



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Aplicación de las redes
neuronales en el reconocimiento
de fallas de turbina de gas**

TESIS

Que para obtener el título de

Ingeniero Mecánico

P R E S E N T A

Salvador Felipe Altamirano

DIRECTOR DE TESIS

Dr. Víctor Hugo Jacobo Armendáriz



Ciudad Universitaria, Cd. Mx., 2016

Prólogo

La turbina de gas tiene grandes aplicaciones en la rama de la ingeniería, y es importante tenerla en buena condición de operación para poder desempeñar su función requerida. Para poder analizar las condiciones de salud de un motor de turbina de gas es requerido un banco de datos, las empresas privadas no dan a conocer dicha información con facilidad, además la extracción de un banco de pruebas tiene grandes costos económicos si no se les da la adecuada aplicación, encontrar un simulador de motores de turbina ayuda en la creación de dichos datos y así poder aplicar algún método de diagnóstico para el reconocimiento de fallas presentes en el motor.

El simulador de flota de motores ProDiMES es un software capaz de darnos este banco de datos. Los algoritmos usados para el análisis de los parámetros del conducto de flujo son capaces de predecir fallas y pronosticar la vida útil de la turbina de gas. En el motor de turbina se pueden presentar diferentes fallas, dependiendo en qué condiciones de salud estén y en el componente donde se presente. ProDiMES genera muchas fallas del conducto de flujo y es importante verificar como algún método de diagnóstico las identifica.

La teoría de reconocimiento de patrones ayuda en la tarea de identificación de fallas y es capaz de adaptarse a un gran conjunto de datos para poder predecir donde se encuentra el problema para cualquier herramienta de reconocimiento, la clasificación de fallas es necesaria y se forma con datos de ProDiMES. Una herramienta de reconocimiento son las redes neuronales la cual ayuda a identificar las fallas individuales procedentes de un conjunto de datos. En este trabajo se hace una propuesta de clasificación donde se utilizan como herramienta de reconocimiento las siguientes redes: Perceptrón Multicapa; Red de Base Radial y Red Neuronal Probabilística. Se usan los criterios probabilísticos para estimar la eficiencia del diagnóstico en la herramienta de reconocimiento.

Una vez hecha la clasificación con el conjunto de datos procedentes de ProDiMES, se ordenan de tal forma para que las redes puedan trabajar con ella. Se hace la optimización por cada red neuronal para poder mostrar los mejores resultados de diagnóstico. El resultado de este trabajo arroja que la red de base radial es mejor en términos numéricos pero las otras redes tienen ventajas en términos del tiempo de cálculo, también la diferencia de reconocimiento entre cada una de ellas no es grande por lo que se puede usar cualquiera dependiendo de la exigencia del usuario. Se hace la comparativa de este trabajo con un artículo internacional procedente de ASME Turbo Expo 2015, mostrando que nuestros resultados de identificación de fallas son buenos e incluso competitivos a los de su investigación.

Contenido

Prólogo.....	i
Introducción.....	vi
Hipótesis	viii
Objetivos.....	viii
CAPÍTULO I Turbinas de Gas	viii
1.1 Importancia de la Turbina de Gas.....	1
1.2 Operación básica y componentes principales de la turbina de Gas	1
1.2.1 Compresor	2
1.2.2 Cámara de combustión	4
1.2.3 Turbina	5
1.3 Tipos de Turbinas para aeronaves	6
1.3.1 Turborreactor.....	7
1.3.2 Turbo fan.....	7
1.3.3 Turbohélice	8
1.3.4 Turbo eje	9
CAPÍTULO II Diagnóstico de la Turbina de Gas.....	10
2.1 Diagnóstico de fallas	10
2.2 Análisis de conducto de flujo	11
2.2.1 Etapas del proceso de diagnóstico	12
2.2.2 Bases teóricas del ACF.....	12
2.2.3 Modelos usados para ACF.....	13
2.3 Modelo del diagnóstico para ACF.....	13
2.3.1 Introducción a los modelos.....	13
2.3.2 Modelo por Polinomios	15
2.4 Concepto de desviación para el ACF.....	15
2.5 Clasificación de fallas	16
2.5.1 Principios para preparar la clasificación	17
2.5.2 Probabilidades del diagnóstico correcto	18
2.5.3 Matriz decisión de detección	19
CAPÍTULO III Software ProDiMES	20
3.1 Introducción	20
3.2 Simulador de Flota de Motores (SFM)	20

3.3 Descripción de la Interfaz gráfica de usuario	21
3.4 Generador de casos.....	24
3.5 C-MAPSS	25
3.6 Parámetros de salida del simulador	26
CAPÍTULO IV Redes Neuronales Artificiales.....	27
4.1 Introducción	27
4.2 Neurona Biológica.....	27
4.3 Neurona artificial	28
4.4 Funciones de activación de las redes neuronales	30
4.5 Tipos de Redes Neuronales	31
4.5.1 Perceptrón Multicapa	31
4.5.2 Red de Base Radial.....	33
4.5.3 Red Neuronal Probabilística	35
4.6 Teoría de reconocimiento de patrones	36
4.6.1 Adquisición de datos	36
4.6.2 Extracción de características	37
4.6.3 Clasificación.....	37
CAPÍTULO V Implementación de las redes en el reconocimiento de fallas	38
5.1 Ingreso de datos a ProDiMES	39
5.2 Generación de flota de motores e identificación de parámetros.....	40
5.3 Manipulación de los parámetros de salida	41
5.4 Uso de los coeficientes del modelo de referencia	42
5.5 Modelo general para la Turbina de gas.....	43
5.6 Corrección del modelo de la Turbina	44
5.7 Clasificación de fallas	45
5.8 Conjunto de validación y estructura de la red neuronal	47
5.8.1 Estructura de la red.....	48
5.9 Optimización de las redes neuronales.....	49
5.9.1 Perceptrón multicapa.....	49
5.9.2 Red de Base Radial.....	50
5.9.3 Red Probabilística	51
5.10 Resultados	53
5.10.1 Probabilidades de diagnostico	53
5.10.2 Análisis de detección.....	54
Conclusiones	57

Bibliografía	59
Anexo: Código de programación.....	61

Introducción

Las fallas y procesos de deterioro en las turbinas de gas afectan fuertemente en la fiabilidad, disponibilidad y costos de operación de la misma. Para mitigar estos efectos negativos los sistemas de monitoreo se han convertido en sistemas prácticos, los cuales emplean la información de los parámetros medidos de la turbina de gas y hacen posible el análisis de la condición de salud del motor para después predecir sus cambios e identificar qué tipo de problemas están presentes.

Los sistemas de monitoreo de salud para la turbina de gas y los algoritmos de diagnóstico del conducto de flujo, son capaces de identificar diferentes fallas así como los mecanismos de degradación de los componentes de dicha turbina. Con la teoría del análisis de conducto de flujo se identifican las variables medidas en el compresor, turbina y cámara de combustión de la turbo máquina.

Gracias al simulador usado ProDiMES, se pueden obtener datos de una flota de motores con diferentes fallas para poder construir una adecuada clasificación de fallas. Se trabajan dos conjuntos de datos, uno de entrenamiento y otro de validación, para el primer conjunto se usa el software para la simulación de nuevos datos de los cuales se manipularan para las entradas de las redes neuronales y para el segundo conjunto se usa una base ya hecha por los creadores del software. Para la propuesta de clasificación son usadas tres redes neuronales como técnicas de reconocimiento de patrones.

Este trabajo está organizado de la siguiente manera:

Capítulo 1: Se introduce al conocimiento de las turbinas de gas, dando a conocer la importancia que éstas tienen en la actualidad. Se dan a conocer los componentes básicos de la turbina de gas, describiendo al compresor, turbina y cámara de combustión. Y por último los tipos de motores que se dan en el área aeronáutica, ya que en este trabajo se utiliza alguna de éstas para su posterior análisis.

Capítulo 2: Se toca el tema del diagnóstico de fallas y se da a conocer por ser una de las bases importantes para nuestro trabajo, así como la teoría de análisis de conducto de flujo, tema importante para el reconocimiento de las variables de trabajo. El modelo de diagnóstico aplicado a este trabajo es el de polinomios y se da una introducción al tema. La desviación es una parte indispensable para proponer la clasificación de fallas, por lo que se dan las bases teóricas para conocerla.

Capítulo 3: ProDiMES es la fuente para los datos de clasificación, es por ello que en este apartado se explica el contenido del programa. Se da a conocer la interacción con la interfaz gráfica de usuario y todas las posibles fallas que puede generar para el motor turbo ventilador. El simulador usa el modelo estático en estado estacionario, desarrollado en el ambiente Matlab. También se explican las condiciones de operación en las que trabaja dicho software.

Capítulo 4: Se introduce la neurona biológica, artificial y las funciones de activación para su correcto funcionamiento. Para el uso de la clasificación de fallas es necesaria alguna red

neuronal para poder aplicar el diagnóstico de fallas por lo que se dan a conocer las tres redes neuronales implementadas, explicándolos más a detalle para su comprensión y manipulación.

Capítulo 5: Por último este apartado se usa para poder explicar la manipulación del banco de datos creado por ProDiMES, como es que se aplicaron los modelos matemáticos para cada motor y la separación de las fallas simuladas. Se explica la formación de clases y la optimización para cada una de las redes neuronales usadas para después dar a conocer el cálculo de precisión del diagnóstico. Se analizan los resultados de clasificación de fallas y se dan las conclusiones correspondientes de cada una.

Hipótesis

Las redes neuronales es un área de la inteligencia artificial que al ser implementadas para el reconocimiento de patrones, permiten bajo ciertas consideraciones mejorar la clasificación e identificación de las fallas en turbinas.

Objetivos

Generales

Mejorar el método de clasificación de fallas mediante la aplicación de las redes neuronales y utilizar el concepto de análisis de conducto de flujo para mejorar el diagnóstico de fallas en las turbinas de gas..

Específicos

- Obtener un banco de datos para el diagnóstico de fallas en turbinas mediante el software ProDiMES
- Exponer la importancia de las turbinas de gas en la industria y la necesidad de un diagnóstico preventivo.
- Dar a conocer las ventajas del uso del simulador de motores ProDiMES
- Desarrollar un algoritmo capaz de identificar las fallas de una flota de motores turboventilador.
- Llegar a una mejor confiabilidad de diagnóstico mediante la teoría de reconocimiento de patrones.

CAPÍTULO I

Turbinas de Gas

1.1 Importancia de la Turbina de Gas

La turbina de gas es considerada una gran máquina de potencia, ésta tiene la capacidad de generar energía dependiendo de su tamaño, peso y características, es por ellos que las turbinas han sido llevadas en cualquier tipo de industria en los últimos 60 años (Boyce, 2012).

Las turbinas de gas han sido requeridas para las industrias energéticas, así como en los servicios públicos y las plantas mercantes, la industria petroquímica, y servicios públicos de todo el mundo. Su tamaño compacto, bajo peso, y la aplicación de combustible múltiple hace que sea una planta de energía natural para las plataformas en alta mar. Hoy en día hay turbinas de gas que funcionan con gas natural, nafta, diésel, metano, crudo, gases de bajo Btu, aceite, combustible vaporizado, y gases de biomasa.

Las turbinas han cambiado por las exigencias que hoy en día tenemos, llevando a cambiar sus sistemas internos, como conducto de flujo, arranque, aceite, combustible, etc. Esto ha dado como resultado el aumento de la eficiencia de las turbinas, los cuales estaban de un 15% hasta llegar a un 45% (Boyce, 2012).

El avance en la eficacia de la turbina dio pauta para el crecimiento de la tecnología de materiales, nuevos recubrimientos y nuevos esquemas de enfriamiento, lo que se traduce en una velocidad de calentamiento de 7582 BTU / kW-hr (8000 kJ / kW-h) a 6824 BTU / kW-hr (7199 kJ / kW-h). El factor limitante para la mayoría de las turbinas de gas ha sido la temperatura de entrada de la turbina. Con los nuevos esquemas de enfriamiento utilizando vapor o aire acondicionado, y los avances en la metalurgia en álabes, se han alcanzado temperaturas más altas en la turbina. Las nuevas turbinas de gas han dado temperaturas de entrada de hasta 2600 ° F (1427 ° C), y relaciones de presión de 40: 1 con una eficiencia de 45% o mayor.

1.2 Operación básica y componentes principales de la turbina de Gas

Comúnmente se habla de turbina de gas por separado ya que tienen características de diseño diferentes para cada aplicación, funcionan con sustancias gaseosas y no se espera un cambio de fase.

El turborreactor (*turbojet*) es el tipo más antiguo de los motores de reacción y su concepto fue desarrollado de manera independiente en los años de 1930 por dos ingenieros: Frank White y Hans von Ohain. Al primero se le da el reconocimiento por patentarlo y construir un motor funcional, pero Von Ohain fue el primero en utilizarlo para propulsar un avión. La configuración de lo que es un turborreactor se puede observar en la Figura 1.1

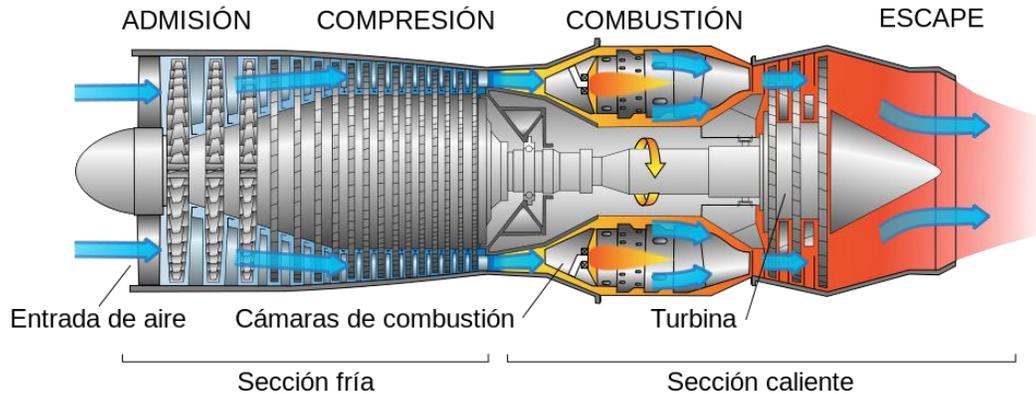


Figura 1.1 Diseño típico para un turborreactor (Jeff Dahl, Spanish translation by Xavigavax 2008)

En la figura se puede ver que aire de entrada fluye a través del compresor llevándolo a una presión más alta. En la cámara de combustión la energía se añade mediante la pulverización de combustible en el aire y encendiéndolo con chispa para generar flujo de alta temperatura. Este gas de alta temperatura entra en la etapa de la turbina produciendo una salida de trabajo en el eje, dicho trabajo se utiliza para accionar el compresor y otros dispositivos tales como un generador eléctrico. La energía que no se utiliza sale mediante los gases de escape por lo que estos tienen temperaturas o velocidades altas.

Básicamente el motor de turbina de gas se puede dividir dentro de 3 componentes generales, el compresor, la cámara de combustión y la turbina.

1.2.1 Compresor

Básicamente hay dos tipos de compresores, el de flujo centrífugo y el de flujo axial. Ambos tipos son impulsados por la turbina del motor y por lo general están acoplados al eje de la turbina.

El compresor radial (Figura 1.2) es una unidad única o dos etapas que consiste básicamente en un impulsor, un colector de difusor, y un colector de compresor. El colector de compresor dirige el aire comprimido en la cámara de combustión. A medida que el impulsor gira a alta velocidad, el aire es aspirado en el inductor. La fuerza centrífuga proporciona una alta aceleración de este aire y hace que se mueva hacia el exterior desde el eje de rotación hacia la salida, donde es expulsado a alta velocidad y alta energía cinética. En el colector de difusor el aumento de presión se produce por conversión de la energía cinética del movimiento en energía de presión estática. La compresión total se reparte entre el rotor y el difusor.

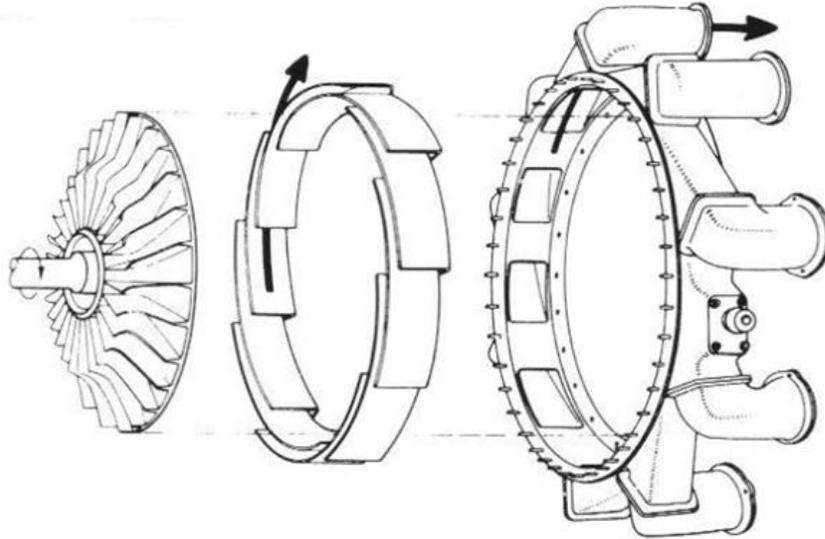


Figura 1.2 Componentes de un compresor radial de etapa singular, (Treager, 1995)

El compresor de flujo axial (Figura 1.3) consiste en uno o más conjuntos de rotores que llevan álabes en la de sección de perfil aerodinámico. Estos conjuntos están montados entre los cojinetes en las carcasas que incorporan las paletas del estator. El compresor es una unidad de varias etapas debido a que el aumento de la presión por cada etapa es pequeña; una etapa consiste en una fila de álabes giratorios seguido de una fila de álabes fijas en el estator. Durante el funcionamiento, el rotor gira a altas velocidades por la turbina, de manera que el aire se induce de forma continua en el compresor, acelerando las cuchillas rotativas, fluyendo hacia atrás en la fila adyacente de las paletas del estator, el aire es desacelerado el siguiente pasaje del estator y la energía cinética se traduce en presión. Los álabes del estator también sirven para corregir la desviación dada del aire mediante los alabes del rotor y así presentar al aire un ángulo correcto para la siguiente etapa de los alabes del rotor. La última fila de los álabes del estator normalmente actúa como alisadores de aire para eliminar la turbulencia del aire antes de la entrada en el sistema de combustión a una velocidad axial razonablemente uniforme. Cuando varias etapas de compresión operan en serie sobre un eje, es necesario variar el ángulo de los álabes del estator para que el compresor funcione de forma eficaz a velocidades por debajo de la condición de diseño.

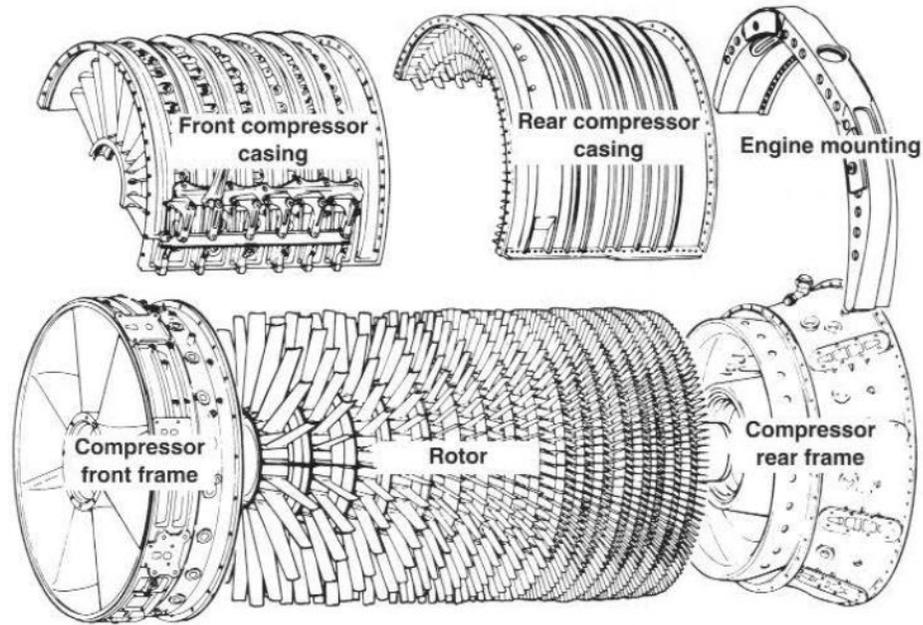


Figura 1.3 Compresor axial desmontado, Fuente: (General Electric 1995)

Con respecto a las ventajas y desventajas de los dos tipos, el compresor centrífugo es generalmente más robusto que el compresor axial y también es más fácil de desarrollar y fabricar. El compresor axial sin embargo maneja mucho más aire que el compresor centrífugo y por lo tanto tiene una mayor eficiencia.

1.2.2 Cámara de combustión

La cámara de combustión (Figura 1.4) tiene la difícil tarea de la quema de grandes cantidades de combustible, suministrado a través de las boquillas de pulverización de combustible con amplios volúmenes de aire suministrado por el compresor, la liberación del calor de tal manera que se expanda y acelere el aire para dar una corriente suave de gas calentado de manera uniforme en todas las condiciones requeridas por la turbina. Esta tarea debe llevarse a cabo con la pérdida mínima de la presión y con la máxima liberación de calor para el limitado espacio disponible.

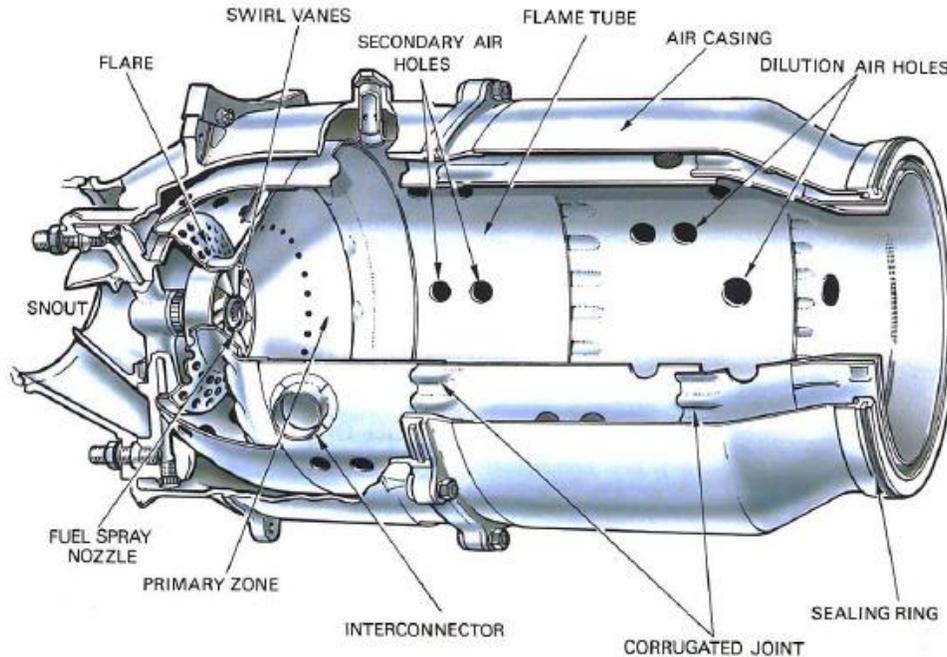


Figura 1.4 Diagrama para una cámara de combustión (Rolls Royce, Jet engine 1996)

La cantidad de combustible añadido al aire dependerá de la elevación de la temperatura requerida. Sin embargo, la temperatura máxima se limita dentro del intervalo de 850 a 1700 K. por los materiales de los que se fabrican los álabes de la turbina y materiales de las toberas. Dado que la temperatura del gas requerida en la turbina varía con el empuje del motor, y en el caso del motor de turbo-hélice de la potencia requerida, la cámara de combustión debe también ser capaz de mantener una combustión estable y eficaz en un amplio intervalo de condiciones de funcionamiento del motor.

