



**UNIVERSIDAD MICHOCANA DE SAN
NICOLÁS DE HIDALGO**

**FACULTAD DE INGENIERÍA ELÉCTRICA
POSGRADO**

**DIAGNÓSTICO DE FALLAS EN UN SISTEMA NO
LINEAL EMPLEANDO REDES NEURONALES**

TESIS

QUE PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS EN INGENIERÍA ELECTRICA

PRESENTA

ING. OMAR CUEVAS SILVA

ASESOR:

DR. JUAN ANZUREZ MARIN

CO-ASESOR

DR. NUN PITALÚA DÍAZ

AGOSTO 2009



A mi padres, Pablo y Anita

A mis hermanos, Cesar, Ulises y Paty

Agradecimientos

A mis Padres Pablo Cuevas Contreras y Anita Silva Silva con quienes con su ejemplo y dedicación me han permitido lograr esta meta y a quienes nunca podré pagar de ninguna manera todo lo que han hecho por mi.

Un agradecimiento especial a mi Asesor el Dr. Juan Anzures Marín, por toda la dedicación que me brindó y con cuyas enseñanzas y consejos me ayudó a la culminación de esta Tesis.

A CONACYT quien con su apoyo económico me permitió realizar esta Tesis.

A todos y cada uno de los Profesores que tuve en mis estudios de Maestría en la División de Estudios de Posgrado de la Facultad de Ingeniería Eléctrica de la Universidad Michoacana de San Nicolás de Hidalgo por sus enseñanzas.

A los integrantes de la mesa Sinodal, quienes con sus recomendaciones y sugerencias enriquecieron esta Tesis.

Omar Cuevas Silva

Morelia, Junio 2009

Resumen

En el presente trabajo de tesis, se propone una metodología para el diseño y aplicación de un sistema de diagnóstico de fallas en sistemas no lineales usando redes neuronales. El sistema diagnóstico propuesto comprende dos etapas principales: la generación de señales residuales y la evaluación de las mismas. En la etapa de generación de las señales residuales, se usan un conjunto de redes neuronales para modelar el sistema no lineal, con y sin fallas, para generar señales de error también conocidas como residuos, mediante las cuales es posible extraer las características más significativas en lo que respecta al comportamiento. La etapa de evaluación de residuos es la encargada de indicar la localización y/o tipo de falla, además de tomar las acciones pertinentes ante la ocurrencia de ésta. De esta manera, se cubren los aspectos de detección (etapa de generación de residuos) y aislamiento (etapa de evaluación) de fallas involucrados en un sistema de diagnóstico de fallas.

Por otra parte, el tipo de red neuronal elegida para el modelado del sistema y sus fallas permite contemplar una sencilla prueba de estabilidad para sistemas no cuyo punto de equilibrio sea el origen. Esta característica es de suma importancia ya que, a diferencia de los sistemas lineales, para sistemas no lineales las pruebas de estabilidad son mucho más complejas.

Adicionalmente, se generó una interfase visual bajo la plataforma Visual Basic 6.0, la cual facilita el desempeño y distribución de las diferentes tareas empleando sus librerías dinámicas (DLL's), así como la apreciación de los resultados por parte del usuario.

El ejemplo de aplicación utilizado es un sistema de dos tanques interconectados para el cual se considera indispensable su continuo funcionamiento. Así pues, el sistema de diagnóstico de fallas deberá permitir su correcta operación aún en presencia de los diferentes escenarios de falla considerados, los cuales son; sistema libre de fallas, falla en el sensor 1, falla en el sensor 2 y falla en ambos sensores. Los resultados de la simulación son mostrados en la interfase grafica mediante los residuos de cada uno de los escenarios mencionados.

Abstract

In this thesis work, a methodology for the design implementation of a system fault diagnosis in nonlinear systems using neural networks is proposed. The fault diagnosis system proposed comprises two main stages: the residual generation signals and its evaluation. In generation residuals stage, a set of neural networks is used for modeling the nonlinear system with and without faults, to generate error signals also known as residues, which it is possible to extract most significant features in regard to behavior. The residue evaluation stage is responsible to indicate location and / or type of fault, moreover to taking appropriate actions before the fault shows up. This will cover the aspect on detection (residual generation stage) and isolation (residual evaluation stage) involved in a fault diagnosis system.

Moreover, the type of neural network chosen for modeling the system and its failures let a simple stability test for systems whose origin is its equilibrium point. This feature is extremely important because, unlike linear systems, the stability testing on nonlinear systems are much more complex.

Additionally, a visual interface in Visual Basic 6.0 platform was created using its dynamic library (DLL's), this provides better performance and distribution of tasks involved.

The study case is a two tanks interconnected system, in which its continuous operation is considered essential. So therefore, the system of detection and isolation faults must allow its operation still in presence of different scenes considered, which are; free faults system, fault in sensor 1, fault in sensor 2 and fault in both sensors. The simulation results are displayed in the graphical interface through the residues from each mentioned scene.

Índice general

Resumen	iii
Abstract	iv
Lista de figuras	vii
Lista de tablas	x
Lista de publicaciones	xi
1 Introducción	1
1.1 Motivación.	2
1.2 Estado del arte.	3
1.2.1 <i>Sistemas de potencia</i>	7
1.2.2 <i>Líneas de transmisión</i>	7
1.2.3 <i>Robots y servomecanismos</i>	8
1.2.4 <i>Motores</i>	9
1.2.5 <i>Turbinas de vapor</i>	10
1.2.6 <i>Control simbólico</i>	11
1.2.7 <i>Detector de partículas y plantas nucleares</i>	12
1.3 Objetivo y contribución	13
1.4 Organización de la tesis	15
2 Redes neuronales en el diagnóstico de fallas	16
2.1 Las redes neuronales	17
2.1.1 Tolerancia a fallas	18
2.1.2 Cualidades del Perceptrón	19
2.1.3 Aproximación de funciones	21
2.2 Tipos de Redes	23
2.2.1 Estructura	23
2.2.2 Algoritmos de aprendizaje	24
2.3 Redes neuronales para modelado	25

2.3.1	Incertidumbre paramétrica y estructural	25
2.3.2	Identificación de sistemas dinámicos	26
2.3.3	Estructuras de modelos no lineales basados en redes neuronales	29
2.3.4	Redes dinámicas multicapa	33
2.4	Control robusto, control adaptivo y control inteligente	37
2.5	Estructura de un sistema de detección y aislamiento de fallas (FDI)	38
2.5.1	Generador y evaluación de residuos neuronal	40
3	Metodología de diagnóstico	43
3.1	Generación de residuos	45
3.2	Toma de decisiones	46
4	Diagnóstico de fallas en un sistema de nivel de líquidos	49
4.1	Modelo del sistema	49
4.2	Modelo neuronal del sistema de dos tanques	52
4.3	Diseño de la interfaz en Visual Basic	53
4.4	Respuesta de los modelos	55
4.4.1	Sistema libre de falla	56
4.4.2	Falla en el sensor 1	60
4.4.3	Falla en el sensor 2	63
4.4.4	Falla en ambos sensores	66
4.5	Zonas de incertidumbre	69
5	Conclusiones y trabajo futuro	73
A	Algoritmo <i>BackPropagation</i>	77
A.1	Propagación hacia delante	78
A.2	Propagación hacia atrás	78
A.3	Consideraciones del algoritmo	80
B	Glosario	81
	Bibliografía	84

Lista de figuras

1.1	Diagrama de flujo del esquema de adaptación a fallas	9
1.2	Esquema de entrenamiento con un sistema experto	10
1.3	Proceso de detección y diagnóstico de fallas en el motor	10
1.4	Conversión simbólica a numérica	11
1.5	Esquema de diagnóstico experto para un detector de partículas	12
1.6	Red neuronal usada por el sistema experto	13
1.7	Estructura jerárquica para el diagnóstico de fallas en un sistema de gran escala	14
2.1	Modelo del Perceptrón	18
2.2	Funciones de activación más comunes	18
2.3	Separación lineal del espacio de entrada (b) de un Perceptrón (a)	19
2.4	Clasificación de los patrones en las funciones AND, OR y XOR	20
2.5	Solución al problema de clasificación en la función XOR	20
2.6	Delimitación de una región en un espacio de entrada con tres Perceptrones . .	21
2.7	Mapeo de múltiples Perceptrones	22
2.8	Clasificación general de las redes neuronales.	23
2.9	Estructura ARX (AutoRegressive, eXternal input).	26
2.10	Esquema del Modelo de identificación Espacio-Estado	28
2.11	Estructuras; (a) NNFIR, (b) NNARX, (c) NNARMAX, (d) NNOE, (e) NNSSIF	30
2.12	Identificación en paralelo	31
2.13	Identificación en serie-paralelo	32
2.14	Red Dinámica multicapa	33
2.15	Patrones de entrenamiento de la red	34
2.16	Diagrama de fase del sistema descrito por las ecuaciones 2.13.	36
2.17	Identificación de un sistema de ecuaciones	37
2.18	Error medio-cuadrático en la identificación.	38
2.19	Dinámica alrededor del origen	39
2.20	Sistema de control adaptivo	40
2.21	Esquema general de diagnóstico de fallas basado en el modelo	41
2.22	Esquema de detección de fallas neural basado en banco de modelos del proceso	42

3.1	Procedimiento básico de identificación de sistemas	44
3.2	Procedimiento básico de implementación del sistema FDI neuronal	45
3.3	Generador de residuos neuronal	46
3.4	Toma de decisiones	47
3.5	Esquema de acciones de los consecuentes	48
4.1	Sistema de tanques de agua (FESTO Compact unit PCS)	50
4.2	Esquema de conexiones del sistema de nivel de líquido de dos tanques interconectados	51
4.3	Diagrama de fase para el sistema sin entrada	52
4.4	Modelado neuronal del sistema de dos tanques	54
4.5	Ventana de visualización	55
4.6	Esquema completo del sistema FDI	56
4.7	Comportamiento normal del sistema sin falla	57
4.8	Señales de los cuatro modelos neuronales en condiciones normales de operación	58
4.9	Señales de error de los cuatro modelos neuronales en condiciones normales de operación	58
4.10	Señales de error booleano de los cuatro modelos neuronales en condiciones normales de operación	59
4.11	Comportamiento en presencia de falla en el sensor 1	60
4.12	Señales de los cuatro modelos neuronales en presencia de la falla en el sensor 1	61
4.13	Señales de error de los cuatro modelos neuronales en presencia de la falla en el sensor 1	62
4.14	Señales de error booleano de los cuatro modelos neuronales en presencia de la falla en el sensor 1	62
4.15	Comportamiento en presencia de falla en el sensor 2	63
4.16	Señales de los cuatro modelos neuronales en presencia de la falla en el sensor 2	64
4.17	Señales de error de los cuatro modelos neuronales en presencia de la falla en el sensor 2	65
4.18	Señales de error booleano de los cuatro modelos neuronales en presencia de la falla en el sensor 2	65
4.19	Comportamiento en presencia de falla en ambos sensores	66
4.20	Señales de los cuatro modelos neuronales en presencia de falla en ambos sensores	67
4.21	Señales de error de los cuatro modelos neuronales en presencia de la falla en ambos sensores	68
4.22	Señales de error booleano de los cuatro modelos neuronales en presencia de falla en ambos sensores	68
4.23	Esquema final de reglas para la toma de decisiones	70
4.24	Esquema final de acciones	71
4.25	Secuencia con múltiples fallas	72
A.1	Procedimientos del algoritmo <i>Back-Propagation</i>	77

A.2 Propagación hacia adelante	78
A.3 Esquema conceptual de la propagación hacia atrás	79
A.4 Propagación hacia atrás	79

Lista de tablas

1.1	Especificaciones de la conversión Numérico/Simbólico	11
3.1	Reglas involucradas en la toma de decisiones	47
4.1	Resumen de incertidumbres	69
4.2	Reglas para las combinaciones de incertidumbre	69
A.1	Ecuaciones de error para los nodos	78

Lista de publicaciones

Publicaciones relacionadas con el presente trabajo son:

- "Unknown Inputs Observers Desing for Fault detection in a Two-Tank Hydraulic System " en "Electronics, Robotics, and automotive Conference", CERMA 2008, en Cuernavaca Morelos. ISBN-13: 978-0-7695-3320.
- "Modelado de sistemas no lineales usando redes neuronales en visual Basic 6.0" en el primer "Congreso Nacional de Ingeniería y Arquitectura" 2008, en la UMSNH. ISBN: 978-969-9322-40-5.
- "Estabilidad de Sistemas No-lineales: Sistema de Nivel de Líquidos de Dos Tanques Interconectados" en la "Revista de Ingeniería Eléctrica y Electrónica y Computación, Vol. 5, No. 2 Diciembre 2008" del departamento de Ing. Eléctrica y Electrónica en el Instituto Tecnológico de Sonora. ISSN 1870-9532.
- "Identificación de Sistemas no Lineales usando Redes Dinámicas Recurrentes" en la "V Semana Nacional de Ingeniería Electrónica del Centro Universitario de la Ciénega" del 7, 8 y 9 de octubre 2009, Universidad de Guadalajara, Ocotlán Jalisco (Publicación aún sujeta a revisión).
- "The Fault Diagnosis Problem: Residual generators design using neural networks in a two-tanks interconnected system" en "Electronics, Robotics, and automotive Conference", CERMA 2009, en Cuernavaca Morelos (Publicación aceptada para presentarse en septiembre de 2009).

Además de las participaciones mencionadas anteriormente, este trabajo de tesis forma parte de un proyecto de investigación PROMEP-2007 bajo el programa de reincorporación de exbecarios promep EXB-132.

Capítulo 1

Introducción

En este capítulo se presentan los antecedentes al problema del diagnóstico de fallas en sistemas no lineales, así como las definiciones básicas sobre el tema de diagnóstico de fallas. Asimismo se define la motivación, el objetivo del presente trabajo y finalmente la organización de la tesis.

Los sistemas de diagnóstico y aislamiento de fallas (por sus siglas en ingles "Fault Detection and Isolation, FDI") están motivados por la necesidad de conocer y en su caso predecir la ocurrencia de fallas en los sistemas de control. Para los sistemas de alta seguridad, las consecuencias debido a un mal funcionamiento pueden ser extremadamente serias en términos de mortalidad humana, impacto ambiental y pérdidas económicas. Por esta razón existe una creciente necesidad de esquemas de supervisión en línea y diagnóstico de fallas para incrementar la confiabilidad de todos los sistemas en general.

Para mantener el alto nivel de seguridad, desempeño y confiabilidad en los sistemas de alto riesgo o procesos controlados es importante que los errores, fallas en componentes y operación anormal del sistema sean detectados rápidamente para evitar que el sistema sobrepase los niveles normales de operación; de igual manera la fuente y severidad de cada mal funcionamiento debe ser diagnosticado para tomar una acción correctiva. El operador humano puede corregir los errores en el sistema, por ejemplo, deshabilitando la parte del proceso que presenta el mal funcionamiento y reconfigurando el lazo de control o señales de referen-

cia. Sin embargo, la tendencia de auto corrección-reconfiguración es inherente en el control automático.

El proceso de detección de fallas oportuno mientras la planta se encuentra aún en operación en una región controlable es muy útil para impedir el progreso de los eventos anormales y reducir la pérdida de productividad. Aun para los sistemas cuya seguridad no es crítico, las técnicas de diagnóstico de fallas en línea pueden ser usadas para mejorar la eficiencia de la planta, disponibilidad y confiabilidad.

En esta tesis se aborda el problema de diagnóstico de fallas usando redes neuronales para modelar un sistema no lineal y sus fallas. Como caso de estudio se considera el modelo de un sistema hidráulico de dos tanques; dicho sistema es un prototipo de laboratorio que se encuentra físicamente en el laboratorio Ing. Eléctrica de la Universidad Autónoma de Baja California (UABC) campus Mexicali.

1.1 Motivación.

Las cuestiones de seguridad y su relación con el posible peligro en la operación de un sistema son la primera motivación de los sistemas de diagnóstico de fallas. Ya que da lugar a la búsqueda de métodos automáticos que realicen las tareas de detección y aislamiento de fallas. En sistemas donde la operación continúa de sus procesos es prioridad, el impacto de algunas fallas puede interrumpir su operación y generar resultados no deseados. Esta es otra razón mas para los sistemas de diagnóstico.

Si bien no es nueva la aplicación de las redes neuronales en el mundo, su uso mayormente a quedado en los usuarios que poseen en general una formación más científica o técnica. Por ello, el motivo principal de este trabajo es mostrar la utilidad y la facilidad de aplicación de un algoritmo de diagnóstico de fallas basado en redes neuronales aplicado a un sistema no lineal, teniendo como principal objetivo el modelado de la planta y sus fallas.

Otra motivación importante es que, en la actualidad, para el diseño de un algoritmo de control y de diagnóstico de fallas en tiempo discreto las ecuaciones de diferencias que definen a un sistema se pueden pasar directamente al sistema real a través de un programa de

computadora y una tarjeta de adquisición de datos diseñada por el usuario, con lo cual se facilita la implementación y los costos están al alcance de prácticamente cualquier persona o institución que posea una computadora de escritorio [5].

Todo esto motiva el desarrollo del presente trabajo de investigación; en el sentido de presentar una alterna de solución para el problema de diagnóstico que provea un claro ejemplo de aplicación de las redes neuronales en un caso práctico.

1.2 Estado del arte.

El punto de partida del desarrollo de métodos de diagnóstico de fallas coincide con el desarrollo alcanzado en la década de los setentas por las computadoras, con el nacimiento del primer microprocesador (1972). Posteriormente, en los años ochentas inician las aplicaciones con técnicas de inteligencia artificial, tales como redes neuronales y lógica difusa; en la década de los noventas creció notablemente el interés por el diagnóstico de fallas en las plantas químicas. En la última década, se han estudiado técnicas robustas [1], [2].

Debido a la evidente utilidad de esta área, la Federación Internacional de Control Automático (IFAC) creó el primer comité SAFEPROCESS en 1991 y posteriormente en 1993 se convierte en un comité técnico que realiza reuniones cada tres años. Este comité tiene entre sus tareas estandarizar conceptos y definiciones en esta área, en beneficio de la industria y la comunidad científica [3].

Históricamente existen dos enfoques en los sistemas de diagnóstico en los que se consideran la redundancia por hardware y redundancia analítica o funcional.

- Redundancia por hardware: Si un sensor, actuador, computadora o programa para controlar una variable en particular falla en el sistema, la redundancia del mismo proporcionará información correcta pues cada componente individualmente es duplicado. De manera tal que el sistema mantenga un desempeño adecuado independientemente del escenario de falla.
- Redundancia Analítica o funcional: Se basa en emplear relaciones entre diferentes

mediciones, en lugar de duplicar cada componente individualmente. La idea es emplear relaciones entre las diferentes variables medidas en el proceso bajo análisis; por ejemplo, entradas-salidas, salidas-salidas, entradas-entradas. De esta manera, en un esquema de Redundancia Analítica no es necesario hardware adicional y por esta razón es potencialmente más utilizable que el enfoque de redundancia por hardware.

Para sistemas lineales, el diagnóstico de fallas, se ha analizado ampliamente en [1]; [14]; [4]; [15]; [6]. Mientras que para no lineales se han propuesto varios enfoques por ejemplo, basado en observadores, espacio de paridad y estimación de parámetros, así como observadores difusos y redes neuronales artificiales. En algunos sistemas no lineales se ha demostrado que es suficiente realizar una linealización alrededor de un punto de operación.

De acuerdo a la información que existe en la literatura sobre las técnicas de diagnóstico de fallas se distinguen tres grandes grupos; según estén basados en el análisis de las señales, el conocimiento de información entrada-salida o el modelo del sistema [12]. Las técnicas basadas en el modelo son las más utilizadas puesto que proporcionan de forma natural herramientas para resolver el problema. A continuación se describen los tres grupos de técnicas haciendo énfasis en aquellas basadas en el modelo y el conocimiento del sistema, ya que se emplearán en la presente tesis.

- Los *métodos basados en el análisis de señales* corresponden a aquellos que emplean técnicas estadísticas o minería de datos. Una de las aplicaciones más importantes es en el área de sistemas eléctricos de potencia; donde, a partir de un conjunto de datos de un sistema libre de fallas se comparan con datos en tiempo real del sistema. De esta manera, mediante el análisis estadístico de los datos se determinan si se presentan fallas en el sistema. Una de las técnicas estadísticas más empleadas es llamada “Análisis de los Componentes Principales (PCA)”, siendo su principal área de aplicación el monitoreo de procesos industriales; dicha técnica permite reducir la dimensión del modelo de la planta mediante el uso de dependencias lineales entre las variables [14].
- Los *métodos basados en el conocimiento de la información*, corresponden a los métodos de clasificación; uno de los ejemplos de esta técnica y que ha sido muy usado son las

Redes Neuronales Artificiales (ANN). En todas las aplicaciones donde se emplean las redes neuronales se pueden identificar dos fases principales; la fase de aprendizaje o entrenamiento y la fase de prueba. En la fase de entrenamiento, se usan un conjunto de datos o patrones de entrenamiento para determinar los parámetros de diseño (pesos) que definen el modelo neuronal. Una vez entrenado este modelo, se usa la llamada fase de prueba, en la que se procesan los patrones de prueba que constituyen la entrada habitual de la red. Las redes neuronales reúnen características útiles para el diagnóstico de fallas, ya que debido a su capacidad de aprender pueden manejar igual procesos no lineales como procesos indeterminados. Además, cuentan con otra propiedad interesante; la capacidad de adaptarse en línea, es decir que la fase de prueba y de entrenamiento se puede hacer en tiempo real; puesto que son muy tolerantes a las señales medidas que cuentan con componentes de ruido.

- Los *métodos basados en el modelo* es una de las técnicas más usadas para la detección y aislamiento de fallas, la cual en general identifica el nivel o tendencia de una señal en particular (llamada residuo) y toma una acción cuando esta alcanza un umbral establecido. Estas técnicas mediante límites o umbrales son sencillas de implementar; sin embargo, tienen la posibilidad de generar falsas alarmas en eventos debidos a ruido, variaciones de entradas y cambios en los puntos de operación si no se efectúa una adecuada selección de los umbrales. La principal ventaja de los métodos basados en el modelo es precisamente que no se requieren componentes físicos adicionales para realizar el algoritmo de detección de fallas, ya que puede ser implementado mediante software. Por lo tanto, los únicos requerimientos son la capacidad de almacenamiento y cálculo en la computadora. Algunos de los métodos basados en el modelo que han sido reportados en la bibliografía son:

- Métodos basados en observadores [15] [1] [16].
- Métodos basados en espacio de paridad [17].
- Métodos basados en estimación de parámetros [18] [19].

- Métodos de diagnóstico en sistemas dinámicos no lineales [1] [20].
- Métodos robustos de diagnóstico de fallas [22] [1] [21].

Dentro del marco de los métodos basados en el conocimiento de la información, han aparecido una gran variedad de ejemplos de aplicación de sistemas FDI en los últimos años, se puede observar en la cantidad de artículos publicados recientemente. En los que incluyen optimización mediante algoritmos genéticos [24], algoritmos basados en colonias de hormigas [25], sistemas empleando wavelets [26], métodos combinados [27], redes neurodifusas [28] y sistemas inspirados en la naturaleza del sistema inmune [29]. Sin embargo la mayoría preservan en cierto grado las características del diagnóstico de fallas clásico, como los presentados en [30], [31], [32]. Algunos de estos ejemplos de aplicaciones FDI con redes neuronales se comentan enseguida.

En [33] se discute el uso de una red neuronal para la detección de fallas en sensores aplicado a una estructura de control de una planta lineal de primer orden sujeto a alta incertidumbre del modelo. La principal ventaja del enfoque propuesto sobre los otros métodos es la habilidad del método para capturar características no lineales, la posibilidad de entrenamiento en línea y la rapidez durante la implementación en línea. En [34] se da una revisión del control con tolerancia a fallas, concentrándose en el control robusto, la redundancia paralela y analítica, y finalizando con los sistemas basados en reglas y las redes neuronales artificiales. La conclusión del autor es que los controles inteligentes tolerantes a fallas pueden mejorar las características de los sistemas.

En [35] se describe un marco de referencia para deducir un modelo de diagnóstico de falla empleando redes neuronales en un nivel abstracto. Los modelos con falla son una representación abstracta de los defectos físicos que pueden ocurrir en un sistema y son de especial ayuda en la determinación de la confiabilidad de un sistema con redes neuronales. La naturaleza abstracta de cada modelo con falla incrementa la posibilidad de que sean de carácter genérico. Un modelo genérico de fallas es obviamente una característica deseable, ya que permitirá hacer comparaciones válidas y fáciles entre diversos sistemas neuronales y no-neuronales. Además de que pueden describir tipos de fallas, dónde y cómo ocurren en el sistema. Sin

embargo, se torna más difícil expresar realmente cómo es visto un sistema, convirtiéndolo cada vez más en un nivel abstracto, especialmente en la definición de cómo se manifiesta una falla [35].

