



UNIVERSIDAD DE GUANAJUATO

CAMPUS IRAPUATO - SALAMANCA
DIVISIÓN DE INGENIERÍAS

*“Análisis de funciones modales usando entropía para
la detección de fallas en motores de inducción”*

TESIS

QUE PARA OBTENER EL GRADO DE:

MAESTRA EN INGENIERÍA ELÉCTRICA

(Opción: Instrumentación y Sistemas Digitales)

PRESENTA:

Ing. Yulimar Lorena Roa Terán

DIRECTOR:

Dr. David Camarena Martínez

Salamanca, Gto., a 13 de Noviembre del 2017.

M. EN I. J. ANTONIO ALVAREZ JAIME
COORDINADOR DE ASUNTOS ESCOLARES
PRESENTE.-

Por medio de la presente, se otorga autorización para proceder a los trámites de impresión, empastado de tesis y titulación al alumno (a) Yulimar Lorena Iba Terán del Programa de Maestría en Ingeniería Eléctrica y cuyo número de NUA es: 145572 del cual soy director. El título de la tesis es: Análisis de funciones modales usando entropía para la detección de fallos en motores de inducción.

Hago constar que he revisado dicho trabajo y he tenido comunicación con los dos sinodales asignados para la revisión de la tesis, por lo que no hay impedimento alguno para fijar la fecha de examen de titulación.

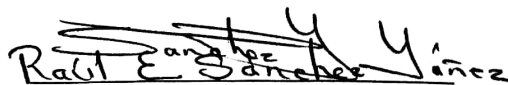
ATENTAMENTE


David Camarena Martinez

NOMBRE Y FIRMA
DIRECTOR DE TESIS
SECRETARIO


David Camarena Martinez

NOMBRE Y FIRMA
DIRECTOR DE TESIS


Raúl E. Sánchez Yáñez

NOMBRE Y FIRMA
PRESIDENTE


Xiomara González Ramirez

NOMBRE Y FIRMA
VOCAL

Dedicatoria

A Dios Todopoderoso por guiar todos mis pasos.

A mis padres, hermanos y sobrina, apoyo y fuente de inspiración en todos mis logros.

Yulimar Lorena Roa Terán

Agradecimientos

Al M.I. Jesús Joaquín Yáñez Borjas, por confiar en mi y recomendarme para realizar los estudios de maestría en la Universidad de Guanajuato, DICIS.

A mi asesor de tesis, Dr. David Camarena Martínez, por su apoyo y orientación en el desarrollo de este trabajo de tesis.

Al Dr. Juan Gabriel Aviña Cervantes, por brindarme la oportunidad de realizar estudios de maestría en la Universidad de Guanajuato, DICIS.

A mis compañeros de maestría, por el apoyo y ayuda brindada en los momentos requeridos. En especial a Lubin y a Ricardo por las orientaciones brindadas para el desarrollo de este trabajo.

A Moisés, por su apoyo y compañía.

Yulimar Lorena Roa Terán

Agradecimientos Institucionales

Expreso mi más sincera gratitud a la Universidad de Guanajuato, especialmente a la División de Ingenierías del Campus Irapuato - Salamanca por la formación y el apoyo que he recibido. A todos los profesores, mis agradecimientos por su arduo trabajo durante estos dos años que me permitieron obtener el grado de Maestra en Ingeniería.



Este trabajo fue realizado gracias al apoyo recibido del Consejo Nacional de Ciencia y Tecnología de México, CONACYT, con la beca otorgada en la convocatoria titulada "BECAS NACIONALES 2015 TERCER PERIODO", bajo el número de becario 718689.



También, agradezco al laboratorio de procesamiento digital de señales (*digital signal processing*, DSP) por brindarme un espacio de trabajo cómodo y agradable.



Signal Processing Group
Electronics Engineering Department

Resumen

Los motores de inducción son partes esenciales de la industria. La fiabilidad y disponibilidad de estas máquinas eléctricas se pueden mejorar significativamente mediante la instalación de sistemas de diagnóstico que supervisen la condición de trabajo en dichos dispositivos. Debido a esto, la detección temprana de fallas de motores de inducción ha sido una tarea de constante investigación en el área de procesamiento digital de señales. Por lo general, el diagnóstico se realiza analizando señales de vibración del motor en estado estacionario o durante el transitorio de arranque. Esta investigación propone una metodología basada en el método de descomposición de señales llamado descomposición empírica de modos por conjuntos completos (*complete ensemble empirical mode decomposition*, CEEMD) y entropía de información para la detección de fallas en motores de inducción a través del análisis de señales de vibración. La metodología propuesta consta de varios pasos. En primer lugar, se utiliza la CEEMD para descomponer las señales en varias funciones de modo intrínseco. Posteriormente, se realiza un proceso de extracción de características, a través de parámetros estadísticos y de complejidad. Finalmente, se entrenan tres algoritmos de clasificación para determinar cuál de ellos es mejor para identificar la condición de operación del motor de inducción. Esta metodología se desarrolló utilizando un software matemático y se validó mediante el uso de señales de vibración reales. Los resultados del uso de la metodología propuesta muestran que puede utilizarse para detectar fallas de barra rota, defecto en rodamientos y desbalance de carga.

Palabras clave: motores de inducción, detección de fallas, CEEMD, extracción de características, entropía de información, SVM, *AdaBoost*.

Abstract

Induction motors are essential parts of the industry. The reliability and availability of these electrical machines can be significantly improved by the installation of diagnosis systems to monitoring the work condition in these devices. Due to this, early detection of induction motors faults has been under constant research in the digital signal processing field. Usually, the diagnosis is performed by analyzing vibration signals at a steady-state motor operation or during a start-up transient. This research proposes a methodology based on the complete ensemble empirical mode decomposition (CEEMD) method and information entropy for fault detection in induction motors through the analysis of vibration signals. The proposed methodology consists of several steps. First, the CEEMD is used for decomposing the signals into several intrinsic mode functions. Subsequently, a feature extraction process is performed through statistical and complexity parameters. Finally, three classification algorithms are trained to determine which of them is better to identify the induction motor work condition. This methodology was developed by using a mathematical software and it was validated by using real vibration signals. The results of using the proposed methodology show that it can be used to detect broken rotor bar, bearing defects and unbalanced load.

Index terms: induction motors, fault detection, CEEMD, feature extraction, information entropy, SVM, AdaBoost.

Índice general

1. Introducción	1
1.1. Objetivos	4
1.1.1. Objetivo general	4
1.1.2. Objetivos específicos	4
1.2. Justificación	5
1.3. Estructura del trabajo	6
2. Generalidades	7
2.1. Motores de inducción	7
2.2. Señales de vibración en los motores de inducción	10
2.3. Fallas en motores de inducción	11
2.3.1. Barra rota	11
2.3.2. Desbalance de carga	13
2.3.3. Defecto en rodamientos	15
3. Metodología de análisis de funciones modales	19
3.1. Descomposición de modo empírico por conjuntos completos	20
3.2. Extracción de características	25
3.2.1. Entropía de información	26

<i>ÍNDICE GENERAL</i>	IX
3.3. Clasificadores	27
3.3.1. Máquinas de soporte vectorial	28
3.3.2. <i>AdaBoost</i> (<i>adaptive boosting</i>)	42
3.3.3. Árboles de decisión	45
4. Descripción del procedimiento propuesto	51
4.1. Sistema de adquisición de datos	52
4.2. Descomposición de las señales de vibración	55
4.3. Extracción de características	56
4.4. Clasificación e identificación de fallas	58
4.4.1. Máquinas de soporte vectorial (<i>SVMs</i>)	58
4.4.2. <i>AdaBoost</i>	59
4.4.3. Árboles de decisión	60
5. Resultados	61
5.1. Descomposición de las señales de vibración	62
5.2. Extracción de características - Resultados	64
5.2.1. Extracción de parámetro de complejidad. Entropía.	68
5.2.2. Extracción de parámetro estadístico de dispersión. Desviación estándar	78
5.3. Clasificación e identificación de fallas - Resultados	86
5.3.1. Fase de entrenamiento - Resultados	86
5.3.2. Fase de pruebas - Resultados	91
6. Conclusiones	99
Bibliografía	101

Índice de figuras

2.1. Estator y rotor de un motor de inducción [35].	8
2.2. Campo magnético giratorio que se produce en el estator [35].	8
2.3. Corriente inducida y fuerza de Lorentz en el rotor [35].	9
2.4. Partes de un motor de inducción [39].	9
2.5. a) Dibujo de la jaula de ardilla de un rotor [44]. b) Rotor de jaula de ardilla típico [45].	12
2.6. Espectro de las señales de vibración de un motor con falla de barra rota y condición sano [46].	13
2.7. Espectro de las señales de vibración de un motor con falla por desbalance de carga. Condición de operación: carga completa. Vibración axial [49].	14
2.8. a) Rodamiento típico [51]. b) Partes de un rodamiento [36].	15
2.9. Componentes de un rodamiento con sus dimensiones correspondientes [55].	17
3.1. Procedimiento para el análisis de las señales de vibración.	19
3.2. EMD aplicada a una señal compuesta por dos ondas seno: a) la señal original, b) envolventes superior e inferior (rojo) y la media (azul punteado), c) la primera descomposición y d) el primer residuo [60]. . .	22
3.3. Ejemplo de clasificación binaria.	29
3.4. a) Espacio de características con dos clases. b) Múltiples líneas de separación entre muestras de dos clases diferentes.	30
3.5. Espacio de características con línea recta de separación entre clases. . .	30

3.6. Vectores de soporte de cada clase.	31
3.7. Margen de separación entre clases.	31
3.8. Modificación de los vectores de soporte.	32
3.9. Modificación de las muestras que no son vectores de soporte.	33
3.10. Conjuntos no linealmente separables.	33
3.11. Muestras erróneamente clasificadas.	34
3.12. Transformación del espacio de características.	35
3.13. Función de clasificación.	36
3.14. Condiciones de clasificación.	37
3.15. Árbol de decisión sencillo que facilita la tarea de clasificación para asignación de título al nombre [83].	47
4.1. Procedimiento propuesto.	52
4.2. Fallas inducidas artificialmente a) Barra rota (BR), b) Defecto en rodamientos (DR), c) Desbalance de carga (DC).	54
4.3. Fallas múltiples y combinadas.	54
4.4. Esquema general del proceso de descomposición de las señales de vibración para cada condición.	56
4.5. Esquema general del proceso de extracción de características.	57
5.1. Señal de vibración del motor con condición <i>BR_DC_DR</i>	62
5.2. Descomposición de la señal de vibración del motor con condición <i>BR_DC_DR</i>	63
5.3. Representación de los valores del parámetro media correspondientes a todos los ejes en la IMF 5 de las pruebas realizadas.	65
5.4. Representación de los valores del parámetro coeficiente de asimetría correspondientes a todos los ejes en la IMF 5 de las pruebas realizadas.	66

5.5. Representación de los valores del parámetro coeficiente de curtosis correspondientes a todos los ejes en la IMF 5 de las pruebas realizadas.	67
5.6. Procedimiento para la extracción del valor de entropía de Shannon a cada modo o IMF obtenido en la descomposición.	68
5.7. Representación de los valores de la característica de entropía correspondientes a todos los ejes en la IMF 1 de las pruebas realizadas.	70
5.8. Representación de los valores de la característica de entropía correspondientes al eje x en la IMF 2 de las pruebas realizadas.	71
5.9. Representación de los valores de la característica de entropía correspondientes a los ejes y y z en la IMF 2 de las pruebas realizadas.	72
5.10. Representación de los valores de la característica de entropía correspondientes al eje z en la IMF 4 de las pruebas realizadas.	73
5.11. Representación de los valores de la característica de entropía correspondientes a los ejes y y z en la IMF 5 de las pruebas realizadas.	74
5.12. Representación de los valores de la característica de entropía correspondientes a los ejes x y y en la IMF 6 de las pruebas realizadas.	75
5.13. Representación de los valores de la característica de entropía correspondientes al eje z en la IMF 6 de las pruebas realizadas.	76
5.14. Representación de los valores de la característica de entropía correspondientes a todos los ejes en la IMF 7 de las pruebas realizadas.	77
5.15. Representación de los valores de la característica de desviación estándar correspondientes al eje x en la IMF 1 de las pruebas realizadas.	79
5.16. Representación de los valores de la característica de desviación estándar correspondientes a los ejes y y z en la IMF 1 de las pruebas realizadas.	80
5.17. Representación de los valores de la característica de desviación estándar correspondientes a los ejes y y z en la IMF 4 de las pruebas realizadas.	81
5.18. Representación de los valores de la característica de desviación estándar correspondientes a los ejes y y z en la IMF 5 de las pruebas realizadas.	82
5.19. Representación de los valores de la característica de desviación estándar correspondientes al eje x en la IMF 6 de las pruebas realizadas.	83

5.20. Representación de los valores de la característica de desviación estándar correspondientes al eje y en la IMF 6 de las pruebas realizadas.	84
5.21. Representación de los valores de la característica de desviación estándar correspondientes al eje y en la IMF 7 de las pruebas realizadas.	85
5.22. Comportamiento de la pérdida por resubstitución de AdaBoost.	90
5.23. Matriz de confusión para SVM con 20 valores sintéticos.	93
5.24. Matriz de confusión para SVM con 50 valores sintéticos.	93
5.25. Matriz de confusión para SVM con 100 valores sintéticos.	94
5.26. Matriz de confusión para AdaBoost con 20 valores sintéticos.	95
5.27. Matriz de confusión para AdaBoost con 50 valores sintéticos.	95
5.28. Matriz de confusión para AdaBoost con 100 valores sintéticos.	96
5.29. Matriz de confusión para árboles de decisión con 20 valores sintéticos. .	97
5.30. Matriz de confusión para árboles de decisión con 50 valores sintéticos. .	97
5.31. Matriz de confusión para árboles de decisión con 100 valores sintéticos.	98

Índice de tablas

2.1. Características de las señales de vibración en motores con fallas en los rodamientos	18
3.1. Tipos de funciones Kernel	42
4.1. Señales obtenidas en la adquisición.	53
5.1. Media y desviación estándar de la característica de entropía en la IMF 1 de las pruebas realizadas.	69
5.2. Media y desviación estándar de la característica de entropía en la IMF 2 de las pruebas realizadas.	70
5.3. Media y desviación estándar de la característica de entropía en la IMF 2 de las pruebas realizadas.	71
5.4. Media y desviación estándar de la característica de entropía del eje z en la IMF 4 de las pruebas realizadas.	72
5.5. Media y desviación estándar de la característica de entropía de los ejes y y z en la IMF 5 de las pruebas realizadas.	73
5.6. Media y desviación estándar de la característica de entropía en la IMF 6 de las pruebas realizadas.	74
5.7. Media y desviación estándar de la característica de entropía en la IMF 6 de las pruebas realizadas.	75
5.8. Media y desviación estándar de la característica de entropía en la IMF 7 de las pruebas realizadas.	76

5.9. Media y desviación estándar de la característica de desviación estándar en la IMF 1 de las pruebas realizadas.	78
5.10. Media y desviación estándar de la característica de desviación estándar en la IMF 1 de las pruebas realizadas.	79
5.11. Media y desviación estándar de la característica de desviación estándar de los ejes y y z en la IMF 4 de las pruebas realizadas.	80
5.12. Media y desviación estándar de la característica de desviación estándar de los ejes y y z en la IMF 4 de las pruebas realizadas.	81
5.13. Media y desviación estándar de la característica de desviación estándar de los ejes y y z en la IMF 5 de las pruebas realizadas.	82
5.14. Media y desviación estándar de la característica de desviación estándar de los ejes x y y en la IMF 6 de las pruebas realizadas.	83
5.15. Media y desviación estándar de la característica de desviación estándar del eje y en la IMF 7 de las pruebas realizadas.	84

Capítulo 1

Introducción

Los procesos en la industria demandan maquinaria con alto desempeño, con cualidades que garanticen un mejor aprovechamiento de tiempo y recursos, además, que no genere altos costos por daños en los equipos. En este sentido, en los últimos años, ha surgido una creciente demanda de motores de inducción ya que; debido a sus características de robustez, bajo costo, fácil mantenimiento y que la fuente de alimentación que requieren es accesible; se han convertido en elementos clave de la industria, siendo el más usado el del tipo jaula de ardilla. Sin embargo, como cualquier dispositivo, los motores de inducción presentan limitaciones que, si se exceden, podrían generar fallas en sus componentes principales, el rotor y el estator [1].

Las fallas en motores de inducción deben ser corregidas inmediatamente al ser detectadas puesto que un motor defectuoso representa un riesgo por las consecuencias que generaría, como una interrupción de los procesos industriales, una baja en la producción y, en casos extremos, daños a los componentes que estén alrededor o, incluso, poner en riesgo la vida de los operarios. Un procedimiento continuo de inspección de la condición de trabajo de los motores puede detectar fallas eléctricas y mecánicas a tiempo, con lo cual se evitarían altos costos de reparaciones y paradas no programadas [2]. En consecuencia, la tendencia de las investigaciones en el área ha sido desarrollar metodologías y procedimientos para proveer el mantenimiento adecuado a los motores de inducción y otorgarles estabilidad en el funcionamiento.

Haciendo uso de señales de voltaje, corriente y vibración obtenidas a partir de motores de inducción, diferentes metodologías han sido propuestas para la detección de fallas, en las cuales fueron aplicadas diversas técnicas de análisis en frecuencia y en tiempo-frecuencia. Por ejemplo, en [2], fue propuesto un procedimiento de detección de fallas múltiples en motores de inducción mediante la medición de señales de corriente y de vibración obtenidas durante la etapa del transitorio y estado estacionario de arranque de la máquina. Para el análisis de estas señales, fueron utilizadas las técnicas

de la transformada rápida de Fourier (*fast Fourier transform*, FFT) y la transformada Wavelet discreta (*discrete Wavelet transform*, DWT). Por otro lado, en [3], fueron empleadas las señales de corriente del estator para detectar falla de barra rota en un motor jaula de ardilla. En este caso, el procedimiento consistió en la aplicación de DWT a las señales de corriente para calcular la energía asociada a la falla del rotor en el ancho de banda de la frecuencia donde se localiza la falla.

No obstante, las técnicas tradicionales de procesamiento de señales, que incluyen análisis en el dominio de tiempo y en el dominio de la frecuencia, se basan en que el proceso que genera las señales es estacionario y lineal, lo cual no ocurre en la mayoría de los casos, ya que las fallas en motores son de carácter no estacionario y generan eventos transitorios [4]. Además, técnicas como la FFT poseen resolución de frecuencia limitada, fuga espectral y baja tolerancia al ruido [5]. Considerando las desventajas de las técnicas previamente mencionadas, se han hecho avances referentes a los métodos de análisis tiempo-frecuencia y se han aplicado al diagnóstico de las fallas en motores de inducción. Entre estos avances se encuentra la transformada Hilbert Huang que usa la descomposición de modo empírico (*empirical mode decomposition*, EMD), propuesta en 1998 por Huang [6], que corresponde a un método adaptativo de análisis en tiempo, adecuado a señales no lineales y no estacionarias [7].

EMD descompone la señal complicada (señal no estacionaria) en un conjunto de componentes completos y casi ortogonales (señales estacionarias, aproximadamente) llamados funciones de modo intrínseco (*intrinsic mode functions*, IMFs) [8]. Estas IMFs representan el modo oscilatorio natural embebido en la señal y sirven como las funciones de base, que son determinadas por la propia señal que fue descompuesta [4]. Desde que la EMD fue propuesta en 1998, ha sido estudiada y aplicada en diversas áreas como el control de procesos [9], medicina y biología [10], reconocimiento de voz, identificación del sistema, etc. A pesar de las ventajas que posee y el aporte en numerosas investigaciones, la EMD presenta limitaciones para descomponer señales en las cuales los rangos de amplitud-frecuencia se encuentran muy cerca unos de otros [11]. La mezcla de modos es el nombre que recibe el principal problema que presenta la EMD, y es un fenómeno causado por la aparición de diferentes escalas de tiempo que se encuentran mezcladas en un único componente IMF, lo que introducirá variaciones adicionales, y ficticias, en la IMF resultante y en los valores de frecuencia instantánea [12].

Para corregir el problema de la mezcla de modos, Wu y Huang [13] plantearon un nuevo método basado en EMD, llamado descomposición de modo empírico por conjuntos (*ensemble empirical mode decomposition*, EEMD). En este caso, con EEMD se añade ruido blanco finito a la señal original. La finalidad del ruido añadido es proveer un marco de referencia uniforme en el espacio tiempo-frecuencia para que la porción de la señal que se obtiene en la descomposición coincida con escalas aptas de una IMF [13]. Sin embargo, esa porción de ruido que se le añade a la señal ocasiona que la señal

reconstruida desde los IMFs contenga ruido residual, y diversas realizaciones de la señal con ruido pueden producir diferente cantidad de modos [14].

Con la finalidad de superar los efectos del ruido que presenta EEMD, fue propuesto por Torres *et al.* [14] un algoritmo denominado descomposición de modo empírico por conjuntos completos (*complete ensemble empirical mode decomposition*, CEEMD). Este nuevo método consiste en añadir ruido pero en cada etapa del proceso, y la descomposición final toma en cuenta un único residuo para obtener cada modo [5]. Las ventajas de este método, comparado con EMD y EEMD, es que se logra una completa reconstrucción de la señal original, se producen menos modos que en EEMD y la cantidad de iteraciones es la mitad comparado con EEMD, lo que representa un menor gasto de procedimientos de cómputo [14]. Por lo tanto, esta técnica es adecuada para el procesamiento de las señales en metodologías de detección de fallas en motores. CEEMD ha sido aplicado en diversas investigaciones, como en [15] donde emplean las señales de corriente del estator de un motor de inducción, para el estudio y evaluación de la condición de trabajo de la máquina con la finalidad de detectar falla de barras rotas. En [5] también aplican CEEMD para el desarrollo de una metodología de detección de fallas diversas en motores que funcionan en estado estacionario, para ello emplean las señales de vibración y los sonidos acústicos provenientes del motor. En otras áreas como economía [16] y estudios sísmicos [17] también se ha usado CEEMD.

El procesamiento que se realiza con CEEMD no extrae directamente información relevante de las señales. Por tanto, es necesario extraer características para crear una base de datos de la información contenida en las señales, con la cual preparar los datos de entrada que se requieren para la correcta clasificación de las fallas. Según la definición de [18], el proceso mediante el cual se representan las medidas originales de una señal en atributos más eficaces, se denomina extracción de características. Normalmente, las características que se obtienen son valores numéricos utilizables para el aprendizaje automático, mediante los cuales se reduce la cantidad de recursos que se requieren para describir el extenso conjunto de datos que contiene la señal de vibración del motor. Existen diversos tipos de características a extraer de las señales, entre ellas se cuentan los parámetros estadísticos; media, desviación estándar, coeficiente de asimetría, curtosis; y parámetros de complejidad; entropía, dimensión fractal, exponente de Lyapunov.

La información de entropía es un parámetro de complejidad eficaz que describe la cantidad de incertidumbre de un sistema, ha sido utilizada en muchas aplicaciones, como comunicaciones, ciencias espaciales y mecánica. En el diagnóstico de fallas en motores también se ha usado, como en [19], donde extraen el valor de entropía de las señales de vibración para detectar fallas múltiples y combinadas en máquinas rotativas. En [20], usan entropía para detectar falla en los rodamientos del motor. También, otras investigaciones [21] [22] usan información de entropía para detectar otro tipo de fallas en motores.

Otro enfoque que se aplica en el diagnóstico de la condición de trabajo de los motores es el procedimiento de clasificación, con el cual es posible catalogar y etiquetar las fallas para su correcta identificación. El procesamiento de las señales de vibración y la extracción de valores característicos a través de parámetros como la entropía, permiten obtener un conjunto de datos particulares con respecto a las señales correspondientes a cada falla y con esto es posible implementar un procedimiento de clasificación. Se han propuesto varias técnicas para resolver el problema de clasificación, entre las cuales se incluyen árboles de decisión [23], redes neuronales [24], el método *k*-nn (*k*-nearest neighbor) [25], clasificador Bayesiano [26] y máquinas de soporte vectorial (*support vector machines*, SVMs) [27]. Los clasificadores han sido extensamente usados en áreas de reconocimiento de patrones, procesamiento de imágenes [28] y también para la detección de fallas en motores como, por ejemplo, en [29] realizan una metodología para detectar falla de barras rotas y para clasificar la gravedad de la falla utilizando homogeneidad. A su vez, en [30] usan SVMs para detectar fallas en rodamientos por medio del análisis de vibraciones durante el arranque del motor.

1.1. Objetivos

1.1.1. Objetivo general

Diseñar una metodología, combinando una técnica de descomposición de señales e información de entropía, aplicada a señales de vibración para la detección de fallas múltiples y combinadas en motores de inducción.

1.1.2. Objetivos específicos

1. Realizar pruebas de la metodología propuesta en señales de vibración provenientes de diversas condiciones de falla de un motor de inducción.
2. Encontrar un método eficiente de clasificación automática para las diferentes fallas de un motor de inducción.
3. Realizar una comparación del porcentaje de efectividad de la metodología propuesta, en la detección y clasificación de fallas múltiples y combinadas de los motores de inducción, con respecto a las reportadas anteriormente.

1.2. Justificación

Los motores de inducción son elementos claves para los diversos procesos que se llevan a cabo en la industria. Son baratos, robustos y más fáciles para hacerles mantenimiento, comparado con otras alternativas. Su uso no se limita a la industria, sino que también son empleados en el hogar para tareas sencillas como la distribución del agua. Sin embargo, los motores de inducción son sensibles a fallas, las cuales afectan su correcto funcionamiento, ponen en riesgo el entorno de operación de la máquina y ocasiona pérdidas en los procesos. Las fallas más comunes que presentan los motores van relacionadas con defectos de las partes que los conforman, como el estator, el rotor, los rodamientos, el eje giratorio, los devanados del estator, etc.

En los últimos años, los estudios basados en la detección temprana de fallas en motores han experimentado un gran crecimiento. El objetivo principal de estas investigaciones ha sido estudiar y procesar las señales de voltaje, corriente o vibración de la máquina con la finalidad de desarrollar metodologías para la identificación y prevención de fallas. De esta manera, es posible minimizar los efectos de un motor dañado y realizar mantenimientos preventivos.

A diferencia de los sonidos y movimientos normales de un motor, las fallas, el uso y el desgaste, causan diferentes ruidos y vibraciones con diferente amplitud y frecuencia. Cada falla, ya sea eléctrica o mecánica, presente en el motor produce una vibración con una frecuencia específica [31]. Por tanto, mediante el estudio de las señales de vibración, que describen las características dinámicas del motor, se pueden detectar esas frecuencias específicas, con las cuales es posible identificar el tipo de falla que presenta la máquina.

Las señales de vibración se obtienen del motor mediante diversos sensores, como acelerómetros. Para el estudio de estas señales se emplean métodos de análisis en frecuencia o tiempo-frecuencia, como la transformada de Fourier, la descomposición de modo empírico (EMD), la transformada de Wavelet, entre otras. Recientemente, el método de descomposición de modo empírico por conjuntos complementarios (CEEMD) ha sido usado para el seguimiento de la condición de trabajo de los motores de inducción [15, 32, 33]. A través de este método se extraen los diversos modos de oscilación que contiene la señal de vibración y con procedimientos de extracción de características, como entropía, media, desviación estándar, entre otros, es posible construir un descriptor para clasificar y etiquetar las fallas.

Es por ello que, en este trabajo se propone una metodología para la identificación de fallas múltiples y combinadas en motores, mediante la integración del método de descomposición de señales, CEEMD, con entropía, para extraer características relevantes de las señales, y un clasificador para la identificación de las condiciones de operación barra rota, defectos en rodamientos, desbalance de carga y su combinación

correspondiente, así como también un estado de motor sano para comparación. La contribución de este trabajo es un enfoque diferente, en el área de procesamiento de señales, para el seguimiento de la condición de trabajo y la detección automática de fallas en motores de inducción.

1.3. Estructura del trabajo

El trabajo de tesis se encuentra organizado en cinco capítulos, los cuales se describen brevemente a continuación:

Capítulo I. Introducción. Contiene una breve reseña de los trabajos previos que se han realizado en el área de estudio, seguido por los objetivos propuestos para el trabajo y, finalmente, se expone el motivo de la realización de esta investigación.

Capítulo II. Generalidades. Se especifican los conceptos asociados a los motores de inducción y sus fallas más comunes. A su vez, se realiza una breve descripción de los términos correspondientes a las señales de vibración en motores de inducción.

Capítulo III. Metodología de análisis de funciones modales. Se especifican los conceptos asociados a las técnicas de procesamiento de señales que se emplearon. Además, una breve descripción de la importancia de la extracción de características en señales. Finalmente, algunos algoritmos de aprendizaje de máquina, enfocados en el desarrollo de clasificadores, son descritos.

Capítulo IV. Descripción del procedimiento propuesto. Se describe el procedimiento, basado en las técnicas de procesamiento de señales, extracción de características y clasificadores, realizado para el diagnóstico de la condición de trabajo de los motores de inducción.

Capítulo V. Resultados. Se exponen los resultados obtenidos luego de aplicado el procedimiento propuesto.

Capítulo VI. Conclusiones. Contiene las conclusiones correspondientes a los resultados obtenidos, basadas en la metodología planteada.

Capítulo 2

Generalidades

A continuación se presenta una breve descripción de los conceptos básicos que se emplean para la realización de esta investigación. Además, se describen los términos concernientes a fallas en motores de inducción y sus características más relevantes.

2.1. Motores de inducción

Los dispositivos eléctricos considerados máquinas de corriente alterna (CA) rotativas se dividen en dos grupos. El primero de ellos es el de los generadores, los cuales convierten energía mecánica en energía eléctrica, y el segundo grupo son los motores, encargados de convertir energía eléctrica en energía mecánica. A su vez, las máquinas de CA se dividen en dos clases principales, por un lado, están las máquinas síncronas y, por otro lado, las máquinas de inducción o asíncronas.

En primer lugar, las máquinas síncronas comprenden los motores y generadores, dispositivos a los cuales se les suministra la corriente de campo magnético por medio de una fuente de potencia de CA que es externa. En segundo lugar, las máquinas asíncronas o de inducción son motores y generadores cuya corriente de campo magnético se suministra a sus devanados de campo por medio de inducción magnética (acción transformadora) [34]. Sin embargo, una máquina asíncrona presenta muchas desventajas como generador y rara vez se utiliza de este modo, por lo tanto, se les llama solamente motores de inducción.

Los motores de inducción son las máquinas eléctricas más comúnmente usadas. Tienen dos partes principales, el estator y el rotor, como se muestra en la Figura 2.1, de [35]. El estator es la parte estacionaria y el rotor es la parte giratoria [36]. El estator se hace apilando finas láminas ranuradas de acero muy permeable dentro

de una carcasa de acero. En las ranuras de estas láminas se ubican los devanados del estator [34]. El rotor está ubicado en el interior del estator de forma coaxial, y consiste en un conjunto de finas láminas ranuradas, llamadas laminaciones, de sustancia electromagnética (núcleo especial de acero), y se disponen juntas presionadas en forma de una trayectoria magnética cilíndrica. En las ranuras de estas laminaciones se ubican las barras conductoras.

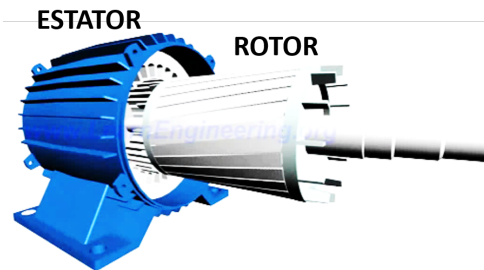


Figura 2.1. Estator y rotor de un motor de inducción [35].

El funcionamiento de un motor de inducción consiste en que, cuando una corriente trifásica de CA pasa a través de los devanados del estator, se produce un campo magnético giratorio [37], como se observa en la Figura 2.2, de [35]. Este campo magnético induce, de acuerdo con la ley de Faraday, una fuerza electro motriz (FEM) en el rotor. La FEM genera una corriente en las barras del rotor, que se encuentran cortocircuitadas por anillos a cada extremo. La corriente que circula por las barras, inmersas en un campo magnético, produce una fuerza magnética (fuerza de Lorentz), gracias a la cual el rotor empieza a girar. En la Figura 2.3, de [35], se ilustra este proceso. Por esa razón, es conocido como motor de inducción, la electricidad es inducida en el rotor por inducción magnética, en lugar de producirse por conexiones eléctricas directas, y la potencia mecánica rotacional se transfiere a través del eje del rotor. De modo que, en un motor de inducción, la energía eléctrica entra por el estator y pasa al rotor, donde se produce la energía mecánica.

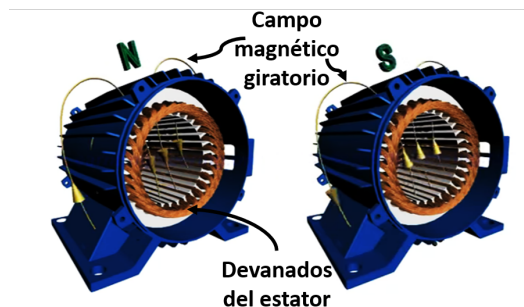


Figura 2.2. Campo magnético giratorio que se produce en el estator [35].

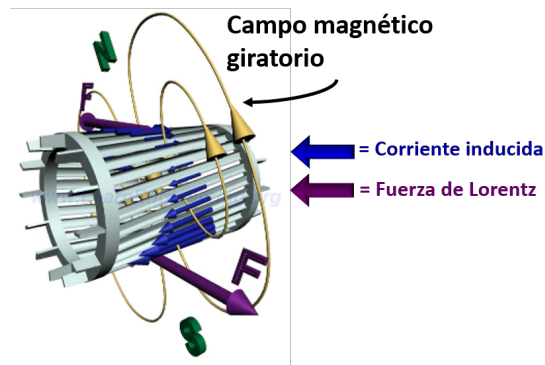


Figura 2.3. Corriente inducida y fuerza de Lorentz en el rotor [35].

Otra de las partes principales de un motor de inducción son los rodamientos (o cojinetes). La máquina contiene dos, uno se ubica en el extremo de conducción y otro en el de no conducción, y se usan como apoyo para el eje giratorio. El uso de los rodamientos es apropiado para ubicar correctamente el rotor dentro del estator y para minimizar la fricción [38]. Adicionalmente, el motor contiene lo siguiente, el eje giratorio (hecho de acero) para transmitir el torque a la carga; ventilador, ubicado en el extremo de no conducción, y se encarga del enfriamiento tanto para el estator como para el rotor; y por último, una caja de terminales, ubicada en la parte superior o a ambos lados, para recibir las conexiones de la fuente de alimentación eléctrica. En la Figura 2.4, de [39], se puede apreciar la ubicación de las partes mencionadas.

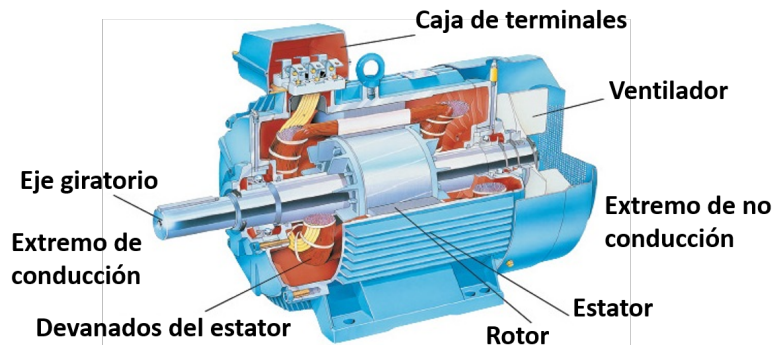


Figura 2.4. Partes de un motor de inducción [39].

2.2. Señales de vibración en los motores de inducción

Una vibración se puede definir como la oscilación de un cuerpo con respecto a un punto de referencia (equilibrio), o como cualquier movimiento que se repite después de un intervalo de tiempo [40]. Desde el punto de vista energético, la vibración es un fenómeno que alterna el intercambio entre energía potencial y energía cinética [41]. Todos los cuerpos presentan una señal de vibración en la cual dejan ver cada una de sus características, las cuales pueden ser evidencia de algún defecto y, en muchos casos, afecta grandemente la naturaleza de los diseños de ingeniería.

