

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/245025192>

# Introducción a los Sistemas Inteligentes.

Book · December 2009

CITATIONS

8

READS

1,878

4 authors:



Ulises Castro Peñaloza

Universidad Politécnica de Baja California

7 PUBLICATIONS 12 CITATIONS

[SEE PROFILE](#)



Nun Pitalua-Diaz

Universidad de Sonora (Unison)

16 PUBLICATIONS 85 CITATIONS

[SEE PROFILE](#)



Jose A. Ruz-Hernandez

Universidad Autónoma del Carmen

63 PUBLICATIONS 210 CITATIONS

[SEE PROFILE](#)



Ruben Lagunas-Jimenez

Universidad Autónoma de Campeche

14 PUBLICATIONS 57 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:

Project

Modelos predictivos inteligentes en aplicaciones de ingeniería en energía [View project](#)







**Universidad Autónoma de Baja California**



## **Universidad Autónoma de Baja California**

**Dr. Gabriel Estrella Valenzuela  
Rector**

**Dr. Felipe Cuamea Velázquez  
Secretario general**

**M.C. Judith Isabel Luna Serrano  
Vicerrectora Campus Ensenada**

**Arq. Aarón Gerardo Bernal Rodríguez  
Vicerrector Campus Mexicali**

**Dr. Alfonso Vega López  
Vicerrector Campus Tijuana**

**L.C.C. Ricardo Moreno García  
Secretario de Rectoría e Imagen Institucional**

Introducción a los sistemas inteligentes / Nun Pitalúa Díaz...  
[et al.]. -- Mexicali, Baja California : Universidad Autónoma  
de Baja California; Sonora : Universidad de Sonora, 2009  
177 p. ; 21 cm. -- (Selección anual para el libro universitario  
2007-2008)

ISBN por la Unison 978-607-7782-17-9; ISBN por la UABC 978-607-7753-47-6  
1. Sistemas inteligentes de control. I. Pitalúa Díaz, Nun. II. Universidad Au-  
tónoma de Baja California. III. Universidad de Sonora.  
IV.t. V.s.

TJ217.5 I58 2009

FAM/ecr031105

**©D.R. 2008 Nun Pitalúa Díaz, Ulises Castro Peñaloza,  
José Antonio Ruz Hernández, Rubén Lagunas Jiménez**

Las características de esta publicación son propiedad de la  
Universidad Autónoma de Baja California.  
Departamento de Editorial Universitaria, Av. Reforma 1375.  
Col. Nueva Mexicali, Baja California, México.  
Teléfono: (686) 552-1056.  
Correo electrónico: editorial@info.rec.uabc.mx  
<http://www.uabc.mx>

ISBN por la Unison 978-607-7782-17-9; ISBN por la UABC 978-607-7753-47-6

Coordinación de editorial: Rosa María Espinoza Galindo  
Formación: Yohana Ruiz Dávila  
Edición: Bolfeta y asociados S.C.

Nun Pitalúa Díaz, Ulises Castro  
Peñaloza, José Antonio Ruz Hernández,  
Rubén Lagunas Jiménez

# Introducción a los sistemas inteligentes

**Selección Anual para el Libro Universitario 2007-2008**



## **AGRADECIMIENTOS**

A Dios por permitirnos colaborar en esta obra.

A Gisel y Luisa  
A Laura, Ana y Mario  
A Yela, Zaira, Marisol y Armando  
quienes dan sentido a todo.

A nuestros padres y hermanos por sus consejos y solidaridad.  
A nuestros mentores, colegas, estudiantes y amigos por su motivación.

A Rosa María Espinoza Galindo por su fe y asistencia en la realización de este libro

A la Universidad de Sonora, Universidad Autónoma de Baja California, Universidad Autónoma del Carmen y a la Universidad Autónoma de Campeche por su continuo apoyo en nuestras investigaciones y proyectos.



## **PRÓLOGO**

En la actualidad los sistemas inteligentes son motivo de impacto entre las diversas ramas de la ciencia debido a sus distintas aplicaciones, ya sea desde un punto de vista ambiental como clasificadores en los diversos rangos de desechos orgánicos e inorgánicos, hasta un punto de vista industrial en el que pueden ser empleados para el adecuado control del nivel de líquido en un sistema de tanques.