1.2.1 *Sistemas de potencia*

En [36] se describe una estrategia con redes neuronales para detección de fallas incipientes debidas al efecto de altas impedancias en un sistema eléctrico de distribución de potencia. Este tipo de fallas pueden no ser fácilmente detectadas por los dispositivos de protección modernos. El enfoque presentado consiste en registrar muestras de la subestación de corriente durante condiciones normales y anormales de operación. Estas muestras son empleadas como patrones de entrenamiento en la red, haciendo posible la detección de fallas para cada caso individual. Un conjunto de 50 casos de entrenamiento se generaron empleando el programa EMTP de los cuales los casos de las fallas por alta impedancia (high-impedance fault, HIF) representan el 30 % de los casos. La red neuronal fue configurada con 2 capas ocultas, 200 nodos de entrada y 1 de salida. El número de nodos en la primera capa fue de 200 y 400 en la segunda. En la fase de detección, de 15 HIF casos presentados, la red respondió correctamente en 10. Y de 85 casos sin falla, la red respondió erróneamente a 17 causando falsas alarmas. Algunos otros ejemplos de aplicaciones en los sistemas de potencia se presentan en [37], [38] y [39].

1.2.2 *Líneas de transmisión*

En [40] se explora la posibilidad de usar redes neuronales para identificar fallas que pueden ocurrir en un sistema de potencia AC-DC. Se proponen tres diferentes arquitecturas de redes para realizar el diagnóstico. Se concluye que las redes neurales pueden ser usadas para distinguir fallas típicas que pueden ocurrir en un sistema de potencia de ac-dc. El sistema puede sensar los voltajes de AC en cualquiera de los buses como valores rms, con o sin la información del ángulo de fase, o los valores instantáneos de las ondas seno muestreadas.

En [41] se presenta un método para crear un árbol de decisiones y aplicarlo a un sistema de

potencia de líneas de transmisión con diagnóstico de fallas.

1. Primero, se consideran un conjunto de fallas; como la falla de línea a tierra, falla línea a línea, como una categoría a ser clasificada. Posteriormente se consideran los atributos de voltajes y corrientes de línea como factores determinantes en la categoría.
2. De estos atributos, el mejor conjunto es seleccionado para crear el árbol de dediciones más eficiente.
3. El procedimiento se deriva de encontrar fallas en las líneas de distribución con el árbol de dediciones.

El algoritmo para el diagnóstico puede ser determinado automáticamente sin asistencia humana.

En [42] se describe una red neuronal discriminante para protección de las líneas de transmisión. El discriminador usa muestras de voltaje y corrientes del sistema de potencia para tomar decisiones. Se emplea 12 neuronas en la capa de entrada, 4 neuronas en la capa oculta y 1 neurona en la capa de salida. Otras aplicaciones similares pueden verse en [43], [44], [45], [46] y [47].

1.2.3 *Robots y servomecanismos*

En [48] se desarrolla el diagnóstico de fallas y adaptación del control para un sistema robótico (ver la figura 1.1). Son propuestos observadores no lineales basados en la disponibilidad del modelo y se construyen observadores basados en un conjunto de fallas posibles. La adaptación del control es desarrollado usando un controlador normal más un compensador neuronal para capturar las características no lineales de las fallas. Posteriormente si la falla es aislada exitosamente, se presenta un controlador adaptivo basado en la información de la falla obtenida por el esquema de aislamiento de la falla.

Otros sistemas diagnósticos en robots pueden verse en [49], [50], [51], [52] y en [53].

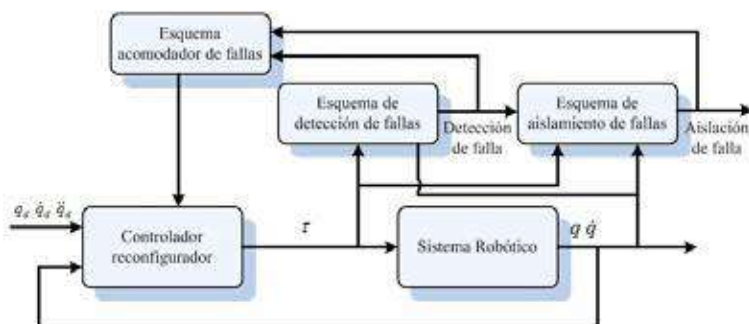


Figura 1.1: Diagrama de flujo del esquema de adaptación a fallas

1.2.4 Motores

En [54] se anota que la posibilidad de fallas incipientes es inherente en las máquinas eléctricas debido al desgaste involucrado en la conversión de energía eléctrica a mecánica y viceversa. Monitoreando este desgaste de manera oportuna, pueden mejorarse los costos de mantenimiento y la confiabilidad de los sistemas. Las redes neuronales pueden ser entrenadas por el conocimiento de un sistema experto para mejorar la detección de fallas como se muestra en la figura 1.2. Con I como la corriente en estado estable, w la velocidad del rotor, N y B son el número de vueltas en el bobinado principal y el coeficiente de amortiguamiento, respectivamente, y son una descripción cuantitativa de las condiciones del motor. Además, son asociadas dentro de tres niveles de condición {good, fair, bad} para generar N_c y B_c , la descripción cualitativa. \hat{N}_c y \hat{B}_c son las respectivas estimaciones hechas por el detector de fallas neuronal. Después de ser entrenada, la red deberá contener el conocimiento necesario para la detección de las fallas (ver figura 1.3) sin ayuda del sistema experto.

En [55] se discuten las consideraciones de diseño de las redes neuronales así como la implementación, tamaño del conjunto de datos de entrenamiento, valores de los parámetros de entrenamiento, y los criterios de paro en la implementación del diagnóstico de fallas en motores.

En [56] se investiga dos de las más comunes fallas incipientes en motores de inducción, con la suposición de que todo lo demás se encuentra en condiciones normales: falla de aislamiento en

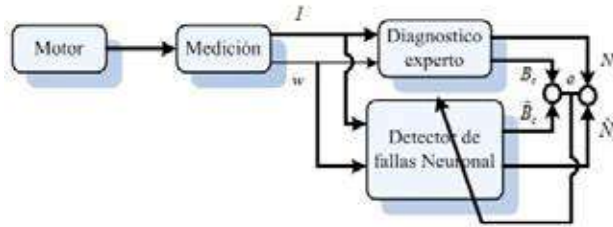


Figura 1.2: Esquema de entrenamiento con un sistema experto

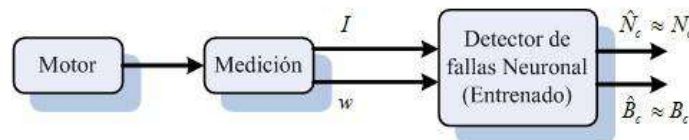


Figura 1.3: Proceso de detección y diagnóstico de fallas en el motor

el estator, desgaste en los rodamientos. El detector de fallas en línea desarrollado en el documento, esta compuesto de dos partes: un filtro neuronal para disturbios y ruido (Disturbance and Noise Filter Artificial Neural Network, NDF-ANN), y un detector de fallas incipientes para el motor de inducción (Incipient Fault Detector for Induction Motors Artificial Neural Network, IFDIM-NNA). Los resultados muestran que el detector de falla es robusto al ruido y disturbios durante la simulación en tiempo real. La combinación de disturbios, ruidos y fallas del motor mejoraron excelentemente, dando una eficiencia del 95% a 99% en el diagnóstico de fallas.

1.2.5 Turbinas de vapor

En [57] una red neuronal y algoritmos genéticos fueron introducidos dentro de la minería de datos y diagnóstico de fallas de una turbina de vapor, pudiendo sobrepasar el cuello de botella de los métodos tradicionales conocidos. La precisión del diagnóstico fue del 84% probando así la factibilidad de desarrollar el sistema de diagnóstico de una turbina de vapor.

Regiones de salida de la planta	Salida del convertidor numérico/simbólico
$h(t) = h_1$	y_1
$h_1 < h(t) < h_2$	y_2
$h_2 \leq h(t) \leq h_3$	y_3
$h_3 < h(t) < h_4$	y_4
$h(t) = h_4$	y_5

Tabla 1.1: Especificaciones de la conversión Numérico/Simbólico

1.2.6 Control simbólico

En [58] se propone un tipo de controlador, denominado controlador simbólico y una red tipo perceptrón multicapa para clasificar datos numéricos y asignar los símbolos apropiados a varias de las clases generadas. Esta conversión numérica a simbólica resulta en un tipo de extracción de información la cual es similar la llamada *reducción de datos* en reconocimiento de patrones. El controlador simbólico usa símbolos como entrada Y_j y una entrada de referencia R_j , además de generar la entrada de control a la planta U_j . El convertidor numérico-simbólico transforma los datos medidos $y(t)$ en símbolos Y_j . El convertidor simbólico-numérico (ver figura 1.4 y tabla 1.1) transforma los datos medidos U_j en la entrada numérica $u(t)$. Su uso en control autónomo para diagnóstico de falla se discute para un sistema tanque almacenador de agua, un sistema biológico y una aeronave.

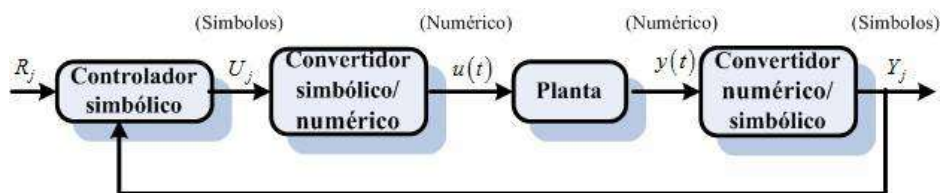


Figura 1.4: Conversión simbólica a numérica

1.2.7 *Detector de partículas y plantas nucleares*

En [59] se usa un sistema de diagnóstico experto para mejorar el diagnóstico de fallas en un detector de posición de partículas físicas. Una parte de este sistema experto es compuesto por una red neuronal, como se muestra en la figura 1.5, desempeñando la función de clasificación de fallas. El objetivo de la red es responder a algunos malfuncionamientos en el sistema y sus salidas son usadas por el sistema experto e integradas como información adicional para mejorar el diagnóstico.

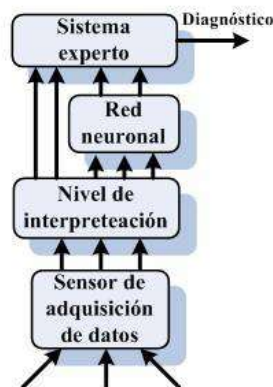


Figura 1.5: Esquema de diagnóstico experto para un detector de partículas

La red empleada se muestra en la figura 1.6. Esta posee tres capas de unidades de procesamiento. La capa de entrada esta constituida por 35 unidades sujeta a los valores de histograma a ser clasificados. La capa de salida tiene cuatro unidades de salida, las cuales determinan la clase de falla en el sistema. La capa oculta esta constituida por cuatro unidades. Según el autor el método empleado parece ser apropiado para resolver este tipo de problemas y alienta a la implementación física del prototipo.

En [60] se reporta que el método propuesto fue exitosamente aplicado para diagnosticar fallas en un presurizador conectado en la planta de potencia nuclear Kori, unidad 2 localizada en la provincia Kyungsang, Korea del sur. En este, se propone que el sistema sea dividido en

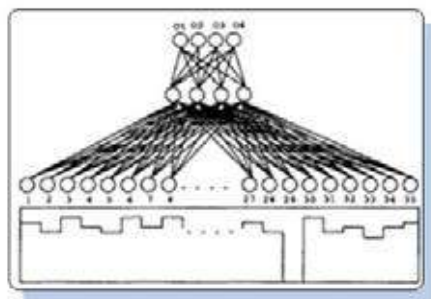


Figura 1.6: Red neuronal usada por el sistema experto

varios pequeños subsistemas (ver figura 1.7) a fin de que cada red neuronal tome cargo del diagnóstico de fallas de cada subsistema en tiempo real con eficiencia deseable.

Es decir el sistema completo para el diagnóstico es compuesto jerárquicamente de redes neuronales tipo modulares de bajo nivel y un nivel superior supervisor. Otra aplicación similar se manejan en [61].

Otros diversos ámbitos de la ingeniería han sido estudiados con importantes resultados en la implementación de las redes neuronales dentro de los sistemas FDI, como son; la industria del automóvil en [62], [63], [64], procesamiento de señales [65], antenas [66], alarmas [67], misiles [68], aeronáutica [69], [70], satélites [71], [72], [73], actuadores hidráulicos [74], administración de yacimientos petroleros [75], sistemas mecánicos [76], inversores [77], [78], Circuitos analógicos [79], [80], [81], transformadores [82], [83], [84], [85], [86], [87], [88], [89], [90], procesos industriales [91], generadores síncronos [92], [93], [94], [95], [96], generadores hidroeléctricos [97]. Por mencionar sólo algunos.

1.3 Objetivo y contribución

El presente trabajo tiene como objetivo el desarrollar un algoritmo para el diagnóstico de fallas basado en redes neuronales empleando como plataforma de programación Visual Basic 6.0. El algoritmo integrará un conjunto de herramientas modulares disponibles como librerías

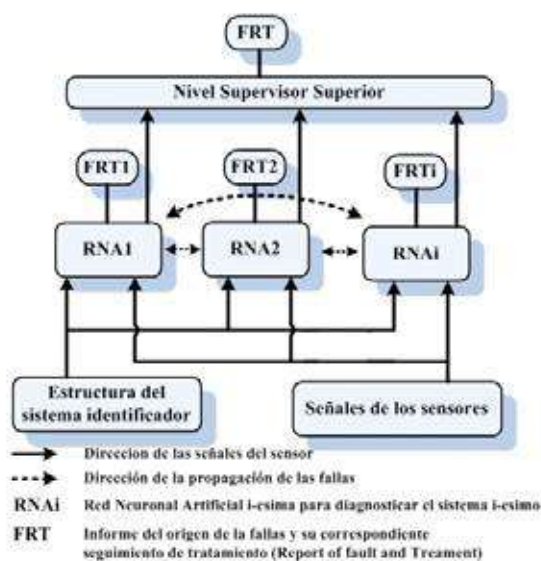


Figura 1.7: Estructura jerárquica para el diagnóstico de fallas en un sistema de gran escala

dinámicas (Dynamic Link Library, DLL's) que contendrán los métodos y funciones de la implementación. Dichas librerías son el resultado de la implementación del sistema FDI con redes neuronales en Visual Basic y serán de gran utilidad para futuros trabajos. Pues, además de poder incluirse en otros programas, pueden ser usadas incluso en otras plataformas de desarrollo como Labview y Visual C.

La investigación y desarrollo de este trabajo es por si misma una recopilación de los diferentes términos y conceptos implicados en el diseño de un sistemas de diagnóstico de fallas.

Con el desarrollo de este proyecto de investigación se tiene la oportunidad el poder comparar, en el aspecto de los FDI, los métodos basados en el modelo y los métodos basados en el conocimiento de la información. Por ello se espera obtener resultados importantes y relevantes en el área del diagnóstico de fallas.

Fuera del ámbito técnico, la gente que no conoce las redes neuronales tiene en general la creencia que las mismas son difíciles de usar, que requieren conocimientos avanzados o que son lentas e ineficientes. Posiblemente esto se deba a los fracasos en los inicios de la inteligencia artificial que postergaron su desarrollo. Por lo tanto, un segundo objetivo de este trabajo

es el de ofrecer una interpretación sencilla sobre los conceptos arraigados acerca de las redes neuronales, demostrando su rapidez y facilidad de uso en un sistema de diagnóstico de fallas.

1.4 Organización de la tesis

En el capítulo 2 se hace una revisión de las características, propiedades y clasificación de las redes neuronales. Así como la importancia y papel que desempeña en los sistemas de diagnóstico de fallas. En el capítulo 3 se presenta el modelo del sistema de tanques interconectados. Esto con la finalidad de simularlo para los distintos escenarios de falla y entrenar un conjunto de redes neuronales que identifiquen dichos escenarios. En el capítulo 4 se describe el algoritmo para implementar el sistema FDI, así como la estrategia de entrenamiento para las redes neuronales. En el capítulo 5 se presenta el esquema completo para el sistema FDI implementado y las gráficas de respuesta de cada uno de los modelos neuronales en los diferentes escenarios de falla. Y finalmente en el capítulo 6 se dan algunas conclusiones del presente trabajo y posibles aportaciones a futuro.

Capítulo 2

Redes neuronales en el diagnóstico de fallas

En esta sección se proporciona una introducción comprensible de algunos modelos importantes de redes neuronales que son necesarios como parte de los antecedentes para los capítulos posteriores.

La sección inicia con una descripción de las redes neuronales y las cualidades del perceptrón simple, continúa con una clasificación de las redes neuronales considerando la estructura o el algoritmo de aprendizaje, posteriormente se presentan las estructuras de modelos no lineales basados en redes neuronales a la vez que se destaca a las redes dinámicas recurrentes para identificación de sistemas dinámico. Más adelante se continúa con una discusión entre las cualidades del control robusto, control adaptivo y el control inteligente desde el punto de vista de la robustez ante incertidumbres en el sistema. Finalmente se considera la estructura general de un sistema de detección y aislamiento de fallas mediante la generación y evaluación de residuos empleando redes neuronales como modelos del sistema y de sus fallas.

2.1 Las redes neuronales

Las redes neuronales son usadas para resolver una gran variedad de problemas incluyendo los aspectos de modelado y de control de sistemas dinámicos complejos. Por ejemplo, en el control clásico se maneja un modelo matemático; en el control difuso ([115], [114]), un conjunto de reglas; y las redes neuronales principalmente consisten de datos numéricos que describen el comportamiento del proceso [98].

Las redes neuronales ofrecen una arquitectura altamente estructurada, con capacidades de aprendizaje y generalización, que intenta imitar el mecanismo neurológico del cerebro. Una red neuronal almacena conocimiento de una manera distribuida dentro de sus pesos, los cuales han sido determinados por medio del entrenamiento (aprendizaje) con muestras conocidas. La habilidad de generalización para entradas nuevas está basada en la estructura algebraica inherente de la red neuronal [99].

Las redes neuronales, o más específicamente, las redes neuronales artificiales (RNA) son modelos inspirados de nuestro conocimiento del sistema nervioso biológico [98]. De acuerdo con [101], se sabe que el cerebro humano posee de 10^{10} a 10^{11} neuronas comunicadas con una red nerviosa altamente interconectada. Aunque las RNA no alcanzan la complejidad del cerebro, hay dos aspectos similares entre las redes biológicas y las artificiales. Primero, los bloques de construcción de ambas son sencillos elementos computacionales (aunque las RNA son mucho más simples que las biológicas) altamente interconectados. Segundo, la conexión entre neuronas determina la función de la red.

Una red neuronal artificial es una colección de neuronas artificiales. El modelo matemático de una neurona biológica, como el Perceptrón (figura 2.1), en su forma simple son vistas como unidades elementales de procesamiento de información en un sistema neuronal.

La señal total de entrada es una combinación lineal de todas las entradas multiplicadas cada una por un coeficiente o peso. Una vez que se ha calculado la señal de entrada total, se aplica una función de activación, generalmente no lineal, para obtener la salida de dicha neurona. Existen múltiples funciones de activación (la única restricción es que deben ser funciones continuamente derivables). Por ejemplo, una de las funciones de activación más empleadas

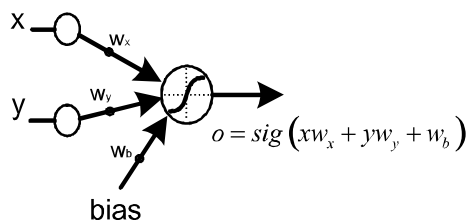


Figura 2.1: Modelo del Perceptrón

es la función sigmoide. En la figura 2.2 se muestran algunas de las funciones de activación más comunes. Estas pueden ser diferenciables y no diferenciables, como la función sigmoide y escalón unitario respectivamente; además de positivas y de promedio cero, como la función gaussiana y la función tangente hiperbólica respectivamente.

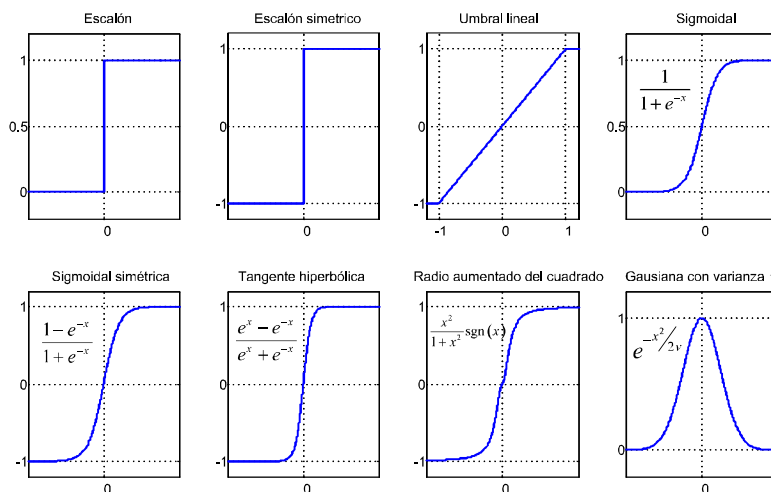


Figura 2.2: Funciones de activación más comunes

2.1.1 Tolerancia a fallas

Las redes neuronales son elementos con capacidad inherente de tolerancia a fallas. A diferencia de los sistemas computacionales tradicionales, los cuales pierden su funcionalidad en cuanto

sufren un pequeño error de memoria ya que cada pieza de información de datos se encuentran almacenados en un espacio único, localizado y direccionable. En las redes neuronales, si se produce una falla en un pequeño número de neuronas, el comportamiento del sistema no sufre una caída súbita, aunque se vea influenciado. Es decir, pueden seguir realizando su función (con cierta degradación) aunque se destruya parte de la red.

Otro aspectos respecto a la tolerancia a fallas es que las redes pueden aprender a reconocer patrones con ruido, distorsionados o incompletos, ésta es una tolerancia a fallos respecto a los datos.

La razón por la que las redes neuronales tienen estas características es que tienen su información distribuida, y no localizada, en las conexiones entre neuronas. La mayoría de las interconexiones entre los nodos de la red tendrán unos valores en función de los estímulos recibidos, y se generará un patrón de salida que represente la información almacenada, existiendo cierto grado de redundancia en este tipo de almacenamiento.

2.1.2 Cualidades del Perceptrón

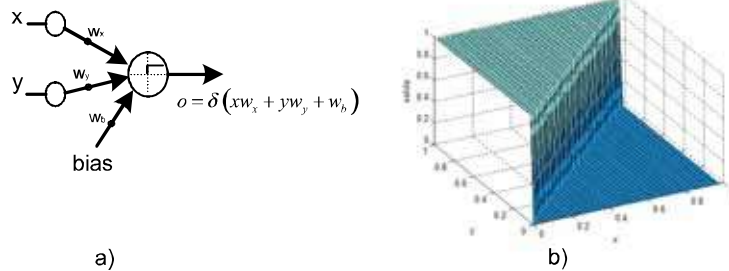


Figura 2.3: Separación lineal del espacio de entrada (b) de un Perceptrón (a)

Para una red tipo Perceptrón con dos entradas, un bias y función *hardlimit* (figura 2.3a), se puede establecer dos regiones separadas por una frontera lineal en el espacio de entrada de los patrones (figura 2.3b), previo entrenamiento de sus pesos. Por tanto, puede resolver problemas de la función OR y AND; no siendo así para problemas que no son linealmente

separables, como el problema de la función XOR. Como se puede ver en la figura 2.4.

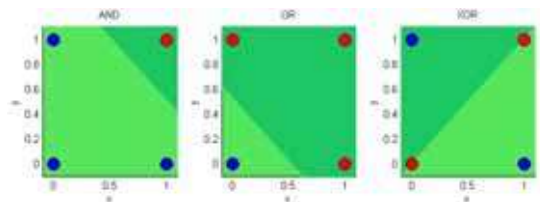


Figura 2.4: Clasificación de los patrones en las funciones AND, OR y XOR

La clasificación de patrones en la función XOR, se puede resolver usando dos Perceptrones para generar dos rectas ubicadas de manera que delimiten el espacio de entrada en tres regiones de las cuales la intersección, que incluye los datos a clasificar, pertenece a una clase y las dos regiones restantes a otra. Para tener la intersección en un único nodo de salida, es necesario contar con otra capa con una neurona cuyas entradas serán las salidas de las neuronas anteriores.