1.2.3 Turbina

La función de la turbina es **conducir** cualquier carga útil (el compresor y accesorios, generador eléctrico, bomba de gas natural) mediante la extracción de una parte de la presión y la energía cinética de los gases de combustión de alta temperatura. Altas tensiones están involucrados en este proceso, y para un funcionamiento eficiente, las puntas de los alabes de la turbina pueden girar a velocidades de más de 458 (m/s). El flujo continuo de gas a la que la turbina está expuesta puede tener una temperatura de entrada entre 850 y 1700 K. y puede alcanzar una velocidad de más de 762 (m/s) en partes de la turbina.

Para producir el par de accionamiento, la turbina puede constar de varias etapas, cada una empleando una fila de álabes guía de tobera estacionarias y una fila de álabes móviles (Figura 1.5). El número de etapas depende de la relación entre la potencia requerida del flujo de gas, la velocidad de rotación a la que debe ser producido y el diámetro de la turbina permitido.

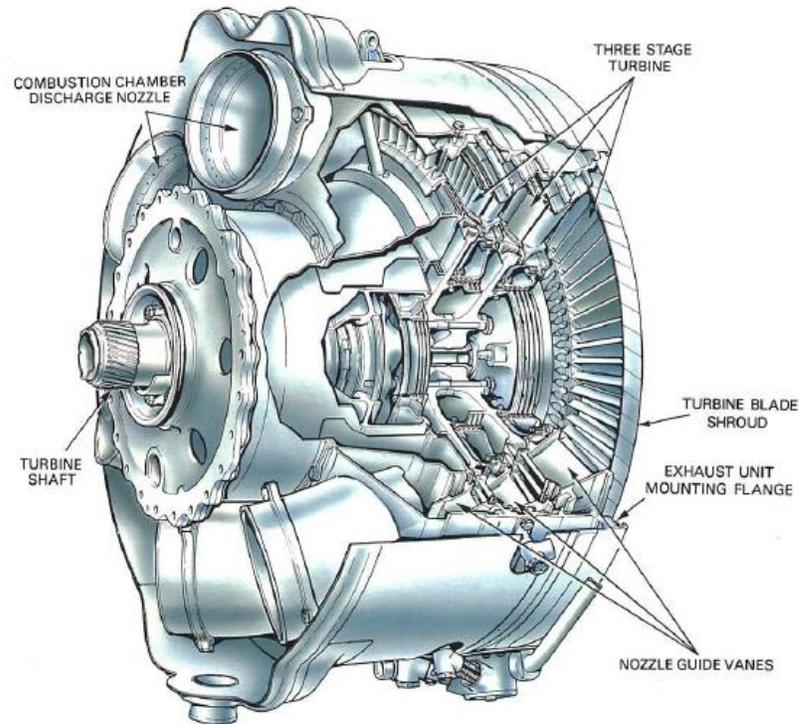


Figura 1.5 Turbina de triple etapa con un eje simple (Rolls Royce, Jet engine 1996)

El número de ejes, y por lo tanto las turbinas, varía con el tipo de motor; motores de alta relación de compresión por lo general tienen dos ejes, con compresores de alta y baja presión. En algunos motores el par de accionamiento se deriva de una turbina libre de potencia. Este método permite que la turbina funcione a su velocidad óptima, ya que es mecánicamente independiente de los demás ejes de la turbina y el compresor.

1.3 Tipos de Turbinas para aeronaves

(Hünecke, 2003) toma la clasificación de los motores según su tarea. Una distinción está hecha con referencia a las características de diseño tal como el número de carretes, el principio de compresión, la distribución del conducto de flujo con el motor, uso de los gases de escape.

Básicamente dice que hay cuatro tipos de motores usados para la aviación:

1. Turborreactor (*Turbojet*)
2. Turbo ventilador (*Turbofan*)
3. Turbohélice (*Turboprop*)
4. Turbo eje (*Turboshaft*)

Los motores Turborreactor y Turboventilador proporcionan fuerzas de propulsión directamente por las fuerzas de reacción generadas de los gases de escape. El motor Turboventilador, en particular, se clasifica de acuerdo a la porción de masa del conducto de

flujo que es desviado sobre el motor, también se tienen dos clasificaciones para el turbo ventilador, de bajo índice de derivación (*low-bypass*) y alto índice de derivación (*high-bypass*)

En un motor Turbohélice, la energía que se extrae de los gases calientes es usado para proporcionar movimiento al hélice. El gas caliente cuando es extraído de las toberas, transmite la mayoría de su energía a las turbinas, con la pequeña energía restante se genera el empuje.

Un motor turbo eje, todo la energía de los gases calientes es extraído y convertido para darle potencia al eje, o en su caso darle torque a la turbina libre. Este tipo de turbinas es típicamente usado en helicópteros, también son implementamos mínimamente para unidades de poder auxiliares que proporcionan potencia eléctrica o neumática para operar aeronaves en tierra.

1.3.1 Turborreactor

Como ya se vio anteriormente el turborreactor es el más antiguo de los motores de reacción. Estos son eficientes a velocidades supersónicas aunque pueden ser muy ruidosas. Hoy en día el uso de estos motores se ha disminuido por la causa de ruido, causando la utilización de turbo hélices con velocidades bajas, cuando se requiere de velocidades altas, optan por usar los motores turbo ventilador. El uso mayor para los turborreactores es en el área de misiles de crucero de mediano alcance, debido a su gran velocidad de escape, baja área frontal y relativa simplicidad.

1.3.2 Turbo ventilador

Este tipo de motores es el más usado para el área de aviación. Se caracterizan por disponer de un ventilador (*fan*) en la parte frontal del motor. El aire entrante se divide en dos caminos: flujo de aire primario y flujo secundario o flujo derivado (*bypass*). El flujo primario penetra al núcleo del motor como lo son compresores y turbinas y el flujo secundario se deriva a un conducto anular exterior y concéntrico con el núcleo. Los turbo fan tienen varias ventajas respecto a los turborreactores: consumen menos combustible, lo que los hacen más económicos, producen menor contaminación y reducen el impacto ambiental.

Para los motores de bajo índice de derivación, este ronda entre 0.2 y 2. Fue usado por la aviación civil hasta que apareció el de alta derivación, su operación es óptima entre Mach 1 y 2, y es utilizado en la actualidad para la aviación militar. A continuación un ejemplo de este motor en la Figura 1.6.



Figura 1.6 Turbo fan [low bypass] Pratt & Whitney JT8D, (KPWM Spotter 2004)

Los motores de *high-bypass* contienen un alto índice, mayor a 5. Estos son usados más en la modernidad. La mayor parte del empuje proviene del compresor, alrededor del 80%. El 20% restante proviene de los gases de escape de la tobera. El Boeing 787 tiene un alto índice de derivación de 10, en la Figura 1.7 se puede ver el avión que contiene este tipo de motor.



Figura 1.7 Aero nave con Turbo fan (high bypass) Boeing 787, (All Nippon Airways Boeing 787-8 Dreamliner JA801A 2013)

1.3.3 Turbohélice

Aproximadamente el 90% del empuje es producido por la hélice y el 10% restante por los gases de escape. Sus condiciones óptimas de operación son entre 250 y 400 mph y entre 18 000 y 30 000 ft. Es usado principalmente para pequeños y medianos aviones subsónicos. El motor Kuznetsov NK-12MA produce 15 000 HP, y es uno de los más potentes turbohélices hoy en día (Figura 1.8).



Figura 1.8 Turbohélice Kuznetsov NK-12, (Petebutt 2010)

1.3.4 Turbo eje

Este tipo de motor es similar al turbohélice ya que entrega su potencia a través de la hélice, pero no la mueve directamente. El vehículo aéreo más usado para este motor es para el helicóptero. Su diagrama de funcionamiento se muestra en la Figura 1.9.

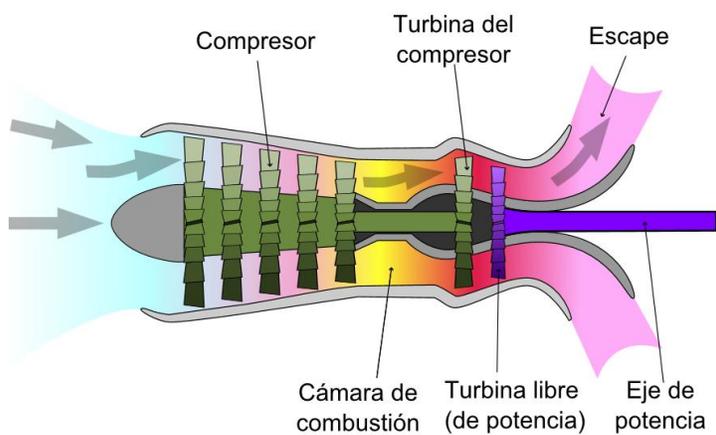


Figura 1.9 Diagrama de un Tubo eje (Turboshaft operation 2009)

CAPÍTULO II

Diagnóstico de la Turbina de Gas

2.1 Diagnóstico de fallas

El diagnóstico de fallas ha sido objeto de numerosas investigaciones en las últimas décadas. Las investigaciones en disciplinas tan diversas como la medicina, la ingeniería, las ciencias, los negocios y las finanzas han desarrollado metodologías para la detección de fallas o condiciones de anomalías para identificar o aislar qué componente en un sistema es defectuoso y decidir sobre el potencial de impacto hacia una falla o componente que ha fallado en la condición del sistema.

La diversidad de los dominios de aplicación está adoptado para la pluralidad de tecnologías de apoyo que han surgido en los últimos años en un intento de diagnosticar este tipo de eventos perjudiciales. En cuanto a las tecnologías de apoyo como lo son la distribución estadística, redes bayesianas, sistemas expertos, máquinas de estados finitos y herramientas de computación flexible (redes neuronales, lógica difusa, algoritmos genéticos, etc.) han sido sugeridos como posibles medios de diagnóstico y pronóstico para un sistema mecánico (Konrad & Isermann, 1996); (Mylaraswamy & Venkatasubramanian, 1997).

Con el fin de estimar la vida útil restante o el tiempo hasta el fallo de un componente, el mantenimiento basado en condición ayuda a maximizar el tiempo de actividad del motor, buscando determinar las fallas sin la existencia de falsas alarmas. Por lo tanto el diagnóstico de turbina de gas tiene como objetivo determinar con precisión las condiciones de fallas incipientes. El objetivo del diseñador es lograr el mejor rendimiento posible de una rutina de diagnóstico al minimizar las falsas alarmas de falla, mientras se reduce la demora de tiempo entre la iniciación y la detección de un evento de falla (Vachtsevanos, et al., 2006).

El monitoreo y diagnóstico de la turbina pertenecen a un espacio común en el monitoreo de condiciones en maquinaria, algunos ejemplos lo son las naves espaciales, aviones, diferentes plantas de energía, procesos industriales y de fabricación. Estos pueden ser considerados como sistemas complejos de ingeniería, volviéndose más sofisticados y complejos llevando a que decrezca la fiabilidad del sistema. Para mantener una alta fiabilidad, se aplican diversas herramientas de diagnóstico ya que son capaces de detectar e identificar las fallas incipientes, reduciendo la tasa de fallas en bruto, diversos factores de deterioros pueden ser responsables de estas fallas.

Un motor de turbina de gas puede ser considerado como una máquina muy compleja y costosa. Por ejemplo, el número total de datos en los componentes del motor y sus subsistemas puede llegar a 20,000 o más. Las turbinas de alta potencia cuestan muchos millones de dólares, estos precios pueden ser considerados como potenciales pérdidas directas debido a un posible fallo de la turbina de gas. En ocasiones las pérdidas indirectas suelen ser mucho mayores, por ello es de vital importancia que la turbina de gas se le proporcione un sistema de monitoreo eficaz.

Los sistemas de monitoreo para las turbinas de gas están basados en mediciones, variables reconocidas y señales. Tales sistemas no necesitan de parada ni de desmontaje del motor ya que operan en tiempo real, proporcionando datos de análisis y de registro en línea de diagnóstico.

2.2 Análisis de Conducto de Flujo (Gas Path Analysis)

Métodos universales tales como el análisis de vibraciones, análisis de aceite, combustible y arranque se aplican para el diagnóstico, el sistema de monitoreo incluye también otra técnica específica para las turbinas de gas, es decir el Análisis de Conducto de Flujo (ACF). Sus algoritmos están basados en el desarrollo de la teoría turbina de gas y de las mediciones del conducto de flujo (presiones, temperaturas, velocidades de rotación, consumo de combustible, entre otros) en donde se concentra en los tres principales componentes del motor, el compresor, la cámara de combustión y la turbina de alta o baja potencia. El ACF se puede considerar como una parte principal de un sistema de monitoreo de turbina de gas. Como un enfoque representativo para el diagnóstico de turbina de gas, el análisis de conducto de flujo ha sido elegido para esta tesis.

El análisis de conducto de flujo proporciona el conocimiento necesario de desempeño de los componentes de la turbina de gas, dejando al descubierto los mecanismos de degradación gradual y de fallas que puedan presentarse. Además las fallas del conducto de flujo, el mal funcionamiento de los sistemas de medición y control también pueden ser detectados e identificados.

Se puede afirmar que la simulación de la turbina de gas es una parte integral del proceso de diagnóstico. La función principal de los modelos es dar referencia al rendimiento de la turbina de gas con el fin de calcular las diferencias entre las mediciones actuales contra los de referencia. Estas diferencias (o desviaciones) sirven como fiables índices de degradación. Los datos registrados para una turbina real en ocasiones son insuficientes para formar una clasificación representativa debido a la rara y ocasional aparición de fallas reales, para tener un registro suficiente es necesario del registro de varias turbinas, haciendo los costos de simulación de fallas para un banco de pruebas muy costosa, es por esto que los modelos matemáticos están involucrados para simplificar el inconveniente de recopilación. Dichos modelos se conectan a los mecanismos de degradación con las variables de conducto de flujo, ayudando de esta manera a una clasificación de fallas necesaria para el diagnóstico.

La Figura 2.1 (Loboda & Yepifanov, 2010) representan una clasificación de las técnicas de análisis del conducto de flujo.

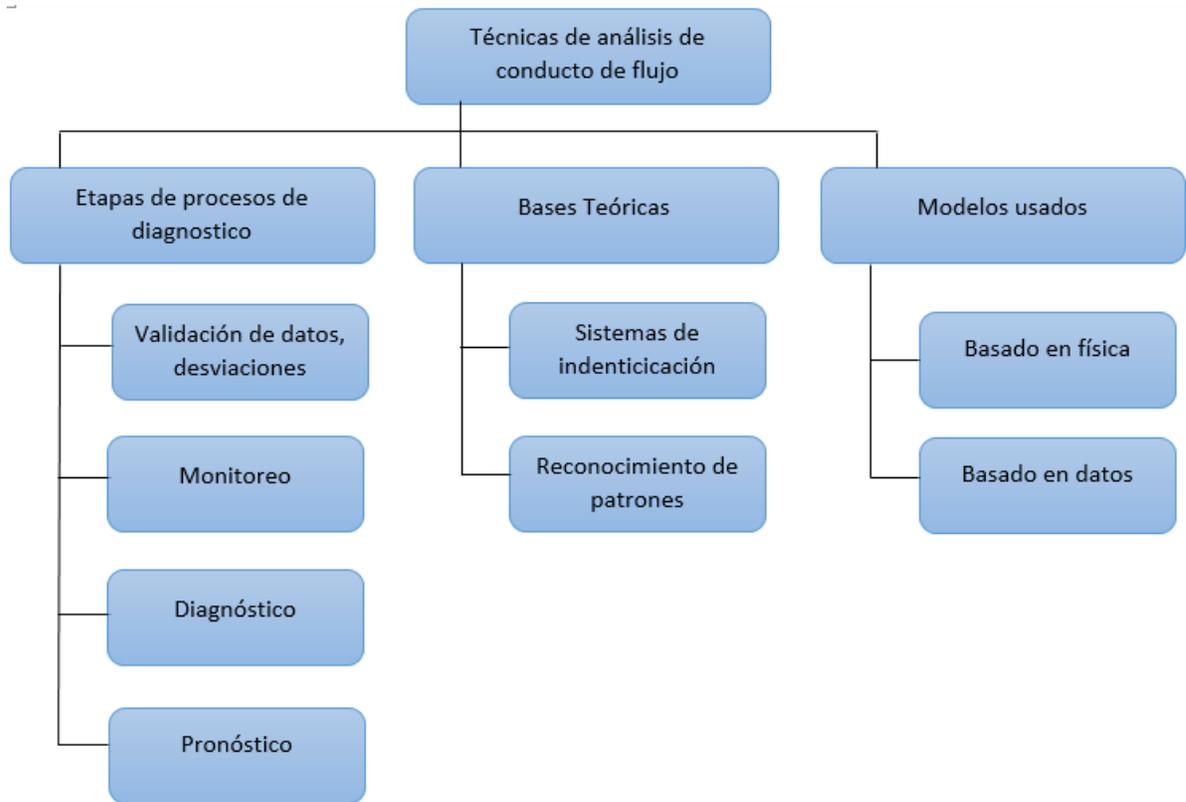


Figura 2.1 Clasificación de las técnicas de diagnóstico del conducto de flujo.

A continuación se da una breve descripción de las técnicas ACF.

2.2.1 Etapas del proceso de diagnóstico

Un proceso de diagnóstico total se puede dividir en tres etapas generales e interrelacionados: Supervisión de motores comunes de salud (detección de fallas), un diagnóstico detallado de identificación (falla), y pronóstico (predicción de vida útil restante del motor). Puesto que los datos de entrada deben ser máximamente exactos, una importante etapa preliminar de validación de datos precede a estas etapas de diagnóstico principales. Adicionalmente a la filtración de datos y el cálculo promedio, también se incluye un procedimiento de cálculo de las desviaciones, que se utilizan prácticamente en todos los métodos de monitoreo, diagnósticos y pronósticos.

2.2.2 Bases teóricas del ACF

Se tienen dos enfoques para el análisis del conducto de flujo que ayudan al diagnóstico correcto de la turbina de gas. El primer enfoque utiliza técnicas de identificación del sistema que por lo general significa identificación del modelo termodinámico. Los modelos utilizados relacionan las variables monitoreadas del conducto de flujo con los parámetros de falla que

permiten la simulación de la degradación de los componentes del motor. El objetivo de la identificación para la turbina de gas es encontrar tales fallas que reduzcan al mínimo la diferencia entre el modelo generado y las variables monitoreadas medidas. La simplificación del proceso de diagnóstico se logra porque las fallas determinadas contienen información sobre el estado actual de cada componente.

El segundo enfoque se basa en la teoría de reconocimiento de patrones y en su mayoría utiliza modelos basados en datos. La clasificación de fallas necesaria puede ser compuesta en cada clase de falla y se forma a partir de datos reales o simulados. Cuando se fija el conjunto para cada clase de fallas una técnica de reconocimiento basada en datos es aplicado, se hace uso de las redes neuronales ya que pueden ser fácilmente manipuladas sin un conocimiento detallado del sistema.

2.2.3 Modelos usados para ACF

Los modelos utilizados en el análisis de conducto de flujo se pueden dividir en dos categorías, basadas en la física y basadas en datos. El modelo basado en la física (modelo termodinámico) requiere un conocimiento detallado del sistema (turbina de gas) y en general presenta un software más o menos complejo. El modelo basado en datos da una relación entre las variables de entrada y salida que se pueden obtener sobre la base de los datos reales o simulados disponibles sin la necesidad de conocimiento del sistema.

2.3 Modelo del diagnóstico para ACF

2.3.1 Introducción a los modelos

Hoy en día se encuentran disponibles varios modelos matemáticos para el uso en el diagnóstico. Los más representativos son el modelo estático no lineal, modelo estático lineal, modelo dinámico no lineal y el modelo por polinomios.

Desde los estudios en (Saravanamuttoo & Lakshminarasimha, 1985), la aplicación del modelo termodinámico para los estados estacionarios se ha convertido en una práctica común. Dicho modelo incluye una descripción completa sucesiva de todos los componentes del conducto de flujo como dispositivo de entrada, el compresor, cámara de combustión, turbina, y dispositivo de salida. Este modelo también puede ser clasificado como no lineal, unidimensional, y basada en componentes.

El modelo termodinámico calcula un vector \vec{Y} (mx1) de las variables monitoreadas del conducto de flujo como una función de un vector de \vec{U} de condiciones de operación (variables de control y las condiciones ambientales), así como un vector $\vec{\theta}$ (rx1) de los parámetros de falla, que puede también ser nombradas como parámetros de salud o factores de corrección en función de los problemas de direccionamiento. Dada la explicación anterior, el modelo termodinámico o modelo estático no lineal tiene la siguiente estructura:

$$\vec{Y} = F(\vec{U}, \vec{\theta}) \quad (2.1)$$

El modelo termodinámico representa un software complejo. El número de ecuaciones algebraicas puede llegar a 15 o más y el software incluye docenas de subprogramas. La mayor parte de los subprogramas pueden ser diseñados como módulos universales independientes de una turbina de gas simulada, lo que simplifica la creación de un nuevo modelo de motor.

El modelo estático lineal presenta la linealización de la dependencia no lineal entre las variables del conducto de flujo y los parámetros de falla determinados en una condición fija de funcionamiento \vec{U} . El modelo está dado por una expresión vectorial:

$$\delta\vec{Y} = H\delta\vec{\theta} \quad (2.2)$$

Se conecta un vector $\delta\vec{\theta}$ de pequeños cambios relativos de los parámetros de falla con un vector $\delta\vec{Y}$ correspondiente a las desviaciones relativas de las variables monitorizadas por una matriz H de coeficientes de influencia.

