



**UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO**

FACULTAD DE INGENIERÍA

**DIAGNÓSTICO DE FALLAS DE UN SISTEMA DINÁMICO
UNIVARIABLE CON BASE EN REDES NEURONALES NO
SUPERVISADAS**

T E S I S
QUE PARA OBTENER EL TÍTULO DE:
INGENIERO ELÉCTRICO ELECTRÓNICO
(MODULO DE SISTEMAS DIGITALES)
P R E S E N T A:
EDUARDO PÉREZ GUZMÁN

DIRECTOR DE TESIS:
DR. HÉCTOR BENÍTEZ PÉREZ



MÉXICO, D.F.

2003



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

ÍNDICE GENERAL

PRÓLOGO ¡ERROR! MARCADOR NO DEFINIDO.

1 GENERALIDADES ¡ERROR! MARCADOR NO DEFINIDO.

1.1 INTRODUCCIÓN ¡ERROR! MARCADOR NO DEFINIDO.

1.2 IDENTIFICACIÓN MEDIANTE ESTIMACIÓN DE PARÁMETROS ¡ERROR! MARCADOR NO DEFINIDO.

1.2.1 INTRODUCCIÓN A LA IDENTIFICACIÓN DE SISTEMAS DINÁMICOS **¡ERROR! MARCADOR NO DEFINIDO.**

1.2.2 MODELOS DE SISTEMAS DINÁMICOS LINEALES E INVARIANTES EN EL TIEMPO **¡ERROR! MARCADOR NO DEFINIDO.**

1.2.3 MÉTODOS PARAMÉTRICOS PARA LA IDENTIFICACIÓN DE SISTEMAS **¡ERROR! MARCADOR NO DEFINIDO.**

1.3 VARIABLES DE ESTADO ¡ERROR! MARCADOR NO DEFINIDO.

1.4 ESTIMACIÓN DE ESTADOS ¡ERROR! MARCADOR NO DEFINIDO.

1.4.1 INTRODUCCIÓN A LA ESTIMACIÓN DE ESTADOS **¡ERROR! MARCADOR NO DEFINIDO.**

1.4.2 OBSERVADORES **¡ERROR! MARCADOR NO DEFINIDO.**

1.4.3 EL FILTRO DE KALMAN **¡ERROR! MARCADOR NO DEFINIDO.**

1.5 REDES NEURONALES ¡ERROR! MARCADOR NO DEFINIDO.

1.5.1 INTRODUCCIÓN A LAS REDES NEURONALES **¡ERROR! MARCADOR NO DEFINIDO.**

1.5.2 REDES TIPO SOFM **¡ERROR! MARCADOR NO DEFINIDO.**

1.5.3 LA RED DE KOHONEN **¡ERROR! MARCADOR NO DEFINIDO.**

1.6 RESUMEN ¡ERROR! MARCADOR NO DEFINIDO.

1.7 REFERENCIAS ¡ERROR! MARCADOR NO DEFINIDO.

2 PROPUESTA DE DIAGNÓSTICO DE FALLAS ¡ERROR! MARCADOR NO DEFINIDO.

2.1 INTRODUCCIÓN ¡ERROR! MARCADOR NO DEFINIDO.

2.2 DIAGNÓSTICO DE FALLAS ¡ERROR! MARCADOR NO DEFINIDO.

2.3	MODELO DE DIAGNÓSTICO DE FALLAS EXPERIMENTAL	¡ERROR! MARCADOR NO DEFINIDO.
2.4	CONCLUSIONES	¡ERROR! MARCADOR NO DEFINIDO.
2.5	REFERENCIAS	¡ERROR! MARCADOR NO DEFINIDO.
3	<u>MODELO DE ESTUDIO</u>	¡ERROR! MARCADOR NO DEFINIDO.
3.1	INTRODUCCIÓN	¡ERROR! MARCADOR NO DEFINIDO.
3.2	ELABORACIÓN DEL EXPERIMENTO Y OBTENCIÓN DE DATOS	¡ERROR! MARCADOR NO DEFINIDO.
3.2.1	DESCRIPCIÓN GENERAL DEL MODELO Y LOS EXPERIMENTOS	¡ERROR! MARCADOR NO DEFINIDO.
3.2.2	CARACTERIZACIÓN DE LOS ELEMENTOS	¡ERROR! MARCADOR NO DEFINIDO.
3.2.3	CARACTERIZACIÓN DEL SISTEMA	¡ERROR! MARCADOR NO DEFINIDO.
3.3	OBTENCIÓN DEL MODELO ANALÍTICO, EL MODELO ARMAX	¡ERROR! MARCADOR NO DEFINIDO.
3.4	ESTIMACIÓN DEL FILTRO DE KALMAN	¡ERROR! MARCADOR NO DEFINIDO.
3.5	IMPLANTACIÓN DE LA RED NEURONAL, UNA RED SOM	¡ERROR! MARCADOR NO DEFINIDO.
3.5.1	INTRODUCCIÓN A LOS ALGORITMOS DE LA RED SOM.	¡ERROR! MARCADOR NO DEFINIDO.
3.5.2	PROPAGACIÓN DE LA SEÑAL.	¡ERROR! MARCADOR NO DEFINIDO.
3.5.3	APRENDIZAJE Y ENTRENAMIENTO.	¡ERROR! MARCADOR NO DEFINIDO.
3.6	CONCLUSIONES	¡ERROR! MARCADOR NO DEFINIDO.
3.7	REFERENCIAS	¡ERROR! MARCADOR NO DEFINIDO.
4	<u>RESULTADOS</u>	¡ERROR! MARCADOR NO DEFINIDO.
4.1	INTRODUCCIÓN	¡ERROR! MARCADOR NO DEFINIDO.
4.2	VALORACIÓN DEL MODELO ANALÍTICO DEL SISTEMA Y EL FILTRO DE KALMAN	¡ERROR! MARCADOR NO DEFINIDO.
4.3	RESULTADOS DEL ENTRENAMIENTO DE LA RED NEURONAL	¡ERROR! MARCADOR NO DEFINIDO.
4.4	COMPORTAMIENTO DEL SISTEMA DE DIAGNÓSTICO DE FALLAS	¡ERROR! MARCADOR NO DEFINIDO.
5	<u>CONCLUSIONES</u>	¡ERROR! MARCADOR NO DEFINIDO.
A	<u>APÉNDICE</u>	¡ERROR! MARCADOR NO DEFINIDO.
A.1	ECUACIONES DEL FILTRO DE KALMAN	¡ERROR! MARCADOR NO DEFINIDO.
B	<u>APÉNDICE</u>	¡ERROR! MARCADOR NO DEFINIDO.

B.1 CÓDIGO DE LOS PROGRAMAS

B.1.1 IDENTIFICACIÓN DEL SISTEMA

B.1.2 FILTRO DE KALMAN

B.1.3 RED NEURONAL

¡ERROR! MARCADOR NO DEFINIDO.

¡ERROR! MARCADOR NO DEFINIDO.

¡ERROR! MARCADOR NO DEFINIDO.

¡ERROR! MARCADOR NO DEFINIDO.

BIBLIOGRAFÍA GENERAL

¡ERROR! MARCADOR NO DEFINIDO.

PRÓLOGO

Las instalaciones de manufactura modernas son a gran escala, extremadamente complejas y operan con una gran número de variables de control en lazo cerrado. La detección y diagnóstico de fallas temprana y certera para estas plantas puede minimizar el tiempo de baja de los procesos, incrementar la seguridad en las operaciones de la planta y reducir los costos de manufactura. Las plantas se han vuelto altamente instrumentadas, resultando en la existencia de más información útil en la detección y diagnóstico de fallas. Esto ha propiciado el surgimiento de esfuerzos académicos e industriales concentrados en el desarrollo de métodos de monitoreo más efectivos.

Este texto comprende dos objetivos principales: desarrollar un modelo de diagnóstico de fallas con un procesamiento heurístico basado en una red neuronal e implantar este modelo experimentalmente en un sistema mecánico como caso de estudio.

Es un modelo de diagnóstico de fallas un proceso de monitoreo. El objetivo de los procesos de monitoreo es asegurar el éxito de las operaciones planeadas mediante el reconocimiento de comportamientos anómalos. La información no sólo mantiene al operador de la planta y el personal de mantenimiento mejor informado, sino que los asiste en tomar acciones de remedio apropiadas para eliminar los comportamientos anormales.

En el primer capítulo se expone la teoría necesaria tanto en el desarrollo del modelo de diagnóstico de fallas como en el análisis del sistema mecánico propuesto. Los sistemas que se estudian son lineales e invariantes en el tiempo por su mayor uso en la descripción de

sistemas dinámicos físicos dada su menor complejidad matemática. Dentro de los sistemas que se estudian se consideran aquellos de parámetros concentrados, sin embargo, en el desarrollo de algunos temas se utilizan modelos que por la generalidad que representan son de parámetros variables en el tiempo.

Los sistemas a discutir y sus estructuras se estudian en el dominio del tiempo discreto, ya que la utilidad de los modelos obtenidos esta en la implantación de algoritmos a procesarse sobre plataformas digitales. Se presentan algunas referencias para casos en el tiempo continuo que permiten un mejor entendimiento de algunos de los conceptos desarrollados.

La descripción funcional del modelo de diagnóstico de fallas se expone en el capítulo 2. En éste se describen los elementos presentes en el algoritmo de diagnóstico y a partir de esquemas se expone la forma de cómo la información fluye y resulta útil en el proceso de monitoreo. En el capítulo 3 se describe el desarrollo experimental necesario para la caracterización de los elementos del sistema mecánico y del modelo de diagnóstico; es este capítulo la conclusión práctica de los conceptos desarrollados en el primer capítulo.

El capítulo 4 evalúa el desarrollo experimental analizando resultados. Esta evaluación de resultados se realiza elemento por elemento y de forma global. La evaluación global consiste en observar el comportamiento interactivo de los elementos con base en del modelo propuesto para el diagnóstico de fallas. Finalmente, el capítulo 5 resume las conclusiones generales del proyecto; en ellas no solo se concluye respecto a los objetivos alcanzados, sino que se abordan algunos resultados no satisfactorios reflexionando sobre las circunstancias que pudieron originarlos por considerarse útiles en futuras aplicaciones.

1 GENERALIDADES

1.1 Introducción

En este capítulo se exponen los conceptos básicos involucrados en la obtención de modelos para la descripción de sistemas dinámicos y su representación matemática, con el objeto de proporcionar un panorama general y completo que permita justificar los procedimientos a realizarse en el desarrollo práctico. Con el mismo objetivo se desarrollan temas que por su contenido son necesarios para el entendimiento de los procesos involucrados en el tema central del trabajo: la implantación de un algoritmo para el diagnóstico de fallas.

En la sección 1.2 se discute el tema de identificación de sistemas mediante estimación de parámetros con base en modelos en ecuaciones en diferencias. El desarrollo de modelos en variables de estado es discutido brevemente en la sección 1.3 como introducción a la sección 1.4 en la cual, se aborda el tema de los observadores y como caso particular el *Filtro de Kalman*.

Finalmente, en la sección 1.5 se expone una descripción general de lo que es una red neuronal artificial: sus principales características y utilidades. Se presenta de manera más amplia la estructura general y principios de las redes tipo **SOM** (*Self-Organizing Maps*) incluida la red de *Kohonen*, fundamental para el modelo de diagnóstico de fallas propuesto.

1.2 Identificación mediante Estimación de Parámetros

1.2.1 Introducción a la Identificación de Sistemas Dinámicos

Los métodos de síntesis comúnmente usados en la teoría de control se basan en el conocimiento del modelo matemático del sistema que se pretende controlar. A la determinación de dicho modelo a partir del conocimiento previo sobre el proceso y de experiencias básicas se le conoce con el nombre de *identificación*. [1] [2]

Es importante hacer notar que la determinación de dicho modelo, al que se denominará como *analítico*, no es necesariamente la representación exacta del sistema, sino que es una aproximación adecuada al tipo de procesamiento que de éste se requiera y cuya determinación depende del diseñador. A partir de ello resulta fácil inferir la existencia de diversos modelos analíticos para un mismo sistema, tantos como criterios que el diseñador desee considerar y herramientas que utilice.

Aníbal Ollero [3] considera dos enfoques diferentes para la construcción del modelo:

1. Mediante la determinación de las ecuaciones y parámetros siguiendo exclusivamente las leyes físicas que rigen su comportamiento.
2. Con una identificación experimental, considerando al sistema como una *caja negra* con determinadas entradas y salidas, cuyas mediciones determinaran la evolución del sistema y su comportamiento.

Con respecto al primer enfoque, basta notar la complejidad que implica la determinación de todas las leyes físicas que intervienen en un proceso, así como la dificultad que representa la medición de las características físicas de los elementos que intervienen en el mismo. Realizar estas mediciones puede ser inaccesible en algunos casos dependiendo de la estructura física del sistema.

Por lo que concierne al segundo enfoque, la adopción de *hipótesis* sobre las características del sistema suele ser de vital importancia. Esta premisa es consecuencia de la incapacidad de deducir la evolución del sistema a partir de la evolución de sus elementos constitutivos. Para deducir la evolución del sistema, en la práctica se pueden aplicar las siguientes etapas:

1. *Etapas de Análisis*, en la cual se tienen en cuenta las leyes físicas principales involucradas en el sistema, con el fin de establecer hipótesis sobre su estructura matemática básica y sus propiedades.
2. *Etapas Experimentales*, en la cual se adoptan las hipótesis anteriormente obtenidas y se validan con información experimental.

Entre los criterios a observar en la elaboración de las hipótesis del sistema resulta importante hacer hincapié en el hecho de que elevar la complejidad del modelo no es sinónimo de incrementar su confiabilidad. Resulta importante tener en cuenta alternativas que nos permitan trabajar con un modelo más simple. Si el sistema es no lineal se puede pensar en adoptar un modelo lineal con variaciones relativamente pequeñas alrededor de un punto de operación. De la misma forma pueden utilizarse hipótesis simplificadoras para describir el comportamiento del sistema mediante un modelo de orden reducido cuya identificación sea más simple. En sistemas lineales con múltiples entradas, una opción es utilizar el *principio de superposición* y ante el problema de identificar modelos de parámetros variables en el tiempo, se puede considerar el utilizar aquellos de parámetros concentrados describiendo su variabilidad en el tiempo mediante la adición de perturbaciones como entradas suplementarias.

Hechas las aclaraciones anteriores, al hablar de modelos obtenidos mediante identificación en los siguientes capítulos de este texto, será de aquellos obtenidos bajo el segundo criterio, es decir, aquellos cuya evolución se deduce a partir del comportamiento de su salida con respecto al de su o sus entradas.

Ljung [4] menciona que la construcción del modelo para un sistema involucra tres entidades básicas:

- 1) Los datos.
- 2) Un conjunto de modelos candidatos.
- 3) Una regla, mediante la cual los modelos puedan ser validados usando los datos.

Desarrollando un poco más las entidades propuestas por Ljung,

1. *El registro de datos.* Los datos de entrada y salida generalmente son tomados en experimentos de identificación especialmente diseñados, donde el diseñador debe determinar qué señales medir y cuándo medirlas, así como las características de las señales de excitación. El objetivo de diseñar el experimento es el de poder escoger estos elementos de tal forma que los datos obtenidos contengan la mayor cantidad de **información** posible. Se exceptúa el caso en el cual el diseñador no pueda afectar el experimento y sea necesario usar información sobre la operación normal del sistema.
2. *El conjunto de modelos.* Esta es sin duda la más importante, y al mismo tiempo, la más difícil elección en el procedimiento de identificación. Es aquí donde el conocimiento *a priori*, la intuición y la percepción de la naturaleza de las cosas se deben combinar.
3. *Determinación de “el mejor” modelo del conjunto, de acuerdo a los datos.* Este es el **método de identificación**. La validación de la calidad del modelo está basada típicamente en cómo se desempeña el modelo al tratar de reproducir las mediciones obtenidas.

Finalmente, el proceso de identificación se sintetiza con el algoritmo descrito en la Figura 1.1.[4] Durante el proceso el modelo puede ser deficiente por diversas razones:

- El procedimiento matemático falló en encontrar un modelo de acuerdo a nuestro criterio.

- El criterio propuesto no es el adecuado.
- El conjunto de modelos no es apropiado dado que no contiene suficientes características para la descripción del sistema.
- El conjunto de datos recolectados no contiene la suficiente información para una validación adecuada de los modelos.

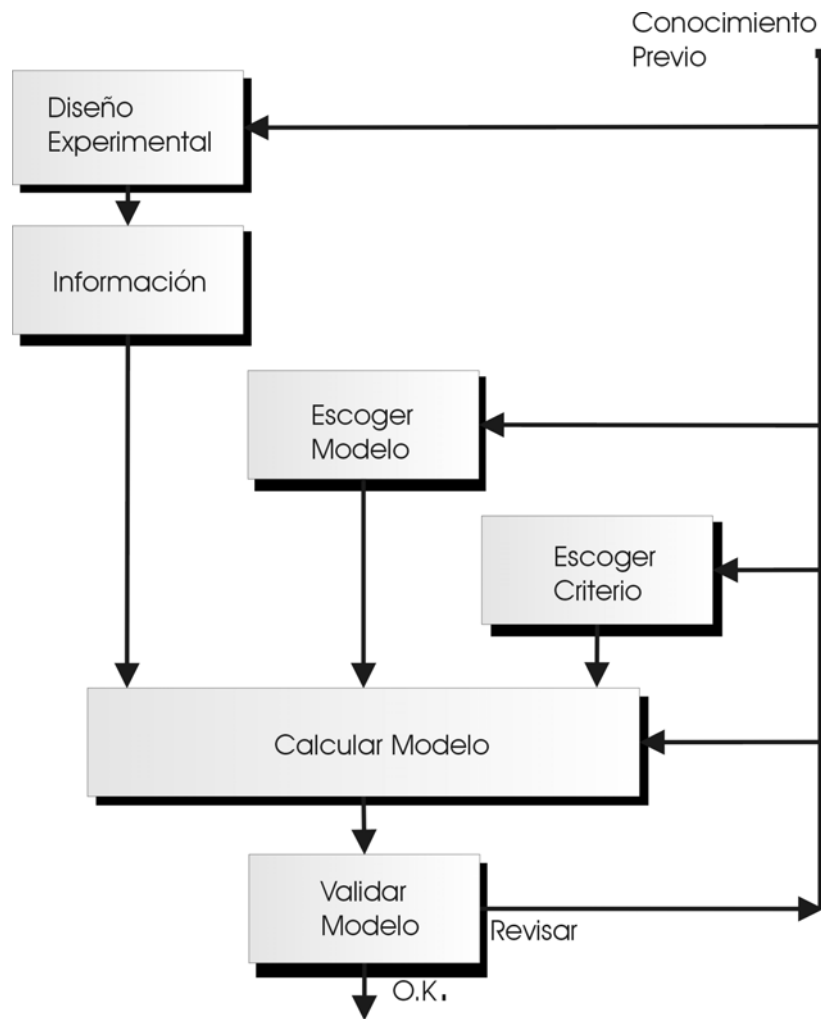


Figura 1.1 Algoritmo del proceso de Identificación.

Se concluye entonces, que el proceso de identificación consiste en la iteración de los elementos descritos en la Figura 1.1 pero sobre todo en el uso de un conocimiento subjetivo por parte del diseñador; cualidades como experiencia, intuición y capacidad de abstracción de los fenómenos permiten que un razonamiento lógico como el descrito en el proceso de

identificación converja con mayor prontitud en un modelo adecuado para un sistema en particular.

1.2.2 Modelos de Sistemas Dinámicos Lineales e Invariantes en el Tiempo

Existen diversos modelos para la representación de un sistema dinámico de acuerdo a sus características. Esencialmente un sistema dinámico lineal está completamente determinado por su función de transferencia $G(t)$ y alteraciones aditivas de la señal, dando como resultado la siguiente descripción básica,

$$y(t) = G(q)u(t) + H(q)e(t) \quad (1.1)$$

con

$$G(q) = \sum_{k=1}^{\infty} g(k)q^{-k}, H(q) = 1 + \sum_{k=1}^{\infty} h(k)q^{-k} \quad (1.2)$$

Generalmente no es posible conocer el valor de los coeficientes de los mecanismos físicos que rigen el comportamiento del sistema *a priori*. Esto conduce a que los coeficientes en cuestión entren en el modelo como **parámetros** a determinar. Denotando estos parámetros con el vector θ , obtenemos el siguiente modelo paramétrico,

$$y(t) = G(q, \theta)U(t) + H(q, \theta)e(t) \quad (1.3)$$

La ecuación anterior define un conjunto de modelos en función de θ cuyo estimador $\hat{y}(t|\theta)$ esta dado por la ecuación,

$$\hat{y}(t|\theta) = H^{-1}(q, \theta)G(q, \theta)u(t) + [1 - H^{-1}(q, \theta)]y(t) \quad (1.4)$$

La forma más directa de parametrizar G y H es representándolas como funciones racionales y permitiendo que los parámetros sean los coeficientes del numerador y denominador. Estas estructuras como se menciona con anterioridad, son conocidas como **modelos de caja negra**.

Probablemente la relación de entrada y salida más simple es la obtenida mediante la descripción como una ecuación en diferencias,

$$y(t) + a_1 y(t-1) + \dots + a_{n_a} y(t-n_a) = b_1 u(t-1) + \dots + b_{n_b} u(t-n_b) + e(t) \quad (1.5)$$

donde el término $e(t)$ es considerado ruido blanco y entra directamente como un error en la ecuación en diferencias. Este modelo es usualmente llamado **modelo de ecuación de error** cuyos parámetros ajustables son,

$$\theta = [a_1, a_2, \dots, a_{n_a}, b_1, \dots, b_{n_b}]^T \quad (1.6)$$

Mediante la siguiente sustitución matemática,

$$A(q) = 1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a}$$

$$B(q) = b_1 q^{-1} + \dots + b_{n_b} q^{-n_b}$$

se observa que (1.5) corresponde con (1.3) siendo,

$$G(q, \theta) = \frac{B(q)}{A(q)}, H(q, \theta) = \frac{1}{A(q)} \quad (1.7)$$

Esta estructura también es conocida por sus siglas en inglés como **modelo ARX**, donde **AR** hace referencia a la parte auto regresiva $A(q)y(t)$ y **X** a la entrada extra $B(q)u(t)$ llamada variable exógena.[4] El flujo de las señales puede representarse en un esquema como el de la Figura 1.2.

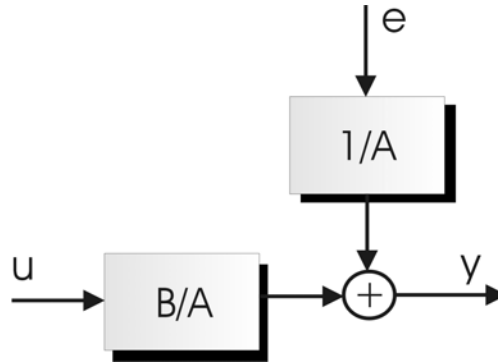


Figura 1.2 Estructura del modelo ARX.

Este modelo probablemente no es de lo más natural desde un punto de vista físico ya que en el se asume que el ruido blanco va a través del denominador A el cual determina la dinámica del sistema antes de ser añadido a la salida. Sin embargo, el modelo tiene una propiedad importante que lo hace la primera elección en muchos casos: su estimador define una regresión lineal. Este estimador está dado por la ecuación,

$$\hat{y}(t|\theta) = B(q)u(t) + [1 + A(q)]y(t) \quad (1.8)$$

obtenida de sustituir (1.7) en (1.4), e introduciendo el vector,

$$\varphi(t) = [-y(t-1), \dots, -y(t-n_a), u(t-1), \dots, u(t-n_b)]^T \quad (1.9)$$

entonces (1.8) puede ser rescrita de la siguiente forma,

$$\hat{y}(t|\theta) = \theta^T \varphi(t) = \varphi^T(t)\theta \quad (1.10)$$

Esta es la propiedad tan importante a la que se alude con anterioridad. El estimador es un producto escalar entre un vector conocido de datos $\varphi(t)$ y el vector de parámetros θ . Este modelo es conocido en estadística como regresión lineal y el vector $\varphi(t)$ como el vector de regresión. Su importancia radica entonces, en el hecho de que métodos de estimación muy poderosos y simples pueden ser aplicados para la determinación de θ .

La principal desventaja en la estructura del modelo **AR** es la carencia de una adecuada libertad para la descripción de las propiedades del término de las alteraciones $e(t)$. Se puede añadir cierta flexibilidad describiendo la ecuación de error como un promedio en movimiento del ruido blanco,

$$\begin{aligned} & y(t) + a_1 y(t-1) + \dots + a_{n_a} y(t-n_a) \\ & = b_1 u(t-1) + \dots + b_{n_b} u(t-n_b) + e(t) + c_1 e(t-1) + \dots + c_{n_c} e(t-n_c) \end{aligned} \quad (1.11)$$

con,

$$C(q) = 1 + c_1 q^{-1} + \dots + c_{n_c} q^{-n_c}$$

(1.11) puede ser rescrita como,

$$A(q)y(t) = B(q)u(t) + C(q)e(t) \quad (1.12)$$

que claramente corresponde a (1.4) con,

$$G(q, \theta) = \frac{B(q)}{A(q)}, H(q, \theta) = \frac{C(q)}{A(q)} \quad (1.13)$$

y,

$$\theta = [a_1, a_2, \dots, a_{n_a}, b_1, \dots, b_{n_b}, c_1, c_2, \dots, c_{n_c}]^T \quad (1.14)$$

Como consecuencia de la parte en movimiento $C(q)e(t)$ del modelo (*moving average*, **MA**), éste es llamado **modelo ARMAX**, sugerido por Aström y Bohlin (1965). Este modelo se ha convertido en una herramienta estándar en control y econometría tanto para la descripción de los sistemas como para el diseño de control.

El estimador para la Ecuación (1.12) es obtenido insertando (1.13) en (1.4), lo cual da,

$$\hat{y}(t|\theta) = \frac{B(q)}{C(q)}u(t) + \left[1 - \frac{A(q)}{C(q)}\right]y(t)$$

o,

$$C(q)\hat{y}(t|\theta) = B(q)u(t) + [C(q) - A(q)]y(t) \quad (1.15)$$

Esto significa que la estimación es obtenida filtrando u y y a través de un filtro cuya dinámica en el denominador está determinada por $C(q)$. Para empezar dicha estimación en el tiempo $t = 0$, es necesario el conocimiento de,

$$\begin{aligned} &\hat{y}(0|\theta) \dots \hat{y}(-n_c + 1|\theta) \\ &y(0) \dots y(-n^* + 1) \\ &u(0) \dots u(-n_b + 1) \end{aligned}$$

$$n^* = \max(n_c, n_a)$$

Si no están disponibles los datos, pueden ser tomados como cero, en cuyo caso la estimación difiere de la verdadera con un error que decae como $c \cdot \mu^t$, donde μ es la máxima magnitud de los ceros de $C(z)$. Es también posible comenzar la recursividad en el tiempo $\max(n^*, n_b)$ e incluir las condiciones iniciales desconocidas $\hat{y}(k|\theta), k = 1, \dots, n_c$, en el vector θ .

La estimación (1.15) puede ser rescrita en forma análoga a (1.10) como sigue. Agregando $[1 - C(q)]\hat{y}(t|\theta)$ en ambos lados de (1.15) dando como resultado,

$$\hat{y}(t|\theta) = B(q)u(t) + [1 - A(q)]y(t) + [C(q) - 1] [y(t) - \hat{y}(t|\theta)] \quad (1.16)$$

Introduciendo el error de estimación,

$$\varepsilon(t, \theta) = y(t) - \hat{y}(t | \theta)$$

y el vector,

$$\varphi(t, \theta) = [-y(t-1) \dots -y(t-n_b), u(t-1) \dots u(t-n_b), \varepsilon(t-1, \theta) \dots \varepsilon(t-n_c, \theta)]^T \quad (1.17)$$

Entonces (1.15) puede ser rescrita como,

$$\hat{y}(t | \theta) = \varphi^T(t, \theta) \theta \quad (1.18)$$

Se puede observar la similitud con la regresión lineal descrita en (1.10). La ecuación (1.18) no es en sí una regresión lineal, consecuencia del efecto no lineal de θ en el vector $\varphi(t, \theta)$. Para marcar la similitud con (1.10) se le puede denominar como una **regresión pseudolineal**. [4]

En lugar de modelar la ecuación de error (1.5) como un promedio en movimiento, como en (1.11), esta puede ser descrita como una auto regresión dando el modelo,

$$A(q)y(t) = B(q)y(t) + \frac{1}{D(q)}e(t) \quad (1.19)$$

con

$$D(q) = 1 + d_1 q^{-1} + \dots + d_{n_d} q^{-n_d}$$

el cual de manera análoga a la terminología anterior, puede ser llamado **modelo ARARX**. Es posible usar la descripción **ARMA** de la ecuación de error, y llevarla a una estructura "**ARARMAX**" más general,

$$A(q)y(t) = B(q)u(t) + \frac{C(q)}{D(q)}e(t) \quad (1.20)$$

la cual por supuesto contiene a (1.5), (1.11), (1.19) como casos particulares.