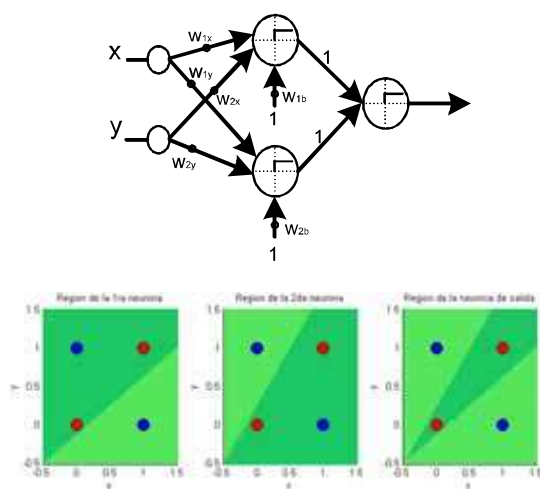


Figura 2.5: Solución al problema de clasificación en la función XOR

Cuando las salidas de las neuronas en la segunda capa y sus pesos de conexión con una neurona

de la capa de salida son todos iguales a 1, la salida de esta última se activará únicamente cuando todas las neuronas de la segunda capa sean activas, semejante a la operación AND. Con dos capas, la region resultante se forma por la intersección de regiones compuestas por cada neurona de la segunda capa, ya que cada una de ellas se comporta como un Perceptrón simple. De esta manera se encuentra una solución al problema de la función XOR.

Es fácil observar que para clasificar un punto, o un pequeño grupo de ellos, en el espacio de entrada es necesario un número mínimo de Perceptrones para distinguir dicho punto del resto. Por ejemplo, para el espacio generado por 2 entradas es necesario como mínimo 3 Perceptrones para restringir una pequeña zona del resto, como se muestra en la figura 2.6. Empleando en paralelo estos conjuntos excluyentes, se puede construir un mapeo mas complejo como el mostrado es la figura 2.7.

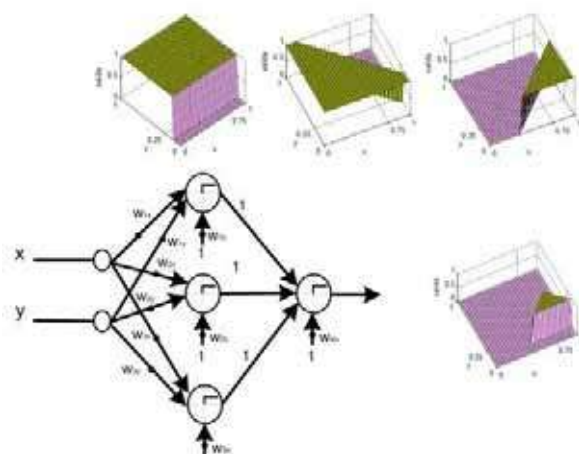


Figura 2.6: Delimitación de una región en un espacio de entrada con tres Perceptrones

2.1.3 Aproximación de funciones

La capacidad de aproximación universal de las redes neuronales ha sido estudiada por varios investigadores incluyendo Cybenko (1989), Hornik (1989), y Sandberg (1991). El resultado básico de la capacidad de aproximación menciona que cualquier función suave $f(x)$ puede

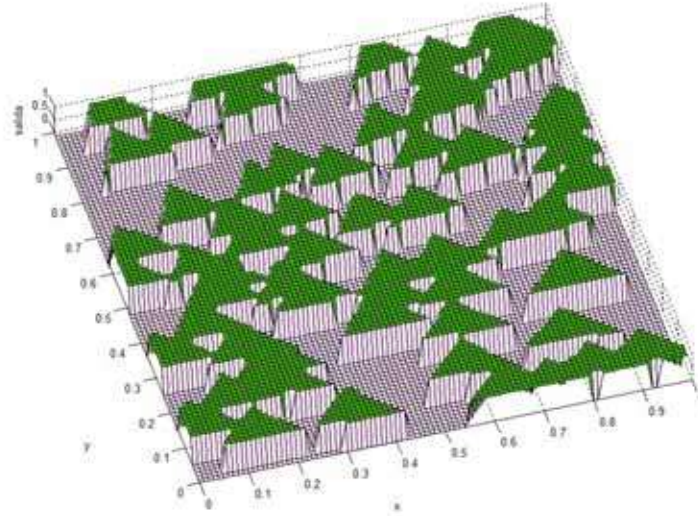


Figura 2.7: Mapeo de múltiples Perceptrones

ser aproximada con precisión arbitraria dentro de un conjunto compacto usando una red neuronal de dos capas y pesos adecuados. Este resultado ha sido mostrado usando funciones de activación *sigmoidal*, *hardlimit* y otras. Específicamente, si $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m$ es una función suave, entonces se tiene un conjunto compacto $S \in \mathbb{R}^n$ y un número ϵ , en el que existe una red neuronal tal que

$$f(x) = W^T \sigma(V^T x) + \epsilon$$

para todo $x \in S$, con un número L suficientemente grande de neuronas en la capa oculta. El valor de ϵ es llamado error de aproximación de la función, y decrementa cuando el número L de neuronas en la capa oculta incrementa. Es decir, si se requiere incrementar el conjunto compacto S , generalmente L deberá incrementarse correspondientemente [102].

A pesar de que los resultados de aproximación han sido mostrados para funciones suaves con un número finito de discontinuidades, debe notarse que el problema de determinar la cantidad de neuronas en la capa oculta para una adecuada precisión no ha sido resuelta, sin embargo existen criterios generales, cuando se usan cierto tipo de funciones de activación (como las redes de base radial), que denotan un máximo de particiones uniformes que permiten la

aproximación con una precisión deseada. Además, la selección de los pesos en la red no está limitado a un procedimiento específico, ya que existen varios métodos para encontrarlos.

2.2 Tipos de Redes

La forma como se interconectan las neuronas de una red neuronal determina su estructura y sus conexiones son ajustadas o entrenadas para lograr un comportamiento deseado. Es decir, las redes pueden ser clasificadas de acuerdo a sus estructuras o algoritmos de aprendizaje [99]. Un esquema general de esta clasificación se muestra en la figura 2.8.

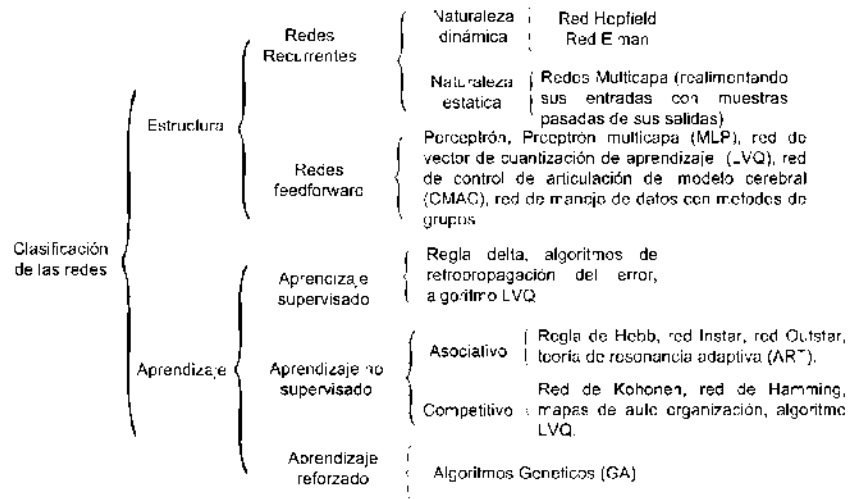


Figura 2.8: Clasificación general de las redes neuronales.

2.2.1 Estructura

En términos de su estructura, las RNA pueden ser divididas en dos tipos: redes neuronales unidireccionales (Feedforward neural networks) y redes neuronales recurrentes (Recurrent Neural Networks).

Redes tipo forward (FFNN's, Feedforward neural networks)

Las neuronas son normalmente agrupadas en capas. Las señales fluyen desde la capa de entrada hasta la capa de salida vía conexiones unidireccionales, las neuronas son conectadas de una capa a la siguiente, pero no dentro de la misma capa. Normalmente realizan un mapeo estático entre un espacio de entrada y un espacio de salida. La salida es una función sólo de la entrada en ese instante. Como ejemplo se tienen: Perceptrón multicapa (Multi-Layer Perceptron, MLP), red de vector de cuantización de aprendizaje (Learning Vector Quantitative, LVQ), red de control de articulación del modelo cerebral (Cerebral Model Articulation Control, CMAC) y red de manejo de datos con método de grupos (Group Method Data Handling, GMDH).

Redes tipo Recurrente (RNN's, Recurrent neural networks)

Las salidas de algunas neuronas son retroalimentadas a las mismas neuronas o a neuronas en capas precedentes. Así, las señales pueden fluir tanto en direcciones hacia adelante como hacia atrás. Tienen una memoria dinámica ya que sus salidas en un instante dado reflejan las entradas actuales, así como las entradas y salidas previas. Por ejemplo: Red de Hopfield, Red Elman, Red Jordan y Redes dinámicas multicapa.

2.2.2 Algoritmos de aprendizaje

Las RNA son entrenadas por dos tipos principales de algoritmos de aprendizaje: algoritmos de aprendizaje supervisado y no supervisado. Además, existe un tercer tipo, aprendizaje reforzado, que puede ser considerado como una forma especial del aprendizaje supervisado.

Aprendizaje supervisado

Ajusta los pesos de las neuronas de acuerdo a la diferencia entre las salidas deseada y actual de la red correspondiente a una entrada dada. Por ejemplo: regla delta [109], la regla delta generalizada o algoritmo de retropropagación [110] y el algoritmo LVQ [111].

Aprendizaje no supervisado

Sólo los patrones de entrada son presentados a la red y automáticamente adapta los pesos de sus conexiones para agrupar los patrones de entrada dentro de grupos con características similares. Por ejemplo: algoritmo de Kohonen, teoría de resonancia adaptiva de Carpenter y Grossberg y los algoritmos de aprendizaje competitivo.

Aprendizaje reforzado

Es un caso especial del aprendizaje supervisado. En lugar de utilizar un maestro, se emplea un crítico sólo para evaluar la calidad de la salida de la RNA correspondiente a una entrada dada. Por ejemplo los algoritmos genéticos (GA).

2.3 Redes neuronales para modelado

2.3.1 Incertidumbre paramétrica y estructural

Los controladores que enfrentan la falta de información cuando alguna parte o toda la dinámica del sistema no son conocidas son un tema importante. Pues, un sistema que no ha sido hecho a la medida siempre tiene incertidumbre estructural o paramétrica [5].

- **Incertidumbre paramétrica:** con el paso del tiempo, las condiciones de operación o climáticas y otros factores del entorno de la planta pueden ocasionar desgaste o degradación de sus componentes, por lo que sus parámetros pueden cambiar.
- **Incertidumbre estructural:** es posible que en el modelo que se este construyendo no se hayan tomado en cuenta por alguna razón ciertas dinámicas de la planta, en este caso la estructura real de la planta tendrá diferencia con respecto a la estructura del modelo matemático.

Resultados recientes muestran que la técnica de redes neuronales es muy efectiva para identificar una gran cantidad de sistemas no lineales complejos cuando no se tiene información completa del modelo.

2.3.2 Identificación de sistemas dinámicos

A menudo la principal, y a veces única, información de que dispone un proceso es una serie numérica temporal. Dicha serie corresponde con los valores de una función, de una o más variables, que contiene la dinámica interna del sistema. El problema radica entonces en la identificación de la función generadora de la serie de valores, esta tarea se ha realizado ampliamente con las redes neuronales [1].

La tarea de la identificación de sistemas consiste esencialmente en encontrar mapeos convenientes que puedan aproximar los mapeos implicados en un sistema dinámico. Un sistema dinámico puede ser descrito por dos tipos de modelos: modelo de entrada salida y modelo estado-espacio [99].

Modelo de Entrada-Salida

Un modelo de entrada-salida describe un sistema dinámico basado en datos de entrada salida. En el dominio discreto, un modelo de entrada-salida puede ser del tipo ARX (AutoRegressive, eXternal input) [23].

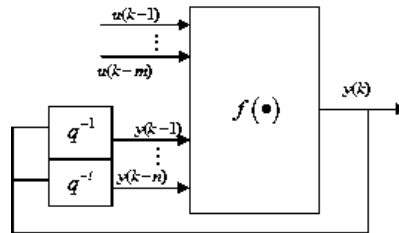


Figura 2.9: Estructura ARX (AutoRegressive, eXternal input).

Un modelo de entrada-salida asume que la salida nueva del sistema se puede predecir mediante las entradas y salidas pasadas del sistema. Si un sistema es además supuesto a ser determinístico, invariante en el tiempo, simple-entrada-simple-salida (SISO), el modelo puede representarse por 2.1.

$$y(k) = f[y(k-1), y(k-2), \dots, y(k-n), u(k-1), u(k-2), \dots, u(k-m)] \quad (2.1)$$

donde m y n son la cantidad de retrasos en las señales de entrada y salida respectivamente. La función f puede ser una función estática que mapea las entradas y salidas pasadas a una nueva salida. Así, desde este enfoque el problema de identificación consiste en una única tarea: encontrar f . Si el sistema es lineal, f es una función lineal y la ecuación 2.1 puede ser rescrita como

$$y(k) = a_1y(k-1), a_2y(k-2), \dots, a_ny(k-n), b_1u(k-1), b_2u(k-2), \dots, b_mu(k-m) \quad (2.2)$$

Modelo de Estado-Espacio

Un sistema dinámico puede ser descrito también por un modelo de estado-espacio. El modelo de estado-espacio de un sistema dinámico de orden n , no-lineal, invariante en el tiempo, multi-entrada-multi-salida (MIMO) es como sigue

$$\begin{aligned} x(k+1) &= \Phi[x(k), u(k)] \\ y(k) &= \Psi[x(k)] \end{aligned} \quad (2.3)$$

donde $x(k) = [x_1(k), x_2(k), \dots, x_n(k)]^T$ es el vector de estados de n componentes del sistema, $u(k) = [u_1(k), u_2(k), \dots, u_r(k)]^T$ es el vector de entrada del sistema, y $y(k) = [y_1(k), y_2(k), \dots, y_m(k)]^T$ es el vector de salida del sistema. Φ y Ψ son mapeos no-lineales. La función $\Phi(\cdot)$ mapea el estado $x(k)$ y la entrada $u(k)$ dentro del nuevo estado $x(k+1)$. La función $\Psi(\cdot)$ es un mapeo que transforma el estado $x(k)$ dentro de la salida $y(k)$.

En la identificación de un sistema pueden conocerse previamente el orden del sistema y desconocer de forma precisa sus parámetros o desconocer totalmente el orden del sistema y sus parámetros. La identificación de un sistema mediante redes neuronales no pretende encontrar las funciones Φ y Ψ , sino un mapa de la relación entre un espacio de entrada y un espacio de salida con base a los patrones con que fue entrenada.

Si el sistema es lineal, la ecuación 2.3 se convierte en

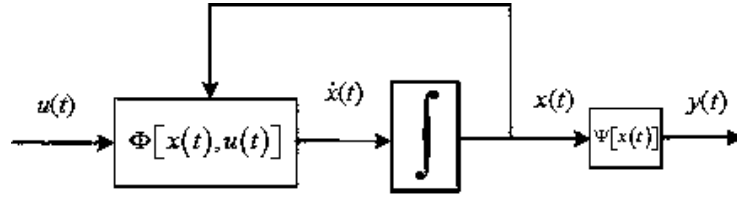


Figura 2.10: Esquema del Modelo de identificación Espacio-Estado

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \end{aligned} \tag{2.4}$$

donde las matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times r}$ y $C \in \mathbb{R}^{m \times n}$.

Estabilidad de Lyapunov

Un punto de equilibrio se dice estable si todas las soluciones que se inicien en las cercanías del punto de equilibrio permanecen en las cercanías del mismo, de otro modo el punto de equilibrio es inestable. Un punto de equilibrio se dice *asintóticamente estable* si todas las soluciones que inicien en las cercanías del punto de equilibrio no sólo permanecen en las cercanías del punto de equilibrio, sino que además tienden hacia el equilibrio a medida que el tiempo se aproxima a infinito [100].

Teorema 1 (*Método Directo*): Sea el origen un punto de equilibrio del sistema no lineal $\dot{x} = f(x)$ donde $f : D \subset \mathbb{R}^n$ es una función continuamente diferenciable y $D \subset \mathbb{R}^n$ es el entorno del origen. Sea

$$A = \left. \frac{\partial f}{\partial x} \right|_{x=0}$$

Entonces, el origen es asintóticamente estable si todos los valores propios de A tienen parte real negativa. El origen es inestable si uno o más valores propios de A tienen parte real positiva.

2.3.3 Estructuras de modelos no lineales basados en redes neuronales

Una variedad de arquitecturas de redes neuronales para control han sido propuestas para el control de sistemas no lineales con incertidumbre. Una de las principales ventajas de las redes neuronales en el diseño de control no lineal es precisamente la capacidad de construir mapeos no lineales complejos. Por esto, las metodologías con redes neuronales se pueden considerar para identificar no linealidades desconocidas en un sistema y después utilizar este modelo identificado para construir una ley de control que logre los propósitos de control deseados [106].

Como se apunta en [108] los modelos clásicos de identificación lineal FIR (Finite Impulse Response), ARX (AutoRegressive, eXternal input), ARMAX (AutoRegressive, Moving Average, eXternal input), OE (Output Error), SSIF (State Space Innovations Form) se pueden expresar en forma de RNA, estos son la contraparte no lineal y son obtenidos por

$$y(t) = g[\varphi(t, \theta), \theta] + e(t) \quad (2.5)$$

o en la forma estimador

$$\hat{y}(t|\theta) = g[\varphi(t, \theta), \theta] \quad (2.6)$$

donde $\varphi(t, \theta)$ es el vector de regresión, el cual contiene entradas pasadas, salidas pasadas, o señales derivadas de las entradas y salidas. Mientras que θ es el vector que contiene los parámetros ajustables conocidos como *pesos* de la red, y g es la función generada por la red. Dependiendo del vector de regresión elegido aparecen diferentes estructuras de modelos. Si el vector regresión es seleccionado como el modelo ARX, la estructura de modelo es llamada NNARX por el acrónimo de *Neural Network ARX*. Asimismo, se reconocen las estructuras NNFIR, NNARMAX, NNOE y NNSSIF. La figura 2.11 muestra estas estructuras equivalentes en redes neuronales artificiales (ANN). m y n son la cantidad de retrasos en las señales de entrada y salida respectivamente y para su implementación es necesario conocer el orden exacto del modelo para asignar así la cantidad de retardos. Sin embargo, en casos prácticos esta información no siempre es posible conocerla y el número de retardos tiene que asignarse experimentalmente como se menciona en [107].

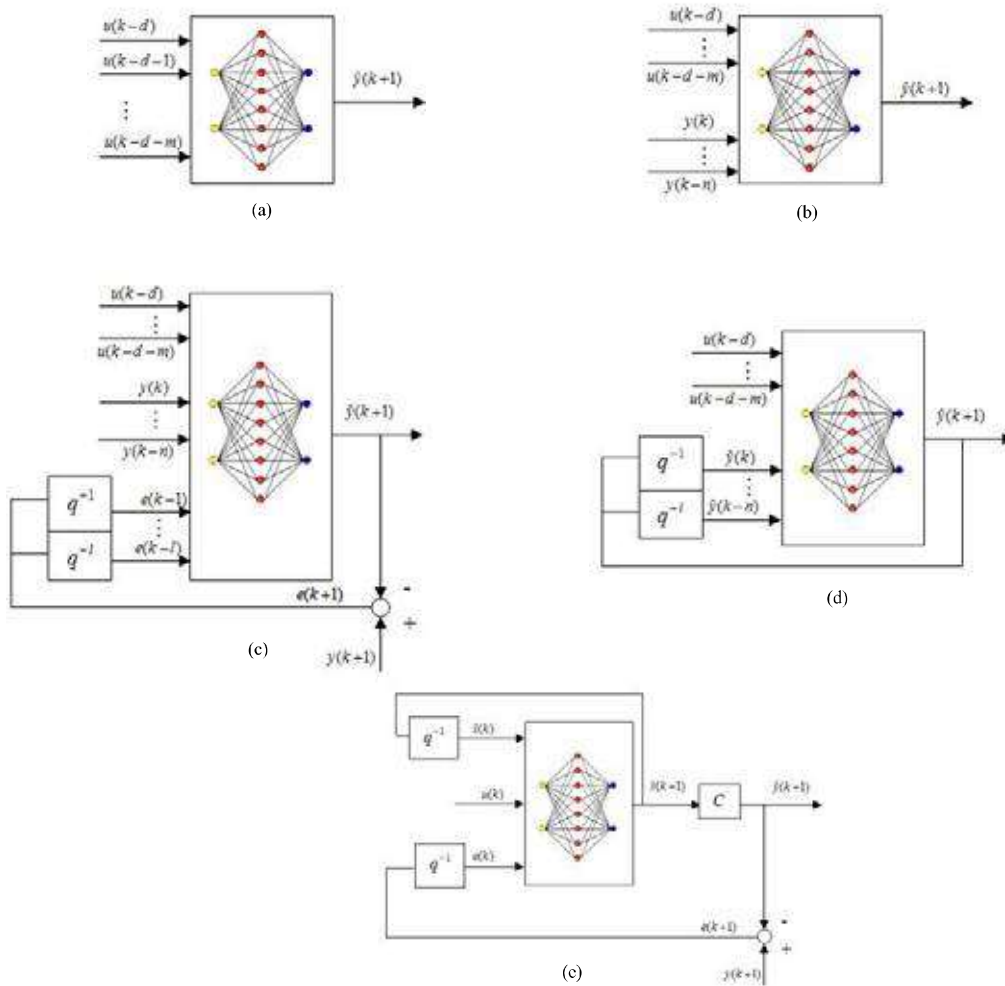


Figura 2.11: Estructuras; (a) NNFIR, (b) NNARX, (c) NNARMAX, (d) NNOE, (e) NNSSIF

Por otro lado, los predictores neuronales son siempre estables debido a que generalmente se incluye funciones de activación acotadas asintóticamente. Esto en particular es importante en el caso no lineal ya que las cuestiones de estabilidad son mucho más complejas que en los sistemas lineales. La ausencia de los problemas relacionados con la estabilidad hace que los modelos NNFIR y NNARX, en particular la estructura NNARX, sean elegidas comúnmente cuando el sistema es determinístico o el nivel de ruido es mínimo. De acuerdo a la configuración de un sistema de identificación, existen dos estructuras de identificación: paralelo y serie-paralelo.

Identificación en paralelo

En la identificación en paralelo, el sistema y la red son dos procesos independientes que comparten las mismas entradas externas. La red y el sistema reciben las mismas entradas externas, pero las salidas del sistema no son utilizadas como entradas a la red. Es decir, las salidas estimadas $\hat{y}(k+1)$ son utilizadas como argumento de entrada al modelo de la red neuronal.

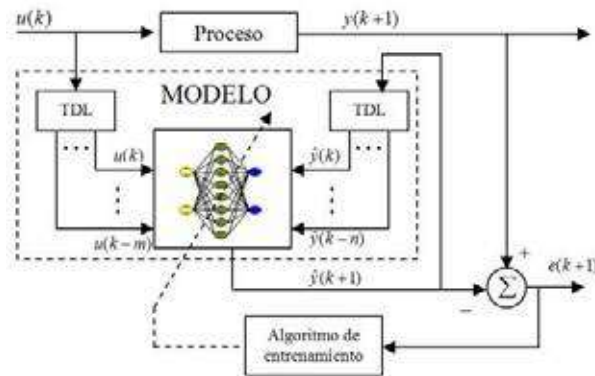


Figura 2.12: Identificación en paralelo

La ecuación 2.7 expresa la salida de la red como una función, generalmente no lineal, de las salidas estimadas anteriores y de la entrada como se muestra en la figura 2.12.

$$\hat{y}(k+1) = f(\hat{y}(k), \hat{y}(k-1), \dots, \hat{y}(k-n), u(k), u(k-1), \dots, u(k-m)) \quad (2.7)$$

Debido a la repetición (recurrencia) de $\hat{y}(k+1)$ en la ecuación 2.7, esta configuración corresponde a una red neuronal recurrente.

Identificación en serie-paralelo

En la identificación serie-paralelo, nuevamente la red y el sistema reciben las mismas entradas externas, pero ahora la salida del sistema es parte de las entradas a la red (ver figura 2.13), es decir, son utilizadas como los argumentos de entrada al modelo de la red neuronal.

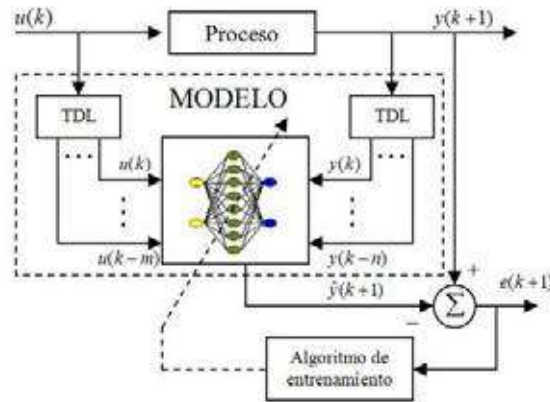


Figura 2.13: Identificación en serie-paralelo

La ecuación que define la salida es:

$$\hat{y}(k+1) = f(y(k), y(k-1), \dots, y(k-n), u(k), u(k-1), \dots, u(k-m)) \quad (2.8)$$

Debido a que no hay repetición (recurrencia) en la ecuación 2.8, el esquema corresponde a una red neuronal feedforward.