Las máquinas rotativas generan vibraciones como parte normal de su actividad. Cada máquina presenta una vibración característica que la hace única, se conoce como firma de vibración. Esta señal contiene la información de cada uno de los componentes del motor debido a que está condicionada por el diseño, fabricación, uso y desgaste de esos componentes. Por lo tanto, una señal de vibración capturada de un motor se compone de la suma de las vibraciones de cada una de sus partes.

Las vibraciones se clasifican en tres grupos:

1. Vibración simple. Su base principal en el dominio del tiempo son las ondas sinusoidales y, por lo tanto, en su espectro aparece una sola frecuencia. Son las más simples y son la representación de las oscilaciones puras. Los parámetros que definen este tipo de vibración son la frecuencia, que indica el período de oscilación de la señal; la amplitud, corresponde a la intensidad de la vibración e indica la severidad de la misma; y la fase.
2. Vibración compuesta. Es la suma de varias vibraciones simples. Normalmente la vibración que presenta un motor de inducción es una compuesta de una serie de vibraciones simples asociadas a sus componentes internos en movimiento. Por lo tanto, la forma de onda de este tipo de vibración puede llegar a ser muy compleja, siendo difícil de interpretar.
3. Vibración aleatoria. No cumple con patrones especiales que se repitan constantemente, siendo demasiado difícil detectar dónde comienza y dónde termina un ciclo. El patrón que las caracteriza se interpreta mejor en el espectro de frecuencias de la señal.

La vibración aleatoria es el patrón que distingue las señales de vibración provenientes de un motor de inducción, las cuales se caracterizan por su amplitud y frecuencia. En primer lugar, la amplitud muestra qué tan fuerte es la vibración y, en segundo lugar, la frecuencia muestra la tasa de oscilación de la vibración. Ambas proveen información

para identificar la raíz de la vibración [31]. Normalmente, estas señales se emplean para estudiar la condición de trabajo del motor y para detectar las fallas que éste presenta, debido a que describen las características dinámicas de la máquina [42].

2.3. Fallas en motores de inducción

Los motores de inducción forman parte de la maquinaria clave en la industria, su uso es muy común y representan un componente importante para los diversos procesos. Sin embargo, estos motores normalmente están expuestos a ambientes ruidosos y bajo una alta demanda de trabajo. Por lo tanto, su buen funcionamiento se pone en riesgo ya que son máquinas susceptibles a diferentes e indeseables tipos de fallas, como las eléctricas, mecánicas y de medio ambiente. Las partes más vulnerables a fallas en los motores son los rodamientos, las barras del rotor y el eje giratorio [43]. En este trabajo se consideran tres de las más comunes, las cuales están relacionadas con las partes mencionadas anteriormente, y se describen a continuación.

2.3.1. Barra rota

El rotor de un motor de inducción puede estar construido de dos formas diferentes. Una es la construcción del tipo jaula de ardilla o rotor de jaula, como se observa en la figura 2.5a, de [44]. Y la otra es la de rotor devanado, la cual, por su construcción, tamaño y mantenimiento, poco se usa [34]. Un rotor de jaula consta de una serie de barras conductoras que se encuentran ubicadas en las ranuras de las laminaciones que componen el rotor y están cortocircuitadas por anillos en sus extremos. La Figura 2.5b, de [45], ilustra a detalle un rotor de jaula de ardilla con las partes que lo componen.

La falla de barra rota se presenta cuando, por diversas causas, una de las barras se encuentra parcialmente agrietada o completamente rota. Las causas por las cuales ocurre esta falla son diversas, entre ellas se cuentan los defectos durante el proceso de fabricación; tensiones térmicas y mecánicas, éstas últimas ocasionadas por fallas en los rodamientos; arranque frecuente del motor al voltaje nominal y también por el desgaste que se produce en el metal del cual están hechas las barras.

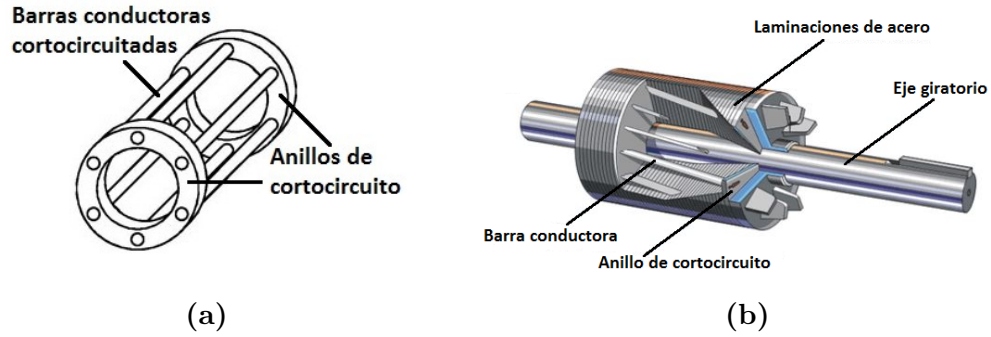


Figura 2.5. a) Dibujo de la jaula de ardilla de un rotor [44]. b) Rotor de jaula de ardilla típico [45].

Una barra rota es el mayor defecto que puede presentar el rotor de un motor de inducción. Una vez que se rompe una barra, la condición de las otras barras también se pone en riesgo por el deterioro progresivo que surge debido a mayores tensiones [46]. Esta falla es perceptible en el motor debido a que altera la perturbación del campo magnético en el rotor, intensificando las modulaciones del par y las vibraciones del motor, las cuales pueden ser medidas mediante sensores de vibración, como acelerómetros, localizados en la carcasa del motor.

Si existe una barra de rotor rota, la corriente no fluirá por ésta. Como resultado, el campo magnético en el rotor alrededor de esa barra en particular no existirá [31]. La fuerza aplicada en ese lado del rotor será diferente a la del otro lado, con lo que se crea una fuerza magnética desequilibrada que gira al tiempo de la velocidad de rotación y modula a una frecuencia igual a la frecuencia de deslizamiento multiplicada por el número de polos, la cual se conoce como frecuencia de paso de polo, f_p [47].

El espectro de la señal de vibración, proveniente de un motor que presenta falla de barra rota, tendrá amplitudes incrementadas en la frecuencia de rotación, f_r , y sus bandas laterales, f_{brb} , [46]:

$$f_{brb} = f_r \pm f_p \quad (2.1)$$

donde, la frecuencia de paso de polo (f_p) se define como:

$$f_p = (f_{sync} - f_r) \cdot P \quad (2.2)$$

siendo P el número de polos del motor y f_{sync} la velocidad síncrona, la cual corresponde a la velocidad de rotación del campo magnético en una máquina rotatoria y viene dada por [34]:

$$f_{sync} = \frac{120 \cdot f_e}{P} \quad (2.3)$$

donde, f_e corresponde a la frecuencia, en Hz , de la red eléctrica a la que se encuentra conectado el motor. Las bandas laterales se producen también en armónicos más altos de la velocidad de rotación ($2f_r, 3f_r, \dots$). En la Figura 2.6, de [46], se muestra el espectro de las señales de vibración obtenidas cuando el motor presenta falla de barra rota y cuando se encuentra en condición sano.

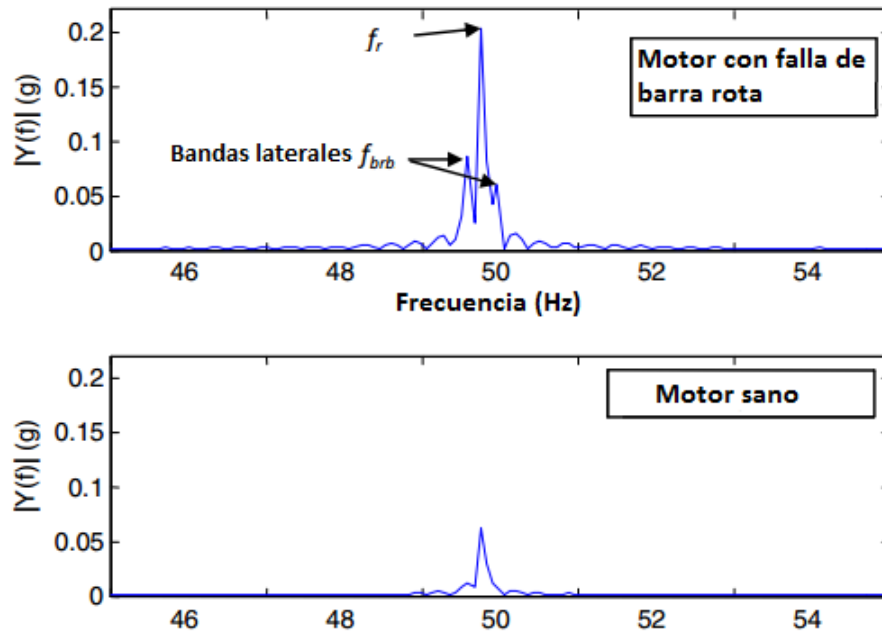


Figura 2.6. Espectro de las señales de vibración de un motor con falla de barra rota y condición sano [46].

En el espectro de la Figura 2.6, se puede apreciar que cuando el rotor tiene una de sus barras rotas, las amplitudes se incrementan tanto en la frecuencia de rotación como en las bandas laterales que se especifican en (2.1), a diferencia de la condición de un motor sano que solo tiene amplitud incrementada en la frecuencia de rotación. En consecuencia, a través del estudio de las señales de vibración es posible determinar si un motor presenta alguna condición que pueda afectar su correcto funcionamiento.

2.3.2. Desbalance de carga

La energía que se obtiene de un motor de inducción es energía mecánica rotativa, que se conoce como par-motor o torque y se ejerce sobre el eje giratorio, o eje de transmisión de potencia. Como consecuencia, lo que la carga demanda del motor es el par o la tendencia de una fuerza para girar un objeto alrededor de un eje.

La falla por desbalance se presenta cuando la carga mecánica del motor de inducción no se encuentra uniformemente distribuida, tomando el centro de masa fuera del eje del motor [48]. Esta falla es considerada como la mayor causa de excentricidades dinámicas o estáticas, o la propagación de defectos en los rodamientos [49]. Entre las principales causas de este tipo de falla se encuentran los defectos de fabricación, desalineamiento interno, curvatura del eje, o también puede ocurrir luego de un largo periodo de operación debido a la adición o sustracción asimétrica de masa alrededor del centro de rotación del rotor.

Los problemas mecánicos que se presenten en la carga del motor pueden causar oscilaciones de velocidad que modulan el motor, además, estos desbalances en el rotor, inducen componentes armónicos específicos en cantidades eléctricas, electromagnéticas y mecánicas [50]. El efecto que genera la falla de desbalance en las señales de vibración del motor, corresponde a un aumento significativo de la amplitud en la frecuencia de rotación, f_r . La Figura 2.7, de [49], muestra el espectro centrado en esta frecuencia, f_r . El espectro mostrado fue extraído de [49] donde se realizan pruebas experimentales para un motor con desbalance de carga.

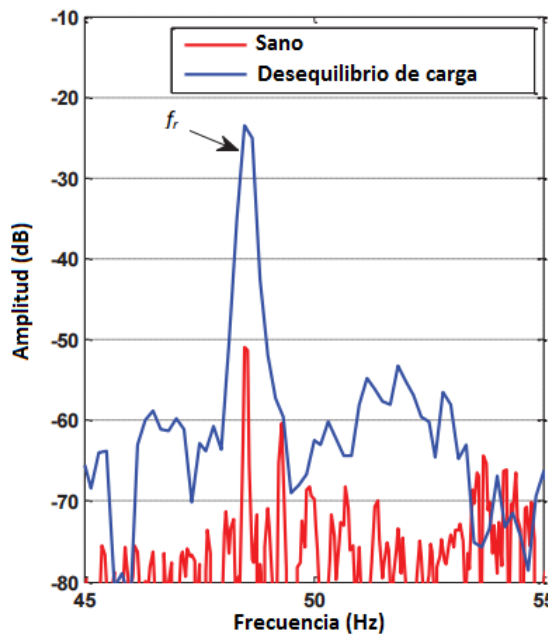


Figura 2.7. Espectro de las señales de vibración de un motor con falla por desbalance de carga. Condición de operación: carga completa. Vibración axial [49].

En la Figura 2.7, se puede observar un incremento significativo de la amplitud del espectro para la condición de desbalance con respecto a la de motor sano. La variación observada en las señales de vibración ante la presencia de un desbalance de carga,

permite emplear estas señales para el estudio de la condición de trabajo de los motores de inducción.

2.3.3. Defecto en rodamientos

El eje giratorio de un motor de inducción se sostiene mediante rodamientos, para ello se emplean dos juegos, cada uno ubicado en los extremos del motor. Además de sostener el eje, con los rodamientos es posible mantener al rotor correctamente localizado para hacerlo girar libremente y sin aumentar las fricciones. Cada juego consiste en dos anillos, uno interno y otro externo, y en el medio de estos dos anillos se ubican pequeñas esferas, las cuales sirven como elementos rodantes. En la Figura 2.8a, de [51], se muestra un rodamiento típico. Normalmente, el anillo interno se une al eje y la carga se transmite a través de las esferas giratorias, permitiendo la reducción de fricción [36]; la Figura 2.8b, de [36], muestra a detalle las partes de un rodamiento.

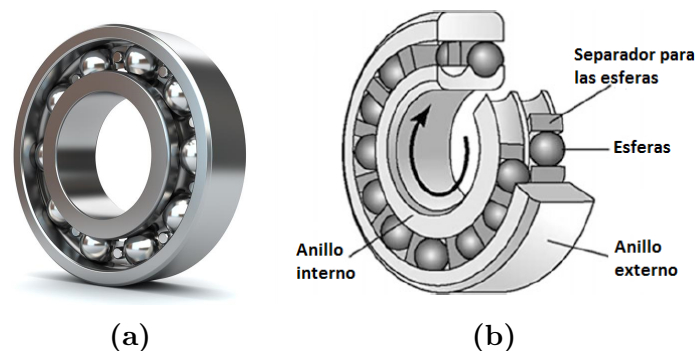


Figura 2.8. a) Rodamiento típico [51]. b) Partes de un rodamiento [36].

Un defecto en rodamientos se presenta cuando existe cualquier daño físico en la superficie de los anillos, en los elementos rodantes (esferas) o en el soporte de las esferas [52]. Las principales causas de estas fallas, además de vibraciones, excentricidades dinámicas o estáticas, son:

1. Corrosión debido a la exposición de los rodamientos a una atmósfera corrosiva o a fluidos corrosivos como ácidos y agua, entre otros [53].
2. Lubricación inadecuada por falta de lubricante y la consecuente aparición de altas temperaturas. Con el uso, se pierde la propiedad del líquido lubricante, provocando un desgaste excesivo de las esferas y anillos y, por lo tanto, un sobrecalentamiento [52].

3. La contaminación corresponde a una de las mayores causas de defectos en los rodamientos, ya que genera altas vibraciones y desgaste. Los lubricantes se contaminan debido a partículas presentes en ambientes industriales.
4. El desgaste debido al movimiento constante que experimentan los rodamientos, lo cual genera la fractura de los elementos que lo componen, esparciendo partículas del material.
5. Cargas excesivas, ajustes muy apretados y subidas de temperatura. Estos factores pueden generar la degradación e incluso acabar con el lubricante.

En la industria, los defectos en rodamientos son el modo de falla más común en los motores eléctricos. Reportes demuestran que cerca del 45 % de los fallos que se presentan en las máquinas de inducción son causados por rodamientos defectuosos [54]. Además, estas fallas se presentan de manera progresiva con el tiempo y causan aumento de la vibración del motor, pudiendo ocasionar un paro inesperado de la máquina.

Los rodamientos defectuosos son considerados una falla localizada que puede producir frecuencias de vibración mecánica específicas, relacionadas tanto con la geometría del rodamiento como con la velocidad. Las frecuencias que corresponden a defectos en los anillos interno, (f_i), y externo, (f_o), del rodamiento son [54]:

$$f_i = \frac{N}{2} f_r \left(1 + \frac{BD}{PD} \cos(\alpha) \right) \quad (2.4)$$

$$f_o = \frac{N}{2} f_r \left(1 - \frac{BD}{PD} \cos(\alpha) \right) \quad (2.5)$$

donde, f_r es la frecuencia rotacional o velocidad del motor, BD es el diámetro de las esferas, PD es el diámetro del soporte de las esferas, N es el número de esferas y α es el ángulo de contacto de las esferas con los anillos.

Además de estas frecuencias, también están las de los defectos en las esferas, f_b , y en el soporte de éstas, f_c , las cuales se describen así [55]:

$$f_b = \frac{PD}{2BD} f_r \left(1 - \frac{BD^2}{PD^2} \cos^2(\alpha) \right) \quad (2.6)$$

$$f_c = \frac{f_r}{2} \left(1 - \frac{BD}{PD} \cos(\alpha) \right) \quad (2.7)$$

En la Figura 2.9, de [55], se muestran a detalle los componentes de un rodamiento con las dimensiones correspondientes, necesarias para las frecuencias previamente mencionadas.

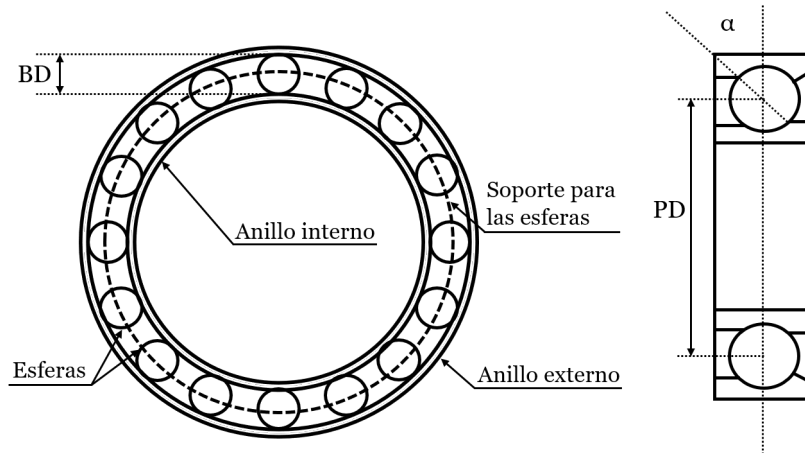


Figura 2.9. Componentes de un rodamiento con sus dimensiones correspondientes [55].

Para (2.4) y (2.5) es necesario contar con información detallada de las características de los rodamientos, la cual podría no estar disponible. En tal caso, si se cuenta con rodamientos que contengan entre seis y 12 esferas, y las fallas sean periódicas, es posible sustituir esas ecuaciones por las siguientes [54]:

$$f_i = 0,6Nf_r \quad (2.8)$$

$$f_o = 0,4Nf_r \quad (2.9)$$

Cuando los rodamientos contengan defectos, en las señales de vibración se generaran algunas de las frecuencias definidas en (2.6) - (2.9). La Tabla 2.1, extracto de la que se encuentra en [56], resume las características de estas señales en presencia de alguna de las fallas previamente mencionadas.

Tabla 2.1. Características de las señales de vibración en motores con fallas en los rodamientos

Ubicación de la falla	Frecuencia en el espectro	Observaciones
Condición sano	f_r, f_o, f_i	Bajo la condición sano puede tener las frecuencias f_r, f_o, f_i y sus armónicos correspondientes, la amplitud es pequeña y no destaca ninguna frecuencia.
Esferas	$2f_b, f_o, f_i$	Para casos severos estas frecuencias pueden ser moduladas por $2f_b, f_c$, y la frecuencia natural puede alterarse también.
Anillos	f_o, f_i	Un incremento severo en los defectos genera armónicos de valores altos donde destacan las frecuencias definidas para fallas en los anillos y si hay poca compensación también aparecerán armónicos en $2f_b$.

En un motor de inducción, la presencia de una barra rota, un desbalance de carga o un rodamiento defectuoso, se refleja en su modo de operación. Específicamente, las vibraciones del motor cambian, volviéndose más intensas, generando un mayor consumo de energía o un desgaste de las partes que lo conforman. En este sentido, un procedimiento de análisis de las señales de vibración del motor, que permita extraer la mayor cantidad de información de ellas, es requerido para prevenir daños o evitar futuras interrupciones de la máquina. Es por ello que, en el siguiente capítulo se presentan los métodos de análisis de señales, propuestos en este trabajo de investigación, para la detección, por medio de señales de vibración, de las fallas expuestas anteriormente, tanto de forma individual como combinadas.

Capítulo 3

Metodología de análisis de funciones modales

El estado de operación de un motor de inducción se refleja en las vibraciones que genera constantemente. De este modo, y como se ha mencionado, la presencia de una barra rota, un desbalance de carga o un rodamiento defectuoso es perceptible si se realiza un análisis en frecuencia, o en tiempo, de las señales de vibración del motor. Considerando esto, para el tratamiento de las señales de vibración, provenientes de un motor que presenta condición de falla, fueron usados los métodos que se presentan en este capítulo, estructurado en tres partes principales. En primer lugar, se describe el método de descomposición de señales. Seguidamente, se describe la extracción de características, como procedimiento para resumir la información contenida en los modos obtenidos durante la descomposición. Finalmente, los tres algoritmos de clasificación, usados para identificar y etiquetar fallas y para validar las características seleccionadas en la etapa anterior, son descritos. En el diagrama de bloques de la Figura 3.1, se muestra un esquema general de estos métodos a emplear, en conjunto, para el análisis de las señales.

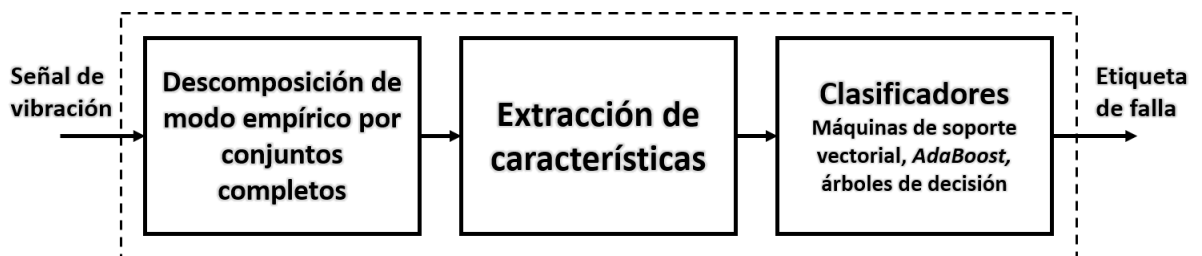


Figura 3.1. Procedimiento para el análisis de las señales de vibración.

3.1. Descomposición de modo empírico por conjuntos completos

Las máquinas rotativas generan vibraciones como parte normal de su actividad. Sin embargo, cuando alguno de los componentes de la máquina presenta un defecto o una falla, las características de esas vibraciones cambian, permitiendo que, a través de un estudio detallado, se pueda identificar el lugar y el tipo de falla, realizar la reparación correspondiente y planificar programas de mantenimiento. El análisis de las vibraciones se basa en interpretar las señales de vibración, caracterizadas por ciertos valores de aceleración, velocidad y desplazamiento, con los cuales se puede obtener información de la vibración [57].

Las señales que se obtienen de las máquinas rotativas, como motores de inducción, pueden clasificarse como estacionarias o no estacionarias [15]. En primer lugar, las señales estacionarias se caracterizan por sus propiedades estadísticas invariables en el tiempo, y pueden ser estudiadas correctamente usando técnicas espectrales como la transformada rápida de Fourier (*fast Fourier transform*, FFT). En segundo lugar, las señales no estacionarias son esencialmente transitorias, lo cual hace inválida la suposición de la estacionariedad de la señal por la FFT, reduciendo así su efectividad [15].

Por otro lado, aunque muchos fenómenos pueden ser aproximados por sistemas lineales, también tienen la tendencia a ser no lineales cuando las variaciones en amplitud se hacen finitas [6]. Esta situación ocurre con las señales de vibración del motor, que presentan la propiedad de no linealidad, ya que el sistema que las genera no cumple con las propiedades de superposición y escalamiento, por lo tanto, se deben usar métodos dinámicos no lineales para el procesamiento y análisis correspondiente [58].

La descomposición de modo empírico (*empirical mode decomposition*, EMD) es un método adaptativo mediante el cual cualquier conjunto de datos complicados puede descomponerse en un número finito de funciones de modo intrínseco (*intrinsic mode functions*, IMFs). El método está basado en las escalas de tiempo físicas de la señal, las cuales caracterizan las oscilaciones del sistema y también permiten aplicar la descomposición en señales provenientes de procesos no estacionarios y no lineales [6].

Las IMFs reciben su nombre debido a que son una representación de los modos de oscilación que contiene la señal descompuesta. Por lo tanto, la IMF en cada ciclo, definido por los cruces por cero, contiene un modo de oscilación, para lo cual satisface las siguientes condiciones [6]:

1. En todo el conjunto de datos, el número de extremos y el número de cruces por cero deben ser iguales o diferir como máximo en uno. Es decir, el número de polos y ceros es igual o difieren como máximo por uno.

2. En cualquier punto, el valor medio de la envolvente definida por los máximos locales y la envolvente definida por los mínimos locales es cero. Es decir, la envolvente superior e inferior deben ser simétricas respecto a la línea de tiempo.

El principio básico del método de EMD, descrito en [59], es analizar una señal $x(t)$ entre dos extremos consecutivos (mínimos o máximos) y definir una parte de alta frecuencia local, llamada detalle, $d(t)$, y una tendencia local, $a(t)$, tal que $x(t) = d(t) + a(t)$. La primera IMF está compuesta de los detalles locales, $d(t)$, obtenidos de todos los extremos consecutivos de $x(t)$. El procedimiento de separación de alta contra el de baja frecuencia se repite de forma iterativa sobre el residuo $a(t)$, dando lugar a un nuevo IMF y a un nuevo residuo.

De acuerdo con [59], el procedimiento propuesto por Huang *et al.* en [6] para descomponer la señal de entrada $x(t)$ se puede resumir en los siguientes pasos:

1. Identificar todos los extremos (mínimos y máximos locales) de $x(t)$.
2. Obtener la envolventes superior ($e_{max}(t)$) e inferior ($e_{min}(t)$) por medio de la interpolación de los máximos y los mínimos locales, respectivamente.
3. Calcular la tendencia local como el valor medio entre la envolvente superior y la envolvente inferior, así $a(t) = (e_{min}(t) + e_{max}(t))/2$.
4. Calcular el componente de detalle como $d(t) = x(t) - a(t)$.
5. Iterar sobre la tendencia local que queda como residuo, $a(t)$.

Las IMFs deben tener valor medio cero y todos sus máximos y mínimos locales deben ser positivos y negativos, respectivamente, esto de acuerdo a las condiciones definidas previamente para la IMF. Si el componente de detalle, que se obtiene en el paso cuatro, no sigue las condiciones mencionadas, entonces los pasos uno al cuatro se repiten usando $d(t)$ en lugar de $x(t)$. Este proceso, denominado *sifting*, se repite hasta que el nuevo $d(t)$ es considerado como una IMF. Para el siguiente IMF, el proceso *sifting* se aplica sobre el residuo $a(t) = x(t) - d(t)$ [59].

Tal como señala Zão *et al.* en [59], a partir del procedimiento de EMD, se puede observar que el número total de extremos es reducido de una IMF a la siguiente. La forma de onda de cada modo puede interpretarse como una señal de amplitud y frecuencia modulada con valor medio de cero. La primera IMF tendrá oscilaciones más rápidas que la segunda, la cual a su vez será de fluctuaciones más rápidas que la tercera, y así sucesivamente con todas las IMFs obtenidas. Ésto significa que, a cada intervalo de tiempo, la EMD realiza una separación de alta frecuencia contra baja frecuencia

entre todas las IMFs. Por lo tanto, los primeros modos presentarán el contenido de alta frecuencia de la señal y la última componente tendrá la menor frecuencia.

Debido a que el procedimiento de EMD se aplica solamente si hay, por lo menos, dos extremos en el último residuo $a(t)$ calculado, cualquier señal de entrada $x(t)$ puede descomponerse en un número finito de IMFs. Si la m -ésima IMF es expresada como IMF_m y un total de M IMFs son extraídas de $x(t)$, entonces

$$x(t) = \sum_{m=1}^M IMF_m(t) + r(t), \quad (3.1)$$

donde, $r(t)$ es el último residuo obtenido del procedimiento de EMD. En la Figura 3.2, de [60], se muestra una representación gráfica del procedimiento que desarrolla la EMD, empleando una onda compuesta por dos señales senoidales a diferentes frecuencias.

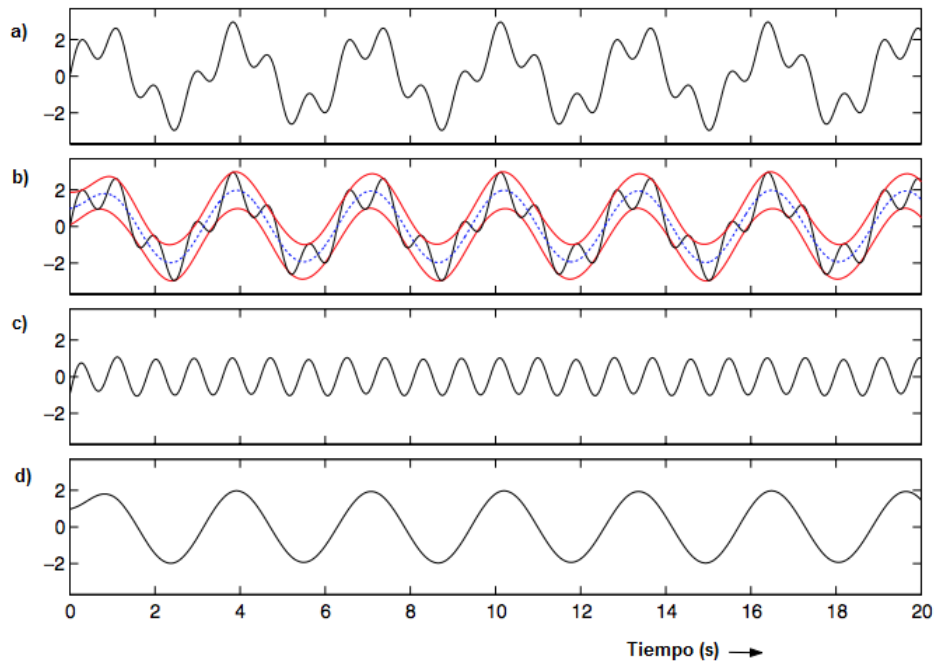


Figura 3.2. EMD aplicada a una señal compuesta por dos ondas seno: a) la señal original, b) envolventes superior e inferior (rojo) y la media (azul punteado), c) la primera descomposición y d) el primer residuo [60].

Las señales de voltaje, corriente y vibración que se obtienen de un motor de inducción, presentan características complejas y con una simple observación no es suficiente para extraer de ellas información con la cual estudiar la condición de trabajo de estos motores. Es por ello que, la descomposición mediante EMD es un buen procedimiento para obtener información relevante de estas señales, las cuales, por ser de naturaleza no lineales y no estacionarias, no admiten el uso de métodos como la

FFT. Sin embargo, EMD presenta un problema denominado la mezcla de modos, en el cual una única IMF consta de señales con escalas muy dispares o una señal de escala similar que reside en diferentes componentes IMF [13].

Para solucionar el problema de la mezcla de modos, fue propuesto un nuevo método de análisis de datos asistido por ruido [61], basado en EMD, denominado descomposición de modo empírico por conjuntos (*ensemble empirical mode decomposition*, EEMD). Con este método se añade ruido blanco gaussiano, de amplitud finita, a la señal original antes de iniciar el proceso de descomposición. La ventaja de añadir ruido blanco esta en que, gracias a la característica estadística de distribución uniforme del ruido, la señal con ruido blanco se vuelve continua en diferentes escalas de tiempo evitando que se generen escalas dispares. El proceso de EEMD, presentado en [61], es el siguiente:

1. A la señal original, $x(t)$, se le añade ruido blanco,

$$y(t) = x(t) + \varepsilon w(t), \quad (3.2)$$

donde, $y(t)$ es una señal artificial; ε es la amplitud del ruido añadido, la cual es una fracción del valor de desviación estándar de la señal original; $w(t)$ es la serie aleatoria de ruido blanco, cuya distribución normal es de media cero y desviación estándar constante.

2. A través del procedimiento original de EMD se descompone la nueva señal artificial, $y(t)$, en IMFs.
3. Se repiten los pasos uno y dos N cantidad de veces con diferentes series de ruido blanco. Con cada repetición se obtiene un componente del conjuntos de IMFs de la forma:

$$\{\{c_j^1(t)\}, \{c_j^2(t)\}, \{c_j^3(t)\}, \dots, \{c_j^N(t)\}\}$$

donde, $j = (1, 2, 3, \dots, n)$ es el número de IMFs, N representa el número del conjunto y $\{c_j^1(t)\}$ es el componente de IMFs luego de desarrollar EMD la primera vez.

4. Se calcula el valor medio del conjunto de las IMFs correspondientes, $c_j(t)$, de la descomposición como el resultado final de:

$$c_j(t) = \frac{1}{N} \sum_{b=1}^N c_j^b(t) \quad (3.3)$$

De acuerdo con (3.3), el resultado final indica que cada modo de EEMD es el valor medio del conjunto de componentes IMF. Por otro lado, en EEMD, el número de

conjuntos y la amplitud del ruido tienen dos parámetros esenciales. La relación entre la cantidad de conjuntos, la amplitud del ruido y el efecto del ruido añadido se obtiene por medio de la siguiente regla:

$$\varepsilon_n = \frac{\varepsilon}{\sqrt{N}} \quad (3.4)$$

o

$$\ln(\varepsilon_n) + \frac{\varepsilon}{2} \ln(N) = 0 \quad (3.5)$$

donde, ε_n es el valor de la desviación estándar de la diferencia entre la señal de entrada y las IMFs correspondientes.

Como ya se ha visto, mediante EEMD se realiza el mismo procedimiento de EMD pero sobre un conjunto de señales con ruido gaussiano [14]. Añadir ruido gaussiano resuelve el problema de la mezcla de modos mediante el relleno de todo el espacio tiempo-frecuencia; sin embargo, este procedimiento genera otros problemas y es que la señal reconstruida contiene residuos de ruido y a diferentes realizaciones de la señal más ruido se puede producir diferente cantidad de modos. Con la finalidad de solucionar este nuevo problema, un nuevo método fue propuesto por Torres *et al.* [14] denominado descomposición de modo empírico por conjuntos completos (*complete ensemble empirical mode decomposition*, CEEMD). A través de CEEMD, se provee una mejor reconstrucción de la señal original sin el problema de la mezcla de modos y sin tanto gasto de procesamiento en hardware. El procedimiento, según [15], es el siguiente:

En primer lugar, se añade un porcentaje de ruido blanco gaussiano a la señal que se quiere descomponer y se obtiene la primera componente EMD de la señal con ruido. La descomposición se repite N veces usando diferentes realizaciones de ruido y se calcula la media de todo el conjunto para determinarla como la primera descomposición (IMF_1) de la señal original. Entonces,

$$IMF_1 = \frac{1}{N} \sum_{i=1}^N E_i[x(t) + \varepsilon w_i] \quad (3.6)$$

donde, IMF_1 es la primera componente EMD de la señal original $x(t)$; w_i es el ruido blanco gaussiano de media cero y varianza unitaria; ε es un coeficiente fijo; E_i produce la i -ésima componente IMF y N es el número de realizaciones. Luego, el primer residuo de la señal r_1 se define por:

$$r_1 = x(t) - IMF_1 \quad (3.7)$$

Seguidamente, se realizan las descomposiciones $r_1 + \varepsilon E_1[w_i]$, $i = 1, 2, \dots, N$; hasta que se cumpla con la primera condición para que sea una componente IMF adecuada y definir la media del conjunto como la segunda componente IMF_2 ,

$$IMF_2 = \frac{1}{N} \sum_{i=1}^N E_1[r_1 + \varepsilon E_1[w_i]] \quad (3.8)$$

Para $k = 2, 3, \dots, K$, se calcula el k -ésimo residuo: $r_k = r_{k-1} - IMF_k$, luego se extrae la primera componente IMF de $r_k + \varepsilon E_k[w_i]$, $N = 1, 2, \dots, N$ y se calcula otra vez la media del conjunto para obtener la componente IMF_{k+1} de la señal original,

$$IMF_{k+1} = \frac{1}{N} \sum_{i=1}^N E_i[r_k + \varepsilon E_k[w_i]] \quad (3.9)$$

El proceso continúa hasta que el último residuo no tiene más de dos polos o ceros, produciendo:

$$R = x(t) - \sum_{k=1}^K IMF_k \quad (3.10)$$

donde, R es el residuo final y K es el total de componentes IMF encontradas. Por último, la señal original puede ser expresada como:

$$x(t) = \sum_{k=1}^K IMF_k + R \quad (3.11)$$

A partir de (3.11) se puede considerar CEEMD como un método de descomposición completo, que comparado con los EMD y EEMD, no sólo resuelve el problema de la mezcla de modos sino que también provee una buena reconstrucción de la señal original.

