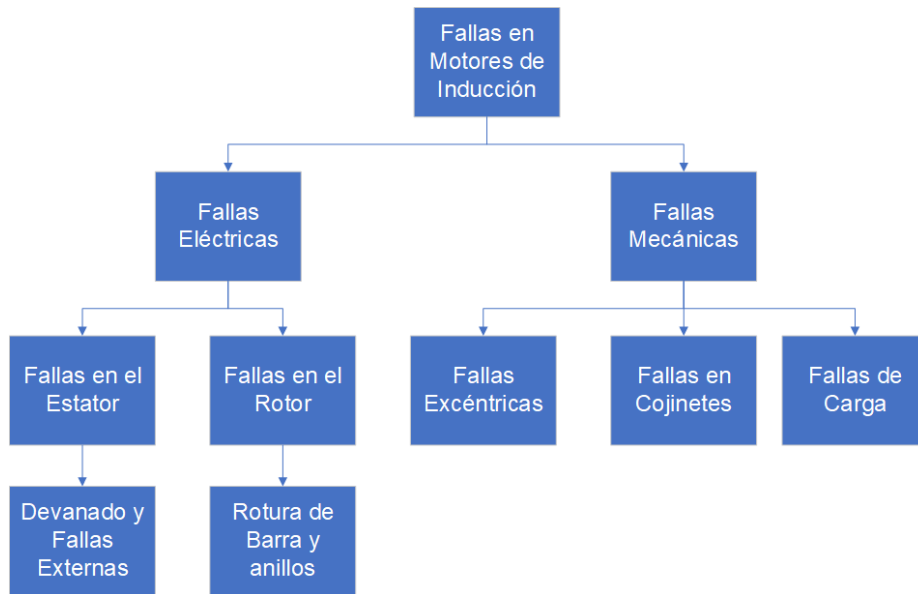


Figura 10. Clasificación de las fallas en los motores de inducción. Fuente: Siddiqui y Sahai, 2014 pág. 3

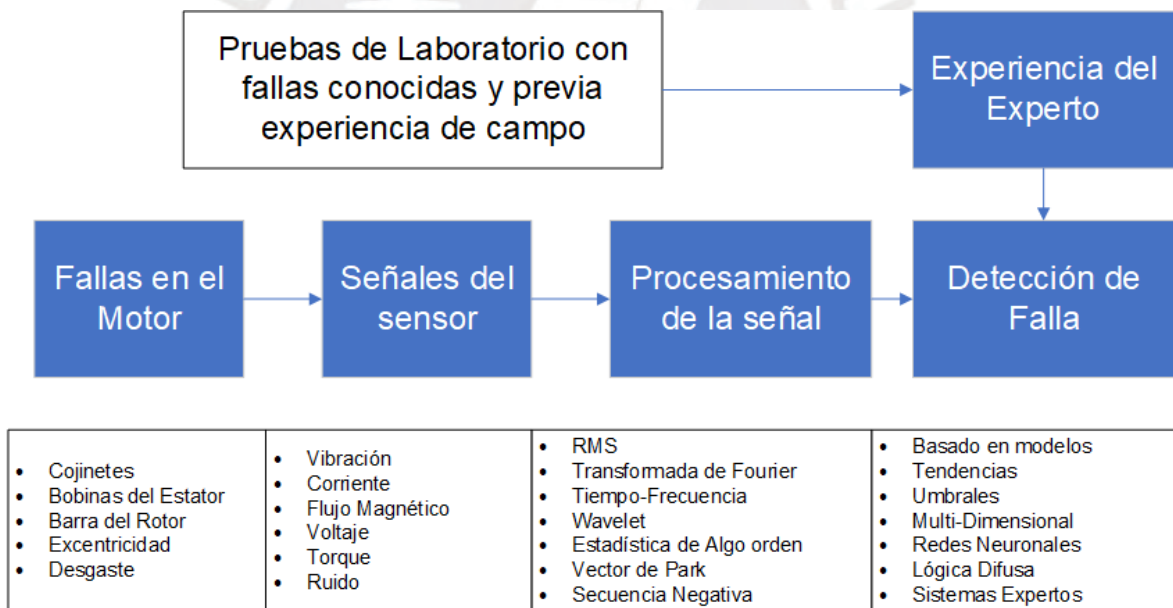


Figura 11. Diagrama de bloques del proceso de monitoreo en-línea. Fuente: Siddiqui y Sahai, 2014, pág. 13

Además, mencionan los componentes y frecuencias de fallas en estas máquinas, según datos proporcionados por la IEEE. Las fallas más frecuentes en las bombas centrífugas se muestran en la tabla siguiente y en la Figura 12.

Componentes con falla	Frecuencia Registrada (%)
Cojinetes	44
Fallas en estator	26
Fallas en rotor	8
Otros	22

Tabla 2. Componentes del motor de inducción que presentan fallas y sus frecuencias. Fuente: Siddiqui y Sahay, 2014, pág. 2

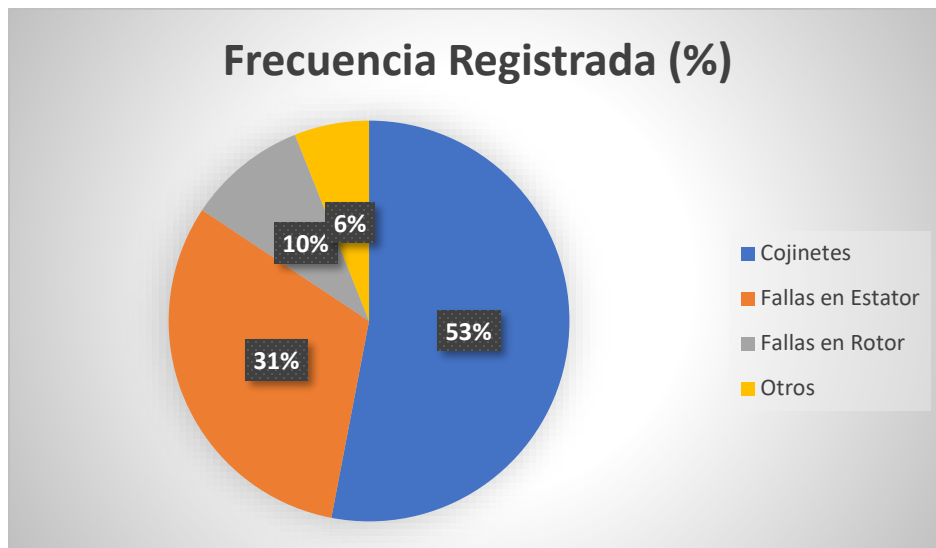


Figura 12. Diagrama de pastel de las fallas en motores de inducción. Elaboración propia.

1.3.3. Normativa

Máquinas vibratoriales

La evaluación de fallas en motores y máquinas de vibración está regulada bajo las normativas ISO 10816-1 (ISO, 1995). A continuación, se muestra una tabla con los límites en velocidad de vibración para las máquinas según su categoría.

Velocidad de Vibración R.M.S
(mm/s)

	Clase I	Clase II	Clase III	Clase IV
0.28	A	A	A	A
0.45				
0.71				
1.12	B	A	A	
1.80				
2.80	C	B	B	A
4.50				
7.10	D	C	C	B
11.20				
18.00	D	D	D	C
28.00				
45.00			D	D

Tabla 3. Límites en velocidad de vibración, Fuente: ISO 10816-1

Donde:

Zona A: La vibración de máquinas nuevas normalmente debería estar en esta zona.

Zona B: Máquinas con vibración dentro de esta zona son consideradas aceptables por un período largo de operación.

Zona C: Máquinas con vibración dentro de esta zona son consideradas insatisfactorias por un período de largo de operación continua. Generalmente, esta máquina debe operar por un período limitado en esta condición hasta su reparación.

Zona D: Los valores de vibración dentro de esta zona son normalmente consideradas lo suficientemente severas para causar daño a la máquina.

Clasificación de las máquinas:

Clase I: Partes individuales de motores y máquinas, conectadas a la máquina completa en su condición de operación normal. (Motores Eléctricos de Producción **hasta los 15 kW** son ejemplos típicos de máquinas en esta categoría) (ISO, 1995)

Clase II: Máquinas medianas (típicamente motores eléctricos **entre los 15kW y 75kW** de salida) sin cimientos especiales, motores o máquinas rígidas montadas (**hasta los 300 kW**) en cimientos especiales. (ISO, 1995)

Clase III: Máquinas grandes para mover carga y otras con masas rotatorias montadas con cimientos pesados y rígidos los cuales son relativamente rígidos en la dirección de la medición de vibración. (ISO, 1995)

Clase IV: Máquinas grandes para mover carga y otras con masas rotatorias montadas con cimientos relativamente suaves en la dirección de las mediciones de vibración (por ejemplo, turbogeneradores y turbinas de gas con salidas mayores a 10 MW). (ISO, 1995)

Además, cabe resaltar que la normativa ISO 10816-1 brinda las instrucciones para realizar las mediciones de manera adecuada, estableciendo las señales a medir, la instrumentación necesaria y ubicación de los puntos adecuados para realizar la medición.

1.4. Estrategias de Mantenimiento

Las estrategias, o también llamadas filosofías de mantenimiento (Scheffer, 2004), son acciones a tomar para garantizar el correcto funcionamiento de los componentes del sistema. Podemos distinguir tres tipos de estrategias de mantenimiento, según A-MAQ (A-MAQ, 2019), las cuales serán descritas a continuación.

1.4.1. Mantenimiento Correctivo

El mantenimiento correctivo se realiza cuando se presentan las fallas, averías y malfuncionamientos. Previamente, no se realiza ninguna acción de prevención o control. Una de las ventajas de esta filosofía de mantenimiento es que, dado que las máquinas operan hasta llegar a su límite, se logran alcanzar rendimientos altos de producción. Además, no se tienen costos de mantenimiento previos. Como desventajas, se tiene que, al exigir el máximo de los equipos, estos pueden acortar sus tiempos de vida y presentar averías que resulten en reparaciones muy costosas. Las máquinas averiadas o con comportamiento errático reducen la calidad de los productos finales y pueden parar la producción de la industria ocasionando a la larga, costos mucho mayores al ahorrado por no ejecutar acciones de control.

1.4.2. Mantenimiento Preventivo

El mantenimiento preventivo establece períodos de mantenimiento según un análisis previo de los equipos. Se conoce y se programan tiempos de parada en la producción, en donde las máquinas son revisadas y reparadas para continuar con su operación normal. Este enfoque presenta ventajas que corrigen las desventajas del mantenimiento correctivo, al evaluar las máquinas periódicamente, se logran reducir la ocurrencia de fallas y averías. Sin embargo, presenta desventajas como la intervención en equipos que no requieren mantenimiento, aumentando así, la probabilidad de fallas y los costos que se asumen para realizar esta tarea de mantenimiento preventivo: horas hombre, materiales y equipos de inspección y horas de parada.

1.4.3. Mantenimiento Predictivo

En el mantenimiento predictivo se analizan las máquinas y equipos mediante un monitoreo de variables específicas que predicen el comportamiento y la aparición de fallas. Esta inspección se realiza con frecuencia y, aplicando la tecnología y la experiencia necesaria. En base a los resultados reportados, se planifican paradas de mantenimiento. Entre las ventajas de esta filosofía tenemos que sólo se corrigen los equipos y máquinas que lo necesitan, mediante un seguimiento del rendimiento de los equipos se pueden prevenir fallas y averías. Como desventaja se tiene el costo en la implementación de estos controles.

Para el mantenimiento predictivo de fallas en procesos industriales, existen dos alternativas. (Lu, 2019). La primera consiste en una inspección periódica manual realizada por los operadores, utilizando equipos de instrumentación. Esta alternativa requiere que el operador esté capacitado y tenga la experiencia suficiente para determinar si ocurre una falla.



Figura 13. Inspección Manual, Fuente: <https://www.dmc.pt/es/analise-de-vibracoes-em-motores-eletricos/>

La segunda alternativa consiste en una red de sensores en línea, que monitorean procesos claves y recopilan la información en un servidor para su posterior análisis. Esta alternativa es costosa debido a que se necesita de una red de sensores distribuida en la planta.

En la industria se requiere que los diversos componentes trabajen a su máxima capacidad y garantizar la calidad del producto, por ello la necesidad de un diagnóstico oportuno de las fallas ocasionadas por la vibración en motores y bombas, para evitar paradas innecesarias,

sobrecostos y poner en riesgo la seguridad. El mantenimiento predictivo es necesario para diagnosticar las fallas descritas anteriormente, pero como se mencionó, existen casos en donde el acceso es restringido para un operador debido al espacio y al riesgo para la salud. La tesis propondrá una solución a esta problemática mediante un algoritmo que integre procesamiento de señales e Inteligencia Artificial utilizando un robot como medio para la toma de datos.

1.4.4. Mantenimiento predictivo mediante Análisis Vibracional

Análisis Vibracional

Vibración

La vibración es el movimiento oscilatorio de un cuerpo alrededor de una posición de equilibrio (White, 2010). Esta vibración se considera de cuerpo entero cuando todas las partes de este se mueven en la misma dirección en todo momento. Este tipo de vibración puede ser descrita como la combinación de movimientos individuales en los 6 grados de libertad: direcciones ortogonales y rotacionales x,y y z. La vibración es causada por una fuerza de excitación, la cual puede venir de una fuente externa o ser propia del cuerpo.

Movimiento Armónico Simple

El Movimiento Armónico Simple (MAS) es el desplazamiento periódico en una dirección de una masa controlada por un único resorte. Si una masa es desplazada del centro de equilibrio y después soltada dentro de un sistema sin fricción, la masa oscilará en un movimiento ondulatorio periódico tal como se muestra en la Figura 14.

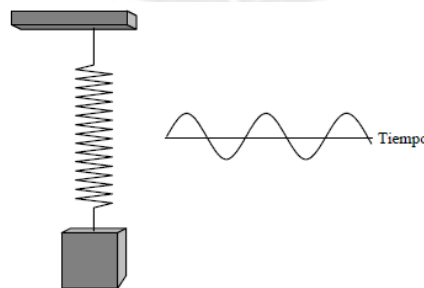


Figura 14. Movimiento Armónico Simple. Fuente: White, 2010, pág. 17

La señal que representa este movimiento ideal es:

$$d = A \operatorname{sen}(\omega t + \theta) \dots \text{(Ecuación 1.1)}$$

Donde:

d = desplazamiento instantáneo

A = desplazamiento máximo

t = tiempo

$$\omega = 2\pi f$$

θ = fase

De esta señal podemos encontrar características como son el período (T) y frecuencia de la onda ($f = 1/T$).

Vibración de las máquinas

Por la propia naturaleza de las máquinas rotativas, estas tienen que vibrar (Sarmiento, 2020). Esta vibración cambia cuando el estado de una máquina se altera, por lo que mediante el análisis vibracional es posible detectar y diagnosticar una amplia gama de fallas, a diferencia de otras técnicas de detección de fallas como ultrasonido y termografía.

La vibración presenta las siguientes características:

Amplitud

La amplitud de la vibración varía dependiendo de la variable a mostrar:

Desplazamiento $x = A \operatorname{sen}(\omega t) \dots \text{(Ecuación 1.2)}$

Velocidad $x = A \omega \operatorname{sen}(\omega t) \dots \text{(Ecuación 1.3)}$

Aceleración $x = -A \omega^2 \operatorname{sen}(\omega t) \dots \text{(Ecuación 1.4)}$

La señal de vibración de las máquinas puede representarse como un valor global. Este valor global puede presentarse como:

Valor Pico – Es el valor máximo positivo. Se utiliza cuando se trabaja con amplitudes de Aceleración.

Valor Pico-Pico – Es la diferencia de los valores máximo y mínimo. Se utiliza cuando se trabaja con Desplazamiento

Valor RMS (Velocidad) – Representa el valor máximo positivo por raíz de 2

Frecuencia

La frecuencia es un parámetro que indica la cantidad de eventos en relación al tiempo. En este caso los eventos son los periodos de onda. La unidad es Hertz (Hz). Sin embargo, para el análisis vibracional es usual trabajar en unidades de Ciclos Por Minuto (CPM), lo cual es llevar las unidades de Hz en segundos a minutos con la multiplicación por el factor de 60. Las máquinas rotativas presentan una frecuencia característica conocida como Revoluciones Por Minuto (RPM), la cual nos indica cuántas vueltas ha dado el rotor en 1 minuto. Las RPM son un dato proporcionado por el fabricante, pero este puede no ser exacto con las revoluciones reales de la máquina (Sarmiento, 2020). Por lo que se suele realizar una medición con un tacómetro o mediante luz estroboscópica. A esta frecuencia de rotación precisa y real, se le conoce como Frecuencia Fundamental y se representa como 1X.

Armónicos

La señal de vibración presenta armónicos o señales con frecuencias múltiplos enteros a la frecuencia fundamental, cuando empiezan a presentarse fallas (Sarmiento, 2020). Estas señales armónicas serán importantes cuando se detecten las fallas utilizando las tablas estandarizadas.

Fase

La fase es la desviación de una onda con respecto al cero. A pesar de tratarse de un distanciamiento o desfase entre dos señales, la fase es una medición del ángulo, en unidades de grados sexagesimales o radianes (Sarmiento, 2020).

La Figura 15 muestra dos señales sinusoidales con desfase. La señal de color roja está desfasada de la azul en 90°.

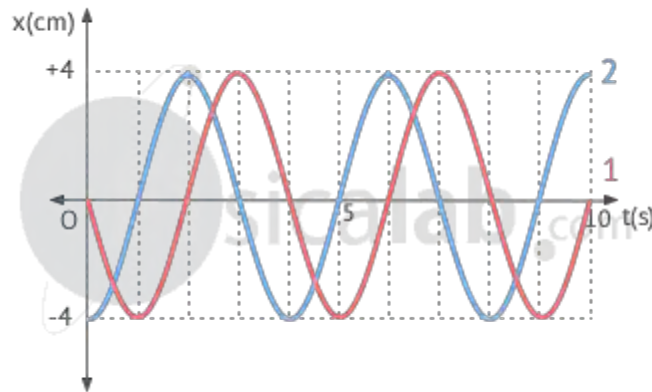


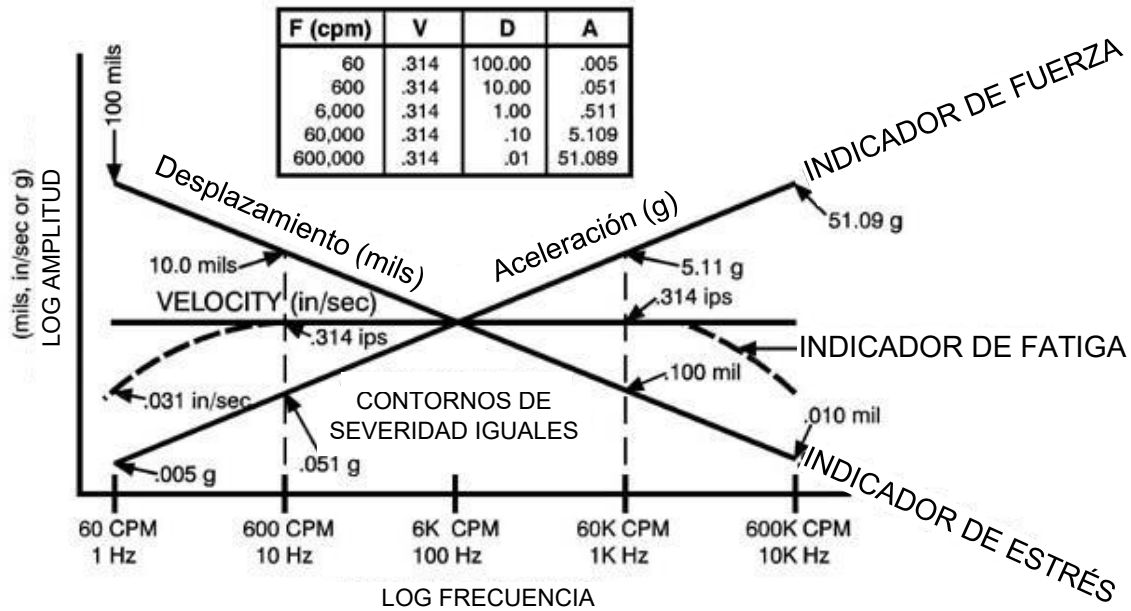
Figura 15. Ejemplo del desfase entre 2 señales, Fuente: <https://www.fiscalab.com/ejercicio/752e>:

Medición e Interpretación de la vibración

Para determinar cuál es la variable de vibración que se requiere reportar, se debe considerar la frecuencia a la que trabajará el equipo (Sarmiento, 2020). La Figura 16 muestra un gráfico en donde se muestra cómo elegir la variable de vibración a utilizar dependiendo de la frecuencia.

En base al gráfico se utilizan las siguientes reglas prácticas:

- Entre los 60 CPM a los 600 CPM medir el desplazamiento de la vibración. Estas frecuencias se presentan cuando tenemos máquinas grandes como molinos.
- Entre los 600 CPM a los 60,000 CPM medir la velocidad de la vibración. Estas frecuencias se presentan cuando tenemos fallas estructurales como, por ejemplo, desbalances y desalineamientos en motores.
- Para más de 60,000 CPM, medir la aceleración de la vibración. Por ejemplo, en fallas en motores, estas frecuencias muestran los problemas en rodamientos.



COMPARACIÓN DE DESPLAZAMIENTO, VELOCIDAD Y ACCELERACIÓN DE VIBRACIÓN

Figura 16. Comparación del Desplazamiento, Velocidad y Aceleración de Vibración. Fuente: Sarmiento, 2020

Para evaluar las fallas de rodamientos, se deben encontrar las frecuencias características de estos (Sarmiento, 2020):

- BPFO = Frecuencia de falla en pista del aro exterior
- BPFi = Frecuencia de falla en pista en el aro interior
- BSF= Frecuencia de falla de los elementos rodantes
- FTF = Frecuencia de falla de la jaula

En base a estas frecuencias se pueden distinguir 4 etapas del daño en rodamientos:

1. Primeros síntomas de fallas aparecen en rango de frecuencia de 250,000 a 350,000 Hz, luego caen a 20,000 – 60,000 Hz (1,200,000 a 3,600,000 CPM)

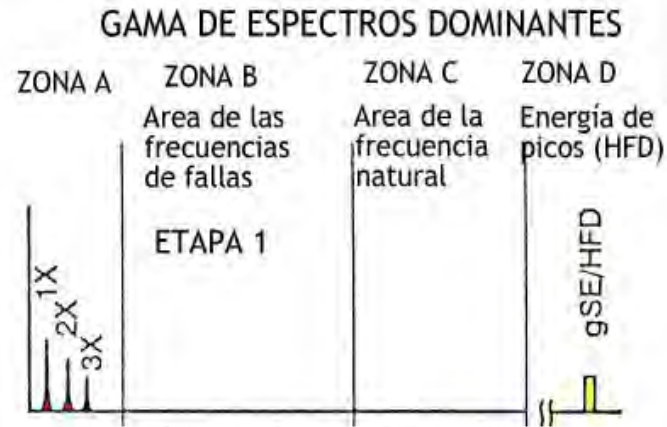


Figura 17. Primeros síntomas de falla en rodamientos. Fuente: Sarmiento, 2020

2. Los defectos del rodamiento generan la excitación de frecuencias naturales de los componentes entre 30K y 120K CPM.

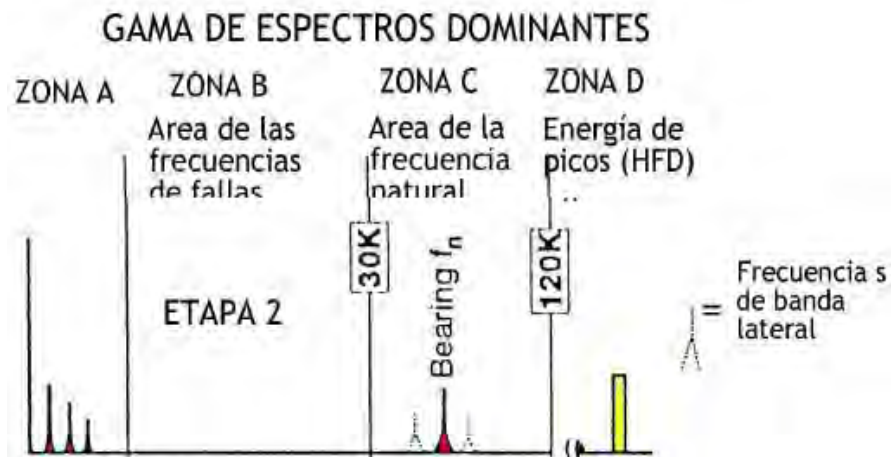


Figura 18. Segunda etapa de daños en rodamientos. Fuente: Sarmiento, 2020

3. Aparecen las frecuencias de defecto del rodamiento y sus armónicos.

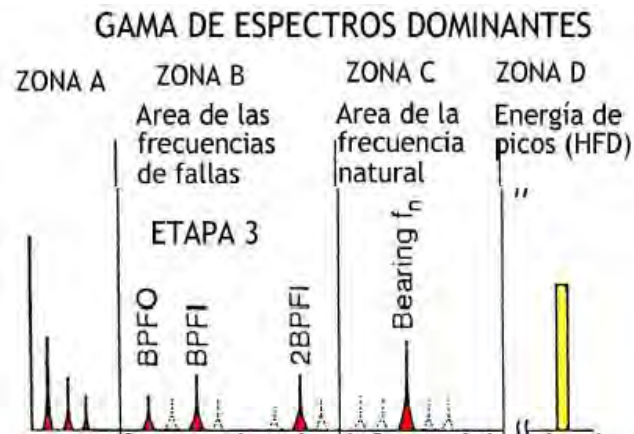


Figura 19. 3ra Etapa de daños en rodamientos. Fuente: Sarmiento, 2020

4. Frecuencia fundamental se ve afectada y se presentan armónicos. El ruido aumenta.

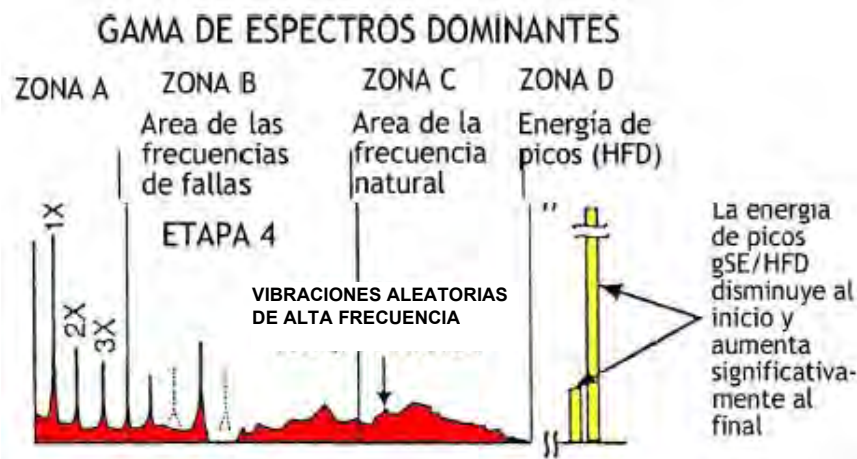


Figura 20. Etapa final de daños en rodamientos. Fuente: Sarmiento, 2020

Metodología para el análisis vibracional

De acuerdo a lo visto en (Sarmiento, 2020), a continuación, se describen los pasos para el análisis vibracional en máquinas rotatorias.

A. Medición Global

Para una medición global y clasificación de una máquina, se mide el valor global de vibración de esta.

Para ello se utiliza un vibrómetro, el cual tiene un sensor y una etapa de acondicionamiento que capturarán y prepararán la señal física en valores de voltaje el cual será registrado en el instrumento y arrojará un valor global.

B. Evaluación con Normativas

El valor registrado en el paso anterior es comparado con los valores de vibración permitidos según estándares internacionales como la ISO 10816.

De requerirse un estudio más detallado de la condición de la máquina, se deberá:

C. Recolección de señales de vibración

Utilizando un recolector de señales y un sensor, se recolectan las señales de vibración del equipo.

D. Evaluación por software

Se requiere de un software especializado para analizar el espectro de las señales medidas para determinar el tipo de falla presentada.

A continuación, se muestra en la Figura 21, un diagrama de flujo del procedimiento descrito.

