CARACTERIZACIÓN DE LAS SEÑALES DE DESBALANCEÓ Y DESALINEACIÓN EN MÁQUINAS ROTÁTIVAS POR MEDIO DE LA TRANSFORMADA CEPSTRUM PARA DIFERENCIAR LAS AMPLITUDES DE VIBRACIÓN EN EL BANCO DE VIBRACIONES DE LAS UNIDADES TECNOLÓGICAS DE SANTANDER.

PROYECTO DE INVESTIGACIÓN

AUTOR
JAIRO ANDRÉS ZANGUÑA RUIZ

UNIDADES TECNOLÓGICAS DE SANTANDER
FACULTAD DE CIENCIAS NATURALES E INGENIERIAS
INGENIERIA ELECTROMECANICA
BUCARAMANGA
FECHA DE PRESENTACIÓN: 01-Nov-2017

CARACTERIZACIÓN DE LAS SEÑALES DE DESBALANCEÓ Y DESALINEACIÓN EN MÁQUINAS ROTÁTIVAS POR MEDIO DE LA TRANSFORMADA CEPSTRUM PARA DIFERENCIAR LAS AMPLITUDES DE VIBRACIÓN EN EL BANCO DE VIBRACIONES DE LAS UNIDADES TECNOLÓGICAS DE SANTANDER.

PROYECTO DE INVESTIGACIÓN

AUTOR
JAIRO ANDRÉS ZANGUÑA RUIZ

Trabajo de Grado para optar al título de
Ingeniero Electromecánico

DIRECTOR
Camilo Leonardo Sandoval Rodríguez

GRUPO DE INVESTIGACIÓN EN SISTEMAS DE ENERGÍA, AUTOMATIZACIÓN Y CONTROL – GISEAC

UNIDADES TECNOLÓGICAS DE SANTANDER
FACULTAD DE CIENCIAS NATURALES E INGENIERIAS
INGENIERIA ELECTROMECANICA
BUCARAMANGA
FECHA DE PRESENTACIÓN: 01-Nov-2017

Nota de Aceptación

_____

_____

_____

_____

_____

Firma del jurado

_____

Firma del Jurado

DEDICATORIA

El presente proyecto quiero dedicárselo a mi Madre y Hermanos por su comprensión, acompañamiento, motivación y colaboración en general para poder dedicarle el tiempo necesario para la elaboración de este proyecto.

También quiero dedicarle la elaboración de este proyecto a mis Tías maternas, por su apoyo y comprensión, cuando no pude dicar el tiempo necesario a las actividades de reunión familiar.

AGRADECIMIENTOS

Agradezco al Magister en Ingeniería Electrónica Camilo Leonardo Sandoval Rodríguez docente de las Unidades Tecnológicas de Santander por sus aportes en la explicación de los diferentes factores que intervienen en la transformada cepstrum.

Agradezco al Ingeniero Mecánico Carlos Gerardo Cárdenas Arias docente de las Unidades Tecnológicas de Santander por sus aportes en la edición y contextualización del proyecto en el ámbito mecánico.

Agradezco al Ingeniero Mecatrónico Alexander Quintero Ruiz docente de las Unidades Tecnológicas de Santander por sus concejos y aportes sobre la programación en Matlab.

Agradezco al Ph.D en Ingeniería Mecánica Omar Lengerke Pérez rector de las Unidades Tecnológicas de Santander por el préstamo del banco de vibraciones sin el cual no sería posible la realización de este proyecto.

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

RESUMEN EJECUTIVO

El presente documento utilizó la transformada Cepstrum en máquinas rotátivas para diferenciar las amplitudes en las vibraciones, debidas al desbalance y la desalineación respecto a un grupo de referencia. Él desbalanceó es un fenómeno producido en la fabricación de piezas para máquinas rotátivas debido a la distribución desigual de masa y la desalineación es un fenómeno producido por la falla en la concentricidad entre el eje conductor y el eje conducido. Estos dos fenómenos contribuyen a la reducción de la vida útil de los elementos de apoyo, acoples y piezas en general de la máquina.

La realización del presente documento requiere de tres etapas generales. Inicia realizando labores de nivelación en las bases donde se soporta el banco de vibración, obteniendo los mejores registros, con el fin de establecer el grupo de control para las condiciones de balanceó y alineación. Él desbalanceó se realizó con una masa conocida ubicada en las dos distancias radiales que posee el primer y segundo volante. La desalineación se realizó corriendo los soportes deslizantes hacia atrás 0.5, 1.0 y 1.5 grados.

En la segunda etapa se creó el algoritmo en Matlab, el cual permitió analizar todas las muestras respecto a la transformada Cepstrum, la caracterización del cepstrum y la correlación de la distancia euclidiana. En la última etapa se analizaron los datos obtenidos identificando las diferencias que puedan existir en los registros analizados.

El proyecto se centró en la utilización de Matlab para encontrar diferencias entre los dos fenómenos mencionados y el grupo de referencia a una frecuencia de 30 Hz. Los resultados obtenidos permitieron determinar que es posible encontrar diferencias con la metodología planteada.

PALABRAS CLAVE. Desbalanceo, desalineación, vibraciones mecánicas, Transformada Cepstrum.

INTRODUCCIÓN

Las máquinas rotativas presentan distintos estados de funcionamiento durante su vida útil, los cuales pueden ser de tipo normal o de funcionamiento óptimo; de funcionamiento con fallas, las cuales pueden ser debidas al poco estudio de la máquina y de funcionamiento crítico, el cual puede afectar de forma permanente a diferentes piezas que componen la máquina (Sánchez, Pérez, Sancho y Rodríguez, 2007).

Debido a esto la industria comenzó a adoptar distintos tipos de mantenimiento en máquinas rotativas con el fin de abordar las fallas que se puedan producir en estos equipos; de forma preventiva antes de que se produzca la falla, de forma correctiva cuando se produce la falla y se indaga porque se produjo; y de falla cuando se busca reemplazar las piezas dañadas (Sánchez, et. al, 2007).

En este proyecto se abordan los análisis necesarios para prevenir que las fallas en máquinas rotativas puedan afectar de forma negativa al funcionamiento de la industria por medio del análisis de vibraciones (Torres y Batista, 2010)

En las máquinas rotativas se presentan dos afectaciones importantes estudiadas en este proyecto. La primera es debida al mal montaje de la máquina sea por la mala fabricación de la bancada o por el mal acople mecánico entre ejes conductor y conducido; esta afectación se le conoce como falla por alineación (Gómez de León, 1998).

La segunda afectación tiene que ver con el desgaste, incorrecta selección de un componente averiado, fabricación incorrecta de las piezas que componen la máquina, fractura de alguna pieza dinámica de la máquina, entre otras posibles causas; por las cuales se puede producir la falla por desbalanceo (Raó, 2012).

Aunque el tipo de falla pueda ser detectado a tiempo, surgen otros problemas inherentes a las fallas en las máquinas rotativas, las cuales tienen que ver con otros factores como; interpretación de gráficas, criterio y subjetividad del analizador.

La interpretación de gráficas en equipos analizadores de vibraciones o en software especializados como Labview, depende de la experiencia, del conocimiento de la gráfica de vibración y del criterio del analizador; para determinar si las fallas visibles y en ocasiones audibles son debidas a desbalance, desalineación, solturas mecánicas, rodamientos desgastados, flechas dobladas o problemas eléctricos (Torres y Batista, 2010).

Debido a lo anterior la subjetividad del analizador (el cual puede ser un experto en vibraciones, un director de mantenimiento, un técnico de mantenimiento, entre otros profesionales que puedan estar relacionados al estudio de máquinas rotativas por medio del análisis de vibraciones); puede intervenir como otra variable en la determinación del tipo de fallo que pueda presentar la máquina, por lo que surge la problemática que enmarca la posibilidad de encontrar un método que permita moderar la interpretación de gráficas de vibración, la subjetividad del analizador; y además, sirva de apoyo a los profesionales del

mantenimiento, para una correcta decisión en la determinación de fallas en máquinas rotativas.

La metodología aplicada en la solución de este proyecto se estructura de acuerdo a la investigación que es del tipo descriptiva porque se desean caracterizar fallas en el banco de vibraciones; y correlacional porque se establece un grupo de referencia para analizar las fallas de desbalanceo y desalineación en máquinas rotativas (Sampieri, Fernández-collado y Baptista, 2006).

El método empleado es del tipo deductivo porque se parten de teorías globales sobre la transformada cepstrum en la solución de problemas particulares como el planteado en este proyecto (Bernal, 2006); y la técnica utilizada para dar validez a todos los resultados planteados en este proyecto es del tipo experimental (Sampieri, et. al, 2006).

En la solución a la metodología planteada se utilizó el programa Matlab para analizar tres tipos generales de muestras; él grupo de control, él grupo de desbalanceo y él grupo de desalineación. En Matlab se crearon distintos algoritmos con el fin de poder seleccionar muestras correspondientes a una misma variable que se encuentren agrupadas en referencia a la amplitud en la frecuencia fundamental del banco de vibraciones. También se crearon algoritmos para calcular la transformada cepstrum, la distancia euclidiana y diferentes características en el análisis de 6 variables estadísticas.

Es conveniente el análisis de estas fallas, porque están ampliamente relacionados con la vida útil y el funcionamiento de las máquinas. Además, una correcta identificación de los problemas que puede presentar la máquina, permiten a la industria mayor productividad, eficiencia y un menor costo de mantenimiento. También, es importante su determinación con el fin de proporcionar mayores herramientas de aprendizaje a los futuros profesionales. La posibilidad de caracterizar estos fenómenos por medio de la transformada cepstrum permitirá realizar otros estudios, con el fin de profundizar en el análisis de vibraciones en las máquinas rotativas.

1. DESCRIPCIÓN DEL TRABAJO DE INVESTIGACIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

En la actualidad, las máquinas rotativas se pueden clasificar según el tipo de uso dado; convertir energía mecánica en energía eléctrica (generadores) y convertir energía eléctrica en energía mecánica (motores); los motores eléctricos son máquinas rotativas que funcionan con corriente continua y corriente alterna (Martin, 2012).

Él acople mecánico se produce entre dos ejes uno conductor y otro conducido. El mantenimiento de las máquinas rotativas acopladas se ubica principalmente en el alineamiento de los ejes y él balanceó de los elementos en movimiento que intervienen en el equipo (Sánchez, Pérez, Sancho y Rodríguez, 2007).

La alineación perfecta de los ejes no existe, aunque existen instrumentos como la regla calibrada metálica, los relojes comparadores y los dispositivos electrónicos de ajuste por láser (Manzano, 2014); los cuales, permiten ajustar las bancadas donde reposan las máquinas conductoras y conducidas (Harper, 2004). La desalineación es un problema generado por una mala alineación, se puede dividir según Gómez de León (1998) en desalineación paralela, angular y combinada.

Él balanceó de máquinas se realiza de acuerdo con la posición del elemento en movimiento, refiriéndose a si es estático o dinámico. Los elementos estáticos son considerados sistemas en un plano los cuales por lo general son discos, engranajes, volantes, entre otros. Los elementos dinámicos son considerados sistemas en dos planos, por lo general son rotores y ejes de las máquinas (Raó, 2012).

La recolección de datos referentes a las vibraciones en los equipos se realiza por medio de sensores que miden la aceleración, la corriente o por instrumentos termográficos (Oxford University Press, 1998).

En el análisis de vibraciones, intervienen muchos factores que no dependen de la vibración de la máquina, algunos de estos factores son: interpretación de gráficas, criterio y subjetividad del analizador. La interpretación de gráficas en equipos analizadores de vibraciones o en software especializados como Labview, depende de la experiencia, del conocimiento de la gráfica de vibración y del criterio del analizador; para determinar si las fallas visibles y en ocasiones audibles son debidas a desbalance, desalineación, solturas mecánicas, rodamientos desgastados, flechas dobladas o problemas eléctricos (Torres y Batista, 2010). Es aquí, donde la subjetividad del analizador, puede hacer la diferencia en la determinación del problema de la máquina, ocasionando muchas veces gastos innecesarios como: cambio de piezas en buen estado, tiempos de falla prolongados e inseguridad en la decisión del tipo de falla.

De acuerdo con lo anterior surge la problemática que enmarca la posibilidad de encontrar un método que permita moderar la interpretación de gráficas de vibración, la subjetividad

del analizador; y además, sirva de apoyo a los profesionales del mantenimiento, para una correcta decisión en la determinación de fallas en máquinas rotátivas, planteando la pregunta referente a ¿Cómo caracterizar señales de desbalanceo y desalineación en máquinas rotátivas por medio de la transformada Cepstrum para diferenciar las amplitudes de vibración respecto a un grupo de control?

En la solución a la problemática planteada se utilizará el programa Matlab para analizar tres tipos de muestras: él grupo de control, él grupo de desbalanceo y él grupo de desalineación. En Matlab se creará un algoritmo que permita llevar los datos del dominio del tiempo, al dominio de la Quefrency. Se espera que los datos en la Quefrency tiendan a una gráfica sinusoidal de la cual se tomarán los valores máximos y mínimos, como otras características que permitan encapsular el tipo de onda para las muestras tomadas. Con las características propias de cada grupo de muestras se creará un algoritmo de condiciones, que permita identificar qué tipo de falla se presenta en la máquina.

1.2. JUSTIFICACIÓN

La alineación y él balanceó en las máquinas eléctricas rotátivas tiene gran importancia para la industria, refiriéndose a la capacidad de las empresas de ser productivos y competitivos en la economía. La subjetividad del analizador para diferenciar el desbalanceo en elementos asociados a las máquinas y la desalineación de los acoples mecánicos, implica en la búsqueda de métodos que faciliten la identificación de estas fallas asociadas a las máquinas, mejorando la capacidad del profesional para decidir y la eficiencia de los departamentos de mantenimiento.

Es conveniente el análisis de estas fallas, porque están ampliamente relacionados con la vida útil y el funcionamiento de las máquinas. Además, una correcta identificación de los problemas que puede presentar la máquina, permiten a la industria mayor productividad, eficiencia y un menor costo de mantenimiento. También, es importante su determinación con el fin de proporcionar mayores herramientas de aprendizaje a los futuros profesionales. La posibilidad de caracterizar estos fenómenos por medio de la transformada cepstrum permitirá realizar otros estudios, con el fin de profundizar en el análisis de vibraciones en las máquinas rotátivas.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Caracterizar las señales de desbalanceo y desalineación en máquinas rotátivas por medio de la transformada Cepstrum para diferenciar las amplitudes de vibración en el banco de vibraciones de las Unidades Tecnológicas de Santander.

1.3.2. OBJETIVOS ESPECÍFICOS

1.3.2.1 Obtener los registros correspondientes a las señales de desbalanceo y desalineación en máquinas rotátivas por medio de la adquisición de datos disponible en el laboratorio de automatización de las Unidades Tecnológicas de Santander con el fin de aplicar la transformada cepstrum a los registros obtenidos.

1.3.2.2 Determinar las características asociadas a las dos condiciones de operación, mediante el desarrollo de un algoritmo en Matlab aplicando la transformada Cepstrum.

1.3.2.3 Establecer por medio de la distancia euclidiana las diferencias existentes entre los descriptores (características a partir del Cepstrum) que definen las condiciones de desbalanceo y desalineación en máquinas rotátivas.

1.4. ESTADO DEL ARTE / ANTECEDENTES

1.4.1. Desalineación y desbalanceó en máquinas rotátivas.

La alineación y él balanceó de una máquina, asegura su correcto funcionamiento y una vida prolongada llena de eficiencia y buen trabajo. Aunque, es una afirmación poco valedera, debido que en las máquinas rotátivas, se puede afirmar que el desgaste de las piezas que lo componen está relacionado en proporción a su funcionamiento. Es el desgaste de las piezas, el cual empieza a ocasionar fallas, que se replican de forma similar en todas las máquinas de acuerdo a las horas de funcionamiento. Según Olarte, Botero y Cañón (2010) la determinación de las causas por las cuales se producen fallas comunes en la máquina, es de gran importancia con el fin de estudiarlas, por medio de un control y monitoreo de las vibraciones.

Estas fallas comunes según Torres y Batista (2010); Aldaz (2015); Acosta, Molina y Cevallos (2013); y Moreno, Becerra y Rendón (2014) son: desbalanceo, desalineación, holguras mecánicas, flexión del eje y desgaste. Según Aldaz (2015) estos fenómenos se producen a una frecuencia síncrona múltiplo de la velocidad de giro del eje; las frecuencias de rodamientos, frecuencias de resonancia y cavitación se producen a una frecuencia asíncrona de la velocidad de rotación.

Lémoli (2015) realizó un artículo desde su experiencia, él analizó los problemas que producen en una máquina eléctrica, él desbalanceo y la desalineación. Él definió la desalineación como “la condición en que los ejes de la máquina conductora y conducida no

poseen la misma línea de centros". Esta definición es muy adecuada de acuerdo a Manzano y Harper.

Aldaz (2015) en su tesis de grado pretendió realizar un "trabajo investigativo de las vibraciones mecánicas por desalineación paralela y angular,… donde se puede llegar a determinar un mantenimiento pre-falla,…" (p.16); con el fin de evitar pérdidas económicas por el paro inesperado de máquinas.

Acosta, et al., (2013) pretendieron conocer los efectos producidos por la vibración de una máquina en funcionamiento, analizando la velocidad y después realizando un análisis espectral de la señal. Por medio del procedimiento realizado, lograron encontrar algunas frecuencias a las cuales surgen ciertos problemas en las máquinas, siendo estas: de 0 a 10 Hz holgura inadecuada y desplazamiento axial del rotor, de 0.4x a 0.5x inestabilidad dinámica en los cojinetes, en 1x defectos asociados al desbalance mecánico o hidráulico, de 2x a 5x solturas mecánicas, de 5x a 20x frecuencias de rodamientos, de 20x a 100x resonancias derivadas de vibración en máquinas vecinas; y mayores a 100 Hz excentricidad rotoestatorica.

La desalineación angular, Aldaz (2015) la definió como "Existe desalineación angular cuando las líneas centrales de dos ejes se cortan formando un ángulo… en una máquina se produce a 1x y 2x de la velocidad de giro" (p.52). Lémoli (2015) la estableció como un análisis de acuerdo al espectro característico de un sistema motor-bomba donde encontró que "la máquina presenta altas vibraciones axiales al 1x y 2x de la frecuencia de giro,…" (párr. 5).

La desalineación paralela, Aldaz (2015) la definió como "Existe desalineación paralela cuando las líneas centrales de dos ejes se encuentran desfasadas una distancia con respecto a la línea centroidal del eje de cada uno de los equipos acoplados… se produce a 1x, 2x y 3x de la velocidad de giro" (p.53). Por otra parte, por medio de un análisis del espectro de la máquina Lémoli (2015) encontró que "presenta altas vibraciones radiales en 1x y 2x de la frecuencia de giro,…" (párr.6). Aldaz (2015) explicó, que en bajas revoluciones la desalineación en general tiene su mayor amplitud a 1x, la cual corresponde con la frecuencia característica de un problema referente al desbalanceo. Cuando la máquina presente estos dos efectos o solo uno de ellos, se puede afirmar que se encuentra desalineado el eje conductor con respecto al eje conducido.

Respecto al desbalanceo, Lémoli (2015) encontró que balancear una máquina es "ajustar la distribución de masa para que este uniformemente distribuida sobre su centro geométrico,… Cabe resaltar que él desbalanceo disminuye en gran proporción la vida útil de la máquina,…" (párr.9). Por otro lado Flórez y Asiain (2011), definieron él balanceó como "Un sistema es equilibrado si durante su funcionamiento la resultante de todas las fuerzas y sus respectivos pares son de magnitud, dirección y sentido constantes" (p.74). También, Estupiñan, San Martin y Canales (2006); definieron él desbalanceo como "Es la condición donde el eje de inercia del rotor no coincide con su eje de rotación…" (p.147). También es necesario aclarar que el balanceó se le realiza

a elementos de la máquina que presentan movimiento en su funcionamiento normal, además, todos los elementos después de su fabricación son considerados como desbalanceados por distintos efectos presentes en la construcción de ellos. Según Benítez (2013), las causas que generan desbalanceo son; porosidad, no uniformidad de la densidad del material, tolerancias de ajustes en la máquina, desgaste, mantenimiento y limpieza, cambio de partes, acoples, excentricidad; entre otros.

Cuando él desbalanceo y la desalineación, no son diferenciables trabajando la máquina a bajas revoluciones Aldaz (2015) explicó, que es muy conveniente “realizar un análisis complementario de fase el cual permitirá distinguir mejor cual problema es el que se presenta a esta frecuencia” (p.52).

1.4.2. Transformada Cepstrum.

La transformada cepstrum según Oppenheim y Schafer (2004), surgió con Bogert en 1963, éste refiriéndose a una novedosa forma para analizar señales en el dominio del tiempo, la cual no pertenecía al dominio de la frecuencia; se puede definir el cepstrum como el espectro del logaritmo natural del espectro.

Agudelo (2008) realizó una metodología con el fin de caracterizar las señales sísmicas procedentes de dos volcanes del país de Colombia conocidos como Volcán nevado del Ruiz y Volcán Galeras; con el apoyo en la obtención de datos sismológicos del Observatorio Vulcanológico y Sismológico de las ciudades de Manizales y Pasto. Con el fin de obtener la caracterización de las señales sismológicas, Agudelo utilizó dos modelos paramétricos conocidos como Autorregresivos (AR) y Autorregresivos de Media Móvil (ARMA); y la transformada Cepstrum la cual refiere es “conocida como el espectro del espectro” (p.54).

Para la transformada Cepstrum, Agudelo (2008) utilizó dos formas conocidas como Cepstrum real y compleja. Donde refiere en sus resultados que la transformada Cepstrum en comparación con los modelos paramétricos utilizados le permitió obtener mejores resultados en la caracterización de señales sísmicas de largo periodo y de tipo vulcano-Tectónico.

Quiroga, Trujillo, y Quintero (2012) realizarón un artículo basado en el estudio de elementos de apoyo en máquinas rotátivas. El estudio fue basado en el análisis de vibración en rodamientos aplicando la transformada rápida de Fourier, envolvente y Cepstrum. En el artículo el Cepstrum se define como “Es un método utilizado en el estudio de vibraciones para determinar periodicidades en el análisis espectral de una señal separando el efecto de funciones de transferencia variables en el tiempo en una convolución de señales…” (párr. 20). El análisis de vibración les permitió reconocer las fallas en rodamientos en la pista exterior e interior en el dominio de la Quefrency midiendo la gamnitude de las ondas ocasionadas por fallas en las pistas.

Brien, Molisani y Burdisso (2013) con su artículo, pretendieron utilizar la transformada Cepstrum, con el fin de cuantificar y detectar la posición de una fuente sonora en la formación del mapa de ruido. Para ello han identificado los métodos más utilizados en

la obtención de fotografías acústicas como Beamforming, Robust Adaptive Beamforming, Delay and Sum, Multiple Signal Clasification y la transformada Cepstrum. La transformada Cepstrum, ellos la utilizaron en su expresión compleja, donde refieren que es la transformada inversa de Fourier del logaritmo complejo de la transformada de Fourier. Ellos refieren algunas características de utilizar la transformada cepstrum; permite la detección de patrones repetidos dentro de las señales espectrales, detección de la periodicidad y de espaciado entre frecuencias; y separación de familias de armónicas. También refieren algunos términos utilizados en el cepstrum, para frecuencia quefrency, espectro cepstrum, fase saphe, amplitud gamnitude, filtración liftering, armonico rahmonic y periodo repiod. Con la utilización del cepstrum ellos encontraron familias de armónicas que no son necesarias en el estudio, así como obtener la señal original del objeto entre toda la señal obtenida, permitiendo encontrar características apropiadas a la necesidad real del estudio.

Morales (2011) en su tesis de maestría utilizó la transformada cepstrum por medio de los coeficientes cepstrales de predicción lineal LPCC, como un complemento en la caracterización de señales de vibración en máquinas rotativas; siendo la finalidad de su proyecto, realizar un caracterizador de fallos utilizando programación en C. También Gajic y Paliwal (2001) realizarón un artículo donde buscaban probar distintos métodos en el reconocimiento de voz, donde utilizaron los LPCC y los coeficientes cepstrales en la frecuencia Mel MFCC; donde encontraron que en promedio un 75% de los datos pueden ser identificados en los 12 primeros coeficientes.

Gómez (2011) utilizó el MFCC como caracterizador de las señales producidas por el habla, con el fin de resolver problemas de reconocimiento del habla en el entorno de red; donde encontró que los MFCC permiten reducir los problemas de ancho de banda y reconocimiento de estas señales. También Peláez (2002) encontró que los MFCC poseen las mejores características para el reconocimiento de voz en sistemas de comunicación ligados a una red.

1.4.3. Estadísticos

Morales (2011) en su tesis de maestría utilizó distintos cálculos estadísticos para la caracterización de los fenómenos en máquinas rotativas, entre los que se encuentran el valor medio, la desviación estándar, la raíz media cuadrática, la Curtosis, la asimetría y el factor de forma; encontrando que los estadísticos permiten atribuir características a fenómenos en máquinas rotativas cuando se cumplen condiciones para una señal estacionaria. También Lei, He y Zi (2007) en su artículo utilizaron los cálculos estadísticos en la caracterización de la predicción de fallos de rodamientos en máquinas rotativas. Jack y Nandi (2002) en su artículo utilizaron los cálculos estadísticos como un método para el pre-procesado de las señales en máquinas rotativas.

2. MARCOS REFERENCIALES

2.1. MARCO TEÓRICO.

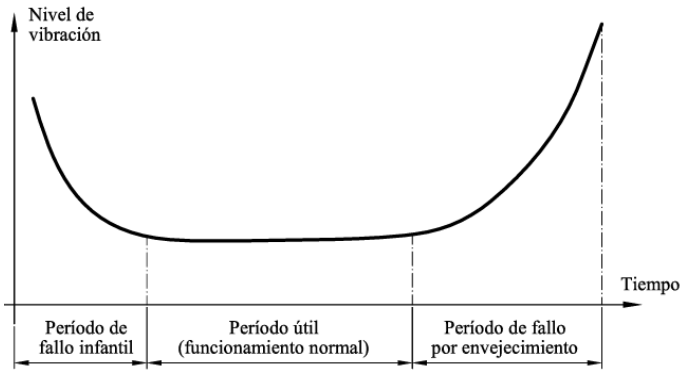
2.1.1. Vibración en máquinas.

La vibración u oscilación es un movimiento periódico en el intervalo del tiempo (Raó, 2012, p.13). La vibración en máquinas es producto de un fallo originado en muchas ocasiones, por el uso de la máquina y por la falta de mantenimiento en tiempos prolongados (Sánchez, et al., 2007, p.287).

La vibración en las máquinas sigue un ciclo denominado curva de bañera Figura 1, la cual describe tres etapas fundamentales. En la primera etapa, cuando la máquina es nueva o recién reparada la curva es decreciente debido a que todas las piezas de la maquina se ajustan al funcionamiento y no existen desgastes en elementos como rodamientos, engranajes, entre otros (Sánchez, et al., 2007, p.288).

Después de una corta transición, en la segunda etapa la máquina se estabiliza en una recta la cual sería descrita como el funcionamiento normal de la máquina o su vibración natural. Por último, aparece en la curva un ascenso repentino de la vibración, debido a desgastes excesivos en las piezas de la máquina (Sánchez, et al., 2007, p.288).

Figura 1. Vibración en máquinas, curva bañera.



Fuente: (Sánchez, et al., 2007, p.288).

La vibración en máquinas puede considerarse natural o normal debido al diseño, la bancada, las condiciones de operación, entre otros (Sánchez, et al., 2007, p.288). La vibración deja de ser normal y se vuelve crítica, cuando supera los límites establecidos por la norma ISO 10816 – 1995 respecto a la criticidad de la vibración en máquinas según su clasificación.

Posibles factores debidos al desgaste y envejecimiento que pueden influir en el deterioro de la máquina son visibles por medio de la detección de vibración. Estos factores son

desbalanceo, desalineación, fatiga, cambio de propiedades del material, corrosión, entre otros (Sánchez, et al., 2007, p.22).

Él desbalanceo produce altas vibraciones en la máquina y es debido al exceso de masa en una o más piezas de la máquina. Es visible en el funcionamiento de la máquina cuando esta vibra con una intensidad mayor que en condiciones normales. Puede ser observado por medio de sensores ubicados en los apoyos donde se visualiza con una frecuencia igual al giro del eje desbalanceado (Sánchez, et al., 2007, p.264).

2.1.2. Máquinas rotativas.

“Las máquinas rotativas son aquellas encargadas de convertir energía eléctrica en energía mecánica actuando como motores, o de convertir energía mecánica en energía eléctrica actuando como generadores…” (Martin, 2012, p. 27). Estas máquinas se encuentran a menudo en las centrales hidroeléctricas, o en campos de generación de energía eólica.

Los motores eléctricos se pueden clasificar según la energía con la cual son alimentados, corriente alterna (AC) o corriente directa (DC). Los motores de corriente alterna son conocidos también como motores de inducción y se pueden clasificar en síncronos y asíncronos (monofásicos, bifásicos y trifásicos) (Martin, 2012, p. 27).

2.1.3. Acople mecánico de máquinas rotativas.

Él acople mecánico es un elemento de máquina que sirve para unir dos ejes necesariamente separados por fabricación, con el fin de evitar las desventajas que trae consigo la fabricación de un solo eje; el cual, soporte todos los elementos de movimiento de la máquina, como se produce en máquinas eléctricas y líneas de trasmisión de ejes, motores de arranque, bancos de prueba, generadores, entre otros; resulta de gran importancia para unir ejes paralelos o ejes en distintos planos (Sánchez, Pérez, Sancho y Rodríguez, 2007).

La función principal del acople mecánico es la de trasmitir potencia mediante la unión de un eje impulsor a un eje impulsado, pero no es la única. Existen distintos tipos de acoples mecánicos, los rígidos y los flexibles; los cuales, cumplen funciones secundarias, como permitir el desalineamiento, pequeños desplazamientos axiales, fusible mecánico en sobrecargas, amortiguar vibraciones, entre otros (Sánchez, et al., 2007).

Los acoplamientos rígidos son muy utilizados en uniones que requieren gran precisión en la alineación de sus ejes, lo que genera gran confianza en cuanto a la desalineación y él desbalanceo que se pueda producir, debido a la vibración generada por un cimiento imperfecto. Existen cuatro tipos de acoplamientos básicos: de bridas, de manguito, de manguito partido y de eje intermedio. En tal caso, donde el eje acoplado por un sistema rígido presente algún tipo de falla en la unión de los ejes, los apoyos presentarán pérdida de su vida útil por la inducción de fatiga en sus materiales (Sánchez, et al., 2007).

Los acoples flexibles a diferencia de los rígidos, están diseñados con el fin de compensar él desbalanceo axial, radial y angular; debido a esta capacidad del acople, son los más utilizados en la industria, puesto que, ninguna unión se considera totalmente perfecta, ayudando a disminuir el riesgo de fatiga en los ejes. Existen distintos tipos de acoples flexibles entre los más usados son: de engranaje, de cadena, de rejilla de acero, de elementos metálicos y con elemento elastómero (Sánchez, et al., 2007).

2.1.4. Alineación

La alineación en los motores eléctricos es necesaria, para asegurar un correcto funcionamiento tanto de la máquina impulsora como del conjunto de elementos acoplados; cumple la función de unir los ejes de forma concéntrica y paralela principalmente, por medio de un elemento mecánico conocido como acople (Sánchez, et al., 2007, p.201).

Como se ha mencionado anteriormente la alineación perfecta no existe, debido a distintas causas asociadas a la construcción de las máquinas y condiciones de funcionamiento; aunque, existen distintos elementos utilizados para lograr una alineación con una precisión muy cercana a la ideal (Manzano, 2014).

Los elementos más comunes a utilizar son reglas calibradas y relojes comparadores. El primero consiste en una regla metálica lo más perfectamente construida que sirve como indicador a simple inspección de la linealidad entre los dos ejes acoplados. La regla se coloca en dos posiciones fundamentales vertical y horizontal si fuese un reloj visto de frente seria a las 12 y a las 9 (Harper, 2004).

Los relojes comparadores son elementos con precisión de 0.01 in o 0.05 mm según el fabricante; se utilizan colocando uno en cada eje a acoplar, uno en forma vertical y el otro en forma horizontal, con el fin de conseguir las dos posiciones de alineación necesaria, colinealidad entre los ejes y concentricidad entre sus diámetros (Manzano, 2014).

La utilización de los dos relojes es necesaria, con el fin de comparar sus valores y hacerlos lo más cercano posibles en medida; de esta forma, se hace rotar los ejes de forma manual hasta conseguir una posición invertida de los dos relojes, es decir, mirando un reloj de frente seria posición invertida de 12 las 6 y posición invertida de las 9 las 3 (Manzano, 2014). Existen en el mercado otro instrumento muy utilizado como lo es la alineación por láser, la cual facilita la calidad de la alineación por medio de un software, brindando mayor confiabilidad en el proceso como tal.

2.1.5. Desalineación.

La desalineación se presenta en la unión de dos ejes como ya se había mencionado, cabe recordar que la alineación perfecta es ideal por lo que surgen tres tipos comunes de desalineamiento: desalineación paralela, desalineación angular y desalineación combinada. La desalineación aunque existe en todas las máquinas acopladas es importante procurar disminuir sus amplitudes de desalineación, en capítulos anteriores se mencionaron

algunos instrumentos que permiten conseguir una alineación casi perfecta, por lo que, permiten reducir a valores seguros la posibilidad de fatiga en los ejes acoplados.

La desalineación paralela entre dos ejes acoplados mecánicamente existe, aún si, las máquinas están alineadas en forma angular; se produce cuando el plano horizontal de los ejes no coincide, debido a que la máquina impulsora es más alta en sus cimientos o más baja que el eje del conjunto de elementos impulsados, esto da como resultado; si inspeccionamos una máquina con una regla metálica en el plano superior horizontal, será evidente que existe una luz entre los ejes de las dos máquinas (Sánchez, et al., 2007). Aun si la regla metálica coincidiera con las dos superficies, se considera que exista alineación paralela parcial, debido a que observando con relojes comparadores aun existiría una desalineación de los dos ejes, debido a esto la alineación paralela por sí sola no asegura una correcta alineación del eje.

La desalineación angular entre dos ejes acoplados mecánicamente se produce, cuando el plano vertical u horizontal o ambos no coincide, debido a que alguno de los ejes a acoplar está inclinado hacia algún extremo; para realizar una alineación inicial, se puede utilizar una regla metálica, con la cual, se logre coincidencia entre el extremo superior, inferior, derecho e izquierdo (Gómez de león, 1998, p.202). Aun, cuando se logre una coincidencia entre los extremos de los ejes, es muy posible que exista desalineación angular. Para asegurar que exista una alineación en el eje vertical, se puede colocar relojes comparadores sobre los ejes, con el fin de medir los extremos y lograr que coincidan en valor numérico.

La desalineación combinada es la forma comúnmente encontrada, en el acoplamiento de ejes mecánicos, es la unión de la desalineación paralela y angular en un par de ejes, siendo una situación más real; debido a que es poco probable que los ejes en un principio, solo presenten algún tipo de desalineamiento (Sánchez, et al., 2007, p.204). La desalineación combinada se considera que existe en cualquier par de ejes acoplados mecánicamente, así, cuando se vaya a realizar la alineación de los ejes se puede proceder de la siguiente manera; primero, realizar una alineación con una regla metálica colocándola en los cuatro puntos esenciales de alineación; según un reloj análogo, las 12, 3, 6 y 9; segundo, utilizar relojes comparadores con el fin de evidenciar los valores pertenecientes a estos cuatro puntos y lograr por medio de la alineación una coincidencia numérica de estos puntos.

2.1.6. Balanceó

Él balanceó de máquinas eléctricas acopladas a volantes, ventiladores, engranajes, entre otras máquinas, se hace necesario; debido a la fabricación misma de los elementos mencionados anteriormente, ya que, en los procesos industriales existe la posibilidad de que las propiedades químicas de los materiales, se sitúen en mayor proporción a un lado de los elementos, siendo desbalanceados por construcción (Raó, 2012).

Él balanceó existe en dos formas: estático y dinámico. Él balanceó estático también es conocido como balanceó en un plano; él balanceó dinámico también es conocido como balanceó en dos planos. Cabe resaltar que él balanceó de cualquier elemento en

movimiento se realiza adicionando o retirando masa del objeto lo cual permite que este encuentre el equilibrio estático que requiere (Raó, 2012).

Él balanceó estático se puede determinar en cualquier elemento en movimiento de la siguiente forma. Existen dos situaciones, si la máquina esta acoplada o si la máquina está libre de acoplamiento.

En el primer caso, vigilar los elementos pertenecientes al segundo eje, supongamos que este eje posee solo un volante. Con el motor apagado hacer girar el volante en dirección de las manecillas del reloj y esperar a que este frene de forma libre, cuando se detenga el eje, realizar una marca en el plano horizontal con un lápiz, el más utilizado en metales es el rojo; repetir este proceso al menos unas 10 veces. Si las marcas realizadas en las 10 repeticiones coinciden en puntos muy cercanos a la primera marca, se dice que el volante se encuentra desbalanceado. Si las marcas realizadas aparecen en forma arbitraria en toda la circunferencia del volante, se dice que está balanceado (Raó, 2012).

En el segundo caso, colocar un volante en el eje de la máquina que previamente fue balanceado. Con la máquina apagada hacer girar el volante en dirección de las manecillas del reloj, dejando que este se detenga libremente. Cuando se detenga marcar con él lápiz rojo y repetir 10 veces el mismo procedimiento. Si las marcas coinciden o se encuentran muy cercanas a la primera marca, el eje del motor se encuentra desbalanceado. Si las marcas aparecen de forma aleatoria en la circunferencia del volante, se dice que el eje del motor está balanceado (Raó, 2012).

Él balanceó dinámico se presenta en los ejes principalmente por tener una longitud muy larga considerada con los elementos como volantes, engranajes y ventiladores. Él desbalanceo en movimiento teóricamente debería ser eliminado por la velocidad de rotación (w); pero, cuando el desbalance es muy grande, él motor en el arranque e incluso en sus primeros segundos de rotación, vibra de forma descontrolada; aunque, esta vibración puede ser atribuida a un desajuste en los cimientos o a los rodamientos involucrados; si al hacer una inspección minuciosa se encuentra que estas variables no intervienen en el proceso, se puede considerar que el eje de rotación se encuentra desbalanceado (Raó, 2012).

Determinar el punto exacto para balancear un eje desbalanceado dinámicamente es muy complejo de realizar, por lo que, en muchos casos la corrección debe ser reemplazada por dos masas en sus extremos. El procedimiento para balancear el eje se debe realizar de la siguiente forma. Desmontar todos los elementos involucrados en el eje y colocarlo solo con los rodamientos, después hacer girar el eje de forma manual fijándose en uno de los extremos, realizando la marcación con lápiz rojo correspondiente, realizar 10 repeticiones sería necesario para determinar el punto de desbalanceo, después realizar el mismo procedimiento pero fijándose en el otro extremo del eje (Raó, 2012).

Al revisar la linealidad de las dos marcas con respecto al plano horizontal, veremos que una marca aparece adelantada unos grados o retrasada unos grados respecto a la otra; la falla en la linealidad se debe a que desconocemos el punto exacto del desbalanceo, así de esta forma, podríamos balancear el eje colocando dos masas en cada extremo; cuanto más

alejado este la línea de marcación 1 de 2, más grande será la diferencia en las masas; solo a partir de la experimentación se podrá determinar cuáles son las masas que permitirán darle el balance al eje en general (Raó, 2012).

2.1.7. Acelerómetro.

El acelerómetro es un instrumento de medida que permite registrar la aceleración (lineal o trasversal) de un objeto en movimiento con respecto al tiempo. Su funcionamiento es basado en la Ley de Hooke donde la fuerza de inercia producida por la aceleración es medida por medio de un resorte (Oxford University Press, 1998).

El acelerómetro piezoeléctrico está basado en el mismo funcionamiento descrito anteriormente, sirve para medir vibraciones con un amplio intervalo de frecuencias. Su principal uso se puede encontrar en observatorios sismológicos, con el fin de registrar los movimientos sísmicos y en la industria; para registrar las vibraciones en las máquinas rotátivas (Oxford University Press, 1998).

2.1.8. Transformada de Fourier.

La transformada de Fourier nombrada así en honor al matemático y físico francés Jean Baptiste Joseph Fourier. Es una expresión matemática que relaciona las series de Fourier por medio de una integral (Zill, Cullen, 2008). Normalmente, utilizada en señales, se utiliza para llevar funciones periódicas en el dominio del tiempo $n(t)$ al dominio de la frecuencia $n(w)$ (James, et al., 2002).

La integral de Fourier en el intervalo de $(-\infty, \infty)$ está dada por Ecu (1) (Zill, Cullen, 2008, p.375).

$$f(x) = \frac{1}{\pi}\int_0^{\infty}\left[\left(\int_{-\infty}^{\infty} f(x)cos \propto dx\right)cos \propto x + \left(\int_{-\infty}^{\infty} f(x)sen \propto dx\right)sen \propto x\right]d \propto \quad (1)$$

$$a(\propto) = \int_{-\infty}^{\infty} f(x)cos \propto dx \quad (2)$$

$$b(\propto) = \int_{-\infty}^{\infty} f(x)sen \propto dx \quad (3)$$

Las Ecu (2) y (3); nos permiten sustituir las integrales definidas en el intervalo $(-\infty, \infty)$ por unas constantes que nos permiten ver de manera general, la forma fundamental de la serie de Fourier. A partir de la Ecu (1) se puede expresar por medio de constantes en (4) (Zill, Cullen, 2008, p.376).

$$f(x) = \frac{1}{\pi}\int_0^{\infty}[a(\propto)cos \propto x + b(\propto)sen \propto x]d \propto \quad (4)$$

La convergencia de la integral de Fourier se puede establecer, analizando la Ecu (4) como $f(x) = f(x+) + f(x-)$ con los límites del intervalo $(-\infty, \infty)$, cuando tiende a "0" por la izquierda y por la derecha. De esta forma, la integral de Fourier converge al promedio de las dos funciones como se aprecia en (5) (Zill, Cullen, 2008, p.376).

$$convergencia\ de\ f(x) = \frac{f(x+) + f(x-)}{2} \tag{5}$$

Los coeficientes $a(\propto)$ y $b(\propto)$ representan las funciones coseno y seno respectivamente, está relacionado con $f(x)$, la cual es una función par e impar. Por lo que $a(\propto)$ representa a las funciones pares y $b(\propto)$ representa a las funciones impares (Zill, Cullen, 2008, p.377).

La integral compleja de Fourier está dada por las ecuaciones (6) y (7) donde se analiza funciones con parte real y parte imaginaria.

$$f(x) = \frac{1}{2\pi}\int_{-\infty}^{\infty} c(\propto)^{-i\propto x} d \propto \tag{6}$$

donde

$$c(\propto) = \int_{-\infty}^{\infty} f(x)^{i\propto x} dx \tag{7}$$

La transformada de Fourier tiene un homologo que es la transformada inversa de Fourier lo que permite llevar una función del dominio del tiempo al dominio de la frecuencia y viceversa.

2.1.9. Propiedades de la transformada de Fourier.

Linealidad.

La propiedad de linealidad de la transformada de Fourier enuncia que: si las funciones $n(t)$ y $m(t)$ son periódicas y tienen transformada de Fourier por si solas $N(w)$ y $M(w)$, la suma de las funciones $f(t) = n(t) + m(t)$, también es periódica, por lo que posee transformada de Fourier $F(w) = N(w) + M(w)$ (James, et al., 2002, p.375).

Desplazamiento en tiempo.

Sea una función $n(t)$ periódica, con transformada de Fourier $N(w)$, desplazada en el tiempo $n(t - t_0)$ entonces la transformada de Fourier será desplazada en la frecuencia como se aprecia en (8) (James, et al., 2002, p.377).

$$n(t - t_0) = e^{-jwt_0} N(e^{jw}) \tag{8}$$

Desplazamiento en la frecuencia.

Sea una función $n(t)$ periódica, con transformada de Fourier $N(w)$, multiplicada por $e^{jw_0 t}$ da como resultado una transformada de Fourier corrida en la frecuencia como se aprecia en (9) (James, et al., 2002, p.378).

$$e^{jw_0 t} n(t) = N(w - w_0) \tag{9}$$

Inversión temporal.
Sea $n(t)$ una función con transformada de Fourier $N(w)$, entonces si se invierte la secuencia temporalmente a la función $n(-t)$ la transformada de Fourier seria $N(-w)$ (Oppenheim y Schafer, 2011, p.59)

Convolución $(*)$*.*
Sea $n(t)$ y $h(t)$ dos funciones continuas en el dominio del tiempo con transformadas de Fourier $N(w)$ y $H(w)$ respectivamente. El teorema de convolución indica que si las funciones son múltiplos en el tiempo en la frecuencia también serán múltiplos, como se aprecia en (10) (Oppenheim y Schafer, 2011, p.60).

$$sea; \qquad \begin{matrix} m(t) = n(t)h(t) = n(t) * h(t) \\ M(w) = N(w)H(w) \end{matrix} \tag{10}$$

2.1.10. Transformada Cepstrum.

El cepstrum surgió con Bogert, Healy y Turkey (1963) cuando publicaron un artículo llamado: "The Quefrency Analysis of time series for Echoes: Cepstrum, Pseudoautocovariance, cross-cepstrum, and saphe cracking" (Oppenheim y Schafer, 2004, p.95); traducido seria "El análisis (cufrencial) de series temporales para ecos: cepstrum, pseudoautovarianza, cepstrum cruzado y ruptura de forma". Encontraron una forma de analizar las señales que no pertenecía ni al dominio de la frecuencia, ni del tiempo, por lo que definieron nuevos términos para el análisis de las señales. Frecuencia "Quefrency", espectro "cepstrum", fase "saphe", amplitud "gamnitude", filtración "liftering", armonico "rahmonic" y periodo "repiod" (Brien, et al., p.8).

El cepstrum posee varias definiciones entre las que se destacan; "Se llama cepstrum…, a una transformada del espectro de Fourier, más o menos modificado, de una señal" (Nava, 2013, p.301). "El cepstrum de una señal es la transformada de Fourier del logaritmo…, de la señal estudiada, razón por la cual en ocasiones se le conoce como el espectro del espectro" (Agudelo, 2008, p. 54).

Considerando una señal muestreada $x(t)$ la cual es la suma de dos señales $a(t)\ y\ b(t)$. La señal $b(t)$ es una copia de $a(t)$ desplazada en el tiempo (t_0) y escalada (β) como se aprecia en (11) (Oppenheim y Schafer, 2011, p. 954).

$$x(t) = a(t) + b(t) = a(t) + \beta a(t - t_0) \tag{11}$$

Aplicando a $x(t)$ la propiedad de convolución, para llevar al dominio de la frecuencia como se aprecia en (12).

$$X(w) = A(w) + \beta A(w - t_0) \tag{12}$$

Llevando $x(t)$ a la forma de la transformada de Fourier en tiempo discreto como se aprecia en (13).

$$X\left(e^{j\omega}\right) = A\left(e^{j\omega}\right)\left[1 + \beta e^{-j\omega t_0}\right] \quad (13)$$

El módulo de $X\left(e^{j\omega}\right)$ se aprecia en (14).

$$X\left(e^{j\omega}\right) = A\left(e^{j\omega}\right)[1 + \beta^2 + 2\beta cos(wt_0)]^{1/2} \quad (14)$$

El cepstrum real es el log de $X\left(e^{j\omega}\right)$. Finalmente la expresión queda como se aprecia en (15).

$$\log\left|X\left(e^{j\omega}\right)\right| = \log\left|A\left(e^{j\omega}\right)\right| + \frac{1}{2}\log|1 + \beta^2 + 2\beta cos(wt_0)| \quad (15)$$

Aunque, las expresiones anteriores forman parte del cálculo del cepstrum para una señal de tipo infinita, es conveniente determinar una ecuación que permita encontrar el cepstrum para una señal finita; facilitando de esta forma el computo del cepstrum (Oppenheim y Schafer, 2011).

El cepstrum para una señal finita es posible calcularlo por medio de la transformada discreta de Fourier. En la ecuación (16) se puede ver como se debe aplicar la transformada discreta de Fourier para el cálculo del cepstrum (Faúndez, 2000).

$$X[n] = IFFT\{log|FFT(x[n])|\} \quad (16)$$

Para obtener el cepstrum real $X[n]$ de una señal finita en el tiempo $x[n]$ se le debe aplicar la transformada discreta de Fourier *FFT;* al valor absoluto de la *FFT* se le debe aplicar el logaritmo en base 10 *log* y por ultimo aplicar la transformada discreta de Fourier inversa *IFFT* (Faúndez, 2000).

Los coeficientes cepstrum en la escala Mel (CCM) han sido utilizados para el reconocimiento de voz con muy buenos resultados respecto a su caracterización; utilizando filtros triangulares pasa banda de gran solapamiento. Para poder determinar los CCM se debe modificar el *log* por el logaritmo en la escala Mel (Salcedo, 2011).

El logaritmo en la escala Mel es una aproximación de la escala perceptual humana donde se modifica la frecuencia original de la señal a una frecuencia de percepción auditiva. En la ecuación (17) se puede ver la definición de la escala Mel $Mel[f]$ donde f es la señal en el dominio de la frecuencia (Salcedo, 2011).

$$Mel[f] = 2595 * \log\left(1 + \frac{f}{700}\right) \quad (17)$$

Para calcular los CCM se debe modificar el logaritmo en base 10 de la ecuación (16) por el logaritmo en la escala Mel ecuación (17) dando como origen la ecuación (18) donde la f de

(17) es remplazada por el valor absoluto de la transformada discreta de Fourier de la señal en el tiempo.

$$CCM[n] = IFFT\left\{2595 * \log\left(1 + \frac{|FFT(x[n])|}{700}\right)\right\} \quad (18)$$

2.1.11. Distancia Euclidiana.

La distancia euclidiana es una medida de distancia métrica, de su cercanía o alejamiento, entre dos o más variables en el espacio euclidiano. También se puede definir como la diferencia entre los valores de las variables que componen un grupo de muestras estudiadas (Herrera, 2000). Para calcular la distancia euclidiana se puede utilizar el teorema de Pitágoras. La distancia euclidiana entre dos muestras es $|y_{i1} - y_{i2}|$, y_{i1} es el valor de la muestra uno y y_{i2} es el valor de la segunda muestra (Rodríguez, Álvarez, Bravo, 2001).

Para un número infinito de muestras se utiliza la ecuación dada en (19) (Rodríguez, Álvarez, Bravo, 2001).

$$M_1(x_1, x_2) = \sqrt{\sum_{i=1}^{n} (y_{i1} - y_{i2})^2} \quad (19)$$

Donde x_1 es la muestra 1, x_2 es la muestra 2, y_{i1} es el valor de la variable de la muestra 1, y_{i2} es el valor de la muestra 2 y n es el número de muestras.

La ecuación que permite relacionar dos muestras está dada por la ecuación (20) (Rodríguez, Álvarez, Bravo, 2001).

$$M_1(x_1, x_2) = \sqrt{(y_{11} - y_{12})^2 + (y_{21} - y_{22})^2} \quad (20)$$

La distancia euclidiana aumenta numéricamente de acuerdo con el número de variables involucradas. Para obtener mejores resultados, los datos de las variables deben tener el mismo número de cifras significativas (Rodríguez, Álvarez, Bravo, 2001).

Suponiendo que las variables A y B pertenecen al grupo de la muestra M donde se requiere encontrar la distancia entre estas dos características. Ahora y_{11} y y_{21} son datos de la variable A y y_{12} y y_{22} son datos de la variable B, para encontrar la distancia euclidiana se debe realizar la diferencia al cuadrado de cada uno de los datos, sumar el

resultado de las diferencias y obtener la raíz de la suma de las diferencias, dando como resultado la distancia de B con respecto de A (Herrera, 2000).

2.1.12. Tamaño de la muestra.

El tamaño de la muestra es una representación significativa de la población de estudio. Para determinar su medida dentro de cualquier proyecto se debe revisar si existe un precedente en la selección de muestra; además, se debe establecer si la población de estudio es finita o infinita (Martínez, 2012).

El tamaño de muestra para poblaciones infinitas ecuación (21), se denota con la letra *n*; *Z* es la confianza de la muestra, *P* es la probabilidad de éxito, *Q* es la probabilidad de fracaso y *E* es la precisión o error máximo admisible del muestreo (Martínez, 2012, p.303).

$$n = \frac{Z^2 PQ}{E^2} \tag{21}$$

La confianza de la muestra por lo general se debe manejar en un rango del 90% al 100%; siendo la más utilizada para muestreo en poblaciones infinitas una confianza del 95% $Z = 1.6448$ (Martínez, 2012).

En poblaciones infinitas sin precedente se debe tomar la probabilidad de éxito y de fracaso iguales con un valor de $P = Q = 0.5$; él error máximo admisible se recomienda no supere el 5%, aunque, para poblaciones sin precedente no se especifica un límite para el error máximo. Cuando el tamaño de la muestra no sea un valor entero siempre se debe aproximar al siguiente valor entero más cercano (Martínez, 2012).

2.1.13. Estadísticos.

El valor medio o media aritmética es el valor que representa la mitad entre dos o más números en relación sobre la cantidad de números sumados. Su valor es afectado por los denominados valores atípicos extremos sean mínimos o máximos (Martínez, 2012).

El valor medio en un conjunto de datos dentro de un rango razonable representa una tendencia de los valores a un valor representativo para todos. En la ecuación (22) se define el valor medio $T1$ de una muestra $x(n)$ como la suma de las amplitudes de la señal entre la longitud de la señal N (Martínez, 2012).

$$T1 = \frac{\sum_{n=1}^{N} x(n)}{N} \tag{22}$$

La desviación estándar es la raíz de la sumatoria de la diferencia cuadrada de los valores de la señal con su valor medio entre el total de datos menos 1. Su medida indica que tan

dispersos se encuentran los datos del valor medio siendo cero un valor ideal donde los datos de la señal y su media coinciden en valor (Martínez, 2012).

En la ecuación (23) se expresa la desviación estándar para una señal donde $x(n)$ es el dato enésimo de la señal, $T1$ es el valor medio y N es la cantidad total de datos (Martínez, 2012).

$$T2 = \sqrt{\frac{\sum_{n=1}^{N}(x(n) - T1)^2}{N - 1}} \tag{23}$$

Para poder relacionar la distancia euclidiana se utiliza el coeficiente de variación el cual se define como la desviación estándar sobre el valor medio de la señal ecuación (24) (Martínez, 2012).

$$CV = \frac{T2}{T1} \tag{24}$$

La raíz media cuadrática o RMS es la raíz de la sumatoria del cuadrado de un número entre la cantidad total de datos. Al depender en gran medida del valor medio de los datos el RMS también es susceptible a los valores atípicos extremos (Martínez, 2012).

En la ecuación (25) para una señal la RMS $T3$ es la raíz de la sumatoria de cada uno de los datos de la señal $x(n)$ al cuadrado, entre la cantidad de datos N (Martínez, 2012).

$$T3 = \sqrt{\frac{\sum_{n=1}^{N}(x(n))^2}{N}} \tag{25}$$

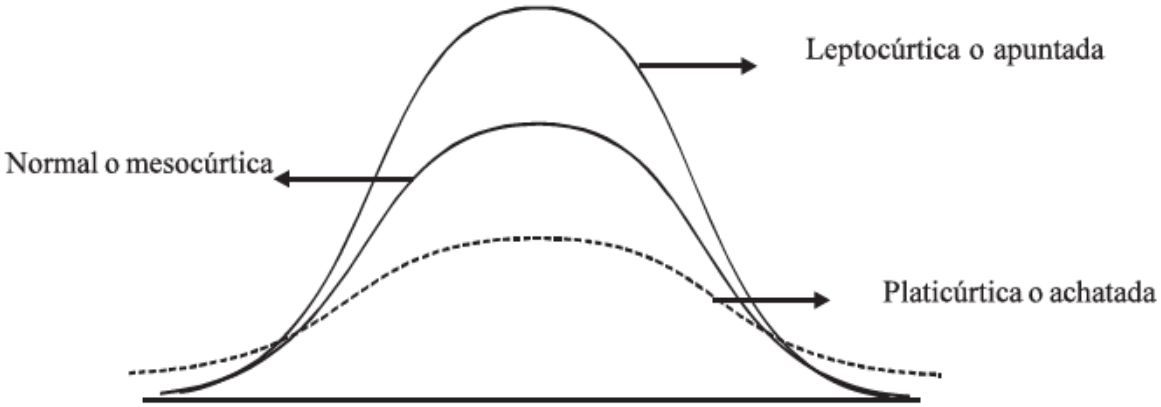
La Curtosis es un método matemático de la medida de dispersión en los datos; en la cual se analiza la forma de la curva, siendo esta muy similar a la de una campana Figura 2. La Curtosis también nos ayuda a prever sin ver el grafico como van a estar distribuidos los datos; muy agrupados hacia el eje de simetría, más o menos agrupados o poco agrupados (Martínez, 2012).