La matriz H puede ser fácilmente calculada por medio del modelo termodinámico. Las variables del conducto de flujo \vec{Y} son primeramente calculados por el modelo para parámetros de falla nominal $\vec{\theta}_0$. A continuación, las pequeñas variaciones son introducidas por turnos en los parámetros de fallo y el cálculo de las variables \vec{Y} se repite para cada parámetro corregido. Por último, para cada pareja Y_i y θ_j el coeficiente de influencia correspondiente se obtiene por la siguiente expresión:

$$H_{ij} = \frac{\delta Y_i}{\delta \theta_j} = \left(\frac{Y_i(\vec{\theta}_j) - Y_i(\vec{\theta}_0)}{Y_i(\vec{\theta}_0)} \right) / \left(\frac{\theta_j - \theta_{0j}}{\theta_{0j}} \right) \quad (2.3)$$

Aunque los métodos para diagnosticar en estados estacionarios están más desarrollados y numerosos que los métodos transitorios, los estudios actuales demuestran un creciente interés en el diagnóstico de la turbina de gas durante el funcionamiento dinámico (Loboda, 2007) y (Ogaji & Sampath, 2003). Un modelo de conducto de flujo termodinámico (modelo dinámico) está en aumento de demanda. A diferencia del modelo estático no lineal Ec. (2.1), en el modelo dinámico, una variable de tiempo t es añadida para fijar el argumento de la función \vec{Y} y el vector \vec{U} esta dado como función del tiempo, un ejemplo de la estructura del modelo dinámico se presenta a continuación:

$$\vec{Y} = F(\vec{U}(t), \vec{\theta}, t) \quad (2.4)$$

Para los modelos descritos con anterioridad, está el modelo por polinomios, en donde (Loboda, 2007) ha demostrado que dan la suficiente aproximación para representar un motor sano de turbina de gas. Es por ello que se usa el modelo de polinomios debido a su simplicidad, ya que no es necesario saber en detalle todos los componentes de una turbina de gas. A continuación se da a conocer en mayor medida el concepto del modelo por polinomios

2.3.2 Modelo por Polinomios

Los polinomios de segundo orden dan aproximación suficiente para la representación de una turbina de gas. Con la supervisión de la variable del conducto de flujo Y , se da como ejemplo una función con tres argumentos de condiciones de operación U , tal polinomio tiene la forma siguiente:

$$Y(\vec{U}) = a_0 + a_1u_1 + a_2u_2 + a_3u_3 + a_4u_1u_2 + a_5u_1u_3 + a_6u_2u_3 + a_7u_1^2 + a_8u_2^2 + a_9u_3^2 \quad (2.5)$$

Se puede observar en la ecuación anterior la aparición de 10 coeficientes a , lo cual es el resultado de la combinación lineal de los argumentos de condiciones de operación U .

Los polinomios de todas las variables monitoreadas se pueden dar en la siguiente forma generalizada:

$$\vec{Y}_0^T = \vec{V}^T \mathbf{A} \quad (2.6)$$

donde \vec{Y}_0^T es un vector (1 x m) de variables monitoreadas, \vec{V}^T es un vector (1 x k) de los componentes 1, u_1 , u_2 , ..., u_2^2 , u_3^2 , y \mathbf{A} representa una matriz (k x m) de coeficientes desconocidos a_i para todas las variables monitoreadas m . Dado que las mediciones de los puntos de operación en estado estacionario no son suficientes para calcular los coeficientes, los datos recogidos en diferentes n puntos diferentes son incluidos dentro del conjunto de entrenamiento y envuelto dentro de los cálculos. Con las nuevas matrices $\mathbf{Y}(n \times m)$ y $\mathbf{V}(n \times k)$ formadas a partir de estos datos, la Ecuación 2.5 se transforma en un sistema lineal

$$\mathbf{Y} = \mathbf{V}\mathbf{A} \quad (2.7)$$

2.4 Concepto de desviación para el ACF

Las desviaciones, también conocidas como residuos o deltas, se definen como las diferencias entre los valores medidos y los de referencia del motor. Como la referencia depende de una condición de funcionamiento sano del motor, se puede escribir como $\vec{Y}_0(\vec{U})$ usualmente llamado función o modelo de referencia, dicha función tiene la estructura de la Ec. 2.5. Con este modelo, la desviación para cada una de las variables monitoreadas $Y_{i,i=1,m}$ es calculada de forma relativa:

$$\delta Y_i^* = \left(\frac{Y_i^* - Y_0(\vec{U})}{Y_0(\vec{U})} \right) \quad (2.8)$$

donde Y_i^* indica un valor medido.

La desviación consiste en la influencia sistemática (señal) inducida por la degradación o fallas del motor y un componente de ruido, lo que es explicado por errores en el sensor y un inadecuado modelo de referencia. Cuando es calculada correctamente, las desviaciones pueden tener una calidad relativamente alta (relación señal-ruido) y pueden ser potencialmente buenos indicadores de salud del motor. Dado el éxito de todas las principales etapas de diagnóstico,

depende directamente de la calidad de las desviaciones, los mejores esfuerzos deben aplicarse para mantener los errores de desviación a un mínimo nivel.

Las herramientas de instrumentación moderna y de registro de datos permiten la colección de un gran volumen de banco de pruebas y de datos de diferentes tipos. Normalmente, los datos de los sensores del motor son usados en el diagnóstico los cuales son previamente filtrados, promediados, y periódicamente registrados (una vez por vuelo, día u hora) en estado estacionario. Cada sección de medición (*snapshot*) incluye condiciones de operación del motor \vec{U} , que establece un punto de operación del motor, y monitorea las variables medidas \vec{Y} . Cuando se graba durante un largo período de tiempo, estas secciones de medición pueden proporcionar información valiosa sobre el deterioro y las fallas del sistema estudiado. La causa más común del deterioro de las turbinas de gas estacionarias es la contaminación del compresor, en donde los datos con ciclos de contaminación y lavado se utilizan ampliamente con el fin de verificar las técnicas de diagnóstico.

Mediante el análisis directo de las propias variables, es difícil discriminar los efectos de degradación del rendimiento de grandes cambios debido a diferentes modos de funcionamiento. Para sacar la información de diagnóstico útil a partir de los datos registrados en bruto, el proceso total de diagnóstico de una turbina de gas por lo general incluye un procedimiento preliminar de computación de las desviaciones. Una vez obtenidas las desviaciones se pueden generar las etapas subsecuentes para el diagnóstico, así como realizar la clasificación adecuada de datos para la manipulación necesaria.

2.5 Clasificación de fallas

Un sistema de ingeniería dejará de funcionar satisfactoriamente en un punto de su vida útil, las fallas dependen de la naturaleza del sistema, donde son inducidas por muchos factores, en algunos casos actúan solos y en otros en varias combinaciones de éstas. Varios parámetros como la vibración, descargas, ruido, calor, frío, polvo, corrosión, humedad, presión, etc., son conocidos por deteriorar las condiciones óptimas de operación de plantas, maquinaria y procesos industriales. Estos son varios medios para detectar y monitorear los primeros indicios de fallas en sistemas eléctricos, mecánicos, electrónicos, neumáticos, hidráulicos, etc., en todas sus manifestaciones y proveer una ayuda al diagnóstico para establecer tiempos de mantenimiento efectos para predecir y prevenir fallas en estos sistemas en tiempo. Una bien diseñada estrategia de monitoreo incorporada a un buen mantenimiento nos dará un método efectivo para reducir costos de producción, de operación y mano de obra. ProDiMES proporciona 18 fallas diferentes que pueden presentarse en una turbina de gas tipo turbo ventilador, y se apega a la teoría del análisis de conducto de flujo, en la Tabla 2.1 se muestran las fallas que pueden ser simuladas.

Tabla 2.1 Fallas para el análisis de conducto de flujo

ID	Descripción de la falla
1	Ventilador

2	Baja presión en el compresor
3	Alta presión en el compresor
4	Alta presión en la turbina
5	Baja presión en la turbina
6	Actuador del estator variable
7	Válvula variable de sangrado
8	Velocidad física del ventilador
9	Velocidad del núcleo
10	Presión total a la entrada del Compresor
11	Presión total a la salida del Compresor
12	Temperatura total a la entrada del Compresor
13	Temperatura total a la salida del Compresor
14	Temperatura total a la entrada de la Turbina
15	Flujo de combustible
16	Presión total a la entrada del ventilador
17	Temperatura a la entrada del ventilador
18	Presión ambiental

Para hacer posible un adecuado diagnóstico para la turbina de gas es necesario de una correcta clasificación de las fallas procedentes del banco de pruebas proporcionado por el software, ya que el banco de pruebas arroja cientos de datos individuales por cada motor simulado.

2.5.1 Principios para preparar la clasificación

Para los fines de diagnóstico del motor esta variedad de los datos dispersos, tiene que ser dividido en un número limitado de clases, donde estarán aislados por grupos individuales de falla. En la teoría de reconocimiento de patrones, supone que el estado del motor D solo puede pertenecer a una de las clases creadas q

$$D_1, D_2, \dots, D_q \quad (2.9)$$

El objetivo del diagnóstico será entonces la búsqueda de la clase D_q que esté más cerca del estado actual del motor D . Se asume que cada clase corresponde a un componente del motor y está descrita por los parámetros de fallas.

La presentación detallada de la clasificación depende del volumen de la información acerca de los estados posibles del motor y el tipo de modelo usado. Dentro de nuestro ámbito la clase se determina como todas las fallas correspondientes al componente del motor. Las clases se presentan por la función de densidad de desviaciones \vec{Z}^* (enfoque probabilístico) o por un conjunto de desviaciones o patrones (enfoque estadístico).

En la Figura 2.2 se muestra un esquema de clasificación de fallas, la clase puede estar descrita en la región Ω_j^* del espacio diagnóstico elegido \vec{Z}^* mediante la medición de distribución de la función de densidad $f(\vec{Z}^*/D_j)$. Cada parámetro diagnosticado Z_i es calculado como una

desviación del valor del estado actual Y_i y el valor del estado normal Y_{oi} . En la figura la clase D_1 esta formada por un parámetro de estado y clase D_2 por dos parámetros. Las regiones Ω_1 y Ω_2 incluyen los valores reales de los parámetros del conducto de flujo y las regiones Ω_1^* y Ω_2^* contienen los valores medidos.

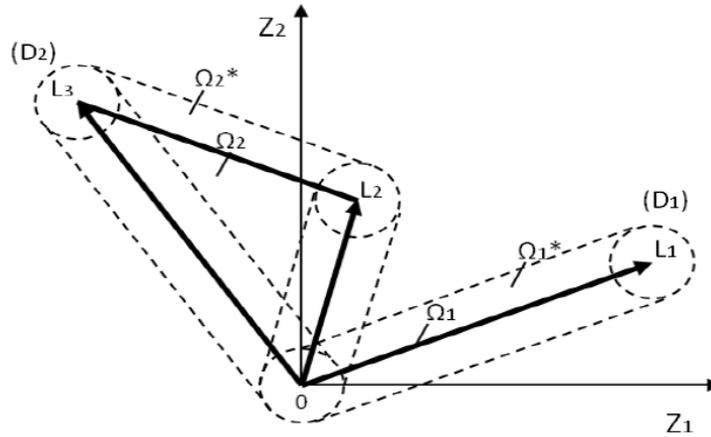


Figura 2.2 Clases de fallas múltiples y singular (Loboda, 2007)

También se pueden formar las clases de fallas como conjuntos de patrones, vectores de desviaciones \vec{Z}^* . De esta manera, los conjuntos de todas las clases se unen en una muestra de datos llamada muestra de aprendizaje. Se llama así porque se usan estos datos en el proceso de entrenamiento de las herramientas de reconocimiento, como es nuestro caso las redes neuronales.

Los errores de medición tienen una influencia en el grado de intersección de las regiones $\Omega_1^*, \Omega_2^*, \dots, \Omega_q^*$ y como resultado, en el nivel de reconocimiento de fallas y diagnóstico del motor. Una nomenclatura posible de diagnóstico d_1, d_2, \dots, d_q corresponde a la clasificación aceptada en (Ec. 2.9).

2.5.2 Probabilidades del diagnóstico correcto

Para verificar un algoritmo de diagnóstico determinado con la ayuda del conjunto de aprendizaje Z_L , un conjunto más es requerido, esto se hace mediante el conjunto Z_V llamado conjunto de validación.

Cada patrón en el conjunto de validación pertenece a una clase conocida. Por eso, al comparar esta clase con el D_j con el diagnóstico d_l , podemos calcular las probabilidades $Pd_{lj} = P(d_l/D_j)$ y componer una matriz de probabilidades (Tabla 2.2). Sus elementos de la diagonal Pd_{ll} forman un vector \vec{P} de probabilidades de diagnóstico correctos que son índices para distinguir las clases. Los elementos que no están en la diagonal ayudan a identificar las causas de una baja confiabilidad del diagnóstico. Estos elementos forman probabilidades de falso diagnóstico.

Tabla 2.2 Matriz de probabilidades de diagnóstico

Diagnóstico	Clases			
	D_1	D_2	...	D_j
d_1	Pd_{11}	Pd_{12}	...	Pd_{1j}
d_2	Pd_{21}	Pd_{22}	...	Pd_{2j}
...
d_l	Pd_{l1}	Pd_{l2}	...	Pd_{lj}

2.5.3 Matriz decisión de detección

Una matriz de decisión (2×2) refleja la capacidad de un algoritmo para la detección de los casos de falsas alarmas de las fallas a analizar. Su diagonal principal refleja el número de predicciones correctas (verdaderos positivos y verdaderos negativos) y sus elementos fuera de la diagonal reflejan el número de predicciones incorrectas (falsos negativos y falsos positivos).

Esto refleja la verdadera tasa positiva y la tasa negativa a lo largo de la diagonal principal, y la tasa de falsos positivos y falsos negativos en la diagonal como se muestra en la Tabla 2.3

Tabla 2.3 Matriz decisión de detección

	Do	Df
do	TNR	FNR
df	FPR	TPR

La nomenclatura de la matriz decisión de detección se toman de (UsersGuide ProDiMES 2009), en donde se describen de la siguiente manera:

TPR (*True Positive Rate*) refleja el número de fallas de detecciones correcto divididos por el número de casos de falla.

FNR (*False Negative Rate*) refleja el número de fallas de detecciones incorrectas divididas por el número de casos de fallas

FPR (*False Positive Rate*) refleja el número de fallas de detecciones incorrectas divididas por el número de casos de no fallas

TNR (*True Negative Rate*) refleja el número de fallas de detecciones incorrectas divididas por el número de casos de no fallas.

CAPÍTULO III

Software ProDiMES

3.1 Introducción

Hoy en día se ve reflejado un incremento de investigación en el área de administración de salud de motores o por sus siglas en inglés (*EHM-Engine Health Management*). ProDiMES es una herramienta para el desarrollo y evaluación de tecnología de diagnóstico en el conducto de flujo el cual es un software libre y que está abierta al público.

ProDiMES es el resultado del trabajo de varios investigadores dentro de la administración de salud de motores con el fin de desarrollar una herramienta que fuera capaz de simular una flota de motores de aviación. Para la correcta simulación se recogieron datos de un gran número de motores con gran rango de condiciones de operación, niveles de deterioro, condiciones conocidas de falla y sin falla. Con el fin de facilitar la evaluación comparativa se decidió utilizar un modelo estático de motor conocido como turbo-ventilador de alta derivación procedente de la NASA. Este modelo por sus siglas en inglés (*C-MAPSS Comercial Modular Aero-Propulsion System Simulation*) aplica un modelado Aero-termodinámico no lineal del conducto de flujo.

Con ProDiMES se evita el uso de motores reales para los cuales en muchas ocasiones la información requerida está limitada por sus mismos usuarios. Con el debido uso del software, se permite la simulación de un gran número de motores con un amplio rango de fallas y magnitudes como también con diferentes condiciones de operación. Es por ello que ProDiMES da pauta inicial para el desarrollo y evaluación de metodologías de diagnóstico del conducto de flujo.

ProDiMES tiene su código desarrollado mediante la plataforma de MATLAB. Para permitir a los usuarios el desarrollo independiente y la evaluación de soluciones de diagnóstico, la herramienta hace el uso de un simulador de flota de motores, o por sus siglas en inglés (*EFS-Engine Fleet Simulator*), a continuación se hace una descripción más detallada sobre el simulador.

3.2 Simulador de Flota de Motores (SFM)

SFM produce mediciones de motores con diferentes niveles de ruido, tanto para el estado de despegue y crucero del avión. Para emular los datos cada motor requerido experimenta perfiles de deterioro y operación únicos. El usuario es capaz de elegir la falla en los componentes principales de la turbina como también en sus sensores y actuadores. La Figura 3.1 ilustra la estructura del simulador de flota de motores.

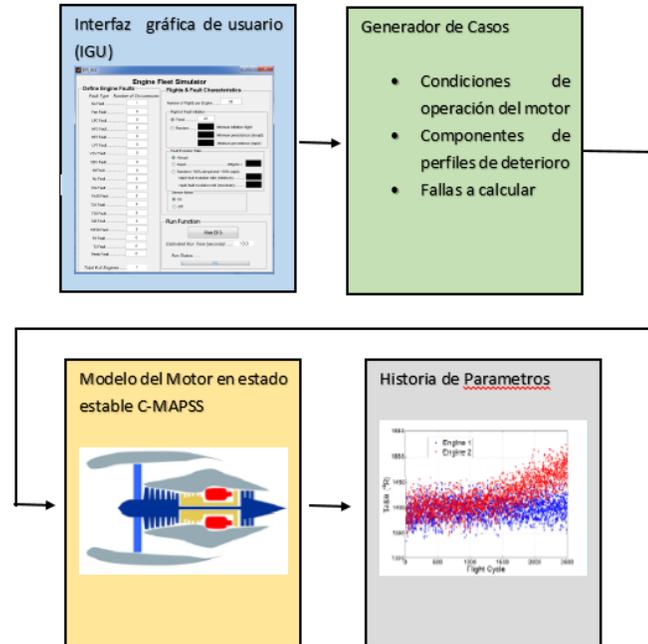


Figura 3.1 Arquitectura del Simulador de la flota de motores, (UsersGuide ProDiMES 2009)

La Figura 3.1 describe la Interfaz gráfica de usuario (IGU) donde acepta entradas específicas de acuerdo al número de motores en la flota, el número de ocurrencias de 18 diferentes tipos de fallas del conducto de flujo, el vuelo de inicio de falla, y la tasa de evolución de falla. El generador de casos se diseñó para introducir fallas únicas, perfiles de deterioro y perfiles de operación para cada motor en la flota. Usando el modelo C-MAPSS y perfiles introducidos se generan datos para toda la flota de motores.

3.3 Descripción de la Interfaz gráfica de usuario

La interfaz de usuario fue diseñada para ser flexible, es decir que tiene la facilidad de generar datos para propósitos de desarrollo y validación de métodos de diagnóstico. Es aquí donde se definen el número de fallas generadas para la flota de motores.

Se debe de destacar que en la simulación las fallas son singulares por motor, es decir que no se simulan fallas múltiples en el mismo motor, así que si se desean 20 fallas de un tipo, se simularan 20 motores con esa falla, y así sucesivamente para cada tipo de falla.

El número máximo de fallas que se pueden simular son 18, más un caso sin falla, lo que suma un total de 19 posibles motores individuales por falla individual. La interfaz también permite especificar lo siguiente: el número de vuelos sobre los cuales los datos de salida son registrados para cada motor; el vuelo de inicio de falla (ya sea en un número de vuelo fijo o aleatoriamente dentro de una ventana específica de vuelos); la tasa a la cual la falla evoluciona, ya sea abrupta (instantánea) o rápida (sobre un número de ciclos de vuelo); y ruido de sensor activado o desactivado. En la Figura 3.2 se refleja lo descrito.

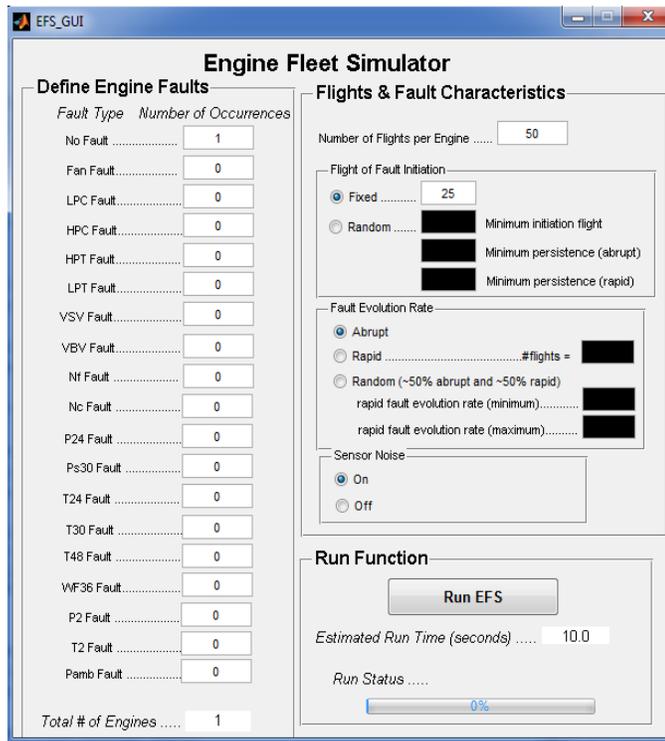


Figura 3.2 Interfaz de usuario, (ejecutando Simulador ProDiMES 2016)

En el lado izquierdo de la figura, se puede notar que están los 18 tipos de falla, más el caso sin falla. La suma de esas entradas define el número total de motores en la flota, el cual es mostrado en la parte inferior izquierda de IGU.

Se tiene el cuadro etiquetado con “*Number of Flights per Engine*”, el cual especifica el número de vuelos de datos que son generados para cada motor. Existe un rango de entradas que va desde un mínimo de un vuelo, hasta un máximo de 5000.

Se puede seleccionar el vuelo de inicio de falla. Los usuarios pueden seleccionar el vuelo “*fixed*” de inicio de falla, que indica que todos los motores en la flota experimentan falla en el mismo vuelo, o vuelo “*random*” de inicio de falla, que indica que el vuelo de inicio de falla es aleatorio asignado para cada motor.

En caso de escoger “*fixed*” de inicio de falla, el usuario tiene que especificar el número de vuelo, pero si se elige “*random*” de inicio de falla, el usuario debe especificar el vuelo posible mínimo de inicio de falla, y el mínimo número posible de vuelos que la falla persiste. Esas entradas limitan la ventana de vuelos sobre los cuales la falla puede iniciar.

La tasa de evolución de falla se refiere a que tan rápido evolucionan las fallas, o crecen en magnitud. El software ProDiMES proporciona 3 diferentes tasas de evolución de falla: abrupta, rápida, y aleatoria. Las fallas que aparecen instantáneamente, pero que no crecen en magnitud

a lo largo del tiempo, son consideradas como tasas de fallas abruptas. Las fallas que inician y crecen en magnitud a lo largo del tiempo son consideradas como fallas de tasas rápidas. Para la tasa rápida, las fallas incrementan linealmente en su magnitud, desde el vuelo de inicio de falla hasta el vuelo donde la falla no cambia en magnitud. Para ambas tasas de evolución, están asignadas las distribuciones de magnitud de falla mostrada en la Tabla 3.1.