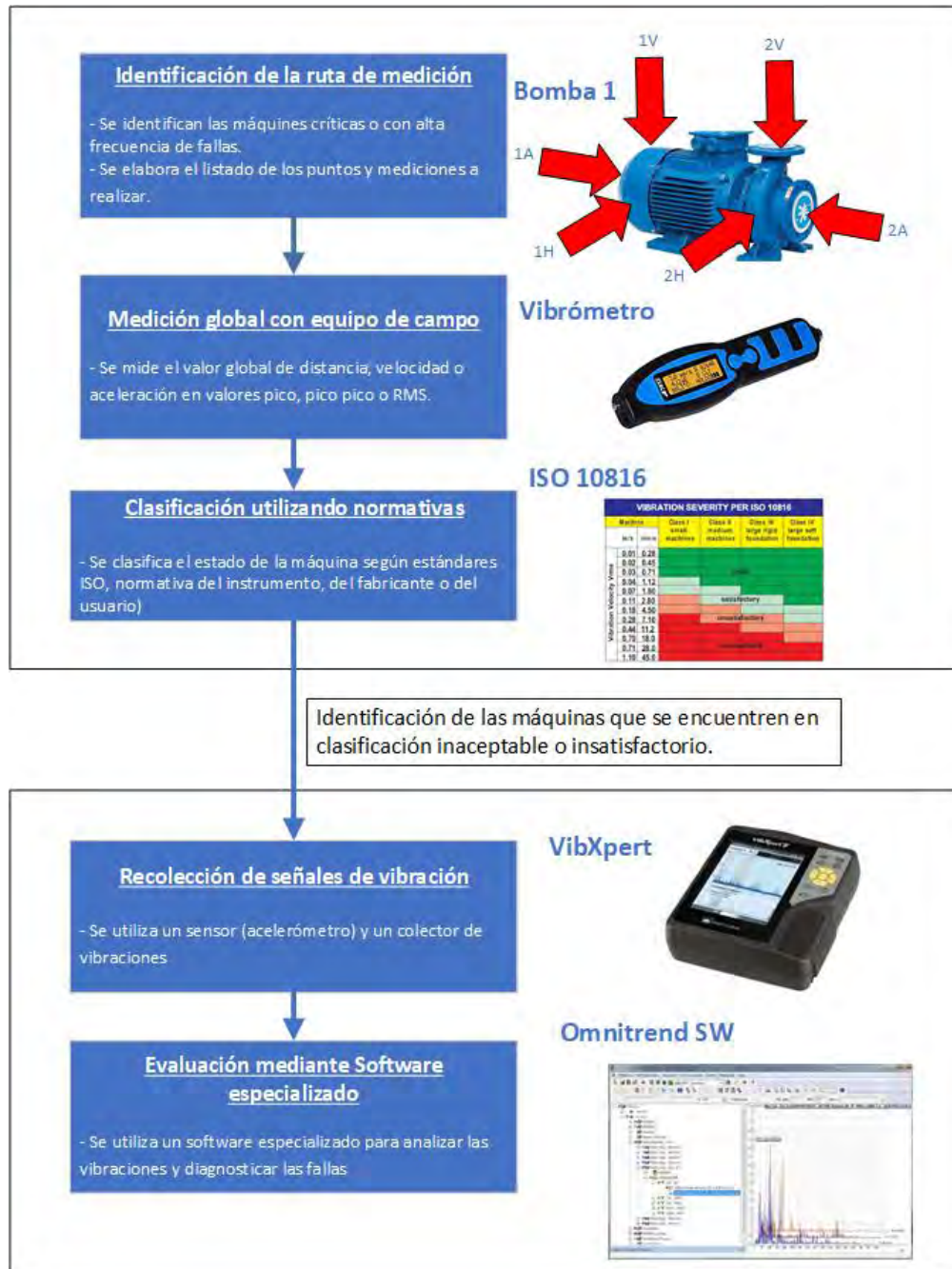


Figura 21. Procedimiento para el Análisis Vibracional. Elaboración propia

Comparación con Tablas estandarizadas

Para que el analista de vibraciones pueda dar un veredicto acerca del tipo de falla, se requiere la comparación con una tabla estandarizada (Sarmiento, 2020). Estas tablas contienen las diferentes fallas que se presentan generalmente en las máquinas y su espectro correspondiente. Con esto, el analista de vibraciones hace una comparación del espectro medido en la máquina y clasifica la falla.

Tablas de Charlotte

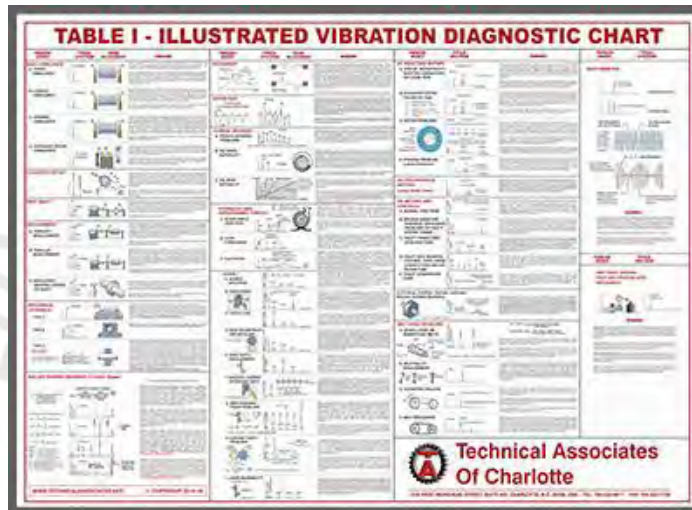


Figura 22. Tabla de Charlotte. Fuente: <https://www.technicalassociates.net/>

Tabla de Análisis de Vibraciones de Reliability Maintenance Solutions (RMS)



Figura 23. Tabla de Análisis de Vibraciones de RMS. Fuente: <https://rms-reliability.com/training/>

1.5. Robots para el diagnóstico de fallas

En los últimos años, el uso de robots para detección de este tipo de fallas en la industria se ha incrementado debido a que son una alternativa con bastante potencial en actividades que involucren el acceso a lugares remotos, inaccesibles o que pongan en peligro la vida de un operador. Los robots son programados para cumplir tareas conocidas de forma eficiente y segura, sin embargo, es necesaria una capa de control que asegure la precisión y repetitividad de la detección. En (Nagatani et al., 2011), (Abhishek et al., 2017) y (Osumi, 2017), los autores describen la aplicación de estos robots en el diagnóstico de fallas y en tareas de rescate. Se explica cómo los robots contribuyeron en tareas de rescate en los desastres nucleares de Hiroshima y en diferentes siniestros como los terremotos. Durante estos acontecimientos, los robots como el Quince ayudaron en el rescate de las personas afectadas. Los robots Quince y DRASB, diseñados para rescate de personas se muestran en las Figura 24 y 25.



Figura 24. Robot Quince. Fuente: Tadokoro, 2015

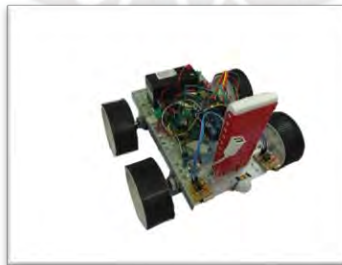


Figura 25. Robot DRASB. Fuente: Abhishek, et. al, 2017

Dentro de la categoría de robots de inspección, actualmente existen robots que viajan a través de la tubería en búsqueda de fallas dentro de estas. Estos robots cuentan con una serie de sensores como cámaras, inclinómetro, sensor inercial, etc., y son lo suficientemente pequeños para entrar y recorrer las tuberías. La Figura 26 muestra un robot (Departamento de Robótica y Mecatrónica de la Facultad de Ingeniería Mecánica de la Universidad AGH de Ciencia y Tecnología de Cracovia, 2017) para la inspección de tuberías.

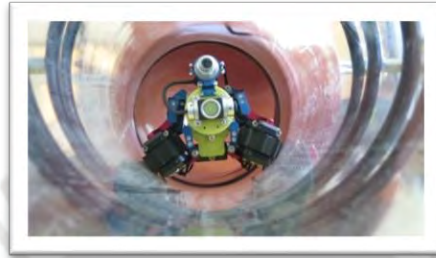


Figura 26. Robot para inspección de tuberías. Fuente: Departamento de Robótica y Mecatrónica de la Facultad de Ingeniería Mecánica de la Universidad AGH de Ciencia y Tecnología de Cracovia, 2017. Link: <https://www.youtube.com/watch?v=NKH9povlrsw>

Boston Dynamics (Boston Dynamics, 2020), Energy Robotics (Boston Robotics, 2020) y Sprint Robotics (Sprint Robotics, 2020) son algunos ejemplos de compañías dedicadas al desarrollo y aplicación de soluciones robóticas para mejorar la calidad y seguridad en diversas actividades como mantenimiento, inspección y limpieza. El Robot SPOT desarrollado por Boston Robotics y Energy Robotics tiene acondicionadas cámaras térmicas, comunicación 4G inalámbrica y un computador embebido que le permiten realizar tareas de supervisión e inspección en plantas de forma autónoma (Energy Robotics, 2020).



Figura 27. Robot SPOT desarrollado por Energy Robotics y Boston Robotics. Fuente: Boston Robotics, <https://www.bostondynamics.com/>, <https://www.energy-robotics.com/>

En nuestro país también se tienen empresas como 4Helix Labs y N&A SAC, dedicadas a la implementación de robótica e inteligencia artificial para el sector industrial con el objetivo de hacer más eficientes y seguras los trabajos de inspección y mantenimiento en diversos equipos. 4Helix Labs desarrolla robots móviles tele-operados, para monitorear plantas industriales y prevenir desastres en infraestructura (PIMEN, 2020). N&A SAC es una empresa dedicada al rubro de la soldadura industrial, por lo que desarrollaron un robot de inspección de la calidad en el acabado de la soldadura en cilindros de Gas Licuado de Petróleo, utilizando sistemas laser, ecosonda y visión artificial (PIMEN, 2020).

1.6. Metodologías para el Diagnóstico de Fallas

Las fallas pueden diagnosticarse con dos metodologías: una basada en modelos y otra en datos (Mrugalski, 2014). La metodología basada en modelos requiere del conocimiento exacto del proceso. En cambio, con la metodología basada en datos, se puede estimar el comportamiento del sistema con información de entradas y salidas.

1.6.1. Basado en Modelos

La metodología de diagnóstico de fallas basado en modelos, se basan en leyes físicas, químicas y matemáticas para representar el comportamiento teórico e ideal de un sistema (Isermann, 2011). A partir de la medición de las señales de entrada y salida del sistema, se pueden hallar parámetros o estados de comparación para detectar cambios que lleven a fallas.

La gran desventaja de esta metodología es que se requiere de un modelo muy preciso para alcanzar los resultados esperados. En la industria, existen muchos casos particulares y diversidad de sistemas, en su mayoría no lineales y dinámicos, lo cual hace más complicada su aplicación. Además de perturbaciones, incertidumbres, ruidos que alteran el modelo teórico.

1.6.2. Basado en Datos

El diagnóstico de fallas basado en datos, trabaja a partir de la experiencia del operador y una base de datos recopilada tras varias pruebas y adquisiciones previas. La ventaja de esta metodología sobre la basada en modelos, es que, en base a datos registrados del sistema, es posible detectar las fallas incluso en casos muy particulares con distintos tipos de perturbación.

1.7. Procesamiento de datos

Las señales adquiridas por los diversos sensores deben pasar por una etapa de procesamiento para poder ser diagnosticadas. A continuación, se describen el análisis vibracional y diagnóstico por propagación de onda.

1.7.1. Análisis Vibracional

En (Stiller et al., 2017), los autores hacen un estudio de los diferentes métodos para analizar vibraciones. A continuación, se hace una breve descripción de los métodos y se exponen sus ventajas y desventajas.

Transformada Rápida de Fourier

El análisis de una señal mediante la transformada de Fourier, permite evaluar su comportamiento en el dominio de la frecuencia. La teoría de Fourier se basa en el hecho que se podemos representar una señal por la suma de componentes sinusoidales. La ventaja de esta metodología es que es sencilla de implementar (Adnan et al., 2015). Las desventajas son que sólo trabaja bien cuando el sistema es lineal y cuando los datos son periódicos o estacionarios. En la mayoría de casos, se trabajará con sistemas no lineales y dinámicos.

Wavelets

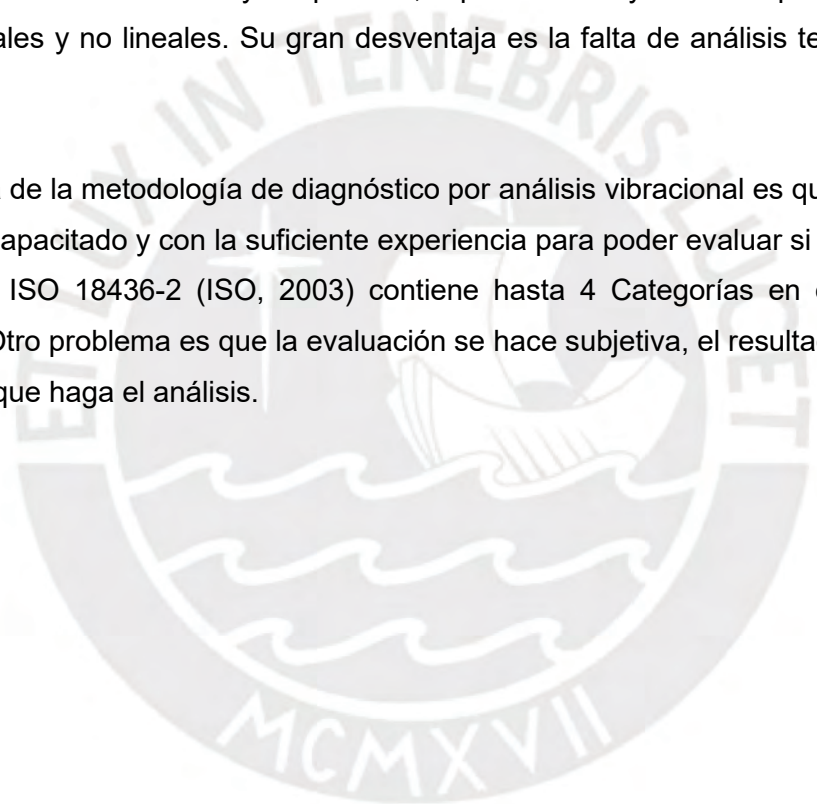
El análisis de la señal mediante Wavelets o también conocidas como ondículas, es muy similar a cómo funciona la transformada de Fourier. Se establecen wavelets elementales con las cuales se descompone la señal. Las ventajas son el análisis de señales dinámicas,

multi resolución, la obtención de la función base por convolución y funciones de escalamiento. Las desventajas que tiene el método son que es de naturaleza no adaptativa y no puede resolver la modulación entre ondas.

Transformada de Hilbert Huang (HHT)

La transformada de Hilbert Huang es una representación Tiempo-Frecuencial de una señal. Consta de dos partes: Descomposición Empírica de Modos y la transformada de Hilbert. Durante el capítulo 2 se desarrollará más a detalle esta metodología. Entre sus ventajas está: su uso para cualquier tipo de señal, su alta resolución tiempo-frecuencial, análisis frecuencial con métodos locales y adaptativos, espectro nítido y su buena performance con sistemas lineales y no lineales. Su gran desventaja es la falta de análisis teórico, físico y matemático.

La desventaja de la metodología de diagnóstico por análisis vibracional es que requiere de un operador capacitado y con la suficiente experiencia para poder evaluar si existen fallas. La normativa ISO 18436-2 (ISO, 2003) contiene hasta 4 Categorías en el análisis de vibraciones. Otro problema es que la evaluación se hace subjetiva, el resultado dependerá del operador que haga el análisis.



1.8. Inteligencia Artificial

En los últimos años, la inteligencia artificial ha obtenido mayor interés en la industria, debido al aporte que representa su uso, principalmente en la reducción de costos, la automatización de procesos de producción y como ayuda en las labores que exigen mucha demanda de tiempo o tengan un alto grado de peligrosidad para el ser humano.

La Inteligencia artificial es muy estudiada en la industria, así como, en distintos campos como el de los juegos, traducción y comprensión de idiomas, robótica, finanzas, medicina y administración de base de datos (Forbes, 2018).

1.8.1. Definiciones de Inteligencia Artificial

Según Russell y Norvig en (Russel, 2004) a lo largo de los años la definición de la inteligencia artificial ha seguido cuatro enfoques:

- *Sistemas que piensan como humanos*, según (Haugeland, 1985). La inteligencia artificial es hacer que las computadoras piensen como humanos como la toma de decisión, resolución de problemas y aprendizaje.
- *Sistemas que actúan como humanos*, según (Rich y Knight, 1991). La inteligencia artificial es lograr que las computadoras realicen las tareas que hacen los humanos, como, por ejemplo, en aplicaciones de robótica.
- *Sistemas que piensan racionalmente*, según (Winston, 1992). La inteligencia artificial consiste en tratar de imitar el pensamiento lógico racional del ser humano. Por ejemplo, cálculos que hacen posible percibir, razonar y actuar.
- *Sistemas que actúan racionalmente*, (Nilsson, 1998). La inteligencia artificial consiste en tratar de simular de forma racional el comportamiento humano, es decir, está relacionada con conductas inteligentes en artefactos.

La Inteligencia Artificial, en resumen, busca imitar al ser humano tanto en su pensamiento como comportamiento. Lo que se desea es que las maquinas imiten el proceso de aprendizaje, razonamiento y la autocorrección que tienen los seres humanos.

1.8.2. Inteligencia artificial aplicado en el diagnóstico de fallas

La Inteligencia Artificial permite diseñar sistemas automáticos de diagnóstico de fallas a través del análisis de patrones y tendencias, que se generan con la información de datos históricos adquiridos por sistemas de monitoreo, para detectar y reportar fallas con el fin de tomar acciones preventivas de mantenimiento.

Las principales estrategias para la identificación y diagnóstico de fallas son las basadas en redes neuronales, lógica difusa y sistemas inmunes (Ramírez, 2018). Así mismo, su elección depende de algunas características importantes como su robustez, adaptabilidad, costo computacional e identificación de fallas múltiples. En los siguientes capítulos se desarrollará más a detalle la metodología escogida.



1.9. Propuesta de Tesis y Objetivos

1.9.1. Propuesta de Tesis

En base al estudio realizado en las secciones anteriores y tomando en cuenta la problemática de la influencia de las fallas en los componentes industriales, la necesidad de una inteligencia artificial para un diagnóstico objetivo y la introducción de robots para llegar a lugares inaccesibles y peligrosos, se propone el siguiente trabajo de tesis, el cual consiste en diseñar e implementar un sistema de diagnóstico de fallas que utilizará un robot móvil para recorrer una planta industrial para tomar muestras de vibraciones para detectar anomalías en máquinas rotativas. Cabe indicar que el diseño del robot móvil y su brazo robótico no forma parte del alcance de esta tesis. La metodología a utilizar será el procesamiento de la señal de sensores de vibración por análisis vibracional y el diagnóstico mediante Inteligencia Artificial. A continuación, se describen los objetivos a lograr con el presente trabajo.

1.9.2. Objetivo General y Objetivos Específicos

Objetivo General

Diseñar e Implementar un sistema de diagnóstico de fallas para un proceso industrial utilizando un robot móvil e Inteligencia Artificial, mediante la medición y análisis de vibración.

Objetivos Específicos

- Estudiar el estado del arte en metodologías de diagnóstico de fallas, en particular las que utilizan métodos vibracionales, así como la utilización de robots móviles para la toma de datos.
- Diseñar un sistema de adquisición de datos vibracionales posicionado conectado al terminal de un brazo robótico.
- Desarrollar el sistema de diagnóstico de fallas para motores y bombas utilizando análisis vibracional e Inteligencia Artificial.
- Simular el desempeño del sistema propuesto en Python.
- Implementar el sistema de diagnóstico de fallas conectado al sistema de adquisición de datos del brazo robótico.

Capítulo 2. Procesamiento para el diagnóstico de fallas

2.1. Introducción

En este capítulo se describen los métodos de análisis vibracional y de propagación de onda, incluye la adquisición y acondicionamiento de las señales, y la programación de los algoritmos basados en la transformada de Hilbert Huang.

2.2. Adquisición de datos

El presente trabajo de Tesis utilizará un robot móvil como plataforma para la adquisición de las señales de vibración. Para la implementación y pruebas se propone utilizar un brazo robot con sensores para la medición de las señales de interés. Se utilizará un acelerómetro triaxial para medir la aceleración de vibración de las bombas y motores.

2.2.1. Medición con acelerómetro

El brazo robot deberá colocar el acelerómetro en los puntos adecuados según la normativa ISO 10816-1 (ISO, 1993) como se muestra en las Figuras 28-32. Para un correcto diagnóstico, se deben registrar mediciones en los 3 ejes: Axial, Radial y Tangencial (Sarmiento, 2020).

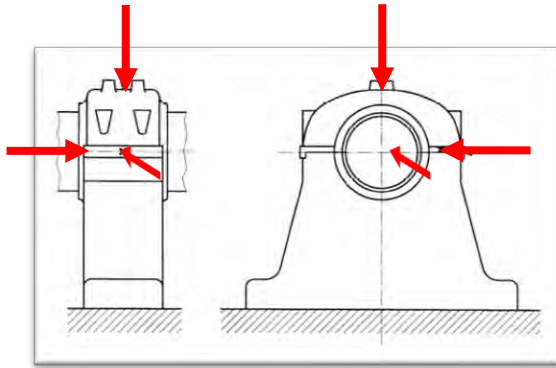


Figura 28. Puntos de medición en cojinetes tipo pedestal. Fuente: ISO 10816-1

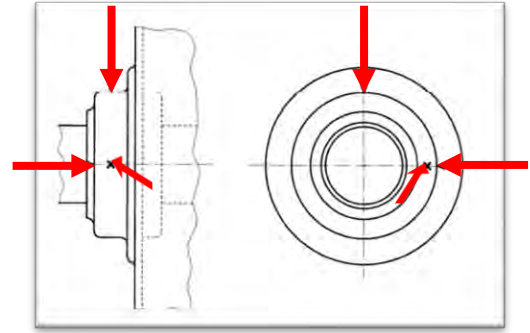


Figura 29. Puntos de medición en cojinetes tipo chasis. Fuente: ISO 10816-1

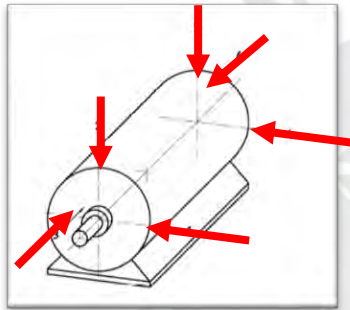


Figura 30. Puntos de medición para máquinas eléctricas pequeñas. Fuente: ISO 10816-1

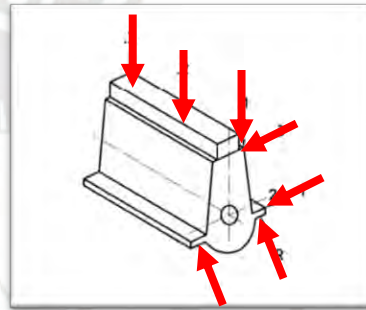


Figura 31. Puntos de medición para motores recíprocos. Fuente: ISO 10816-1

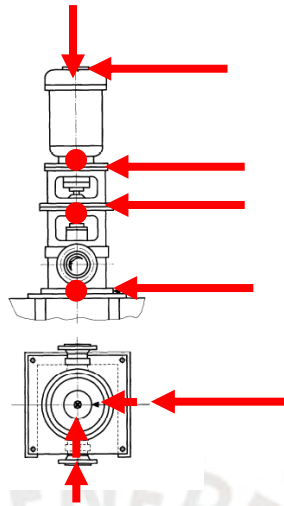


Figura 32. Puntos de medición para máquinas verticales. Fuente: ISO 10816-1

2.3. Acondicionamiento de la señal

Antes de ser procesadas, las señales deberán pasar por una etapa de pre amplificación y filtrado, para reducir el ruido y mejorar la lectura de la señal física. En la etapa de pre amplificación se evaluarán los parámetros dependiendo de los requisitos de la tarjeta de adquisición. Para el filtrado de la señal se consideran dos filtros: Filtro Pasa Banda y Filtro DC. El primer filtro nos permitirá trabajar con las frecuencias relevantes y acortar el ancho de banda, mientras que el segundo será una etapa previa a la transformada de Hilbert Huang, en donde se removerá la señal continua.

2.4. Procesamiento de datos

En el Capítulo 1 se mencionaron diversas metodologías para procesar los datos de vibración como la Transformada de Fourier, Wavelets y la Transformada de Hilbert Huang. Elegimos trabajar con la transformada de Hilbert-Huang, debido a que es la única que puede trabajar señales dinámicas y no lineales, presenta los datos en tiempo y frecuencia, y por su implementación sencilla.

A continuación, se describe esta metodología con mayor detalle.

2.4.1. La Transformada de Hilbert Huang

La Transformada de Hilbert Huang es propuesta por Huang en el año 1998 (Wang y Ji, 2018). Consta de dos partes: Descomposición Empírica de Modos y la Transformada de Hilbert.

2.4.1.1. Descomposición Empírica de Modos (EMD)

Esta etapa descompone una señal en sus Funciones de Modo Intrínseco (IMF). Estas funciones son señales completas, adaptivas y representaciones casi ortogonales de la señal analizada las cuales, deben cumplir las siguientes condiciones (Wang y Ji, 2018):

- El número de extremos (máximos y mínimos) y el número de cruces por cero deben ser iguales o al menos diferenciarse en 1.
- En cualquier punto, el valor medio de la envolvente definida por máximo local y la envolvente definida por el mínimo local es cero.

Para ello se utilizan las señales envolventes superior e inferior de la señal y se calcula la media entre ellas. Luego se halla la resta entre la señal y la media. Si este resultado cumple con las condiciones necesarias para ser considerada IMF, se reemplaza la señal inicial con la diferencia entre IMF y la media. El proceso se itera hasta cumplir una condición de parada. La Figura 33 muestra el algoritmo.

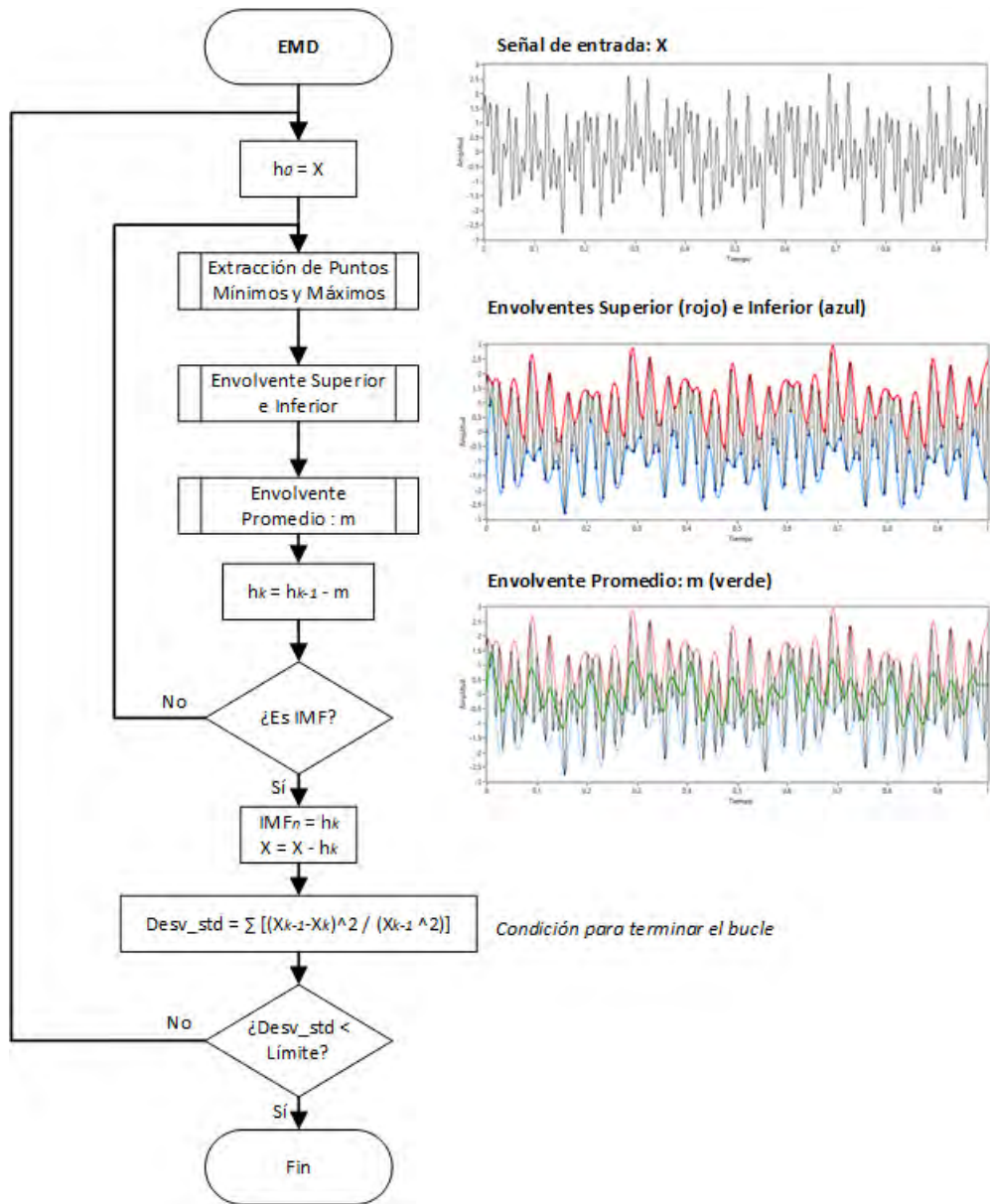


Figura 33. Diagrama de flujo del algoritmo HHT. Elaboración propia

2.4.1.2. Transformada de Hilbert

La transformada de Hilbert es una representación de una señal en el dominio temporal y frecuencial. De esta señal, se obtiene la componente imaginaria según la siguiente ecuación.

$$y(t) = \frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{x(\tau)}{t-\tau} d\tau \dots \text{(Ecuación 2.1)}$$

Donde:

P es el primer valor de Cauchy

Las componentes real e imaginaria pueden juntarse en la siguiente función analítica:

$$z(t) = x(t) + iy(t) = a(t)e^{i\theta(t)} \dots \text{(Ecuación 2.2)}$$

Con esta representación se puede hallar la magnitud y la fase.

$$a(t) = \sqrt{x^2 + y^2} \dots \text{(Ecuación 2.3)}$$

$$\theta(t) = \tan^{-1} \frac{y}{x} \dots \text{(Ecuación 2.4)}$$

También es posible hallar la energía de la señal.

La Transformada de Hilbert Huang recibe como entrada una señal real y obtiene como salidas las IMFs, la frecuencia instantánea en función del tiempo y la energía representada en tiempo y frecuencia.