En la ecuación (26) la Curtosis $T4$ depende de la sumatoria de la diferencia de orden cuarto de la señal $x[n]$ y el valor medio $T1$ entre la longitud total de la señal N menos 1 por la desviación estándar $T2$ elevada al cuarto orden (Martínez, 2012).

$$T4 = \frac{\sum_{n=1}^{N}(x(n) - T1)^4}{(N - 1) * (T2)^4} \tag{26}$$

La Curtosis tiene tres tipos de curva diferenciadas Figura 2, puede ser mesocúrtica o curva de referencia cuando $T4 = 0$; puede ser leptocúrtica o apuntada cuando la Curtosis $T4 > 0$ y puede ser platicúrtica o achatada cuando la Curtosis $T4 < 0$ (Martínez, 2012).

Figura 2. Tipos de curvas en la Curtosis.



Fuente: (Martínez, 2012, p. 171).

La simetría ampliamente utilizada en la geometría de elementos circulares, cuadrados o elementos de forma irregular, es una característica dada a las curvas de un conjunto de datos cuando el sesgo es nulo, también existe sesgo positivo y negativo dando lugar a una curva asimétrica (Lind, Marchal, y Wathen, 2012).

La asimetría indica valores extremos dentro de los datos los cuales pueden ser positivos o negativos. Cuando la asimetría es nula su valor numérico es cero, indicando que los datos son simétricos. Cuando la asimetría es positiva indica que los datos están inclinados en mayor medida hacia su extremo máximo. Cuando la asimetría es negativa indica que los datos están inclinados en mayor medida hacia su extremo mínimo (Lind, Marchal, y Wathen, 2012).

En la ecuación (27) la asimetría depende de la sumatoria de la diferencia cubica de la señal $x[n]$ y el valor medio $T1$ entre la longitud total de la señal N menos 1 por la desviación estándar $T2$ elevada al cubo (Lind, Marchal, y Wathen, 2012).

$$T5 = \frac{\sum_{n=1}^{N}(x(n) - T1)^3}{(N-1) * (T2)^3} \tag{27}$$

El factor de forma (28) es un medidor que refleja si los datos evaluados se encuentran dentro de una misma forma, para esto se apoya en el valor RMS y el valor medio de los datos. Cuando el factor de forma es igual a la unidad significa que los datos evaluados tienen la misma forma; un factor de forma muy cercano a 1 que es lo común significa que los datos están desviados un poco del valor RMS, pero son muy similares (Hermosa, 1999).

$$T6 = \frac{T3}{T1} \tag{28}$$

2.1.14. Estudio descriptivo.

El estudio descriptivo busca dar a conocer las propiedades, características y perfiles de objetos, personas, fenómenos, entre otros; que se puedan someter a un análisis sin examinar o analizar los datos en sí mismo. También, mide o recoge información de manera independiente o conjunta de distintas variables pertenecientes al elemento a investigar sin intentar relacionarlas (Sampieri, Fernández-collado y Baptista, 2006, p. 102).

Además, el estudio descriptivo permite al investigador definir y visualizar lo que se va a estudiar (conceptos, variables, componentes, entre otros) y lo que se va a medir (personas, grupos, animales, objetos, entre otros) (Sampieri, et al., 2006, p. 102).

2.1.15. Estudio correlacional.

La correlación de variables o estudio correlacional busca encontrar de manera cuantitativa o cualitativa, la relación entre dos o más conceptos, categorías o variables en un contexto en particular. La relación de variables depende del estudio a realizar. La relación mínima se da entre dos variables, pero también, se puede relacionar tres variables o múltiples variables (Sampieri, et al., 2006, p. 105).

Para que exista correlación, está siempre se debe dar entre características del mismo sujeto de estudio. La relación entre las variables estudiadas siempre debe ponerse a prueba debido a que es posible que las hipótesis planteadas puedan no tener relación (Sampieri, et al., 2006, p. 105).

La correlación es útil a medida que permite determinar cómo está relacionada una variable respecto a otras con las cuales presuntamente está relacionada. La correlación puede ser positiva o negativa; si es positiva, significa que las variables están directamente relacionadas es decir si una variable toma un valor positivo con cierta amplitud su homologo tenderá a tomar un valor positivo con una amplitud muy cercana. Si es negativa, significa que las variables son opuestas o están inversamente relacionadas (Sampieri, et al., 2006, p. 105).

2.1.16. Método deductivo.

El método deductivo permite obtener conclusiones parciales partiendo de teorías generales, las cuales se consignan en el marco teórico a una situación particular la cual es el proyecto estudiado, dando como resultado, él desarrollo de los objetivos previstos y las conclusiones que validan o niegan, él estudio realizado (Bernal, 2006, 56).

El método deductivo permite al investigador inferir una solución a un cuestionamiento planteado partiendo de sucesos similares que pueden ser relacionados para llegar a una conclusión particular, cabe aclarar que la conclusión obtenida debe ser verificada por medio de una técnica, la cual en sí, permite establecer la certeza de la solución planteada (Bernal, 2006).

2.1.17. Técnica experimental.

La técnica experimental se refiera a la manipulación de una o más variables independientes con el fin de ver qué cambios produce en una o más variables dependientes en un entorno de control para el investigador (Sampieri, et al., 2006, p. 160).

(Sampieri, et al., 2006) Los requisitos para poder realizar una investigación enfocada a la experimentación son: Primero, la manipulación de la variable independiente una o más, así mismo, que la variable independiente tenga uno o varios grados de libertad, es decir; que al modificar la variable independiente se observen cambios pertinentes en la variable dependiente.

La variable dependiente se debe medir solamente, es decir, no se le realiza ningún efecto de manipulación. Segundo, debe existir relación entre la variable independiente y dependiente debido a que una falsa relación o nula causara una invalidez de la información encontrada. Por último, debe existir control en los grupos seleccionados para el experimento, esto se logra teniendo un grupo denominado de control al cual se le realizan los mismos procedimientos, pero la variable independiente y sus grados de libertad son nulos para este grupo.

2.2. MARCO NORMATIVO.

2.2.1. Norma ISO 10815 – 1995

La norma ISO 10816 – 1995 establece parámetros para identificar situaciones donde las máquinas rotátivas pueden fallar por la severidad de la vibración. Entre los parámetros que se establecen, se encuentra el tipo de máquina clasificado por la potencia y el tamaño del eje, el tipo de soporte en el cual está montada: rígido o flexible; revoluciones o velocidad y zonas críticas de funcionamiento basado en un gráfico de severidad. A continuación se detalla la norma según la fuente encontrada Sinais Ingeniería.

Sinais Ingeniería (2013), establece las condiciones y procedimientos generales para la medición y evaluación de la vibración, utilizando mediciones realizadas sobre partes no rotátivas de las máquinas. El criterio general de evaluación se basa tanto en la monitorización operacional como en pruebas de validación que han sido establecidas fundamentalmente con objeto de garantizar un funcionamiento fiable de la máquina a largo plazo.

La severidad de la vibración se clasifica conforme a los siguientes parámetros:

- Tipo de máquina.
- Potencia o altura de eje.
- Flexibilidad del soporte.

Clasificación de acuerdo al tipo de máquina, potencia o altura de eje.
Sinais Ingeniería (2013), las significativas diferencias en el diseño, tipos de descanso y estructuras soporte de la máquina, requieren una división en grupos. Las máquinas de estos

grupos pueden tener eje horizontal, vertical o inclinado y además pueden estar montados en soportes rígidos o flexibles.

- **Grupo 1**: Máquinas rotatorias grandes con potencia superior 300 kW. Máquinas eléctricas con altura de eje H >= 315 mm.
- **Grupo 2**: Máquinas rotatorias medianas con potencia entre 15 y 300 kW. Máquinas eléctricas con altura de eje 160 =< H =< 315 mm.
- **Grupo 3**: Bombas con impulsor de múltiples álabes y con motor separado (flujo centrífugo, axial o mixto) con potencia superior a 15 kW.
- **Grupo 4**: Bombas con impulsor de múltiples álabes y con motor integrado (flujo centrífugo, axial o mixto) con potencia superior a 15 kW.

NOTA: La altura del eje H de una máquina está definida como la distancia medida entre la línea de centro del eje y el plano basal de la máquina misma. La altura del eje H de una máquina sin patas o de una máquina con pies levantados o cualquier máquina vertical, se debe tomar como la altura de eje H de una máquina horizontal en el mismo marco básico. Cuando el soporte es desconocido, la mitad del diámetro de máquina puede ser utilizada.

Evaluación.
Descripción de la criticidad respecto a la vibración en máquinas rotátivas según la Figura 3.

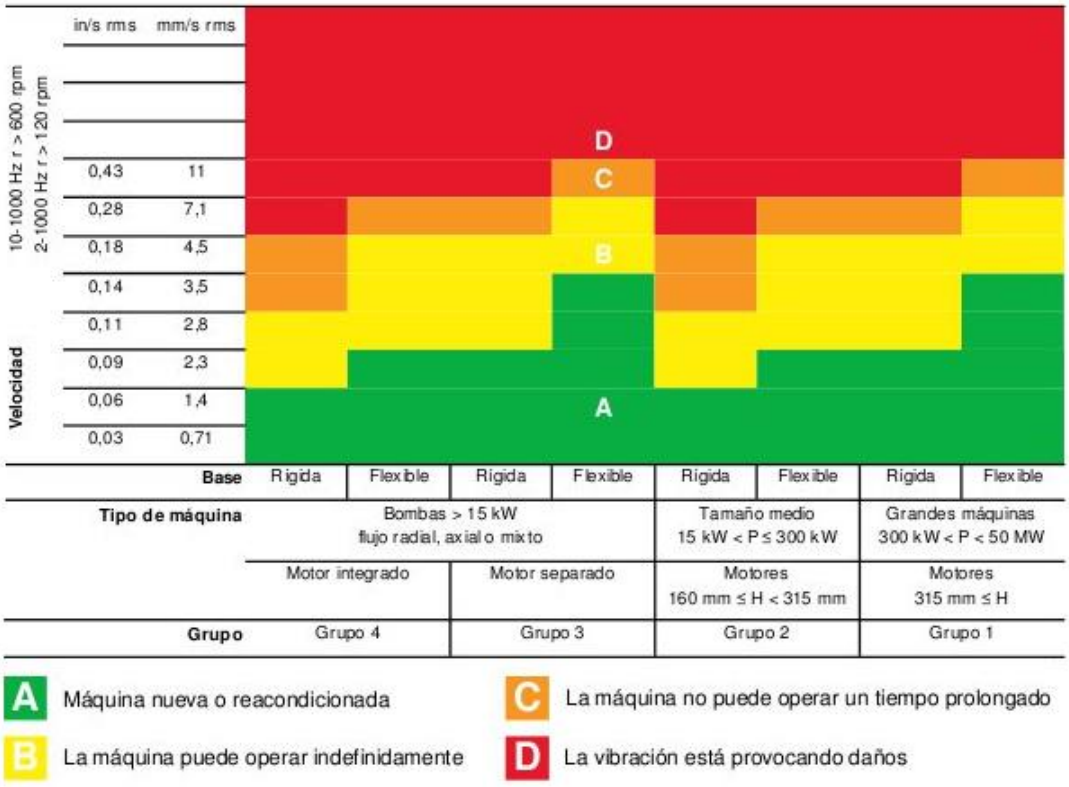
- **Zona A**: Valores de vibración de máquinas recién puestas en funcionamiento o reacondicionadas.
- **Zona B**: Máquinas que pueden funcionar indefinidamente sin restricciones.
- **Zona C**: La condición de la máquina no es adecuada para una operación continua, sino solamente para un período de tiempo limitado. Se deberían llevar a cabo medidas correctivas en la siguiente parada programada.
- **Zona D**: Los valores de vibración son peligrosos, la máquina puede sufrir daños.

Clasificación según la flexibilidad del soporte
Sinais Ingeniería (2013), si la primera frecuencia natural del sistema máquina-soporte en la dirección de la medición es mayor que su frecuencia principal de excitación (en la mayoría de los casos es la frecuencia de rotación) en al menos un 25%, entonces el sistema soporte puede ser considerado rígido en esa dirección. Todos los otros sistemas soportes pueden ser considerados flexibles.

En algunos casos el sistema máquina-soporte puede ser considerado rígido en una dirección de medición y flexible en la otra dirección. Por ejemplo, la primera frecuencia natural en la dirección vertical puede estar sobre la frecuencia principal de excitación mientras que la frecuencia natural horizontal puede ser considerablemente menor. Tales sistemas serían rígidos en el plano vertical; y flexibles en el plano horizontal. En estos

casos, la vibración debe ser evaluada de acuerdo a la clasificación del soporte que corresponda en la dirección de la medición.

Figura 3. Clasificación según la máquina y su criticidad con respecto a la vibración del equipo.



Fuente: (Sinais Ingeniería, 2013).

2.2.2. Norma ISO 1940:1 2003

La norma ISO 1940:1 2003 modificada a la ISO 21940-11: 2016, establece definiciones y condiciones de desbalanceo en las máquinas rotátivas con el fin de priorizar las especificaciones mínimas que se debe cumplir en la fabricación. Especifica las revoluciones, el tipo de máquina, los tipos de desbalanceo, y los grados de desbalanceo permitidos para una máquina nueva según su uso. A continuación se detallan los aspectos más importantes de la norma.

Balance.
ISO (2003), Procedimiento por el cual la distribución en masa de un rotor es comprobada y, si es necesario, se ajusta para asegurar que el desbalance residual o la vibración de los cojinetes y / o fuerzas en los rodamientos a una frecuencia correspondiente a la velocidad de servicio están dentro de límites especificados.

Desbalance.

ISO (2003), condición que existe en un rotor cuando la fuerza de la vibración o movimiento se imparte a sus rodamientos como resultado de las fuerzas centrifugas en la Tabla 1, se establece el tipo de maquinaria el grado máximo de desbalance permitido.

Tabla 1. Grado de desbalance permitido en maquinaria.

Tipos de maquinaria: Ejemplos generales	**Balance, grado de calidad** G	**Magnitud** $e_{per} * \Omega$ mm/s
Turbinas de gas para aviones Centrífugas (separadores, decantadores) Motores y generadores eléctricos (de al menos 80 mm de altura de eje), de velocidades nominales máximas de hasta 950 RPM Motores eléctricos de alturas de eje inferiores a 80 mm Ventiladores, Engranajes, Maquinaria en general Herramientas de máquina, Máquinas de papel Máquinas de plantas de proceso, Bombas Turbocompresores, Turbinas de agua	G 6,3	6,3
Compresores Unidades de ordenador Motores y generadores eléctricos (de al menos 80 mm de altura del eje), de velocidades máximas nominales superiores a 950 RPM Turbinas de gas y turbinas de vapor Unidades para máquinas de herramienta Máquinas textiles	G 2,5	2,5

Fuente: (ISO, 2003).

Desbalance dinámico.

Condición en la que el eje principal central tiene cualquier posición relativa al eje secundario.

Nota 1 En casos especiales, muchos de ellos pueden ser paralelos o pueden cortar el eje.
Nota 2 la medida cuantitativa del desbalance dinámico puede ser dada por dos vectores de desbalance complementarios en dos planos especificados (perpendiculares al eje) que representan completamente el desbalance total del rotor en un estado constante (rígido).

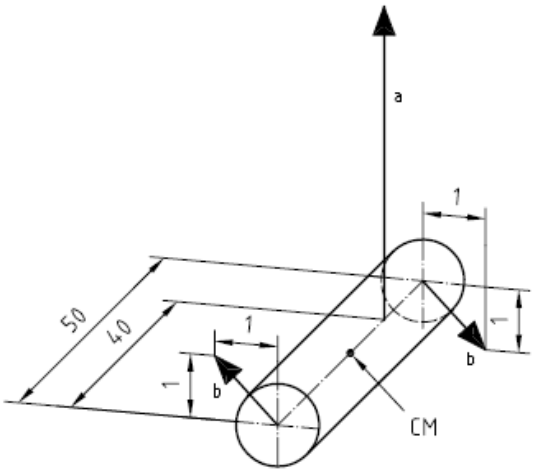
Representación del desbalance.

ISO (2003), el mismo desequilibrio de un rotor en un estado constante (rígido) puede ser representado por cantidades vectoriales de varias maneras, como se muestra en la Figura 4 a) a f).

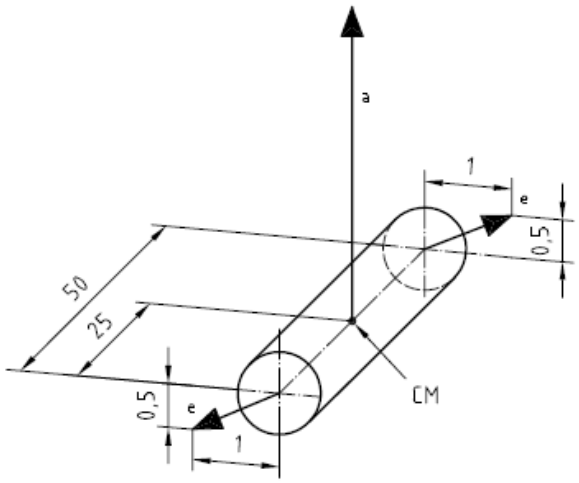
La Figura 4 a) a c) muestran diferentes representaciones en términos de desbalance resultante y el par de desbalance resultante, mientras que la Figura 4 d) a f) son en términos de un desbalance dinámico en dos planos.

Figura 4. Cantidades vectoriales del desbalanceo.

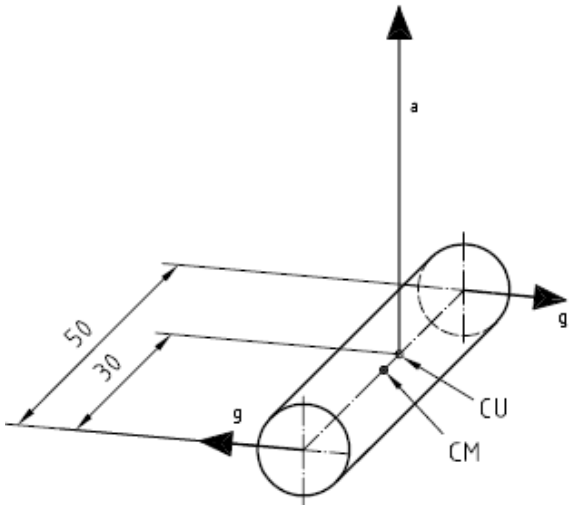
Dimensiones en milimetros



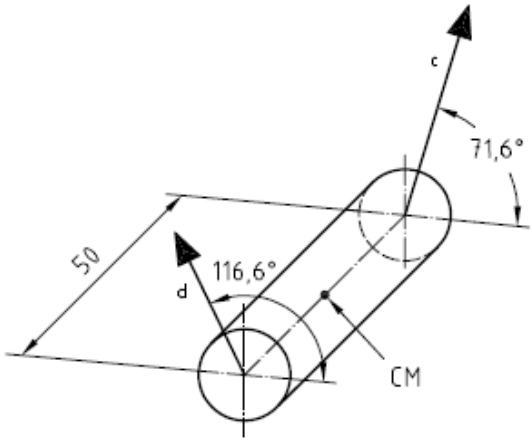
a) Un vector de desbalance resultante junto con un par de desbalance asociado en los extremos del plano.



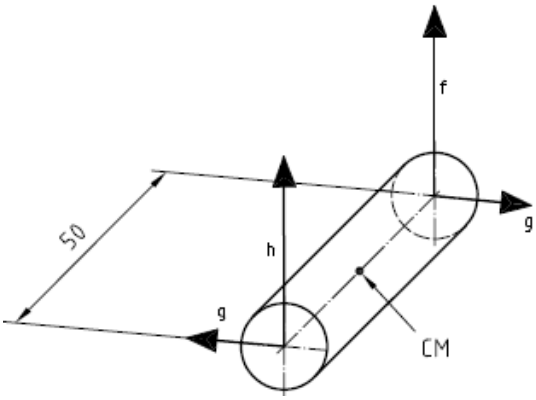
b) Caso especial de a), vector de desbalance situado en el centro de masa CM (desequilibrio estático), junto con un par de desbalance asociado en los extremos del plano.



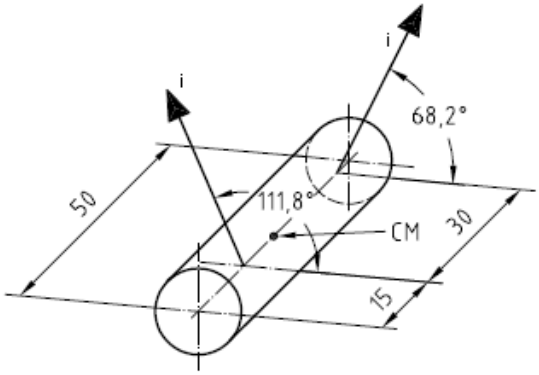
c) Caso especial de a), vector de desbalance resultante situado en el centro del desbalance CU. El par de desbalance asociado es mínimo y se sitúa en un plano ortogonal al vector de desbalance resultante.



d) Un vector de desbalance en cada uno de los extremos del plano.



e) Dos componentes de desbalance a 90° en cada uno de los extremos del plano.



f) Un vector de desbalance en cada uno de los otros dos planos.

a Desbalance igual a 5 g*mm.
b Desbalance igual a 1.41 g*mm.
c Desbalance igual a 3.16 g*mm.
d Desbalance igual a 2.24 g*mm.
e Desbalance igual a 1.12 g*mm.

f Desbalance igual a 3 g*mm.
g Desbalance igual a 1 g*mm.
h Desbalance igual a 2 g*mm.
i Desbalance igual a 2.69 g*mm.

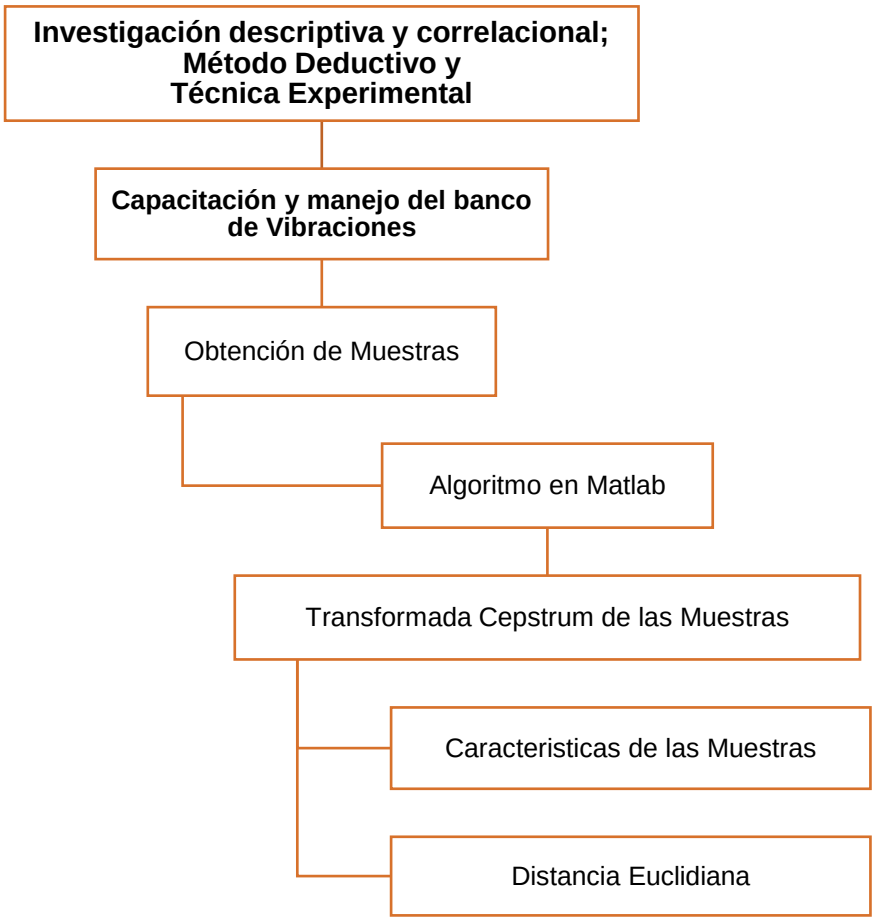
Fuente: (ISO, 2003).

3. DESARROLLO DEL TRABAJO DE GRADO

3.1. METODOLOGIA.

La metodología forma parte importante en el desarrollo de proyectos investigativos, debido al aporte en la sucesión de pasos necesarios para la verificación de las hipótesis planteadas. En esta sección se detalla el tipo de investigación, el enfoque, el método y la técnica. La metodología planteada en diagrama de flujo Figura 5; empleada para el desarrollo de los objetivos específicos planteados en este proyecto, se explica entre la sección 3.2, y la sección 3.3.

Figura 5. Metodología del Proyecto.



Fuente (Autor).

Tipo de investigación y enfoque.

El presente proyecto se ajusta a dos tipos de investigación; descriptivo y correlacional. El proyecto es descriptivo porque pretende encontrar las características propias de un fenómeno en máquinas rotátivas, el cual se subdivide en desbalanceo y desalineación. El proyecto es correlacional porque busca encontrar de manera cuantitativa la relación de tres

tipos de grupos principales; de control, de desbalanceo y de desalineación. La relación entre el grupo de referencia y cada una de las variables de estos grupos se medirá por medio de la distancia euclidiana en la sección 3.3.2.5.

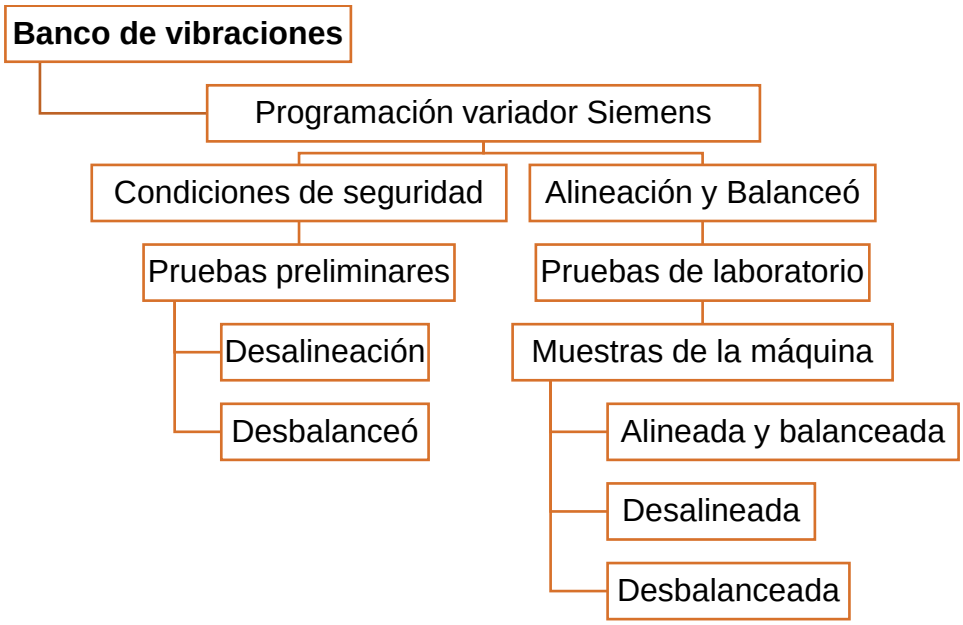
Método.
El método acorde a los objetivos específicos de este proyecto es el método deductivo, debido a que se parten de teorías generales consignadas en el marco teórico a situaciones particulares como los objetivos planteados en este proyecto.

Técnica.
Las variables de este proyecto son acordes a la técnica experimental, debido a que existe la manipulación de las variables independientes con varios grados de libertad, además, se requiere de un grupo de referencia o control para validar los datos obtenidos.

3.2. BANCO DE VIBRACIONES.

En esta sección se especifican todas las actividades Figura 6, necesarias para el desarrollo del primer objetivo específico 1.3.2.1, el cual tiene como finalidad la obtención de los registros de las diferentes muestras necesarias para validar el proyecto. También se profundiza en la forma como se abordó este objetivo, los desafíos que presento la obtención de los registros; así como, el procedimiento para normalizar el estado de la máquina a la condición de balanceada y alineada.

Figura 6. Diagrama de actividades de la sección 3.2.



Fuente: (Autor)

3.2.1. El variador.

En esta sección no se entrara en detalles relevantes a la construcción del variador, tampoco profundiza en su programación; solo pretende especificar aspectos relevantes al manejo del banco de vibraciones, los cuales son importantes para evitar choques eléctricos que puedan afectar de manera irreversible a todos los componentes electrónicos del banco.

El variador siemens Figura 7, se ajusta a las condiciones de placa de los motores en los cuales interviene, aunque existe una función de ajuste rápido; en esta sección, se aborda el procedimiento realizado para programar el variador Tabla 2; a las características necesarias para la ejecución de las pruebas de laboratorio.

Para iniciar la programación se debe hacer uso de los botones Fn y P situados junto a las flechas de dirección Figura 7. Al oprimir el botón Fn se accede al menú de programación del variador y con el botón P se sale del menú de programación, este botón presenta otras opciones en las cuales no se profundiza. Después de oprimir Fn, se procede a desplazarse en el menú de programación con la tecla para subir.

Para iniciar la programación se debe buscar el parámetro $P003$ oprimir P y buscar la opción 2 la cual permite modificar los parámetros del variador Tabla 2. Después, se debe buscar el parámetro $P0010$, oprimir P y seleccionar la opción 1 la cual pre-ajusta la configuración del variador. Después, se debe verificar si el pre-ajuste es apropiado para las características de placa del motor. El parámetro $P0100$ opción 2 selecciona la frecuencia a 60 Hz. El parámetro $P0304$ permite verificar la tensión nominal del motor, si el pre-ajuste no pertenece al voltaje nominal del motor se puede utilizar las flechas de subir y bajar Figura 7 para corregir el valor del voltaje.

Los siguientes parámetros son para verificación del pre-ajuste respecto a las características de placa del motor. El parámetro $P0305$ permite modificar la corriente nominal, el parámetro $P0307$ permite modificar la potencia nominal en KW, el parámetro $P0308$ permite modificar el factor de potencia, el parámetro $P0310$ permite verificar la frecuencia seleccionada y el parámetro $P0311$ permite verificar la velocidad nominal del motor.

Los siguientes parámetros se utilizaron para ajustar el motor a las condiciones de seguridad mencionadas en la sección 3.2.2. El parámetro $P1000$ opción 3 fija la frecuencia del motor; por lo que, después de guardados los valores programados no permite modificar la frecuencia del motor hasta una nueva programación. El parámetro $P1080$ permite modificar la frecuencia mínima de trabajo, para este parámetro se seleccionó una frecuencia de 30 Hz. El parámetro $P1082$ permite modificar la frecuencia máxima de trabajo, para este parámetro se seleccionó una frecuencia de 30 Hz. El parámetro $P1120$ permite modificar el tiempo de arranque del motor, para este parámetro se seleccionó un tiempo de 20 segundos y el parámetro $P3900$ opción 3, guarda todos los parámetros de la programación realizada al motor.

Cabe resaltar que es muy importante verificar antes de accionar el motor, que las características de placa sean las mismas que se programan en el variador; con el fin de evitar daños por sobrecargas a componentes electrónicos del banco.

Figura 7. Variador Siemens micromaster 420.

Fuente: (Autor).

Tabla 2. Parámetros de programación variador Siemens.

Parámetro	Configuración
P0003	Opción 2, modifica los parámetros del variador.
P0010	Opción 1, programación del motor.
P0100	Opción 2, Selección de frecuencia a 60 Hz.
P0304	Modifica la tensión nominal.
P0305	Modifica la corriente nominal.
P0307	Modifica la potencia en KW
P0308	Modifica el factor de potencia.
P0310	Frecuencia nominal.
P0311	Velocidad nominal.
P1000	Opción 3, No permite modificar la frecuencia.
P1080	Selección de frecuencia mínima.
P1082	Selección de frecuencia máxima.
P1120	Selección tiempo de arranque.
P1121	Selección tiempo de frenado.
P3900	Opción 3, Guarda las modificaciones.

Fuente: (Siemens, sf.).

3.2.2. Manejo del Banco de Vibraciones.

En la Figura 8, se presenta una fotografía del banco de vibraciones; el cual se compone de un variador Siemens, un motor Weg de medio caballo de fuerza, un eje secundario con dos volantes, un acople flexible, un breaker principal, dos switch de encendido, un switch de paro de emergencia, una bancada de hierro anclada a una mesa de madera y está a su vez anclada al suelo por medio de una base metálica.

El manejo del banco de vibraciones es secuencial y repetitivo por lo que en primera instancia no presenta una complicación importante. Para iniciar, se debe accionar un breaker principal el cual a su vez forma parte de la protección de todos los componentes del banco.

En la Figura 8, en la parte superior izquierda se pueden ver dos switch de color negro, las luces piloto, y el paro de emergencia de color rojo. El primer switch al accionarse enciende el variador siemens micromaster 420 Figura 7 y permite accionar el segundo switch el cual energiza el motor Figura 9; antes de energizar el motor, se debe programar el variador sección 3.2.1.

Las pruebas preliminares realizadas al banco constan de la realización de los fallos de desbalanceo y desalineación. Para las pruebas preliminares de desbalanceo, se produjo un desbalance por prueba en cada uno de los volantes, para una masa y dos masas por volante; también se realizó la prueba colocando masa en los dos volantes con el fin de verificar el comportamiento en general del banco.

Para las pruebas preliminares de desalineación se realizó el corrimiento de los apoyos para distintos grados corriendo el apoyo más lejano del motor respecto al apoyo más cercano y corriendo ambos apoyos el mismo grado.

Las limitantes que presenta el banco de vibraciones encontradas en las pruebas preliminares en cuanto a su construcción y seguridad son tres principalmente. La velocidad es una de las principales limitantes en la Figura 9, se puede observar las características de placa del motor Weg; el cual posee una velocidad de 3340 RPM.

La velocidad es una limitante no por defectos propios del motor; sino, por la construcción como tal del banco; debido a que a la frecuencia nominal del motor, la bancada en general vibra con una intensidad inesperada para una prueba en condiciones de balanceo y alineación aparente; indicando que la velocidad nominal del motor sobrepasa las condiciones de estabilidad del banco, considerando esta situación como un riesgo en general para el operador y para el banco; debido a que podría fallar prematuramente por fatiga los distintos elementos de sujeción. Al no poderse reducir la velocidad del motor se hace uso del variador sección 3.2.1 para cambiar su frecuencia de 60 Hz a 30 Hz donde el banco permanece firme respecto a la rotación del motor.

La segunda limitante que presenta el banco es respecto a su seguridad según las pruebas preliminares para la falla de desalineación, si bien, el eje secundario se puede desalinear en forma práctica hasta 5° respecto al apoyo más cercano al motor; frente a la rotación, los

rodamientos que son los elementos del banco más afectados en esta prueba; presentan ruidos inusuales, afectando su funcionamiento en tiempos prolongados del fallo; por lo que, se determina de forma practica el ángulo en el cual los rodamientos no presenten ruidos inusuales, con el fin de preservar la funcionalidad del banco de vibraciones.

Los ángulos encontrados que cumplen esta condición son todos aquellos ángulos menores de 1.5°; los cuales no presentan ruidos inusuales cuando se realiza una prueba de media hora con la falla provocada al eje secundario.

Figura 8. Banco de vibraciones.

Fuente: (Autor).

La última limitante también es de seguridad respecto a las pruebas preliminares de desbalanceo. Al realizar estas pruebas se encontró de forma práctica que los apoyos se mantienen estables para las pruebas de una y dos masas en cada uno de los volantes; sin embargo, para la prueba donde se adiciona masa a los dos volantes se utilizó el paro de emergencia para proteger el banco de vibraciones; debido a que a la mitad del tiempo de arranque se indujo en el banco una inestabilidad critica afectando la seguridad de los apoyos y poniendo en riesgo al operador.

Debido a las pruebas preliminares se pudo establecer condiciones de seguridad para el banco de vibraciones, las cuales permitan realizar fallos en el banco, sin causar un daño permanente en uno o más elementos que lo conforman. También con las pruebas preliminares se pudo establecer condiciones de seguridad para el operador, entre las más importantes el uso de protectores auditivos, careta de seguridad, guantes dieléctricos y botas dieléctricas.

Figura 9. Características de motor Weg.

Fuente: (Autor).

Después de conocer el funcionamiento del banco de vibraciones y los aspectos relevantes a la seguridad para su manejo, se puede proceder a realizar las pruebas de referencia, de desbalanceo y desalineación, estas pruebas se explicaran a profundidad en la sección 3.2.4.

3.2.3. Balanceó y alineado.

En esta sección se profundiza en todas las acciones encaminadas al balanceó y alineado del banco de vibraciones. El banco de vibraciones cuenta con un programa de seguimiento en Labview utilizando la transformada wavelet, para realizar la caracterización de las señales tomadas por un sensor de aceleración de marca dytran.

Es pertinente realizar el balanceó y alineado del banco de vibraciones debido a que se pudo comprobar que el banco requería de alineación por los espectros formados en el programa Labview para las pruebas preliminares de la sección 3.2.2, pero no se pudo distinguir que tipo de desalineación presentaba el banco de vibraciones; por lo que, se asume que el

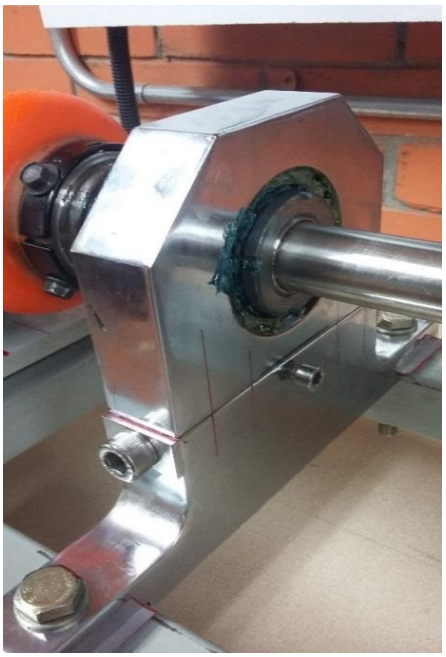
banco posee una desalineación combinada. Las acciones realizadas para lograr el alineamiento del banco y la comprobación del balanceó en cada uno de los volantes se describe a continuación.

Las primeras acciones para lograr alinear el banco fueron, determinar los instrumentos necesarios para poder realizar el despiece del banco en general; es decir, encontrar las medidas de los tornillos y tuercas de sujeción de los diferentes elementos que conforman el banco de vibraciones; motor, eje secundario, bancada, mesa y base metálica de anclaje al suelo.

Al realizar una verificación de la referencia de los tornillos con llaves milimétricas, se pudo comprobar de forma experimental que los tornillos de los apoyos deslizantes, la bancada de sujeción del motor y la base metálica de anclaje al suelo Figura 8 pertenecen a llaves No 17. Los tornillos de la bancada metálica Figura 8, donde se apoya el motor y el eje secundario respecto a la mesa pertenecen a llaves No 20. Los tornillos que unen la madera de la mesa a la base metálica de anclaje al suelo pertenecen a llaves No 22 y los tornillos del acople flexible son de referencia 3/8 en pulgadas.

Para poder graduar los apoyos deslizantes Figura 10, estos cuentan con tres tornillos de sujeción tipo brístol; dos ubicados en sus laterales y uno ubicado en su cara frontal. El tornillo principal es el lateral izquierdo, el cual se puede graduar con una llave milimétrica Bristol No 8. El tornillo lateral derecho se puede graduar con una llave No 1/8 en pulgadas y el tornillo frontal se puede graduar con una llave No 3/16 en pulgadas. Después de comprobar los instrumentos necesarios para realizar el desarme del banco, se procedió a su realización.

Figura 10. Apoyo deslizante.



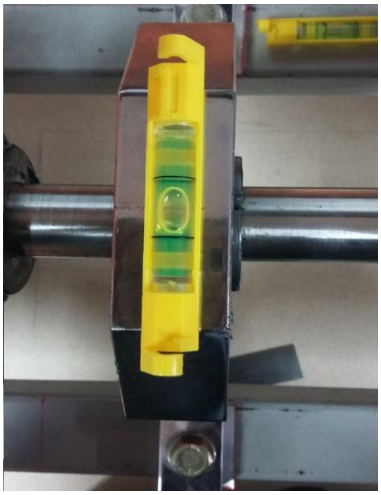
Fuente: (Autor).

En el proceso de armado se contó con 4 niveladores tipo burbuja Figura 11, con el fin de posicionarlos en cada una de las esquinas de la mesa, bancada y apoyos. Se pudo comprobar al colocar los niveladores en la mesa, que esta se encontraba más elevada en su parte frontal respecto a su parte posterior; por lo que, se procedió a su nivelación retirando material tipo goma utilizado inicialmente para su montaje.

Después de nivelar la mesa se procedió a comprobar la nivelación de la bancada, resultando desnivelada; por lo que, se procedió a retirar material tipo caucho utilizado inicialmente para su montaje.

Al verificar la alineación en el motor y los apoyos del eje secundario, se pudo comprobar por medio de los niveladores que estos presentaban desalineación paralela, en menor medida, que la presentada inicialmente antes del desensamble; por lo que, se procedió a la creación de unas galgas para poder realizar esta alineación. En la Figura 12 a) se puede ver la forma de la galga la cual posee las siguientes dimensiones Figura 12 b); $3.6\ cm$ de largo, $3\ cm$ de ancho y un grosor de 0.23 milímetros. La cavidad en la galga donde entra la base del tornillo esta espaciada a lo largo así; $1.3\ cm$ de sus extremos y $1\ cm$ por $1.7\ cm$ de ancho de perforación

Figura 11. Niveles tipo burbuja.

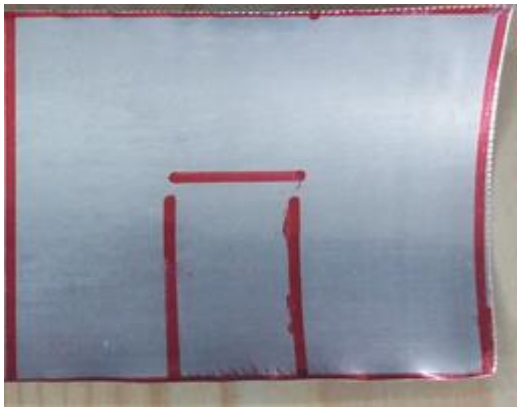


a) Nivelador montado a lo largo del apoyo primario.

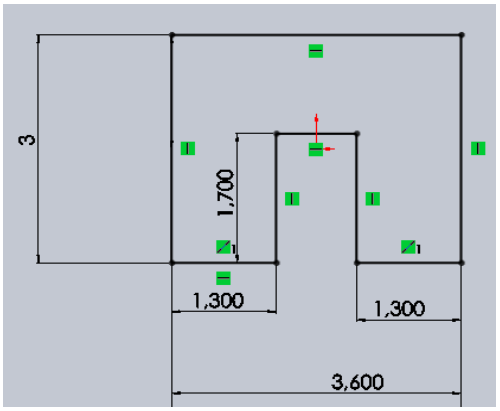
b) Nivelador montado a lo ancho del apoyo primario.

Fuente: (Autor).

Figura 12. Galgas de balanceó para los apoyos.



a) Galga en aluminio.



b) Diseño de la galga en SolidWorks.

Fuente: (Autor).

Al nivelar la bancada, se procedió al montaje del motor y del eje secundario sin realizar la unión por medio del acople flexible; donde se pudo comprobar por medio de los niveles tipo burbuja Figura 11, que las acciones tomadas para alinear paralelamente el banco en general resultaron apropiadas.

Para alinear angularmente el segundo eje respecto al motor, se realizó el corrimiento de los apoyos por medio del tornillo tipo Bristol de 8 mm y una regla metálica delgada; para comprobar los desniveles entre las cabezas de acople de cada uno de los ejes. Se movieron los apoyos hasta que se logró un nivel apropiado en los cuatro puntos principales de alineación angular sección 2.1.4.

En la Figura 13, se evidencian todas las correcciones realizadas en los apoyos; del eje secundario, del motor, de la bancada y de la estructura metálica; de las cuales se describió anteriormente las correcciones necesarias para conseguir una apropiada alineación.

Figura 13. Correcciones en los apoyos por medio de la adición de galgas.

a) Apoyo base metálica suelo.

b) Apoyo bancada del motor y eje secundario respecto a la mesa.

c) Apoyos del motor con la bancada.

d) Apoyo eje secundario más cercano al motor.

Fuente: (Autor).

Para comprobar el balanceo de los volantes, se realizó una comprobación de balanceó en un plano para cada volante como se describe en la sección 2.1.6. Terminados todos los ajustes correspondientes al balanceó y la alineación; se pueden realizar las pruebas de

laboratorio sección 3.2.4; contando con la normalización del banco de vibraciones en su estado de alineado y balanceado.

3.2.4. Pruebas de laboratorio.

Las pruebas de laboratorio son necesarias para identificar los grupos con los cuales se va a realizar el proyecto. También es importante, porque de las pruebas de laboratorio saldrán todas las características que se deseen encontrar, a los fenómenos de desbalanceo y desalineación. El punto para la toma de datos, se localizó en el apoyo más lejano del motor, en posición horizontal a la mitad de la distancia del rodamiento y la adquisición de las señales se realizó por medio del acelerómetro piezoeléctrico dytran. Todas las muestras se realizaron a una frecuencia de 30 Hz debido a las condiciones de seguridad obtenidas de las pruebas preliminares sección 3.2.2.

El montaje y adquisición de datos se realizó por medio de un sensor de marca dytran, una tarjeta de adquisición de National Instruments y un programa en Labview (Forero, Lizcano, Coronel, 2016, p.85);

3.2.4.1 Calculo del tamaño de la muestra.

El cálculo del tamaño de la muestra se realizó como aparece en la sección 2.1.12; aplicando la ecuación (21); para $Z = 1.6448$, $P = Q = 0.5$ y $E = 0.05$.

$$n = \frac{1.6448^2 * 0.5 * 0.5}{0.05^2} = 270.55 \cong 271$$

El tamaño de muestra obtenido con un error del 5% resulta demasiado grande para el estudio de una población infinita sin precedentes, por lo que se corrige a un error del 10%.

$$n = \frac{1.6448^2 * 0.5 * 0.5}{0.10^2} = 67.63 \cong 68$$

El nuevo tamaño obtenido de la muestra con un error del 10% es apropiada para el estudio de fallas en máquinas rotátivas; cabe resaltar que las 68 muestras corresponde realmente a 68 paquetes de datos, con un periodo muestral de 2500 datos, según el programa del banco de vibraciones en Labview.

3.2.4.2 Muestras máquina alineada y balanceada.

Las muestras tomadas de la máquina alineada y balanceada formaron parte del grupo de referencia, en el cual no intervienen las variables de los grupos desalineado y desbalanceado del proyecto. El grupo del referencia será nombrado de ahora en adelante con la sigla GCAB (Grupo de control de alineación y balanceó).

Después de realizar las correcciones de balanceó y alineación en el banco de vibraciones sección 3.2.3; se procedió a realizar la toma de datos correspondiente a la variable GCAB. Se realizarón 3 tomas de datos de aproximadamente 80 muestras, con el fin de poder seleccionar aquellas muestras que se encuentren más agrupas, en referencia a la amplitud de la frecuencia fundamental del banco de vibraciones; pasando un algoritmo de selección en Matlab sección 3.3.2.2. Esto con el fin de validar la repetición de los datos del grupo de referencia.

3.2.4.3 Muestras máquina desbalanceada.

Para el fenómeno de desbalanceo se realizarón diferentes pruebas las cuales incluyen colocar una masa conocida Figura 14, (Es un tornillo de cinco dieciseisavos de pulgada de paso milimétrico, con doble tuerca y arandela; de pulgada y media de vástago); en uno de los volantes del eje secundario, el volante que esté más cerca al motor eléctrico será llamado en adelante *volante primario*.

La masa conocida se colocó a dos distancias en el volante primario y en el volante secundario, las cuales ya presenta cada uno de los volantes. Las variables utilizadas forman dos grupos con dos grados de libertad; para esta prueba las variables se describen en la Tabla 3, las cuales representan todos los casos seleccionados para la prueba de desbalanceo ajustados a las condiciones de seguridad de la sección 3.2.2.

Figura 14. Desbalanceo en el volante primario a una distancia r2.

Fuente: (Autor)

Tabla 3. Variables seleccionadas para la prueba de desbalanceo.

Símbolo	Descripción
DBv1r1m	Colocar una masa m1 en el volante 1 a la distancia r1
DBv1r2m	Colocar una masa m1 en el volante 1 a la distancia r2
DBv2r1m	Colocar una masa m1 en el volante 2 a la distancia r1
DBv2r2m	Colocar una masa m1 en el volante 2 a la distancia r2

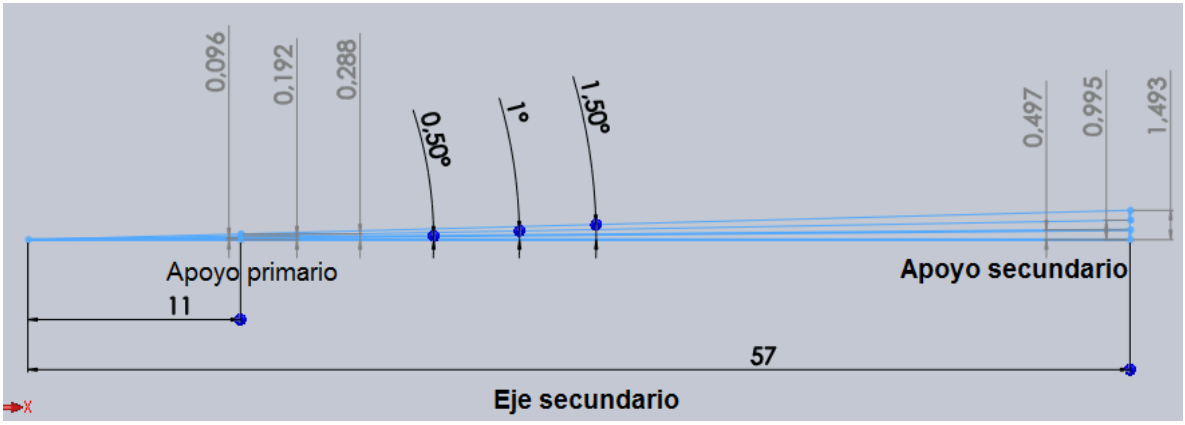
Fuente: (Autor).

3.2.4.4 Muestras máquina desalineada.

Para el fenómeno de desalineación se realizarón corrimientos en forma diagonal y hacia atrás del eje secundario respecto al eje primario, en sus dos puntos de apoyo Figura 8. El punto de apoyo del eje secundario más próximo al motor es nombrado de ahora en adelante *apoyo primario*. El apoyo primario se corrió de forma lineal así; para $0.5°$ grados se corrió $0.096 \cong 0.1$ cm, para $1.0°$ grados se corrió $0.192 \cong 0.2$ cm y para $1.5°$ grados se corrió $0.288 \cong 0.3$ cm.

Para el apoyo secundario los desplazamientos lineales fueron así; para $0.5°$ grados se corrió $0.497 \cong 0.5$ cm, para $1.0°$ grado se corrió $0.995 \cong 1.0$ cm y para $1.5°$ grados se corrió $1.493 \cong 1.5$ cm. En la Figura 15 se puede apreciar de forma sencilla los valores descritos anteriormente; la precisión en las medidas se debe a un dibujo del eje secundario en SolidWorks; donde se establece una longitud desde él acople hasta el segundo apoyo de $57\,cm$, y una distancia del acople con el apoyo primario de $11\,cm$. Las variables seleccionadas forman un grupo con tres grados de libertad; en la Tabla 4 se identifican las variables para cada prueba ajustadas a las condiciones de seguridad de la sección 3.2.2.

Figura 15. Eje secundario en SolidWorks.



Fuente: (Autor).

Tabla 4. Variables seleccionadas para la prueba de desalineación.

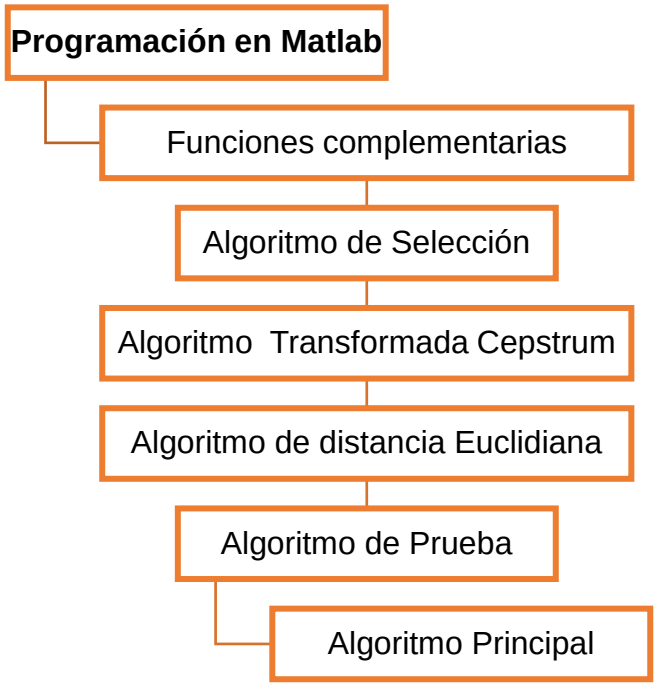
Símbolo	Descripción
DA05g	Movimiento de los apoyos para 0.5° grados
DA10g	Movimiento de los apoyos para 1.0° grados
DA15g	Movimiento de los apoyos para 1.5° grados

Fuente: (Autor).

3.3. Matlab.

En esta sección se especifican todas las actividades necesarias Figura 16, para la realización de los objetivos específicos 1.3.2. Matlab al ser un programa interdisciplinar, cuenta con varios recursos que permiten crear distintas actividades; por medio de la ejecución de códigos de programa los cuales permiten realizar lo que el programador ideo.

Figura 16. Diagrama de actividades de la sección 3.3.



Fuente: (Autor).

3.3.1. Programación básica en Matlab.

En esta sección se explica la programación básica en Matlab que debe conocer el lector, para poder entender los diferentes llamados, las operaciones, la creación de archivos .m y funciones; que se utilizarán en este proyecto. Si el lector desea profundizar sus conocimientos en Matlab puede referirse a (Ramírez, 2014).

3.3.1.1 Operaciones.

En Matlab se pueden realizar operaciones matemáticas desde suma de variables escalares hasta operaciones con matrices o vectores. Los operadores utilizados en este proyecto aparecen descritos en Tabla 5 donde se describe su acción en Matlab.

Tabla 5. Operadores de Matlab.

Operador	Descripción
+	Suma los argumentos que estén involucrados.
-	Resta los argumentos que estén involucrados
*	Multiplicación.
.*	Multiplicación interna de vectores o matrices con un escalar.
/	División.
^	Potencia de un número, matriz o vector.
‘ ’	Comillas simples; utilizadas para ingresar vectores tipo *char*.
%	Comentarios.

Fuente: (Ramírez, 2014).

3.3.1.2 Tipos de datos.

En Matlab se puede trabajar con distintos tipos de datos; en esta sección se mencionan los tipos de dato con los cuales se realizó este proyecto.