Lo anterior se debe a su cualidad de emular el pensamiento humano por medio de diversas técnicas, ya sea por medio de la experiencia como en los sistemas difusos, por medio del entrenamiento a partir de datos como en las redes neuronales o por medio de procesos evolutivos como los utilizados por los algoritmos genéticos.

Esta obra presenta la teoría fundamental de los sistemas inteligentes en sus diversas ramas: sistemas difusos, redes neuronales y algoritmos genéticos, además se incluye un capítulo referente a sistemas neurodifusos cuya combinación de las propiedades de los

sistemas difusos y redes neuronales les permiten una capacidad de auto aprendizaje y sus cálculos computacionales son rápidos por el uso de operaciones numéricas difusas.

Los autores agradecen a la Universidad Autónoma de Baja California y a la Universidad de Sonora por la oportunidad de difundir el contenido de este libro a la sociedad académica e industrial de México.

Por último se desea que el contenido de esta obra sirva de apoyo para todo aquel investigador o estudiante que se encuentre interesado o investigando sobre este tema.

## **INTRODUCCIÓN**

Cualquier tipo de proceso requiere un efectivo manejo de variables para controlar uno o varios sistemas, es por ello que se diseñan diversos tipos de técnicas para controlar dichas variables. Sin embargo, la versatilidad de éstas pueden limitarse cuando se aplican a otro sistema diferente, esto es, el desempeño del control puede no ser adecuado si los valores de operación del nuevo sistema son diferentes.

Los esquemas basados en técnicas de lógica difusa han sido utilizados para aproximar el comportamiento de diversos sistemas, ya que las técnicas de diseño obedecen principalmente a una serie de reglas generadas por humanos expertos para así obtener un comportamiento adecuado. Lo anterior da origen a un nuevo tipo de sistema el cual se denomina sistema difuso.

El sistema difuso puede ser analizado como un mapeo entre un vector de datos de entrada (error) y una salida escalar (salida de control). Este comportamiento entrada-salida se define por medio del conjunto de reglas, las cuales establecen cómo un valor de salida puede ser deducido desde un valor de entrada.

El capítulo I de este libro pretende introducir al lector en los antecedentes de los sistemas difusos y a los fundamentos de la lógica difusa. Así mismo, se analizan las partes que componen un sistema difuso. Finalmente se muestra un ejemplo del diseño y desempeño de un sistema de este tipo que se utiliza para fines de control (sistema de control difuso).

Otro de los sistemas que ha sido ampliamente usado en las últimas tres décadas han sido las Redes Neuronales Artificiales (RNA). Inspiradas en el funcionamiento del sistema nervioso de los seres humanos y de los vertebrados. En general estos algoritmos han permitido resolver problemas complejos en diversas áreas de la ciencia y la ingeniería de manera similar a la forma en que los seres humanos llevamos a cabo tareas cotidianas y que, para una computadora, son demasiado complejas.

En el capítulo II se hace una descripción de las RNA planteando, primeramente, los fundamentos biológicos que sustentan los modelos matemáticos. Posteriormente, se describe el modelo de una neurona artificial, denominado perceptrón simple, así como su algoritmo de entrenamiento. Más adelante, se describe la solución de un problema de clasificación y su construcción utilizando amplificadores operacionales. Seguido del perceptrón simple, se describe uno de los modelos de RNA más comúnmente utilizados, el perceptrón multicapa, finalizando con un ejemplo en el que se muestra la solución de un problema y su construcción mediante circuitos electrónicos.

El tercer capítulo, aborda el tema de los sistemas neurodifusos, los cuales se refieren a la combinación de la teoría de conjuntos difusos y de las redes neuronales para el aprovechamiento de las ventajas que ambas aportan. De forma general, la diferencia principal que existe entre los sistemas difusos y los neurodifusos

radica en que los primeros son construcciones expertas, mientras que los sistemas neurodifusos son entrenados a partir de datos experimentales. En su mayoría, los primeros métodos de diseño para el control difuso estaban basados en consideraciones heurísticas. Actualmente, el diseño se ha enfocado en el desarrollo de técnicas de control difuso basadas en el modelo. Con este modelo de comportamiento aproximado al proceso puede diseñarse un controlador.