Estos forman entonces la familia de modelos relacionados a la ecuación de error descrita mediante la Figura 1.3, y sus relaciones con (1.4) así como las expresiones de sus estimadores resultan evidentes.

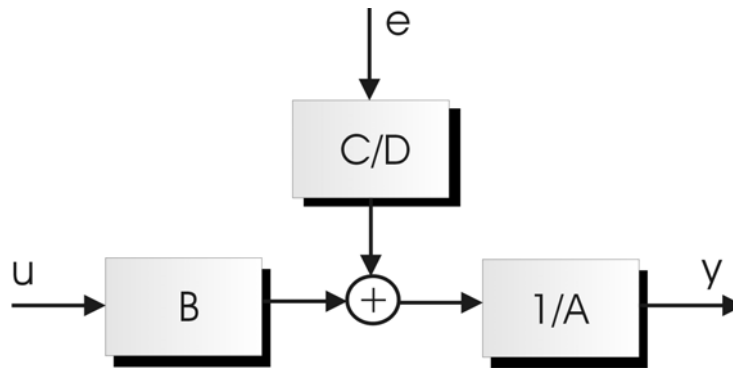


Figura 1.3 La familia del modelo de la ecuación de error.

Las estructuras para los modelos basados en la ecuación de error corresponden a descripciones donde las funciones de transferencia G y H tienen al polinomio A como factor común en sus denominadores. Desde un punto de vista físico resulta más natural parametrizar estas funciones de transferencia de manera independiente.

Si se supone que la relación entre la entrada y la salida sin alteraciones w , puede ser escrita como una ecuación en diferencias y que las alteraciones consisten en mediciones de ruido blanco, entonces obtenemos la siguiente descripción,

$$w(t) + f_1 w(t-1) + \dots + w_{n_f} y(t-n_f) = b_1(t-1) + \dots + b_{n_b} u(t-n_b) \quad (1.21)$$

$$y(t) = w(t) + e(t) \quad (1.22)$$

con,

$$F(q) = 1 + f_1 q^{-1} + \dots + f_{n_f} q^{-n_f}$$

de esta forma podemos escribir el modelo como,

$$y(t) = \frac{B(q)}{F(q)} u(t) + e(t) \quad (1.23)$$

con el flujo de las señales mostrado en la Figura 1.4.

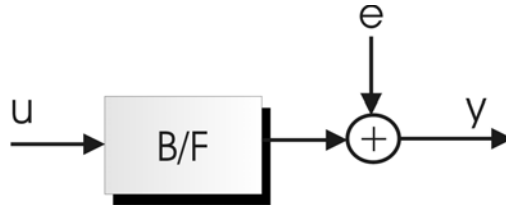


Figura 1.4 La estructura del modelo de error a la salida

Este modelo es llamado *ecuación de error a la salida (output error, OE)*, cuyo vector de parámetros a determinar es ,

$$\theta = [b_1, b_2, \dots, b_{n_b}, f_1, f_2, \dots, f_{n_f}]^T \quad (1.24)$$

Ya que $w(t)$ en (1.21) nunca es observada, en forma estricta debe llevar un índice θ , debido a que esta construida de u usando (1.21). Esto es,

$$w(t, \theta) + f_1 w(t-1, \theta) + \dots + f_{n_f} w(t-n_f, \theta) = b_1 u(t-1) + \dots + b_{n_b} u(t-n_b) \quad (1.25)$$

Comparando con (1.3), encontramos que $H(q, \theta) = 1$, lo cual da como estimador natural,

$$\hat{y}(t|\theta) = \frac{B(q)}{F(q)}u(t) = w(t,\theta) \quad (1.26)$$

El paso natural a seguir en el desarrollo del modelo de error a la salida es introducir en el modelo las propiedades del error en la salida. Describiéndolo como un modelo **ARMA** nos da como resultado,

$$y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t) \quad (1.27)$$

En cierto sentido, esta es la parametrización de dimensión finita más natural a partir de la descripción (1.3): las funciones de transferencia G y H son parametrizadas de forma independiente como funciones racionales. El modelo (1.27) fue sugerido y tratado por Box y Jenkins (1970).[5] Este modelo a su vez representa la familia de modelos relacionados con la ecuación de error a la salida, cuyo flujo de señales se muestra en la Figura 1.5 y su estimador esta descrito por,

$$\hat{y}(t|\theta) = \frac{D(q)B(q)}{C(q)F(q)}u(t) + \frac{C(q)-D(q)}{C(q)}y(t) \quad (1.28)$$

Las estructuras discutidas anteriormente, permiten la obtención de un modelo general, en cuyo caso uno o varios de sus cinco polinomios son reducidos a la unidad dependiendo de la aplicación. Esta estructura general esta dada por,

$$A(q)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t) \quad (1.29)$$

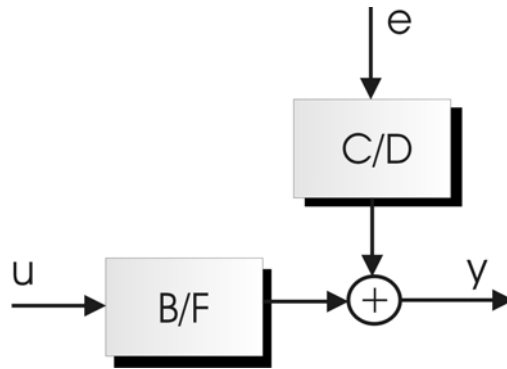


Figura 1.5 La estructura del modelo BJ

Finalmente, se pueden considerar los modelos de caja negra más comúnmente usados en la práctica como casos especiales del modelo descrito por (1.29), los cuales se resumen en la Tabla 1.1 propuesta por Ljung.[4] Estos modelos de caja negra son aplicables a sistemas **SISO** (*single-input single-output*), esto es, sistemas de una variable a la entrada y una a la salida.

Polinomios usados en (1.29)	Nombre del modelo
B	<i>FIR (respuesta finita al impulso)</i>
AB	<i>ARX</i>
ABC	<i>ARMAX</i>
AC	<i>ARMA</i>
ABD	<i>ARARX</i>
ABCD	<i>ARARMAX</i>
BF	<i>OE</i>
BFCD	<i>BJ (Box-Jenkins)</i>

Tabla 1.1 Modelos SISO de caja negra.

1.2.3 Métodos Paramétricos para la Identificación de Sistemas

Es posible efectuar múltiples clasificaciones de los métodos de estimación de parámetros en el proceso de identificación del sistema. Considerando las medidas disponibles y las señales aplicadas al sistema durante la identificación, Aníbal Ollero [3] establece la siguiente clasificación:

1. *Métodos que utilizan la respuesta ante señales de ensayo sobre el sistema.*
2. *Métodos que utilizan medidas históricas de funcionamiento de la planta.* Son imprescindibles cuando no es posible realizar ensayos como en el caso de sistemas socioeconómicos y de muchos sistemas industriales en los que no es posible o resulta costoso modificar las condiciones de trabajo.
3. *Métodos de identificación en línea.* Cuando su aplicación resulta posible sin perturbar significativamente las condiciones de trabajo del sistema.
4. *Métodos de identificación en tiempo real,* en los cuales es muy interesante la característica recursiva.

Considerando ahora las características del modelo que se pretende obtener se establece otra clasificación de los métodos de la siguiente forma:

1. *Métodos para la identificación de un modelo paramétrico,* en los cuales se trata de obtener los valores numéricos de los coeficientes de las funciones o matrices de transferencia o de los elementos de las matrices de representación mediante variables de estado.
2. *Métodos para la identificación de un modelo no paramétrico,* como en el caso de las curvas de módulos y argumentos de una respuesta en frecuencia las cuales constituyen la única descripción necesaria para la aplicación de métodos de síntesis empleando diagramas de Nyquist, Bode y Nichols.

No obstante la diferencia entre los elementos de la clasificación anterior, existe una relación natural entre métodos: a partir de un modelo paramétrico, es inmediata la obtención de curvas de respuesta en frecuencia y en un modelo no paramétrico, se puede

parametrizar empleando coeficientes tales como márgenes de fase y de ganancia, ancho de banda entre otros.

El presente estudio se enfoca a los métodos paramétricos los cuales resultan de mayor utilidad para la comprensión del método utilizado en el desarrollo experimental. De esta forma dando continuidad a los modelos obtenidos en la sección anterior se establece de forma general el método para la identificación de sistemas usando la estimación de parámetros.

Una vez elegida cierta estructura M con modelos particulares $M(\theta)$ parametrizados, donde cada modelo representa una forma de estimar salidas futuras, y recolectado un conjunto de datos sobre el sistema Z donde,

$$Z^N = [y(1), u(1), y(2), u(2), \dots, y(N), u(N)] \quad (1.30)$$

El problema a enfrentar es el de cómo utilizar la información contenida en Z^N para seleccionar un valor apropiado $\hat{\theta}_N$ del vector de parámetros, y en consecuencia un modelo particular apropiado $M(\hat{\theta}_N)$ del conjunto M . Expresado con más rigor matemático, se debe realizar un mapeo de los datos Z^N al conjunto D_M ,

$$Z^N \rightarrow \hat{\theta}_N \in D_M \quad (1.31)$$

Un mapeo expresado en esta forma, representa un método de estimación de parámetros.[4]

Este problema se puede resolver encontrando una prueba que permita evaluar la habilidad de los diversos modelos para “seguir” o reproducir la información observada. La importancia de un modelo radica en su capacidad de estimar la salida del sistema, en consecuencia se ha de juzgar su desempeño basados en ese aspecto. Entonces, queda expresado el error en la estimación dado por un cierto modelo $M(\theta^*)$ como,

$$\varepsilon(t, \theta^*) = y(t) - \hat{y}(t | \theta^*) \quad (1.32)$$

Entonces un “buen” modelo es aquel que produzca “pequeños” errores de estimación de la salida. Cabe señalar que existe una considerable flexibilidad al seleccionar diversas funciones estimadoras con una correspondiente libertad al definir lo que es un “buen” modelo. Como guía se puede considerar el siguiente párrafo expresado por Ljung: [4]

Basados en Z^N se puede calcular el error de estimación $\varepsilon(t, \theta)$ usando (1.32). En el tiempo $t = N$, se selecciona $\hat{\theta}_N$ de tal forma que los errores de estimación $\varepsilon(t, \hat{\theta}_N)$, $t = 1, 2, \dots, N$, se vuelvan tan pequeños como sea posible.

La pregunta ahora es como definir “pequeño”. Una aproximación es crear una norma escalar o función criterio para medir el tamaño de ε . Otra aproximación es obligar que $\varepsilon(t, \hat{\theta}_N)$ no tenga correlación con una secuencia de información dada. Esto corresponde a decir que ciertas proyecciones de $\varepsilon(t, \hat{\theta}_N)$ son cero.

La primer aproximación resulta de gran utilidad para introducir el criterio de los **mínimos cuadrados**. Este criterio permite aprovechar el que los modelos **ARX** y **ARMAX** describan una regresión lineal y pseudolineal respectivamente, para la estimación de sus parámetros. Este criterio se expone a continuación.

La secuencia del error de estimación en (1.32) puede ser vista como un vector en R^N . El “tamaño” de este vector puede ser medido usando cualquier norma en R^N , cuadrática o no cuadrática. Ahora se evalúa cuán grande es la secuencia del error de estimación: Sea la secuencia de error de estimación filtrada a través de un filtro lineal estable $L(q)$,

$$\varepsilon_F(t, \theta) = L(q)\varepsilon(t, \theta), \quad 1 \leq t \leq N$$

entonces usando la norma siguiente,

$$V_N(\theta, Z^N) = \frac{1}{N} \sum_{t=1}^N \ell(\varepsilon_F(t, \theta)) \quad (1.33)$$

donde $\ell(\cdot)$ es una función escalar (típicamente positiva).

La función $V_N(\theta, Z^N)$ es, para una Z^N dada, una función escalar definida del modelo paramétrico θ . Es una medida natural de la validez del modelo $M(\theta)$. La estimación $\hat{\theta}_N$ esta entonces definida por la minimización de (1.33):

$$\hat{\theta}_N = \hat{\theta}_N(Z^N) = \arg \min V_N(\theta, Z^N) \quad (1.34)$$

Esta forma de estimar θ puede ser evaluada mediante varios procedimientos dependiendo de la elección de $\ell(\cdot)$, del prefiltro $L(\cdot)$ y del la estructura del modelo. Uno de estos procedimientos es el de los mínimos cuadrados.

A partir de la ecuación (1.10),

$$\hat{y}(t|\theta) = \theta^T \varphi(t) = \varphi^T(t) \theta$$

el error de predicción se vuelve,

$$\varepsilon(t, \theta) = y(t) - \varphi^T(t) \theta$$

y la función criterio resultante de (1.33) y (1.34) con $L(q) = 1$ y $\ell(\varepsilon) = \frac{1}{2} \varepsilon^2$ es,

$$V_N(\theta, Z^N) = \frac{1}{N} \sum_{t=1}^N \frac{1}{2} [y(t) - \varphi^T(t) \theta]^2 \quad (1.35)$$

Este es el criterio de los mínimos cuadrados para la regresión lineal (1.10). La única característica de este criterio, que se desarrolla a partir de la parametrización lineal y del criterio cuadrático, es que es una función cuadrática en θ . En consecuencia puede ser minimizada analíticamente, lo cual da, demostrando que la inversa indicada existe,

$$\hat{\theta}_{LS}^N = \arg \min V_N(\theta, Z^N) = \left[\frac{1}{N} \sum_{t=1}^N \varphi(t) \varphi^T(t) \right]^{-1} \frac{1}{N} \sum_{t=1}^N \varphi(t) y(t) \quad (1.36)$$

que es la estimación de mínimos cuadrados (*least-square estimate*, **LSE**).[4]

Introduciendo la matriz $d \times d$,

$$R(N) = \frac{1}{N} \sum_{t=1}^N \varphi(t) \varphi^T(t) \quad (1.37)$$

y el vector columna de dimensión d ,

$$f(N) = \frac{1}{N} \sum_{t=1}^N \varphi(t) y(t) \quad (1.38)$$

se tiene,

$$\hat{\theta}_{LS}^N = \arg \min V_N(\theta, Z^N) = [R(N)]^{-1} f(N)$$

El método de los mínimos cuadrados es un caso especial de la ecuación (1.32) con propiedades interesantes.

Si se supone que la información observada es generada por,

$$y(t) = \varphi^T(t)\theta_0 + v_0(t) \quad (1.39)$$

para cierta secuencia $\{v_0(t)\}$. Se puede pensar en θ_0 como en un “valor cierto” del vector de parámetros. Sustituyendo (1.39) en (1.36) se tiene,

$$\hat{\theta}_{LS} = [R(N)]^{-1} \frac{1}{N} \sum_{t=1}^N \varphi(t) [\varphi^T(t)\theta_0 + v_0(t)]$$

$$\hat{\theta}_{LS} = \theta_0 + [R(N)]^{-1} \frac{1}{N} \sum_{t=1}^N \varphi(t)v(t)$$

Propiedades deseables de $\hat{\theta}_N$ son:

1. $\hat{\theta}_N$ es muy cercano a θ_0 .
2. $\hat{\theta}_N$ converge a θ_0 conforme N tiende a infinito.

Se puede observar que si $v_0(t)$ en (1.39) es pequeño comparado con $\varphi(t)$ entonces el término del error,

$$[R(N)]^{-1} \frac{1}{N} \sum_{t=1}^N \varphi(t)v(t)$$

es pequeño, y en consecuencia $\hat{\theta}_N$ se aproxima a θ_0 .

1.3 Variables de Estado

En un sistema discreto definimos estado como la mínima cantidad de información registrada sobre el sistema necesaria para determinar tanto la salida como los futuros estados del sistema dada una entrada conocida. [6] De manera más precisa, un conjunto de variables x es considerado un vector de estado si dos funciones F y G pueden ser encontradas tales que,

$$\begin{aligned} x(kT + T) &= F[x(kT), u(kT)] \\ y(kT) &= G[x(kT), u(kT)] \end{aligned} \quad (1.40)$$

donde $y(kT)$ es el vector de salida en el tiempo kT y $u(kT)$ el vector de entrada en el tiempo kT . El tiempo kT es el resultado de muestrear la señal continua en el tiempo para trasladarla a un espacio discreto y que por tratarse este de un desarrollo de conceptos en el tiempo discreto se denomina simplemente como t . De esta forma se puede expresar (1.40) como,

$$\begin{aligned} x(t+1) &= F[x(t), u(t)] \\ y(t) &= G[x(t), u(t)] \end{aligned} \quad (1.41)$$

Una descripción paramétrica típica en variables de estado de un sistema dinámico es

$$\begin{aligned} x(t+1 | \theta) &= A(t, \theta)x(t) + B(t, \theta)u(t) \\ y(t | \theta) &= C(t, \theta)x(t) + D(t, \theta)u(t) \end{aligned} \quad (1.42)$$

siendo A , B , C y D matrices que describen la dinámica del sistema. Si el sistema es de parámetros concentrados, los elementos de las matrices A , B , C y D son constantes. Esta descripción se muestra en la Figura 1.6.

Como una simplificación en lo sucesivo, las salidas de los elementos de retardo unitario en el tiempo del diagrama de simulación de un sistema serán consideradas como el estado del sistema. Estas salidas proveen una elección conveniente y suficiente del vector de estados.

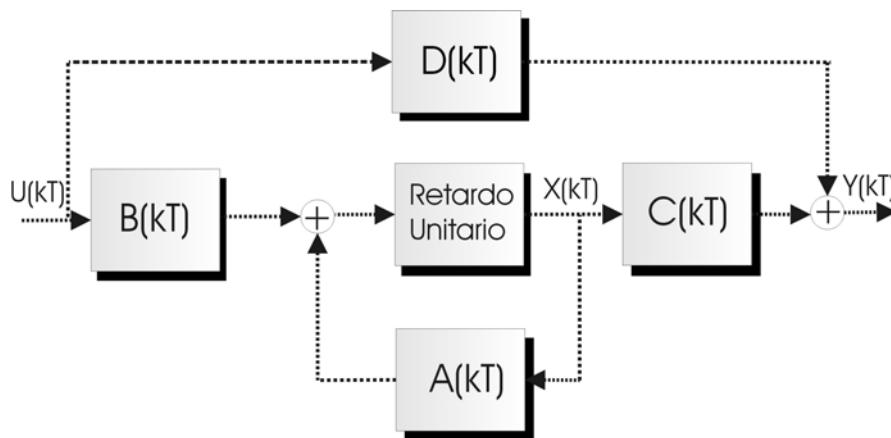


Figura 1.6 Flujo de señales de un sistema representado en variables de estado.

1.4 Estimación de Estados

1.4.1 Introducción a la Estimación de Estados

En esta sección se aborda el tema de la estimación de las variables de estado de un sistema dinámico. La diferencia fundamental en la estimación de estados respecto a la estimación de parámetros descrita anteriormente, radica en que las magnitudes a estimar varían en el tiempo, lo cual hace que su solución sea más compleja, no obstante se aplican en lo sucesivo las nociones básicas sobre estimación ya presentadas.

Como ya se mencionó, las variables de estado resumen el comportamiento dinámico y contienen toda la información necesaria para estimar la evolución de un sistema. Entonces, todo comportamiento anormal del sistema se verá reflejado en magnitudes anormales de los residuos, que son los errores entre la estimación de la salida en variables de estado y la salida medida, además de reflejarse en magnitudes anormales de las variables de estado del sistema y sus estimaciones.

Es importante recordar que las variables de estado son entidades abstractas sin referencia directa o ninguna referencia a magnitudes físicas medibles. El estimador del vector de estados puede considerarse entonces como un observador que tiene en cuenta las componentes aleatorias del sistema.

1.4.2 Observadores

En numerosas ocasiones no es posible medir directamente el estado de un sistema debido a que no existe una correspondencia directa con magnitudes físicas, o bien a que no se disponen de todos los sensores necesarios. La noción del concepto de *observabilidad* está relacionada con la posibilidad de obtener el estado del sistema a partir de las entradas y salidas que son las únicas medidas que, en principio, son accesibles.

Strejc [7] define que el estado $x(t_0)$ de un sistema es observable si para cada instante $k = k_0$ existe un N entero finito tal que el conocimiento de las secuencias:

$$\begin{aligned} u(t_0), u(t_0 + 1), \dots, u(t_0 + N) \\ y(t_0), y(t_0 + 1), \dots, y(t_0 + N) \end{aligned} \quad (1.43)$$

es suficiente para determinar el estado $x(t_0)$.

De la misma forma Ollero [3] dice que el estado $x(t_0)$ puede ser reconstruido si para cada instante $t = t_0$ existe un N entero finito tal que el conocimiento de las secuencias:

$$\begin{aligned} u(t_0 - N), u(t_0 - N + 1), \dots, u(t_0) \\ y(t_0 - N), y(t_0 - N + 1), \dots, u(t_0) \end{aligned} \quad (1.44)$$

es suficiente para determinar el estado $x(t_0)$.

Estos conceptos se extienden también a la observación y reconstrucción completa de un sistema.

Como se sabe, la respuesta del sistema (1.42) ante una secuencia de entrada $u(t)$ esta dada por,

$$y(t) = Cx(t) = CA^t x(0) + C \sum_{j=0}^{t-1} A^{t-1-j} Bu(j) \quad (1.45)$$

El primer sumando del segundo término corresponde a la evolución del sistema en régimen libre, es decir, para $u(t) = 0, \dots, N$ desde un estado inicial arbitrario $x(0)$. El segundo corresponde al régimen forzado por $u(t)$ a partir del estado $x(0) = 0$.

Para el estudio de la observabilidad resulta interesante considerar únicamente el término de respuesta en régimen libre.

Si se define $y^a(t)$ con la expresión,

$$y^a(t) = y(t) - C \sum_{j=0}^{t-1} A^{t-1-j} Bu(j) \quad (1.46)$$

se observa que a partir de $y(t)$ y de la secuencia $u(1), \dots, u(t)$ es posible determinar el valor de $y^a(t)$. De esta forma el estudio de la observabilidad del sistema (1.42) puede realizarse a través del estudio de un sistema cuya salida sea,

$$y^a(t) = CA^t x(0) \quad (1.47)$$

es decir, un sistema con entrada nula, las mismas matrices C y A , y el mismo estado inicial.

La expresión (1.47) para $t = 0, \dots, N-1$ puede expresarse de la forma,

$$vy = \psi x(0)$$

siendo

$$vy \square [y(0), \dots, y(N-1)]^T$$

un vector de $q \times N$ elementos correspondientes a las q salidas del sistema y,

$$\mathfrak{G} = [C, CA, \dots, CA^{N-1}]^T$$

una matriz de $q \times N$ filas y n columnas a la que se denomina *matriz de observabilidad*.

Si el sistema tiene una sola salida, $y(t)$ es un escalar, vy es un vector de dimensión N y ψ es una matriz de N filas y n columnas. De esta forma, si $N = n$ teniendo en cuenta que vy es conocido, la expresión anterior se convierte en un sistema de n ecuaciones con las n componentes de $x(0)$ como incógnitas. La existencia de esta solución exige que ψ sea no singular o lo que es lo mismo,

$$\text{rango } \mathfrak{G} = n \quad (1.48)$$

Puede demostrarse que la igualdad (1.48) del rango de la matriz de observabilidad a la dimensión del vector de estados es una condición necesaria y suficiente de observabilidad. Cuando se cumple esta condición es posible resolver el problema de la observación mediante la expresión,

$$x(0) = \mathfrak{G}^{-1}vy$$

1.4.3 El Filtro de Kalman

Los primeros trabajos significativos sobre estimación de señales se deben a Wiener, que durante la Segunda Guerra Mundial obtuvo resultados que son fundamentales en el tratamiento de series temporales. De esta forma se desarrolla el denominado filtro de Wiener que permite obtener la estimación de las variables de estado de sistemas en tiempo continuo. Posteriormente Ragazzini y Franklin extienden los resultados a sistemas muestreados.[4]

El filtro de Kalman o Kalman-Bucy se desarrolla en el marco de la teoría moderna de control, y está adaptado a la utilización de procesos digitales. En efecto la estimación es recurrente, lo que permite una fácil programación del algoritmo. Por otra parte el filtro de Kalman puede generalizarse para la estimación de señales cuyas propiedades estadísticas varían en el tiempo.

La estrategia de estimación describe Anibal Ollero [4], consiste en la repetición de los dos pasos siguientes:

- i. *Ciclo de propagación*: Estimar el estado $\hat{x}(t+1|t)$ y la covarianza $P(t+1|t)$ del vector de error de estimación.
- ii. *Ciclo de actualización*: Corregir las estimaciones del ciclo de propagación utilizando las medidas $y(t+1)$.

Las ecuaciones recursivas de estimación para $\hat{x}(t+1|t)$, $P(t+1|t)$ así como la actualización de \hat{x} se describen en el apéndice A y son usadas en el capítulo 3.

1.5 Redes Neuronales

1.5.1 Introducción a las Redes Neuronales

El nacimiento de las redes neuronales artificiales fue motivado inicialmente por las estructuras biológicas en los cerebros de los humanos y animales, las cuales son extremadamente poderosas para tareas como lo son el procesamiento de información, el aprendizaje y los procesos de adaptación. Las principales características de las redes neuronales son:

- 1) Un gran número de unidades simples.
- 2) Un alto número de unidades paralelas.
- 3) Unidades fuertemente interconectadas.
- 4) Robustez en caso de fallo individual de las unidades.
- 5) Aprendizaje a partir de la información suministrada a la red.

Estas características permiten que las redes neuronales artificiales sean apropiadas para implementaciones en procesos en los cuales sea necesario un rápido manejo de la información. La mayoría de las redes neuronales artificiales usadas en ingeniería en la actualidad están tan relacionadas con las matemáticas, estadística y optimización como con el fundamento biológico que las inspiró.

Usando la terminología aplicada a las redes neuronales, un modelo es considerado una red neuronal si sus funciones fundamentales son todas del mismo tipo y con una descripción como se muestra en la Figura 1.7. El nodo a la salida es llamado neurona de salida, y todas las neuronas de salida juntas se consideran la *capa de salida*. Cada uno de los M nodos en el centro que realiza una función fundamental es llamado neurona oculta y todas ellas juntas forman la *capa oculta*. Finalmente las entradas son en ocasiones denominadas neuronas de entrada y todas ellas consideradas como la *capa de entrada*. No obstante estas neuronas solo distribuyen las entradas hacia todas las neuronas de la capa oculta y no realizan ningún cálculo real.

Debido a que en la literatura no existe un acuerdo de si las neuronas de entrada deben ser consideradas como una capa o no, la red de la Figura 1.7 puede ser considerada una red bicapa o tricapa.[8] Esta descripción es muy general ya que existen redes neuronales que no pueden ser descritas mediante esta estructura, pero resulta útil al ofrecer un panorama de partida para la representación de una red neuronal.

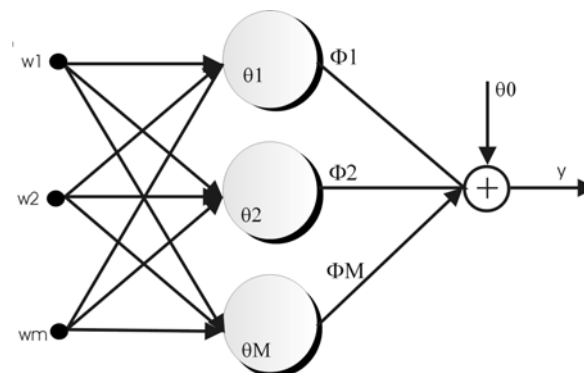


Figura 1.7 Esquema general de una red neuronal.

Para una red neuronal, los parámetros lineales asociados con la neurona de salida son denominados *pesos de salida*,

$$\theta_i^{(1)} = w_i$$

La neurona de salida es usualmente una combinación lineal de las neuronas ocultas o funciones fundamentales $\Phi_i(\cdot)$ con un *offset* adicional w_0 , el cual es usualmente llamado sesgo o umbral como se observa en la Figura 1.8.

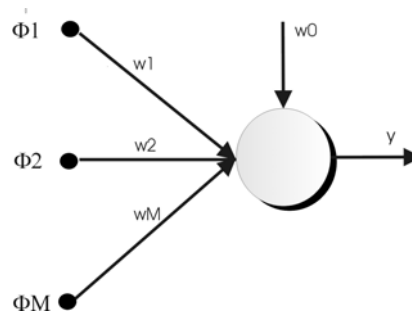


Figura 1.8 Diagrama de una neurona

La elección de una red neuronal con una estructura determinada depende de los resultados que se deseen obtener de dicha red, pero independientemente de la estructura que se elija, la red realiza dos procesos fundamentales: un proceso de aprendizaje y un proceso de entrenamiento.[9]

El aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. Los cambios que se producen durante el proceso de aprendizaje se reducen a la destrucción, modificación y creación de conexiones entre las neuronas. En las redes neuronales artificiales, la creación de una nueva conexión implica que el peso de la misma adquiere un valor distinto de cero. De la misma forma, una conexión se destruye cuando su peso se vuelve cero. Se puede afirmar que este proceso ha terminado (la red ha aprendido) cuando los valores de los pesos permanecen estables ($dw/dt = 0$).[9]

Existen criterios fundamentales, llamados reglas de aprendizaje, que clasifican el cómo la red modifica los valores de los pesos: aprendizaje supervisado y aprendizaje no supervisado, así como, aprendizaje en línea y aprendizaje fuera de línea.