Sin embargo, la Transformada de Hilbert Huang presenta dos problemas en su primera etapa, los cuales influyen en la precisión del resultado final:

- Efectos de final
- Mezcla de modos

A. Efectos de final

Durante la etapa EMD se utilizan las señales envolventes superior e inferior de la señal a evaluar. Para ello se aplica una interpolación spline cúbica a partir de los máximos y mínimos locales encontrados. El problema es que tanto al inicio como al final de la

envolvente no se tiene información que permita interpolar estas zonas de manera ideal. La interpolación presenta caídas o alzas bruscas en estas zonas que no representan a la envolvente. Debido a estas caídas o alzas ingresan señales IMF indeseadas en baja frecuencia, por lo que se obtienen datos erróneos que perjudican el diagnóstico. Para minimizar los efectos de final, (Wang, Ji, 2018) propone utilizar una interpolación lineal local. La metodología consta de utilizar los dos últimos máximos (o mínimos) para encontrar un punto extremo que permita aproximar de manera más coherente al extremo de la señal.

La tabla 4 muestra los errores en las envolventes originales y las envolventes corregidas con el método de la interpolación lineal local.

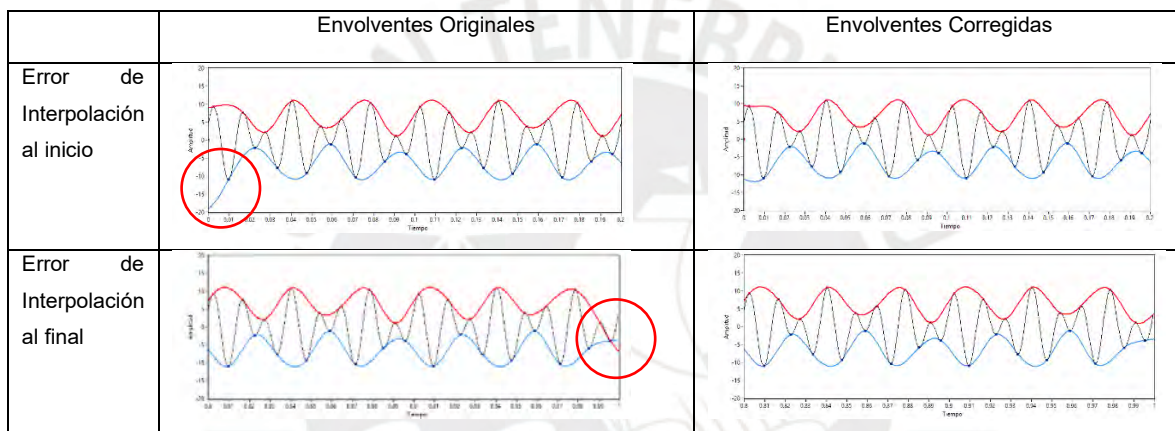


Tabla 4. Corrección de las envolventes utilizando la Interpolación Lineal Local

B. Mezcla de modos

El problema de mezcla de modos se presenta cuando una señal contiene componentes frecuenciales muy cercanos. Este problema puede causar aliasing en la distribución temporal y frecuencial ocasionando que una IMF represente más de una frecuencia. Wang y Ji (Wang y Ji, 2018) mencionan que la principal causa de este problema es que la descomposición de componentes IMF en la etapa EMD no son totalmente ortogonales. Para lograr la ortogonalidad entre IMFs es necesario implementar una función de decorrelación.

Utilizamos la covarianza para analizar la ortogonalidad entre dos variables.

La covarianza es una medida estadística que representa la dispersión entre los valores de una señal.

$$C_x = \frac{\sum(x_i - \bar{x})^2 \cdot n_i}{N} \quad \text{o} \quad C_x = \frac{\sum x_i^2 \cdot n_i}{N} - \bar{x}^2 \dots \text{(Ecuación 2.5)}$$

$$C_y = \frac{\sum(y_i - \bar{y})^2 \cdot n_i}{N} \quad \text{o} \quad C_y = \frac{\sum y_i^2 \cdot n_i}{N} - \bar{y}^2 \dots \text{(Ecuación 2.6)}$$

Entonces la correlación entre las dos variables será:

$$C_{xy} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y}) \cdot n_i}{N} \quad \text{o} \quad C_{xy} = \frac{\sum x_i \cdot y_i \cdot n_i}{N} - \bar{x} \cdot \bar{y} \dots \text{(Ecuación 2.7)}$$

Finalmente se puede expresar como:

$$C_{xy} = E\{(x - E(x))(y - E(y))\} = E(xy) - E(x)E(y) \dots \text{(Ecuación 2.8) } \llcorner$$

Donde x e y son señales de longitud n : $x = x(n)$, $y = y(n)$

A partir de la ecuación de correlación definida se obtienen las conclusiones:

1. Si $C_{xy} = 0$, $x(n)$ e $y(n)$ no están correlacionadas.
2. Si $E(xy) = 0$, $x(n)$ e $y(n)$ son ortogonales.
3. Finalmente, si $x(n)$ e $y(n)$ no están correlacionadas ($C_{xy} = 0$) y $E(x) = 0$ ó $E(y) = 0$, entonces $x(n)$ e $y(n)$ son ortogonales.

Se concluye que la ortogonalidad entre x e y se consigue cuando las señales no están correlacionadas para variables aleatorias con media cero.

Para solucionar estos problemas, Wang y Ji (Wang y Ji, 2018) proponen utilizar una función de decorrelación en la etapa EMD. Se define el coeficiente 'r' como:

$$r = \frac{y^T(n) \cdot x(n)}{y^T(n) \cdot y(n)} \dots \text{(Ecuación 2.9)}$$

Donde $r \cdot y(n)$ representa la correlación cruzada de $x(n)$ e $y(n)$. El dato $v(n)$, el cual no está correlacionado con $y(n)$, se obtiene con la siguiente ecuación:

$$v(n) = x(n) - r \cdot y(n) \dots \text{(Ecuación 2.10)}$$

Según la definición del coeficiente de correlación, r debe satisfacer la condición $|r| \leq 1$. Este caso no siempre se cumple debido a las amplitudes de las IMF. Pero al normalizar las señales y asumiendo un r_N como la correlación escalada, esto no afecta el resultado. La Figura 34 presenta el diagrama de flujo de la metodología.

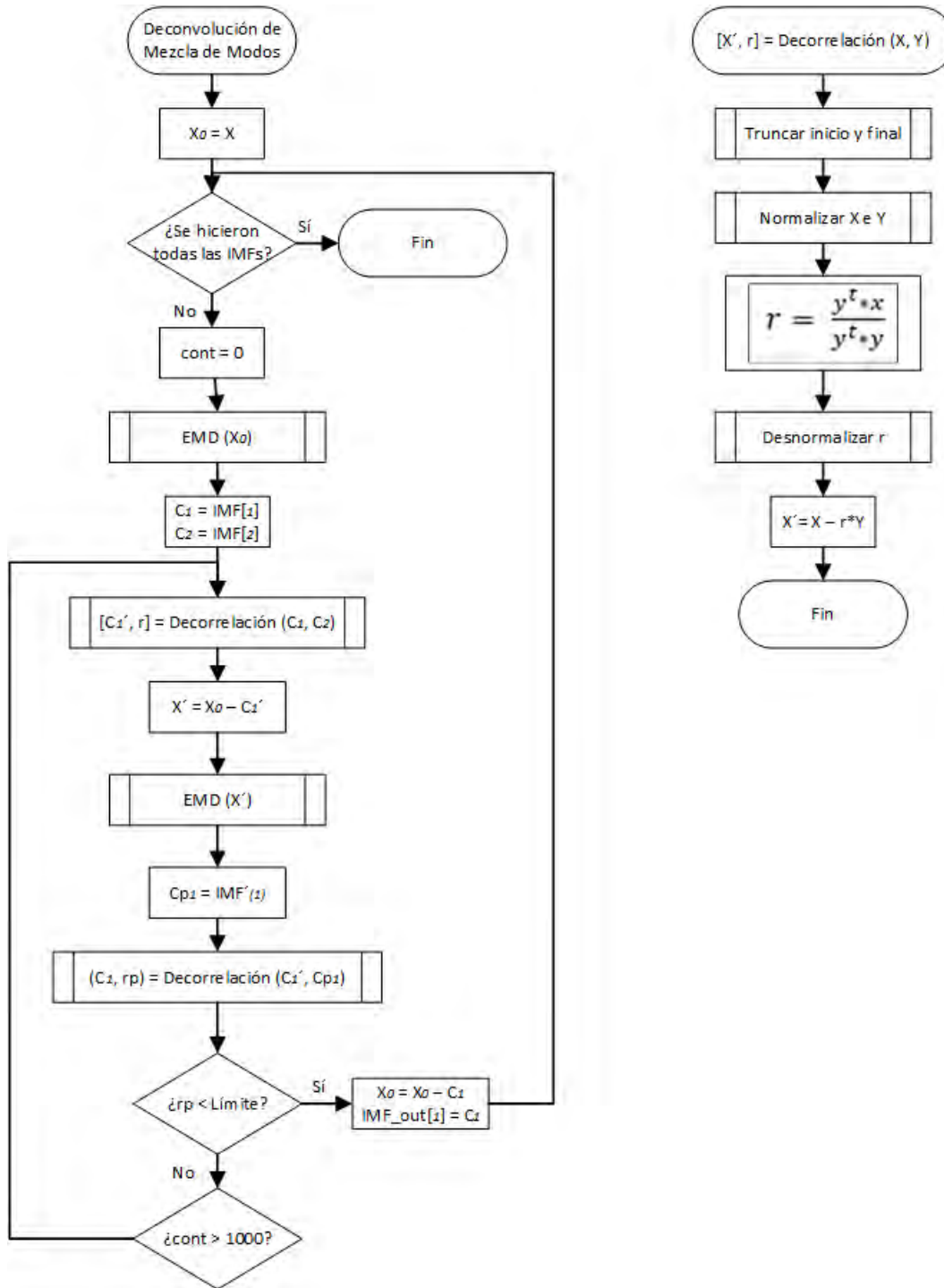


Figura 34. Diagrama de flujo del algoritmo Descomposición de Mezcla de Modos. Elaboración propia

2.4.2. Evaluación de la Transformada de Hilbert Huang

2.4.2.1. Evaluación con Señales sintética

Definimos la siguiente señal de entrada compuesta por 5 tonos:

$$X = \cos(2\pi 80 t) + 0.8 \sin(2\pi 50 t) + 0.6 \sin(2\pi 25 t) + 0.4 \cos(2\pi 10 t) + 0.3 \cos(2\pi 3 t)$$

La gráfica de la señal de entrada se muestra en la figura 35.

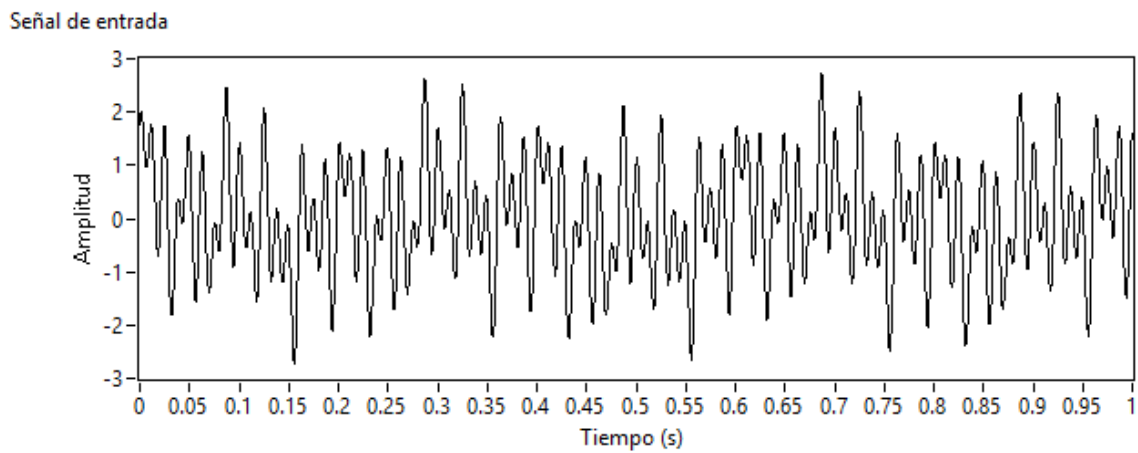


Figura 35. Señal de entrada con 5 tonos. Elaboración propia

El espectro de la señal sintética utilizando la Transformada de Fourier se muestra en la figura 36.

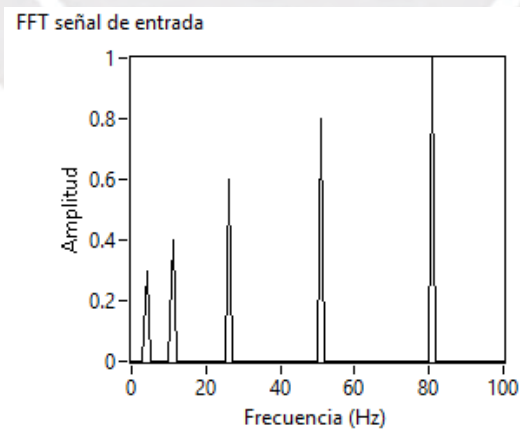


Figura 36. Espectro de la señal de entrada utilizando la Transformada de Fourier. Elaboración propia

Finalizada la etapa EMD, se consiguen las siguientes IMF.

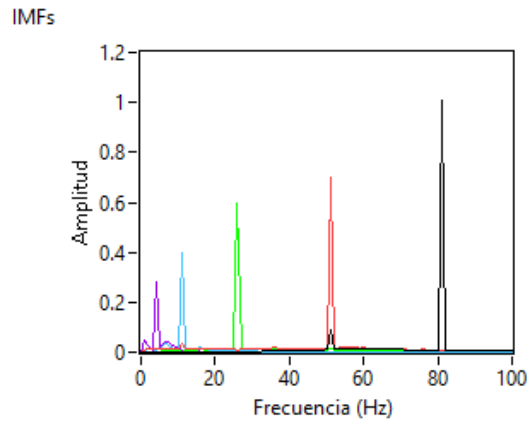


Figura 37. IMF obtenidas en la etapa EMD. Elaboración propia

Se pueden apreciar los 5 picos, correspondientes a las frecuencias que conforman la señal sintética.

Las IMF encontradas se muestran en la tabla 5.

Señales IMF	
<p>Tono de 80 Hz</p>	<p>Tono de 50 Hz</p>
<p>Tono de 25 Hz</p>	<p>Tono de 10 Hz</p>
<p>Tono de 3 Hz</p>	

Tabla 5. Tonos encontrados en la señal

Los tonos encontrados corresponden a las componentes de la señal original.

2.4.2.2. Señales reales

Utilizamos el banco de datos de la Universidad de Case Western Reserve (Case Western Reserve, 2019), la cual contiene señales reales de mediciones de vibración de motores en estado óptimo y con diferentes fallas en rodamientos.

Motor sin fallas

Condiciones:

- Muestra: X098_FE_1772
- Medición en parte del ventilador
- RPM: 1772
- Frecuencia de muestreo: 12,000 Hz
- Número de muestras: 1,200
- Tiempo de medición: 0.1 segundos

La señal original se muestra en la Figura 36.

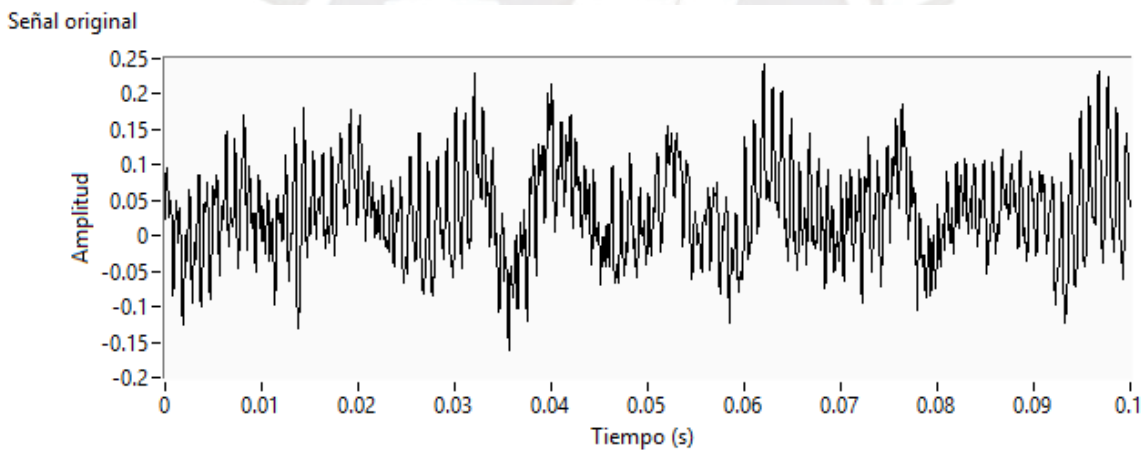


Figura 38. Señal original X098_FE_1772. Elaboración propia

El espectro de la señal sin fallas utilizando la Transformada de Fourier se muestra en la Figura 39.

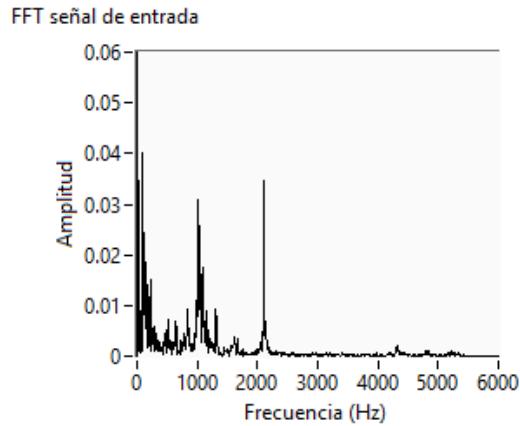


Figura 39. Espectro de la señal de entrada utilizando la Transformada de Fourier. Elaboración propia

El ancho de bando de los espectros utilizado para las gráficas será de 6 KHz, esto nos permitirá comparar los resultados de las señales con y sin fallas utilizando la misma referencia. Se puede apreciar que el espectro presenta componentes frecuenciales de 1Khz, 2.1 KHz y en bajas frecuencias.

Las siguientes consideraciones fueron utilizadas en el programa implementado en LabVIEW:

- Cantidad de IMF a encontrar: 5
- Límite: 0.0002

Luego de la etapa EMD se obtienen IMF con el espectro mostrado en la Figura 40.

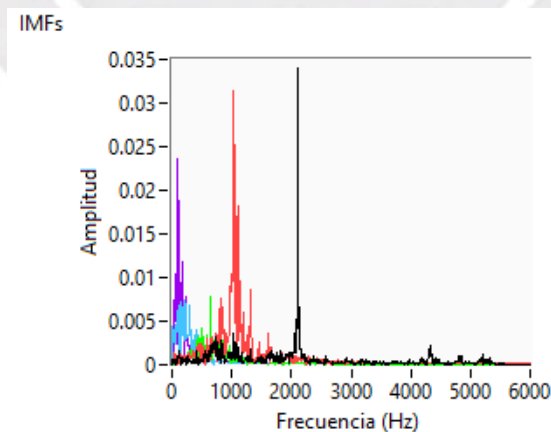


Figura 40. Espectro de las IMF encontradas. Elaboración propia

Al sumar las IMF encontradas y comparando con la señal original se encontró una correlación de 93%.

Las IMF encontradas se muestran en la tabla 6.

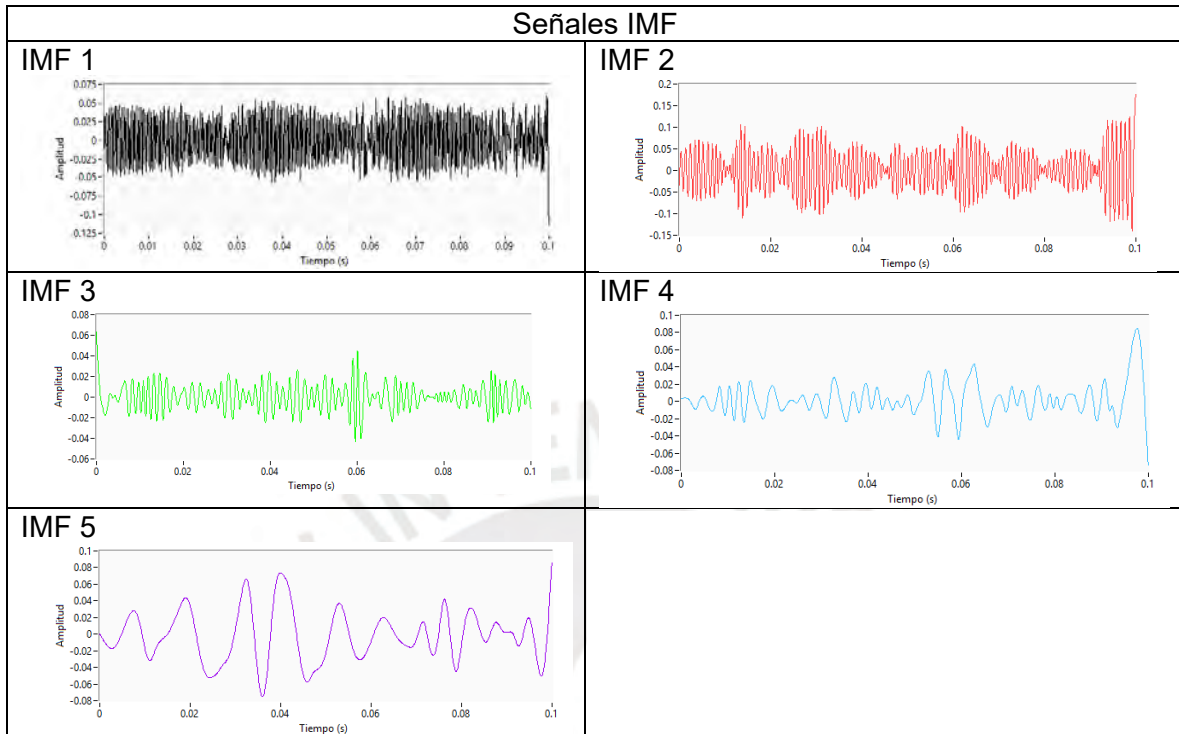


Tabla 6. IMF encontradas

Motor con fallas en rodamientos (Diámetro de falla 0.007")

Condiciones:

- Muestra: X283_FE_time_1777_ball_0.007
- Medición en parte del ventilador
- RPM: 1777'
- Frecuencia de muestreo: 12,000 Hz
- Número de muestras: 1,200
- Tiempo de medición: 0.1 segundos

La señal original se muestra en la Figura 41.

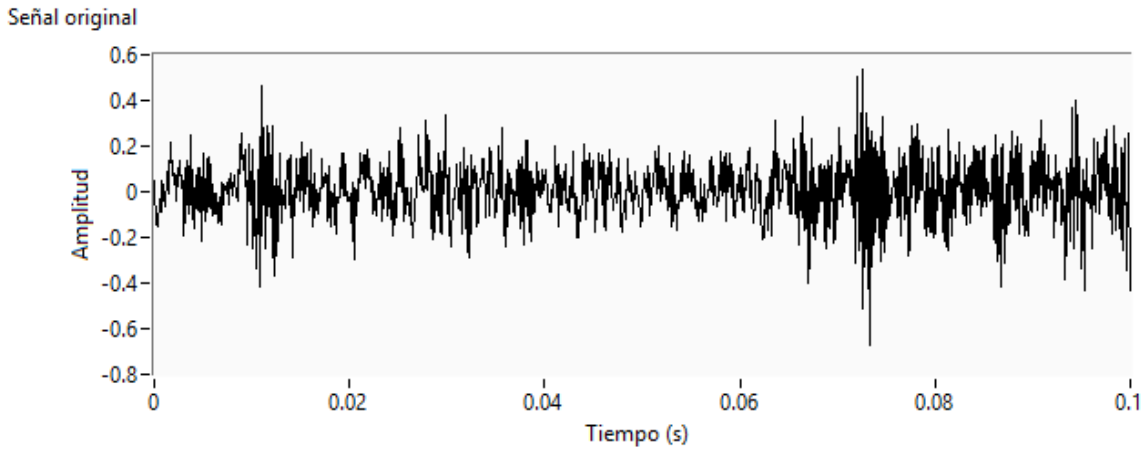


Figura 41. Señal original de un motor con falla en rodamientos. Elaboración propia

El espectro de la señal de un motor con fallas en rodamientos utilizando la Transformada de Fourier se muestra en la Figura 42.

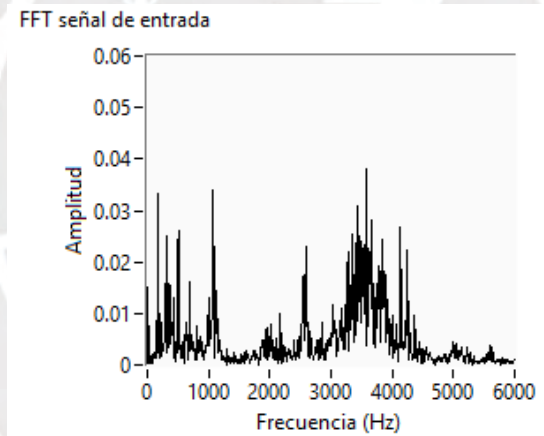


Figura 42. Espectro de la señal de un motor con falla en rodamientos. Elaboración propia

En el caso de un motor con fallas en rodamientos, se puede apreciar que el espectro presenta componentes frecuenciales de 1Khz, en bajas frecuencias y en frecuencias de 3 kHz a 4 kHz. Esta presencia inusual puede ser indicio de fallas según se comentó en la sección 1.4.4.

Las siguientes consideraciones fueron utilizadas en el programa implementado en LabVIEW:

- Cantidad de IMF a encontrar: 5

- Límite: 0.0002

Luego de la etapa EMD se obtienen IMF con el espectro en frecuencia mostrado en la Figura 43.

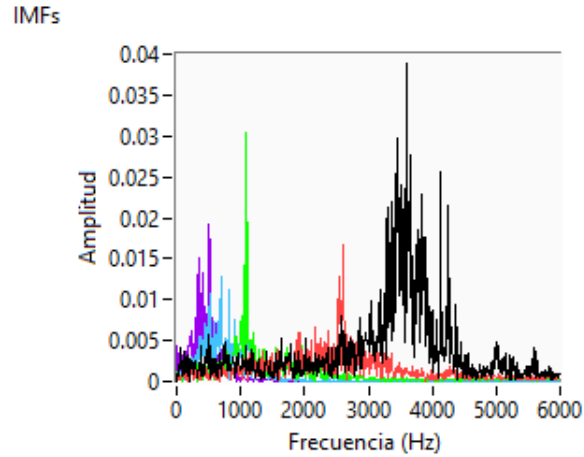


Figura 43. Respuesta en frecuencia de las IMF encontradas. Elaboración propia

Al sumar las IMF encontradas y comparando con la señal original se encontró una correlación de 92%.

Las IMF encontradas se muestran en la tabla 7.

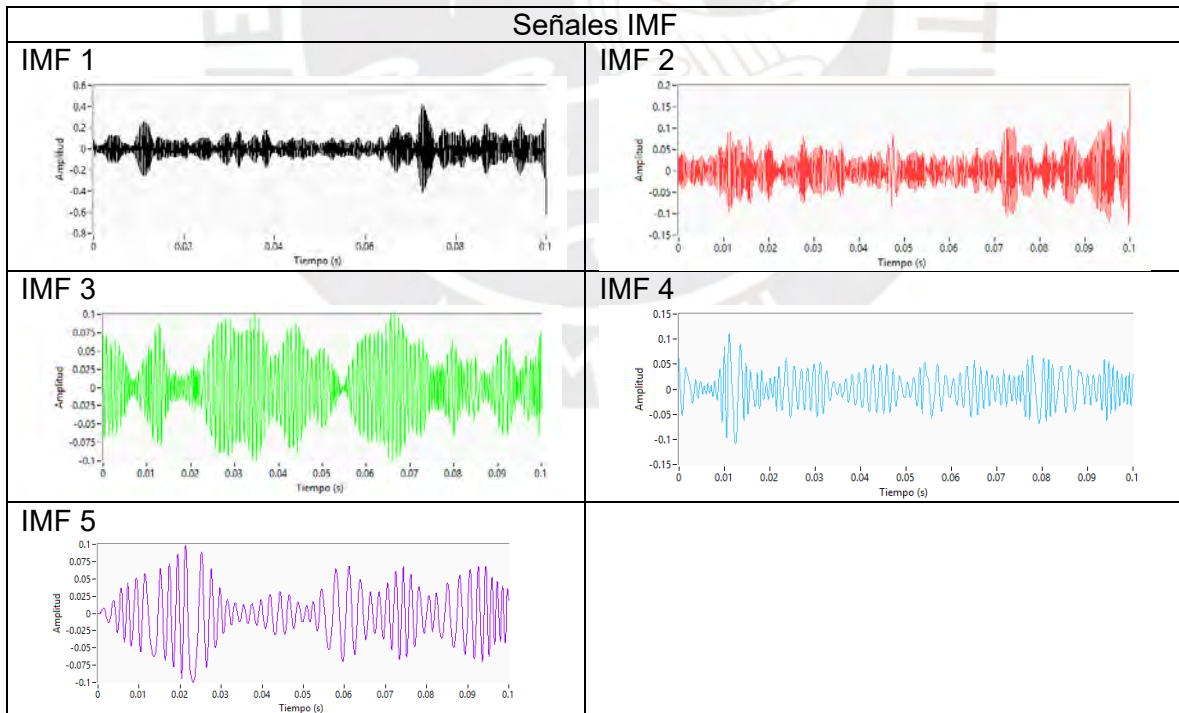


Tabla 7. IMF encontradas

2.4.3. Entradas para la Red Neuronal

Para la evaluación del sistema utilizaremos la Base de Datos MAFAULDA (MAFAULDA, 2018). Las entradas que alimentarán la Red Neuronal serán las primeras 3 IMF. Como ejemplos, en la tabla 8 se muestran las primeras 3 IMF de las señales medidas en los 3 ejes de un motor en condiciones normales y en la tabla 9, las primeras 3 IMF de las señales medidas en los 3 ejes de un motor con falla en rodamientos.