A nivel computacional, los modelos neurodifusos pueden considerarse como estructuras matemáticas flexibles, similares a las redes neuronales. Éstos pueden aproximar una clase muy extensa de sistemas no lineales complejos, con un grado de exactitud deseado una vez que han sido sometidos a un proceso de aprendizaje.

Las ventajas que presentan los sistemas neurodifusos, como resultado de combinar la lógica difusa y las redes neuronales, son entre otras, las siguientes: manejan cualquier clase de información (numérica, lingüística o lógica); manejan información imperfecta, vaga, imprecisa o parcial; resuelven conflictos por colaboración y agregación; no necesitan un conocimiento previo de la relación de los datos del sistema; poseen capacidad de autoaprendizaje, autoorganización y autosintonización; poseen un proceso de toma de decisiones similar al operador humano y sus cálculos computacionales son rápidos por el uso de operaciones numéricas difusas. Con estas ventajas es evidente pensar que los sistemas neurodifusos pueden aprovecharse para modelar plantas o procesos con la finalidad de desarrollar y aplicar esquemas de control basados en el modelo.

En este capítulo se presentan los antecedentes de los sistemas neurodifusos. Así mismo, se explican brevemente las características de las aproximaciones con modelos del tipo caja blanca, negra y

gris. Se definen las características de un sistema tipo ANFIS y se plantean las fórmulas y ecuaciones a utilizar en la implementación de los algoritmos de aprendizaje: básico e híbrido. El primero se basa en el método del gradiente descendente para sintonizar los parámetros del ANFIS y el segundo, combina el primero con el estimador de mínimos cuadrados.

Además se estudian las propiedades del ANFIS como un aproximador universal. Se desarrolla una aplicación que consiste en obtener un modelo neurodifuso que aproxime una función no lineal bidimensional con base en el entrenamiento híbrido de este tipo de sistemas.

Por último, se describen los esquemas de control basados en modelos neurodifusos: control neurodifuso inverso, control con modelo neurodifuso interno y control predictivo basado en el modelo neurodifuso. Así mismo, se realiza el estudio de un caso de aplicación.

El capítulo final introduce al lector en el tema de los algoritmos genéticos, los cuales están inspirados en los procesos evolutivos que se presentan en la naturaleza, donde la idea central es la supervivencia de los individuos más aptos y la modificación constante de sus descendientes para adaptarse al entorno donde habitan. Estos principios son en parte emulados por los algoritmos genéticos cuando se usan para resolver problemas de optimización. El algoritmo genético es un proceso iterativo en el cual, a través de las generaciones, se conservan y reproducen los individuos mejor calificados con el objetivo de que, al final del proceso, se tenga una solución óptima del problema de optimización.

Este capítulo presenta los antecedentes de los algoritmos genéticos y se analizan los procesos de codificación, inicialización de la población, decodificación y evaluación. Con base en lo

anterior, se determinan los procesos iterativos de evaluación, reproducción, selección, cruzamiento y mutación. Así también, se muestra la estructura de un algoritmo genético simple.

Por último, se agrega un apéndice referente a los conceptos más utilizados en los sistemas de control.



# CAPÍTULO 1. SISTEMAS DIFUSOS

## 1.1 ANTECEDENTES

La lógica clásica sólo admite una preposición de dos valores: verdadero o falso (lógica binaria o bivaluada). Existen otras lógicas que admiten además un tercer valor posible (lógica trivaluada) e incluso múltiples valores de verdad (lógica multivaluada).

La lógica difusa puede usarse para explicar el mundo en el que vivimos puesto que sigue el comportamiento humano de razonar, sacando conclusiones a partir de hechos observados.

En décadas recientes ha surgido un crecimiento considerable en el número de aplicaciones basadas en la lógica difusa. Esto se debe, principalmente, a las membresías relevantes que presentan los sistemas difusos frente a otros tipos de sistemas en diversas aplicaciones. El objetivo de todo sistema que utilice una lógica difusa es describir los grados de los enunciados de salida (dados por una serie de reglas) en términos de los de entrada. Más aún, algunos sistemas son capaces de refinar los grados de veracidad de los enunciados de salida conforme se refinan los de entrada.