3.2. Extracción de características

El diagnóstico de fallas en los motores de inducción se realiza por medio del estudio de las características representativas existentes en las señales de voltaje, corriente o vibración, provenientes de un motor con determinadas condiciones de operación. Estas características van relacionadas con el comportamiento de la señal, tanto en el dominio del tiempo como en el de la frecuencia, con las variaciones de amplitud, con los modos de oscilación, etc.

El proceso mediante el cual se representan las medidas originales de una señal en atributos más eficaces, se denomina extracción de características [18]. Normalmente, las características que se obtienen son valores numéricos utilizables para el aprendizaje automático, y mediante los cuales se reduce la cantidad de recursos que se requieren para describir el extenso conjunto de datos que contiene la señal de voltaje, corriente o vibración del motor. Al arreglo de esos valores numéricos se le denomina vector de características y son comúnmente usados en las áreas de reconocimiento de patrones y aprendizaje de máquina. También, para el desarrollo de metodologías confiables de diagnóstico y detección de fallas, se emplean vectores de características, algunos de

alta dimensión a fin de evitar el riesgo de perder información de las señales que sea potencialmente necesaria [62].

Un vector de características se compone de valores numéricos representativos de un objeto en específico, por ejemplo; si el objeto fuera una imagen, los píxeles de la imagen representarían uno de los valores característicos; si el objeto fuera una señal, sus valores característicos podrían ser estadísticos como la desviación estándar, la media, entre otros; valores propios de la señal como su valor cuadrático medio o valores que estén directamente relacionados con la naturaleza de la señal, como los parámetros de complejidad.

Las señales, por su naturaleza, pueden presentar un comportamiento muy aleatorio, mientras que otras parecen demostrar la aparición de ciertos patrones en varios intervalos [63]. Para comparar series de tiempo y distinguir su comportamiento regular, caótico y aleatorio, fueron desarrolladas las medidas de complejidad [64]. De este modo, la evaluación de la complejidad o la variabilidad de una señal, ofrece ciertas ventajas en el procesamiento de señales. Por un lado, la complejidad es una descripción intuitiva y, como consecuencia, facilita la interpretación de los resultados de una medición. Por otro lado, las medidas invariantes permiten la comparación entre diferentes condiciones, ya que son insensibles a medidas absolutas, como la amplitud y la frecuencia [63]. Los principales parámetros de complejidad son: entropía, dimensión fractal y exponente de Lyapunov. Para el caso que ocupa este trabajo, se usa la información de entropía como parámetro de complejidad para la extracción de características relevantes en las señales de vibración de un motor con diferentes condiciones de operación.

3.2.1. Entropía de información

La teoría de la información de Shannon, llamada solamente teoría de la información, fue propuesta en 1948 por C.E. Shannon [65], y es una medida de la cantidad de incertidumbre contenida en cualquier señal aleatoria. En [65], Shannon propuso una medida de la cantidad de información en una distribución, a la que llamó ‘entropía’. La entropía, $H(P)$, de una distribución P mide la incertidumbre inherente en P , es decir, el concepto de entropía se concibe como la cantidad de desorden que contiene o libera un sistema.

En [66] ejemplifican el concepto de la siguiente forma: imagine a un observador que sabe que X está distribuido de acuerdo con P , el observador advierte que $X = x$. La entropía de P representa la incertidumbre del observador sobre el resultado de x antes de que él observe. Ahora, imagine al observador como un receptor del mensaje que transporta el valor de X . Desde estos dos puntos de vista, la entropía representa la cantidad media de información que el observador ha obtenido después de recibir un resultado de x de la variable aleatoria X . Como indican en [67] cuanto mayor sea

la entropía de información, mayor será el grado de desorden de información. De esta forma, se puede hablar de la cantidad de información que lleva una señal y la definición matemática es la siguiente:

Consideremos un evento aleatorio (o variable aleatoria) B con n posibles resultados $B_1, B_2, B_3, \dots, B_i$, y cada B_i con una probabilidad $p(B_i)$ (con $i = 1, 2, 3, \dots, n$), entonces la entropía de información $H(B)$ de un evento aleatorio B esta dada por [65]:

$$H(B) = - \sum_{i=1}^n p(B_i) \log_2(p(B_i)) \quad (3.12)$$

donde, $0 \leq p(B_i) \leq 1$, $\sum_{i=1}^n p(B_i) = 1$.

La expresión definida en (3.12) da una idea de la cantidad de información y mide con mayor precisión la aleatoriedad, por tanto, si un evento tiene probabilidad $p(B_i) = 1$, entonces $H = 0$ lo que refleja que no hay mucha información. A su vez, dicha expresión será aplicada a las funciones modales o IMFs, descritas en la ecuación (3.9), y obtenidas luego de realizar la descomposición mediante CEEMD. De la misma forma, estos valores de entropía, obtenidos de cada IMF, serán los que se utilicen como características para el procedimiento de clasificación e identificación de fallas. Como se observa en (3.12), $H(B)$ depende solamente de la probabilidad del evento y la unidad con la cual se mide la entropía es *bits* cuando se usan logaritmos de base 2, por otro lado, si la entropía se mide usando logaritmos naturales o neperianos, su unidad es el *nat*.

3.3. Clasificadores

Las fallas que se presentan en los motores de inducción generan características específicas en las señales que describen el comportamiento de la máquina como las de voltaje, corriente o vibración. Como se mencionó previamente, en el espectro de las señales de vibración aparecen amplitudes elevadas en frecuencias donde normalmente no aparecerían si el motor estuviera operando bajo condiciones normales de operación o sin fallas. Por tanto, a través de las señales del motor se pueden extraer características relevantes referentes al comportamiento del dispositivo, con las cuales establecer un conjunto de patrones para procedimientos de clasificación, identificación y etiquetado de las fallas.

Una de las partes esenciales en todo procedimiento de detección de fallas es el referente al proceso de clasificación. Para las áreas de aprendizaje de máquina y estadística, la clasificación representa el problema de identificar la categoría a la cual pertenece un nuevo dato, considerando un entrenamiento previo, en el cual la pertenencia de un conjunto de datos en una categoría en específico ya se ha establecido [68]. En términos de aprendizaje de máquina, el proceso de clasificación es considerado

una instancia del aprendizaje supervisado [69]. Para implementar la clasificación se usa un algoritmo conocido como *clasificador*, cuyos componentes, y en general de aprendizaje de máquina, son *los datos de entrenamiento*, correspondientes a un conjunto de datos que se emplean para entrenar un modelo, compuestos de variables de entrada (vector de características) y salida conocida (objetivo o *target*); *los datos de prueba*, que representan los datos utilizados para probar el rendimiento del modelo después de haber sido entrenado; *las características*, son las variables de entrada de datos de entrenamiento que se emplean para entrenar un modelo, por ejemplo, edad, ingresos, monto del préstamo, etc.; *el objetivo*, salida del modelo entrenado, por ejemplo, riesgo de incumplimiento.

De este modo, los clasificadores han sido extensamente usados en diversas áreas de investigación, como procesamiento de imágenes, reconocimiento de patrones o procesamiento de señales. Entre los algoritmos de clasificación más comunes se encuentran los árboles de decisión, las redes neuronales, el método de k -nn (o *k-nearest neighbor*) y las máquinas de soporte vectorial, los cuales se pueden implementar para la clasificación en dos clases, clasificador binario; o para la clasificación en más de dos clases, clasificador multiclase. A continuación, se explicaran a detalle los clasificadores basados en máquinas de soporte vectorial, *AdaBoost* y árbol de decisión.

3.3.1. Máquinas de soporte vectorial

Las máquinas de soporte vectorial (*support vector machines*, SVMs) son un conjunto de algoritmos de aprendizaje supervisado que generan funciones de mapeo de entrada-salida a partir de un conjunto de datos, ya etiquetados, de entrenamiento. Las funciones de mapeo pueden ser de clasificación o de regresión [70]. Los algoritmos fueron introducidos por Vapnik en 1995 [68] para resolver tareas de aprendizaje de máquina tales como regresión y reconocimiento de patrones.

El sistema de aprendizaje de las SVMs emplea un espacio de características de alta dimensión, donde el algoritmo de aprendizaje reconoce patrones complejos a partir de conjuntos de datos y predice las clasificaciones de otros datos no vistos previamente, realizando aprendizaje discriminativo [71] para resolver problemas de clasificación y regresión, por lo tanto, las SVMs están divididas en:

1. Clasificación de soporte vectorial (*support vector classification*, SVC), el cual es usado para describir la clasificación mediante métodos de soporte vectorial.
2. Regresión de soporte vectorial (*support vector regression*, SVR), que se emplea para describir regresión por medio de métodos de soporte vectorial.

Los algoritmos de clasificación que se obtienen a partir de las máquinas de soporte vectorial pertenecen a la familia de clasificadores lineales generalizados, la cual tiene

tanto la capacidad de minimizar el error de clasificación empírico como de maximizar el margen geométrico [72].

En su formulación básica, SVM corresponde a un clasificador binario y lineal. Por un lado, un clasificador binario es aquel que permite distinguir únicamente entre dos clases, por ejemplo, en la Figura 3.3, distinguir entre la clase frutas y verduras de color rojo y la clase frutas y verduras de otros colores. Por otro lado, un clasificador lineal es aquel en el que la frontera de clasificación es una línea recta en un espacio bidimensional y, en general, en un espacio multidimensional, es un hiperplano [73].



Figura 3.3. Ejemplo de clasificación binaria.

Existen procedimientos de clasificación donde se requiere hacer la distinción de los objetos en más de dos clases, es decir, se requiere un clasificador multiclase, para lo cual, la formulación básica de SVM se modifica con la finalidad de extender el clasificador a un problema de clasificación de más de dos clases. También, algunos procedimientos presentan la particularidad de que las clases no pueden separarse de forma perfecta con una frontera lineal y, al igual que con el aspecto anterior, la formulación básica se modifica para tratar con conjuntos de datos que no se puedan separar de forma lineal.

3.3.1.1. Conjuntos linealmente separables

Para el caso en el que se tienen dos clases y se quiere definir una frontera lineal entre ellas, no va a existir una manera única de generar un hiperplano de separación. En el ejemplo de la Figura 3.4a, representación de la que se encuentra en [74], se pueden trazar líneas de separación entre las muestras de la clase uno y las muestras de la clase dos, obedeciendo diferentes criterios, como se observa en la Figura 3.4b, representación de la que se encuentra en [75], donde se trazan diversas líneas para separar las clases. Para construir estas líneas de separación, existen los modelos discriminativos, cuyo objetivo es clasificar las muestras por medio de un conjunto de entrenamiento que sea suficientemente representativo. SVM es un ejemplo de estos modelos discriminativos [73].

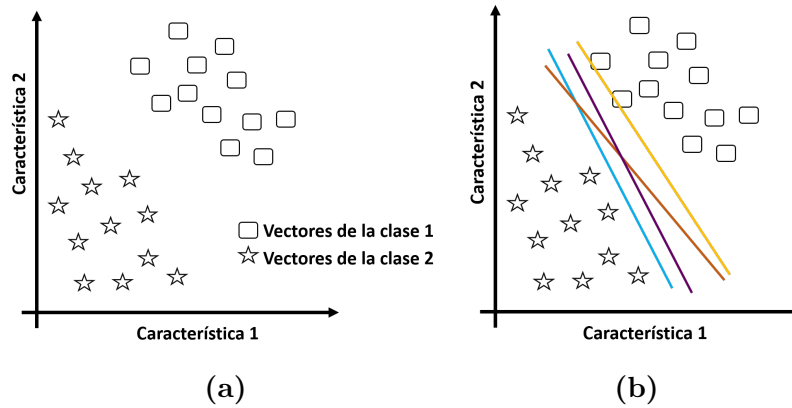


Figura 3.4. a) Espacio de características con dos clases. b) Múltiples líneas de separación entre muestras de dos clases diferentes.

SVM puede describirse como un sistema de clasificación binaria cuya solución es un clasificador lineal. Dentro de los clasificadores lineales, la característica principal de los SVMs es que la solución se basa en lo que se conoce como margen máximo a partir de los vectores de soporte [70]. Resumiendo, como clasificador lineal, la solución del SVM será un hiperplano que divide el espacio en dos regiones y que para dos dimensiones, es una línea recta como se observa en la Figura 3.5, representación de la que se encuentra en [74].

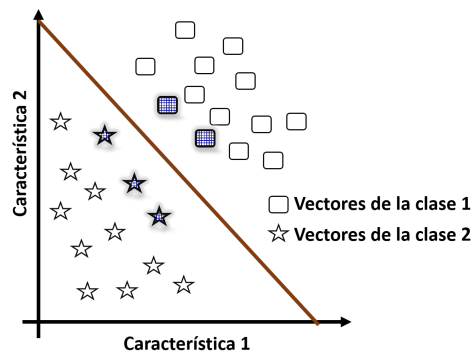


Figura 3.5. Espacio de características con línea recta de separación entre clases.

El modo de encontrar esta línea (o hiperplano) en el SVM es encontrar la frontera a partir de las muestras de entrenamiento. La particularidad específica del SVM es que calcula el hiperplano (la solución) tomando en cuenta, solamente, un número limitado de muestras del conjunto de entrenamiento con unas propiedades concretas. A estas muestras, de gran importancia, se les denomina *vectores de soporte* [70]. Por tanto, cada clase tendrá un conjunto de vectores de soporte, los cuales se muestran en el ejemplo de la Figura 3.6, representación de la que se encuentra en [74].

Los vectores de soporte se eligen de manera que la distancia entre los planos que los contienen, sea máxima. A esta distancia se le conoce con el nombre de *margen*. Con esta condición de *margen máximo*, se logra definir la región más amplia del espacio de características que separa las dos clases y que, además, esta carente de muestras [76].

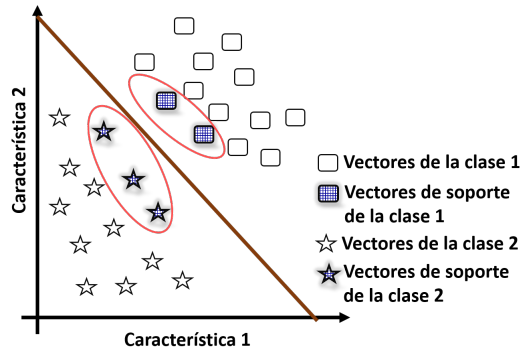


Figura 3.6. Vectores de soporte de cada clase.

Por lo tanto, la región de separación queda definida por los dos hiperplanos que contienen a los vectores de soporte, ilustrados en la Figura 3.7, representación de la que se encuentra en [75]. Para lo cual, el plano intermedio de la región representa la solución del clasificador SVM y que, en términos matemáticos, se plantea como un problema de optimización donde la función a optimizar es el margen.

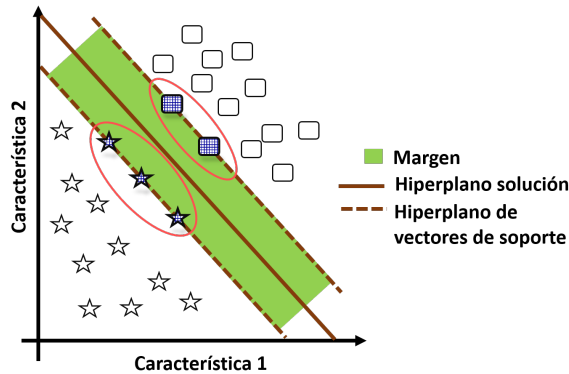


Figura 3.7. Margen de separación entre clases.

A través de la formulación del problema de optimización, se obtiene el máximo margen de seguridad para la clasificación de nuevas muestras que no pertenecen al conjunto de entrenamiento del clasificador. Por esta razón, cualquier elección alternativa de los vectores de soporte y, por lo tanto, cualquier solución alternativa al hiperplano de separación entre clases da como resultados un margen más estrecho [70], lo cual no

representa una solución óptima ya que se tendría un margen de seguridad menor en la clasificación de muestras que se encuentren próximas a la frontera entre clases.

La solución, basada en el uso de vectores de soporte, proporciona a SVM algunas propiedades importantes:

- * Por un lado, debido a que el hiperplano solución depende, únicamente, de la elección de los vectores de soporte, si se desplazan las muestras asociadas a los vectores de soporte, se modifica la solución del SVM, como se puede ver en la Figura 3.8 [70].

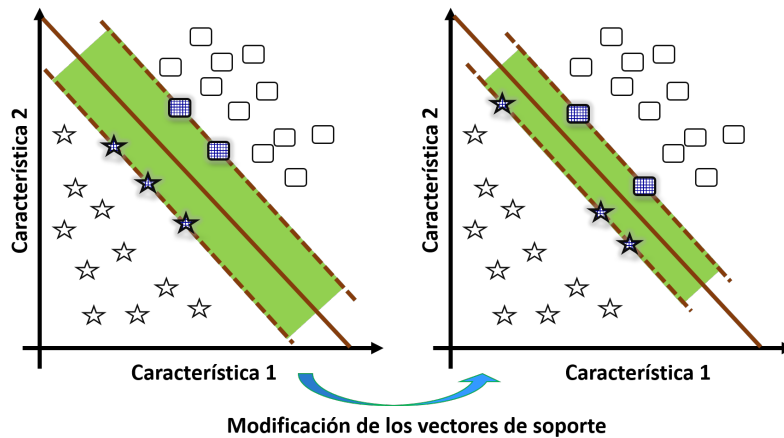


Figura 3.8. *Modificación de los vectores de soporte.*

- * Por otro lado, y debido a que las muestras que no son vectores de soporte no intervienen en la definición de la frontera de separación, si se desplaza cualquier muestra que no sea un vector de soporte, la solución del SVM permanece inalterada, como se observa en la Figura 3.9 [70].

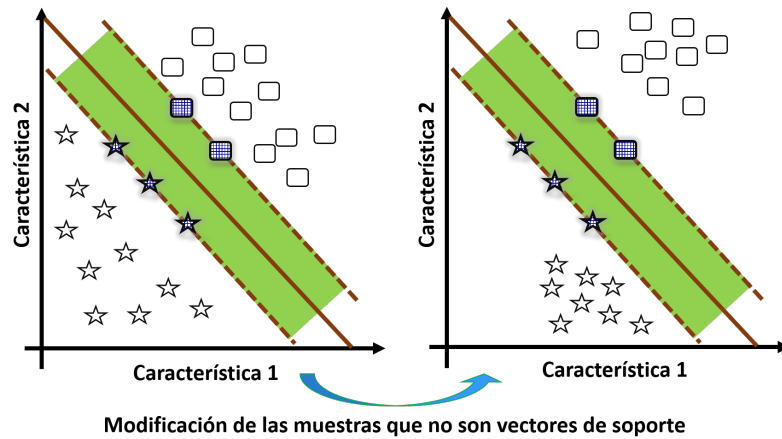


Figura 3.9. *Modificación de las muestras que no son vectores de soporte.*

Estas dos características proporcionan al SVM cierta robustez estadística y, particularmente, lo hacen robusto a lo que se conoce como *overfitting*, que es una condición con la cual se pierde la capacidad de generalización, para otros datos, debido a un ajuste elevado a la muestras de entrenamiento. Sin embargo, el SVM generaliza de una manera robusta sin que se vea influido por *outliers* que se encuentren fuera de la zona donde están definidos los vectores de soporte [73].

Esta explicación del modelo básico del SVM como clasificador lineal sirve, únicamente, para clasificar conjuntos que sean linealmente separables y también completamente separables, es decir, sin solapamiento entre las clases. A continuación se explica el modelo para el caso de clasificaciones entre clases del tipo que se muestra en la Figura 3.10, representación de la que se encuentra en [77].

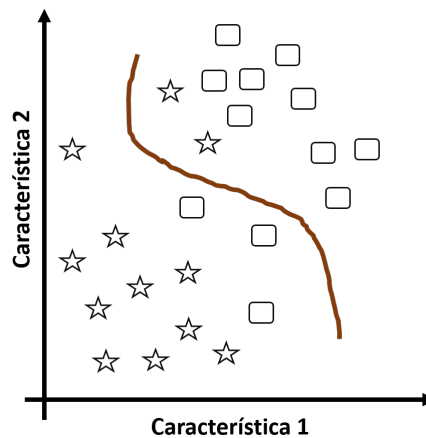


Figura 3.10. *Conjuntos no linealmente separables.*

3.3.1.2. Conjuntos no linealmente separables

Para que el clasificador SVM pueda ser eficiente en problemas donde los datos no sean completamente separables o linealmente separables (los más habituales en conjuntos de datos reales), existen dos alternativas [73]:

- * Por un lado, permitir un cierto solapamiento entre las clases a partir de relajar la condición de margen máximo. Procedimiento que se conoce como margen suave (*soft margin*).
- * Por otro lado, extender el modelo a conjuntos que no sean linealmente separables mediante la transformación del espacio de características original en otro espacio que sí sea linealmente separable. Proceso que se conoce como truco del kernel (*kernel trick*).

Margen suave

Esta relajación de la condición del margen implica que algunas muestras caigan dentro de la zona del margen, es decir, que queden erróneamente clasificadas. En la Figura 3.11, representación de la que se encuentra en [78], se ilustra un ejemplo de esta alternativa [73].

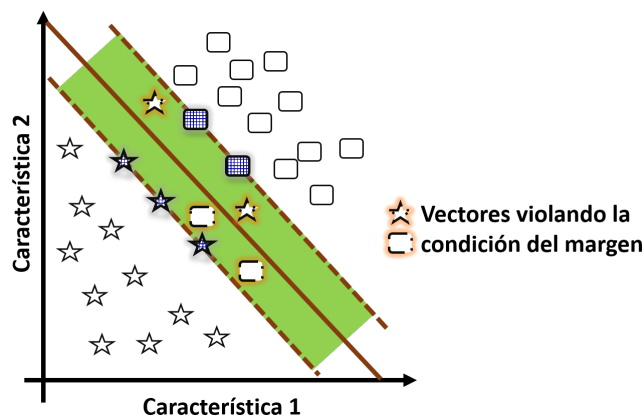


Figura 3.11. Muestras erróneamente clasificadas.

La ventaja que otorga este procedimiento es que el SVM se hace robusto al ruido asociado a las muestras y va a permitir tener cierta tolerancia a errores, de manera que se puede gestionar el problema de solapamiento de clases y trabajar con conjuntos que no sean separables. Esta tolerancia a errores, matemáticamente, se gestiona mediante lo que se conoce como *slack variables* [71] y lleva consigo un factor de regularización que

controla la compensación entre errores de entrenamiento y los márgenes rígidos, creando así un margen blando (*soft margin*) que permita algunos errores en la clasificación a la vez que los penaliza.

Truco del kernel

Para el caso de conjuntos que no sean linealmente separables, el esquema de SVM se aplica realizando un mapeo del conjunto de características desde el espacio original, en el que los datos no son linealmente separables, a un espacio de dimensión superior, en el que los datos sí van a ser linealmente separables. En la Figura 3.12, representación de la que se encuentra en [79], se ilustra un ejemplo de dos clases que, en una dimensión no son separables de forma lineal, pero que cuando se transforma a un espacio de dos dimensiones se vuelve un problema que sí es linealmente separable y, por tanto, se puede resolver utilizando SVM de forma estándar [73]. La ventaja con este procedimiento es que no es necesario encontrar la transformación de un espacio a otro, ya que la formulación matemática del SVM facilita los procedimientos por medio del producto escalar y el truco del kernel.

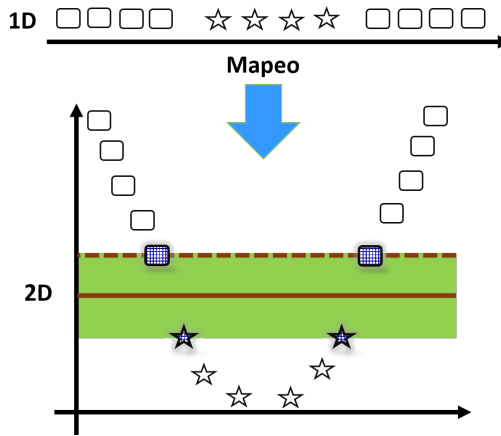


Figura 3.12. Transformación del espacio de características.

3.3.1.3. Formulación matemática

Para un conjunto de características, representativas de una señal y agrupadas en un vector denotado por $x_i \in R^n, i = 1 \dots M$, donde M es el número de muestras de entrenamiento y n es el número de características extraídas, la solución al hiperplano de separación entre clases, para un clasificador lineal SVM, se expresa de la siguiente manera [70]:

$$w^T \cdot x_i + b = 0 \tag{3.13}$$

donde, w es el vector ortogonal al hiperplano y b representa el coeficiente de intersección o sesgo de desplazamiento.

A partir de este hiperplano solución, en un clasificador que sea lineal y binario, las muestras positivas van a ser aquellas que, cuando se aplica la formulación del hiperplano se obtiene un valor mayor que cero ($w^T \cdot x_i + b > 0$); mientras que, para las muestras negativas, se obtiene un valor menor que cero ($w^T \cdot x_i + b < 0$), como se ilustra en la Figura 3.13. De esta manera, la función de clasificación se puede expresar mediante (3.14), dependiendo del signo al aplicar el hiperplano a las muestras [70].

$$f(x) = \text{sgn}(w^T \cdot x_i + b) \quad (3.14)$$

En el caso de SVM, este hiperplano solución corresponde al hiperplano medio entre los dos hiperplanos que contienen los vectores de soporte, denotados por h^+ y h^- . Estos dos hiperplanos se definen en (3.15) [76], como hiperplanos paralelos al hiperplano solución, lo cual se ilustra en la Figura 3.13, representación de la que se encuentra en [70].

$$\begin{aligned} h^+ &\rightarrow w^T \cdot x_i + b = +1 \\ h^- &\rightarrow w^T \cdot x_i + b = -1 \end{aligned} \quad (3.15)$$

En este caso, $+1$ y -1 es de elección arbitraria, lo cual mantiene la generalidad y para cualquier otra opción genera el mismo resultado.

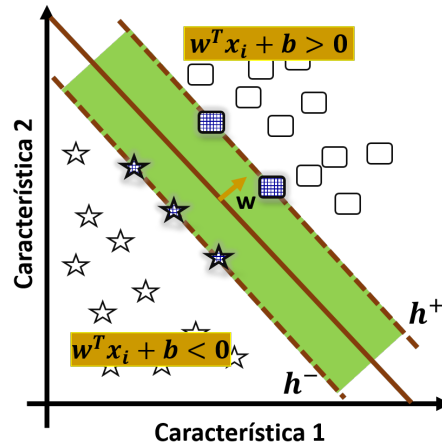


Figura 3.13. Función de clasificación.

Seguidamente, la condición de clasificación, en el caso del SVM, va a implicar que todas las muestras que estén correctamente clasificadas van a ubicarse en la zona más allá del margen de cada clase, representada por las áreas en gris de la Figura 3.14. Por lo tanto, todas las muestras positivas deben cumplir la condición que se indica en (3.16a), y todas las muestras negativas deben cumplir la que se indica en (3.16b) [70].

$$w^T \cdot x_i + b \geq +1 \quad (3.16a)$$

$$w^T \cdot x_i + b \leq -1 \quad (3.16b)$$

Asumiendo que las muestras positivas se identifican con $+1$ y las negativas con -1 , las condiciones de clasificación de (3.16a) y (3.16b) se pueden integrar en una única condición de clasificación, la cual tienen que cumplir todas las muestras de entrenamiento y se expresa de la siguiente forma [70]:

$$y_i(w^T \cdot x_i + b) \geq 1 \quad (3.17)$$

donde, $y_i = \pm 1$ corresponde al valor de salida o categorías asociadas de x_i .

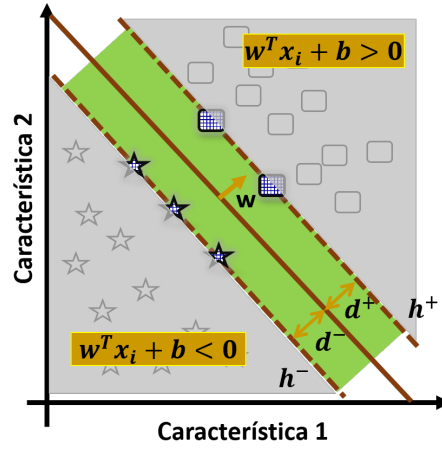


Figura 3.14. Condiciones de clasificación.

Así, el margen, que está delimitado por la región verde en la Figura 3.14, se puede calcular a partir de la distancia entre los dos planos, h^+ y h^- , del siguiente modo:

$$d^+ = d^- = \frac{|wx + b|}{\|w\|} = \frac{1}{\|w\|} \quad (3.18)$$

y, por consiguiente, el margen viene dado por la suma de las distancias d^+ y d^- , así [70]:

$$margen = d^+ + d^- = 2 \frac{1}{\|w\|} \quad (3.19)$$

y depende únicamente de w .

Como se mencionó anteriormente, en un clasificador SVM lineal, el mejor hiperplano es aquel que tiene el mayor margen de separación entre las clases. A fin de encontrar el mejor hiperplano para la clasificación, el margen, representado en (3.19), se debe maximizar o, lo que es equivalente, se debe minimizar el problema inverso [80], representado por la siguiente función:

$$\Phi(w) = \frac{1}{2} \|w\|^2 \quad (3.20)$$

la cual es proporcional al producto escalar $\|w\|^2 = w^T w$.

Además, esta minimización estará sujeta a la condición de clasificación que tienen que cumplir todas las muestras de entrenamiento y que está definida en (3.17). Como resultado de esto, se genera un problema de optimización cuadrática que se resuelve de manera estándar y se plantea de la siguiente forma [73]:

$$\begin{aligned} & \underset{w,b}{\text{Minimizar}} \quad \Phi(w) = \frac{1}{2} \|w\|^2 \\ & \text{Sujeto a} \quad y_i(w^T \cdot x_i + b) \geq 1, \quad i = 1, \dots, M \end{aligned} \quad (3.21)$$

Desarrollo matemático del problema de optimización cuadrática de SVM lineal

Como ya se ha visto, la solución del SVM consiste en obtener los valores w y b del hiperplano solución, representado por (3.13), a través del problema de optimización descrito en (3.21). El anterior, es un problema clásico de optimización de programación cuadrática (*Quadratic Programming*, QP, *optimization*) con restricciones de desigualdad, que se resuelve por medio de los puntos de inflexión de la función auxiliar conocida como Lagrangiano (L) [70], cuya formulación general es la siguiente:

$$L(X, \alpha) = f(X) + \sum_j \alpha_j g(X) \quad \forall \alpha_j \geq 0 \quad (3.22)$$

donde, $f(X)$ es la función que se quiere optimizar, $g(X)$ representa las restricciones a las cuales está sujeta la optimización y α_j son los multiplicadores de Lagrange. De esta forma, haciendo la equivalencia de estos parámetros con los del problema de optimización de (3.21) y recordando que , se tiene que:

$$\begin{aligned} f(X) & \rightarrow \Phi(w) \\ g(X) & \rightarrow y_i(w^T \cdot x_i + b) - 1 \geq 0 \end{aligned} \quad (3.23)$$

Con esta transformación, la solución al problema de optimización se obtiene, de manera general, primero, minimizando el Lagrangiano con respecto a las variables de la función (3.13), es decir de w y b y, después, maximizando con respecto a los multiplicadores α_j [70], de la siguiente forma:

$$\left. \begin{aligned} & \underset{w,b}{\text{Minimizar}} \quad \Phi(w) = \frac{1}{2} w^T w \\ & \text{Sujeto a} \quad y_i(w^T \cdot x_i + b) \geq 1 \end{aligned} \right\} \Rightarrow \max_{\alpha} (\min_{w,b}(\Phi(w))) \quad (3.24)$$

formulación que recibe el nombre de SVM primal, por estar en el espacio de los parámetros w y b (*espacio primal*).

Luego, el Lagrangiano se crea, siguiendo las equivalencias de (3.23), a partir de la función $\Phi(w)$ a minimizar y de la condición de clasificación $y_i(w^T \cdot x_i + b) - 1 \geq 0$ multiplicada por los coeficientes de Lagrange (α_j), con lo cual se obtiene la siguiente expresión [70]:

$$L(w, b, \alpha) = \frac{1}{2}w^T w - \sum_{i=1}^M \alpha_i [y_i(w^T x_i + b) - 1] \quad (3.25)$$

Por otra parte, la minimización del Lagrangiano y la búsqueda de los puntos de inflexión de la función implica que [70]:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^M \alpha_i y_i x_i = 0 \Rightarrow w = \sum_{i=1}^M \alpha_i y_i x_i \quad (3.26)$$

$$\frac{\partial L}{\partial b} = - \sum_{i=1}^M \alpha_i y_i = 0 \Rightarrow \sum_{i=1}^M \alpha_i y_i = 0 \quad \alpha_i \geq 0 \quad (3.27)$$

Seguidamente, bajo condiciones de convexidad, se puede incluir en la función (3.25) el valor obtenido para w en (3.26), con lo que se obtiene una nueva expresión para el Lagrangiano que, luego de desarrollar matemáticamente, queda de la siguiente forma:

$$\begin{aligned} L(w, b, \alpha) &= -\frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_i \alpha_i \\ &= \sum_{i=1}^M \alpha_i - \frac{1}{2} \sum_{i,j=1}^M y_i y_j \alpha_i \alpha_j x_i^T x_j \end{aligned} \quad (3.28)$$

y recibe el nombre de Lagrangiano dual ($L_d(\alpha)$), ya que su formulación está en el espacio de los multiplicadores de Lagrange, el cual se conoce como *espacio dual* [70].

Para encontrar el mejor hiperplano de separación entre clases, $L_d(\alpha)$ debe ser maximizado con respecto a valores α_i positivos y con respecto a la restricción de igualdad de (3.27), con lo cual se genera un nuevo problema de optimización, conocido como SVM dual y que tiene la siguiente formulación [70]:

$$\begin{aligned} \text{Maximizar}_{\alpha} \quad \Theta(\alpha) &= \sum_{i=1}^M \alpha_i - \frac{1}{2} \sum_{i,j=1}^M y_i y_j \alpha_i \alpha_j x_i^T x_j \\ \text{Sujeto a} \quad \sum_{i=1}^M \alpha_i y_i &= 0, \quad \alpha_i \geq 0 \end{aligned} \quad (3.29)$$

Finalmente, y a modo de resumen, a partir del SVM original (SVM primal), representado en (3.24), se obtiene un nuevo problema de optimización (SVM dual) cuya solución, por un lado, se consigue empleando algún método numérico estándar y, por otro lado, representa la solución del SVM. A su vez, por medio de esta solución se obtienen los valores óptimos de los multiplicadores α_i que, como se ha visto, mediante (3.26) y (3.13), permiten encontrar la formulación del hiperplano solución (w y b , respectivamente) y, por lo tanto, definir la función de clasificación ya sea, en forma directa, a partir de los parámetros del hiperplano (3.30a) o, en forma indirecta, a partir de los multiplicadores α_i y los vectores de soporte (3.30b).

$$f(x) = \text{sgn}(w^T \cdot x + b) \quad (3.30a)$$

$$f(x) = \text{sgn} \left(\sum_{i=1}^M y_i \alpha_i x^T x_i + b \right) \quad (3.30b)$$

A partir de la formulación del SVM, se extraen las siguientes propiedades:

1. La solución de w viene dada como una combinación lineal de las muestras de entrenamiento (3.26). Por lo tanto, y debido a que los multiplicadores de Lagrange (α_i) son todos positivos o nulos, la solución para el hiperplano (w) va a ser la combinación lineal, únicamente, de aquellas muestras que tengan asociado un multiplicador $\alpha_i > 0$, siendo estas muestras de entrenamiento las que definen los vectores de soporte. O, dicho de otra forma, aquellas muestras de entrenamiento que no tengan un $\alpha_i = 0$ serán los vectores de soporte [70].
2. En la formulación dual se puede observar que las muestras; tanto en la formulación del problema de optimización (3.29), como en la función de clasificación (3.30b); aparecen, únicamente, como productos escalares. Esta propiedad es la que se utiliza para adaptar el problema al caso de conjuntos no linealmente separables, mediante el uso del truco del kernel [73].
3. La solución del SVM es no paramétrica, es decir, no hay necesidad de ajustar ningún parámetro durante el entrenamiento. Sin embargo, en los casos donde los conjuntos no son linealmente separables y donde se quiera tener tolerancia a errores de clasificación, sí es necesario realizar el ajuste de algunos parámetros [73].