2.3.4 Redes dinámicas multicapa

Las redes multicapa son de naturaleza estática, es decir, su salida no evoluciona con el tiempo (para un patrón de entrada existe una salida asociada), pero pueden adquirir un comportamiento dinámico (para un patrón de entrada la salida posee un estado transitorio y converge a un valor en el estado estacionario) realimentando sus entradas con estados anteriores de sus salidas. Como se puede ver en la figura 2.14, la red esta compuesta de una capa estática la cual generalmente posee un número de neuronas superior al número de variables de estado del sistema a identificar, la salida de la capa estática va a un sumador donde se le resta el valor anterior de la variable de estado Z_i identificada por el sistema, de esta operación se genera la derivada de cada una de las i variables de estado identificadas por el sistema.

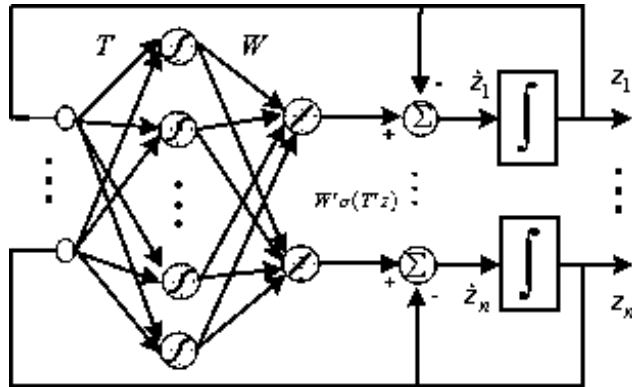


Figura 2.14: Red Dinámica multicapa

La red recurrente dinámica multicapa cuyo comportamiento lo describe la ecuación 2.9 puede identificar el comportamiento de un sistema autónomo ($u = 0$) descrito por la ecuación 2.10.

$$\frac{\partial}{\partial t} z = \bar{f}(z) = Az + W' \sigma(T'z) \quad (2.9)$$

$$\frac{\partial}{\partial t} x = f(x) = Ax + f_0(x) \quad (2.10)$$

donde $x, z \in \mathfrak{R}^n$, $A \in \mathfrak{R}^{n \times n}$, $f(x) : \mathfrak{R}^n \rightarrow \mathfrak{R}^n$, $\bar{f}(x) : \mathfrak{R}^n \rightarrow \mathfrak{R}^n$, $f_0(x) : \mathfrak{R}^n \rightarrow \mathfrak{R}^n$, $W \in \mathfrak{R}^{n \times N}$, $T \in \mathfrak{R}^{N \times n}$, $\sigma(z) = [\sigma(z_1), \sigma(z_2), \dots, \sigma(z_n)]$ y función de transferencia $\sigma(\theta) = \tan sig(\theta)$, n es el número de variables de estado del sistema y N el número de neuronas en la capa oculta. Según [13], sin pérdida de generalidad, si el origen se asume como punto de equilibrio, el sistema 2.10 será identificado con la red 2.9 alrededor de su región de atracción y se garantiza que el error en la aproximación $e(t)$ es limitado.

La etapa estática que hace parte de la red multicapa dinámica recurrente generalmente es entrenada con algún de propagación inversa (Backpropagation), ya que son más eficientes y rápidos. Los patrones de entrenamiento de la capa estática de la figura 2.14 son diferentes combinaciones de valores de las variables de estado y los patrones objetivo están dados por la suma de cada variable de estado con su correspondiente derivada como se muestra en la figura 2.15.

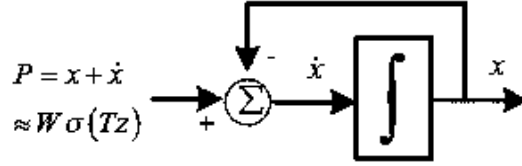


Figura 2.15: Patrones de entrenamiento de la red

Así, el error generado para el entrenamiento será:

$$e(t) = P - W' \sigma(T' z)$$

La red después de entrenada tiene la estructura de la ecuación 2.11.

$$\frac{\partial}{\partial t} \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} = \begin{bmatrix} -z_1 \\ -z_2 \\ \vdots \\ -z_n \end{bmatrix} + \begin{bmatrix} W_{11} & W_{12} & \cdots & W_{1n} \\ W_{21} & W_{22} & \cdots & W_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ W_{N1} & W_{N2} & \cdots & W_{Nn} \end{bmatrix}^T \begin{bmatrix} (T_{11}z_1 + T_{12}z_2 + \cdots + T_{1n}z_n) \\ (T_{21}z_1 + T_{22}z_2 + \cdots + T_{2n}z_n) \\ \vdots \\ (T_{N1}z_1 + T_{N2}z_2 + \cdots + T_{Nn}z_n) \end{bmatrix} \quad (2.11)$$

Para garantizar que la red ha identificado la dinámica del sistema, el Jacobiano de la red en el origen 2.12 debe tener valores propios muy cercanos a los del sistema que ha sido aproximado.

$$J_m = -I_n + W'T' \quad (2.12)$$

Ejemplo 2 Para el sistema de la ecuación 2.13, una red neuronal identificara su comportamiento autónomo alrededor de un punto de equilibrio y se verificara la similitud entre ambos vía eigenvalores en el origen como punto de equilibrio.

$$\begin{aligned} \dot{x} &= (x - y)(y^2 + x^2 - 1) \\ \dot{y} &= (x + y)(y^2 + x^2 - 1) \end{aligned} \quad (2.13)$$

Haciendo $\dot{x} = \dot{y} = 0$, y con $y = x = 0$ el punto de equilibrio es $(0,0)$. Para los eigenvalores es necesario evaluar el jacobiano en el punto de equilibrio

$$A = \begin{bmatrix} y^2 + x^2 - 1 + 2(x - y)x & -y^2 - x^2 + 1 + 2(x - y)y \\ y^2 + x^2 - 1 + 2(x + y)x & y^2 + x^2 - 1 + 2(x - y)y \end{bmatrix} \Bigg|_{\substack{x=0 \\ y=0}} = \begin{bmatrix} -1 & 1 \\ -1 & -1 \end{bmatrix}$$

y obtener el

$$\det(sI - A) = \begin{bmatrix} s + 1 & -1 \\ 1 & s + 1 \end{bmatrix} = (s^2 + 1 - i)(s^2 + 1 + i)$$

Los eigenvalores tienen parte real negativa, así que es un foco estable. Para la identificación con una red neuronal recurrente se propone una red de 2 nodos de entrada para los estados, 4 nodos en la capa oculta y sus 2 respectivas salidas. La figura 2.17(a) muestra la evolución del sistema de ecuaciones (línea continua) y la estimada de la red (línea punteada) para las condiciones iniciales C.I. $(x, y) = (-0.5, 0.5)$. Dicha condición inicial, corresponde a una de las trayectorias del diagrama de fase de la figura 2.16. Con un coeficiente de aprendizaje de 0.001 y después de una iteración la red tiene el comportamiento mostrado en la figura 2.17(b). Para 5 y 14 iteraciones se muestran en las figuras 2.17(c) y 2.17(d) respectivamente. Después de 600 iteraciones, el error medio cuadrático desciende por debajo de 1×10^{-5} , como se muestra en la figura 2.18. Finalmente, el conjunto de pesos de la capa oculta y la capa de salida quedan

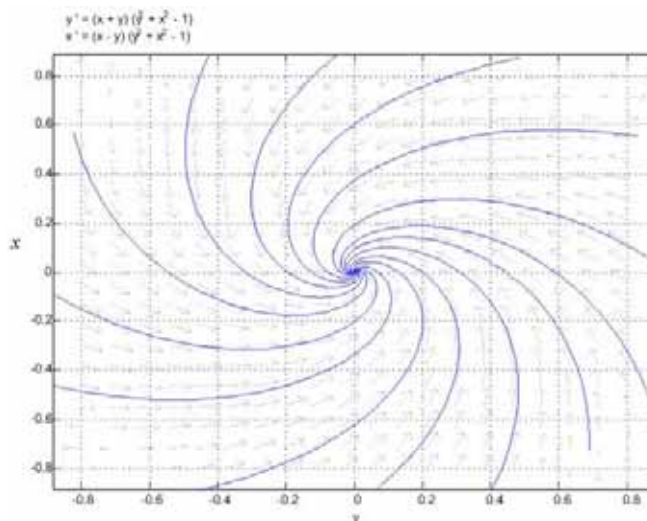


Figura 2.16: Diagrama de fase del sistema descrito por las ecuaciones 2.13.

como:

$$T' = \begin{bmatrix} 1.193544 & 0.19517461 & -0.498339 & -0.6378439 \\ 0.403326 & 0.4307488 & 0.875252 & 0.6924773 \end{bmatrix}$$

$$W' = \begin{bmatrix} 0.7759846 & -0.2509433 \\ -0.56269885 & -1.100868262 \\ 0.2300762454 & 0.65037617 \\ 0.68568607 & 0.20634516 \end{bmatrix}$$

Así el jacobiano de dicha configuración es obtenido de la ecuación 2.12 y queda como:

$$J_{Red} = \begin{bmatrix} -0.7356592 & -0.970098005 \\ 0.74679001 & -0.86327678 \end{bmatrix}$$

y sus eigenvalores

$$\lambda_1 = -0.79947304 + 0.84875711i$$

$$\lambda_2 = -0.79947304 - 0.84875711i$$

similares a los obtenidos para el sistema original. En la figura 2.19 se puede ver la respuesta para cuatro condiciones iniciales y se verifica su dinámica alrededor del origen.

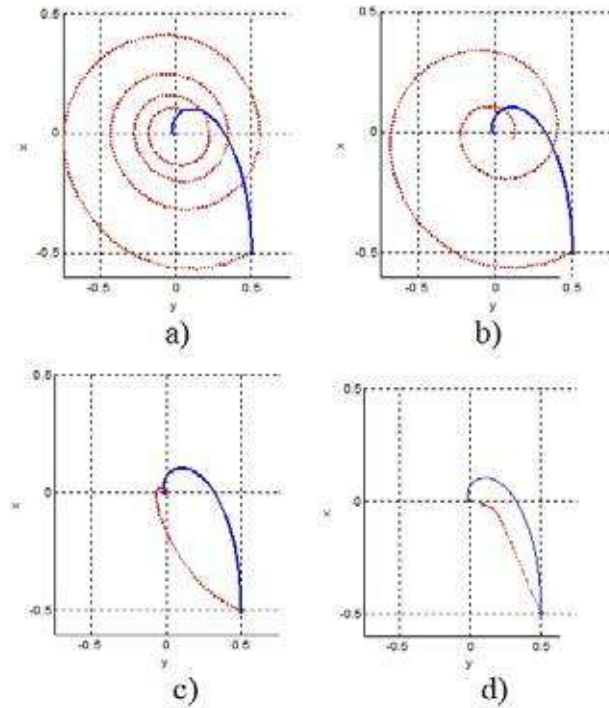


Figura 2.17: Identificación de un sistema de ecuaciones

2.4 Control robusto, control adaptivo y control inteligente

Como se menciona en [112], uno de los propósitos primarios del control con retroalimentación clásico es el de incrementar la robustez en un sistema de control. El control clásico provee robustez sobre un rango de incertidumbre relativamente pequeño. Tradicionalmente en los sistemas de control, que operan con incertidumbre, típicamente dependen de la intervención humana, en alguna parte del diseño, para funcionar apropiadamente. Sin embargo, en algunas aplicaciones en tiempo real la intervención humana es inaceptable y se deben aplicar técnicas para manejar la incertidumbre. Por esta razón, las técnicas de control adaptivo fueron desarrolladas para sistemas que requieren mejor desempeño sobre largos rangos de incertidumbre debidas a grandes variaciones en valores de parámetros, condiciones ambientales y señales de entrada. Generalmente incorporan un segundo lazo de retroalimentación con la capacidad

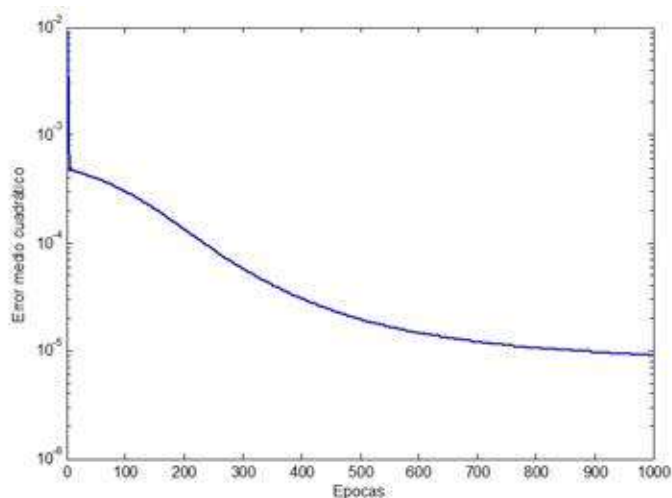


Figura 2.18: Error medio-cuadrático en la identificación.

de rastrear los parámetros del sistema (ver figura 2.20), las condiciones ambientales y las características de la entrada [112]. Es decir puede variar parámetros en los elementos de compensación del interior del lazo y mantener aceptables características de desempeño.

El diseño de un sistema de control inteligente es similar al del control adaptivo. Sin embargo, hay diferencias; por ejemplo, para el control inteligente el rango de incertidumbre puede ser substancialmente mayor que la que puede ser tolerada por los sistemas adaptivos. Uno de los enfoques para los sistemas de control inteligente usa redes neuronales. Empleando las RNA el problema de control puede ser considerado como un problema de reconocimiento de patrones, donde los patrones a ser reconocidos son los cambios de las señales mapeadas dentro de las señales de acción para un desempeño específico del sistema [112].

2.5 Estructura de un sistema de detección y aislamiento de fallas (FDI)

Un sistema que tiene la oportunidad de detectar, aislar e identificar fallas es llamado sistema de diagnóstico de fallas [107]. La idea básica es generar señales, conocidas como residuos,

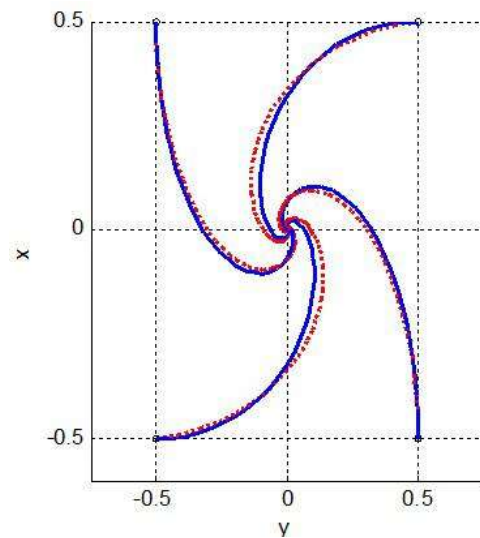


Figura 2.19: Dinámica alrededor del origen

que reflejen inconsistencias entre las condiciones de operación nominal y con falla. Usualmente son usados métodos analíticos basados en Observadores, Estimación de parámetros o Ecuaciones de paridad. Desafortunadamente una desventaja de estos es la precisión del modelo matemático. Una solución alternativa es empleando inteligencia artificial. Puede ser aplicada efectivamente tanto al modelado del proceso como a la toma de decisiones en su comportamiento.

El objetivo de un sistema de diagnóstico de fallas es determinar la localización y el momento de ocurrencia de las posibles fallas basadas en datos accesibles o conocimiento del comportamiento del proceso, empleando su modelo matemático o cuantitativo. El esquema basado en el modelo consiste de dos partes: la generación y evaluación de residuos [1], como se puede ver en la figura 2.21. Pueden ser vistos como un proceso secuencial de extracción de síntomas y de diagnóstico actual.

El proceso de extracción de residuos está basado en la comparación entre las mediciones y las salidas del sistema predictor. En condiciones normales el residuo es cercano a cero y en presencia de fallas, debe aparecer una desviación de cero. El módulo de evaluación del residuo

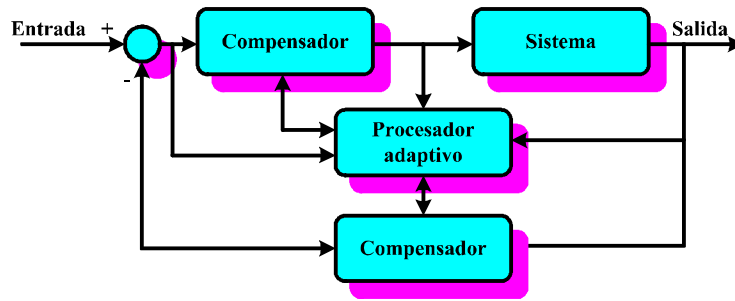


Figura 2.20: Sistema de control adaptivo

se dedica al análisis de la señal residual, que en el caso más sencillo se basa en la comparación lógica con un umbral, para determinar si ha ocurrido una falla o no, y aislar la falla en un dispositivo en particular del sistema.

2.5.1 Generador y evaluación de residuos neuronal

Para la generación de residuos, la red neuronal reemplaza el modelo analítico que describe el proceso bajo condiciones normales de operación. La red tiene que ser entrenada para esta tarea. Los datos de entrenamiento pueden ser colectados directamente del proceso o de un modelo simulado. También es necesaria la adquisición de los datos en diferentes situaciones de falla. Estos datos generalmente no están disponibles en el proceso real. El entrenamiento puede hacerse en o fuera de línea, dependiendo de la disponibilidad de los datos.

El entrenamiento en línea es muy atractivo en especial cuando se adecua una red a ambientes no constantes o sistemas no estacionarios. Después de finalizar el entrenamiento, la red neuronal está lista para la generación de residuos en línea. En la figura 2.22 se muestra un esquema diagnóstico empleando un conjunto de modelos neuronales, previamente entrenados, como generadores de residuos similar al descrito con observadores difusos dedicados en [12]. Cada modelo neural representa una clase de comportamiento del proceso. Un modelo representa al sistema funcionando en condiciones normales. Cada uno de los modelos sucesivos representan una situación de falla, respectivamente [107]. En ausencia de fallas, los residuos son debidos a ruido y disturbios no modelados y el *modelo 0* tendrá la caracterís-

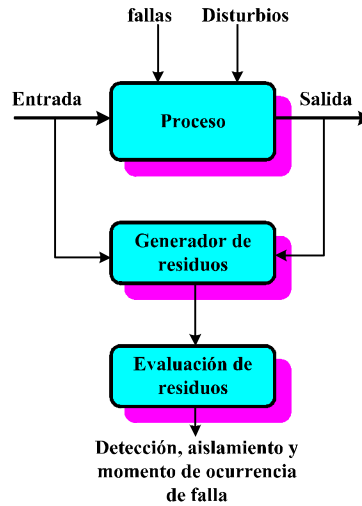


Figura 2.21: Esquema general de diagnóstico de fallas basado en el modelo

tica de no activar su correspondiente residuo. En presencia de fallas, existirá una desviación característica en los residuos.

Después de que el residuo ha sido generado, un mecanismo de toma de decisión deberá evaluar los residuos para determinar la ocurrencia y localización de las fallas. Tradicionalmente, es implementado un método de prueba lógica con un umbral fijo, adaptivo o estadístico. La tarea principal de este mecanismo de decisión es la de clasificar los residuos dentro de un número distinguido de patrones correspondientes a distintos tipos de panoramas de fallas. Por lo tanto, esta etapa también puede ser basada en los principios de reconocimiento de patrones cuya función es la de mapear un vector de características de entrada dentro de varias clases de decisiones. En los sistemas FDI, estas clases de decisiones son los diferentes tipos y/o localización de las fallas en el sistema. En general, las redes neuronales artificiales pueden ser aplicadas para resolver tanto el modelado, en la generación de residuos, como a la toma de decisiones (evaluación de residuos).

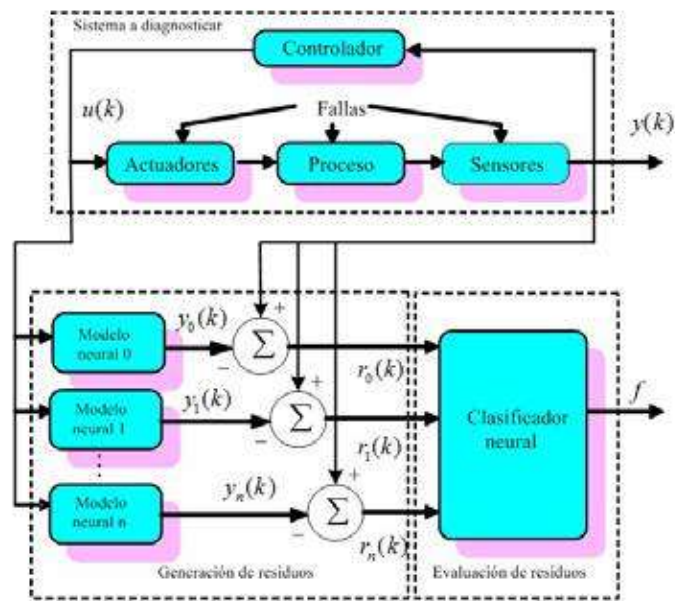


Figura 2.22: Esquema de detección de fallas neural basado en banco de modelos del proceso

Capítulo 3

Metodología de diagnóstico

Para la implementación del sistema de diagnóstico de fallas (FDI) es necesario primeramente realizar la identificación propia del sistema. Recordando que, si la identificación es basada exclusivamente en los datos medidos, el proceso de identificación es llamado modelado de caja negra (*BlackBox modelling*), si es puramente modelado físico de un sistema es llamado modelado de caja blanca (*WhiteBox modelling*) y cuando existe un cierto nivel de conocimiento del sistema y es utilizado para mejorar el modelo empírico se usa concepto de modelado de caja gris (*GrayBox modelling*), se puede ubicar fácilmente a los sistemas de identificación con redes neuronales como modelado de caja negra [107].

El problema de identificación puede ser enfocado de diferentes maneras dependiendo del nivel de comprensión previo acerca del sistema. Sin embargo, cuando intentamos identificar el modelo de un sistema dinámico es común practicar las fases la experimentación, selección de una estructura, la estimación del modelo y su validación. Como se muestra en la figura 3.1.

El propósito de la experimentación es recolectar un grupo de datos que describan como se comporta el sistema sobre diferentes rangos de operación variando las entradas y observando los impactos sobre las salidas. En la etapa de selección de la estructura de modelo, se considera la búsqueda del modelo dentro de un conjunto de modelos candidatos. Por ejemplo, las estructuras de modelos de identificación lineal FIR (Finite Impulse Response), ARX (AutoRe-

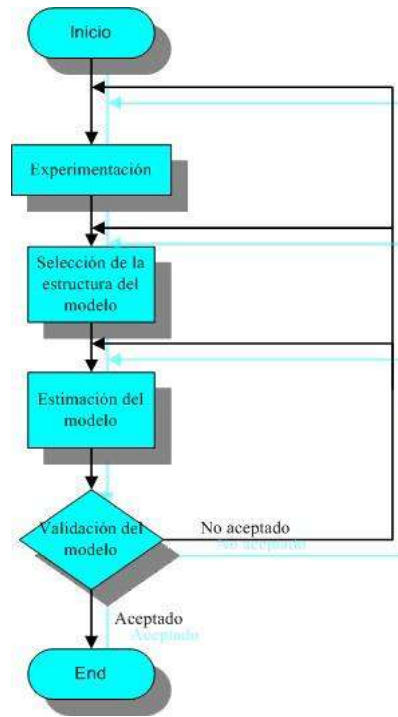


Figura 3.1: Procedimiento básico de identificación de sistemas

gressive, eXternal input), ARMAX (AutoRegressive, Moving Average, eXternal input), OE (Output Error), SSIF (State Space Innovations Form). En la etapa de estimación del modelo se elige en modelo que mejor desempeño tenga de acuerdo con ciertos criterios. La estrategia más común es elegir el modelo que provea la mejor predicción un paso adelante en términos de un error cuadrático mínimo entre las salidas del sistema y las estimadas del modelo propuesto. No obstante, el mismo proceso en redes neuronales es llamado *entrenamiento* o *aprendizaje*. La validación radica en usar el modelo estimado/entrenado e investigar si cumple con los requerimientos necesarios [108]. La línea de retroalimentación en la validación del sistema se refiere a una corrección necesaria en alguna de las etapas anteriores.

Al igual que en el diagrama de flujo de la figura 3.1, en el sistema diagnóstico propuesto cubre los aspectos descritos en la figura 3.2.