El aprendizaje supervisado se caracteriza porque el proceso de aprendizaje se realiza mediante un entrenamiento controlado por un agente externo (supervisor o maestro) que determina la respuesta que

debería generar la red a partir de una entrada determinada. En este tipo de aprendizaje se suelen considerar a su vez tres formas de llevarlo a cabo:

- 1) Aprendizaje por corrección de error.
- 2) Aprendizaje por refuerzo.
- 3) Aprendizaje estocástico.

Las redes con aprendizaje no supervisado no requieren influencia externa para ajustar los pesos de las conexiones entre sus neuronas. La red no recibe ninguna información por parte del entorno que le indique si la salida generada en respuesta a una determinada entrada es o no correcta, por esto se dice que estas redes son capaces de auto organizarse.

En algunos casos la salida de una red neuronal con aprendizaje no supervisado representa el grado de familiaridad o afinidad entre la información que se le está presentando en la entrada y las informaciones que se le han mostrado hasta entonces (en el pasado). En otro caso podría realizar un establecimiento de categorías (*clustering*), indicando la red a la salida a qué categoría pertenece la información presentada a la entrada. Una variación de esta categorización es el “prototipado”, donde la red obtiene prototipos representantes de las clases a las que pertenecen las informaciones de entrada. También el aprendizaje no supervisado permite realizar una codificación de los datos de entrada, generando a la salida una versión codificada de menor dimensión. Finalmente, algunas redes lo que realizan es un mapeo de características (*feature maps*), obteniéndose en las neuronas de salida una disposición geométrica que representa un mapa topográfico de las características de los datos de la entrada.

Los algoritmos de aprendizaje no supervisado pueden englobarse en dos tipos:

- 1) Aprendizaje hebbiano
- 2) Aprendizaje competitivo y cooperativo

Finalmente el proceso de entrenamiento es un proceso en el cual los pesos modificados por las entradas presentadas a la red durante el aprendizaje se llevan a un punto en el cual se vuelven estables mediante mecanismos de iteración, es decir se vuelven invariantes en el tiempo consolidando de esta manera el aprendizaje de la red neuronal.

1.5.2 Redes Tipo SOFM

Las redes SOFM, define Haykin [10], (*self-organizing feature maps*) son redes no supervisadas cuyo algoritmo de aprendizaje es de tipo competitivo: las neuronas de salida de la red compiten entre ellas para ser activadas o disparadas, con el resultado de que solo una neurona, o una neurona por grupo, esta encendida en un tiempo dado. Las neuronas ganadoras en la competencia son llamadas neuronas “ganadora toma todo” (*winner-takes-all*).

En una red SOFM, las neuronas son colocadas en los nodos de un entramado que es usualmente de una o dos dimensiones. Las neuronas se vuelven selectivas a encenderse a diversos patrones de entrada o tipos de patrones durante el proceso de aprendizaje competitivo. Las regiones de las neuronas encendidas de esa manera tienden a reordenarse con respecto a ellas mismas de tal forma que un sistema coordinado significativo es creado sobre el entramado. Una red SOFM está caracterizada, como se menciona en la sección anterior, por la formación de mapas topográficos de los patrones de entrada en los cuales las regiones espaciales de las neuronas en el entramado corresponden a características intrínsecas de los patrones de entrada.

El desarrollo de esta clase de redes neuronales artificiales fue motivado por una característica del cerebro humano: el cerebro esta organizado en diversos lugares de forma tal que entradas sensoriales diferentes están representadas por mapas ordenados topológicamente. En particular, entradas sensoriales como las táctiles (Kaas), visuales (Hubek y Wiesel) y acústicas (Suga) son mapeadas sobre diferentes áreas de la corteza cerebral de un manera topológicamente ordenada.

Haykins [10] menciona que el uso de mapas computacionales ofrece las siguientes ventajas:

1. ***Procesamiento eficiente de la información.*** El sistema nervioso es requerido para analizar eventos acontecidos en un ambiente dinámico de manera continua. Esto, requiere el uso de estrategias de procesamiento que permitan un procesamiento ágil de grandes volúmenes de información. Los mapas computacionales desempeñados por unidades de procesamiento en paralelo, son ideales para estas tareas. Proveen un método para la rápida clasificación y procesamiento de estímulos complejos, y la representación de los resultados en una forma simple y sistemática.
2. ***Simplicidad en el acceso a la información procesada.*** El uso de mapas computacionales simplifica los esquemas de conectividad requeridos para utilizar la información en procesadores de ordenes mayores.
3. ***Forma común de representación.*** Una representación mapeada común de los resultados de diversos tipo.

Para entender mejor el desarrollo de las redes SOFM, se discute primero el uso de la retroalimentación lateral (*lateral feedback*) como un mecanismo para modificar la forma de excitación aplicada a la red. Por retroalimentación lateral se entiende una forma particular de retroalimentación dependiente de la distancia lateral al punto de su aplicación. Esta retroalimentación produce conexiones no implícitas con efectos inhibidores o excitadores, dependiendo de la distancia.

Siguiendo motivos biológicos, la retroalimentación lateral es descrita como una función de tipo sombrero mexicano, como se muestra en la Figura 1.9. De acuerdo a esta figura, podemos distinguir tres áreas de interacción lateral entre neuronas:

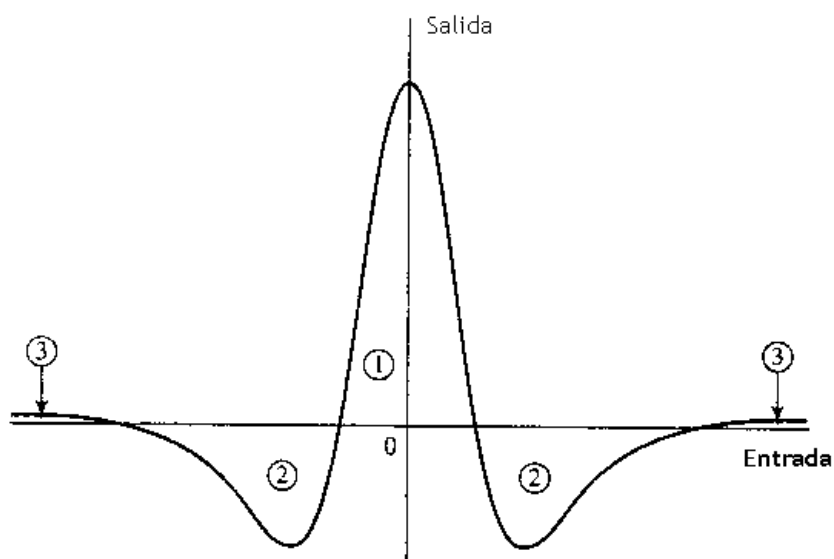


Figura 1.9 Función de sombrero mexicano.

- 1) Una pequeña área de excitación lateral
- 2) Una penumbra de acción inhibidora
- 3) Un área de débil excitación que rodea el área de penumbra, la cual es usualmente ignorada.

Esta red exhibe dos importantes características. Primero, la red tiende a concentrar su actividad eléctrica en regiones (*local clusters*), referenciadas como burbujas de actividad. Segundo, la localización de las burbujas son determinadas por la naturaleza de las señales de entrada.

Sea que x_1, x_2, \dots, x_p describan las señales de entrada (excitaciones) aplicadas a la red, donde p es el número de terminales de entrada. Ahora sean $w_{j1}, w_{j2}, \dots, w_{jp}$ los pesos sinápticos de la neurona j , $c_{j,-K}, \dots, c_{j,-1}, c_{j,0}, c_{j,1}, \dots, c_{j,K}$ los pesos de la retroalimentación lateral conectados a la neurona j , donde K es el “radio” de la interacción lateral. Finalmente sean y_1, y_2, \dots, y_N las señales de salida de la red, donde N es el número de neuronas en la red. Entonces se puede expresar la señal de salida (respuesta) de la neurona j como sigue,

$$y_j = \varphi \left(I_j + \sum_{k=-K}^K c_{jk} y_{j+k} \right), j = 1, 2, \dots, N \quad (1.49)$$

donde $\varphi(\cdot)$ es una función no lineal que limita el valor de y_j y asegura que $y_j \geq 0$.

El término I_j sirve a la función como estímulo, representando todo el control externo ejercido sobre la neurona j por el efecto de las señales de entrada a través de los pesos; esto es,

$$I_j = \sum_{l=1}^p w_{jl} x_l \quad (1.50)$$

Generalmente, el estímulo I_j es una función suave de índice espacial j .

La solución a la ecuación no lineal (1.49) es encontrada mediante iteración, usando una técnica de “relajación”. Esta puede ser reformulada como una ecuación en diferencias de la siguiente forma,

$$y_j(n+1) = \varphi \left(I_j + \beta \sum_{k=-K}^K c_{jk} y_{j+k}(n) \right), j = 1, 2, \dots, N \quad (1.51)$$

donde n representa tiempo discreto. Así pues, $y_j(n+1)$ es la neurona j en el tiempo $n+1$, y $y_{j+k}(n)$ es la neurona de salida $j+k$ en el tiempo previo n . El parámetro β en el argumento en el miembro izquierdo de (1.51) controla la velocidad de convergencia del proceso de relajación.

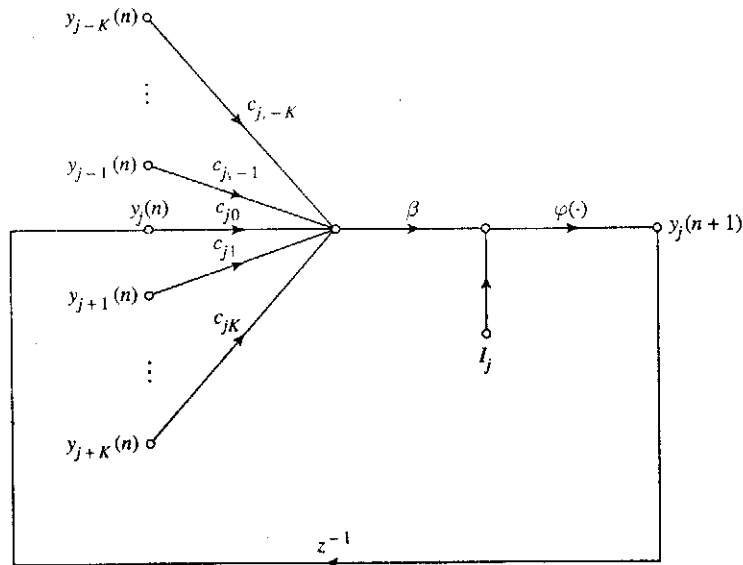


Figura 1.10 Diagrama de la ecuación de relajación.

La ecuación de relajación (1.51) representa un sistema con retroalimentación como se muestra en la Figura 1.10, donde z^{-1} es el operador de retraso unitario. El parámetro β juega el papel de un factor de retroalimentación en el sistema. El sistema incluye tanto retroalimentación negativa como positiva, correspondiendo estas a las partes excitadoras e inhibitoras de la función de sombrero mexicano respectivamente. La acción limitante de la función de activación no lineal $\varphi(\cdot)$ provoca que la respuesta espacial $y_j(n)$ se establezca de cierta manera, dependiendo de los valores asignados a β . Si β es lo suficientemente grande, entonces el último paso correspondiente a $n \rightarrow \infty$, los valores de y_j tienden a concentrarse dentro de un grupo espacial limitado, esto es, una “burbuja de actividad”. La burbuja está centrada en un punto donde la respuesta inicial $y_j(0)$ debida al estímulo I_j es máxima.

El ancho de la burbuja de actividad depende de la relación entre las interconexiones de excitación a inhibición laterales. Se puede asegurar lo siguiente:

- 1) Si la retroalimentación positiva se hace más fuerte, la burbuja de actividad se vuelve más amplia.
- 2) Si la retroalimentación negativa se refuerza, la burbuja de actividad se vuelve más aguda o pronunciada.

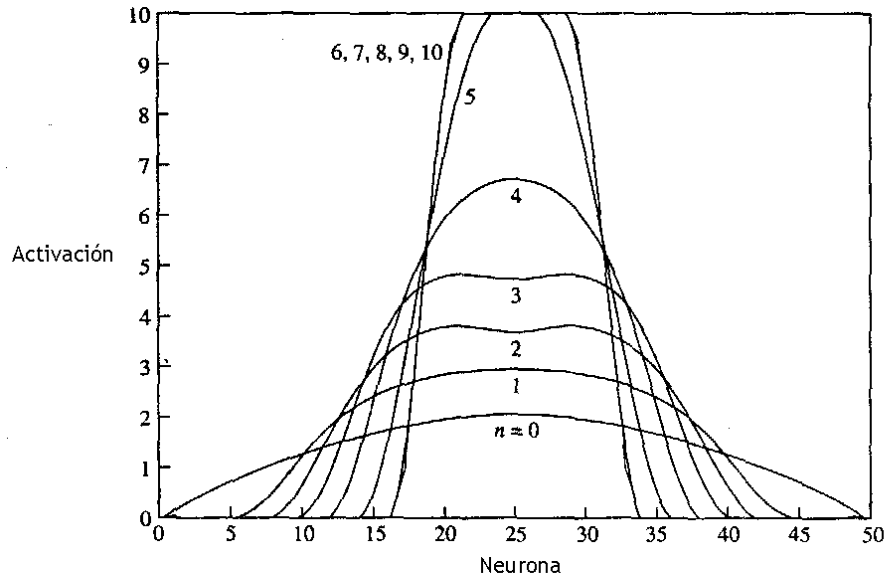
Por supuesto que si la retroalimentación neta que actúa en el sistema es muy negativa, se previene la formación e la burbuja de actividad. Estos aspectos se muestran en la Figura 1.11 para valores de $\beta = 2$ (a) y $\beta = 0.75$ (b).

Una vez que se ha formado la burbuja de actividad, se puede idealizar la salida de la neurona j como sigue:

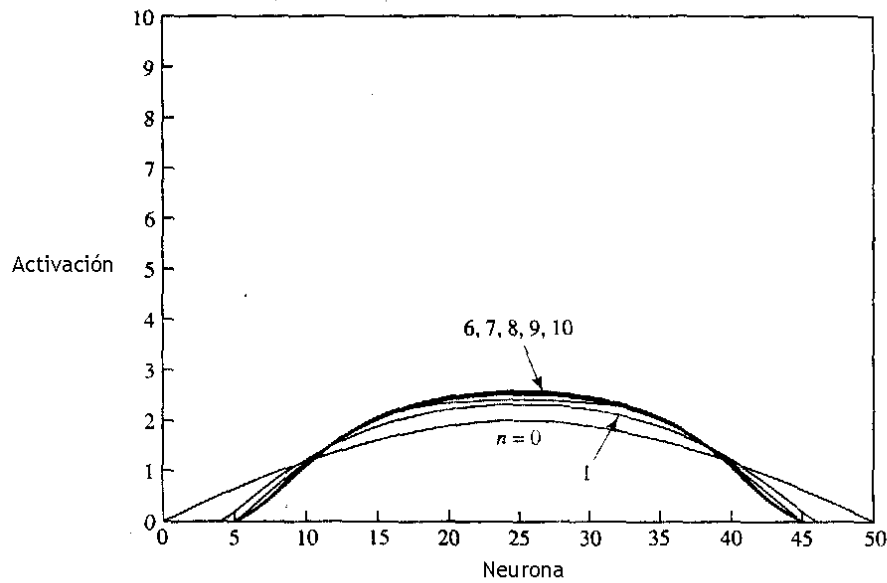
$$y_j = \begin{cases} a, & \text{neurona } j \supset \text{ burbuja} \\ 0, & \text{neurona } j \not\subset \text{ burbuja} \end{cases}$$

donde a es un valor limitador de la función no lineal $\varphi(\cdot)$ definiendo la relación de entrada-salida de la neurona j .

El entendimiento de la formación de las burbujas de actividad es básico para la interpretación de los mapas topográficos en la redes neuronales tipo SOFM y el caso particular de la red de Kohonen.



(a)



(b)

Figura 1.11 Formación de las burbujas de actividad.

1.5.3 La red de Kohonen

El principal objetivo del algoritmo SOFM desarrollado por Kohonen (1982) es el transformar un señal de entrada de dimensión arbitraria en un mapa discreto de una o dos dimensiones, y desempeñar

dicha transformación adaptable en una manera ordenada y topológica. Varios patrones de activación son presentados a la red uno a la vez.

El funcionamiento de la red es relativamente simple. Cuando se presenta a la entrada una información $E_k = (e_1^{(k)}, \dots, e_N^{(k)})$, cada una de las M neuronas de la capa de salida la recibe a través de las conexiones dispersión con pesos w_{ij} . También estas neuronas reciben las correspondientes entradas debidas a las conexiones laterales con el resto de las neuronas de salida y cuya influencia mutua depende de la distancia a la que se encuentren.

El modelo de Kohonen es uno de los más útiles en computación neuronal, a pesar de sus limitaciones en cuanto a la duración del proceso de aprendizaje y a la imposibilidad de aprender nuevos datos sin tener que volver a repetir completamente el proceso de aprendizaje con todos los patrones. Su utilidad radica en su capacidad de establecer clases o categorías de datos sin supervisión.[9]

Como aplicaciones, destacan las relacionadas con el reconocimiento de patrones (voz, texto, imágenes), codificación de datos, compresión de imágenes y resolución de problemas de optimización.

Los elementos esenciales de la red neuronal son los siguientes:

1. *Un entramado unidimensional o bidimensional de neuronas* que calculen funciones de discriminación simples de entradas recibidas de una entrada de dimensión arbitraria.
2. *Un mecanismo que compare* estas funciones discriminantes y seleccione la neurona con el mayor valor de la función discriminante.
3. *Una red interactiva* que active la neurona seleccionada y su vecindad simultáneamente.
4. *Un proceso adaptable* que permita a las neuronas activadas incrementar sus valores de función discriminante en relación a la señal de entrada.

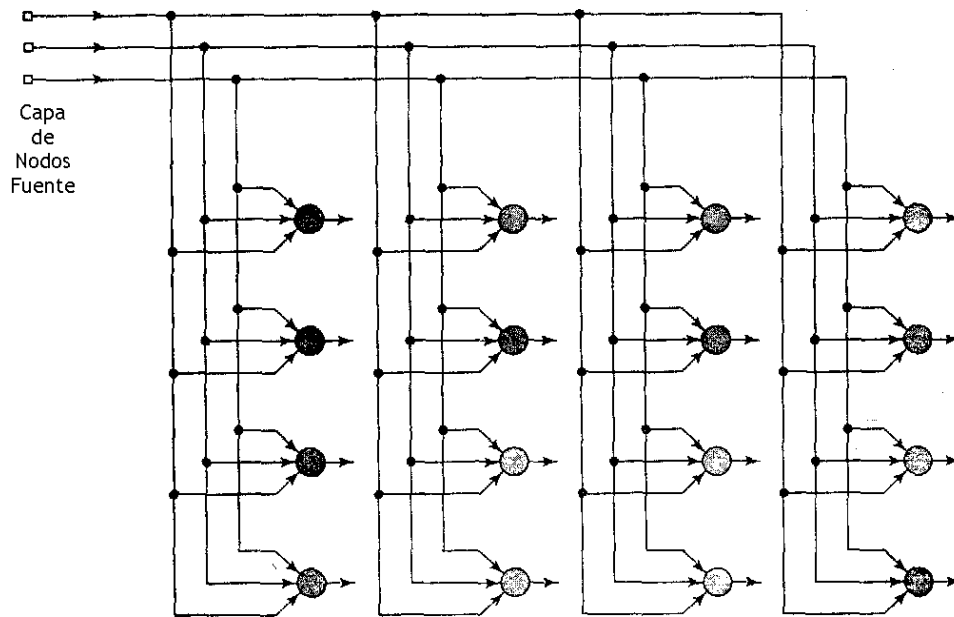


Figura 1.12 Entramado bidimensional.

A continuación se desarrolla el algoritmo a partir de un entramado bidimensional de neuronas como el mostrado en la Figura 1.12. En la figura se conectan los mismos conjuntos de entrada a todas las neuronas del entramado. El vector de entrada que representa el conjunto de señales de entrada está descrito por,

$$x = [x_1, x_2, \dots, x_p]^T$$

y el vector de pesos sinápticos de la neurona j se describe mediante,

$$w_j = [w_{j1}, w_{j2}, \dots, w_{jp}]^T, j = 1, 2, \dots, N$$

Para encontrar el mejor enlace del vector de entrada x con el vector de pesos w_j , simplemente se comparan los productos internos $w_j^T x$ para $j = 1, 2, \dots, N$ y se selecciona el mayor. Esto garantiza que el mismo umbral es aplicado a todas las neuronas. Es importante notar a su vez que $w_j^T x$ es idéntico al estímulo I_j de (1.51). De esta manera, seleccionando a la neurona con el mayor producto interno $w_j^T x$, se habrá en efecto determinado la ubicación de la burbuja de actividad.

En la formulación de un algoritmo adaptable, es conveniente normalizar el vector de pesos w_j a la norma constante euclídeana (distancia). En esta situación, el mejor criterio de enlace es equivalente a la distancia mínima euclídeana entre los vectores. En concreto, si se usa el índice $i(x)$ para identificar a la neurona que mejor enlaza al vector de entrada x , se determina entonces $i(x)$ aplicando la condición

$$i(x) = \arg_j \min \|x - w_j\|, j = 1, 2, \dots, N \quad (1.52)$$

donde $\|(\cdot)\|$ es la norma euclídeana del vector argumento. La neurona i que satisface esta condición es llamada la neurona ganadora dado el vector de entrada x . Mediante el uso de (1.52) un espacio continuo de entrada puede ser mapeado en un conjunto discreto de neuronas. Dependiendo de la aplicación de interés, la respuesta de la red puede ser tanto el índice de la neurona ganadora (su posición en el entramado), como el vector de pesos que esté más cercano al vector de entrada.

La topología de las iteraciones en el algoritmo SOFM define qué neuronas en el entramado son en realidad vecinas. Sea $\Lambda_{i(x)}(n)$ la vecindad topológica de la neurona ganadora $i(x)$. La vecindad $\Lambda_{i(x)}(n)$ es escogida de tal forma que sea una función del tiempo discreto n , por ello también se refiere a $\Lambda_{i(x)}(n)$ como la función de vecindad.

Numerosas simulaciones han demostrado que los mejores resultados en auto organización de la red son obtenidos si la función de vecindad $\Lambda_{i(x)}(n)$ es seleccionada siendo muy amplia en un principio y permitiéndole encogerse conforme el tiempo n . [10] Este comportamiento es equivalente a iniciar usando una fuerte retroalimentación lateral positiva y después reforzar la retroalimentación lateral negativa.

Para que la red sea auto organizable, los pesos sinápticos w_j de la neurona j deben cambiar en relación con el vector x . Incluyendo un término no lineal $-g(y_j)w_j$ donde $g(y_j)$ es una función escalar positiva de la respuesta y_j , cuya única restricción es que el término constante de su expansión en serie de Taylor sea 0, esto es,

$$g(y_j) = 0, \quad \text{para } y_j = 0 \quad \text{y para toda } j$$

Dada esa función podemos expresar la ecuación diferencial que define el mapa computacional resultado del algoritmo SOFM como

$$\frac{dw_j}{dt} = \eta y_j x - g(y_j) w_j, \quad j = 1, 2, \dots, N \quad (1.53)$$

donde t denota el tiempo continuo y η es la velocidad de aprendizaje del algoritmo.

Simplificando (1.53) en el sentido de la formación de la burbuja de actividad y la función de vecindad $\Lambda_{i(x)}(n)$ podemos decir que,

$$y_j = \begin{cases} 1, & \text{activa } j \in \Lambda_{i(x)} \\ 0, & \text{in activa } j \notin \Lambda_{i(x)} \end{cases}$$

de la misma forma se puede expresar la función $g(y_j)$ como,

$$g(y_j) = \begin{cases} \alpha, & y_j = 1 \quad j \in \Lambda_{i(x)} \\ 0, & y_j = 0 \quad j \notin \Lambda_{i(x)} \end{cases}$$

donde α es una constante positiva. De acuerdo a las dos condiciones anteriores se puede simplificar (1.53) de la manera siguiente,

$$\frac{dw_j}{dt} = \begin{cases} \eta x - \alpha w_j, & j \in \Lambda_{i(x)} \\ 0, & j \notin \Lambda_{i(x)} \end{cases} \quad (1.54)$$

Sin perder generalidad usando el mismo factor de escalamiento para el vector de entrada x y el vector de pesos w_j . En otras palabras, haciendo $\alpha = \eta$ quedando (1.54) expresada como,

$$\frac{dw_j}{dt} = \begin{cases} \eta(x - w_j), & j \in \Lambda_{i(x)} \\ 0, & j \notin \Lambda_{i(x)} \end{cases} \quad (1.55)$$

De acuerdo con (1.55) el vector de pesos w_j tiende a seguir al vector de entrada x conforme el tiempo t se incrementa.

Finalmente formalizando para el tiempo discreto (1.55) dado el vector de pesos $w_j(n)$ de la neurona j en el tiempo discreto n , se puede calcular el valor actualizado $w_j(n+1)$ como,

$$w_j(n+1) = \begin{cases} w_j(n) + \eta(n)[x - w_j(n)], & j \in \Lambda_{i(x)}(n) \\ w_j(n), & c.o.c. \end{cases} \quad (1.56)$$

es entonces (1.56) la fórmula de actualización deseada para el cálculo del mapa de características (*feature map*).

El efecto de la ecuación (1.56) es el de mover el vector de pesos w_j de la neurona ganadora i hacia el vector de entrada x . El algoritmo entonces dirige a un ordenamiento topológico del mapa de características en el espacio de entrada en el sentido de que las neuronas adyacentes en el entramado tienden a tener vectores de peso similares.

1.6 Resumen

En el capítulo la representación matemática de los sistemas se aborda a partir de dos enfoques principales: ecuaciones en diferencias y variables de estado. Esta división va de acuerdo al manejo que del sistema se pretende.

Las ecuaciones en diferencias son un representación adecuada para efectuar el proceso de identificación del sistema mecánico, primer etapa a realizarse en el proyecto. Dentro de los diversos modelos expuestos, el modelo **ARMAX** es importante por dos de sus características principales: el tratamiento del ruido $e(t)$ y la identificación de sus parámetros θ mediante un mecanismo semejante a una regresión lineal.

La representación en variables de estado resulta adecuada para la realización de un observador del sistema mecánico. Los observadores sirven como elementos de monitoreo de los estados del sistema y sus resultados son de utilidad en el diagnóstico de fallas. El filtro de Kalman, un observador no lineal, tiene características importantes para el manejo de señales con cierto carácter estocástico. El filtro constituye el elementos cuantitativo o

analítico dentro del modelo de diagnóstico de fallas, el cual se describe en el siguiente capítulo.

Las redes neuronales son el proceso cualitativo o heurístico dentro del algoritmo de diagnóstico de fallas. La red **SOM** en particular, presenta dos características relevantes: su aprendizaje no supervisado, y la auto organización de sus pesos como una codificación de los patrones de entrada. Estas cualidades son de utilidad para la evaluación de escenarios que efectúa la red en el procedimiento de diagnóstico de las fallas.

1.7 Referencias

- [1] Graupe, D.: *Identification of Systems*. Krieger, 1975.
- [2] Eykhoff, P.: *System Identification, Parameter and State Estimation*. Wiley, 1979.
- [3] Ollero, A.: *Control por Computadora*. Alfaomega Marcombo, 1991.
- [4] Ljung, L.: *System Identification*. Cambridge Press, 1985.
- [5] Dugard, L. y Landau, I.: *Recursive Output Error Identification Algorithms*. Automatica, 1980.
- [6] DeRusso, P.: *State Variables for Engineers*. Prentice Hall, 1999
- [7] Strejc, V.: *State Space Theory of Discrete Linear Control*. Wiley, 1981.
- [8] Nelles O.: *Nonlinear System Identification*. Springer, 2000.
- [9] *Redes Neuronales Artificiales: Fundamentos, Modelos y Aplicaciones*.
- [10] Haykin: *Neural Networks a Comprehensive Foundation*., 1994

1 PROPUESTA DE DIAGNÓSTICO DE FALLAS

1.1 Introducción

En este capítulo se da una breve introducción al los conceptos más frecuentes al hablar de diagnóstico de fallas. A su vez, se explica el concepto de diagnóstico de fallas, sus principales elementos y formas de implantación en sistemas dinámicos. El objetivo principal del capítulo es describir teóricamente el modelo de diagnóstico de fallas a implantarse experimentalmente.

En la sección 2.2, se explica la importancia de los procesos de diagnóstico de fallas en la seguridad de los sistemas industriales. Se abordan los elementos que constituyen un modelo de diagnóstico y la terminología asociada al mismo. Si bien, la implantación de un modelo para el diagnóstico de fallas en un proceso industrial incrementa su confiabilidad, el poder tener un modelo robusto no es tarea fácil. El problema de la robustez también se aborda en esta sección.