Tabla 3.1 Fallas posibles en el simulador

ID	Descripción de la falla	Magnitud de la falla
0	Sin-falla	---
1	Falla Fan	1 a 7%
2	Falla LPC	1 a 7%
3	Falla HPC	1 a 7%
4	Falla HPT	1 a 7%
5	Falla LPT	1 a 7%
6	Falla VSV	1 a 7%
7	Falla VBV	1 a 19%
8	Falla de sensor Nf	± 1 a 10σ
9	Falla de sensor Nc	± 1 a 10σ
10	Falla de sensor P24	± 1 a 10σ
11	Falla de sensor Ps30	± 1 a 10σ
12	Falla de sensor T24	± 1 a 10σ
13	Falla de sensor T30	± 1 a 10σ
14	Falla de sensor T48	± 1 a 10σ
15	Falla de sensor Wf	± 1 a 10σ
16	Falla de sensor P2	± 1 a 10σ
17	Falla de sensor T2	± 1 a 10σ
18	Falla de sensor Pamb	± 1 a 19σ

Las primeras 5 fallas corresponden al ventilador(Fan), Compresor de Baja Presión (LPC), Compresor de Alta Presión (HPC), Turbina de Alta Presión (HPT), y Turbina de Baja Presión (LPT). Estas fallas mencionadas se ajustan con los parámetros de falla: eficiencia y capacidad de flujo del componente de motor. Las fallas 6 y 7, corresponden a actuadores; estator variable (VSV) y válvula variable de sangrado (VBV). Las fallas restantes, 8 a 18, representan defectos en los sensores; velocidad física del ventilador (Nf), velocidad fija del núcleo (Nc), Presión total a la entrada en LPC (P24), Presión total a la salida en HPC (Ps30), Temperatura total a la entrada en LPC (T24), Temperatura total a la salida en HPC (T30), Temperatura total a la entrada en HPT (T48), flujo de combustible (Wf), Presión total a la entrada del ventilador (P2), Temperatura a la entrada del ventilador (T2), y Presión ambiental (Pamb).

El botón “*Sensor Noise*” sirve para activar o desactivar el ruido en los sensores. Si el ruido de sensor es seleccionado, cada motor experimenta ruido de sensor distribuido normalmente con media cero (medición no repetida). Después de que todas las entradas necesarias han sido especificadas, los usuarios pueden correr EFS presionando el botón “*Run EFS*”. Una vez presionando este motor, los usuarios son consultados para determinar si cualquier archivo de salida existente de EFS en el directorio actual debe ser sobrescrito o no. Si se selecciona “*Yes*”, los archivos existentes serán sobrescritos. Si se selecciona “*No*”, se interrumpe el programa

dando oportunidad a los usuarios de archivar los archivos de salida de EFS generados previamente que deseen.

3.4 Generador de casos

Después de haber especificado los parámetros deseados y presionar el botón “*Run EFS*”, inicia el proceso de generación de historia de parámetros para cada motor. El generador de casos asigna aleatoriamente un perfil de deterioro e historia de operación únicos para cada motor.

Las condiciones de operación como altitud, presión, número de Mach, temperatura ambiente y los parámetros para la configuración de poder en crucero y despegue son generados aleatoriamente para simular un vuelo de avión comercial.

Para ver de forma representativa como varían las condiciones de operación, se muestran las distribuciones de los parámetros usados para el simulador de la flota de motores en la Figura 3.3 y 3.4.

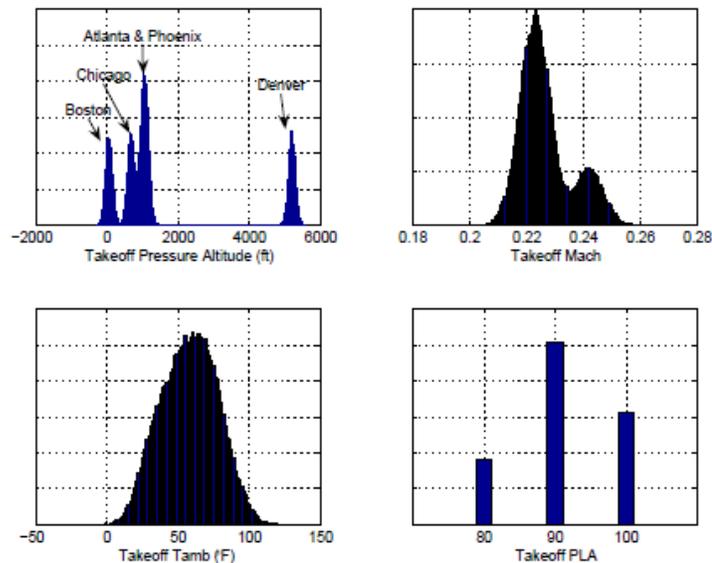


Figura 3.3 Distribuciones de condiciones de operación en despegue, (UsersGuide ProDiMES 2009)

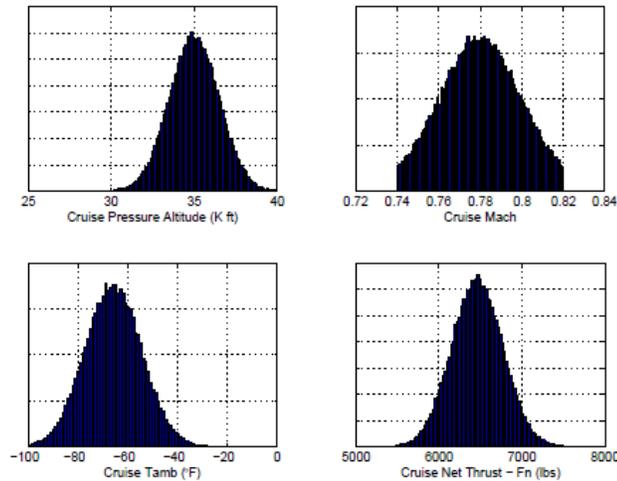


Figura 3.4 Distribuciones de condiciones de operación en crucero, (UsersGuide ProDiMES 2009)

Estas condiciones así como variables medidas en conducto se registran en ProDiMES, cada motor tiene uno de los 19 tipos de fallas a simular, incluyendo el estado sin falla (ver Figura 3.2 y Tabla 3.1).

El simulador incluye una tasa de deterioro gradual por cada motor, ya que cada motor de avión tiene un deterioro natural debido a la contaminación, erosión y corrosión de los alabes estatores y rotores de la turbo máquina, y que se ve reflejado a lo largo de vida de uso. Para cuestiones de diagnóstico este deterioro no se considera como falla, ya que tiene una tasa menor de desarrollo comparado con las fallas. Un motor de avión real se deteriora en función de su uso y del ambiente en el que opera, es decir que dos motores no se deterioran a la misma tasa o a lo largo del mismo perfil.

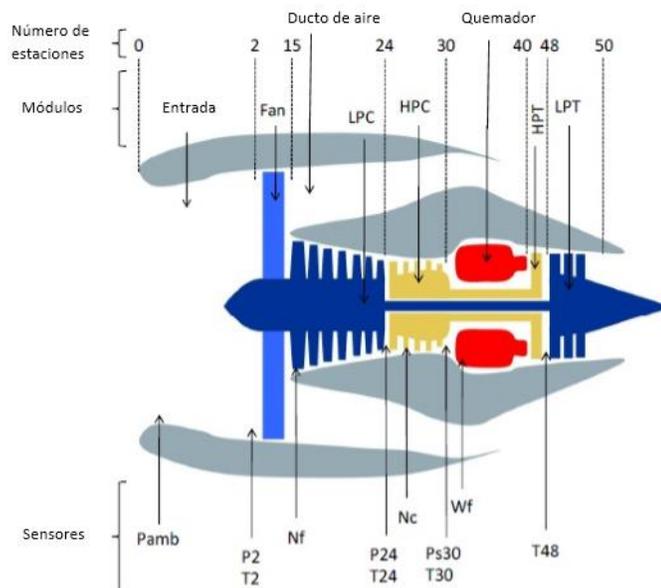
3.5 C-MAPSS

C-MAPSS *Comercial Modular Aero-Propulsion System Simulation* (C-MAPSS) es el modelo estático ejecutado dentro de ProDiMES, trasciende de un modelo no-lineal, desarrollado para un motor turbo-ventilador con el objetivo de investigación en control y diagnóstico. El modelo genera datos de motor para las distintas fallas por cada motor, cada vuelo, cada condición de operación ya sea de crucero o despegue, especificado por el generador de casos. Este modelo tiene dos estados, velocidad del ventilador y del núcleo, y tres actuadores que incluyen el flujo de combustible, álabes estatores variables y una válvula de sangrado variable.

Una versión en estado estacionario modificada de C-MAPSS, también llamada C-MAPSS en estado estacionario, es usada por EFS. Este modelo en estado estacionario ha sido validado en contra del modelo C-MAPSS original. La versión del estado estacionario tiene muchas diferencias notables del original: el sistema de control de ciclo cerrado se ha removido, el código ha sido convertido de Simulink a un archivo MATLAB Mex, y el solucionador transitorio ha sido remplazado por un solucionador en estado estacionario para permitir la convergencia rápida de los puntos de operación. C-MAPSS en estado estacionario se ejecuta dentro de EFS para

generar datos de salida simulados para cada motor, en cada vuelo, para cada condición de operación de despegue y crucero, especificados por el generador de casos.

En la Figura 3.5, se muestra el número de estaciones, con respecto a la localización de los sensores y módulos del motor.



3.5 Números de estaciones, módulos y sensores de C-MAPSS en estado estacionario, (UsersGuide ProDiMES 2009)

3.6 Parámetros de salida del simulador

Cuando ProDiMES hace la simulación de los motores con sus fallas correspondientes, este arroja un archivo de salida llamado “EFS_Output.mat” el cual contiene dos archivos llamados “efs_output_to” y “efs_output_c”, estos tienen el tamaño de 3 dimensiones (#motores x #vuelos x 11 parámetros medidos), para “efs_output_to” tiene los datos para la condición despegue y “efs_output_c” la condición para crucero. Los 11 parámetros medidos que arroja ProDiMES son los de la Tabla 3.2.

Tabla 3.2 Parámetros de salida por motor

ID	Símbolo	Descripción	unidad
1	Nf	Velocidad física del ventilador	rpm
2	Nc	Velocidad física del núcleo	rpm
3	P24	Presión total a la salida LPC	psia
4	Ps30	Presión estática a la salida de HPC	psia
5	T24	Temperatura total a la salida de LPC	°R
6	T30	Temperatura total a la salida de HPC	°R
7	T48	Temperatura total a la salida de HPT	°R
8	Wf	Flujo de combustible	pps
9	P2	Presión total a la entrada del ventilador	psia
10	T2	Temperatura a la entrada del ventilador	°R
11	Pamb	Presión ambiental	psia

CAPÍTULO IV

Redes Neuronales Artificiales

4.1 Introducción

La inteligencia artificial (IA) es la rama de la informática que estudia y desarrolla sistemas inteligentes, intenta comprender la filosofía y la psicología que conciernen con la inteligencia, con el que se esfuerza para construir entidades también como se pueda entender (Russel & Norving, 2003).

Los objetivos centrales de la investigación en IA incluyen el razonamiento, el conocimiento, la planificación, el aprendizaje, la comunicación, la percepción y la capacidad de moverse y manipular objetos. Hay un gran número de herramientas que se utilizan en la IA, incluyendo versiones de búsqueda y optimización matemática, lógica, métodos basados en la probabilidad y la economía, y muchos otros.

Este campo se basa en la afirmación de que una habilidad central de los seres humanos, la inteligencia, puede ser descrita de manera que puede ser simulada por una máquina. Esto plantea cuestiones filosóficas acerca de la naturaleza de la mente y la ética de crear seres artificiales, cuestiones que han sido abordadas por el mito, la ficción y la filosofía desde la antigüedad. La inteligencia artificial ha sido objeto de gran optimismo, pero también ha sufrido retrocesos impresionantes. Hoy en día se ha convertido en una parte esencial de la industria, tecnología y muchos de los problemas más difíciles en la informática.

4.2 Neurona Biológica

La neurona es la unidad fundamental del sistema nervioso y en particular del cerebro. Cada neurona es una simple unidad procesadora que recibe y combina señales desde y hacia otras neuronas. Si la combinación de entradas es suficientemente fuerte la salida de la neurona se activa.

Para saber la configuración de una neurona véase la Figura 4.1. Donde consta de un cuerpo celular y un núcleo, también se puede observar que hay otros elementos específicos, como es el axón, que es una ramificación de salida de la neurona, a través de él se propagan una serie de impulsos electro-químicos. Se puede ver que la neurona cuenta con un gran número de ramificaciones de entrada, estas se llaman dendritas, y se encargan de propagar la señal al interior de la neurona (Kriesel, 2005).

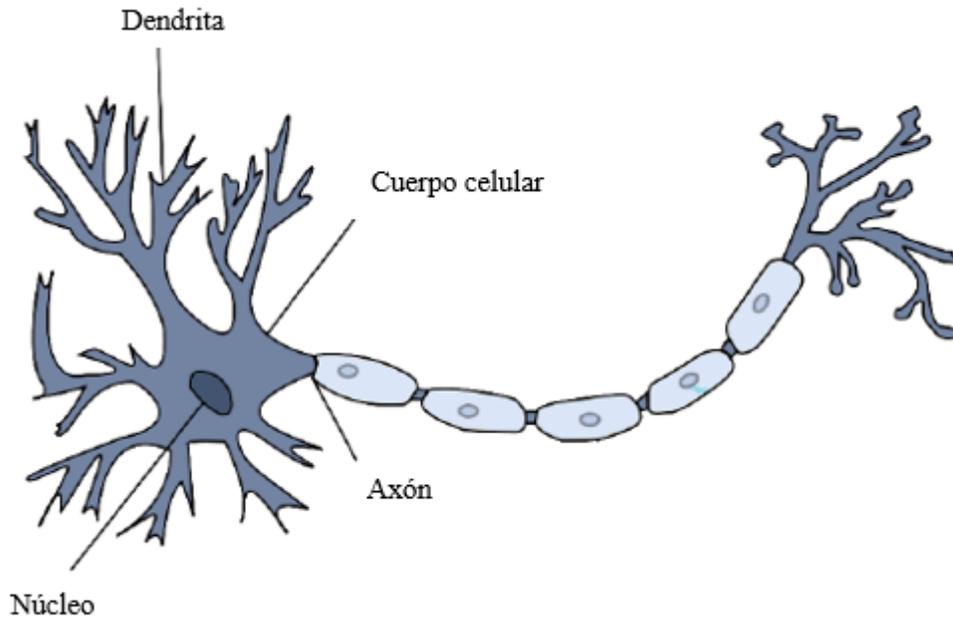


Figura 4.1 Representación de una neurona biológica, modificado de (Kriesel, 2005)

Una de las características más impresionantes de las redes neuronales artificiales es su capacidad para aprender. La Red Neuronal Artificial se inspira en el sistema nervioso biológico, en particular, el cerebro humano. Se cree que durante el proceso de aprendizaje la estructura neural del cerebro se altera, aumentando o disminuyendo la fuerza de sus conexiones sinápticas en función de su actividad. La información más relevante es más fácil de recordar, ya que tiene conexiones sinápticas más fuertes mientras que la información menos relevante tendrá conexiones sinápticas más débiles, por lo que es más difícil de recordar.

4.3 Neurona artificial

Existen modelos muy diversos de redes de neuronas las cuales siguen filosofías de diseño, reglas de aprendizaje y funciones de construcción de respuestas muy distintas. Una primera clasificación se hace en función del recorrido que sigue la información dentro de la red, y así se distinguen redes alimentadas hacia adelante y redes con retro-alimentación.

Las neuronas poseen una función que les permiten cambiar de nivel de activación a partir de las señales que reciben; a dicha función se le denomina función de transición de estado o función de activación. Las señales que recibe cada neurona pueden provenir del exterior o de las neuronas a las cuales está conectada. El nivel de activación de una célula depende de las entradas recibidas y de los valores sinápticos (Isasi & Galván, 2004).

La Figura 4.2 muestra un modelo típico para una neurona artificial. Aquí un grupo de entradas $x_1, x_2 \dots x_n$ son introducidas en una neurona artificial. Estas entradas, definidas por un vector X , corresponden a las señales de la sinapsis de una neurona biológica. Cada señal se

multiplica por un peso asociado w_1, w_2, \dots, w_n antes de ser aplicada a la sumatoria. Cada peso corresponde a la fuerza de una conexión sináptica, es decir el nivel de concentración iónica de la sinapsis, y se representa por un vector W .

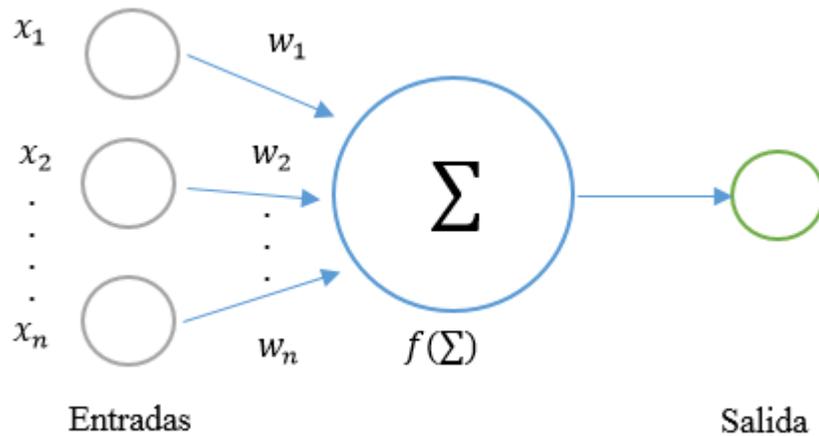


Figura 4.2 Esquema de una neurona con proceso típico.

La sumatoria que corresponde al cuerpo de la neurona, suma todas las entradas ponderadas algebraicamente, produciendo una salida que se denomina E , como en la siguiente forma:

$$E = x_1w_1 + x_2w_2 + \dots + x_nw_n \quad (4.1)$$

Cuando las células se conectan entre sí, se le denomina patrón de conectividad o arquitectura de la red. La estructura básica de la interconexión entre células es la red multicapa, mostrado en la Figura 4.3. Esta es la representación de una estructura típica conocido como Retro-propagación.

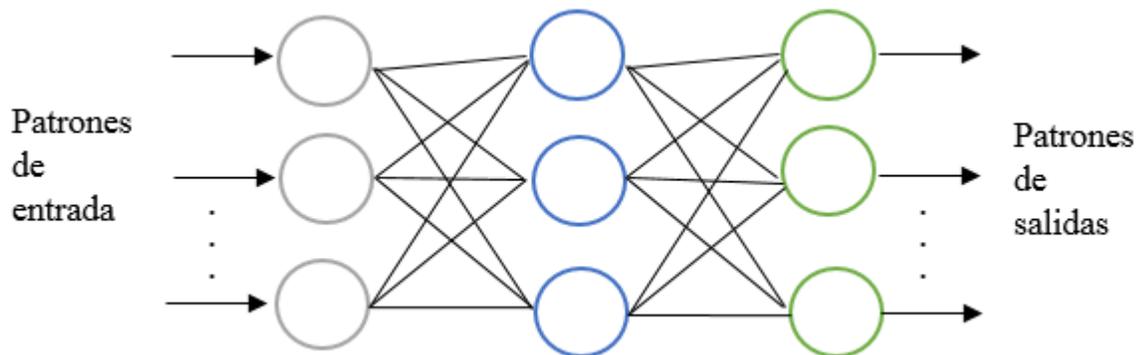


Figura 4.3 Esquema básico de una red.

A la izquierda se ve una serie de entradas a la neurona; cada una llega de la salida de otra neurona de la red. Una vez calculada la salida de una neurona, ésta se propaga, vía conexiones de salida, a las células destino. Todas las conexiones de salida reciben el mismo valor de salida.

El primer nivel lo constituyen las células de entrada, estas unidades reciben los valores de unos patrones representados como vectores que sirven de entrada a la red. En la parte central, se les conoce como capas ocultas, cuyas unidades corresponden a rasgos particulares que pueden aparecer en los patrones de entrada, de estas capas intermedias puede haber una o varias. El último nivel es el de las células de salida y este sirve como patrones de salida de toda la red.

4.4 Funciones de activación de las redes neuronales

La función f es llamada función de activación, las más utilizadas son la función sigmoideal y la función tangente hiperbólica. Dichas funciones poseen como imagen un rango continuo de valores dentro de los intervalos $[0,1]$ y $[-1,1]$, respectivamente, y vienen dadas por las siguientes expresiones:

Función sigmoideal:

$$f_1(x) = \frac{1}{1 + e^{-x}} \quad (4.2)$$

Función tangencial hiperbólica

$$f_2(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (4.3)$$

El comportamiento de estas funciones se describe mediante las siguientes figuras.

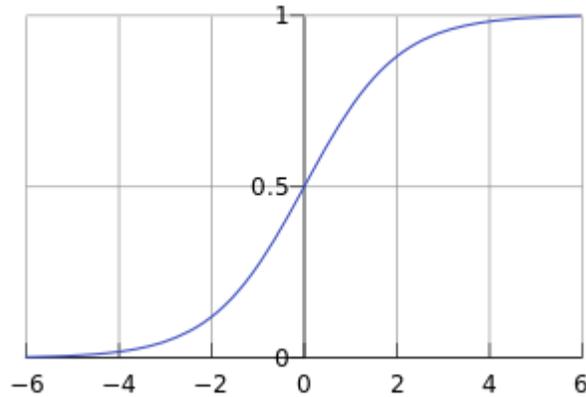


Figura 4.4 Función sigmoidea $f_1(x)$, (Internet, Google Imágenes)

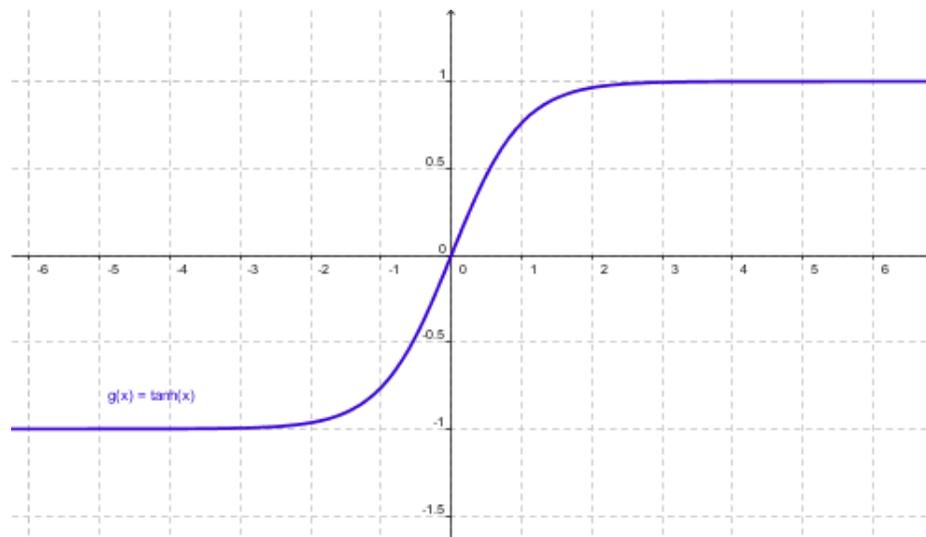


Figura 4.5 Función tangencial $f_2(x)$, (Internet, Google Imágenes)

4.5 Tipos de Redes Neuronales

4.5.1 Perceptrón Multicapa

Un Perceptrón Multicapa (MLP: *Multilayer Perception*) es un modelo de red neuronal artificial *feedforward* que mapea conjuntos de datos de entrada en un conjunto de salidas apropiadas. Un MLP se compone de varias capas de nodos de un grafo dirigido, con cada capa totalmente conectada a la siguiente (Figura 4.6). A excepción de los nodos de entrada, cada nodo es una neurona (o elemento de tratamiento) con una función de activación lineal. MLP utiliza una técnica de aprendizaje supervisado denominado retro propagación para la formación de la red. MLP es una modificación de la Perceptrón lineal estándar y puede distinguir los datos que no son separables linealmente.