En el primer bloque, son definidas la cantidad de fallas y la cantidad de entradas y salidas

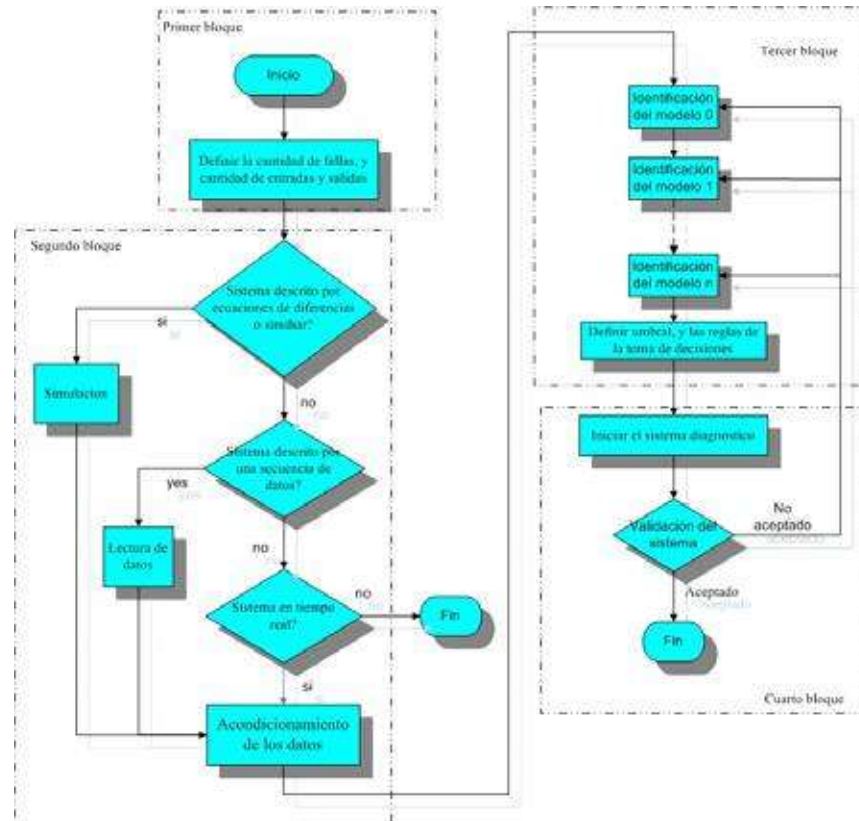


Figura 3.2: Procedimiento básico de implementación del sistema FDI neuronal

del sistema. El segundo bloque define la fuente de los datos. En el tercer bloque, los modelos son identificados dentro de cada una de sus respectivas fallas, incluyendo el modelo libre de fallas además de definir las reglas en la toma de decisiones. En el cuarto bloque, el sistema diagnóstico es activado y una etapa de validación es necesaria para investigar si el sistema diagnóstico satisface completamente con las características descadas.

3.1 Generación de residuos

En general las redes neuronales pueden ser aplicadas tanto en el modelado como en la toma de decisiones. Para la generación de residuos el modelo si falla, así como sus modelos con

falla, operan en paralelo para la generación de las señales. En la figura 3.3 se muestra nuevamente el esquema de generación de residuos empleando un conjunto de modelos, previamente entrenados para esta tarea. Cabe notar que las señales en s_1 , s_2 y s_3 son aprovechadas para la generación de los residuos, así como pueden servir de argumentos en el aislamiento de las fallas.

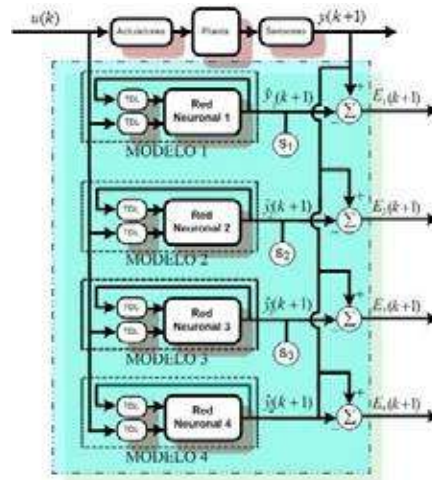


Figura 3.3: Generador de residuos neuronal

3.2 Toma de decisiones

En este trabajo la toma de decisiones consiste primeramente de una prueba en el error basada en un umbral. El umbral es fijado en 0.05, es decir 5%. Posteriormente un conjunto de reglas basadas en relaciones son incluidas. Similar en las empleadas por [117]. Las premisas, o vector de antecedentes booleanos generados por la prueba de umbral a la salida de la red, son identificadas por estas reglas. Cada primicia tiene dos funciones de membresía, R y \bar{R} , cuyas formas corresponden a; escalón unitario y escalón unitario complementario, ambos ubicados en el origen. Este tipo de membresía, implica una transición súbita en la respuesta del FDI. La figura 3.4 muestra las relaciones entre las señales de entrada (antecedentes) y las señales de salida (consecuentes). Las reglas identifican las combinaciones establecidas en base a la

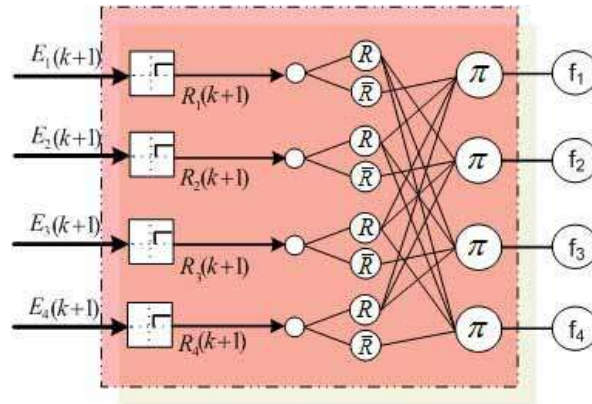


Figura 3.4: Toma de decisiones

IF	THEN
\bar{R}_1 And R_2 And R_3 And R_4	f_1 is true
R_1 And \bar{R}_2 And R_3 And R_4	f_2 is true
R_1 And R_2 And \bar{R}_3 And R_4	f_3 is true
R_1 And R_2 And R_3 And \bar{R}_4	f_4 is true

Tabla 3.1: Reglas involucradas en la toma de decisiones

operación *And* denotada por π . Su interpretación sería la descrita por la tabla 3.1. La idea del presente enfoque es realizar las modificaciones pertinentes en base al vector de antecedentes de entrada para aislar la falla.

El aislamiento de la falla puede ser interpretada como la sustitución del sistema con falla por alguno equivalente o la suma de varios que contribuyan a mantener en línea del sistema completo o la mayor parte de él. Sin embargo, para el caso del sistema de tanques, el modelo neuronal recurrente libre de falla puede ser suficiente para cubrir esta necesidad de aislamiento de la falla. Por ejemplo, la primera regla de toma de decisiones implica que el modelo libre de fallas funciona de manera correcta sin activar el residuo y los demás modelos con el residuo activado. El consecuente de esta regla es f_1 en estado activo e inactivo de lo contrario. La figura 3.5 muestra el esquema consecuente e indica que cuando f_1 esta activo,

$$f_2 = f_3 = f_4 = 0 \text{ y } g(k + 1) = y(k + 1).$$

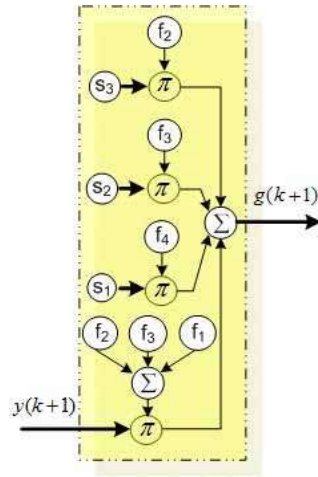


Figura 3.5: Esquema de acciones de los consecuentes

Para el consecuente de la segunda regla (f_2), el *modelo 2* con falla en el *sensor 1* no activa el residuo, es decir su comportamiento es similar al del sistema, mientras que los demás modelos si activan su correspondiente residuo y $g(k+1) = y(k+1) + s_3 = y(k+1) + \hat{y}_3(k+1)$. Es decir, una porción de información es usada para mejorar la salida final $g(k+1)$. Así consecutivamente para las demás reglas; si f_3 es activado entonces $g(k+1) = y(k+1) + s_2 = y(k+1) + \hat{y}_2(k+1)$ y si f_4 es activado entonces $g(k+1) = s_1 = \hat{y}_1(k+1)$. Esta última implicación puede ser cambiada por $g(k+1) = s_2 + s_3 = \hat{y}_2(k+1) + \hat{y}_3(k+1)$.

Para que este concepto tenga éxito es necesario que el modelo neuronal que se usa para aislar la falla sea de tipo recurrente. De lo contrario, las salidas estimadas tendrían un comportamiento diferente con la aparición de las fallas y no serían de utilidad para el proceso de aislamiento.

Capítulo 4

Diagnóstico de fallas en un sistema de nivel de líquidos

En este capítulo se desarrolla el modelo de un sistema de dos tanques con la intención de simularlo en condiciones normales de operación y de falla para generar los datos de entronamiento de cada uno de los modelos neuronales. También se comentan algunas de las características del programa en visual basic que desempeña el esquema FDI completo. Además, se inspeccionan las respuestas de cada uno de los diferentes panoramas de falla. Y finalmente, se consideran las ambigüedades en las respuestas y se propone una manera de incluirles para mejorar el desempeño del sistema FDI.

4.1 Modelo del sistema

El sistema no lineal real empleado en este trabajo para la aplicación de la técnica de diagnóstico de fallas usando redes neuronales, es el sistema de nivel de líquidos de dos tanques interconectados (unidad compacta FESTO PCS [113]) que se muestra en la figura 4.1.

La figura 4.2 muestra el esquema de conexiones del la unidad FESTO. En ella se puede observar que, sin acción de la bomba y por la diferencia de alturas entre los recipientes, la cantidad de líquido en el tanque 1 será transferida al tanque 2 debido a los flujo q_1 y q_2 .

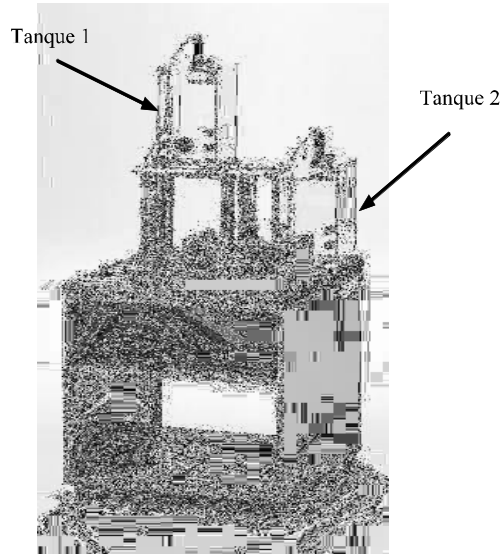


Figura 4.1: Sistema de tanques de agua (FESTO Compact unit PCS)

El modelo matemático del sistema de la figura 4.2 está dado por las ecuaciones 4.1.

$$\begin{aligned} \dot{h}_1 &= -\frac{q_1\sqrt{h_1}}{C} - \frac{q_2\sqrt{h_1}}{C} + \frac{u}{C} \\ \dot{h}_2 &= \frac{q_1\sqrt{h_1}}{C} + \frac{q_2\sqrt{h_1}}{C} - \frac{u}{C} \end{aligned} \quad (4.1)$$

Donde C es la capacitancia de cada tanque (m^2), h_1 y h_2 es el nivel de cada tanque (m), q_1 y q_2 es el valor de la rapidez de flujo en cada tanque (m^3/s) y u es el flujo de entrada (m^3/s). Debido a que no existen pérdidas ni ganancias en el sistema, los mismos elementos aparecen en las dos ecuaciones pero con signo distinto. Entonces se cumple que:

$$\begin{aligned} h_{Total} &= h_1 + h_2 \\ h_2 &= h_{Total} - h_1 \end{aligned} \quad (4.2)$$

El nivel en alguno de los tanques con el volumen total del sistema es de $0.356m$. Es decir, $0.0091m^3$ ($160mm * 160mm * 356mm$) y la capacitancia del recipiente es de $C = 0.0256m^2$. Así la diferencia de altura ganada en el primer tanque es igual a la diferencia perdida del segundo;

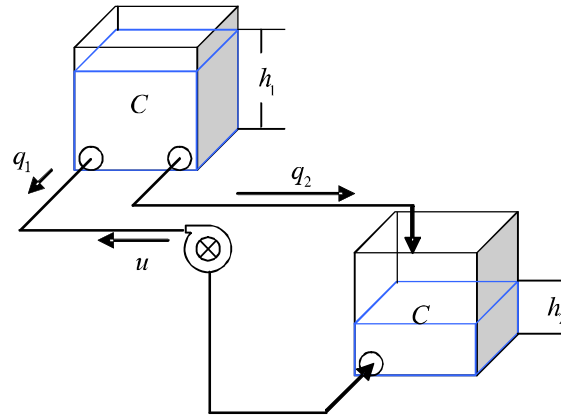


Figura 4.2: Esquema de conexiones del sistema de nivel de líquido de dos tanques interconectados

$$\dot{h}_2 = -\dot{h}_1$$

Las constantes relacionadas con los flujos de salida (q_1 y q_2) en 4.1, se obtuvieron de forma experimental. Considerando que $q_1 = q_2$, las constantes tienen un valor de $q_1 = q_2 = 0.0000379m^3/s$. Por lo tanto 4.1 queda como:

$$\begin{aligned} \dot{h}_1 &= -\frac{0.00000379\sqrt{h_1}}{C} - \frac{0.00000379\sqrt{h_1}}{C} + \frac{u}{C} \\ \dot{h}_2 &= \frac{0.00000379\sqrt{h_1}}{C} + \frac{0.00000379\sqrt{h_1}}{C} - \frac{u}{C} \end{aligned} \quad (4.3)$$

Una vez determinada la constante del flujo de salida es posible realizar el retrato de fase del sistema para corroborar la dinámica del sistema no lineal sin entrada.

En la figura 4.3 se muestra el retrato de fase, donde se observa que, con la condición inicial $P[h_1, h_2] = P[0.356, 0]$, la trayectoria de evolución del sistema indica la transferencia del volumen total del tanque 1 al tanque 2. Es decir se dirige al punto de equilibrio $P[0, 0.356]$, conservando durante el trayecto siempre la misma cantidad de volumen total. Más aun, todo el eje de las ordenadas son puntos de equilibrio.

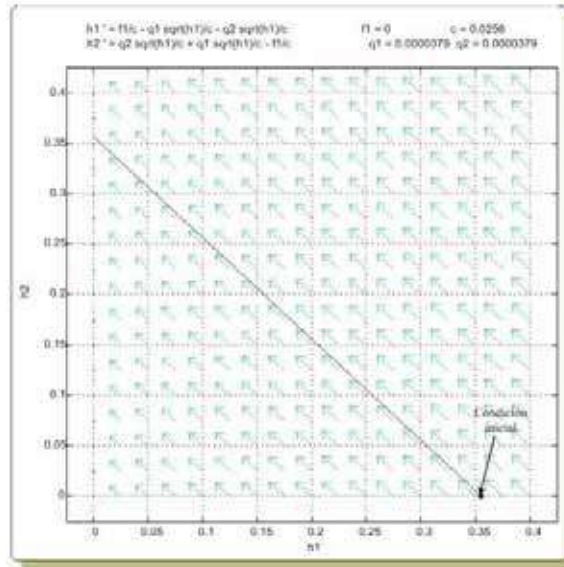


Figura 4.3: Diagrama de fase para el sistema sin entrada

4.2 Modelo neuronal del sistema de dos tanques

Para modelar con una red dinámica recurrente el sistema autónomo de dos tanques, no fueron necesarias las unidades de retardo ya que con sólo el ingreso de los estados y una capa oculta de cuatro nodos la red neuronal se consiguió un desempeño aceptable. Sin embargo, y debido a que el origen no es su punto de equilibrio fue necesario incluir los bias en el entrenamiento. Los pesos de la capa oculta y la capa de salida después del entrenamiento se muestran en las ecuaciones 4.4 y 4.5.

$$T = \begin{bmatrix} -0.177399570038039 & 0.529707980269448 & 0.466368674951268 & 0.556954077159651 \\ 0.186339880522207 & -0.430374348111460 & -0.505635453524371 & 0.978941739093427 \end{bmatrix} \quad (4.4)$$

$$W = \begin{bmatrix} 0.533737614036035 & 0.375746327327467 \\ -0.391023964861136 & -0.920279517585123 \\ 0.756930991609755 & 0.431557047708116 \\ -0.708777613329179 & -0.454473003620532 \end{bmatrix} \quad (4.5)$$

Los bias de cada respectiva capa son:

$$B_T = \begin{bmatrix} 0.208251452188225 \\ -0.264493413601478 \\ -0.351132585734485 \\ -0.431258356617935 \end{bmatrix}$$

$$B_W = \begin{bmatrix} 0.012113677313105 \\ 0.064608687106487 \end{bmatrix}$$

La figura 4.4 muestra del lado derecho la evolución del sistema (línea azul) y la red neuronal (línea roja) para la condición inicial $P[h_1, h_2] = P[0.356, 0]$. Del lado izquierdo se pueden apreciar los errores de estimación de los estados. En los cuales, se puede observar que ninguno de ellos supera el 5% de error.

El tiempo de muestreo en la simulación del sistema fue de 10 segundos. Sin embargo, para obtener las estimaciones de la red este dato tuvo que ser escalado por 0.01 en el procedimiento de integración debido a que un paso de cálculo de esa magnitud genera inestabilidad en el mismo.

Para la identificación de la dinámica del sistema con entrada, es necesario agregar su respectivo nodo a la capa de entrada además de incrementar la cantidad de nodos en la capa oculta.

4.3 Diseño de la interfaz en Visual Basic

La implementación tanto de la red neuronal como del monitoreo y registro de los datos del sistema se desarrolla en visual Basic 6.0. Lo anterior, debido a la facilidad de incluir módulos,

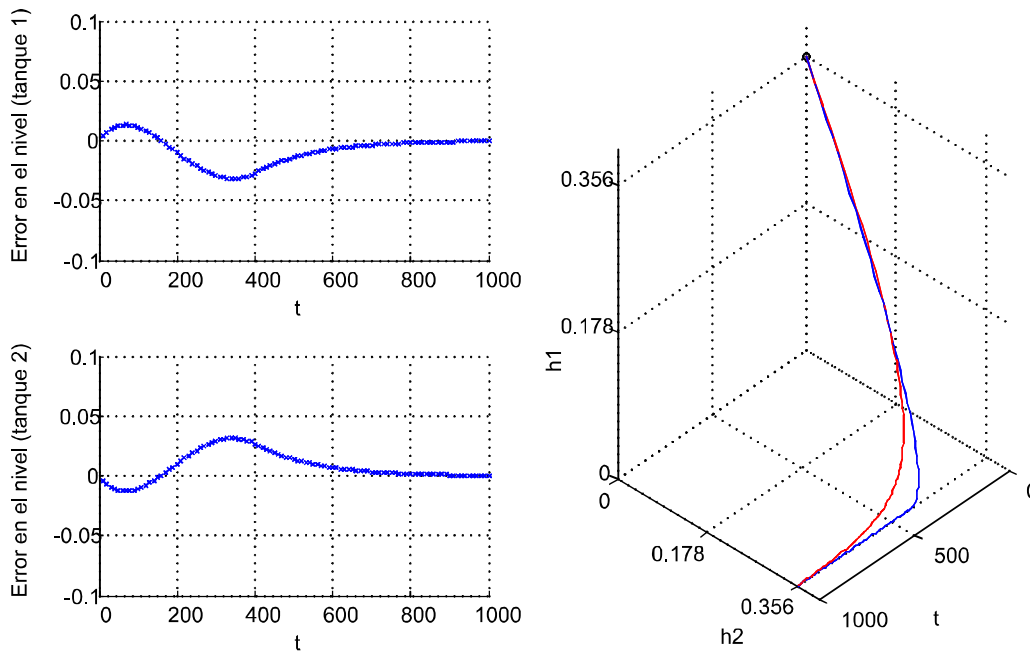


Figura 4.4: Modelado neuronal del sistema de dos tanques

clases de objetos y controles OCX que facilitan la organización y distribución de código en cada programa. Así entonces, existe una clase encargada de crear, salvar, cargar e inicializar con datos aleatorios la red neuronal. Además de métodos que realizan la propagación hacia adelante de las señales de entrada y la propagación del error hacia atrás como se especifica en el algoritmo Back-propagation. También se incluye una clase que se encarga de manejar las líneas de retardo (Taped Delay Lines, TDL) y su concatenación empleada en la alimentación de la red. Una clase más se emplea en el programa para desplegar los resultados gráficamente tanto de los datos del sistema como de la salida de la red.

Con el objetivo de facilitar la visualización de la distribución de datos en la red neuronal, se ha incluido en el programa una ventana de visualización (ver figura 4.5) con la capacidad de representar las conexiones y valores en cada nodo interno del sistema neuronal. Los nodos de entrada, salida y los nodos maestros pueden ser apreciados numéricamente, mientras que los internos se identifican por tonos de colores; azul para el valor máximo asintótico (1), rojo

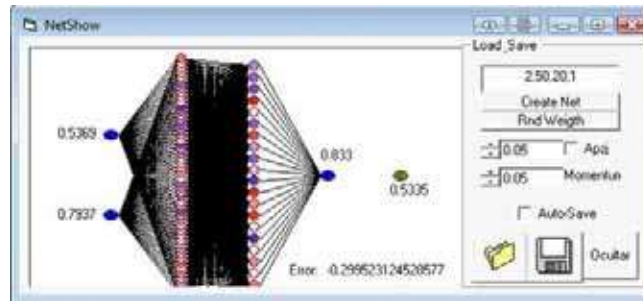


Figura 4.5: Ventana de visualización

para el mínimo asintótico (0) y blanco para el valor medio (0.5) pasando por los diversos tonos. En esta ventana se tienen las opciones básicas sobre la red; crear, cargar, salvar, auto-salvar, asignar datos aleatorios, asignar el coeficiente de aprendizaje, asignar el término de momento.

El programa cuenta con una pantalla principal donde se muestra el esquema FDI completo (figura 4.6) con opción de graficar los nodos de mayor relevancia como; la salida de cada una de las redes, el error de estimación de cada modelo, los residuos, la entrada y las salidas del sistema simulado así como la señal de salida $g(k)$ del esquema FDI. El esquema FDI fue implementado con los bloques comentados en el capítulo 3.

4.4 Respuesta de los modelos

Para fines prácticos los escenarios de fallas a considerar son; sistema libre de fallas, falla en el *sensor* 1, falla en el *sensor* 2 y falla en ambos sensores. Sin embargo, pueden ser incluidas mas fallas para incrementar lo robusto del sistema. La aparición de las fallas ocurre de manera súbita llevando a corto las líneas provenientes de los sensores. Una vez terminado el entrenamiento de cada uno de los modelos, la siguiente fase es integrarlos para validar su desempeño como sistema diagnóstico.

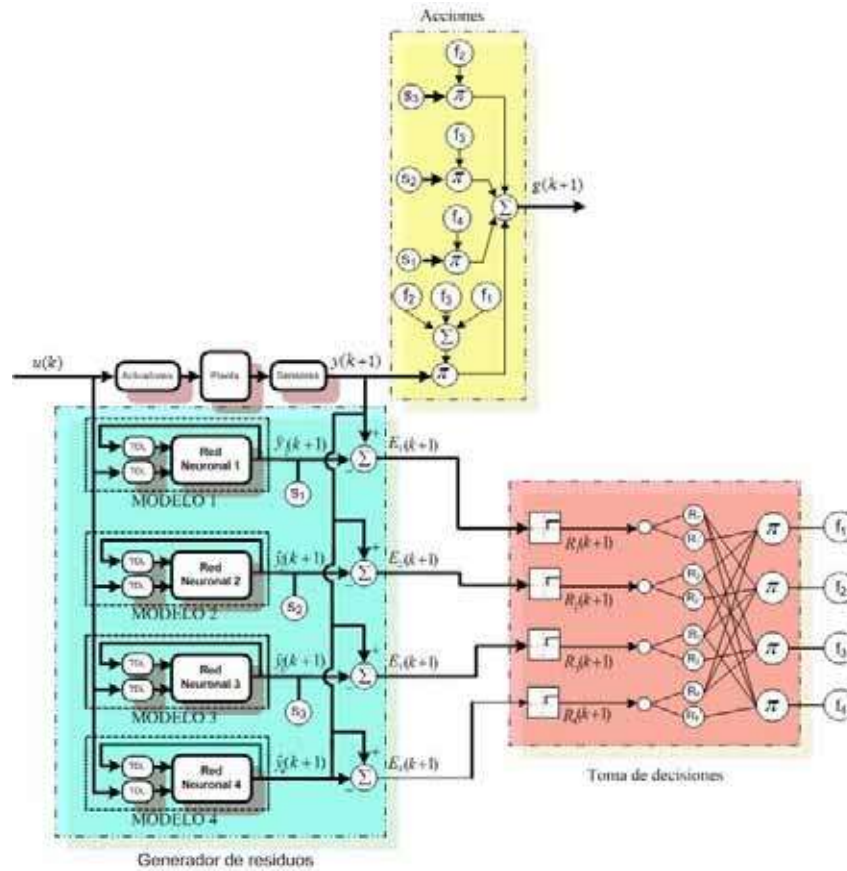


Figura 4.6: Esquema completo del sistema FDI

4.4.1 Sistema libre de falla

En de la figura 4.7 se muestran las gráficas para un intervalo de tiempo con las señales del sistema de tanques. La gráfica superior-izquierda muestra la señal de entrada, la inferior-izquierda el comportamiento del sistema. En el esquema libre de falla, la señal de los sensores (gráfica superior centro) coincide plenamente con las señales del sistema. La gráfica a la derecha son las señales del vector de salida $g(k)$ considerada en el esquema de la figura 3.5 del capítulo 3. La figura inferior-centro es el error entre las señales del sistema y las señales del sistema FDI.