Por estas propiedades es que ciertos sistemas de lógica difusa aparecen una labor de aprendizaje, y son excelentes mecanismos de control de procesos. Lo anterior propicia que la lógica difusa se posicione dentro del área de la inteligencia artificial y ha dado origen a sistemas expertos de tipo difuso (Aceves López y Aguilar Martín, 2006), (Driankov, Hellendoorn, y Reinfrank, 1996), (Farinwata, Filev, y Langari, 2000), (Hopgood, 2001), (Lagunas Jiménez y Pitalúa Díaz, 2007), (Li, Chen, y Huang, 2001), (Matía, 1994), (Nguyen, Prasad, Walker, y Walker, 2003), (Passino y S., 1998), (Pitalúa Díaz, 2005), (Wang, A Course in Fuzzy Systems and Control, 1997), (Wang, Stable and Optimal Fuzzy Control of Linear Systems, 1998), (Ying, 2000), (Zadeh, 1965), (Zhang, Li, y Zhang, 2001) y (Zilouchian y Jamshidi, 2001).

## 1.2 TEORÍA DE CONJUNTOS DIFUSOS

El concepto de conjunto expresa una colección bien definida de elementos en la que es posible determinar para un objeto cualquiera, en un universo dado, si acaso éste pertenece o no al conjunto. La decisión, naturalmente, es sí pertenece, o bien, si no pertenece. El conjunto difuso surge de la necesidad del modelado de expresiones lingüísticas, caracterizadas por su naturaleza imprecisa e incierta (Driankov, Hellendoorn, y Reinfrank, 1996), (Matía, 1994), (Passino y S., 1998), (Pitalúa Díaz, 2005), (Wang, A Course in Fuzzy Systems and Control, 1997) y (Zadeh, 1965).

*Definición 1.* Sea  $U$  un conjunto universo constituido por la reunión de elementos denotados por  $x$ . Un conjunto difuso  $A$  en  $U$ , se define como el conjunto de pares ordenados:

$$\mu_A(x) = \{\mu_A(x) | x \in U\} \quad (1.1)$$

donde  $\mu_A(x)$  es la función membresía o pertenencia que se expresa como:

1. Una función de cualquier conjunto de parámetros  $p(x)$  del elemento  $x$ .

$$\mu_A(x) = \mu_A(p_1(x), p_2(x), \dots, p_n(x)) \quad (1.2)$$

2. Por enumeración de pares definidos sobre elementos discretos (Chen, 2006) del conjunto.

$$A = \sum_{i=1}^n \frac{\mu_A(x_i)}{x_i} \quad (1.3)$$

3. Cuando  $U$  es continua,  $A$  es comúnmente descrita como:

$$A = \int_x \frac{\mu_A(x)}{X} \quad (1.4)$$

donde:

- Σ no representa una suma, sino una agregación de pares.
- ∫ representa una enumeración incontable de los elementos  $x \in U$  con la función de membresía asociada  $\mu_A(x)$ .
- $\mu_A(x)$  no representa ningún cociente, sino un par (posibilidad/elemento).

La función de membresía establece una relación entre los elementos  $x$  y los grados de pertenencia en el intervalo [0, 1]. El grado de pertenencia designa la certeza con que cada elemento pertenece al conjunto.

*Definición 2.* El soporte de un conjunto difuso  $A$  en un universo discurso  $U$  es el conjunto que contiene todos los elementos de  $U$ , tales que:

$$S(A) = \{x \in U \mid \mu_A(x) > 0\} \quad (1.5)$$

*Definición 3.* El elemento único es un conjunto difuso cuyo soporte es un punto singular en  $U$ .

$$A = \frac{\mu}{y} \quad (1.6)$$

donde  $\mu$  es el grado de membresía del soporte  $y$  en el elemento único  $A$ .