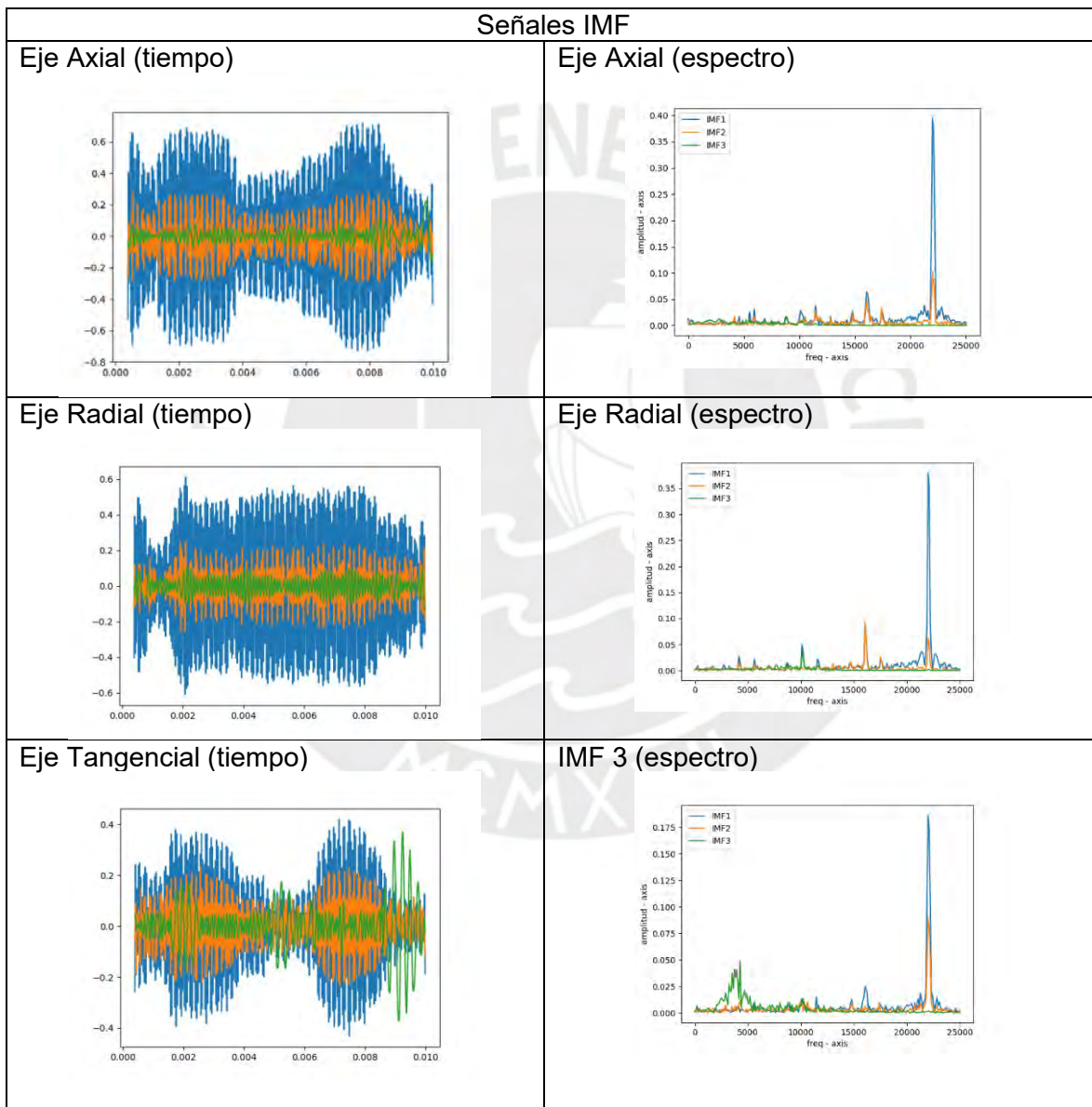


Tabla 8. Señales IMF de motor en estado normal

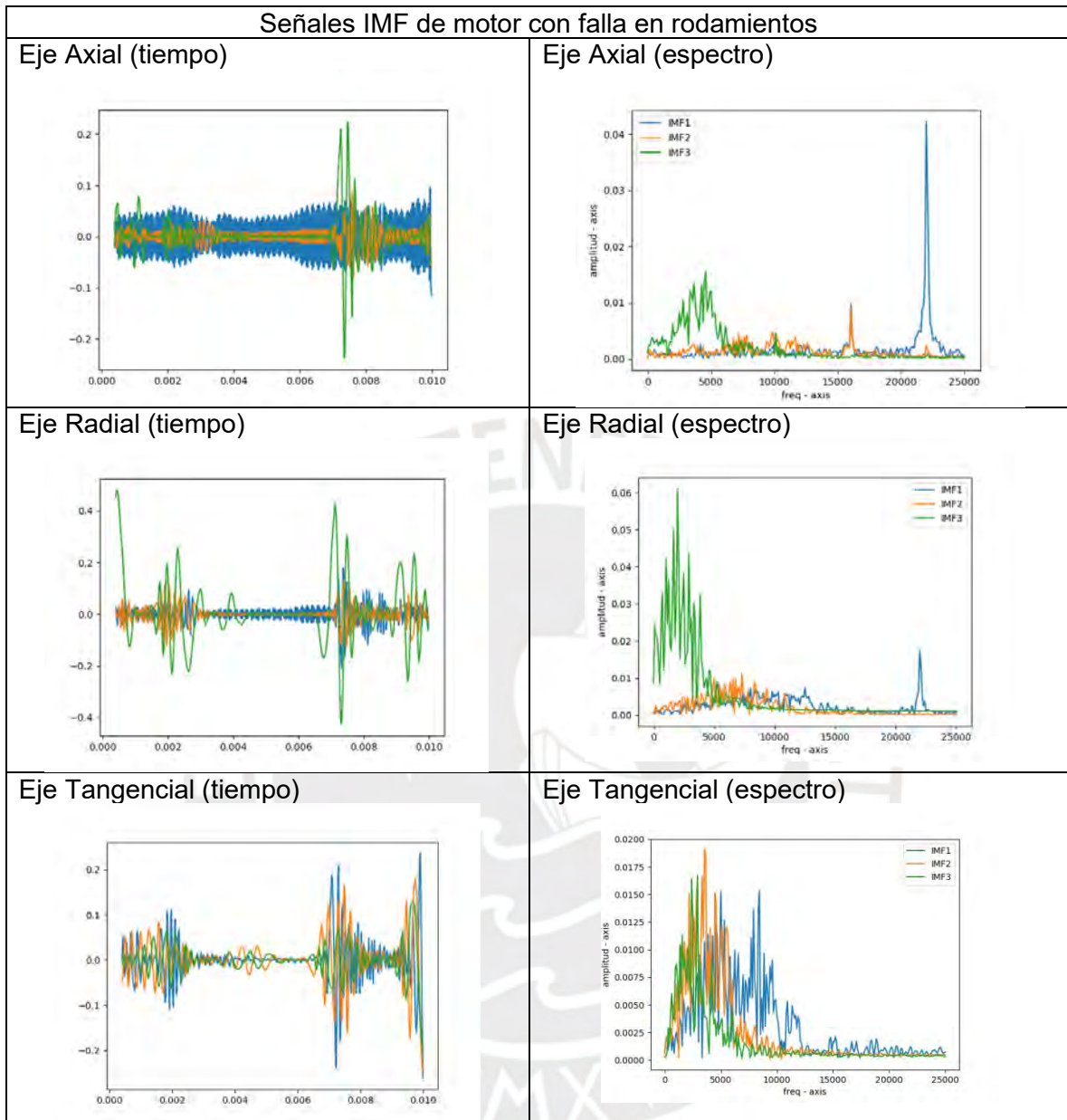


Tabla 9. Señales IMF de motor con fallas en rodamientos

En resumen, mediante la Transformada de Hilbert Huang, corregida con el algoritmo de interpolación lineal local, se encontrará la información relevante que representa a una señal adecuada y a una con fallas. El siguiente capítulo presentará el diseño de la Red Neuronal para el diagnóstico de las fallas utilizando como entradas las IMFs.

2.5. Conclusiones del Capítulo

- Las mediciones de las señales de vibración se realizarán mediante un brazo robot, el cual posicionará los sensores según se indica en la normativa ISO 10816-1 (ISO, 1993).
- Las señales medidas deberán pasar por una etapa de amplificación y filtrado para su posterior procesamiento con la Transformada de Hilbert Huang. Los parámetros para el acondicionamiento se determinarán según las tarjetas de adquisición utilizadas.
- La Transformada de Hilbert Huang permite representar a las señales de forma temporal y frecuencial, gracias a ello es posible detectar la falla y conocer cuando ocurrió. Para evitar los inconvenientes que tiene la etapa EMD, se utilizará el algoritmo de Interpolación Lineal Local.



Capítulo 3. Diseño de la Red Neuronal Convolutiva

3.1. Introducción

En este capítulo se presenta la Red Neuronal, su equivalencia con las neuronas en el ser humano, sus características, componentes, funcionamiento y aplicaciones. Además, se hace una breve introducción a Deep Learning o Aprendizaje Profundo y a la necesidad de optar por una Red Neuronal Convolutiva, exponiendo sus ventajas y desventajas. Dentro del diseño de la Red Neuronal Convolutiva, se presentarán los algoritmos a implementar mediante las librerías de TensorFlow y Keras, y su programación en el entorno Python.

3.2. Redes Neuronales

Las Redes Neuronales son estructuras conexionistas que imitan las neuronas del ser humano, las cuales interactúan entre sí para procesar información y aprender (Ramírez et al, 2018). Tienen la capacidad de resolver tareas de reconocimiento, predicción, clasificación, control de procesos automáticos y modelamiento de sistemas lineales y no lineales. Se utilizan en el marco de la Inteligencia Artificial para entrenar y representar sistemas de reconocimiento. Una red neuronal consiste en un sistema de entradas y salidas (x_n e \tilde{y}_n respectivamente), la cual está formada por diferentes capas: entradas, neuronas intermedias y salidas. Las capas intermedias y la cantidad de neuronas son parámetros configurables. La Figura 44 muestra un esquema básico de una Red Neuronal con 3 entradas, 4 neuronas intermedias y 2 salidas. Los pesos de los coeficientes V y W son inicializados aleatoriamente. Estos pesos serán actualizados mediante la optimización de una función de costo.

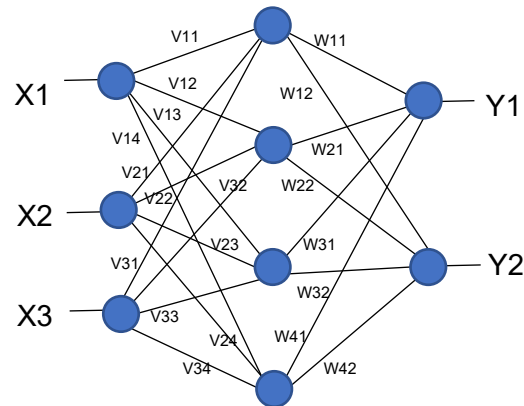


Figura 44. Ejemplo de Red Neuronal. Elaboración propia.

La Red Neuronal tiene 2 etapas: Entrenamiento y Validación. Para que una Red Neuronal logre relacionar entradas y salidas, se necesita una gran cantidad de datos (entradas x_n y sus salidas correspondientes \tilde{y}_n). Durante la etapa de Entrenamiento, se utiliza una función de costo J para minimizar el error entre los datos de salida proporcionados al inicio \tilde{y}_n y las salidas estimadas de la red y_n . Este error se va sumando para todas las entradas y sirve para ajustar los parámetros en las capas intermedias. La función de costo J , por ejemplo, puede ser el Error Cuadrático Medio, el cual eleva al cuadrado la diferencia entre salidas reales y salidas estimadas. Este error ajusta iterativamente los pesos V y W en las capas intermedias.

$$J = \sum_{i=1}^n (y_i - \tilde{y}_i)^2 \dots \text{(Ecuación 3.1)}$$

$$V_{ij} = V_{ij} + \eta * \frac{dJ}{dV} \dots \text{(Ecuación 3.2)}$$

$$W_{jk} = W_{jk} + \eta * \frac{dJ}{dW} \dots \text{(Ecuación 3.3)}$$

Donde η es el factor de aprendizaje, el cual se utiliza para regular la influencia del error en la actualización de los pesos. Las Redes Neuronales tienen la ventaja de que no necesitan un modelo físico o matemático para encontrar la relación de entradas y salidas, debido a que sólo requieren de datos de entrada y salida de un sistema. Las desventajas de esta metodología son la gran cantidad de datos necesarios para entrenar la Red y ser capaz de reconocer con precisión y confiabilidad diferentes casos, la alta capacidad computacional para el manejo, almacenamiento y procesamiento de datos, y la poca robustez ante cambios bruscos para una misma salida. Estas dos últimas desventajas son las que nos impulsan a trabajar con técnicas de Aprendizaje Profundo.

3.3. Aprendizaje Automático

El Aprendizaje Automático (o “Machine Learning”) es un conjunto de métodos que pueden detectar patrones en una gran cantidad de data, y luego utilizar los patrones hallados para realizar predicciones sobre datos futuros o tomar de decisiones (Murphy, 2012). Goodfellow (Goodfellow, 2016) lo define como la aplicación estadística mediante el uso de computadoras, para estimar funciones complejas. Un algoritmo de Aprendizaje Automático tiene la capacidad de “aprender” de una forma similar a como solemos hacerlo las personas: interpretando la relación entre las entradas y salidas conocidas de un sistema mediante ensayo-error, para luego identificar la salida correspondiente a una nueva entrada (Google Developer, 2020).

3.4. Aprendizaje Profundo

El Aprendizaje Profundo es una rama del Aprendizaje Automático cuyo objetivo es replicar la arquitectura de aprendizaje del cerebro humano, aprendiendo parámetros característicos en varios niveles de procesamiento con niveles de abstracción cada vez más altos (Goodfellow, 2016). Los algoritmos de Aprendizaje Profundo son potentes y flexibles debido a que representan el mundo como una jerarquía interconectada de conceptos, en donde cada concepto se relaciona con conceptos cada vez más simples y abstractos (Murphy, 2012). La Figura 45 muestra el Aprendizaje Profundo incluido como un tipo de Aprendizaje Automático, el cual es utilizado en muchas aplicaciones de Inteligencia Artificial.

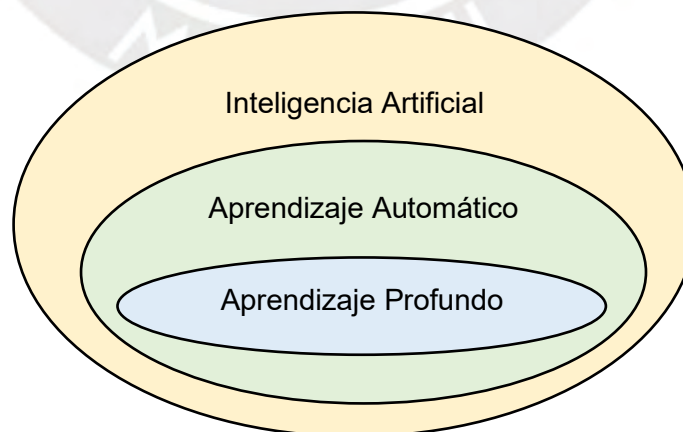


Figura 45. Diagrama de Venn representando Inteligencia Artificial, Aprendizaje Automático y Aprendizaje Profundo. Elaboración propia.

3.5. Redes Neuronales Convolucionales (CNN)

Las Redes Neuronales Convolucionales son una metodología de Aprendizaje Profundo en donde se requiere trabajar con grandes cantidades de datos, teniendo amplio uso y desarrollo en aplicaciones de procesamiento de imágenes (Saha, 2018). Las CNN son una de las técnicas más eficientes de aprendizaje profundo y su uso en diagnóstico de fallas en motores es bastante amplio (Lu, 2019). Estas son redes retroalimentadas en donde el flujo de información va en una sola dirección desde las entradas hasta las salidas. Se basaron en la biología del ser humano, en donde el cerebro alterna capas de células simples y complejas. Las CNN pueden presentar diferentes arquitecturas, pero generalmente contienen una etapa de convolución y extracción contenidos dentro de un módulo. Posteriormente, se tiene 1 o más capas conectadas de redes neuronales. Una arquitectura convencional de CNN incluye una primera etapa de convolución y extracción seguida por una red neuronal (Rawat, 2017). Las Redes neuronales Convolucionales combinan 3 funcionalidades para asegurar robustez e invarianza ante cambios de posición y distorsión: campos receptivos locales, pesos compartidos y submuestreo espacial o temporal (LeCun, 2007). Los campos receptivos locales son pesos dentro de un vecindario de la capa anterior que permiten extraer un valor representativo para la siguiente capa, con la finalidad de extraer aspectos representativos como puntos, esquinas o bordes. Cuando se tienen distorsiones o cambios bruscos, las posiciones de los parámetros representativos de la entrada suelen reubicarse, afectando la clasificación. Este problema es solucionado recorriendo toda la entrada con los campos receptivos locales, por lo que los parámetros representativos los encontrarán sin problemas. Una siguiente etapa de promedio y submuestreo reducirán la resolución y la sensibilidad a cambios bruscos o distorsión. La tabla 10 muestra la comparativa entre las Redes Neuronales y las Redes Neuronales Convolucionales.

	Redes Neuronales (NN)	Red Neuronal Convolutacional (CNN)
Invarianza frente a cambios bruscos y distorsión	No cuentan con mecanismo para lidiar con estos problemas.	Mediante el aprendizaje de aspectos, las CNN son robustas ante este tipo de distorsiones.
Procesamiento de datos	Cuando la cantidad de información se incrementa, el tiempo y costo computacional se hace inmanejable.	Durante las capas de convolución y extracción, la información se va reduciendo y haciéndose más abstracta, por lo que el tiempo y costo computacional es manejable.
Aplicaciones	Datos en función del tiempo,	Procesamiento de Imágenes - Reconocimiento de Patrones. Datos en función del tiempo

Tabla 10. Tabla Comparativa Redes Neuronales y Redes neuronales Convolucionales

La arquitectura básica de una Red Neuronal Convolutacional se muestra en la Figura 46. El problema consiste en reconocer el tipo de vehículo teniendo como entrada una imagen. La entrada es una imagen representada por una matriz de 3 dimensiones (X, Y, Color) de un auto. En la primera etapa, la imagen entra a un primer módulo con capas de Convolución y Extracción. La Convolución extrae los parámetros característicos de la imagen por cada uno de los tipos de filtros definidos, haciendo que la dimensión cambie según la cantidad de estos. La Extracción de tipo Valor Máximo o Valor Promedio, obtendrá el valor representativo para una ventana o vecindario de valores. El resultado del primer módulo entrará a un segundo módulo con capas de Convolución y Extracción, y así lo hará para todos los módulos definidos. Mientras mayor cantidad de módulos, el modelo crecerá haciéndose más complejo y profundo. Una vez terminado el Aprendizaje de Aspectos, tendremos una capa de aplanamiento en donde las dimensiones de la entrada se ajustarán a un Vector de 1 dimensión. Posteriormente, se tendrá una Red Neuronal y una capa de

activación para la Clasificación. La Figura 47 muestra una arquitectura similar aplicada para el reconocimiento de dígitos numéricos.

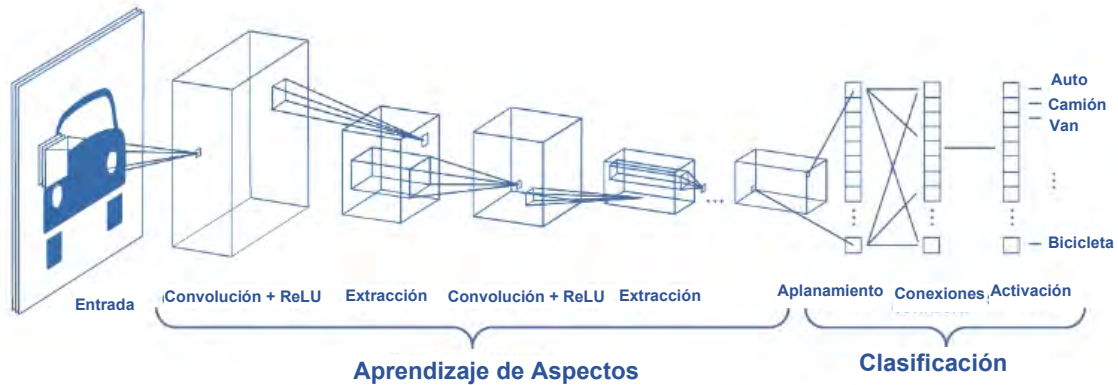


Figura 46. Red Neuronal Convolutiva. Fuente: Saha, 2018.

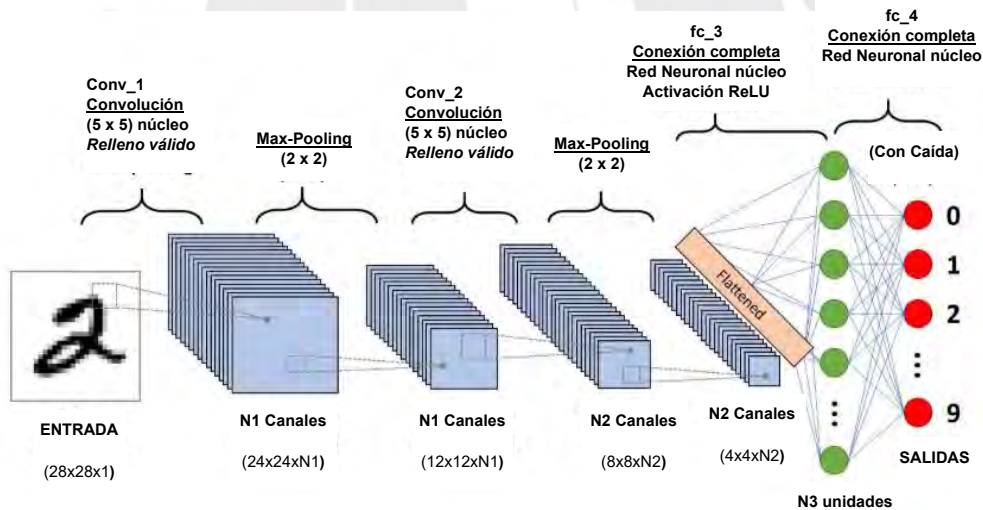


Figura 47. Ejemplo de implementación de una Red Neuronal Convolutiva para reconocimiento de dígitos escritos a mano. Fuente: Saha, 2018.

Las CNN son una variante de Red Neuronal para trabajar con matrices y arreglos en 2 dimensiones. Su objetivo es encontrar una relación entre entradas y salidas, para lo cual se divide en 2 etapas: Aprendizaje de Aspectos y Clasificación.

3.5.1. Aprendizaje de Aspectos

Debido a la gran cantidad de información contenida en la entrada, en la etapa de Aprendizaje de Aspectos, la señal de entrada es reducida convenientemente, manteniendo características o “features” relevantes que permitan una clasificación precisa y evitan la sobrecarga de información. Los datos de entrada se reducen, pero mantienen aspectos característicos que serán de gran utilidad para la clasificación. Estos conforman un descriptor o identificador de la señal, el cual no tiene un significado físico, pero es una representación robusta de la señal. Esta reducción de información se consigue mediante la función de convolución.

3.5.1.1. Convolución

La convolución es una operación matemática que permitirá obtener los parámetros característicos de la señal o imagen. En el caso de imágenes, una convolución se puede realizar con un filtro o kernel, la cual es una matriz con valores definidos y de un tamaño menor al de la imagen (Saha, 2018). Los filtros recorren todos los datos mediante la técnica de ventaneo. La cantidad de filtros y el tamaño de estos definirán las dimensiones de la nueva matriz tridimensional a la salida. A continuación, se presenta, de manera gráfica, cómo funciona la convolución en una imagen.

La Figura 48 muestra la foto de una casa. La imagen de 231 x 218 píxeles en escala de grises puede representarse como una matriz de 2 dimensiones en donde para cada par (X, Y) se asigna un valor de intensidad. Al tratarse de una imagen de 8 bits, los valores de intensidad se encontrarán entre 0 y 255.



Figura 48. Imagen de una casa, en escala de grises, 8-bit, 231 x 218 píxeles, Fuente: <https://shodhganga.inflibnet.ac.in/bitstream/10603/171122/2/11.%20chapter%203.pdf>

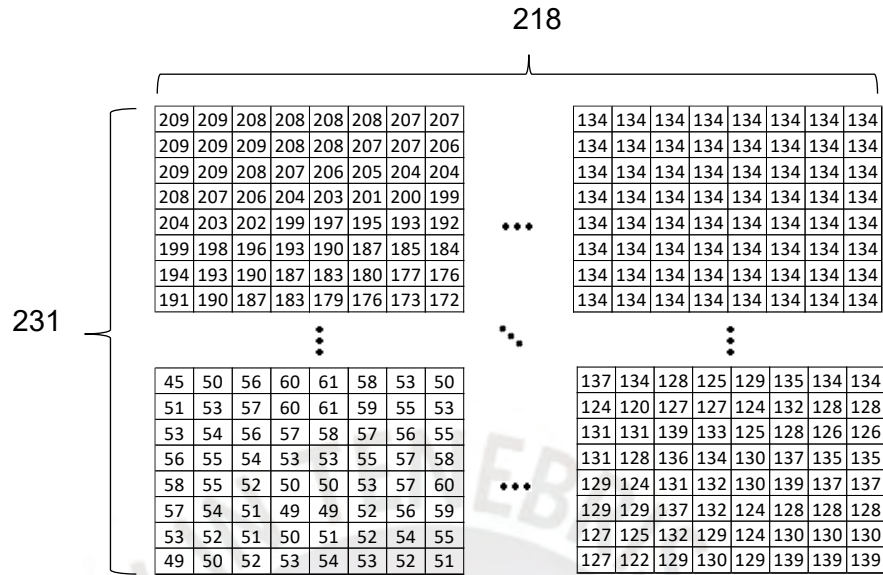


Figura 49. Representación matricial de la imagen con valores de intensidad. Elaboración propia.

Estos valores pueden ser normalizados para tener valores entre 0 y 1, dividiendo por 255.

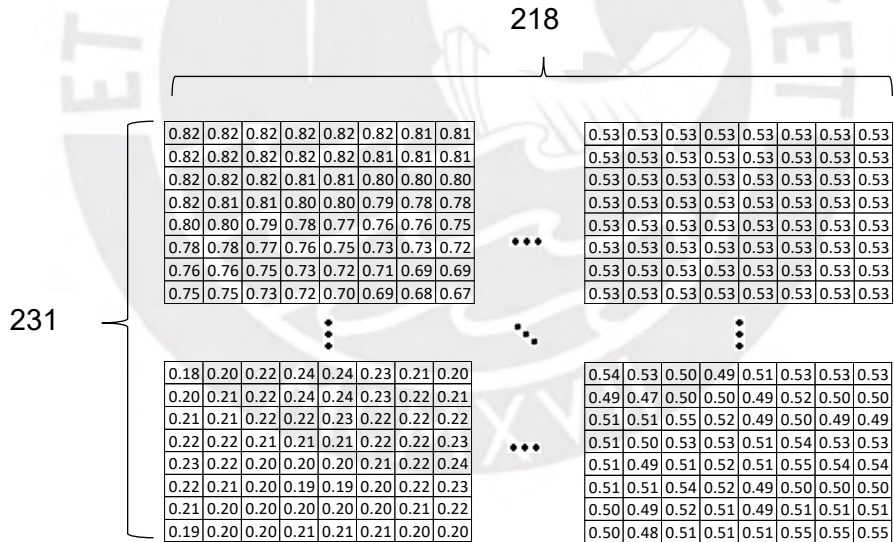


Figura 50. Representación matricial de la imagen con valores de intensidad normalizados (I_n). Elaboración propia.

Definimos el siguiente filtro o “kernel” para realizar la convolución y mostrar los parámetros característicos obtenidos para la imagen.

Filtro F_v , tamaño de [3x3]

-1	0	1
-1	0	1
-1	0	1

Realizando la operación de convolución de la imagen normalizada I_n con el filtro F_v , obtenemos la imagen resultado I_v .

$$I_v = I_n * F_v \dots \text{ (Ecuación 3.4)}$$

La convolución 2D es un proceso iterativo, en donde un filtro recorrerá la imagen y realizará operaciones entre sus elementos y los valores de la sección correspondiente de la imagen. En la primera iteración el filtro se ubica en la parte superior izquierda. La convolución, multiplica punto por punto los elementos del filtro con la porción de la imagen y posteriormente, suma los resultados, obteniendo la salida para la posición respectiva. A continuación, el filtro se moverá a la derecha y repetirá el mismo procedimiento. Al alcanzar la última columna, el filtro regresará a la primera columna y continuará con la siguiente fila. La Figura 51 muestra el procedimiento.