Ya que los modelos han sido entrenados en diferentes situaciones de falla (figura 4.8), las

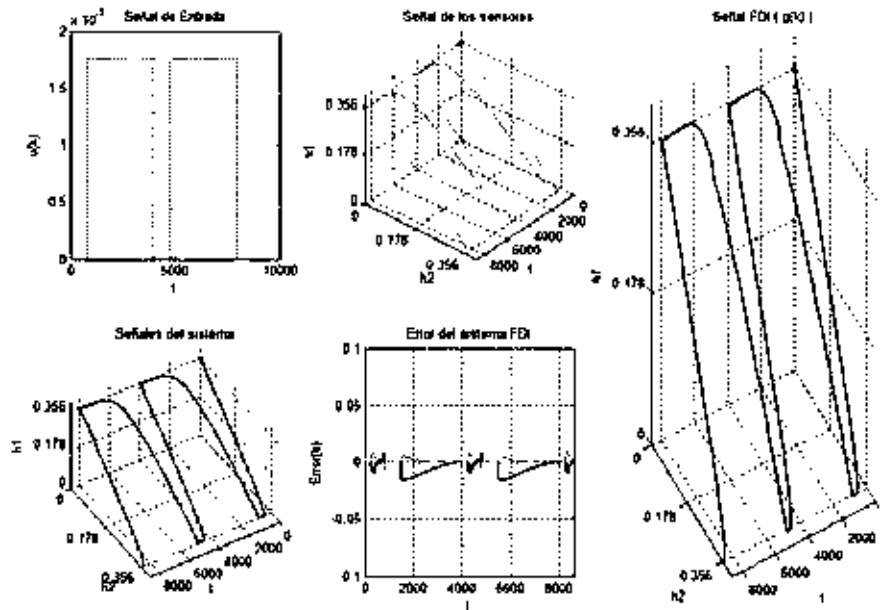


Figura 4.7: Comportamiento normal del sistema sin falla

respuestas de cada uno de ellos generan distintos residuos que son usados para la toma de decisiones. En las figuras 4.9 y 4.10 se muestran las señales de error de estimación y residuos booleanos, respectivamente, para cada uno de los modelos del sistema FDI.

En condiciones normales de operación, el residuo del modelo libre de falla no es activo debido a que su error de estimación no supera un umbral asignado de 0.05 (5% de error). Los modelos restantes si activan su correspondiente residuo. Sin embargo, para el *modelo 2* y para el *modelo 3*, en diferentes instantes sus correspondientes residuos no son activados. Esto es debido a la gran similitud entre las señales de los sensores y la salida de dichos modelos en ese instante. Su interpretación es que existe una *zona de incertidumbre* que dificulta decidir si el sistema se encuentra en la región de operación normal o en falla. En la siguiente sección se propone la manera de contemplar estas incertidumbres de manera que el sistema FDI no genere falsas alarmas.

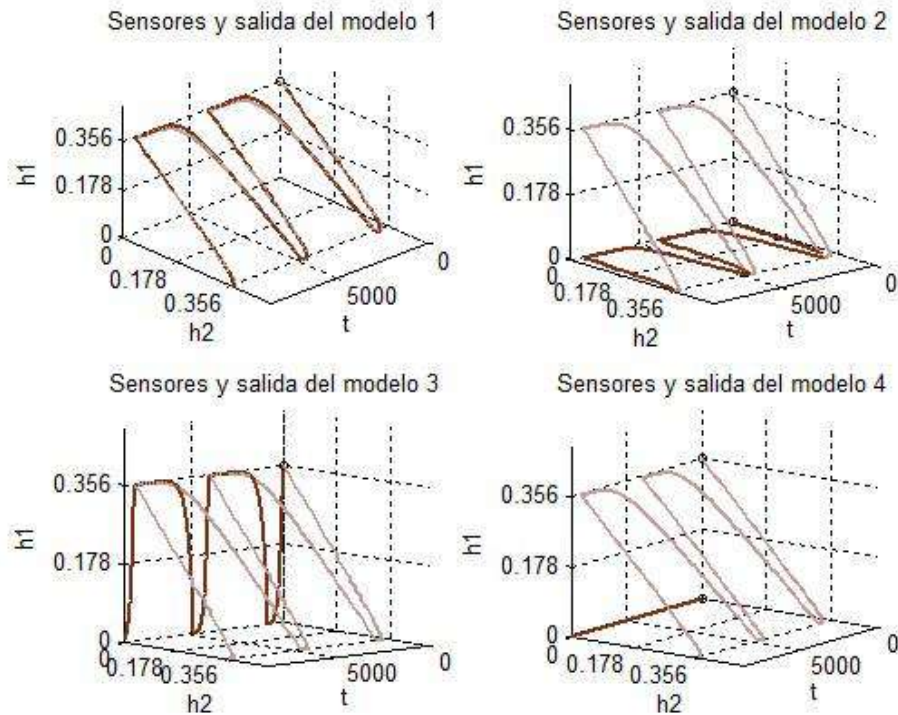


Figura 4.8: Señales de los cuatro modelos neuronales en condiciones normales de operación

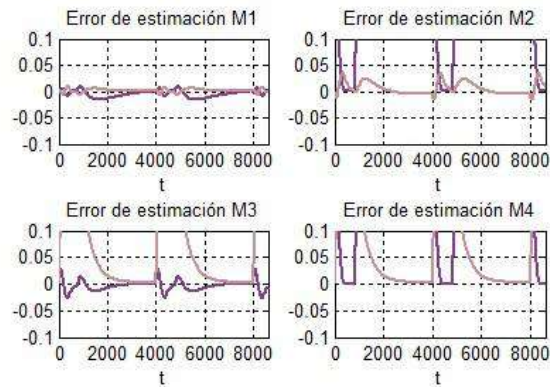


Figura 4.9: Señales de error de los cuatro modelos neuronales en condiciones normales de operación

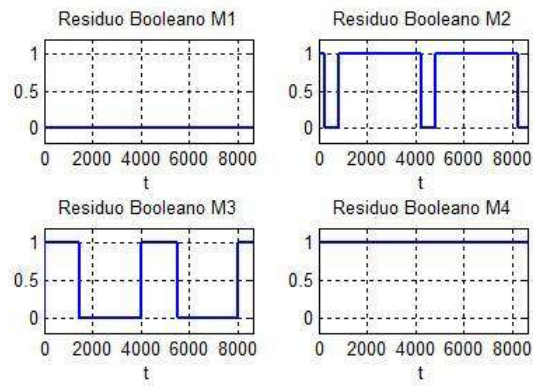


Figura 4.10: Señales de error booleano de los cuatro modelos neuronales en condiciones normales de operación

4.4.2 Falla en el sensor 1

La señal del *sensor 1* es llevado a corto circuito quedando sólo apreciable la señal del *sensor 2*, como se muestra en la gráfica superior-centro de la figura 4.11. La salida (figura a la derecha) y el error en el sistema FDI (figura inferior-centro) verifica lo robusto de la salida $g(k)$ ante la presencia de la falla. Cabe notar que en este panorama de falla la salida corresponde a la dada por los sensores mas el *modelo 3* y la señal de los sensores, figura superior-centro, son sólo la proyección de las señales del sistema con $h_1 = 0$. En la figura 4.12 se muestra la evolución de cada uno de los modelos neuronales con su característico comportamiento.

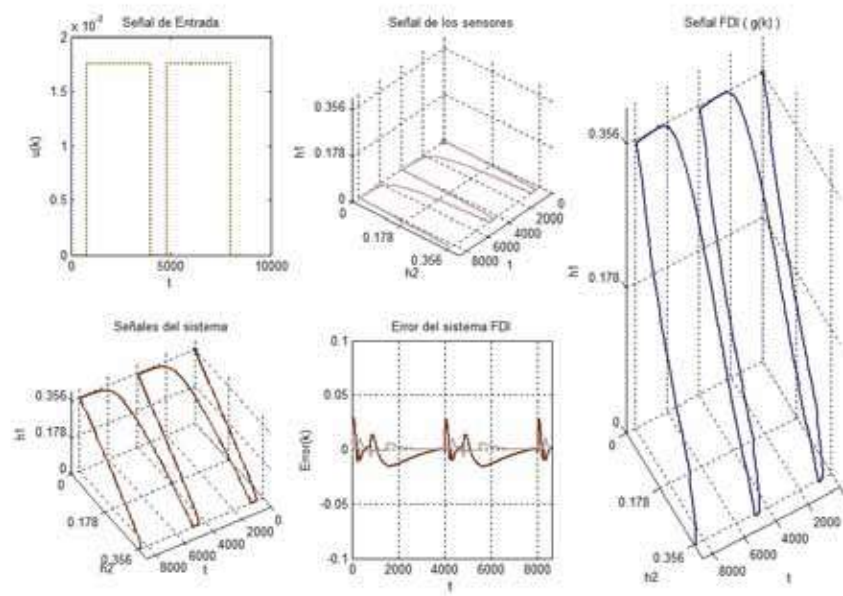


Figura 4.11: Comportamiento en presencia de falla en el sensor 1

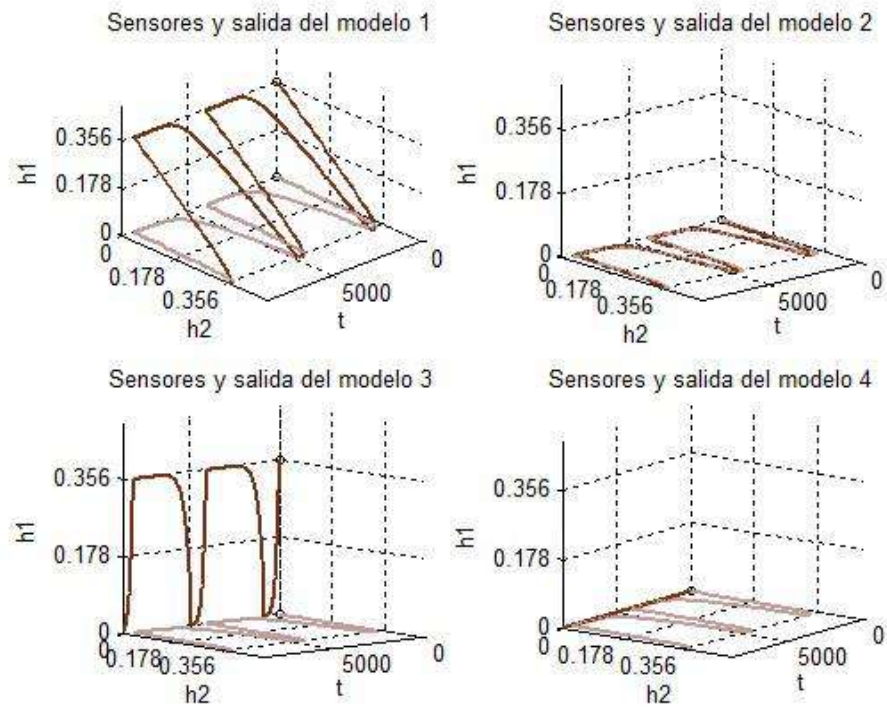


Figura 4.12: Señales de los cuatro modelos neuronales en presencia de la falla en el sensor 1

El *modelo 2* se comporta de manera similar a las señales de los sensores y su residuo booleano no es activado. Los modelos restantes sí activan sus residuos debido a la desviación del comportamiento aún existente. En este panorama de falla también existen zonas de incertidumbre atribuidas al *modelo 1* y el *modelo 4*.

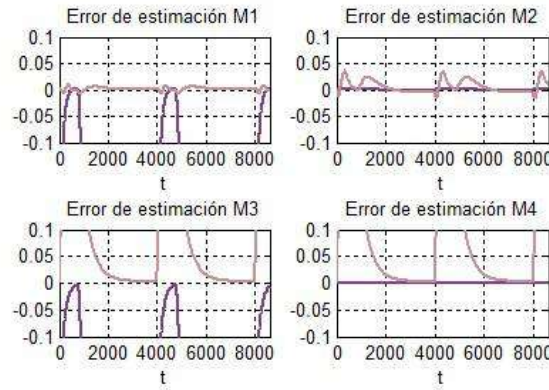


Figura 4.13: Señales de error de los cuatro modelos neuronales en presencia de la falla en el sensor 1

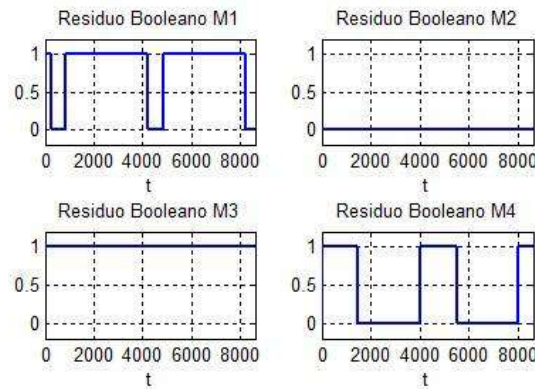


Figura 4.14: Señales de error booleano de los cuatro modelos neuronales en presencia de la falla en el sensor 1

4.4.3 Falla en el sensor 2

De manera similar al caso anterior, como se muestra en la gráfica superior-centro de la figura 4.15 se incluye una falla en el *sensor 2* y se comporta en corto circuito es decir, la señal de los sensores sera la proyección del sistema con $h_2 = 0$. La gráfica a la derecha muestra la señal $g(k)$ con su característica protección en presencia de la falla.

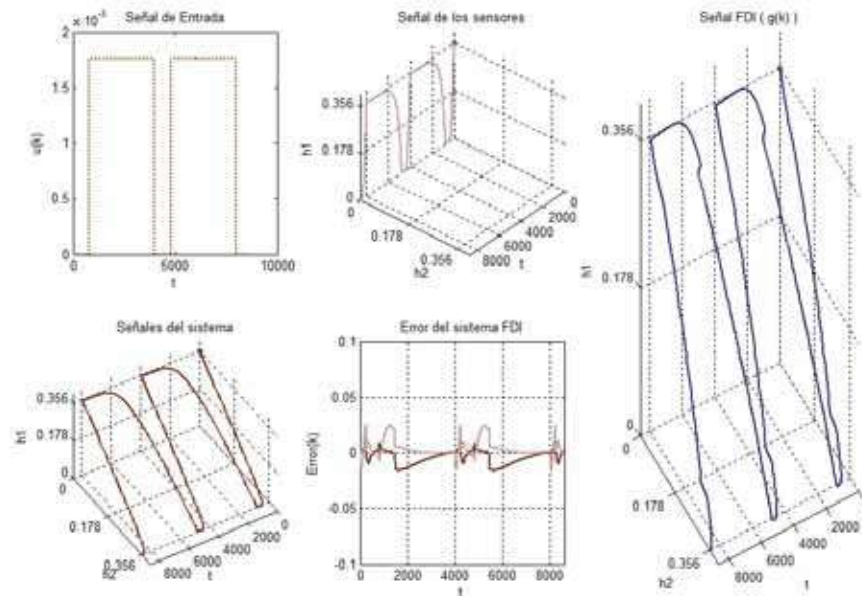


Figura 4.15: Comportamiento en presencia de falla en el sensor 2

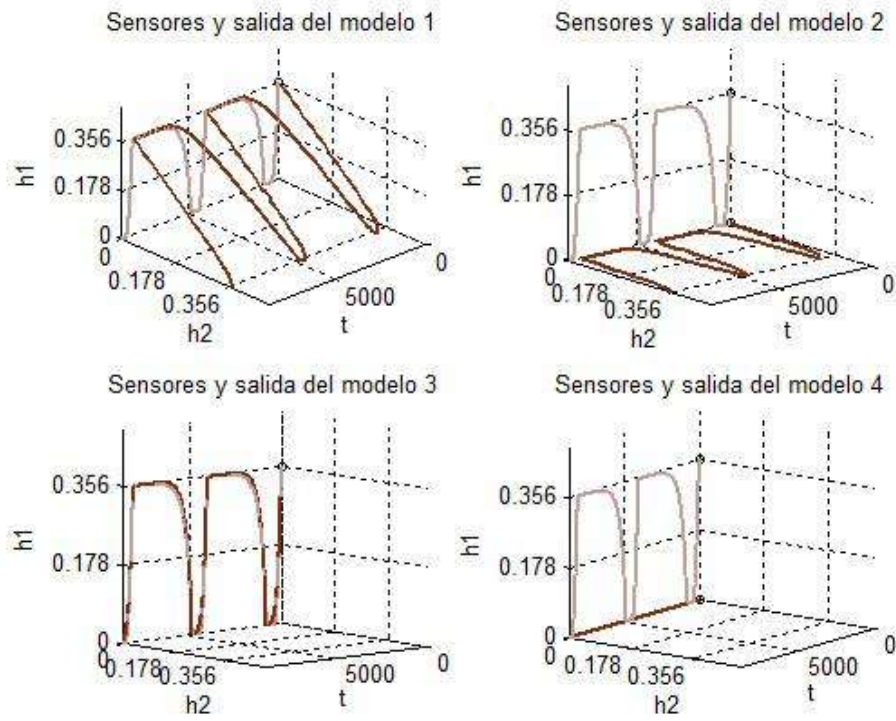


Figura 4.16: Señales de los cuatro modelos neuronales en presencia de la falla en el sensor 2

Las señales de los modelos para este panorama de falla son mostrados en la figura 4.16. En ella se puede ver que el *modelo 3* consigue un comportamiento similar al del sistema con falla y su residuo no es activado, mientras que los demás modelos si lo hacen. Las zonas de incertidumbre involucran al *modelo 1* y al *modelo 4*.

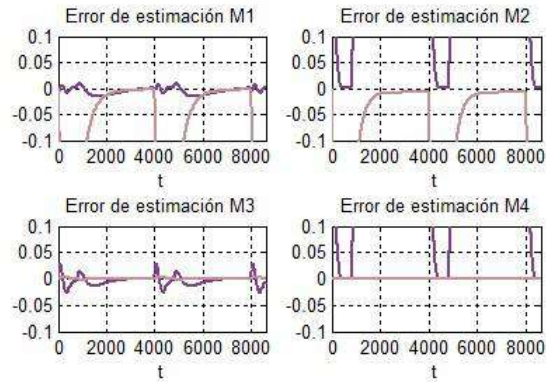


Figura 4.17: Señales de error de los cuatro modelos neuronales en presencia de la falla en el sensor 2

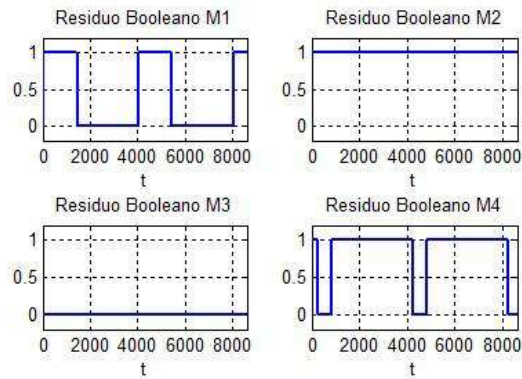


Figura 4.18: Señales de error booleano de los cuatro modelos neuronales en presencia de la falla en el sensor 2

4.4.4 Falla en ambos sensores

Finalmente ambos sensores son llevados a corto y se corrobora que la señal de salida $g(k)$ no se ve afectada por la falta de información, como se muestra a la derecha en la figura 4.19, ya que la señal del sistema es sustituida en su totalidad por el modelo neuronal sin falla o bien, por una suma de los modelos con falla.

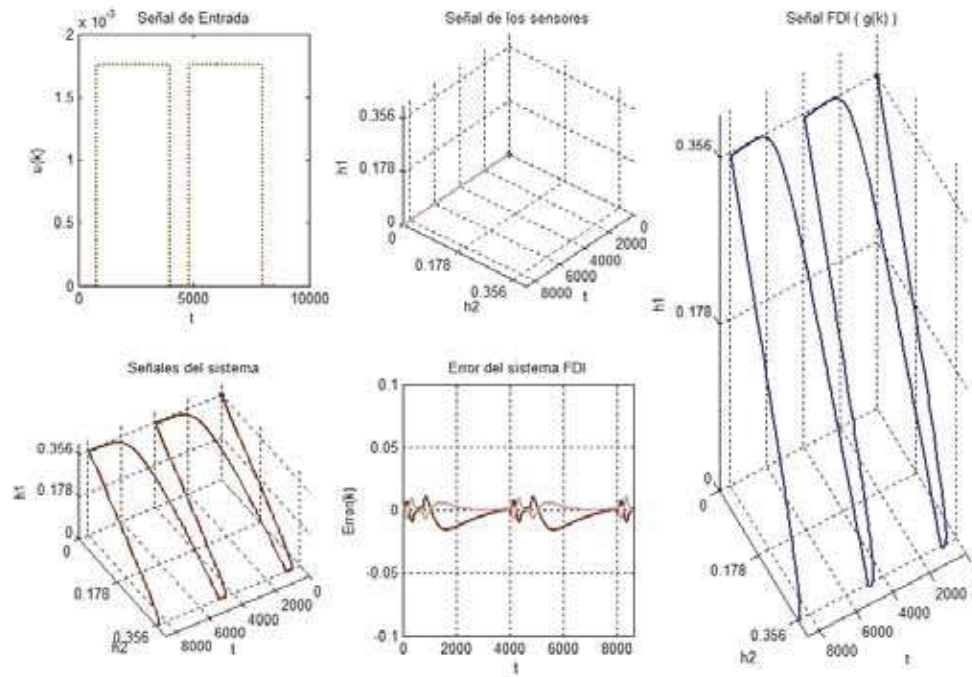


Figura 4.19: Comportamiento en presencia de falla en ambos sensores

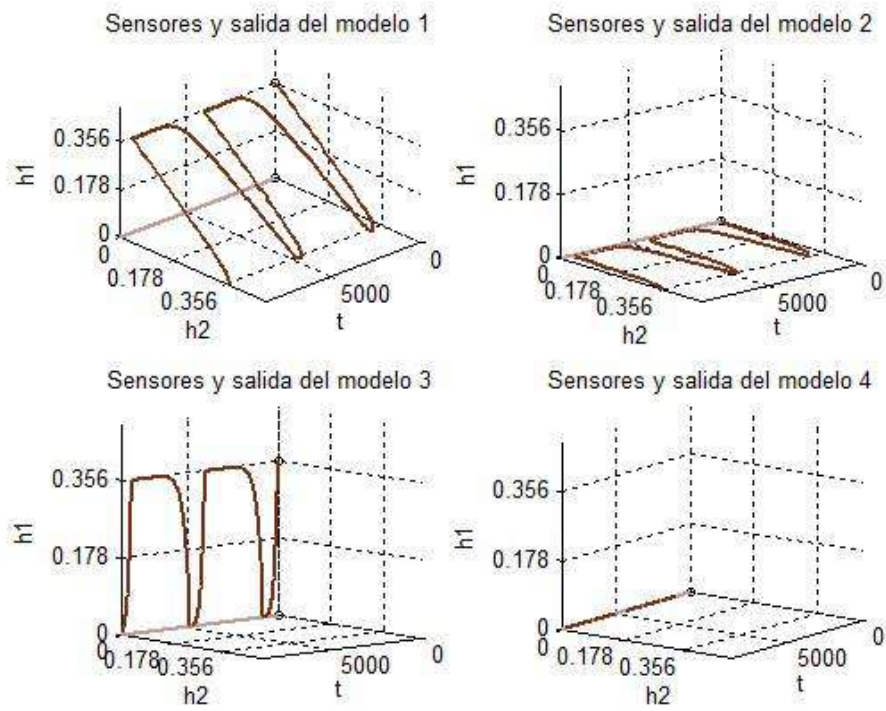


Figura 4.20: Señales de los cuatro modelos neuronales en presencia de falla en ambos sensores

En la figura 4.20, el *modelo 4* se ve identificado con las señales de los sensores y como resultado desactiva el *residuo 4*. Los modelos restantes indican un excedente del umbral y activan sus residuos. También aparecen las zonas de incertidumbre que implican al *modelo 2* y al *modelo 3*. En las figuras 4.21 y 4.22 se muestran las señales error generado y su residuo booleano correspondiente para cada uno de los cuatro subsistemas.