Caso no linealmente separable. El truco del kernel

Mediante el algoritmo original de hiperplano de margen máximo presentado por Vapnik en 1963 [81], se construye un clasificador lineal binario. Sin embargo, normalmente, los datos representativos de los casos de estudio reales no se presentan en dos dimensiones, por lo tanto, un algoritmo de clasificación SVM debe tratar con

problemas donde existen más de dos variables predictoras, curvas de separación no lineales, conjuntos de datos que no pueden ser completamente separados o clasificaciones en más de dos dimensiones. Es por ello que, en 1992, fue propuesto por Boser *et. al* un procedimiento para crear clasificadores no lineales [82] que se aplica al método del hiperplano de margen máximo y recibe el nombre de *kernel trick*.

Con este nuevo procedimiento, el algoritmo resultante es, formalmente, similar al que se emplea para tratar conjuntos linealmente separables. La estrategia consiste en utilizar una función de mapeo, identificada como $\varphi(x)$, para aumentar la dimensionalidad del espacio de características con la esperanza de que, en este nuevo espacio de características, las clases sean linealmente separables. Para lograrlo, se realiza un mapeo de los vectores de entrada $x \in R^n$ en vectores $\varphi(x)$ pertenecientes a un espacio de características, F , de mayor dimensión (donde φ representa el mapeado: $R^n \rightarrow R^f$). Este nuevo espacio de características se representa de la siguiente forma [70]:

$$x \in R^n \rightarrow \varphi(x) = [\phi_1(x), \phi_2(x), \dots, \phi_n(x)]^T \in R^f, \quad (3.31)$$

Por medio de este mapeo, se espera que, en un espacio φ , el algoritmo de aprendizaje sea capaz de separar linealmente los datos de x mediante la aplicación de la formulación de SVM presentada previamente. También, se espera que este planteamiento vuelva a conducir a la solución del problema de optimización cuadrática con restricciones similares en un espacio φ [70].

Para llevar a cabo este procedimiento, debido a que en la formulación del SVM dual (3.29) los vectores de datos aparecen únicamente como productos escalares ($x_i^T x_j$), será suficiente definir una función kernel ($K(x, z)$) que represente el producto escalar en el espacio mapeado por $\varphi(x)$. De esta forma, no va a ser necesario definir explícitamente la función $\varphi(x)$, sino que será suficiente con encontrar esta función kernel, representada en (3.32) [70], entre dos muestras.

$$K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j) = [\phi_1(x_i), \phi_2(x_i), \dots, \phi_n(x_i)]^T [\phi_1(x_j), \phi_2(x_j), \dots, \phi_n(x_j)] \quad (3.32)$$

Debido a que esta función kernel representa un producto escalar, se puede integrar de forma directa en la solución del SVM (3.29) sustituyendo a los productos escalares entre las muestras en la formulación original, tanto en la función a optimizar como en la función de clasificación. Por consiguiente, la formulación queda de la siguiente forma:

$$\underset{\alpha}{\text{Maximizar}} \quad \Theta(\alpha) = \sum_{i=1}^M \alpha_i - \frac{1}{2} \sum_{i,j=1}^M y_i y_j \alpha_i \alpha_j K(x_i, x_j) \quad (3.33a)$$

$$b = y_i - w^T \varphi(x_i) = y_i - \sum_{i,j=1}^M y_i \alpha_i K(x_i, x_j) \quad (3.33b)$$

$$f(x) = \operatorname{sgn} \left(\sum_{i=1}^M y_i \alpha_i K(x_i, x_j) + b \right) \quad (3.33c)$$

Existen diferentes ejemplos de funciones kernel que en la práctica resultan de gran utilidad, de forma que, no hay que buscar una función de kernel para cada problema distinto. Entre las funciones kernel más habituales, destacan las que se muestran a continuación, en la Tabla 3.1, extracto de la que se presenta en [70].

Tabla 3.1. Tipos de funciones Kernel

Función Kernel	Tipo de clasificador
$K(x_i, x_j) = x_i^T x_j$	Lineal.
$K(x_i, x_j) = e^{-\ x_i - x_j\ ^2}$	Función de base radial (Gaussiano).
$K(x_i, x_j) = (x_i^T x_j + 1)^d$	Polinomial completo de grado d entero positivo.
$K(x_i, x_j) = \tanh(p_1 x_i^T x_j + p_2)$	Sigmoide o perceptron multicapa con $p_1 > 0$ y $p_2 < 0$.
$K(x_i, x_j) = \frac{1}{\sqrt{\ x_i - x_j\ ^2 + \beta}}$	Función multicuadrática inversa.

De las funciones presentadas, se puede observar que la función de kernel define una función de similitud entre dos muestras. Siendo éste el valor que se requiere tanto para la solución del SVM (3.33a), como para después aplicarlo en la función de clasificación (3.33c). Por otro lado, las funciones kernel introducen nuevos parámetros, como p_1 , p_2 y β , cuyos valores óptimos se encuentran durante el aprendizaje del clasificador utilizando validación cruzada.

3.3.2. AdaBoost (adaptive boosting)

AdaBoost, o *adaptive boosting*, consiste en un algoritmo de aprendizaje de máquina que tiene como principio de operación un método general denominado *Boosting*, mediante el cual se mejora la precisión de cualquier algoritmo de aprendizaje por medio de la creación de un clasificador fuerte a partir de una serie de clasificadores débiles [83]. Para esto, se realiza la construcción de un modelo a partir de los datos de entrenamiento, luego se crea un segundo modelo que intenta corregir los errores del primer modelo. Finalmente, ambos modelos se unen hasta que el conjunto de entrenamiento se predice perfectamente o un número máximo de modelos han sido agregados [84].

Boosting se basa en que encontrar muchas reglas aproximadas puede ser mucho más fácil que encontrar una única regla de predicción muy precisa. Para aplicar *Boosting* se debe iniciar con un método o algoritmo mediante el cual encontrar dichas reglas

aproximadas. El algoritmo basado en *Boosting* hace un llamado de estos algoritmos de aprendizaje “débiles” o “de base” en repetidas ocasiones, siempre que se le da un subconjunto diferente de los ejemplos de entrenamiento. Cada vez que es llamado, el algoritmo de aprendizaje de base genera una nueva regla de predicción débil, y luego de varias rondas, el algoritmo *Boosting* debe combinar estas reglas débiles en una única regla de predicción que, se espera, será mucho más precisa que cualquier otra de las reglas débiles [85].

Tal y como se ha mencionado, se requiere de un algoritmo que esté basado en *Boosting* para implementar el proceso descrito anteriormente. Se han propuestos varios, entre ellos, el de mayor renombre es *AdaBoost*, propuesto por Schapire y Freund en 1995 [83], como un método general para generar un clasificador “fuerte” a partir de un conjunto de clasificadores “débiles”. Este algoritmo puede ser utilizado para reducir significativamente el error de cualquier algoritmo de aprendizaje que, consistentemente, genera clasificadores cuyo rendimiento es un poco mejor que la predicción aleatoria [84].

La premisa bajo la cual fue propuesto *AdaBoost* es que “dos cabezas son mejores que una”, de modo que, la esencia del aprendizaje combinado y el viejo dicho es coincidente ya que producir una gran cantidad de clasificadores “débiles” y combinarlos bajo ciertas reglas para obtener un único clasificador “fuerte”, generará excelentes resultados de clasificación [86]. De este modo, *AdaBoost* es considerado el clasificador de mayor éxito en el aprendizaje combinado o de conjunto y corresponde a la técnica del algoritmo de aprendizaje de máquina en la cual el “clasificador débil” es aprendiz, y cuyos resultados de la clasificación serán cercanos a los de las etiquetas reales, mientras que un “clasificador fuerte” contendrá resultados cuya coincidencia, con las etiquetas reales, será perfecta [87].

De este modo, mediante *AdaBoost*, los clasificadores débiles son seleccionados de forma iterativa a partir de un número de clasificadores débiles candidatos, posteriormente son combinados de forma lineal para formar un clasificador fuerte. Para ello, el conjunto de clasificadores débiles se expresa de la siguiente forma [88]:

$$H = \{\tilde{h}_f\} \quad (3.34)$$

Por otro lado, el conjunto de datos de la muestra de entrenamiento se expresa así:

$$\{(x_1, y_1), \dots, (x_i, y_i), \dots, (x_n, y_n)\} \quad (3.35)$$

donde, x_i representa el i -ésimo vector de características; $y_i \in \{+1, -1\}$ es la etiqueta del i -ésimo vector de características, que indica si el vector de características presenta un comportamiento u otro; y n es el tamaño del conjunto de datos.

Por otra parte, $\{w_1, \dots, w_i, \dots, w_n\}$ corresponde a los pesos de las muestras que indican el grado de importancia de las muestras y, en términos estadísticos, representan

una estimación de la distribución de la muestra. Seguidamente, el procedimiento de *AdaBoost*, expuesto en [88], se describe de la siguiente forma:

1. Inicializar los pesos, $w_i(1) = 1/n$ ($i = 1, \dots, n$) satisfaciendo la condición $\sum_{i=1}^n w_i(1) = 1$.
2. Examinar para $t = 1, \dots, T$ lo siguiente:
 - a. Establecer ε_j como la suma de los errores de clasificación ponderados para los clasificadores débiles h_j .

$$\varepsilon_j = \sum_{i=1}^n w_i(t)[y_i \neq h_j(x_i)] \quad (3.36)$$

Escoger, a partir de los clasificadores débiles construidos, el clasificador débil, $h(t)$, que minimiza la suma de los errores de clasificación ponderados, del siguiente modo:

$$h(t) = \arg \min_{h_j \in H} \varepsilon_j \quad (3.37)$$

- b. Calcular la suma de los errores de clasificación ponderados, $\varepsilon(t)$, para el clasificador débil, $h(t)$, que ha sido escogido.

- c. Establecer

$$\alpha(t) = \frac{1}{2} \log \left(\frac{1 - \varepsilon(t)}{\varepsilon(t)} \right) \quad (3.38)$$

- d. Actualizar los pesos de la siguiente forma:

$$w_i(t+1) = \frac{w_i(t) \exp(-\alpha(t)y_i h(t)(x_i))}{Z(t)} \quad (3.39)$$

donde, $Z(t)$ corresponde a un factor de normalización que se define así:

$$Z(t) = \sum_{k=1}^n w_k(t) \exp(-\alpha(t)y_k h(t)(x_k)) \quad (3.40)$$

3. Finalmente, el clasificador fuerte se define por:

$$H(x) = \operatorname{sgn} \left(\sum_{t=1}^T \alpha(t) h(t)(x) \right) \quad (3.41)$$

De este procedimiento, en el paso dos, se puede observar que los pasos siguientes se ejecutarán por $t = 1, \dots, T$, siendo T el número de ciclos de aprendizaje para los clasificadores débiles.

Finalmente, entre las ventajas que presenta *AdaBoost* como procedimiento de clasificación se pueden mencionar [88]:

- * Corrige las clasificaciones erróneas que realizan los clasificadores débiles, por lo tanto, es menos susceptible a *overfitting* comparado con la mayoría de los algoritmos de aprendizaje.
- * Existen conjuntos de datos reales que pueden presentar diversos tipos de características, lo cual dificulta encontrar relaciones entre ellas. Es por ello que, mediante la combinación de clasificadores débiles, aplicada a diversos tipos de características, en la construcción de clasificadores fuertes, hace que la relación entre las características diversas sea manejada de forma natural sin necesidad de conversiones entre éstas.
- * Si se usan clasificadores débiles simples, *AdaBoost* resultará un algoritmo rápido. Lo anterior lo convierte en uno de los algoritmos más populares en el área de reconocimiento de patrones.

3.3.3. Árboles de decisión

Los árboles de decisión corresponden a un modelo predictivo que se emplea para el aprendizaje de máquina con el objetivo de predecir el valor de una variable destino en función de diversas variables de entrada [89]. A su vez, se refiere tanto a un modelo concreto de decisión, utilizado para apoyar la toma de decisiones, como a un método para construir dichos modelos automáticamente a partir de datos. Por un lado, como modelo, los árboles de decisión se refieren a una estructura concreta de información o conocimiento, que sirve para dar soporte en la toma de decisiones en tareas de clasificación y regresión. Por otro lado, como método, comprende varias técnicas para construir modelos de árboles de decisión de forma automatizada empleando algoritmos y medidas de diversas áreas, como la estadística, el aprendizaje de máquina y la inteligencia artificial [90].

Debido a las ventajas sustanciales de flexibilidad, intuición, sencillez y eficiencia computacional que presentan, las técnicas de aprendizaje basadas en árboles de decisión se han utilizado con éxito para resolver una amplia gama de problemas de clasificación. Según Friedl y Brodley en [91], los árboles de decisión se definen como un procedimiento de clasificación que divide, de forma recursiva, un conjunto de datos (o espacio de características) en subdivisiones más pequeñas basándose en un conjunto de pruebas definidas.

La estructura representativa de los árboles de decisión está compuesta de *nodos* y *ramas* que conectan los nodos [92]. Se distinguen dos tipos de nodos básicos: nodos hoja y nodos no- hoja (el *nodo raíz* es un nodo no- hoja especial). Por un lado, cada nodo no- hoja, también conocido como nodo terminal, es etiquetado con un atributo o una pregunta y las ramas que emanan de este tipo de nodo corresponden a los posibles

valores del atributo o las respuestas a la pregunta. Por otro lado, los nodos hoja son etiquetados con una clase o categoría [90].

En la Figura 3.15, de [90], se ilustra una estructura de un árbol de decisión basado en un ejemplo simple de asignación de título al nombre, el cual, por un lado, contiene tres nodos hoja con las etiquetas *Señor*, *Señora* y *Señorita*; y dos nodos no-hoja identificados con los atributos *Género* (éste es el nodo raíz) y *Estado civil*. Por otro lado, las ramas son etiquetadas con los valores de los atributos correspondientes: *masculino* y *femenino* (para el atributo *Género*); *casado* y *no casado* (para el atributo *Estado civil*).

Cabe destacar que, un árbol de decisión cuyos nodos hoja son identificados con etiquetas de clase discretas se denomina árbol de clasificación. Caso contrario, si se emplean valores continuos o rangos de valores, se denomina árbol de regresión [90]. A su vez, una estructura de árbol de decisión representa un grafo acíclico dirigido que satisface las siguientes propiedades:

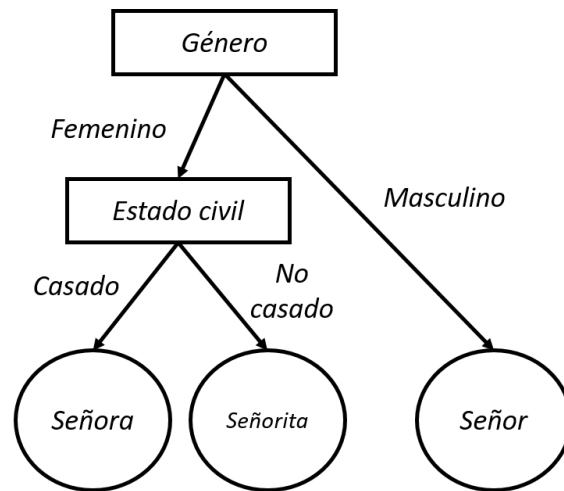
1. Hay exactamente un *nodo*, denominado la *raíz*, en el cual no entran bordes (ramas).
2. Cada *nodo*, diferente de la *raíz*, tiene exactamente un borde entrante (rama).
3. Existe un único camino desde la *raíz* a cada uno de los otros nodos.

También, cada camino desde el nodo raíz a un nodo hoja puede ser interpretado como una regla de decisión, la cual contiene una condición y su parte de conclusión correspondiente. Ésto puede ser expresado mediante la notación *IF-THEN* o el símbolo para implicación lógica, de la siguiente forma [90]:

$$\begin{array}{l} \text{IF } \textit{condición} \text{ THEN } \textit{conclusión} \\ \textit{condición} \Rightarrow \textit{conclusión} \end{array}$$

Para lo cual, si la información de entrada cumple con todas las condiciones descritas, se afirma la conclusión correspondiente. Por lo tanto, la estructura de decisión se puede formular como un conjunto de reglas de decisión, como se ilustra en la Figura 3.15, de [90].

Es importante mencionar que, normalmente, los árboles de decisión se visualizan con el nodo raíz en la parte superior y los nodos-hoja en la parte inferior, como se ilustra en la Figura 3.15. Sin embargo, es posible encontrar estructuras con el nodo raíz en la parte inferior del diagrama y los nodos-hoja en la parte superior.



Reglas de decisión correspondientes:

R_1 : IF Masculino THEN Señor

R_2 : IF Femenino AND Casado THEN Señora

R_3 : IF Femenino AND No casado THEN Señorita

Figura 3.15. *Árbol de decisión sencillo que facilita la tarea de clasificación para asignación de título al nombre [83].*

3.3.3.1. Construcción de los árboles de decisión

Existen dos enfoques básico para la construcción de árboles de decisión. En primer lugar, se pueden generar de forma manual, pero éste es un procedimiento que requiere tiempo y un esfuerzo considerable. En segundo lugar, los árboles de decisión se pueden construir, automáticamente, a partir de los datos disponibles mediante algoritmos adecuados de aprendizaje de máquina, para lo cual se requieren de dos pasos básicos, que son:

1. Obtener los datos del problema que involucra la toma de decisiones o fenómeno estudiado.
2. Deducir el árbol de decisión (o el equivalente conjunto de reglas de decisión) mediante un proceso inductivo que generaliza a partir de los datos.

Para el proceso de construcción del árbol de decisión, el algoritmo de aprendizaje determina, a partir de los datos o instancias, qué atributo debe colocarse en el nodo raíz, con lo cual proporciona una idea del atributo que tiene la influencia más fuerte, respecto a los datos o instancias subyacentes. De este modo, el aprendizaje por árboles de decisión

generaliza a partir de las observaciones, mediante un proceso de inducción que toma como entrada un conjunto de observaciones específicas o instancias, y genera sobre ellas, reglas generales. Para facilitar el aprendizaje automatizado, las observaciones o instancias de aprendizaje (denominadas conjunto de entrenamiento) suelen expresarse en una tabla, donde una fila representa una instancia y una columna representa, o un atributo y sus valores, o las etiquetas de clase [90].

Haciendo uso de las instancias de entrenamiento como entrada, el algoritmo de aprendizaje genera un modelo de árboles de decisión por medio de la división recursiva de las instancias a través de los siguientes pasos principales [90]:

1. *Inicializar*. Proporciona el conjunto completo de observaciones (conjunto de entrenamiento) como entrada al algoritmo de aprendizaje del árbol de decisión.
2. *Analizar atributos*. Determina el atributo que produce el grupo o partición más uniforme de instancias basado en la etiqueta de clase de las instancias.
3. *Crear el nodo*. Si se alcanza el umbral de uniformidad predefinido, crea un nodo hoja y lo etiqueta con el rótulo de la clase correspondiente. De lo contrario, crea un nodo no- hoja que prueba el atributo, asigna a ese nodo los datos de la partición correspondiente y regresa al paso dos.

El resultado final del procedimiento de aprendizaje del árbol de decisión descrito anteriormente es un modelo de árbol de decisión en el que la totalidad, o la mayoría, de los datos de un nodo hoja muestran la misma etiqueta de clase. Por otro lado, un reto en el aprendizaje de árboles de decisión es maximizar la uniformidad o pureza de los datos asignados a los nodos hoja del modelo, asegurando, a su vez, que el modelo generaliza bien a datos invisibles, es decir, datos que no aparecen en el conjunto de entrenamiento. Lo cual se conoce como habilidad de generalización o sesgo de compensación [90].

En resumen, la asignación del atributo al nodo considerado es la parte crucial del algoritmo de construcción del árbol de decisión. Por lo general, la elección del atributo se basa en medidas de uniformidad, pureza y homogeneidad, de referencia para el proceso descrito previamente, y que se calculan sobre el correspondiente conjunto de datos de entrenamiento. Entre estas medidas se encuentran: ganancia de información (basado en la teoría de la información o entropía de Shannon), impureza de Gini, reducción de la varianza y distancia de Kolmogorov-Smirnov [93]. Existen dos tipos de árboles de decisión: binario y no binario. Para el caso binario, los nodos se dividen en dos nodos secundarios y las ramas son etiquetadas por algunos subconjuntos complementarios del grupo de posibles valores para los atributos. Para el caso no binario, los nodos se dividen en la misma cantidad de nodos como posibles valores para que los atributos existan y cada rama se etiqueta con cada valor posible.

Existen diversos algoritmos de aprendizaje de árboles de decisión, los cuales difieren en dos tópicos: tipo de árbol (binario o no binario) y tipo de medida. Algunos de estos algoritmos son:

1. ID3 (*iterative dichotomiser 3*), propuesto por Quinlan en 1986 [92], produce árboles no binarios. Emplea la teoría de información (o entropía de Shannon) como medida de impureza, por lo que, la función de medición se denomina ganancia de información [93]. A través de este procedimiento se obtiene el algoritmo básico de partición recursiva para un número arbitrario de clases y para atributos con valores discretos
2. C4.5 (sucesor de ID3 y su versión actualizada), propuesto por Quinlan en 1993, [94], también se basa en la entropía de información de Shannon. Mediante este algoritmo es propuesta una función adicional denominada información dividida, que toma valores altos para atributos con grandes dominios [93]. Además, C4.5 incorpora varias mejoras de significativa importancia, las cuales son necesarias para abordar problemas reales, incluyendo el manejo de valores numéricos y atributos faltantes, evitar *overfitting* y mejorar la escalabilidad [95].

Cabe destacar que, ambos algoritmos (ID3 y C4.5), eligen la prueba que extrae la cantidad máxima de información de un conjunto de casos, dada la restricción de que sólo se probará un atributo [96].

3. CART (*classification and regression tree*), propuesto por Breiman *et. al* en 1984 [23]. Mediante este algoritmo se construyen árboles de decisión binarios, para ello se divide un nodo en dos nodos secundarios repetidamente, comenzando con el nodo raíz que contiene toda la muestra de aprendizaje. El algoritmo emplea el índice económico de Gini como criterio para elegir las mejores variables características [97].

Los árboles de decisión también son usados con frecuencia como parte de métodos compuestos, entre ellos *random forests* [98], o *AdaBoost* [83].

Para concluir, los métodos presentados previamente se adaptan adecuadamente a los requerimientos de este trabajo de investigación, específicamente para el tratamiento de las señales de vibración. Lo anterior debido a que, en primer lugar, con CEEMD es posible extraer los modos de oscilación intrínsecos presentes en las señales de vibración. Con la extracción de estos modos se obtienen dos ventajas de singular importancia. Por un lado, se genera una mayor cantidad de muestras de una misma señal para futuras inferencias estadísticas. Por otro lado, cada modo contiene información específica referente al comportamiento de la señal, lo cual, sumado al proceso de extracción de características, permite establecer un patrón de comportamiento de cada condición de

operación del motor estudiada. En segundo lugar, con la extracción de características se obtiene un vector de valores representativos de las señales de vibración del motor, que sirven para la validación a realizar en la siguiente etapa. Finalmente, mediante los algoritmos de clasificación y el vector de características, obtenido en la etapa anterior, se identifican y etiquetan las diferentes condiciones de operación estudiadas, además, se valida la selección de características y se aprueba la pertinencia de la metodología propuesta para un futuro conjunto de datos. El procedimiento de aplicación de estos métodos, a señales de vibración reales, se muestra a continuación en el siguiente capítulo.

Capítulo 4

Descripción del procedimiento propuesto

La detección de fallas en motores de inducción es objeto de estudio continuo para los procedimientos de vigilancia y control de la maquinaria en la industria. Una metodología adecuada para la inspección de los motores, evitaría pérdida de tiempos en la producción y aumento de los costos por reparación de equipos. También, para el mantenimiento preventivo de los motores es una ventaja contar con estos procedimientos puesto que se reducen los tiempos de paradas programadas.

Para el desarrollo de procedimientos de detección de fallas se requiere de un estudio detallado de las características de los motores bajo determinadas condiciones de operación. Las señales de vibración de un motor están compuestas por la suma de las vibraciones de cada una de las partes que lo conforman. Por medio de estas señales, se pueden estudiar las características dinámicas de los motores [19]. Por esa razón, dichas señales se emplean en los procedimientos de vigilancia y control de la condición de trabajo de estas máquinas, y también para la detección de fallas que pudieran presentar.

En lo concerniente a esta investigación, un procedimiento para la detección de fallas en motores de inducción es propuesto, para ello se emplean las señales de vibración de un motor con diferentes condiciones de operación: sano, con barra rota, con desbalance de carga y con defecto en rodamientos, además de la combinación de estas últimas tres. La metodología propuesta consiste en tres etapas, en las cuales se procesan y analizan las señales de vibración para el logro de los objetivos planteados. En primer lugar, se realiza la descomposición de las señales. Seguidamente, se extraen, de cada señal descompuesta, un parámetro de complejidad y varios parámetros estadísticos (de posición, de dispersión y de forma) para construir con ellos un vector de características que resuma la información contenida en las señales. Finalmente, se usa ese vector

de características como entrada a tres clasificadores para identificar y etiquetar las fallas. En el diagrama de bloques de la Figura 4.1, se muestra un esquema general del procedimiento.

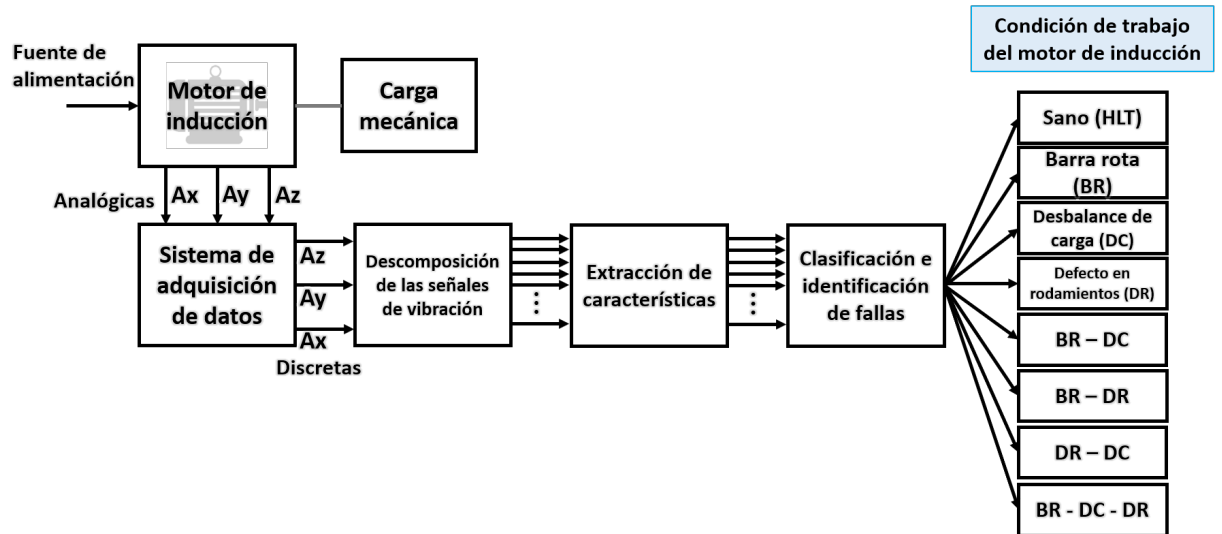


Figura 4.1. Procedimiento propuesto.

Las etapas de la Figura 4.1 se describen a continuación:

4.1. Sistema de adquisición de datos

Las señales de vibración que se usaron son las provenientes de los tres ejes (A_x , A_y y A_z) de un motor de inducción trifásico modelo WEG 00136APE48T, con capacidad de 1 hp, dos polos, 28 barras y alimentado por una fuente de CA de 220 V a 60 Hz. Un alternador ordinario, que representa el 25% de la carga total del motor, se empleó como carga mecánica. Para la adquisición de las señales de vibración fue usado un acelerómetro triaxial modelo LIS3L02AS4, basado en la tecnología MEMS (*Micro-Electro-Mechanical Systems*), fabricado por *STMicroelectronics*.

El sistema de adquisición de datos constaba de un convertidor analógico-digital de 12 bits, cuatro canales, con salida serial, modelo ADS7841. El sistema de instrumentación empleaba una frecuencia de muestreo de 1.5 kHz para obtener 4096 muestras (en cada eje) durante la etapa del transitorio de arranque del motor. Para sincronizar la adquisición de datos con el encendido y apagado del motor, éste fue controlado por un relé. Además, para adquirir las señales se realizaron 20 pruebas por condición de operación y por eje del motor, para ocho condiciones de operación (siete fallas y la

condición sano) y tres ejes (x , y y z). De este modo, se obtuvieron un total de 480 señales, clasificadas por condición, eje y número de prueba como se observa en la Tabla 4.1.

Tabla 4.1. Señales obtenidas en la adquisición.

N°	condición	Ejes		
		x	y	z
1.	BR	$Prueba1, \dots,$ $Prueba20$	$Prueba1, \dots,$ $Prueba20$	$Prueba1, \dots,$ $Prueba20$
2.	BR_DC	$Prueba1, \dots,$ $Prueba20$	$Prueba1, \dots,$ $Prueba20$	$Prueba1, \dots,$ $Prueba20$
3.	BR_DC_DR	$Prueba1, \dots,$ $Prueba20$	$Prueba1, \dots,$ $Prueba20$	$Prueba1, \dots,$ $Prueba20$
4.	BR_DR	$Prueba1, \dots,$ $Prueba20$	$Prueba1, \dots,$ $Prueba20$	$Prueba1, \dots,$ $Prueba20$
5.	DC	$Prueba1, \dots,$ $Prueba20$	$Prueba1, \dots,$ $Prueba20$	$Prueba1, \dots,$ $Prueba20$
6.	DR	$Prueba1, \dots,$ $Prueba20$	$Prueba1, \dots,$ $Prueba20$	$Prueba1, \dots,$ $Prueba20$
7.	DR_DC	$Prueba1, \dots,$ $Prueba20$	$Prueba1, \dots,$ $Prueba20$	$Prueba1, \dots,$ $Prueba20$
8.	HLT	$Prueba1, \dots,$ $Prueba20$	$Prueba1, \dots,$ $Prueba20$	$Prueba1, \dots,$ $Prueba20$

De la Tabla 4.1, HLT hace referencia a la condición *sano*, BR a la condición *barra rota*, DC a la condición *desbalance de carga* y DR a la condición *defecto en rodamientos*. Seguidamente, y con la finalidad de extraer las señales de vibración para cada condición, las fallas fueron artificialmente inducidas en las partes del motor correspondientes (barras del rotor, eje de rotación y rodamientos) de la siguiente forma:

Fallas individuales

La condición de barra rota (BR) se produjo artificialmente mediante la perforación de un hoyo de 2.0 mm de diámetro para causar una rotura parcial de la barra. La Figura 4.2a muestra el rotor con una barra rota usado durante la adquisición de las señales. Para el caso de la condición de falla por defecto en rodamientos (DR), se perforó un hoyo de 1.19 mm de diámetro en el anillo externo del rodamiento, para lo cual se empleó una broca de tungsteno. En la Figura 4.2b se muestra el daño artificial que se le hizo al rodamiento. La condición de desbalance de carga (DC) se produjo añadiendo una masa a una polea del motor. En la Figura 4.2c, se puede observar cómo queda la polea con la masa añadida.

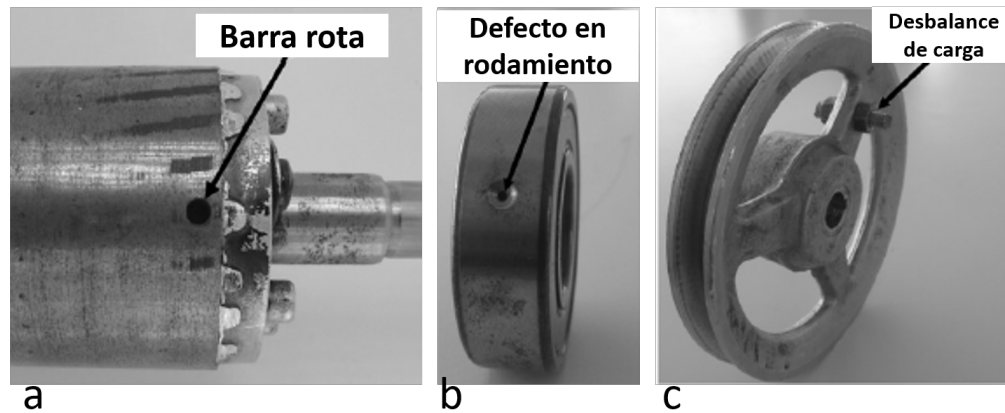


Figura 4.2. Fallas inducidas artificialmente a) Barra rota (BR), b) Defecto en rodamientos (DR), c) Desbalance de carga (DC).

Fallas múltiples y combinadas

Las fallas múltiples y combinadas se obtuvieron mezclando cada falla individual con una o dos de las fallas restantes. Por tanto, se obtuvieron señales para las siguientes combinaciones: barra rota con desbalance de carga (*BR_DC*); barra rota con defecto en rodamientos (*BR_DR*); defecto en rodamientos con desbalance de carga (*DR_DC*); barra rota con desbalance de carga y con defecto en rodamientos (*BR_DC_DR*). En la Figura 4.3 se muestran las diferentes combinaciones para las fallas múltiples.

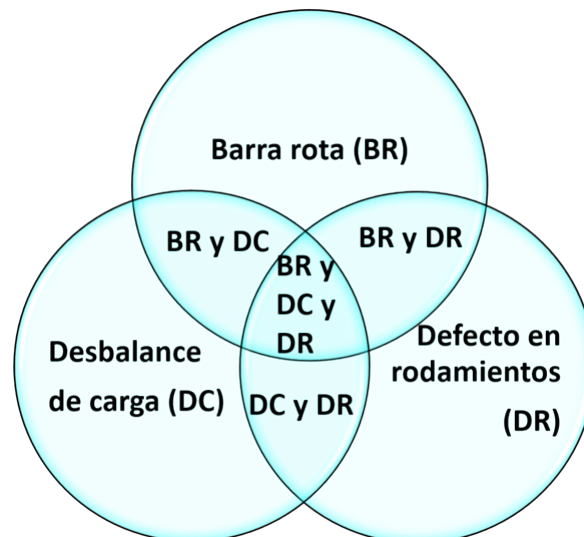


Figura 4.3. Fallas múltiples y combinadas.

4.2. Descomposición de las señales de vibración

La descomposición de las señales se realizó con la finalidad de obtener los modos de oscilación intrínsecos, físicamente significativos, de las señales de vibración del motor. El método de descomposición usado fue el de CEEMD. En este procedimiento fueron usadas las 480 señales de vibración, las cuales, como se había mencionado, están clasificadas por condición, eje y número de prueba.

El algoritmo para la descomposición de las señales mediante CEEMD fue presentado en el ICASSP-11 (*International Conference on Acoustics, Speech and Signal Processing, 2011*) por Torres *et al.* en [14] y se encuentra disponible de forma gratuita para su uso en Matlab® en [99]. El algoritmo es llamado desde su función principal, `ceemd.m`, desde la cual se obtienen los modos de la señal, la forma de usarla es la siguiente:

```
[modes its] = ceemd(x, Nstd, NR, MaxIter);
```

donde, “modes” contiene en forma de matriz, los modos físicamente significativos que se obtuvieron de la descomposición, cada columna de esta matriz es un modo. Por su parte, “its” es el número de iteraciones del proceso *sifting* por cada modo por cada realización (una columna para cada realización). Los parámetros de la función son los siguientes:

- * `x`: señal que se quiere descomponer.
- * `Nstd`: desviación estándar del ruido añadido.
- * `NR`: número de realizaciones o cantidad de veces que se realiza la descomposición para encontrar los modos correctos.
- * `MaxIter`: número máximo de iteraciones permitidas para el proceso *sifting*.