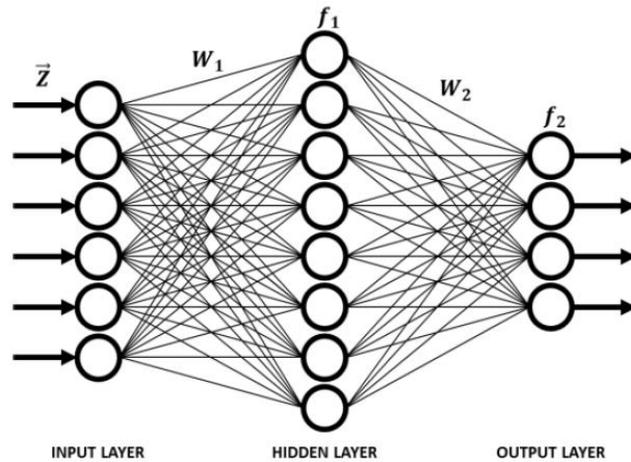


Figura 4.6 Estructura de la red de Perceptrón Multicapa, (Pérez & Loboda, 2014).

Capas

El Perceptrón multicapa consta de tres o más capas (una entrada y una capa de salida con una o más capas ocultas) de nodos de forma no lineal que activan y por lo tanto se considera una red neural de profundidad. Cada nodo en una capa se conecta con un cierto w_{ij} peso a cada nodo en la capa siguiente.

Aprendizaje

El algoritmo de aprendizaje de retro propagación se puede dividir en dos fases: la propagación y actualización de peso.

Fase 1: Propagación

Cada propagación implica los siguientes pasos:

1. Propagación hacia adelante de la entrada de un patrón de entrenamiento a través de la red neuronal con el fin de generar activaciones de salida de la propagación.
2. Propagación hacia atrás de la activación de salida de propagación a través de la red neuronal utilizando el objetivo de patrón de entrenamiento con el fin de generar los deltas de todas las salidas.

Fase 2: Actualización de Peso

Para cada sinapsis de peso se siguen los siguientes pasos:

1. Se multiplica la delta de salida y la activación de entrada para obtener el gradiente del peso.
2. Se resta una relación (en porcentaje) del gradiente del peso. Esta proporción (porcentaje) influye en la velocidad y la calidad del aprendizaje; se llama la tasa de aprendizaje. Cuanto mayor sea la relación, más rápidos que los trenes de neuronas; cuanto menor sea la relación, más precisa será el entrenamiento. El signo del gradiente de un peso indica

donde está aumentando el error; es por esto que el peso debe ser actualizado en la dirección opuesta.

Para que el rendimiento sea satisfactorio, se repiten las Fases 1 y 2.

4.5.2 Red de Base Radial

En el campo de la modelización matemática, la Red de Base Radial (*Radial Basis Network-RBN*) es una red neuronal artificial que utiliza funciones de base radial como funciones de activación. La salida de la red es una combinación lineal de funciones de base radial de las entradas y los parámetros de las neuronas (Broomhead & Lowe, 1988). Estas redes tienen muchos usos, incluyendo la función de aproximación, la predicción de series de tiempo, clasificación y control del sistema.

La Función de Base Radial (FBR) tiene típicamente tres capas: una capa de entrada, una capa oculta con una función de activación no lineal y una capa de salida lineal. La entrada puede ser modelada como un vector de números reales $x \in \mathbb{R}^n$. La salida de la red es entonces una función escalar del vector de entrada, $f: \mathbb{R}^n \rightarrow \mathbb{R}$, y viene dada por:

$$f(x) = \sum_{i=1}^N w_i h(\|x - c_i\|) \quad (4.5)$$

donde N es el número de neuronas en la capa oculta, c_i es el vector centro de la neurona i , y w_i es el peso de la neurona i en la neurona de salida lineal. Las funciones que dependen sólo de la distancia a partir de un vector centro son radialmente simétricas sobre ese vector, de ahí el nombre de función de base radial. En la forma básica todas las entradas están conectadas a cada neurona oculta. La norma se toma típicamente como la distancia euclidiana y la función de base radial h comúnmente se toma como Gauss (Figura 4.7).

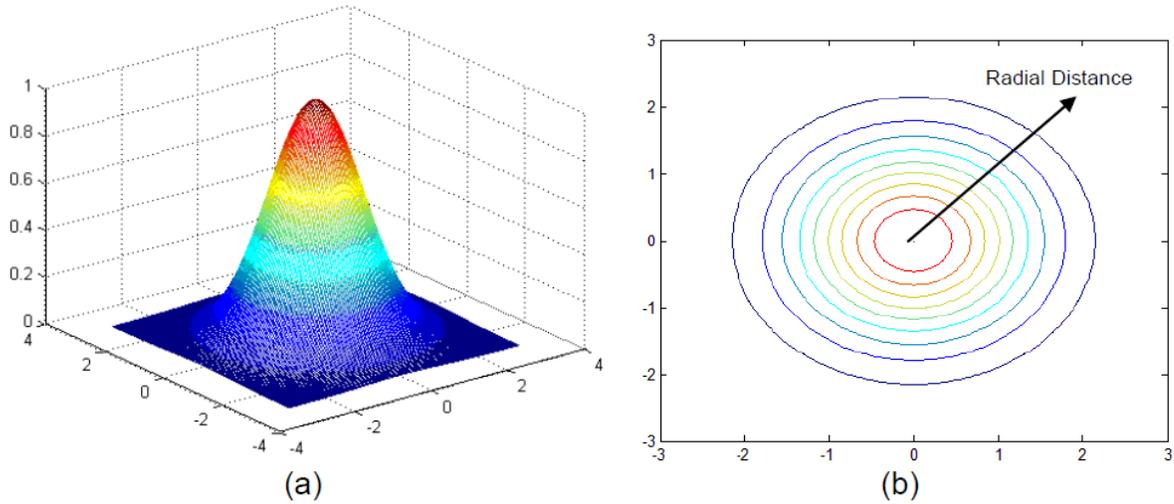


Figura 4.7 Área de influencia de una neurona Gaussiana con spread=1 (a), Función de base radial (b), (Pérez & Loboda, 2014)

Capas de la Red de base Radial

Capa de entrada: Esta capa alimenta cada una de las neuronas de la capa oculta con los valores de entrada. Entre la entrada y la capa oculta no hay ningún peso.

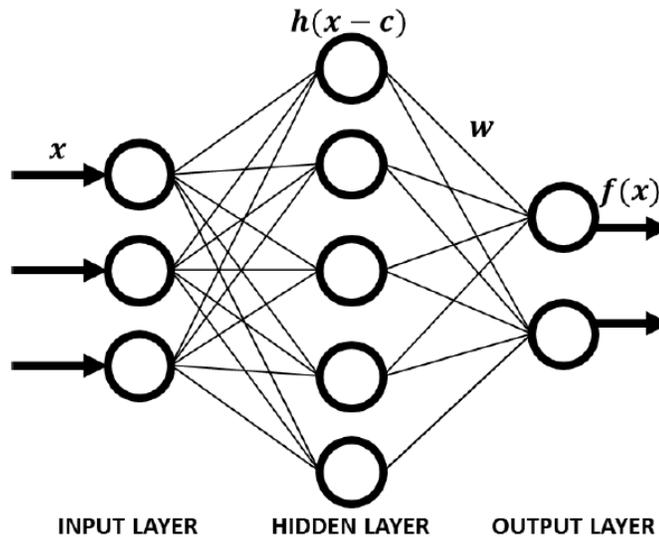


Figura 4.8 Estructura de una red de base radial, (Pérez & Loboda, 2014)

Capa oculta: Esta capa tiene un número variable de neuronas (el número óptimo se determina por el proceso de formación). Cada neurona consiste en una función de base radial centrada en un punto con el mayor número de dimensiones, ya que hay variables de predicción. La propagación de la función de la red de base radial puede ser diferente para cada dimensión. Los centros y los diferenciales son determinados por el proceso de formación. Cuando es presentado el vector de entrada de las capas de entrada, una neurona oculta calcula la distancia euclidiana del caso de prueba desde el punto central de la neurona y luego se aplica la función de base

radial al núcleo de esta distancia utilizando los valores de cálculo. El valor resultante se pasa a la capa de salida.

Capa de salida: El valor de la salida de una neurona en la capa oculta se multiplica por un peso asociado con la neurona y se pasa a la suma que añade los valores ponderados y presenta esta suma como la salida de la red. Para problemas de clasificación, hay una salida para cada categoría objetiva, cuando se le da el valor de 1 la salida pertenece a la categoría, y cuando es 0 no hay pertenencia mutua.

4.5.3 Red Neuronal Probabilística

Una red neuronal probabilística (*Probabilistic Neural Network - PNN*), un tipo específico de redes de base radial, está destinado a los problemas de clasificación. Como puede verse en la Figura 4.9 se muestra la estructura de la PNN que consta de tres capas.

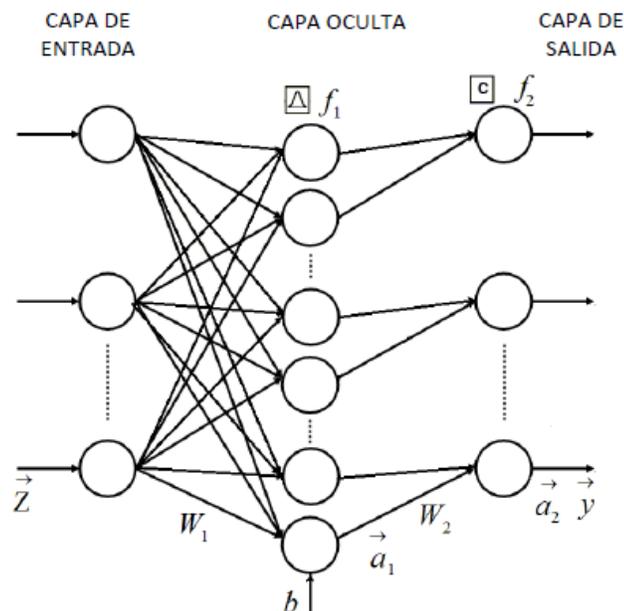


Figura 4.9 Estructura de la red neuronal probabilística, (Pérez, et al., 2014).

La capa oculta se forma y funciona igual que la misma capa del algoritmo de base radial. La capa se construye a partir de los patrones unidos en una matriz de aprendizaje W_1 . Cada patrón de aprendizaje \vec{w}_{1j} especifica un centro de la red de base radial de una neurona oculta. Una distancia euclidiana L_j entre el centro de la función \vec{w}_{1j} y el patrón entrada \vec{x} es calculado en primer lugar. La distancia L_j dividida por la propagación b es una entrada de neurona oculta. Y una función de base radial f_{1j} calcula entonces una salida de la neurona $a_{1j} = f_{1j} = e^{-L_j^2/b^2}$. Cuanto más cerca del vector de entrada esté situado del centro de la neurona, mayor será la salida de la neurona. Por lo tanto, los elementos de un vector de \vec{a}_1 indican cuán cerca está el patrón de entrada a los patrones de aprendizaje.

Las salidas o diferentes capas de clasificación de la capa de salida de la red de base radial. Cada neurona de clasificación corresponde a una de las clases analizadas. Una entrada de la neurona de clasificación, se interpreta como una probabilidad de esta clase, se calcula como una suma de las señales a_{1j} relacionadas con los patrones de aprendizaje de la clase. En otras palabras, cada neurona oculta está conectada con sólo una neurona de clasificación y los pesos de conexión son igual a uno. Para realizar una conexión de este tipo, una matriz W_2 es compuesto de una manera particular desde cero y de elementos de uno, para después el producto $W_2\vec{a}_1$ pueda ser calculado. Este es un vector de entrada para la capa de clasificación y se compone de las probabilidades de todas las clases consideradas. Por último, la capa de clasificación de la función de transferencia f_2 produce un 1 correspondiente a la mayor probabilidad, y 0 para las otras salidas de la red. Por lo tanto, el PNN clasifica el vector de entrada \vec{x} en una clase específica sobre la base de medida de probabilidad. Teniendo en cuenta que esta red toma decisiones probabilísticos en vez de determinísticos, es más cercano a la realidad que el Perceptrón.

4.6 Teoría de reconocimiento de patrones

El reconocimiento de patrones es una rama de la inteligencia artificial que se centra en los procesos de ingeniería, la informática y las matemáticas relacionadas con los objetos físicos o abstractos con el fin de extraer información y establecer las propiedades entre conjuntos de este tipo de objetos (también llamados patrones). El proceso de reconocimiento de patrones está conformada principalmente por tres etapas: adquisición de datos, extracción de características y clasificación (Figura 4.10)



Figura 4.10 Proceso de reconocimiento de patrones.

4.6.1 Adquisición de datos

La adquisición de datos es el proceso de las señales de muestreo que miden las condiciones físicas del mundo real y la conversión de las muestras resultantes en valores numéricos digitales que pueden ser manipulados por una computadora. Los sistemas de adquisición de datos convierten típicamente formas de onda analógicas en valores digitales para su procesamiento. Los componentes de los sistemas de adquisición de datos incluyen:

- Los sensores que convierten los parámetros físicos en señales eléctricas.

- Acondicionamiento de señal de circuitos para convertir señales de los sensores en una forma que se puede convertir en valores digitales.
- Convertidores de analógico a digital, que convierten las señales del sensor condicionados a valores digitales.

Las aplicaciones de adquisición de datos son controlados con el uso de software, utilizando diversos lenguajes de programación para el desarrollo del mismo, como lo son BASIC, C, Fortran, Java, Lisp, Pascal, etc.

4.6.2 Extracción de características

La extracción de características implica la simplificación de la cantidad de recursos necesarios para describir un gran conjunto de datos con precisión. Al realizar el análisis de datos complejos, uno de los principales problemas se deriva de la cantidad de variables implicadas. El análisis con un gran número de variables en general, requiere una gran cantidad de memoria y potencia de computación o un algoritmo de clasificación que sobreajuste los ejemplos de entrenamiento y generalice pobremente a las nuevas muestras.

4.6.3 Clasificación

La clasificación en ciertas ocasiones tiene el problema en identificar cuál es el conjunto de categorías que pertenece una nueva observación, en la base del conjunto de entrenamiento de las observaciones que contienen los datos, cuya pertenencia a una categoría se conoce. Un algoritmo que implementa la clasificación, especialmente en una aplicación concreta, es conocido como un clasificador. El término "clasificador" a veces también se refiere a la función matemática, implementada por un algoritmo de clasificación que mapea datos de entrada a una categoría. En el aprendizaje de máquina, las observaciones son a menudo conocidas como instancias, las variables explicativas se denominan características y las posibles categorías para ser predichas son clases.

CAPÍTULO V

Implementación de las redes en el reconocimiento de fallas

Para el reconocimiento de fallas, aplicando redes neuronales se realizó lo siguiente: Utilización del Software ProDiMES, el cual permitió la simulación de motores turbo ventilador, con esta información de flota de motores se hizo la extracción necesaria para la creación de modelos de turbinas de gas los cuales fueron necesarios para el diagnóstico de fallas. Una vez obtenidos los modelos requeridos, se implementó un sistema de clasificación de fallas adecuado para las redes neuronales, posteriormente se optimizaron dichas redes para la obtención de las mejores probabilidades. Con los resultados obtenidos, se comparó contra lo publicado en el artículo “*Regression-Based Modelling of a Fleet of Gas Turbine Engine for Performance Trending*” presentado en la Turbo Expo (Borguet, et al., 2015). En la Figura 5.1 se muestra un diagrama que describe en términos generales lo hecho en este trabajo

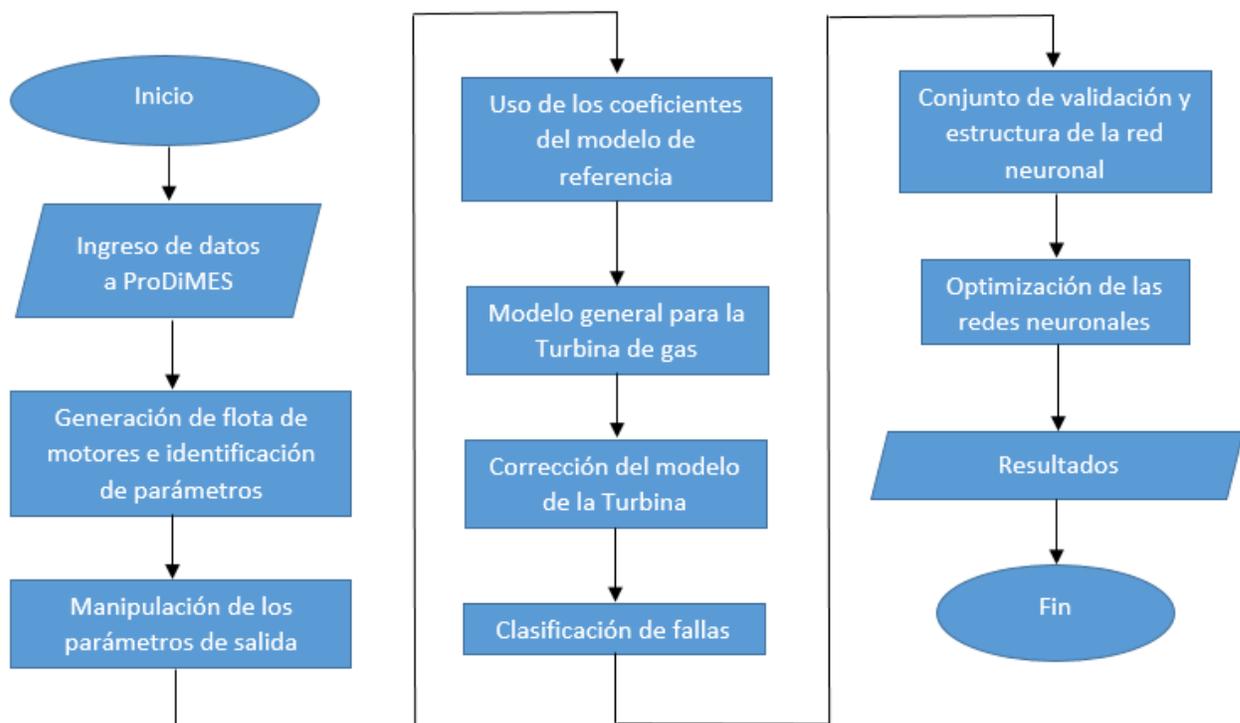


Figura 5.1 Estructura general del desarrollo del trabajo

En las siguientes secciones se da una descripción más detallada acerca del diagrama de la Figura 5.1

5.1 Ingreso de datos a ProDiMES

Como se vio en la sección 3.3, ProDiMES tiene la posibilidad de simular 18 fallas distintas de motores turbo ventilador y también simular motores sin influencia de fallas (estado nominal). El banco de datos de la flota de motores que ofrece el software sirvió para la creación de modelos de turbinas de gas necesarios para aplicar el método de diagnóstico propuesto.

ProDiMES ofrece un archivo llamado “EFS_Output.data” en la carpeta “Example Solution”, este archivo tiene las características siguientes: 190 motores, 50 vuelos por motor y 11 parámetros de salida, los parámetros de salida fueron descritos en la sección 3.6. Los 190 motores del archivo de salida corresponden a las 18 fallas posibles más el estado nominal, teniendo así 10 motores por cada falla.

Para poder aplicar correctamente el diagnóstico de fallas fue necesario de un conjunto de entrenamiento y un conjunto de validación, el archivo “EFS_Output.data” fue usado como conjunto de validación. Para crear el conjunto de entrenamiento se hizo uso del simulador ProDiMES, en donde se propusieron 10 motores por cada estado de falla y 50 vuelos por cada motor, se propuso lo anterior para tener el mismo número de motores y vuelos con falla respecto al conjunto de validación y así tener una mejor comparativa. Para la opción del inicio de vuelo de falla se escogió 11 debido a que en los primeros 10 vuelos no hay influencia de fallas y 11 es el primer vuelo en donde si hay dicha influencia. En la opción de tasa de evolución de falla se escogió “Rápido” esto porque así se pudo analizar un desarrollo gradual de la falla a partir del vuelo 11, como se simulaban 50 vuelos por motor solo se pudo definir la tasa de evolución a 34 de acuerdo a las especificaciones de ProDiMES, dentro del rango los vuelos tienen influencia de falla que van del vuelo 11 al 44. Con estos parámetros definidos en ProDiMES se pudo simular 180 motores con diferentes fallas.

En el artículo de (Cisneros, et al., 2015) se propusieron varios escenarios con distinto número de motores nominales para la creación del modelo de referencia con el menor error, dando como mejor resultado 100 motores, por lo que para este trabajo se tomó el mismo número para el estado nominal. Es así que la Interfaz de Usuario de ProDiMES, se simulaban 100 motores en estado nominal. Juntando todos los motores simulados se obtuvo un total de 280 motores para el conjunto de entrenamiento. Los datos que se ingresaron en el simulador ProDiMES se muestran en la Figura 5.2.

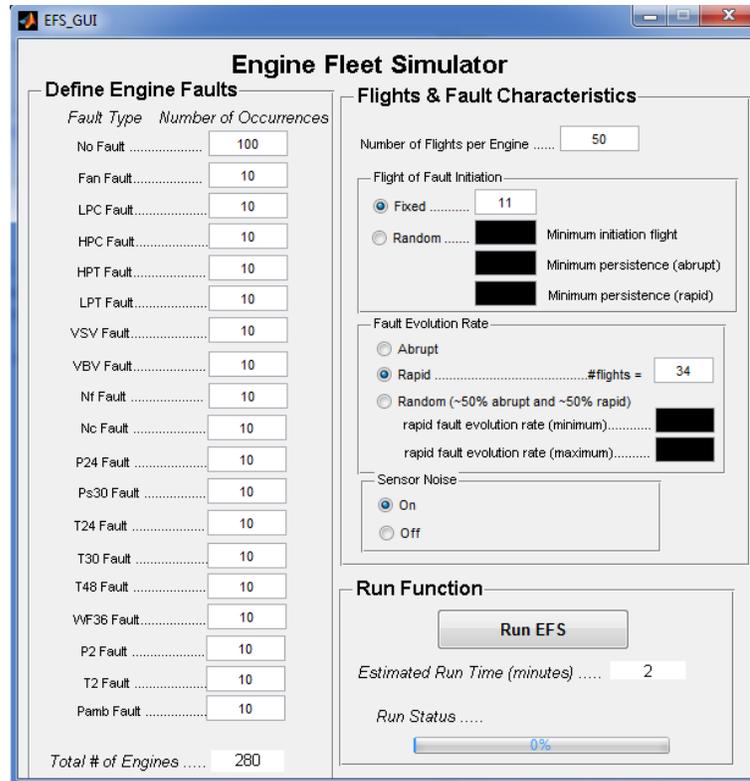


Figura 5.2 Ingreso de los datos a ProDiMES, (ejecutando Simulador ProDiMES 2016).