En la sección 2.3 se da un panorama general de las diversas alternativas en la implantación de modelos de diagnóstico para finalizar con la elaboración del modelo a aplicarse en el sistema en estudio. Se describen las partes que lo conforman así como su función dentro del modelo y el funcionamiento global del mismo. Finalmente, en la sección 2.3 se concluye expresando los posibles alcances del modelo.

1.2 Diagnóstico de Fallas

Para mantener un alto nivel de seguridad, desempeño y viabilidad en procesos controlados es importante que los errores del sistema, las fallas y la operación anormal del sistema sean detectados a tiempo y que el origen y la gravedad del mal funcionamiento sean diagnosticados para que puedan ser tomadas acciones correctivas.

En el campo de la seguridad en los sistemas controlados, el proporcionar al operador información de forma rápida y en un formato adecuado resulta una prioridad. Una falla simple puede convertirse en una falla múltiple si el operador ignora o malinterpreta las señales de alarma. Dado que las consecuencias son la pérdida de seguridad, la degradación ambiental o pérdidas económicas, lo que es realmente importante es la detección de comportamientos anormales del sistema, llamado *detección de cambios*. De manera más usual se usa el término de *detección de fallas* englobando tanto los cambios en el sistema como las fallas en sus componentes.[1] En realidad, una falla es entendida como un cambio indeseable en el sistema que tiende a degradar el desempeño global del sistema, aún cuando no represente un malfuncionamiento de los componentes físicos.

La detección de fallas desde un punto de vista simplista consiste en una decisión binaria: algo está mal o todo está bien. Expresado de esta manera parece demasiado trivial que existe un campo de investigación al respecto. La realidad es que para que sea útil, la presencia de las fallas debe conocerse antes de que se vuelvan serias. El desconocimiento del verdadero comportamiento del sistema, perturbaciones desconocidas y mediciones imprecisas, todas combinadas hacen de la tarea de detección temprana de fallas algo imprecisa y difícil de concretar en su totalidad. Entre más conocimiento se tenga sobre el proceso, sea este matemático, cualitativo o simbólico, mejor será la oportunidad de obtener una detección de la falla certera y robusta. Es este requerimiento del uso de toda la información disponible del sistema lo que hace este campo tan atractivo y un reto.

Un esquema de detección de fallas robusto es por definición capaz de la detección confiable de pequeñas o tempranas fallas en los componentes del sistema antes que los operadores humanos o la protección automática del sistema intervengan. Los métodos para la detección temprana de fallas suaves (conocidas como fallas incipientes) o de pequeñas desviaciones en los procesos de la planta, abre un campo de retos interesantes y posibilidades para el monitoreo del desempeño de la planta y mantenimiento preventivo.

En una aplicación real, la detección de la falla es seguida por el aislamiento de la falla, tarea en la cual el origen de la falla es determinado, por ejemplo, qué sensor o actuador tiene un mal funcionamiento. El proceso combinado de detección y aislamiento es referido en la literatura de la ingeniería de control como la función FDI (*fault detection and identification*). Un proceso de monitoreo el cual sea usado para detectar y aislar las fallas y cuantificar su gravedad es llamado *sistema de diagnóstico de fallas*. [1]

Los cuatro procedimientos asociados con el proceso de monitoreo son: detección de falla, identificación de falla, diagnóstico de falla y recuperación del proceso; como se muestra en la Figura 1.1.[2] Dado que en la literatura no existe un estándar para definir estos términos se adopta la terminología dada por Raich y Cinar.[3]

1. **Detección de la falla** es determinar si una falla ha ocurrido.
2. **Identificación de la falla** es determinar las variables de observación mas relevantes en el diagnóstico de la falla.
3. **Diagnóstico de la falla** es determinar qué falla ocurrió, en otras palabras, determinar la causa de las variaciones en las observaciones.
4. **Recuperación del proceso** también llamada *intervención*, es remover el efecto de la falla.

El aislamiento de las fallas incipientes lleva consigo su problema de robustez. Este problema ciertamente no es el mismo que el de robustez asociado a la detección ya que una falla puede ser detectada en una manera bastante robusta pero el procedimiento usado puede proveer un ambiguo aislamiento de la falla. Si hay un error en el aislamiento de la información (la falla es localizada incorrectamente) las consecuencias pueden ser serias con la implicación de que la etapa de detección de la falla resultó inútil.

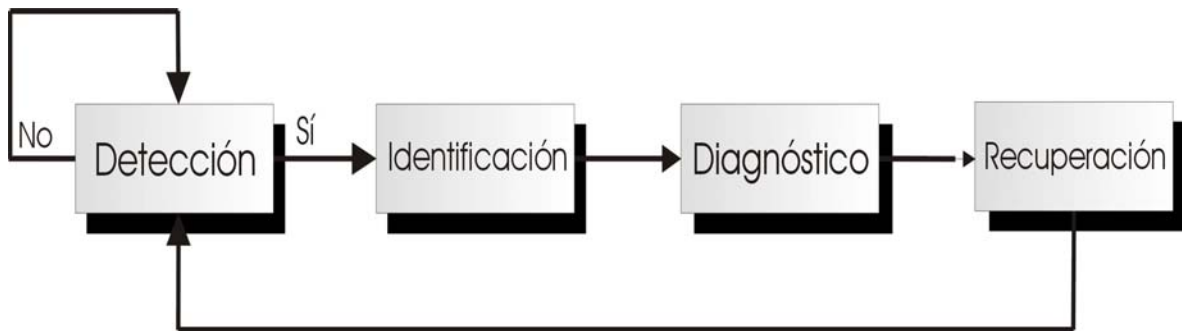


Figura 1.1 Proceso de monitoreo.

El mecanismo FDI puede implantarse usando replicación del *hardware* del sistema (computadoras, sensores, actuadores, etc.) en lo que es llamado redundancia del hardware: las salidas de componentes idénticos son comparadas. De forma alternativa, FDI se puede implantar usando información analítica o funcional del sistema monitoreado, por ejemplo, basado en un modelo matemático del sistema. Esta última aproximación es conocida como redundancia analítica o también diagnóstico de fallas con base en modelo.[1]

Con el uso de información matemática, el principio básico de FDI es comparar la respuesta actual del sistema y la reconstruida por el modelo matemático del sistema. A pesar de que esto es usualmente

asociado a modelado cuantitativo, la comparación puede llevarse a cabo mediante técnicas de modelado cualitativo. La combinación de métodos cuantitativos y cualitativos puede proporcionar un poderoso acercamiento al diagnóstico de fallas en una aplicación real.

La finalidad de una detección y diagnóstico temprano de las fallas es la de tener suficiente tiempo para contrarrestarlas sea reconfigurando el sistema, con mantenimiento o reparación de los componentes.

1.3 Modelo de Diagnóstico de Fallas Experimental

Hay dos rutas de acercamiento para el diagnóstico de fallas: basándose en modelos analíticos o en modelos heurísticos.

En la primer acercamiento, son usados modelos cuantitativos (ecuaciones diferenciales, espacios de estados, funciones de transferencia, etc.), los cuales generalmente utilizan resultados del campo de la teoría de control. Estos métodos están basados en la estimación de parámetros, estimación de estados o conceptos de paridad de estados. La filosofía de este acercamiento es que una falla causa cambios a ciertos parámetros físicos o estados. Monitoreando los parámetros estimados o los estados es entonces posible detectar y aislar cada falla. En esta aproximación es necesario tener un conocimiento *a priori* sobre las relaciones entre el sistema, el punto donde se considera actúan sus fallas y los cambios en los parámetros o estados producidos. Sin embargo, son difíciles de obtener modelos teóricos comprensibles o en algunos casos imposibles de derivar. Los modelos matemáticos pueden consumir mucho tiempo en su desempeño y rara vez pueden replicar las funciones del proceso por completo.

La segunda ruta está basada en el uso de métodos de inteligencia artificial. La aproximación formal usa razonamiento y modelado cualitativo. Para los acercamientos basados en simulación cualitativa, son usados modelos cualitativos del proceso para predecir el comportamiento del proceso bajo condiciones normales de operación y también bajo ciertas condiciones de falla. La detección de fallas se desempeña entonces comparando el comportamiento predicho con las observaciones actuales. En el campo del diagnóstico de fallas usando la inteligencia artificial se usan redes neuronales, lógica difusa y métodos neuroborrosos.

El modelo de diagnóstico de fallas a implantarse en el sistema de estudio, el cual se describe en el tercer capítulo, es un modelo híbrido el cual combina redundancia analítica

con el uso de observadores de espacios de estados, e inteligencia artificial con el uso de una red neuronal.

Inicialmente, se aborda el modelo analizando cómo detecta una falla un sistema con redundancia analítica mediante el uso de observadores lineales y el análisis de residuos.

Los residuos son variables las cuales son cero en circunstancias ideales, los residuos son diferentes de cero como resultado de una falla, ruido o errores en el modelo matemático del sistema.

El objetivo es la detección de fallas en un sistema dinámico mediante la medición de la entrada disponible y las variables de salida $U(t)$ y $Y(t)$ (Ver Figura 1.2). Se considera que el proceso trabaja en una configuración de lazo abierto. El proceso de parámetros concentrados y lineal es descrito mediante la ecuación diferencial,

$$y(t) + a_1 y^{(1)}(t) + \dots + a_n y^{(n)}(t) = b_0 u(t) + b_1 u^{(1)}(t) + \dots + b_m u^{(m)}(t)$$

$$y(t) = Y(t) - Y_{00}, u(t) = U(t) - U_{00}$$

donde U_{00}, Y_{00} son valores en estado estable conocidos y $y^{(n)}(t) = d^n y(t) / dt^n$.

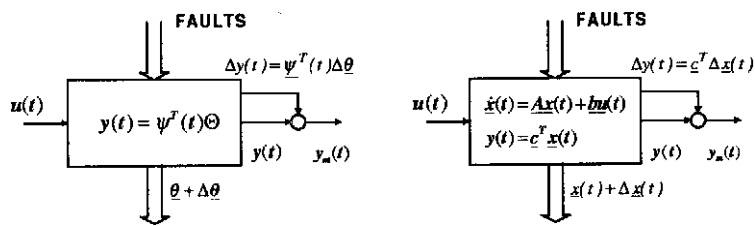


Figura 1.2 Modelo afectado por fallas.

Fallas aditivas a la entrada o a la salida se muestran en la Figura 1.3 y pueden ser modeladas por,

$$y(t) + a_1 y^{(1)}(t) + \dots + a_n y^{(n)}(t) = b_0 u(t) + b_1 u^{(1)}(t) + \dots + b_m u^{(m)}(t) + f_y + b_0 f_u$$

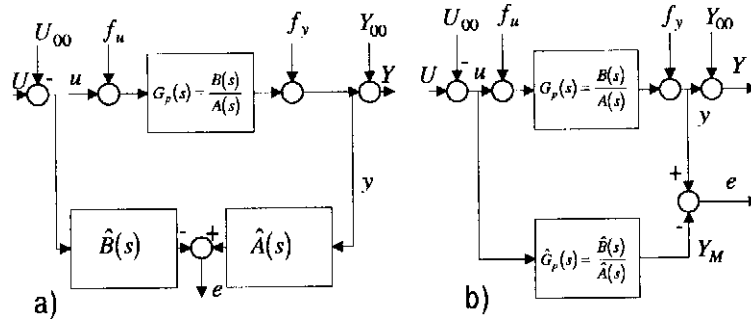


Figura 1.3 Sistema afectado por fallas a la entrada y a la salida.

El proceso lineal puede ser descrito a su vez en su forma de variables de estado como,

$$\begin{aligned} \dot{x} &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \quad (2.1)$$

Asumiendo que la estructura así como todos los parámetros A , B , C del proceso son conocidos, un observador de los estados es usado para reconstruir las variables de estado basado en la entrada y la salida medidas como se muestra en la Figura 1.4,

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x}(t) + Bu(t) + He(t) \\ e(t) &= y(t) - C\hat{x}(t) \end{aligned} \quad (2.2)$$

donde $e(t)$ es el error a la salida como se muestra en la Figura 1.3. Para la estimación del error de estado a partir de,

$$\begin{aligned} \hat{\tilde{x}}(t) &= x(t) - \hat{x}(t) \\ \dot{\hat{\tilde{x}}}(t) &= [A - HC]\hat{\tilde{x}}(t) \end{aligned}$$

el error de estado disminuye asintóticamente,

$$\lim_{t \rightarrow \infty} \hat{x}(t) = 0$$

cuando el observador es estable, lo cual se consigue con un diseño apropiado de la retroalimentación del observador H . El proceso es ahora influenciado por las perturbaciones y fallas como sigue,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + Fv(t) + Lf_L(t) \\ y(t) &= Cx(t) + Nw(t) + Mf_M(t) \end{aligned} \quad (2.3)$$

Es importante notar que $v(t)$ y $w(t)$ representan las perturbaciones no medibles a la entrada y la salida respectivamente, mientras que $f_L(t)$ indica la falla de entrada actuando a través de L en $x(t)$ y $f_M(t)$ indica la falla en la salida actuando a través de M como un cambio en la salida $\Delta y(t)$.

La diferencia $e(t)$ entre la salida y la estimación del observador expresada en (2.2) representa el residuo del sistema.

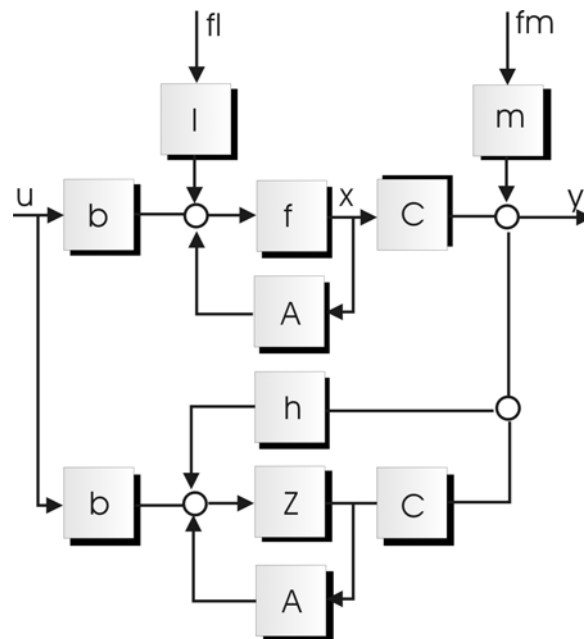


Figura 1.4 Sistema y observador.

Para disminuir la influencia del ruido en el residuo del sistema, es necesario dar cierto tratamiento estadístico a las perturbaciones $v(t)$ y $w(t)$. El origen de $w(t)$ es atribuido a ruido dentro del sistema (sensores, actuadores, etc.) el cual puede ser determinado mediante mediciones y obtener sus características estadísticas. El origen de $v(t)$ es ajeno al sistema, son perturbaciones externas y aleatorias.

En el modelo propuesto el tratamiento del ruido se da modelando al sistema bajo una estructura **ARMAX**, esto es,

$$\begin{aligned} y(t) + a_1 y(t-1) + \dots + a_{n_a} y(t-n_a) \\ = b_1 u(t-1) + \dots + b_{n_b} u(t-n_b) + e(t) + c_1 e(t-1) + \dots + c_{n_c} e(t-n_c) \end{aligned}$$

con las siguientes consideraciones:

- 1) Los términos $e(t) + c_1 e(t-1) + \dots + c_{n_c} e(t-n_c)$ describen el comportamiento de las perturbaciones $w(t)$, esto es,

$$w(t) = e(t) + c_1 e(t-1) + \dots + c_{n_c} e(t-n_c)$$

donde la matriz C contiene las características estadísticas que describen al ruido.

- 2) Las perturbaciones $v(t)$ son consideradas ruido blanco, con las características propias del ruido blanco, esto es

$$\sigma = 0$$

A partir de este tratamiento de las perturbaciones y considerando un modelo adecuado del sistema, se puede considerar que las contribuciones en el residuo $e(t)$ provienen en su mayor parte de las fallas. Entonces, un patrón lógico es generado mostrando cuales residuos

pueden ser considerados normales y cuales indican una falla. Un patrón como éste es llamado la *forma* o el *sello de la falla*. [1]

El paso final del procedimiento es el análisis de los patrones lógicos o sellos de la falla obtenidos a partir de los residuos y, para el modelo propuesto, los estados estimados del sistema los cuales se consideran también como residuos. El procesamiento de los residuos se realiza a través de la red neuronal. Cuando una red neuronal es entrenada, la relación entre las fallas y sus causas puede ser identificada y almacenada como pesos de la red. Esta conclusión da la pauta para la elección de la red neuronal, cuyos procedimientos de aprendizaje y entrenamiento deben adecuarse al resultado esperado: una clasificación de las fallas. La red entrenada puede ser usada entonces para diagnosticar fallas asociando el mal funcionamiento observado con la falla correspondiente.

Resumiendo, el observador permite la detección de la falla mientras que la red neuronal aísla la falla identificándola obteniéndose como resultado un sistema de diagnóstico de fallas completo como el mostrado en la Figura 1.5.

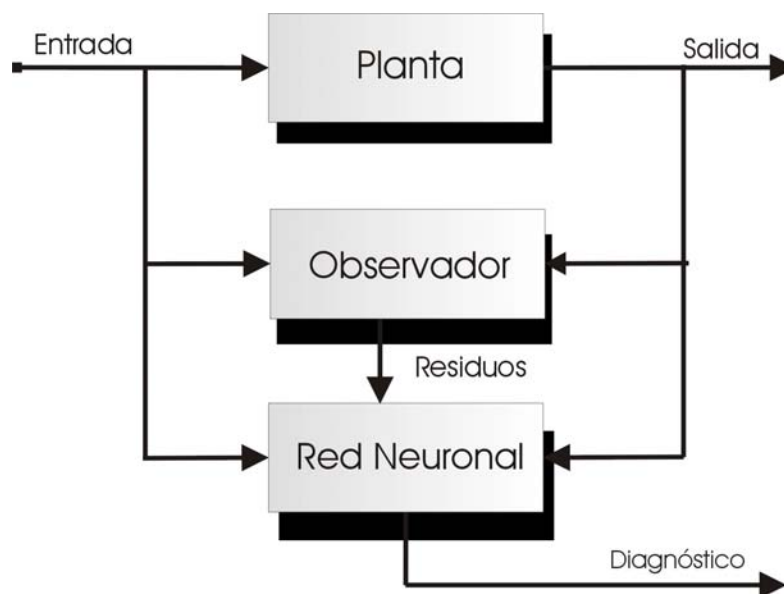


Figura 1.5 Esquema básico del modelo de diagnóstico de fallas.

Es la Figura 1.5 la base para el modelo a implantarse en el sistema el cual se muestra en la Figura 1.6.

En este modelo (el de la Figura 1.6), un observador de tipo no lineal el cual es un filtro de Kalman es acoplado al modelo ARMAX de la planta, de esta forma el modelo adquiere robustez respecto a los ruidos principalmente $v(t)$. Las entradas del observador son la entrada $u(t)$ y la salida $y(t)$ de la planta y sus salidas, es decir los residuos, son la estimación de la salida $\hat{x}(t)$ y el error de estimación $e(t)$.

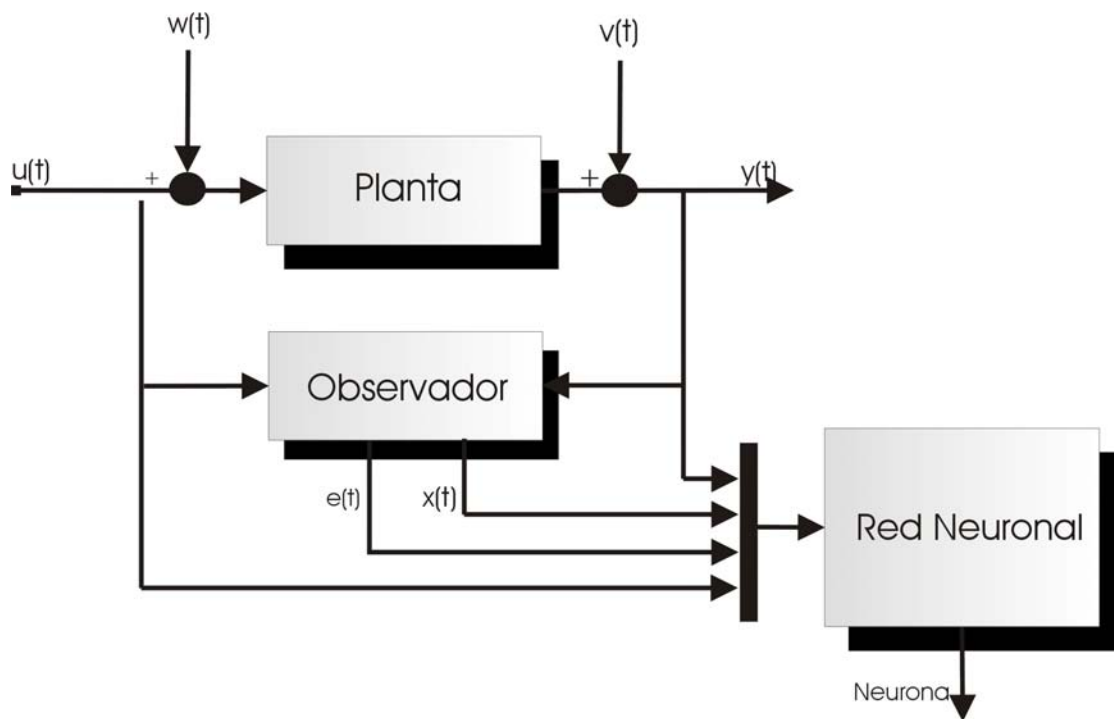


Figura 1.6 Modelo experimental de diagnóstico de fallas.

La red neuronal es una red SOM. Esta red, cuyas características principales fueron analizadas en el primer capítulo ofrece un comportamiento adecuado para proporcionar el resultado esperado que es, una clasificación de escenarios. Esta red es entrenada para un número finito de escenarios posibles del sistema entre ellos el comportamiento estable del sistema y ciertas fallas características.

Al ser la red SOM una red con aprendizaje no supervisado y auto organizable ofrece la ventaja de que durante el entrenamiento la clasificación de los escenarios se realiza sin intervención externa, quedando en función de los parámetros establecidos en el algoritmo de programación de la red.

Entrenada la red, es colocada en el sistema como una función evaluadora. En línea con el sistema la red no entrega a la salida una clasificación de los escenarios sino un valor de pertenencia a los mismos. Este valor está definido por la neurona de salida activada por el patrón de entrada: la neurona ganadora. Resulta evidente que esta asociación de la neurona ganadora con el escenario requiere de un conocimiento previo por parte de la persona o el algoritmo que supervise al sistema. Este conocimiento previo se da desde el momento del entrenamiento de la red, cuando se determina que tipo de fallas entrenar.

El resultado de la red se puede evaluar de la siguiente forma: para un escenario entrenado, la red debe entregar un patrón característico conocido de las neuronas de salida activadas en función de que el vector de pesos asociado a dicha neurona debe ser una buena estimación del vector de excitación o patrón de entrada, mientras que para un escenario desconocido el comportamiento de la red debe ser errático en algún sentido.

El patrón de excitación de la red neuronal, está constituido además de por los residuos, por la entrada y la salida del sistema. Este patrón es un vector de la forma (2.4). La forma de este vector es la misma utilizada como entrada de la red durante su entrenamiento.

$$x_p(t) = [u(t), y(t), \hat{x}(t), e(t)]^T \quad (2.4)$$

Finalmente, puede considerarse el conjunto observador y red neuronal una función **FDI** híbrida (analítica y heurística) quedando definido de manera teórica el modelo de diagnóstico de fallas a evaluarse en el sistema mecánico.

1.4 Conclusiones

A partir de lo expuesto en el capítulo, se puede concluir que un modelo de diagnóstico de fallas debe ser un conjunto de elementos de monitoreo interrelacionados con el objeto de ser una función discriminante entre aquello que es permisible dentro del funcionamiento de un proceso y aquello que no, es más, se puede afirmar que el proceso de diagnóstico de fallas no concluye solamente en ser la función discriminante (proceso de identificación) sino que va más allá al proporcionar información acerca de las características de aquellos comportamientos que afectan el buen funcionamiento del sistema.

El modelo expuesto pretende involucrar las cualidades mencionadas. Éste puede considerarse limitado en cuanto a la forma en que la información acerca de la falla es proporcionada, consecuencia de la interpretación que de los resultados de la red neuronal se debe hacer, pero resulta un vínculo apropiado para cualquier otra aplicación con un mayor refinamiento en la interpretación y visualización de los resultados, al contener suficiente información sobre las características del sistema.

1.5 Referencias

- [1] Patton, R.: *Issues of Fault Diagnosis for Dynamic Systems*. Springer, 2000.
- [2] Chiang, L.: *Fault Detection and Diagnosis in Industrial Systems*. Springer, 2001
- [3] Cinar, A.: *Statistical Process Monitoring and Disturbance Diagnosis*. Springer, 1996.

1 MODELO DE ESTUDIO

1.1 Introducción

En el desarrollo de este capítulo, se ponen en práctica los conocimientos teóricos descritos a lo largo del texto implantándose en un sistema dinámico real. Estos son: la obtención del modelo analítico del sistema, su identificación experimental, la implementación de un algoritmo de detección de fallas y la de uno de diagnóstico, con el objeto de describir de forma global tanto al sistema de tipo experimental a estudiar como a los algoritmos que en el se han de implantar.

El sistema dinámico en estudio es un sistema mecánico compuesto esencialmente por dos elementos: un actuador lineal, el cual es un transductor de un par motor a una fuerza longitudinal y una masa de carga. El sistema se muestra en la Figura 1.1.

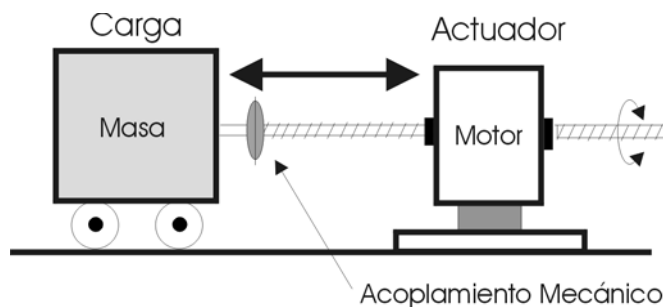


Figura 1.1 Sistema dinámico.

El actuador es un motor a pasos modificado, el cual es controlado mediante un tren de pulsos. El desplazamiento de la masa es medido mediante un **LVDT** (*linear voltage transformer*) que es un sensor de desplazamiento longitudinal de respuesta lineal y con un principio de funcionamiento de carácter electromagnético. Tanto las señales de control como las mediciones obtenidas son procesadas en una computadora personal a través de una tarjeta de adquisición de datos en tiempo real y bajo un ambiente gráfico en MATLABTM completándose de esta manera los elementos físicos involucrados en los procesos de experimentación y evaluación de los algoritmos. El esquema global del sistema se muestra en la Figura 1.4.

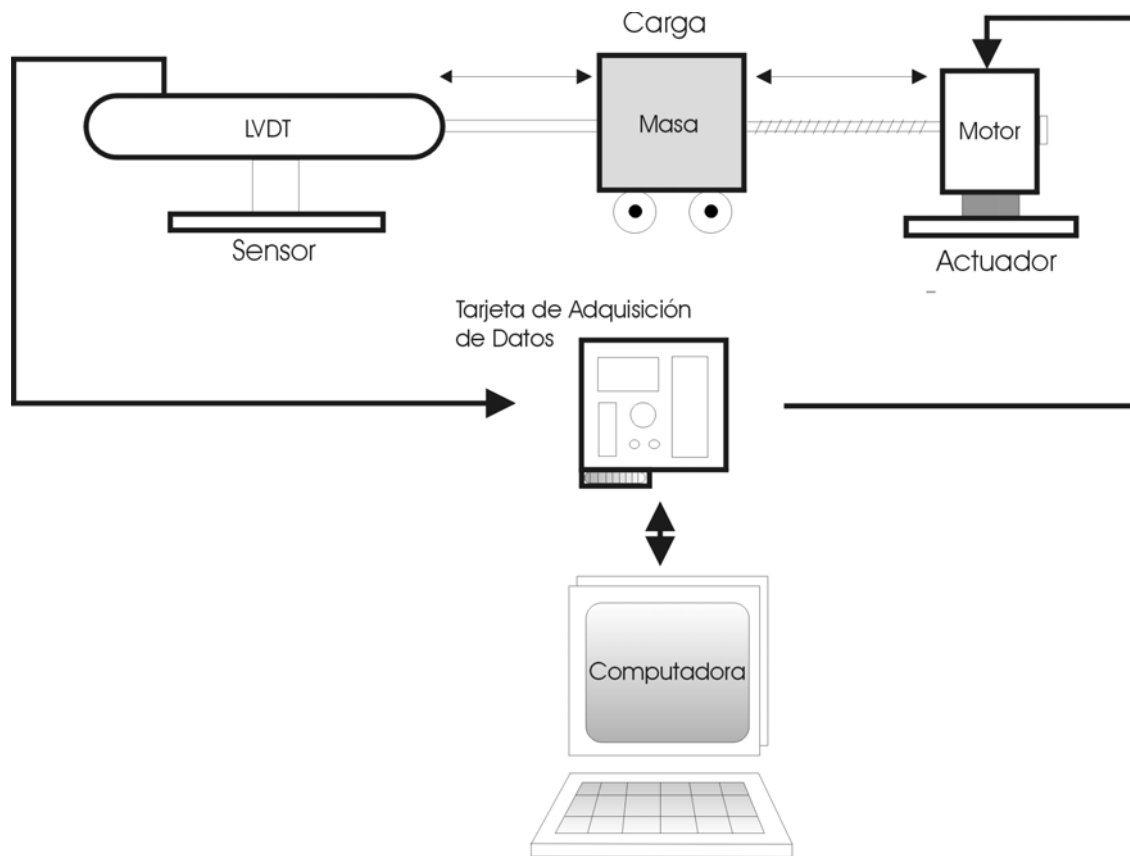


Figura 1.2 Sistema global.