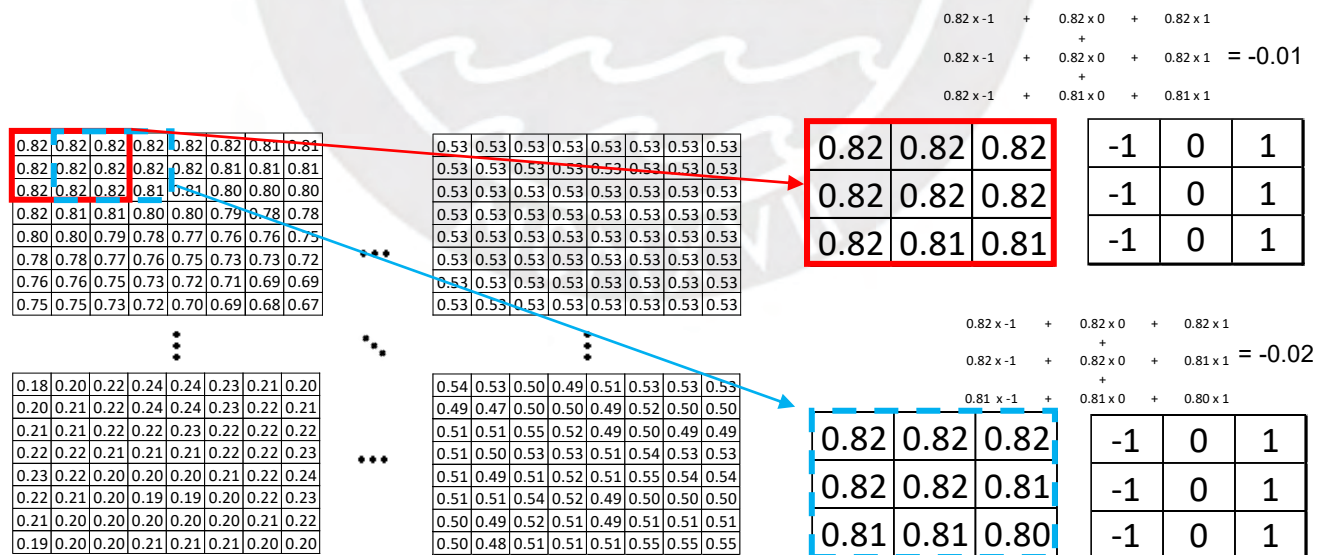


Figura 51. Convolución de la imagen normalizada I_n y el filtro F_v . Elaboración propia.

El resultado se puede observar en la siguiente Figura 52.



Figura 52. Imagen Resultado de la convolución.
Elaboración propia.

Aplicando la función de convolución con el filtro F_v , se han encontrado las líneas verticales de la imagen. El resultado se muestra en la Figura 53.

Si utilizamos el siguiente filtro:

Filtro F_h , tamaño de [3x3]

1	1	1
0	0	0
-1	-1	-1



Figura 53. Resultado de la convolución con el filtro F_h .
Elaboración propia.

De la convolución con el filtro F_h , han encontrado las líneas horizontales de la imagen.

En la etapa de Aprendizaje de Aspectos, se pueden definir la cantidad n filtros, de los cuales obtendremos n imágenes resultado de la convolución respectiva. Estos filtros tendrán coeficientes aleatorios, los cuales se irán ajustando de acuerdo a los resultados obtenidos en la etapa de la Clasificación.

3.5.1.2. Función de activación

Una función de activación relaciona una entrada con su salida.

Unidad Lineal Rectificada (ReLU)

Función que devuelve el mismo valor de entrada, si y solo sí, es mayor a 0. De lo contrario, entrega el valor de 0.

$$y = x, \text{ si } x > 0 \dots \text{ (Ecuación 3.5)}$$

$$y = 0, \text{ si } x \leq 0 \dots \text{ (Ecuación 3.6)}$$

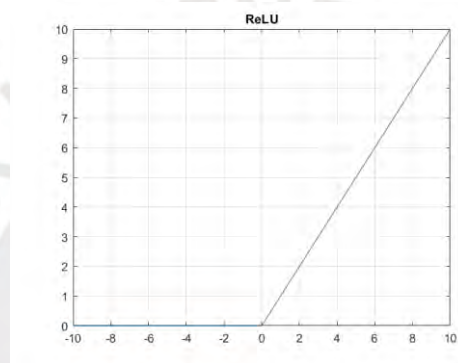


Figura 54. Función de Activación tipo ReLU. Elaboración propia.

Sigmoidea

La función de activación de tipo sigmoidea relaciona la entrada y salida de acuerdo a la siguiente función:

$$y = \frac{1}{1+e^{-x}} \dots \text{ (Ecuación 3.7)}$$

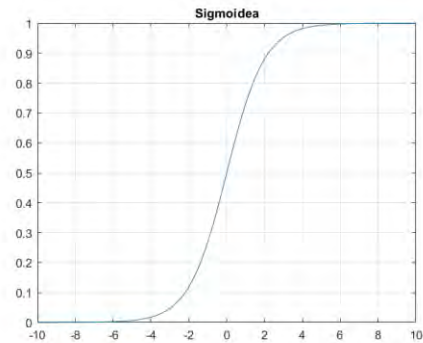


Figura 55. Función de Activación de tipo Sigmoidea. Elaboración propia.

Softmax

La función de activación de softmax (Google Developers, 2020) se utiliza cuando se tienen salidas de clases múltiples. La función Softmax asigna probabilidades decimales a cada tipo de salida las cuales suman 1 en total. Esta función se implementa previo a la capa de resultado y debe tener la misma cantidad de nodos que la capa de salida.

$$p(y = j|x) = \frac{e^{(w_j^T x + b_j)}}{\sum_{k \in K} e^{(w_k^T x + b_k)}} \dots \text{ (Ecuación 3.8)}$$

3.5.1.3. Extracción

La etapa de Extracción o “*Pooling*” busca extraer un valor representativo de la región de interés en la imagen (Saha, 2018). Para ello se busca dentro de la ventana el valor máximo (Max Pooling) o el valor promedio (Average Pooling). La Figura 61 muestra un ejemplo del proceso de Pooling.

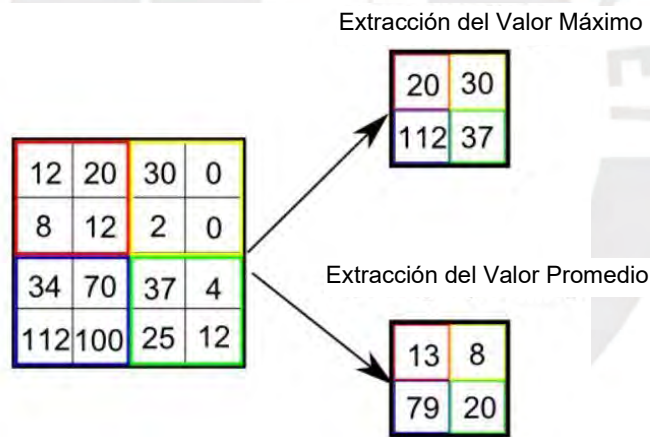


Figura 56. Extracción o Pooling. Fuente: Saha, 2018.

3.5.1.4. Parámetros y capas adicionales de la Red Neuronal Convolutiva

Muestra

Una muestra, o “sample”, es un elemento de un paquete de datos, como por ejemplo una imagen en una red convolutiva (Keras, 2020).

Lote

El lote, o “batch”, es un grupo de N muestras (Keras, 2020). Los lotes son procesados en paralelo y de manera independiente. Generalmente, el lote representa mejor la distribución de la entrada. Mientras más grande sea el lote, mejor será la aproximación, pero el tiempo y costo computacional será mayor.

Épocas

La cantidad de épocas o “epochs” define cuántas veces se iterará el aprendizaje y optimización sobre la data de entrada (Keras, 2020). Todos los datos se recorrerán durante la etapa de entrenamiento (Google Developer, 2020).

Caída

La capa de Caída o “Dropout” consiste en una técnica de regularización para reducir el sobreentrenamiento (Rawat, 2017). Esta convierte las unidades en 0 de forma aleatoria de acuerdo a la frecuencia configurada (TensorFlow, 2020). Hinton (Hinton, 2012) demuestra que al aplicar un Dropout del 50% disminuye el error durante la clasificación.

Aplanamiento

La capa de aplanamiento o “flatten”, transforma las dimensiones de la entrada sin afectar el tamaño del lote (Keras, 2020). La capa de aplanamiento construye un vector de 1 dimensión, el cual será la entrada a la Red Neuronal.

3.5.2. Clasificación

La etapa de Clasificación trabaja como una Red Neuronal convencional la cual tiene capas de entrada, intermedias y de salida. Las entradas de esta red serán los aspectos característicos obtenidos en la etapa anterior. Durante cada iteración, la red neuronal ajustará los parámetros tanto de las capas intermedias como de los filtros de la etapa anterior. Para ello se define una función de costo y un optimizador.

3.5.2.1. Funciones de Costo

Como se mencionó anteriormente, una función de costo busca minimizar el error de predicción entre las salidas actuales y las estimadas por la Red Neuronal.

Error Cuadrático Medio (MSE)

Un modelo predictivo de regresión busca predecir un valor real. Para ello existen funciones como el Error de Media Cuadrada o MSE. Este error es calculado como el promedio de la diferencia de la salida real y predicha. Su utilización se prefiere cuando se trabaja bajo condiciones de distribución de variable objetivo tipo Gaussiano (Brownlee, 2020).

$$J = \sum_{i=1}^n (y_i - \tilde{y}_i)^2 \dots \text{(Ecuación 3.9)}$$

Función de Costo Entropía Cruzada Binaria

La Entropía Cruzada es una medida de la diferencia entre dos distribuciones probabilísticas para variables aleatorias (Brownlee, 2020). Esta función se utiliza cuando se trabaja bajo condiciones de máxima similitud. Sólo se utiliza cuando se tienen 2 clases.

Función de Costo Entropía Cruzada Categórica

Al igual que la Entropía Cruzada Binaria, esta función obtiene la diferencia entre las salidas actuales y predicciones, con la diferencia de que esta puede trabajar para más de 2 clases (Keras, 2020).

3.5.2.2. Optimizadores

Para poder minimizar el error obtenido en la función de costo, se define un algoritmo de optimización.

Gradiente Descendiente

El algoritmo de optimización por Gradiente Descendiente es uno de los más populares para optimizar Redes Neuronales (Ruder, 2016). Esta se basa en la función Gradiente y busca

la minimización de esta para una función de costo J dada. Se tienen 3 variantes del algoritmo:

Gradiente Descendiente por Lote

Se calcula el gradiente de la función J con respecto al tiempo para los parámetros θ del sistema.

$$\theta = \theta - \eta * \nabla_{\theta} J(\theta) \dots \text{(Ecuación 3.10)}$$

Donde:

θ = Vector de Parámetros

$\nabla_{\theta} J(\theta)$ = Función Gradiente de J en θ

η = Factor de aprendizaje

Este cálculo es lento y consumo demasiados recursos computacionales debido a que se trabaja sobre toda la data actualizando 1 sola vez (Ruder, 2016).

Gradiente Descendiente Estocástico

El Gradiente Descendiente Estocástico (o SGD) se realiza para cada muestra $x^{(i)}$ con salida $y^{(i)}$

$$\theta = \theta - \eta * \nabla_{\theta} J(\theta; x^{(i)}; y^{(i)}) \dots \text{(Ecuación 3.11)}$$

Donde:

$x^{(i)}; y^{(i)}$ representan cada par muestra y salida.

A diferencia del Gradiente Descendiente por Lote, realiza una actualización por cada ejecución, por lo tanto, es mucho más rápido. Su desventaja está en que las actualizaciones

frecuentes ocasionan fluctuaciones que pueden ocasionar estancamientos en mínimos locales. Para compensar este problema, el factor de aprendizaje η puede reducirse.

Gradiente Descendiente Mini-Lote

Este algoritmo es un punto medio entre los dos anteriores (Ruder, 2016). Las actualizaciones se dan por cada Mini-Lote de n muestras de entrenamiento.

$$\theta = \theta - \eta * \nabla_{\theta} J(\theta; x^{(i:i+n)}, y^{(i:i+n)}) \dots \text{(Ecuación 3.12)}$$

Donde:

$x^{(i:i+n)}, y^{(i:i+n)}$ representan las muestras y salidas evaluadas por lote.

Así se logra reducir la fluctuación en la actualización de los parámetros y es más eficiente en tiempo y recursos computacionales.

AdaGrad

El algoritmo AdaGrad (Gradiente Adaptativo) actualiza los parámetros modificando el factor de aprendizaje, incrementándolo en parámetros inusuales y reduciéndolo en parámetros comunes (Ruder, 2016).

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_{i,ii} + \epsilon}} * \nabla_{\theta} J(\theta_{t,i}) \dots \text{(Ecuación 3.13)}$$

Donde G_i es una matriz diagonal con elementos i, i siendo la suma de los cuadrados de la gradiente con respecto al tiempo de θ_i hasta el paso t . El parámetro ϵ es un factor de control para evitar divisiones por 0. La principal ventaja del algoritmo AdaGrad es eliminar el ajuste manual del factor de aprendizaje. Por otro lado, su principal desventaja es la

acumulación de gradientes elevados al cuadrado en el denominador, el cual va creciendo por cada iteración.

ADADELTA

El algoritmo Adadelata es una mejora del algoritmo Adagrad (Ruder, 2016). Para evitar la acumulación en el denominador, esta se realiza en función de una ventana. Se utiliza el promedio de la gradiente $E|g^2|_t$ para cada paso de tiempo t:

$$g_{t,i} = \nabla_{\theta_i} J(\theta_{t,i}) \dots \text{(Ecuación 3.14)}$$

$$E|g^2|_t = \gamma E|g^2|_{t-1} + (1 - \gamma)g^2 \dots \text{(Ecuación 3.15)}$$

Donde γ es una variable de control.

Reemplazando en el algoritmo Adagrad se tiene:

$$\Delta\theta_t = - \frac{\eta}{\sqrt{E|g^2|_t + \epsilon}} g_t \dots \text{(Ecuación 3.16)}$$

Definiendo el error de raíz cuadrática media como:

$$RMS|\Delta\theta|_t = \sqrt{E|g^2|_t + \epsilon} \dots \text{(Ecuación 3.17)}$$

Finalmente, se obtiene:

$$\Delta\theta_t = - \frac{RMS|\Delta\theta|_{t-1}}{RMS|g|_t} g_t \dots \text{(Ecuación 3.18)}$$

$$\theta_{t+1} = \theta_t + \Delta\theta_t \dots \text{(Ecuación 3.19)}$$

RMSProp

El algoritmo RMSProp trabaja con el momento para generar la actualización de los parámetros (Kingma, 2015). Al igual que el algoritmo Adadelata, busca solucionar el problema de Adagrad (Ruder, 2016).

$$E|g^2|_t = 0.9 E|g^2|_{t-1} + 0.1g^2 \dots \text{(Ecuación 3.20)}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E|g^2|_t + \epsilon}} g_t \dots \text{(Ecuación 3.21)}$$

Adaptive Moment Estimation (Adam)

Es un algoritmo de optimización, desarrollado en (Kingma, 2015), el cual calcula el factor de aprendizaje de manera adaptativa para cada parámetro (Ruder, 2016). Al igual que los algoritmos Adadelta y RMSprop, Adam almacena el promedio de los gradientes cuadrados anteriores v_t de manera decreciente, pero, además, guarda el promedio de los gradientes pasados m_t .

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \dots \text{(Ecuación 3.22)}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \dots \text{(Ecuación 3.23)}$$

El algoritmo Adam propone la siguiente regla de actualización

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t \dots \text{(Ecuación 3.24)}$$

Donde $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$

Los resultados obtenidos en (Ruder, 2016) demuestran que Adam es el algoritmo con mejor performance.

3.5.3. Diseño de una Red Neuronal Convolutiva para el diagnóstico de fallas en componentes industriales

En general, las CNN se utilizan para trabajar con datos matriciales en 2 dimensiones como son las imágenes (O'Shea et. al, 2015). En ocasiones, para trabajar con señales de 1D, se transforma estos datos en datos 2D. Por ejemplo, en (Zihlmann, 2017), se utiliza el espectrograma para representar una señal de ECG a una imagen 2D.

En el presente trabajo de tesis, se diseñará una Red Neuronal Convolutiva de una dimensión debido a que se utilizan señales obtenidas mediante análisis vibracional, es decir que los datos de entrada son señales de vibración en el tiempo, por lo tanto, son señales unidimensionales. Además, se consideran las ventajas de la CNN de 1D frente a distorsiones y cambios bruscos de la señal y la robustez de la clasificación al trabajar con parámetros característicos de las señales. Como alternativa es posible acondicionar la señal y obtener una imagen que represente la entrada, pero, esta opción no es considerada, dado la capacidad computacional requerida para trabajar con imágenes o entradas en 2 dimensiones condiciona la implementación.



Diseño de la Red Neuronal Convolutiva de 1D

Para nuestro caso trabajaremos con señales en función del tiempo, por lo tanto, serán de 1 dimensión. Utilizaremos el siguiente esquema:

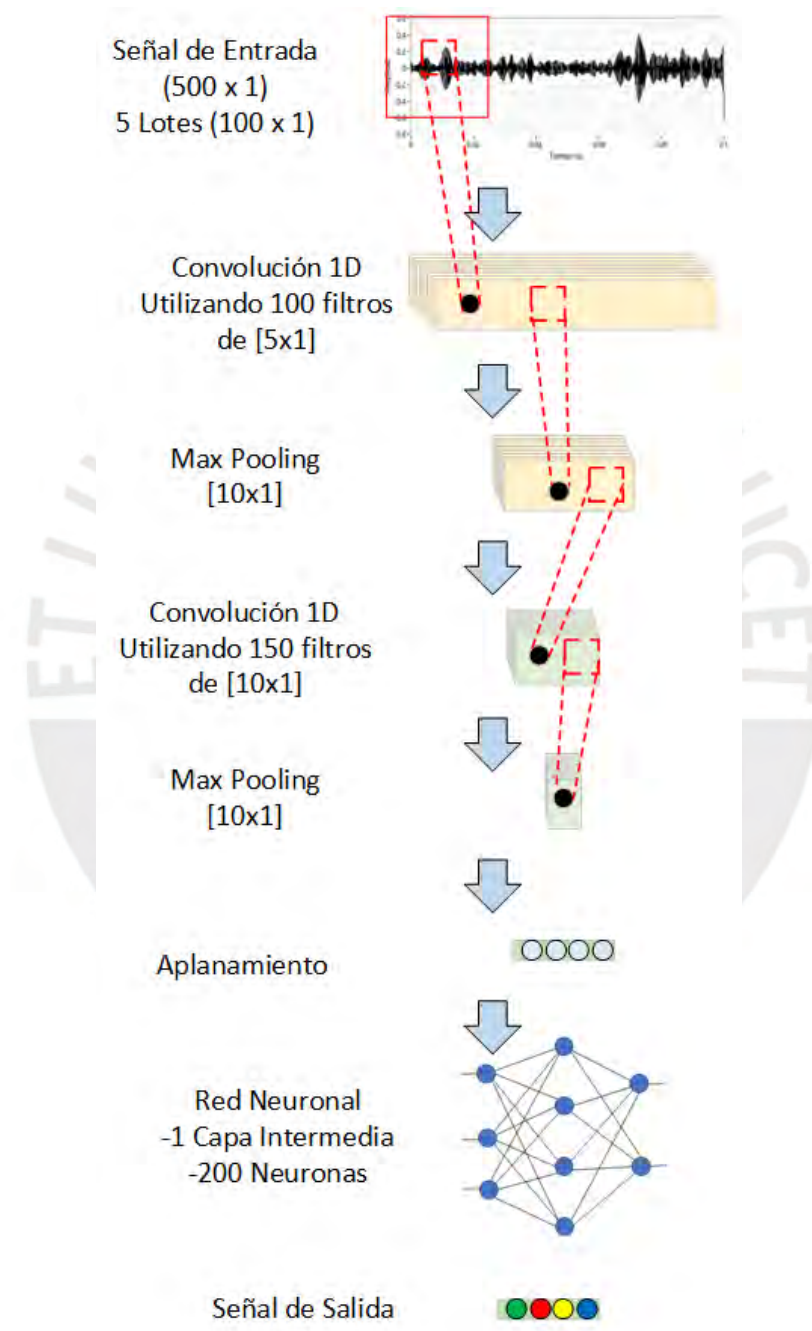
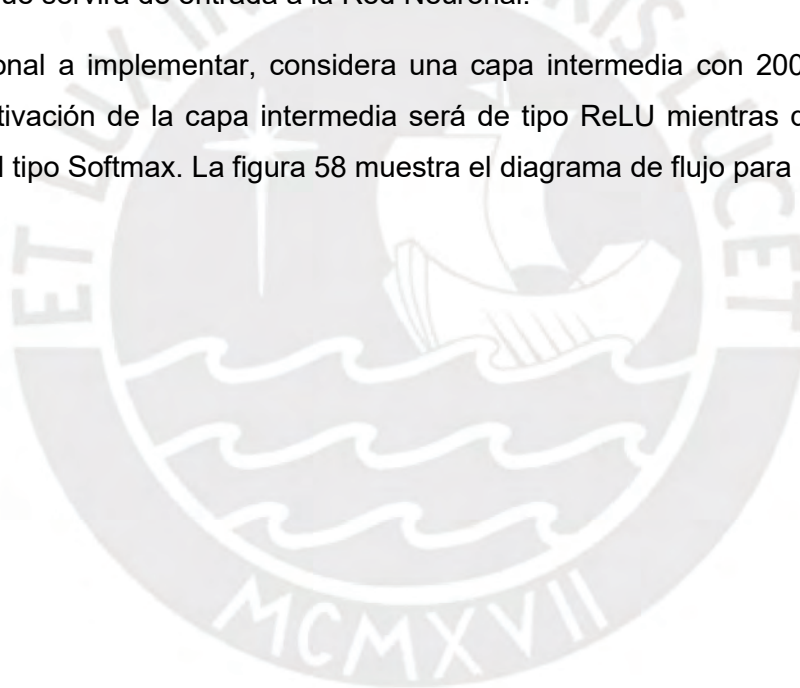


Figura 57. Esquema de implementación de Red Neuronal Convolutiva 1D. Elaboración propia.

Los datos de entrada serán las señales de vibración y ruido adquiridas por los sensores. La cantidad de muestras serán 500, equivalentes a 10 milisegundos y estas serán divididas en 5 lotes, teniendo 100 muestras cada una.

Se implementan 2 etapas de Aprendizaje de Aspectos, las cuales incluyen una capa de convolución 1D y una capa de extracción. En la primera etapa, la convolución 1D considerará 100 filtros de tamaño 5 y la capa de extracción será del tipo Max Pooling y de tamaño 10. En la segunda etapa, la capa de convolución 1D incluirá 150 filtros adicionales de tamaño 10 y una capa de extracción del tipo Max Pooling de tamaño 10. Las funciones de activación de serán del tipo ReLU para las 2 etapas. Con estas 2 etapas de Aprendizaje de Aspectos, garantizamos un vector que contengan aspectos abstractos que representen cada una de las clases. Como siguiente etapa tendremos el aplanamiento de información en un vector que servirá de entrada a la Red Neuronal.

La Red Neuronal a implementar, considera una capa intermedia con 200 neuronas. La función de activación de la capa intermedia será de tipo ReLU mientras que la capa de salida será del tipo Softmax. La figura 58 muestra el diagrama de flujo para el algoritmo.



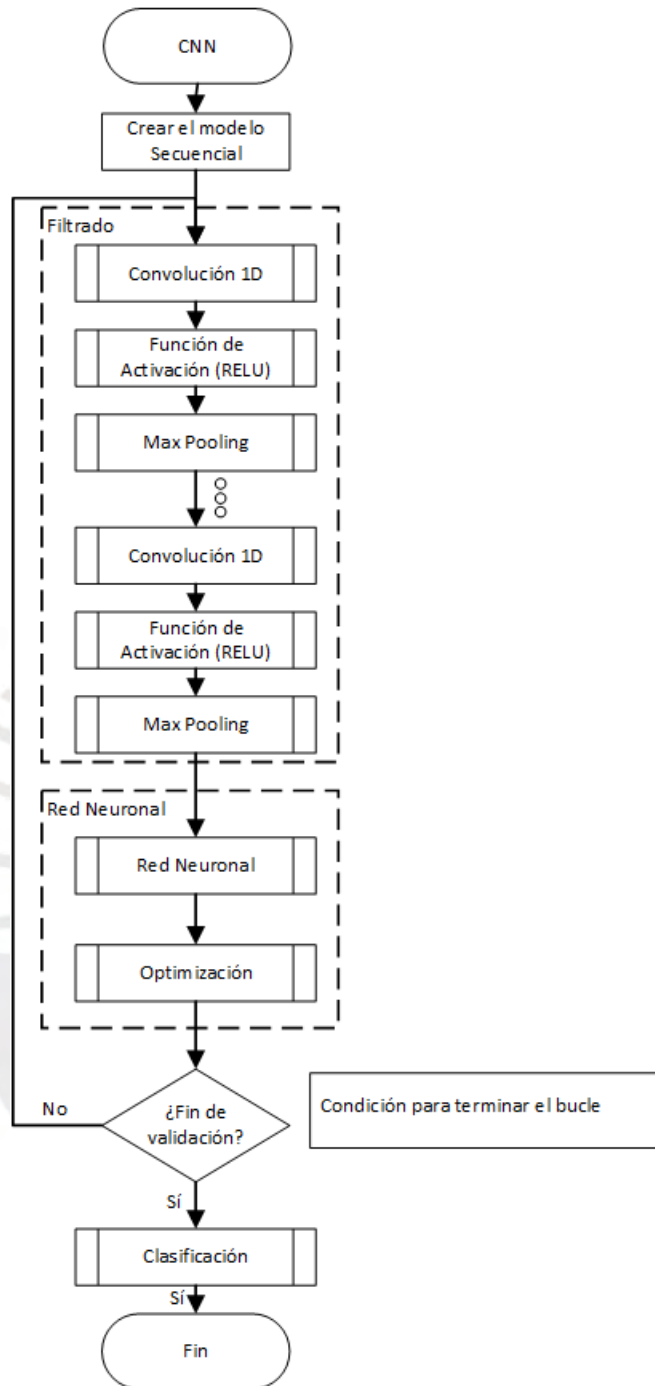


Figura 58. Diagrama de Flujo de algoritmo CNN 1D. Elaboración propia.

El algoritmo empezará creando un modelo de tipo secuencial. Luego entrará en un bucle de 100 iteraciones en donde se ejecutarán las 2 etapas del Aprendizaje de Aspectos y la Red Neuronal. Finalmente, para la Optimización se utilizará la función correlación cruzada categórica y el optimizador de ADAM.

3.5.4. Programación de la red neuronal diseñada.

El lenguaje de Programación a utilizar para implementar la Red Neuronal Convolutiva será Python.

3.5.4.1. Python

Python es un lenguaje de programación interpretado, el cual tiene la particularidad de ser gratuito (Python, 2020). Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, programación funcional. Dentro de las librerías o paquetes de funciones de Python tenemos la librería **NumPy** para operar con vectores y matrices las cuales nos darán mayor soporte y funciones matemáticas de alto nivel (Numpy, 2020) y **Pandas**, una interfaz de programación de aplicaciones basadas en Numpy enfocada a trabajar los datos como columnas (Google Developers, 2020). Python tiene varias versiones para descargar (Python, 2020). Elegimos trabajar con la versión 3.6, dado que tiene soporte para trabajar con las librerías de Keras y TensorFlow.

3.5.4.2. Keras

Keras, es una librería gratuita de Python para el desarrollo y evaluación de modelos de aprendizaje profundo (Keras, 2020). Dentro de sus funciones principales están la creación de un modelo, adición de capas, compilación, entrenamiento y validación de los modelos. Como primer paso se deberá definir un modelo, el cual puede ser Secuencial o Multicapa. El **Modelo Secuencial** es el más simple para trabajar, ya que considera sólo 1 tipo de entradas y 1 tipo de salidas. Una vez definido el modelo, se procede a adicionar las capas. Mediante la función `add(Dense())`, se crean las capas de la Red Neuronal. Como entradas se solicitan la cantidad de neuronas y el tipo de función de activación. La función `add()` también se utiliza para crear las etapas de Convolución en 1 dimensión y el Max Pooling.

Para la función Convolución 1D se deben considerar los siguientes parámetros:

Argumentos	Descripción	Tipo de dato
Filtros	Dimensión del espacio de salida, establece el número de filtros de salidas en la convolución.	Numérico Entero

Tamaño de filtro (o kernel)	Longitud de la ventana de convolución	Numérico Entero o lista de un solo elemento numérico entero
Activación	Función de activación	ReLU, Sigmoid, Softmax, etc.

Tabla 11. Parámetros de la función Convolución 1D

Para la función Max Pooling 1D se debe ingresar el tamaño de la ventana a ser reducida.

Una vez estructurado el modelo, se procederá a compilarlo. Esto se realiza mediante la función `compile()`. Como entradas a la función es mandatorio seleccionar una función de costo y un optimizador.

El modelo compilado deberá ser entrenado utilizando la función `fit()`. Como entradas obligatorias se tienen los datos de entrenamiento y sus respectivas etiquetas, la cantidad de épocas y el tamaño de lote.

Finalmente, se deberá validar el modelo utilizando la función `evaluate()`. Como entradas obligatorias se tienen los datos a validar y sus respectivas etiquetas, y el tamaño de lote.

3.4.3.3. TensorFlow

TensorFlow es una plataforma de extremo a extremo de código abierto para aprendizaje automático a través de un rango de tareas, y desarrollado por Google (TensorFlow, 2020). Esta satisface la necesidad de implementar sistemas capaces de construir y entrenar redes neuronales para detectar y descifrar patrones y correlaciones, análogos al aprendizaje y razonamiento usados por los humanos. (TensorFlow, 2020). TensorFlow se utiliza principalmente para Aprendizaje Máquina (Google Developers, 2020) pero también en tareas que requieran cómputo numérico mediante gráficos de flujo de datos.

Tensor

Es la estructura principal para los programas de TensorFlow (Google Developers, 2020). Un tensor es un arreglo multidimensional de datos, los cuales pueden ser valores escalares, vectores o matrices con diferentes tipos de unidades.