Los tipos de datos se demarcan en Matlab dentro de la ventana $\boldsymbol{Workspace}$ en la columna $Class$. Los números reales enteros o decimales se denotan en Matlab como *doublé*. Los números complejos se denotan en Matlab como $doublé\ (complex)$. Las palabras se denotan en Matlab como $char$ y las estructuras se denotan en Matlab como $struct$.

En Matlab se pueden realizar operaciones con datos Tabla 5 del tipo *doublé* y $doublé\ (complex)$ contenidos en un vector, en una matriz, en un vector celdas y en una estructura tipo vector o tipo matriz. Por lo general los datos tipo $char$ se utilizan para realizar comparaciones o para guardar nombres de archivos dentro de un vector de $char$.

3.3.1.3 Vectores y Matrices.

Los vectores y las matrices en Matlab son los espacios de trabajo especializados de este programa. Los vectores se pueden definir en Matlab desde la ventana $\boldsymbol{Command\ Windows}$; donde se puede elegir si se desea introducir, un vector fila o un vector columna.

Los vectores fila se pueden ingresar nombrando la variable en este caso (x), a la cual pertenecen e ingresando los datos que estos contienen dentro de corchetes, separados por espacios en blanco o por comas, ejemplo.

$$x = [1\ 2\ 3\ 4\ 5] \quad \text{Ej. 1}$$
$$x = [1,2,3,4,5] \quad \text{Ej. 2}$$

Los vectores columna se pueden ingresar en Matlab nombrando la variable y separando los datos por *(;)* dentro de corchetes, ejemplo.

$$x = [1; 2; 3; 4; 5] \quad \text{Ej. 3}$$

Las matrices en Matlab se definen como $m\ x\ n$; en donde m hace referencia a las filas y n hace referencia a las columnas. La matriz más utilizada en este trabajo es la matriz $\boldsymbol{zeros}$; la cual ayuda a predefinir el espacio requerido en la memoria de procesamiento del computador haciendo más rápido los algoritmos. También la matriz $\boldsymbol{zeros}$ es conocida como una función la cual devuelve ceros respecto a la matriz ingresada. Las matrices también requieren que se defina la variable a la cual van a ser asignadas y debe ingresarse por medio de paréntesis, ejemplo.

$$a = zeros(10,3) \quad \text{Ej. 4}$$

Según la definición anterior a es una matriz de 10 filas y 3 columnas; la función $\boldsymbol{zeros}$ hace que el vector llene sus espacios internos con ceros.

3.3.1.4 Condicionales.

Los condicionales $\boldsymbol{if}$ y $\boldsymbol{if\ else}$; así como en lógica computacional o lógica booleana son instrucciones que permiten comparar diferentes situaciones, que puedan desarrollarse dentro de los algoritmos de Matlab; con el fin de tomar una decisión respecto a la comparación. En la Tabla 6 se describen algunos operadores utilizados como decisión dentro de los condicionales.

El condicional $\boldsymbol{if}$ se utiliza cuando se quiere compara dos argumentos y si la comparación es verdadera, la instrucción dentro del condicional se ejecuta, ejemplo.

```
If 1>a(1,1)  % Es mayor 1 que la posición (1,1) de la matriz a
    b=1;     % Sí, es cierto la variable b es igual a 1
end
```
Ej. 5

En el ejemplo Ej. 5 se evidencia una forma recursiva de utilizar los comparadores la cual en este proyecto se encontrara inmersa dentro ciclos tipo $\boldsymbol{For}$. Otra forma de utilizar los condicionales es con variables tipo $\boldsymbol{char}$, ejemplo.

```
If  strcmp('GCAB',r)==1 % Función para comparar argumentos tipo char
    b=2;     % Sí, es cierto la variable b es igual a 2
end
```
Ej. 6

En el ejemplo Ej. 6 se hace uso de la función $\boldsymbol{strcmp}$; la cual debe usarse como aparece en el ejemplo; es ***GCAB*** igual a la variable $\boldsymbol{r}$ si es cierto la variable b será igual a 2. Esta función permite comparar variables tipo $\boldsymbol{char}$ regresando como respuesta lógica; 1 si son

iguales y 0 si son diferentes. Esta función es especialmente utilizada en este trabajo porque es sensible a las letras mayúsculas o minúsculas limitando la decisión a lo que desea el programador.

El condicional ***if else***, es un adicional al condicional tipo ***if*** ya que si la condición de comparación no se cumple se realiza las operaciones que se encuentren dentro del ***else***; ejemplo.

```
If 1>a(1,1)  % Es mayor 1 que la posición (1,1) de la matriz a
    b=1;     % Sí, es cierto la variable b es igual a 1
else                                                          Ej. 7
    b=2;     % Sí, es falso la variable b es igual a 2
end
```

En el ejemplo Ej. 7 si el condicional no se cumple la variable ***b*** tomara el valor de 2. Con lo descrito anteriormente se resumen las instrucciones que serán utilizadas dentro de los algoritmos creados en Matlab para este proyecto.

Tabla 6. Operadores para los condicionales ***if*** y ***if else***.

Operador	Definición
==	Igual, igual; permite comparar si los argumentos son iguales.
>=	Mayor o igual que.
<=	Menor o igual que.
>	Mayor que.
<	Menor que.
&&	Condicional tipo Y.
||	Condicional tipo O.
~	Diferente que.
strcmp	Función utilizada para comparar variables tipo ***char***; con devolución lógica ***1*** si son iguales ***0*** si no son iguales.

Fuente: (Ramírez, 2014).

3.3.1.5 Ciclos *For* y *while*.

Los ciclos *For* y *while* permiten realizar repeticiones tantas veces como los rangos ingresados en el contador. Por lo general son utilizados para realizar diferentes operaciones sección 3.3.1.1 con variables internas de matrices o vectores sección 3.3.1.3.

El ciclo *For* es una repetición interna de las instrucciones que se encuentren dentro de su definición. Para ingresar al ciclo *For* basta con definir una variable y un rango en el cual se deben repetir las instrucciones en Matlab.

En este proyecto se utilizan los ciclos *For* por lo general para leer los datos de las muestras siendo el rango por lo general de 1 hasta el tamaño encontrado por la función ***length***, ejemplo.

```
for  i=1:5   % Los : denotan rango
     if x(i)<=3  % Es el vector x en su posición i igual que 3
         b(i)=x(i);      % Si, es cierto construya el vector b
     else
         a(i)=x(i);   % si, no es cierto construya el vector a
     end
end
```

Ej. 8

En el ejemplo Ej. 8 se usa el ciclo For unido a un condicional con el fin de crear dos vectores b y a. Se debe tener especial cuidado con el contador del ciclo For, debido a que si el contador excede el tamaño; sea del vector o de una matriz, Matlab terminará lo que esté realizando para sacar un error de tamaño de matriz o vector.

El ciclo $while$ realiza operaciones mientras el condicional con el cual se defino no sea cierto, cuando el condicional se cumple el ciclo se termina, ejemplo.

```
r='N     % se predefine la variable r para ingresar al ciclo
while  strcmp(r,'Y')==1   % Los : denotan rango
     r = input('Desea continuar digite Y', 's'); % input es una función para solicitar
                                                 % el ingreso de datos
end
```

Ej. 9

En el ejemplo Ej. 9 se utiliza el ciclo $while$ para solicitar al usuario si desea continuar con algunas instrucciones que venía realizando el programa. Esto se realiza por medio de la función $input$ la cual permite solicitarle al usuario una decisión; el condicional hace que el ciclo se repita mientras la comparación de caracteres no sea cierta cuando el condicional sea verdadero el ciclo se terminara.

3.3.1.6 Estructuras, archivos .m y funciones.

Las estructuras o $struct$ en Matlab son variables de tipo apuntador, las cuales permiten almacenar matrices, vectores de números o de caracteres, escalares, entre otros datos; que pueden ser guardados dentro de la estructura.

El comportamiento de la estructura es muy similar a una matriz vea sección 3.3.1.3, donde por medio de paréntesis se apunta a la característica que se desee extraer de la estructura. Las estructuras son relevantes en este proyecto, debido a que la mayoría de algoritmos fueron diseñados con esta condición; con el fin de facilitar el manejo de las distintas variables con las cuales se va a trabajar.

La definición de una estructura en Matlab es muy fácil de realizar. Una estructura puede ser tan profunda como llamados internos se le desee dar, ejemplo.

```
y(1).y=x; % El operador .y define la estructura "y" con el apellido .y
y(2).y=a;
y(3).y=b;
y(2,1).y=a;
y(3,1).y=b;
```

Ej. 10

En el ejemplo Ej. 10 las primeras tres definiciones de la estructura y guardan en la primera fila el valor de la variable x, a y b. En las dos últimas definiciones de la estructura y se utiliza un llamado tipo matriz, para indicarle que debe guardar los vectores a y b en la segunda y tercera fila de la columna 1.

Para poder utilizar los valores del vector x; y de la matriz ***a***, se debe llamar la estructura de la siguiente forma, ejemplo.

```
x1=y(1).y(1); % El operador .y define la estructura "y" con el apellido .y
x2=y(2).y;
```

Ej. 11

En el ejemplo Ej. 11 se guardan en variables distintas los llamados a la estructura y; donde, en $x1$ se está guardando el primer valor de la variable x guardado en la estructura y Ej. 10, en su primera fila y columna. En $x2$ se está guardando los valores de la variable a, la cual inicialmente era una matriz de ceros de 10 filas y 3 columnas Ej. 4.

Los archivos .m o scripts son en general ventanas en las cuales se puede escribir un código o algoritmo en Matlab, guardar y abrir para utilizarlo después. En general, en este proyecto no se trabaja con archivos .m en la calidad de scripts; sino que, se utiliza la opción función de un script.

Las funciones son archivos con extensión .m, donde se da un nombre a la función, valores de entrada y unos retornos o variables de salida. Un archivo .m puede contener solo una función o varias funciones, donde se subdivide las funciones dentro del archivo .m en función principal y sub-funciones, ejemplo.

```
function [b,a,s] = ejemplo (A) % Variable A de entrada
% Sección principal
for  i=1:5   % Los : denotan rango
     if x(i)<=A  % Es el vector x en su posición i igual que la variable A
         b(i)=x(i);      % Si, es cierto construya el vector b
     else
         a(i)=x(i);   % si, no es cierto construya el vector a
     end
end
[s] = suma (b,a);
end
function [s] = suma (c,d) % Sub-función llamada por la función principal ejemplo
% Sub-función 1
s=c+d;
end
```

Ej. 12

En el ejemplo Ej. 12 se define la función ***ejemplo***, la cual es la función principal del archivo con extensión .m llamado ***ejemplo.m***. En esta función se pide como variable de entrada un número el cual modifica el condicional tipo ***if else*** que se encuentra dentro del ciclo ***For***; modificar el condicional es modificar los vectores a y b que se encuentran dentro de sus condiciones.

La sub-función ***suma*** del ejemplo Ej. 12, pertenece al dominio interno de la función ***ejemplo*** y recibe los vectores a y b de la sección principal, realiza la operación suma y devuelve un vector suma llamado s a la sección principal, el cual a su vez es un argumento de salida de la sección principal que puede ser asignado a una variable en el ***Command Windows***.

La diferencia más importante entre un script y una función es que el script permite visualizar todas las variables en el ***Workspace***; mientras que, en la función solo se visualiza en el ***Workspace*** las variables de salida. Esto quiere decir, que si en él ***Command Windows*** se definió la función como $[b] = ejemplo\,(3)$ para el Ej. 12; en la variable b externa de la función solo se guardarán los datos de la variable b interna de la función ***ejemplo***; pero la variable b externa se puede definir con otra variable distinta sin afectar la variable b interna.

Es preciso aclarar al lector que como se explicó anteriormente, en las funciones todas las variables que se utilicen son internas; por tal motivo, no pueden ser visualizadas en el ***Workspace***; siendo esta una de las características más importantes al trabajar con funciones, debido a que se puede hacer uso de las mismas variables en funciones con nombre distinto y estas no crearan conflicto en el algoritmo; en el caso hipotético donde ***a*** siendo un vector doble en la función ***ejemplo*** en la función ***ejemplo1*** puede ser ***a*** una matriz, un escalar o una variable tipo carácter; sin ocasionar un conflicto por el tipo de variable utilizado.

3.3.1.7 Funciones del dominio de Matlab.

En esta sección se realiza una breve descripción de las funciones utilizadas en los distintos algoritmos propuestos en este proyecto que pertenecen a Matlab Tabla 7, es decir son utilizadas dentro de los algoritmos, pero no fueron creadas por el autor; sino que, fueron tomadas de las librerías de Matlab para la realización de distintos procesos.

La función ***cd*** permite ubicar la dirección actual de Matlab en el $Current\ Directory$ a una ubicación especial que ingrese el programador. Es importante aclarar que la dirección debe ser ingresada como un conjunto de caracteres para que la función cd pueda ubicar la dirección asignada, de lo contrario podría ocurrir un error en Matlab.

La función ***ls*** permite leer los nombres de los archivos contenidos en el directorio actual de Matlab y convertirlos a un vector del tipo $char$. Con esta función se debe tener especial cuidado porque no diferencia el tipo de archivo; por lo que, se debe asegurar que la dirección donde va a realizar la lectura de los archivos es correcta.

La función ***load*** lee los datos contenidos dentro de un archivo en este caso de extensión $.txt$. Esta función puede ser declarada o no en una variable dentro del entorno de Matlab por lo que se debe tener especial cuidado cuando se desee recuperar los datos leídos.

La función ***floor*** redondea el número ingresado a su entero inferior más cercano, se debe tener especial cuidado con el tipo de variable que se va a redondear porque se puede incurrir en un error tipológico en Matlab.

La función ***max*** devuelve el valor máximo de un vector, con esta función se debe tener especial cuidado porque solo acepta variables de tipo vector doublé; por lo que, se debe ingresar la variable correcta para no incurrir en errores en Matlab.

Tabla 7. Funciones de Matlab.

Función	Descripción
cd	Ubicar la dirección actual de Matlab.
ls	Lee archivos dentro de una carpeta.
load	Lee datos dentro de un archivo.
floor	Valor entero real por debajo.
max	Máximo de un vector.
min	Mínimo de un vector.
zeros	Crea una matriz o un vector de ceros.
length	Calcula el tamaño de un vector, matriz o estructura.
strcmp	Compara caracteres y devuelve un valor lógico
fopen	Abre un archivo para escribirlo.
fprintf	Escribe datos de salida en un archivo abierto por $fopen$.
fclose	Cierra un archivo abierto por $fopen$.
rceps	Calcula la transformada cepstrum real.
cceps	Calcula la transformada cepstrum compleja.
log	Calcula el logaritmo base 10 de un vector.
abs	Calcula el valor absoluto de un vector
real	Calcula el valor real de un vector.
xlswrite	Crea e imprime datos en Excel.
plot	Graficar datos en 2D.
fft	Calcula la transformada discreta de Fourier.
ifft	Calcula la transformada discreta de Fourier inversa.

Fuente: (Autor).

La función ***min*** devuelve el valor mínimo de un vector, con esta función se debe tener especial cuidado porque solo acepta variables de tipo vector doublé; por lo que, se debe ingresar la variable correcta para no incurrir en errores en Matlab.

La función ***zeros*** crea un vector o una matriz de ceros. Esta función es realmente una matriz a la cual se le deben ingresar el número de filas y columnas, por lo que un vector puede ser de una fila y n columnas o puede ser de m filas y una columna.

La función ***length*** mide la cantidad de filas o columnas que posee un vector, se debe tener especial cuidado con esta función porque devuelve el valor de mayor tamaño; por lo que, se debe especificar si lo que se quiere medir es una fila o una columna para obtener el resultado esperado.

La función ***strcmp*** es un condicional que permite comparar variables de tipo $char$, devolviendo un valor lógico el cual puede ser 1; si son iguales y 0 si son distintas, se debe tener especial cuidado con esta función porque reconoce letras en mayúscula o minúscula; por lo que, puede que no de los resultados esperados si se realiza mal la comparación.

Las funciones con letra inicial $\boldsymbol{f}$ dependen de funciones hermanas por lo que se debe utilizar en las operaciones cada una de sus funciones hermanas. La función $\boldsymbol{fopen}$ abre un archivo para realizar escritura, lectura, entre otras acciones, la función $\boldsymbol{fprinf}$ guarda vectores o matrices independientemente del tipo de variable en un archivo abierto por $\boldsymbol{fopen}$ y la función $\boldsymbol{fclose}$ cierra el archivo abierto para escritura, el archivo debe cerrarse porque puede generar conflictos de no hacerlo, como no poder borrar el archivo no cerrado.

La familia $ceps$ no dependen de las funciones hermanas por lo que calculan la parte real o la parte compleja. La función $\boldsymbol{rceps}$ calcula la transformada cepstrum real como en la ecuación (16) y la función $\boldsymbol{cceps}$ calcula la transformada compleja como en la ecuación (16) pero utilizando el logaritmo complejo.

La función $\boldsymbol{log}$ realiza el logaritmo en base 10 de un vector o un escalar. En muchos casos para crear el logaritmo complejo se le agrega a esta función el ángulo del vector complejo.

La diferencia entre la función $\boldsymbol{abs}$ y la función $\boldsymbol{real}$ es que la función $\boldsymbol{abs}$ calcula el valor absoluto de un vector o un escalar real o imaginario; mientras que, la función $\boldsymbol{real}$ elimina la parte imaginaria de un vector imaginario. La diferencia de aplicar estas dos funciones es que por lo general para tener el valor real positivo de un vector imaginario solo se utiliza la función $\boldsymbol{abs}$, pero esta realiza una aproximación del valor real la cual no es igual al valor de la función $\boldsymbol{real}$. En este proyecto para calcular el valor absoluto de un vector imaginario se utiliza primero la función $\boldsymbol{real}$ y luego la función $\boldsymbol{abs}$ para no modificar el valor real del vector.

La función $\boldsymbol{xlswrite}$ es una función integrada de la aplicación Excel de Microsoft en la cual se puede crear el archivo Excel, se puede nombrar sus libros y escribir datos de salida.

3.3.2. Algoritmos.

En esta sección se profundiza en los algoritmos utilizados para resolver el segundo y tercer objetivo específico planteados en este proyecto, donde se busca caracterizar los registros obtenidos de las muestras, por medio de un algoritmo en Matlab. Entiéndase en este proyecto como algoritmo al conjunto de instrucciones dentro de un archivo .m que puede ser del tipo función o del tipo scripts. El lector después de leer la sección 3.3.1, estará en la capacidad de entender los algoritmos que se presenten en esta sección.

3.3.2.1 Funciones complementarias.

Las funciones complementarias son utilizadas dentro de este proyecto como una herramienta para la realización de diferentes operaciones matemáticas, graficas, de comparación, entre otras; que faciliten el uso de los algoritmos que integran actividades relevantes en este proyecto, como el cálculo de la transformada cepstrum y de la distancia euclidiana.

Las funciones complementarias dentro del entorno de este proyecto resultan de gran utilidad para el programador, debido a que facilita en gran medida la utilización de los datos dentro del entorno de la selección inicial de las muestras más adecuadas.

Es importante resaltar que cada una de las funciones descritas en esta sección aportan grandes resultados a los procesos de transformación de la señal, los algoritmos de caracterización y la distancia euclidiana.

También es importante para el lector comprender que la programación en Matlab no es compleja; de hecho es muy recursiva, permitiendo al programador en cualquier momento revisar y corregir cualquier problema que pueda surgir dentro de la programación de las características que se desee, debe realizar en finalidad el programa.

3.3.2.1.1 Función carga.m.

La función $\boldsymbol{carga.m}$ es una de las funciones complementarias más importantes dentro de este proyecto, su importancia radica en la información que recibe el programador al ejecutar su algoritmo, ya que esta función devuelve el tiempo que tarda en cargar los datos que se estén leyendo por medio de porcentaje.

Esta función interacciona con el usuario, es decir, le indica las actividades que está realizando el programa. Al ser una función independiente puede ser utilizada en el entorno de Matlab desde cualquier directorio; por lo que, presenta una ayuda al programador cuando maneje grandes grupos de variables con muchos datos.

El nombre de la función $\boldsymbol{carga.m}$ fue seleccionado por el lugar donde se ubica por lo general dentro de este proyecto; el cual es la lectura de los datos, por lo que se eligió un nombre similar a lectura de datos pero que fuera un poco más general.

La función $\boldsymbol{carga.m}$ es empleada dentro de algoritmos como $\boldsymbol{PrincipalJZ.m}$, $\boldsymbol{prueba.m}$, $\boldsymbol{seleccion.m}$, entre otros; se creó debido a la demora que presentaba realizar distintas operaciones en Matlab y no se podía conocer si el programa estaba trabajando o se había quedado inmerso en un ciclo infinito; por lo que, esta función realiza una interacción con el usuario donde le explica que proceso está realizando, cuanto tiempo en promedio se demora en terminar dicho proceso y que porcentaje de la operación lleva realizada. En la Tabla 8 se presenta el algoritmo $\boldsymbol{carga.m}$ y en la página 65 se explican algunas de sus características.

Tabla 8. Algoritmo $\boldsymbol{carga.m}$.

```
Línea  Código
  1    function carga (i,t,toc,ln)
  2    %% Sección principal
  3    if ln==1
  4        if i==3
  5            tim=((toc*t)/60)+1; tom=floor(tim); tt=60*(tim-tom);
  6            disp ('////////////////////////////////////////////////////////')
  7            disp ('Por favor espere mientras Matlab carga todos los datos
  8                   encontrados en la carpeta')
  9            fprintf ('%s %.f %s %.f %s\n','de trabajo. Esta acción tarda
```

```
                    aproximadamente', tom, 'min', tt,'seg')
            disp ('////////////////////////////////////////////////////////')
        end
        conteo (i,t);
    end
    if ln==2
        if i==1
            tim=((toc*t)/60)+1; tom=floor(tim); tt=60*(tim-tom);
            disp  ('////////////////////////////////////////////////////////')
            disp ('Por favor espere mientras Matlab selecciona y modifica
                   todos los datos encontrados en la')
            fprintf ('%s %.f %s %.f %s\n','carpeta de trabajo. Esta acción
                    tarda aproximadamente', tom,'min', tt, 'seg')
            disp('////////////////////////////////////////////////////////')
        end
        conteo(i,t);
    end
    if ln==3
        if i==1
            tim=((toc*t)/60)+1; tom=floor(tim); tt=60*(tim-tom);
            disp ('////////////////////////////////////////////////////////')
            disp ('Por favor espere mientras Matlab Grafica los datos.')
            fprintf ('%s %.f %s %.f %s\n','Esta acción tarda
                    aproximadamente', tom, 'min', tt, 'seg')
            disp ('////////////////////////////////////////////////////////')
        end
        conteo(i,t);
    end
    end
     %================================================================
    function conteo(i,t)
    %% Sub-función
    if i==floor((5*t)/100); disp('Por favor espere...Carga  5%');end
    if i==floor((10*t)/100); disp('Por favor espere...Carga 10%');end
    if i==floor((15*t)/100); disp('Por favor espere...Carga 15%');end
    if i==floor((20*t)/100); disp('Por favor espere...Carga 20%');end
    if i==floor((25*t)/100); disp('Por favor espere...Carga 25%');end
    if i==floor((30*t)/100); disp('Por favor espere...Carga 30%');end
    if i==floor((35*t)/100); disp('Por favor espere...Carga 35%');end
    if i==floor((40*t)/100); disp('Por favor espere...Carga 40%');end
    if i==floor((45*t)/100); disp('Por favor espere...Carga 45%');end
    if i==floor(t/2);        disp('Por favor espere...Carga 50%');end
    if i==floor((55*t)/100); disp('Por favor espere...Carga 55%');end
    if i==floor((60*t)/100); disp('Por favor espere...Carga 60%');end
    if i==floor((65*t)/100); disp('Por favor espere...Carga 65%');end
    if i==floor((70*t)/100); disp('Por favor espere...Carga 70%');end
    if i==floor((75*t)/100); disp('Por favor espere...Carga 75%');end
    if i==floor((80*t)/100); disp('Por favor espere...Carga 80%');end
    if i==floor((85*t)/100); disp('Por favor espere...Carga 85%');end
    if i==floor((90*t)/100); disp('Por favor espere...Carga 90%');end
    if i==floor((95*t)/100); disp('Por favor espere...Carga 95%');end
    if i==t; disp('Por favor espere...Carga 100%');end
    end
```

Fuente: (Autor)

La función $\boldsymbol{carga.m}$ es del tipo función sin retorno, principal y sub-funciones. Dentro de la sección principal de la línea 2 a la línea 38 Tabla 8, se realizan las operaciones

correspondientes para informar al usuario; que tipo de operación está realizando el programa y cuánto tiempo se demora en terminarla. La sub-función ***conteo*** línea 40 Tabla 8 a medida que el algoritmo $\boldsymbol{carga.m}$ va siendo llamado por otros programas, devuelve el porcentaje de carga que lleva el programa, realizando la operación después de haber iniciado el llamado a la función.

Todas las variables de entrada línea 1 Tabla 8 de $\boldsymbol{carga.m}$ son escalares. Las variables $\boldsymbol{i}$ y $\boldsymbol{t}$ le indican al programa cuando inicia el llamado y cuando termina. La variable $\boldsymbol{toc}$ lleva el tiempo que demora realizar un ciclo dentro de la función que llama a $\boldsymbol{carga.m}$ y la variable $\boldsymbol{ln}$ indica una selección de mensaje para el usuario.

3.3.2.1.2 Función recortar.m.

La función $\boldsymbol{recortar.m}$ fue diseñada como un complementario de la función $\boldsymbol{cepstrumMel.m}$ Tabla 20 en la cual se calculan los coeficientes cepstrum en la escala Mel. Como su nombre lo indica la función $\boldsymbol{recortar.m}$ extrae partes de un vector y las guarda en otro vector de menor tamaño.

Dentro de este proyecto la función $\boldsymbol{recortar.m}$ fue utilizada para examinar cada uno de los coeficientes de la función $\boldsymbol{cepstrumMel.m}$; dentro de los argumentos de entrada se encuentran límites o rangos que se imponen al vector con el fin de extraer el coeficiente.

La función $\boldsymbol{recortar.m}$ principalmente es utilizada dentro de los algoritmos $\boldsymbol{PrincipalJZ.m}$ y $\boldsymbol{prueba.m}$. La necesidad de esta función surge cuando se encontraron los CCM de la sección 3.3.2.4, con el fin de poder analizar coeficiente por coeficiente. En la Tabla 9 se presenta el algoritmo $\boldsymbol{recortar.m}$ y en la página 67 se explican algunos detalles de su funcionamiento.

Tabla 9. Algoritmo $\boldsymbol{recortar.m}$.

```
Línea  Código
  1    function [yy]=recortar(m,y,carpeta,r1,r2,relacion,ref4)
  2    %% Sección pricipal
  3    if strcmp(relacion,'Y')==1
  4    if strcmp(m,'GCABvsDA05g')==1
  5        ref3=['DA05g_';'GCAB._']; %r1=2; r2=6;
  6        [yy]=comparar(y,carpeta,ref3,r1,r2,relacion);
  7    end
  8    if strcmp(m,'GCABvsDA10g')==1
  9        ref3=['DA10g_';'GCAB._']; %r1=2; r2=6;
 10        [yy]=comparar(y,carpeta,ref3,r1,r2,relacion);
 11    end
 12    if strcmp(m,'GCABvsDA15g')==1
 13        ref3=['DA15g_';'GCAB._']; %r1=9; r2=13;
 14        [yy]=comparar(y,carpeta,ref3,r1,r2,relacion);
 15    end
 16    if strcmp(m,'GCABvsDBv1r1')==1
 17        ref3=['DBv1r1';'GCAB._']; %r1=13; r2=16;
 18        [yy]=comparar(y,carpeta,ref3,r1,r2,relacion);
 19    end
 20    if strcmp(m,'GCABvsDBv1r2')==1
 21        ref3=['DBv1r2';'GCAB._']; %r1=9; r2=13;
```

```
22          [yy]=comparar(y,carpeta,ref3,r1,r2,relacion);
23      end
24      if strcmp(m,'GCABvsDBv2r1')==1
25          ref3=['DBv2r1';'GCAB._']; %r1=6; r2=9;
26          [yy]=comparar(y,carpeta,ref3,r1,r2,relacion);
27      end
28      if strcmp(m,'GCABvsDBv2r2')==1
29          ref3=['DBv2r2';'GCAB._']; %r1=6; r2=9;
30          [yy]=comparar(y,carpeta,ref3,r1,r2,relacion);
31      end
32      end
33      if strcmp(relacion,'N')==1
34          %r1=2; r2=6;
35          [yy]=comparar(y,carpeta,ref4,r1,r2,relacion);
36      end
37      end
38      %=================================================
39      function [yy]=comparar(y,carpeta,ref3,r1,r2,relacion)
40      %% Sub-función
41      u=1; su=0; co=1;  se=1; c1=0; c2=0; t1=length(carpeta(:,1));
42      for ct=3:t1
43          if ct==t1; uu=2; else; uu=ct+u; end
44          cy=length(carpeta(ct,:))-8; ref=carpeta(3+su,1:cy);
45          ref2=carpeta(uu,1:cy);
46          if strcmp(ref,ref2)==1
47              co=co+1;
48          else
49              su=ct;
50              if strcmp(ref,ref3(se,:))==1
51                  co2=co-67;
52                  if strcmp(relacion,'N')==1; se=1; else; se=se+1; end
53                  for k=co2:co
54                      c1=c1+1;
55                      for k1=r1:r2
56                          c2=c2+1; yy1(c2)=y(k).y(k1);
57                      end
58                      c2=0; yy(c1).y=yy1;
59                  end
60              end
61              co2=co; co=co+1;
62          end
63      end
64      end
```

Fuente: (Autor).

La función $\boldsymbol{recortar.m}$ es del tipo función principal y sub-funciones. En su sección principal línea 2 a la línea 37 de la Tabla 9, se selecciona el tipo de relación que se va a realizar en el programa. En la sub-función $\boldsymbol{comparar}$ línea 39 Tabla 9, se realiza el corte del coeficiente dependiendo del rango $\boldsymbol{r1}$ y $\boldsymbol{r2}$ ingresado por el programador.

Las variables de entrada línea 1 Tabla 9 de $\boldsymbol{recortar.m}$ son del tipo $char$, vector de $char$, estructura y escalar. Las variables $\boldsymbol{m}$, $\boldsymbol{relacion}$ y $\boldsymbol{ref4}$; son del tipo $char$, se utilizan en distintos condicionales dentro de la sección principal del programa. La variable $\boldsymbol{carpeta}$ es del tipo vector de $char$; contiene los nombres de los archivos encontrados por otro programa.

La variable $\boldsymbol{y}$ es una estructura la cual contiene para este proyecto 544 datos. Las variables $\boldsymbol{r1}$ y $\boldsymbol{r2}$ son del tipo escalar y realizan el recorte del CCM dentro de $\boldsymbol{recortar.m}$; y la variable de salida $\boldsymbol{yy}$ línea 1 de la Tabla 9, es del tipo estructura con un tamaño de 136 muestras.

3.3.2.1.3 Función MaxyMin.m.

La función $\boldsymbol{MaxyMin.m}$ es una abreviación de máximo y mínimo. Esta función calcula el valor máximo y mínimo de una estructura. El valor máximo y mínimo en el manejo de datos es muy importante debido a que permite establecer rangos dentro de un conjunto de archivos o muestras.

La función $\boldsymbol{MaxyMin.m}$ fue diseñada para que pueda ser utilizada en un futuro cercano como una medida de clasificación de un conjunto de archivos. Está función se basa en encontrar los limites o los rangos en los cuales puede estar incluido un fenómeno en máquinas rotátivas.

La función $\boldsymbol{MaxyMin.m}$ es utilizada dentro de los algoritmos $\boldsymbol{transffourier.m}$, $\boldsymbol{transffourier1.m}$, $\boldsymbol{cepstrum.m}$ y $\boldsymbol{cepstrumMel.m}$. La necesidad de esta función surgió cuando fue necesario identificar los rangos máximos y mínimos de los datos con los cuales se estaba trabajando. Aunque, en algunas funciones donde fue implementada su utilidad no es relevante para el programa en general sus aportes más importantes fueron evidentes después de ejecutar $\boldsymbol{selección.m}$. En la Tabla 10 se presenta el algoritmo $\boldsymbol{MaxyMin.m}$ y en la página 69 se explican algunas de sus características.

Tabla 10. Algoritmo $\boldsymbol{MaxyMin.m}$

```
Línea  Código
  1    function [Max,Min]=MaxyMin(y,carpeta)
  2    u=1; su=0; co=1; co2=0; rmax=0; rmin=1; c1=0; t1=length(carpeta(:,1));
  3    for ct=3:t1
  4        if ct==t1;
  5            uu=2;
  6        else
  7            uu=ct+u;
  8        end
  9        cy=length(carpeta(ct,:))-8; ref=carpeta(3+su,1:cy);
 10        ref2=carpeta(uu,1:cy);
 11        if strcmp(ref,ref2)==1
 12            co=co+1;
 13        else
 14            su=ct;
 15            for i=1+co2:co
 16                yy=y(i).y; Max1=max(yy);
 17                if Max1>=rmax; rmax=Max1; end
 18                if Max1<=rmin; rmin=Max1; end
 19            end
 20            c1=c1+1; Max(c1)=rmax; Min(c1)=rmin; rmax=0; rmin=1;
 21            co2=co; co=co+1;
 22        end
 23    end
 24    end
```

Fuente: (Autor).

La función ***MaxyMin.m*** es del tipo función principal; sus variables de entrada línea 1 Tabla 10 son del tipo estructura y vector *char*. La variable ***y*** es del tipo estructura con un tamaño para este proyecto de 544 datos. La variable ***carpeta*** es del tipo vector *char*, contiene los nombres de los archivos encontrados por otro programa; y las variables de salida línea 1 Tabla 10 ***Max*** y ***Min*** son vectores del tipo doublé los cuales regresan los rangos que diferencian a cada grupo de variables dentro del algoritmo.

3.3.2.1.4 Función grafica_5.m

La función ***grafica_5.m*** es una de las funciones complementarias más importantes dentro de este proyecto. Esta función permite graficar conjunto de archivos o muestras contenidos dentro de variables del tipo estructura.

La importancia de esta función radica en el aumento del poder que posee la función *plot* de Matlab, la cual solo permite graficar vectores. Inicialmente esta función solo graficaba variables del tipo estructura de una longitud fija, pero debido a su gran importancia en el manejo de datos fue ampliada a el grafico de variables de longitud variable.

También se adiciono dentro de su código la posibilidad de graficar variables del tipo vector fijo en el eje x debido a que por lo general los datos de esta variable no cambian respecto a todas las muestras; sin embargo se mantuvo una sub-función que incluye la variabilidad del eje x donde la variable de entrada debe ser del tipo estructura.

La función ***grafica_5.m*** es utilizada por los algoritmos ***prueba.m***, ***seleccion.m*** y ***PrincipalJZ.m***. La necesidad de esta función surge como un método grafico de inspección de los diferentes procesos realizados a la señal de entrada. El aporte más significativo dentro de las distintas funciones por la cual se realiza su llamado, radica en que identifica con colores las distintas variables encontradas por el programa y puede graficar matrices del tipo estructura. En la Tabla 11 se presenta el algoritmo desarrollado para esta función y en la página 72 se explican algunas de sus características.

Tabla 11. Algoritmo ***grafica_5.m***.

Línea	Código
1	`function grafica_5(m,f,y,carpeta,coeficiente,relacion,ref3,color4)`
2	`%% Sección principal`
3	`if strcmp(relacion,'Y')==1`
4	`    if strcmp(m,'GCABvsDA05g')==1`
5	`        ref2=['DA05g_';'GCAB._']; color=0; color1=1;`
6	`        if strcmp(coeficiente,'N')==1`
7	`            grafica_GCAB(f,y,carpeta,ref2,color,color1,relacion);`
8	`        end`
9	`        if strcmp(coeficiente,'Y')==1`
10	`            grafica_GCABxcoefi(f,y,carpeta,ref2,color,color1,relacion);`
11	`        end`
12	`    end`
13	`    if strcmp(m,'GCABvsDA10g')==1`
14	`        ref2=['DA10g_';'GCAB._']; color=2; color1=1;`
15	`        if strcmp(coeficiente,'N')==1`
16	`            grafica_GCAB(f,y,carpeta,ref2,color,color1,relacion);`
17	`        end`

```
            if strcmp(coeficiente,'Y')==1
                grafica_GCABxcoefi(f,y,carpeta,ref2,color,color1,relacion);
            end
        end
        if strcmp(m,'GCABvsDA15g')==1
            ref2=['DA15g_';'GCAB._'];color=3; color1=1;
            if strcmp(coeficiente,'N')==1
                grafica_GCAB(f,y,carpeta,ref2,color,color1,relacion);
            end
            if strcmp(coeficiente,'Y')==1
                grafica_GCABxcoefi(f,y,carpeta,ref2,color,color1,relacion);
            end
        end
        if strcmp(m,'GCABvsDBv1r1')==1
            ref2=['DBv1r1';'GCAB._'];color=4; color1=1;
            if strcmp(coeficiente,'N')==1
                grafica_GCAB(f,y,carpeta,ref2,color,color1,relacion);
            end
            if strcmp(coeficiente,'Y')==1
                grafica_GCABxcoefi(f,y,carpeta,ref2,color,color1,relacion);
            end
        end
        if strcmp(m,'GCABvsDBv1r2')==1
            ref2=['DBv1r2';'GCAB._'];color=5; color1=1;
            if strcmp(coeficiente,'N')==1
                grafica_GCAB(f,y,carpeta,ref2,color,color1,relacion);
            end
            if strcmp(coeficiente,'Y')==1
                grafica_GCABxcoefi(f,y,carpeta,ref2,color,color1,relacion);
            end
        end
        if strcmp(m,'GCABvsDBv2r1')==1
            ref2=['DBv2r1';'GCAB._'];color=6; color1=1;
            if strcmp(coeficiente,'N')==1
                grafica_GCAB(f,y,carpeta,ref2,color,color1,relacion);
            end
            if strcmp(coeficiente,'Y')==1
                grafica_GCABxcoefi(f,y,carpeta,ref2,color,color1,relacion);
            end
        end
        if strcmp(m,'GCABvsDBv2r2')==1
            ref2=['DBv2r2';'GCAB._'];color=0; color1=1;
            if strcmp(coeficiente,'N')==1
                grafica_GCAB(f,y,carpeta,ref2,color,color1,relacion);
            end
            if strcmp(coeficiente,'Y')==1
                grafica_GCABxcoefi(f,y,carpeta,ref2,color,color1,relacion);
            end
        end
    end
    if strcmp(relacion,'N')==1
        if strcmp(m,'FI')==1; grafica_funcint(f,y,carpeta); end
        if strcmp(m,'FIS')==1; grafica_funcintstr(f,y,carpeta); end
        grafica_GCABxcoefi(f,y,carpeta,ref3,color4,color4,relacion);
    end
    end
    %=============================================================
    function grafica_GCAB(f,y,carpeta,ref3,color2,color3,relation)
```

```
76    %% Sub-función 1
77    u=1; su=0; co=1; color=color2; se=1; le=[107,98,114,103,99,121,109];
78    t1=length(carpeta(:,1)); figure(1);
79    for ct=3:t1
80        if ct==t1; uu=2; else; uu=ct+u; end
81        cy=length(carpeta(ct,:))-8; ref=carpeta(3+su,1:cy);
82        ref2=carpeta(uu,1:cy);
83        if strcmp(ref,ref2)==1
84            co=co+1;
85        else
86            su=ct;
87            if strcmp(ref,ref3(se,:))==1
88                co2=co-67;
89                if strcmp(relation,'N')==1; se=1; else; se=se+1;  end
90                color=color+u; letr=le(color);
91                for k=co2:co; hold on; plot(f,y(k).y,char(letr)); end
92                color=color3;
93            end
94            co2=co; co=co+1;
95        end
96    end
97    grid
98    end
99    %============================================================
100   function grafica_GCABxcoefi(f,y,carpeta,ref3,color2,color3,relation)
101   %% Sub-función 2
102   u=1; su=0; co=1; color=color2; se=1; se1=0;
103   le=[107,98,114,103,99,121,109]; t1=length(carpeta(:,1)); figure(1);
104   for ct=3:t1
105       if ct==t1; uu=2; else; uu=ct+u; end
106       cy=length(carpeta(ct,:))-8; ref=carpeta(3+su,1:cy);
107       ref2=carpeta(uu,1:cy);
108       if strcmp(ref,ref2)==1
109           co=co+1;
110       else
111           su=ct;
112           if strcmp(ref,ref3(se,:))==1
113               co2=co-67;
114               if strcmp(relation,'N')==1; se=1; else; se=se+1; end
115               color=color+u;  letr=le(color);
116               for k=co2:co
117                   hold on; se1=se1+1; plot(f,y(se1).y,char(letr));
118               end
119               color=color3;
120           end
121           co2=co; co=co+1;
122       end
123   end
124   grid
125   end
126   %============================================================
127   function grafica_funcint(f,y,carpeta)
128   %% Sub-función 3
129   u=1; su=0; co=1; co2=0; color=0; le=[107,98,114,103,99,121,109];
130   t1=length(carpeta(:,1)); figure(1);
131   for ct=3:t1
132       if ct==t1; uu=2; else; uu=ct+u; end
133       cy=length(carpeta(ct,:))-8; ref=carpeta(3+su,1:cy);
```

```
134         ref2=carpeta(uu,1:cy);
135         if strcmp(ref,ref2)==1
136             co=co+1;
137         else
138             su=ct; color=color+u; letr=le(color);
139             for k=1+co2:co; hold on; plot(f,y(k).y,char(letr)); end
140             if color==7
141                 color=0; le=[107,99,121,109,107,98,114,];
142             end
143             co2=co; co=co+1;
144         end
145     end
146     grid
147     end
148     %============================================================
149     function grafica_funcintstr(f,y,carpeta)
150     %% Sub-función 4
151     u=1; su=0; co=1; co2=0; color=0; le=[107,98,114,103,99,121,109];
152     t1=length(carpeta(:,1)); figure(1);
153     for ct=3:t1
154         if ct==t1; uu=2; else; uu=ct+u; end
155         cy=length(carpeta(ct,:))-8;  ref=carpeta(3+su,1:cy);
156         ref2=carpeta(uu,1:cy);
157         if strcmp(ref,ref2)==1
158             co=co+1;
159         else
160             su=ct; color=color+u; letr=le(color);
161             for k=1+co2:co; hold on;  plot(f(k).x,y(k).y,char(letr)); end
162             if color==7; color=0; end;
163             co2=co;  co=co+1;
164         end
165     end
166     grid
167     end
```

Fuente:(Autor).

La función $\boldsymbol{grafica\_5.m}$ es del tipo función sin retorno, principal y sub-funciones. En su sección principal; línea 2 a la línea 73 de la Tabla 11, se realiza una serie de comparaciones por medio de condicionales.

Las sub-funciones $\boldsymbol{grafica\_GCAB}$ línea 75, $\boldsymbol{grafica\_GCABxcoefi}$ línea 100, $\boldsymbol{grafica\_funcint}$ línea 127; y $\boldsymbol{grafica\_funcintstr}$ línea 149 de la Tabla 11, son llamadas por el usuario dependiendo del tipo de grafica que requiera y del tipo de dato ingresado, siendo estas sub-funciones las que realizan los gráficos de los datos ingresados.

Las variables de entrada línea 1 Tabla 11 del algoritmo $\boldsymbol{grafica\_5.m}$ son del tipo $char$, vector de $char$, estructura, vector doublé y escalar. Las variables $\boldsymbol{m}$, $\boldsymbol{coeficiente}$, $\boldsymbol{relacion}$ y $\boldsymbol{ref3}$ son del tipo $char$ y son utilizadas en general en el algoritmo como parte de los condicionales utilizados en la sección principal. La variable $\boldsymbol{y}$ es del tipo estructura con un tamaño de 544 datos.

La variable $\boldsymbol{f}$ puede ser del tipo estructura o del tipo vector doublé dependiendo de los requerimientos del usuario. La variable $carpeta$ es del tipo vector $char$ el cual contiene los

nombres de los archivos encontrados por otro programa; y la variable $color4$ es del tipo escalar la cual define el color para la sub-función ***grafica_GCABxcoefi*** línea 100 de la función $transffourier.m$.

La función ***transfourier.m*** fue diseñada especialmente para archivos que contienen más de una muestra, esto con el fin de ser implementada dentro de la función ***seleccion.m***. Esta función calcula la transformada discreta real de Fourier para cada uno de las sub-muestras encontradas, también realiza un filtrado a la transformada con el fin de poder visualizar las amplitudes incidentes por medio de la función ***grafica_5.m***.

La función ***transffourier.m*** es utilizada principalmente por ***seleccion.m***. Esta función fue diseñada para comprobar las operaciones realizadas dentro de ***seleccion.m*** respecto a la transformada discreta real de Fourier. En la Tabla 12 se presenta el algoritmo de ***transffourier.m*** y en la página 75 se explican algunas de sus características.

Tabla 12. Algoritmo ***transffourier.m***.

```
Línea  Código
  1    function [dat,ytf]= transffourier (y,num,carpeta,porc)
  2    %% Sección principal
  3    for i=1:length(y)
  4        clear mmmax;
  5        for k=1:length(y(i).y)
  6            yy=abs(real(fft(y(i).y(k).y))/num); l=floor(length(yy)/8);
  7            mmax=max(yy(2:l)); mmmax(k)=mmax; com=mmax*0.126; yc=zeros(l,1);
  8            for j=1:l
  9                if j==1; yc(j)=0; end
 10                if j>1
 11                    if com>yy(j); yc(j)=0; end
 12                    if com<=yy(j); yc(j)=yy(j); end
 13                end
 14            end
 15            ytf(i).y(k).y=yc;
 16        end
 17        maximo(i).m=mmmax; carp(i,:)=carpeta(i+2,:);
 18   end
 19      [dat] = selec(ytf,maximo,carpeta,porc);
 20   end
 21   %=============================================================
 22   function [dat] = selec(y1tf,maximo,carp,porc)
 23   %% Sub-función 1
 24   su=0;co=1;u=1; co2=0;
 25   for jk=3:length(carp(:,1))
 26       if jk==length(carp(:,1)); uu=1; else; uu=jk+u; end
 27       cy=length(carp(jk,:))-8; ref=carp(3+su,1:cy); ref2=carp(uu,1:cy);
 28       if strcmp(ref,ref2)==1
 29           co=co+1;
 30       else
 31           su=jk;[rrmax]=MAXIMO(co2,co,maximo); rmin=porc*rrmax;
 32           for i=1+co2:co
 33               vec=zeros(20,1);
 34               for ii=1:length(y1tf(i).y)
 35                   mmaxx=maximo(i).m; xmax=mmaxx(ii);
 36                   if rrmax>=xmax && rmin<=xmax
 37                       ytf(i).y(ii).y=y1tf(i).y(ii).y;
```

```
                    carpet(i,:)=carp(i,:); vec(ii)=ii;
                else
                    ytf(i).y(ii).y=0; doutcarp(i,:)=carp(i,:);
                    dout(i).f(ii).f=y1tf(i).y(ii).y;
                end
            end
            num(i).n=vec;
        end
        co2=co; co=co+1;
    end
end
dat(1).d=ytf; dat(2).d=carpet; dat(3).d=dout; dat(4).d=doutcarp;
dat(5).d=num;
end
%============================================================
function [rrmax]=MAXIMO(co2,co,maximo)
%% Sub-función 2
rrrmax=0; cont=0; cont1=0; cont2=0; cont3=0; cont4=0;
cont5=0; cont6=0; cont7=0; cont8=0; cont9=0;
for k=1+co2:co
    ref=maximo(k).m; rmax=max(ref);
    if rmax>rrrmax; rrrmax=rmax; end
end
ju=rrrmax/10;
for k1=1+co2:co
    reff=maximo(k1).m; Rmax=max(reff);
    if (ju*10)>=Rmax && (ju*9)<=Rmax;  cont=cont+1;    end
    if (ju*9)>=Rmax && (ju*8)<=Rmax;   cont1=cont1+1;  end
    if (ju*8)>=Rmax && (ju*7)<=Rmax;   cont2=cont2+1;  end
    if (ju*7)>=Rmax && (ju*6)<=Rmax;   cont3=cont3+1;  end
    if (ju*6)>=Rmax && (ju*5)<=Rmax;   cont4=cont4+1;  end
    if (ju*5)>=Rmax && (ju*4)<=Rmax;   cont5=cont5+1;  end
    if (ju*4)>=Rmax && (ju*3)<=Rmax;   cont6=cont6+1;  end
    if (ju*3)>=Rmax && (ju*2)<=Rmax;   cont7=cont7+1;  end
    if (ju*2)>=Rmax && (ju*1)<=Rmax;   cont8=cont8+1;  end
    if (ju*1)>=Rmax && 0<=Rmax;        cont9=cont9+1;  end
end
if cont>=cont1 && cont>=cont2 && cont>=cont3 && cont>=cont4 &&
   cont>=cont5 && cont>=cont6 && cont>=cont7 && cont>=cont8 &&
   cont>=cont9;   rrmax=ju*10;
end
if cont1>=cont &&  cont1>=cont2 && cont1>=cont3 && cont1>=cont4 &&
   cont1>=cont5 && cont1>=cont6 && cont1>=cont7 && cont1>=cont8 &&
   cont1>=cont9;   rrmax=ju*9;
end
if cont2>=cont  && cont2>=cont1 && cont2>=cont3 && cont2>=cont4 &&
   cont2>=cont5 && cont2>=cont6 && cont2>=cont7 && cont2>=cont8 &&
   cont2>=cont9;   rrmax=ju*8;
end
if cont3>=cont && cont3>=cont1 && cont3>=cont2 && cont3>=cont4 &&
   cont3>=cont5 && cont3>=cont6 && cont3>=cont7 && cont3>=cont8 &&
   cont3>=cont9;   rrmax=ju*7;
end
if cont4>=cont  && cont4>=cont1 && cont4>=cont2 && cont4>=cont3 &&
   cont4>=cont5 && cont4>=cont6 && cont4>=cont7 && cont4>=cont8 &&
   cont4>=cont9;   rrmax=ju*6;
end
if cont5>=cont  && cont5>=cont1 && cont5>=cont2 && cont5>=cont3 &&
```

```
96        cont5>=cont4 && cont5>=cont6 && cont5>=cont7 && cont5>=cont8 &&
97        cont5>=cont9;   rrmax=ju*5;
98    end
99    if cont6>=cont  && cont6>=cont1 && cont6>=cont2 && cont6>=cont3 &&
100       cont6>=cont4 && cont6>=cont5 && cont6>=cont7 && cont6>=cont8 &&
101       cont6>=cont9;   rrmax=ju*4;
102   end
103   if cont7>=cont  && cont7>=cont1 && cont7>=cont2 && cont7>=cont3 &&
104       cont7>=cont4 && cont7>=cont5 && cont7>=cont6 && cont7>=cont8 &&
105       cont7>=cont9;   rrmax=ju*3;
106   end
107   if cont8>=cont  && cont8>=cont1 && cont8>=cont2 && cont8>=cont3 &&
108       cont8>=cont4 && cont8>=cont5 && cont8>=cont6 && cont8>=cont7 &&
109       cont8>=cont9;   rrmax=ju*2;
110   end
111   if cont9>=cont  && cont9>=cont1 && cont9>=cont2 && cont9>=cont3 &&
112       cont9>=cont4 && cont9>=cont5 && cont9>=cont6 && cont9>=cont7 &&
113       cont9>=cont8;   rrmax=ju*1;
114   end
115   end
```

Fuente: (Autor).

La función $\boldsymbol{transffourier.m}$ es del tipo función principal y sub-funciones. En su sección principal de la línea 2 a la línea 20 de la Tabla 12, se realiza la lectura de los datos y la transformada discreta real de Fourier. También realiza una ventana de la señal que es ocho veces menor al tamaño original del archivo leído y realiza un llamado a una sub-función llamada $\boldsymbol{selec}$ línea 22 Tabla 12. Las sub-funciones de $\boldsymbol{transffourier.m}$ son $\boldsymbol{selec}$ y $\boldsymbol{MAXIMO}$.

La sub-función $\boldsymbol{selec}$ realiza una selección de los archivos encontrados respecto a la frecuencia fundamental de la transformada discreta real de Fourier, pero antes de realizar la selección llama a la sub-función $\boldsymbol{MAXIMO}$ línea 53 Tabla 12.

La sub-función $\boldsymbol{MAXIMO}$ realiza por medio de una serie de condicionales una revisión de todos los archivos correspondientes a un mismo grupo con el fin de encontrar en que rango se repiten en mayor número las amplitudes de la frecuencia fundamental.

Las variables de entrada de $\boldsymbol{transffourier.m}$ línea 1 Tabla 12, son del tipo estructura, vector de $char$ y escalar. La variable $\boldsymbol{y}$ es del tipo estructura, contiene todos los archivos leídos por otra función. La variable $\boldsymbol{carpeta}$ es del tipo vector de $char$, contiene todos los nombres de los archivos leídos por otra función. Las variables $\boldsymbol{num}$ y $\boldsymbol{porc}$ son del tipo escalar; $\boldsymbol{num}$ es el periodo de muestreo y $\boldsymbol{porc}$ es una precisión en la selección la cual puede variar de 0 a 0.99.

Las variables de salida $\boldsymbol{dat}$ y $\boldsymbol{ytf}$ línea 1 Tabla 12, son del tipo estructura. La variable $\boldsymbol{ytf}$ contiene la transformada discreta real de Fourier con un filtro pasa bajo y la variable $\boldsymbol{dat}$ contiene la selección de los archivos, los archivos eliminados, la posición donde se localiza la muestra seleccionada dentro de los archivos leídos y el nombre del archivo.

3.3.2.1.5 Función transffourier1.m.

La función $\boldsymbol{transffourire1.m}$ calcula la transformada de Fourier discreta real para variables de tipo estructura con una sola profundidad, siendo esta la gran diferencia con la función $\boldsymbol{transffourire.m}$ la cual maneja doble profundidad en la variable del tipo estructura.

La función $\boldsymbol{transffourire1.m}$ es utilizada principalmente por $\boldsymbol{prueba.m}$. Esta función fue diseñada como un punto de verificación después de ejecutar la función $\boldsymbol{seleccion.m}$; cabe resaltar al lector que $\boldsymbol{seleccion.m}$ realiza un llamado interno de la función $\boldsymbol{Seleccionmuestras.m}$; entonces, la función $\boldsymbol{transffourire1.m}$ permite verificar los archivos que se guardaron por medio de $\boldsymbol{Seleccionmuestras.m}$. En la Tabla 13 se presenta el algoritmo de $\boldsymbol{transffourire1.m}$ y se explican algunas de sus características.

Tabla 13. Algoritmo $\boldsymbol{transffourire1.m}$.

```
Línea  Código
  1    function [ytf,xx,Maxtf,Mintf]= transffourier1 (x,y,num,carpeta)
  2    for i=1:length(y)
  3        yy=abs(real(fft(y(i).y))/num);
  4        l=floor(length(yy)/8);
  5        yc=zeros(l,1);
  6        xx=zeros(l,1);
  7        for j=1:l
  8            xx(j)=x(1).x(j);
  9            if j==1; yc(j)=0; end
 10            if j>1
 11                yc(j)=yy(j);
 12            end
 13        end
 14        ytf(i).y=yc;
 15   end
 16   [Maxtf,Mintf]=MaxyMin(ytf,carpeta);
 17   end
```

Fuente: (Autor).

La función $\boldsymbol{transffourire1.m}$ es del tipo función principal. En su algoritmo realiza la transformada discreta real de Fourier, realizando una ventana a la señal que es ocho veces menor al tamaño de los archivos ingresados, pero sin realizar un filtro pasa bajo; dentro de su algoritmo se realiza un llamado a la función $\boldsymbol{MaxyMin.m}$ de la sección 3.3.2.1.3.