5.2 Generación de flota de motores e identificación de parámetros

Una vez que se ejecutó el simulador ProDiMES con los datos ingresados previamente, este arrojó un nuevo archivo de salida “EFS_Output.data” ubicado en la subcarpeta “EFS” de ProDiMES. Teniendo este nuevo archivo de salida, se generaron dos archivos adicionales, “efs_output_c” y “efs_output_to”, condición de crucero para el primero y condición de despegue del avión para el segundo, en este trabajo se limitó a trabajar solo con la primera opción.

El nuevo archivo generado “efs_output_c” tuvo una dimensión de 3, con tamaño (280 x 50 x 11), cada número corresponde a lo siguiente: (No. Motores totales simulados x No. vuelos por motor x Parámetros de salida). La lista de los parámetros de salida se muestra en la Tabla 5.1.

Tabla 5.1. Parámetros de salida que arroja ProDiMES

ID	Símbolo	Variable
1	Nf	Velocidad física del ventilador
2	Nc	Velocidad física del núcleo
3	P24	Presión total a la entrada en LPC
4	Ps30	Presión total a la salida en HPC
5	T24	Temperatura total a la salida en LPC
6	T30	Temperatura total a la salida en HPC
7	T48	Temperatura total a la entrada en HPC

8	Wf	Flujo de combustible
9	P2	Presión total a la entrada del ventilador
10	T2	Temperatura a la entrada del ventilador
11	Pamb	Presión ambiental

En la sección 2.3.1, se mencionó que es necesario identificar los parámetros correspondientes a variables monitoreadas y los de condiciones de operación, esto para hacer posible el desarrollo de los modelos de diagnóstico del conducto de flujo. De la Tabla 5.1 se tienen dichos parámetros mencionados. La Tabla 5.2 muestra las 7 variables monitoreadas las cuales son necesarias para el cálculo de las desviaciones (sección 2.4) y en la Tabla 5.3 están los 4 parámetros de condiciones de operación, responsables en determinar el régimen de la Turbina de Gas.

Tabla 5.2 Variables monitoreadas

ID	Símbolo	Variable
1	Nc	Velocidad física del núcleo
2	P24	Presión total a la entrada en LPC
3	Ps30	Presión total a la salida en HPC
4	T24	Temperatura total a la salida en LPC
5	T30	Temperatura total a la salida en HPC
6	T48	Temperatura total a la entrada en HPC
7	Wf	Flujo de combustible

Tabla 5.3 Condiciones de operación

ID	Símbolo	Variable
1	Nf	Velocidad física del ventilador
2	P2	Presión total a la entrada del ventilador
3	T2	Temperatura a la entrada del ventilador
4	Pamb	Presión ambiental

5.3 Manipulación de los parámetros de salida

El nuevo archivo generado por el simulador “efs_output_c” fue convertido en un nuevo arreglo YU_e de 2 dimensiones, esto para poder manipular con mayor facilidad los datos obtenidos y así tener un orden más sencillo y simplificado para las redes neuronales, el nuevo archivo quedó con un tamaño de 14,000 x 11, donde 14,000 fue el resultado de concatenar 280 motores con 50 vuelos.

En la sección 5.1 se identificaron dos grupos de parámetros (Tabla 5.2 y 5.3) necesarios para el análisis de conducto de flujo, para la creación del modelo de diagnóstico mediante polinomios (sección 2.3.2) fueron necesarios los parámetros de condiciones de operación y los coeficientes en estado sano del motor (Cisneros, et al., 2015), del arreglo YU_e se tomaron solo los coeficientes de operación para el desarrollo de los modelos de diagnóstico, donde se creó un nuevo arreglo U_e con tamaño 14,000 x 4, y otro el arreglo Y_e con tamaño 14,000 x 7, este último

contiene las variables monitoreadas. La Figura 5.3 describe el proceso de separación de los datos de crucero procedentes de ProDiMES.

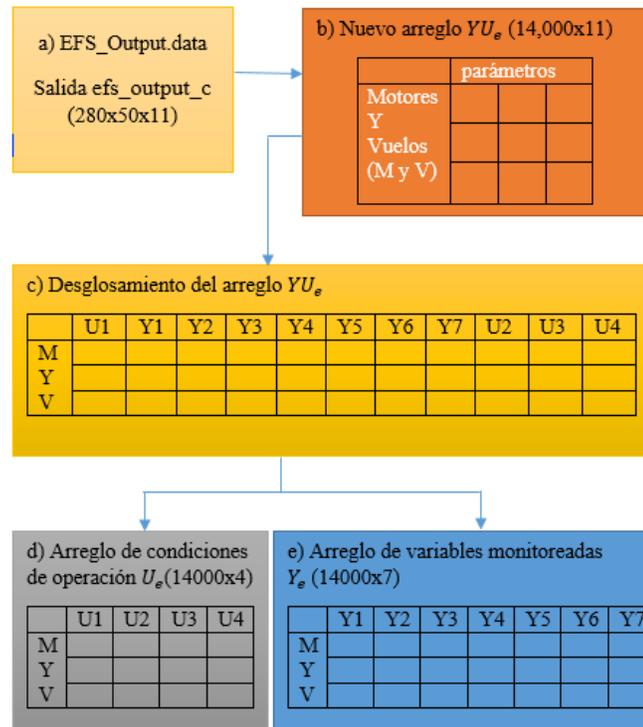


Figura 5.3 Diagrama de separación para los datos simulados por ProDiMES.

5.4 Uso de los coeficientes del modelo de referencia

Para avanzar en el proceso de clasificación, fue necesario la obtención de coeficientes para un modelo de referencia que describa el estado sano del motor, es decir que las fallas no están involucradas dentro de los datos simulados. Este trabajo se hizo en (Cisneros, et al., 2015) donde se muestra la obtención de dichos coeficientes para tener un modelo en condiciones nominales, los coeficientes se ven aplicados en el trabajo (Felipe, et al., 2015), donde se muestran probabilidades de diagnóstico con otro variante de clasificación de fallas.

Para la creación del modelo de referencia en estado sano en (Cisneros, et al., 2015) se necesitó ejecutar una historia de parámetros, mediante el simulador de la flota de motores ProDiMES, los datos ingresados mediante la Interfaz de usuario fueron las siguientes: 100 motores y 5,000 vuelos con ruido deshabilitado. Con esta información se obtuvo un conjunto de coeficientes necesarios para hacer una función de aproximación, los cuales fueron requeridos para el cálculo de las desviaciones.

Cuando se obtuvieron los coeficientes necesarios para la aproximación de un motor sano, se pudo desarrollar la siguiente ecuación:

$$Y_0 = c_0 + c_1 Nf + c_2 P^2 + c_3 T^2 + c_4 P_{amb} +$$

$$\begin{aligned}
& + c_5 N f P 2 + c_6 N f T 2 + c_7 N f P_{amb} + \\
& + c_8 P 2 T 2 + c_9 P 2 P_{amb} + c_{10} T 2 P_{amb} + \\
& + c_{11} N f^2 + c_{12} P 2^2 + c_{13} T 2^2 + c_{14} P_{amb}^2
\end{aligned} \tag{5.1}$$

La Ecuación 5.1 es la representación polinomial de un motor simulado, ya visto en la sección 2.3.2. La expresión 5.1 contiene los parámetros de la Tabla 5.3, que pertenecen a las condiciones de operación del motor. En total se obtuvieron 15 coeficientes (c_0 - c_{14}) debido a la combinación lineal de las condiciones de operación.

Para la obtención de los coeficientes se propusieron varios variantes para determinar la función que tuviera menor error en la aproximación de un motor sano, la Figura 5.4 muestra el comportamiento de motores con los mejores coeficientes encontrados, en el *eje x* se observan los vuelos por motor y en el *eje y* las desviaciones para cada variable monitoreada (DPs30 – desviación de la presión total a la salida del Compresor, DT24 – desviación en la temperatura total a la salida Compresor), en los vuelos 90, 180, 270, 360 y 450 están los cambios por cada motor, en el segundo gráfico, del vuelo 180 a 270, se puede observar de forma más clara como el motor tiene una desviación diferente a los del resto pero con diferente comportamiento respecto a las desviaciones de cada variable monitoreada.

Los cambios graduales de cada motor representados en la gráfica se deben a que cualquier motor tiene tendencia al envejecimiento, así como la vida útil es individual por cada motor.

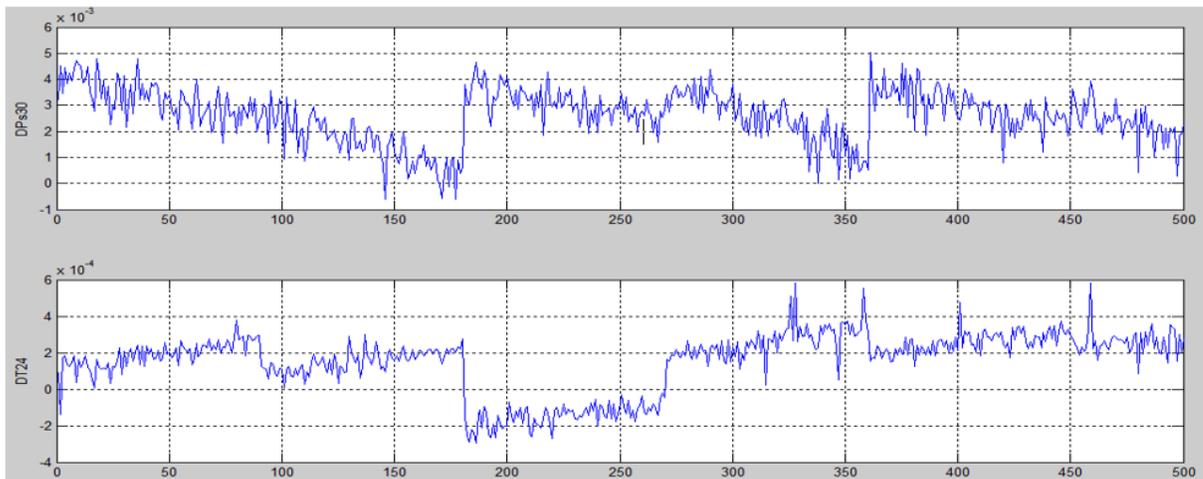


Figura 5.4 Comportamiento de motores en estado sano ampliado a 500 vuelos, (Cisneros, et al., 2015).

5.5 Modelo general para la Turbina de gas

El cálculo de las desviaciones con menor error sirve para la representación de un modelo que describa el comportamiento de una turbina de gas, información necesaria para el diagnóstico y aplicación en la clasificación de fallas. La desviación representa la diferencia entre el valor medido Y_i^* y el valor de referencia Y_0 . Es por esto que se tiene la siguiente Ecuación 5.2.

$$\delta Y_i = \frac{Y_i^* - Y_0}{Y_0}, \quad i = 1, \dots, m \quad (5.2)$$

Con los coeficientes calculados en estado sano de la sección 5.3, se creó un nuevo arreglo XX , este tiene un tamaño de 15×7 , en donde el primer valor corresponde a los coeficientes calculados del modelo de referencia en estado sano y el segundo término a las 7 variables monitoreadas Y .

Con las condiciones de operación se generó un nuevo arreglo A_e , que es el resultado de la combinación lineal del arreglo U_e las cuales fueron simulados anteriormente. El arreglo A_e tiene un tamaño de $14,000 \times 15$, usando los coeficientes del modelo en estado sano se puede calcular la Ecuación 5.3.

$$Y_{egral} = A_e * XX \quad (5.3)$$

Con la Ecuación 5.3 se tiene un modelo general que describe el comportamiento de cada turbina de gas. El arreglo Y_{egral} tiene un tamaño de $14,000 \times 7$, esto quiere decir que tenemos valores de referencia para 14,000 vuelos, involucrados 280 motores.

5.6 Corrección del modelo de la Turbina

Para el modelo general obtenido hubo una diferencia en las características de cada motor, debido a que el generador de casos asigna aleatoriamente una degradación inicial diferente para cada uno. Se corrigió esto mediante el cálculo de un modelo individual por cada motor. Esto se hizo mediante el procedimiento siguiente: 1) Cálculo de las desviaciones entre el modelo general y los valores monitoreados, solo con los primeros 10 vuelos por cada motor, debido a que no hay influencia de fallas. 2) Obtención de la diferencia promedio de 10 diferencias de cada variable Y , 3) Construcción del modelo individual corregido de cada motor

- 1) Con el concepto de desviación y la Ecuación 5.2 se obtuvo la siguiente expresión:

$$\delta Y_{egral} = \frac{Y_e - Y_{egral}}{Y_{egral}} \quad (5.4)$$

donde Y_e es el arreglo hecho de los datos simulados previamente Figura 5.3 e), y Y_{egral} es el modelo general calculado por la Ecuación 5.3

- 2) Con el cálculo de la desviación de la Ecuación 5.4 se tomaron los 10 primeros vuelos de cada motor para tener una corrección del modelo, donde se creó un nuevo arreglo $\delta Y_{egral10}$, se calculó el promedio de los 10 vuelos por motor dando el arreglo δY_{egprom} y así una desviación promedio necesaria para la creación del modelo corregido, Figura 5.5 c).

- 3) Gracias a la obtención de δY_{egprom} , se puede crear una ecuación del modelo de referencia individual corregido. Esta se formó de la siguiente manera:

$$Y_{eic} = Y_{egral} * (1 + \delta Y_{egprom}) \quad (5.5)$$

donde Y_{eic} representa el modelo individual para cada variable Y de cada motor.

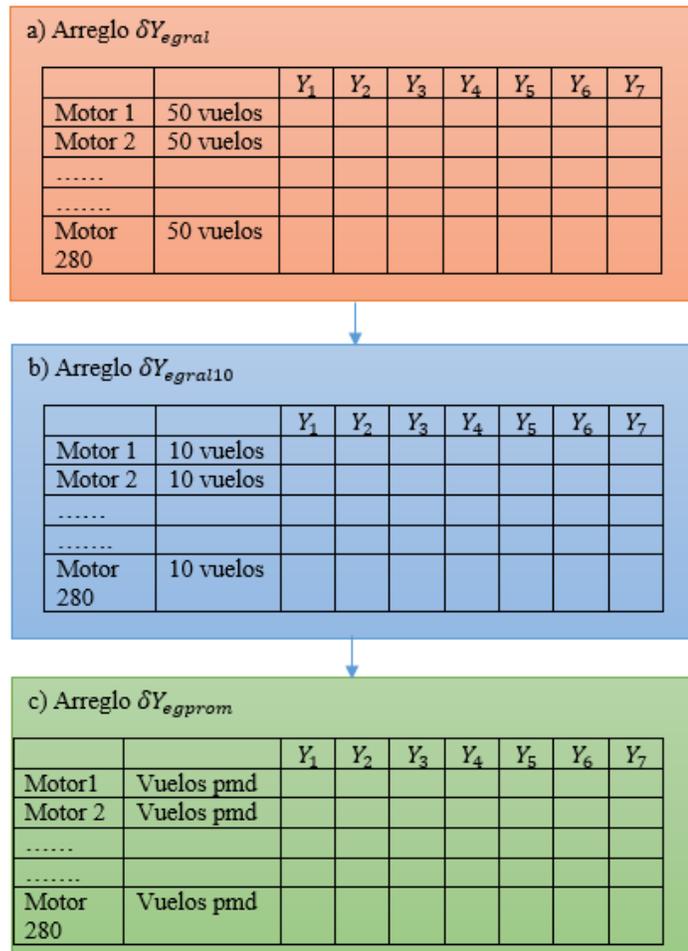


Figura 5.5 Proceso para la corrección del modelo.

5.7 Clasificación de fallas

Para la formación de la clasificación se crearon clases para cada tipo de falla, esto se hizo mediante cálculo de las desviaciones y con la corrección del modelo de la Turbina (Ecuación 5.5). Con los datos del arreglo Y_e de variables monitoreadas y el arreglo Y_{eic} se pudo formar la Ecuación 5.6:

$$\delta Y_{eic} = \frac{Y_e - Y_{eic}}{Y_{eic}} \quad (5.6)$$

En ProDiMES se tienen 18 tipos de fallas más el estado nominal, dando la posibilidad de crear 19 grupos diferentes, es así que se crearon 19 clases. La clase 1 corresponde a los motores donde no hay falla, la clase 2 corresponde a la falla del ventilador (ID de falla 1, Tabla 5.4), y así sucesivamente hasta llegar a la clase 19 que corresponde a la falla a causa de la presión ambiental (ID de falla 18, Tabla 5.4)

Tabla 5.4 Total de clases a formar

ID de falla	Descripción de falla	ID de clase
0	No hay falla	1
1	Falla en Fan	2
2	Falla en LPC	3
3	Falla en HPC	4
4	Falla en HPT	5
5	Falla en LPT	6
6	Falla en VSV	7
7	Falla en VBV	8
8	Falla en sensor de Nf	9
9	Falla en sensor de Nc	10
10	Falla en sensor de P24	11
11	Falla en sensor de Ps30	12
12	Falla en sensor de T24	13
13	Falla en sensor de T30	14
14	Falla en sensor de T48	15
15	Falla en sensor de Wf	16
16	Falla en sensor de P2	17
17	Falla en sensor de T2	18
18	Falla en sensor de Pamb	19

Para la formación de las clases con fallas, se obtuvieron 10 motores por cada uno, la tasa de evolución de falla fue de 40, por lo que se falla se presentó del vuelo 11 al 50, es así que se tomaron los 40 vuelos de ese rango por motor, dando así 400 vuelos por cada falla, teniendo 18 clases con 400 vuelos para su formación.

En la simulación del estado sin falla se tuvieron 100 motores con 50 vuelos, dando así 5000 vuelos sin influencia de falla, se propuso tomar 40 vuelos como se hizo en las que si tenían influencia de falla, la formación de la clase 1 quedó en 4000 vuelos ya que se tenían 100 motores simulados.

La Figura 5.6 muestra 7 clases diferentes correspondientes a las primeras fallas, que van de la falla 1 a la 7. La Figura 5.7 muestra 5 clases que corresponden a fallas en sensores que van de la falla 8 a la 13.

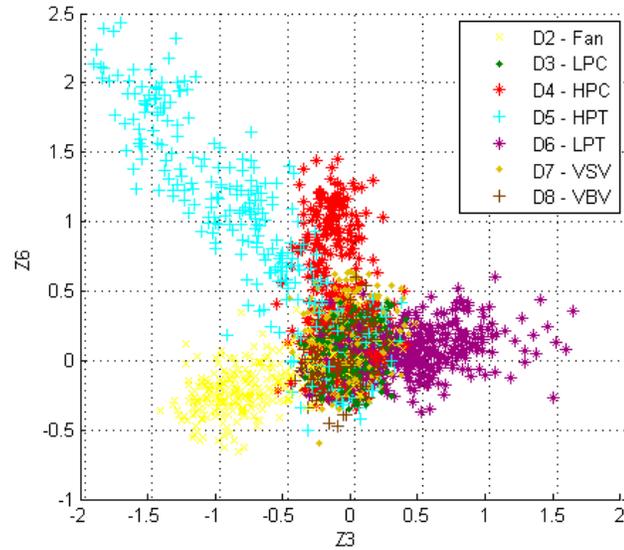


Figura 5.6 Muestra de las primeras clases con falla (clase 2-8).

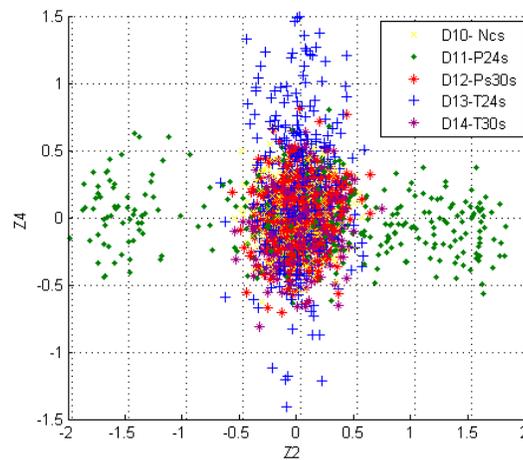


Figura 5.7 Muestra de otras clases (clase 9-14).

5.8 Conjunto de validación y estructura de la red neuronal

Cuando se usa alguna red neuronal para el reconocimiento de fallas, son necesarios dos conjuntos de datos, el primero corresponde al de entrenamiento (ya creado previamente), y el segundo corresponde al de validación.

De la carpeta *ExampleSolution* de ProDiMES, contiene un muestra *EFS_Output.data*, la cual sirvió para hacer un conjunto de validación, de igual manera como en la fase de entrenamiento, se encontraron dos archivos adicionales, *efs_output_c* y *efs_output_to*, para ser comparado correctamente con el de entrenamiento se tomó el que pertenece a la condición de cruce con

un tamaño de 190 x 50 x 11. Teniendo este tamaño se sabe que son 190 motores simulados con 50 vuelos por cada motor, el archivo contiene las 18 fallas más el estado sin falla.

Así como se pasó de un arreglo de 3 dimensiones a 2 dimensiones en la etapa de entrenamiento, aquí también se hizo lo mismo, dando un nuevo arreglo YU_v con tamaño de 9,500 x 11. De igual manera se dividieron en dos partes, el arreglo U_v para las condiciones de operación, y el arreglo Y_v para las variables monitoreadas.

El uso del modelo general también fue necesario aquí, quedando la siguiente expresión:

$$Y_{vgral} = A_v * XX \quad (5.7)$$

Se aplicó la corrección del modelo y los pasos correspondientes, dando las siguientes expresiones:

$$\delta Y_{vgral} = \frac{Y_v - Y_{vgral}}{Y_{vgral}} \quad (5.8)$$

$$Y_{vic} = Y_{vgral} * (1 + \delta Y_{vgral}) \quad (5.9)$$

donde Y_{vic} representa el modelo individual corregido para cada motor de la muestra de prueba.

Y por último se aplicó la desviación para la formación del conjunto de validación:

$$\delta Y_{vic} = \frac{Y_v - Y_{vic}}{Y_{vic}} \quad (5.10)$$

5.8.1 Estructura de la red

Antes de utilizar cualquier tipo de red neuronal, se deben conocer las entradas y las salidas en cada capa para alimentar a la red (sección 4.5), donde las capas de salida contienen 1 y 0 para darle reconocimiento a la pertenencia de una clase. Cuando la salida es 1 significa que el dato analizado corresponde a una clase específica, cuando es 0 significa que no corresponde.

El número de nodos en la capa de entrada para la red neuronal fue de 7, dicho número fue el que representó a las variables monitoreadas ya que es el número máximo de estas variables para los arreglos Y_e y Y_v . Como se formaron 19 clases tanto para el conjunto de entrenamiento y validación, 19 nodos en la capa de salida fueron con los que trabajó la red neuronal.