Un esquema general del procedimiento que se realiza mediante CEEMD se muestra a continuación:

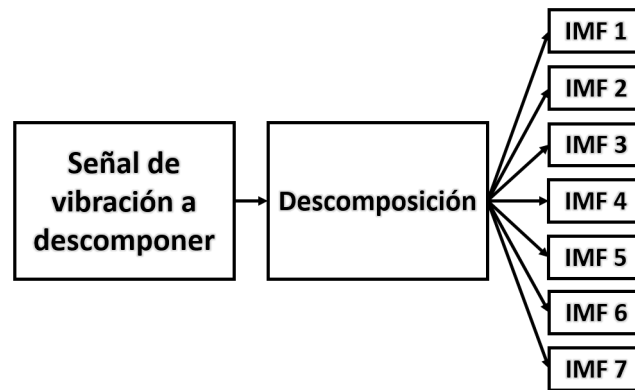


Figura 4.4. Esquema general del proceso de descomposición de las señales de vibración para cada condición.

Como se mencionó previamente, en el procedimiento de captura de las señales, se obtuvieron un total de 480 señales de vibración clasificadas por condición, eje y número de prueba. A cada una de estas 480 señales se les realizó la descomposición, para lo cual el algoritmo determina, de acuerdo a las condiciones que debe cumplir una IMF, cuántos modos obtener. Tal como se observa en el diagrama de la Figura 4.4, se obtienen siete modos para cada señal.

4.3. Extracción de características

El proceso de extracción de características consiste en representar las medidas originales de una señal en atributos más eficaces, los cuales resumen la cantidad de recursos que se requieren para describir el comportamiento de las señales de vibración del motor. Por lo tanto, y considerando que los modos de oscilación obtenidos mediante el proceso de descomposición no dejan ver, de forma directa, las características relevantes e identificativas de las señales; sino que cada uno de ellos es una parte intrínseca de la señal de vibración original; varios parámetros estadísticos (de posición, de dispersión y de forma) y un parámetro de complejidad (la entropía) son calculados a partir de cada uno de estos modos de oscilación, para construir un vector de características que resuma, de forma eficaz, los atributos que presentan las señales de vibración obtenidas con cada condición de operación del motor.

Entre el conjunto de parámetros estadísticos considerados, se encuentran:

- * De posición: la media.
- * De dispersión: la desviación estándar.
- * De forma: el coeficiente de asimetría y el coeficiente de curtosis o apuntamiento.

Un esquema de este proceso se muestra a continuación en la Figura 4.5.

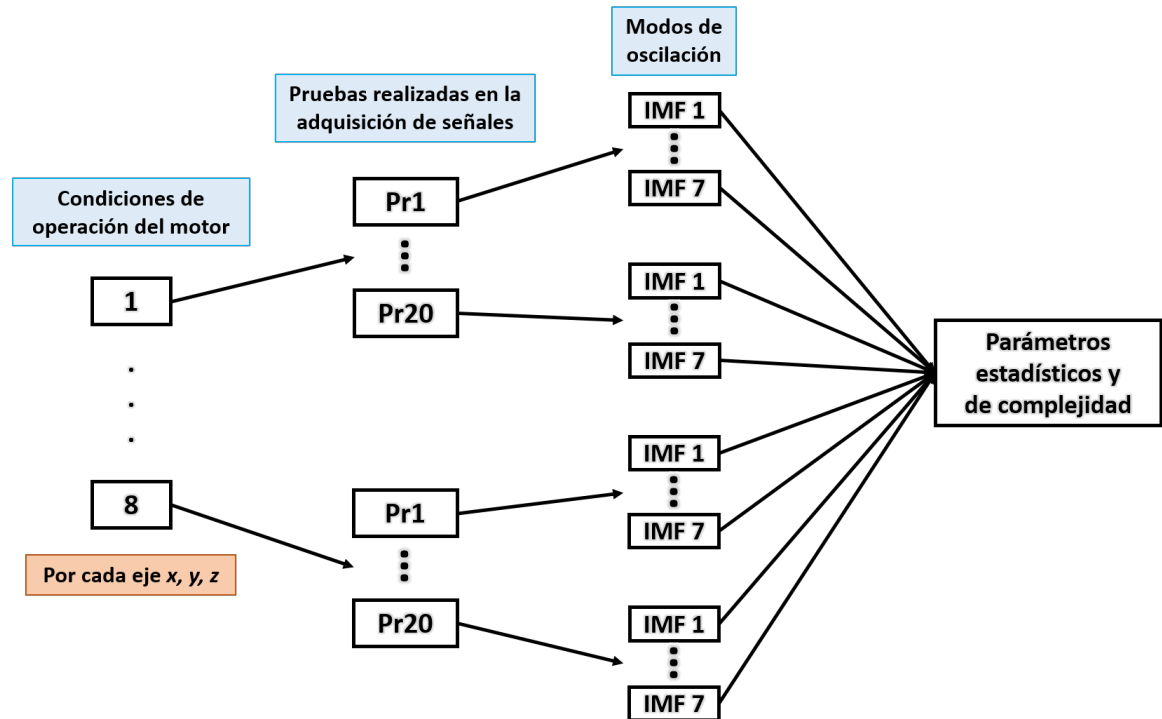


Figura 4.5. Esquema general del proceso de extracción de características.

En el esquema de la Figura 4.5 se observa que cada parámetro (estadístico y de complejidad) fue calculado a partir de cada uno de los siete modos correspondientes a cada señal de vibración obtenida en cada una de las 20 pruebas realizadas, por eje, a un motor con ocho condiciones de operación distintas.

Además de resumir los recursos que se requieren para describir el comportamiento de las señales de vibración, por medio de estos parámetros se construye un patrón que identifica si el motor tiene un comportamiento normal o si presenta alguna falla. Por lo tanto, con este patrón será posible determinar si un futuro conjunto de señales, diferentes a las obtenidas, presenta una condición de operación u otra.

También, este procesamiento estadístico simple está basado en un análisis discriminante que se emplea para identificar el conjunto de características robustas que proporcionan la mayor discriminación entre las categorías de fallas, representadas mediante las señales de vibración del motor. Por consiguiente, haciendo uso de ciertos criterios de selección de características, algunos parámetros de los calculados, que contengan poca información discriminante, serán descartados, resultando en un subconjunto de características con un número reducido de parámetros.

4.4. Clasificación e identificación de fallas

El vector de características, obtenido a partir de los parámetros estadísticos y de complejidad mencionados previamente, representa el punto de partida para el proceso de clasificación e identificación de la condición de operación del motor. Para este proceso, con la finalidad de obtener suficientes resultados que demuestren la pertinencia del método propuesto, fueron empleados los siguientes tres algoritmos de clasificación:

1. Máquinas de soporte vectorial (*support vector machines*, SVMs).
2. Árboles de decisión.
3. *AdaBoost* (*adaptive boosting*).

Cada uno de estos algoritmos es entrenado para obtener clasificadores con la capacidad de identificar la condición de operación sano, la presencia de alguna falla individual o sus posibles combinaciones. El conjunto de datos de entrenamiento para cada algoritmo contiene lo siguiente:

- * Como variables de entrada: los modos de oscilación (o IMFs) correspondientes a los parámetros estadísticos y de complejidad escogidos en la etapa anterior.
- * Como salida conocida (u objetivo): las condiciones de operación del motor, ocho en total. Cada una de las cuales contiene 20 valores diferentes por variable de entrada, debido a que durante el proceso de adquisición de señales fueron realizadas 20 pruebas.

Siendo ocho salidas conocidas (u objetivos) se requieren clasificadores multiclase, que soporten conjuntos de datos que no estén linealmente separados. Los tres algoritmos de clasificación mencionados previamente se ajustan de forma correcta con este requerimiento. La configuración de cada algoritmo se realizó empleando las funciones correspondientes del *toolbox* de estadística y aprendizaje de máquina de Matlab® , de la siguiente forma:

4.4.1. Máquinas de soporte vectorial (*SVMs*)

Para obtener un clasificador multiclase basado en SVMs, se llevaron a cabo los siguientes pasos:

1. Definir la plantilla de aprendizaje que requiere el algoritmo SVM. En este caso se consideran conjuntos de datos no linealmente separables, para lo cual se usa el truco del kernel. Se especifica mediante el comando de Matlab “`templateSVM`”, de la siguiente forma:

```
svm_temp = templateSVM('KernelFunction','polynomial',...
                      'PolynomialOrder',3);
```

donde, “`svm_temp`” es la plantilla de aprendizaje y los parámetros de la función son los siguientes:

- * ‘`KernelFunction`’: indica que se va a definir una función de transformación (función kernel) para tratar con conjuntos de datos no linealmente separables.
 - * ‘`polynomial`’: hace referencia al tipo de función kernel a utilizar, la cual requiere de un grado (tal y como se mostró previamente en la Tabla 3.1).
 - * ‘`PolynomialOrder`’, 3: se usa para definir el grado de la función kernel.
2. Definir los parámetros para entrenar el algoritmo. En este caso se emplea la función “`fitcecoc`” que retorna un clasificador basado en el modelo de códigos de salida de corrección de errores (*error-correcting output codes*, ECOC) para el aprendizaje multiclase. Este modelo reduce el problema de clasificación de tres o más clases a un conjunto de clasificadores binarios [100]. La función se define de la siguiente forma:

```
Model = fitcecoc(mt_valsel,v,'Learners',svm_temp);
```

donde, “`Model`” corresponde al modelo de clasificación basado en ECOC y los parámetros definidos para esta función son los siguientes:

- * `mt_valsel`: corresponde al conjunto de datos de entrenamiento.
- * `v`: es un vector columna que contiene las etiquetas de clases, es decir, de las ocho condiciones de operación para las cuales se entrena el clasificador.
- * ‘`Learners`’: indica que los clasificadores binarios, del modelo ECOC, deben usar la plantilla de aprendizaje `svm_temp`.

4.4.2. *AdaBoost*

Para obtener un clasificador multiclase, basado en *AdaBoost*, el único paso consistió en lo siguiente:

Definir el modelo de clasificación. Se realiza mediante la función “fitensemble”, la cual retorna un modelo entrenado con múltiples algoritmos de aprendizaje para clasificación o regresión. Los requerimientos fueron especificados con los siguientes parámetros:

```
model_ada = fitensemble(mt_valsel,v,'Bag',T,'Tree',...  
                        'Type','Classification');
```

donde, “model_ada” es el modelo de clasificación y los parámetros de la función son:

- * mt_valsel y v: son los datos de entrenamiento y las etiquetas de clases, respectivamente.
- * 'Bag': define el método para unir los clasificadores débiles, los cuales tendrán T ciclos de aprendizaje.
- * 'Tree': para definir el tipo de clasificadores débiles a utilizar.
- * 'Type','Classification': indica que la tarea a realizar es la clasificación.

4.4.3. Árboles de decisión

Para obtener un clasificador basado en árboles de decisión se realizó lo siguiente:

Definir el clasificador. Se realiza empleando la función “fitctree”, la cual provee un clasificador basado en árboles de decisión para clasificación multiclase. Los parámetros especificados fueron los siguientes:

```
Modeladec = fitctree(mt_valsel,v);
```

donde, “mt_valsel” y “v” son los datos de entrenamiento y las etiquetas de clases, respectivamente.

Los resultados obtenidos luego de realizar los procedimientos descritos previamente, se muestran a continuación en el siguiente capítulo.

Capítulo 5

Resultados

La detección temprana de fallas en motores de inducción actualmente representa una tarea de alta demanda, debido a la importancia de estas máquinas en la industria. Muchos son los eventos indeseados que se evitarían, gracias a un correcto procedimiento de detección y prevención de fallas no solo de los motores, sino de toda la maquinaria de la industria en general. En particular, la metodología propuesta en este trabajo permite la correcta detección de ocho condiciones de operación de los motores de inducción.

El desarrollo de este capítulo está enfocado en la presentación de los resultados obtenidos luego de aplicar el procedimiento propuesto en esta investigación. Para llevar a cabo cada paso de la metodología, fueron empleadas las señales de vibración correspondientes a las ocho condiciones de operación de un motor de inducción trifásico, las cuales son, siete condiciones de falla y una condición de motor sano. Las señales se obtuvieron de un banco de prueba en el cual fueron realizadas 20 pruebas con la finalidad de tener una mayor cantidad de muestras ya que, siguiendo la estadística, mientras mayor es el número de muestras en una población, mejores resultan las inferencias hechas a partir de los cálculos estadísticos.

La estructura del capítulo se divide en tres etapas de acuerdo a la metodología propuesta. Se presentan, en primer lugar, los resultados concernientes a la descomposición de las señales de vibración mediante CEEMD. Seguidamente, los datos de la extracción de características de los modos obtenidos en la descomposición. Y por último, las matrices de confusión referentes al proceso de clasificación e identificación de las fallas. A continuación, se desarrollan las etapas previamente mencionadas.

5.1. Descomposición de las señales de vibración

A partir del método de descomposición, CEEMD, aplicado a las señales de vibración, de cada condición del motor, se obtuvieron los diferentes modos de oscilación intrínsecos (IMFs). Se obtuvieron siete modos en total, el primero de ellos contiene el modo de oscilación más rápido de la señal original y el último contiene el más lento. A su vez, éstos contienen información de gran utilidad con la cual es posible establecer un patrón para la clasificación e identificación de fallas en motores. Los valores de cada descomposición se guardan en una matriz de 4096×7 , una columna por modo, cada uno de 4096 muestras de longitud.

En la Figura 5.1 se muestra la señal de vibración correspondiente a la condición de falla múltiple, barra rota con desbalance de carga y defecto en rodamientos. Seguidamente, en la Figura 5.2, se muestran los modos obtenidos luego de aplicar la descomposición a la señal de vibración de la Figura 5.1.

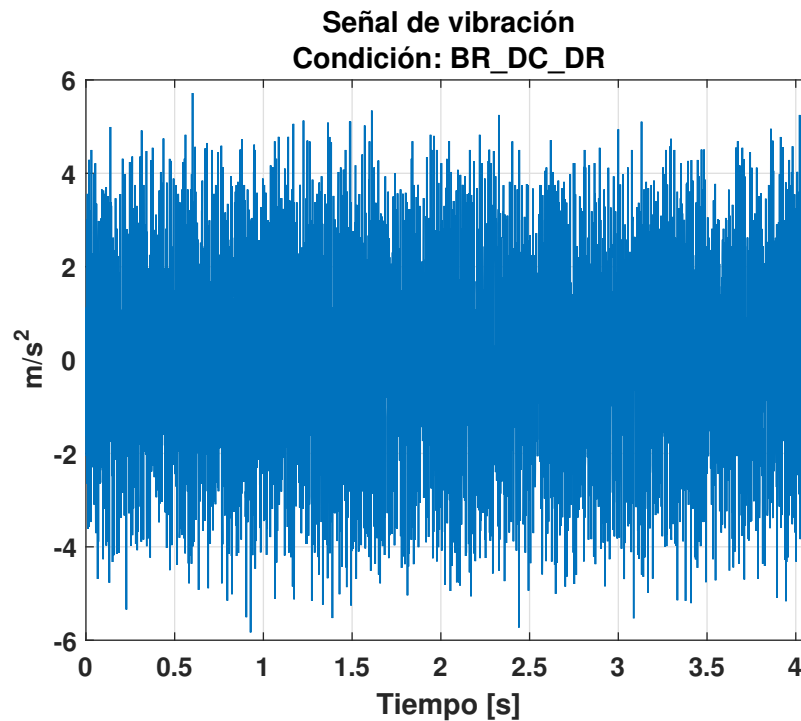


Figura 5.1. Señal de vibración del motor con condición BR_DC_DR.

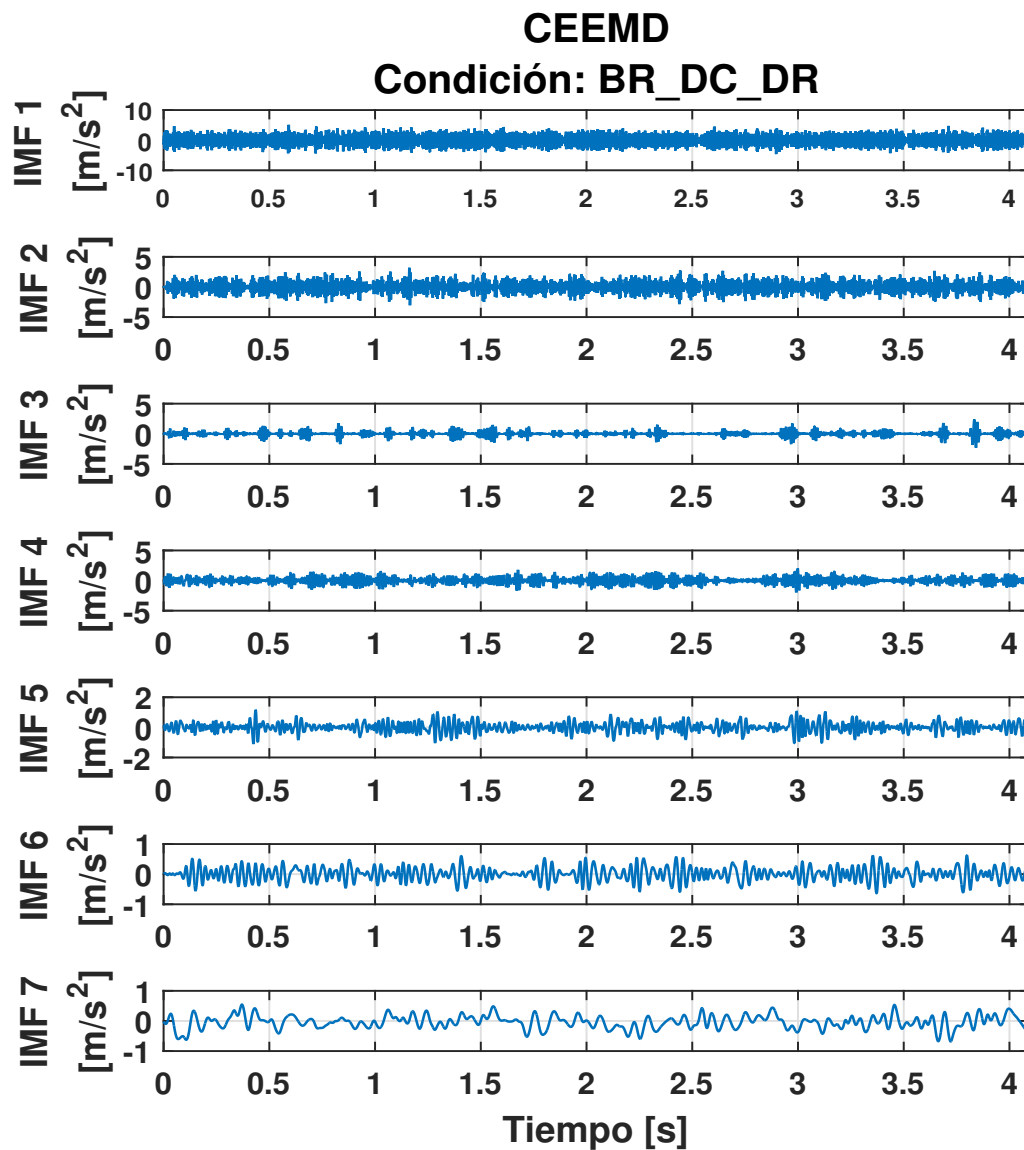


Figura 5.2. Descomposición de la señal de vibración del motor con condición BR_DC_DR.

Todas las señales de vibración, de cada condición, fueron descompuestas a la forma que se observa en la Figura 5.2, de las cuales se obtuvieron 3300 modos, aproximadamente. En resumen, son siete modos por condición, eje y número de prueba.

5.2. Extracción de características - Resultados

A través de diversas investigaciones se ha podido constatar que, por medio de un conjunto de características, es posible representar completamente la información contenida en las señales de vibración referentes a fallas en motores de inducción. Y con esta información, proveer un diagnóstico completo de la condición de operación de este tipo de motores [101].

En esta investigación, con la finalidad de crear un vector de características representativo de la información contenida en los modos de oscilación obtenidos durante la descomposición, fue extraído un conjunto de parámetros el cual estuvo compuesto de parámetros estadísticos como la media, la desviación estándar, el coeficiente de asimetría y el coeficiente de curtosis o apuntamiento, así como también de la entropía como parámetro de complejidad.

Todos estos parámetros fueron evaluados considerando cuan separados entre sí se encontraban los valores correspondientes a las condiciones de operación del motor que se están tratando. De este modo, por medio de este grado de separación es posible identificar si la característica (o parámetro) sirve como un buen factor discriminante o no en el proceso de clasificación e identificación de fallas con otras señales diferentes. En base a esto, unos parámetros fueron seleccionados para formar parte del vector de características mientras que otros se descartaron.

Para representar estos valores fueron usadas campanas gaussianas debido a que, con estas gráficas, además de observar la distribución de los valores alrededor de su valor medio, se puede apreciar cuáles tienen solapamiento unos con otros y cuales se encuentran más separados entre sí. Por un lado, para los parámetros estadísticos de posición y de forma se obtuvo lo siguiente:

- * Media. Presenta valores muy unidos entre sí. A través de la Figura 5.3 se puede observar el comportamiento para cada condición referente a todos los ejes de un modo en específico. Los demás modos presentan un comportamiento similar. Por lo tanto, este parámetro se descarta para el vector de características.

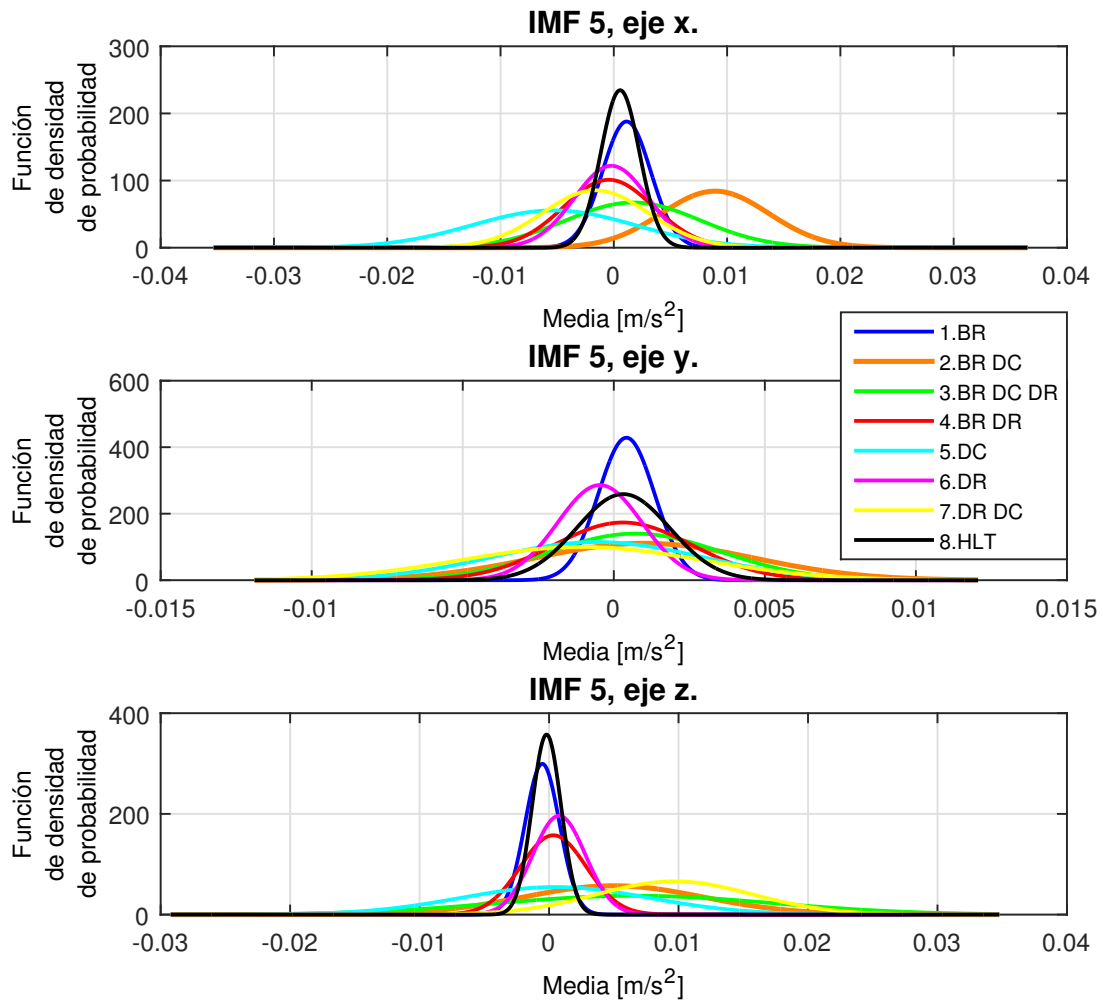


Figura 5.3. Representación de los valores del parámetro media correspondientes a todos los ejes en la IMF 5 de las pruebas realizadas.

- * Coeficiente de asimetría (*skewness*). Las gráficas expuestas en la Figura 5.4 dejan ver cuán unidos se encuentran los valores entre cada condición de operación. A diferencia del anterior, y sólo en algunos modos, existe un poco más de separación. Sin embargo, ésta no era suficiente para considerar este parámetro como una característica con buen factor discriminante.

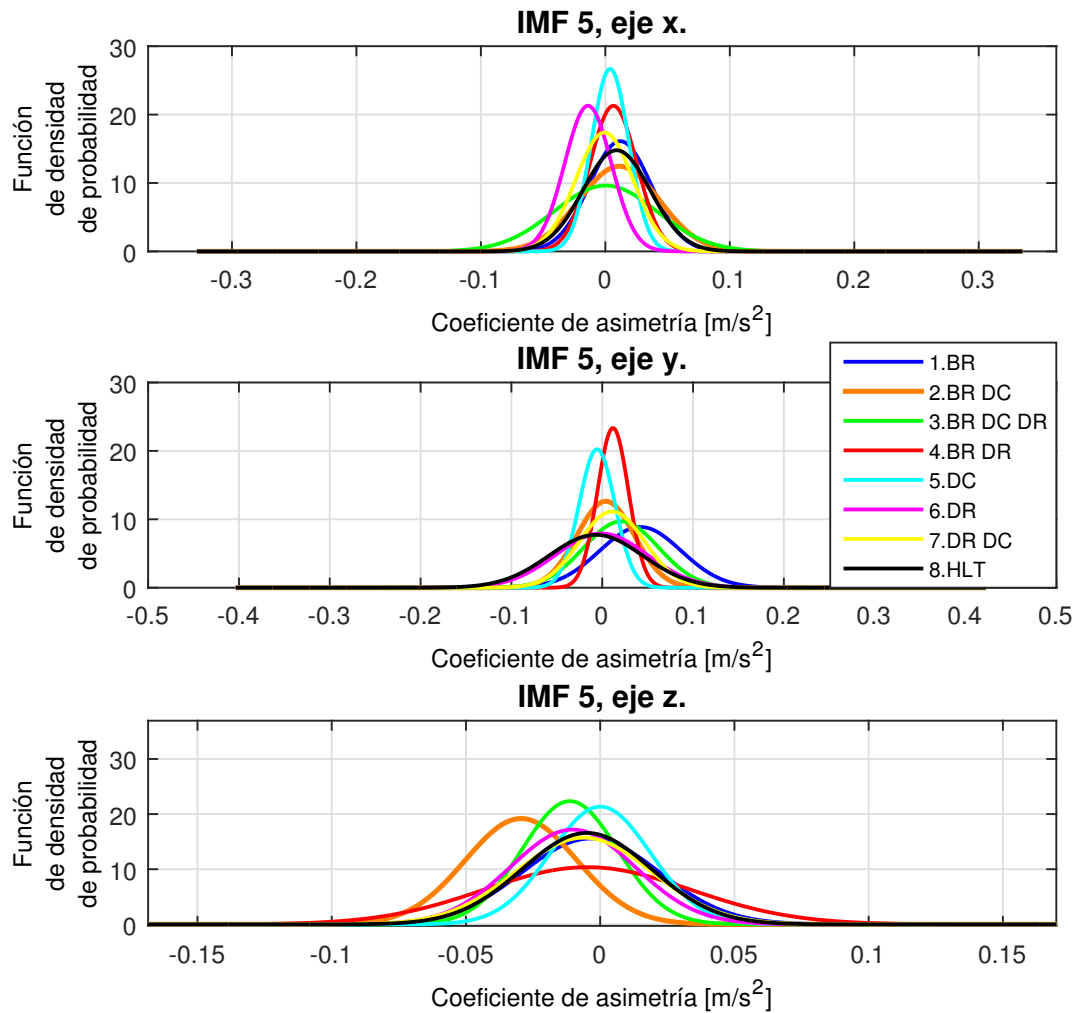


Figura 5.4. Representación de los valores del parámetro coeficiente de asimetría correspondientes a todos los ejes en la IMF 5 de las pruebas realizadas.

* Coeficiente de curtosis o apuntamiento. En este caso, todos los modos presentaron valores muy unidos entre sí. Una muestra de ello es presentada en la Figura 5.5, en las cuales se observa el comportamiento de los valores de cada condición para los tres ejes de un modo en específico. En estas gráficas, a pesar de que el eje z muestra valores un poco separados, los demás modos presentan un comportamiento más parecido al del eje y . Considerando ésto, la curtosis también se descarta para el vector de características.

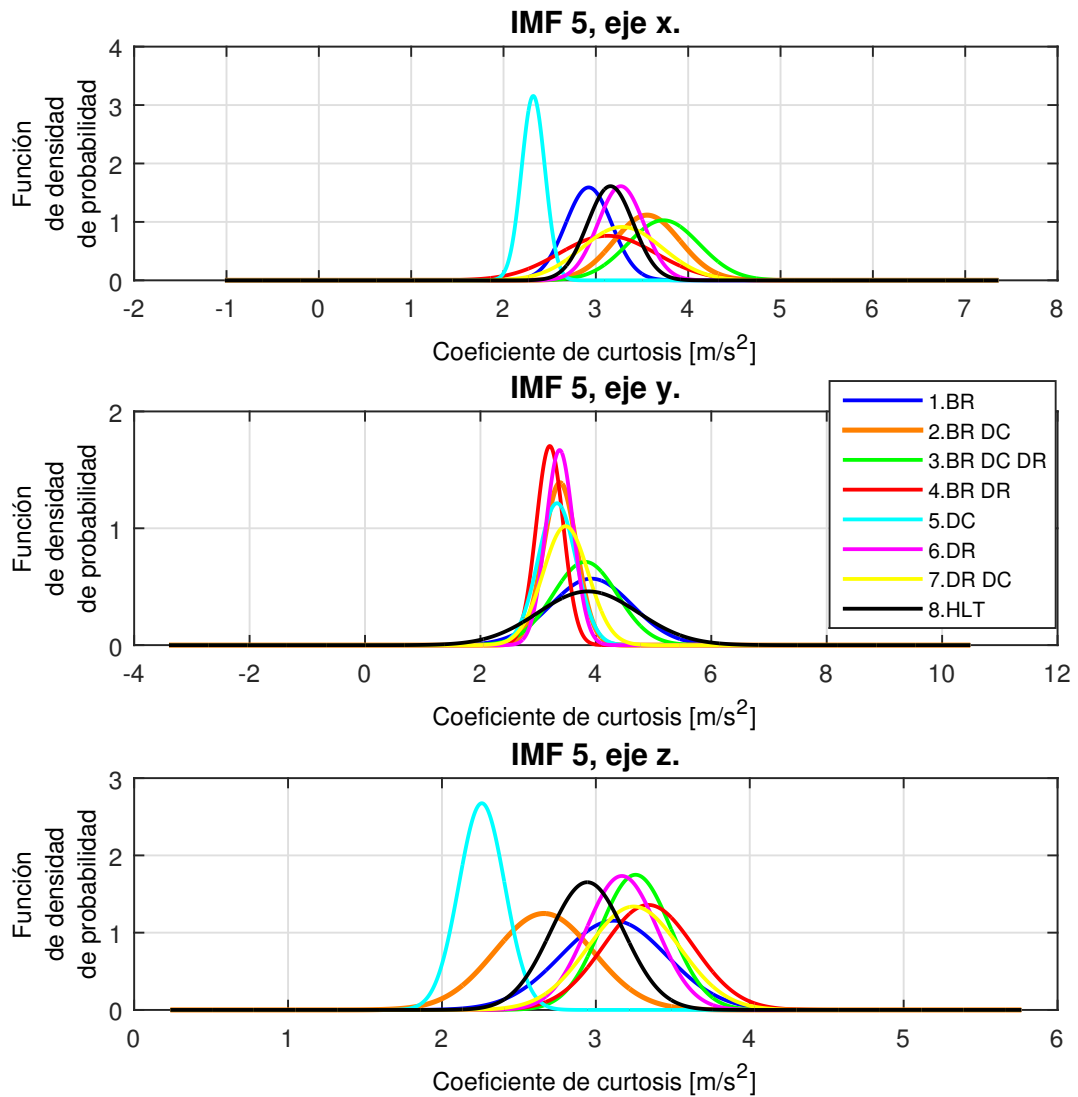


Figura 5.5. Representación de los valores del parámetro coeficiente de curtosis correspondientes a todos los ejes en la IMF 5 de las pruebas realizadas.

Tomando en cuenta las afirmaciones anteriores, estos tres parámetros son descartados debido a que puede ser muy forzado para un clasificador identificar a qué clase (o condición) pertenece una señal diferente si se emplean estos valores como patrones de decisión.

Por otro lado, el parámetro estadístico de dispersión, desviación estándar, y el parámetro de complejidad, entropía, cuyos valores sí presentaron mayor separación entre las condiciones de operación, sí fueron considerados para formar parte del vector

de características. Es importante destacar que, aún cuando estos valores presentan un buen grado de discriminación, fue aplicado un segundo proceso de selección, también considerando separación entre condiciones, pero, en este caso, fueron descartados algunos modos (o IMFs) y algunos ejes también. De forma general, el cálculo de estos valores, las gráficas y los resultados obtenidos se muestran a continuación.

5.2.1. Extracción de parámetro de complejidad. Entropía.

Para la extracción de entropía, se realizó en Matlab® un algoritmo basado en la Ecuación (3.12), que define la entropía o teoría de la información de Shannon [65]. El algoritmo tiene el siguiente procedimiento:

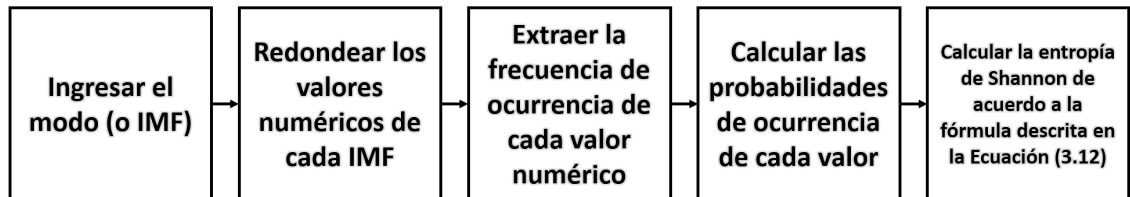


Figura 5.6. Procedimiento para la extracción del valor de entropía de Shannon a cada modo o IMF obtenido en la descomposición.

Del esquema de la Figura 5.6, en la primera etapa se leen los datos contenidos en el vector columna que contiene los valores numéricos de cada modo o IMF. En la segunda etapa, se realiza un redondeo de esos valores con la finalidad de tener valores enteros para el cálculo de las probabilidades siguientes. En la tercera etapa, se calculan las frecuencias de ocurrencia de cada valor, a través de un histograma, con la finalidad de obtener un panorama de la distribución de cada valor con respecto a las muestras obtenidas. En la cuarta etapa, se calculan las probabilidades de ocurrencia de cada valor de la señal, empleando la definición básica de probabilidad, número de casos favorables entre el número de casos posibles. Finalmente, se aplica la sumatoria definida en la Ecuación (3.12) para calcular la entropía de Shannon. Este procedimiento se realiza de forma iterativa a cada modo de cada condición. Los resultados obtenidos se presentan a continuación.