Las variables de entrada de $\boldsymbol{transffourire1.m}$ línea 1 Tabla 13, son del tipo estructura, escalar y vector $char$. Las variables $\boldsymbol{x}$ y $\boldsymbol{y}$ son del tipo estructura, estas variables contienen los valores de los archivos leídos por otra función. La variable $\boldsymbol{num}$ es del tipo escalar y contiene el periodo de muestreo de los archivos. La variable $\boldsymbol{carpeta}$ es del tipo vector $char$; contiene los nombres de todos los archivos leídos por otro programa.

Las variables de salida de $\boldsymbol{transffourire1.m}$ línea 1 Tabla 13, son del tipo estructura y vector doublé. La variable $\boldsymbol{ytf}$ es del tipo estructura, contiene las transformadas discretas reales de Fourier. Las variables $\boldsymbol{xx}$, $\boldsymbol{Maxtf}$ y $\boldsymbol{Mintf}$ son del tipo vector doublé los cuales contienen rangos dependiendo de la variable $\boldsymbol{ytf}$.

3.3.2.2 Algoritmo Selección y selección de muestras.

El algoritmo selección es una función llamada $\boldsymbol{seleccion.m}$ la cual realiza un barrido dentro de las muestras con el fin de buscar y seleccionar los datos que se encuentren más agrupados de acuerdo a un porcentaje que debe ingresar el programador.

La importancia de este algoritmo dentro de la programación en Matlab para este proyecto resulta de la afirmación ¡Son todas las muestras tomadas para el grupo de control, el grupo de desbalanceo y el grupo de desalineación iguales; y es posible utilizarlas todas!

Pues bien dentro de esta sección se le explicara al lector por que no todas las muestras tomadas sirven para el análisis en este proyecto. Como preliminar se le indica al lector que la variable de influencia es la dispersión de los datos.

Para empezar con el algoritmo se tuvo que establecer un precedente de referencia. En este caso la referencia fue la transformada real de Fourier la cual permite ver las amplitudes de la señal correspondiente al tiempo; en frecuencia.

Debido a que con la transformada de Fourier es posible ver los picos de la señal a la frecuencia de trabajo de la máquina, se presume también que es posible comparar si las señales tomadas como muestra están agrupadas o no en el pico que identifique a la prueba que se vaya a realizar.

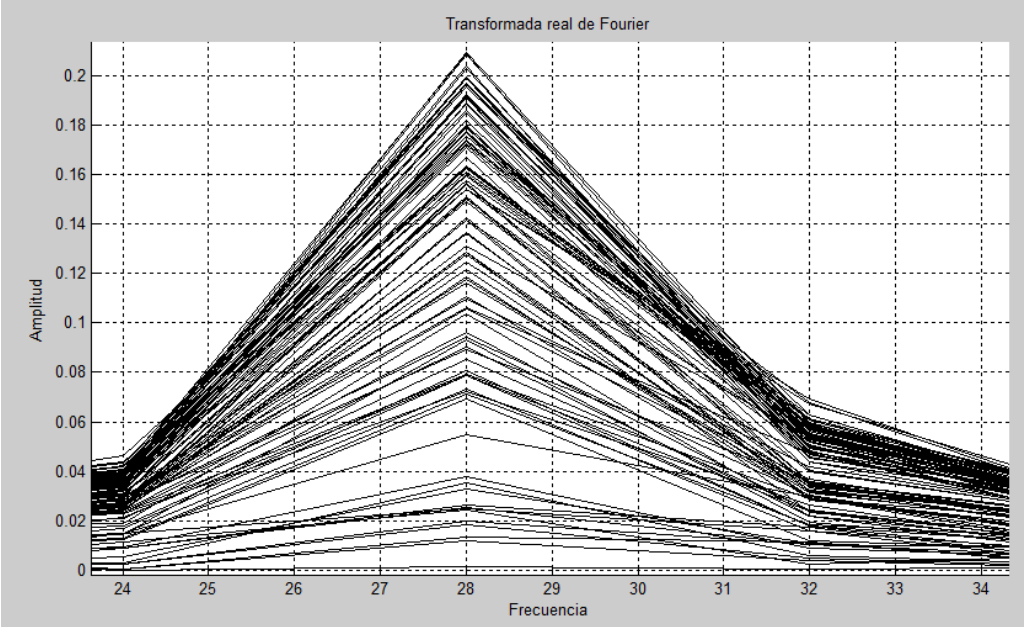
En las Figura 17 y Figura 18, se graficaron 102 muestras para el grupo de referencia. Para realizar este grafico se utilizó el script $\boldsymbol{prueba.m}$ donde se realiza el llamado a las funciones $\boldsymbol{transffourier1.m}$ sección 3.3.2.1.5 y $\boldsymbol{grafica\_5.m}$ sección 3.3.2.1.4. Después de graficar en la Figura 17, se puede establecer porque todas las muestras correspondientes a un mismo grupo no son iguales y no se pueden utilizar.

Los datos correspondientes a la variable GCAB graficados en la Figura 17, en la cual se pueden visualizar que las muestras graficadas son dispersas y aparecen contenidas dentro de un rango con amplitud máxima de 0.2092 y una amplitud mínima muy a aproximada a cero para la frecuencia de 28 Hz. También cabe resaltar de la Figura 17 que existe un agrupamiento de los datos entre la amplitud máxima y la amplitud de 0.14 para la variable GCAB.

Después del análisis realizado a la Figura 17 se puede concluir que dentro de los archivos del grupo de referencia, se debe realizar una selección de los archivos que se encuentren más agrupados; con el fin de establecer un rango de repetición de los archivos, el cual representa la forma de cada grupo.

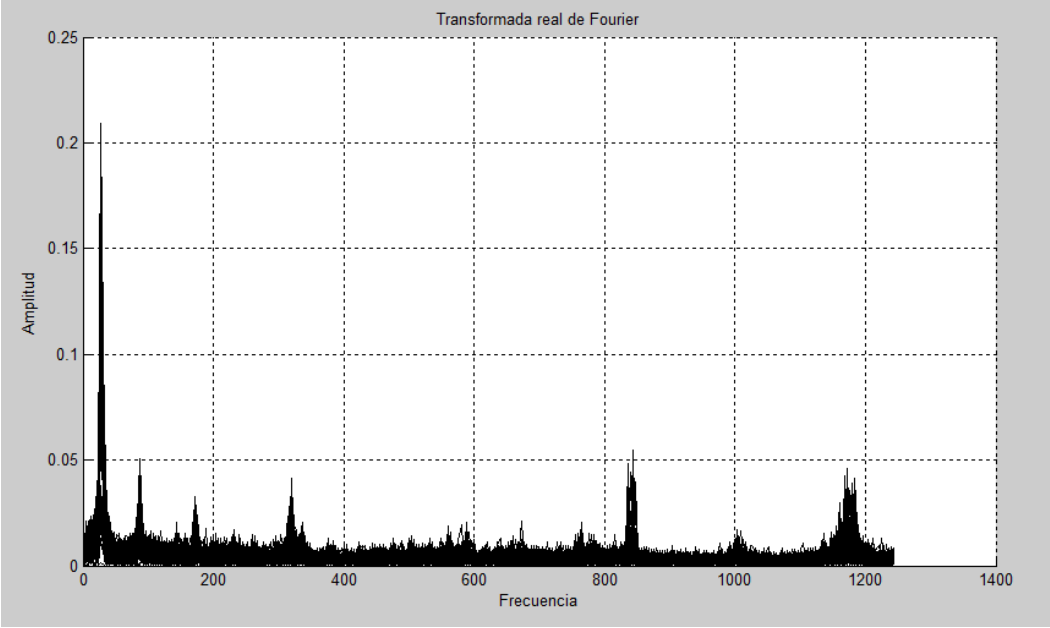
En la gráfica de la Figura 18 no se puede visualizar algún parámetro que pueda incidir en la decisión para la selección de muestras; por lo que, para los grupos de fallas solo se compartirá con el lector las figuras para 28 Hz, que es el punto específico que muestra la frecuencia fundamental del banco de vibraciones.

Figura 17. Transformada real de Fourier, muestras grupo GCAB para 28 Hz



Fuente: (Autor).

Figura 18. Transformada de Fourier para el grupo GCAB para 1200 Hz.



Fuente: (Autor).

La importancia del algoritmo de selección radica en su capacidad para buscar datos que se encuentren ubicados dentro de un rango posible en la amplitud de la frecuencia fundamental, el cual se repita con mayor frecuencia que en otros rangos.

Es posible la selección de datos por medio de un reconocimiento de estos mismos. El programa reconoce la amplitud máxima en la frecuencia fundamental por medio de distintos condicionales, los cuales revisan si a medida que se va realizando la lectura de las muestras existe una muestra con mayor amplitud que la previamente guardada.

Lo anterior con el fin de que el programa reconozca la amplitud máxima de la señal para las distintas muestras tomadas; después de realizar este proceso el programa pasa las distintas muestras a un proceso de selección.

Al realizar este proceso, el programa pasa a una segunda etapa la cual es reconocer en una cuadricula de 10 rangos, donde se agrupan las muestras tomadas en mayor medida. Después de encontrar el rango en el cual se repiten las muestras, el algoritmo devuelve el rango con sus respectivas muestras seleccionadas.

El algoritmo de selección no solo realiza la selección de las muestras, también sobrescribe las muestras, con el fin de que el programador obtenga al final de ejecutar el programa; solo las muestras que le sirven, para un factor de precisión que este mismo ingresa en el programa.

Teniendo en cuenta lo anterior la pertinencia del algoritmo de selección para este programa es inmensa, de tal forma que sin este algoritmo no se podría llegar a comenzar a trabajar con los datos, para realizarle transformación o medición; debido a que los datos inicialmente son muy dispersos de acuerdo a la referencia tomada respecto a la transformada real de Fourier.

El algoritmo realizado para la selección de datos es un archivo con extensión .m del tipo función, por lo que las variables establecidas van a pertenecer al dominio interno de la función, garantizando con esto que no se vaya a entrar en conflicto con otras variables que puedan existir en el ***Workspace***. En la Tabla 14 se presenta el algoritmo ***seleccion.m*** y en la página 80 se explican algunas de sus características.

Tabla 14. Algoritmo *seleccion.m*.

```
Línea  Código
  1    function [var,ytf,carpeta]=seleccion(num3,porc,PWD)
  2    carpeta=ls(PWD);cont=0; u=size(carpeta(:,1)); t=max(u)-2;
  3    for i=1:t; tic;
  4        texto=carpeta(i+2,:); Dato1=zeros(200000,4); Dato1=load(texto);
  5        ref3=max(size(Dato1));
  6        if ref3>num3
  7            num=num3; cont=1; n=1; num2=num3;
  8            ref=floor(length(Dato1)/num); clear xx yy;
  9            for j=1:ref
 10                Dato=zeros(2500,2); Dato=Dato1(1:num,1);
 11                Dato2=Dato1(n:num2,2);
 12                xx(j).x=Dato(:,1); yy(j).y=Dato2(:,1);
```

```
13                n=n+num; num2=num2+num;
14            end
15            x(i).x=xx;  y(i).y=yy;    ti=toc; carga(i,t,ti,1);
16        else
17            break
18        end
19    end
20    if cont==1
21        [dat,ytf]= transffourier (y,num,carpeta,porc);
22        [var]= Seleccionmuestras (x,y,dat,carpeta,num3);
23    else
24    disp('////////////////////////////////////////////////////////////////')
25        disp('Los Archivos ingresados no pueden ser seleccionados')
26        disp('El periodo muestral es igual o inferior al periodo ingresado ')
27    disp('////////////////////////////////////////////////////////////////')
28    end
29    end
```

Fuente: (Autor).

La función $\boldsymbol{seleccion.m}$ es del tipo función principal. Esta función pide al programador tres argumentos de entrada y retorna tres argumentos de salida. También es de notar que la función hace llamado a tres funciones externas las cuales no son sub-funciones de $\boldsymbol{seleccion.m}$; por el contrario, son funciones totalmente independientes que realizan diferentes tareas, las cuales son llamadas por la sección principal que en este caso pertenece a la función $\boldsymbol{seleccion.m}$.

Los argumentos de entrada línea 1 Tabla 14 del algoritmo $\boldsymbol{seleccion.m}$ como se mencionó son tres $\boldsymbol{num3}$, $\boldsymbol{porc}$ y $\boldsymbol{PWD}$. La variable $\boldsymbol{num3}$ representa el periodo de muestreo de los datos a seleccionar por la función; es decir, es una entra del tipo real entero. La variable $\boldsymbol{porc}$ representa la precisión con la cual se desea realizar la selección, la cual varía en porcentaje de 0 a 100 y como variable de entrada, varía de 0 a 0.99 es decir la variable es del tipo real decimal y $\boldsymbol{PWD}$ representa la dirección en la cual se encuentran los paquetes de muestras, es decir la variable es del tipo $char$.

Los argumentos de salida línea 1 Tabla 14 del algoritmo $\boldsymbol{seleccion.m}$ son $\boldsymbol{var}$, $\boldsymbol{ytf}$ y $\boldsymbol{carpeta}$. La variable $\boldsymbol{var}$ es una matriz de $m\,x\,n$ con un número indeterminado de columnas pero con un número determinado de filas el cual pertenece a $\boldsymbol{num3}$. Esta matriz devuelve todos los datos seleccionados que fueron modificados en las carpetas de archivo $.txt$ las cuales fueron dispuestas por el usuario en una carpeta perteneciente al dominio de Matlab variable $\boldsymbol{PWD}$.

La variable $\boldsymbol{ytf}$ devuelve una estructura del tamaño de los datos colocados en la carpeta de Matlab. La estructura de retorno contiene las transformadas de Fourier de las variables seleccionadas con una modificación de filtro pasa bajo devolviendo realmente las amplitudes características de los datos.

La variable $\boldsymbol{carpeta}$ es un vector tipo $char$ de una columna y de varias filas como datos se encuentren en la carpeta de dominio de Matlab más dos filas propias de la función que realiza esta acción.

Las funciones de llamada en la función principal del algoritmo ***seleccion.m*** pertenecientes al autor son; ***carga***, ***transffourier*** y ***Seleccionmuestras***. Las funciones ***carga*** y ***transfourier*** se explicaron en la sección 3.3.2.1. El llamado de la función ***carga*** dentro del ciclo *For* permite al programa interactuar con el usuario con el fin de que este conozca un tiempo aproximado de la lectura de los datos ingresados.

Revisando el algoritmo de ***seleccion.m*** Tabla 14, se puede llegar a la conclusión que este algoritmo solo lee los datos, los guarda en variables tipo estructura y termina su proceso interno. El lector se preguntará donde realiza el algoritmo el resto de operaciones que debería realizar. La respuesta es muy sencilla, el algoritmo por lo general no debe hacer más de una acción compleja, debido a que podría demorar más de lo apropiado; por lo que, se llama a la función independiente *Seleccionmuestras.m* para que realice las actividades de selección.

Es importante aclarar que las funciones ***seleccion.m*** y ***Seleccionmuestras.m*** no pueden trabajar por separado, debido a que la primera función lee datos de un grupo de archivos pero la segunda función no; por lo que son mutuamente dependientes. En la Tabla 15 se presenta el algoritmo ***Seleccionmuestras.m*** el cual depende directamente de la función ***transffourier.m*** y en la página 82 se explican algunas de sus características.

Tabla 15. Algoritmo *Seleccionmuestras.m*.

```
Línea  Código
  1    function [var,carp1]= Seleccionmuestras (x,y,dat,carpeta,num)
  2    %% Sección principal
  3    l=2; j=1; ct3=1; u=1; su=0; co=1; co2=0; var=zeros(num,11);
  4    for ct=3:length(carpeta(:,1))
  5        if ct==length(carpeta(:,1));  uu=2; else; uu=ct+u;  end
  6        cy=length(carpeta(ct,:))-8;
  7        ref=carpeta(3+su,1:cy); ref2=carpeta(uu,1:cy);
  8        if strcmp(ref,ref2)==1
  9            co=co+1;
 10        else
 11            su=ct;
 12            for ct2=1+co2:co
 13                if ct3<=co
 14                dato2=dat(5).d(ct2).n;
 15                if ct2>=ct3;
 16                vx(ct2).x=zeros(num,1); vy(ct2).y=zeros(num,1);
 17                end
 18                for ct4=1:length(y(ct2).y)
 19                    if ct3<=co
 20                    if dato2(ct4)>0
 21                        vx(ct3).x=x(ct2).x(ct4).x;
 22                        vy(ct3).y=y(ct2).y(ct4).y; ct3=ct3+1;
 23                    end
 24                    else
 25                        break
 26                    end
 27                end
 28                else
 29                    break
 30                end
 31            end
```

```
32              co2=co; ct3=co+1; co=co+1;
33          end
34      end
35      t1=length(y);
36      for ii=1:t1
37          tic;
38          xx=vx(ii).x; yy=vy(ii).y; reF=max(yy); n=carpeta(ii+2,:);
39          if reF>0;
40              var(:,j)=xx; var(:,l)=yy; escritura(xx,yy,n);
41              nn=carpeta(ii+2,:); j=j+2; l=l+2;
42          else
43              cero(n); j=j+2; l=l+2;
44          end
45          ti=toc; carga(ii,t1,ti,2);
46      end
47      carp1(ii,:)=nn;
48      end
49      %=========================================================
50      function escritura(xx,yy,carpeta)
51      %% Sub-función 1
52      n=carpeta(1,:); yi1=fopen(n,'w');
53      for ij=1:length(xx);
54          c=xx(ij); cy=yy(ij);
55          fprintf(yi1,'%.6f \t %.6f \r\n',c,cy);
56      end
57      fclose(yi1);
58      end
59      %=========================================================
60      function cero(carpeta)
61      %% Sub-función 2
62      n1=carpeta(1,:); yI1=fopen(n1,'w');
63      c1=zeros(num,1); cy1=zeros(num,1);
64      fprintf(yI1,'%.f \t %.f \r\n',c1,cy1);
65      fclose(yI1);
66      end
```

Fuente: (Autor).

La función $\boldsymbol{Seleccionmuestras.m}$ es un archivo con extensión .m el cual es del tipo función principal y sub-funciones sección 3.3.1.6. La función en general pide 4 argumentos de entrada y entrega dos argumentos de salida. Además de la sección principal posee dos funciones que heredan características del algoritmo principal llamados $\boldsymbol{escritura}$ y $\boldsymbol{cero}$.

Los argumentos de entrada línea 1 Tabla 15 de la función son $\boldsymbol{x, y, dat}$ y $\boldsymbol{carpeta}$. Las variables $\boldsymbol{y, x}$ y $\boldsymbol{dat}$ son del tipo estructura. La variable $\boldsymbol{carpeta}$ es un vector tipo $char$. La función $\boldsymbol{escritura}$ junto con la función $\boldsymbol{cero}$ modifican los archivos del directorio actual de Matlab. Las funciones $\boldsymbol{escritura}$ y $\boldsymbol{cero}$ se crearon por separado porque realizan una acción distinta aunque su finalidad sea modificar los archivos.

Las dos funciones se diferencian en que la función $\boldsymbol{escritura}$ guarda los datos seleccionados y la función $\boldsymbol{cero}$ guarda con matriz cero los archivos que no alcanzaron a ser seleccionados debido al porcentaje ingresado por el usuario.

Limitantes de las funciones $\boldsymbol{seleccion.m}$ *y* $\boldsymbol{Seleccionmuestras.m}$.
Las limitaciones de la pareja de algoritmos de selección se debe establecer por el programador; por lo que, se deben generar protecciones para el algoritmo con el fin de que el programa no incurra en algún error matemático, el cual pueda establecer resultados imprecisos o erróneos.

Las primeras limitaciones tienen que ver con los argumentos de entrada; solo se pueden ingresar variables de tipo número entero o decimal, de tipo $char$ o de tipo estructura solicitadas por el algoritmo, cualquier otro dato ingresado genera un error; incluso si las variables ingresadas son del tipo mencionado, pero son ubicadas dentro del llamado de forma errónea; también se deben establecer errores por parte del programa para su protección.

Otra limitante muy importante es que la lectura de datos siempre debe hacerse desde una carpeta abierta; que este en la dirección actual de Matlab, de lo contrario los programas pueden presentar errores tipológicos en Matlab. Este problema puede ser solucionado por el programador, si dentro de las variables de entrada, se establece la dirección donde se encuentran contenidos los archivos que debe analizar el programa.

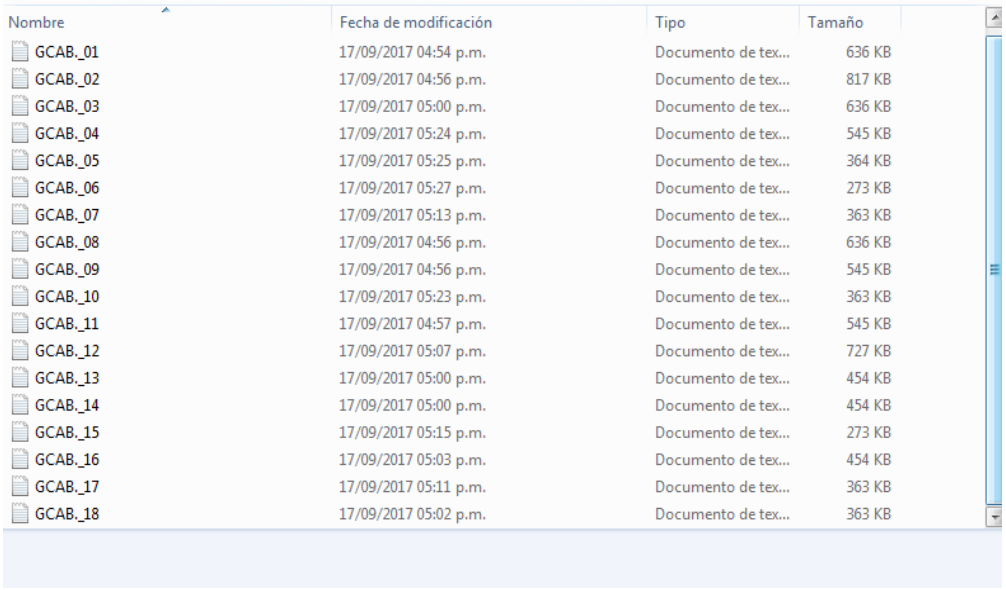
Una limitante importante de estos programas radica en el periodo muestral; si los datos ingresados por el usuario en la carpeta dentro del directorio de Matlab, poseen un periodo muestral distinto; puede que el usuario no obtenga los resultados esperados, debido a que los programas fueron diseñados pensando en que la toma de datos se realiza con el mismo equipo de medición; por lo que, la premisa seria que el periodo muestral permanece invariante en el tiempo, si todos los datos fueron tomados desde el mismo equipo de muestreo.

Ejemplos.
En esta sección se presentan varios ejemplos de lo que realizan las parejas de funciones de selección; desde la lectura de archivos $.txt$ hasta la selección de muestras, con un número de datos entre $25{,}000$ y $27{,}500$ por archivo o muestra. Se debe realizar un análisis con el fin de comprender que ocurre cuando la variable $\boldsymbol{porc}$ (el porcentaje exigido por el usuario), cambia del 90 al 100 por ciento respecto a la selección y la escritura en los archivos de entrada.

Para cada uno de los ejemplos planteados en esta sección se seleccionaron grupos de archivos para la variable GCAB de 18 paquetes de muestras Figura 19, donde la variable de entrada $\boldsymbol{porc}$ será modificada para los valores de $0.9, 0.95$ y 0.99 respectivamente para cada uno de los ejemplos.

En la Figura 19 se puede ver los primeros 18 archivos de la variable GCAB como originalmente se tomaron de Labview. Revisando la columna de tamaño se puede ver que el tamaño de los archivos varía de 300 a 800 KB aproximadamente. Para cada uno de estos archivos el muestreo realizado en Labview tiene un tamaño de muestra de 2,500 datos y un muestreo de 10,000 datos por segundo.

Figura 19. Muestras GCAB originales.

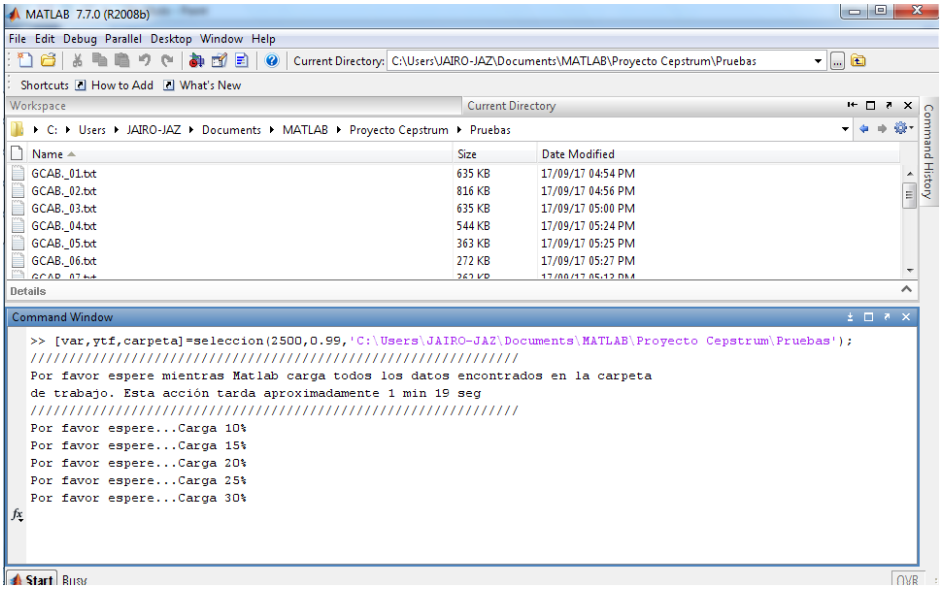


Nombre	Fecha de modificación	Tipo	Tamaño
GCAB._01	17/09/2017 04:54 p.m.	Documento de tex...	636 KB
GCAB._02	17/09/2017 04:56 p.m.	Documento de tex...	817 KB
GCAB._03	17/09/2017 05:00 p.m.	Documento de tex...	636 KB
GCAB._04	17/09/2017 05:24 p.m.	Documento de tex...	545 KB
GCAB._05	17/09/2017 05:25 p.m.	Documento de tex...	364 KB
GCAB._06	17/09/2017 05:27 p.m.	Documento de tex...	273 KB
GCAB._07	17/09/2017 05:13 p.m.	Documento de tex...	363 KB
GCAB._08	17/09/2017 04:56 p.m.	Documento de tex...	636 KB
GCAB._09	17/09/2017 04:56 p.m.	Documento de tex...	545 KB
GCAB._10	17/09/2017 05:23 p.m.	Documento de tex...	363 KB
GCAB._11	17/09/2017 04:57 p.m.	Documento de tex...	545 KB
GCAB._12	17/09/2017 05:07 p.m.	Documento de tex...	727 KB
GCAB._13	17/09/2017 05:00 p.m.	Documento de tex...	454 KB
GCAB._14	17/09/2017 05:00 p.m.	Documento de tex...	454 KB
GCAB._15	17/09/2017 05:15 p.m.	Documento de tex...	273 KB
GCAB._16	17/09/2017 05:03 p.m.	Documento de tex...	454 KB
GCAB._17	17/09/2017 05:11 p.m.	Documento de tex...	363 KB
GCAB._18	17/09/2017 05:02 p.m.	Documento de tex...	363 KB

Fuente: (autor)

En la Figura 20 se puede hacer una revisión del funcionamiento de la función complementaria carga. También en esta figura se puede apreciar cómo se debe llamar a la función ***seleccion.m*** desde el ***Command Windows***.

Figura 20. Llamado desde ***Command Window*** a ***seleccion.m.***



Fuente: (Autor).

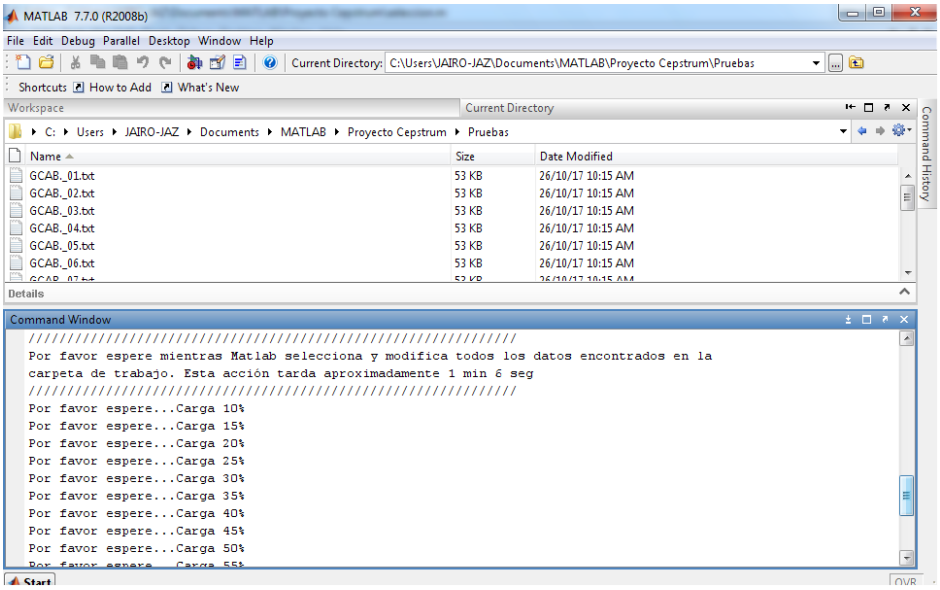
En la parte superior de la Figura 20 se puede ver la dirección del ***current dictory*** la cual pertenece a una carpeta llamada ***pruebas*** que se encuentra en el espacio de trabajo de Matlab. En la parte inferior se puede ver el llamado de la función ***carga*** donde se realiza una interacción con el usuario en la cual le dice cuanto tiempo demora el programa en cargar los datos y comienza a enviarle mensajes en porcentaje de la carga de estos.

En la Figura 21 se puede ver el llamado a la función ***Seleccionmuestras.m*** en el ***Command Windows*** por medio de la función complementaria $carga$. El lector podría preguntarse cómo es posible que se pueda ver el llamado a esta función desde el ***Command Windows***, si esta función pertenece al espacio de trabajo interno de ***seleccion.m***; cabe resaltar al lector que está en lo cierto en pensar que ***Seleccionmuestras.m*** no puede verse en ***Command Windows***.

Sin embargo, cuando el programa realiza la selección de muestras en los algoritmo de selección, para la función ***Seleccionmuestras.m*** significa un gran trabajo, en el procesamiento de los datos; el cual requiere un tiempo prolongado, debido a esto internamente esta función llama a la función ***carga*** para informar al usuario que ha comenzado la modificación de los datos y cuánto tiempo podría demorarse esta acción; en la parte inferior de la Figura 21 es lo que se puede ver que realiza el programa.

Después de revisar la Figura 20 y la Figura 21; se puede concluir que la función complementaria $carga$, realiza apropiadamente su interacción con el usuario; donde se puede ver en estas figuras que le está informado que operación interna se está realizando en el programa, cuanto tiempo en promedio se demora en terminar esa operación y cuanto porcentaje de datos leídos se ha realizado por el programa.

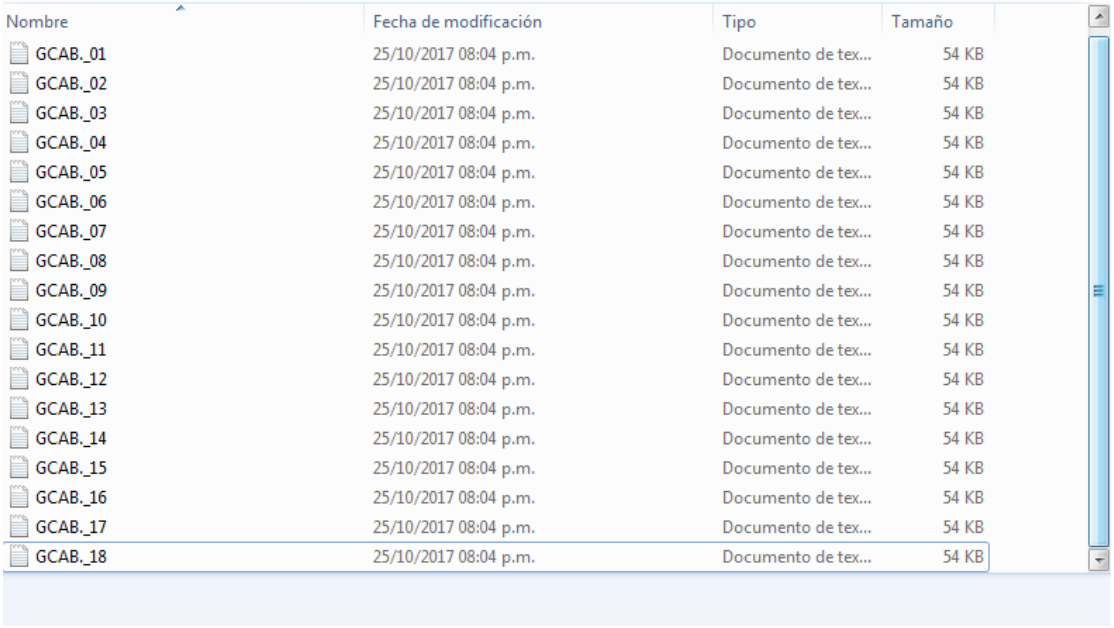
Figura 21. Llamada a la función ***Seleccionmuestras.m.***



Fuente: (Autor).

En la Figura 22 se aplicó para los archivos iniciales de la Figura 19 un $porc = 0.9$. En esta figura se puede ver el proceso terminado de la función $\boldsymbol{seleccion.m}$ en la carpeta de trabajo de Matlab llamada pruebas. Revisando la columna tamaño se puede ver que todos los archivos han cambia de tener un tamaño entre 300 y 800 KB en la Figura 19; a tener un tamaño de aproximadamente 54 KB después de ejecutar los algoritmos de selección. Todos los archivos tienen el mismo tamaño porque el periodo muestral es el mismo para todas las muestra y también porque para un $porc = 0.9$ existen 18 muestras que se pueden seleccionar.

Figura 22. Muestras GCAB después de ejecutar $seleccion.m$ para $porc = 0.9$.



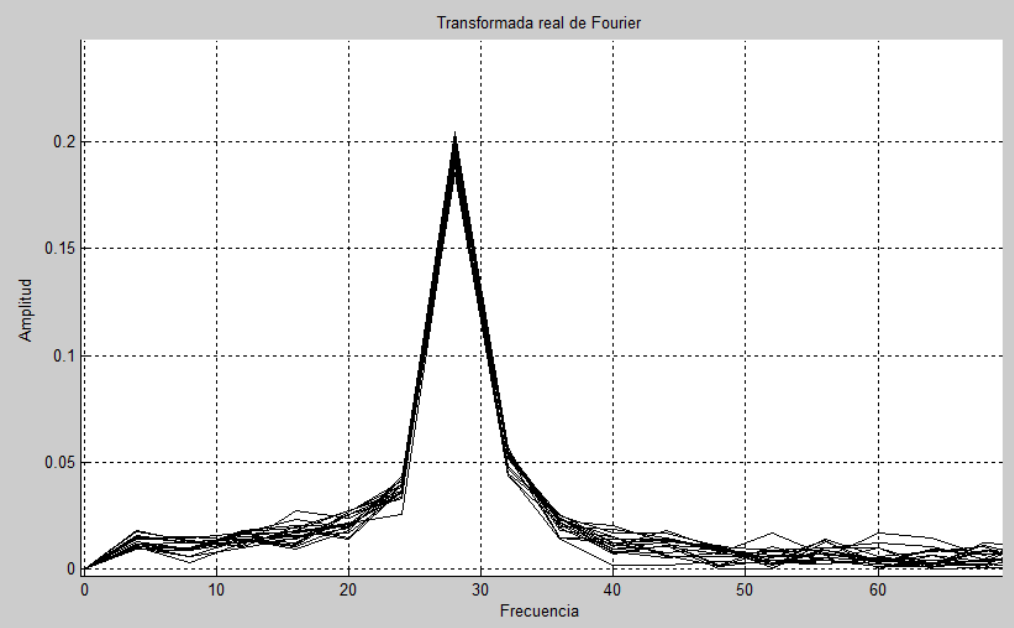
Nombre	Fecha de modificación	Tipo	Tamaño
GCAB._01	25/10/2017 08:04 p.m.	Documento de tex...	54 KB
GCAB._02	25/10/2017 08:04 p.m.	Documento de tex...	54 KB
GCAB._03	25/10/2017 08:04 p.m.	Documento de tex...	54 KB
GCAB._04	25/10/2017 08:04 p.m.	Documento de tex...	54 KB
GCAB._05	25/10/2017 08:04 p.m.	Documento de tex...	54 KB
GCAB._06	25/10/2017 08:04 p.m.	Documento de tex...	54 KB
GCAB._07	25/10/2017 08:04 p.m.	Documento de tex...	54 KB
GCAB._08	25/10/2017 08:04 p.m.	Documento de tex...	54 KB
GCAB._09	25/10/2017 08:04 p.m.	Documento de tex...	54 KB
GCAB._10	25/10/2017 08:04 p.m.	Documento de tex...	54 KB
GCAB._11	25/10/2017 08:04 p.m.	Documento de tex...	54 KB
GCAB._12	25/10/2017 08:04 p.m.	Documento de tex...	54 KB
GCAB._13	25/10/2017 08:04 p.m.	Documento de tex...	54 KB
GCAB._14	25/10/2017 08:04 p.m.	Documento de tex...	54 KB
GCAB._15	25/10/2017 08:04 p.m.	Documento de tex...	54 KB
GCAB._16	25/10/2017 08:04 p.m.	Documento de tex...	54 KB
GCAB._17	25/10/2017 08:04 p.m.	Documento de tex...	54 KB
GCAB._18	25/10/2017 08:04 p.m.	Documento de tex...	54 KB

Fuente: (Autor).

En la Figura 23 revisando detenidamente los resultados obtenidos en la gráfica se puede llegar a la conclusión que los primeros 18 archivos de la variable GCAB están agrupados para un $porc = 0.9$ por lo que la sub-función $\boldsymbol{cero}$ de $\boldsymbol{Seleccionmuestras}$ no intervino dentro del proceso.

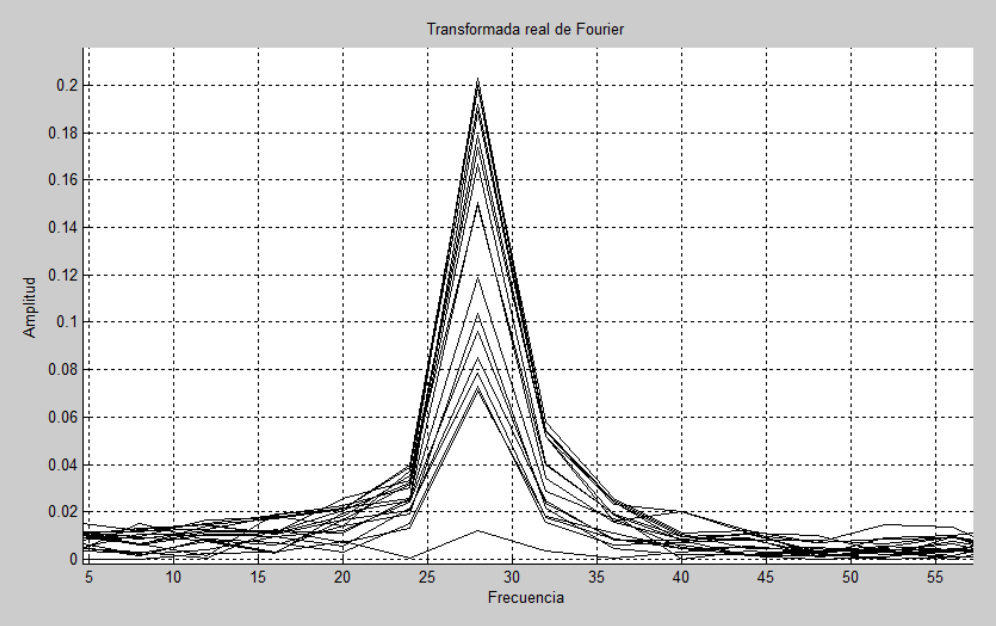
También se pude ver en la Figura 23 de forma gráfica el cambio respecto a la Figura 19 que contiene los datos originales de GCAB. Cuando se ejecuta la función $\boldsymbol{seleccion.m}$ para un $porc = 0.9$ se puede validar la necesidad y la pertinencia del programa con la Figura 23. Para dar claridad al lector respecto a la pertinencia de los algoritmos de selección en la Figura 24 se graficaron los primeros 18 archivos de la variable GCAB sin ejecutar la función $seleccion.m$.

Figura 23. Grafica primeras 18 muestras GCAB ejecutando $seleccion.m$.



Fuente: (Autor).

Figura 24. Grafica primeras 18 muestras GCAB sin ejecutar $seleccion.m$.

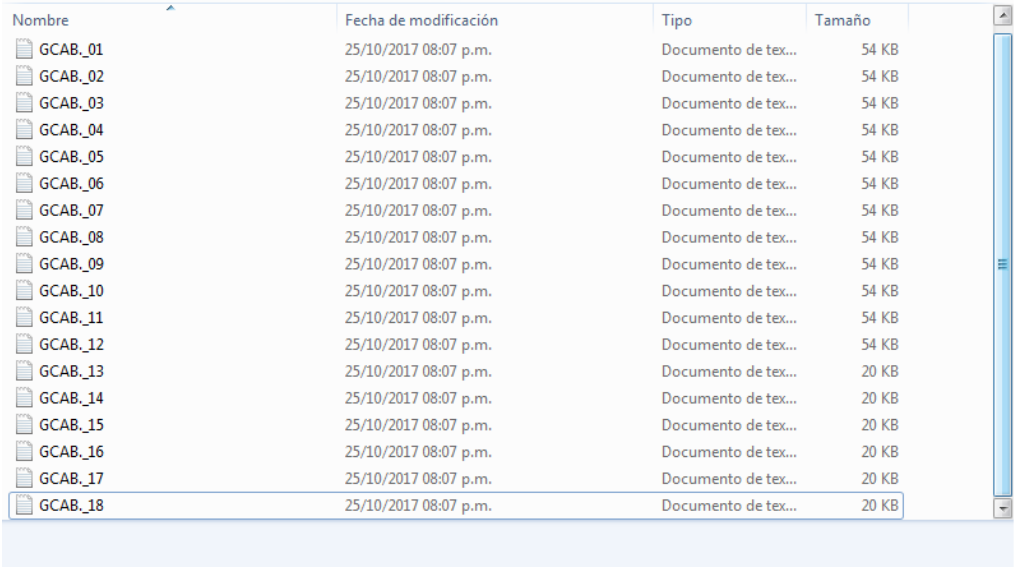


Fuente: (Autor).

Después de revisar la Figura 23 y la Figura 24; se puede concluir con seguridad que el programa $\boldsymbol{seleccion.m}$ permite al usuario seleccionar las muestras de sus archivos. Ahora se desea revisar que ocurre cuando la precisión de selección aumenta para; $porc = 0.95$ y $porc = 0.99$, se comparte con el lector solo los cambios efectuados en los archivos cuando se ha ejecutado $\boldsymbol{seleccion.m}$.

En la Figura 25 se aplicó para los archivos iniciales de la Figura 19 un $porc = 0.95$. En esta figura se puede ver el proceso terminado de la función $\boldsymbol{seleccion.m}$ en la carpeta de trabajo de Matlab llamada pruebas. Revisando la columna tamaño se puede ver que los archivos han cambia de tener un tamaño entre 300 y 800 KB en la Figura 19; a tener un tamaño de aproximadamente 54 KB después de ejecutar los algoritmos de selección para un grupo de 12 muestras; eso quiere decir que 6 muestras con un tamaño aproximado de 20 KB no fueron seleccionadas por los algoritmos. Todos los archivos no tienen el mismo tamaño porque para un $porc = 0.95$ existen solamente 12 muestras que se pueden seleccionar.

Figura 25. Muestras GCAB ejecutando $\boldsymbol{seleccion.m}$ para $porc = 0.95$.

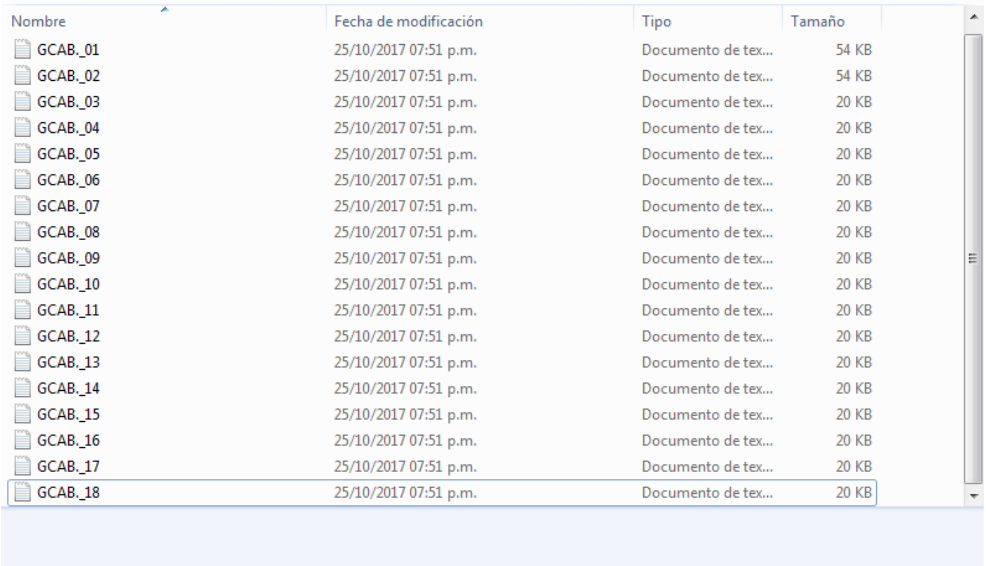


Nombre	Fecha de modificación	Tipo	Tamaño
GCAB._01	25/10/2017 08:07 p.m.	Documento de tex...	54 KB
GCAB._02	25/10/2017 08:07 p.m.	Documento de tex...	54 KB
GCAB._03	25/10/2017 08:07 p.m.	Documento de tex...	54 KB
GCAB._04	25/10/2017 08:07 p.m.	Documento de tex...	54 KB
GCAB._05	25/10/2017 08:07 p.m.	Documento de tex...	54 KB
GCAB._06	25/10/2017 08:07 p.m.	Documento de tex...	54 KB
GCAB._07	25/10/2017 08:07 p.m.	Documento de tex...	54 KB
GCAB._08	25/10/2017 08:07 p.m.	Documento de tex...	54 KB
GCAB._09	25/10/2017 08:07 p.m.	Documento de tex...	54 KB
GCAB._10	25/10/2017 08:07 p.m.	Documento de tex...	54 KB
GCAB._11	25/10/2017 08:07 p.m.	Documento de tex...	54 KB
GCAB._12	25/10/2017 08:07 p.m.	Documento de tex...	54 KB
GCAB._13	25/10/2017 08:07 p.m.	Documento de tex...	20 KB
GCAB._14	25/10/2017 08:07 p.m.	Documento de tex...	20 KB
GCAB._15	25/10/2017 08:07 p.m.	Documento de tex...	20 KB
GCAB._16	25/10/2017 08:07 p.m.	Documento de tex...	20 KB
GCAB._17	25/10/2017 08:07 p.m.	Documento de tex...	20 KB
GCAB._18	25/10/2017 08:07 p.m.	Documento de tex...	20 KB

Fuente: (Autor).

En la Figura 26 se aplicó para los archivos iniciales de la Figura 19 un $porc = 0.99$. En esta figura se puede ver el proceso terminado de la función $\boldsymbol{seleccion.m}$ en la carpeta de trabajo de Matlab llamada pruebas. Revisando la columna tamaño se puede ver que los archivos han cambia de tener un tamaño entre 300 y 800 KB en la Figura 19; a tener un tamaño de aproximadamente 54 KB después de ejecutar los algoritmos de selección para un grupo de 2 muestras; eso quiere decir que 16 muestras con un tamaño aproximado de 20 KB no fueron seleccionadas por los algoritmos. Todos los archivos no tienen el mismo tamaño porque para un $porc = 0.99$ existen solamente 2 muestras que se pueden seleccionar.

Figura 26. Muestras GCAB ejecutando $seleccion.m$ para $porc = 0.99$.



Fuente: (Autor).

Después de revisar y analizar la Figura 19, Figura 22, Figura 25 y Figura 26; se puede concluir que los algoritmos diseñados para la selección y reemplazo de archivos con los datos seleccionados, realiza las operaciones que se plantearon debía realizar en la página 79.

En la sección 4.1 se incluirán los resultados obtenidos al aplicar los algoritmos de selección para cada una de las variables de los grupos de referencia, desalineación y desbalanceo.

3.3.2.3 Algoritmo de Prueba.

El algoritmo prueba fue diseñado como su nombre lo indica para la realización de pruebas para cada uno de los diferentes procesos que se fueron desarrollando en el entorno de Matlab y que tienen que ver directamente con la solución de alguno de los objetivos específicos planteados en este proyecto.

La importancia del algoritmo prueba radica en el desarrollo, mejoramiento e interrelación de las diferentes funciones de este proyecto, donde se fueron moldeando cada una de las funciones a medida que se fue ejecutando este script.

El algoritmo prueba representa para el autor un apoyo muy importante en la programación, aunque no es la función principal de este programa, permite ejecutar de forma oportuna cada una de las nuevas funciones que se fueron diseñando. Una de las ventajas más importantes de este algoritmo es que permite ver cada una de las variables utilizadas en el

código, dentro de ***Workspace***, lo que permite revisar cada uno de los resultados después de ejecutar este algoritmo.

EL algoritmo prueba es un archivo .m del tipo script el cual fue diseñado para realizar los procesos de forma rápida, sin realizar tantos informes al programador. El script se llama ***prueba.m*** y junto con los algoritmos de selección forman parte de un recurso especial diseñado para el programador; debido a que permite realizar tareas complejas de gran trascendencia a la hora de tomar una decisión; con la característica principal que no utiliza muchos recursos del sistema operativo como memoria, video, red, entre otros; que pudieran disminuir el rendimiento del equipo en el cual es ejecutado.

En la Tabla 16 se presenta el algoritmo ***prueba.m*** y en la página 91 se explican algunas de sus utilidades en el entorno de la programación del algoritmo principal.

Tabla 16. Algoritmo tipo script ***prueba.m***.

Línea	Código
1	`clear;clc; carpeta=ls(pwd); u=size(carpeta(:,1)); t=max(u)-2;`
2	`for i=1:t; tic;`
3	`    texto=carpeta(i+2,:);`
4	`    Dato1=zeros(200000,4);  Dato1=load(texto);  num=2500;`
5	`    Dato=zeros(num,2);   Dato=Dato1(1:num,1:2);`
6	`    x(i).x=Dato(:,1);  y(i).y=Dato(:,2);`
7	`    ti=toc; carga(i,t,ti,1);`
8	`end`
9	`%=======================================================`
10	`% transformadas discreta real de Fourier`
11	`[ytf,xx,Maxtf,Mintf]= transffourier1 (x,y,num,carpeta);`
12	`fs=10000; f=floor(fs*(0:((num-1)/8)-1)/num); % frecuencia`
13	`%=======================================================`
14	`% cepstrum`
15	`xx1=xx(1:128);f1=f(1:128);`
16	`[env,tc,x1,Maxenv,Minenv]=cepstrum(x,y,num,carpeta);`
17	`[tc1,x2,Maxtc1,Mintc1]=cepstrumMel(x,y,num,carpeta);`
18	`%=======================================================`
19	`% Seleccion de los coeficientes en escala Mel`
20	`xm(1,:)=x2(2:6); xm1(1,:)=x2(9:13);`
21	`xm2(1,:)=x2(13:16); xm3(1,:)=x2(6:9);`
22	`[yy1]=recortar('GCABvsDA05g',tc1,carpeta,2,6,'Y',0);`
23	`[yy2]=recortar('GCABvsDA10g',tc1,carpeta,2,6,'Y',0);`
24	`[yy3]=recortar('GCABvsDA15g',tc1,carpeta,9,13,'Y',0);`
25	`[yy4]=recortar('GCABvsDBv1r1',tc1,carpeta,13,16,'Y',0);`
26	`[yy5]=recortar('GCABvsDBv1r2',tc1,carpeta,9,13,'Y',0);`
27	`[yy6]=recortar('GCABvsDBv2r1',tc1,carpeta,6,9,'Y',0);`
28	`[yy7]=recortar('GCABvsDBv2r2',tc1,carpeta,6,9,'Y',0);`
29	`%=======================================================`
30	`% Estadisticos`
31	`[EAB1,EAB2,EAB3,EAB4,EAB5,EAB6]=EstadisticosCuefrencia(yy1,'GCAB');`
32	`[E5g1,E5g2,E5g3,E5g4,E5g5,E5g6]=EstadisticosCuefrencia(yy1,'DA05g');`
33	`[E1g1,E1g2,E1g3,E1g4,E1g5,E1g6]=EstadisticosCuefrencia(yy2,'DA10g');`
34	`[E15g1,E15g2,E15g3,E15g4,E15g5,E15g6]=EstadisticosCuefrencia(yy3,'DA15g'`
35	`);`
36	`[Ev1r1m1,Ev1r1m2,Ev1r1m3,Ev1r1m4,Ev1r1m5,Ev1r1m6]=EstadisticosCuefrencia`
37	`(yy4,'DBv1r1');`

```
[Ev1r2m1,Ev1r2m2,Ev1r2m3,Ev1r2m4,Ev1r2m5,Ev1r2m6]=EstadisticosCuefrencia
(yy5,'DBv1r2');
[Ev2r1m1,Ev2r1m2,Ev2r1m3,Ev2r1m4,Ev2r1m5,Ev2r1m6]=EstadisticosCuefrencia
(yy6,'DBv2r1');
[Ev2r2m1,Ev2r2m2,Ev2r2m3,Ev2r2m4,Ev2r2m5,Ev2r2m6]=EstadisticosCuefrencia
(yy7,'DBv2r2');
%=====================================================
% Distancia euclidiana
[CEPDA1]=distanciaeuclidiana (yy1);
[CEPDA2]=distanciaeuclidiana (yy2);
[CEPDA3]=distanciaeuclidiana (yy3);
[CEPD1]=distanciaeuclidiana (yy4);
[CEPD2]=distanciaeuclidiana (yy5);
[CEPD3]=distanciaeuclidiana (yy6);
[CEPD4]=distanciaeuclidiana (yy7);
```

Fuente: (Autor).

El algoritmo $\boldsymbol{prueba.m}$ al ser del tipo script no tiene nombre dentro del archivo donde fue escrito. Las primeras líneas hasta el ciclo For realizan la misma acción que la función $\boldsymbol{seleccion.m}$; guardan los datos contenidos en una variable perteneciente al dominio de Matlab. Después del ciclo For se puede ver que se realiza la llamada a diferentes funciones entre ellas; $\boldsymbol{transffourier1.m}$, $\boldsymbol{grafica\_5.m}$, $\boldsymbol{cepstrum.m}$, $\boldsymbol{cepstrumMel.m}$, $\boldsymbol{recortar.m}$, $\boldsymbol{Estadisticoscuefrencia.m}$, y $\boldsymbol{distanciaeuclidiana.m}$.

Las funciones complementarias $transffourier1.m, grafica\_5.m$ y $recortar.m$ fueron abordadas en la sección 3.3.2.1 y explicadas detalladamente en las secciones 3.3.2.1.5, 3.3.2.1.4 y 3.3.2.1.2.

Los algoritmos correspondientes a la transformada cepstrum; $\boldsymbol{cepstrum.m}$ y $\boldsymbol{cepstrumMel.m}$ son explicados en la sección 3.3.2.4. El algoritmo para la caracterización de las variables pertenecientes a los grupos de referencia, desalineación y desbalanceo; $\boldsymbol{Estadisticoscuefrencia.m}$ es explicado en la sección 3.3.2.6 y el algoritmo para la determinación de la distancia euclidiana; $\boldsymbol{distanciaeuclidiana.m}$ es explicado en la sección 3.3.2.5. Esto con el fin de que el lector pueda dirigirse a cada una de las secciones donde se explica detalladamente cada uno de los códigos propuestos.