### 1.3 FUNCIONES DE MEMBRESÍA TÍPICAS

*Función Triangular:* Definida por sus límites inferior  $a$  y superior  $b$ , y el valor modal  $m$ , tal que  $a < m < b$ . La siguiente figura muestra una función triangular típica:

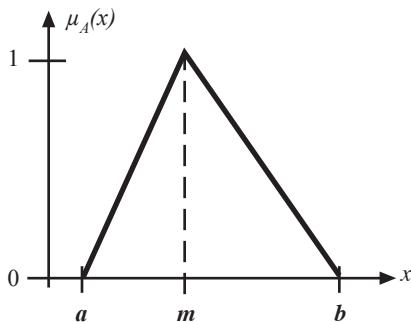


Figura 1.1. Función triangular

$$\mu_A(x) = \begin{cases} 0 & \text{si } x \leq a \\ (x-a)/(m-a) & \text{si } x \in (a,m] \\ (b-x)/(b-m) & \text{si } x \in (m,b] \\ 0 & \text{si } x \geq b \end{cases} \quad (1.7)$$

*Función  $\Gamma$  (gamma).* Definida por su límite inferior  $a$  y el valor  $k > 0$  (ver Figura 1.2).

$$\mu_A(x) = \begin{cases} 0 & \text{si } x \leq a \\ 1 - e^{-k(x-a)^2} & \text{si } x \in (a,m] \end{cases} \quad (1.8)$$

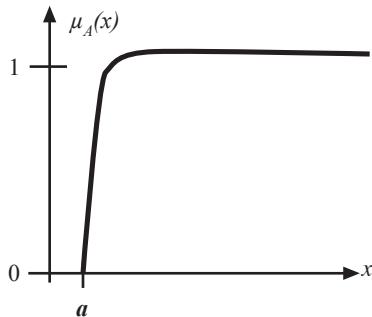


Figura 1.2. Función gamma

*Función Trapezoidal.* Definida por sus límites inferior  $a$  y superior  $d$ , y los límites de su soporte,  $b$  y  $c$ , inferior y superior respectivamente (ver Figura 1.3).

$$\mu_A(x) = \begin{cases} 0 & \text{si } x \leq a \text{ o } x \geq d \\ (x-a)/(b-a) & \text{si } x \in (a,b] \\ 1 & \text{si } x \in (b,c] \\ (d-x)/(d-c) & \text{si } x \geq b \end{cases} \quad (1.9)$$

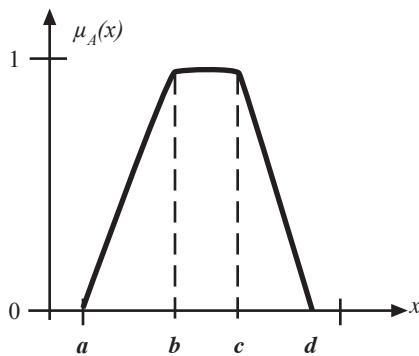


Figura 1.3. Función trapezoidal

*Función Gaussiana.* Definida por su valor medio  $m$  y el valor  $k > 0$  (ver figura 1.4).

$$\mu_A(x) = e^{-k(x-m)^2} \quad (1.10)$$

Es la típica campana de Gauss. Cuanto mayor es  $k$ , más estrecha es la campana.

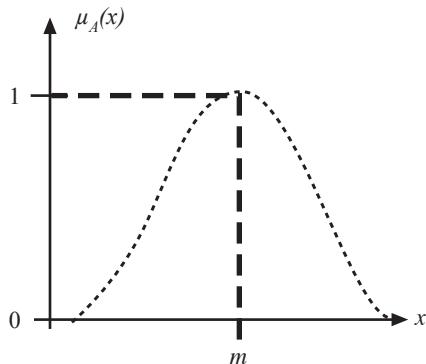


Figura 1.4. Función gaussiana

## 1.4 OPERACIONES ENTRE CONJUNTOS DIFUSOS

De la misma manera que en la teoría clásica de los conjuntos se definen las operaciones de unión, intersección y complemento, entre conjuntos difusos (Driankov, Hellendoorn, y Reinfrank, 1996), (Matía, 1994), (Passino y S., 1998), (Pitalúa Díaz, 2005) y (Wang, A Course in Fuzzy Systems and Control, 1997), las operaciones son definidas en términos de sus funciones de membresía como:

$$\text{Unión} \quad \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) \quad (1.11)$$

$$\text{Intersección} \quad \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad (1.12)$$

$$\text{Complemento} \quad \mu_{\neg A}(x) = 1 - \mu_A(x) \quad (1.13)$$

Zadeh estableció como definición de las leyes de membresía, las cotas máximas y mínimas a los conjuntos de intersección y unión (Zadeh, 1965).