5.2.1.1. Resultados obtenidos

La entropía de información de Shannon, o simplemente entropía de información, mide el nivel de peculiaridad de las señales o combinaciones. La representación de este parámetro también se realizó mediante campanas gaussianas, pero en este caso

para seleccionar cuáles IMFs eran las más apropiados para formar parte del vector de características. Para comenzar, en la Tabla 5.1, se presentan los valores de media y desviación estándar, requeridos para construir las campanas gaussianas, referentes a cada condición de operación del motor, cada eje, y a la IMF 1.

Tabla 5.1. Media y desviación estándar de la característica de entropía en la IMF 1 de las pruebas realizadas.

N°	condición	Ejes		
		x	y	z
1.	<i>BR</i>	$\mu = 10,4772$ $\sigma = 0,0925$	$\mu = 10,5902$ $\sigma = 0,1356$	$\mu = 11,0059$ $\sigma = 0,0242$
2.	<i>BR_DC</i>	$\mu = 10,9482$ $\sigma = 0,0355$	$\mu = 10,9736$ $\sigma = 0,0470$	$\mu = 11,0573$ $\sigma = 0,0539$
3.	<i>BR_DC_DR</i>	$\mu = 10,9293$ $\sigma = 0,0510$	$\mu = 10,9069$ $\sigma = 0,0318$	$\mu = 10,9299$ $\sigma = 0,0588$
4.	<i>BR_DR</i>	$\mu = 10,9174$ $\sigma = 0,0366$	$\mu = 10,9096$ $\sigma = 0,1080$	$\mu = 10,9307$ $\sigma = 0,0344$
5.	<i>DC</i>	$\mu = 10,9525$ $\sigma = 0,0343$	$\mu = 10,7084$ $\sigma = 0,0465$	$\mu = 10,8841$ $\sigma = 0,0424$
6.	<i>DR</i>	$\mu = 10,8328$ $\sigma = 0,0304$	$\mu = 10,9123$ $\sigma = 0,0229$	$\mu = 11,0355$ $\sigma = 0,0176$
7.	<i>DR_DC</i>	$\mu = 10,9751$ $\sigma = 0,0224$	$\mu = 10,9811$ $\sigma = 0,0262$	$\mu = 10,9786$ $\sigma = 0,0408$
8.	<i>HLLT</i>	$\mu = 10,6752$ $\sigma = 0,1161$	$\mu = 10,5028$ $\sigma = 0,0642$	$\mu = 10,8$ $\sigma = 0,0305$

La representación correspondiente a los datos de la Tabla 5.1, se muestra en la Figura 5.7. En este caso, los valores de entropía se encuentran muy cerca unos con otros, por tanto, a efectos de un proceso de clasificación se requieren más valores para formar un vector de características bien definido, para ello se toman en cuenta los valores de entropía referentes a las IMFs restantes.

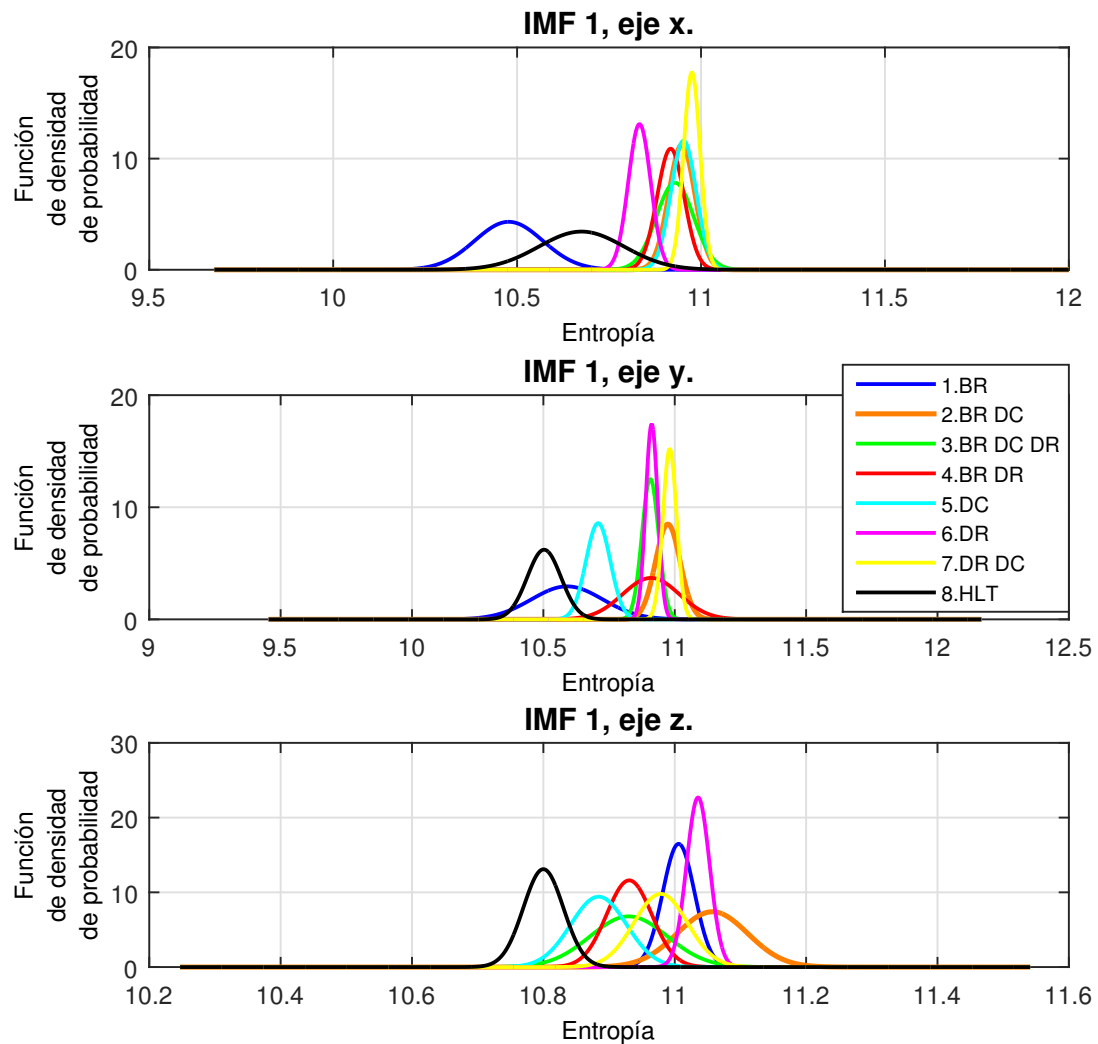


Figura 5.7. Representación de los valores de la característica de entropía correspondientes a todos los ejes en la IMF 1 de las pruebas realizadas.

Seguidamente, en las Tablas 5.2 y 5.3, se muestran los resultados de la IMF 2.

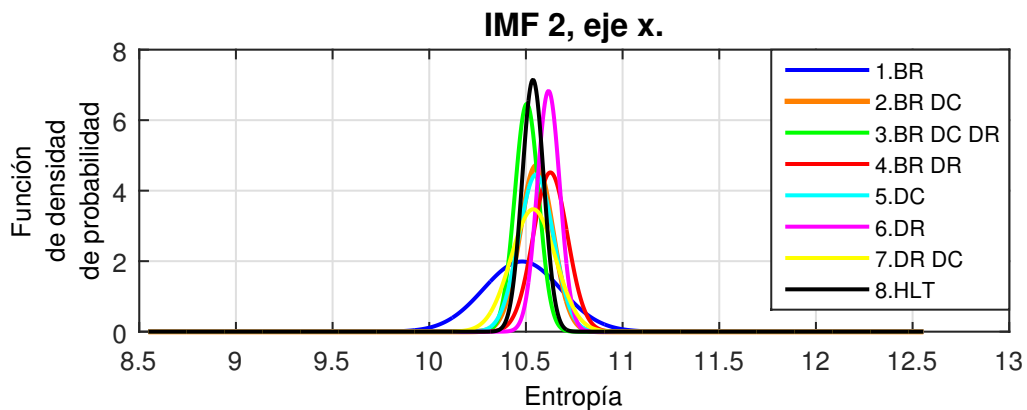
Tabla 5.2. Media y desviación estándar de la característica de entropía en la IMF 2 de las pruebas realizadas.

Nº	condición	Ejes		
		x	y	z
1.	BR	$\mu = 10,48$ $\sigma = 0,2004$	$\mu = 10,5902$ $\sigma = 0,1356$	$\mu = 11,0059$ $\sigma = 0,0242$

Tabla 5.3. Media y desviación estándar de la característica de entropía en la IMF 2 de las pruebas realizadas.

N°	condición	Ejes		
		x	y	z
2.	<i>BR_DC</i>	$\mu = 10,5547$ $\sigma = 0,0846$	$\mu = 10,9736$ $\sigma = 0,0470$	$\mu = 11,0573$ $\sigma = 0,0539$
3.	<i>BR_DC_DR</i>	$\mu = 10,5044$ $\sigma = 0,0617$	$\mu = 10,9069$ $\sigma = 0,0318$	$\mu = 10,9299$ $\sigma = 0,0588$
4.	<i>BR_DR</i>	$\mu = 10,6265$ $\sigma = 0,0883$	$\mu = 10,9096$ $\sigma = 0,1080$	$\mu = 10,9307$ $\sigma = 0,0344$
5.	<i>DC</i>	$\mu = 10,5531$ $\sigma = 0,0895$	$\mu = 10,7084$ $\sigma = 0,0465$	$\mu = 10,8841$ $\sigma = 0,0424$
6.	<i>DR</i>	$\mu = 10,6169$ $\sigma = 0,0584$	$\mu = 10,9123$ $\sigma = 0,0229$	$\mu = 11,0355$ $\sigma = 0,0176$
7.	<i>DR_DC</i>	$\mu = 10,5378$ $\sigma = 0,1149$	$\mu = 10,9811$ $\sigma = 0,0262$	$\mu = 10,9786$ $\sigma = 0,0408$
8.	<i>HLT</i>	$\mu = 10,5365$ $\sigma = 0,0558$	$\mu = 10,5028$ $\sigma = 0,0642$	$\mu = 10,8$ $\sigma = 0,0305$

Las Figuras 5.8 y 5.9 muestran la representación gráfica de los valores de las Tablas 5.2 y 5.3. En estos casos, como en el anterior, los valores están muy cercanos entre sí. La misma situación ocurre con los valores de todos los ejes de la IMF 3, los ejes x y y de la IMF 4, el eje x de la IMF 5, por lo tanto, éstos serán descartados para el proceso de clasificación.

**Figura 5.8.** Representación de los valores de la característica de entropía correspondientes al eje x en la IMF 2 de las pruebas realizadas.

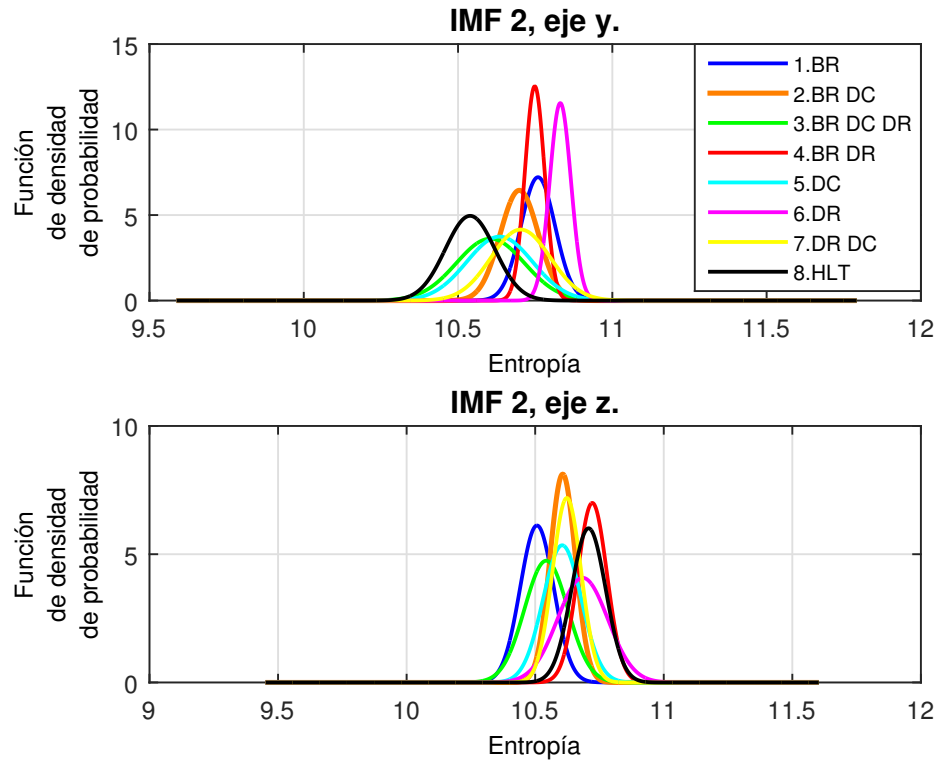


Figura 5.9. Representación de los valores de la característica de entropía correspondientes a los ejes y y z en la IMF 2 de las pruebas realizadas.

Por otro lado, los valores del eje z de la IMF 4, los ejes y y z de la IMF 5, los de todos los ejes de las IMFs 6 y 7, sí fueron considerados para el vector de características del proceso de clasificación. Los valores de entropía correspondientes al eje z de la IMF 4, se muestran a continuación en la Tabla 5.4.

Tabla 5.4. Media y desviación estándar de la característica de entropía del eje z en la IMF 4 de las pruebas realizadas.

Nº	condición	Eje z
1.	<i>BR</i>	$\mu = 8,9241$, $\sigma = 0,1407$
2.	<i>BR_DC</i>	$\mu = 10,0136$, $\sigma = 0,0685$
3.	<i>BR_DC_DR</i>	$\mu = 10,3607$, $\sigma = 0,1138$
4.	<i>BR_DR</i>	$\mu = 10,5284$, $\sigma = 0,1147$
5.	<i>DC</i>	$\mu = 10,1324$, $\sigma = 0,1006$
6.	<i>DR</i>	$\mu = 9,9290$, $\sigma = 0,0945$
7.	<i>DR_DC</i>	$\mu = 10,4240$, $\sigma = 0,1763$
8.	<i>HLT</i>	$\mu = 9,4750$, $\sigma = 0,0896$

Seguidamente, en la Figura 5.10 se muestran las campanas de Gauss correspondientes a los valores de entropía presentados en la Tabla 5.4.

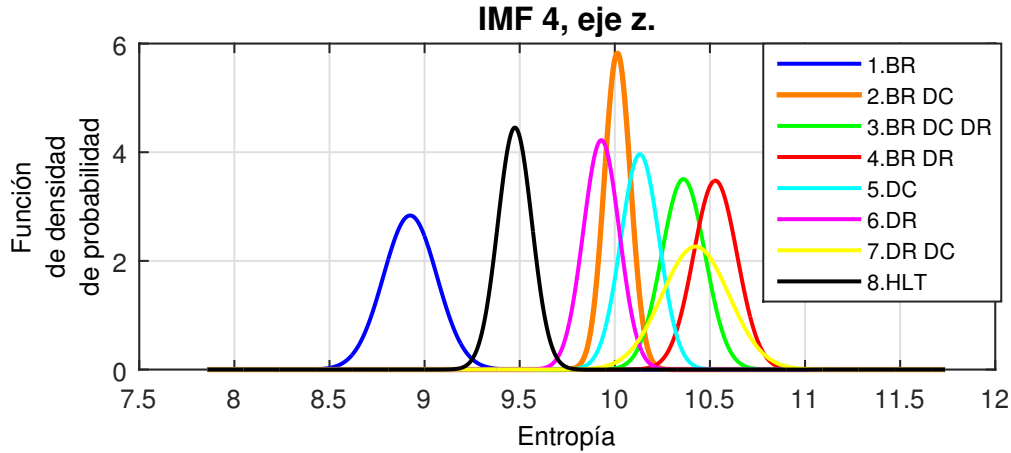


Figura 5.10. Representación de los valores de la característica de entropía correspondientes al eje z en la IMF 4 de las pruebas realizadas.

En la Figura 5.10, es perceptible que los valores de entropía se encuentran más separados entre ellos, por tanto este conjunto de datos sí es más apropiado para el vector de características. A continuación se muestran los valores de los ejes y y z de la IMF 5.

Tabla 5.5. Media y desviación estándar de la característica de entropía de los ejes y y z en la IMF 5 de las pruebas realizadas.

Nº	condición	Ejes	
		y	z
1.	BR	$\mu = 8,5587, \sigma = 0,1189$	$\mu = 9,0796, \sigma = 0,1064$
2.	BR_DC	$\mu = 9,9229, \sigma = 0,1141$	$\mu = 10,5020, \sigma = 0,0892$
3.	BR_DC_DR	$\mu = 9,8946, \sigma = 0,1162$	$\mu = 10,6283, \sigma = 0,0865$
4.	BR_DR	$\mu = 9,7082, \sigma = 0,0710$	$\mu = 10,0709, \sigma = 0,1240$
5.	DC	$\mu = 9,7305, \sigma = 0,2047$	$\mu = 10,7627, \sigma = 0,0441$
6.	DR	$\mu = 9,6807, \sigma = 0,0984$	$\mu = 9,6973, \sigma = 0,0682$
7.	DR_DC	$\mu = 10,2488, \sigma = 0,1327$	$\mu = 10,5620, \sigma = 0,1078$
8.	HLT	$\mu = 9,1737, \sigma = 0,1119$	$\mu = 9,1208, \sigma = 0,1034$

Para los datos presentados en la Tabla 5.5, la representación es la que se muestra a continuación.

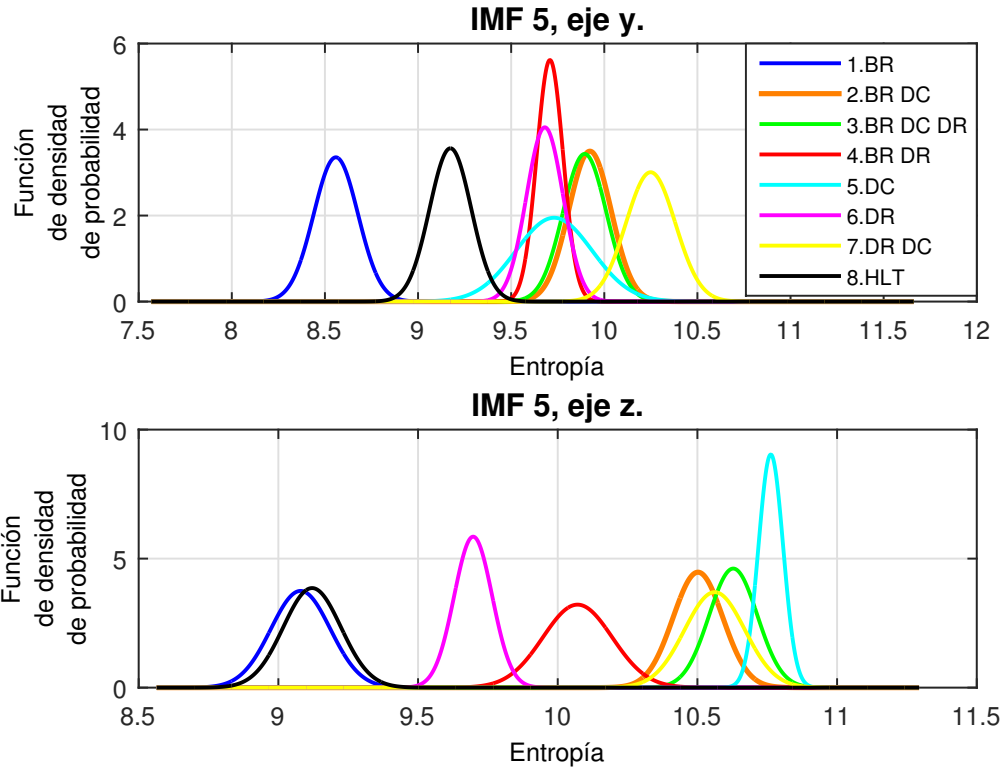


Figura 5.11. Representación de los valores de la característica de entropía correspondientes a los ejes y y z en la IMF 5 de las pruebas realizadas.

En la Figura 5.11, se observa una mayor separación de los valores. Sin embargo, hay situaciones como la que se presenta entre las condiciones BR y HLT (curvas azul y negro, respectivamente) del eje z , en las cuales las gráficas están casi solapadas, en este caso, se puede compensar con la inclusión de los valores del eje y u otros valores característicos. Los resultados correspondientes a todos los ejes de la IMF 6 se presentan a continuación en las Tablas 5.6 y 5.7.

Tabla 5.6. Media y desviación estándar de la característica de entropía en la IMF 6 de las pruebas realizadas.

N°	condición	Ejes		
		x	y	z
1.	BR	$\mu = 8,6787$	$\mu = 8,0079$	$\mu = 8,4905$
		$\sigma = 0,1632$	$\sigma = 0,1438$	$\sigma = 0,1564$
2.	BR_DC	$\mu = 9,9548$	$\mu = 9,6210$	$\mu = 9,6621$
		$\sigma = 0,1663$	$\sigma = 0,1158$	$\sigma = 0,2616$
3.	BR_DC_DR	$\mu = 10,2884$	$\mu = 9,6853$	$\mu = 10,1571$
		$\sigma = 0,1182$	$\sigma = 0,0819$	$\sigma = 0,0907$

Tabla 5.7. Media y desviación estándar de la característica de entropía en la IMF 6 de las pruebas realizadas.

Nº	condición	Ejes		
		x	y	z
4.	BR_DR	$\mu = 9,6504$	$\mu = 9,1259$	$\mu = 9,6036$
		$\sigma = 0,0961$	$\sigma = 0,1462$	$\sigma = 0,1548$
5.	DC	$\mu = 9,5673$	$\mu = 9,1290$	$\mu = 9,5991$
		$\sigma = 0,2441$	$\sigma = 0,1680$	$\sigma = 0,2106$
6.	DR	$\mu = 9,3006$	$\mu = 9,2195$	$\mu = 9,3292$
		$\sigma = 0,1054$	$\sigma = 0,1265$	$\sigma = 0,0877$
7.	DR_DC	$\mu = 9,7955$	$\mu = 9,8038$	$\mu = 10,0792$
		$\sigma = 0,0998$	$\sigma = 0,0981$	$\sigma = 0,1861$
8.	HLT	$\mu = 8,7151$	$\mu = 8,7406$	$\mu = 8,3792$
		$\sigma = 0,1770$	$\sigma = 0,1094$	$\sigma = 0,1174$

Seguidamente, las Figuras 5.12 y 5.13 muestran la representación gráfica de los valores referentes a los ejes x y y y al eje z de la IMF 6, respectivamente.

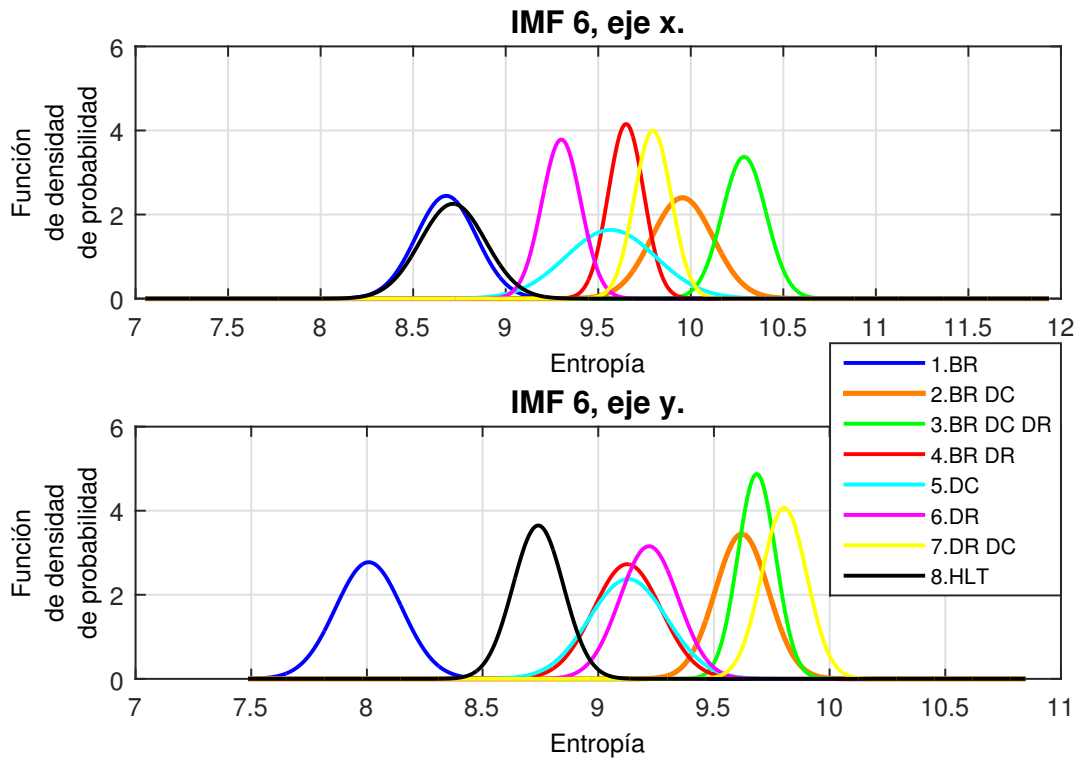


Figura 5.12. Representación de los valores de la característica de entropía correspondientes a los ejes x y y en la IMF 6 de las pruebas realizadas.

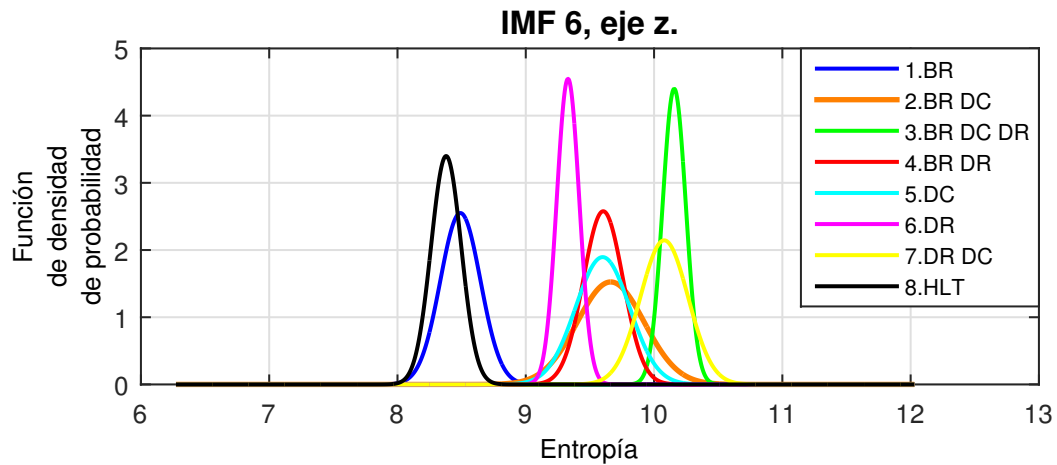


Figura 5.13. Representación de los valores de la característica de entropía correspondientes al eje z en la IMF 6 de las pruebas realizadas.

Finalmente, para complementar los valores representados en las Figuras 5.12 y 5.13, se muestran en la Tabla 5.8, los valores de entropía referentes a todos los ejes de la IMF 7.

Tabla 5.8. Media y desviación estándar de la característica de entropía en la IMF 7 de las pruebas realizadas.

N°	condición	Ejes		
		x	y	z
1.	BR	$\mu = 8,6290$	$\mu = 8,1089$	$\mu = 8,4870$
		$\sigma = 0,1374$	$\sigma = 0,1613$	$\sigma = 0,1734$
2.	BR_DC	$\mu = 9,8843$	$\mu = 9,6974$	$\mu = 9,6398$
		$\sigma = 0,0965$	$\sigma = 0,1052$	$\sigma = 0,1349$
3.	BR_DC_DR	$\mu = 9,8829$	$\mu = 9,5024$	$\mu = 9,9538$
		$\sigma = 0,0783$	$\sigma = 0,0662$	$\sigma = 0,1077$
4.	BR_DR	$\mu = 9,7825$	$\mu = 9,1694$	$\mu = 9,6382$
		$\sigma = 0,0574$	$\sigma = 0,0663$	$\sigma = 0,1782$
5.	DC	$\mu = 9,7145$	$\mu = 9,1936$	$\mu = 9,6248$
		$\sigma = 0,0602$	$\sigma = 0,0871$	$\sigma = 0,1135$
6.	DR	$\mu = 9,3236$	$\mu = 9,4533$	$\mu = 9,4238$
		$\sigma = 0,0798$	$\sigma = 0,0505$	$\sigma = 0,0696$
7.	DR_DC	$\mu = 9,9504$	$\mu = 10,0368$	$\mu = 10,0985$
		$\sigma = 0,0712$	$\sigma = 0,1634$	$\sigma = 0,1517$
8.	HLT	$\mu = 8,5411$	$\mu = 8,7260$	$\mu = 8,2543$
		$\sigma = 0,0970$	$\sigma = 0,1086$	$\sigma = 0,1014$

Seguidamente, la representación de estos valores se muestran en la Figura 5.14.

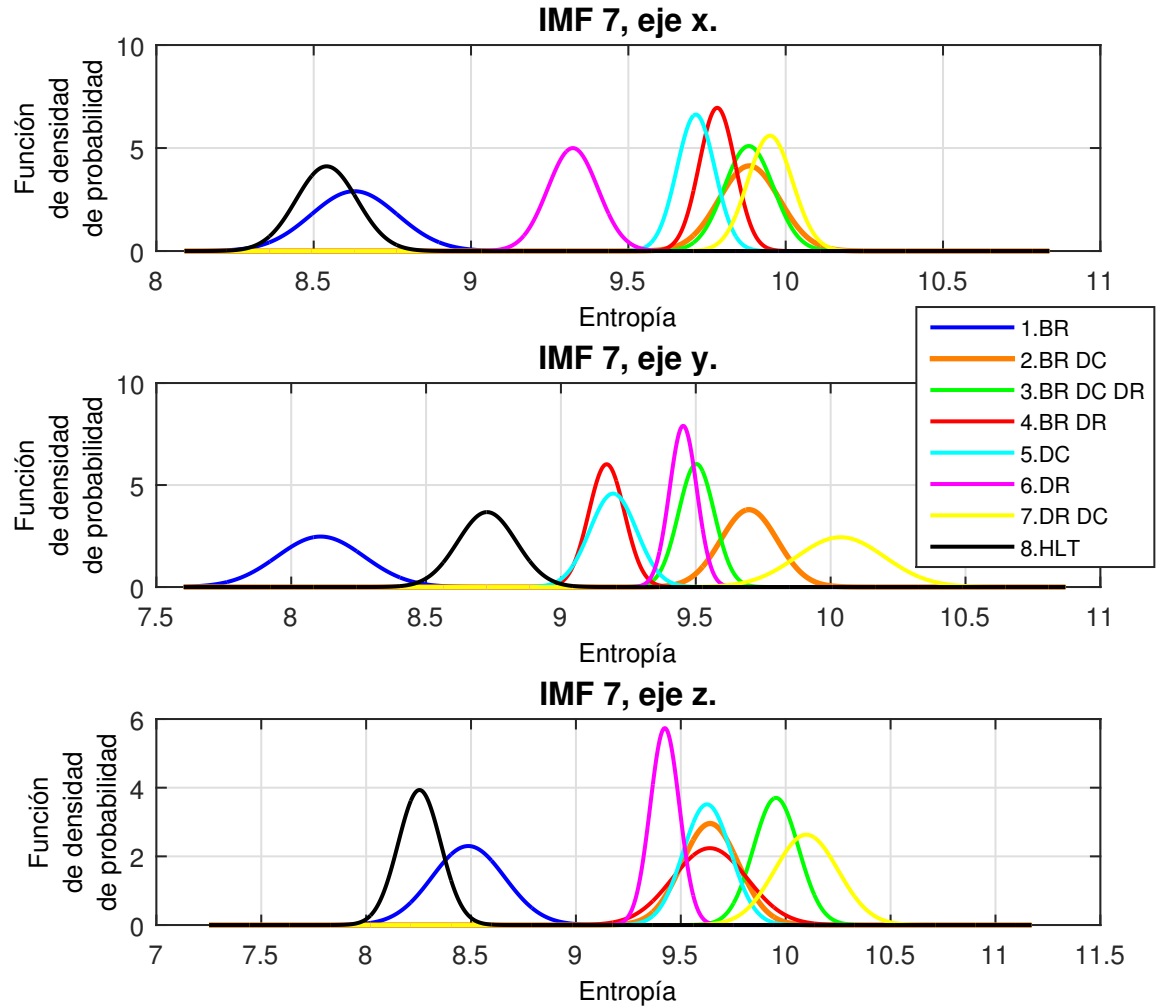


Figura 5.14. Representación de los valores de la característica de entropía correspondientes a todos los ejes en la IMF 7 de las pruebas realizadas.

Como se observa en la Figura 5.14, estos valores de entropía presentan solapamiento de unos con otros. A pesar de esto, el parámetro estadístico de dispersión, la desviación estándar, presenta un comportamiento favorable que ayuda a complementar los valores obtenidos con la entropía. La información referente al parámetro de desviación estándar como parte del vector de características se muestra a continuación.

5.2.2. Extracción de parámetro estadístico de dispersión. Desviación estándar

La medida que se obtiene con el cálculo de la desviación estándar representa el grado de dispersión de los datos con respecto al valor promedio de éstos. En este caso, se emplea la desviación estándar como parámetro característico para complementar, junto con la entropía, el vector de características representativas que se usa en el proceso de clasificación de las fallas. Para el cálculo de la desviación estándar se empleó la función básica de Matlab®, $\text{std}(x)$, siendo x el vector que contiene los valores de la señal correspondiente a cada modo obtenido mediante CEEMD. Para el cálculo de este valor, Matlab® se basa en la fórmula general definida para la desviación estándar, que es la siguiente:

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N |A_i - \mu|^2} \quad (5.1)$$

donde, A es un vector de variables aleatorias, N es la dimensión del vector y μ es el valor medio de A , el cual se define se la siguiente forma:

$$\mu = \frac{1}{N} \sum_{i=1}^N A_i \quad (5.2)$$

5.2.2.1. Resultados obtenidos

La desviación estándar, o desviación típica, fue extraída a partir de los modos o IMFs obtenidos de la descomposición con la finalidad de definir otro parámetro descriptivo de la información contenida en las señales de vibración. Los resultados de este procedimiento se presentan a continuación. Inicialmente, en las Tablas 5.9 y 5.10, se muestran los valores para la IMF 1.

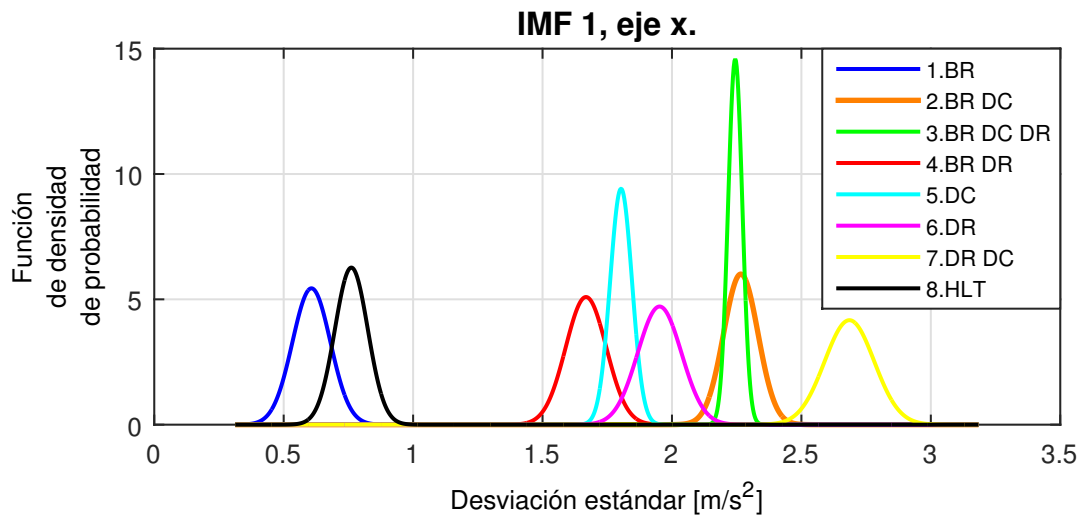
Tabla 5.9. Media y desviación estándar de la característica de desviación estándar en la IMF 1 de las pruebas realizadas.