3.6. Pruebas del algoritmo

3.6.1. Información de la Base de Datos

Utilizaremos la Base de datos MAFAULDA (MAFAULDA, 2020) para entrenar nuestro modelo y validar los resultados. A continuación, se presentan las especificaciones del Banco de Pruebas utilizado:

Banco de Pruebas	SpectraQuest's Machinery Fault Simulator (MFS) Alignment-Balance-Vibration (ABVT)
Motor	¼ CV DC
Rango de Frecuencia	700-3600 rpm
Peso del sistema	22 kg
Diámetro del eje	16 mm
Longitud del eje	520 mm
Rotor	15.24 cm
Distancia de rodamientos	390 mm
Número de rodamientos	8
Diámetro de rodamientos	0.7145 cm
Diámetro de jaula	2.8519 cm
FTF	0.375 CPM/rpm
BPFO	2.998 CPM/rpm
BPFI	5.002 CPM/rpm
BSF	1.871 CPM/rpm

Tabla 12. Base de Datos MAFAULDA (MAFAULDA, 2020)

Información del sistema de adquisición de datos:

- 3 x Sensores Industriales IMI, Acelerómetros Modelo 601A01 en el eje Axial, Radial y Tangencial
 - Sensibilidad: 100 mV por g
 - Rango de Frecuencia: 16-600000 CPM (0.27-10.000 Hz)
 - Rango de Medición: $\pm 50 \text{ g}$ ($\pm 490 \frac{m}{s^2}$)

- 1 x Sensor IMI, Acelerómetro Triaxial Modelo 604B31, el cual entrega información de los ejes Axial, Radial y Tangencial.
 - Sensibilidad: 100 mV por g
 - Rango de Frecuencia: 30-300000 CPM (0.5-5.000 Hz)
 - Rango de Medición: $\pm 50 \text{ g}$ ($\pm 490 \frac{m}{s^2}$)
- 1 x Tacómetro analógico Monarch Instrument MT-190
- 1 x Micrófono Shure SM81 con rango de frecuencia de 20 – 20,000 Hz
- 2 x Tarjetas National Instruments NI9234, con módulos de adquisición de 4 canales analógicos, con frecuencia de muestro de 51.2 kHz.

Medición de Datos

Los datos fueron generados a una frecuencia de muestreo de 50 kHz durante 5 segundos, por lo que cada archivo tiene 250000 muestras.

Para esta evaluación, utilizaremos 66 mediciones de 5000 muestras medidas por los acelerómetros 601A01 en cada uno de los tres ejes a una frecuencia de muestreo de 50 kHz. Las Figuras 59 y 60 muestran la distribución de los datos.

Datos de Entrenamiento: 56 Datos

- 14 Datos Normales
- 14 Datos de Desalineamiento Horizontal 2mm
- 14 Datos de Desbalance 35g
- 14 Datos de Fallas en Rodamientos 20 g y 35 g

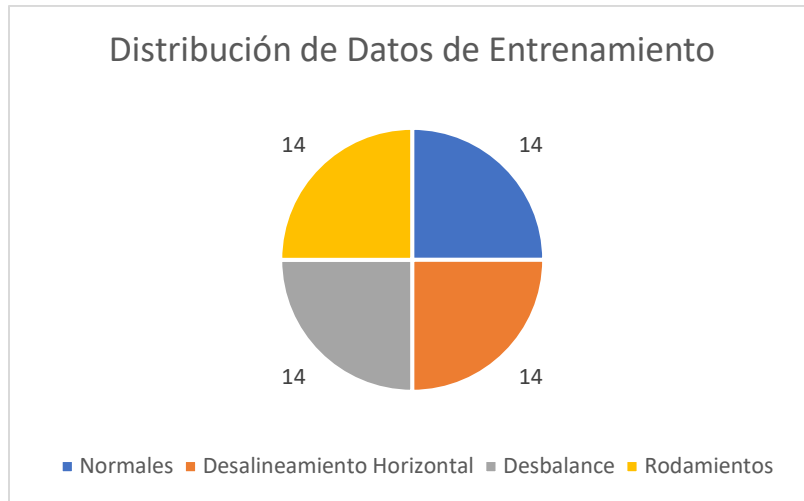


Figura 59. Distribución de Datos de Entrenamiento. Elaboración propia.

Datos de Validación: 10 Datos

- 2 Datos Normales
- 2 Datos de Desalineamiento Horizontal 2mm
- 3 Datos de Desbalance 35g
- 3 Datos de Fallas en Rodamientos 20 g y 35 g

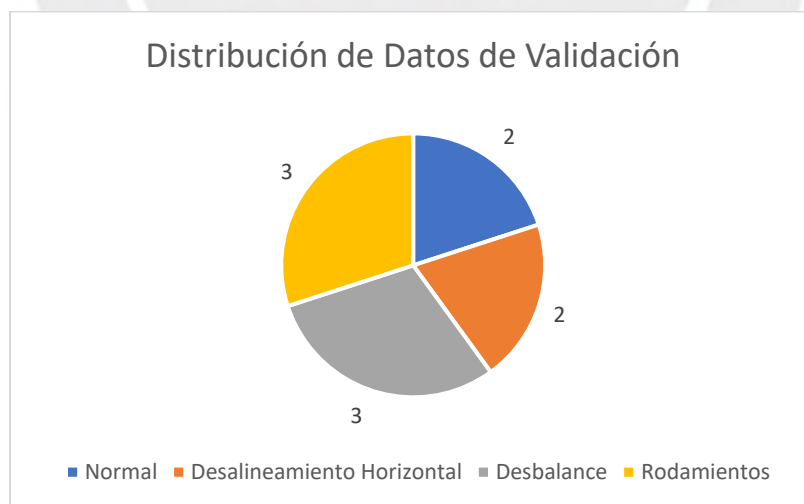


Figura 60. Distribución de Datos de Validación. Elaboración propia.

3.7. Evaluación de parámetros

Para encontrar los parámetros óptimos del sistema, variaremos uno por uno los parámetros más relevantes para evaluar el desempeño en el porcentaje de entrenamiento, error y porcentaje de validación. La figura 61 muestra las siguientes condiciones iniciales en el diagrama de flujo. Estas condiciones parten del ejemplo estudiado en (Brownlee, 2020) para el modelamiento de un sistema de reconocimiento de actividad física en base a mediciones de aceleración.

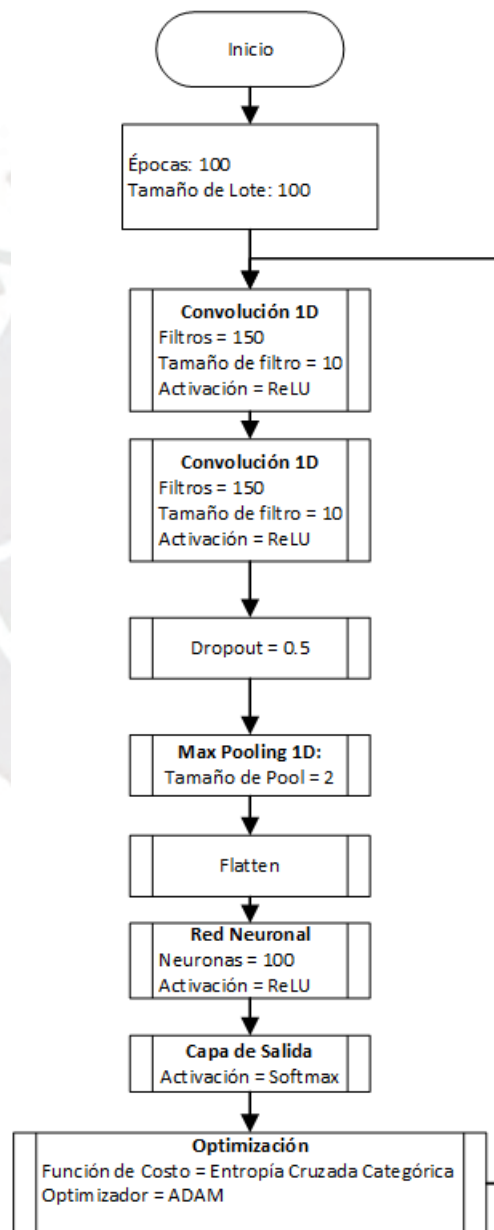


Figura 61. Condiciones Iniciales del Modelo. Elaboración propia.

En los siguientes diagramas de flujo, variaremos el parámetro indicado en una X y evaluaremos los resultados obtenidos.

Variación de épocas

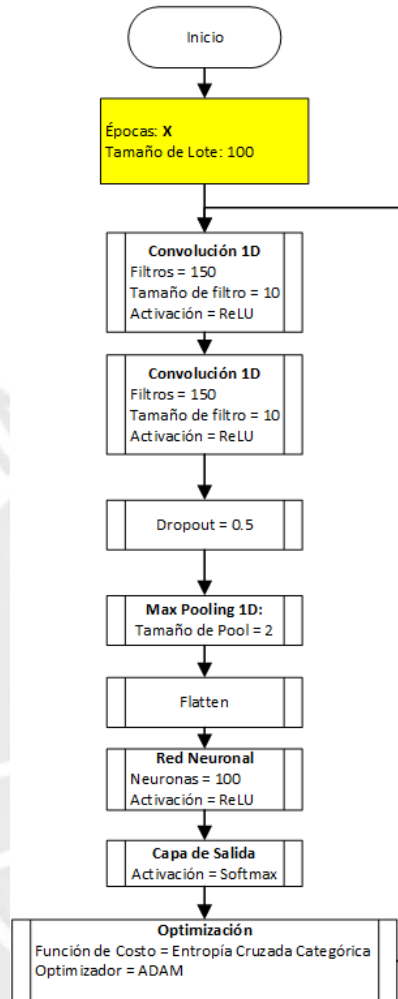


Figura 62. Variación de cantidad de épocas. Elaboración propia.

Épocas (X)	Porcentaje de efectividad en la etapa de Entrenamiento (%)	Error	Porcentaje de efectividad en Validación (%)
2	25	1.6192	30
5	67.86	1.0972	50
10	82.14	0.4589	50
20	100	0.0013	80
50	100	0.000077803	80
100	100	0.000081957	80

Tabla 13. Resultados de la variación de épocas

Variación de Tamaño de Lote (Batch size)

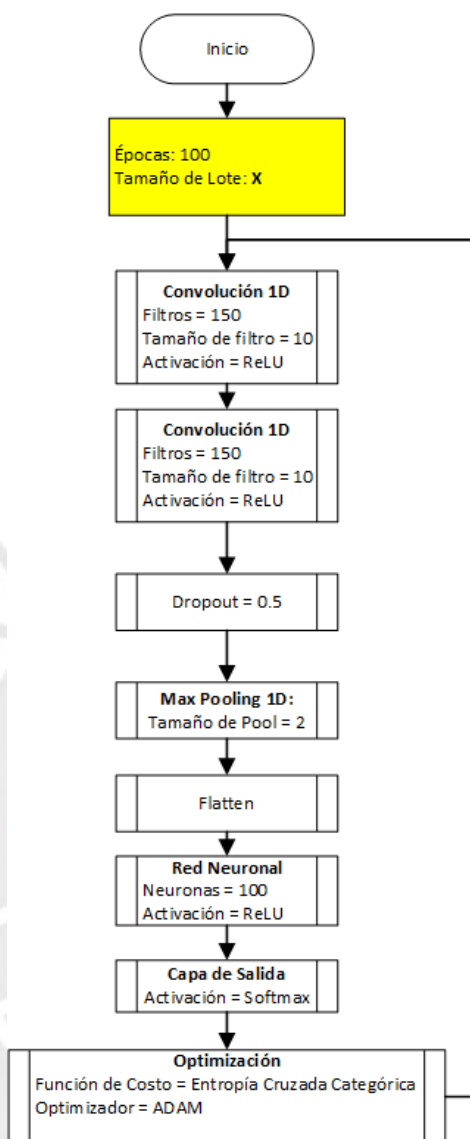


Figura 63. Variación del tamaño del lote. Elaboración propia.

Tamaño de Lote (X)	Porcentaje de efectividad en la etapa de Entrenamiento (%)	Error	Porcentaje de efectividad en Validación (%)
10	100	0.000026657	80
20	100	0.000051994	80
50	100	0.000076637	80
100	100	0.0000063753	80
250	100	0.00014237	80
500	100	0.000079848	80

Tabla 14. Resultados de la variación del tamaño de lote

1era Etapa de Convolución 1D: Variación del número de filtros

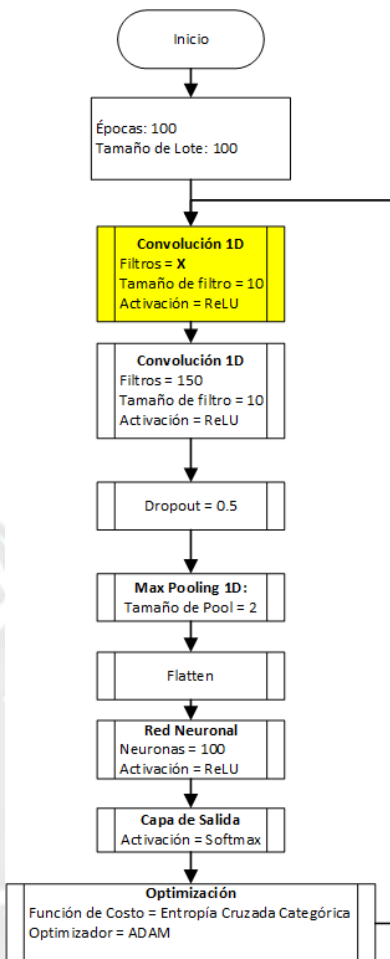


Figura 64. Variación de la cantidad de filtros en la 1era Etapa de Convolución 1D. Elaboración propia.

Número de Filtros (X)	Porcentaje de efectividad en la etapa de Entrenamiento (%)	Error	Porcentaje de efectividad en Validación (%)
10	100	0.000080742	80
20	100	0.00008615	80
50	100	0.000009611	80
100	100	0.0000074737	80
150	100	0.000018659	80
200	100	0.000011644	80

Tabla 15. Resultado de variación de número de Filtros en primera etapa Convolución 1D

1era Etapa de Convolución 1D: Variación del tamaño de filtro

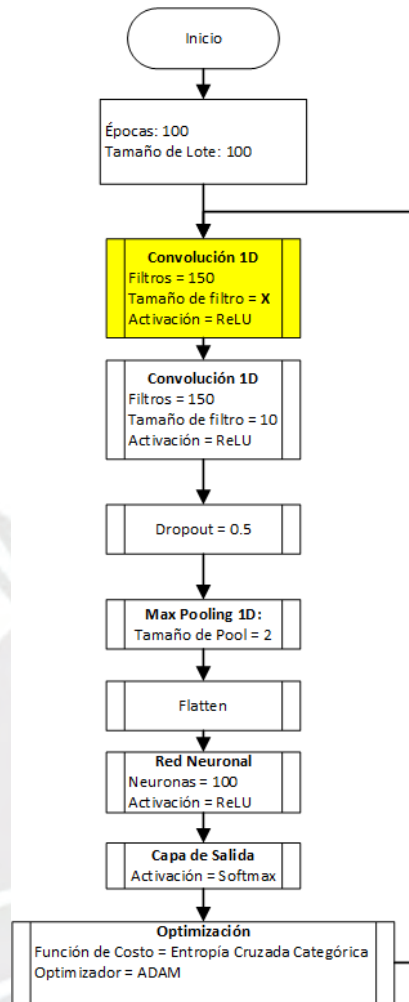


Figura 65. Variación del tamaño de filtro en la primera etapa de Convolución 1D. Elaboración propia.

Tamaño de Filtros (X)	Porcentaje de efectividad en la etapa de Entrenamiento (%)	Error	Porcentaje de efectividad en Validación (%)
3	100	0.000010496	80
5	100	0.0000082571	80
10	100	0.00001098	80
20	100	0.00005213	80
50	100	0.000028458	80
100	100	0.000029287	80

Tabla 16. Resultado de variación del tamaño del filtro en la 1era etapa de Convolución 1D

1era Etapa de Convolución 1D: Variación de la Función de Activación

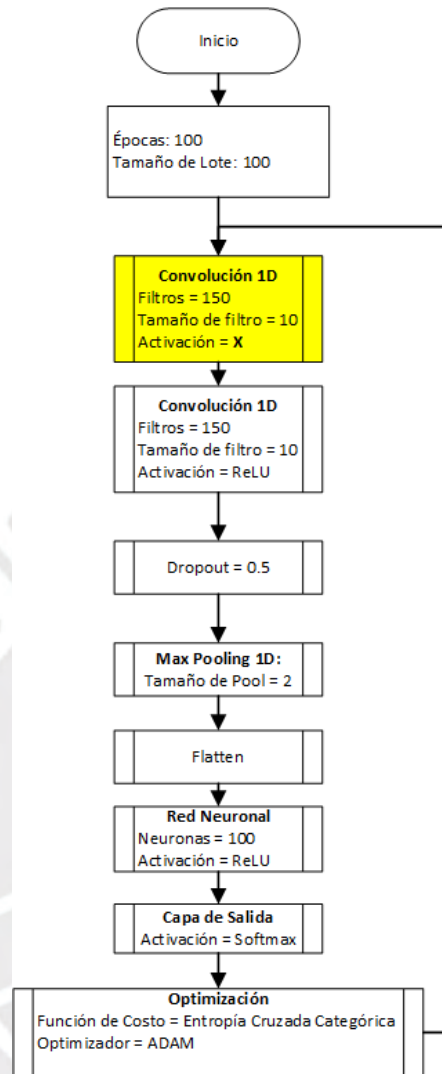


Figura 66. Variación de la función de activación en la primera etapa de Convolución 1D. Elaboración propia.

Función de Activación (X)	Porcentaje de efectividad en la etapa de Entrenamiento (%)	Error	Porcentaje de efectividad en Validación (%)
ReLU	100	0.0000038934	80
Sigmoidea	25	1.3863	20
Softmax	91.07	0.3306	50

Tabla 17. Resultado de variación de la función de activación en la 1era etapa de Convolución 1D

2da Etapa de Convolución 1D

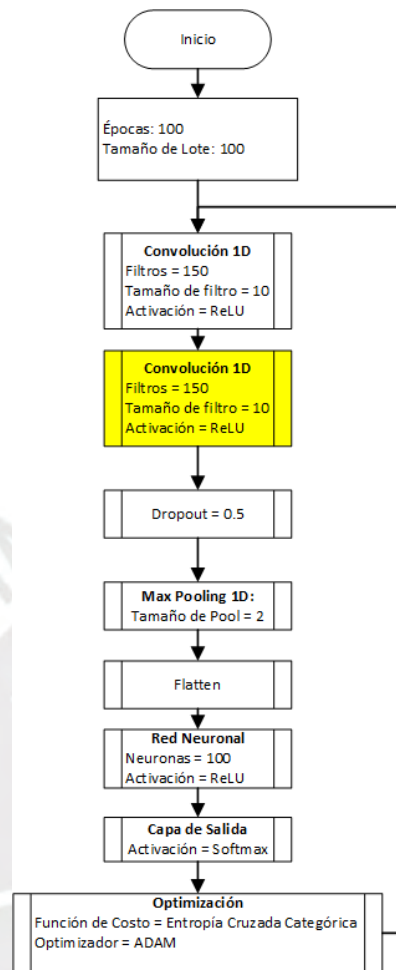


Figura 67. Variación de la segunda etapa de Convolución 1D. Elaboración propia.

2da Etapa de Convolución 1D (X)	Porcentaje de efectividad en la etapa de Entrenamiento (%)	Error	Porcentaje de efectividad en Validación (%)
Sin 2da Etapa de Convolución 1D	100	0.00071255	80
Con 2da Etapa de Convolución 1D	100	0.0000072546	80

Tabla 18. Resultado de variación de la 2da etapa de Convolución 1D

Variación de parámetro Dropout

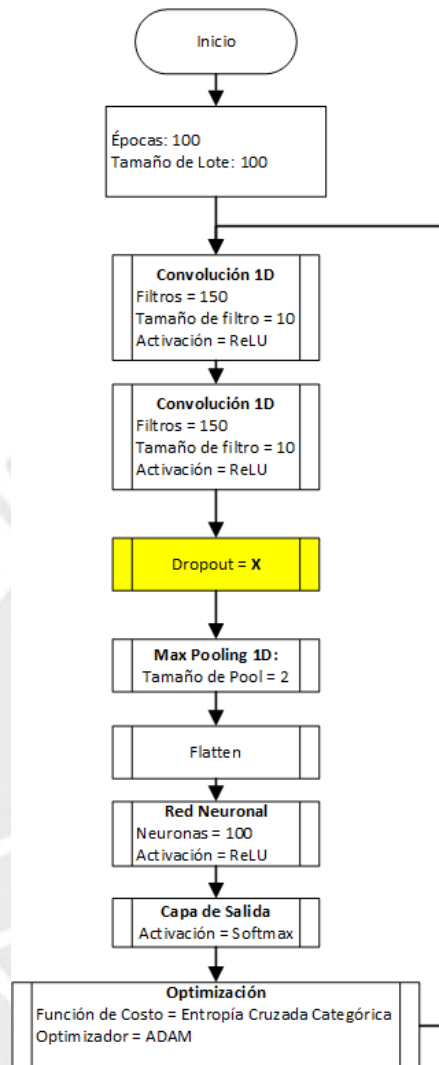


Figura 68. Variación del parámetro Dropout. Elaboración propia.

Dropout (X)	Porcentaje de efectividad en la etapa de Entrenamiento (%)	Error	Porcentaje de efectividad en Validación (%)
Sin Dropout	100	0.0000022011	60
0.1	100	0.0000047406	60
0.25	100	0.00002228	80
0.5	100	0.00017472	80

Tabla 19. Resultados de la variación del parámetro Dropout

Variación en etapa de Extracción (Pooling)

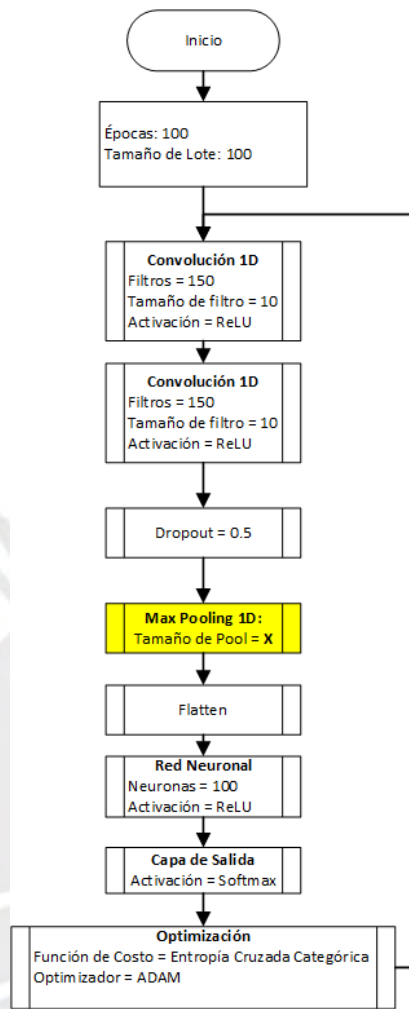


Figura 69. Variación de la etapa de Extracción. Elaboración propia.

Tamaño del pooling (X)	Porcentaje de efectividad en la etapa de Entrenamiento (%)	Error	Porcentaje de efectividad en Validación (%)
2	100	0.000011016	80
3	100	0.000030818	80
5	100	0.00022495	80
10	100	0.00040003	90
20	100	0.00056589	90
50	100	0.00046608	90

Tabla 20. Resultados de la variación en la etapa de Extracción

Red Neuronal: Variación en número de neuronas

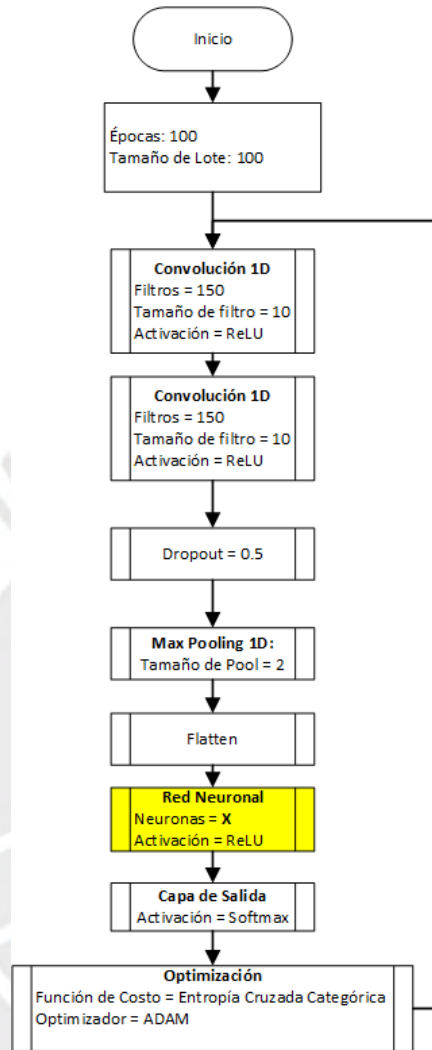


Figura 70. Variación de la cantidad de neuronas en la Red Neuronal. Elaboración propia.

Cantidad de neuronas en Capa Intermedia de Red Neuronal (X)	Porcentaje de efectividad en la etapa de Entrenamiento (%)	Error	Porcentaje de efectividad en Validación (%)
Sin Capa Intermedia	100	0.00042613	80
10	100	0.000042495	70
25	100	0.00034648	80
50	100	0.0000091363	80
100	100	0.00002456	80
200	100	0.0000071672	80

Tabla 21. Resultado de variación de la cantidad de neuronas en la Red Neuronal

Red Neuronal: Variación en función de activación de la capa intermedia

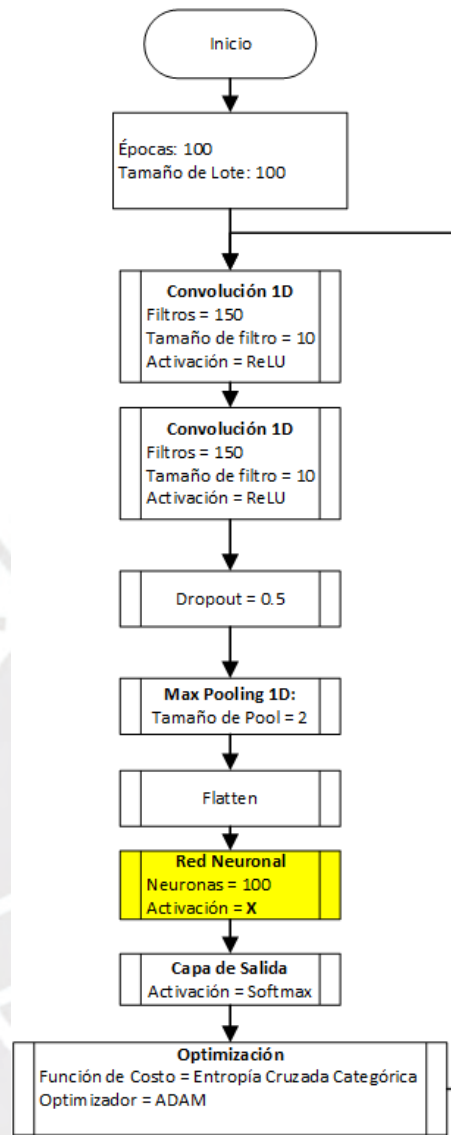


Figura 71. Variación de la función de activación de la Red Neuronal. Elaboración propia.

Función de Activación en Capa Intermedia (X)	Porcentaje de efectividad en la etapa de Entrenamiento (%)	Error	Porcentaje de efectividad en Validación (%)
ReLU	100	0.00004878	80
Sigmoidea	100	0.0052	80
Softmax	91.07	1.0423	80

Tabla 22. Resultado de la variación de la función de activación en la Red Neuronal

Red Neuronal: Variación de la función de activación de la capa de salida

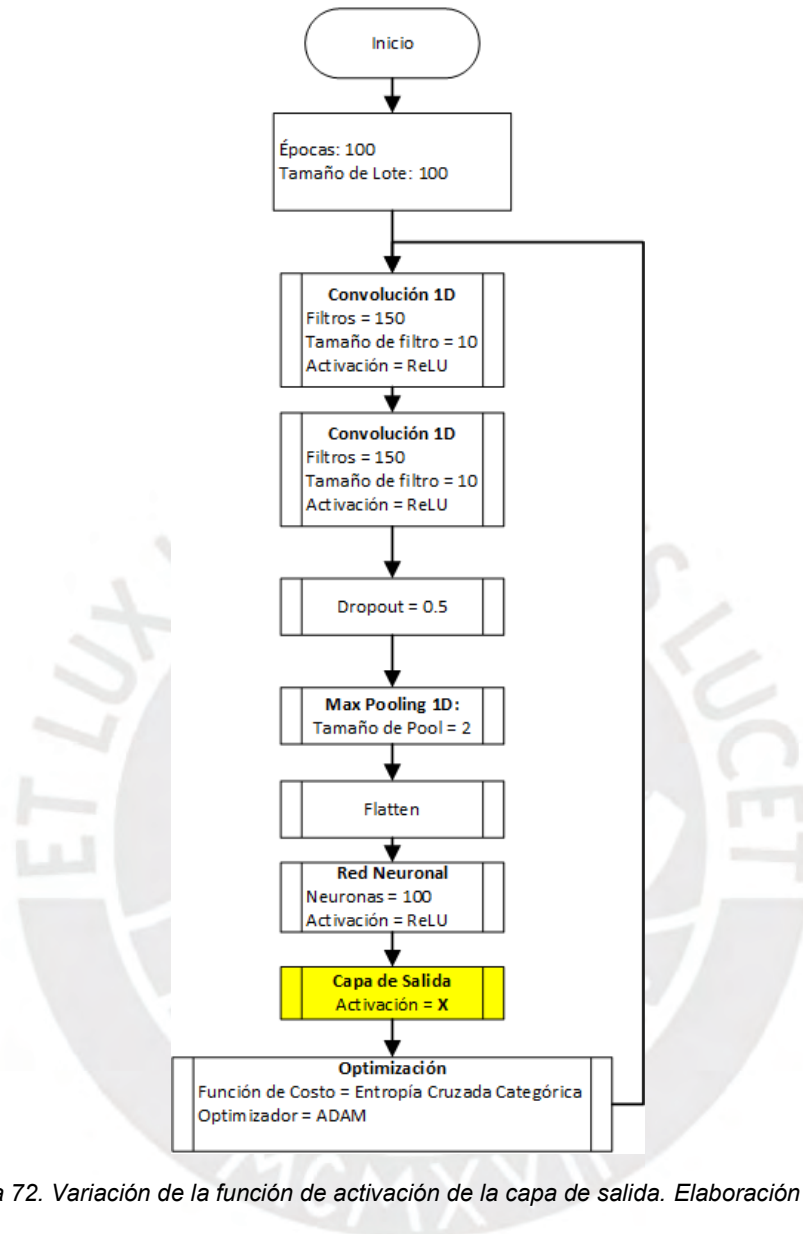


Figura 72. Variación de la función de activación de la capa de salida. Elaboración propia.