El conjunto vacío  $\Phi$  tiene como función de membresía una constante igual a cero, y el conjunto completo una constante igual a uno.

*Unión difusa (norma-s):* La función  $s(\mu_A(x), \mu_B(x))$  es una norma s si satisface lo siguiente:

- $s(1,1)=1, s(0,a)=s(a,0)=a$
- $s(a,b)=s(b,a)$
- Si  $a \leq a'$  y  $b \geq b'$ , entonces  $s(a,b) \leq s(a',b')$
- $s(s(a,b),c)=s(a,s(b,c)),$  por ejemplo:  

$$s(\mu_A(x), \mu_B(x)) = \max(\mu_A(x), \mu_B(x))$$

$$s(\mu_A(x), \mu_B(x)) = \min(1, \mu_A(x), \mu_B(x))$$

*Intersección difusa (norma-t):* La función  $t(\mu_A(x), \mu_B(x))$  es una norma s si satisface lo siguiente:

- $t(1,1)=1, t(0,a)=t(a,0)=a$
- $t(a,b)=t(b,a)$
- Si  $a \leq a'$  y  $b \geq b'$ , entonces  $s(a,b) \leq s(a',b')$
- $t(t(a,b),c)=t(a,t(b,c)),$  por ejemplo:  

$$t(\mu_A(x), \mu_B(x)) = \min(\mu_A(x), \mu_B(x))$$

$$t(\mu_A(x), \mu_B(x)) = \mu_A(x) * \mu_B(x)$$

## 1.5 RELACIONES DIFUSAS

En la teoría clásica para relacionar dos conjuntos se efectua el producto cartesiano de sus elementos. Como siempre, la diferencia residirá en que, mientras en la teoría clásica los conjuntos están o no en relación, en una relación difusa cada par ordenado de los dos elementos puede gozar de una función de membresía que caracterice en qué grado dichos elementos están relacionados. Por tanto una relación difusa, es un conjunto difuso definido en el producto cartesiano de los conjuntos clásicos  $U_1 \times U_2 \times \dots, U_n$  (Driankov, Hellendoorn, y Reinfrank, 1996), (Matía, 1994), (Passino y S., 1998), (Pitalúa Díaz, 2005) y (Wang, A Course in Fuzzy Systems and Control, 1997).

La relación difusa se puede representar de manera general como:

$$Q = \{(x_1, x_2, \dots, x_n), \mu_Q(x_1, x_2, \dots, x_n)\} | (x_1, x_2, \dots, x_n) \in U_1 \times U_2 \times \dots, U_n \} \quad (1.14)$$

*Ejemplo 1.1.* La relación difusa  $R = x$  aproximadamente igual que  $y$ , también definida como  $R(X, Y)$ , donde los conjuntos difusos son:

$$\begin{aligned} X &= \{x_1, x_2\} \\ Y &= \{y_1, y_2\} \end{aligned}$$

y la estructura matricial es:

TABLA 1.1 RELACIÓN DIFUSA  $X$  APROXIMADAMENTE IGUAL QUE  $Y$

X/Y	Y <sub>1</sub>	y <sub>2</sub>	y <sub>3</sub>	y <sub>4</sub>
x <sub>1</sub>	1.0	0.5	0.1	0.0
x <sub>2</sub>	0.5	1.0	0.5	0.1
x <sub>3</sub>	0.1	0.5	1.0	0.5
x <sub>4</sub>	0.0	0.0	0.5	1.0

En la tabla anterior se expresa el grado en que los elementos de ambos conjuntos difusos  $X$  y  $Y$  están relacionados entre sí. Como se puede observar, el pico de la función de membresía se encuentra centrado en aquellas posiciones tales que los dos elementos son iguales, expresando así que los elementos se encuentran relacionados al máximo (función de membresía = 1) cuando son iguales.

## 1.6 OPERACIONES CON RELACIONES DIFUSAS

Se utilizan tres operaciones que son la composición, proyección y extensión cilíndrica (Driankov, Hellendoorn, y Reinfrank, 1996), (Matía, 1994), (Passino y S., 1998), (Pitalúa Díaz, 2005) y (Wang, A Course in Fuzzy Systems and Control, 1997). A continuación se define cada una de ellas.