N°	condición	Ejes		
		x	y	z
1.	BR	$\mu = 0,6072$	$\mu = 0,6275$	$\mu = 2,6935$
		$\sigma = 0,0733$	$\sigma = 0,0660$	$\sigma = 0,2863$
2.	BR_DC	$\mu = 2,2660$	$\mu = 3,1195$	$\mu = 3,6869$
		$\sigma = 0,0664$	$\sigma = 0,1831$	$\sigma = 0,2306$
3.	BR_DC_DR	$\mu = 2,2443$	$\mu = 1,9240$	$\mu = 2,2384$
		$\sigma = 0,0273$	$\sigma = 0,0619$	$\sigma = 0,1159$

Tabla 5.10. Media y desviación estándar de la característica de desviación estándar en la IMF 1 de las pruebas realizadas.

Nº	condición	Ejes		
		x	y	z
4.	BR_DR	$\mu = 1,6679$	$\mu = 1,7431$	$\mu = 2,0088$
		$\sigma = 0,0783$	$\sigma = 0,1517$	$\sigma = 0,1434$
5.	DC	$\mu = 1,8034$	$\mu = 1,1659$	$\mu = 1,9514$
		$\sigma = 0,0424$	$\sigma = 0,0412$	$\sigma = 0,0661$
6.	DR	$\mu = 1,9521$	$\mu = 2,0374$	$\mu = 3,7111$
		$\sigma = 0,0846$	$\sigma = 0,0237$	$\sigma = 0,0177$
7.	DR_DC	$\mu = 2,6856$	$\mu = 3,6335$	$\mu = 3,9603$
		$\sigma = 0,0957$	$\sigma = 0,2473$	$\sigma = 0,1309$
8.	HLT	$\mu = 0,7614$	$\mu = 0,8854$	$\mu = 1,2472$
		$\sigma = 0,0637$	$\sigma = 0,0561$	$\sigma = 0,0869$

La representación de estos valores se muestra en las Figuras 5.15 y 5.16 para el eje x y para los ejes y y z , respectivamente. En este caso, al igual que con la entropía, hay valores que se superponen unos con otros, dificultando, de este modo, escoger cuáles son correctos para el proceso de clasificación y cuáles no. En este sentido, se han descartado todos los valores referentes a los tres ejes de la IMF 2 y de la IMF 3, el eje x de la IMF 4 y de la IMF 5, el eje z de la IMF 6 y los ejes x y z de la IMF 7.

**Figura 5.15.** Representación de los valores de la característica de desviación estándar correspondientes al eje x en la IMF 1 de las pruebas realizadas.

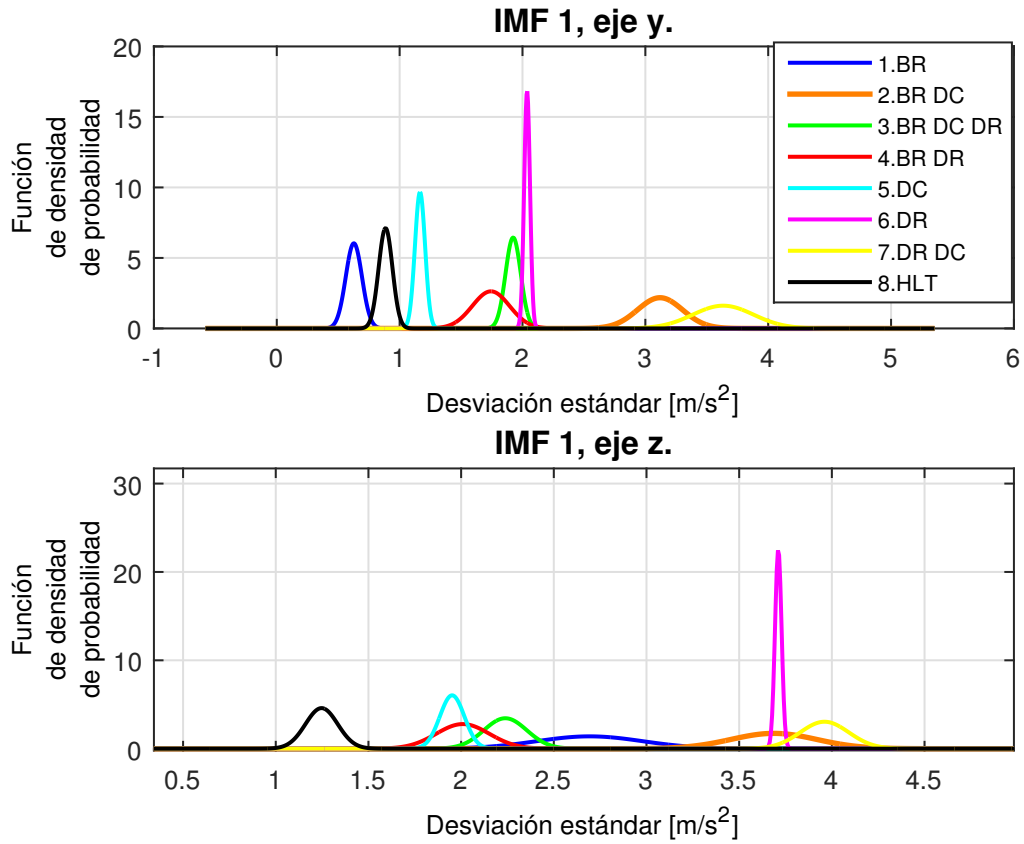


Figura 5.16. Representación de los valores de la característica de desviación estándar correspondientes a los ejes y y z en la IMF 1 de las pruebas realizadas.

En las Figuras 5.15 y 5.16, se observa que los valores se presentan con mayor separación entre sí. Para compensar los que se solapan, fueron utilizados los referentes a los ejes y y z de la IMF 4 y de la IMF 5, los ejes x y y de la IMF 6 y el eje y de la IMF 7. Los resultados correspondientes a la IMF 4 se presentan a continuación en las Tablas 5.11 y 5.12.

Tabla 5.11. Media y desviación estándar de la característica de desviación estándar de los ejes y y z en la IMF 4 de las pruebas realizadas.

Nº	condición	Ejes	
		y	z
1.	<i>BR</i>	$\mu = 0,1906$, $\sigma = 0,0185$	$\mu = 0,1350$, $\sigma = 0,0141$
2.	<i>BR_DC</i>	$\mu = 0,4244$, $\sigma = 0,0297$	$\mu = 0,3209$, $\sigma = 0,0188$
3.	<i>BR_DC_DR</i>	$\mu = 0,4915$, $\sigma = 0,0377$	$\mu = 0,4547$, $\sigma = 0,0287$
4.	<i>BR_DR</i>	$\mu = 0,3597$, $\sigma = 0,0184$	$\mu = 0,5767$, $\sigma = 0,0521$

Tabla 5.12. Media y desviación estándar de la característica de desviación estándar de los ejes y y z en la IMF 4 de las pruebas realizadas.

Nº	condición	Ejes	
		y	z
5.	DC	$\mu = 0,4029$, $\sigma = 0,0225$	$\mu = 0,3641$, $\sigma = 0,0311$
6.	DR	$\mu = 0,4573$, $\sigma = 0,0224$	$\mu = 0,2952$, $\sigma = 0,0236$
7.	DR_DC	$\mu = 0,5755$, $\sigma = 0,0478$	$\mu = 0,5958$, $\sigma = 0,1268$
8.	HLT	$\mu = 0,3777$, $\sigma = 0,0387$	$\mu = 0,1993$, $\sigma = 0,0135$

Y la representación de estos valores se muestra en la Figura 5.17.

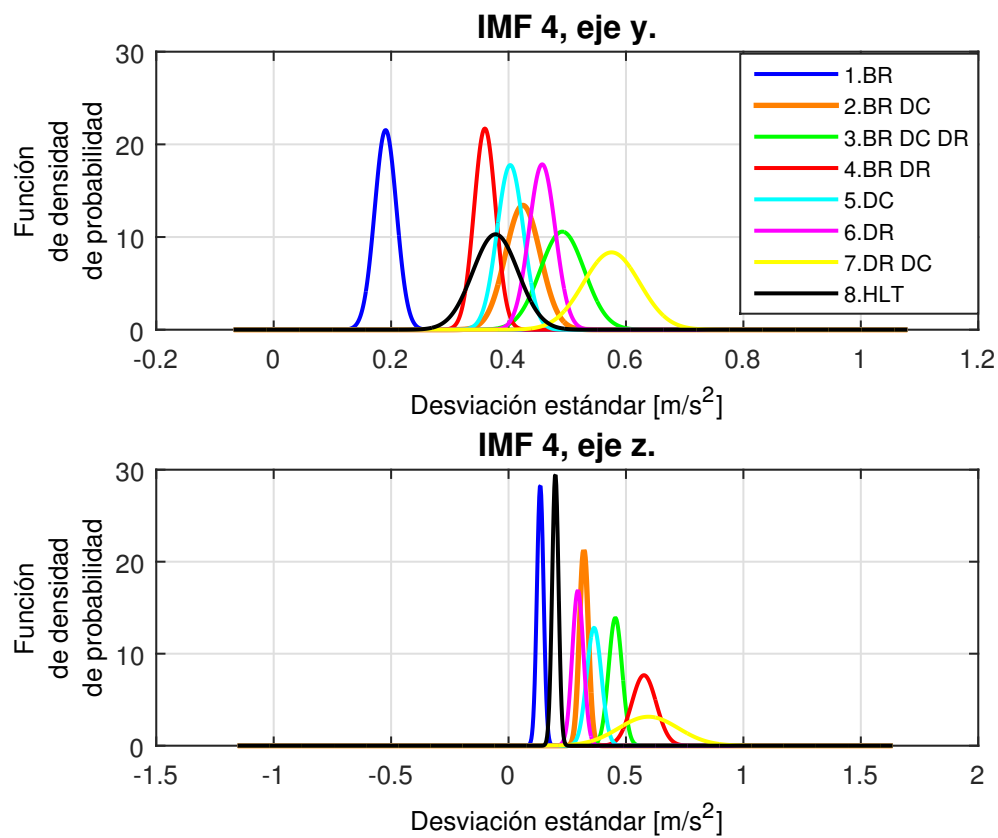


Figura 5.17. Representación de los valores de la característica de desviación estándar correspondientes a los ejes y y z en la IMF 4 de las pruebas realizadas.

Para la IMF 5 fueron seleccionados los valores de desviación estándar referentes a los ejes y y z , estos resultados se muestran a continuación en la Tabla 5.13.

Tabla 5.13. Media y desviación estándar de la característica de desviación estándar de los ejes y y z en la IMF 5 de las pruebas realizadas.

N°	condición	Ejes	
		y	z
1.	BR	$\mu = 0,0996$, $\sigma = 0,0090$	$\mu = 0,1457$, $\sigma = 0,0113$
2.	BR_DC	$\mu = 0,2957$, $\sigma = 0,0284$	$\mu = 0,4878$, $\sigma = 0,0385$
3.	BR_DC_DR	$\mu = 0,2943$, $\sigma = 0,0221$	$\mu = 0,5905$, $\sigma = 0,0440$
4.	BR_DR	$\mu = 0,2416$, $\sigma = 0,0153$	$\mu = 0,3382$, $\sigma = 0,0343$
5.	DC	$\mu = 0,2529$, $\sigma = 0,0411$	$\mu = 0,6331$, $\sigma = 0,0305$
6.	DR	$\mu = 0,2386$, $\sigma = 0,0181$	$\mu = 0,2392$, $\sigma = 0,0126$
7.	DR_DC	$\mu = 0,3997$, $\sigma = 0,0496$	$\mu = 0,5690$, $\sigma = 0,0738$
8.	HLT	$\mu = 0,1624$, $\sigma = 0,0140$	$\mu = 0,1490$, $\sigma = 0,0121$

La representación de los valores de la Tabla 5.13, se muestran en la Figura 5.18.

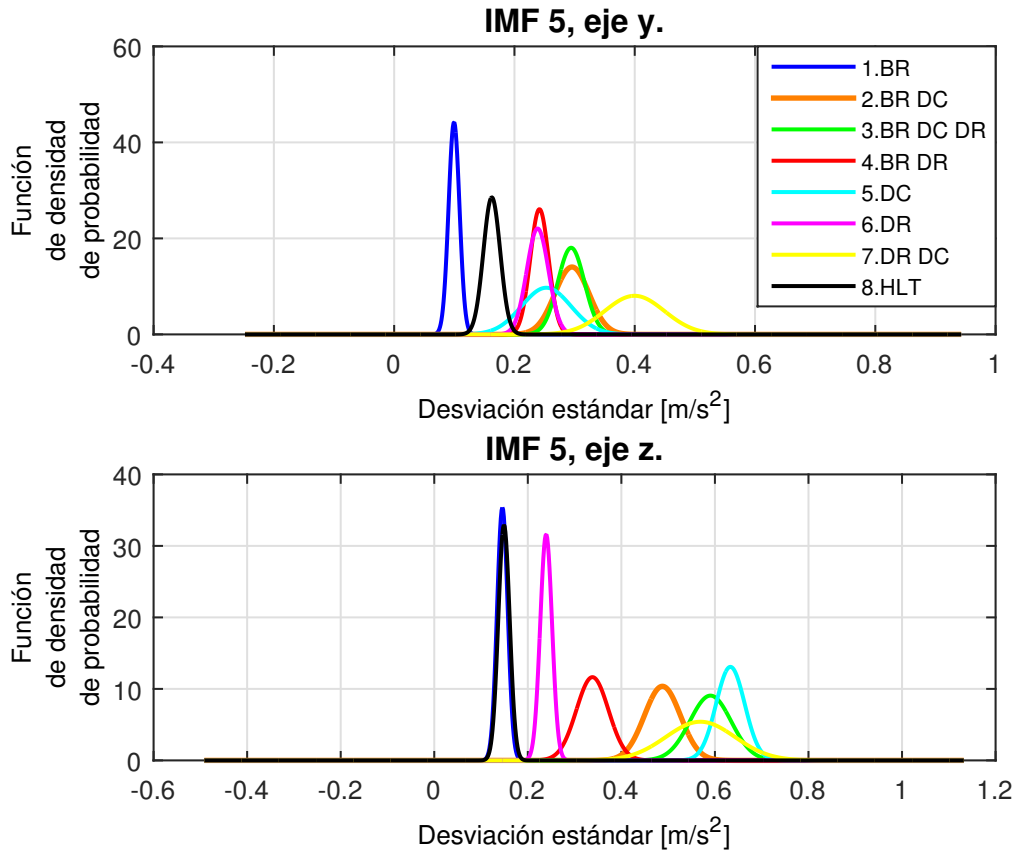


Figura 5.18. Representación de los valores de la característica de desviación estándar correspondientes a los ejes y y z en la IMF 5 de las pruebas realizadas.

Los resultados presentados en la Figura 5.18 también presentan solapamiento unos con otros. Por tanto, fueron considerados más valores, como los de los ejes x y y del IMF 6 que se presentan en la Tabla 5.14.

Tabla 5.14. Media y desviación estándar de la característica de desviación estándar de los ejes x y y en la IMF 6 de las pruebas realizadas.

Nº	condición	Ejes	
		x	y
1.	BR	$\mu = 0,1104$, $\sigma = 0,0128$	$\mu = 0,0673$, $\sigma = 0,0065$
2.	BR_DC	$\mu = 0,3045$, $\sigma = 0,0444$	$\mu = 0,2256$, $\sigma = 0,0220$
3.	BR_DC_DR	$\mu = 0,4006$, $\sigma = 0,0406$	$\mu = 0,2348$, $\sigma = 0,0156$
4.	BR_DR	$\mu = 0,2387$, $\sigma = 0,0210$	$\mu = 0,1518$, $\sigma = 0,0172$
5.	DC	$\mu = 0,2396$, $\sigma = 0,0369$	$\mu = 0,1535$, $\sigma = 0,0200$
6.	DR	$\mu = 0,1745$, $\sigma = 0,0137$	$\mu = 0,1623$, $\sigma = 0,0154$
7.	DR_DC	$\mu = 0,2639$, $\sigma = 0,0207$	$\mu = 0,2636$, $\sigma = 0,0216$
8.	HLT	$\mu = 0,1112$, $\sigma = 0,0143$	$\mu = 0,1139$, $\sigma = 0,0092$

La representación de los valores de la Tabla 5.14 se muestra en las Figuras 5.19 y 5.20.

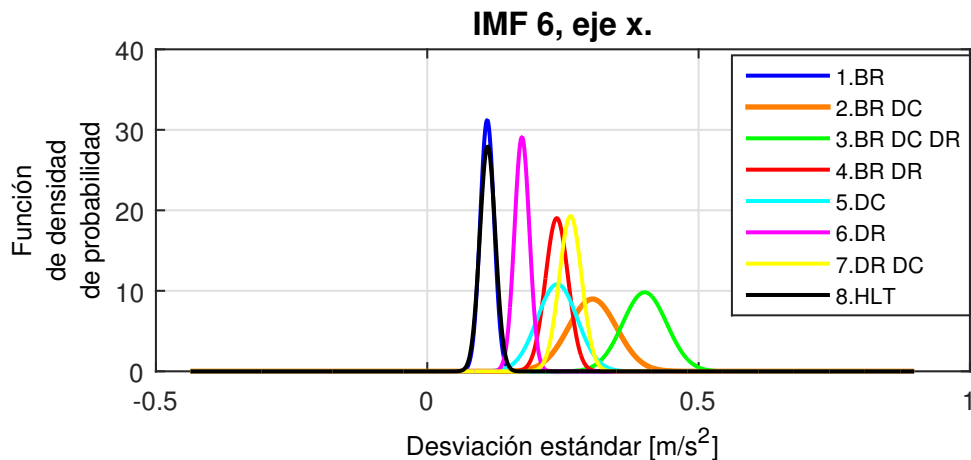


Figura 5.19. Representación de los valores de la característica de desviación estándar correspondientes al eje x en la IMF 6 de las pruebas realizadas.

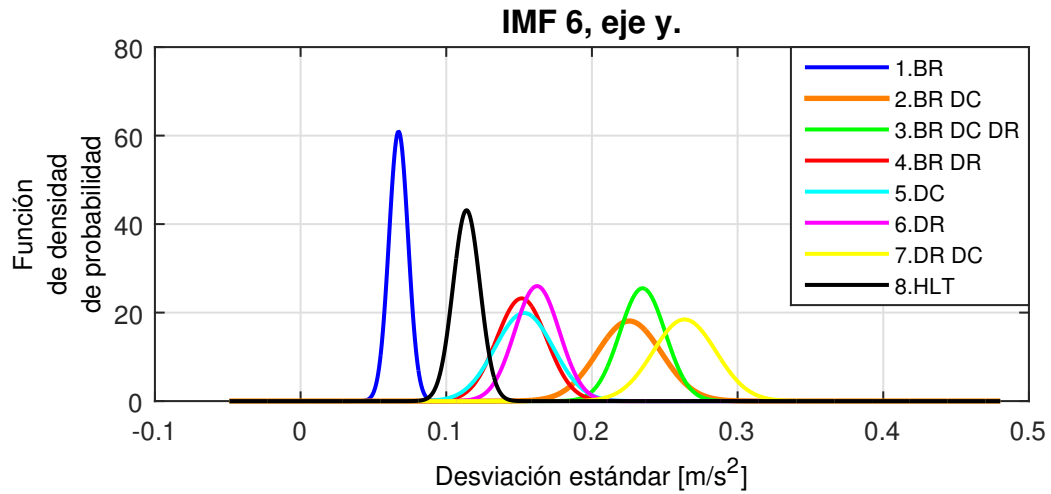


Figura 5.20. Representación de los valores de la característica de desviación estándar correspondientes al eje y en la IMF 6 de las pruebas realizadas.

Para complementar los valores representados en las Figuras 5.19 y 5.20 y todos los anteriores, también se emplearon los del eje y de la IMF 7 cuyos resultados se muestran a continuación en la Tabla 5.15.

Tabla 5.15. Media y desviación estándar de la característica de desviación estándar del eje y en la IMF 7 de las pruebas realizadas.

Nº	condición	Eje y
1.	BR	$\mu = 0,0796$, $\sigma = 0,0103$
2.	BR_DC	$\mu = 0,2430$, $\sigma = 0,0216$
3.	BR_DC_DR	$\mu = 0,2058$, $\sigma = 0,0108$
4.	BR_DR	$\mu = 0,1602$, $\sigma = 0,0077$
5.	DC	$\mu = 0,1623$, $\sigma = 0,0115$
6.	DR	$\mu = 0,1999$, $\sigma = 0,0094$
7.	DR_DC	$\mu = 0,3237$, $\sigma = 0,0427$
8.	HLT	$\mu = 0,1180$, $\sigma = 0,0086$

Finalmente, la representación de estos valores es la que se muestra en la Figura 5.21.

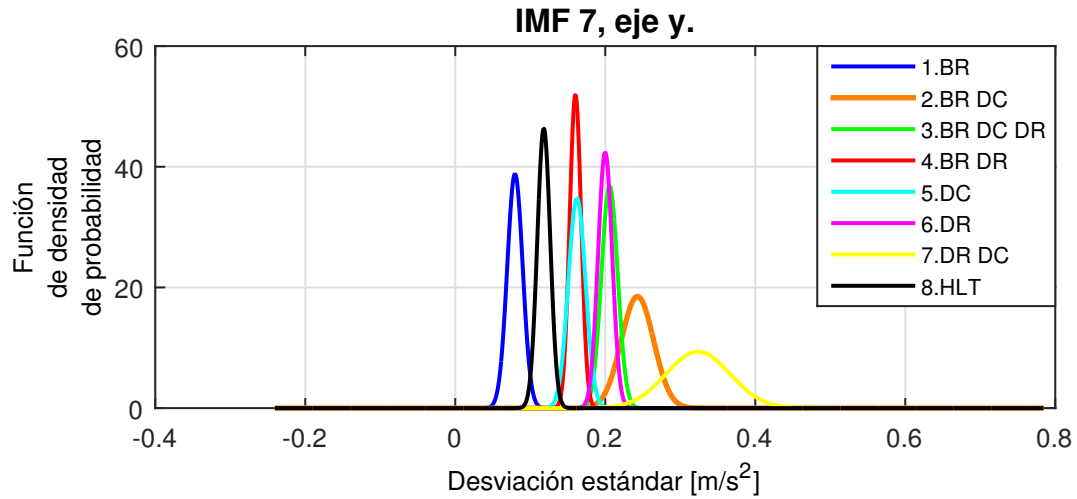


Figura 5.21. Representación de los valores de la característica de desviación estándar correspondientes al eje y en la IMF 7 de las pruebas realizadas.

Luego de realizar todo el análisis correspondiente para seleccionar los valores de los parámetros, los IMFs y los ejes más apropiados, el vector de características usado para el entrenamiento de los clasificadores en la siguiente etapa queda compuesto por:

* Variables de entrada o características.

✓ Parámetro estadístico de dispersión, desviación estándar. Compuesta de los IMFs y los ejes con mayor separación entre los valores del parámetro para cada condición de operación. Éstos son los 10 valores de desviación estándar en:

- | | |
|----------------------|-----------------------|
| 1. Eje x de IMF 1. | 6. Eje y de IMF 5. |
| 2. Eje y de IMF 1. | 7. Eje z de IMF 5. |
| 3. Eje z de IMF 1. | 8. Eje x de IMF 6. |
| 4. Eje y de IMF 4. | 9. Eje y de IMF 6. |
| 5. Eje z de IMF 4. | 10. Eje y de IMF 7. |

✓ Parámetro de complejidad, entropía. Compuesta de los IMFs y los ejes con mayor separación entre los valores del parámetro para cada condición de operación. Éstos son los nueve valores de entropía en:

- | | |
|----------------------|----------------------|
| 1. Eje z de IMF 4. | 6. Eje z de IMF 6. |
| 2. Eje y de IMF 5. | 7. Eje x de IMF 7. |
| 3. Eje z de IMF 5. | 8. Eje y de IMF 7. |
| 4. Eje x de IMF 6. | 9. Eje z de IMF 7. |
| 5. Eje y de IMF 6. | |

- * Variables de salida conocida u objetivo. Son las condiciones de operación del motor, ocho en total y cada una con 20 valores diferentes por variable de entrada.

Por lo tanto, se obtiene una matriz de características compuesta de 19 columnas, cada una correspondiente a las variables de entrada, y 160 filas, 20 filas por condición de operación.

5.3. Clasificación e identificación de fallas - Resultados

En esta etapa fueron entrenados y probados tres algoritmos de clasificación, máquinas de soporte vectorial, *AdaBoost* y árboles de decisión. La finalidad de este proceso es, que a partir de las señales de vibración correspondientes a las ocho condiciones de operación del motor, entrenar los clasificadores para diferentes señales que se obtengan con la finalidad de realizar un diagnóstico de la condición de trabajo de los motores. Esta etapa se desarrolló en dos fases, una de entrenamiento y otra de prueba. A continuación se muestran los resultados obtenidos en cada fase.

Para el entrenamiento, fue usada la matriz de características obtenida a partir de los parámetros estadístico y de complejidad; desviación estándar y entropía, respectivamente; los IMFs y los ejes con los valores más adecuados de acuerdo al grado de separación entre cada condición de operación de entrenamiento. Para la prueba, fueron usados valores sintéticos generados a partir de los valores de entropía y desviación estándar contenidos en la matriz de características usada para el entrenamiento.

5.3.1. Fase de entrenamiento - Resultados

En esta fase, cada algoritmo de clasificación fue entrenado con la matriz de características obtenida a partir de los parámetros estadístico y de complejidad; desviación estándar y entropía, respectivamente; los IMFs y los ejes con los valores más adecuados de acuerdo al grado de separación entre cada condición de operación de entrenamiento.

Después de realizar el entrenamiento correspondiente a cada algoritmo, se llevó a cabo un proceso de validación de cada uno con la finalidad de comprobar cuán predictivo es el modelo obtenido. Los métodos usados para esta validación fueron los siguientes:

1. Error de resubstitución. Corresponde a la tasa de error de clasificación que se obtiene al usar los datos de entrenamiento. Es decir, la pérdida calculada para los

datos con los cuales fue entrenado el algoritmo. Dado que el cálculo de este error es sobre los datos de entrenamiento, la medida que se obtiene es muy optimista. Sin embargo, la motivación para obtener esta medida fue para asegurar que los parámetros con los cuales se entrenó el algoritmo fueran correctos para los datos de entrenamiento.

Dado que esta medida del error de resubstitución es bastante optimista, otros métodos para medir el desempeño de los algoritmos fueron usados. En este caso, fue empleada la validación cruzada, la cual, según [102], remueve el optimismo de la resubstitución empleando datos de prueba que no son usados durante el diseño del clasificador.

2. Validación cruzada. Este método consiste en dividir los datos en dos subconjuntos, uno para el entrenamiento del algoritmo y otro para probar el desempeño. En específico, fue usada la validación cruzada de k iteraciones, conocida como *k-fold cross-validation*, en la cual, siguiendo la definición expuesta en [103], los datos iniciales son divididos de forma aleatoria en k subconjuntos o “*folds*” mutuamente excluyentes, D_1, D_2, \dots, D_k , de igual tamaño, aproximadamente. El proceso de entrenamiento y prueba se realiza k veces. En la iteración i , la partición D_i es reservada como conjunto de prueba, y las particiones restantes se usan de forma colectiva para entrenar el modelo. Por lo tanto, en la primera iteración, los subconjuntos D_2, \dots, D_k sirven de forma colectiva como el conjunto de entrenamiento para obtener el primer modelo, el cual es probado con D_1 ; en la segunda iteración se entrena con los subconjuntos D_1, D_3, \dots, D_k y se prueba con D_2 ; y así sucesivamente. En este proceso, cada muestra se usa el mismo número de veces para entrenamiento y una sola vez para prueba.
3. Validación cruzada dejando uno por fuera. También conocida como *Leave-one-out cross-validation*, *LOOCV*. Es un caso especial de la validación anterior en la cual k se ajusta al número de tuplas iniciales. Es decir, se separan los datos de forma que en cada iteración se tenga una sola muestra para los datos de prueba y todo el resto queda para los datos de entrenamiento.

De estos últimos dos métodos, se obtiene una medida del error de clasificación que indica cuánto se equivoca el clasificador en la predicción de la clase a la cual pertenecen los valores de prueba. A continuación se muestran los resultados correspondientes de estas medidas para cada algoritmo de clasificación utilizado.

5.3.1.1. Máquinas de soporte vectorial (SVMs) - Resultados de entrenamiento

Luego de realizar el entrenamiento correspondiente al algoritmo SVM mediante los pasos expuestos en el capítulo anterior, se obtuvieron los siguientes resultados:

1. Error de resubstitución. Para calcularlo se utilizó el comando de Matlab® `resubLoss(Model)`, que tiene como parámetro el modelo de clasificación que se obtuvo luego de entrenar el algoritmo. El valor resultante fue de 0% de error, lo cual indica que, como se esperaba, el algoritmo es capaz de predecir correctamente las clases a las cuales pertenecen las muestras de entrenamiento. Además, este resultado indica que los parámetros establecidos para obtener el modelo se adaptan correctamente con el conjunto de datos de entrenamiento.
2. Validación cruzada de k iteraciones. Para obtener los resultados correspondientes a esta medida, fueron considerados dos valores diferentes de k . El cálculo se realizó mediante el comando de Matlab® , `crossval`, cuyos parámetros fueron los siguientes:

$$\text{cvmdl_svm} = \text{crossval}(\text{Model_svm}, 'kfold', k);$$

donde, `cvmdl_svm` es el modelo de validación cruzada obtenido, contiene la cantidad de características con las cuales fue entrenado el algoritmo, además del número de observaciones realizadas para la validación, el nombre de las clases para las predicciones, entre otros; `Model_svm` corresponde al modelo de clasificación; `'kfold'` se usa para definir el tipo de validación cruzada a realizar; y k para definir el número de subconjuntos o “*folds*” a utilizar. Los resultados, para dos valores de k diferentes, se muestran a continuación.

k	<code>kfoldLoss(cvmdl_svm)</code>
5	0%
10	0%

Con estos resultados, se demuestra que, aún cuando el algoritmo emplea un número mayor o menor de datos de entrenamiento y prueba, es capaz de predecir correctamente la clase a la cual pertenece una muestra específica del conjunto de datos de prueba que se obtuvo con cada valor diferente de k .

3. Validación cruzada dejando uno fuera, *LOOCV*. Así como el caso anterior, esta validación fue calculada empleando el comando de Matlab® , `crossval` con

“Leaveout” en lugar de “kfold” para definir la validación tipo LOOCV. El resultado obtenido fue de 0 %.

De este resultado y todos los anteriores se concluye que, el algoritmo de clasificación basado en SVMs genera predicciones exitosas en la identificación de las clases correspondientes a los valores de prueba que se obtuvieron del conjunto de datos de entrenamiento. Esta afirmación indica que, posiblemente, el clasificador basado en SVM sea capaz de identificar y etiquetar, con errores muy mínimos, conjuntos de datos diferentes.

5.3.1.2. *AdaBoost* - Resultados de entrenamiento

Al igual que con el algoritmo anterior, las medidas de validación para *AdaBoost* fueron el error de resubstitución, validación cruzada de k iteraciones y dejando uno fuera. A su vez, tal y como se mencionó previamente, *AdaBoost* utiliza varios ciclos de aprendizaje para una determinada cantidad de clasificadores débiles. Por lo tanto, también se muestra una gráfica en la cual se refleja el comportamiento de la pérdida de resubstitución acumulada sobre el número de ciclos de aprendizaje. A continuación se muestran los resultados de cada método de validación.

1. Error de resubstitución. El valor obtenido para esta medida fue de 0 %. Con lo cual se demuestra que, al probar con los mismos valores con los cuales fue entrenado el algoritmo, esta medida se usa para comprobar la pertinencia de los parámetros establecidos durante el entrenamiento, con los datos correspondientes de la matriz de características.
2. Validación cruzada de k iteraciones. Con este método se utilizaron los mismos valores que se usaron con SVM, esto motivado a que se procuraron valores de k con los cuales se obtuviera un número entero de subconjuntos de entrenamiento y prueba. Los resultados obtenidos fueron los siguientes:

k	kfoldLoss (cvmdl_ada)
5	0,063 %
10	0 %

En este caso, el resultado obtenido con $k = 5$, un error mínimo diferente de cero, indica que, de todas las clases, el algoritmo identificó la gran mayoría para el conjunto de datos de prueba y que con datos diferentes, las predicciones serán bastante aproximadas. Por su parte, con $k = 10$, si se obtuvo un error exacto de cero, lo que indica que, para

AdaBoost, mientras mayor sean los datos del conjunto de entrenamiento, mejores serán las predicciones futuras a conjuntos de datos diferentes.

3. Validación cruzada dejando uno fuera, *LOOCV*. El valor del error obtenido para este caso fue de 0%, con lo cual se comprueba que, como ocurre con un *kfold* de 10, con este algoritmo, mientras mayor es el número de datos de entrenamiento, mejores serán las predicciones sobre un conjunto de datos diferentes.
4. Curva de comportamiento de la pérdida por resubstitución. La gráfica que se muestra a continuación en la Figura 5.22, refleja cómo se comporta la medida del error por resubstitución de los datos de entrenamiento, a medida que se aumenta la cantidad de ciclos de aprendizaje para los clasificadores débiles que usa *AdaBoost* para la creación de un clasificador fuerte.

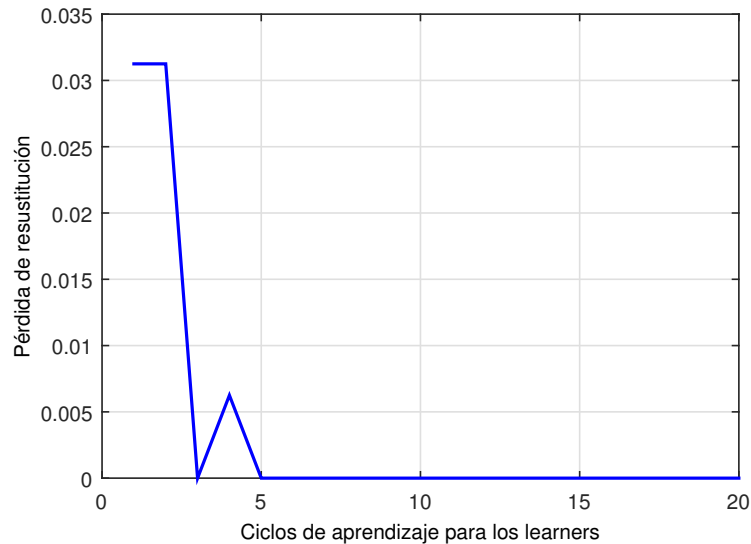


Figura 5.22. Comportamiento de la pérdida por resubstitución de *AdaBoost*.

En este gráfica, se observa que a mayor cantidad de ciclos de aprendizaje menor es la pérdida por resubstitución. Esto se debe a que, en cada ciclo de aprendizaje, los clasificadores débiles que emplea *AdaBoost* corrigen los errores del modelo obtenido en el ciclo anterior y así sucesivamente hasta que el conjunto de entrenamiento se predice perfectamente o hasta que se agota la cantidad de clasificadores débiles.

5.3.1.3. Árboles de decisión - Resultados de entrenamiento

Para este algoritmo se emplearon los mismos métodos de validación que con los algoritmos anteriores. A continuación se muestran los resultados correspondientes.

1. Error de resubstitución. El resultado de esta medida fue de 0 % y, al igual que con los casos anteriores, esto comprueba que los parámetros definidos para el algoritmo se ajustan correctamente a los datos de entrenamiento.
2. Validación cruzada de k iteraciones. Al igual que en los casos anteriores, se usaron dos valores de k , los resultados se muestran a continuación:

k	kfoldLoss (cvmdl_ade)
5	0,0438 %
10	0,05 %

Mediante los resultados de validación cruzada obtenidos con $k=5$ y $k=10$, se observa que, a pesar de no ser tan exactos como lo fue para los dos algoritmos previos, sí es un buen valor. Esta diferencia con respecto a SVM y *AdaBoost* se debe a que, árboles de decisión, por tratarse de un procedimiento donde la distinción entre clases se hace por medio de condiciones simples de decisión, puede ser muy sensible cuando los valores, representativos de las clases, son muy cercanos entre sí y la diferencia entre uno y otro es muy pequeña, del orden de los decimales.