Teniendo en cuenta lo anterior cabe aclarar que el script $\boldsymbol{prueba.m}$ realiza la lectura de los datos de cada uno de los archivos correspondiente a cada variable y comienza a pasar información a cada una de las funciones independientes, con el fin de que el procesamiento de la señal no se recargue en un solo archivo .m y de esta forma el procesador de los computadores no colapse por la carga computacional a la cual es sometido después de ejecutar el script $\boldsymbol{prueba.m}$.

3.3.2.4 Algoritmo Transformada cepstrum

En esta sección se explicaran los algoritmos utilizados para resolver el segundo objetivo específico 1.3.2.2 de este proyecto el cual resulta en extraer características de los registros por medio de la transformada cepstrum.

En esta sección se explican las distintas expresiones empleadas para determinar la transformada cepstrum que permita diferenciar de forma gráfica cada una de las variables utilizadas en este proyecto, con el fin de que el lector conozca la forma como se abordó esta problemática.

Es importante aclarar que para poder utilizar alguna de las expresiones de la transformada cepstrum que se van a emplear en esta sección, se requirió inicialmente solucionar el problema de la selección de muestras abordado en la sección 3.3.2.2.

La transformada cepstrum como se explicó en la sección 2.1.10 es definida en muchas ocasiones como el espectro del espectro; esto en si hace referencia a la transformada discreta real de Fourier sección 2.1.8, la cual es el primer espectro de la señal analizada. El segundo espectro se refiere a la transformada discreta real de Fourier inversa; como fue expresado en la ecuación (16). En esta sección se explican dos expresiones derivadas de esta ecuación la transformada cepstrum real y la transformada cepstrum compleja.

3.3.2.4.1 Transformada cepstrum real.

La transformada cepstrum real de una señal en el tiempo se expresa como en la ecuación (16). El término real se refiere exactamente a la transformada discreta de Fourier la cual debe ser real. Es necesario explicar que la transformada discreta de Fourier en Matlab tiene parte real y parte imaginaria, por lo que su resultado es un vector complejo. En la Tabla 17 se presenta el algoritmo para la transformada cepstrum real llamado $\boldsymbol{cepstrum.m}$ y en la página 94 se explican algunas de sus características.

Tabla 17. Algoritmo $\boldsymbol{cepstrum.m}$.

```
Línea  Código
  1    function [env,tc,x1,Maxenv,Minenv]=cepstrum(x,y,num,carpeta)
  2    % Sección principal
  3    for i=1:length(y)
  4        ceps=abs(rceps(y(i).y)/num); L=length(ceps)/4; xx=x(i).x;
  5        [rrmax,yc,x1]=MAXIMO(ceps(1:L),xx(1:L));
  6        envol=abs(real(fft(yc)));
  7        tc(i).y=yc; env(i).y=envol;
  8    end
  9    [Maxenv,Minenv]=MaxyMin(env,carpeta);
 10    end
 11    % ============================================================
 12    function [rrmax1,vec1,x2]=MAXIMO(maximo,x1)
 13    % Sub-función 1
 14    cont=0;  cont1=0; cont2=0; cont3=0; cont4=0;
 15    cont5=0; cont6=0; cont7=0; cont8=0; cont9=0;
 16    rmax=max(maximo); ju=rmax/10;
 17    for k1=1:length(maximo)
 18        Rmax=maximo(k1);
 19        if Rmax>0
 20        if (ju*10)>=Rmax && (ju*9)<=Rmax;  cont=cont+1;    end
 21        if (ju*9)>=Rmax && (ju*8)<=Rmax;   cont1=cont1+1;  end
 22        if (ju*8)>=Rmax && (ju*7)<=Rmax;   cont2=cont2+1;  end
 23        if (ju*7)>=Rmax && (ju*6)<=Rmax;   cont3=cont3+1;  end
```

```
24        if (ju*6)>=Rmax && (ju*5)<=Rmax;   cont4=cont4+1;  end
25        if (ju*5)>=Rmax && (ju*4)<=Rmax;   cont5=cont5+1;  end
26        if (ju*4)>=Rmax && (ju*3)<=Rmax;   cont6=cont6+1;  end
27        if (ju*3)>=Rmax && (ju*2)<=Rmax;   cont7=cont7+1;  end
28        if (ju*2)>=Rmax && (ju*1)<=Rmax;   cont8=cont8+1;  end
29        if (ju*1)>=Rmax && 0<=Rmax;        cont9=cont9+1;  end
30        end
31    end
32    if cont>=cont1 && cont>=cont2 && cont>=cont3 && cont>=cont4 && cont>=cont5
33       && cont>=cont6 && cont>=cont7 && cont>=cont8 && cont>=cont9
34       rrmax1=ju*10; [vec1,x2]=vecmax(maximo,rrmax1,x1);
35    end
36    if cont1>=cont  && cont1>=cont2 && cont1>=cont3 && cont1>=cont4 &&
37       cont1>=cont5 && cont1>=cont6 && cont1>=cont7 && cont1>=cont8 &&
38       cont1>=cont9
39        rrmax1=ju*9; [vec1,x2]=vecmax(maximo,rrmax1,x1);
40    end
41    if cont2>=cont  && cont2>=cont1 && cont2>=cont3 && cont2>=cont4 &&
42       cont2>=cont5 && cont2>=cont6 && cont2>=cont7 && cont2>=cont8 &&
43       cont2>=cont9
44       rrmax1=ju*8; [vec1,x2]=vecmax(maximo,rrmax1,x1);
45    end
46    if cont3>=cont  && cont3>=cont1 && cont3>=cont2 && cont3>=cont4 &&
47       cont3>=cont5 && cont3>=cont6 && cont3>=cont7 && cont3>=cont8 &&
48       cont3>=cont9
49       rrmax1=ju*7; [vec1,x2]=vecmax(maximo,rrmax1,x1);
50    end
51    if cont4>=cont  && cont4>=cont1 && cont4>=cont2 && cont4>=cont3 &&
52       cont4>=cont5 && cont4>=cont6 && cont4>=cont7 && cont4>=cont8 &&
53       cont4>=cont9
54       rrmax1=ju*6; [vec1,x2]=vecmax(maximo,rrmax1,x1);
55    end
56    if cont5>=cont  && cont5>=cont1 && cont5>=cont2 && cont5>=cont3 &&
57       cont5>=cont4 && cont5>=cont6 && cont5>=cont7 && cont5>=cont8 &&
58       cont5>=cont9
59       rrmax1=ju*5; [vec1,x2]=vecmax(maximo,rrmax1,x1);
60    end
61    if cont6>=cont && cont6>=cont1 && cont6>=cont2 && cont6>=cont3 &&
62       cont6>=cont4 && cont6>=cont5 && cont6>=cont7 && cont6>=cont8 &&
63       cont6>=cont9
64       rrmax1=ju*4; [vec1,x2]=vecmax(maximo,rrmax1,x1);
65    end
66    if cont7>=cont && cont7>=cont1 && cont7>=cont2 && cont7>=cont3 &&
67       cont7>=cont4 && cont7>=cont5 && cont7>=cont6 && cont7>=cont8 &&
68       cont7>=cont9
69       rrmax1=ju*3; [vec1,x2]=vecmax(maximo,rrmax1,x1);
70    end
71    if cont8>=cont && cont8>=cont1 && cont8>=cont2 && cont8>=cont3 &&
72       cont8>=cont4 && cont8>=cont5 && cont8>=cont6 && cont8>=cont7 &&
73       cont8>=cont9
74       rrmax1=ju*2; [vec1,x2]=vecmax(maximo,rrmax1,x1);
75    end
76    if cont9>=cont && cont9>=cont1 && cont9>=cont2 && cont9>=cont3 &&
77       cont9>=cont4 && cont9>=cont5 && cont9>=cont6 && cont9>=cont7 &&
78       cont9>=cont8
79       rrmax1=ju*1; [vec1,x2]=vecmax(maximo,rrmax1,x1);
80    end
81    end
```

```
% ============================================================
function [vec2,x2]=vecmax(maximo,rrmax,x1)
% Sub-función 2
for JI=1:(2^7)
    max1=maximo(JI); x2(JI,1)=x1(JI);
    if max1<=rrmax; vec2(JI,1)=0; else; vec2(JI,1)=max1; end
end
end
```

Fuente: (Autor).

La función ***cepstrum.m*** es del tipo función principal y sub-funciones, las cuales son llamadas por la sección principal de la línea 2 a la línea 10 Tabla 17; así como los datos necesarios para ejecutar las sub-funciones son aportados por la sección principal.

La sección principal del algoritmo ***cepstrum.m*** calcula el cepstrum real a partir de la función *rceps* la cual es propia de Matlab sección 3.3.1.7. En esta función se aplica la definición de la ecuación (16). En la Figura 27 se presenta los argumentos que recibe la función *rceps* de Matlab.

Figura 27. Argumentos función *rceps*.

```
[x1]=rceps(y);
[x1]= real(ifft(log(abs(fft(x)))));
```

Fuente: (MathWorks, 2017).

La función *rceps* recibe un solo argumento y devuelve un solo argumento en la Figura 27. El argumento de entrada ***y*** debe ser del tipo vector; por supuesto el argumento que devuelve ***x1*** debe ser del mismo tipo. En la segunda línea de la Figura 27 se puede ver que se aplica internamente dentro de esta función. Esta función llama a las funciones ***fft, abs, log*** y ***ifft*** del dominio de Matlab relacionadas con la transformada discreta de Fourier.

Teniendo claro que realiza la función *rceps* de Matlab se utilizó dentro de la sección principal de ***cepstrum.m*** donde por medio del ciclo *For* se extraen los datos internos de la estructura ***y*** y se guardan temporalmente en el vector *ceps*. Después de realizar esto dentro del ciclo *For* se llama por cada vector extraído en *ceps* la sub-función ***MAXIMO***.

La sub-función ***MAXIMO*** línea 12 Tabla 17 requiere dos argumentos de entrada y devuelve tres argumentos de salida. Los argumentos de entrada son ***maximo*** y ***x1***. La variable ***maximo*** es un vector de la cuarta parte del tamaño del vector *ceps*. La variable ***x1*** es un vector de la cuarta parte de un vector llamado ***xx***, el cual por cada ciclo del *For* guarda los vectores de una estructura llamada ***x*** la cual es un argumento de entrada de la función ***cepstrum.m***.

En general la sub-función ***MAXIMO*** realiza un filtrado paso bajo de los datos ingresados, con el fin de obtener solo el valor pico del cepstrum real. Dentro de la sub-función ***MAXIMO*** por medio de un grupo de condicionales, se realiza la llamada a la segunda sub-función de ***cepstrum.m*** llamada ***vecmax***. La sub-función ***vecmax*** es la que realmente realiza el

filtrado paso bajo; dependiendo de una referencia de paso bajo, la cual fue calculada por la sub-función ***MAXIMO*** donde se calcula la cantidad de datos que más se repiten.

Terminada la acción de ***vecmax***, este devuelve un vector a ***MAXIMO***, el cual a su vez se lo pasa a la sección principal. En la sección principal podemos encontrar este vector con el nombre de ***yc***; es este último el que contiene realmente la transformada real cepstrum.

Después de terminada la acción de la sub-función ***MAXIMO***; al vector ***yc*** se le calcula la envolvente de la transformada cepstrum; la cual es la transformada discreta real de Fourier y se guarda en un vector temporal llamado ***envol***.

Después de terminadas estas acciones, se procede a guardar cada ciclo en dos estructuras llamadas ***tc*** transformada cepstrum y ***env*** envolvente. Una vez terminado el ciclo *For* antes de salir de la sección principal, se llama a la función externa llamada ***MaxyMin.m*** de la sección 3.3.2.1.3.

En la sección de resultados 4.2.1 se evidencia que la transformada cepstrum real Figura 46 y las distintas envolventes Figura 47 y Figura 48; no presentan una diferenciación notoria respecto a cada una de las variables, por lo que es necesario explorar otro tipo de transformada cepstrum.

3.3.2.4.2 Transformada cepstrum compleja.

Es necesario explicar que la transformada cepstrum compleja se desarrolló debido a que el resultado obtenido aplicando la transformada cepstrum real, no proporciono algún coeficiente que permitiera diferenciar en forma gráfica las variables estudiadas.

La transformada cepstrum compleja de una señal en el tiempo se expresa como en la ecuación (16). El término compleja se refiere exactamente a la transformada discreta de Fourier la cual debe ser de tipo compleja. Es necesario explicar que la transformada discreta de Fourier en Matlab tiene parte real y parte imaginaria por lo que su resultado es un vector complejo. En la Tabla 18 se presenta la sección principal del algoritmo ***cepstrumcomplex.m*** y se explican algunas de sus características.

Tabla 18. Sección principal del algoritmo ***cepstrumcomplex.m***.

```
Línea  Código
  1    function [tcC,env1,x1]=cepstrumComplex(x,y,num)
  2    % Sección principal
  3    for i=1:length(y)
  4        ceps=abs(cceps(y(i).y)/num);
  5        L=length(ceps)/4;
  6        xx=x(i).x;
  7        [rrmax,yc,x1]=MAXIMO(ceps(1:L),xx(1:L));
  8        envol=abs(real(fft(yc)));
  9        tcC(i).y=yc;
 10        env1(i).y=envol;
 11    end
 12    end
```

Fuente: (Autor).

La función ***cepstrumcomplex.m*** es del tipo función principal y sub-funciones. Esta función se diferencia con la función ***cepstrum.m*** en su sección principal, donde en la línea 4 se aplica la función propia de Matlab llamada *cceps* sección 3.3.1.7.

La función *cceps* aplica la ecuación (16), pero como el logaritmo en base 10 en Matlab es para números reales, se le realiza una modificación para sumarle al valor del algoritmo el ángulo de la transformada discreta de Fourier.

En la Tabla 19 se presenta parte del algoritmo *cceps* de Matlab, el cual es de tipo libre por lo que se puede ver todo su código ejecutando en Matlab *type* seguido del nombre de la función.

Tabla 19. Algoritmo de Matlab *cceps*.

```
Línea  Código
  1    function [xhat,nd,xhat1] = cceps(x,n)
  2    %CCEPS Complex cepstrum.
  3    %   Author(s): L. Shure, 6-9-88
  4    %   Copyright 1988-2004 The MathWorks, Inc.
  5    %   $Revision: 1.8.4.4 $  $Date: 2007/12/14 15:03:55 $
  6    %   References:
  7    %     [1] Oppenheim, A.V. and Schafer, R.W.  Discrete-Time Signal
  8    %         Processing, Prentice-Hall, 1989.
  9    %     [2] Programs for Digital Signal Processing, IEEE Press,
  10   %         John Wiley & Sons, 1979, algorithm 7.1.
  11   %   [3] Steiglitz, K. and Dickinson, B.  "Computation of the complex
  12   % cepstrum by factorization of the Z-transform," Proc. Int. Conf.
  13   % ASSP, 1977, 723-726
  14   error(nargchk(1,2,nargin,'struct'));
  15   % Check for valid input signal
  16   [errid,errmsg] = chkinput(x);
  17   if ~isempty(errmsg), error(errid,errmsg); end
  18   if nargin < 2
  19       h = fft(x);
  20   else
  21       h = fft(x,n);
  22   end
  23   [ah,nd] = rcunwrap(angle(h));
  24   logh = log(abs(h))+i*ah;
  25   xhat = real(ifft(logh));
  26   end
```

Fuente: (Shure), Copyright 1988-2004 The MathWorks, Inc.

La función *cceps* Tabla 19 de Matlab, realiza la transformada discreta de Fourier de la señal ***x*** línea 19 Tabla 19, también determina su parte positiva, además calcula el logaritmo en base 10; y llamando una sub-función ***rcunwrap*** línea 23 Tabla 19 le suma el ángulo complejo a lo anterior y lo guarda temporalmente en la variable ***logh***. Después calcula la inversa de la transformada discreta de Fourier de ***logh*** línea 25 Tabla 19 y retorna la parte real en la variable ***xhat***.

En la sección de resultados 4.2.2 se evidencia que la transformada cepstrum compleja Figura 47 y su envolvente Figura 48 no presentan una diferenciación notoria respecto a cada una de las variables, por lo que es necesario explorar otro tipo de transformada cepstrum.

3.3.2.4.3 Transformada cepstrum en escala Mel.

Es necesario explicar que la transformada cepstrum en escala Mel se desarrolló debido a que los resultados obtenidos aplicando la transformada cepstrum real y la transformada cepstrum compleja; no proporcionan coeficientes que permitan diferenciar en forma gráfica las variables estudiadas.

La transformada cepstrum en escala Mel también llamada coeficientes cepstrum en la escala Mel (CCM); de una señal en el tiempo se expresa como en la ecuación (18). La escala Mel es una ecuación logarítmica (17) que permite simular la frecuencia de sonido del ser humano sección 2.1.10 página 30. En la Tabla 20 se presenta la sección principal del algoritmo $\boldsymbol{cepstrumMel.m}$ y se explican algunas de sus características.

Tabla 20. Sección principal algoritmo $\boldsymbol{cepstrumMel.m}$.

```
Línea  Código
  1    function [tc,x1,Maxtc,Mintc]=cepstrumMel(x,y,num,carpeta)
  2    % Sección principal
  3    for i=1:length(y)
  4        ceps=abs(real(ifft(2595*log(1+(abs(real(fft((y(i).y))))/num)/700))));
  5        L=length(ceps)/4;
  6        xx=x(i).x;
  7        [rrmax,yc,x1]=MAXIMO(ceps(2:L),xx(2:L));
  8        tc(i).y=yc;
  9    end
 10    [Maxtc,Mintc]=MaxyMin(tc,carpeta);
 11    end
```

Fuente: (Autor).

La función $\boldsymbol{cepstrumMel.m}$ es del tipo función principal y sub-funciones. Esta función se diferencia de la función $cepstrum.m$ en la sección principal, donde en la línea 4 se aplica la ecuación (18) para el cálculo de los CCM.

En la sección de resultados 4.2.3 se evidencia que la transformada cepstrum en la escala Mel representa un 71.43% de diferenciación por el método de visualización de los CCM; pero se puede establecer que este método no es válido para este proyecto, por lo que se debe encontrar diferencias por medio de la distancia euclidiana la cual es un método valido para este proyecto.

3.3.2.5 Algoritmo distancia Euclidiana.

En esta sección se utiliza la distancia euclidiana para correlacionar los grupos de desbalanceo y desalineación con la variable GCAB del grupo de referencia. El objetivo

principal de la distancia euclidiana es encontrar diferencias de cada una de las variables respecto al grupo de referencia teniendo en cuenta los coeficientes de los CCM encontrados en la sección de resultados 4.2.3.

Para determinar la distancia euclidiana se utiliza la ecuación (19) de la sección 2.1.11 la cual se utiliza para determinar la distancia euclidiana para un número n de muestras. A continuación se establecen las siglas que representan la distancia euclidiana respecto a las variables a correlacionar para cada uno de los grupos.

Distancia del grupo desalineado con la variable GCAB.
La distancia del grupo desalineado con el grupo de referencia se realiza respecto a la amplitud de los CCM encontrados en la Tabla 29. La correlación mencionada se nombra de ahora en adelante como CEPDA (Correlación euclidiana de la prueba de desalineación). En la Tabla 21 se puede ver el CEPDA para cada una de las relaciones de las variables del grupo desalineado.

Tabla 21. CEPDA para la relación de variables del grupo desalineado.

Relación	No CEPDA	No CCM
GCABvsDA05g	CEPDA1	1
GCABvsDA10g	CEPDA2	1
GCABvsDA15g	CEPDA3	3

Fuente: (Autor).

Distancia del grupo desbalanceado con la variable GCAB.
La distancia del grupo desbalanceado con el grupo de referencia se realiza respecto a la amplitud de los CCM encontrados en la Tabla 29. La correlación mencionada se nombra de ahora en adelante como CEPD (Correlación euclidiana de la prueba de desbalance). En la Tabla 22 se puede ver el CEPD para cada una de las relaciones de las variables del grupo desbalanceado.

Tabla 22. CEPD para la relación de variables del grupo desbalanceado.

Relación	No CEPD	No CCM
GCABvsDBv1r1	CEPD1	4
GCABvsDBv1r1	CEPD2	3
GCABvsDBv1r1	CEPD3	2
GCABvsDBv1r1	CEPD4	2

Fuente: (Autor).

En la Tabla 23 se presenta el algoritmo utilizado para calcular la distancia euclidiana de los CCM llamado $distanciaeuclidiana.m$ y se explican algunas de sus características.

Tabla 23. Algoritmo $distanciaeuclidiana.m$.

```
Línea  Código
  1    function [eu]=distanciaeuclidiana (tc)
  2    ct=0; ct1=0; ct2=0;
  3    for i=1:(length(tc)/2)
  4        tc11=tc(i).y;
  5        for j2=1:length(tc(i).y)
  6            if tc(i).y(j2)==0; ct1=ct1+1; end
  7        end
  8        tc1(i)=(sum(tc11))/(length(tc11)-ct1);  ct1=0;
  9    end
 10    for j=69:length(tc)
 11        tc12=tc(j).y;
 12        for i3=1:length(tc(j).y)
 13            if tc(j).y(i3)==0; ct2=ct2+1;  end
 14        end
 15        ct=ct+1; tc2(ct)=(sum(tc12))/(length(tc12)-ct2);  ct2=0;
 16    end
 17    t1=length(tc1);
 18    for i2=1:t1;  e(i2)=(tc1(i2)-tc2(i2))^2; end
 19    e1=sum(e); eu=sqrt(e1);
 20    end
```

Fuente: (Autor).

La función $distanciaeuclidiana.m$ calcula la distancia entre los grupos de desbalanceo y desalineación respecto al grupo de referencia GCAB. La variable de entrada tc línea 1 Tabla 23 es una estructura que contiene la relación en parejas de las variables de los grupos desalineado y desbalanceo; respecto a la variable GCAB. La variable eu línea 1 Tabla 23 es un escalar el cual devuelve la distancia entre las dos variables.

Como la variable de entrada tc es una estructura de relación su tamaño es de 136 muestras para este proyecto; por lo que se utilizan dos ciclos For línea 3 y línea 10; para extraer los datos de cada una de las muestras y comenzar a relacionar cada muestra de las variables de falla, respecto a cada muestra de la variables de referencia por medio de un tercer ciclo For línea 18, como se realiza en la ecuación (19).

En la sección de resultados 4.3 se evidencia que la distancia euclidiana representa un 100% de diferenciación según el coeficiente analizado del CCM, por lo que se pueden establecer las características correspondientes a cada coeficiente de los CCM por medio del análisis estadístico.

3.3.2.6 Algoritmos de caracterización.

En esta sección se realiza la caracterización de cada uno de los coeficientes del CCM encontrados en la sección 3.3.2.4.3 y diferenciados en un 100% por medio de la distancia euclidiana sección 3.3.2.5.

El análisis estadístico como lo refiere el estado del arte sección 1.4.3 ha sido empleado para encontrar diferencias entre variables que pueden ser relacionadas. En esta sección se

diseña un algoritmo que permita valorar las distintas variables estadísticas establecidas en la sección 2.1.13.

En la Tabla 24 se presenta el algoritmo ***EstadisticosCuefrencia.m***, el cual realiza todas las operaciones estadísticas de la sección 2.1.13 y en la página 101 se explican algunas de sus características.

Tabla 24. Algoritmo ***EstadisticosCuefrencia.m***.

```
Línea  Código
  1    function [E1,E2,E3,E4,E5,E6]=EstadisticosCuefrencia(yc,m)
  2    % Sección principal
  3    if strcmp(m,'GCAB')==1
  4        [E1,E3,E6]=estadis1(yc,69,136); [E2,E4,E5]=estadis2(yc,69,136,E1);
  5    end
  6    if strcmp(m,'DA05g')==1
  7        [E1,E3,E6]=estadis1(yc,1,68); [E2,E4,E5]=estadis2(yc,1,68,E1);
  8    end
  9    if strcmp(m,'DA10g')==1
 10        [E1,E3,E6]=estadis1(yc,1,68); [E2,E4,E5]=estadis2(yc,1,68,E1);
 11    end
 12    if strcmp(m,'DA15g')==1
 13        [E1,E3,E6]=estadis1(yc,1,68); [E2,E4,E5]=estadis2(yc,1,68,E1);
 14    end
 15    if strcmp(m,'DBv1r1')==1
 16        [E1,E3,E6]=estadis1(yc,1,68); [E2,E4,E5]=estadis2(yc,1,68,E1);
 17    end
 18    if strcmp(m,'DBv1r2')==1
 19        [E1,E3,E6]=estadis1(yc,1,68); [E2,E4,E5]=estadis2(yc,1,68,E1);
 20    end
 21    if strcmp(m,'DBv2r1')==1
 22        [E1,E3,E6]=estadis1(yc,1,68); [E2,E4,E5]=estadis2(yc,1,68,E1);
 23    end
 24    if strcmp(m,'DBv2r2')==1
 25        [E1,E3,E6]=estadis1(yc,1,68); [E2,E4,E5]=estadis2(yc,1,68,E1);
 26    end
 27    end
 28    % ============================================================
 29    function [E1,E3,E6]=estadis1(yc,ref,ref1)
 30    % Sub-función 1
 31    cot=1;
 32    for i=ref:ref1
 33        cot3=0; cot1=length(yc(i).y);
 34        for i2=1:cot1
 35            if yc(i).y(i2)==0; cot3=cot3+1; end
 36        end
 37        yc1(cot)=sum(yc(i).y)/(cot1-cot3);
 38        yc2(cot)=(yc1(cot))^2; cot=cot+1;
 39    end
 40    E1=sum(yc1)/(cot-1); E3=sqrt(sum(yc2)/(cot-1)); E6=E3/E1;
 41    end
 42    % ============================================================
 43    function [E2,E4,E5]=estadis2(yc,ref,ref1,E1)
 44    % Sub-función 2
 45    cot=1;
 46    for i=ref:ref1
```

```
47          cot3=0; cot1=length(yc(i).y);
48          for i2=1:cot1
49              if yc(i).y(i2)==0; cot3=cot3+1; end
50          end
51          yc3(cot)=((sum(yc(i).y)/(cot1-cot3))-E1)^2; yc4(cot)=(yc3(cot))^2;
52          yc5(cot)=((sum(yc(i).y)/(cot1-cot3))-E1)^3; cot=cot+1;
53      end
54      E2=sqrt(sum(yc3)/(cot-2)); E4=sum(yc4)/(cot-2)*(E2)^4;
55      E5=sum(yc5)/(cot-2)*(E2)^3;
56      end
```

Fuente: (Autor).

La función $\boldsymbol{EstadisticosCuefrencia.m}$ es del tipo función principal y sub-funciones. Esta función realiza las operaciones de; valor medio ($E1$) ecuación (22), desviación estándar ($E2$) ecuación (23), raíz media cuadrática ($E3$) ecuación (25), Curtosis ($E4$) ecuación (26), Asimetría ($E5$) ecuación (27) y factor de forma ($E6$) ecuación (28); de la sección 2.1.13.

Los argumentos de entrada línea 1 Tabla 24 son del tipo estructura y $char$. La variable $\boldsymbol{yc}$ corresponde a una estructura con los coeficientes da cada variable y la variable m es un comparador utilizado en la sección principal con el fin de ubicar yc en el coeficiente correcto.

Las sub-funciones $estadis1$ y $estadis2$; calculan los estadísticos para cada variable de los grupos de referencia, desbalanceo y desalineación; los cuales son guardados en las variables $E1, E2, E3, E4, E5$ y $E6$. Estas variables son de salida y son del tipo escalar.

En la sección de resultados 4.4 se evidencian los resultados obtenidos en la caracterización de los coeficientes CCM. Después de realizar todas las etapas propuestas en la metodología Figura 5 de este proyecto, se debe unificar todos los algoritmos diseñados para cada una de las etapas; con el fin de que se realice todo el análisis en un solo algoritmo de Matlab.

3.3.2.7 Algoritmo principal.

En esta sección se realiza la unificación de todos los algoritmos vistos, los cuales permiten realizar cada uno de los objetivos planteados en este proyecto. El algoritmo principal debe realizar la selección de los datos, rescribir las muestras seleccionadas, realizarle a las muestras la transformada cepstrum Mel, calcular la distancia euclidiana de los CCM y determinar las características estadísticas de cada uno de los CCM.

El algoritmo principal debe ser capaz de recibir paquetes de datos, los cuales deben ser ubicados en una carpeta cuya dirección debe ser tenida en cuenta por el algoritmo para comenzar a realizar la selección de los datos.

Además, el algoritmo principal debe estar en la capacidad de solicitar el periodo muestral de los datos, para poder realizar la selección de muestras y también debe solicitar la precisión de selección de las muestras.

El algoritmo principal al realizar la unificación de todos los algoritmos debe tener la capacidad de poder diferenciar los argumentos de entrada, con el fin de no incurrir en errores matemáticos; por lo que, el algoritmo principal debe estar provisto de unas protecciones que permitan protegerlo de algún error humano.

En la Tabla 25 se presenta el algoritmo que cumple con todas las condiciones requeridas para el análisis de los datos, el cual se llama ***PrincipalJZ.m*** y en la página 104 se describen algunas de sus características.

Tabla 25. Algoritmo ***PrincipalJZ.m.***

```
Línea  Código
  1    function [CCM,CCM2,carpeta]=PrincipalJZ(num,porc,direccion)
  2    %% Protecciones
  3    if ischar(direccion)==1
  4        if num>0 && num==floor(num)
  5            if porc<1 && porc>0
  6                [CCM,CCM2,carpeta]=JZ(num,porc,direccion);
  7            else
  8                disp('////////////////////////////////////////////////////')
  9                disp('La variable ingresada en 2 no es un número decimal
 10                      positivo entre 0 y 1')
 11                disp('[...,...,...]=PrincipalJZ(...,2,...)')
 12                disp('Intente ingresar un número decimal positivo asi:')
 13                disp('0.99,0.95,0.9,0.85')
 14                disp('////////////////////////////////////////////////////')
 15            end
 16        else
 17            disp('////////////////////////////////////////////////////')
 18            disp('La variable ingresada en 1 no es un número entero positivo')
 19            disp('[...,...,...]=PrincipalJZ(1,...,...)')
 20            disp('Intente ingresar números enteros asi:')
 21            disp('1000,2000,2500,3000')
 22            disp('////////////////////////////////////////////////////')
 23        end
 24    else
 25        disp('////////////////////////////////////////////////////////')
 26        disp('La variable ingresada en 3 no es una dirección valida')
 27        disp('[...,...,...]=PrincipalJZ(...,...,3)')
 28        disp('Intente ingresar la dirección de la siguiente forma entre comillas')
 29        disp('C:\Users\...\Documents\MATLAB\Proyecto Cepstrum\...')
 30        disp('////////////////////////////////////////////////////////')
 31    end
 32    end
 33    %=========================================================================
 34    function [CCM,CCM2,carpeta]=JZ(num,porc,direccion)
 35    %Sub-función 1
 36    disp('La dirección actual de Matlab es:')
 37    cd; oldFolder = cd (direccion);
 38    if strcmp(oldFolder,cd)==1; else; oldFolder = cd (direccion); end
 39    %% Selección
 40    seleccion(num,porc,oldFolder);
 41    %% Programa Pricipal
 42    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 43    carpeta=ls(oldFolder); u=size(carpeta(:,1)); t=max(u)-2;
```

```
    for i=1:t; tic;
        texto=carpeta(i+2,:);  Dato1=zeros(200000,4);  Dato1=load(texto);
        Dato=zeros(num,2);     Dato=Dato1(1:num,1:2);
        x(i).x=Dato(:,1);  y(i).y=Dato(:,2);
        ti=toc; carga(i,t,ti,1);
    end
    %% Transformada Discreta real de Fourier
    [ytf,xx]= transffourier1 (x,y,num,carpeta);
    fs=10000; f=floor(fs*(0:((num-1)/8)-1)/num);
    %% Transformada Cepstrum en escala Mel
    xx1=xx(1:128);
    [tc1,x2]=cepstrumMel(x,y,num,carpeta);
    %% Selección de los coeficientes en escala Mel
    xm(1,:)=x2(2:6); xm1(1,:)=x2(9:13); xm2(1,:)=x2(13:16); xm3(1,:)=x2(6:9);
    [yy1]=recortar('GCABvsDA05g',tc1,carpeta,2,6,'Y',0);
    [yy2]=recortar('GCABvsDA10g',tc1,carpeta,2,6,'Y',0);
    [yy3]=recortar('GCABvsDA15g',tc1,carpeta,9,13,'Y',0);
    [yy4]=recortar('GCABvsDBv1r1',tc1,carpeta,13,16,'Y',0);
    [yy5]=recortar('GCABvsDBv1r2',tc1,carpeta,9,13,'Y',0);
    [yy6]=recortar('GCABvsDBv2r1',tc1,carpeta,6,9,'Y',0);
    [yy7]=recortar('GCABvsDBv2r2',tc1,carpeta,6,9,'Y',0);
    %% Distancia euclidiana
    [CEPDA1]=distanciaeuclidiana (yy1);
    [CEPDA2]=distanciaeuclidiana (yy2);
    [CEPDA3]=distanciaeuclidiana (yy3);
    [CEPD1]=distanciaeuclidiana (yy4);
    [CEPD2]=distanciaeuclidiana (yy5);
    [CEPD3]=distanciaeuclidiana (yy6);
    [CEPD4]=distanciaeuclidiana (yy7);
    %% Caracterización
    % Estadisticos
    [EAB1,EAB2,EAB3,EAB4,EAB5,EAB6]=EstadisticosCuefrencia(yy1,'GCAB');
    [E5g1,E5g2,E5g3,E5g4,E5g5,E5g6]=EstadisticosCuefrencia(yy1,'DA05g');
    [E1g1,E1g2,E1g3,E1g4,E1g5,E1g6]=EstadisticosCuefrencia(yy2,'DA10g');
    [E15g1,E15g2,E15g3,E15g4,E15g5,E15g6]=EstadisticosCuefrencia(yy3,'DA15g');
    [Ev1r1m1,Ev1r1m2,Ev1r1m3,Ev1r1m4,Ev1r1m5,Ev1r1m6]=EstadisticosCuefrencia
    (yy4,'DBv1r1');
    [Ev1r2m1,Ev1r2m2,Ev1r2m3,Ev1r2m4,Ev1r2m5,Ev1r2m6]=EstadisticosCuefrencia
    (yy5,'DBv1r2');
    [Ev2r1m1,Ev2r1m2,Ev2r1m3,Ev2r1m4,Ev2r1m5,Ev2r1m6]=EstadisticosCuefrencia
    (yy6,'DBv2r1');
    [Ev2r2m1,Ev2r2m2,Ev2r2m3,Ev2r2m4,Ev2r2m5,Ev2r2m6]=EstadisticosCuefrencia
    (yy7,'DBv2r2');
    %% Variables de Salida
    % Variable CCM tipo estructura de Salida
    CCM(1,1).y=tc1;CCM(1,2).y=xx1;CCM(1,3).y='Coeficientes CCM';
    CCM(2,1).y=yy1;CCM(2,2).y=xm;CCM(2,3).y='GCABvsDA05g';
    CCM(3,1).y=yy2;CCM(3,2).y=xm;CCM(3,3).y='GCABvsDA10g';
    CCM(4,1).y=yy3;CCM(4,2).y=xm1;CCM(4,3).y='GCABvsDA15g';
    CCM(5,1).y=yy4;CCM(5,2).y=xm2;CCM(5,3).y='GCABvsDBv1r1';
    CCM(6,1).y=yy5;CCM(6,2).y=xm1;CCM(6,3).y='GCABvsDBv1r2';
    CCM(7,1).y=yy6;CCM(7,2).y=xm2;CCM(7,3).y='GCABvsDBv2r1';
    CCM(8,1).y=yy7;CCM(8,2).y=xm2;CCM(8,3).y='GCABvsDBv2r2';
    %% Impresión de caracteristicas y Distancia Euclidiana
    Var1={'Estadisticos','','','','','','';
        'Variable','Valor medio','Desviación Estandar','Raiz media
        cuadratica','Curtosis','Asimetria','Factor de forma' ;
        'GCAB',EAB1,EAB2,EAB3,EAB4,EAB5,EAB6;
```

```
102         'DA05g',E5g1,E5g2,E5g3,E5g4,E5g5,E5g6;
103         'DA10g',E1g1,E1g2,E1g3,E1g4,E1g5,E1g6;
104         'DA15g',E15g1,E15g2,E15g3,E15g4,E15g5,E15g6;
105         'DBv1r1',Ev1r1m1,Ev1r1m2,Ev1r1m3,Ev1r1m4,Ev1r1m5,Ev1r1m6;
106         'DBv1r2',Ev1r2m1,Ev1r2m2,Ev1r2m3,Ev1r2m4,Ev1r2m5,Ev1r2m6;
107         'DBv2r1',Ev2r1m1,Ev2r1m2,Ev2r1m3,Ev2r1m4,Ev2r1m5,Ev2r1m6;
108         'DBv2r2',Ev2r2m1,Ev2r2m2,Ev2r2m3,Ev2r2m4,Ev2r2m5,Ev2r2m6};
109     xlswrite1('Proyecto Cepstrum.xls',Var2,'Estadisticos','I1');
110     Var3={'Relación','Distancia Euclidiana','No CCM','Rango Cuefrencia','';
111         '','','','Minimo','Maximo';
112         'GCABvsDA05g',CEPDA1,1,min(xm),max(xm);
113         'GCABvsDA10g',CEPDA2,1,min(xm),max(xm);
114         'GCABvsDA15g',CEPDA3,3,min(xm1),max(xm1);
115         'GCABvsDBv1r1',CEPD1,4,min(xm2),max(xm2);
116         'GCABvsDBv1r2',CEPD2,3,min(xm1),max(xm1);
117         'GCABvsDBv2r1',CEPD3,2,min(xm3),max(xm3);
118         'GCABvsDBv2r2',CEPD4,2,min(xm3),max(xm3)};
119     %% Informe de Programa Finalizado correcto
120     disp('////////////////////////////////////////////////////////////')
121     disp('Todos los Calculos terminarón correctamente.')
122     disp('Sí, desea graficar los resultados; utilice grafica_5')
123     disp('Revise las caracteristicas encontradas en la carpeta Proyecto Cepstrum')
124     disp('en el documento de excel llamado Proyecto Cepstrum.xls')
125     disp('////////////////////////////////////////////////////////////')
126     end
```

Fuente: (Autor)

La función $\boldsymbol{PrincipalJZ.m}$ es del tipo función principal y sub-funciones. En la sección principal de la función denominado *%%Protecciones* de la línea 2 a la línea 32 Tabla 25; se integran una serie de condicionales los cuales son encargados de proteger la sub-función $\boldsymbol{JZ}$ del ingreso de variables erróneas, aconsejando al usuario de cómo debe ingresar los datos una vez se haya producido el error.

En la sub-función $\boldsymbol{JZ}$ de la línea 34 a la línea 126 Tabla 25; se realizan todas las operaciones pertinentes desde la selección de las muestras hasta la impresión de la distancia euclidiana y de las características de los CCM.

En *%%Selección* se realizan todas las operaciones pertinentes encaminadas a seleccionar y reemplazar los archivos ingresados; para esto en la línea 1 Tabla 25 los argumentos de entrada de la función $\boldsymbol{PrincipalJZ.m}$ solicitan el periodo muestral y el porcentaje de efectividad en las variables num y $porc$.

En *%%Programa principal* se realiza la lectura de los archivos seleccionados y se guardan en variables del tipo estructura. En *%%Transformada discreta real de Fourier* se realiza la transformada discreta real de Fourier por medio de la función $\boldsymbol{transffourier1.m}$ de la sección 3.3.2.1.5.

En *%%Transformada Cepstrum en escala Mel* se realiza la transformada por medio de la función $\boldsymbol{cepstrumMel.m}$. En *%%Selección de los coeficientes en escala Mel* se realiza la selección para cada uno de los coeficientes identificados para cada variable relacionada con GCAB, por medio de la función complementaria $\boldsymbol{recortar.m}$ de la sección 3.3.2.1.2.

En *%%Distancia euclidiana* se determina la distancia de cada una de las variables respecto al grupo de referencia por medio de la función $\boldsymbol{distanciaeuclidiana.m}$ de la sección 3.3.2.5, aplicando la ecuación (19) de la sección 2.1.11.

En *%%Variables estadísticas* se realiza la caracterización de los CCM por medio de cálculos estadísticos que incluyen el valor medio, la desviación estándar, la raíz media cuadrática, la Curtosis, la asimetría y el factor de forma; por medio de la función $\boldsymbol{EstadisticosCuefrencia.m}$ de la sección 3.3.2.6.

En *%%Impresión de características y Distancia Euclidiana* se crea un archivo de Excel con el nombre Proyecto cepstrum, en el cual se imprimen los resultados de la caracterización y de la distancia euclidiana.

Las variables de entrada línea 1 Tabla 25 de la función $\boldsymbol{PrincipalJZ.m}$ son del tipo *doublé* y $char$. Las variables $\boldsymbol{num}$ y $\boldsymbol{porc}$ son del tipo doublé; $\boldsymbol{num}$ debe ser un número entero, representa el periodo muestral de un grupo de datos y $\boldsymbol{porc}$ debe ser un número decimal; representa la precisión en la selección de las muestras. La variable $\boldsymbol{direccion}$ es del tipo $char$ y representa la dirección donde reposan los archivos para analizar.

Las variables de salida línea 1 Tabla 25 son del tipo estructura y vector $char$. La variable $\boldsymbol{CCM}$ es del tipo estructura y contiene los respectivos coeficientes cepstrales de cada una de las variables relacionadas con GCAB; y la variable $\boldsymbol{carpeta}$ es del tipo vector $char$ contiene los nombres de todos los archivos encontrados en la dirección ingresada.

Por último, para dar por terminada esta sección después de explicar cada una de las secciones de la Tabla 25 y las variables que intervienen a la entrada y salida de la función $\boldsymbol{PrincipalJZ.m}$ se le recomienda al lector revisar todas las secciones correspondientes a Matlab sección 3.3.

3.4. Alternativa de clasificación.

En esta sección se presenta una alternativa de clasificación, la cual puede ser analizada en proyectos posteriores a este trabajo. Después de la revisión y el análisis de la sección 3.3 se pudo llegar a la conclusión de que los CCM permiten diferenciar de forma visual las variables relacionadas respecto al grupo de referencia en un 71.43%. También se pudo llegar a concluir que la distancia euclidiana permite diferenciar en un 100% las variables relacionadas respecto al grupo de referencia.

La problemática que puede existir en la implementación de un clasificador por medio de los CCM, resulta en encontrar como se puede ubicar cada variable de este proyecto, dentro de un rango que permita la diferenciación en el clasificador respecto a otra variable.

Después del análisis de los resultados de caracterización sección 4.4 se puede llegar a la conclusión de que los estadísticos de valor medio y de raíz media cuadrática son los únicos estadísticos que pueden ser cuantificables.

Esta sección se encarga de determinar nuevos CCM y nuevas características dependiendo del valor medio encontrado para cada variable, en la sección de resultados para la caracterización de los CCM.

En la Tabla 26 se presenta la función $EstadisticosCuefrenciaVM.m$ la cual depende del valor medio de la función $EstadisticosCuefrencia.m$ y en la página 108 se explican algunas de sus características.

Tabla 26. Algoritmo $EstadisticosCuefrenciaVM.m$.

```
Línea  Código
  1    function [Est]=EstadisticosCuefrenciaVM(yc,EVM,m)
  2    %% Sección pricipal
  3    if strcmp(m,'GCAB')==1
  4        [yc1max,yc1min]=DivisionVM(yc,69,136,EVM);
  5        [E1max,E3max,E6max]=estadis1(yc1max,1,68);
  6        [E2max,E4max,E5max]=estadis2(yc1max,1,68,E1max);
  7        [E1min,E3min,E6min]=estadis1(yc1min,1,68);
  8        [E2min,E4min,E5min]=estadis2(yc1min,1,68,E1min);
  9    end
 10    if strcmp(m,'DA05g')==1
 11        [yc1max,yc1min]=DivisionVM(yc,1,68,EVM);
 12        [E1max,E3max,E6max]=estadis1(yc1max,1,68);
 13        [E2max,E4max,E5max]=estadis2(yc1max,1,68,E1max);
 14        [E1min,E3min,E6min]=estadis1(yc1min,1,68);
 15        [E2min,E4min,E5min]=estadis2(yc1min,1,68,E1min);
 16    end
 17    if strcmp(m,'DA10g')==1
 18        [yc1max,yc1min]=DivisionVM(yc,1,68,EVM);
 19        [E1max,E3max,E6max]=estadis1(yc1max,1,68);
 20        [E2max,E4max,E5max]=estadis2(yc1max,1,68,E1max);
 21        [E1min,E3min,E6min]=estadis1(yc1min,1,68);
 22        [E2min,E4min,E5min]=estadis2(yc1min,1,68,E1min);
 23    end
 24    if strcmp(m,'DA15g')==1
 25        [yc1max,yc1min]=DivisionVM(yc,1,68,EVM);
 26        [E1max,E3max,E6max]=estadis1(yc1max,1,68);
 27        [E2max,E4max,E5max]=estadis2(yc1max,1,68,E1max);
 28        [E1min,E3min,E6min]=estadis1(yc1min,1,68);
 29        [E2min,E4min,E5min]=estadis2(yc1min,1,68,E1min);
 30    end
 31    if strcmp(m,'DBv1r1')==1
 32        [yc1max,yc1min]=DivisionVM(yc,1,68,EVM);
 33        [E1max,E3max,E6max]=estadis1(yc1max,1,68);
 34        [E2max,E4max,E5max]=estadis2(yc1max,1,68,E1max);
 35        [E1min,E3min,E6min]=estadis1(yc1min,1,68);
 36        [E2min,E4min,E5min]=estadis2(yc1min,1,68,E1min);
 37    end
 38    if strcmp(m,'DBv1r2')==1
 39        [yc1max,yc1min]=DivisionVM(yc,1,68,EVM);
 40        [E1max,E3max,E6max]=estadis1(yc1max,1,68);
 41        [E2max,E4max,E5max]=estadis2(yc1max,1,68,E1max);
 42        [E1min,E3min,E6min]=estadis1(yc1min,1,68);
 43        [E2min,E4min,E5min]=estadis2(yc1min,1,68,E1min);
 44    end
 45    if strcmp(m,'DBv2r1')==1
```

```
        [yc1max,yc1min]=DivisionVM(yc,1,68,EVM);
        [E1max,E3max,E6max]=estadis1(yc1max,1,68);
        [E2max,E4max,E5max]=estadis2(yc1max,1,68,E1max);
        [E1min,E3min,E6min]=estadis1(yc1min,1,68);
        [E2min,E4min,E5min]=estadis2(yc1min,1,68,E1min);
    end
    if strcmp(m,'DBv2r2')==1
        [yc1max,yc1min]=DivisionVM(yc,1,68,EVM);
        [E1max,E3max,E6max]=estadis1(yc1max,1,68);
        [E2max,E4max,E5max]=estadis2(yc1max,1,68,E1max);
        [E1min,E3min,E6min]=estadis1(yc1min,1,68);
        [E2min,E4min,E5min]=estadis2(yc1min,1,68,E1min);
    end
    Est(1,1).y='E1'; Est(1,2).y=E1max; Est(1,3).y= E1min;
    Est(2,1).y='E2'; Est(2,2).y=E2max; Est(2,3).y= E2min;
    Est(3,1).y='E3'; Est(3,2).y=E3max; Est(3,3).y= E3min;
    Est(4,1).y='E4'; Est(4,2).y=E4max; Est(4,3).y= E4min;
    Est(5,1).y='E5'; Est(5,2).y=E5max; Est(5,3).y= E5min;
    Est(6,1).y='E6'; Est(6,2).y=E6max; Est(6,3).y= E6min;
    end
    %% ======================================================
    function [yc1max,yc1min]=DivisionVM(yc,ref,ref1,EVM)
    %% Sub-función 1
    cot=1;
    for i=ref:ref1
        cot3=0; cot1=length(yc(i).y);
        for i2=1:cot1; if yc(i).y(i2)==0; cot3=cot3+1; end;  end
        yc1(cot)=sum(yc(i).y)/(cot1-cot3);
        if yc1(cot)>EVM;
            yc1max(cot).y=yc(i).y;
        else
            yc1max(cot).y=[0,0,0,0,0];
        end
        if yc1(cot)<EVM
            yc1min(cot).y=yc(i).y;
        else
            yc1min(cot).y=[0,0,0,0,0];
        end
        cot=cot+1;
    end
    end
    %% ======================================================
    function [E1,E3,E6]=estadis1(yc,ref,ref1)
    %% Sub-función 2
    cot=1;cot4=0;
    for i=ref:ref1
        cot3=0;  cot1=length(yc(i).y);
        for i2=1:cot1; if yc(i).y(i2)==0; cot3=cot3+1; end;  end
        if sum(yc(i).y)==0;
            yc1(cot)=0;
        else
            yc1(cot)=sum(yc(i).y)/(cot1-cot3);
        end
        if yc1(cot)==0; cot4=cot4+1; end
        yc2(cot)=(yc1(cot))^2;  cot=cot+1;
    end
    E1=sum(yc1)/(cot-1-cot4); E3=sqrt(sum(yc2)/(cot-1-cot4));
    E6=E3/E1;
```

```
104   end
105   %% ======================================================
106   function [E2,E4,E5]=estadis2(yc,ref,ref1,E1)
107   %% Sub-función 3
108   cot=1; cot4=0;
109   for i=ref:ref1
110       cot3=0; cot1=length(yc(i).y);
111       for i2=1:cot1; if yc(i).y(i2)==0; cot3=cot3+1; end;   end
112       if sum(yc(i).y)==0; cot4=cot4+1; end
113       if sum(yc(i).y)==0
114           yc3(cot)=0;  yc4(cot)=(yc3(cot))^2;  yc5(cot)=0;
115       else
116           yc3(cot)=((sum(yc(i).y)/(cot1-cot3))-E1)^2;
117           yc4(cot)=(yc3(cot))^2;
118           yc5(cot)=((sum(yc(i).y)/(cot1-cot3))-E1)^3;
119       end
120       cot=cot+1;
121   end
122   E2=sqrt(sum(yc3)/(cot-2-cot4)); E4=sum(yc4)/(cot-2-cot4)*(E2)^4;
123   E5=sum(yc5)/(cot-2-cot4)*(E2)^3;
124   end
```

Fuente: (Autor).

La función $EstadisticosCuefrenciaVM.m$ es del tipo función principal y sub-funciones. Las variables de entrada línea 1 Tabla 26 son del tipo estructura, escalar y $char$. La variable yc es una estructura de 136 datos para este proyecto. La variable EVM es del tipo escalar y representa el valor medio obtenido por la función $EstadisticosCuefrencia.m$; y la variable m es del tipo $char$ y contiene el nombre de la variable para calcular los estadísticos. La variable de salida línea 1 Tabla 26 es del tipo estructura.

En la sección principal de la línea 2 a la línea 65 Tabla 26 se establecen una serie de condicionales con el fin de ubicar la variable a la cual se le va a realizar el análisis estadístico. La función $EstadisticosCuefrenciaVM.m$ tiene tres sub-funciones que realizan distintas operaciones dentro del algoritmo.

La sub-función $DivisionVM$ línea 67 Tabla 26 realiza la separación de cada una de las variables de la estructura yc respecto al valor medio EVM, guardando los datos por debajo y por encima del valor medio en dos estructuras distintas.

Las sub-funciones $estadis1$ y $estadis2$ son llamadas por la sección principal de la función $EstadisticosCuefrenciaVM.m$ dos veces, con el fin de calcular las variables estadísticas para las estructuras que se guardaron por debajo y por encima del valor medio en la sub-función $DivisionVM$.

En la sección de resultados 4.6 se evidencia que los estadísticos de valor medio presentan rangos que pueden llegar a influir en la clasificación de las variables relacionadas con los grupos desbalanceado y desalineado, por lo que es posible realizar estudios avanzados en la clasificación de las variables de falla respecto a los estadísticos de valor medio.

4. RESULTADOS

4.1. Algoritmo selección.

En esta sección se presentan al lector todos los resultados obtenidos al ejecutar los algoritmos de selección de la sección 3.3.2.2, para cada una de las variables de los grupos de referencia, desalineación y desbalanceo Tabla 27.

Tabla 27. Relación de variables con su grupo de pertenencia.

Grupo de pertenencia.	Variable
Grupo de referencia.	GCAB
Grupo desalineado.	DA05g
	DA10g
	DA15g
Grupo desbalanceado.	DBv1r1
	DBv1r2
	DBv2r1
	DBv2r2

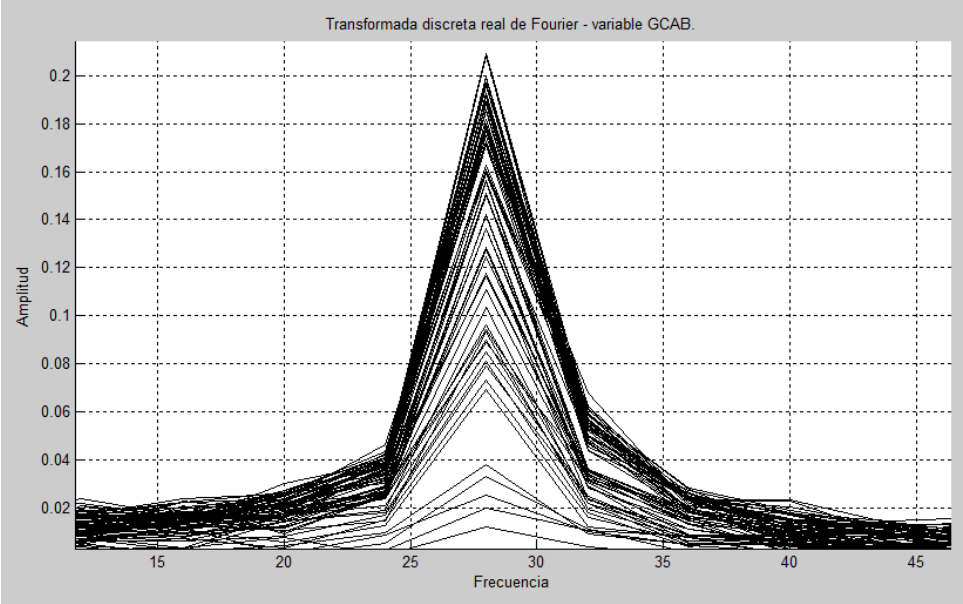
Fuente: (Autor).

Para tener un análisis completo se presentaran las gráficas sin ejecutar los algoritmos de selección y después de ejecutar los algoritmos de selección; con el fin de ratificar la pertinencia de estos algoritmos establecida en la sección 3.3.2.2 página 89. Para cada una de las variables se graficaran 68 muestras para la frecuencia fundamental de 28 Hz aplicando la función complementaria $grafica\_5.m$ Tabla 11 de acuerdo al cálculo realizado para el tamaño de la muestra en la sección 3.2.4.1.

En la Figura 28 se graficaron las muestras correspondientes a la variable GCAB la cual hace parte del grupo de referencia Tabla 27, donde se evidencia que las primeras 68 muestras de cada uno de los 68 paquetes de muestras, se encuentran dispersos de acuerdo a la amplitud en la frecuencia de 28 HZ; por lo que se hace necesario ejecutar los algoritmos de selección sección 3.3.2.2, con el fin de encontrar un rango donde concurra el mayor número de muestras Figura 29.