Función de Activación en Capa de Salida (X)	Porcentaje de efectividad en la etapa de Entrenamiento (%)	Error	Porcentaje de efectividad en Validación (%)
ReLU	25	Nan - División por 0	--
Sigmoidea	25	Nan - División por 0	--
Softmax	100	0.000011796	80

Tabla 23. Resultado de la variación de la función de activación en la capa de salida

Optimización: Variación de la Función de Costo

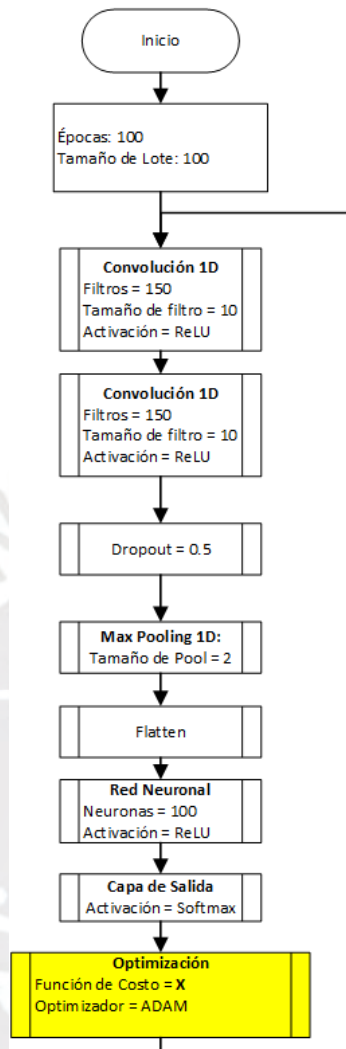


Figura 73. Variación de la función de costo en la optimización. Elaboración propia.

Función de Costo en Optimización (X)	Porcentaje de efectividad en la etapa de Entrenamiento (%)	Error	Porcentaje de efectividad en Validación (%)
Error Cuadrático Medio	100	0.0000030839	50
Entropía Cruzada Categórica	100	0.00010657	80

Tabla 24. Resultado de la variación de la función de costo en la optimización

Optimización: Variación del Optimizador

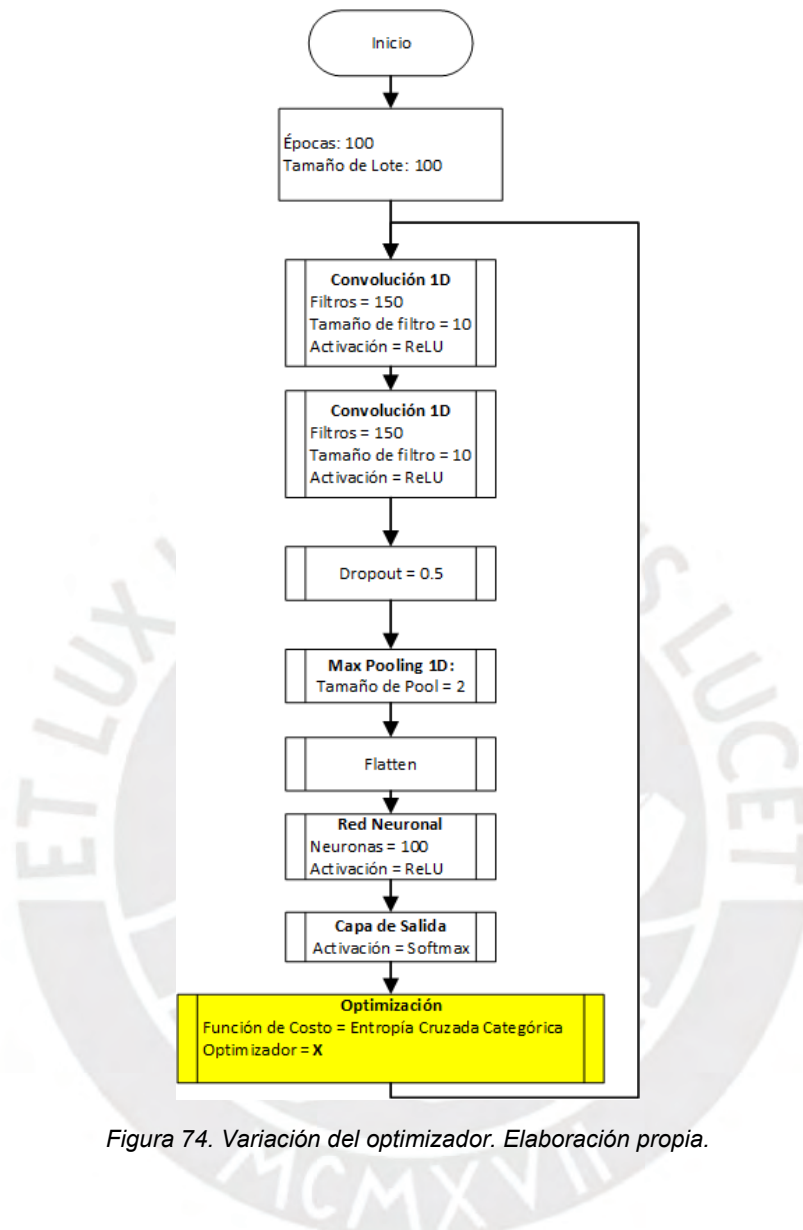


Figura 74. Variación del optimizador. Elaboración propia.

Optimizador (X)	Porcentaje de efectividad en la etapa de Entrenamiento (%)	Error	Porcentaje de efectividad en Validación (%)
Gradiente Descendiente Estocástico (SGD)	89.29	0.5401	60
Adadelta	100	0.0019	80
RMSprop	100	0.00085614	80
ADAM	100	0.0000088337	80

Tabla 25. Resultados de la variación del optimizador

Modelo implementado con las mejores condiciones

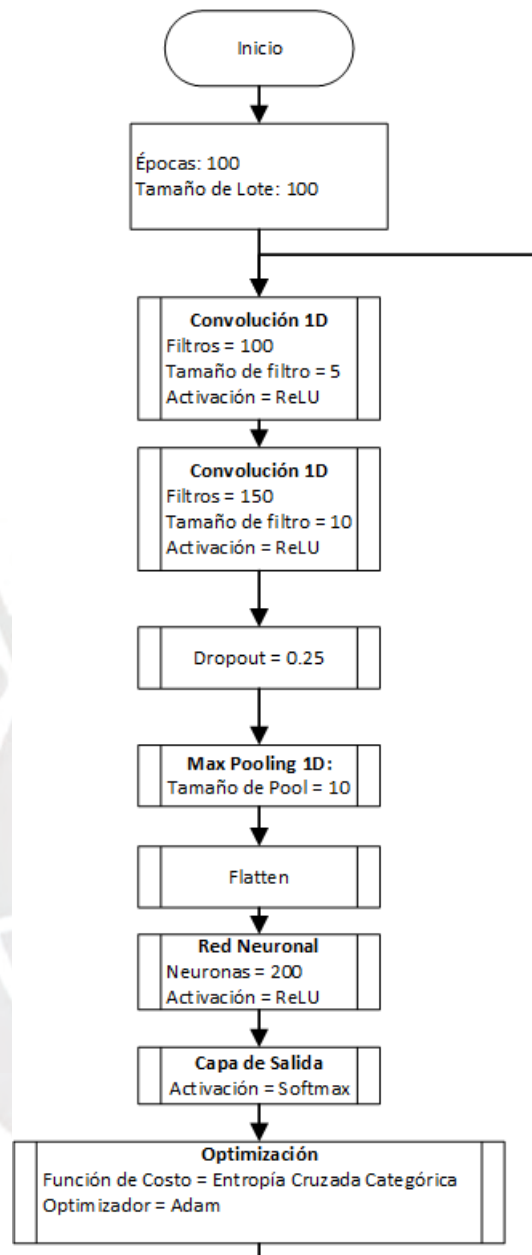


Figura 75. Modelo con los parámetros obtenidos. Elaboración propia.

Condiciones	Porcentaje de efectividad en la etapa de Entrenamiento (%)	Error	Porcentaje de efectividad en Validación (%)
Iniciales	100	0.0000024778	80
Mejoradas	100	0.00019136	90

Tabla 26. Resultados con parámetros mejorados

3.8. Conclusiones del Capítulo

- Trabajar con Aprendizaje Profundo y Redes Neuronales Convolucionales nos permitirá gestionar la gran cantidad de información disponible de manera más eficiente, reduciendo la carga computacional, almacenamiento y procesamiento de datos. Además, nos dará mayor robustez ante cambios bruscos o distorsiones para una misma salida.
- Debido a que las vibraciones son señales temporales de 1 dimensión, utilizaremos una Red Neuronal Convolucional de 1D. Se presentaron el esquema y el diagrama de flujo para su implementación.
- La implementación del software se realizará utilizando el entorno gratuito de Python en su versión 3.6. La librería de Keras contiene funciones para desarrollar y evaluar modelos de Aprendizaje Profundo. TensorFlow es una biblioteca gratuita de código abierto que complementa a Keras.
- En el Capítulo 3 se presentó la validación del modelo implementado para el diagnóstico de fallas implementado en Python utilizando la base de datos MAFAULDA. Se identificaron los parámetros importantes y se definieron los valores adecuados para la aplicación. Se obtuvo un 100% de efectividad durante el entrenamiento y un 90% de efectividad durante la validación.

Capítulo 4. Propuesta de Implementación

4.1. Introducción

El presente trabajo de Tesis busca desarrollar la etapa de diagnóstico de un robot que será capaz de recorrer una planta industrial en busca de fallas en los diversos componentes de esta. El robot deberá adquirir mediciones de vibración y ruido para diagnosticar si existe o no una falla. La capa de control que determinará la presencia de fallas se basará en la Tesis de Maestría.

4.2. Implementación del Hardware

4.2.1. Diagrama general del sistema

La Figura 76 muestra el sistema al implementar. Se utilizará un robot móvil para recorrer la planta en busca de fallas. Para adquirir las señales de vibración en los tres ejes, se utilizará un acelerómetro triaxial. Estas señales son adquiridas y acondicionadas por una tarjeta de adquisición. El procesamiento de las señales y el posterior diagnóstico de las fallas podría ser implementada en el computador NVIDIA Jetson Nano.

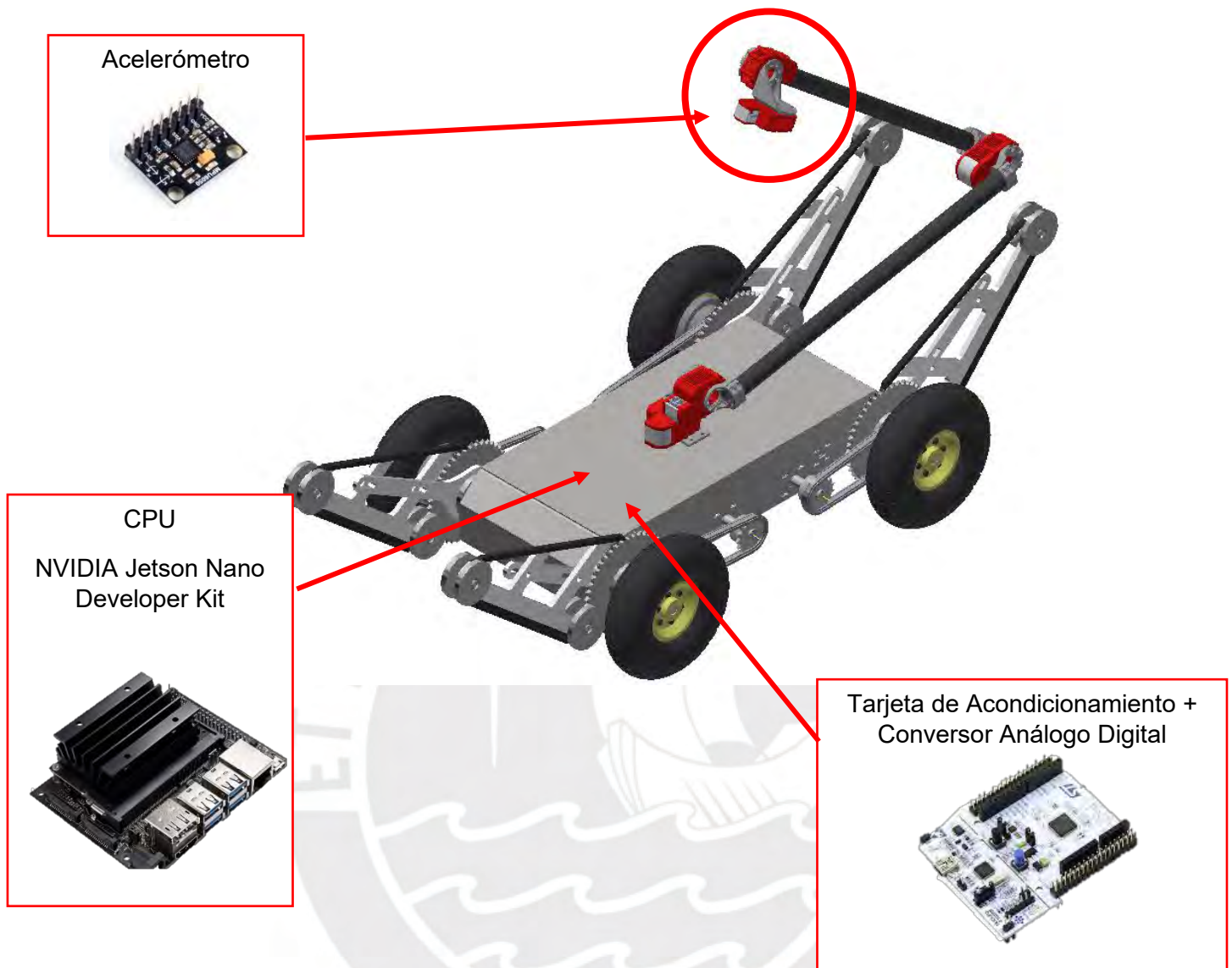


Figura 76. Robot Móvil propuesto. Elaboración propia.

A continuación, se describirán los componentes del sistema presentado.

4.2.2. Sensores

4.2.2.1. Acelerómetros

Para medir la aceleración de la vibración de las bombas y motores, se utilizarán acelerómetros. Se utilizará un acelerómetro triaxial MPU 6050, el cual tiene la gran ventaja de realizar mediciones en los ejes axial, radial y tangencial simultáneamente. La Figura 77 muestra el acelerómetro triaxial MPU 6050 de la marca OSEPP.



Figura 77. Acelerómetro Triaxial MPU 6050. Fuente: <https://www.osepp.com/electronic-modules/sensor-modules/16-accelerometer-gyro-breakout-balancing-module>

Alimentación	VDD Voltaje de alimentación 3-16 V max
Interfaz de comunicación	Interfaz Serial I2C
Señales medidas	Giroscopio de 3 ejes, Acelerómetro de 3 ejes
Convertor Análogo Digital	16-bit ADC

Tabla 27. Especificaciones del acelerómetro triaxial MPU 6050

4.2.2. CPU

4.2.2.1. NVIDIA Jetson Nano

La NVIDIA Jetson Nano es una computadora potente y compacta dedicada para aplicaciones embebidas de Inteligencia Artificial. Tiene gran potencial para aplicaciones de aprendizaje profundo, redes neuronales, visión por computadora, procesamiento de gráficos y multimedia. Este módulo de desarrollo de NVIDIA adiciona los periféricos interface con el usuario a la computadora, conectores para la comunicación y alimentación, y disipador de calor a la computadora Jetson Nano.



Figura 78. NVIDIA Jetson Nano Developer Kit. Fuente: <https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit#write>

Especificaciones Técnicas del módulo de Desarrollo NVIDIA Jetson Nano (NVIDIA, 2020)

Fabricante	NVIDIA
GPU	128-core Maxwell
CPU	Quad-Core ARM A57 @ 1.43 GHz
Memoria	4GB 64-bit LPDDR4 25.6 GB/s
Puertos USB	4x USB 3.0, USB 2.0 Micro-B
Otros	GPIO, I2C, I2S, SPI, UART
Consumo de Potencia	5 W
Precio	\$ 99.00 (NVIDIA Store)

Tabla 28. Especificaciones de la computadora NVIDIA Jetson nano

Listado de periféricos y entradas:

1. Entrada microSD para almacenamiento
2. 40 Pines de expansión
3. Puerto micro-USB para alimentación de entrada 5V o para data
4. Puerto Gigabit Ethernet
5. 4 x Puertos USB 3.0
6. Puerto de salida HDMI
7. Conector DisplayPort
8. Entrada conector Jack para alimentación de 5V
9. Conectores MIPI CSI para cámara



Figura 79. Periféricos de la Jetson Nano Developer Kit. Fuente: <https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit#write>

4.2.4. Acondicionamiento de las señales

La adquisición de las señales de vibración y ruido necesitará una etapa de acondicionamiento, como por ejemplo, la tarjeta STM32 nucleo.



Figura 80. Tarjeta de acondicionamiento STM32 Nucleo. Fuente: <https://www.st.com/en/evaluation-tools/stm32-nucleo-boards.html>

Característica	Cantidad
Procesador	Cortex-M4
Memoria Flash	128Kbytes
Memoria SRAM	32Kbytes
I/Os, periféricos	2 Buses APB, 2 buses AHB y Bus matricial multi-AHB de 32 bits
Acondicionamiento de señales	2 x ADC 12-bit, 4 comparadores, 3 amplificadores, 4 canales DAC, temporizadores de propósito general de 32 bits, 2 temporizadores PWM dedicados para control de motores, 7 contadores de propósito general de 16 bits y 1 temporizador de baja potencia de 16 bits.

Tabla 29. Especificaciones de la tarjeta de acondicionamiento de señal

4.3. Procedimiento para la configuración e instalación del Hardware y Software de la NVIDIA Jetson Nano

A continuación, se describirá el Hardware necesario y la instalación para las pruebas con la NVIDIA Jetson Nano.

4.3.1. Instalación del Hardware

La Figura 81 muestra las conexiones en la tarjeta de desarrollo NVIDIA Jetson Nano.

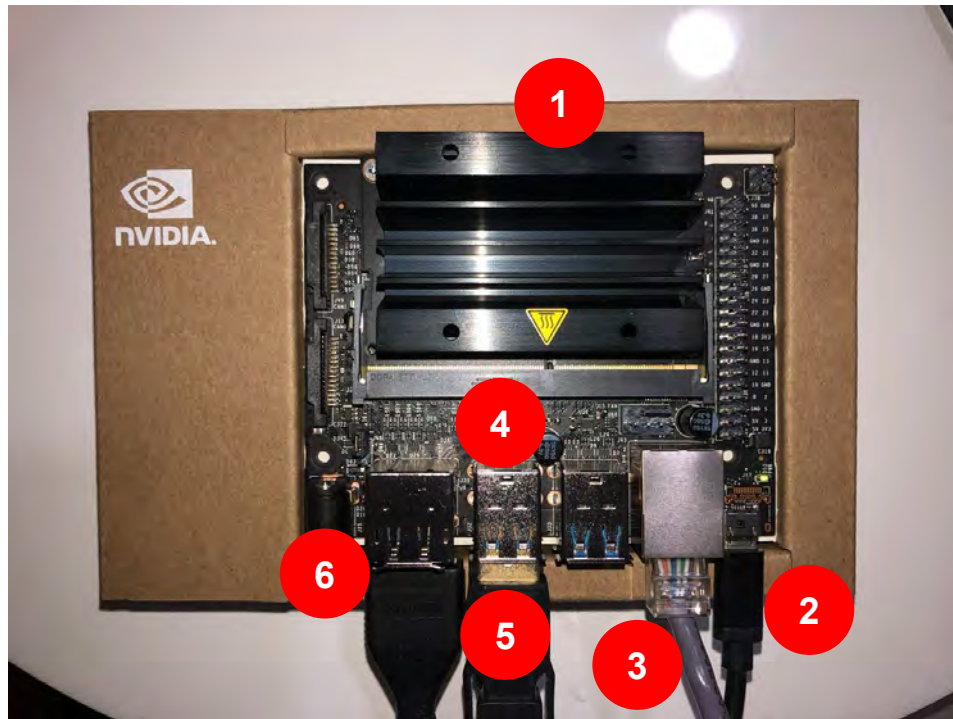


Figura 81. Instalación del Hardware en Jetson Nano Developer Kit. Elaboración propia.

Listado de equipos y periféricos conectados:

1. Entrada microSD para almacenamiento
2. Conexión de alimentación por puerto micro-USB utilizando un cargador de 5V, 2.5 A.
3. Modem para proporcionar conexión a Internet. La Jetson Nano se conecta al modem mediante cable Ethernet.
4. Mouse USB
5. Teclado USB
6. Conexión por Cable HDMI a pantalla

4.3.2. Instalación y configuración del Software

La NVIDIA Jetson Nano trabaja con sistema operativo Linux Ubuntu 18.04. Para cargar el sistema operativo, se debe hacer un flash de la imagen de este a la tarjeta micro SD, utilizando los programas: *SD Card Formatter* y *Balena Etcher*. El paquete de software se descarga desde la página web de NVIDIA (NVIDIA, 2020):

Una vez instalado el sistema operativo y encendido la computadora, como primer paso se verifica la instalación de Python 3.6. En Ubuntu 18.04, la versión 3.6.9 de Python viene preinstalada. Luego se deberán descargar las librerías de Keras y Tensorflow, además de las librerías básicas de Python para poder trabajar con Aprendizaje Profundo y Redes Neuronales. Entre las librerías básicas tenemos numpy, pandas, matplotlib, scipy, entre otras. Con las librerías instaladas y probadas, se ejecutan los algoritmos antes mencionados.



4.4. Evaluación de tiempos de procesamiento en NVIDIA Jetson Nano

Para comparar el performance de la computadora NVIDIA Jetson Nano, evaluaremos el tiempo que le tomará completar las etapas de entrenamiento y validación. Estos tiempos se compararán con la Laptop HP OMEN 2019 utilizada para la programación. El algoritmo se ejecutará 10 veces, obteniéndose el valor promedio. Se realizarán 2 pruebas con cantidades de datos diferentes. La tabla 31 muestra los resultados.

	LAPTOP HP OMEN 2019	NVIDIA JETSON NANO
Procesador	Intel Core i7-8750H CPU @ 2.20GHz	Quad-core ARM A57 @ 1.43 GHz
RAM	16 GB	4 GB 64-bit LPDDR4 25.6 GB/s
GPU	NVIDIA GEFORCE RTX	128-Core Maxwell

	Entrenamiento	Validación	Entrenamiento	Validación
Cantidad de datos	50	18	50	18
Tiempo Promedio de ejecución (s)	4.77	0.66	13.64	6.95
Cantidad de datos	70	25	70	25
Tiempo Promedio de ejecución (s)	6.44	0.67	17.04	7.01

Tabla 30. Comparación de tiempos de procesamiento

De los resultados podemos observar que el tiempo que le toma a la tarjeta NVIDIA Jetson Nano es aproximadamente el triple para la etapa de entrenamiento, mientras que en la validación es aproximadamente 10 veces. Estos resultados refuerzan la necesidad de mejorar los tiempos del algoritmo mediante optimizaciones de la programación.

4.5. Procedimiento para la configuración e instalación del Hardware y Software de la STM32 Nucleo-64 431RB y el acelerómetro triaxial MPU 6050

La tarjeta STM32 Nucleo64 G431 RB nos servirá para leer y acondicionar la señal medida por el acelerómetro. Las señales serán leídas mediante el protocolo I2C y posteriormente compartidas al controlador Jetson Nano, mediante comunicación por el puerto serial.

4.5.1. Instalación del Hardware

La Figura 82 muestra el diagrama de conexiones de la tarjeta con el acelerómetro triaxial.

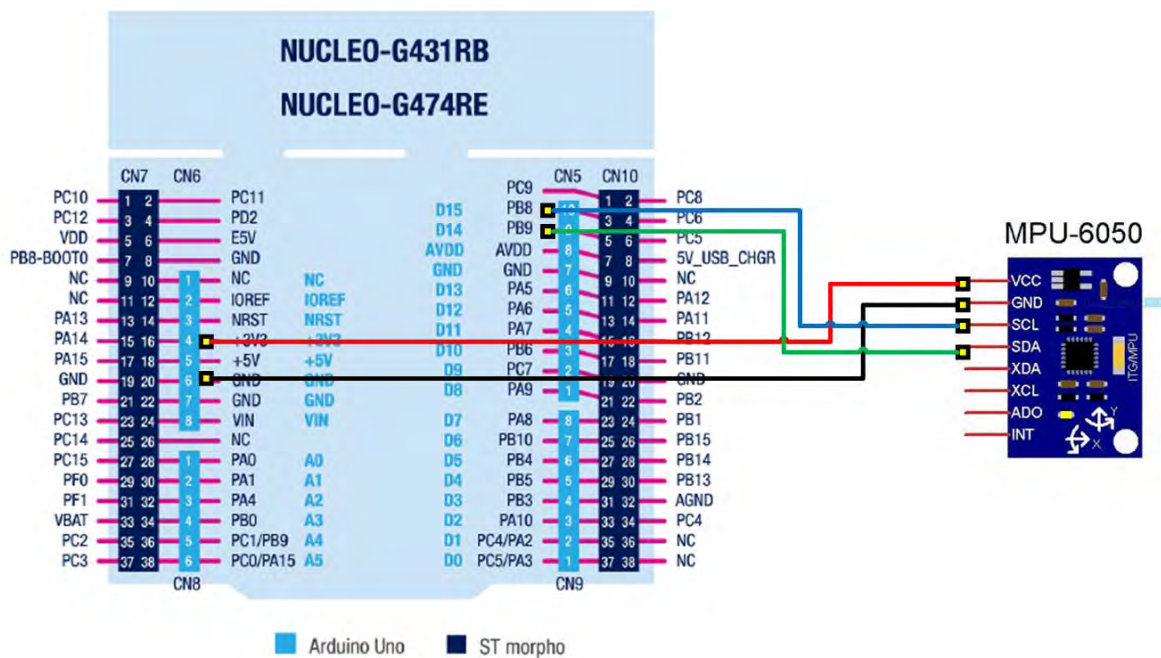


Figura 82. Diagrama de conexiones de la Tarjeta STM32 Nucleo64 G431RB y el acelerómetro triaxial MPU 6050. Elaboración propia: Fuentes: https://github.com/candle-usb/candleLight_fw/issues/21, <https://electronicalugo.com/comenzando-con-arduino-y-mpu6050/>

Finalmente, el Hardware quedaría como se muestra en la Figura 83.

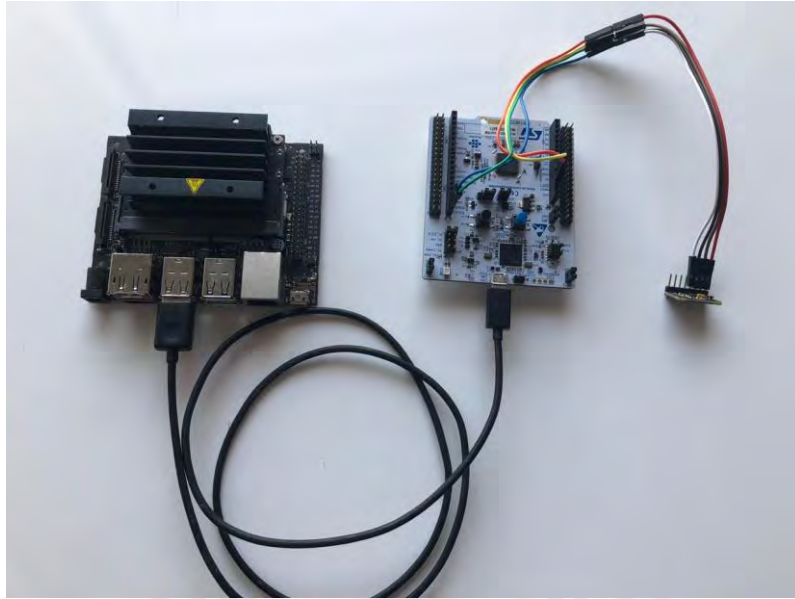


Figura 83. Instalación del Hardware. Elaboración propia.

4.5.1. Software utilizado para la tarjeta STM32 Nucleo64 G431RB

Utilizamos el software STM32 CubeMX mostrado en la figura 84, para la configuración de la tarjeta STM32. El software puede ser descargado gratuitamente de su página web.



Figura 84. Ventana de inicio del Software STM32 CubeMX. Elaboración propia.
Fuente: <https://www.st.com/en/development-tools/stm32cubemx.html>

El primer paso será elegir al tarjeta STM32 Nucleo64 G431RB como se muestra en la Figura 85.



Figura 85. Elección de la tarjeta STM32. Elaboración propia.

Luego se configurarán los puertos, la comunicación I2C y serial, el reloj y la información del proyecto. La Figura 86 muestra la configuración de la tarjeta.