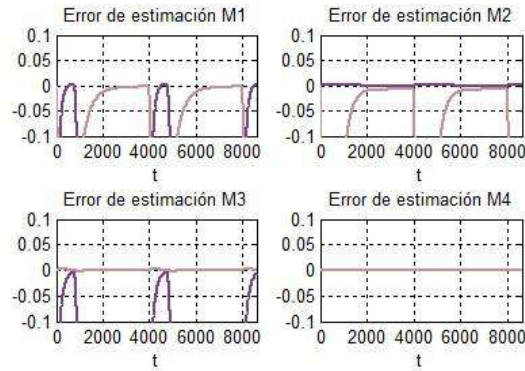


Figura 4.21: Señales de error de los cuatro modelos neuronales en presencia de la falla en ambos sensores

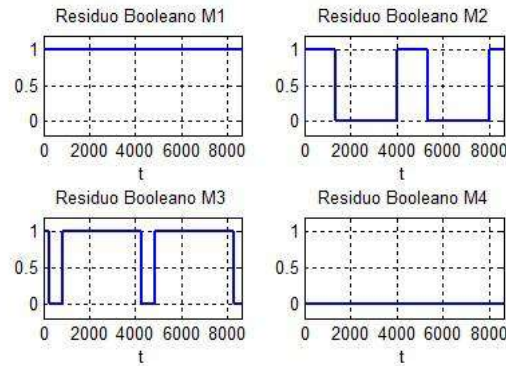


Figura 4.22: Señales de error booleano de los cuatro modelos neuronales en presencia de falla en ambos sensores

	\bar{R}_1	\bar{R}_1	R_1	R_1
	\bar{R}_2	R_2	R_2	\bar{R}_2
	R_3	\bar{R}_3	\bar{R}_3	R_3
	R_4	R_4	\bar{R}_4	\bar{R}_4
Incertidumbres en el sistema sin falla	1	1		
Incertidumbre con falla en el <i>sensor 1</i>	1			1
Incertidumbre con falla en el <i>sensor 2</i>		1	1	
Incertidumbre con falla en ambos sensores			1	1

Tabla 4.1: Resumen de incertidumbres

IF	THEN	ELSE
\bar{R}_1 And \bar{R}_2 And R_3 And R_4	m_1 is <i>true</i>	m_1 is <i>false</i>
\bar{R}_1 And R_2 And \bar{R}_3 And R_4	m_2 is <i>true</i>	m_2 is <i>false</i>
R_1 And \bar{R}_2 And R_3 And \bar{R}_4	m_3 is <i>true</i>	m_3 is <i>false</i>
R_1 And R_2 And R_3 And \bar{R}_4	m_4 is <i>true</i>	m_4 is <i>false</i>

Tabla 4.2: Reglas para las combinaciones de incertidumbre

4.5 Zonas de incertidumbre

Para evitar que esta situación tenga efectos no deseados en la señal de salida, la combinación de residuos en ese instante deberá ser considerada como regla en la toma de decisiones. En este capítulo la simulación de la señal de salida $g(k)$ incluye la adición de estas reglas. Las combinaciones de residuos en zonas de incertidumbre pueden ser resumidos en la tabla 4.1. Estas combinaciones se presentan por pares tanto para el panorama libre de falla como para cada uno de los panoramas con falla. Las reglas para incluir estas combinaciones de incertidumbre en los residuos se muestra en la tabla 4.2.

La figura 3.4 del capítulo 3 es modificada como se muestra en la figura 4.23. En la que se muestra la toma de decisiones incluyendo estas regla como m_1 , m_2 , m_3 , y m_4 . La figura 3.5 del capítulo 3 también es modificada como se muestra en la figura 4.24.

En la figura 4.25 se pueden ver una secuencia del sistema simulado donde aparecen las señales en condición normal de operación y los panoramas de falla en intervalos de 4000 segundos; la falla en el sensor 1 aparece en el segundo 4000, el sensor 2 falla en el segundo 8000 y ambas

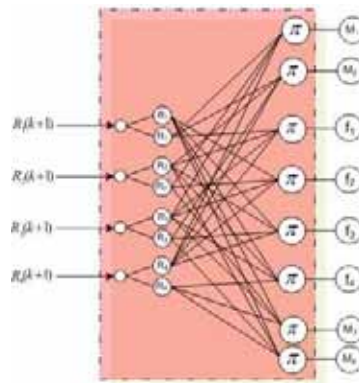


Figura 4.23: Esquema final de reglas para la toma de decisiones

fallas en el segundo 12000. También se puede apreciar que la señal de salida $g(k)$ es protegida usando los datos del modelo neuronal libre de falla y los modelos con falla de acuerdo con la toma de decisiones. También se observan eventos de incertidumbre de los residuos. Sin embargo, su aparición tampoco afecta a la señal $g(k)$ ya que han sido incluidas como reglas en la toma de decisiones.

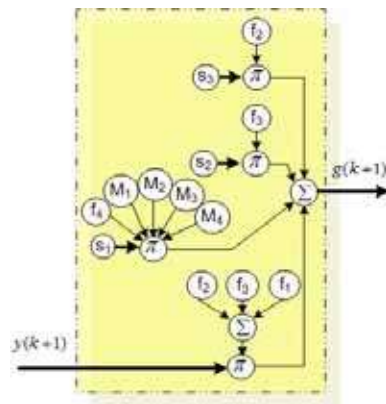


Figura 4.24: Esquema final de acciones

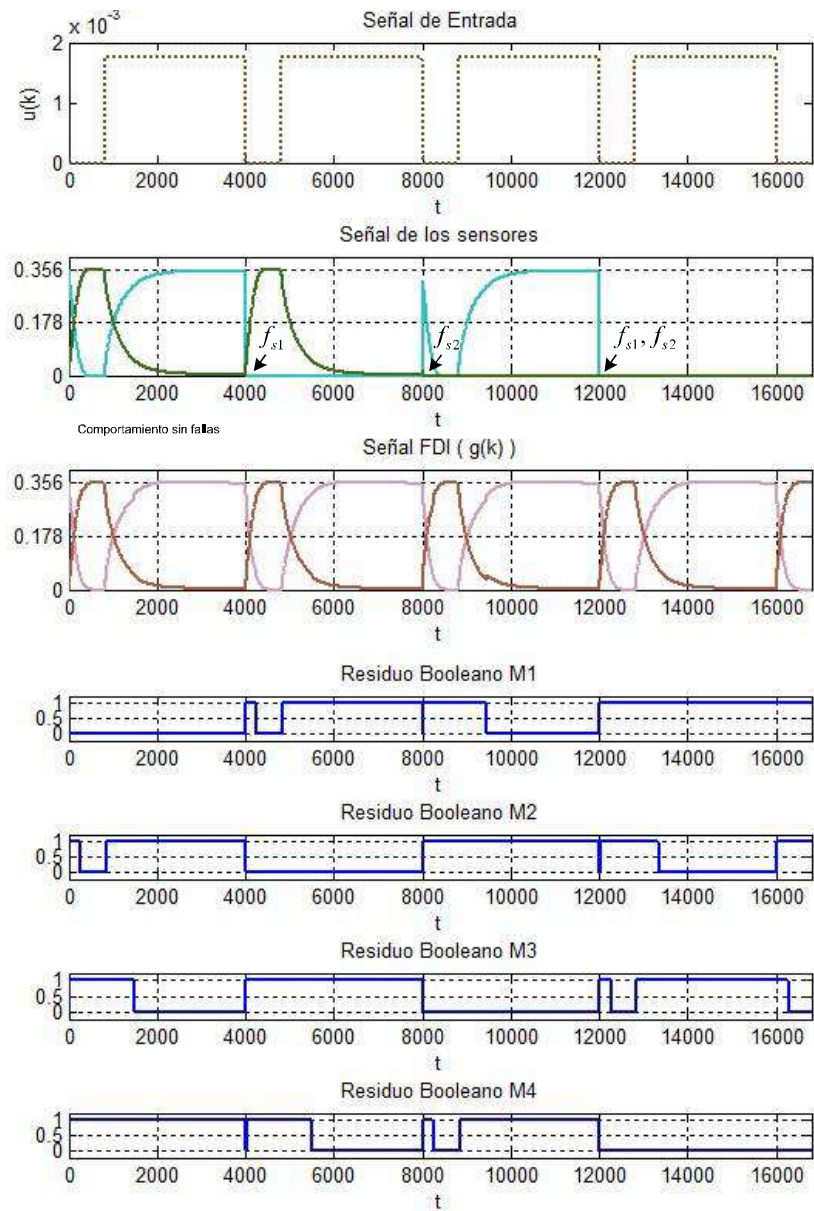


Figura 4.25: Secuencia con múltiples fallas

Capítulo 5

Conclusiones y trabajo futuro

Mediante el desarrollo del presente trabajo de tesis, se ha tenido la oportunidad de investigar uno de los paradigmas inspirados mayormente en nuestra propia comprensión del conocimiento y de como la naturaleza que nos rodea a encontrado la manera de distribuir y aprovechar los recursos en beneficio mismo de las especies que en ella residen. A pesar de que las aplicaciones computacionales de las RNA's distan mucho de las cualidades de sus análogos biológicos (ya que incluso un organismo viviente simple implica delicadas y complejas fases de procesamiento superiores a las capacidades de cualquier computadora), la búsqueda de nuevos espacios de aplicación de las redes neuronales por parte de los investigadores sigue siendo una opción muy atractiva y popular debido a la gran cantidad de documentos publicados con excelentes resultados.

Como mencionan algunos autores, las redes neuronales pueden ser vistas como modelos genéricos. Esta característica es muy deseable en diversas aplicaciones, ya que en ocasiones resulta difícil describir las relaciones entre variables o son vagamente entendibles e incluso pueden ser consideradas cuando el número de variables es muy grande. El empleo de las RNA's como modelos genéricos puede mejorar el modelado y significar una diferencia en la eficiencia de la operación de un sistema, lo que da como consecuencia una mejora en la relación costo-beneficio.

En el presente trabajo, se diseñó un sistema diagnóstico de fallas basado en un conjunto de

modelos que describen al sistema en cuestión (sistema de tanques interconectados) en los panoramas libre de falla y con falla en cada uno de sus sensores. El sistema diagnóstico diseñado, consiguió un excepcional desempeño para cada uno de los escenarios de falla ya que es fácilmente identificable el elemento que fallo en el sistema debido a que el enfoque planteado contempla todas las posibles fallas en todos los posibles elementos, a diferencia de algunas otras técnicas basadas en el modelo (como los basados en observadores), en los cuales de manera sencilla no se consideran fallas múltiples. Con esto, se corrobora la fácil incorporación de los modelos neuronales no simplemente a los sistemas diagnósticos sino a una gran cantidad de aplicaciones en las cuales pueden ser útiles. Para la generación de residuos, resulta muy interesante usar RNA's recurrentes debido a que algunos de los modelos pueden ser usados para el aislamiento de las fallas en sensores. Para la detección y el aislamiento de fallas en actuadores, se deberán incluir otras consideraciones. Adicionalmente, en la toma de decisiones es indispensable incluir algunas reglas adicionales a las establecidas para identificar los modelos (con y sin falla) con el propósito de realizar un correcto aislamiento y evitar comportamientos no deseados en el sistema FDI. La aparición de estos comportamientos no deseados en el sistema FDI es el resultado de la gran similitud, o incertidumbre, entre los modelos y el comportamiento del sistema no lineal en ciertas regiones. Algunas de estas situaciones son comparables a aseverar premeditadamente que un sistema no tiene falla si no se ha puesto en funcionamiento o las señales entradas son nulas y el sistema se encuentra en reposo en un punto de equilibrio. Para poder diferenciar estos panoramas es necesario estudiar el comportamiento del sistema sobre todo su rango de operación e incluir como reglas, en la toma de decisiones, las situaciones de incertidumbre que no implican fallas para evitar falsas alarmas. Este último procedimiento será implementado, como trabajo futuro, mediante un algoritmo o empleando una red neuronal adicional, como toma de decisiones, en la cual el diseñador suministre el conocimiento experto tanto en la evaluación de los residuos como en las situaciones de incertidumbre.

Un tema destacado dentro del contexto de las redes neuronales son los algoritmos de aprendizaje. En los cuales, el aspecto matemático es fundamental para el entendimiento de conceptos tales como la estabilidad. Afortunadamente existen varias publicaciones que abordan esta

problemática, basándose en características específicas en la configuración de la red neuronal, e incluso proponen nuevos algoritmos de aprendizaje, algunos de ellos basados en métodos híbridos o combinados. Unos de estos algoritmos con resultados significativos son los algoritmos genéticos (*genetics algorithm*), ya que se basan en criterios de evaluación sobre un conjunto de modelos o población, lo que aporta ventajas en la búsqueda de la solución. En este trabajo por ejemplo, la selección de la cantidad de nodos en la capa oculta de cada uno de los modelos se ha realizado manualmente (prueba-error). Lo que ocupa invertir tiempo en la identificación de la estructura que desempeñe el mejor comportamiento. Por ello, como trabajo futuro se pretende implementar un algoritmo genético para la elección de la mejor estructura, con la finalidad de mejorar el tiempo invertido en dicha tarea y de contribuir en el diseño de un sistema de diagnóstico.

Por otro lado, también se propone como trabajo futuro aprovechar las cualidades del Perceptrón multicapa como discriminador de regiones lineales (función de activación *hardlimit*). Ya que, varios discriminadores en paralelo, permiten construir mapeos complejos con una cantidad mínima de elementos simples, mejorando con ello la calidad de la predicción de las señales en un sistema. Sin embargo, se puede especular acerca del gran tamaño de la estructura final cuando las regiones a cubrir son bastante grandes, cuestión que una red Perceptrón multicapa común, con función de activación diferenciable y suficientes nodos en la capa oculta, pretende solucionar automáticamente con los algoritmos de entrenamiento clásicos. En este punto, se puede reconocer que la determinación de implementar o no una metodología dependen de los requerimientos de capacidad y prioridad del problema a tratar. Entretanto, el comportamiento de los discriminadores de regiones lineales en diversas tareas también representa una opción digna de analizar, incluso en la identificación de sistemas no lineales y en el diseño de sistemas FDI.

Por último, cabe recordar que los problemas en la vida cotidiana no son todos solucionables, y si lo son, no poseen soluciones únicas. En general los problemas pueden ser considerados como directos e inversos. La resolución de un problema directo involucra análisis o razonamiento progresivo, de premisas a conclusiones, de causas a efectos. Resolver un problema inverso involucra síntesis, o razonamiento regresivo, de conclusiones a premisas o de efectos a causas.

Las redes neuronales por consiguiente, son un mecanismo de solución inversa que nos ilustra como enfrentarse a problemas de diferente índole, dentro de los cuales se puede considerar la detección y aislamiento de fallas en sistemas no lineales un problema de gran importancia por su amplia aplicación.

Apéndice A

Algoritmo *BackPropagation*

En el presente trabajo de tesis, la forma de actualizar los pesos se basa en el *algoritmo backpropagation* (retropropagación del error). El algoritmo *back-propagation* esta formado por dos procedimientos; la propagación hacia adelante de las señales de entrada y la propagacion hacia atrás del error estimado en cada nodo. Matricialmente pueden ser considerados como se muestra el figura A.1.

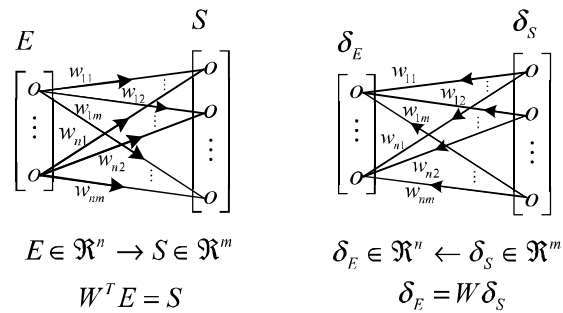


Figura A.1: Procedimientos del algoritmo *Back-Propagation*

$\delta_j = (t_j - O_j) O_j (1 - O_j)$	Para las neuronas de la capa de salida
$\delta_j = O_j (1 - O_j) \sum_k \delta_k w_{jk}$	Para las neuronas de las capas ocultas

Tabla A.1: Ecuaciones de error para los nodos

A.1 Propagación hacia delante

Las nodos de entrada reciben un vector señal de entrada y lo transmiten a cada una de las $m - th$ neuronas de la capa oculta próxima. Estas neuronas calculan la suma de pesos de sus entradas y producen una señal de acuerdo a su función de activación y la envían a la capa oculta siguiente. Así sucesivamente hasta que las señales de la última capa oculta va entonces a cada una de las $r - th$ neuronas de la capa de salida (figura A.2).

FIGURA A.2: Propagación hacia adelante

Figura A.2: Propagación hacia adelante

A.2 Propagación hacia atrás

En la propagación hacia atrás, la salida actual de la red es comparada con el valor deseado de la salida y un error es calculado con las formulas del cuadro A.1 para los nodos de la red.

La actualización de los pesos estará dada por la ecuación A.1.

$$\begin{aligned} \Delta w_{ji} &= \eta \delta_j O_i + m \Delta w_{ji} \\ w_{ji}(k) &= w_{ji}(k-1) + \eta \delta_j O_i + m \Delta w_{ji} \end{aligned} \tag{A.1}$$

donde η se refiere a la tasa de aprendizaje, m es el término conocido como *momento*, δ_j se refiere a la señal de error en la neurona j -th en la capa L , O_j se refiere a la salida de la neurona j -th en capa L y O_i se refiere a la salida de la neurona i -th en capa $L - 1$ [109]. En la figura A.3.

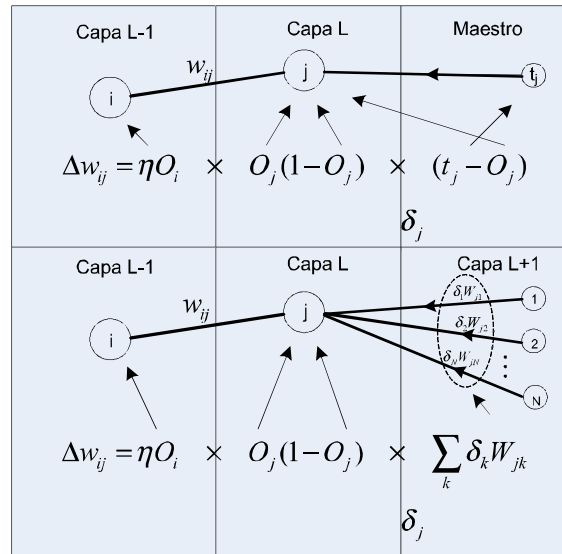


Figura A.3: Esquema conceptual de la propagación hacia atrás

Matricialmente el procedimiento de propagación hacia atrás puede ser considerado como en la figura A.4.

Figura A.4: Propagación hacia atrás

La actualización de pesos puede ser luego de la presentación de cada patrón o después de presentar el conjunto completo de patrones (entrenamiento batch). En ambos casos, una

época (epoch) de entrenamiento es completada cuando todos los patrones de entrenamiento han sido presentados a la red.

A.3 Consideraciones del algoritmo

La tasa de aprendizaje η indica cuanto cambian los pesos. Si η es pequeño, la velocidad de aprendizaje es muy lenta. Si es muy grande, aparecen efectos oscilatorios [118]. El algoritmo hace que el sistema evolucione en la dirección de la línea de máxima pendiente. Es útil comenzar con valores altos de α y del momentum (coeficiente de aprendizaje), e ir reduciéndolos poco a poco. De acuerdo con [118], algunos de los principales problemas del algoritmo *back-propagation* son :

- La dirección de máxima pendiente no siempre es la ruta más directa hacia el mínimo de la función error. Por lo que, puede converger a un mínimo local.
- Si la pendiente es escasa, el entrenamiento es lento.
- Si la pendiente es nula, el algoritmo se detiene.

También algunas deficiencias del algoritmo *back-propagation* comentados en [118] son:

- Parálisis de la red. Los pesos toman valores muy grandes, y la red no cambia.
- Mínimos locales. La superficie de error esta llena de valles y picos. Es difícil localizar el mínimo global, y suele caer en mínimos locales.
- Dificultad de elegir la arquitectura de la red (número de capas, número de unidades en cada capa,...), los parámetros de aprendizaje (η , momento, error aceptable,...).
- Lentitud del aprendizaje.

Sin embargo, es el algoritmo más popular para el entrenamiento de redes neuronales artificiales. El objetivo de emplearlo es la de corroborar sus características y proponer algún mecanismo que agregue alguna mejora en su implementación.

Apéndice B

Glosario

Debido a la variada terminología y diversos enfoques en el campo de diagnóstico de fallas, el comité técnico "Detección de fallas, supervisión y seguridad para procesos técnicos (SAFE-PROCESS)" de la Federación Internacional de Control Automático (IFAC Technical Committee), tuvo la iniciativa para definir una terminología común. Algunas de las definiciones establecidas se mencionan a continuación:[21].

1. Aislamiento de falla: Determinación de la clase, localización y tiempo de detección de la falla. Este es un proceso que complementa el concepto de la detección de falla.
2. Confiabilidad: Habilidad de un sistema para desarrollar una función requerida bajo condiciones especificadas en un rango dado durante un periodo de tiempo especificado.
3. Detección de falla: Determinación de las fallas presentes en un sistema y el tiempo en el que son detectadas.
4. Diagnóstico basado en el modelo: Método que emplea la redundancia analítica, es decir, usa un modelo matemático del sistema.
5. Diagnóstico de falla: Determinación de la clase, tamaño, localización y el tiempo de detección de la falla. Incluye la detección, aislamiento e identificación, es decir, no solamente el proceso de reconocer la presencia de una falla, sino además identificar cual

- componente del sistema ha fallado.
6. Error: Desviación entre un valor medido o calculado (de una variable de salida) y el valor especificado o teóricamente correcto.
 7. Falla: Una interrupción permanente de la habilidad de un sistema para desarrollar una función requerida bajo condiciones de operación especificadas; es decir, un cambio en el comportamiento del sistema debido a entradas externas que exceden los límites de tolerancia pre-establecidos.
 8. Firma de falla: Un efecto o patrón característico de una falla que se refleja en el residuo del sistema físico. Representado en forma matricial.
 9. Identificación de la falla: Determinación del tamaño y el comportamiento variante en el tiempo de la falla.
 10. Modelo cualitativo: Descripción de un sistema con relaciones entre variables del sistema y parámetros en términos heurísticos como: reglas SI-ENTONCES.
 11. Modelo cuantitativo: Descripción de un sistema con relaciones entre variables del sistema y parámetros en términos analíticos tales como: ecuaciones diferenciales.
 12. Modelo de diagnóstico: Conjunto de relaciones estáticas o dinámicas las cuales relacionan una variable de entrada y salida específicas.
 13. Monitoreo: Tarea continua, en tiempo real, que determina las condiciones físicas del sistema, mediante el registro de información, reconocimiento e indicación de anomalías en el comportamiento.
 14. Perturbación: Entrada desconocida que actúa sobre un sistema, provocando una desviación temporal del estado actual.
 15. Protección: Medio mediante el cual un comportamiento altamente peligroso del sistema, si es posible, es eliminado o la consecuencia de dicho comportamiento es evitada.

16. Redundancia analítica: Uso de dos o más (no necesariamente idénticas) formas de determinar una variable, donde una forma emplea el modelo matemático del proceso en forma analítica.
17. Residuo: Indicador de falla, basado en la desviación entre las mediciones y los cálculos basados en el modelo.
18. Seguridad: Habilidad de un sistema de no causar daño a personas, equipo o ambiente.
19. Síntoma: Cambio de una cantidad observable a partir del comportamiento normal.
20. Sistema de diagnóstico de fallas: Un sistema de monitoreo el cual es usado para detectar fallas así como diagnosticar su localización y significado en el sistema. Normalmente consiste de tres funciones básicas: Detección de falla, Aislamiento de falla e Identificación de falla.
21. Supervisión: Monitorco del sistema físico y toma de acciones apropiadas para mantener la operación en caso de falla.
22. Umbral: Valor límite de una desviación de residuo a partir de cero, tal que si se excede, una falla se declara como detectada.

Bibliografía

- [1] J. Chen and R.J. Patton, Robust Model-Based fault Diagnosis for Dynamic Systems, Boston: Kluwer Academic Publishers, 1999.
- [2] Ch. Edwards and S. K. Spurgeon, Sliding Mode Control: Theory and Applications, Bristol: Taylor & Francis Ltd, 1998.
- [3] [24] R. Isermann and P. Ballé, “Trends in the application of model-based fault detection and diagnosis of technical processes”, Control Eng. Practice, Vol. 5, No. 5, 1997, pp. 709-719.
- [4] R. Isermann, “Supervision, fault-detection and fault diagnosis methods-and introduction”, Control Eng. Practice, Vol. 5, No. 5, 1997, pp. 639-652.
- [5] J. de J. Rubio Ávila “Modelado y control para una clase de sistemas no lineales desconocidos en tiempo discreto” tesis del Centro De Investigación Y De Estudios Avanzados Del Instituto Politécnico Nacional, México, DF. 2007.
- [6] A.S. Willsky, “A survey of design methods for failure detection in dynamic system”, Automatica, Vol. 12, No. 6, 1976, pp 601-611.
- [7] T. Kohonen, Associative Memory. A system theoretical approach, Springer Verlag, 1977.
- [8] J. Hopfield. Networks with graded response have collective computational properties like those of two-state networks, Proceedings of the National Academic of Sciences, 3088-3092, 1982.