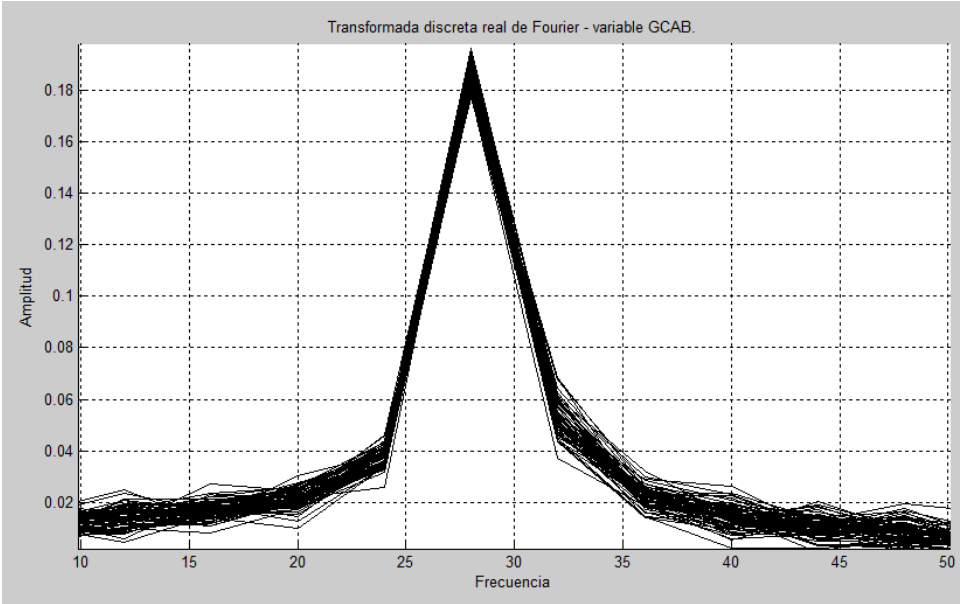
En la Figura 29 se puede observar que después de ejecutar los algoritmos de selección para un $porc = 0.9$, la amplitud máxima es 0.1967 y la amplitud mínima es 0.1774 para un número total de 68 muestras seleccionadas. En esta figura se puede apreciar que para una frecuencia de 28 Hz, las muestras se encuentran agrupadas en una forma muy definida parecida a un triángulo.

Figura 28. Transformada discreta real de Fourier, variable GCAB para 28 Hz; sin ejecutar los algoritmos de selección.



Fuente: (Autor).

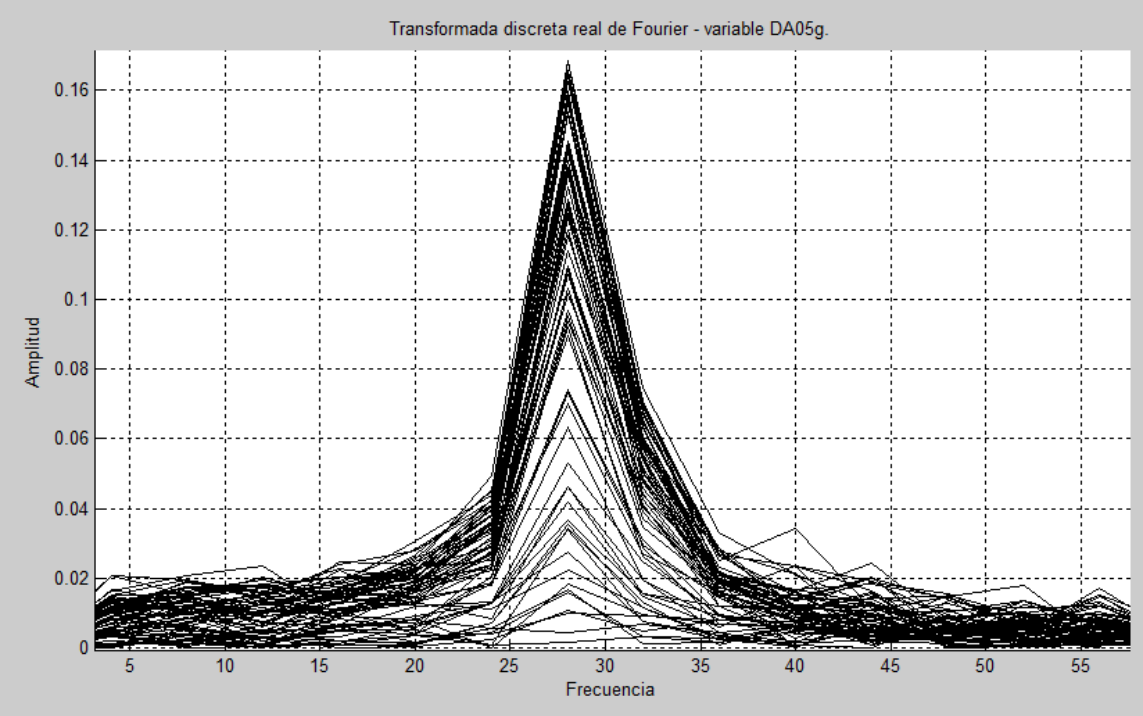
Figura 29. Transformada discreta real de Fourier, variable GCAB para 28 Hz; ejecutando los algoritmos de selección.



Fuente: (Autor).

En la Figura 30 se graficaron las muestras correspondientes a la variable DA05g, la cual hace parte del grupo desalineado Tabla 27, donde se evidencia que las primeras 68 muestras de cada uno de los 68 paquetes de muestras, se encuentran dispersos de acuerdo a la amplitud en la frecuencia de 28 HZ; por lo que se hace necesario ejecutar los algoritmos de selección sección 3.3.2.2, con el fin de encontrar un rango donde concurra el mayor número de muestras Figura 31.

Figura 30. Transformada discreta real de Fourier, variable DA05g para 28 Hz; sin ejecutar los algoritmos de selección.

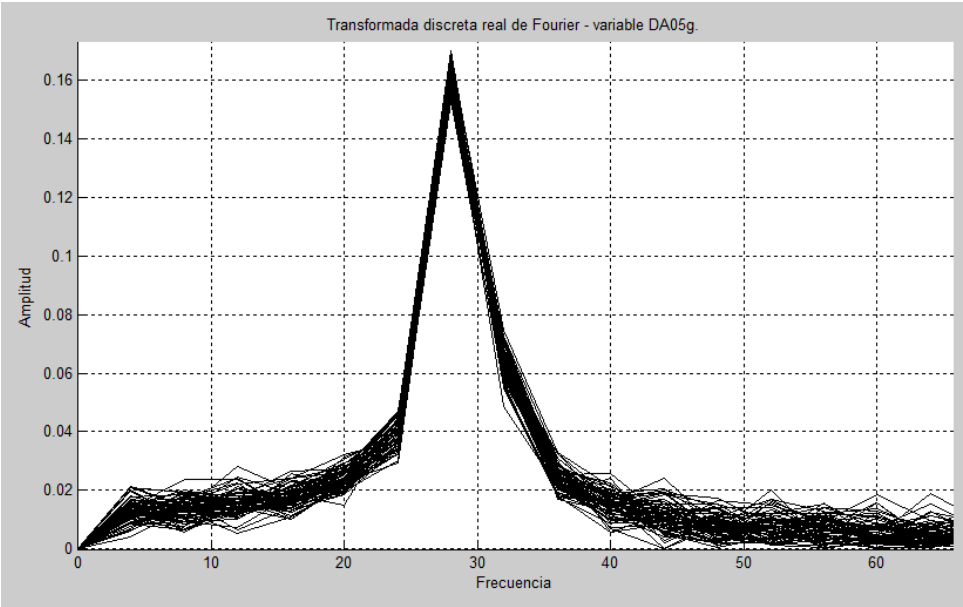


Fuente: (Autor).

En la Figura 31 se puede observar que después de ejecutar los algoritmos de selección para un $porc = 0.9$, la amplitud máxima es 0.1700 y la amplitud mínima es 0.1531 para un número total de 68 muestras seleccionadas. En esta figura se puede apreciar que para una frecuencia de 28 Hz, las muestras se encuentran agrupadas en una forma muy definida parecida a un triángulo.

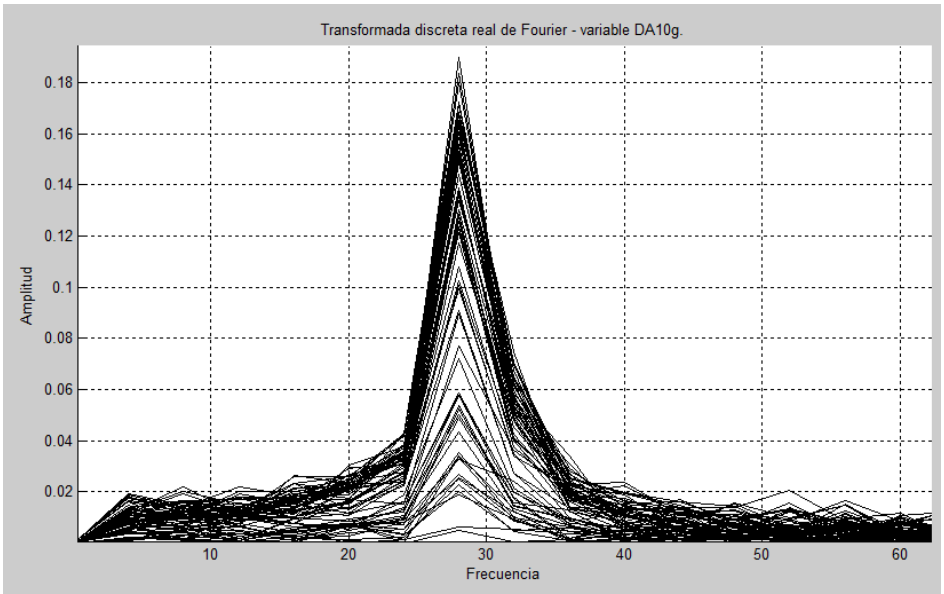
En la Figura 32 se graficaron las muestras correspondientes a la variable DA10g la cual hace parte del grupo desalineado Tabla 27, donde se evidencia que las primeras 68 muestras de cada uno de los 68 paquetes de muestras, se encuentran dispersos de acuerdo a la amplitud en la frecuencia de 28 HZ; por lo que se hace necesario ejecutar los algoritmos de selección sección 3.3.2.2, con el fin de encontrar un rango donde concurra el mayor número de muestras Figura 33.

Figura 31. Transformada discreta real de Fourier, variable DA05g para 28 Hz; ejecutando los algoritmos de selección.



Fuente: (Autor).

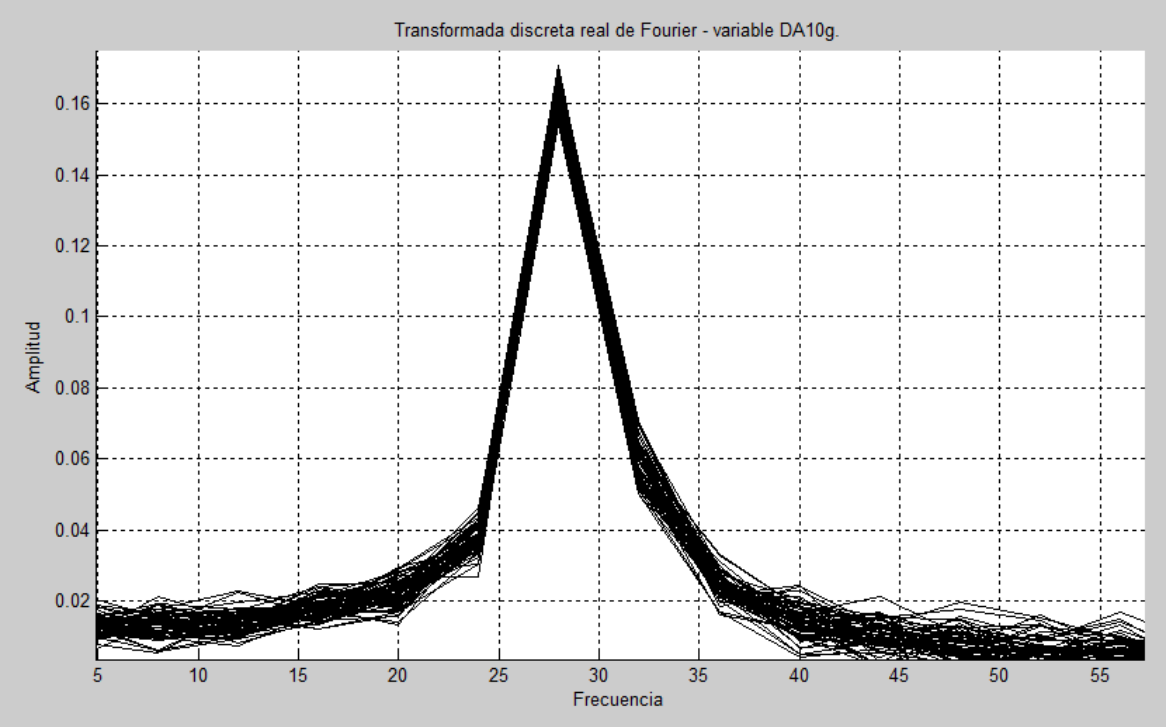
Figura 32. Transformada discreta real de Fourier, variable DA10g para 28 Hz; sin ejecutar los algoritmos de selección.



Fuente: (Autor).

En la Figura 33 se puede observar que después de ejecutar los algoritmos de selección para un $porc = 0.9$, la amplitud máxima es 0.1709 y la amplitud mínima es 0.1540 para un número total de 68 muestras seleccionadas. En esta figura se puede apreciar que para una frecuencia de 28 Hz, las muestras se encuentran agrupadas en una forma muy definida parecida a un triángulo.

Figura 33. Transformada discreta real de Fourier, variable DA10g para 28 Hz; ejecutando los algoritmos de selección.

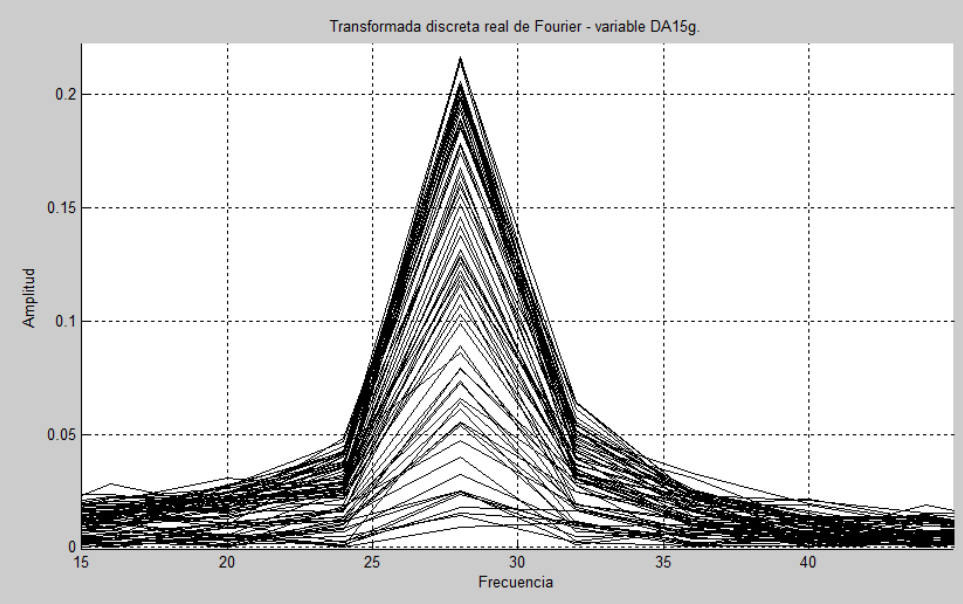


Fuente: (Autor).

En la Figura 34 se graficaron las muestras correspondientes a la variable DA15g la cual hace parte del grupo desalineado Tabla 27, donde se evidencia que las primeras 68 muestras de cada uno de los 68 paquetes de muestras, se encuentran dispersos de acuerdo a la amplitud en la frecuencia de 28 HZ; por lo que se hace necesario ejecutar los algoritmos de selección sección 3.3.2.2, con el fin de encontrar un rango donde concurra el mayor número de muestras Figura 35.

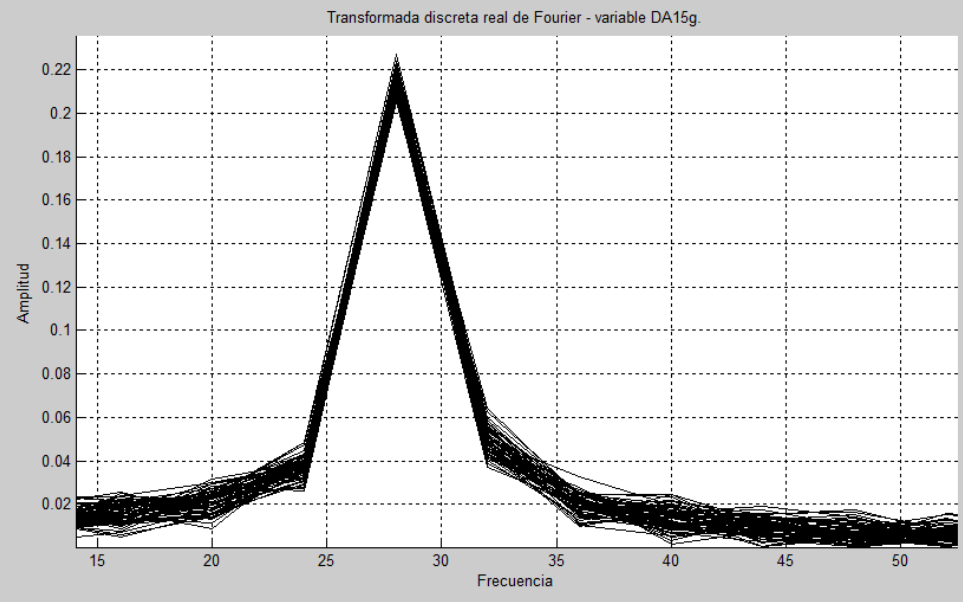
En la Figura 35 se puede observar que después de ejecutar los algoritmos de selección para un $porc = 0.9$, la amplitud máxima es 0.2274 y la amplitud mínima es 0.2055 para un número total de 68 muestras seleccionadas. En esta figura se puede apreciar que para una frecuencia de 28 Hz, las muestras se encuentran agrupadas en una forma muy definida parecida a un triángulo.

Figura 34. Transformada discreta real de Fourier, variable DA15g para 28 Hz; sin ejecutar los algoritmos de selección.



Fuente: (Autor).

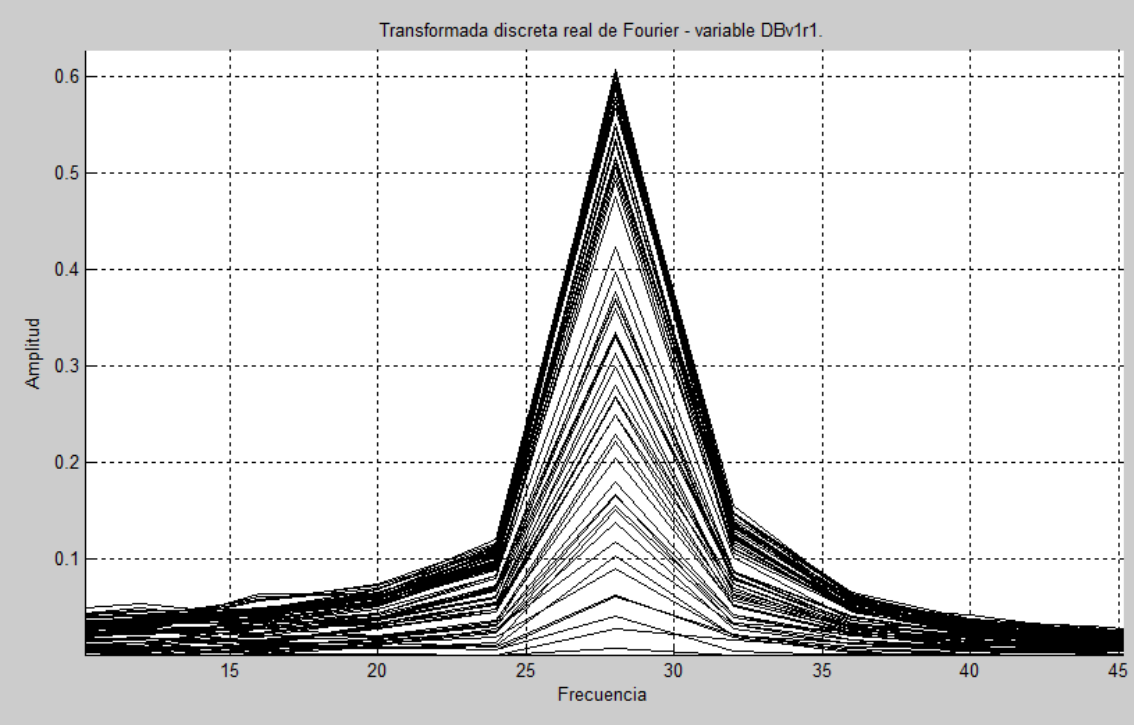
Figura 35. Transformada discreta real de Fourier, variable DA15g para 28 Hz; ejecutando los algoritmos de selección.



Fuente: (Autor).

En la Figura 36 se graficaron las muestras correspondientes a la variable DBv1r1 la cual hace parte del grupo desbalanceado Tabla 27, donde se evidencia que las primeras 68 muestras de cada uno de los 68 paquetes de muestras, se encuentran dispersos de acuerdo a la amplitud en la frecuencia de 28 HZ; por lo que se hace necesario ejecutar los algoritmos de selección sección 3.3.2.2, con el fin de encontrar un rango donde concurra el mayor número de muestras Figura 37.

Figura 36. Transformada discreta real de Fourier, variable DBv1r1 para 28 Hz; sin ejecutar los algoritmos de selección.

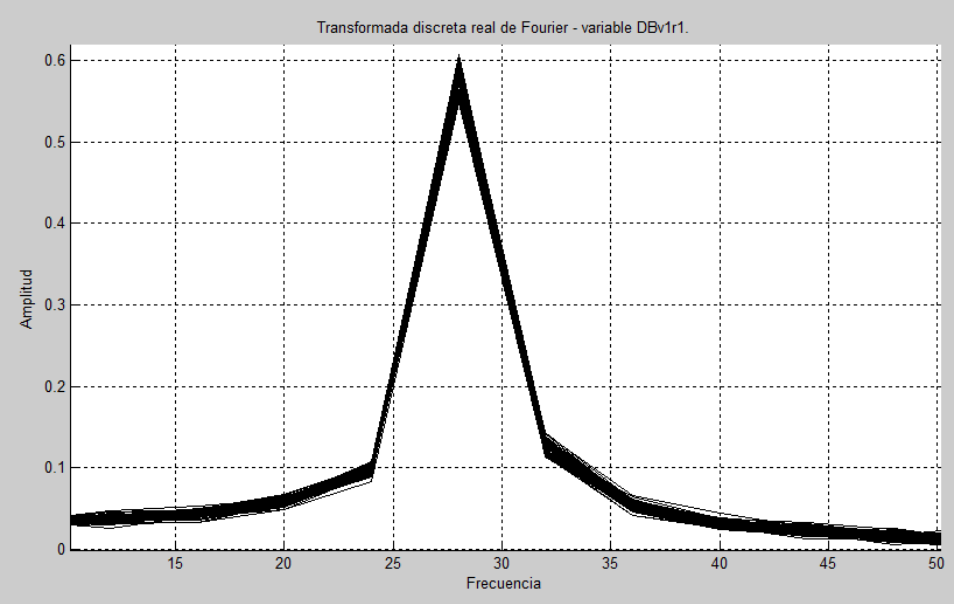


Fuente: (Autor).

En la Figura 37 se puede observar que después de ejecutar los algoritmos de selección para un $porc = 0.9$, la amplitud máxima es 0.6073 y la amplitud mínima es 0.5480 para un número total de 68 muestras seleccionadas. En esta figura se puede apreciar que para una frecuencia de 28 Hz, las muestras se encuentran agrupadas en una forma muy definida parecida a un triángulo.

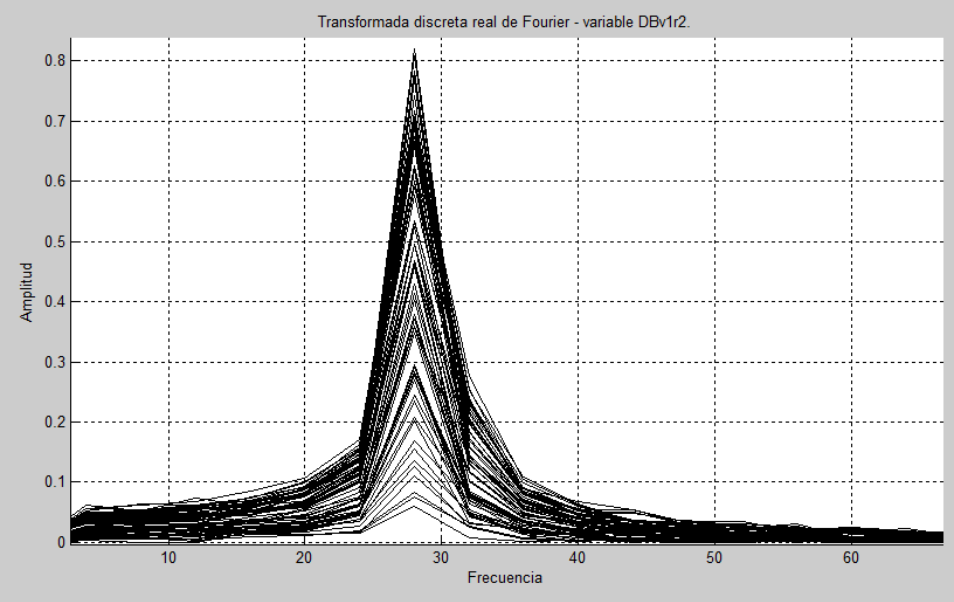
En la Figura 38 se graficaron las muestras correspondientes a la variable DBv1r2 la cual hace parte del grupo desbalanceado Tabla 27, donde se evidencia que las primeras 68 muestras de cada uno de los 68 paquetes de muestras, se encuentran dispersos de acuerdo a la amplitud en la frecuencia de 28 HZ; por lo que se hace necesario ejecutar los algoritmos de selección sección 3.3.2.2, con el fin de encontrar un rango donde concurra el mayor número de muestras Figura 39.

Figura 37. Transformada discreta real de Fourier, variable DBv1r1 para 28 Hz; ejecutando los algoritmos de selección.



Fuente: (Autor).

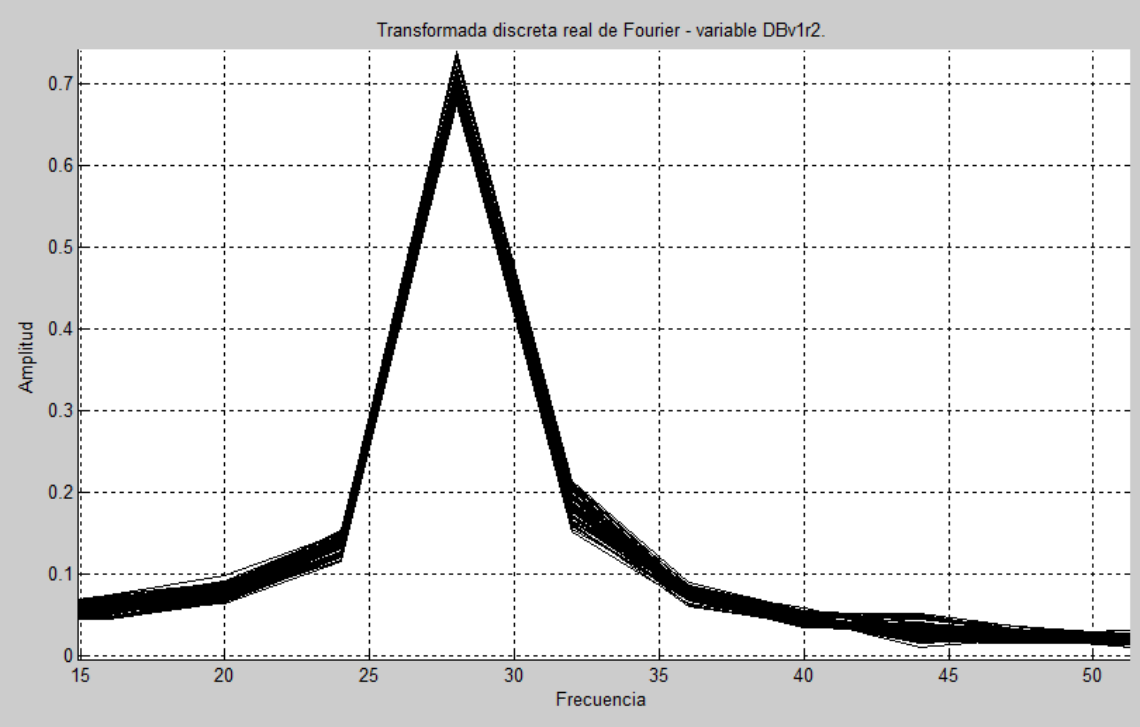
Figura 38. Transformada discreta real de Fourier, variable DBv1r2 para 28 Hz; sin ejecutar los algoritmos de selección.



Fuente: (Autor).

En la Figura 39 se puede observar que después de ejecutar los algoritmos de selección para un $porc = 0.9$, la amplitud máxima es 0.7430 y la amplitud mínima es 0.6743 para un número total de 68 muestras seleccionadas. En esta figura se puede apreciar que para una frecuencia de 28 Hz, las muestras se encuentran agrupadas en una forma muy definida parecida a un triángulo.

Figura 39. Transformada discreta real de Fourier, variable DBv1r2 para 28 Hz; ejecutando los algoritmos de selección.

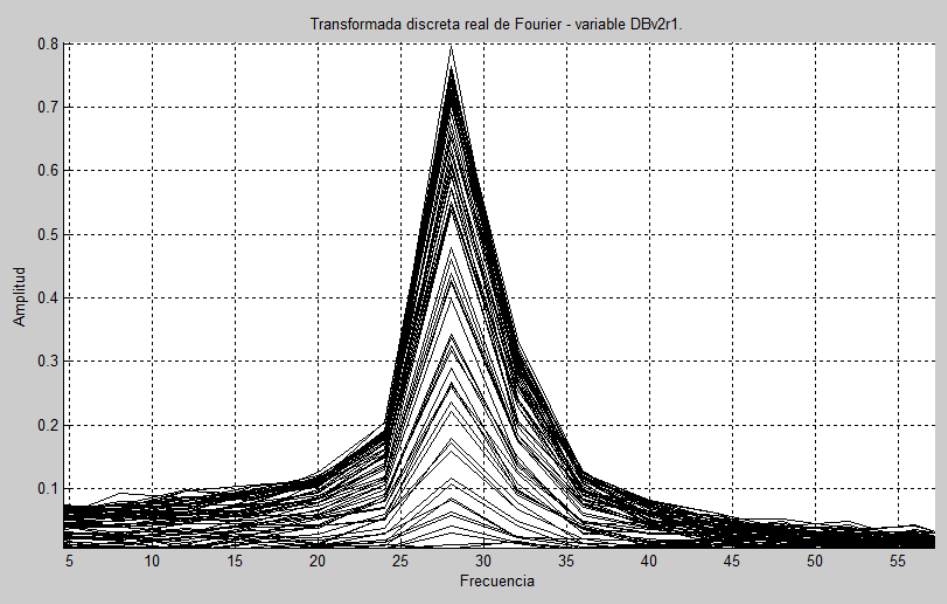


Fuente: (Autor).

En la Figura 40 se graficaron las muestras correspondientes a la variable DBv2r1 la cual hace parte del grupo desbalanceado Tabla 27, donde se evidencia que las primeras 68 muestras de cada uno de los 68 paquetes de muestras, se encuentran dispersos de acuerdo a la amplitud en la frecuencia de 28 HZ; por lo que se hace necesario ejecutar los algoritmos de selección sección 3.3.2.2, con el fin de encontrar un rango donde concurra el mayor número de muestras Figura 41.

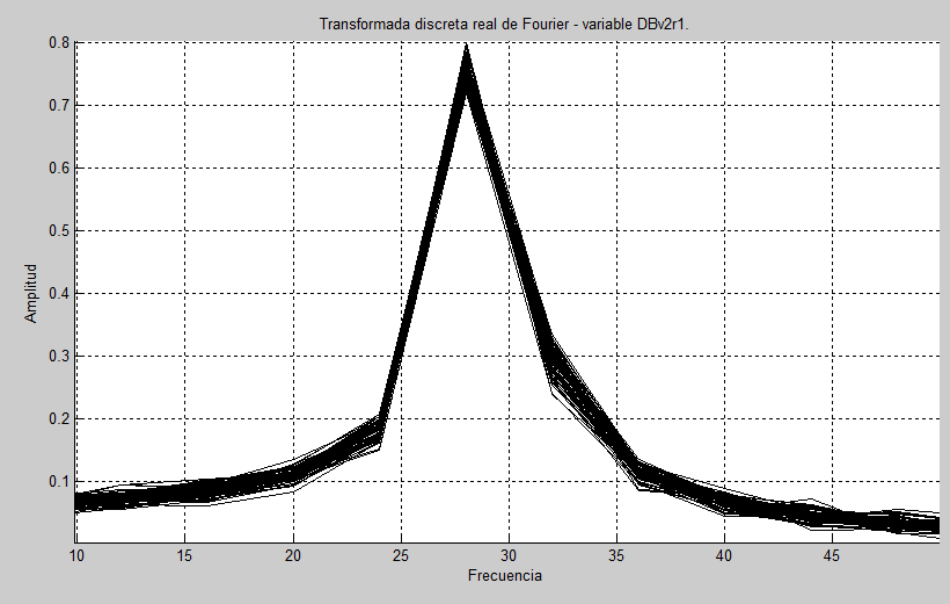
En la Figura 41 se puede observar que después de ejecutar los algoritmos de selección para un $porc = 0.9$, la amplitud máxima es 0.7999 y la amplitud mínima es 0.7203 para un número total de 68 muestras seleccionadas. En esta figura se puede apreciar que para una frecuencia de 28 Hz, las muestras se encuentran agrupadas en una forma muy definida parecida a un triángulo.

Figura 40. Transformada discreta real de Fourier, variable DBv2r1 para 28 Hz; sin ejecutar los algoritmos de selección.



Fuente: (Autor).

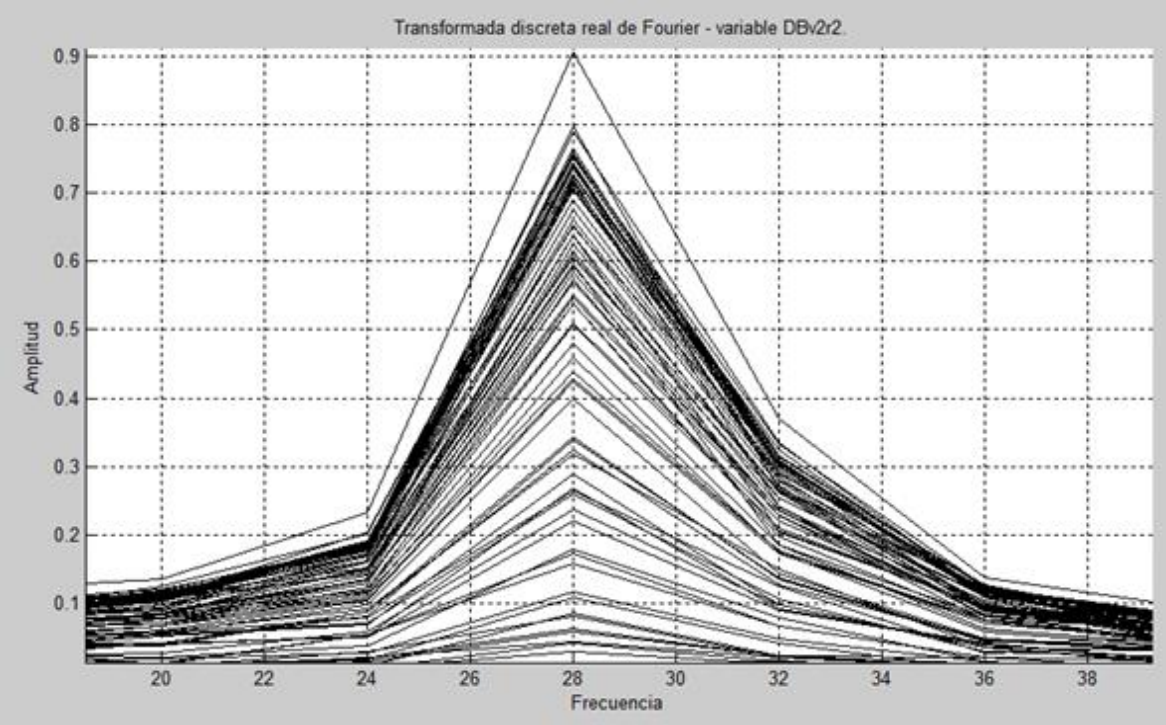
Figura 41. Transformada discreta real de Fourier, variable DBv2r1 para 28 Hz; ejecutando los algoritmos de selección.



Fuente: (Autor).

En la Figura 42 se graficaron las muestras correspondientes a la variable DBv2r2 la cual hace parte del grupo desalineado Tabla 27, donde se evidencia que las primeras 68 muestras de cada uno de los 68 paquetes de muestras, se encuentran dispersos de acuerdo a la amplitud en la frecuencia de 28 HZ; por lo que se hace necesario ejecutar los algoritmos de selección sección 3.3.2.2, con el fin de encontrar un rango donde concurra el mayor número de muestras Figura 43.

Figura 42. Transformada discreta real de Fourier, variable DBv2r2 para 28 Hz; sin ejecutar los algoritmos de selección.

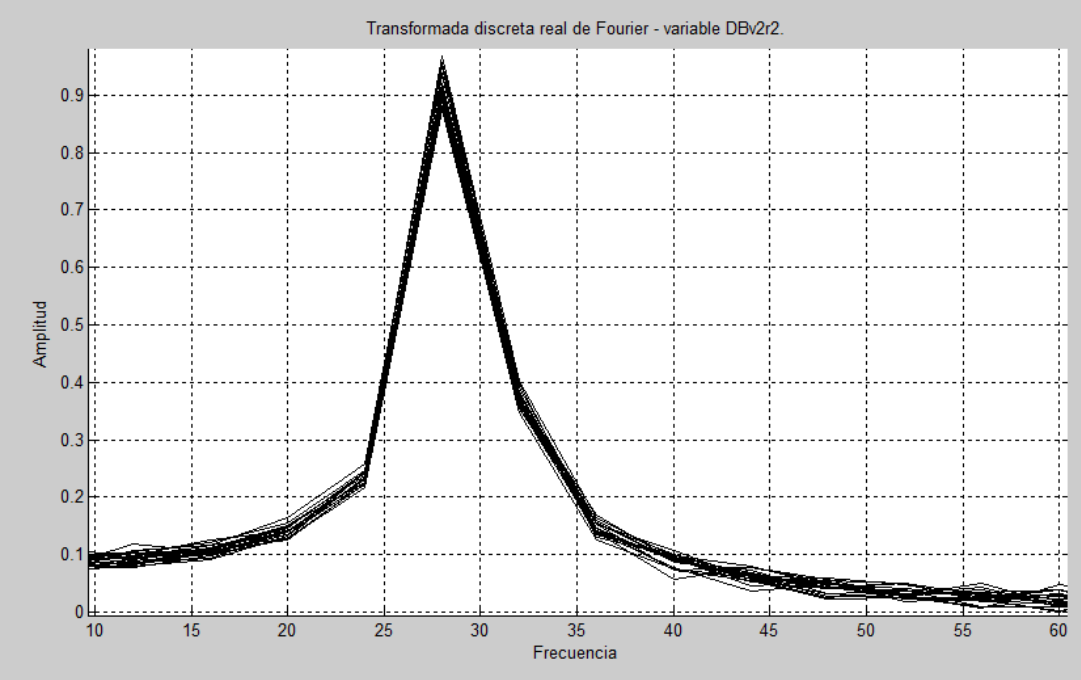


Fuente: (Autor).

En la Figura 43 se puede observar que después de ejecutar los algoritmos de selección para un $porc = 0.9$, la amplitud máxima es 0.9677 y la amplitud mínima es 0.8739 para un número total de 68 muestras seleccionadas. En esta figura se puede apreciar que para una frecuencia de 28 Hz, las muestras se encuentran agrupadas en una forma muy definida parecida a un triángulo.

Después de revisar la Figura 28, Figura 30, Figura 32, Figura 34, Figura 36, Figura 38, Figura 40 y la Figura 42; se puede llegar a la concluir que para todas las variables de este proyecto es indispensable ejecutar los algoritmos de selección de la sección 3.3.2.2, debido a que para la frecuencia fundamental de 28 Hz; todas las variables presentan una dispersión en la amplitud, la cual no permite realizar la transformación de las señales al dominio cepstrum. También de estas figuras se puede ratificar la pertinencia del diseño de los algoritmos de selección.

Figura 43. Transformada discreta real de Fourier, variable DBv2r2 para 28 Hz; ejecutando los algoritmos de selección.



Fuente: (Autor).

Después de revisar la Figura 29, Figura 31, Figura 33, Figura 35, Figura 37, Figura 39, Figura 41 y la Figura 43; para las cuales se ejecutaron los algoritmos de selección, se puede evidenciar que los algoritmos permiten agrupar las muestras en un rango de amplitud, para cada una de las variables aplicando un $porc = 0.9$ que cumple con el número de muestras calculado en la sección 3.2.4.1.

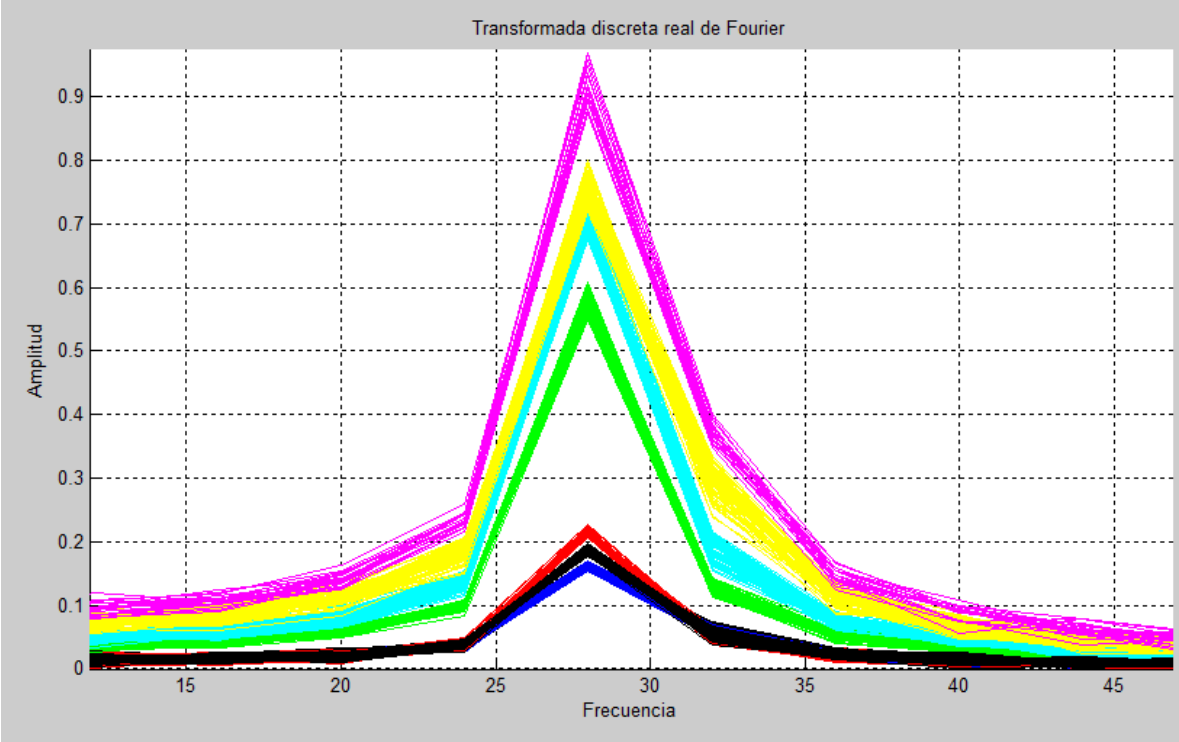
En la Figura 44 se puede resumir los resultados para cada una de las variables de la Tabla 27, después de ejecutar los algoritmos de selección según la asignación de colores de la Tabla 28.

Tabla 28. Asignación de colores a las variables de la Tabla 27.

Variable	**Color**
GCAB	Negro
DA05g	Negro
DA10g	Azul
DA15g	Rojo
DBv1r1m	Verde
DBv1r2m	Cian
DBv2r1m	Amarillo
DBv2r2m	Magenta

Fuente: (Autor).

Figura 44. Transformada discreta real de Fourier, para todas las variables de la Tabla 27.



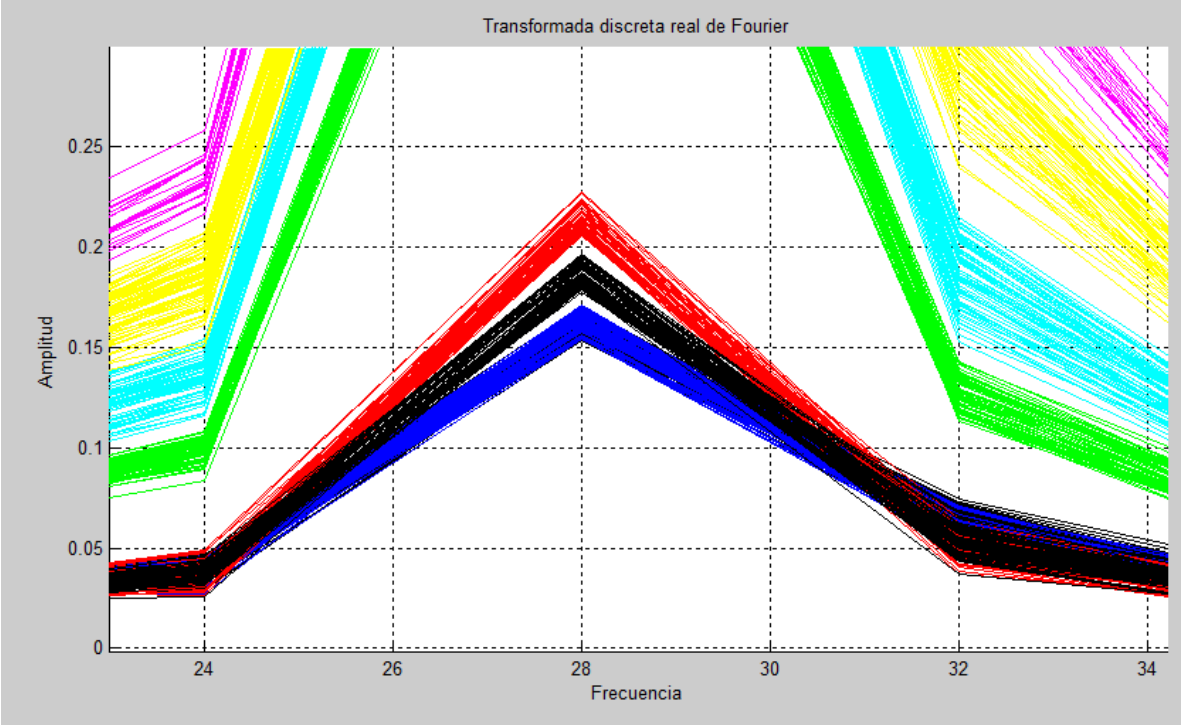
Fuente: (Autor).

Después de revisar la Figura 44 se puede concluir que el grupo desbalanceado tiene mayor amplitud que el grupo de referencia. También se puede concluir que los desbalances que se producen lejos del motor poseen una amplitud mayor, según sea su distancia respecto al plano horizontal del eje secundario; es decir, las variables correspondientes al volante 2 poseen una mayor amplitud que las variables correspondientes al volante 1.

En la Figura 45 se realiza una ampliación de la Figura 44 con el fin de revisar el grupo de desalineación. En esta figura se puede concluir que la gráfica de color negro corresponde a las muestras del grupo de referencia y que la variable DA05g se encuentra detrás de la variable DA10g graficada en azul; según las amplitudes encontradas para cada una de las variables en las Figura 31 y Figura 33.

Con la Figura 44 y la Figura 45 termina esta sección, siendo todo el análisis realizado un gran aporte al primer objetivo específico, referente a la obtención de registros; debido a que permite al programador tener un referente en cuanto a la decisión de si las muestras seleccionadas se agrupan en distintas amplitudes de la transformada de Fourier real y establece las bases para poder transformar al dominio cepstrum las diferentes variables.

Figura 45. Ampliación de la Figura 44 respecto al grupo desalineado.



Fuente: (Autor)

4.2. Transformada cepstrum.

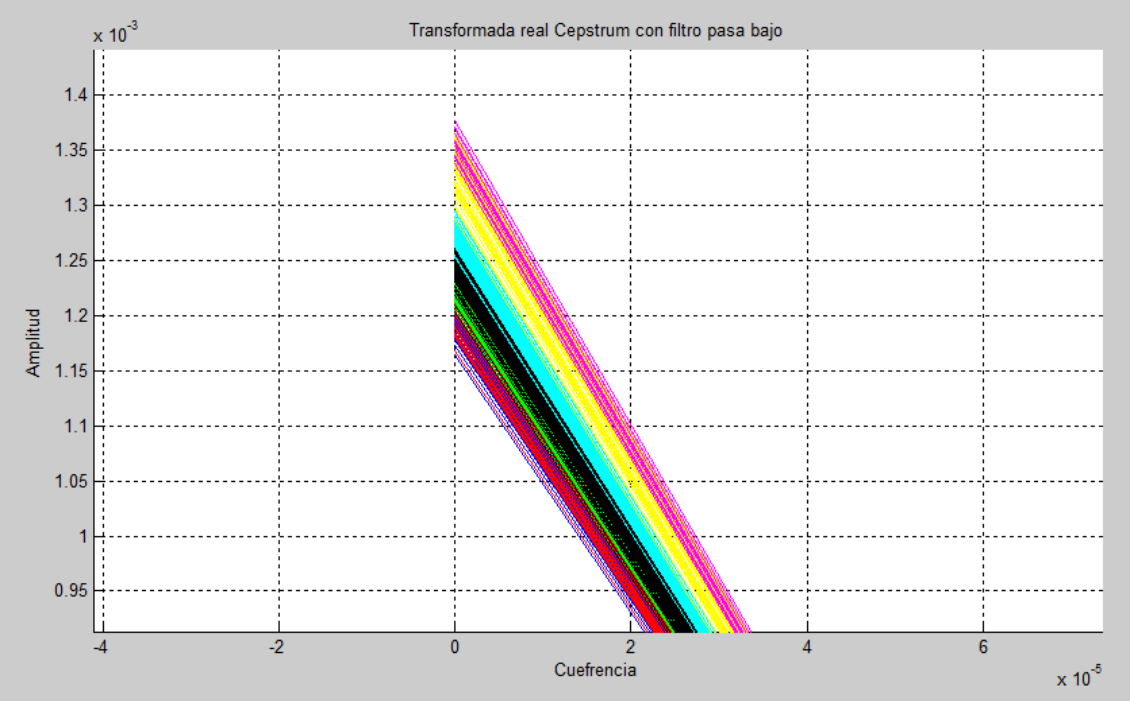
En esta sección se presentan los resultados obtenidos aplicando las diferentes expresiones de la transformada cepstrum adoptadas por este proyecto, para la diferenciación de las variables de la Tabla 28 y aplicando la envolvente al cepstrum; la cual se define como la transformada discreta real de Fourier del cepstrum línea 6 Tabla 17.

4.2.1. Transformada cepstrum real.

En esta sección se presentan los resultados de la transformada cepstrum real propuesta en la sección 3.3.2.4.1, donde se grafican las variables de la Tabla 28; con el fin de encontrar coeficientes característicos que permitan diferenciar las variables entre sí.

En la Figura 46 se graficaron las variables de la Tabla 28, aplicando la transformada cepstrum real con un filtro pasa bajo aplicado a la transformada discreta real de Fourier Tabla 17. Esta grafica contiene un solo coeficiente cepstrum ubicado con amplitud máxima a una cuefrencia de 0.

Figura 46. Grafica de las variables de cada grupo aplicando cepstrum real pasa bajo.



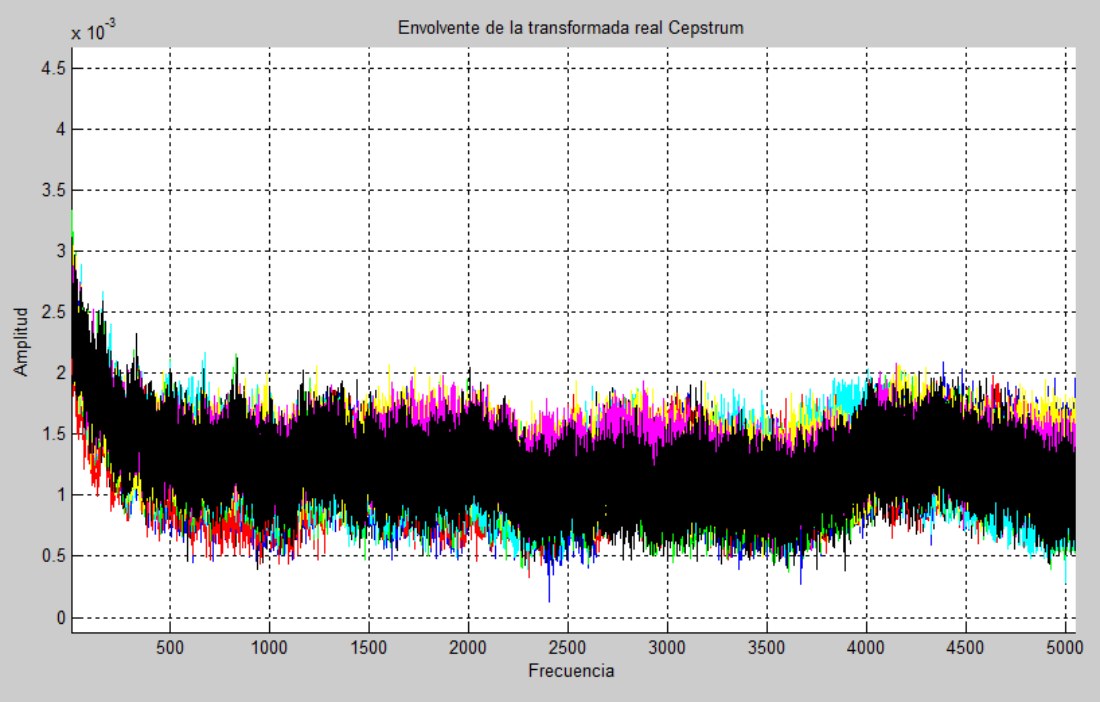
Fuente: (Autor).

Revisando la Figura 46 según la asignación de colores de la Tabla 28, se puede ver que las variables correspondientes al grupo desalineado, aparecen mezcladas entre sí, con el grupo de referencia y con las variables del volante 1 del grupo desbalanceado. Por lo que la transformada cepstrum real con filtro pasa bajo aplicado a la transformada discreta de Fourier no permite diferenciar las variables entre sí.

En la Figura 47 se graficaron las variables de la Tabla 28, eliminando el filtro pasa bajo aplicado a la transformada discreta real de Fourier de la Figura 46 y aplicando la envolvente al cepstrum; con el fin de encontrar coeficientes que puedan diferenciar entre sí a cada una de las variables.

Revisando la Figura 47 según la asignación de colores de la Tabla 28, se puede ver que no existe diferenciación entre las variables correspondientes a los grupos de referencia, desalineado y desbalanceado; por lo que la transformada discreta real de Fourier sin filtro pasa bajo y aplicando la envolvente al cepstrum; no permite encontrar coeficiente definido para diferenciar cada una de las variables estudiadas.

Figura 47. Envolvente transformada real cepstrum.