En la primera sección de este capítulo se describen los experimentos necesarios para la caracterización de los dispositivos físicos transductores, así como los experimentos enfocados en la obtención de información adecuada y útil para el proceso de identificación.

Posteriormente la identificación es realizada en la sección 3.3 concluyendo con la obtención de un modelo matemático de tipo **ARMAX** que describe al sistema en estudio.

1.2 Elaboración del Experimento y Obtención de Datos

1.2.1 Descripción General del Modelo y los Experimentos

Como se menciona en el primer capítulo, es importante analizar toda la información que sobre el proceso se pueda tener *a priori* para, la elaboración de hipótesis sobre su comportamiento, la elección de un conjunto estructuras que lo modelen y la elaboración de experimentos que sean adecuados para la obtención de datos a través de los cuales validar el o los modelos candidatos de una estructura.

En el proceso de síntesis del modelo para el caso de estudio, el cual es propuesto como un modelo analítico de identificación experimental, primero resulta necesario analizar las leyes físicas de importancia que rigen al sistema. Este análisis se realiza a partir del diagrama mostrado en la Figura 1.3. El sistema mecánico representado, describe a un sistema dinámico de segundo orden, fácilmente demostrable mediante el análisis de fuerzas de las leyes de elementos y la síntesis de éstas en las leyes de conjunto.

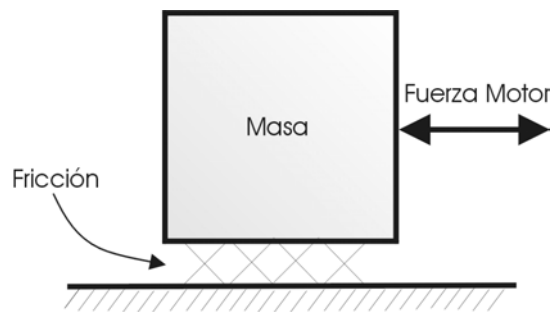


Figura 1.3 Fuerzas que intervienen en el Sistema.

Sean f_M , f_B y f_{motor} fuerzas cuyas leyes de elemento describen el comportamiento de la masa M . Estas fuerzas son entonces la inercia de la masa, la fricción cinética y la fuerza del motor en función de su par T ,

$$\begin{aligned}f_M(t) &= M\ddot{x}(t) \\f_B(t) &= B\dot{x}(t) \\f_{motor}(t) &= F(T, t)\end{aligned}\tag{3.1}$$

y de acuerdo con la tercera ley de Newton establecemos la siguiente ley de conjunto,

$$\sum f(.) = 0\tag{3.2}$$

sintetizando en (3.2) las ecuaciones descritas en (3.1) y de acuerdo con la Figura 1.3

$$f_M(t) + f_B(t) - f_{motor}(t) = 0\tag{3.3}$$

o bien

$$\ddot{x}(t) + \frac{B}{M}\dot{x}(t) = \frac{1}{M}f(T, t)\tag{3.4}$$

siendo $x(t)$ la variable desplazamiento en función del tiempo.

El modelo del sistema a obtener, de acuerdo al estudio propuesto, debe incorporar como entrada la frecuencia del tren de pulsos que excita al motor, y como salida el desplazamiento consecuencia del movimiento traslacional de la masa de carga. Son necesarias entonces, modificaciones al modelo básico descrito en (3.4) para incorporar estas características.

Es necesario cumplir entonces dos premisas: primero, la obtención de una función que nos permita relacionar la frecuencia del tren de pulsos con la fuerza longitudinal obtenida en el actuador lineal y segundo, la transformación del modelo matemático para que esté presente en él la variable de interés a la salida $x(t)$.

Estas modificaciones dificultan en sí el proceso de síntesis del modelo, a lo que hay que añadir la experimentación necesaria para obtener el valor de los coeficientes asociados a las características físicas del sistema: M y B . Sin las herramientas y el equipo necesario, salvar estos obstáculos resulta algo improbable, así pues, abandonaremos en parte esta aproximación de tipo físico experimental, pero sin dejar atrás la estructura básica obtenida en (3.4).

Ahora se procede a buscar una función que relacione la frecuencia del tren de pulsos $Frec_{pulso}$ y la fuerza lineal del actuador f de la forma $f_{motor} = f(Frec_{pulso})$ con base en las siguientes premisas:

- 1) La fuerza longitudinal está en función del par del motor T de manera directa y proporcional, esto es:

$$f_{motor} = kT_{motor}$$

- 2) El par motor es una función dependiente del momento de inercia J y la aceleración angular $\ddot{\theta}$, así como de la fricción f y la velocidad angular $\dot{\theta}$. [1] Al mismo tiempo existe una relación lineal escalada por el “tamaño” del paso del motor entre la frecuencia del tren de pulsos y la velocidad angular.

$$T = J\ddot{\theta}(f_{pulso}) + f_{friccion}\dot{\theta}(f_{pulso})$$

Ahora bien, hay que modificar la estructura de (3.4) de tal forma, que la variable $x(t)$ esté presente sin efectuar un proceso de integración. Este es el caso de modelar la masa como un sistema de masa, resorte y amortiguador como se muestra en la Figura 1.4.

Simultáneamente se incorpora la función $f(Frec_{pulso})$ justificada anteriormente y se obtiene la descripción del sistema

$$\ddot{x}(t) + \frac{B}{M} \dot{x}(t) + \left(\frac{K}{M} + \frac{B_M}{M} \right) x = \frac{1}{M} f(f_{pulso}, t)$$

y mediante el siguiente cambio de variables

$$a_1 = \frac{B}{M}$$

$$a_2 = \left(\frac{K}{M} + \frac{B_M}{M} \right)$$

$$b = \frac{1}{M}$$

obtenemos la descripción del sistema

$$\ddot{x} + a_1 \dot{x} + a_2 x = bf(f_{pulso}, t) \tag{3.5}$$

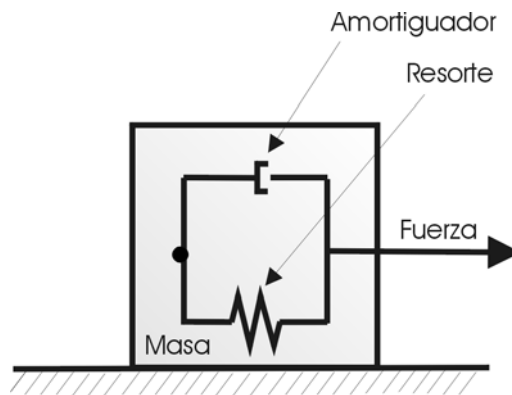


Figura 1.4 Modelado de la masa.

La representación de la masa como un sistema masa, resorte, amortiguador puede parecer arbitraria, pero para los fines de identificación, introducirlo resulta una forma muy práctica de solucionar el problema de aislar la variable de salida de interés, justificándose

como contribuciones a las fuerzas existentes: inercia en el resorte y fricción en el amortiguador.[2]

A partir de la estructura (3.5) el proceso de identificación consiste en determinar el valor de los coeficientes a_1 , a_2 y b , los cuales son los parámetros del modelo. Antes de ello es necesaria una última consideración consecuencia de la naturaleza discreta que toman los datos durante su procesamiento.

Existen métodos para trasladar el modelo de un sistema de un espacio en el tiempo continuo a uno en el tiempo discreto involucrando el tiempo entre muestreos de las señales. Si se quiere respetar la relación directa existente entre los parámetros en la estructura y los coeficientes físicos, lo conveniente es utilizar alguno de estos métodos para trasladar dicha estructura a un espacio del tiempo discreto. Un acercamiento propuesto por Ljung [3], es el trasladar la descripción obtenida en ecuaciones diferenciales a una en variables de estado de la forma,

$$\begin{aligned}\dot{x}(t) &= F(\theta)x(t) + G(\theta)u(t) \\ \theta &= [a_1, a_2, b]^T\end{aligned}\tag{3.6}$$

y suponiendo que la entrada es constante durante el intervalo de muestreo T

$$u(t) = u_k = u(kT), \quad kT \leq t < (k+1)T$$

Entonces la representación descrita en (3.6) puede ser resuelta de $t = kT$ a $t = kT+T$ por

$$x(kT + T) = A_T(\theta)x(kT) + B_T(\theta)u(kT)\tag{3.7}$$

donde

$$A_T(\theta) = e^{F(\theta)T}$$

$$B_T(\theta) = \int_{\tau=0}^T e^{F(\theta)\tau} G(\theta) d\tau$$

Para los fines que se persiguen en este trabajo lo importante es la obtención de una descripción confiable del proceso sin importar que exista relación directa alguna entre los parámetros del modelo y los coeficientes físicos del sistema. La descripción discreta que se obtiene a partir de (3.7) resulta una función matemáticamente más compleja del vector de parámetros θ que aquella obtenida mediante un modelo **ARX** o **ARMAX** directamente parametrizado en términos de θ aún cuando éstos pudieran contener más parámetros. Simplemente consideraremos la “forma” y el orden de (3.5) para trasladarla a un espacio de tiempo discreto, obteniendo la estructura,

$$y(t) - a_1 y(t-1) - a_2 y(t-2) = b_1 u(t-1) \quad (3.8)$$

siendo ésta entonces la estructura básica a identificar.

Determinada la estructura se describen los experimentos necesarios para la obtención de los datos que permitan tener la información necesaria para determinar los parámetros involucrados agrupándolos en tres grandes bloques:

- 1) Aquellos que permitan verificar las características de interés especificadas por el fabricante de los elementos de entrada y salida, que para el caso del actuador es el intervalo de respuesta lineal del desplazamiento longitudinal en función de la frecuencia del tren de pulsos y para el sensor el intervalo de respuesta lineal de la diferencia de potencial al desplazamiento.
- 2) Experimentos de respuesta a entrada escalón, para determinar el nivel de ruido y su comportamiento, así como el de los transitorios para el caso de que los resultados arrojen algún interés en su análisis.

- 3) Experimentos de respuesta a entradas variables en el tiempo, los cuales serán en definitiva los que proporcionen la información necesaria para determinar el valor de los parámetros.

1.2.2 Caracterización de los Elementos

Primero se comprueba la linealidad de la relación entrada-salida del sensor en el intervalo especificado por el fabricante, ya que el sensor se utiliza posteriormente para determinar las características del motor.

Distancia Medida	Volts DC Medidos (Laboratorio)	Volts DC Medidos (Fabricante)	Volts DC Calculados (Fabricante)	Desviación Calculada (Fabricante)
-1.0000	-10.1	-10.1104	-10.1161	0.0057
-0.8000	-8.1	-8.0902	-8.0922	0.0019
-0.6000	-6.1	-6.0842	-6.0682	-0.0160
-0.4000	-4.1	-4.0615	-4.0442	-0.1730
-0.2000	-2.0	-2.0284	-2.0202	-0.0083
+0.2000	2.1	2.0458	2.0278	0.0180
+0.4000	4.1	4.0871	4.0518	0.0353
+0.6000	6.1	6.1024	6.0758	0.0266
+0.8000	8.1	8.0935	8.0998	0.0064
+1.0000	10.1	10.0843	10.1238	0.0395

Tabla 1.1 Valores prueba del LVDT.

Para este fin, se acopla al núcleo metálico del LVDT una extensión que prolonga a éste fuera de la estructura fija del sensor como se muestra en la Figura 1.5. Ésta extensión es lo suficientemente grande para graduar en ella el intervalo de mediciones requerido y para poder manipular el núcleo.

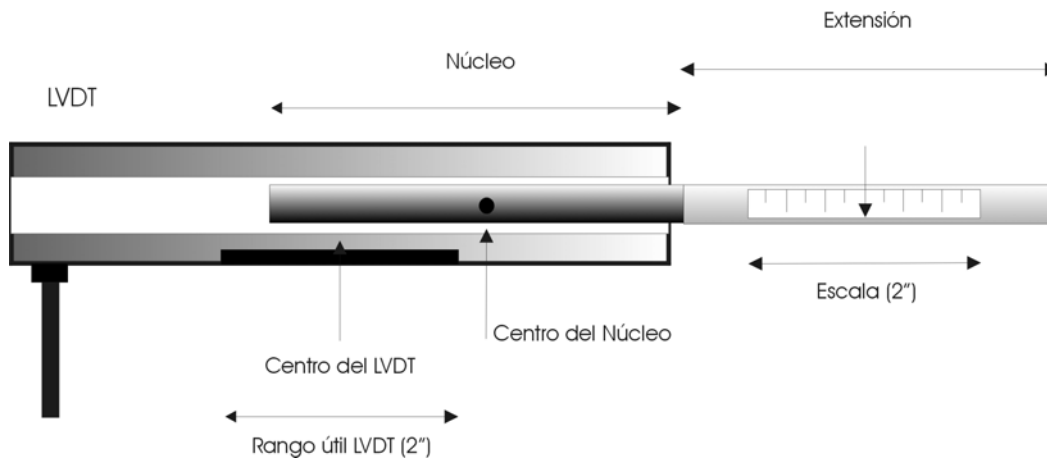
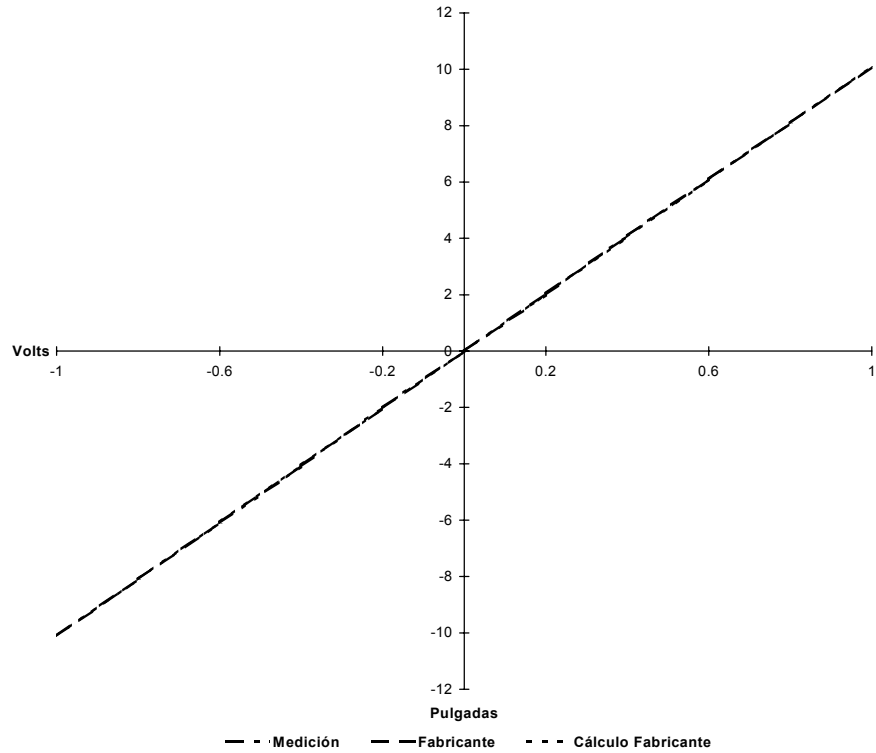


Figura 1.5 Adaptación del núcleo del LVDT.

Con ayuda del osciloscopio se calibra el núcleo en la posición cero, esto es diferencia de potencial a la salida del sensor 0 Volts, se marca la extensión y se gradúa en pulgadas. Con el dispositivo adaptado se obtienen las mediciones descritas en la Tabla 1.1 (la primer columna corresponde a las mediciones en laboratorio y el resto a los datos proporcionados por el fabricante) las cuales concuerdan con las especificadas por el fabricante de acuerdo con la Gráfica 1.1, donde no se aprecia diferencia significativa entre las series de datos y se observa la linealidad del sensor.

El factor de escala del LVDT es de 10.12 [Volts/Pulgada] con una linealidad de 0.2%.

En la caracterización del motor se realizan diversas pruebas. En la primer prueba a la entrada del motor, se consideran cuatro frecuencias: dos por debajo de la frecuencia mínima y dos por encima de la frecuencia máxima especificadas por el fabricante, así como 6 frecuencias dentro del intervalo establecido. Además se añadirán dos mediciones para determinar las frecuencias máxima y mínima para las cuales existe aún desplazamiento del eje del motor.

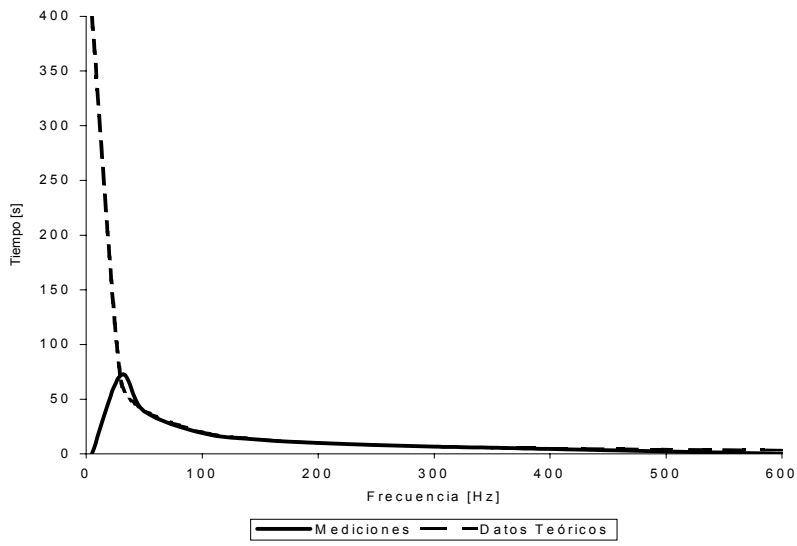


Gráfica 1.1 Respuesta del LVDT.

Frecuencia [Hz]	Tiempo 1 [s]	Tiempo 2 [s]	Tiempo 3 [s]	Tiempo 4 [s]	Tiempo 5 [s]	Promedio [s]	Calculado [s]
5	-	-	-	-	-	-	400.00
30	70.0	72.0	69.0	75.0	73.0	71.8	66.66
50	39.5	38.7	39.2	38.6	38.9	38.98	40.00
100	18.7	18.9	19.2	19.4	19.2	19.08	20.00
150	12.9	13.0	13.0	13.3	13.0	13.04	13.33
200	10.0	10.0	10.2	10.0	10.0	10.04	10.00
250	8.1	8.0	7.9	7.8	8.0	7.96	8.00
300	6.5	6.6	6.6	6.7	6.5	6.58	6.66
350	5.6	5.7	5.6	5.5	5.6	5.6	5.71
600	-	-	-	-	-	-	3.33

Tabla 1.2 Valores prueba del motor.

Para la obtención de los datos se mide el tiempo que tarda el eje en desplazar al núcleo del sensor dos pulgadas, registrándose cuatro mediciones por frecuencia cuya media se considera el valor característico de la muestra obteniéndose la Tabla 1.2. Los datos medidos se comparan con aquellos obtenidos a partir de cálculo directo de acuerdo con las características especificadas para el motor. Los resultados comparativos se muestran en la Gráfica 1.2.



Gráfica 1.2 Respuesta del motor.

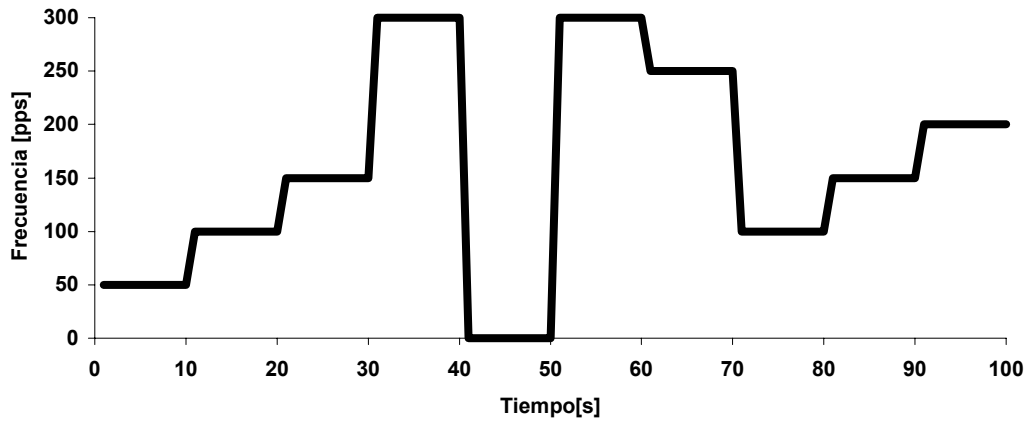
1.2.3 Caracterización del Sistema

Una vez obtenidas las relaciones de entrada salida de los elementos involucrados en el sistema, se expande el estudio a un nivel de interdependencia de los elementos para obtener así las relaciones de entrada salida de los elemento en conjunto, es decir del proceso en estudio.

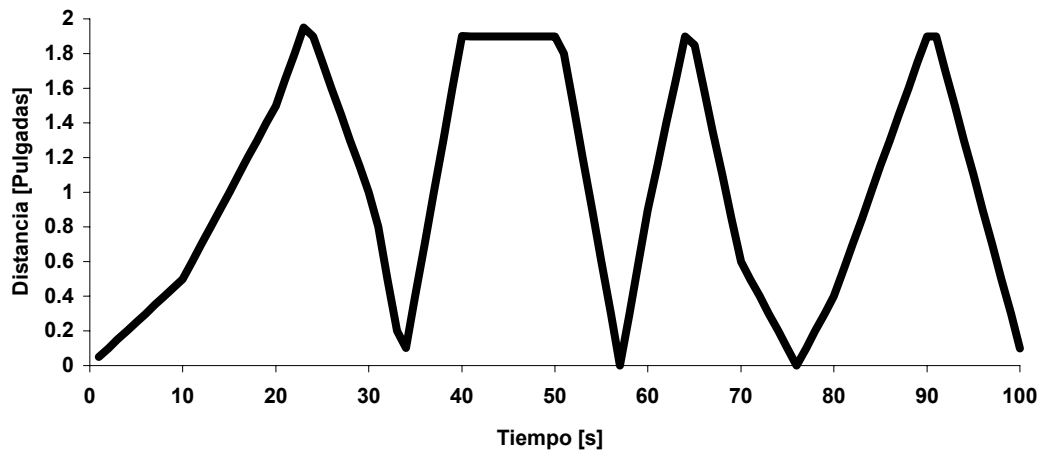
Como se menciona con anterioridad, las variables del sistema cuyas mediciones interesan son: a la entrada, la frecuencia del tren de pulsos que excita al motor de pasos y a la salida, el desplazamiento longitudinal de la masa de carga.

La caracterización del sistema consiste en un experimento muy simple: se excita al sistema mediante un tren de pulsos cuya frecuencia se manipula de forma aleatoria siempre dentro del intervalo de respuesta lineal obtenido para el motor. La respuesta a estos cambio es completamente perceptible ya que la capacidad de respuesta del sensor, el cual es esencialmente un dispositivo electromagnético, es considerablemente mayor a la capacidad de respuesta del actuador, el cual es un dispositivo electromecánico.

La obtención de las mediciones se describe a continuación: se realizan 5 grupos de mediciones compuestos cada uno de 100 mediciones entrada salida. Los grupos de mediciones corresponden a 5 días consecutivos de experimentación donde el sistema es sometido en todos los días a una misma entrada tipo la cual se muestra en la Gráfica 1.3 (a).



(a) Entrada tipo.



(b) Salida del sistema.

Gráfica 1.3 Entrada tipo y respuesta del sistema.

A partir de los resultados obtenidos al término de los cinco días de experimentación se obtiene el promedio aritmético entre los cinco grupos obteniéndose una salida promedio a la entrada tipo como se muestra en la Gráfica 1.3 (b).

1.3 Obtención del Modelo Analítico, el Modelo ARMAX

El algoritmo de diagnóstico de fallas tiene como objetivo detectar aquellos comportamientos anómalos del sistema que no sean atribuibles a desviaciones producidas por la sensibilidad del sistema al ruido. Por lo tanto entre las estructuras que se adecuan a este propósito son los modelo de tipo **ARX** o **ARMAX** descritos ya con anterioridad. Estas estructura nos proporciona cierta robustez en el modelo, convirtiéndolas en estructuras adecuadas por su tolerancia a las desviaciones de los resultados atribuibles a ruido, sea este inherente a los componentes del proceso $w(t)$, o de carácter externo y de inferencia en el sistema $v(t)$.

Los datos contenidos en la Gráfica 1.3 proveen de información suficiente para la obtención de los parámetros contenidos en la estructura de la forma descrita en **¡Error! No se encuentra el origen de la referencia.**, la cual define un modelo **ARMAX** que por su generalidad y utilidad será el modelo a desarrollar. Recapitulando, el modelo **ARMAX** esta definido por,

$$\begin{aligned} y(t) + a_1 y(t-1) + \dots + a_{n_a} y(t-n_a) \\ = b_1 u(t-1) + \dots + b_{n_b} u(t-n_b) + e(t) + c_1 e(t-1) + \dots + c_{n_c} e(t-n_c) \end{aligned}$$

y,

$$\theta = [a_1, a_2, \dots, a_{n_a}, b_1, b_2, \dots, b_{n_b}, c_1, c_2, \dots, c_{n_c}]^T$$

El procedimiento para la obtención de los valores del vector de parámetros θ consiste en aprovechar la estructura descrita en ¡Error! No se encuentra el origen de la referencia. la cual define una regresión pseudolineal de la forma,

$$\hat{y}(t | \theta) = \varphi^T(t, \theta) \theta$$

Esta regresión pseudolineal tiene una solución posible utilizando el método de los mínimos cuadrados donde φ es el vector de regresión definido por,

$$\varphi(t, \theta) = [-y(t-1) \dots -y(t-n_a), u(t-1) \dots u(t-n_b), \varepsilon(t-1, \theta) \dots \varepsilon(t-n_c, \theta)]^T \quad (3.9)$$

Para homologar el modelo (3.8) con el modelo ARMAX descrito por $y(t) - a_1 y(t-1) - a_2 y(t-2) = b_1 u(t-1) + e(t) + c_1 e(t-1) + c_2 e(t-2)$ de la referencia. basta adicionar a la estructura los términos relacionados a las perturbaciones obteniéndose el modelo,

$$y(t) - a_1 y(t-1) - a_2 y(t-2) = b_1 u(t-1) + e(t) + c_1 e(t-1) + c_2 e(t-2) \quad (3.10)$$

y siendo para este caso,

$$n_a = 2$$

$$n_b = 1$$

$$n_c = 2$$

el vector de regresión queda definido entonces como,

$$\varphi(t, \theta) = [-y(t-1), -y(t-2), u(t-1), \varepsilon(t-1, \theta), \varepsilon(t-2, \theta)]^T \quad (3.11)$$

con,

$$\theta = [a_1, a_2, b_1, c_1, c_2]$$

Con los elementos necesarios reunidos, se desarrolla entonces el algoritmo que nos permite determinar el valor de los elementos del vector θ y es la solución de la pseudoregresión lineal mediante el método de los mínimos cuadrados.

Se elabora una matriz M de n vectores $m(N)$ de regresión a partir de la cual el algoritmo recursivo de la pseudoregresión determina el valor de los parámetros contenidos en el vector θ . Los vectores $m(N)$ de medidas tienen la forma de (3.11) y la matriz M queda definida por,

$$M(N) = \begin{bmatrix} -y(1) & y(0) & u(0) & \varepsilon(1) & \varepsilon(0) \\ -y(2) & y(1) & u(1) & \varepsilon(2) & \varepsilon(1) \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ -y(N-1) & y(N-2) & u(N-1) & \varepsilon(N-1) & \varepsilon(N-2) \end{bmatrix}$$

Finalmente el método recursivo desarrollado se programa¹ en un archivo para uso en MATLABTM, los datos fuente utilizados para elaborar la Gráfica 1.3 en una matriz M y se desarrolla el cómputo de los datos obteniéndose los siguientes resultados:

$$\theta = [-0.0016, 0.0008, 0.025, -0.017, 0.3196]$$

A partir de los resultados obtenidos, el modelo ARMAX que describe al sistema queda descrito mediante la ecuación,

$$\begin{aligned} y(t) + -0.0016y(t-1) + 0.0008y(t-2) \\ = 0.025(t-1) + e(t) + -0.017e(t-1) + 0.3196e(t-2) \end{aligned} \quad (3.12)$$

1.4 Estimación del Filtro de Kalman

El modelo de segundo orden descrito en (3.12) es necesario transformarlo en su equivalente en variables de estado para determinar la ganancia del filtro de Kalman. Esta descripción en variables de estado tiene la forma,

$$\begin{aligned} x(t+1) &= F[x(t), u(t)] \\ y(t) &= G[x(t), u(t)] \end{aligned}$$

o bien,

$$\begin{aligned} x(t+1) &= A(t)x(t) + B(t)u(t) + \Gamma(t)w(t) \\ y(t) &= C(t)x(t) + v(t) \end{aligned}$$

¹ Ver Apéndice B, programa *armax.m*

donde el problema a solucionar es el de estimar los estados x basados en las t observaciones de Z como se explica en el primer capítulo. Como se menciona en el primer capítulo, la estrategia de estimación consiste en la repetición de los dos pasos siguientes:

- 1) Estimar $\hat{x}(t+1|k)$ y la covarianza $P(t+1|k)$ del vector de error de estimación.
- 2) Corregir las estimaciones del ciclo de propagación utilizando las medidas $y(k+1)$.