Los nodos de la capa oculta no son fijas ya que la probabilidad de diagnóstico varía según el valor numérico que se proponga, es así que se tuvieron que programar varios algoritmos para las redes con el fin de buscar la mejor probabilidad en el diagnóstico de fallas.

5.9 Optimización de las redes neuronales

5.9.1 Perceptrón multicapa

Para esta red se ajustó el número óptimo para las épocas y el de neuronas ocultas (sección 4.5.1). El número propuesto para las épocas fueron de 100 hasta 5,000, con un salto de 100 en la iteración. Para la capa oculta se propusieron de 15 hasta 81 neuronas, con un salto de 3 neuronas de iteración.

Los datos de salida calculados para la probabilidad de diagnóstico fueron de 50 archivos por cada neurona oculta, dando un total de 1,150 archivos para la red Perceptrón multicapa. En la Figura 5.8 se muestra la influencia que tienen las épocas para algunas neuronas.

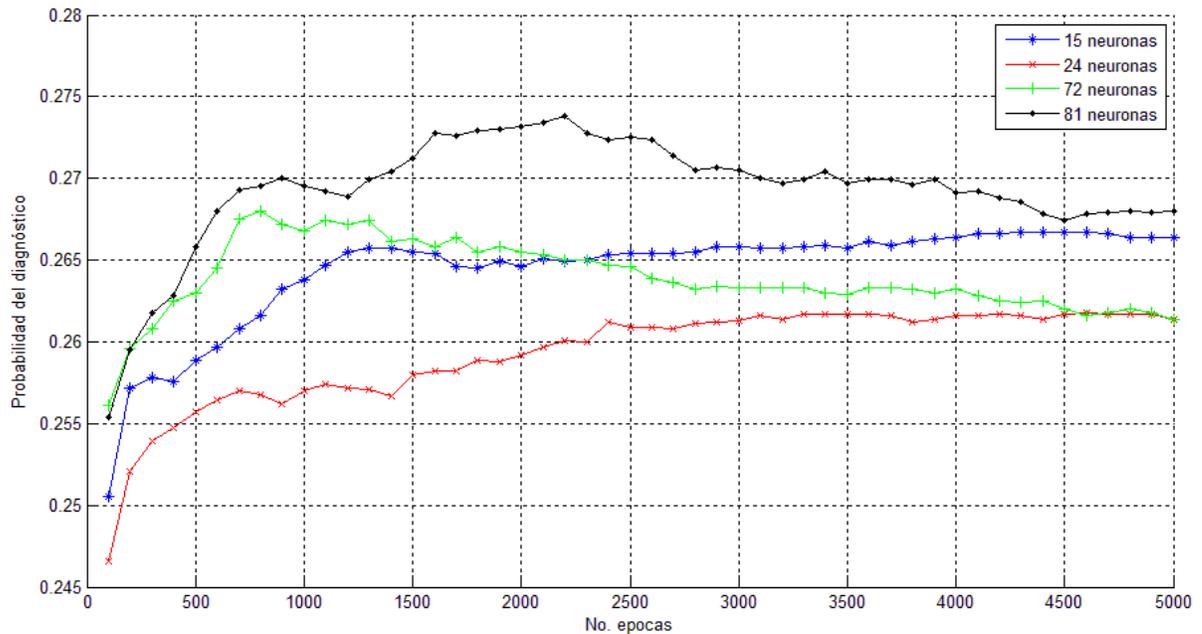


Figura 5.8 Influencia en el No. de épocas para cada capa oculta (MLP).

Se puede notar de la Figura 5.8, que no necesariamente va aumentando la probabilidad de diagnóstico conforme el aumento de las neuronas, ya que 15 neuronas tienen un mejor resultado comparada con el de 24, se nota que después de 3000, las dos capas ocultas tienen una tendencia similar pero con diferentes probabilidades de diagnóstico. En el caso de las dos últimas capas ocultas calculadas el que dio mejor resultado es el de 81 neuronas, dejando al de 72 con gran desventaja, notándose que después de la época 2,250 por debajo del de 15 neuronas.

De los 1,150 archivos calculados, se monitoreo el de mejor probabilidad, dando como resultado óptimo 75 neuronas ocultas acompañado de 900 épocas, con una probabilidad de diagnóstico para validación de 0.2866. En la Figura 5.9 se muestra la tendencia para 75 neuronas ocultas.

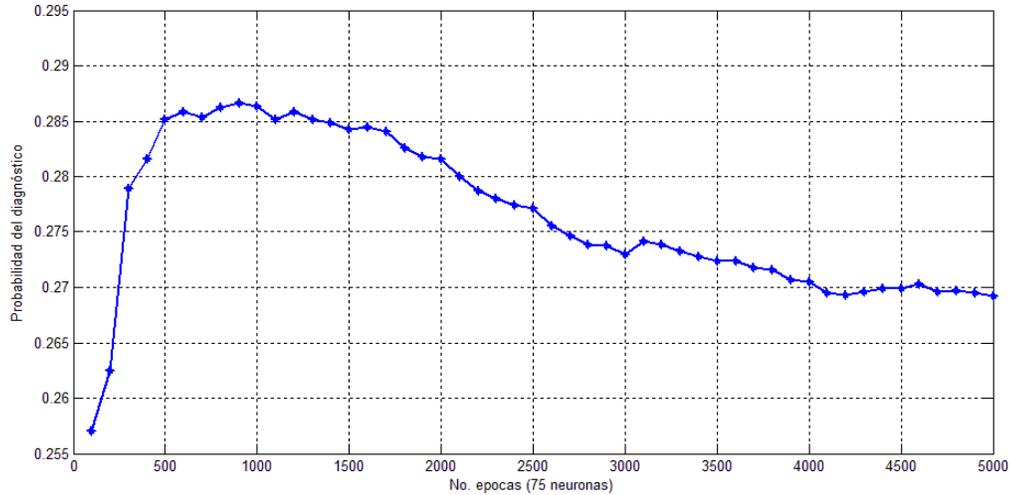


Figura 5.9 Mejor probabilidad de diagnóstico para 75 neuronas (MLP).

5.9.2 Red de Base Radial

El “spread” o propagación, representa la desviación estándar que se utiliza para identificar la amplitud de la campana de Gauss, ya sea en base radial o en red probabilística. Para la optimización de este parámetro, se calculó desde 0.1 hasta 3, con un salto de 0.1 de iteración. Para las neuronas ocultas se calcularon desde 50 hasta 119, con un salto de 3 neuronas de iteración.

Aquí el número total de archivos de salida para el diagnóstico fue de 720, donde 30 archivos correspondieron a cada capa oculta. En la Figura 5.10 se muestran solo 4 gráficas de diferentes capas ocultas.

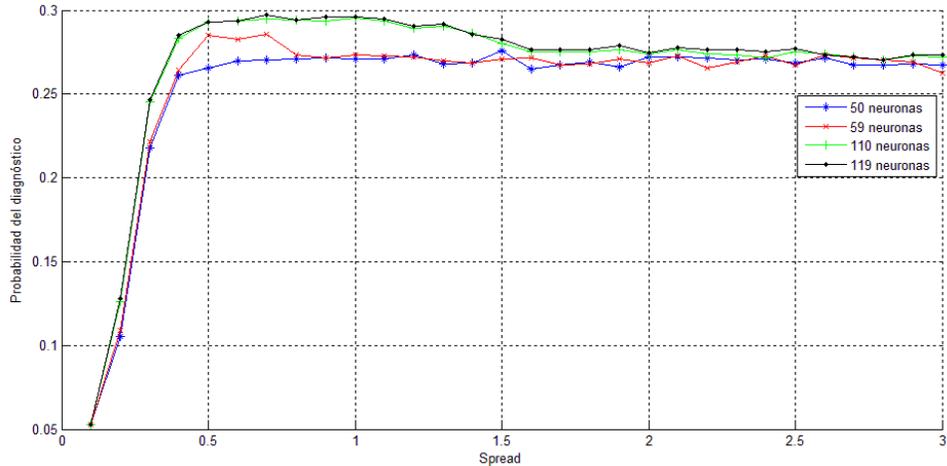


Figura 5.10 Influencia en el No. de épocas para cada capa oculta (RBN).

En la Figura 5.10 se puede notar una mejoría en la probabilidad de diagnóstico con el aumento de neuronas ocultas, pero solo fue válido dentro del rango de 0.4 hasta 2 de spread, después de ese límite la diferencia entre neuronas no fue tan relevante ya que tienden a reflejar los mismos resultados.

De los 720 archivos, el que dio mejor resultado de probabilidad de diagnóstico fue el de 116 neuronas ocultas y 0.7 de spread, dando una probabilidad de 0.2978 para el conjunto de validación. En la Figura 5.11 se muestra la tendencia para 116 neuronas.

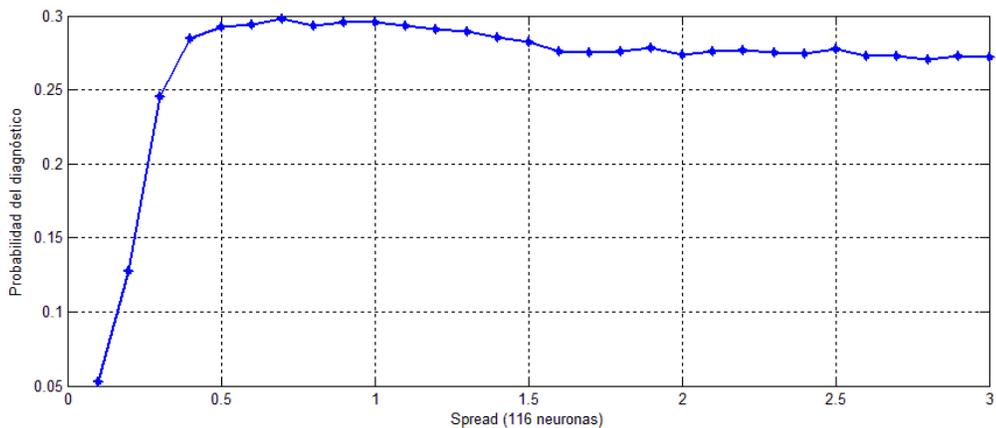


Figura 5.11 Mejor probabilidad de diagnóstico con 116 neuronas (RBN).

5.9.3 Red Probabilística

En esta red, el parámetro que tuvo mayor influencia es el “spread” o propagación. Para ajustarlo al número óptimo se hizo el cálculo de 0.1 hasta 5, con un salto de 0.1 de iteración. Dando solamente 50 archivos de salida para el diagnóstico. Su tendencia se muestra en la Figura 5.12.

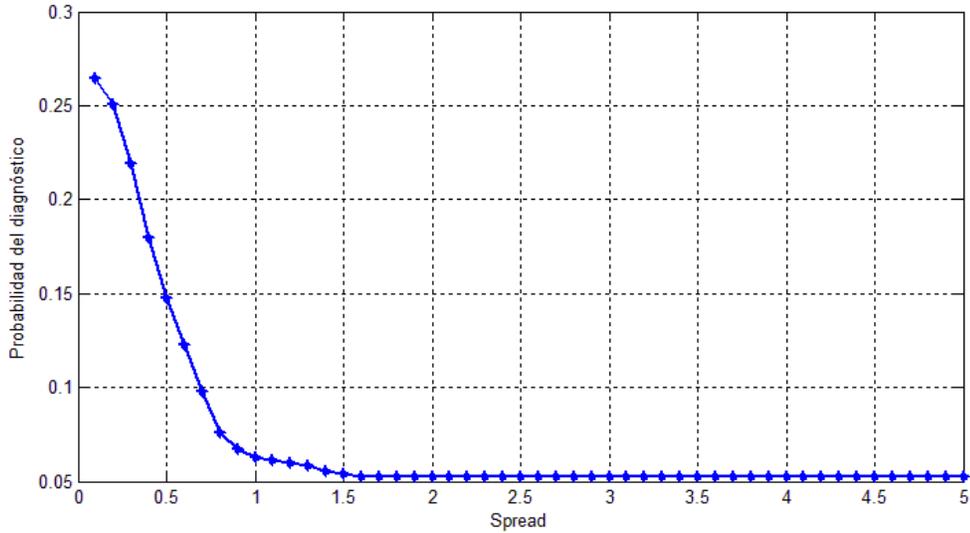


Figura 5.12 Tendencia del spread respecto a la probabilidad (PNN).

De la Figura 5.12 se puede notar que los que dieron mejores resultados en probabilidad se encuentran al principio, ya que después de 1.5 en spread, empezaron a mantenerse constante y con una muy baja probabilidad.

Para asegurarnos de que al principio es donde se obtienen los mejores resultados, se hizo un segundo cálculo, con 50 archivos de salida, donde fueron de 0.01 hasta 0.5 de spread, con un salto de 0.01 de iteración. El resultado se muestra en la Figura 5.13

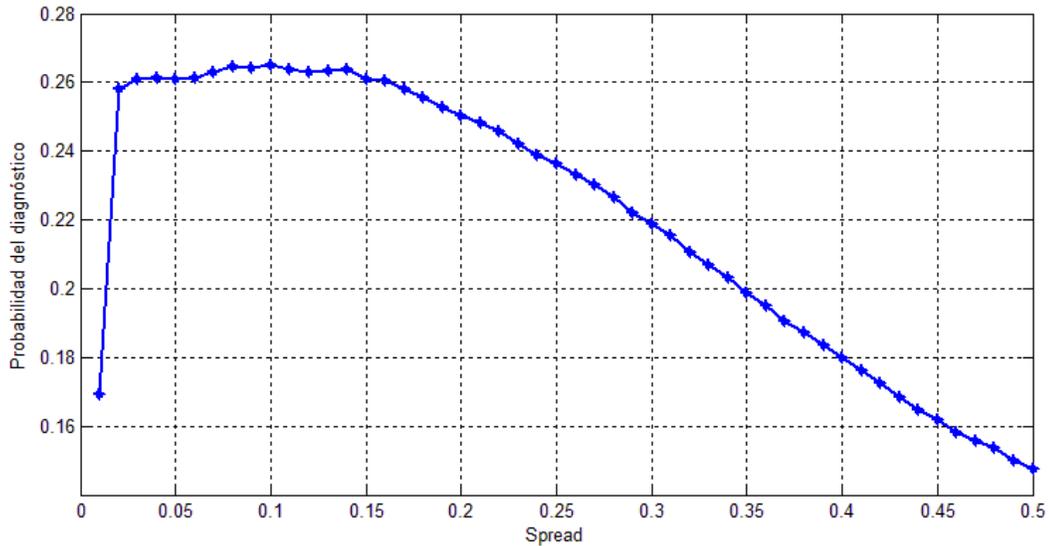


Figura 5.13 Tendencia de spread con las mejores probabilidades (PNN).

Como se observa en la Figura 5.13 los que obtuvieron las mejores probabilidades fueron en el rango de 0.05 hasta 0.151 de spread, quedando 0.1 spread con el mejor resultado de probabilidad y con una probabilidad de diagnóstico de 0.2649.

De las secciones pasadas de este capítulo se describieron las entradas que se le implementaron a ProDiMES así como su manipulación, el desarrollo de los modelos para el diagnóstico, la formación de clases y la aplicación a las redes neuronales, todo esto fue desarrollado mediante el ambiente de programación de Matlab, debido a que ProDiMES fue desarrollado dentro de este entorno, además de ser una potente herramienta para la manipulación de matrices y de gran uso en el área de las redes neuronales.

El desglosamiento del código se encuentra en el apartado *Anexo Código de programación*

5.10 Resultados

5.10.1 Probabilidades de diagnóstico

El mejor resultado para la red MLP fue: 75 neuronas ocultas y 900 épocas con un probabilidad de diagnóstico igual a 28.66%. En RBN fue para: 116 neuronas ocultas y 0.7 de spread con un diagnóstico igual a 29.78%. Por último PNN con: 0.1 de spread con un diagnóstico igual a 26.49%.

Las probabilidades de cada clase para validación y entrenamiento se muestran en la Tabla 5.5 y 5.6.

Tabla 5.5 Probabilidad de diagnóstico para las 3 redes en entrenamiento

No. Clase	Probabilidad para MLP	Probabilidad para RBN	Probabilidad para PNN
Clase 1	0.9441	0.9571	1.0000
Clase 2	0.4500	0.4176	0.9000
Clase 3	0.0559	0.0176	0.8765
Clase 4	0.5706	0.5206	0.9412
Clase 5	0.8294	0.8000	0.9853
Clase 6	0.5794	0.5324	0.9559
Clase 7	0.2794	0.5853	0.9559
Clase 8	0.0471	0.0382	0.8353
Clase 9	0.3882	0.3647	0.9382
Clase10	0.0529	0.0029	0.8912
Clase 11	0.4676	0.4971	0.9412
Clase 12	0.1765	0.0941	0.8882
Clase 13	0.3353	0.2706	0.9000
Clase 14	0.2500	0.3382	0.9147
Clase 15	0.3294	0.4294	0.9147
Clase 16	0.3294	0.3000	0.9176
Clase 17	0.0676	0.0147	0.8735
Clase 18	0.3794	0.3176	0.9235
Clase 19	0.0000	0.0000	0.8324

\bar{P}	0.3438	0.3420	0.9150
-----------	---------------	---------------	---------------

Tabla 5.6 Probabilidad de diagnóstico para las 3 redes en validación

No. Clase	Probabilidad para MLP	Probabilidad para RBN	Probabilidad para PNN
Clase 1	0.9825	0.9875	0.9525
Clase 2	0.5600	0.5475	0.4325
Clase 3	0.0475	0.0150	0.0375
Clase 4	0.4950	0.4700	0.4275
Clase 5	0.5250	0.5250	0.5125
Clase 6	0.5325	0.5300	0.4775
Clase 7	0.2775	0.5175	0.5075
Clase 8	0.0325	0.0300	0.0175
Clase 9	0.1450	0.1625	0.1625
Clase10	0.0000	0.0000	0.0050
Clase 11	0.2750	0.4300	0.4100
Clase 12	0.1750	0.0700	0.0275
Clase 13	0.1450	0.1350	0.1100
Clase 14	0.3050	0.2850	0.2000
Clase 15	0.3575	0.3775	0.3425
Clase 16	0.1425	0.1725	0.1175
Clase 17	0.0100	0.0025	0.0200
Clase 18	0.4375	0.4000	0.2725
Clase 19	0.0000	0.0000	0.0000
\bar{P}	0.2866	0.2978	0.2649

Se ve claramente que las clases de entrenamiento tienen un porcentaje mayor respecto al conjunto de validación lo cual es muy normal ya que las redes neuronales se alimentaron de lo que se simuló con anterioridad, el conjunto de validación que se extrajeron de los archivos del software ProDiMES estuvo muy cercano con las probabilidades de diagnóstico. Se puede ver que el diagnóstico con la red neuronal probabilística tuvo probabilidades diferentes de entrenamiento y validación, esto es porque hubo un sobre entrenamiento en la red, reflejando una probabilidad de 91.5% en entrenamiento, pero el de validación tuvo un comportamiento similar con las otras redes.

5.10.2 Análisis de detección

Como se vio en la sección 2.5.3 la decisión de detección sirvió para ver el número de predicciones correctas que se tuvieron para cada proceso de clasificación. Con el conjunto que se usó para validar las redes neuronales, se obtuvo una matriz de referencia para ProDiMES, mostrado en la Tabla 5.7.

Tabla 5.7 Matriz de decisión de ProDiMES

	<i>Do</i>	<i>Df</i>
<i>do</i>	0.999	0.629

df | 1/1059 0.371

La Tabla anterior sirvió como referencia con este trabajo junto con el artículo “Regression-Based Modelling of a Fleet of Gas Turbine Engine for Performance Trending” (Borguet, et al., 2015). Dicho artículo trabajó con el simulador ProDiMES para hacer uso de la creación de los modelos pero con otra metodología para después hacer el análisis de detección de fallas, allí se propusieron varias configuraciones de detección, en la Tabla 5.8 se muestra el mejor resultado del artículo.

Tabla 5.8 Matriz de decisión del artículo de referencia

	Do	Df
do	0.999	0.729
df	1/1051	0.271

Como se puede observar sus resultados son muy similares a los que arroja ProDiMES, se tuvo una TPR menor, lo cual significa que la detección de fallas correctas no fueron las esperadas pero con el TNR se apegaron muy bien a la detección.

A continuación se muestran las tablas de los resultados de probabilidad de diagnóstico creados en este trabajo.

Tabla 5.9 Matriz de decisión MLP

	Do	Df
do	0.983	0.692
df	0.018	0.308

Tabla 5.10 Matriz de decisión RBN

	Do	Df
do	0.988	0.711
df	0.013	0.289

Tabla 5.11 Matriz de decisión PNN

	Do	Df
do	0.953	0.662
df	0.048	0.338

Se puede observar que las matrices creadas a partir de los resultados de las probabilidades de diagnóstico están parecidas a los de ProDiMES, se obtuvieron números menores en el TNR pero claramente hubo mejoría para TPR comparado con los del artículo de referencia ya que su mejor resultado fue de 27.1% y los resultados obtenidos de esta investigación fueron desde 28.9% hasta 33.8%. A continuación en la Figura 5.14 se reflejan dichas aproximaciones.

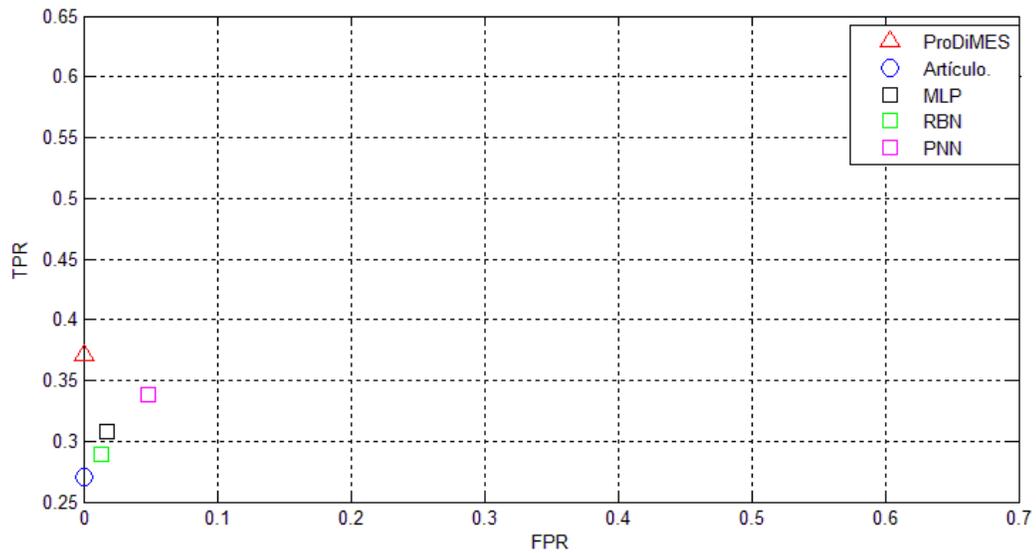


Figura 5.14 Aproximación de los resultados para decisión de detección.

Conclusiones

Hoy en día hay libros para el desarrollo detallado de los modelos que describen el comportamiento general de la turbina de gas, es por ello que existe una serie de paquetes de software comerciales. En este trabajo se usó el software ProDiMES desarrollado por la NASA para la simulación de una flota de motores turbo-ventilador, además de ser un software libre y disponible al público interesado.