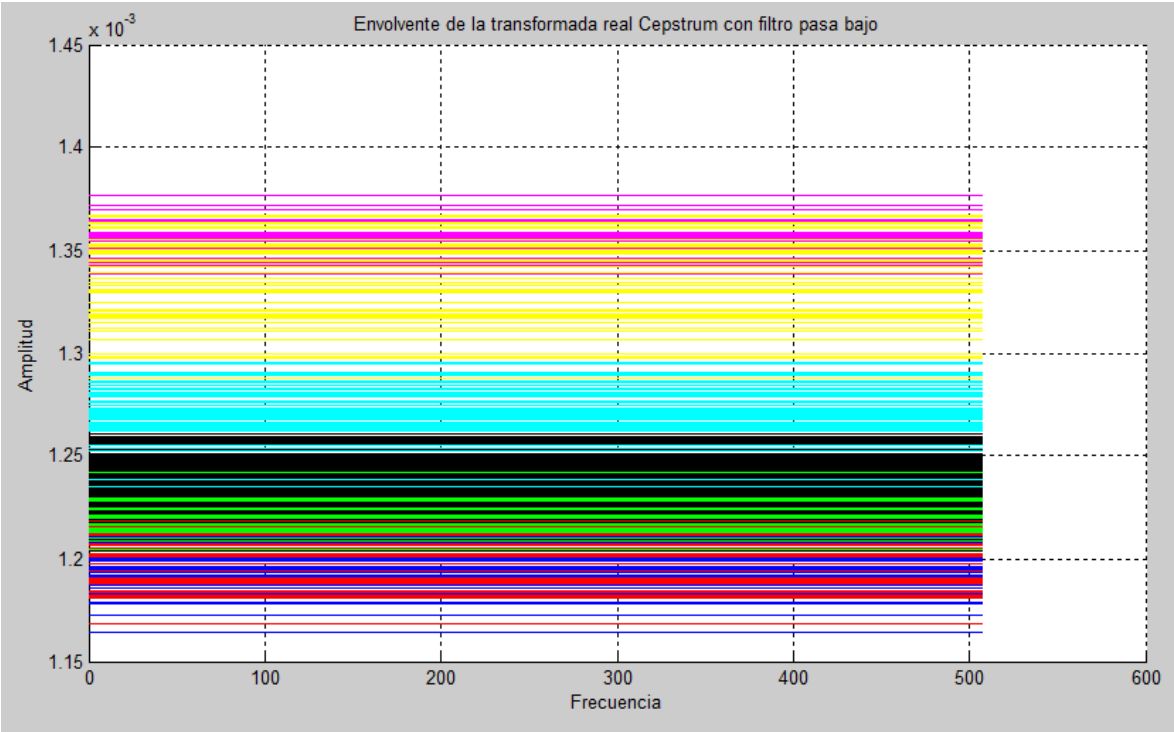
Fuente (Autor).

En la Figura 48 se graficaron las variables de cada uno de los grupos aplicando la envolvente del cepstrum real con un filtro pasa bajo, con el fin de encontrar coeficientes definidos que permitan encontrar diferencias entre las variables de la Tabla 28.

Revisando la Figura 48 según la asignación de colores de la Tabla 28, se puede ver que no existe diferenciación entre las variables correspondientes a los grupos de referencia, desalineado y desbalanceado; por lo que la transformada discreta real de Fourier con filtro pasa bajo y aplicando la envolvente al cepstrum; no permite encontrar coeficiente definido para diferenciar cada una de las variables estudiadas.

Después de revisar cada una de las variantes propuestas Figura 46, Figura 47 y Figura 48; para la transformada cepstrum real se puede llegar a concluir que la transformada cepstrum real y sus distintas envolventes, no permiten obtener coeficientes cepstrum definidos; los cuales puedan establecer alguna diferencia entre las variables estudiadas.

Figura 48. Grafica de las variables de cada grupo aplicando la envolvente del cepstrum real pasa bajo.



Fuente (Autor).

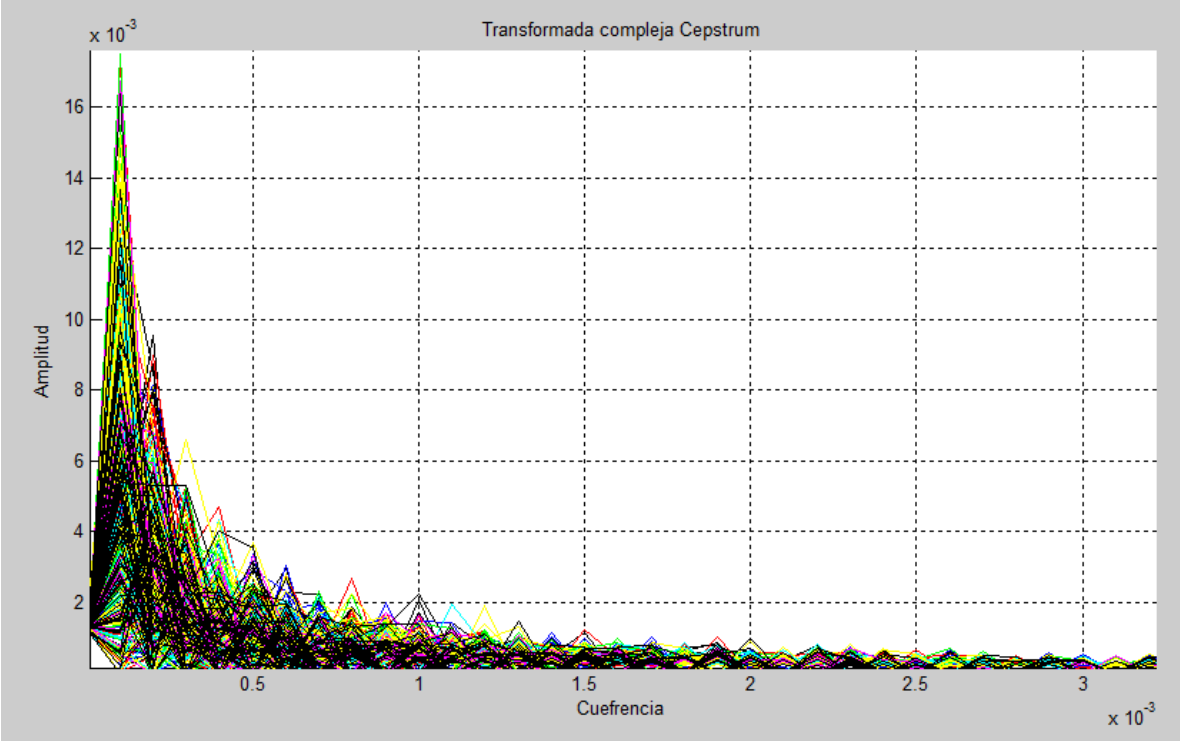
4.2.2. Transformada cepstrum compleja.

En esta sección se presentan los resultados de la transformada cepstrum compleja propuesta en la sección 3.3.2.4.2, donde se grafican las variables de la Tabla 28; con el fin de encontrar coeficientes característicos que permitan diferenciar las variables entre sí.

En la Figura 49 se graficaron las variables de cada uno de los grupos aplicando la transformada compleja cepstrum con filtro pasa bajo; con el fin de encontrar coeficientes definidos que permitan encontrar diferencias entre las variables de la Tabla 28.

Revisando la Figura 49 según la asignación de colores de la Tabla 28, se puede ver que no existe diferenciación entre las variables correspondientes a los grupos de referencia, desalineado y desbalanceado; debido a que los colores de las variables aparecen mezclados en la Figura 49, a pesar de que aparecen coeficientes definidos en forma triangular; por lo que la transformada compleja cepstrum con filtro pasa bajo no permite encontrar coeficientes definidos para diferenciar cada una de las variables estudiadas.

Figura 49. Grafica de las variables; aplicando la transformada compleja cepstrum con filtro pasa bajo.



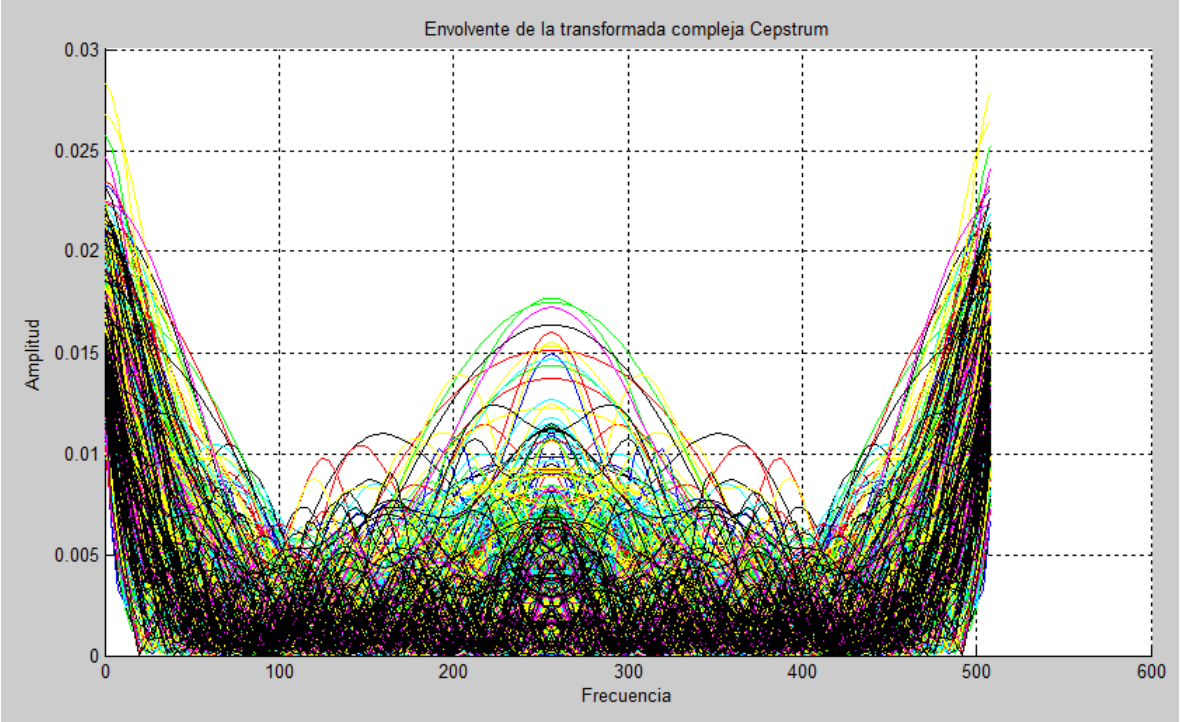
Fuente: (Autor).

En la Figura 50 se graficaron las variables de cada uno de los grupos aplicando la envolvente de la transformada compleja cepstrum con filtro pasa bajo; con el fin de encontrar coeficientes definidos que permitan encontrar diferencias entre las variables de la Tabla 28.

Revisando la Figura 50 según la asignación de colores de la Tabla 28, se puede ver que no existe diferenciación entre las variables correspondientes a los grupos de referencia, desalineado y desbalanceado; debido a que los colores de las variables aparecen mezclados en la Figura 50, a pesar de que aparecen coeficientes definidos del tipo ventana sombrero mexicano; por lo que la envolvente de la transformada compleja cepstrum con filtro pasa bajo, no permite encontrar coeficientes definidos para diferenciar cada una de las variables estudiadas.

Después de revisar cada una de las variantes propuestas Figura 47 y Figura 48; para la transformada cepstrum compleja se puede llegar a concluir que la transformada cepstrum compleja y su envolvente no permiten obtener coeficientes cepstrum definidos los cuales puedan establecer alguna diferencia entre las variables estudiadas.

Figura 50. Grafica de las variables; aplicando la envolvente de la transformada compleja cepstrum con filtro pasa bajo.



Fuente: (Autor).

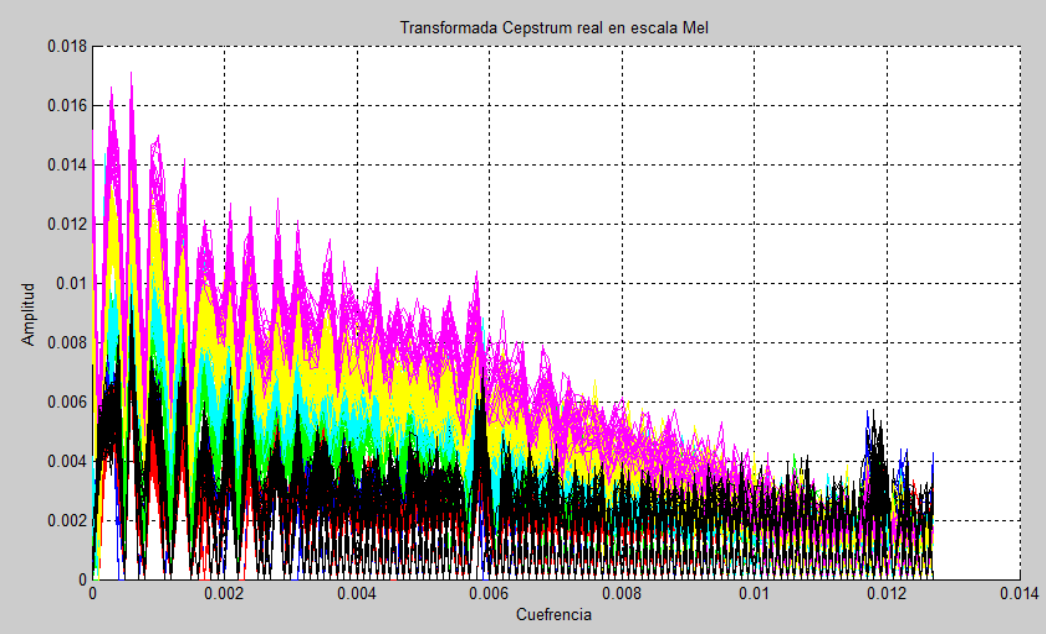
4.2.3. Transformada cepstrum en la escala Mel.

En esta sección se presentan los resultados de la transformada cepstrum en la escala Mel propuesta en la sección 3.3.2.4.3, donde se grafican las variables de la Tabla 28; con el fin de encontrar coeficientes característicos que permitan diferenciar las variables entre sí.

En la Figura 51 se graficaron las variables de cada uno de los grupos aplicando la transformada cepstrum en la escala Mel; con el fin de encontrar coeficientes definidos que permitan encontrar diferencias entre las variables de la Tabla 28.

Revisando la Figura 51 según la asignación de colores de la Tabla 28, se puede ver que existen coeficientes definidos y también existe diferenciación entre las variables correspondientes a los grupos de referencia y desalineado; respecto al grupo desbalanceado, por lo que se revisó detenidamente cada uno de los coeficientes cepstrum; con el fin de verificar si existen diferencias entre cada una de las variables.

Figura 51. Grafica de las variables aplicando la transformada cepstrum en escala Mel.



Fuente (Autor).

En la Tabla 29 se resume los CCM encontrados, después de hacer una revisión detenida en cada uno de ellos respecto al grupo de referencia. En la 2 columna se puede ver las parejas de relación que se conforman de cada una de las variables de los grupos desalineado y desbalanceado Tabla 27, respecto al grupo de referencia. En la 3 columna se puede ver el coeficiente específico que diferencia a cada una de las variables respecto a GCAB. En la 4 columna se puede ver el rango en el eje de cuefrencia que toma cada uno de los coeficientes y la última columna es una nueva asignación de colores para la comparación respecto al grupo de referencia, el cual toma el color azul como parte del grupo.

Tabla 29. Comparación de Variables.

No	Comparación	No CCM	RC * 10-4	Color
1	GCABvsDA05g	1	2 – 6	Azul – Negro
2	GCABvsDA10g	1	2 – 6	Azul – Rojo
3	GCABvsDA15g	3	9 – 13	Azul – Verde
4	GCABvsDBv1r1m	4	13 – 16	Azul – Cian
5	GCABvsDBv1r2m	3	9 – 13	Azul – Amarillo
6	GCABvsDBv2r1m	2	6 – 9	Azul – Magenta
7	GCABvsDBv2r2m	2	6 – 9	Azul – Negro

CCM: Coeficientes cepstrum en la escala Mel.
RC: Rango Cuefrencia.

Fuente (Autor).

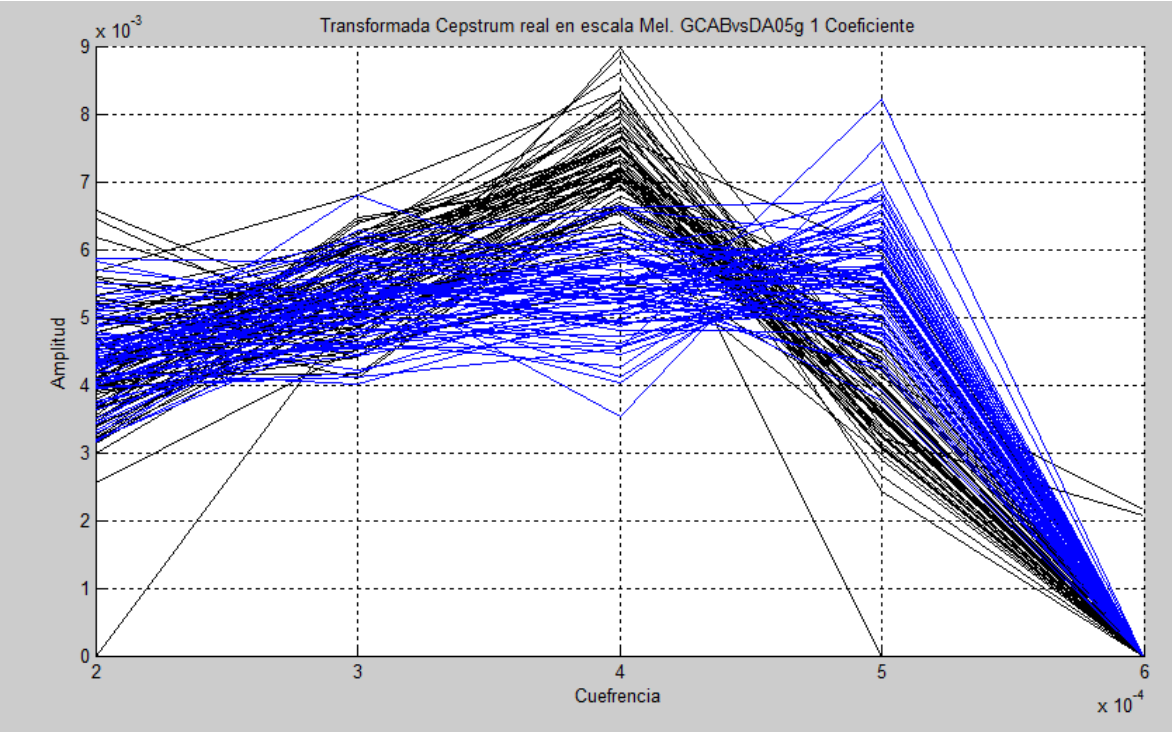
Después de hacer una revisión detenida de los CCM resumida en la Tabla 29, se puede llegar a concluir que los primeros 4 coeficientes del CCM, permiten diferenciar a cada una de las variables de los grupos desbalanceado y desalineado; respecto a la variable GCAB, por lo que es importante ver las gráficas de cada uno de los coeficientes seleccionados con el fin de verificar visualmente los datos obtenidos.

Coeficientes para el grupo desalineado.

En esta sección se presentan los coeficientes del CCM para las variables de grupo desalineado en referencia con la variable GCAB; con el fin de visualizar los coeficientes que diferencian a cada una de las variables comparadas en este grupo.

En la Figura 52 se grafica el primer CCM para la comparación GCABvsDA05g. Este es el coeficiente encontrado en el cual se diferencian las variables relacionadas; se puede ver una pequeña dispersión en la forma de los datos en cada una de las variables. También se puede ver que la variable DA05g en general posee datos por encima de GCAB, pero no es concluyente la diferenciación de las variables por visualización.

Figura 52. Coeficiente No 1 del CCM para la comparación GCABvsDA05g.

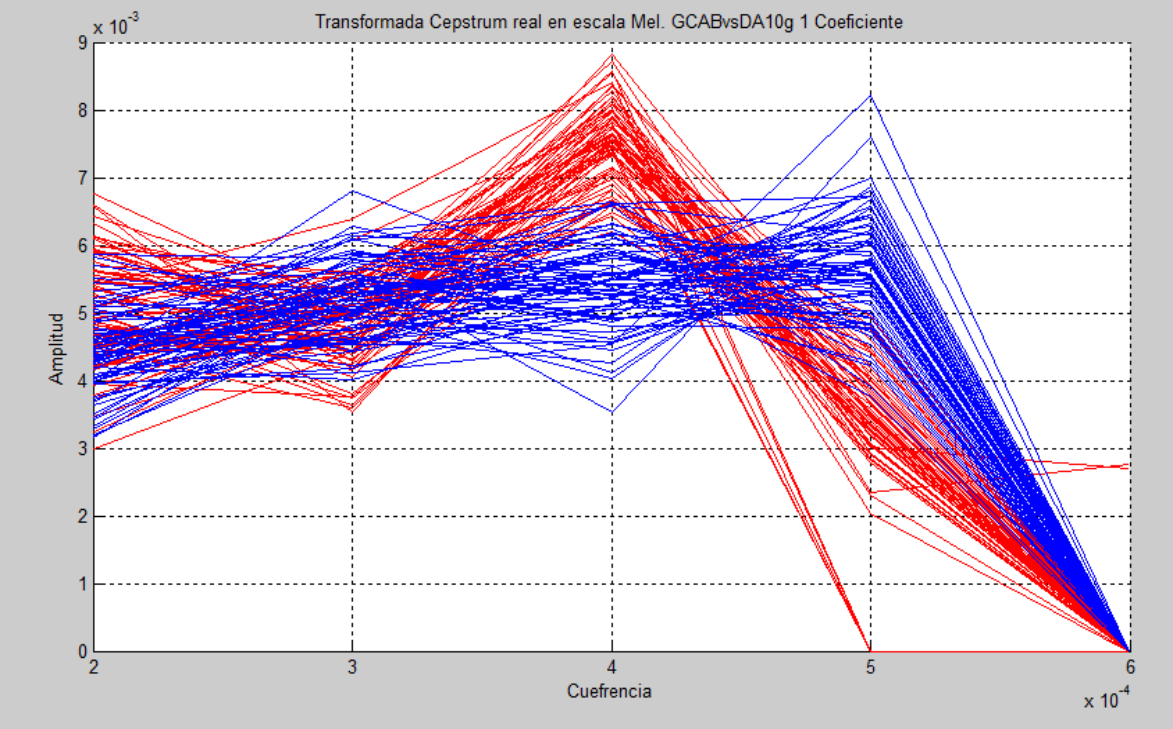


Fuente: (Autor).

En la Figura 53 se grafica el primer CCM para la comparación GCABvsDA10g. Este es el coeficiente encontrado en el cual se diferencian las variables relacionadas; se puede ver una pequeña dispersión en la forma de los datos en cada una de las variables. También se

puede ver que la variable DA10g en general posee datos por encima de GCAB, pero no es concluyente la diferenciación de las variables por visualización.

Figura 53. Coeficiente No 1 del CCM para la comparación GCABvsDA10g.

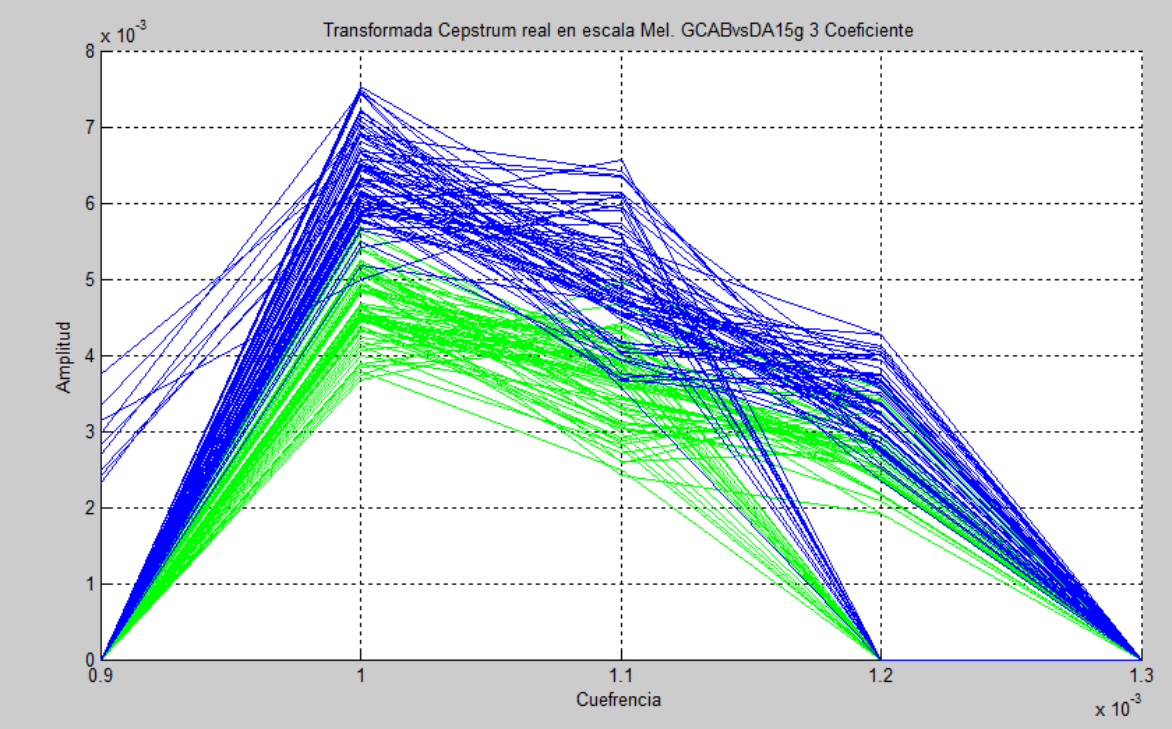


Fuente: (Autor).

En la Figura 54 se grafica el tercer CCM para la comparación GCABvsDA15g. Este es el coeficiente encontrado en el cual se diferencian las variables relacionadas; se puede ver una pequeña dispersión en la forma de los datos en cada una de las variables. También se puede ver que la variable DA15g en general posee datos por debajo de GCAB y en general existe diferenciación de las variables por visualización.

Después de revisar la Figura 52, Figura 53 y la Figura 54; correspondientes a las relaciones del grupo desalineado con el grupo de referencia, graficadas con el fin de encontrar diferencias en las relaciones por visualización; se puede concluir que por visualización solo se diferencia la variable DA15g, las otras dos variables presentan pequeñas diferencias pero por visualización no se puede concluir nada al respecto.

Figura 54. Coeficiente No 3 del CCM para la comparación GCABvsDA15g.



Fuente: (Autor).

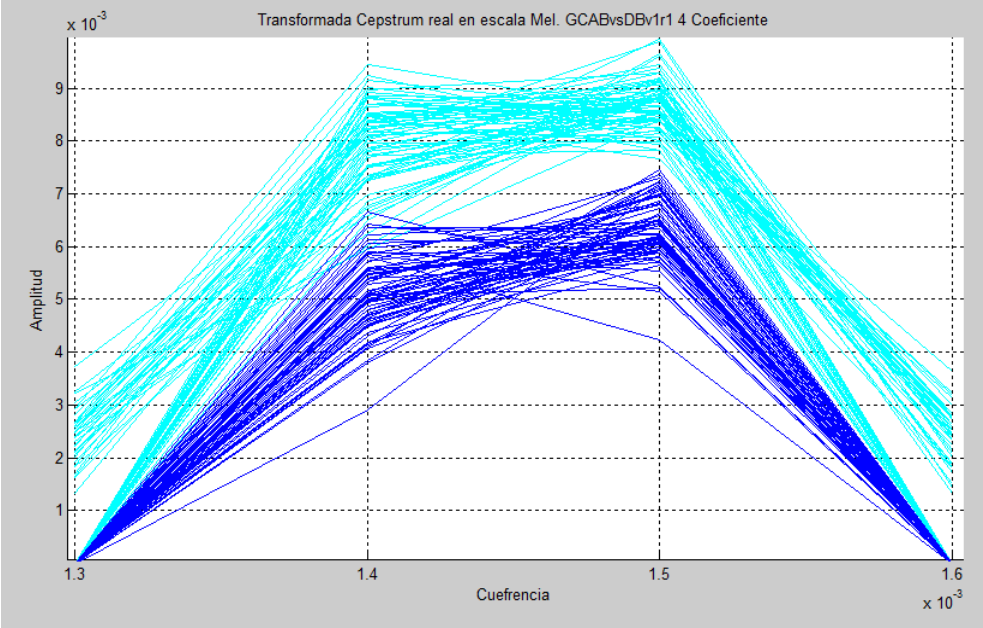
Coeficientes para el grupo desbalanceado.

En esta sección se presentan los coeficientes del CCM para las variables de grupo desbalanceado en referencia con la variable GCAB; con el fin de visualizar los coeficientes que diferencian a cada una de las variables comparadas en este grupo.

En la Figura 55 se grafica el cuarto CCM para la comparación GCABvsDBv1r1. Este es el coeficiente encontrado en el cual se diferencian las variables relacionadas; se puede ver una pequeña dispersión en la forma de los datos en cada una de las variables. También se puede ver que la variable DBv1r1 posee datos por encima de GCAB y en general existe diferenciación de las variables por visualización.

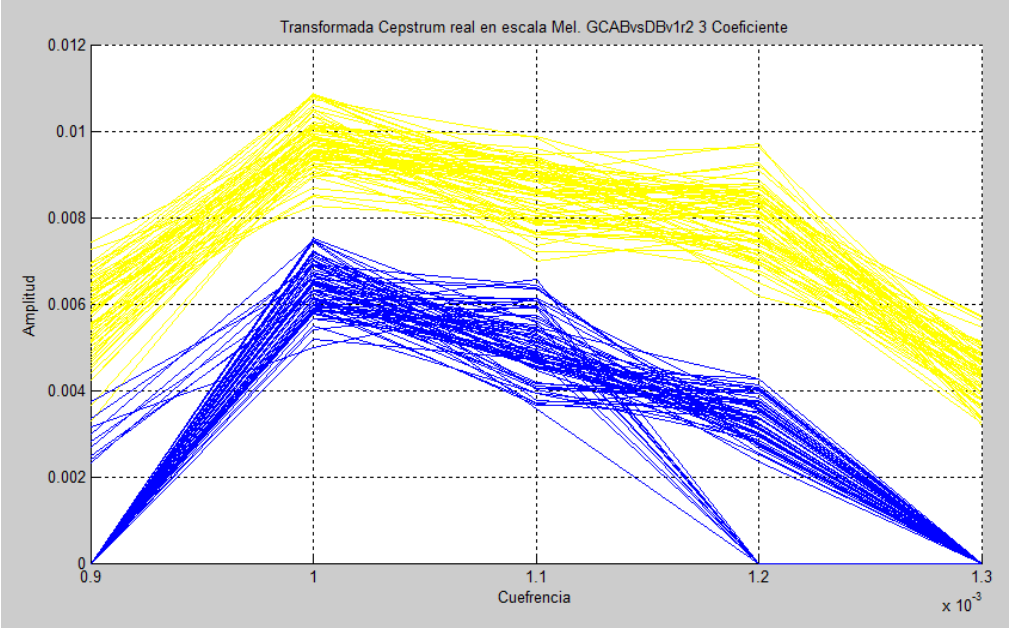
En la Figura 56 se grafica el tercer CCM para la comparación GCABvsDBv1r2. Este es el coeficiente encontrado en el cual se diferencian las variables relacionadas; se puede ver una pequeña dispersión en la forma de los datos en cada una de las variables. También se puede ver que la variable DBv1r2 posee datos por encima de GCAB y en general existe diferenciación de las variables por visualización.

Figura 55. Coeficiente No 4 del CCM para la comparación GCABvsDBv1r1.



Fuente: (Autor).

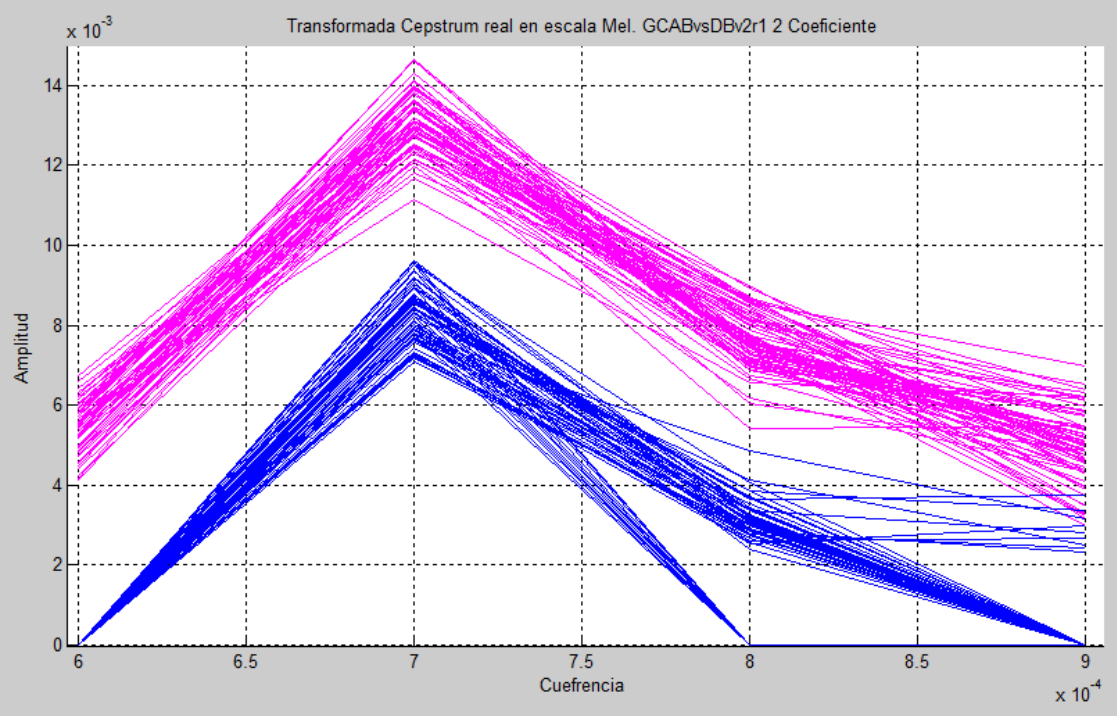
Figura 56. Coeficiente No 3 del CCM para la comparación GCABvsDBv1r2.



Fuente: (Autor).

En la Figura 57 se grafica el segundo CCM para la comparación GCABvsDBv2r1. Este es el coeficiente encontrado en el cual se diferencian las variables relacionadas; se puede ver una pequeña dispersión en la forma de los datos en cada una de las variables. También se puede ver que la variable DBv2r1 posee datos por encima de GCAB y en general existe diferenciación de las variables por visualización.

Figura 57. Coeficiente No 2 del CCM para la comparación GCABvsDBv2r1.

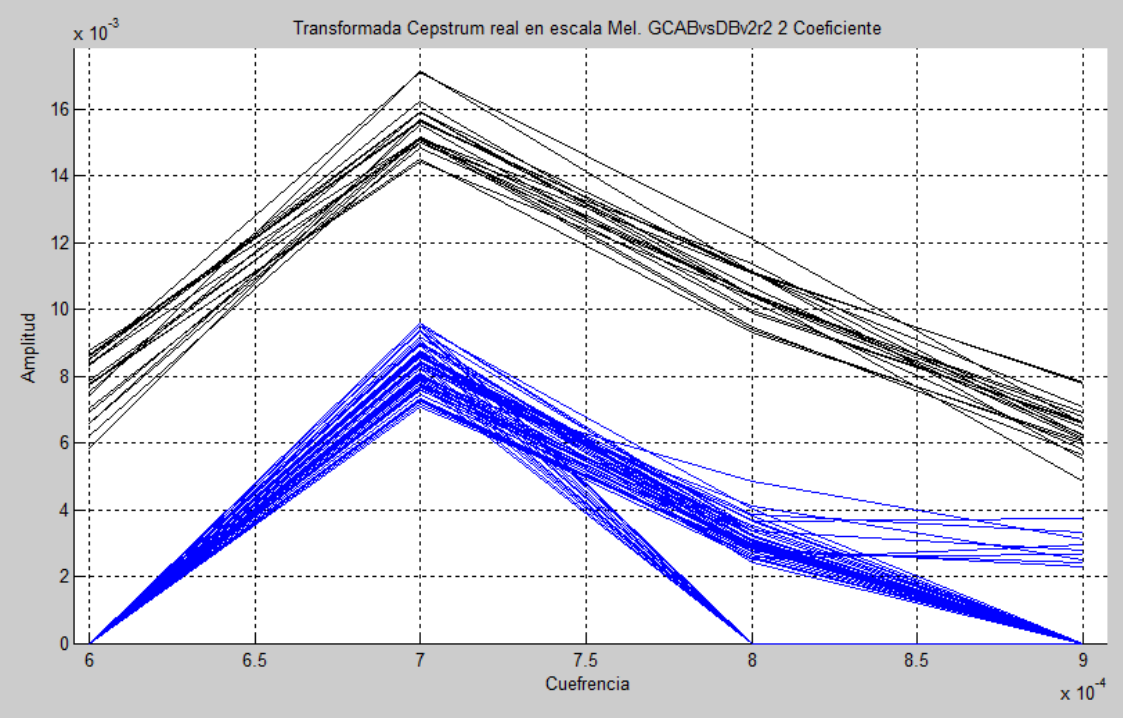


Fuente: (Autor).

En la Figura 58 se grafica el segundo CCM para la comparación GCABvsDBv2r2. Este es el coeficiente encontrado en el cual se diferencian las variables relacionadas; se puede ver una pequeña dispersión en la forma de los datos en cada una de las variables. También se puede ver que la variable DBv2r2 posee datos por encima de GCAB y en general existe diferenciación de las variables por visualización.

Después de revisar la Figura 55, Figura 56, Figura 57 y la Figura 58; correspondientes a las relaciones del grupo desbalanceado con el grupo de referencia, graficadas con el fin de encontrar diferencias en las relaciones por visualización; se puede concluir que para todas las variables del grupo desbalanceado existe diferenciación por visualización respecto al grupo de referencia.

Figura 58. Coeficiente No 2 del CCM para la comparación GCABvsDBv2r2.



Fuente: (Autor).

Después de revisar gráficamente cada una de las variables relacionadas en los CCM respecto al grupo de referencia GCAB se puede concluir que existen 5 relaciones que se pueden diferenciar por visualización y 2 relaciones que no se pueden diferenciar por visualización.

Es necesario explicar que los CCM encontrados para diferenciar cada una de las variables respecto al grupo de referencia tiene un 100% de diferenciación por visualización para el grupo de desbalanceo y un 33.33% de diferenciación por visualización para el grupo de desalineación. Esto quiere decir que en general los CCM representan un 71.43% de diferenciación por visualización para todas las relaciones establecidas en la Tabla 29.

Debido a lo anterior se puede concluir que la diferenciación por visualización es un buen método para diferenciar las variables relacionadas en la Tabla 29, pero no representa un método valido para este proyecto; por lo que, el método de diferenciación por visualización representa para este proyecto, una referencia para comenzar a realizar un análisis respecto a la diferenciación por medio de la distancia euclidiana; la cual hace parte del tercer objetivo específico de este proyecto 1.3.2.3.

4.3. Distancia euclidiana.

En esta sección se presentan los resultados obtenidos aplicando el algoritmo para la distancia euclidiana de la sección 3.3.2.5, en el cual se aplicó la ecuación (19) de la sección 2.1.11; para la diferenciación de las variables relacionadas en la Tabla 29, con el fin de determinar si los CCM encontrados para cada relación, se pueden diferenciar por medio de la distancia euclidiana.

En la Tabla 30 se presentan los datos obtenidos para cada grupo de relación respecto a su distancia euclidiana. Revisando los datos obtenidos se puede concluir que la distancia euclidiana, permite diferenciar todas las variables que estén relacionadas con el grupo de referencia, según sea el coeficiente analizado.

También se puede concluir que para las variables del grupo desalineado existe diferencia por medio de la distancia euclidiana en un 100% y en las variables del grupo desbalanceado existe diferencia por medio de la distancia euclidiana en un 100%; por lo que en general la distancia euclidiana permite diferenciar todas las variables relacionadas en un 100%.

Tabla 30. Distancia euclidiana de las variables relacionadas en la Tabla 29.

No Variable para la DE	**Relación**	**No CCM**	**DE * e-3**
CEPDA1	GCABvsDA05g	1	3.93322
CEPDA2	GCABvsDA10g	1	4.82292
CEPDA3	GCABvsDA15g	3	10.94881
CEPD1	GCABvsDBv1r1	4	11.80011
CEPD2	GCABvsDBv1r2	3	16.63780
CEPD3	GCABvsDBv2r1	2	17.28532
CEPD4	GCABvsDBv2r2	2	32.77406

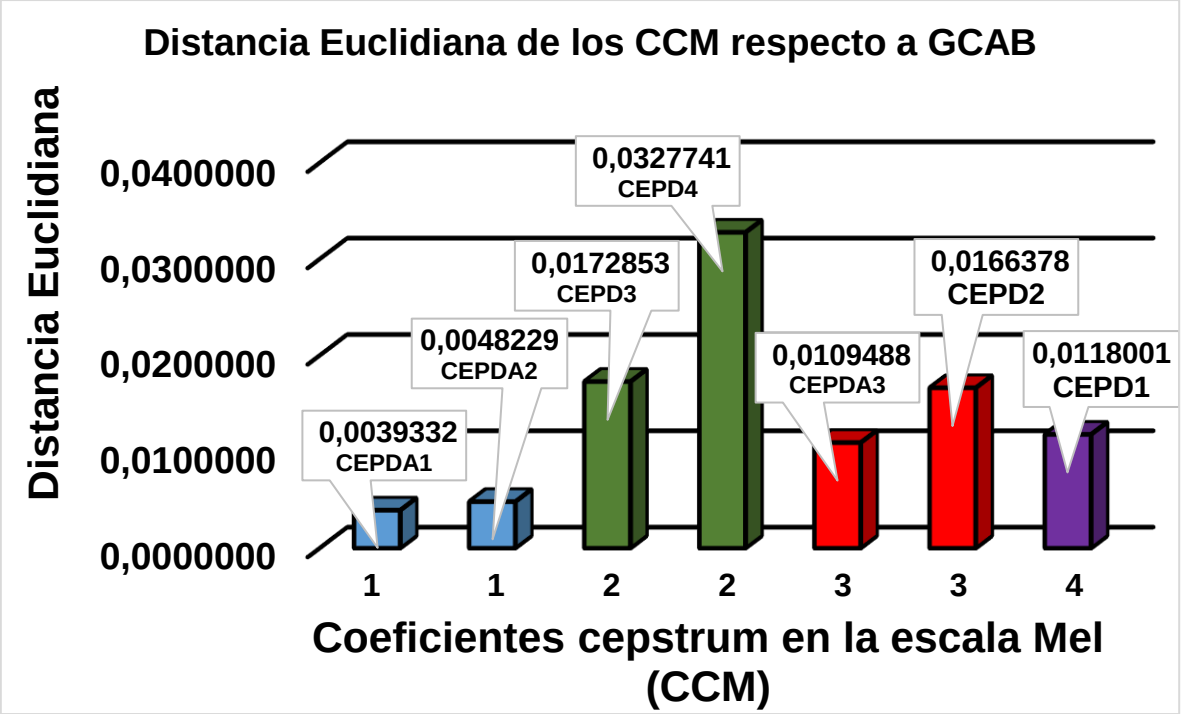
DE: Distancia Euclidiana.
CCM: Coeficiente cepstrum en la escala Mel.

Fuente: (Autor).

En la Figura 59 se graficaron las variables de la distancia euclidiana respecto a cada CCM; con el fin de visualizar de forma gráfica la diferencia encontrada por medio de la distancia euclidiana.

Después de revisar detenidamente la Figura 59 se puede concluir que la distancia euclidiana permite diferenciar en un 100% cada una de las variables relacionadas columna 2 Tabla 30, según el coeficiente de análisis de los CCM.

Figura 59. Distancia euclidiana de los CCM.



Fuente: (Autor).

4.4. Caracterización.

En esta sección se presentan los resultados obtenidos aplicando el algoritmo para la caracterización de los CCM por medio del análisis estadístico de la sección 3.3.2.6, en el cual se aplicaron las ecuaciones de la sección 2.1.13.

En la Tabla 31 se presentan los valores estadísticos obtenidos para cada variable después de aplicar el algoritmo $EstadisticosCuefrencia.m$ Tabla 24.

Después de revisar cada uno de los estadísticos calculados se puede llegar a concluir que el valor medio permite diferenciar cada una de las variables de la Tabla 31, donde las variables correspondientes al grupo desalineado se encuentran por debajo del grupo de referencia y las variables del grupo desbalanceado se encuentran por encima del grupo de referencia.

La variable estadística de desviación estándar E2 no proporciona mucha información por sí sola, pero si se utiliza el coeficiente de variación se puede concluir que las variables en general de la Tabla 31, no superan el 13% del coeficiente de variación; sin embargo la variable DBv1r1 supera este valor y se ubica en un coeficiente de variación aproximado del

20%; por lo que en general todas las variables no tienen una desviación estándar demasiado grande.

Tabla 31. Valores estadísticos para cada variable

Variable	Estadísticos						
	E1 * e-4	E2 *e-4	CV %	E3 *e-4	E4	E5	E6
GCAB	55.3963	6.8916	12,44	55.9504	1,3032E-23	2,3734E-18	1,00918
DA05g	52.2840	3.2627	6,24	52.3842	3,8258E-28	-1,3172E-22	1,00192
DA10g	52.9456	4.3471	8,21	53.1211	6,1306E-27	7,4302E-21	1,00332
DA15g	38.4259	3.6330	9,45	38.5947	8,0445E-28	8,6878E-22	1,00439
DBv1r1m	62.8185	12.5058	19,91	64.0333	1,4672E-23	3,5653E-18	1,01934
DBv1r2m	69.4527	2.8814	4,15	69.5116	1,437E-28	2,1541E-22	1,00085
DBv2r1m	77.7885	3.7489	4,82	77.8775	9,2955E-28	-1,0494E-21	1,00114
DBv2r2m	100.5636	5.0916	5,06	100.6905	9,9209E-27	8,2741E-21	1,00126

E1: Valor medio
E2: Desviación Estándar
E3: RMS Raíz media cuadrática
E4: Curtosis
E5: Asimetría
E6: Factor de forma
CV: Coeficiente de variación

Fuente: (Autor).

La variable estadística de raíz media cuadrática E3 permite diferenciar cada una de las variables de la Tabla 31, donde las variables correspondientes al grupo desalineado se encuentran por encima y por debajo del grupo de referencia y las variables del grupo desbalanceado se encuentran por encima del grupo de referencia.

La variable estadística de Curtosis E4 según la sección 2.1.13, donde la curva de referencia es una campana gaussiana de $E4 = 0$ Figura 2; se puede llegar a concluir que todas las variables de la Tabla 31 son leptocúrticas es decir tiene una Curtosis $E4 > 0$.

La variable estadística de asimetría E5 según la sección 2.1.13, permite definir que una variable es simétrica cuando su valor numérico es $E5 = 0$. Para las variables de la Tabla 31 se puede decir que en general todas las variables son asimétricas con un valor muy cercano a cero.

Según la sección 2.1.13 las variables DA05g y DBv2r1 son asimétricas negativas es decir los datos se encuentran más agrupados por debajo del valor medio. Las demás variables son asimétricas positivas es decir los datos se encuentran más agrupados por encima del valor medio.

La variable estadística de factor de forma E6 según la sección 2.1.13, permite definir que una variable tiene una forma definida cuando $E6 = 1$. Para las variables de la Tabla 31 se puede llegar a concluir que las variables en general, tienen una forma definida muy cercana a uno.

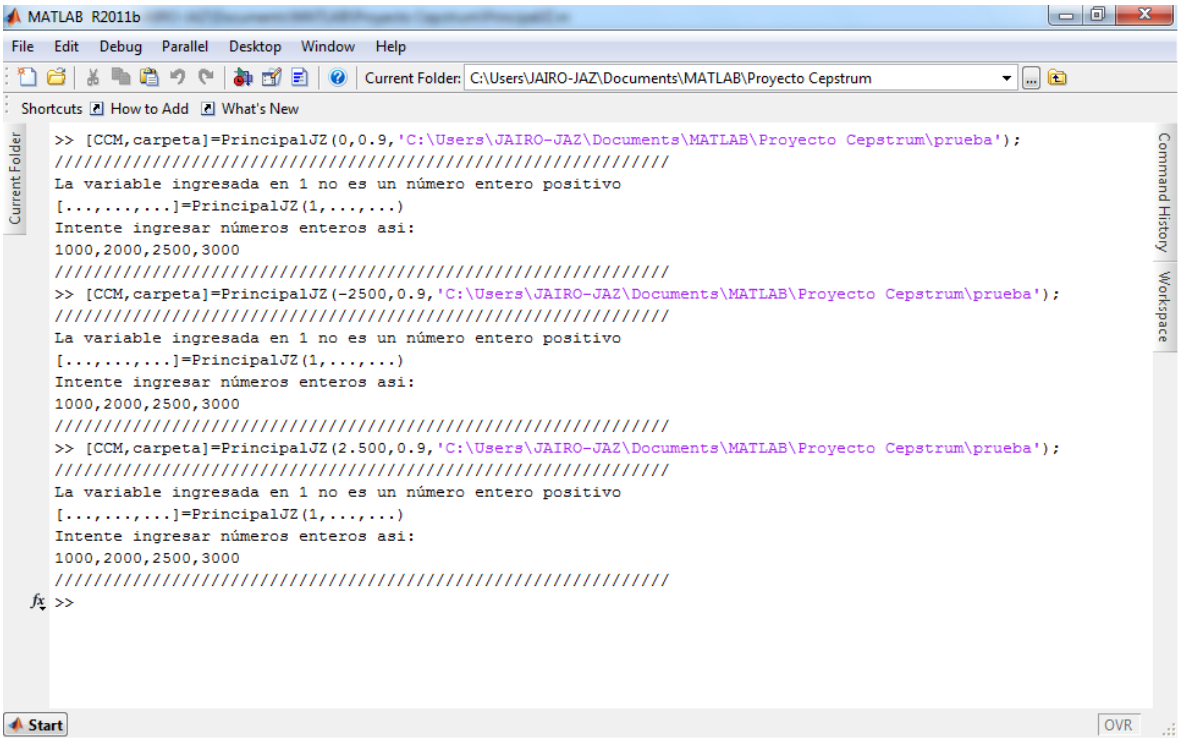
4.5. Algoritmo principal.

En esta sección se evalúan todas las condiciones previstas en este proyecto para el algoritmo de integración llamado $\boldsymbol{PrincipalJZ.m}$ de la sección 3.3.2.7.

Errores al ejecutar $\boldsymbol{PrincipalJZ.m}$.
En esta sección se plantean algunos errores en los cuales puede incurrir el usuario; debido a una selección inadecuada de la variable de entrada, tipológico de Matlab, de dimensión y de orden numérico.

En la Figura 60 se puede ver tres llamados realizados a la función $\boldsymbol{PrincipalJZ.m}$ aplicando un error de variable al argumento de entrada num. En el primer error se ingresó un $num = 0$, donde se le indica al usuario por medio de una impresión en el $\boldsymbol{Command\ Windows}$; que existe un error en la variable de entrada 1 debido a que no es un número entero positivo.

Figura 60. Errores para la variable num.



Fuente: (Autor).

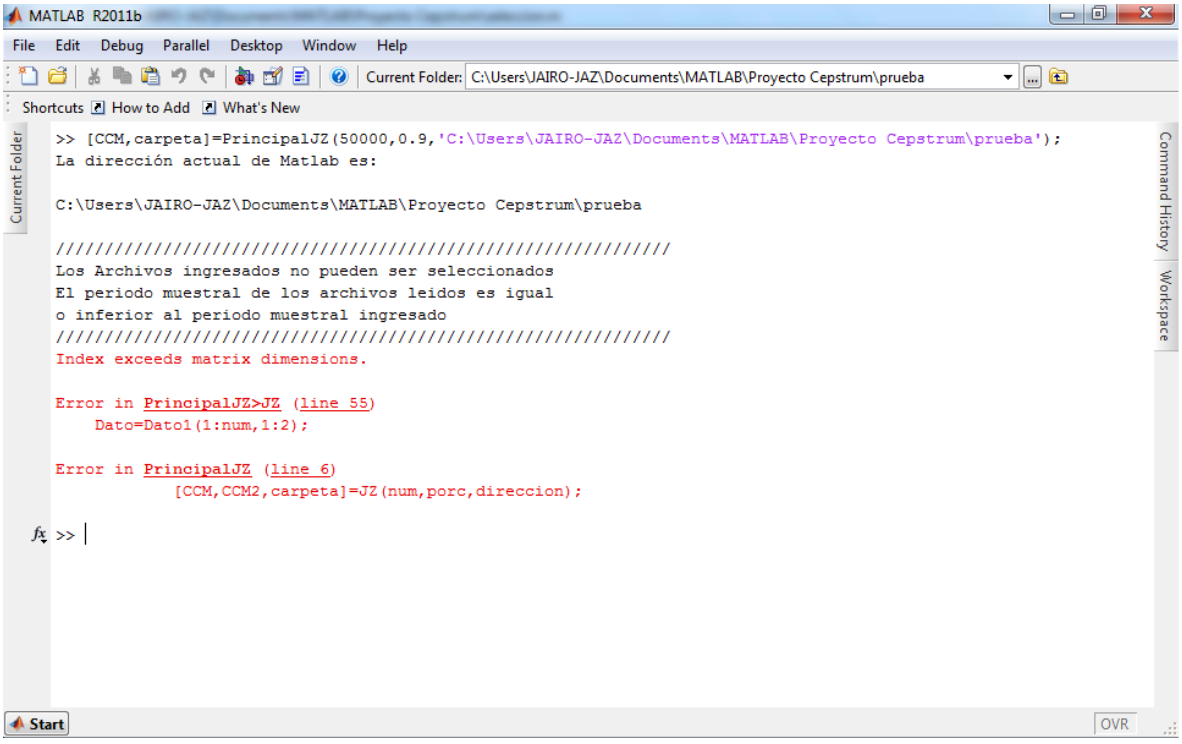
En el segundo error se ingresó un $num = -2500$, donde se le indica al usuario por medio de una impresión en el $\boldsymbol{Command\ Windows}$; que existe un error en la variable de entrada 1 debido a que no es un número entero positivo.

En el tercer error se ingresó un $num = 2.500$, donde se le indica al usuario por medio de una impresión en el ***Command Windows***; que existe un error en la variable de entrada 1 debido a que no es un número entero positivo.

Para los tres casos previstos en este proyecto respecto a la variable de entrada num, se ingresó por el programador un número que no era entero y tampoco era positivo; por lo que no se pudo comenzar a ejecutar el programa. Es necesario explicar que el usuario puede cometer otro tipo de error no previsto por el programador; en todos los casos que pueda existir, Matlab arrojara un error para la condición que este fallando debido a la variable num.

En la Figura 61 se puede ver un error particular para el argumento de entra num al ejecutar la función ***PrincipalJZ.m***. En este caso se ingresó $num = 50000$; el cual excede el tamaño de los archivos sin seleccionar; por lo que se imprime un error por el programa el cual explica que el periodo muestral seleccionado, es decir num, excede el tamaño del primer archivo leído. También se puede ver en rojo un error de dimensión arrojado por Matlab el cual obliga al programa a salir de su ejecución.

Figura 61. Error por una selección inadecuada de la variable num.



Fuente: (Autor).