El carácter recurrente de la estimación es evidente. De esta forma, las ecuaciones mostradas en el Apéndice A, del ciclo de propagación **¡Error! No se encuentra el origen de la referencia.** y las del ciclo de actualización **¡Error! No se encuentra el origen de la referencia.** y **¡Error! No se encuentra el origen de la referencia.** permiten la estimación en línea de los estados del sistema. La estructura del algoritmo de estimación que realiza el filtro de Kalman puede ser la que se indica a continuación,

Inicio

Asignar valores iniciales $x(0)$, $P(0|0)$;

Para $t = 0$ hasta $t = N$ hacer:

Lectura de $u(t)$

|Propagación|

*Calcular $\hat{x}(t+1|t)$ mediante **¡Error! No se encuentra el origen de la referencia.**;*

*Calcular $P(t+1|t)$ mediante **¡Error! No se encuentra el origen de la referencia.**;*

|Actualización|

*Calcular ganancia $L(t+1)$ mediante **¡Error! No se encuentra el origen de la referencia.**;*

*Calcular $P(t+1|t+1)$ mediante **¡Error! No se encuentra el origen de la referencia.**;*

Esperar hasta el instante $k+1$;

Lectura $y(t+1)$;

*Calcular $\hat{x}(t+1|t+1)$ mediante **¡Error! No se encuentra el origen de la referencia.**;*

Terminar

Terminar

Es importante notar que la estimación $\hat{x}(t+1|t+1)$ no se dispone en el instante $t+1$ sino que es necesario realizar el cálculo **¡Error! No se encuentra el origen de la referencia.** a partir de la medida $y(t+1)$.

Sólo si el periodo de muestreo es suficientemente grande, será posible realizar en tiempo real la estimación del vector de estado. En caso contrario puede procederse al cálculo previo, fuera de línea de la ganancia L del filtro de Kalman.

Considerando este segundo enfoque, se supone que el muestreo de nuestra planta no es lo suficientemente grande para realizar el cálculo en línea del vector de estado ya que este cálculo en cada instante t está acompañado por el procesamiento de la red neuronal.

Para poder utilizar las ecuaciones del filtro de Kalman se necesita conocer la matriz R de covarianza de los ruidos $v(t)$ de las medidas, y la matriz Q de covarianzas de los ruidos $w(t)$ del sistema. La matriz R depende del sensor y puede ser identificada a partir de sus características y de experimentación. Esta matriz se obtiene mediante los resultados obtenidos a partir de la Tabla 3.1.

Ahora bien, la elección de la matriz Q apropiada resulta más difícil. Para simplificar este inconveniente, se recurre a la hipótesis de ruido blanco [4]. Físicamente Q está asociada a perturbaciones desconocidas que pueden ser escalones, ruido blanco o cualquier perturbación debida a imperfecciones en el modelo de la planta. Se ha supuesto entonces, que $w(t)$ es ruido blanco con las propiedades estadísticas ya descritas anteriormente.

A partir de estas consideraciones se observa que si se supone que $Q = 0$, para simplificar el problema de diseño, indicando que no existen ruidos que perturben al sistema, las ecuaciones **¡Error! No se encuentra el origen de la referencia.** y **¡Error! No se encuentra el origen de la referencia.** con $P(0|0) = 0$ dan lugar a una ganancia nula al filtro.

Retomando la ganancia del filtro, su cálculo previo es posible debido a que no son necesarias las medidas $y(t+1)$ sino que basta con la mera descripción del sistema. El cálculo de esta ganancia L se obtiene utilizando el algoritmo siguiente:

Inicio

Asignar valor inicial a $P(0 | 0)$;

Para $k= 0$ hasta $k = N$ hacer hacer:

*Calcular $P(k + 1 | k)$ mediante; **Error! No se encuentra el origen de la referencia.**;*

*Calcular $L(k + 1)$ mediante; **Error! No se encuentra el origen de la referencia.**;*

*Calcular $P(k + 1 | k + 1)$ mediante; **Error! No se encuentra el origen de la referencia.**;*

Terminar

El algoritmo propuesto se programa² y ejecuta en MATLABTM donde se obtienen el siguiente resultado:

$$L = [0.056]$$

siendo ésta la ganancia del filtro de Kalman para la planta propuesta.

1.5 Implantación de la Red Neuronal, una Red SOM

1.5.1 Introducción a los Algoritmos de la red SOM.

En esta sección la atención se centra en el proceso de definir los algoritmos de un simulador tipo SOM con base en las características de esta red descritas en el primer capítulo. Los primeros algoritmos son aquellos referentes a la propagación de las señales a través de la red para finalizar con aquellos concernientes al aprendizaje y el entrenamiento de la misma.

La red neuronal tiene la estructura mostrada en la Figura 1.6, esto es; es una red con dos capas de unidades o neuronas, donde las unidades de salida están distribuidas en un entramado bidimensional o

² Ver Apéndice B, programa *kalgan.m*

matriz. Las unidades en la capa de entrada están dispuestas como un vector que provee de estímulo al arreglo rectangular en el que están dispuestas las unidades de salida. Las unidades en la capa de salida están interconectadas para permitir inhibiciones o excitaciones laterales.

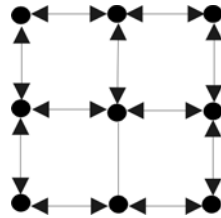


Figura 1.6 Interconectividad de los elementos de salida.

Para facilitar el procesamiento de los algoritmos de la red y su programación es necesario modificar la estructura descrita anteriormente. Con el fin de alcanzar este objetivo se aprovecha la capacidad iterativa que ofrece la elaboración de un programa de computadora desglosando los diferentes escenarios de la red en subrutinas o elementos constitutivos partiendo del algoritmo teórico para la red de Kohonen descrito por Freeman y Skapura [5].

Una primera aproximación es distinguir aquellos elementos básicos del algoritmo como se muestra en la Figura 1.7.

$$SOFM \left\{ \begin{array}{l} \text{propagación de la señal} \\ \text{aprendizaje} \\ \text{entrenamiento} \end{array} \right.$$

Figura 1.7 Elementos básicos del algoritmo SOM.

A partir del cuadro mostrado en la Figura 1.7 podemos elaborar un nuevo esquema con las rutinas a programar, a través del análisis de los elementos básicos de la red. En otras palabras, es necesario identificar los procesos que intervienen en cada uno de los elementos mostrados en la Figura 1.7, obteniéndose el esquema mostrado en la Figura 1.8.

El esquema de la Figura 1.8, es pues, el esquema fundamental a partir del cual se trabajarán los algoritmos para la programación de la red neuronal. La forma recursiva del proceso global de entrenamiento de la red queda representada en la Figura 3.9.

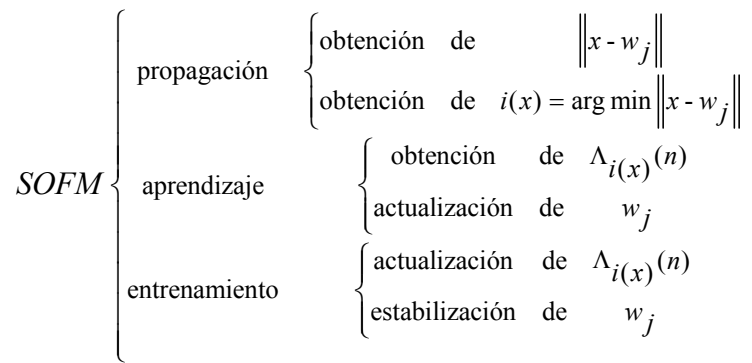


Figura 1.8 Esquema de los procesos del algoritmo SOM.

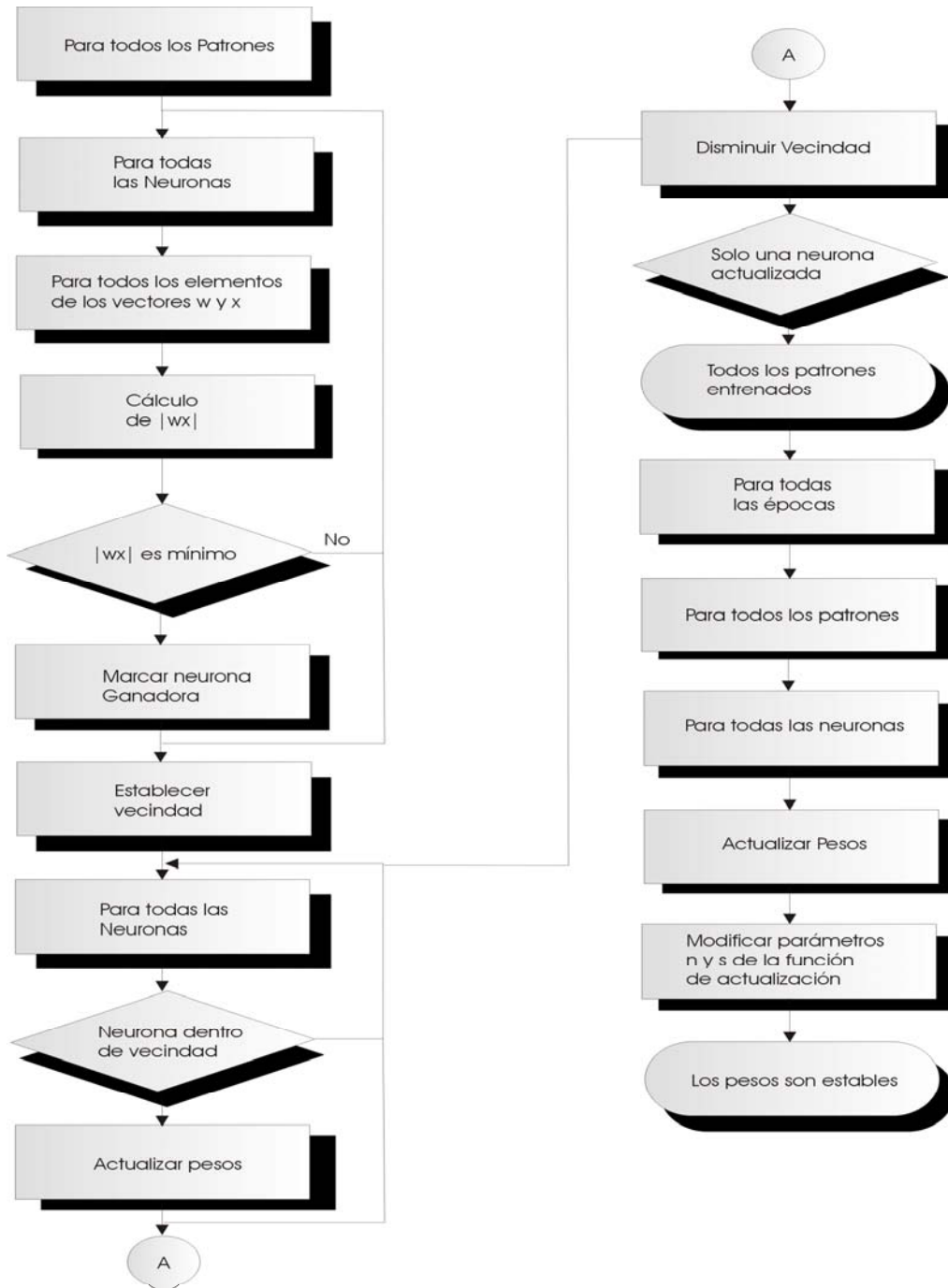


Figura 1.9 Proceso recursivo de aprendizaje de la red neuronal.

1.5.2 Propagación de la señal.

Se desarrollan entonces los algoritmos de programación con base en el esquema de la Figura 1.8 para la red **SOM**. Primero se describen los algoritmos de propagación de la señal. Estos algoritmos describen la forma en la que los patrones de entrada son distribuidos e influenciados por los pesos a las neuronas de salida. Estos algoritmos incluyen la manera en que con base a la función discriminante (norma euclideana) se determina la neurona ganadora a un patrón de entrada dado.

El primer algoritmo corresponde a la obtención de la norma dado un patrón de entrada y el vector de pesos asociado a la neurona. Este algoritmo debe repetirse tantas veces como neuronas en la red neuronal existan para cada patrón.³

Inicio

Obtención del patrón de entrada; $x(n | j)$

Obtención del vector de pesos asociado a la j -ésima neurona; w_j

Longitud del vector $x(n | j)$; $d = \dim x(n | j)$

Para todos los elementos del vector $x(n | j)$; desde $k = 1$ hasta d

$$\text{Calcular } \textit{distancia} = \|x - w_j\|$$

Terminar

Una vez obtenida la norma, el siguiente paso consiste en evaluar la norma para determinar la neurona ganadora. La evaluación es una simple comparación de las normas de cada neurona respecto al patrón que se entrena conservando aquella que sea mínima, y marcando a la neurona que la posee. El algoritmo de evaluación se presenta a continuación.⁴

Inicio

Para la primer neurona; $j = 1$

Determinar la dimensión de neuronas de salida (renglones \times columnas ; $numr \times numc$)

³ Ver Apéndice B, programa *distancia.m*

⁴ Ver Apéndice B, programa *minimo.m*

Para todos los renglones de la matriz de neuronas de salida; desde $r = 1$ hasta numr

Asignar valor arbitrario muy grande; $\text{menor} = 10000$

Para todos los elementos del r -esimo renglón; desde k_1 hasta k_{numc}

Calcular $\text{distancia} = \|x - w_j\|$ con la rutina **distancia**

Si $\text{distancia} < \text{menor}$

Neurona ganadora del r -esimo renglón; $\text{ganadora} = S_{jk}$

Se actualiza el valor arbitrario; $\text{menor} = \text{distancia}$

Se cuenta una neurona más; $j = j + 1$

Terminar

Terminar

Terminar

1.5.3 Aprendizaje y Entrenamiento.

Los procesos de aprendizaje y entrenamiento involucran primero la actualización de los pesos alrededor de la neurona ganadora. Es necesario un algoritmo que determine el tamaño de dicha vecindad, para ello se definen vecindades cuadradas las cuales disminuirán conforme el proceso de entrenamiento de la red avanza. El algoritmo propuesto crea una matriz imagen de ceros del entramado de neuronas de salida. La matriz es marcada con unos para aquellas neuronas que estén dentro de la vecindad y cuyos pesos serán actualizados. El algoritmo es el siguiente,⁵

Inicio

Crear la matriz imagen de ceros del entramado de salida;

Definir la posición espacial de la neurona ganadora en el entramado;

Definir la dimensión de la vecindad (largo de un lado del cuadrado);

Crear un algoritmo que evite se marquen puntos fuera de la matriz imagen (cuando la neurona ganadora esta en los límites de la matriz);

Marcar con un uno las neuronas que estén dentro de la vecindad,

Entregar la nueva matriz imagen;

Fin

⁵ Ver Apéndice B, programa *pertenece.m*

Determinadas las neuronas que pertenecen a una vecindad, el siguiente paso es actualizar los vectores de pesos asociados a dichas neuronas mediante la función descrita. Y la posterior disminución de la vecindad hasta que solo los pesos de la neurona ganadora sean actualizados.

La función de actualización es una modificación de la ecuación **¡Error! No se encuentra el origen de la referencia.** En **¡Error! No se encuentra el origen de la referencia.** todos los vectores de pesos a actualizar (que están dentro de la vecindad) son influidos de la misma forma por la función $\Lambda_{i(x)}(n)$, que es la función de vecindad topológica, tomando un valor máximo. Esta función es modificada en el algoritmo para que asemeje a la función sombrero mexicano explicada en primer capítulo. La elección más conveniente es entonces una función de tipo gaussiana en la cual la amplitud de la vecindad topológica disminuye mientras se incrementa la distancia lateral respecto a la neurona ganadora. La función de actualización queda definida entonces como,

$$w_j(n+1) = w_j(n) + \eta(n)\pi_{ji(x)}(n)[x(n) - w_j(n)] \quad (3.13)$$

donde,

$$\pi_{ji} = \exp\left(-\frac{d_{ji}^2}{2\sigma^2}\right)$$

El algoritmo de actualización se muestra a continuación,⁶

Inicio

*Determinar la neurona ganadora mediante la rutina **minimo**;*

*Determinar las neuronas a actualizar mediante la rutina **pertenece**;*

Para todas las neuronas que están dentro de la vecindad

Actualizar los pesos mediante la función (3.13);

Actualizar la matriz de pesos W_{fin} ;

⁶ Ver Apéndice B, programa *actualiza.m*.

Terminar

Finalmente el proceso de entrenamiento involucra la repetición de los algoritmos anteriores para todos los patrones de entrenamiento y un proceso de estabilización de los pesos en el cual se repite el entrenamiento para todos los patrones donde cada ciclo de entrenamiento es denominado una época y en cada época los parámetros de la función de actualización son modificados de forma exponencial [6] como sigue,

$$\sigma(n) = \sigma_0 \exp\left(-\frac{n}{\tau_1}\right)$$
$$\eta(n) = \eta_0 \exp\left(-\frac{n}{\tau_2}\right)$$

Esta forma de hacer decrecer la vecindad topológica en función del tiempo n fue propuesta por Ritter (1992) con el fin de garantizar la convergencia del algoritmo **SOM**. En el algoritmo propuesto la modificación de los parámetros se hace después de que todos los patrones de entrada x_p tengan una neurona ganadora y el tiempo n corresponde a cada época del periodo de convergencia de los pesos. El algoritmo de entrenamiento es el siguiente,⁷

Inicio

Inicializar parámetros (η_0 , σ_0 , épocas, neuronas de salida, valor inicial de vecindad);

Para todos los patrones de entrenamiento x_p

Mientras el numero de neuronas actualizadas sea mayor o igual a uno

*Correr la rutina **actualiza**;*

Disminuir la vecindad;

Terminar

Para todas las épocas

Para todos los patrones x_p

*Correr rutina **actualiza**;*

Terminar

Modificar parámetros η y σ ;

⁷ Ver Apéndice B, programa *entrenamiento.m*

Terminar

1.6 Conclusiones

El desarrollo experimental mostrado permite vincular los conocimientos teóricos descritos en el primer capítulo a la realización práctica del proyecto de monitoreo.

Definidos en el capítulo 2 los alcances del proyecto, la realización experimental del mismo hasta antes de su evaluación global involucra algunos pasos básicos:

- a) El conocimiento del sistema a monitorear para definir la vía de modelado que mejor convenga de acuerdo a la información que de éste se pueda obtener.
- b) La obtención el modelo mediante métodos de identificación el cual debe responder a las necesidades del proyecto.
- c) El desarrollo experimental de los algoritmos teóricos de los elementos involucrados en el modelo de monitoreo.

Este desarrollo permite tener los elementos necesarios para evaluar el comportamiento del modelo. Primero de forma analítica, lo que permite aislar los errores en el procesamiento de la información por parte de cada elemento y después de forma sintética para observar la evolución interdependiente de dichos elementos.

El desarrollo mostrado a lo largo del capítulo y los resultados a los que se llegaron, así como los algoritmos de programación propuestos son la base para la implantación del modelo de diagnóstico de fallas, y los resultados de su evaluación se describen en el siguiente capítulo.

1.7 Referencias

- [1] Gourishankar, V.: *Conversión de Energía Electromecánica*. International Textbook, 1969.
- [2] Patton, R.: *Issues of Fault Diagnosis for Dynamic Systems*. Springer, 2000.
- [3] Ljung, L.: *System Identification*. Cambridge Press, 1985.
- [4] Ollero, A.: *Control por Computadora*. Alfaomega Marcombo, 1991.
- [5] Freeman, J. y Skapura, D.: *Neural Networks: Algorithms, Applications and Programming Techniques*. Addison Wesley, 1992.
- [6] Haykin: *Neural Networks a Comprehensive Foundation.*, 1994

1 RESULTADOS

1.1 Introducción

El objetivo de este capítulo es presentar las valoraciones de los cálculos obtenidos en el capítulo 3 con el fin de determinar si los elementos cumplen adecuadamente las funciones que de ellos se requieren y evaluar el modelo de diagnóstico de fallas propuesto. Estos resultados se discuten de forma simple auxiliados por material gráfico, el cual ofrece la ventaja de mejorar la comprensión de los fenómenos discutidos verbalmente.

Es importante mencionar que los resultados se obtuvieron a partir de la simulación del sistema mecánico modelado. El sistema mecánico, denominado planta, esta representado por el modelo ARMAX. La simulación es desarrolla en el ambiente gráfico de SIMULINK de MATLABTM.

Se inicia la discusión en la sección 4.2, analizando el comportamiento del modelo analítico de la planta. Posteriormente en la misma sección, se discute el comportamiento del observador de Kalman acoplado a la planta. En la sección 4.3 se evalúan los resultados del entrenamiento de la red **SOM** para diversas entradas tipo (escalones y senos). La sección concluye con los resultados de entrenar a la red a algunos tipos de falla que se pudieran presentar en el sistema como lo son truncamientos en amplitud de la señal de salida. Finalmente en la sección 4.4 se conforma todo el sistema de diagnóstico de fallas (planta, observador y red neuronal) y se discuten los resultados obtenidos.

1.2 Valoración del Modelo Analítico del Sistema y el Filtro de Kalman

Para la evaluación del modelo **¡Error! No se encuentra el origen de la referencia.**, que es la representación **ARMAX** del sistema, se desarrolla el modelo gráfico presentado en la Figura 1.1.

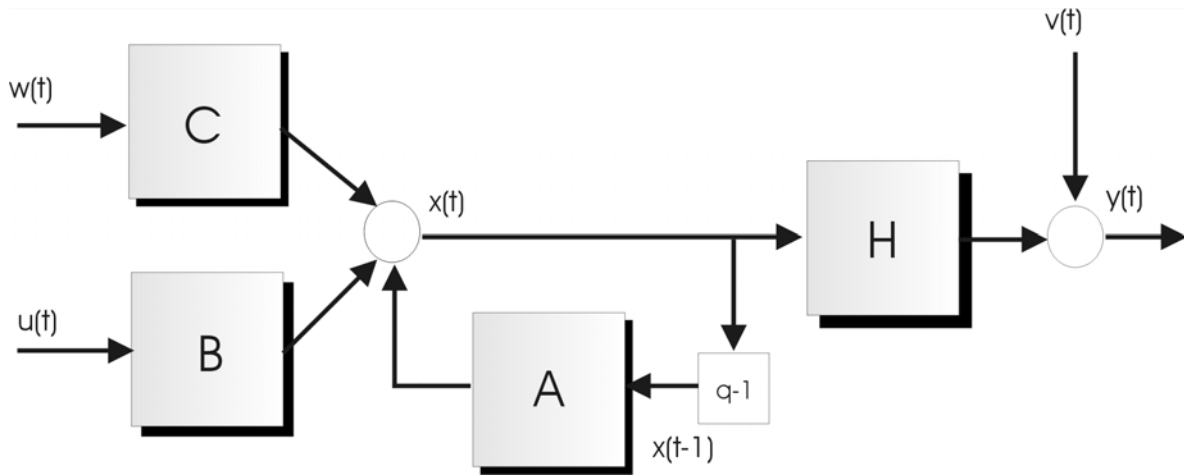
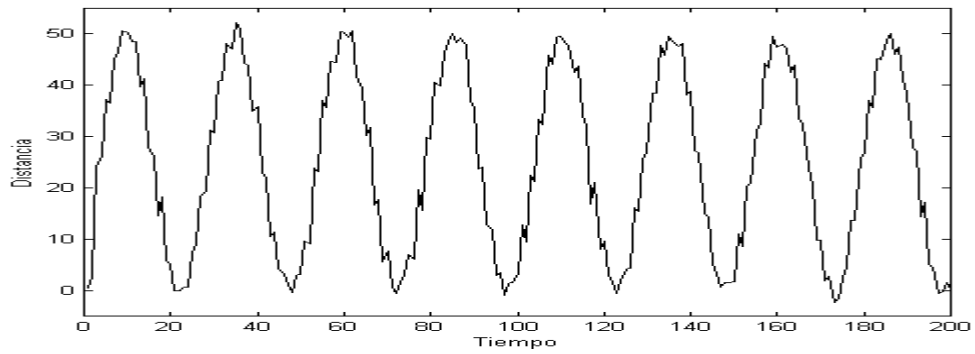
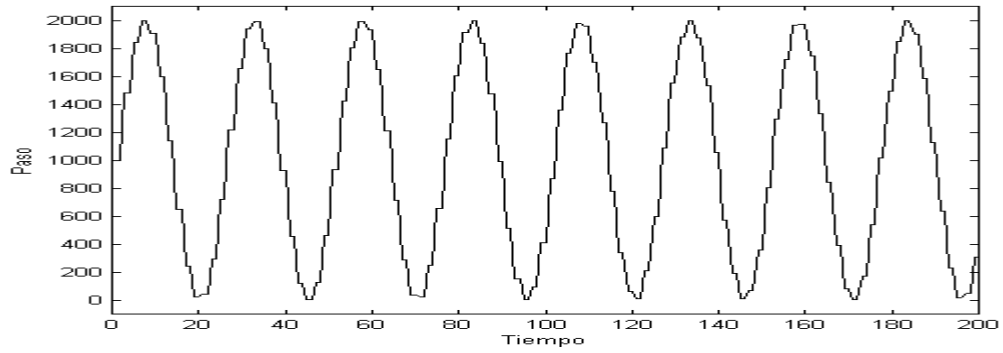


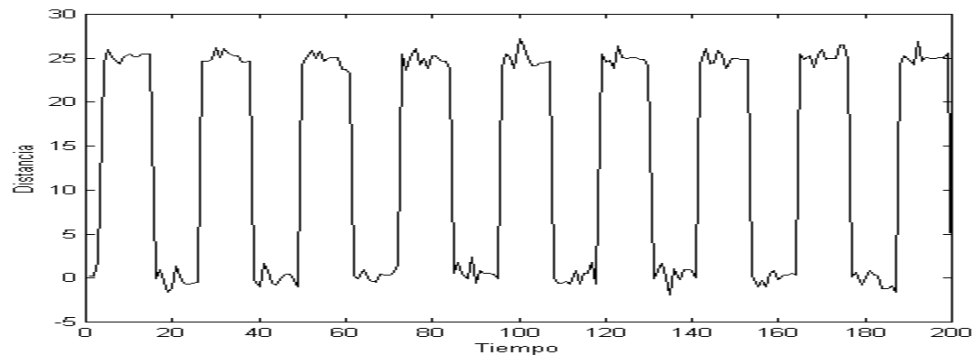
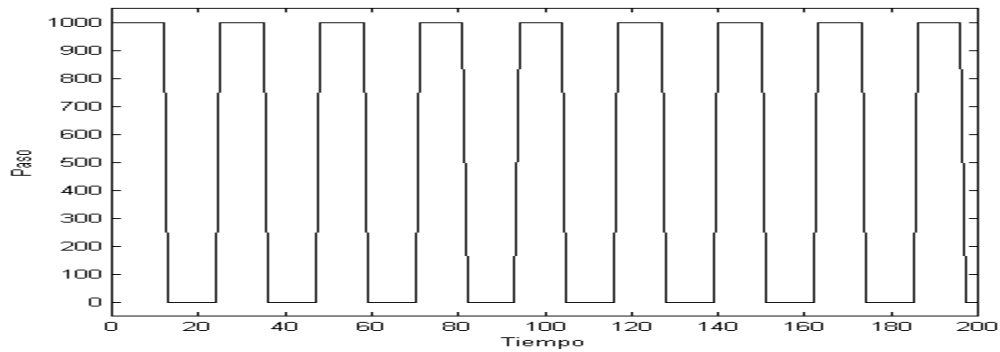
Figura 1.1 Esquemático del modelo ARMAX.

Este modelo es una representación típica de un sistema cuyo comportamiento es influenciado por ruido tanto a la entrada como a la salida. En la simulación el ruido ha sido simplificado a ruido blanco con características de potencia diferentes.

Inicialmente el modelo es excitado por entrada periódicas (senos y trenes de pulso) evaluándose el comportamiento de éste en 1000 unidades de tiempo. En la Gráfica 1.1 se presentan los pares de gráficas correspondientes a la entrada y la salida del sistema (a) para una excitación senoidal y (b) para una entrada de tren de pulsos.



(a) Entrada y salida senoidal.



(b) Entrada y salida tren de pulsos.

Gráfica 1.1 Entrada y salida del sistema.