Sin embargo, ambos son buenos valores con los cuales se comprueba la pertinencia de los parámetros definidos para el algoritmo con los datos de entrenamiento y también, con datos diferentes para predicciones futuras.

3. Validación cruzada dejando uno fuera, *LOOCV*. En este caso se obtuvo un error de 0,0313 %, el cual, tampoco es tan exacto como en SVM y *AdaBoost*. Esta situación se debe al procedimiento que realiza árboles de decisión para clasificar. Además, se comprueba que, por más que se aumente la cantidad de datos entrenamiento, si los valores de cada clase son muy cercanos entre sí, el algoritmo tendrá errores de clasificación mínimos.

5.3.2. Fase de pruebas - Resultados

Una vez concluido el proceso de entrenamiento y validación de cada algoritmo de clasificación, se procedió a realizar la prueba de cada modelo obtenido. El conjunto de datos de prueba, para cada condición de operación, fue obtenido de forma sintética al producir aleatoriamente n valores de las características desviación estándar y entropía, dentro del rango $[\mu - \sigma, \mu + \sigma]$, donde μ es la media y σ es la desviación estándar de los valores de ambas características extraídas de los modos de oscilación de las señales de vibración reales, obtenidas en las 20 pruebas que se realizaron durante la adquisición de señales.

Para validar esta fase, se emplearon matrices de confusión que se obtuvieron a partir de los modelos de clasificación de cada algoritmo con los valores de prueba generados de forma sintética. En cada matriz se refleja el desempeño de cada algoritmo, por medio de la cantidad de datos acertados y errados durante la clasificación, para cada clase. Es importante recordar que cada clase representa una condición de operación del motor en específico, de la siguiente forma:

- ~ Clase 1: condición de motor con barra rota.
- ~ Clase 2: condición de motor con barra rota y desbalance de carga.
- ~ Clase 3: condición de motor con barra rota, desbalance de carga y defecto en rodamientos.
- ~ Clase 4: condición de motor con barra rota y defecto en rodamientos.
- ~ Clase 5: condición de motor con desbalance de carga.
- ~ Clase 6: condición de motor con defecto en rodamientos.
- ~ Clase 7: condición de motor con defecto en rodamientos y desbalance de carga.
- ~ Clase 8: condición de motor sano.

Además, se emplearon tres tamaños para el conjunto de datos de prueba, es decir se usaron tres valores distintos de n . En las matrices obtenidas, cada fila representa el número de predicciones de cada clase, mientras que cada columna representa a las instancias en la clase real. Los resultados se muestran a continuación.

5.3.2.1. Matriz de confusión para el algoritmo de máquinas de soporte vectorial (*SVMs*)

- ✓ Para $n = 20$.

Matriz de confusión para SVM

Clase predicha	1	20 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	0 0.0%	20 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	3	0 0.0%	0 0.0%	20 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	4	0 0.0%	0 0.0%	0 0.0%	20 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	20 12.5%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	20 12.5%	0 0.0%	0 0.0%	100% 0.0%
	7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	20 12.5%	0 0.0%	100% 0.0%
	8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	20 12.5%	100% 0.0%
			100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%
		1	2	3	4	5	6	7	8	
		Clase real								

Figura 5.23. Matriz de confusión para SVM con 20 valores sintéticos.

✓ Para $n = 50$.

Matriz de confusión para SVM

Clase predicha	1	50 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	0 0.0%	48 12.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	3	0 0.0%	0 0.0%	50 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	4	0 0.0%	0 0.0%	0 0.0%	50 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	50 12.5%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	50 12.5%	0 0.0%	0 0.0%	100% 0.0%
	7	0 0.0%	2 0.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	50 12.5%	0 0.0%	96.2% 3.8%
	8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	50 12.5%	100% 0.0%
			100% 0.0%	96.0% 4.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	99.5% 0.5%
		1	2	3	4	5	6	7	8	
		Clase real								

Figura 5.24. Matriz de confusión para SVM con 50 valores sintéticos.

✓ Para $n = 100$.

Matriz de confusión para SVM

	1	2	3	4	5	6	7	8	
1	100 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	98 12.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	100 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	100 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 12.5%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 12.5%	0 0.0%	0 0.0%	100% 0.0%
7	0 0.0%	2 0.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 12.5%	0 0.0%	98.0% 2.0%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 12.5%	100% 0.0%
	100% 0.0%	98.0% 2.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	99.8% 0.2%
	1	2	3	4	5	6	7	8	
	Clase real								

Figura 5.25. Matriz de confusión para SVM con 100 valores sintéticos.

Con los resultados obtenidos para cada tamaño de los valores sintéticos se comprueba que, mediante las características de desviación estándar y entropía, es posible identificar con más de 99% de efectividad las ocho condiciones de operación del motor propuestas. La condición que presenta menor porcentaje de acierto es la de motor con barra rota y desbalance de carga, como se refleja en las matrices de las Figuras 5.24 y 5.25 donde se observa que predicen erróneamente dos muestras y la asigna a la clase siete o condición de motor con defecto en rodamientos y desbalance de carga. Sin embargo, el error en esta predicción se ubica entre el 2% y 4%, lo cual no representa un gran riesgo para predicciones futuras con conjuntos de datos diferentes.

5.3.2.2. Matriz de confusión para el algoritmo *AdaBoost*

✓ Para $n = 20$.

Matriz de confusión para AdaBoost

	1	2	3	4	5	6	7	8	
1	20 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	20 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	20 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	20 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	20 12.5%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	20 12.5%	0 0.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	20 12.5%	0 0.0%	100% 0.0%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	20 12.5%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%
	1	2	3	4	5	6	7	8	
	Clase real								

Figura 5.26. Matriz de confusión para AdaBoost con 20 valores sintéticos.

✓ Para $n = 50$.

Matriz de confusión para AdaBoost

	1	2	3	4	5	6	7	8	
1	50 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	49 12.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	50 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	50 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	50 12.5%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	50 12.5%	0 0.0%	0 0.0%	100% 0.0%
7	0 0.0%	1 0.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	50 12.5%	0 0.0%	98.0% 2.0%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	50 12.5%	100% 0.0%
	100% 0.0%	98.0% 2.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	99.8% 0.2%
	1	2	3	4	5	6	7	8	
	Clase real								

Figura 5.27. Matriz de confusión para AdaBoost con 50 valores sintéticos.

✓ Para $n = 100$.

Matriz de confusión para AdaBoost

	1	2	3	4	5	6	7	8	
1	100 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	100 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	100 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	100 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 12.5%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 12.5%	0 0.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 12.5%	0 0.0%	100% 0.0%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 12.5%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%
	1	2	3	4	5	6	7	8	
	Clase real								

Figura 5.28. Matriz de confusión para AdaBoost con 100 valores sintéticos.

En este caso, al igual que con SVM, el algoritmo falla, con un mínimo error, en la predicción de una muestra que corresponde a la clase dos asignándola a la clase siete, como se refleja en la matriz correspondiente a 50 muestras de prueba, Figura 5.27. Sin embargo, el error en la predicción es mínimo y el algoritmo es capaz de predecir correctamente los casos donde estas condiciones de operación están aisladas (clases uno, cinco y seis).

5.3.2.3. Matriz de confusión para el algoritmo de árboles de decisión

✓ Para $n = 20$.

Matriz de confusión para árbol de decisión

Clase predicha	1	19 11.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	2	0 0.0%	20 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	3	0 0.0%	0 0.0%	20 12.5%	0 0.0%	1 0.6%	0 0.0%	0 0.0%	0 0.0%	95.2% 4.8%
	4	0 0.0%	0 0.0%	0 0.0%	19 11.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	5	0 0.0%	0 0.0%	0 0.0%	1 0.6%	19 11.9%	0 0.0%	0 0.0%	0 0.0%	95.0% 5.0%
	6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	20 12.5%	0 0.0%	0 0.0%	100% 0.0%
	7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	20 12.5%	0 0.0%	100% 0.0%
	8	1 0.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	20 12.5%	95.2% 4.8%
			95.0% 5.0%	100% 0.0%	100% 0.0%	95.0% 5.0%	95.0% 5.0%	100% 0.0%	100% 0.0%	100% 0.0%
		1	2	3	4	5	6	7	8	
		Clase real								

Figura 5.29. Matriz de confusión para árboles de decisión con 20 valores sintéticos.

✓ Para $n = 50$.

Matriz de confusion para árbol de decisión

Clase predicha	1	50 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.5%	96.2% 3.8%
	2	0 0.0%	48 12.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	3	0 0.0%	0 0.0%	50 12.5%	0 0.0%	3 0.8%	0 0.0%	0 0.0%	0 0.0%	94.3% 5.7%
	4	0 0.0%	0 0.0%	0 0.0%	48 12.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
	5	0 0.0%	0 0.0%	0 0.0%	2 0.5%	46 11.5%	0 0.0%	0 0.0%	0 0.0%	95.8% 4.2%
	6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.3%	49 12.3%	0 0.0%	0 0.0%	98.0% 2.0%
	7	0 0.0%	2 0.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	50 12.5%	0 0.0%	96.2% 3.8%
	8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.3%	0 0.0%	48 12.0%	98.0% 2.0%
			100% 0.0%	96.0% 4.0%	100% 0.0%	96.0% 4.0%	92.0% 8.0%	98.0% 2.0%	100% 0.0%	96.0% 4.0%
		1	2	3	4	5	6	7	8	
		Clase real								

Figura 5.30. Matriz de confusión para árboles de decisión con 50 valores sintéticos.

✓ Para $n = 100$.

Matriz de confusión para árbol de decisión

	1	2	3	4	5	6	7	8	
1	99 12.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	100 12.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	100 12.5%	0 0.0%	4 0.5%	0 0.0%	0 0.0%	0 0.0%	96.2% 3.8%
4	0 0.0%	0 0.0%	0 0.0%	94 11.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	6 0.8%	96 12.0%	0 0.0%	0 0.0%	0 0.0%	94.1% 5.9%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	97 12.1%	0 0.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100 12.5%	0 0.0%	100% 0.0%
8	1 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 0.4%	0 0.0%	100 12.5%	96.2% 3.8%
	99.0% 1.0%	100% 0.0%	100% 0.0%	94.0% 6.0%	96.0% 4.0%	97.0% 3.0%	100% 0.0%	100% 0.0%	98.3% 1.7%
	1	2	3	4	5	6	7	8	
	Clase real								

Figura 5.31. Matriz de confusión para árboles de decisión con 100 valores sintéticos.

En este caso, al igual que en los anteriores, se observa que las clases dos y siete son las menos distinguibles entre sí. Ambas clases contienen la condición de motor con desbalance de carga, ya que su detección depende de cuan grave sea la falla y también de la carga mecánica del motor; cuanto mayor es la gravedad de la falla y la carga mecánica, más fácil la detección. Para el resto de las predicciones, y en comparación con SVM y *AdaBoost*, el algoritmo de árboles de decisión no genera un buen clasificador que detecte de forma efectiva las condiciones de operación propuestas. A pesar de que el porcentaje de error, para los tres tamaños del conjunto de datos de prueba, es mínimo, se prefieren los algoritmos de SVM o *AdaBoost* para futuras predicciones con conjuntos de datos diferentes.

Es perceptible el grado de efectividad que presentan los dos algoritmos iniciales, SVM y *AdaBoost*, por encima de árboles de decisión. Esta diferencia se debe al modo de operación de cada algoritmo. En primer lugar, SVM define un margen de clasificación considerando las muestras que se ubican más cercana a la frontera entre clases, conocidas como vectores de soporte. En segundo lugar, la metodología de operación de *AdaBoost* consiste en que, en cada ciclo de aprendizaje, los clasificadores débiles corrigen los errores generados en el ciclo anterior. Estas características, tanto de SVM como de *AdaBoost*, le otorgan a estos algoritmos mayor preferencia por encima de clasificadores basados en árboles de decisión, cuyo rendimiento se refleja en los resultados mostrados previamente.

Capítulo 6

Conclusiones

En este trabajo es propuesta una metodología para la detección de fallas múltiples y combinadas en motores de inducción. El trabajo está orientado al uso del procesamiento de señales para supervisar la condición de operación de estos dispositivos, mediante el análisis de las señales de vibración provenientes de un motor con diferentes condiciones de operación.

A través de los resultados obtenidos se demostró la factibilidad en la detección de fallas, como barra rota, defectos en rodamientos, desbalance de carga o la combinación de estas tres, en motores de inducción que operan en estado estable, por medio del análisis de las señales de vibración. Para este análisis, fue usada la descomposición de modo empírico por conjuntos completos, CEEMD, con la cual fue posible la separación de las señales en sus modos de oscilación intrínsecos, o IMFs, físicamente significativos.

Con esta descomposición, además de extraer los modos de oscilación, fue posible generar una mayor cantidad de muestras para las inferencias estadísticas y de complejidad que se realizaron posteriormente con la extracción de características representativas de la información contenida en los modos.

La extracción de los parámetros estadísticos y de complejidad, desviación estándar y entropía, respectivamente, permitió construir una matriz de características que resumían, eficazmente y con buen grado de discriminación, el conjunto de datos e información presentes en las señales referentes a cada una de las condiciones de operación del motor.

A partir de los clasificadores obtenidos, se validó la extracción y posterior selección de características que resumen la información contenida en las señales de vibración de entrenamiento. Además, los clasificadores permiten la identificación y el etiquetado de las ocho condiciones de operación propuestas en este trabajo. Tanto SVM como

AdaBoost presentaron errores de clasificación mínimos, los cuales demuestran la pertinencia de uno u otro algoritmo para futuras predicciones con conjuntos de datos diferentes.

La combinación de un método de descomposición de señales, como CEEMD, junto con la extracción de características como información de entropía y un algoritmo de clasificación como SVM o *AdaBoost* resultó ser altamente eficiente en la detección de fallas individuales y combinadas. Los resultados presentados demuestran que, como se ha probado en investigaciones previas, la información de entropía es adecuada para abordar problemas complejos así como también para el diagnóstico de la condición de operación de motores de inducción.

Es importante mencionar que, para realizar una comparación de la metodología propuesta con trabajos anteriores, es necesario contar con los métodos o el conjunto de datos empleados en dichos trabajos. Es por ello que, estas comparaciones no son presentadas. Sin embargo, con los clasificadores fue validada la selección de características, realizada para la identificación de la condición de operación en motores de inducción.

Finalmente, el aporte de esta investigación es una metodología basada en la extracción de parámetros, característicos de una señal como la entropía y la desviación estándar, a partir de los modos de oscilación presentes en señales de vibración reales, referentes a un motor con diferentes condiciones de operación. A su vez, el uso de algoritmos de aprendizaje de máquina para la clasificación y posterior identificación de la condición de trabajo de los motores de inducción.

Bibliografía

- [1] S. A. S. Al Kazzaz and G. K. Singh, “Experimental investigations on induction machine condition monitoring and fault diagnosis using digital signal processing techniques,” *Electric Power Systems Research*, vol. 65, pp. 197–221, June 2003.
- [2] E. Cabal-Yepez, R. A. Osornio-Rios, R. J. Romero-Troncoso, J. R. Razo-Hernandez, and R. Lopez-Garcia, “FPGA-based online induction motor multiple-fault detection with fused FFT and Wavelet analysis,” in *2009 International Conference on Reconfigurable Computing and FPGAs*, pp. 101–106, Dec. 2009.
- [3] S. H. Kia, H. Henao, and G. A. Capolino, “Diagnosis of broken-bar fault in induction machines using discrete Wavelet transform without slip estimation,” *IEEE Transactions on Industry Applications*, vol. 45, pp. 1395–1404, July 2009.
- [4] Y. Lei, J. Lin, Z. He, and M. J. Zuo, “A review on empirical mode decomposition in fault diagnosis of rotating machinery,” *Mechanical Systems and Signal Processing*, vol. 35, pp. 108–126, Feb. 2013.
- [5] P. A. Delgado-Arredondo, D. Morinigo-Sotelo, R. A. Osornio-Rios, J. G. Avina-Cervantes, H. Rostro-Gonzalez, and R. J. Romero-Troncoso, “Methodology for fault detection in induction motors via sound and vibration signals,” *Mechanical Systems and Signal Processing*, vol. 83, pp. 568–589, Jan. 2017.
- [6] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, “The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis,” in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, no. 1971 in 454, pp. 903–995, The Royal Society, 1998.
- [7] X. Chen, C. Yin, and W. He, “Feature extraction of gearbox vibration signals based on EEMD and sample entropy,” in *2013 10th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 811–815, July 2013.

- [8] J. Xiang and Y. Zhong, “A fault detection strategy using the enhancement ensemble empirical mode decomposition and random decrement technique,” *Microelectronics Reliability*, 2017.
- [9] L. Luo, Y. Yan, P. Xie, J. Sun, Y. Xu, and J. Yuan, “Hilbert–Huang transform, Hurst and chaotic analysis based flow regime identification methods for an airlift reactor,” *Chemical Engineering Journal*, vol. 181–182, pp. 570–580, Feb. 2012.
- [10] S. Charleston-Villalobos, R. González-Camarena, G. Chi-Lem, and T. Aljama-Corrales, “Crackle sounds analysis by empirical mode decomposition,” *IEEE Engineering in medicine and biology magazine*, vol. 26, no. 1, p. 40, 2007.
- [11] S.-D. Wu, J.-C. Chiou, and E. Goldman, “Solution for mode mixing phenomenon of the empirical mode decomposition,” in *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, vol. 2, pp. V2–500–V2–504, Aug. 2010.
- [12] N. E. Huang, M.-L. C. Wu, S. R. Long, S. S. P. Shen, W. Qu, P. Gloersen, and K. L. Fan, “A confidence limit for the empirical mode decomposition and Hilbert spectral analysis,” *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 459, pp. 2317–2345, Sept. 2003.
- [13] Z. Wu and N. E. Huang, “Ensemble empirical mode decomposition: a noise-assisted data analysis method,” *Advances in Adaptive Data Analysis*, vol. 01, pp. 1–41, Jan. 2009.
- [14] M. E. Torres, M. A. Colominas, G. Schlotthauer, and P. Flandrin, “A complete ensemble empirical mode decomposition with adaptive noise,” in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4144–4147, May 2011.
- [15] A. Garcia-Perez, O. Ibarra-Manzano, and R. J. Romero-Troncoso, “Analysis of partially broken rotor bar by using a novel empirical mode decomposition method,” in *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*, pp. 3403–3408, Oct. 2014.
- [16] L. Tang, W. Dai, L. Yu, and S. Wang, “A novel CEEMD-based EELM ensemble learning paradigm for crude oil price forecasting,” *International Journal of Information Technology & Decision Making*, vol. 14, pp. 141–169, Oct. 2014.
- [17] J. Han and M. van der Baan, “Empirical mode decomposition for seismic time-frequency analysis,” *Journal of Geophysics*, vol. 78, pp. 9–19, Feb. 2013.
- [18] K. Fukunaga, “Chapter 9 - Feature extraction and linear mapping for signal representation,” in *Introduction to Statistical Pattern Recognition (Second Edition)*, pp. 399–440, Boston: Academic Press, 1990. DOI: 10.1016/B978-0-08-047865-4.50015-6.

- [19] E. Cabal-Yepez, M. Valtierra-Rodriguez, R. J. Romero-Troncoso, A. Garcia-Perez, R. A. Osornio-Rios, H. Miranda-Vidales, and R. Alvarez-Salas, "FPGA-based entropy neural processor for online detection of multiple combined faults on induction motors," *Mechanical Systems and Signal Processing*, vol. 30, pp. 123–130, 2012.
- [20] Y. Yu, YuDejie, and C. Junsheng, "A roller bearing fault diagnosis method based on EMD energy entropy and ANN," *Journal of Sound and Vibration*, vol. 294, pp. 269–277, June 2006.
- [21] Y. Zhang, S. Huang, W. Gao, and T. Shen, "Vibration fault diagnosis of steam turbine shafting based on probability neural networks," in *2008 Congress on Image and Signal Processing*, vol. 5, pp. 582–585, May 2008.
- [22] E. Cabal-Yepez, R. J. Romero-Troncoso, A. Garcia-Perez, R. A. Osornio-Rios, and R. Alvarez-Salas, "Multiple fault detection through information entropy analysis in ASD-fed induction motors," in *8th IEEE Symposium on Diagnostics for Electrical Machines, Power Electronics Drives*, pp. 391–396, Sept. 2011.
- [23] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.
- [24] C. Bishop, *Neural Networks for Pattern Recognition*. Advanced Texts in Econometrics, Clarendon Press, 1995.
- [25] S. D. Bay, "Combining nearest neighbor classifiers through multiple feature subsets.," in *Proceedings of the 17th International Conference on Machine Learning*, vol. 98, pp. 37–45, Citeseer, Madison, WI, 1998.
- [26] I. Rish, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, no. 22 in 3, pp. 41–46, IBM New York, 2001.
- [27] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [28] R. E. Sánchez-Yáñez, E. V. Kurmyshev, and F. J. Cuevas, "A framework for texture classification using the coordinated clusters representation," *Pattern Recognition Letters*, vol. 24, pp. 21–31, Jan. 2003.
- [29] R. A. Lizarraga-Morales, C. Rodriguez-Donate, E. Cabal-Yepez, M. Lopez-Ramirez, L. M. Ledesma-Carrillo, and E. R. Ferrucho-Alvarez, "Novel FPGA-based methodology for early broken rotor bar detection and classification through homogeneity estimation," *IEEE Transactions on Instrumentation and Measurement*, vol. PP, no. 99, pp. 1–10, 2017.

- [30] P. Konar and P. Chattopadhyay, “Bearing fault detection of induction motor using wavelet and Support Vector Machines (SVMs),” *Applied Soft Computing*, vol. 11, pp. 4203–4211, Sept. 2011.
- [31] K. Ágoston, “Fault detection of the electrical motors based on vibration analysis,” *Procedia Technology*, vol. 19, pp. 547–553, Jan. 2015.
- [32] X. Xue, J. Zhou, Y. Xu, W. Zhu, and C. Li, “An adaptively fast ensemble empirical mode decomposition method and its applications to rolling element bearing fault diagnosis,” *Mechanical Systems and Signal Processing*, vol. 62–63, pp. 444–459, Oct. 2015.
- [33] L. Zhao, W. Yu, and R. Yan, “Rolling bearing fault diagnosis based on CEEMD and time series modeling,” *Mathematical Problems in Engineering*, vol. 2014, July 2014.
- [34] S. Chapman, *Máquinas eléctricas (5a.ed.)*. McGraw Hill México, 2012.
- [35] LearnEngineering, “How does an induction motor work?” <http://www.learnengineering.org/2013/08/three-phase-induction-motor-working-squirrel-cage.html>, 2013.
- [36] S. Karmakar, S. Chattopadhyay, M. Mitra, and S. Sengupta, “Induction Motor and Faults,” in *Induction Motor Fault Diagnosis - Approach through Current Signature Analysis*, Power Systems, pp. 7–28, Springer Singapore, 2016. DOI: 10.1007/978-981-10-0624-1_2.
- [37] M. Say, *Performance & Design A.C. Machines*. CBS Publishers & Distributors, illustrated ed., 2005.
- [38] S. Karmakar, S. Chattopadhyay, M. Mitra, and S. Sengupta, “Introduction,” in *Induction Motor Fault Diagnosis - Approach through Current Signature Analysis*, Power Systems, pp. 1–6, Springer Singapore, 2016. DOI: 10.1007/978-981-10-0624-1_1.
- [39] “Parts of 3 phase induction motor.” <http://www.easysparepart.com/blog/what-are-parts-of-3-phase-induction-motor-.html>, 2014.
- [40] D. Inman, *Engineering Vibration*. Pearson Prentice Hall, third ed., 2008.
- [41] C. Wang and C. Wang, *Structural Vibration: Exact Solutions for Strings, Membranes, Beams, and Plates*. Taylor & Francis, illustrated ed., 2013.
- [42] A. Garcia-Perez, R. J. Romero-Troncoso, E. Cabal-Yepez, and R. A. Osornio-Rios, “The application of high-resolution spectral analysis for identifying multiple combined faults in induction motors,” *IEEE Transactions on Industrial Electronics*, vol. 58, pp. 2002–2010, May 2011.

- [43] O. V. Thorsen and M. Dalva, “A survey of faults on induction motors in offshore oil industry, petrochemical industry, gas terminals, and oil refineries,” *IEEE Transactions on Industry Applications*, vol. 31, pp. 1186–1196, Sept. 1995.
- [44] A. Bjorn, “Motor asíncrono trifásico..” <https://sites.google.com/site/bjorn512/>, 2012.
- [45] ticgalicia, “El motor de jaula de ardilla. La base de los coches eléctricos..” <http://www.ticgalicia.com/2016/06/el-motor-de-jaula-de-ardilla-la-base-de.html>, 2016.
- [46] Ž. Kanović, D. Matić, Z. Jeličić, M. Rapaić, B. Jakovljević, and M. Kapetina, “Induction motor broken rotor bar detection using vibration analysis - A case study,” in *2013 9th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives (SDEMPED)*, pp. 64–68, Aug. 2013.
- [47] M. Neale, Associates, and G. B. C. for Terotechnology, *A Guide to the condition monitoring of machinery*. H.M. Stationery Office, 1979.
- [48] R. J. Romero-Troncoso, R. Saucedo-Gallaga, E. Cabal-Yepez, A. Garcia-Perez, R. A. Osornio-Rios, R. Alvarez-Salas, H. Miranda-Vidales, and N. Huber, “FPGA-based online detection of multiple combined faults in induction motors through information entropy and fuzzy inference,” *IEEE Transactions on Industrial Electronics*, vol. 58, no. 11, pp. 5263–5270, 2011.
- [49] Y. Gritli, A. O. D. Tommaso, R. Miceli, C. Rossi, and F. Filippetti, “Diagnosis of mechanical unbalance for double cage induction motor load in time-varying conditions based on motor vibration signature analysis,” in *2013 International Conference on Renewable Energy Research and Applications (ICRERA)*, pp. 1157–1162, Oct. 2013.
- [50] A. Bellini, F. Filippetti, C. Tassoni, and G. A. Capolino, “Advances in diagnostic techniques for induction machines,” *IEEE Transactions on Industrial Electronics*, vol. 55, no. 12, pp. 4109–4126, 2008.
- [51] Istockphoto, “Fotos de Ball Bearing e Imágenes de Stock..” <http://www.istockphoto.com/mx/fotos/ball-bearing?excludenudity=true&phrase=ball%20bearing&sort=mostpopular>, 2014.
- [52] S. Nandi, H. A. Toliyat, and X. Li, “Condition monitoring and fault diagnosis of electrical motors-a review,” *IEEE Transactions on Energy Conversion*, vol. 20, pp. 719–729, Dec. 2005.
- [53] P. Eschmann, L. Hasbargen, Weigand, J. Brändlein, and F. K. G. S. KGaA., *Ball and roller bearings: theory, design, and application*. R. Oldenbourg, second, illustrated ed., 1985.

- [54] M. J. M. Gonçalves, R. C. Creppe, E. G. Marques, and S. M. A. Cruz, “Diagnosis of bearing faults in induction motors by vibration signals - Comparison of multiple signal processing approaches,” in *2015 IEEE 24th International Symposium on Industrial Electronics (ISIE)*, pp. 488–493, June 2015.
- [55] H. Toliyat, S. Nandi, S. Choi, and H. Meshgin-Kelk, *Electric Machines: Modeling, Condition Monitoring, and Fault Diagnosis*. Taylor & Francis, illustrated ed., 2012.
- [56] B. Li, M. Y. Chow, Y. Tipsuwan, and J. C. Hung, “Neural-network-based motor rolling bearing fault diagnosis,” *IEEE Transactions on Industrial Electronics*, vol. 47, pp. 1060–1069, Oct. 2000.
- [57] S. Mitra and C. Koley, “Vibration signal analysis of induction motors used in process control operation,” in *2013 IEEE 1st International Conference on Condition Assessment Techniques in Electrical Systems (CATCON)*, pp. 286–291, Dec. 2013.
- [58] W. He, C. Yin, and X. Chen, “Nonlinear methods for rolling bearing fault diagnosis,” in *2013 Ninth International Conference on Natural Computation (ICNC)*, pp. 168–172, July 2013.
- [59] L. Zão, R. Coelho, and P. Flandrin, “Speech enhancement with EMD and Hurst-based mode selection,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, pp. 899–911, May 2014.
- [60] P. J. Oonincx and J. Hermand, “Empirical mode decomposition of ocean acoustic data with constraint on the frequency range,” *Proceedings of the Seventh European Conference on Underwater Acoustics, ECUA 2004*, 2004.
- [61] M. C. Chen, L. S. Chen, and Y. Wei, “Apply ensemble empirical mode decomposition to discover time variants of metro station passenger flow,” in *2017 4th International Conference on Industrial Engineering and Applications (ICIEA)*, pp. 239–243, Apr. 2017.
- [62] M. Kang, M. R. Islam, J. Kim, J. M. Kim, and M. Pecht, “A hybrid feature selection scheme for reducing diagnostic performance deterioration caused by outliers in data-driven diagnostics,” *IEEE Transactions on Industrial Electronics*, vol. 63, pp. 3299–3310, May 2016.
- [63] I. A. Rezek and S. J. Roberts, “Stochastic complexity measures for physiological signal analysis,” *IEEE Transactions on Biomedical Engineering*, vol. 45, pp. 1186–1191, Sept. 1998.
- [64] C. Bandt and B. Pompe, “Permutation entropy: A natural complexity measure for time series,” *Physical Review Letters*, vol. 88, p. 174102, Apr. 2002.

- [65] C. E. Shannon, “The mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, pp. 379–423, 1948.
- [66] P. Grunwald and P. Vitányi, “Shannon information and kolmogorov complexity,” *arXiv preprint cs/0410002*, 2004.
- [67] W. L. Qin, W. J. Zhang, and C. Lu, “Rolling bearing fault diagnosis: A data-based method using eemd, information entropy and one-versus-one svm,” in *2016 12th World Congress on Intelligent Control and Automation (WCICA)*, pp. 1016–1020, June 2016.
- [68] V. N. Vapnik and V. Vapnik, *Statistical Learning Theory*, vol. 1. Wiley New York, 1998.
- [69] E. Alpaydin, *Introduction to Machine Learning*. Adaptive computation and machine learning, MIT Press, Aug. 2014.
- [70] L. Wang, *Support Vector Machines: Theory and Applications*, vol. 177 of *Studies in Fuzziness and Soft Computing*. Springer-Verlag Berlin Heidelberg, 1 ed., 2005.
- [71] G. Mitchell, S. Bahadoorsingh, N. Ramsamooj, and C. Sharma, “A comparison of artificial neural networks and support vector machines for short-term load forecasting using various load types,” in *2017 IEEE Manchester PowerTech*, pp. 1–4, June 2017.
- [72] K. C. Lin, T. Y. Liu, P. H. Chen, and C. T. Lin, “Use Support Vector Machine (SVM) to estimate gas concentration in mixture condition,” in *2017 International Conference on Applied System Innovation (ICASI)*, pp. 744–746, May 2017.
- [73] A. Ben-Hur and J. Weston, “A User’s Guide to Support Vector Machines,” in *Data Mining Techniques for the Life Sciences*, Methods in Molecular Biology, pp. 223–239, Humana Press, 2010. DOI: 10.1007/978-1-60327-241-4.13.
- [74] A. Guazzelli, “Predicciones sobre el futuro, parte 2. Técnicas de modelado predictivo.” <https://www.ibm.com/developerworks/ssa/industry/library/ba-predictive-analytics2/index.html>, 2012.
- [75] OpenCV, “Introduction to Support Vector Machines..” http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html, 2013.
- [76] T. Fletcher, “Support vector machines explained,” *University College London, London*, 2009.
- [77] P. Enigma, “Support Vector Machines..” <https://prateekvjoshi.com/2012/08/24/support-vector-machines/>, 2014.

- [78] K. in “Eight to Late”, “A gentle introduction to support vector machines using R..” <https://eight2late.wordpress.com/2017/02/07/a-gentle-introduction-to-support-vector-machines-using-r/>, 2017.
- [79] E. Golge, “How does the Kernel Trick enable SVMs to perform non-linear classification?.” <https://www.quora.com/How-does-the-Kerne-Trick-enable-SVMs-to-perform-non-linear-classification>, 2012.
- [80] K. Aida-zade, A. Xocayev, and S. Rustamov, “Speech recognition using Support Vector Machines,” in *2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT)*, pp. 1–4, Oct. 2016.
- [81] V. Vapnik and A. Lerner, “Pattern Recognition using Generalized Portrait Method,” *Automation and Remote Control*, vol. 24, pp. 774–780, 1963.
- [82] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152, ACM, July 1992.
- [83] Y. Freund and R. E. Schapire, “A desicion-theoretic generalization of on-line learning and an application to boosting,” in *European conference on computational learning theory*, pp. 23–37, Springer, 1995.
- [84] Y. Freund, R. E. Schapire, *et al.*, “Experiments with a new boosting algorithm,” in *Proceedings of the 13th International Conference on Machine Learning*, vol. 96, pp. 148–156, 1996.
- [85] R. E. Schapire, “The Boosting Approach to Machine Learning: An Overview,” in *Nonlinear Estimation and Classification*, Lecture Notes in Statistics, pp. 149–171, Springer, New York, NY, 2003. DOI: 10.1007/978-0-387-21579-2_9.
- [86] G. Zhao, H. Huang, and X. Lu, “Discriminating earthquakes and explosion events by seismic signals basing on BP-Adaboost classifier,” in *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pp. 1965–1969, Oct. 2016.
- [87] R. Sonavane and P. Sonar, “Classification and segmentation of brain tumor using Adaboost classifier,” in *2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC)*, pp. 396–403, Dec. 2016.
- [88] W. Hu, W. Hu, and S. Maybank, “AdaBoost-Based Algorithm for Network Intrusion Detection,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, pp. 577–583, Apr. 2008.

- [89] L. Lu, L. Di, and Y. Ye, “A Decision-Tree Classifier for Extracting Transparent Plastic-Mulched Landcover from Landsat-5 TM Images,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, pp. 4548–4558, Nov. 2014.
- [90] D. D. Berrar and W. Dubitzky, “Decision Tree,” in *Encyclopedia of Systems Biology* (W. Dubitzky, O. Wolkenhauer, K.-H. Cho, and H. Yokota, eds.), pp. 551–555, Springer New York, 2013. DOI: 10.1007/978-1-4419-9863-7_611.
- [91] M. A. Friedl and C. E. Brodley, “Decision tree classification of land cover from remotely sensed data,” *Remote Sensing of Environment*, vol. 61, pp. 399–409, Sept. 1997.
- [92] J. R. Quinlan, “Induction of decision trees,” *Machine Learning*, vol. 1, pp. 81–106, Mar. 1986.
- [93] L. Rutkowski, M. Jaworski, L. Pietruczuk, and P. Duda, “The CART decision tree for mining data streams,” *Information Sciences*, vol. 266, pp. 1–15, May 2014.
- [94] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993.
- [95] J. Fürnkranz, “Decision Tree,” in *Encyclopedia of Machine Learning and Data Mining* (C. Sammut and G. I. Webb, eds.), pp. 330–335, Springer US, 2017. DOI: 10.1007/978-1-4899-7687-1_66.
- [96] S. L. Salzberg, “C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993,” *Machine Learning*, vol. 16, pp. 235–240, Sept. 1994.
- [97] S. Zhao, Y. Liu, J. Jiang, W. Cheng, M. Zhou, M. Li, and R. Ruan, “Extraction of mangrove in Hainan Dongzhai Harbor based on CART decision tree,” in *2014 22nd International Conference on Geoinformatics*, pp. 1–6, June 2014.
- [98] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, pp. 5–32, Oct. 2001.
- [99] “Complete ensemble empirical mode decomposition with adaptative noise (CEEMDAN).” http://bioingenieria.edu.ar/grupos/ldnlys/meteorres/re_inter.htm, 2014.
- [100] “ClassificationECOC class.” <https://www.mathworks.com/help/stats/classificationecoc-class.html>, 2014.
- [101] X. Wang, Y. Zheng, Z. Zhao, and J. Wang, “Bearing Fault Diagnosis Based on Statistical Locally Linear Embedding,” *Sensors*, vol. 15, pp. 16225–16247, July 2015.

- [102] U. Braga-Neto and E. Dougherty, “Bolstered error estimation,” *Pattern Recognition*, vol. 37, pp. 1267–1281, June 2004.
- [103] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*. The Morgan Kaufmann Series in Data Management Systems, Elsevier Science, 2011.