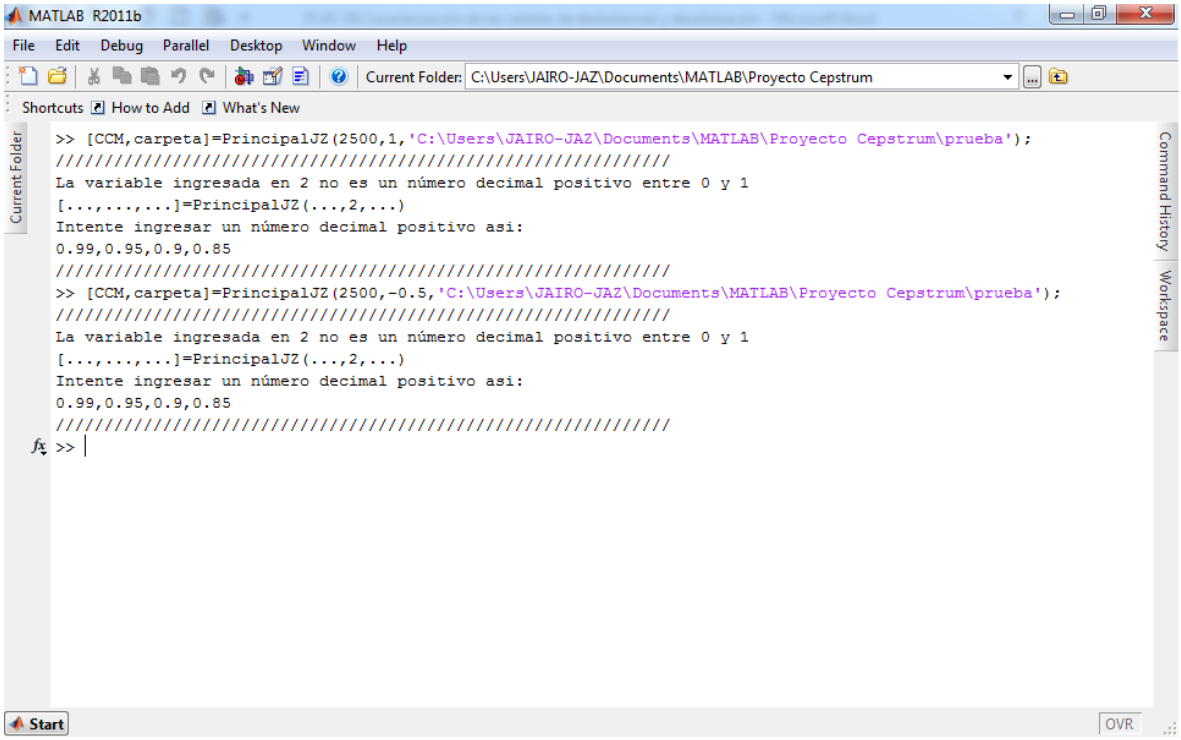
En la Figura 62 se puede ver dos llamados realizados a la función ***PrincipalJZ.m*** aplicando un error de variable al argumento de entrada $porc$. En el primer error se ingresó $porc = 1$, donde se le indica al usuario por medio de una impresión en el

Command Windows; que existe un error en la variable de entrada 2 debido a que no es un número decimal positivo.

En el segundo error se ingresó $porc = -0.5$, donde se le indica al usuario por medio de una impresión en el ***Command Windows***; que existe un error en la variable de entrada 2 debido a que no es un número decimal positivo.

Para los dos casos previstos en este proyecto respecto a la variable de entrada $porc$, se ingresó por el programador un número que no era decimal y tampoco era positivo; por lo que no se pudo comenzar a ejecutar el programa. Es necesario explicar que el usuario puede cometer otro tipo de error no previsto por el programador; en todos los casos que puedan existir, Matlab arrojara un error para la condición que este fallando debido a la variable $porc$.

Figura 62. Errores para la variable $porc$.



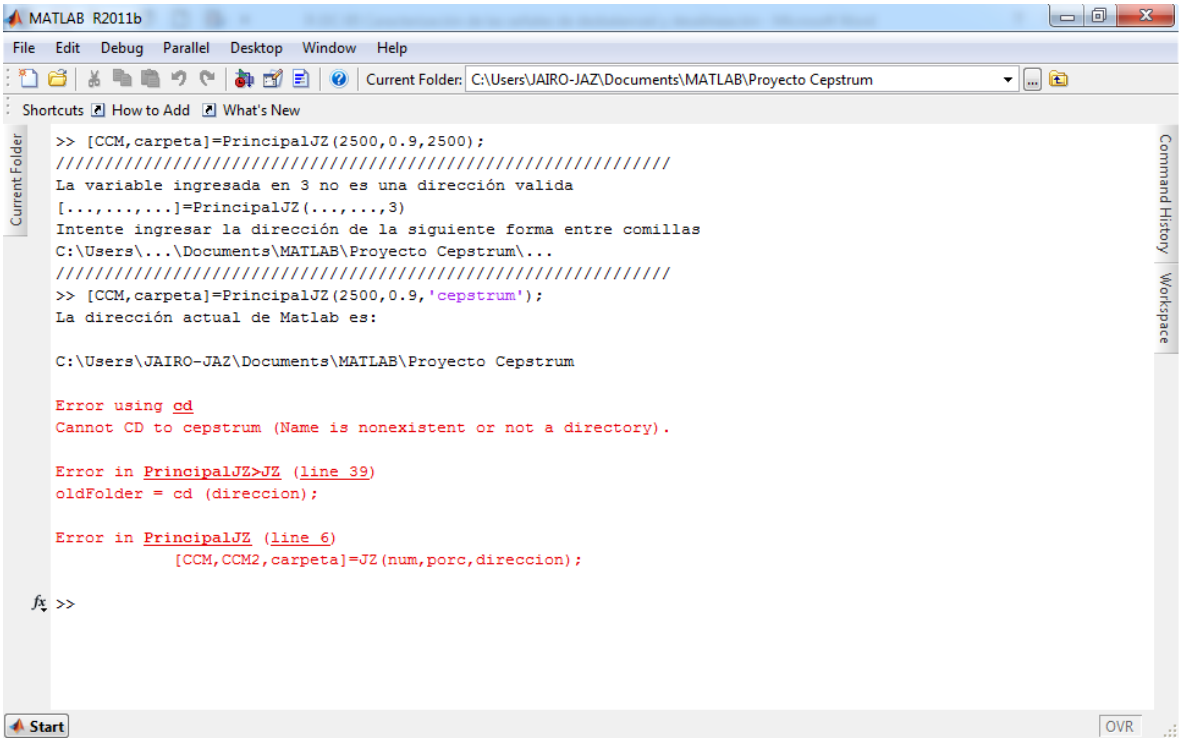
Fuente: (Autor).

En la Figura 63 se puede ver dos llamados realizados a la función ***PrincipalJZ.m*** aplicando un error de variable al argumento de entrada $direccion$. En el primer error se ingresó $direccion = 2500$, donde se le indica al usuario por medio de una impresión en el ***Command Windows***; que existe un error en la variable de entrada 3 debido a que no es una dirección valida en el entorno de Matlab.

En el segundo error se ingresó $direccion = cepstrum$, donde el programa se ejecuta pero inmediatamente Matlab en él ***Command Windows***; informa en letra roja que la dirección que se suponía era válida, realmente no es una dirección válida para Matlab.

Para los dos casos previstos en este proyecto respecto a la variable de entrada $direccion$, se ingresó por el programador un número entero que no era una dirección válida y un vector de caracteres que tampoco era una dirección valida en el entorno de Matlab; por lo que no se pudo ejecutar el programa. Es necesario explicar que el usuario puede cometer otro tipo de error no previsto por el programador; en todos los casos que pueda existir, Matlab arrojara un error para la condición que este fallando debido a la variable $direccion$.

Figura 63. Errores para la variable dirección.



Fuente: (Autor).

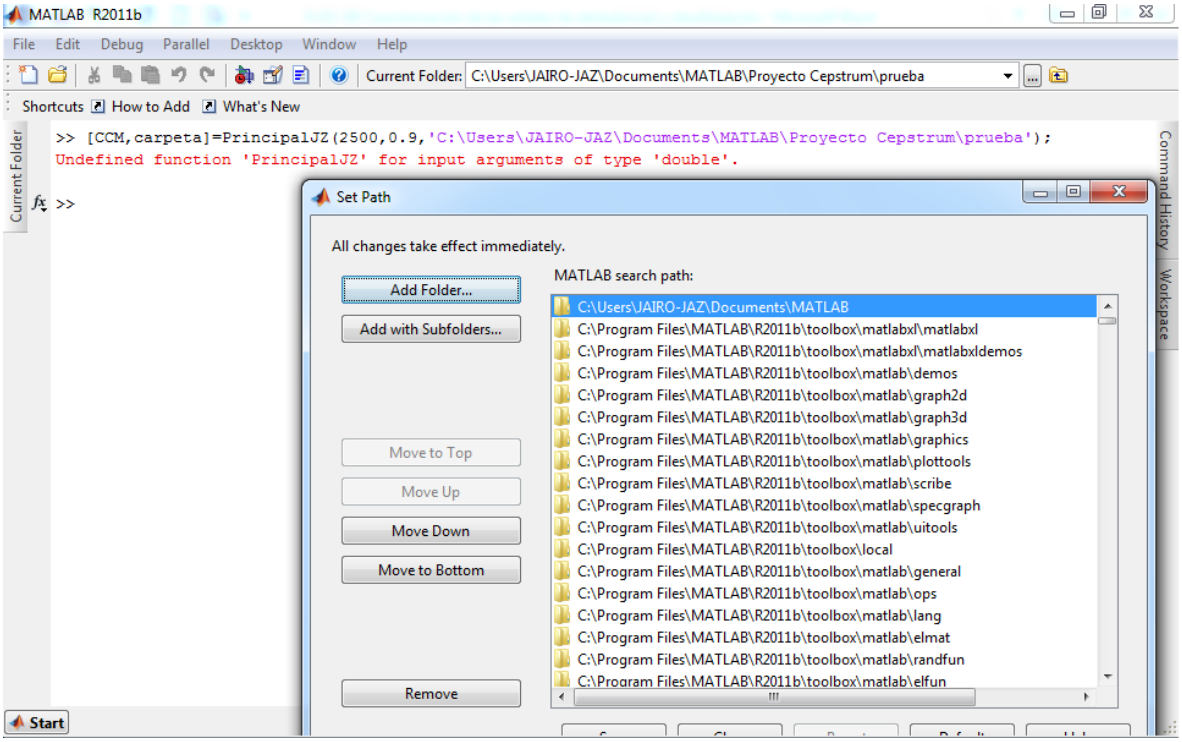
En la Figura 64 se puede ver un error particular para la función ***PrincipalJZ.m***. En este caso el error lo arroja Matlab en el ***Command Windows***, donde indica al usuario que la función ***PrincipalJZ.m*** no existe en el entorno de Matlab. Este error se puede solucionar realizando los siguientes pasos.

En la parte superior derecha de la ventana de Matlab, seleccionar $file$ dentro del menú de $file$ buscar la carpeta $set\ path$; la cual despliega una ventana como en la Figura 64. Dentro de la ventana buscar el botón $Add\ with\ subfolders$, donde se despliega una ventana en la cual se puede buscar la ubicación donde se encuentra la carpeta *proyecto cepstrum;* la cual

contiene todas las funciones necesarias para ejecutar el algoritmo $\boldsymbol{PrincipalJZ.m}$. Después de seleccionar la ubicación de la carpeta dentro de la ventana de $set\ path$ dar clic en el botón $save$.

Realizando los pasos mencionados, se soluciona el problema referido por Matlab respecto al algoritmo $\boldsymbol{PrincipalJZ.m}$.

Figura 64. Error de instalación de la carpeta proyecto cepstrum.

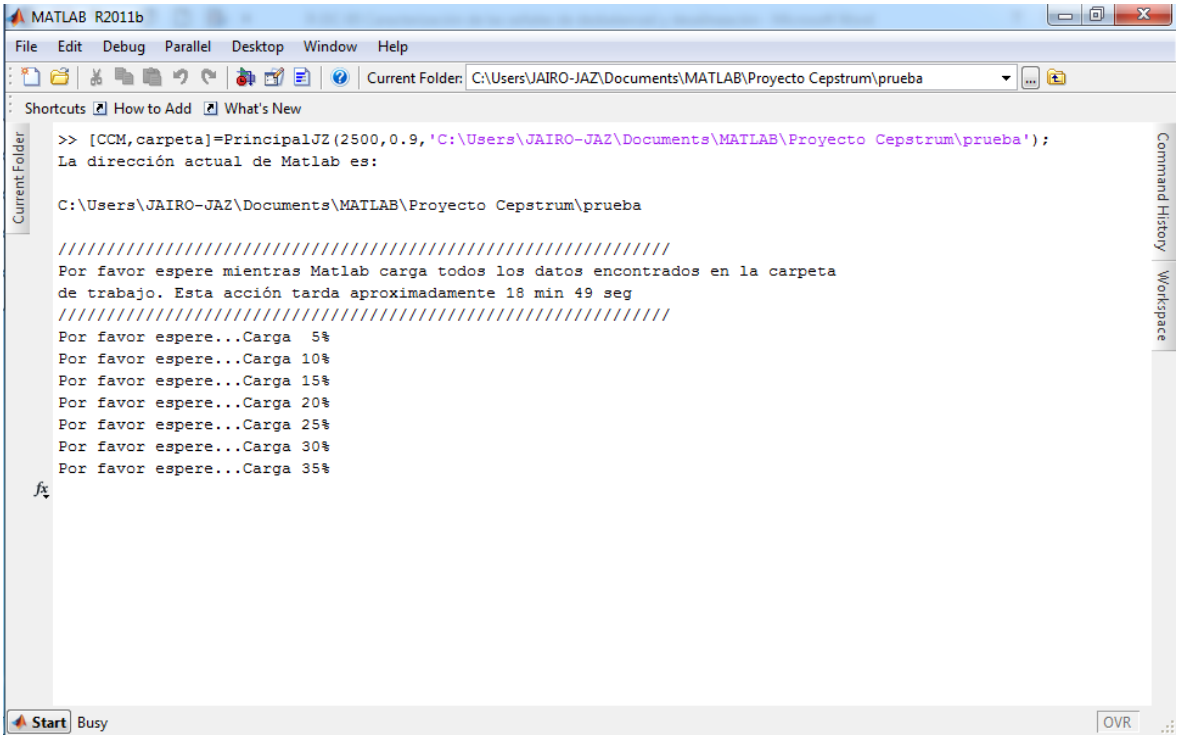


Fuente: (Autor).

Funcionamiento del algoritmo $\boldsymbol{PrincipalJZ.m}$.
En esta sección se describen los procesos visibles en el $\boldsymbol{Command\ Windows}$, cuándo el algoritmo $\boldsymbol{PrincipalJZ.m}$ se ejecuta de forma correcta dentro del entorno de Matlab. Es importante aclarar que el tiempo aproximado para ejecutar todo el programa es de $28\ min$; tiempo que puede variar según el procesador que posee el computador donde sea ejecutado el algoritmo. Este tiempo es relativamente bueno debido a que este proyecto tiene 8 variables que suman 544 archivos con un tamaño promedio de 1 MB por archivo.

En la Figura 65 se puede ver la interacción de la función complementaria $carga.m$ con el usuario; donde esta función es llamada por el algoritmo de selección; se puede llegar a esta conclusión porque de todos los procesos que realiza el programa, el que demanda un mayor tiempo de ejecución es el debido a la lectura de los archivos originales.

Figura 65. Lectura del algoritmo $\boldsymbol{PrincipalJZ.m}$ para seleccionar las muestras.



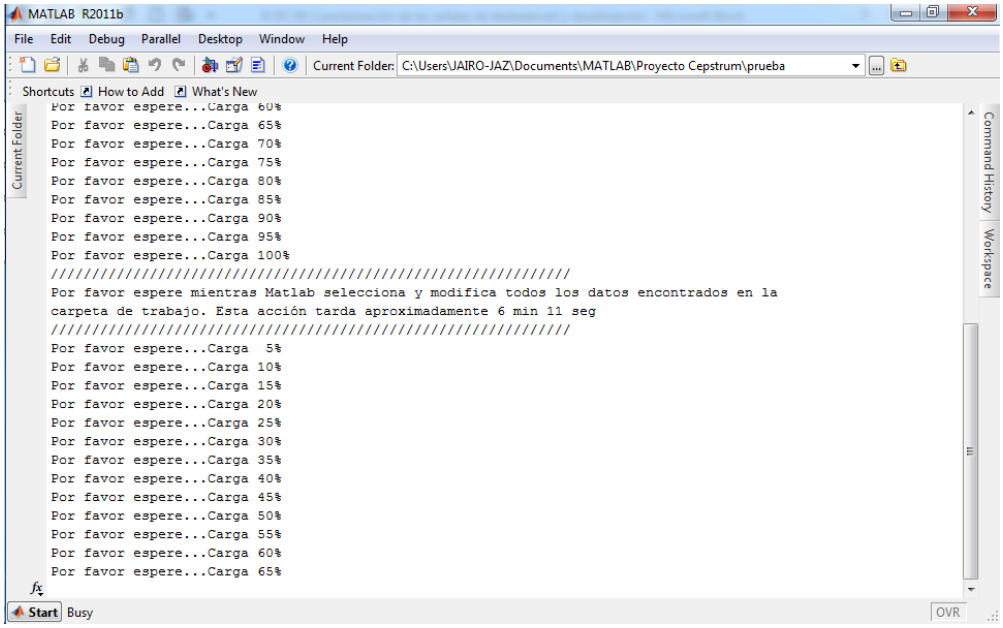
Fuente: (Autor).

En la Figura 66 se puede ver la interacción de la función complementaria $carga.m$, con el usuario donde esta función es llamada por la función $Seleccionmuestras.m$; se puede llegar a esta conclusión por el mensaje de inicio del proceso; en el cual indica que el programa está realizando la selección y modificación de cada uno de los archivos encontrados.

En la Figura 67 se puede ver la interacción de la función complementaria $carga.m$, con el usuario donde esta función es llamada por la sub-función JZ del algoritmo $\boldsymbol{PrincipalJZ.m}$; se puede llegar a esta conclusión por el tiempo que demora la lectura de los datos encontrados por Matlab, debido a que el tiempo de lectura de las muestras que se encuentren modificadas es muy rápido; sin importar el tamaño de la carpeta donde se encuentren localizados.

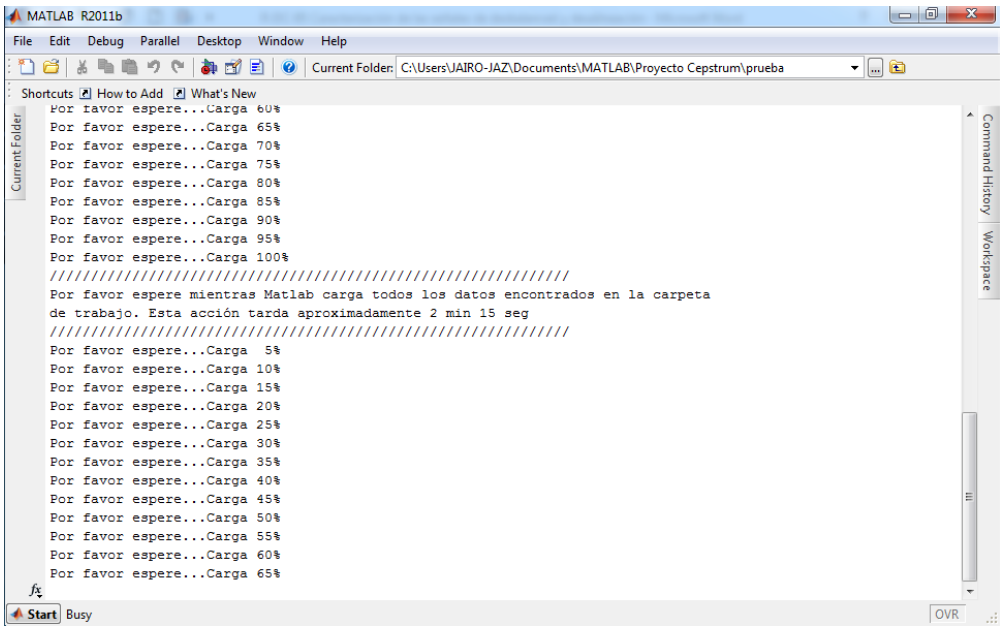
En la Figura 68 y la Figura 69; se puede ver en la parte inferior el mensaje de finalización del algoritmo $\boldsymbol{PrincipalJZ.m}$. También se puede ver en la Figura 68 una ventana llamada variable editor en la cual aparece la variable CCM con su matriz de $8x3$ del tipo estructura; y en la Figura 69 se puede ver una ventana llamada variable editor en la cual aparece la variable $carpeta$ ubicada en la parte final del vector tipo $char$.

Figura 66. Modificación de los archivos por las muestras seleccionadas.



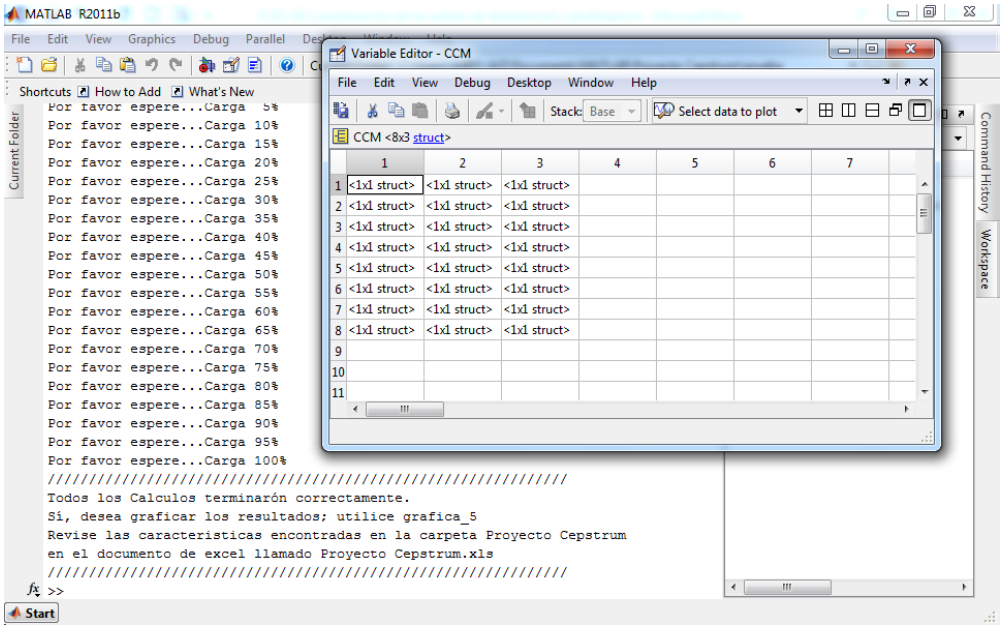
Fuente: (Autor).

Figura 67. Lectura de las muestras seleccionadas para realizar transformada cepstrum, distancia euclidiana y caracterización.



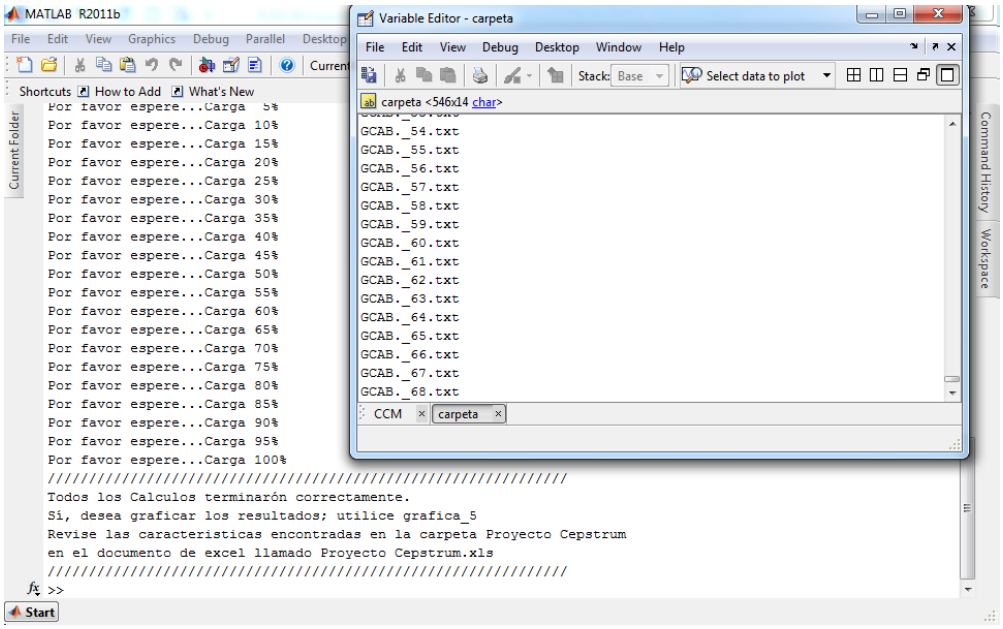
Fuente: (Autor).

Figura 68. Algoritmo $PrincipalJZ.m$ terminado y variable CCM.



Fuente: (Autor).

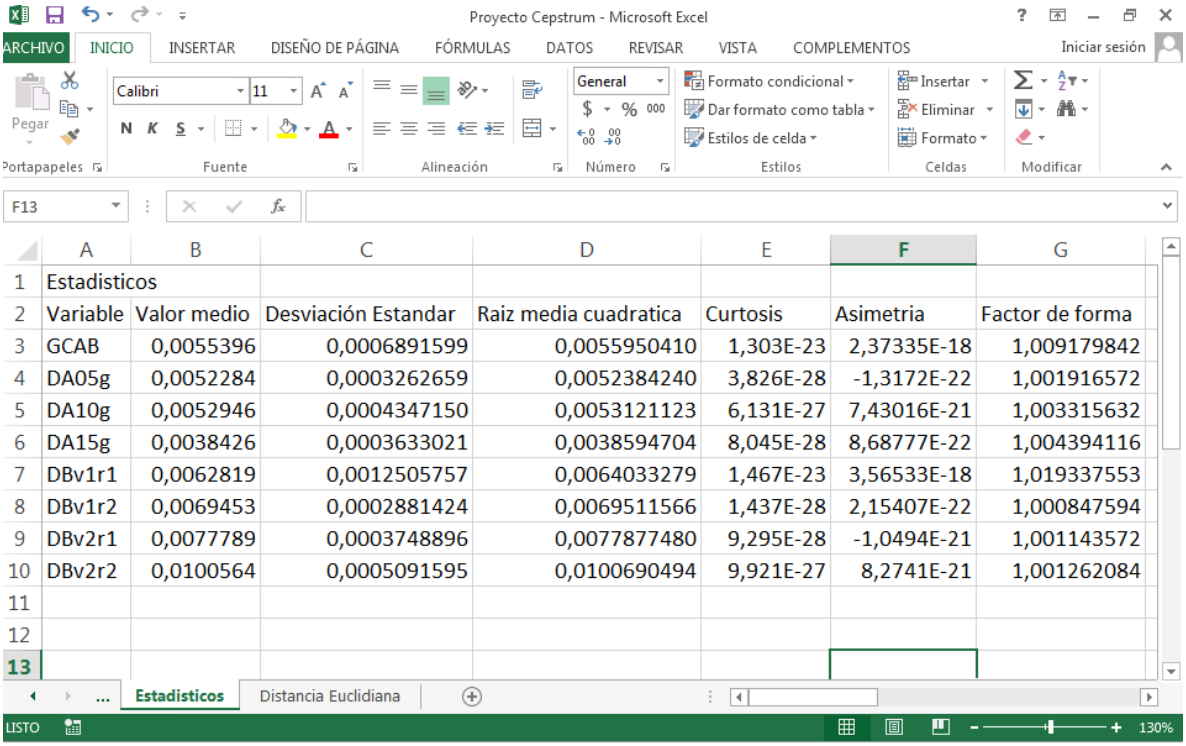
Figura 69. Algoritmo $PrincipalJZ.m$ terminado y variable $carpeta$.



Fuente: (Autor).

En la Figura 70 se puede ver la finalización de la sección de impresión de la sub-función JZ del algoritmo $\boldsymbol{PrincipalJZ.m}$; en la cual se crea un archivo de Excel llamado Proyecto Cepstrum. En esta figura se puede ver que se creó el libro llamado estadísticos en el cual se imprimen los resultados de la caracterización de las variables según el coeficiente del CCM.

Figura 70. Archivo de Excel y libro de características estadísticas.

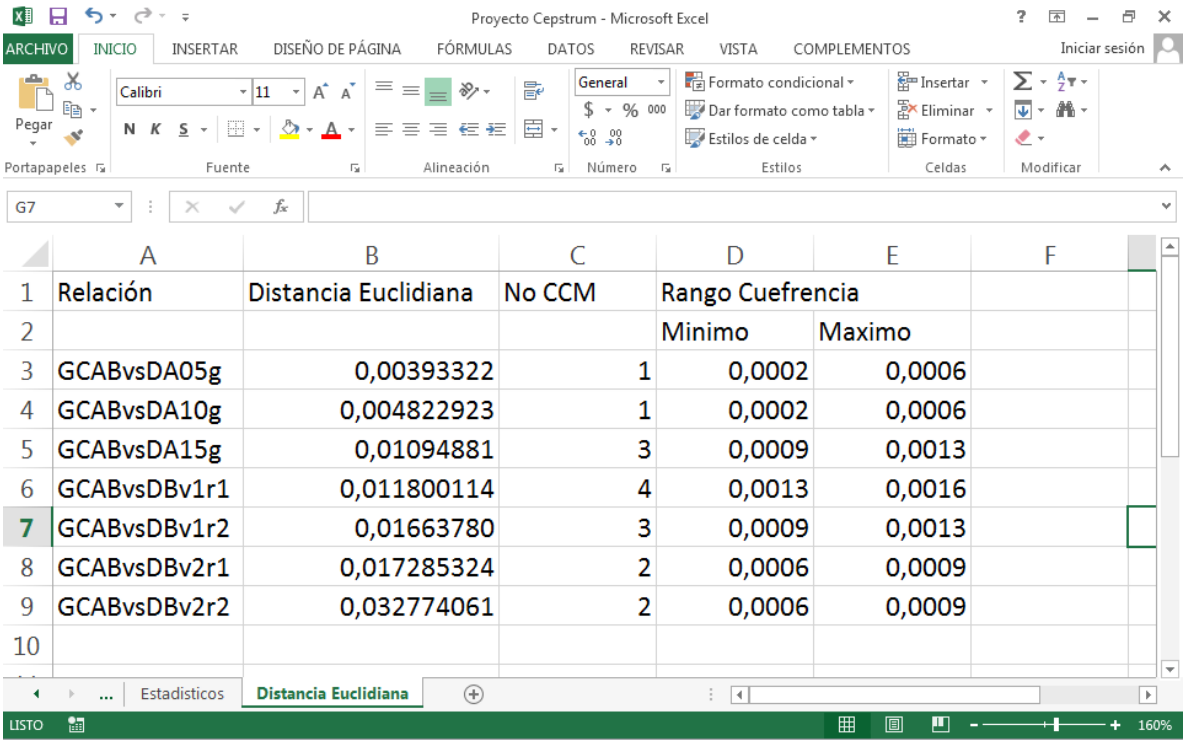


	A	B	C	D	E	F	G
1	Estadisticos						
2	Variable	Valor medio	Desviación Estandar	Raiz media cuadratica	Curtosis	Asimetria	Factor de forma
3	GCAB	0,0055396	0,0006891599	0,0055950410	1,303E-23	2,37335E-18	1,009179842
4	DA05g	0,0052284	0,0003262659	0,0052384240	3,826E-28	-1,3172E-22	1,001916572
5	DA10g	0,0052946	0,0004347150	0,0053121123	6,131E-27	7,43016E-21	1,003315632
6	DA15g	0,0038426	0,0003633021	0,0038594704	8,045E-28	8,68777E-22	1,004394116
7	DBv1r1	0,0062819	0,0012505757	0,0064033279	1,467E-23	3,56533E-18	1,019337553
8	DBv1r2	0,0069453	0,0002881424	0,0069511566	1,437E-28	2,15407E-22	1,000847594
9	DBv2r1	0,0077789	0,0003748896	0,0077877480	9,295E-28	-1,0494E-21	1,001143572
10	DBv2r2	0,0100564	0,0005091595	0,0100690494	9,921E-27	8,2741E-21	1,001262084
11							
12							
13							

Fuente: (Autor).

En la Figura 71 se puede ver la finalización de la sección de impresión de la sub-función JZ del algoritmo $\boldsymbol{PrincipalJZ.m}$; en la cual se crea un archivo de Excel llamado Proyecto Cepstrum. En esta figura se puede ver que se creó el libro llamado distancia euclidiana en el cual se imprimen los resultados de la distancia euclidiana respecto a la pareja de variables relacionada con el grupo de referencia.

Figura 71. Archivo de Excel y libro de distancia euclidiana.



	A	B	C	D	E	F
1	Relación	Distancia Euclidiana	No CCM	Rango Cuefrencia		
2				Minimo	Maximo	
3	GCABvsDA05g	0,00393322	1	0,0002	0,0006	
4	GCABvsDA10g	0,004822923	1	0,0002	0,0006	
5	GCABvsDA15g	0,01094881	3	0,0009	0,0013	
6	GCABvsDBv1r1	0,011800114	4	0,0013	0,0016	
7	GCABvsDBv1r2	0,01663780	3	0,0009	0,0013	
8	GCABvsDBv2r1	0,017285324	2	0,0006	0,0009	
9	GCABvsDBv2r2	0,032774061	2	0,0006	0,0009	
10						

Fuente: (Autor).

Después de realizar un análisis de los posibles errores que pueden presentarse en la ejecución del algoritmo $PrinciaplJZ.m$ y de los procesos que se pueden ver por medio del $Command\ Windows$; se puede llegar a concluir que el algoritmo de integración de todos los procesos relacionados con este proyecto; los realiza de acuerdo a lo establecido en la sección 3.3.2.7 por lo cual el algoritmo $PrinciaplJZ.m$ cumple con todos los objetivos específicos planteados en este proyecto.

4.6. Alternativa de clasificación.

En esta sección se presentan los resultados obtenidos al aplicar el algoritmo propuesto para la clasificación de los CCM en la sección 3.4.

En la Tabla 32 se evidencian los datos obtenidos al ejecutar la función $EstadisticosCuefrenciaVM.m$ con el fin de analizar los datos obtenidos para las variables estadísticas E1 y E3. Los datos obtenidos para los estadísticos E1 y E3 permiten establecer rangos para cada una de las variables de la columna 1 Tabla 32.

Revisando los datos obtenidos en la Tabla 32 se puede concluir que para las variables estadísticas analizadas, las variables DA05g y DA10g se encuentran contenidas casi en un

70% en la variable GCAB, por lo que no permite establecer una buena clasificación de las variables relacionadas con el grupo desalineado.

Respecto a las variables del grupo desbalanceado se puede llegar a concluir que la variable DBv1r2 está contenida en un 99% en la variable DBv1r1. Es importante explicar que las variables en mención no se encuentran en el mismo CCM por lo que realmente DBv1r2 no se encuentra contenida en la variable DBv1r1. Debido a lo anterior las variables correspondientes al grupo desbalanceado si permiten establecer rangos de clasificación respecto al valor medio.

Tabla 32. Variables calculadas respecto a la media.

Variable	**Estadísticos con referencia al valor medio**			
	E1 * e-4		**E3 *e-4**	
	máximo	mínimo	máximo	mínimo
GCAB	63.3774	51.1120	63.6387	51.1915
DA05g	54.6043	49.6737	54.6446	49.7187
DA10g	56.3141	49.7695	56.4294	49.8010
DA15g	41.7704	35.9389	41.8432	35.9896
DBv1r1	75.1348	54.1971	75.7943	54.3056
DBv1r2	71.7752	67.1302	71.7996	67.1456
DBv2r1	80.8291	74.5636	80.8447	74.6016
DBv2r2	105.1396	96.4960	105.1908	96.5142

E1: Valor medio
E3: RMS Raíz media cuadrática

Fuente: (Autor).

En la Tabla 33 se realiza una comparación entre los valores obtenidos en la Tabla 31 respecto a las variables estadísticas de E1 y E3; comparados con los valores obtenidos en la Tabla 32. Esta tabla se realiza con el fin de verificar que la Tabla 31 y la Tabla 32 no presenten cambios significativos entre los valores encontrados.

Tabla 33. Relación entre los estadísticos de las Tabla 31 Tabla 32.

Variable	Datos Tabla 31		Datos Tabla 32		% de Error	
	E1 *e-4	**E3 *e-4**	**E1 *e-4**	**E3 *e-4**	**E1 *e-4**	**E3 *e-4**
GCAB	55.3963	55.9504	57.2447	57.4151	3,33676671	2,61790630
DA05g	52.2840	52.3842	52.1390	52.1816	0.27736902	0.38677454
DA10g	52.9456	53.1211	53.0418	53.1152	0.18177804	0.01114248
DA15g	38.4259	38.5947	38.8546	38.9164	1.11588412	0.83348626
DBv1r1m	62.8185	64.0333	64.6660	65.0500	2.94091266	1.58773715
DBv1r2m	69.4527	69.5116	69.4527	69.4726	3.7466E-14	0.05605059
DBv2r1m	77.7885	77.8775	77.6964	77.7231	0.11844931	0.19818521
DBv2r2m	100.5636	100.6905	100.8178	100.8525	0.25280166	0.16090439

Fuente: (Autor).

Revisando los datos obtenidos en la Tabla 33 se puede concluir que el porcentaje de error entre la Tabla 31 y la Tabla 32; respecto a la variable E1 no supera el 4% en general para todas la variables y respecto a la variable E3 no supera el 3% en general para todas las variables. Debido a esto se puede concluir que las variables estadísticas obtenidas en la Tabla 32 representan aproximadamente el 96% de los datos obtenidos en la Tabla 31.

Debido a que las variables GCAB, DA05g y DA10g no se podrían clasificar por medio de los estadísticos E1 y E3 se realizó una revisión general de las variables correspondientes a los grupos desalineado y desbalanceado con el fin de encontrar CCM en los cuales se diferencien entre sí.

En la Tabla 34 se presentan los resultados obtenidos de realizar una revisión de los CCM con el fin de diferenciar las variables de los grupos desalineado y desbalanceado entre sí.

Tabla 34. Revisión de coeficientes en escala Mel

				Estadísticos con referencia al valor medio				
Variable	**No CCM**	**RC * e-4**	**R**	**Valor medio * e-4**	**E1 * e-4**		**E3 *e-4**	
					máximo	**mínimo**	**máximo**	**mínimo**
DA05g	5	16-20	1	41.2834	45.5471	36.7612	45.6609	36.8632
DA10g	5	16-20	1	34.7058	37.4700	32.5236	37.5749	32.5586
DA10g	1	2-6	2	52.9456	56.3141	49.7695	56.4294	49.8010
DA15g	1	2-6	2	45.5846	49.2479	42.5153	49.3399	42.5668
DBv1r1	5	16-20	3	51.4555	55.0638	49.0792	55.1244	49.1129
DBv1r2	5	16-20	3	60.0922	62.4134	58.2597	62.4532	58.2718
DBv2r1	1	2-6	4	80.3992	86.3957	77.1284	86.5424	77.1638
DBv2r2	1	2-6	4	104.0082	109.6837	100.0353	109.7139	100.0680

E1: Valor medio.
E3: RMS Raíz media cuadrática.
RC: Rango Cuefrencia.
R: Relación entre variables identificada por números.

Fuente: (Autor).

Revisando los datos obtenidos de la Tabla 34 se encontraron coeficientes diferentes a los analizados en la sección 4.2.3 por lo que se puede concluir que se pueden realizar estudios avanzados respecto a la clasificación de las variables de los grupos desalineado y desbalanceado respecto a los estadísticos de valor medio.

5. CONCLUSIONES

El conocimiento del banco de vibraciones en cuanto a sus componentes, permiten determinar las diferentes pruebas que se puedan realizar, sin afectar definitivamente el funcionamiento del mismo; por lo que, es importante realizar pruebas preliminares con el fin de determinar cuáles son las limitantes que presenta el banco de vibraciones; en cuanto a su diseño, por razones de seguridad del banco y por razones de seguridad del operador.

Las muestras obtenidas a partir del sistema de adquisición debieron ser estudiadas por medio de algoritmos en Matlab, con el fin de extraer la mayor cantidad de muestras correspondientes a una misma variable; que se encuentren agrupadas en un rango específico respecto a la amplitud en frecuencia.

EL método visual aplicado a la transformada cepstrum permitió establecer que las transformadas cepstrum real y compleja, en los espectros de la cuefrencia y de la envolvente; no permitían determinar coeficientes cepstrum definidos y diferencias para las fallas de desbalanceo y desalineación en máquinas rotátivas. La transformada cepstrum real en la escala Mel permitió encontrar coeficientes cepstrum definidos y diferencias en las fallas estudiadas en máquinas rotátivas por medio del método visual.

Los coeficientes en la escala Mel para las fallas en máquinas rotativas se pueden encontrar hasta el cuarto coeficiente cepstrum. La distancia euclidiana permitió encontrar diferencias entre los coeficientes cepstrum en escala Mel de acuerdo al coeficiente estudiado para las fallas de desbalanceo y desalineación en máquinas rotátivas.

Las características de los coeficientes cepstrum en escala Mel se determinaron por medio del análisis estadístico teniendo en cuenta las variables de valor medio, desviación estándar, coeficiente de variación, raíz media cuadrática, Curtosis, asimetría y factor de forma.

Las variables estadísticas de valor medio y de raíz media cuadrática permiten encontrar diferencias de acuerdo al coeficiente cepstrum estudiado para cada una de las fallas de desbalanceo y desalineación en máquinas rotátivas.

El coeficiente de variación permitió determinar que las variables estudiadas para las fallas de desbalanceo y desalineación no poseen una desviación estándar muy elevada.

La Curtosis permitió establecer que todas las variables estudiadas para las fallas de desbalanceo y desalineación son del tipo platicúrtica teniendo como referencia la curva gaussiana del tipo mesocúrtica.

La asimetría permitió establecer que en general todas las variables son asimétricas positivas, solo las variables DA05g y DBv2r1 son asimétricas negativas. También permitió establecer que aunque se encuentra que las variables son asimétricas positivas y negativas se encuentran muy cercanas a la simetría.

El factor de forma permitió establecer que todas las variables estudiadas para las fallas de desbalanceo y desalineación según el coeficiente cepstrum en general poseen una forma definida que es igual para todas las muestras de una misma variable.

6. RECOMENDACIONES

El autor recomienda realizar los siguientes estudios los cuales pueden ser derivados de este proyecto.

Realizar un programa en Matlab con el fin de diseñar un clasificador para las fallas de desbalanceo y desalineación en máquinas rotátivas aplicando la transformada cepstrum en escala Mel.

Realizar clasificación de las fallas de desbalanceo y desalineación en máquinas rotátivas para frecuencias superiores e inferiores a la propuesta en este proyecto.

Realizar bancos de vibraciones que permitan estudiar fallas en máquinas rotátivas como flechas dobladas, soltura mecánica, fallas en rodamientos, fallas eléctricas, entre otras; que puedan aportar a una clasificación amplia de la transformada cepstrum en la escala Mel.

Realizar un mejoramiento tecnológico y de diseño en el banco de vibraciones de las Unidades Tecnológicas de Santander con el fin de estudiar otras variables para las fallas de desbalanceo y desalineación en máquinas rotátivas.

Realizar un clasificador de las fallas de desbalanceo y desalineación en máquinas rotátivas basado en el estudio del valor medio como se propuso en este proyecto; con el fin de determinar rangos en los cuales se puedan ubicar las distintas variables estudiadas en este proyecto.

7. REFERENCIAS BIBLIOGRÁFICAS

Acosta, D., Molina, J. y Cevallos H. (2013, Febrero). Adquisición de vibraciones mecánicas de un motor en funcionamiento usando Labview. *Escuela Superior Politécnica del Litoral ESPOL.* Guayaquil, Ecuador. Recuperado de: *http://www.dspace.espol.edu.ec/bitstream/123456789/24133/1/Adquisicion%20de%20vibraciones%20mec%C3%A1nicas_fiec.pdf*

Agudelo, H. (2008, Diciembre). Caracterización de señales sísmicas utilizando modelos paramétricos y transformada cepstrum. *Tecnológicas.* Universidad Tecnológica de Pereira, Pereira, Colombia. Recuperado de: *http://itmojs.itm.edu.co/index.php/tecnologicas/article/viewFile/284/292http://itmojs.itm.edu.co/index.php/tecnologicas/article/viewFile/284/292*

Aldaz, L. A. (2015). *Análisis espectral de señales de vibraciones mecánicas causadas por desalineación como método de mantenimiento predictivo en bombas hidráulicas centrifugas horizontales de flujo radial de 1 hp* (Tesis de Pregrado). Universidad Técnica de Ambato, Ambato, Ecuador. Recuperado de: *http://repositorio.uta.edu.ec/bitstream/123456789/10374/1/Tesis%20I.M.%20271%20-%20Aldaz%20Mayorga%20Luis%20Amable. pdf*

Benítez, R. A. (2013). *Diseño e implementación de un banco didáctico para alineación de elementos rotativos y balanceo de masas en cantiléver* (Tesis de Pregrado). Universidad Autónoma de Occidente, Santiago de Cali, Colombia. Recuperado de: *http://red.uao.edu.co:8080/bitstream/10614/7026/1/T05109.pdf*

Bernal, C. A. (2006) *Metodología de la investigación.* Pearson. (pp. 56) Recuperado de: *https://books.google.com.co/books?id=h4X_eFai59oC&pg=PA56&dq=metodo+deductivo&hl=es-419&sa=X&sqi=2&pjf=1&ved=0ahUKEwiNtKKFzYTUAhVDfiYKHUtAA10Q6AEIITAA#v=onepage&q=metodo%20deductivo&f=false*

Brien, R., Molisani, L. y Burdisso, R. (2013, Noviembre). Técnicas avanzadas para la detección de fuentes sonoras. Recuperado de: *https://www.researchgate.net/profile/ronald_obrien/publication/280625065_tecnicas_avanzadas_para_la_deteccion_de_fuentes_sonoras/links/5600cc0308aeba1d9f84e628.pdf*

Estupiñan, E., San Martin, C. y Canales, L. (2006, Agosto). Desarrollo de un instrumento virtual para el balanceamiento dinámico de rotores. *Ingeniare.* ISNN 0718-3305. Recuperado de: *http://www.scielo.cl/scielo.php?pid=S0718-33052006000100008&script=sci_arttext&tlng= pt*

Faúndez, M. (2000). *Tratamiento digital de voz e imagen y aplicación a la multimedia.* Editorial MARCOMBO. (pp. 54 - 57). Recuperado de: *https://books.google.com.co/books?id=bJAHd5YC61IC&pg=PA55&dq=cepstrum&hl=es-419&sa=X&redir_esc=y#v=onepage&q=cepstrum&f=true*

Flórez, R. y Asiain, T. (2011, Enero). Diagnóstico de Fallas en Máquinas Eléctricas Rotatorias Utilizando la Técnica de Espectros de Frecuencia de Bandas Laterales. *Información Tecnológica*. ISNN 0718-0764. Recuperado de: *http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-07642011000400009*

Forero, W., Lizcano, D. y Coronel, J. (2016) *Implementación de un banco para el análisis de vibraciones mecánicas generadas por desbalanceó y desalineamiento en máquinas rotátivas* (Tesis de Pregrado). Unidades Tecnológicas de Santander, Bucaramanga, Colombia.

Gajic, B y Paliwal, K. (2001). *Robust feature extraction using subband spectral centroid histograms.* Recuperado de: *https://pdfs.semanticscholar.org/0a4d/dd4a354c076fc08c34de1dfd3099fdca02c4.pdf*

Gómez, D. (2011). *Contribuciones al reconocimiento robusto de habla en redes de comunicaciones mediante transparametrización* (Tesis Doctoral). Universidad Carlos III de Madrid, Leganés, España. Recuperado de: *https://e-archivo.uc3m.es/bitstream/handle/10016/14359/Tesis_Diego_Ferney_Gomez_Cajas.pdf?sequence=1*

Gómez de León, F. C. (1998) *Tecnología del mantenimiento industrial*. Publicaciones universidad de Murcia. (pp. 201 - 203) Recuperado de: *https://books.google.com.co/books?id=bOrFC3532MEC&pg=PA202&dq=desalineacion+angular+felix+gomez&hl=es-419&sa=X&ved=0ahUKEwjG99bg8PTTAhUFTCYKHefmB2UQ6AEIITAA#v=onepage&q=desalineacion%20angular%20felix%20gomez&f=false*

Harper, E. (2004) *Instalaciones y montaje Electromecánico*. Limusa (pp. 118 - 119) Recuperado de: *https://books.google.com.co/books?id=yPVngh2E0-AC&pg=PA131&dq=alineacion+enriquez+harper&hl=es-419&sa=X&redir_esc=y#v=onepage&q=alineación%20enriquez%20harper&f=true*

Hermosa**,** A. (1999). *Principios de electricidad y electrónica II.* Editorial Marcombo. (p. 190) Recuperado de: https://books.google.com.co/books?id=OBGdJcvSvCQC&pg=PA190&dq=factor+de+forma&hl=es-419&sa=X&redir_esc=y#v=onepage&q=factor%20de%20forma&f=true

Herrera, A. (2000). *La clasificación numérica y su aplicación en la ecología.* 1 Edición. Sammenycar. (pp. 32 - 34). Recuperado de: *http://site.ebrary.com/lib/utssp/reader.action?docID=10149789*

ISO (2003). ISO 1940-1 Vibraciones mecánicas – Requisitos de calidad de balance para rotores en estado constante. (pp. 2 - 11) Recuperado de: *https://www.iso.org/standard/27092.html*

Jack, L. y Nandi, A. (2002). *Fault detection using support vector machines and artificial neural networks, augmented by genetic algorithms. Mechanical Systems and Signal Processing*. *ELSEVIER.*

James, G., Burley, D., Clements, D., Dike, F., Searl, J., Steele, N. y Wright, J. (2002). Matemáticas avanzadas para Ingeniería. 2 Edición. Pearson. (pp. 362 - 379).

Lei, Y., He, Z. y Zi, Y. (2007). *A new approach to intelligent fault diagnosis of rotating machinery*. *ELSEVIER.* Recuperado de: *https://pdfs.semanticscholar.org/3196/bf8b2f965810f8a53a3d57ecb439feb00c97.pdf*

Lémoli M. (2015, Junio). Desbalance y desalineación en motores eléctricos. Costa Rica, Motortico. Recuperado de: *http://www.motortico.com/biblioteca/MotorTico/2015%20JUN%20-%20Desbalance%20y%20desalinemiento%20en%20Motores%20Elect ricos.pdf*

Lind, D., Marchal, W. y Wathen, S. (2012). *Estadística Aplicada a los Negocios y la Economía*. 15ª Edición. Editorial Mc Graw Hill.

Manzano, J. J. (2014). *Maquinas Eléctricas*. 2 Edición. Paraninfo. (pp. 282 - 283) Recuperado de: *https://books.google.com.co/books?id=Q2vPAgAAQBAJ&pg=PA283&dq=alineacion+manzano+orrego&hl=es-19&sa=X&redir_esc=y#v=onepage&q=alineación%20manzano%20orrego&f=true*

Martin, J. C. (2012). *Maquinas Eléctricas.* Editex. (pp. 26 - 27) Recuperado de: *https://books.google.com.co/books?id=Aiy8AwAAQBAJ&printsec=frontcover&dq=maquinas+electricas+juan+carlos+martin&hl=es-19&sa=X&ved=0ahUKEwj18bOr0TTAhUJQSYKHV1AA34Q6AEIITAA#v=onepage&q=maquinas%20electricas%20juan%20carlos%20martin&f=false*

Martínez, C. (2012). *Estadística y muestreo.* 13ª Edición. ECOE Ediciones. (pp. 303 - 310)

MathWorks. (2017). *Documentation rceps.* Editor MathWorks. Recuperado de: https://www.mathworks.com/help/signal/ref/rceps.html

Morales, O. (2011). *Análisis tiempo-frecuencia de señales de vibraciones mecánicas para la detección de fallos en máquinas rotativas* (Tesis Magister). Universidad Nacional de Colombia, Manizales, Colombia. Recuperado de: *https://core.ac.uk/download/pdf/11054266.pdf*

Moreno, F. E., Becerra, J. A. y Rendón, C. A. (2015, Enero) Diseño de un sistema de análisis temporal y espectral para detectar fallas por vibración en motores eléctricos. *Faculta de Ingeniería*. ISNN 0121-1129. Recuperado de: *https://dialnet.unirioja.es/servlet/articulo?codigo=5170939*

Nava, A. (2013). *Procesamiento de series de tiempo.* 2 Edición. Ediciones científicas universitarias. (pp. 301,302). Recuperado de: *http://site.ebrary.com/lib/utssp/detail.action?docID=11286234&p00=procesamiento+series+tiempo*

Olarte, W., Botero, M. y Cañón, B. (2010, Agosto). Análisis de vibraciones: una herramienta clave en el mantenimiento predictivo. *Scientia et Technica*. Universidad Tecnológica de Pereira, Pereira, Colombia. ISNN 0122-1701. Recuperado de: *https://dialnet.unirioja.es/servlet/ articulo?codigo=4546611*

Oppenheim, A. V. y Schafer, R. W. (2004, Septiembre). From Frecuency to Quefrency: A History of the Cepstrum. *IEEE Journals & Magazines*. Recuperado de: *https://www.researchgate.net/profile/Av_Oppenheim/publication/3321562_From_Frequency_to_Quefrency_A_History_of_the_Cepstrum/links/5519617d0cf2d241f3564c58.pdf*

Oppenheim, A. V. y Schafer, R. W. (2011). *Tratamiento de señales en tiempo discreto.* 3 Edición. Pearson. (pp. 59, 60, 954 - 959)

Peláez, C. (2002). *Reconocimiento de habla mediante transparametrización; una alternativa robusta para entornos móviles e IP* (Tesis Doctoral). Universidad Carlos III de Madrid, Leganés, España. Recuperado de: *https://dialnet.unirioja.es/servlet/dctes?codigo=2162*

Quiroga, J., Trujillo, G. y Quintero, S. (2012, Diciembre). Estudio de fallas incipientes en rodamientos usando la técnica de la envolvente y cepstrum. *Ingeniare*. Recuperado de: *http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0718-33052012000300009&lng= es&nrm=iso&tlng=es*

Raó, SS. (2012) *Vibraciones Mecánicas*. Pearson. (pp. 651-658)

Ramírez, A. (2014) *Tutorial de Matlab (Español).* Publicación en Youtube. Universidad Nacional de Colombia, Manizales, Colombia. Recuperado de: *https://www.youtube.com/watch?v=Wgk_FdfpJqg&list=PLj3KYX7UqPG8uZWqtQ7ZBG1DSou1fLDMS*

Rodríguez, M., Álvarez, S. y Bravo, E. (2001). *Coeficientes de Asociación.* Plaza y Valdés. (pp. 44 - 46). Recuperado de: *http://site.ebrary.com/lib/utssp/detail.action?docID=10862477&p00=coeficientes+asociación*

Salcedo, F. (2011). *MODELOS OCULTOS DE MARKOV del reconocimiento de voz a la música.* Editorial Lulu. (pp. 129 - 131). Recuperado de: *https://books.google.com.co/books?id=L2d0eVFr004C&pg=PA130&dq=coeficiente+cepstrum+en+la+escala+mel&hl=es-419&sa=X&redir_esc=y#v=onepage&q=coeficiente%20cepstrum%20en%20la%20escala%20mel&f=true*

Sampieri, R., Fernández-collado, C. y Baptista, P. (2006) *Metodología de la investigación.* Mc Graw Hill. (pp. 102 - 160)

Sánchez, F.T., Pérez, A., Sancho, J. L. y Rodríguez, J. (2007). *Mantenimiento Mecánico de Maquinas*. Universitat Jaume. (pp. 164 - 253) Doi: *http://dx.doi.org/10.6035/INFiTEC.2007.25*

Sinais Ingeniería (2013) Norma ISO 10816 – 1995. Recuperado de: *http://www.sinais.es/Recursos/Curso-vibraciones/normativa/iso10816.html*

Torres, R. M. y Batista, C. R. (2010, Enero). Análisis vibrodinámico de motores eléctricos. *Ingeniería Mecánica.* ISNN 1815-5944. Recuperado de: *http://scielo.sld.cu/scielo.php?pid=S1815-59442010000100002&script=sci_arttext&tlng=pt*

Zill, D. y Cullen, M. (2008). *Matemáticas avanzadas para Ingeniería.* 3 Edición. Mc Graw Hill. (pp. 380 - 389) Recuperado de: *http://site.ebrary.com/lib/utssp/detail.action?docID=10515233&p00=matematicas+avanzadas*