Se observa en la Gráfica 1.1 que la salida obtenida mediante el modelo analítico sigue adecuadamente a la señal de entrada. Las variaciones que se perciben en la Gráfica 1.1 (a) y (b) para las salidas del sistema son consecuencia en gran parte al ruido aditivo a la salida del modelo.

Al incorporar el Filtro de Kalman al modelo **ARMAX** del sistema evaluado, se considera la descripción del sistema mostrado en la **¡Error! No se encuentra el origen de la referencia.** del Apéndice A. El modelo del sistema y el observador se muestran en la Figura 1.2. Este modelo es excitado con una señal de tipo senoidal obteniéndose los resultados mostrados en la Gráfica 1.2.

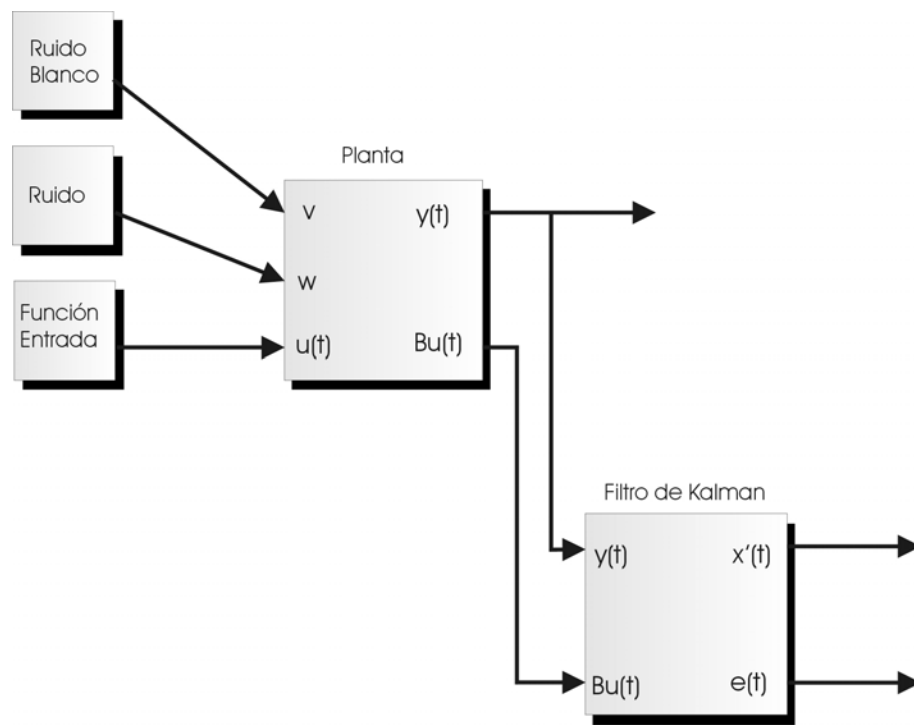
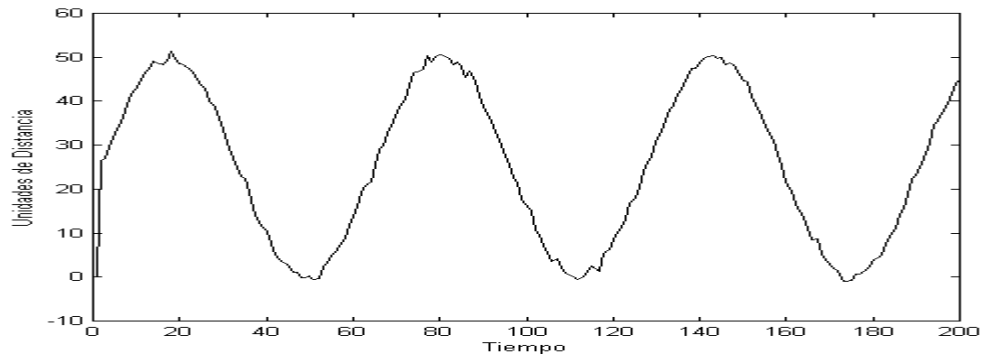


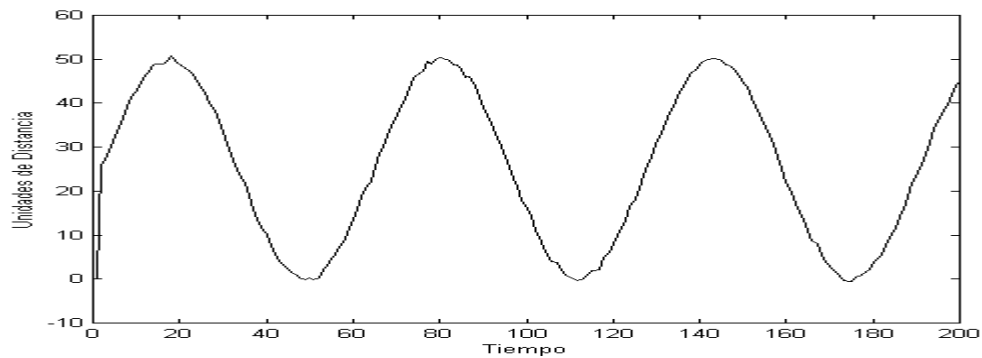
Figura 1.2 Sistema y observador.

Los resultados obtenidos en la Gráfica 1.2 (b) y (c) corresponden a la estimación de $\hat{x}(t)$ y el error $e(t)$ respectivamente, mientras que la Gráfica 1.2 (a) corresponde a la salida $y(t)$ del sistema. Se observa como el filtro de Kalman en su estimación $\hat{x}(t)$ de la salida $y(t)$ atenúa las fluctuaciones de la señal consecuencia de la influencia del ruido blanco $v(t)$. Este resultado es importante ya que de esta manera se está dando robustez al sistema

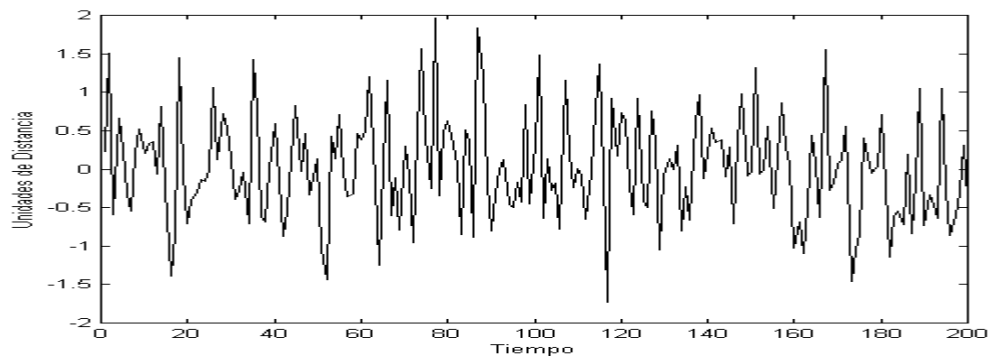
discriminando el ruido y los patrones de entrada a la red neuronal serán más estables haciéndola menos sensible al ruido y mas sensible a las fallas críticas en el proceso.



(a) Salida del sistema.



(b) Salida estimada del observador.

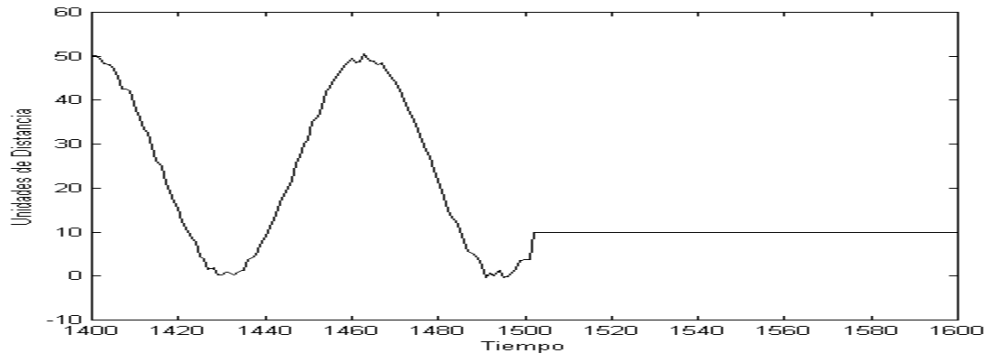


(c) Error entre la salida del sistema y su estimación.

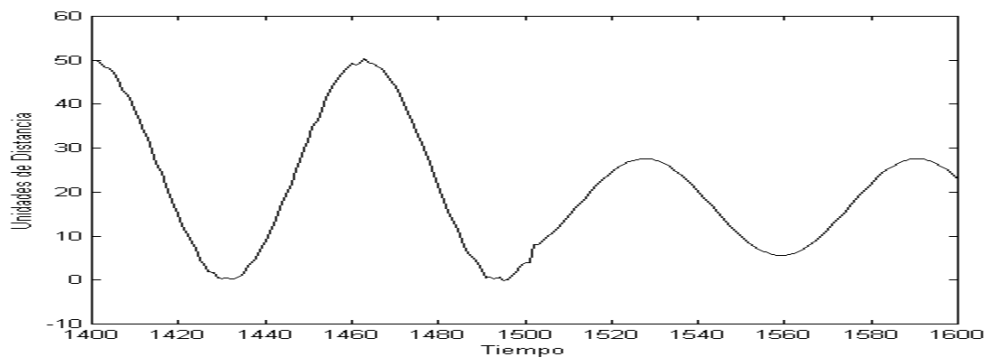
Gráfica 1.2 Resultados del sistema con la incorporación del observador.

Ahora, si bien una de las funciones del filtro de Kalman es la de incrementar la robustez en el sistema, su principal objetivo es el de funcionar como un mecanismo de monitoreo de

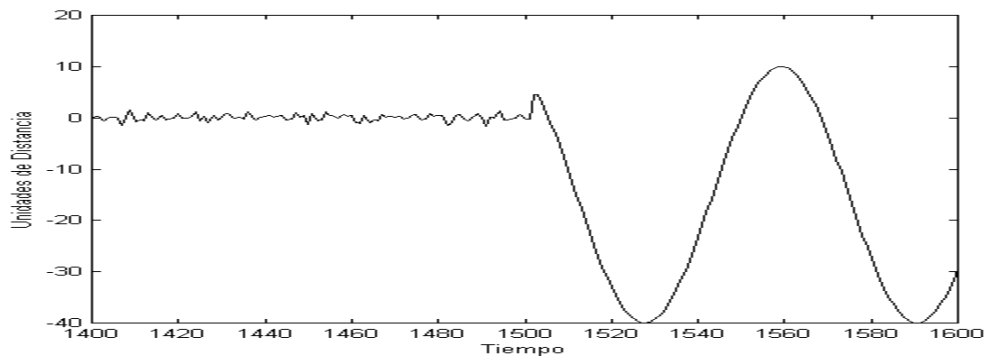
la planta para la detección de las fallas. Consecuencia de este segundo objetivo, se exponen los resultados de exponer al sistema a una de las fallas posibles: truncamiento en amplitud de la salida. La Gráfica 1.3 (a) muestra la salida del sistema con un truncamiento en el tiempo $t = 1500$ mientras que la Gráfica 1.3 (b) y (c) corresponden a la estimación $\hat{x}(t)$ y el error $e(t)$ respectivamente.



(a) Salida del sistema.



(b) Salida estimada.



(c) Error.

Gráfica 1.3 Resultados a una falla.

En la Gráfica 1.3 (b) se observa como el filtro de Kalman intenta seguir la señal de entrada $u(t)$ que es un seno pero es influido por el residuo $e(t)$ en el momento de la actualización. Mientras que el error $e(t)$ se incrementa considerablemente conforme el filtro intenta estimar la salida $\hat{x}(t)$ en función de la entrada $u(t)$. Se concluye entonces que efectivamente existe una modificación de los residuos en el momento de ocurrir la falla por lo que el objetivo de detección por parte del observador se cumple.

1.3 Resultados del Entrenamiento de la Red Neuronal

En el proceso de entrenamiento de la red, primero se analizan los resultados al reorganizar la matriz de pesos W_{fin} para entradas simples y conocidas como lo son: un seno y un tren de pulsos. La importancia de observar estos resultados radica en que los resultados esperados en la matriz W_{fin} pueden ser predecibles, como consecuencia, se pueden detectar fallas en el procesos de aprendizaje de la red que conducirían a una mala interpretación de los escenarios en el modelo de diagnóstico de fallas.

Estos resultados son predecibles en el sentido de que como se mencionó con anterioridad, el vector de pesos debe ser una buena aproximación del vector de entrada. Para las señales seno y tren de pulsos conocidas, la matriz de pesos W_{fin} deberá almacenar esta información como representaciones de las entradas, esto es, se deben ver reflejados los valores del seno y el tren de pulsos respectivamente en los valores de los pesos de la matriz.

Existen ciertos parámetros característicos de la red **SOM**, los cuales están directamente relacionados con la función de actualización del vector de pesos w_j . Estos son: el parámetro de retroalimentación negativa η que influye en la formación de la burbuja de activación y la desviación estándar σ que “físicamente” en el proceso de actualización de la red SOM representa la distancia efectiva de la vecindad.

A estos parámetros se les asignaron los siguientes valores para todas las evaluaciones realizadas a la red neuronal,

$$\eta = 2$$

$$\sigma = 30$$

Estos valores se asignaron de forma empírica a partir de los siguientes criterios:

- Como menciona la literatura valores muy pequeños de η inhiben la formación de la burbuja de actividad mientras que valores muy grandes la constriñen demasiado. El valor de 2 y valores ligeramente más grandes mostraron ser adecuados para la cantidad de vectores de entrenamiento x_p introducidos a la red.
- Un valor de σ muy grande propicia que se inicie el entrenamiento con una función de vecindad topológica similar a un pulso, el cual es la función más simple de vecindad topológica más simple observándose que la red neuronal converge de manera más rápida.

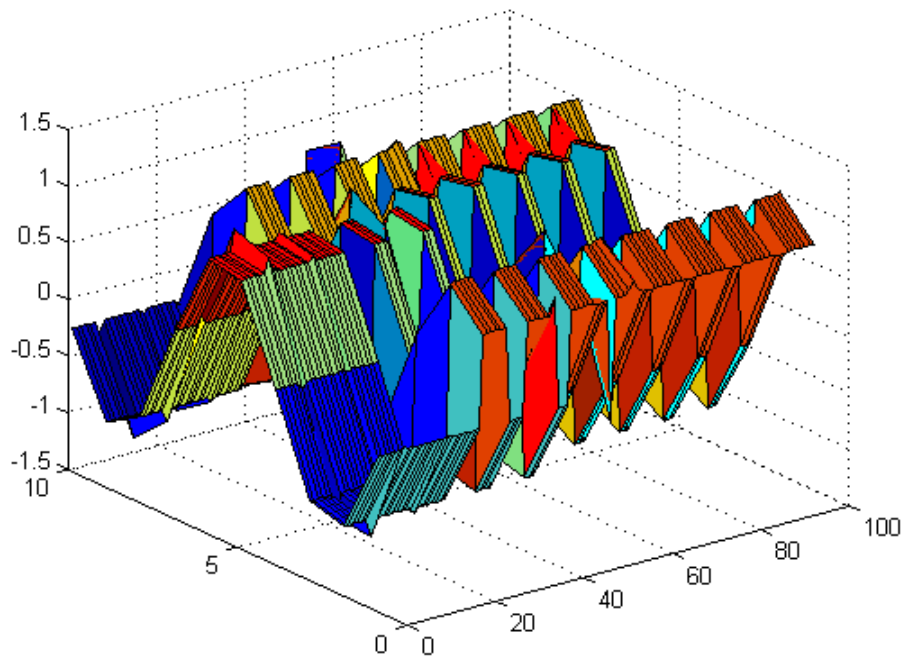
La red neuronal en todos los casos esta conformada por un entramado de neuronas de salida de 10×10 , es decir, 100 neuronas de salida y es entrenada para 1000 vectores de entrada x_p . Las épocas de entrenamiento (ciclos en los cuales se repite el proceso de actualización una vez que todos los patrones de entrada x_p fueron introducidos a la red) son 15 valor que se observo suficiente para que convergieran los pesos w . Las constantes de tiempo τ_1 y τ_2 de las funciones de detrimento de los parámetros η y σ tienen un valor de 5, el cual se determino en función a las épocas de entrenamiento.

La información de los parámetros y factores de entrenamiento de la red neuronal con sus respectivos valores se resume en la Tabla 4.1.

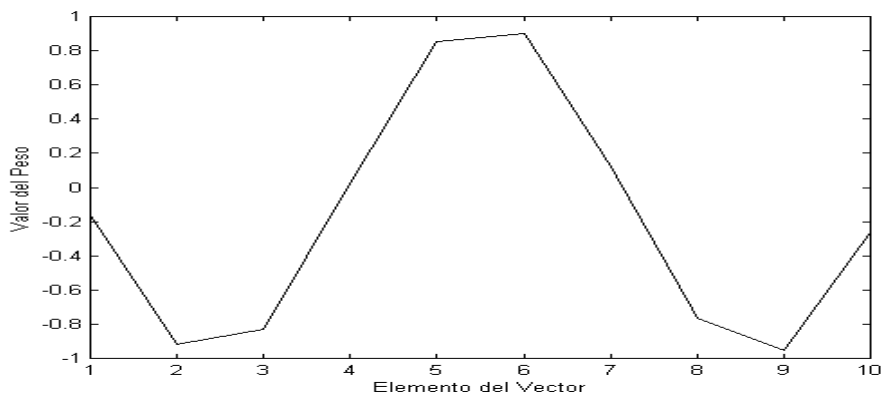
Parámetro o Factor	η	σ	épocas	τ_1	τ_2	Vectores x_p entrenados
Valor	2	30	15	5	5	1000

Tabla 1.1 Valores de entrenamiento.

Para la función seno, se genera una matriz de la función donde cada vector representa una ventana de tiempo. La matriz es una matriz de 10 renglones por 1000 columnas, es decir, la red es entrenada para 1000 ventanas de tiempo obteniéndose los resultados mostrados en la Gráfica 1.4 y Gráfica 1.5.



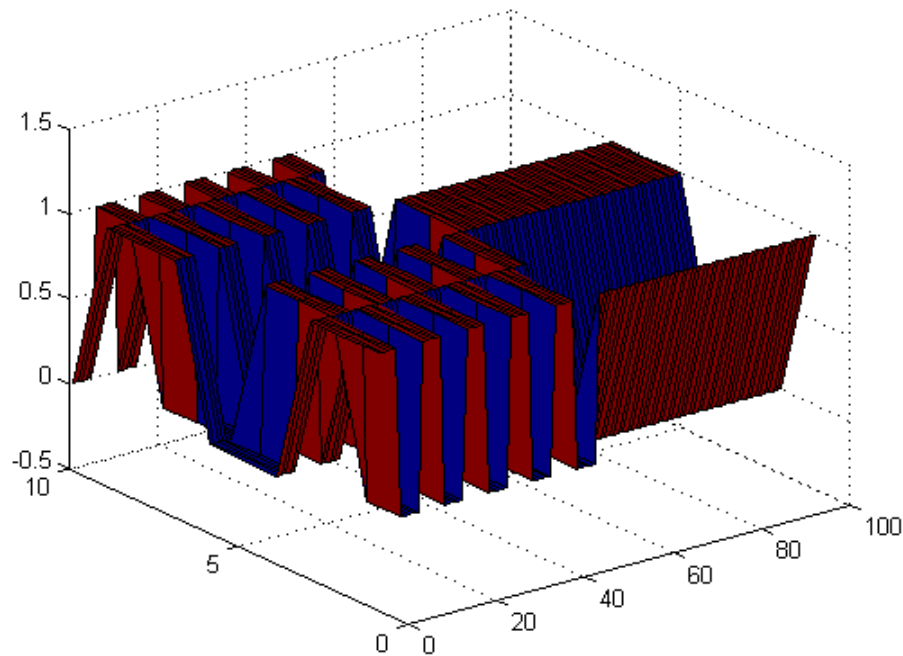
Gráfica 1.4 Matriz de pesos W_{fin} .



Gráfica 1.5 Vector de pesos asociado a la primera neurona.

La Gráfica 1.4 muestra los resultados de la matriz de pesos W_{fin} , donde se observa que efectivamente la información almacenada en los valores de los pesos de la matriz corresponde a una función de tipo senoidal. Un acercamiento a la información contenida en la matriz de pesos correspondiente al primer vector columna de la matriz, es decir, el vector de pesos asociado a la primer neurona se muestra en la Gráfica 1.5. En la figura se observa que efectivamente es un seno la información almacenada.

Los resultados mostrados en la Gráfica 1.6 corresponden al entrenamiento de la red neuronal a una entrada tren de pulsos. Estos resultados son similares a los obtenidos durante el aprendizaje y entrenamiento de la función seno.



Gráfica 1.6 Ordenamiento de la matriz W_{fin} para entrada tren de pulsos.

La Gráfica 1.6 muestra la distribución de los vectores de pesos asociados a las neuronas de salida de la red. El eje x representa el número de elementos de cada vector, el eje y el número de la neurona a la que esta asociado el vector, y el eje z el valor del peso. Es importante mencionar, que aún cuando en las gráficas se muestren de manera continua, en

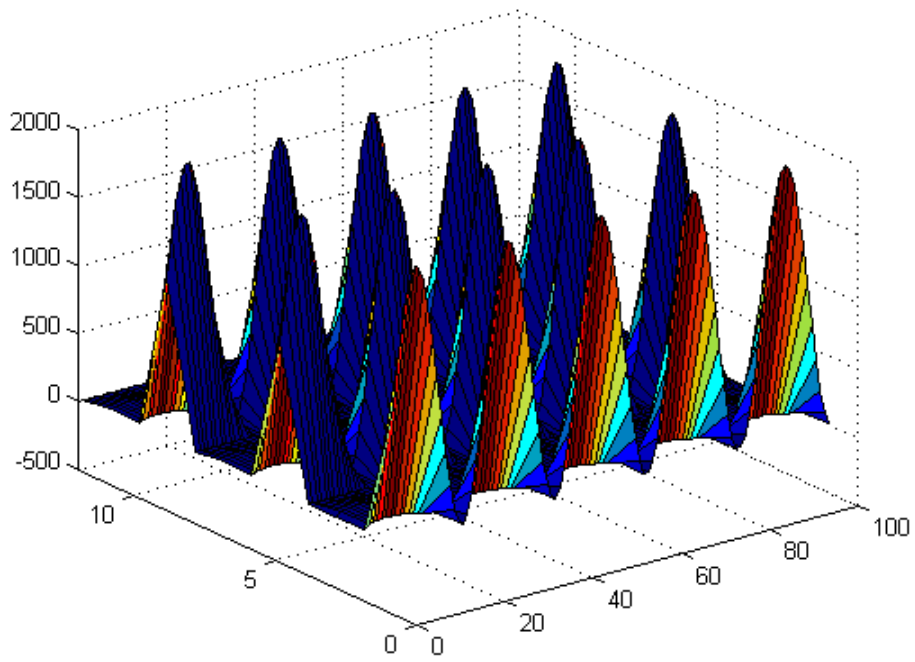
realidad los pesos son puntuales, es decir, las gráficas corresponden a valores discretos. El objeto de mostrar estos valores como distribuciones continuas, es el de proporcionar una visión más clara de los resultados que se obtienen y facilitar de esta manera su interpretación.

A partir de los resultados obtenidos para las señales senoidal y tren de pulso, se puede afirmar que el procedimiento de ordenamiento y asignación de valores de los pesos en la red neuronal es adecuado: la red aprende correctamente.

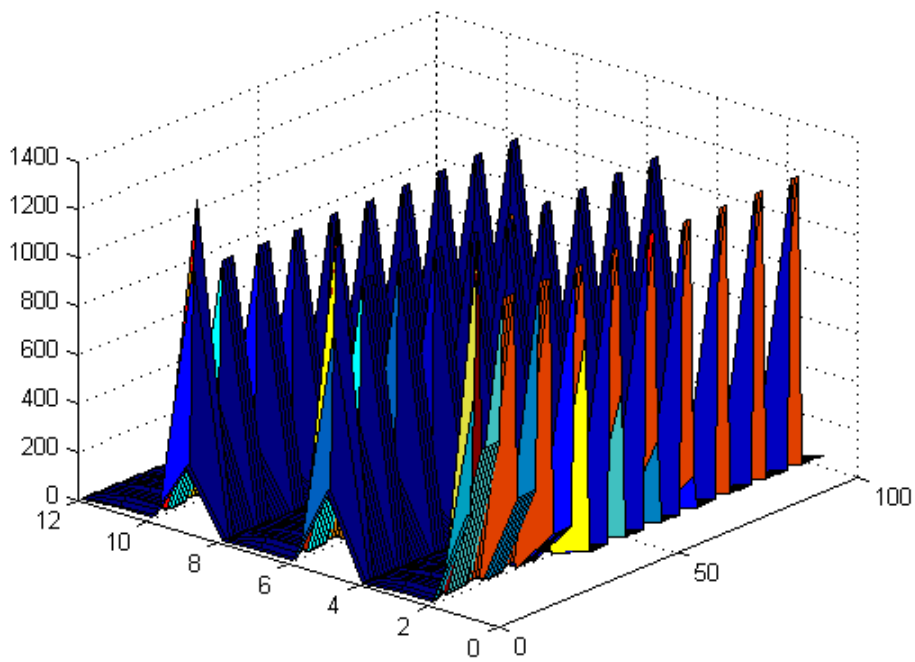
Con base en la conclusión anterior se procede a entrenar la red para valores de entrada con la forma apropiada y necesaria en el modelo de diagnóstico de fallas . El procedimiento lógico es el mismo que para los dos casos anteriores: la red es entrenada introduciendo vectores que representan un patrón y una ventana de tiempo. Cada elemento de la ventana de tiempo está compuesto, como se describe en el capítulo 3, por la entrada $u(t)$, la salida $y(t)$ y los residuos del observador $\hat{x}(t)$ (salida estimada), $e(t)$ (error). Estos elementos constituyen al vector patrón de entrada que tiene la forma,

$$x_p = (u(t)y(t)\hat{x}(t)e(t), u(t-1)y(t-1)\hat{x}(t-1)e(t-1), u(t-2)y(t-2)\hat{x}(t-2)e(t-2))$$

Los elementos del vector x_p son obtenidos a partir del modelo de planta con el observador de la Figura 1.2. La red es entrenada para 1000 ventanas de tiempo o vectores x_p . Un subconjunto de 100 vectores se muestra en la Gráfica 1.7 y la matriz *Wfin* resultante del entrenamiento se muestra en la Gráfica 1.8. Nuevamente se observa que la matriz de pesos refleja el comportamiento del sistema descrito por los vectores x_p .



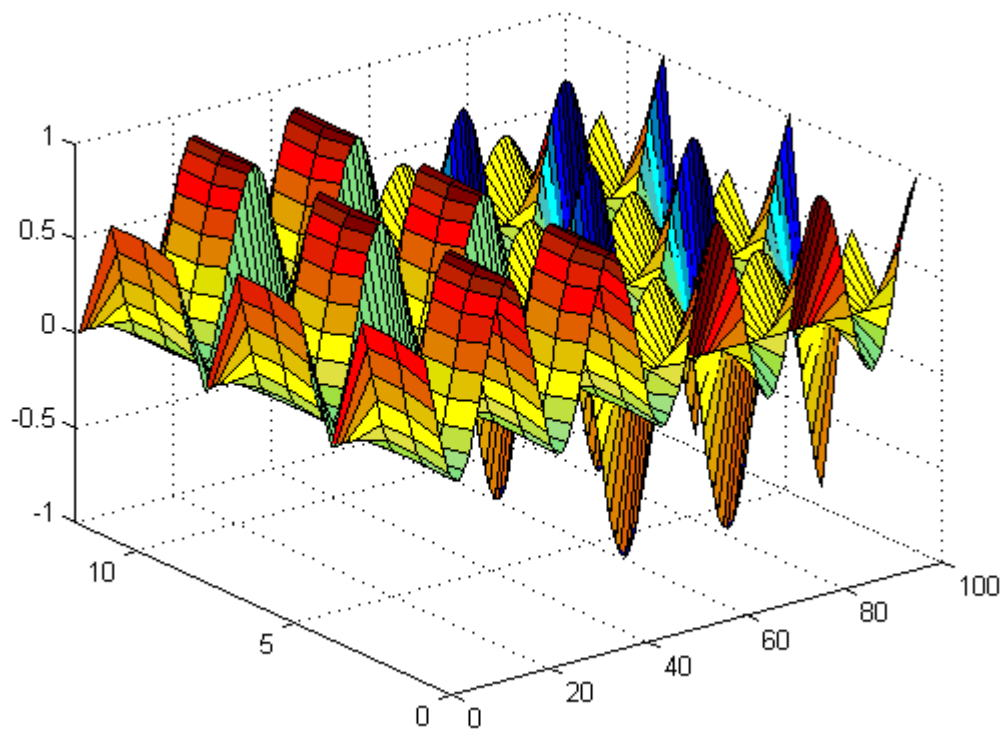
Gráfica 1.7 Muestra del escenario de entrenamiento.



Gráfica 1.8 Matriz W_{fin} . Resultado del entrenamiento.