-
- [9] S. Z. Sarpturk, Y. Istefanopolos and O. Kaynak, On the Stability of Discrete-Time Sliding-Mode Control Systems, IEEE Transactions on Automatic Control, Vol.32, No.10, 930-932, October 1987.
- [10] Y. Fang, T. W. S. Chow and X. D. Li, Use of a Recurrent Neural Network in Discrete Sliding-Mode Control, IEE Proceeding-Control Theory Applications, Vol.146, No.1, 84-90, 1999.
- [11] K.S.Narendra and K.Parthasarathy, Identification and Control of Dynamical Systems Using Neural Networks, IEEE Trans. Neural Networks, Vol.1, No.1, 4-27, 1990.
- [12] J. Anzurez Marin “Diagnóstico de fallas en sistemas no lineales usando lógica difusa y observadores con modos deslizantes” tesis del Centro De Investigación Y De Estudios Avanzados Del Instituto Politécnico Nacional, México, DF. 2007.
- [13] A. Delgado. Inteligencia Artificial y Minirobots. Ecoe Ediciones. Santafé de Bogotá: 1998
- [14] J.J. Gertler, Fault Detection and Diagnosis in Engineering Systems, New York: Marcel Dekker, 1998.
- [15] R.J. Patton, P.M. Frank and R.N. Clark, Fault Diagnosis in Dynamic Systems: Theory and Applications, New York: Prentice Hall, 1989.
- [16] R. J. Patton, P. M. Frank and R. N. Clark, Issues of Fault Diagnosis for Dynamic systems, New York: Springer-Verlag, 2000.
- [17] L.A. Mironovski “Functional diagnosis of linear dynamic system- a survey”, Automation Remote Control, 1980, 41: pp. 1122-1143.
- [18] E.A. García y P.M. Frank, “Deterministic nonlinear observer-based approaches to fault diagnosis: a survey”, Contr. Eng. Practice Vol. 5, No. 5, 1997, pp. 663-670.

-
- [19] V. Krishnaswami, G. C. Luh and G. Rizzoni, "Nonlinear parity equation residual generation for diagnosis of automotive engine fault", *Contr. Eng. Practice* Vol. 3, No. 10, 1995, pp. 1385-1392.
- [20] J.J. Gertler, "Survey of model-based failure detection and isolation in complex plants", *IEEE Contr. Syst. Mag.* Vol. 8, No. 6, 1988, pp. 3-11.
- [21] P.M. Frank, "Fault Diagnosis in Dynamic Systems Using Analytical and Knowledge-Based Redundancy-A survey and some new results", *Automatica*, Vol. 26, No. 3, 1990, pp. 459-474.
- [22] A. Akhenak, M. Chadli, D. Maquin and J. Ragot, "Sliding mode multiple observer for fault detection and isolation", *Conference on Decision and Control, USA, 2003*, pp. 953-958.
- [23] W.C.Chan, C.W.Chan, K.C.Cheung and Y.Wang, Modeling of nonlinear stochastic dynamical systems using neurofuzzy networks, *38th IEEE Conference on Decision and Control, 2643-2648, Phoenix, USA, 1999*.
- [24] Yanghong Tan, Yigang He, Chun Cui, And Guanyuan Qiu "A Novel Method For Analog Fault Diagnosis Based On Neural Networks And Genetic Algorithms" *IEEE Transactions On Instrumentation And Measurement*, Vol. 57, No. 11, November 2008
- [25] Yan-bin Qu Yang Zhang "An Application of the Combination of Ant Colony Algorithm and Neural Network" *Proceedings of the 2006 IEEE International Conference on Information Acquisition August 20 - 23, 2006, Weihai, Shandong, China*
- [26] Qian-jin Guo, Hai-bin Yu, Ai-dong Xu "Modified Morlet Wavelet Neural Networks for Fault Detection" *International Conference on Control and Automation (ICCA2005) June 27-29, 2005, Budapest, Hungary*
- [27] Zhang Ren, Jie Chen, Xiaojing Tang, Weisheng Yan "A Combined Method Based on Neural Network for Control System Fault Detection and Diagnosis" *Proceedings of the*

-
- 2000 IEEE International Conference on Control Applications Anchorage, Alaska, USA
September 25-27, 2000
- [28] Tianqi Yang “Application of Neuro-fuzzy Network for Fault Diagnosis in an Industrial Process” 2007 IEEE International Conference on Control and Automation ThAP-16 Guangzhou, CHINA - May 30 to June 1, 2007
- [29] Gui-hong Zhou, Chun-cheng Zuo¹, Jia-zhong Wang, Shu-xia Liu “A New Fault Diagnosis Model Based on AIR Scheme” Proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC’06)
- [30] Giovanni Betta, Consolatina Liguori, Antonio Pietrosanto “An Advanced Neural-Network-Based Instrument Fault Detection and Isolation Scheme” IEEE Transactions On Instrumentation And Measurement, Vol. 47, No. 2, April 1998
- [31] Rastko R. Selmic, Frank L. Lewis “Multimodel neural networks identification and failure detection of non linear systems” Proceedings of the 40th IEEE Conference on Decision and Control Orlando, Florida USA, December 2001
- [32] A. Bernieri, M. D’Apuzzo, L. Sansone, M. Savastano “A Neural Network Approach for Identification and Fault Diagnosis on Dynamic Systems” IEEE Transactions On Instrumentation And Measurement, Vol. 43, No. 6 , December 1994
- [33] Sinnasamy R. Naidu, Evangelhos Zafiriou, Thomas J. McAvoy.”Use of Neural Networks for Sensor Failure Detection in a Control System” American Control Conference, Pittsburgh, Pennsylvania, June 21-23, 1989.
- [34] Robert F. Stengel “Intelligent Failure-Tolerant Control” Symposium on Intelligent Control, Philadelphia, PA, Sept. 5-7, 1990.
- [35] George Bolt “Fault Models for Artificial Neural Networks” Proceedings of the IJCNN, Singapore. pp.1373-1378 (1991)

-
- [36] S. Ebron, D. L. Lubkeman, M. White "A Neural Network Approach To The Detection Of Incipient Faults On Power Distribution Feeders" Electric Power Research Center College of Engineering North Carolina State University Raleigh, NC 27695-7911. IEEE Transactions on Power Delivery, Vol. 5, No. 2, April 1990
- [37] Ghendy Cardoso, Jr., Jacqueline Gisèle Rolim, Hans Helmut Zürn "Application of Neural-Network Modules to Electric Power System Fault Section Estimation" IEEE Transactions On Power Delivery, Vol. 19, No. 3, July 2004
- [38] LIU Ai-min, LIN Xin , LIU Xiang-dong "Fault Diagnosis Method of High Voltage Circuit Breaker based on (RBF) Artificial Neural Network" 2005 IEEE/PES Transmission and Distribution Conference & Exhibition: Asia and Pacific Dalian, China
- [39] Mohamed A. Awadallah, Medhat M. Morcos "Application of AI Tools in Fault Diagnosis of Electrical Machines and Drives—An Overview" IEEE Transactions On Energy Conversion, Vol. 18, No. 2, June 2003
- [40] N. Kandil, V.K. Sood, K. Khorasani, R.V. Pate1 "Fault Identification In An Ac-Dc Transmission System Using Neural Networks" Transactions on Power Systems, Vol. 7, No. 2, May 1992.
- [41] Masato Togami, Norihiro Abe, Tadahiro Kitahashi, Harunao Ogawa "On the Application of a Machine Learning Technique to Fault Diagnosis of Power Distribution Lines" IEEE Transactions on Power Delivery, Vol. 10, No. 4, October 1995
- [42] T.S. Sidhu, H. Singh, M.S. Sachdev "Design, Implementation and Testing of An Artificial Neural Network Based Fault Direction Discriminator for Protecting Transmission Lines" IEEE Transactions on Power Delivery, Vol. 10, No. 2, April 1995
- [43] Rodrigo Hartstein Salim, Karen Rezende Caino De Oliveira, André Darós Filomena, Mariana Resener, Arturo Suman Bretas "Hybrid Fault Diagnosis Scheme Implementation Power Distribution Systems Automation" IEEE Transactions On Power Delivery, Vol. 23, No. 4, October 2008

-
- [44] A. G. Jongepier, L. van der Sluis “Adaptive Distance Protection of Double-Circuit Lines using Artificial Neural Networks” IEEE Transactions on Power Delivery, Vol. 12, No. 1, January 1997
- [45] M.M. Tawfik, M.M. Morcos “Novel Approach for Fault Location on Transmission Lines” IEEE Power Enganeenng Review, November 1998
- [46] D. V. Coury, D. C. Jorge “Artificial Neural Network Approach to Distance Protection of Transmission Lines” IEEE Transactions on Power Delivery, Vol. 13, No. 1, January 1998
- [47] L. Sousa Martins, J. F. Martins, C. M. Alegria, V. Femilo Pires, “A Network Distribution Power System Fault Location Based on Neural Eigenvalue Algorithm” Paper accepted for presentation at 2003 IEEE Bologna PowerTech Conference, June 23-26, Bologna, Italy
- [48] Sunan N. Huang and Kok Kiang Tan “Fault Detection, Isolation, and Accommodation Control in Robotic Systems” IEEE Transactions On Automation Science And Engineering, Vol. 5, No. 3, July 2008
- [49] Hongmei Liu, Pingechao Ouyang, Shaoping Wang “Fault Detection Based on RBF Neural Network in a Hydraulic Position Servo System” Proceedings of the 6th World Congress on Intelligent Control and Automation, June 21 - 23, 2006, Dalian, China
- [50] Elias N. Skoundrianos, Spyros G. Tzafestas “Fault Diagnosis on the Wheels of a Mobile Robot Using Local Model Neural Networks” IEEE Robotics & Automation Magazine September 2004
- [51] Puneet Goel, Goksel Dedeoglu, Stergios I. Roumeliotis, Gaurav S. Sukhatme “Fault Detection and Identification in a Mobile Robot Using Multiple Model Estimation and Neural Network” Proceedings of the 2000 IEEE International Conference on Robotics & Automation San Francisco, CA April 2000

-
- [52] Arun T. Vemuri, Marios M. Polycarpou, Sotiris A. Diakourtis “Neural Network Based Fault Detection in Robotic Manipulators” *IEEE Transactions On Robotics And Automation*, Vol. 14, No. 2, April 1998
- [53] Arun T. Vemuri, Marios M. Polycarpou “Neural-Network-Based Robust Fault Diagnosis in Robotic Systems” *IEEE Transactions On Neural Networks*, Vol. 8, No. 6, November 1997
- [54] MO-yuen Chow, Robert N. Sharpe, and James C. Hung, “On the Application and Design of Artificial Neural Networks for Motor Fault Detection-Part I” *IEEE Transactions On Industrial Electronics*, Vol. 40, No. 2, April 1993
- [55] MO-yuen Chow, Robert N. Sharpe, and James C. Hung, “On the Application and Design of Artificial Neural Networks for Motor Fault Detection-Part II” *IEEE Transactions On Industrial Electronics*, Vol. 40, No. 2, April 1993
- [56] MO-Yuen Chow, Sui Oi Yee “Methodology For On-Line Incipient Fault Detection In Single-Phase Squirrel-Cage Induction Motors Using Artificial Neural Networks” *IEEE/PES Winter Meeting*, New York, 1991
- [57] Qinglin Guo, Qi Tang “Data Mining based on Improved Neural Network and its Application in Fault Diagnosis of Steam Turbine” *Proceedings of the 7th World Congress on Intelligent Control and Automation* June 25 - 27, 2008, Chongqing, China
- [58] Kevin M. Passino, Michael A. Sartori, and Panos J. Antsaklis. “Neural Computing for Numeric-to-Symbolic Conversion In Control Systems”, *IEEE Control Systems Magazine*, April 1989.
- [59] I. D’Antone “A Neural Network In An Expert Diagnostic System” *Nuclear Science, IEEE Transactions on* Volume 39, Issue 2, Apr 1992 Page(s):58 – 62
- [60] Hak-yeong Chung, Zeungnam Bien, Joo-hyun Park, Poong-hyun Seong “Incipient Multiple Fault Diagnosis in Real Time with Application to Large- Scale Systems” *IEEE Transactions On Nuclear Science*, Vol. 41, No. 4 August 1994

-
- [61] Keehoon Kim, Eric B. Bartlett “Nuclear Power Plant Fault Diagnosis Using Neural Networks with Error Estimation by Series Association” IEEE Transactions On Nuclear Science, Vol. 43, No. 4. August 1996
- [62] YangShulian, Li Wenhai ,Zhen Hua, Xiang Fang “Intelligent condition monitoring and fault diagnosis of a gearbox based on Artificial Neural Network” The Eighth International Conference on Electronic Measurement and Instruments ICEMI’2007
- [63] Daniel Fischer, Eberhard Kaus, Rolf Isermann “Fault Detection For An Active Vehicle Suspension” Proceedings of the American Control Conference Denver. Colorado June 4-6, 2003
- [64] Shubhalaxmi Kher, P.K. Chande, P.C.Sharma “Automobile Engine Fault Diagnosis Using Neural Network” 2001 IEEE Intelligent Transportation Systems Conference Proceedings - Oakland (CA), USA - August 25-29, 2001
- [65] Andrea Bernieri, Giovanni Betta, Consolatina Liguori “On-Line Fault Detection and Diagnosis Obtained by Implementing Neural Algorithms on a Digital Signal Processor” IEEE Transactions On Instrumentation And Measurement, Vol. 45, No 5, October 1996
- [66] K. Rahnamai, P. Arabshahi, T.-Y. Yan, T. Pham, S. G. Finley “An Intelligent Fault Detection and Isolation Architecture For Antenna Arrays” TDA Progress Report 42-132 February 15, 1998
- [67] Mohamed A. H. El-Sayed, Abdullah S. Alfuhaid “Ann-Based Approach For Fast Fault Diagnosis And Alarm Handling Of Power Systems” Proceedings of the 5th International Conference on Advances in Power System Control. Operation and Management, APSCOM 2000. Hong Kong, October 2000.
- [68] Dong Xu, Mei Wu, and Jinwen An “Design of An Expert System Based on Neural Network Ensembles for Missile Fault Diagnosis” Proceedings of the 2003 IEEE International Conference on Robotics, Intelligent Systems and Signal Processing Changsha, China - October 2003

-
- [69] Neil H. Eklund and Kai F. Goebel “Using Neural Networks and the Rank Permutation Transformation to Detect Abnormal Conditions in Aircraft Engines” 2005 IEEE Mid-Summer Workshop on Soft Computing in Industrial Applications Helsinki University of Technology, Espoo, Finland, June 28-30, 2005
- [70] Mohammad Azam, Krishna Pattipati, Jeffrey Allanach, Scott Poll, Ann Patterson-Hine “In-flight Fault Detection and Isolation in Aircraft Flight Control Systems “ IEEEAC paper #1429, Version 6, Updated December 8, 2004
- [71] Z. Q. Li, L. Ma, K. Khorasani ”A Dynamic Neural Network-based Reaction Wheel Fault Diagnosis for Satellites” 2006 International Joint Conference on Neural Networks Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada July 16-21, 2006
- [72] H.A. Talebi and R.V. Patel “An Intelligent Fault Detection and Recovery Scheme for Reaction” Wheel Actuator of Satellite Attitude Control Systems” Proceedings of the 2006 IEEE International Conference on Control Applications Munich, Germany, October 4-6, 2006
- [73] Iz Al-Dein, Al-Zyoud, K. Khorasani “Neural Network-based Actuator Fault Diagnosis for Attitude Control Subsystem of a Satellite” World Automation Congress (WAC) 2006, July 24-26, Budapest, Hungary
- [74] Davor Linaric, Vladimir Koroman “Fault Diagnosis of a Hydraulic Actuator using Neural Network” ICIT 2003 - Maribor, Slovenia
- [75] Hua Zong and Yun Zhang “A Real-time Recognition of Working Patterns to Fault Diagnosis Based on BP Neural Network” Proceedings of the 6th World Congress on Intelligent Control and Automation, June 21 - 23, 2006, Dalian, China
- [76] S. N. Huang, K. K. Tan, T. H. Lee, “Automated Fault Detection and Diagnosis in Mechanical Systems” IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews, Vol. 37, No. 6, November 2007

-
- [77] Surin Khomfoi and Leon M. Tolbert “Fault Diagnostic System for a Multilevel Inverter Using a Neural Network” *IEEE Transactions On Power Electronics*, Vol. 22, No. 3, May 2007
- [78] Narendra Bawane, A.G. Kothari “Artificial Neural Network Based Fault Identification Of HVDC Converter” *SDEMPED 2003 -Symposium on Diagnostics for Electric Machines, Power Electronics and Drives Atlanta, CA, USA, 24-26 August 2003*
- [79] Farzan Aminian, Mehran Aminian, H. W. Collins, Jr. “Analog Fault Diagnosis of Actual Circuits Using Neural Networks” *IEEE Transactions On Instrumentation And Measurement*, Vol. 51, No. 3, June 2002
- [80] Deng Ying, He Yigang “On the Application of Artificial Neural Networks to Fault Diagnosis in Analog Circuits with Tolerances” *Proceedings of ICSP2000*
- [81] Haiying Yuan, Guangju Chen, Sanbao Shi, Huawei Chen “Research on Fault Diagnosis in Analog Circuit Based on Wavelet-Neural Network” *Proceedings of the 6th World Congress on Intelligent Control and Automation, June 21 - 23, 2006, Dalian, China*
- [82] M. Tripathy, R.P. Maheshwari and H.K. Verma “Probabilistic neural-network-based protection of power transformer” *IET Electr. Power Appl.*, 2007, 1, (5), pp. 793–798
- [83] Ying-Jun Guo, Li-Hua Sun, Yong-Chun Liang, Hai-Chao Ran, Hui-Qin Sun “The Fault Diagnosis Of Power Transformer Based On Improved RBF Neural Network” *Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 19-22 August 2007*
- [84] Jing Luo, Ping-Chen Zai, Yun-Ni Jian “Fault Diagnosis Of Power Transformer Based On Ellipsoidal Basis Functional Neural Network” *Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition, Beijing, China, 2-4 Nov. 2007*
- [85] Xiong Hao And Sun Cai-Xin “Artificial Immune Network Classification Algorithm For Fault Diagnosis Of Power Transformer” *IEEE Transactions On Power Delivery*, Vol. 22, No. 2, April 2007

-
- [86] F.Charfi, F.Sellami, K. Al-Haddad “Fault Diagnostic in Power System Using Wavelet Transforms and Neural Networks” IEEE ISIE 2006, July 9-12, 2006, Montreal, Quebec, Canada
- [87] Yann-Chang Huang “Evolving Neural Nets for Fault Diagnosis of Power Transformers” IEEE Transactions On Power Delivery, Vol. 18, No. 3, July 2003
- [88] A. S. Farag, M. Mohandes, Ali Al-Shaikh “Diagnosing Failed Distribution Transformers Using Neural Networks” IEEE Transactions On Power Delivery, Vol. 16, No. 4, October 2001
- [89] Li Honglei, Xiao Dengming and Chen Yazhu “Wavelet ANN Based Transformer Fault Diagnosis” Proceedings of The 6th International Conference on Properties and Applications of Dielectric Materials June 21-26,2000, Xi’an Jiaotong University, Xi’an, China
- [90] Zhenyuan Wang, Yilu Liu, Paul J. Griffin “Neural Net and Expert Systems Diagnose Transformer Faults” IEEE Computer Applications in Power, January 2000
- [91] Yunosuke Maki, Kenneth A. Loparo “A Neural-Network Approach to Fault Detection and Diagnosis in Industrial Processes” IEEE Transactions On Control Systems Technology, Vol. 5, No. 6, November 1997
- [92] A. I. Taalab, H. A. Darwish, T. A. Kawady “Ann-Based Novel Fault Detector For Generator Windings Protection” IEEE Transactions on Power Delivery, Vol. 14, No. 3, July 1999
- [93] Hatem A. Darwish, Abdel-Maxoud I. Taalab, Tamer A. Kawady “Development and Implementation of an ANN-Based Fault Diagnosis Scheme for Generator Winding Protection” IEEE Transactions On Power Delivery, Vol. 16, No. 2, April 2001
- [94] Shing Chiang Tan, Chee Peng Lim “Application of an Adaptive Neural Network With Symbolic Rule Extraction to Fault Detection and Diagnosis in a Power Generation Plant” IEEE Transactions On Energy Conversion, Vol. 19, No. 2, June 2004

-
- [95] Yong-Guang Ma, Liang-Yu Ma, Jin Ma “RFB Neural Network Based Fault Diagnosis For The Thermodynamic System Of A Thermal Power Generating Unit“ Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005.
- [96] Jose A. Ruz-Hernandez, Edgar N. Sanchez, Dionisio A. Suarez “Neural Networks-based Scheme for Fault Diagnosis in Fossil Electric Power Plants” Proceedings of International Joint Conference on Neural Networks, Montreal, Canada, July 31 - August 4, 2005
- [97] Zhong Liu, Jianzhong Zhou, Min Zou, Yongchuan Zhang, Liangliang Zhan “A New Method for Intelligent Fault Diagnosis of Hydroelectric Generating Unit” 2007 IEEE International Conference on Control and Automation Guangzhou, CHINA - May 30 to June 1, 2007.
- [98] Hung T. Nguyen: A First Course in fuzzy and Neural Control. Chapman & Hall CRC; ISBN 1-58488-1, USA, 2002.
- [99] Andres Ferreyra Ramirez; tesis”Redes Neuronales Difusas para modelado via agrupamiento en-línea: aplicación a un condensador de aspiración” Centro De Investigación Y De Estudios Avanzados Del Instituto Politécnico Nacional Departamento De Control Automático, México, D.F., 2005.
- [100] Hassan. K. Khalil, “Nolinear systems”, New Jersey, Prentice Hall, 2002.
- [101] Juan Reyes Reyes “Pasividad Robusta de Sistemas No Lineales via Redes Neuronales Dinámicas” tesis del Centro De Investigación Y De Estudios Avanzados Del IPN, 2007.
- [102] Jagannathan Sarangapani of the University of Missouri "Neural Network Control of Nonlinear Discrete-Time Systems" Taylor & Francis Group, LLC 2006.
- [103] J. A. Anderson and E. Rosenfeld "Neurocomputing Foundation of Research", Cambridge:The MIT Press, 1988

-
- [104] J. S. Denker "Neural Networks for Computing", New York; American Institute of Physics, 1986.
- [105] Haykin S, Neural Networks. A Comprehensive Foundation.- New Jersey: Prentice Hall, 1999.
- [106] S.Haykin, Neural Networks- A Comprehensive Foundation, Macmillan College Publ. Co., New York, 1994.
- [107] Patan K. Korbics J.: Artificial Neural Network to fault diagnosis.- Wydawnictwa Naukowo-Techniczne WNT, Polonia, 2002.
- [108] M. Norgaard; Neural Networks for modelling and control of Dynamic Systems; Springer; ISBN1-85233-227-1, Glasgow, Scotland, U.K., 2000.
- [109] Abhijit S. Pandya,Robert B. Macy "Pattern Recognition with Neural network in C++". CRC press. Florida Atlantic University.
- [110] T. Kohonen, The Self-Organizing Map, Proceedings of the IEEE, Vol. 78, No. 9, 1464-1480, 1990.
- [111] P. j. Werbos, Backpropagation through time:What it does and how to do it, Proceedings of the IEEE, Vol. 78, No. 10, 1550-1560, 1990.
- [112] Behnam Bavarian. "Introduction to Neural Networks for Intelligent Control" IEEE International Conference on Neural Networks, San Diego, California, June 21-24, 1987.
- [113] FESTO compact unit PCS, online information. Available: <http://www.festo.com>.
- [114] M. K. Passino, S. Yurkovich, Fuzzy Control, Addison-Wesley, USA, 1998.
- [115] K. Tanaka, H. O. Wang "Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach" , John Wiley & Sons, Inc. 2001.

- [116] Yang Xiaoqing, Cai Tao, Chen Jie “Process of Threshold of Fault Diagnosis Expert System in LAN Based on BP Neural Network “ Proceedings of the 27th Chinese Control Conference July 16-18, 2008, Kunming, Yunnan, China.
- [117] Bose, B. K., “Power electronics and variable frequency drives technology and applications,” Proc. IEEE INT. SYMP., Budapest, pp. 559-627, 1993.
- [118] http://www2.uca.es/dept/leng_sist_informaticos/preal/23041/transpas/E-Backpropagation/ppframe.htm