Los fabricantes de motores rara vez ponen a disposición la información necesaria para saber el comportamiento del motor, con la ayuda de un simulador confiable, se pudo extraer los datos necesarios para el análisis de fallas. Los modelos generales de los motores hechos por los fabricantes están sintonizados para que coincida con el nivel de rendimiento inicial de un motor dado y se utilizan a continuación para evaluar el estado de salud de la turbina de gas durante su vida operativa, ProDiMES pudo ofrecer los datos necesarios para la simulación de condiciones sanas y defectuosas de una flota de motores, y así crear modelos necesarios para el diagnóstico de fallas.

Con la etapa de pruebas de modelos de referencia, se permitió encontrar la mejor aproximación del rendimiento del motor sano, esta etapa fue una parte importante del proceso de diagnóstico, ya que afectó directamente a la precisión de probabilidad. El modelo de referencia en estado sano que se usó, tuvo bajos niveles de errores, y gracias a ello se pudo reflejar adecuadamente las condiciones nominales del motor.

En este trabajo se pudo demostrar la aplicación que tienen las redes neuronales para el reconocimiento de patrones, que en este caso son las fallas presentes que se simulan durante la vida útil del motor y así aplicar el diagnóstico para todas las clases de fallas creadas en la clasificación. Se utilizaron tres redes neuronales como las técnicas de reconocimiento de patrones: Perceptrón Multicapa (MLP), Red Base Radial (RBN) y Red Neuronal Probabilística (PNN).

En los resultados de la sección (5.7.1) se pueden observar los datos de probabilidad de diagnóstico tanto en entrenamiento y validación que no están muy alejadas entre sí, lo cual dice que la red neuronal realizó correctamente su tarea. Los mejores resultados en probabilidad de diagnóstico para el conjunto de validación fue RBN con 29.78%, en segundo lugar fue MLP con 28.66% y el último PNN con 26.49%. El que se llevó más tiempo de ejecución fue RBN, después MLP y por último PNN.

La optimización de las redes sirvió para darse cuenta que las tendencias de probabilidad en el diagnóstico varían en función de los parámetros de cada red, donde se vio que para MLP, no necesariamente aumentando las neuronas ocultas se obtendría un mejor resultado. Caso contrario de RBN, en donde si hubo mejoría en sus resultados conforme el aumento de sus neuronas ocultas, pero sólo por tramos se debe de considerar, ya que después de cierta brecha, el número de neuronas ocultas no se tuvo gran diferencia para lo esperado en probabilidad. Es

interesante ver como para PNN, la propagación o “spread” tiene mayor fuerza con valor de 0.1, dejando a las demás como datos poco eficientes.

En el análisis de monitoreo tenemos que se obtuvieron datos parecidos a los del artículo, con números menores en TNR que van desde 95.3 a 98.8 lo cual no se aleja al máximo de referencia de 99.9. Se obtuvieron números positivos en cuanto a TPR superando la detección de fallas correctas lo que dan como bueno nuestro trabajo. Los resultados obtenidos son competitivos para el uso de las redes neuronales como herramienta de diagnóstico, donde se pueden proponer otras configuraciones para mejorar los índices de detección.

Bibliografía

Borguet, S., Leonard, O. & Dewallef, P., 2015. Regression-Based Modelling of a Fleet of Gas Turbine Engines for performance trending. *Proceedings of ASME Turbo Expo 2015*.

Boyce, M. P., 2012. *Gas Turbine Engineering Handbook*. Cuarta ed.

Broomhead, D. S. & Lowe, D., 1988. Radial Basis Functions, Multi-Variable Functional Interpolation and Adaptive Networks. *Complex Systems*.

Cisneros, F. d. J., Felipe, S. & Loboda, I., 2015. Elaboración del modelo de Referencia de una turbina de gas mediante el software ProDiMES. *Congreso Nacional de Ingeniería Electromecánica y de Sistemas*.

Felipe, S., Cisneros, F. d. J. & Loboda, I., 2015. Reconocimiento de las Fallas de una Turbina de Gas basado en el software universal ProDiMES. *Congreso Nacional de Ingeniería Electromecánica y de Sistemas*.

Giampaolo, A., 2006. *Gas Turbine Handbook. Principles and Practices*. Tercera ed.

Hilera, J. R. G. & Martinez, V. J. H., 2000. *Redes Neuronales Artificiales: fundamentos, modelos y aplicaciones*: Alfaomega.

Hünecke, K., 2003. *Jet Engines - Fundamental of Theory, Design and Operations*.

Isasi, P. & Galván, I. M., 2004. *Redes Neuronales Artificiales. Un enfoque práctico*.

Kamboukos, P. & Mathioudakis, K., 2005. Comparison of Linear and Nonlinear Gas Turbine Performance Diagnostics. *Journal of Engineering for Gas Turbines and Power*.

Konrad, H. & Isermann, R., 1996. Diagnosis of Different Faults in Milling Using Drive Signals and Process Models. *Proceedings of the 13th World Congress International Federation of Automatic Control*.

Kriesel, D., 2005. *A brief introduction to Neural Networks*.

Loboda, I., 2007. *Gas Turbine Condition Monitoring and Diagnosis*.

Loboda, I. & Yepifanov, S., 2010. A Mixed Data-driven and Model Based Fault Classification for Gas Turbine Diagnosis. *Proceedings of GT2010: ASME Turbo Expo 2010*.

Mylaraswamy, D. & Venkatasubramanian, V., 1997. A Hybrid Framework for Large Scale Process Fault Diagnosis. *Computers & Chemical Engineering*.

Ogaji, S. O. T. & Sampath, S., 2003. Gas Path Fault Diagnosis of a Turbofan Engine From Transient Data Using Artificial Neural Networks. *Proceedings of IGTI/ASME Turbo Expo 2003*.

Pérez, J. L. & Loboda, I., 2014. A flexible fault Classification for gas Turbine diagnosis.

Pérez, J. L., Loboda, I. & González, I., 2014. Clasificación flexible de fallas para el diagnóstico paramétrico del turbinas de gas. *Congreso Internacional de Ingeniería electromecánica y de sistemas*.

- Pinelli, M. & Spina, P. R., 2003. Gas Path Fault Diagnosis of a Turbofan Engine from Transient Data Using Artificial Neural Network. *Proceedings of IGTI/ASME TURBO Expo 2003*.
- Roa, B. K. N., 1996. *Handbook of Condition Monitoring*.:Elsevier.
- Rummelhart, D. E. & McClelland, J. L., 1986. *Parallel Distributed Processing Explorations in the Microstructure of Cognition*.
- Russel, D. E. & Norving, P., 2003. *Artificial Intelligence: A modern approach*. s.l.:Prentice Hall.
- Sampath, S. & Sigh, R., 2006. An integrated Fault Diagnostics Model Using Genetic Algorithm and Neural Networks. *Journal of Engineering for Gas Turbines and Power*.
- Saravanamuttoo, H. I. H. & Lakshminarasimha, A. N., 1985. A Preliminary Assessment of Compressor Fouling. *ASME International Gas Turbine and Aeroengine Congress*.
- Simon, D., 2009. *Propulsion Diagnostic Method Evaluation Strategy (ProDiMES). User's Guide*.
- Treager, I., 1995. *Aircraft Gas Turbine Engine Technology*.
- Vachtsevanos, G., Lewis, F. L., Roemer, M. & Hess, A., 2006. *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*.

Anexo: Código de programación

A continuación se da una pequeña descripción acerca del algoritmo desarrollado:

En la primera parte del código se definió un apartado llamado “Parámetros de salida ProDiMES”, allí se definió el número de los parámetros de salida (variables monitoreadas y condiciones de operación) con el que trabajó el simulador así como la asignación de variables para cada componente.

A continuación se creó el apartado “Método de red neuronal”, allí se definieron las épocas, neuronas y propagación que son variables importantes para la optimización de cada red implementada (sección 5.9)

“Datos del conjunto de entrenamiento y validación”, en este apartado se definieron las entradas implementadas en la interfaz gráfica de usuario, cada variable definida de este apartado es de importancia debido a que con éstas se definieron los tamaños de los arreglos necesarios para la creación de modelos, cálculo de desviaciones para cada clase y entradas de las redes.

Una vez que ya se definieron las variables implementadas en el algoritmo, se cargan los archivos correspondientes al conjunto de entrenamiento y validación procedentes de ProDiMES así como la carga de los coeficientes necesarios para la creación de los modelos. Es allí donde se hace el cambio de dimensión de cada muestra para su mejor manipulación.

Los datos de temperatura que arroja el simulador están en grados Rankine, para tenerlos en términos del sistema internacional, se crea un apartado para convertirlos a Kelvin, posteriormente se manipulan los coeficientes de operación, necesarios para la creación de modelos de turbina de gas.

Después se desarrollaron los arreglos para las condiciones de operación con las siguientes líneas:

```
for k=1:kcoef
    AS(:,k)=US(:,npol1(k)).*US(:,npol2(k));
end;
```

donde “kcoef” representó el número de coeficientes que tuvo el modelo polinomial, npol1 y npol2 se describieron en el primer apartado del Código de programación, “US” contiene las variables de las condiciones de operación, es así que AS representó la combinación lineal de las 4 condiciones de operación.

Posteriormente se creó el apartado para la creación del modelo general, visto en la sección 5.5, se expresa el arreglo para los modelos generales de la turbinas de gas con las siguientes líneas:

```
for k=1:ky
    YES=AS*XX(:,k);
end;
```

“ky” es el número de variables monitoreadas trabajadas, “XX” representó el arreglo correspondiente a los coeficientes y por último “YES” a la Ecuación 5.3

Una vez creados los modelos para el conjunto de entrenamiento y validación se procedió a corregir cada modelo, para disminuir la individualidad de degradación en cada motor y así obtener mejores resultados de probabilidad, obteniendo la siguiente línea:

```
yp=YES.*(1+DYS0p);
```

“yp” representa la ecuación 5.5. La formación de clases vista en la sección 5.7 se hizo presente después, allí se calculan las desviaciones finales para cada motor, para así crear las clases correspondientes a cada situación de falla.

```
DYZ=(YS-yp)./yp;
```

donde “YS” contiene las variables monitoreadas que ejecutó ProDiMES, “DYZ” contiene las desviaciones para cada clase asignada.

Ya creadas las clases para cada conjunto vino el apartado de “entradas para las redes neuronales”, es aquí donde se definieron los arreglos para la pertenencia de cada clase, es decir los arreglos necesarios para comparar el conjunto de validación respecto a conjunto de entrenamiento.

Los arreglos que se definieron con anterioridad, son implementados en la sección de “entrenamiento de la red y simulación”, allí se desarrolló el código necesario para el implementación de cada red neuronal, así como las variantes importantes para la optimización de cada una.

Una vez implementada cada red se hizo el proceso de diagnóstico, arrojando las probabilidades de existencia de fallas, posteriormente se hizo la impresión de resultados para cada red propuesta.

A continuación se muestra el código completo desarrollado en Matlab.

```
clear; clc;

tic
rand('state',1); randn('state',1);
tcpu=cputime;

%*****Parámetros de salida ProDiMES*****

%Nf, P2, T2, Pamb, Nc, P24, Ps30, T24, T30, T48, Wf,
% 1 2 3 4 1 2 3 4 5 6 7
ku=4;%Número de variables "U"
ky=7;%Número de variables "Y"
kuyt=11; %Número total de variables "U", "Y" y tiempo
tk=1.8; ktk=4; ntk=[5 6 7 10]; % Para conversión Rankine-Kelvin

kcoef=15; % Número de coeficientes
```

```

npol1=[5 1 2 3 4 1 1 1 2 2 3 1 2 3 4 6 6 7]; % Columnas de U para formar
combinaciones
npol2=[5 5 5 5 5 2 3 4 3 4 4 1 2 3 4 5 6 6];

nul=1;
nu2=1; DZ=[.03,.035,.01,.009,.03,.017,.06]; %argumento de la VELOCIDAD DEL
VENTILADOR (Nf)

upro=[1 9 10 11]; %Variables "U" de ProDiMeS
ypro=[2 3 4 5 6 7 8]; %Variables "Y" de ProDiMeS

%*****Método de red neuronal*****
epcs=100; %épocas
neu=15; %neuronas
spd=0.1; %spread

nm=6; %Numero para la selección de la red neuronal a implementar

if nm==6 % Perceptrón multicapa
    nnp=neu; i_EarlyStop=0; errt=0.026;
elseif nm==13 % Red de Base Radial
    nrb=neu; spread=spd; errt=0.026;
elseif nm==14 % Perceptrón multicapa
    spread=spd;
end;

epc=epcs;

%*****Datos conjunto de entrenamiento y
validación*****

%(entrenamiento)

kdfT=19; %totalidad de clases

vpm2=50; %#Vuelos por motor

kdf1=1; %#clase sin falla
msf=100; %# motores sin falla
vpmsf=34; %# vuelos por motor sin falla
ktd1=msf*vpmsf;

kdf2=18; %#clases con falla
mcf=10; %#motores con falla
vpmcf=34; %#vuelos por motor
ktd2=mcf*vpmcf; %vuelos por clase con falla

fau=kdf1+kdf2;

mot2=kdf1*msf+kdf2*mcf;

%(validación)

vpm2v=50;

```

```

mV=10;

kdfv=19; %numero de clases para validación
ktdv=400; %numero de puntos por cada clase para validación

mot2v=mV*kdfv;

%*****Carga de archivos ProDiMES y
coeficientes*****

load ('XXr_v1A.mat') %Carga de coeficientes para los modelos

k=0;
ent=['EFS_Output_e.mat']; %Carga del conjunto de entrenamiento
load(ent);
for j=1:mot2
    k=k+1;
    ent2(k, :, :)=efs_output_c(j, :, :);
end

for j=1:11 % Variables "Y" y "U"
    for i=1:mot2 % Número de motores
        for k=1:vpm2 % Número de vuelos
            n2=(i-1)*vpm2+k; % Contador de vuelo
            UYS(n2,j)=ent2(i,k,j); %Archivo "base"
        end
    end
end

k=0;
valid=['EFS_Output_v.mat']; %Carga del conjunto de validación
load(valid);
for j=1:mot2v
    k=k+1;
    valid2(k, :, :)=efs_output_c(j, :, :);
end

for j=1:11 % Variables "Y" y "U"
    for i=1:mot2v % Número de motores
        for k=1:vpm2v % Número de vuelos
            n2=(i-1)*vpm2v+k; % Contador de vuelo
            UYSv(n2,j)=valid2(i,k,j); %Archivo "base"
        end
    end
end

%*****Conversión de Rankine a Kelvin*****

for k=1:ktk
    UYS(:,ntk(k))=UYS(:,ntk(k))/tk; %UYS[95000x11]
end;

for k=1:ktk

```

```

    UYSv(:,ntk(k))=UYSv(:,ntk(k))/tk; %UYSv[9500x11]
end;

%*****Manipulación de los coeficientes de operación*****

n1=14000;
TMPb=ones(n1,1);
TMPb(:)=UYS(:,nu1);
UYS(:,nu1)=UYS(:,nu2);
UYS(:,nu2)=TMPb(:);

TMPb=ones(n2,1);
TMPb(:)=UYSv(:,nu1);
UYSv(:,nu1)=UYSv(:,nu2);
UYSv(:,nu2)=TMPb(:);

for k=1:ku
    US(:,k)=UYS(:,upro(k));
end;
u1=ones(n1,1); u2=zeros(n1,1);
US(:,ku+1)=u1; US(:,ku+2)=u2;

for k=1:ku
    USv(:,k)=UYSv(:,upro(k));
end;
u1=ones(n2,1); u2=zeros(n2,1);
USv(:,ku+1)=u1; USv(:,ku+2)=u2;

%*****Combinación lineal de las condiciones de
operación*****

for k=1:kcoef
    AS(:,k)=US(:,npol1(k)).*US(:,npol2(k)); %AS son las combinaciones de U
(entrenamiento)
end;

for k=1:kcoef
    ASv(:,k)=USv(:,npol1(k)).*USv(:,npol2(k)); %ASv son las combinaciones
de U (validación)
end;

%*****Valores medidos*****

for k=1:ky
    YS(:,k)=UYS(:,ypro(k)); %Datos medidos (entrenamiento)
end;

for k=1:ky
    YSv(:,k)=UYSv(:,ypro(k)); %Datos medidos (validación)
end;

%*****Creación del modelo general*****

```

```

for k=1:ky
    YES(:,k)=AS*XX(:,k); %Polinomios del Modelo general (entrenamiento)
    DYS(:,k)=(YS(:,k)-YES(:,k))./YES(:,k);

end;

for k=1:ky
    YESv(:,k)=ASv*XX(:,k); %Polinomios del Modelo general (validación)
    DYSv(:,k)=(YSv(:,k)-YESv(:,k))./YESv(:,k);
end;

%*****Corrección de los modelos creados*****

%Formación de matriz de corrección "DY0p"
k=0;
for j=1:mot2
    for i=1:10 % primeros 10 vuelos sin falla
        posm=(j-1)*vpm2; k=k+1;
        DYS0(k,:)=DYS(posm+i,:);
    end
    k2=(j-1)*10+1;
    DYS0p(j,:)=mean(DYS0(k2:k,:)); %Matriz de correcciones (promedio)
end

k=0;
for j=1:mot2v
    for i=1:10 % primeros 10 vuelos sin falla
        posm=(j-1)*vpm2v; k=k+1;
        DYS0v(k,:)=DYSv(posm+i,:);
    end
    k2=(j-1)*10+1;
    DYS0pv(j,:)=mean(DYS0v(k2:k,:)); %DYS0pv=Matriz de correcciones
    (promedio) [190x7]
end

%*****Desviaciones y Formación de clases*****

%Desviación para clase 1
k=0;
for m=1:msf
    for i=1:vpmsf
        posm=(m-1)*50; k=k+1;
        yp=YES(posm+i,:).*(1+DYS0p(m,:)); %[yp]=Modelo individual/motor
        DYZ(:,k)=(YS(posm+i,:)-yp)./yp; %Desviaciones Clase 1 (sin falla)
    end
end

%Desviaciones para clases 2-19 entrenamiento
k=0;
for m=101:mot2
    for i=11:50
        posm=(m-1)*50; k=k+1;
        yp=YES(posm+i,:).*(1+DYS0p(m,:));
        DYZF(:,k)=(YS(posm+i,:)-yp)./yp; %Desviaciones Clase 2-19
    end
end

```

```

end

%Desviaciones para clases 1-19 validación
k=0;
for m=1:tot2v
    for i=11:vpm2v
        posm=(m-1)*vpm2v; k=k+1;
        ypv=YESv(posm+i,:).*(1+DYS0pv(m,:));
        DYZFv(:,k)=(YSv(posm+i,:)-ypv)./ypv; %Desviaciones Clase 1-19
    end
end

%*****Entradas para las redes neuronales *****

P1=[DYZ,DYZF];
a=3*std(P1');

for i=1:ky
    P1(i,:)=P1(i,+)/a(i);
end

P2=[DYZFv];
a=3*std(P2');

for i=1:ky
    P2(i,:)=P2(i,+)/a(i);
end

[~,DimZ1]=size(DYZ);
vector1=ones(1,ktd1);

[~,DimZ2]=size(DYZF);
vector2=ones(1,ktd2);

T1a=zeros(kdfT,DimZ1);
for i=1:kdf1
    n=(i-1)*ktd1+1;
    T1a(i,n:i*ktd1)=vector1;
end

T1b=zeros(kdfT,DimZ2);
for i=1:kdf2
    n=(i-1)*ktd2+1;
    T1b(1+i,n:i*ktd2)=vector2;
end

T1=[T1a,T1b];

[~,DimZv]=size(P2(1,:));
vector2=ones(1,ktdv);

T2=zeros(kdfv,DimZv);
for i=1:kdfv

```

```

n=(i-1)*ktdv+1;
T2(i,n:i*ktdv)=vector2;
end

%*****Entrenamiento de la red y simulación*****
MiMa=minmax(P1);
if nm==6 % Perceptrón multicapa
net=newff(MiMa,[nnp,kdfT],{'tansig','logsig'},'trainrp');
net.trainParam.epochs=epc;
net.trainParam.show=20;
net.trainParam.goal=errt;
if i_EarlyStop==1
val.P=P2;
val.T=T2;
[net,tr]=train(net,P1,T1,[],[],val);
else
[net,tr]=train(net,P1,T1);
end;

elseif nm==13 %Función base radial
nd=3;
net=newrb(P1,T1,errt,spread,nrb,nd);
elseif nm==14 %Red neuronal probabilística
net=newpnn(P1,T1,spread);
end;

A1=sim(net,P1); %Simulación entrenamiento
A2=sim(net,P2); %Simulación validación

%*****Calculo y presición del diagnostico*****

PD1a=zeros(kdfT);

for idf=1:kdf1
for itd=1:ktd1
n=(idf-1)*ktd1+itd;
[~,imax]=max(A1(:,n));
PD1a(imax,idf)=PD1a(imax,idf)+1;
end;
end;

num=n;
PD1b=zeros(kdfT);
for idf=2:kdf2+1
for itd=1:ktd2
n=num+(idf-2)*ktd2+itd;
[~,imax]=max(A1(:,n));
PD1b(imax,idf)=PD1b(imax,idf)+1;
end;
end;

PD1_1=PD1a(:,1);
PD1_1=PD1_1/ktd1;

```

```

PD1_2=PD1b(:,2:19);
PD1_2=PD1_2/ktd2;

PD1=[PD1_1,PD1_2];

PDT1=diag(PD1);
psr1=mean(PDT1);
PD1T=PD1';

PD2=zeros(kdfv);
for idf=1:kdfv
    for itd=1:ktdv
        n=(idf-1)*ktdv+itd;
        [~,imax]=max(A2(:,n));
        PD2(imax,idf)=PD2(imax,idf)+1;
    end;
end;
PD2=PD2/ktdv;
PDT2=diag(PD2);
psr2=mean(PDT2);
PD2T=PD2';

%*****Impresión de matriz de confusión*****

redi=['mlp',num2str(neu),'neu-',num2str(epcs),'epochs.dat'];
%redi=['rbn',num2str(neu),'neu-',num2str(spd),'spread.dat'];
%redi=['pnn-',num2str(spd),'spread.dat'];

fw1=fopen(redi,'wt');

fprintf(fw1,'POINTST %5i      POINTSV %5i      METHOD %2i
%2i\n',ktd1,ktdv,nm);
count=fprintf(fw1,'Regime %2i   Seed %2i\n',1,0);

for j=1:kdfT
    F(j,:)='%6.4f ';
end
[p s]=size(F(1,:)); ca=' \n'; st=reshape(F',1,(kdfT*s));
fo=[st ca];
fprintf(fw1,num2str(fo),PD1T,PDT1);
fprintf(fw1,'%6.4f \n',psr1);
fprintf(fw1,num2str(fo),PD2T,PDT2);
fprintf(fw1,'%6.4f \n',psr2);
texec=toc; % Tiempo de ejecución
count=fprintf(fw1,'TIME %8.2f \n',texec);
fclose(fw1);

```