Finalmente para concluir la evaluación del proceso de aprendizaje y entrenamiento de la red neuronal, se introduce al sistema una falla. Esta falla es un truncamiento en amplitud como el mostrado en la Gráfica 1.3 para el filtro de Kalman cuyo truncamiento ocurre en el tiempo $t = 1500$.

Los valores son normalizados a la norma unitaria, con el fin de evitar que los valores de entrada $u(t)$ considerablemente mayores a $y(t)$, $\hat{x}(t)$ y $e(t)$ influyan excesivamente en el proceso ya que ahora las variaciones de los residuos son considerables y de importancia como se observa en la Gráfica 1.9. Esta gráfica corresponde al momento de ocurrir la falla, el valor de 50 tiene equivalencia con el tiempo 1500.

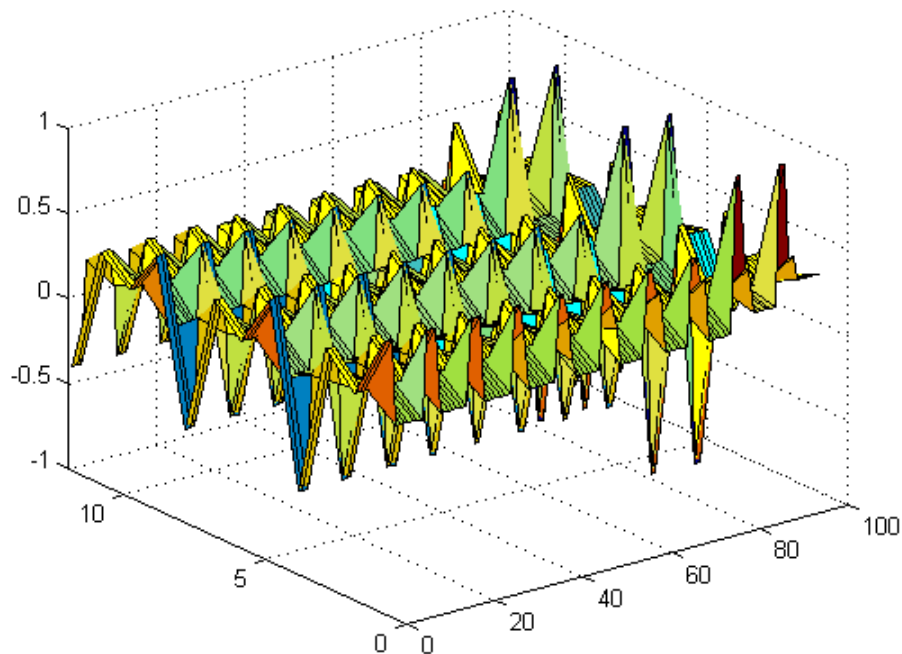


Gráfica 1.9 Patrón de entrenamiento con dos escenarios distintos.

Los resultados del entrenamiento se muestran en la Gráfica 1.10. Se observan claramente los dos escenarios: el primer escenario del vector de pesos asociado a la primer

neurona al vector asociado a la neurona 60 y el segundo del vector asociado a la neurona 60 a la 100. El primer escenario corresponde al comportamiento normal del sistema y el segundo escenario corresponde al comportamiento del sistema con la falla.

Con este resultado se finaliza la evaluación de la red neuronal y su procesos de entrenamiento considerándose satisfactorio y adecuado a su función: ser la función clasificadora de escenarios, es decir, proporcionar información sobre el tipo de falla.



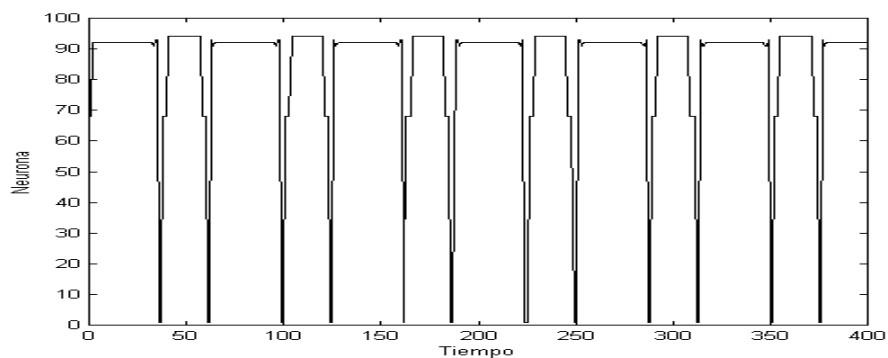
Gráfica 1.10 Resultado del entrenamiento. Matriz W_{fin} .

1.4 Comportamiento del Sistema de Diagnóstico de Fallas

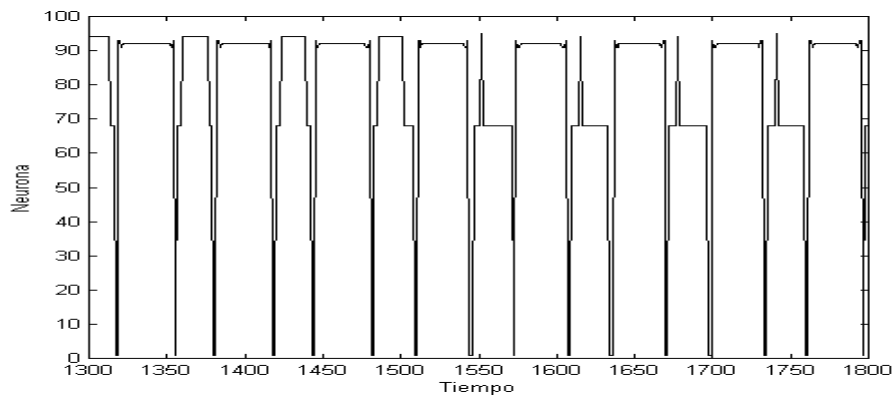
El último paso de la evaluación, consiste en integrar todos los elementos (planta, observador y red neuronal) en el esquema de diagnóstico de fallas descrito en la Figura 2.6 y observar el comportamiento de la sistema y el resultado de la red neuronal como función evaluadora de escenarios. El resultado esperado es la obtención de patrones definidos y estables para cada uno de los escenarios entrenados, sean estos fallas o no.

Primero se evalúa el comportamiento del sistema en condiciones de operación normales. La entrada $u(t)$ del sistema es de tipo senoidal y a sido previamente entrenada. El resultado se observa en la Gráfica 1.11 donde se aprecia visiblemente un patrón con periodo igual al de la señal de entrada por lo que el comportamiento es el deseado.

El siguiente paso consiste en evaluar este comportamiento sometiendo al sistema a una falla. La Gráfica 1.12 corresponde a la salida del red neuronal en función de diagnóstico. El diagnóstico tiene como base la matriz de pesos W_{fin} representada en la Gráfica 1.10. La falla ocurre en el tiempo $t=1500$ y se observa como existe un cambio en el patrón de salida de la red. De esta forma resultan claramente identificables los dos escenarios, antes y después de la falla por lo que se concluye que el objetivo de diagnóstico del sistema proyectado se cumple.

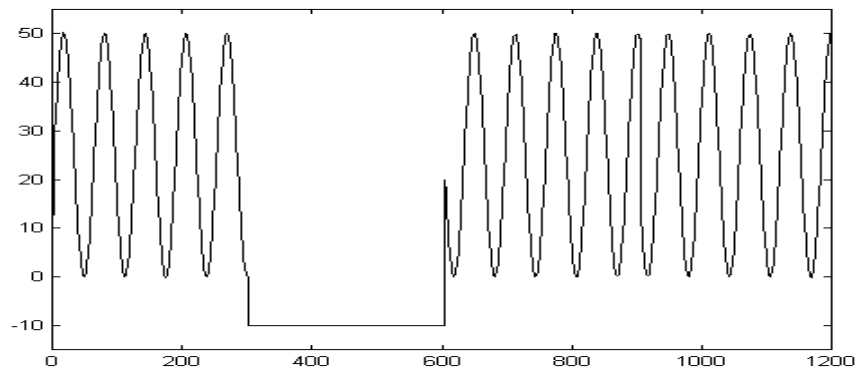


Gráfica 1.11 Patrón de comportamiento normal del sistema.



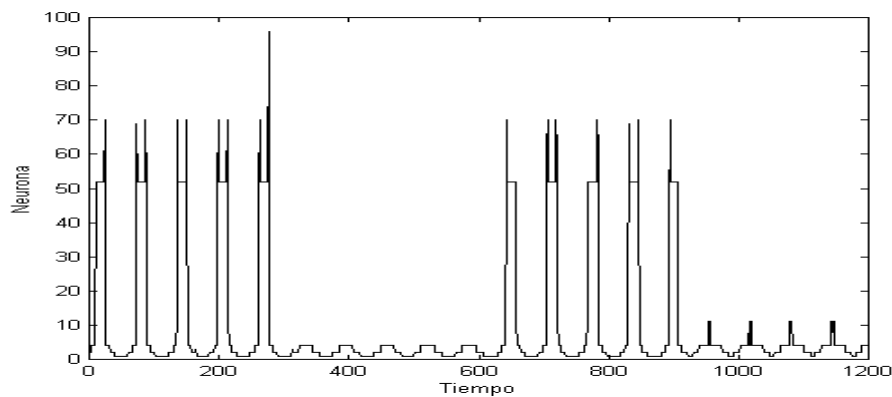
Gráfica 1.12 Patrones del escenario normal y la falla.

Finalmente la última evaluación corresponde a el diagnóstico de fallas diversas. El sistema es sometido a dos tipos de fallas: primero un truncamiento en amplitud y posteriormente un desfase en tiempo. La señal de entrada al sistema $u(t)$ es de tipo senoidal y la salida $y(t)$ afectada por las fallas se muestra en la Gráfica 1.13. La secuencia es la siguiente: comportamiento normal de $t = 0$ a $t = 300$, falla de amplitud de $t = 301$ a $t = 600$, corrección de la falla y comportamiento normal de $t = 601$ a $t = 900$ y finalmente una segunda falla de desfase de $t = 901$ a $t = 1200$.



Gráfica 1.13 Salida $y(t)$ consecuencia de dos fallas.

El resultado del diagnóstico efectuado por el sistema se muestra en la Gráfica 1.14 donde claramente se distinguen los tres escenarios propuestos en la secuencia descrita para la variable de salida $y(t)$. Este resultado concluye de manera satisfactoria la evaluación del sistema de diagnóstico de fallas.



Gráfica 1.14 Patrones de salida del diagnóstico.

1 CONCLUSIONES

A partir del análisis de los resultados efectuado en el capítulo anterior se consideran cumplidos satisfactoriamente los dos objetivos planteados al inicio del proyecto:

1. Relacionar los conceptos teóricos con el desarrollo práctico de un modelo de diagnóstico de fallas con base en una red neuronal.
2. Evaluar el modelo teórico propuesto en un modelo dinámico obtenido a partir de un sistema real.

El modelo **ARMAX** resultó un modelo apropiado de la planta con características adecuadas en el manejo de los ruidos. Si bien el sistema experimental resultó tener una muy buena relación señal a ruido, en el manejo de las simulaciones de la planta la potencia de éste se reforzó observándose de la misma forma un comportamiento adecuado del modelo.

El filtro de Kalman se utilizó por debajo de sus capacidades, la estimación de los estados proporcionada por el filtro se limitó únicamente a la estimación de la salida $y(t)$ y no se consideró el segundo elemento (consecuencia de ser el modelo de segundo orden) del vector de estados estimado $\hat{x}(t)$. La utilización de los dos elementos del vector de estado proporciona mayor información respecto al comportamiento del sistema y una mayor confiabilidad en los resultados proporcionados, pero el hecho de manejar un modelo de orden pequeño se consideró como razón suficiente para utilizar solamente la estimación del estado actual y disminuir la cantidad de información procesada por la plataforma digital utilizada.

Para el caso de la red neuronal, se observaron ciertas limitaciones en la manera en que los datos deben de ser introducidos en el algoritmo de aprendizaje para que el entrenamiento de la red tenga un ordenamiento de los pesos adecuado.

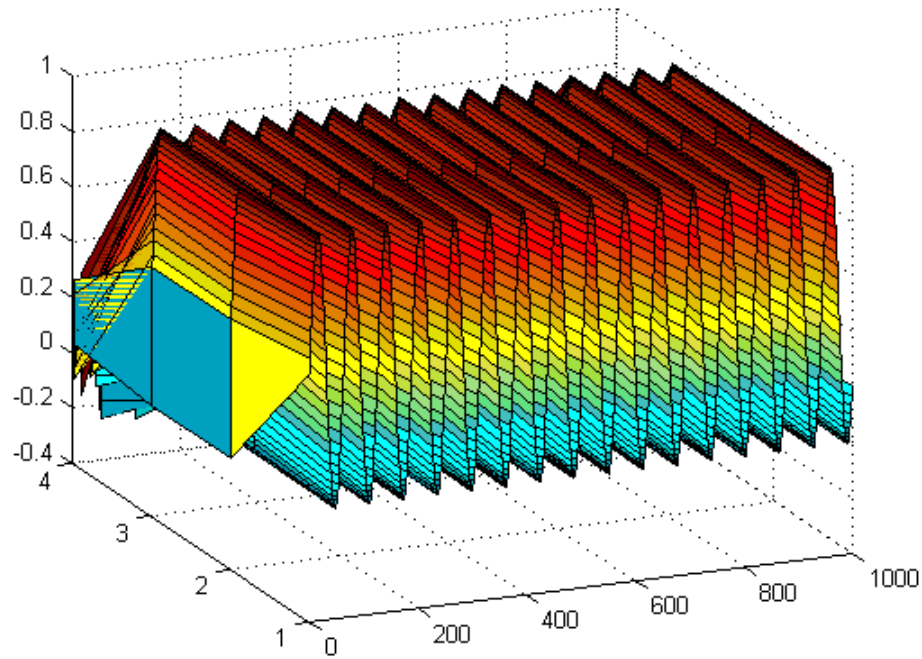
Una de estas limitaciones es referente a la “forma” de los datos o patrones de entrenamiento: los elementos de los vectores de entrada x_p y sus valores. Los patrones deben ser claramente diferenciables entre si y característicos, esto es, deben contener información particular y concreta para un correcto ordenamiento de los pesos. Como consecuencia del proceso de selección de la neurona ganadora, en el cual se considera como discriminante la distancia entre el vector de entrada x_p y el vector de pesos asociado a la j -ésima neurona w_j , en magnitud más no en dirección y sentido, se debe garantizar de cierta forma que los valores de los elementos de los vectores de entrenamiento no se compensen entre si dando como resultado la misma magnitud para vectores x_p diferentes. Esto originaría la concentración de la información en un número muy limitado de neuronas haciéndola de poca utilidad en el proceso.

Otra limitación, la cual es particular al proceso estudiado, tiene que ver con la cantidad y el tipo de información contenida en cada patrón. Los primeros resultados en el entrenamiento de la red se obtuvieron introduciendo un patrón a la red que correspondía a las características del sistema en un tiempo t . De esta forma los vectores x_p de entrenamiento correspondían a las características del sistema en los tiempos $t = 0, 1, \dots, 1000$. La hipótesis que sustentaba el entrenamiento de la red de esta forma era que al incorporar la red al sistema de diagnóstico de fallas, esta pudiera procesar la información y ofrecer un resultado de forma inmediata para cada muestreo del sistema. Los resultados obtenidos se muestran en la Gráfica 5.1 y en la Gráfica 5.2.

La Gráfica 5.2. corresponde al ordenamiento de la matriz W_{fin} después de entrenarla con una función seno como entrada $u(t)$ y un vector x_p de la forma,

$$x_p = [u(t), y(t), \hat{x}(t), e(t)]$$

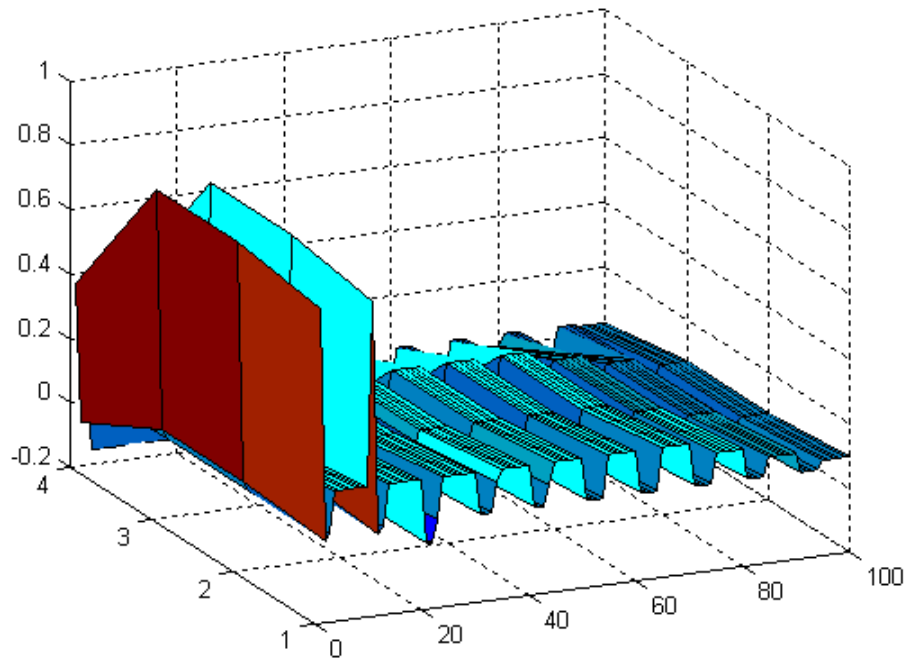
La muestra de 1000 vectores que se entrenaron se muestra en la Gráfica 1.1.



Gráfica 1.1 Patrones de entrenamiento.

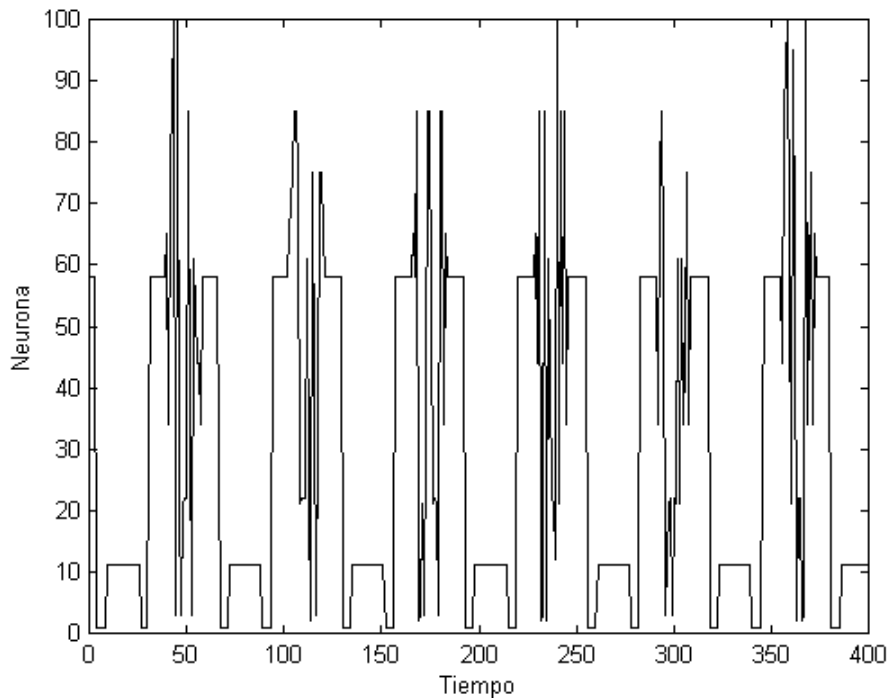
Resulta claro que el ordenamiento de los pesos en la matriz W_{fin} no es el adecuado, hay un cierto número de neuronas cuyos valores son actualizados adecuadamente, mientras que el resto de los vectores w_{ij} tienden a asemejarse al patrón entrenado pero nunca alcanzan los valores adecuados. Este comportamiento puede ser consecuencia de la primer limitante antes expuesta de la siguiente forma. Al inicio del entrenamiento el proceso de actualización que en su inicio abarca varias neuronas permite que una redistribución de más vectores de pesos. A medida que la vecindad se constriñe los vectores de pesos actualizados disminuyen lo cual es natural de acuerdo al procedimiento propuesto por Kohonen. El problema surge cuando al tenerse un nuevo patrón, si este es muy semejante a un valor ya entrenado (lo cual es muy probable para una función periódica) impedirá que una nueva neurona sea ganadora ya que su valor estará más “cerca” al de alguna neurona ya entrenada.

La consecuencia es la observada en la Gráfica 1.2: una distribución concentrada de los patrones entrenados lo cual resulta en un desperdicio de recurso, ya que un gran número de neuronas nunca serán activadas al introducirse la matriz W_{fin} en línea con el sistema.



Gráfica 1.2 Matriz W_{fin} entrenada.

Si el modelo de diagnóstico de fallas se prueba con valores entrenados de esta manera, el resultado que se obtiene es deficiente. La Gráfica 5.3 muestra el resultado de diagnóstico si se considera la matriz de pesos mostrada en la Gráfica 1.2. Si bien se alcanza distinguir cierto patrón en el resultado de la red neuronal, existe una gran variación principalmente en las crestas de la señal. Este comportamiento se puede definir como sensibilidad al ruido el cual afecta principalmente al elemento $e(t)$ del vector de entrada. El hecho de entrenar mediante ventanas de tiempo proporciona cierta robustez a las variaciones del residuo $e(t)$.



Gráfica 1.3 Resultado del diagnóstico.

El modelo de diagnóstico de fallas de la forma en la que fue propuesto resulta entonces en una forma conveniente de registrar escenarios de fallas posibles. Este modelo involucra necesariamente un conocimiento previo de la dinámica del sistema así como el conocimiento estadístico de su evolución para determinar que tipos de fallas o escenarios resulta conveniente utilizar.

La principal ventaja que ofrece esta en el procesamiento de la información. El hecho de considerar elementos de inteligencia artificial como las redes neuronales en el procesamiento de información en lugar de métodos analíticos basados en línea reduce el tiempo de procesamiento y de obtención de resultados además de proporcionar métodos más simples para acceder el conocimiento que sobre la planta o el proceso se tiene al modelo en desarrollo. Resulta claro que la forma en que la red fue entrenada para adquirir el conocimiento que de la planta se tenía resulto más sencilla que el intentar un modelo matemático para la descripción de los escenarios y la dinámica de estos.

Queda como campo de experimentación abierto la evaluación del modelo de diagnóstico de fallas con base en el principio heurístico propuesto en este trabajo para sistemas de mayor complejidad analítica: sistemas no lineales y variantes en el tiempo. Esta evaluación permitiría comprobar la efectividad tanto en el diseño de modelos como en el tiempo de procesamiento de los modelos heurísticos en el tratamiento de sistemas analíticamente complejos.

A APÉNDICE

A.1 Ecuaciones del Filtro de Kalman

Para llegar a establecer las ecuaciones del filtro se considera la descripción paramétrica del sistema en variables de estado (1) bajo la influencia de ruido $v(t)$. Descomponiendo el término de ruido $v(t)$ en contribuciones $w(t)$ y $v(t)$,

$$\begin{aligned}x(t+1) &= A(\theta)x(t) + B(\theta)u(t) + \Gamma w(t) \\y(t) &= C(\theta)x(t) + v(t)\end{aligned}\tag{1}$$

donde los ruidos del proceso $w(t)$ y de las mediciones $v(t)$ son secuencias aleatorias independientes con media cero y covarianzas,

$$\begin{aligned}Ew(t)w^T(t) &= \Gamma(\theta) \\Ev(t)v^T(t) &= Q(\theta) \\Ew(t)v^T(t) &= 0\end{aligned}\tag{2}$$

siendo Γ y Q matrices semidefinidas positivas como corresponde a las matrices de covarianza.

Para la obtención de la covarianza $P(t+1|t)$ tenemos que,

$$P(t+1|t) = E \{ [x(t+1) - \hat{x}(t+1|t)][x(t+1) - \hat{x}(t+1|t)]^T \} \quad (3)$$

sustituyendo en la ecuación anterior la dinámica de la ecuación (1),

$$P(t+1|t) = E \{ [Ax(t) + Bu(t) - A\hat{x}(t|t)][x^T(t)A^T + w^T(t)\Gamma^T - \hat{x}^T(t|t)A^T] \} \quad (4)$$

$$P(t+1|t) = AE \{ [x(t) - \hat{x}(t|t)][x(t) - \hat{x}(t|t)]^T \} A^T + \Gamma Q \Gamma^T$$

donde,

$$P(t|t) = E \{ [x(t) - \hat{x}(t|t)][x(t) - \hat{x}(t|t)]^T \} \quad (5)$$

por lo tanto,

$$P(t+1|t) = AP(t|t)A^T + \Gamma Q \Gamma^T \quad (6)$$

Las ecuaciones para el ciclo de propagación están dadas por,

$$\hat{x}(t+1|t) = A(t)\hat{x}(t|t) + B(t)u(t) \quad (7)$$

$$P(t+1|t) = A(t)P(t|t)A^T(t) + \Gamma Q \Gamma^T$$

y para el ciclo de actualización tenemos

$$\hat{x}(t+1|t+1) = \hat{x}(t+1|t) + L(t+1)[y(t+1) - C(t+1)\hat{x}(t+1|t)] \quad (8)$$

$$P(t+1|t+1) = P(t+1|t) - L(t+1)C(t+1)P(t+1|t)$$

siendo,

$$L(t+1) = P(t+1|t)C^T(t+1)[C(t+1)P(t+1|t)C^T(t+1) + R]^{-1} \quad (9)$$

la cual es obtenida como la solución positiva semidefinida de la ecuación estacionaria de Ricatti.[4]

El filtro de Kalman, definido por la ecuaciones (7) y (8) puede representarse como se muestra en la Figura A.1.[3]

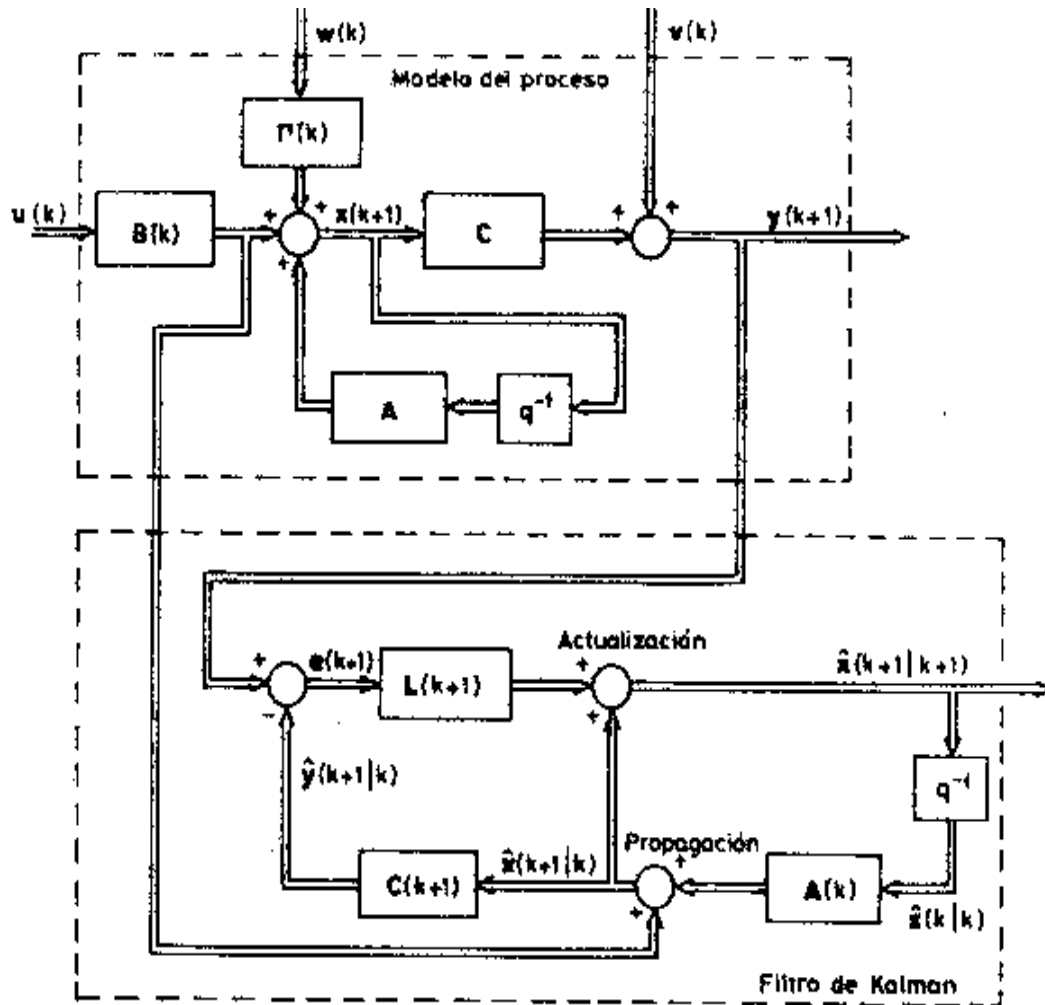


Figura A.1 Filtro de Kalman.