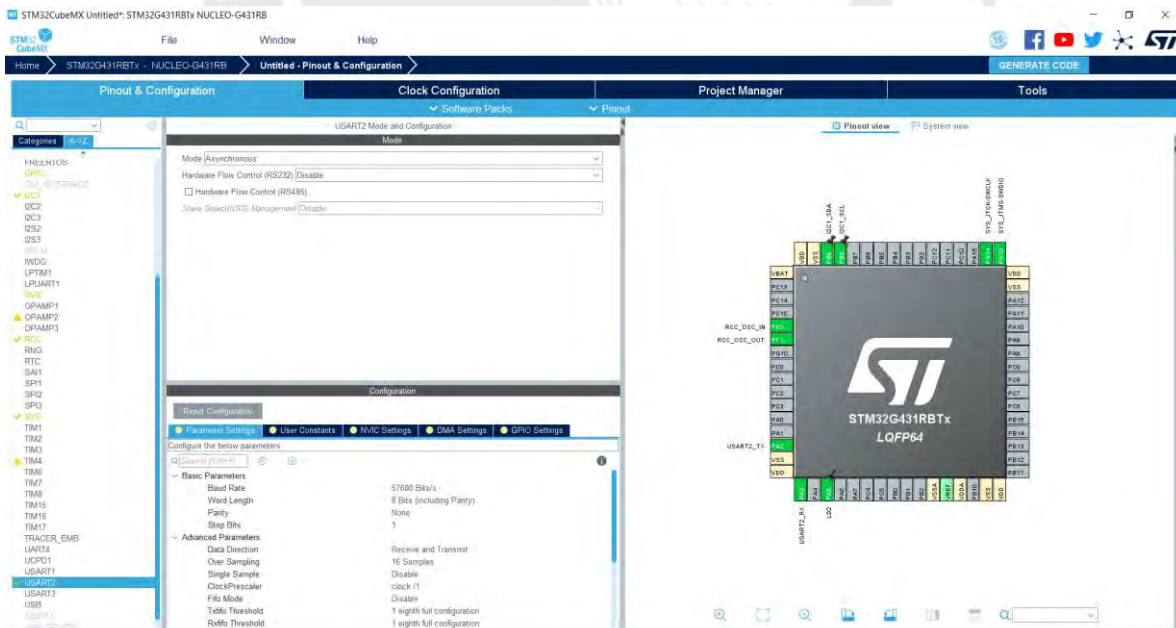


Figura 86. Configuración de puertos, comunicación, reloj y proyecto. Elaboración propia.

Luego de configurar la tarjeta STM32, utilizamos el software System Workbench for STM32 de AC6 Tools, para la programación de esta. La Figura 87 muestra el Software mencionado, el cual está disponible de forma gratuita desde la página web de STM.



Figura 87. Ventana de inicio del Software 4 STM32. Elaboración propia. Fuente: <https://www.st.com/en/development-tools/sw4stm32.html>



4.6. Procedimiento para la medición

Procedimiento de medición utilizando un robot móvil con brazo acondicionado con sensores:

4.6.1. Para la adquisición de datos en motores y bombas:

Caso A. Utilizando un acelerómetro triaxial:

- a. El robot se acercará a la bomba o al motor de interés lo suficiente para que el brazo pueda colocar los sensores con facilidad en 1 eje (axial, radial o tangencial).
- b. El brazo deberá colocar el acelerómetro triaxial suavemente y de manera **perpendicular** a la superficie en 1 eje (axial, radial o tangencial).
- c. La medición deberá durar al menos 5 segundos.
- d. La señal medida será acondicionada y filtrada.
- e. El controlador ejecutará el algoritmo de diagnóstico de fallas.
- f. De presentarse una falla, el controlador clasificará el tipo de falla.
- g. El brazo retirará el acelerómetro triaxial.
- h. El robot continúa con el siguiente equipo.

Caso B. Utilizando un acelerómetro de 1 eje:

- a. El robot se acercará a la bomba o al motor de interés lo suficiente para que el brazo pueda colocar los sensores con facilidad en los 3 ejes (axial, radial y tangencial).
- b. El brazo deberá colocar el acelerómetro suavemente y de manera **perpendicular** a la superficie en 1 eje.
- c. La medición deberá durar al menos 5 segundos.
- d. La señal medida será acondicionada y filtrada.
- e. El brazo retirará el acelerómetro.
- f. El brazo continuará la medición del siguiente eje, repitiendo los pasos b-e.
- g. Cuando se terminen de medir los 3 ejes, el controlador ejecutará el algoritmo de diagnóstico de fallas.
- h. De presentarse una falla, el controlador clasificará el tipo de falla.
- i. El robot continúa con el siguiente equipo.

Ejemplo para la medición de aceleración en los 3 ejes

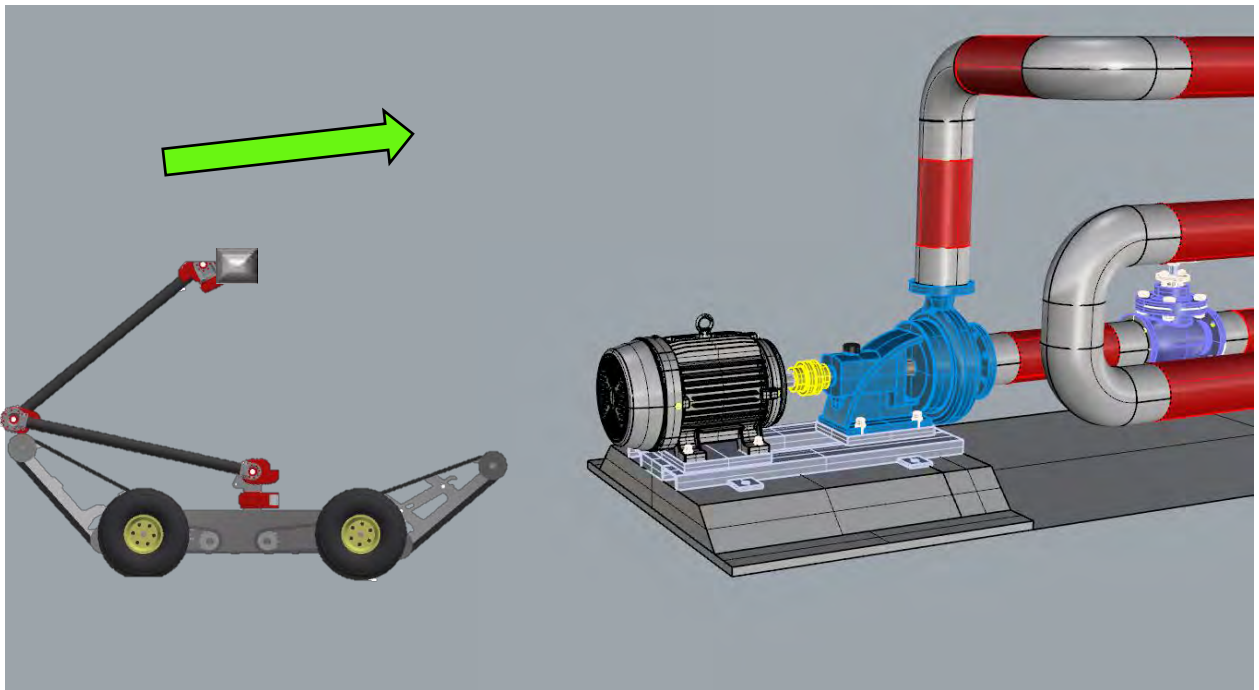


Figura 88. El robot se acerca al equipo de interés

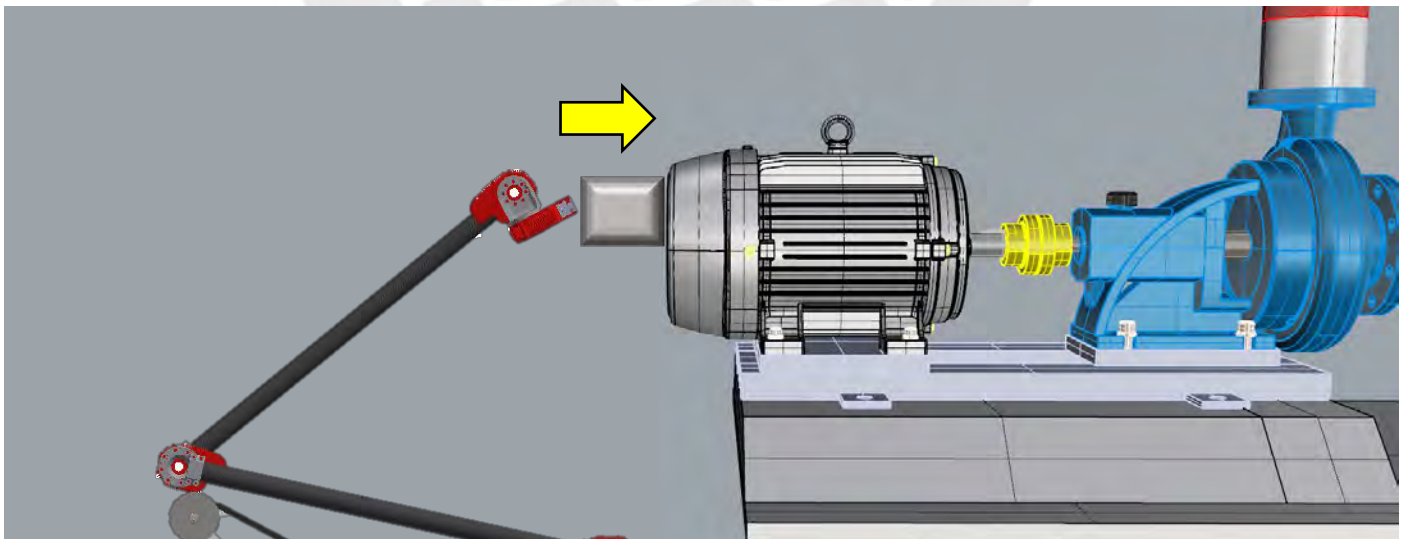


Figura 89. El brazo coloca el acelerómetro perpendicular a la superficie del eje axial. Elaboración propia.

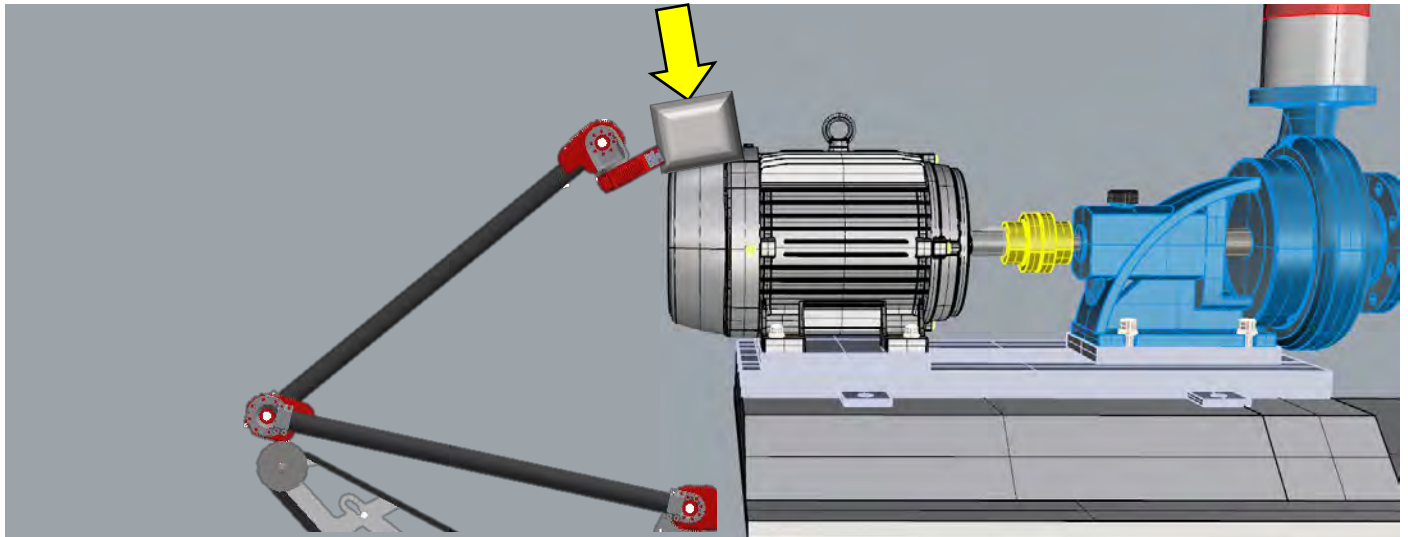


Figura 90. El brazo coloca el acelerómetro perpendicular a la superficie del eje radial. Elaboración propia.

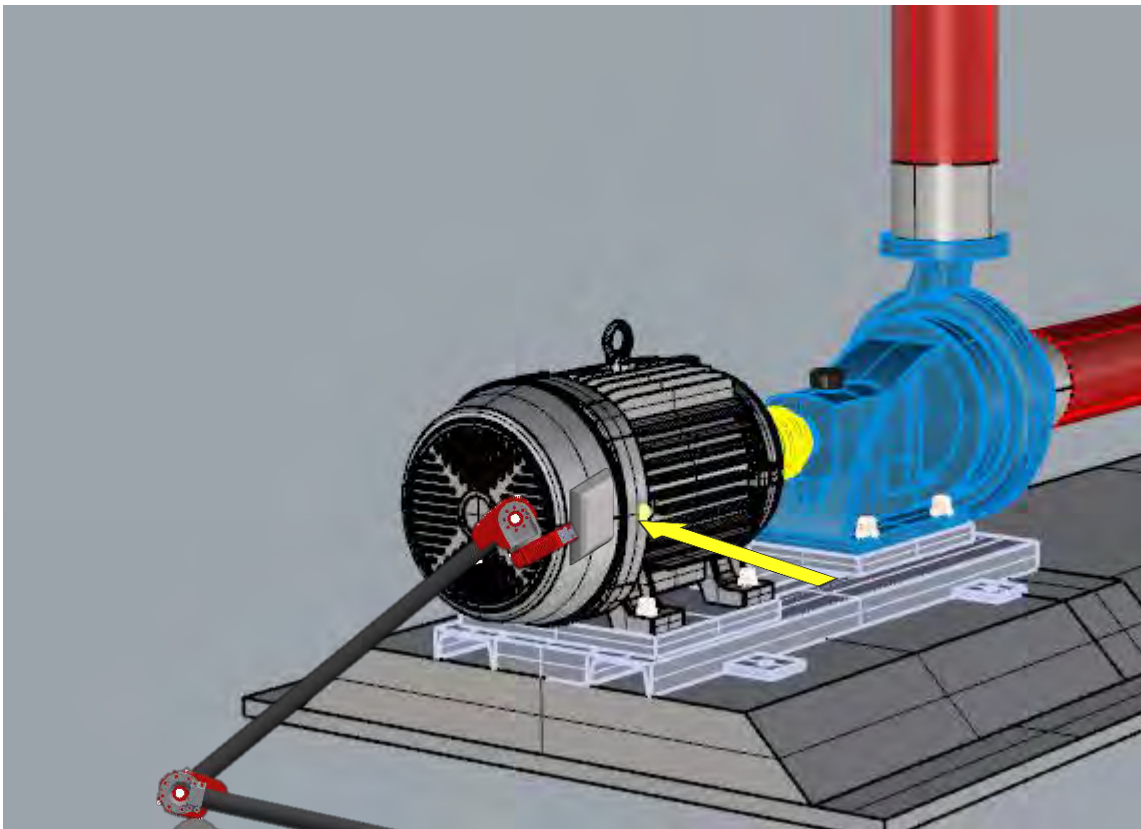


Figura 91. El brazo coloca el acelerómetro perpendicular a la superficie del eje tangencial. Elaboración propia.

4.7. Conclusiones

- En el Capítulo 4, se presentaron tablas informativas con las características del acelerómetro triaxial para la medición de la vibración, la computadora Jetson Nano y la tarjeta de acondicionamiento.
- El tiempo de procesamiento de la computadora NVIDIA Jetson Nano es aproximadamente tres veces más lento que en la laptop HP OMEN. Esto es comprensible dada la capacidad en procesadores, memoria y tarjeta gráfica.
- Se describió la instalación y configuración del Hardware y Software para la implementación del sistema de diagnóstico de fallas y se detallaron los pasos a seguir para una medición correcta.



Conclusiones

- En el Capítulo 1 se presentaron los componentes industriales, como las bombas y motores, y las metodologías, tanto tradicionales como el estado del arte, para el diagnóstico de sus fallas, resaltando el uso de metodologías de análisis vibracional. Además, se presentaron ejemplos de robots móviles los cuales están diseñados para realizar de forma segura, tareas de mantenimiento, supervisión, inspección y rescate.
- En el Capítulo 2 se justificó la elección de la Transformada de Hilbert Huang, como algoritmo base para el procesamiento de los datos de vibración, debido a la representación de datos en tiempo y frecuencia, su buena performance con sistemas lineales y no lineales, y su implementación sencilla. Los algoritmos para obtener las Funciones de Modo Intrínseco se explicaron en las Figuras 33 y 34 mediante diagramas de flujo. Finalmente, se presentaron pruebas con señales sintéticas y reales, demostrando gráficamente las diferencias entre señales a condiciones normales y con fallas.
- En el Capítulo 3 se presentó la Red Neuronal Convolutiva y sus etapas de Aprendizaje de Aspectos y Clasificación. Se demostró la importancia de la Convolución y Extracción de Aspectos Característicos para reducir los datos de entrada y a su vez transformarlos en datos más representativos. Se justificó su elección mediante la tabla 10, en donde se describieron las ventajas, desventajas y aplicaciones en comparación con las redes neuronales convencionales. Los diagramas de flujo del algoritmo de la Red Neuronal Convolutiva implementado, se mostraron en las Figuras 57 y 58. Los parámetros utilizados fueron optimizados para obtener el mejor aprendizaje.
- Para validar del modelo implementado en Python, se utilizó la base de datos de la universidad Case Western Reserve y MAFAULDA. Se identificaron los parámetros importantes y se definieron los valores adecuados para la aplicación, obteniéndose un 100% de efectividad durante el entrenamiento y un 90% de efectividad durante la validación.
- Con los resultados obtenidos en el Capítulo 3, podemos concluir que el algoritmo implementado logra clasificar las condiciones normales y de fallas por desalineamiento y rodamientos a un porcentaje del 90%. Aún bajo estas condiciones, no es capaz de clasificar el tipo de falla a un porcentaje mayor al 90%,

pero esto se lograría con una mayor cantidad de datos de entrenamiento. En la tabla 1 del Anexo 2, se muestra el resultado de la evaluación de señales normales y de fallas en rodamientos, en el cual se logra la clasificación con un porcentaje de 93.89%.

- De la tabla 30, se concluye que el tiempo de procesamiento en el controlador NVIDIA Jetson Nano no supera el tiempo de la laptop utilizada. Esto es entendible dado la capacidad computacional de esta. El código aún puede ser optimizado para reducir el tiempo.
- En el capítulo 4 se presentó el diseño del sistema de adquisición de datos vibracionales conectado al terminal de un brazo robótico. Se describió el sensor y tarjetas de adquisición y controlador utilizados. Finalmente, se mostró la implementación de este mediante diagramas de conexiones e instrucciones para una medición adecuada.



Recomendaciones

- Se recomienda entrenar al sistema con la mayor cantidad de datos posibles. Mientras más datos tengamos, mejor será la validación.
- El posicionamiento del acelerómetro es crítico para que la señal de vibración sea representativa. Esta debe seguir las recomendaciones de la normativa presentada.
- Debido al tiempo de procesamiento evaluado, se recomienda que la etapa de entrenamiento se realice en la laptop. Posteriormente el modelo entrenado se grabará en la Jetson Nano.



Bibliografía

- [1]. Isermann; R. (2006). ***Fault-Diagnosis Systems. An Introduction from Fault Detection to Fault Tolerance.*** Springer
- [2]. Isermann; R. (2011). ***Fault-Diagnosis Applications. Model-based condition monitoring: Actuators, drives, machinery, plants, sensors, and fault-tolerant systems.*** Springer
- [3]. Perez-Zuniga, C., Chantery, E., Travé-Massuyès, L.; Sotomayor, J. (2015). ***Decentralized diagnosis in a spacecraft attitude determination and control system.*** Journal of Physisc: Conf Series vol. 659(1) pp. 1-12
- [4]. Perez, C.G., Chantery, E., Travé-Massuyès, L.; Sotomayor, J. (2016). ***Fault driven minimal structurally overdetermined set in a distributed context.*** 27th International Workshop on Principles of Diagnosis DX2016
- [5]. Perez-Zuniga, C., Chantery, E., Travé-Massuyès, L.; Sotomayor, J. (2017). ***Fault-driven structural diagnosis approach in a distributed context.*** IFAC PapersOnline 50-1 14254-14259
- [6]. Perez-Zuniga, C., Chantery, E., Travé-Massuyès, L.; Sotomayor, J.; Artigues, C. (2018). ***Decentralized diagnosis via structural analysis and integer programming.*** IFAC-PapersOnline 51-24 pp. 168-175
- [7]. Perez-Zuniga, C.G., Sotomayor-Moriano, J.; Chantery, E., Travé-Massuyès, L.; Soto, M. (2019). ***Flotation Process Fault Diagnosis Via Structural Analysis.*** IFAC-PapersOnline Volume 52, Issue 14, Pages 225-230.
- [8]. Tadokoro, S. (2013). ***Priority Issues of Disaster Robotics in Japan.*** IEEE R10-HTC2013. Sandai, Japan, August 26-29, 2013. 2013 IEEE Region 10 Humanitarian Technology Conference
- [9]. Gharahbagheri, H. (2016). ***Industrial Fault Detection and Diagnosis Using Bayesian Belief Network.*** Tesis de Maestría, Facultad de Ingeniería y Ciencia Aplicada, Memorial University of Newfoundland
- [10]. Diccionario de la Real Academia Española (RAE), Ruta: <https://dle.rae.es/?id=HXktHg8|HXnzSUT|HXpaYZn>
- [11]. Adnan, N.; Ghazali, M.; Amin, M.; Hamat, A. (2015). ***Leak detection in gas pipeline by acoustic and signal processing – A review.*** 3rd International Conference of Mechanical Engineering Research (ICMER 2015)
- [12]. Scheffer, C.; Girdhar, P. (2004). ***Practical Machinery Vibration Analysis and Predictive Maintenance.*** Elsevier.

- [13]. Siddiqui, K.; Sahay. K. (2014). **Health Monitoring and Fault Diagnosis in Induction Motor – A review**. International Journal of Advanced Research in Electrical Electronics and Instrumentation Engineering. ResearchGate.
- [14]. ISO 10816-1. **Mechanical vibration – Evaluation of machine vibration by measurements on non-rotating parts – Part 1: General Guidelines**. INTERNATIONAL STANDARD ORGANIZATION.
- [15]. A-MAQ S.A., Medellín, Colombia. (2019) Link: <https://a-maq.com/category/video/>
- [16]. Lu, S.; Qian, G.; He, Q.; Liu, F.; Liu, Y; Wang, Q. (2019). **Insitu Motor Fault Diagnosis Using Enhanced Convolutional Neuronal Network in an Embedded System**. IEEE Sensors Journal
- [17]. White, G. (2010). **Introducción al Análisis de Vibraciones**. Azima DLI
- [18]. Sarmiento, S. (2020). **Análisis Vibracional**. Curso de Análisis Vibracional, TECSUP 2020
- [19]. Técnicos Asociados de Charlotte. Technical Associates of Charlotte <https://www.technicalassociates.net/>
- [20]. Reliability Maintenance Solutions <https://rms-reliability.com/training/>
- [21]. Tadokoro, S (2015). **Challenge of Disaster Robotics**. Proceedings of the 34th Chinese Control Conference. July 28-30, 2015, Hangzhou, China. IEEE Xplore
- [22]. Nagatani, K.; Kiribayashi, S. Okada, Y.; Tadokoro S.; Nishimura T.; Yoshida T.; Koyanagi, E.; Hada, Y. (2011). **Redesign of rescue mobile robot Quince**. Proceedings of the 2011 IEEE International Symposium on Safety, Security and Rescue Robots. Kyoto, Japan, November 1-5 2011
- [23]. Abhishek J.; Nagarjun C.S.; Srinivas R. (2017). **The DRASB – Disaster Response and Surveillance Bot**. 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)
- [24]. Osumi, H. (2014). **Application of Robot Technologies to the Disaster Sites**. Report of JSME Research Committee on the Great East Japan Earthquake Disaster. The Japan Society of Mechanical Engineers.
- [25]. Department of Robotics and Mechatronics, Faculty of Mechanical Engineering and Robotics, AGH University of Science and Technology in Krakow. **Tracked mobile robot for pipeline inspection with an active**

- adaptation system.** (2017). Link: <https://www.youtube.com/watch?v=NKH9povlrsw>
- [26]. Boston Dynamics (2020). Link: <https://www.bostondynamics.com/>
- [27]. Energy Robotics (2020). Link: <https://www.energy-robotics.com/>
- [28]. Sprint Robotics (2020). Link: <https://www.sprintrobotics.org/overview/>
- [29]. Innovate Perú. Fondo para la Ciencia y Tecnología. Ministerio de la Producción. 15ª Convocatoria del Concurso PIMEN. Link: https://www.innovateperu.gob.pe/fincyt/doc/pimen/15/resultados/RESULTADOS_FINAL_Reconsideraci%C3%B3n_15%C2%B0PIMEN_2810191353.pdf
- [30]. Innóvate Perú. Fondo para la Ciencia y Tecnología. Ministerio de la Producción. 9ª Convocatoria del Concurso PIMEN. Link: https://innovateperu.gob.pe/fincyt/doc/pimen/09/resultados/Proyectos_Resultados_Finales_9PIMEN.pdf
- [31]. Mrugalski, M. (2014). **Advanced Neural Network-Based Computational Schemes for Robust Fault Diagnosis.** Studies in Computational Intelligence book series (SCI, volume 510)
- [32]. Stiller, M.; Wagner, J.; Thyroff, D.; Hahn, I. (2017). **Comparison of different noise analysis methods for error detection on induction machines.** 2017 IEEE 11th International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives (SDEMPED)
- [33]. ISO 18436-2. **Condition monitoring and diagnostics of machines – Requirements for training and certification of personnel – Part 2: Vibration condition monitoring and diagnostics.** INTERNATIONAL STANDARD ORGANIZATION.
- [34]. Revista Forbes. (2018). Link: <http://forbes.es/business/42128/inteligencia-artificial-en-que-sectores-tiene-mas-importancia/>
- [35]. Norvig, P.; Russell, S. (2004). **Inteligencia Artificial Un Enfoque Moderno.** 2da. Edición. Pearson Prentice Hall.
- [36]. Ramírez, J.; Sarmiento, H.; López-Lezama, J. (2018) **Diagnóstico de fallas en procesos industriales mediante inteligencia artificial.** Revista Espacios. Vol.39, Año 2018 Página 12.
- [37]. Wang, H.; Ji, Y. (2018). **A Revised Hilbert-Huang Transform and Its Application to Fault Diagnosis in a Rotor System.** Artículo de la revista sensors. MDPI.

- [38]. Huang, E.; Wu, Z. (2007). **A Review on HHT and its application to geophysical studies**. American Geophysical Union.
- [39]. Matworks (2016). Link:
<https://la.mathworks.com/matlabcentral/fileexchange/19681-hilbert-huang-transform>
- [40]. Case Western Reserve University. Base de datos de mediciones de motor (2019). Link:
<https://csegroups.case.edu/bearingdatacenter/pages/download-data-file>
- [41]. Base de datos de vibraciones en máquinas MAFAULDA (Machinery Fault Database)
http://www02.smt.ufrj.br/~offshore/mfs/page_01.html
- [42]. Murphy, K. (2012). **Machine Learning A Probabilistic Perspective**, Massachusetts Institute of Technology
- [43]. Goodfellow, I.; Bengio, Y.; Courville, A. (2016). **Deep Learning**, Massachusetts Institute of Technology
- [44]. Saha, S. (2018) **A Comprehensive Guide to Convolutional Neural Networks** – the ELI5 way. <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- [45]. Rawar, W.; Wang, Z. (2017) **Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review**. ResearchGate, Neural Computation June 2017
- [46]. LeCun, Y.; Bengio, Y. (2007) **Convolutional Networks for Images, Speech, and Time-Series**
- [47]. Ince, T.; Kiranyaz, S.; Eren, L.; Askar, M.; Gabbouj, M. (2016) **Real-Time Motor Fault Detection by 1-D Convolutional Neural Networks**. IEEE Transactions on Industrial Electronics, Vol 63, No. 11, Noviembre 2016
- [48]. Hinton, G.; Srivastava, N.; Krizhevsky, A., Sutskever, I.; Salakhutdinov, R. (2012). **Improving neural networks by preventing co-adaptation on feature detectors**
- [49]. O’Shea, K.; Nash, R. (2015). **An Introduction to Convolutional Neural Networks**. ResearchGate. Department
- [50]. Google Developers <https://developers.google.com/machine-learning/crash-course/>

- [51]. Keras <https://keras.io/>
- [52]. TensorFlow <https://www.tensorflow.org/>
- [53]. Brownlee, J. (2020). Machine Learning Mastery – Making Developers Awesome at Machine Learning. <https://machinelearningmastery.com/>
- [54]. Ruder, S. (2016). **An overview of gradient descent optimization algorithms**. Insight Centre for Data Analytics, NUI Galway
- [55]. Kingma, D.; Lei Ba, J. (2015). **ADAM: A method for stochastic optimization**. ICLR 2015
- [56]. Zihlmann, M.; Perekrestenko, D.; Tschannen, M. (2017) **Convolutional Recurrent Neural Networks for Electrocardiogram Classification**. ETH Zurich, Communication Technology Laboratory.
- [57]. Python <https://www.python.org/>
- [58]. NVIDIA Developer webpage
<https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit#write>