### 1.6.1 COMPOSICIÓN

Si se definen dos relaciones difusas  $R(X,Y)$  y  $S(Y,Z)$  compartiendo el mismo espacio  $Y$ , se puede generar una nueva relación  $Q(X,Z)$  a partir de la composición de las relaciones  $R$  y  $S$ . La generación de la composición se puede realizar a partir de las funciones de membresía de cada una de las relaciones de la siguiente forma: si  $\mu_R(x,y)$  es la función de membresía asociada con  $R$  y  $\mu_S(y,z)$  es la asociada con  $S$ , entonces la composición  $R \otimes S$  tendrá la siguiente definición:

$$\mu_{R \otimes S}(x,z) = \sup_{\forall y \in Y} (\mu_R(x,y) * \mu_S(y,z)) \quad (1.15)$$

donde  $\sup$  es el operador máximo para universos discretos y “ $*$ ” puede ser una norma-s. Las más usadas son el mínimo y el producto algebraico. Esta combinación de operadores da origen a la composición *max-min* (máximo-mínimo) y *max-prod* (máximo producto) respectivamente. Debe notarse que la formulación anterior también se cumple en la composición de relaciones clásicas usando funciones de membresía categóricas (con los valores cero y uno únicamente).

$$\mu_{R \circ S}(x, z) = \max_{y \in Y} \min[\mu_R(x, y), \mu_S(y, z)] \quad (1.16)$$

$$\mu_{R \circ S}(x, z) = \max_{y \in Y} [\mu_R(x, y) \cdot \mu_S(y, z)] \quad (1.17)$$

### 1.6.2 PROYECCIÓN

*Definición 4.* Sea  $Q$  una relación en  $U_1 \times \dots \times U_n$  y  $\{i_1, \dots, i_k\}$  una subsecuencia de  $\{1, \dots, n\}$ , entonces la proyección de  $Q$  sobre  $U_{i_1} \times \dots \times U_{i_k}$  es una relación difusa  $Q_P$  en  $U_{i_1} \times \dots \times U_{i_k}$  con función de membresía:

$$\mu_{Q_P}(x_{i_1}, \dots, x_{i_k}) = \max_{x_{i_1} \in U_{i_1}, \dots, x_{j(n-k)} \in U_{j(n-k)}} \mu_Q(x_1, \dots, x_n) \quad (1.18)$$

Ejemplo 1.2. Obtenga la proyección de la relación difusa  $Q = x$  esta muy cerca  $z$  donde:

$$\begin{aligned} X &= \{x_1, x_2\} \\ Z &= \{z_1, z_2\} \end{aligned}$$

TABLA 1.2 RELACIÓN DIFUSA  $X$  ESTA MUY CERCA DE  $Z$ 

X/Z	$Z_1$	$Z_2$
$x_1$	0.6	1
$x_2$	0.4	0.8

Al tomar el elemento máximo correspondiente al conjunto que se este analizando, se genera la siguiente enumeración:

$$\begin{aligned} Q_{P(x)} &= \frac{1}{x_1} + \frac{0.8}{x_2} \\ Q_{p(z)} &= \frac{0.6}{z_1} + \frac{1}{z_2} \end{aligned} \quad (1.19)$$

### 1.6.3 EXTENSIÓN CILÍNDRICA

En el caso de que se quiera realizar la composición de dos conjuntos cuyas bases de parámetros sean diferentes, será necesario definir una base de parámetros comunes. A este proceso se le denomina extensión cilíndrica.

*Definición 1.5.* Sea  $Q_P$  una relación en  $U_1 \times \dots \times U_n$  y  $\{i_1, \dots, i_k\}$  una subsecuencia de  $\{1, \dots, n\}$ , entonces la extensión cilíndrica de  $Q_P$  a  $U_{i_1} \times \dots \times U_{i_k}$  es una relación difusa  $Q_{PE}$  en  $U_{i_1} \times \dots \times U_{i_k}$  con función de membresía:

$$\mu_{Q_{PE}}(x_1, \dots, x_n) = \mu_{Q_P}(x_{i_1}, \dots, x_{i_k}) \quad (1.20)$$

*Ejemplo 1.3.* Obtenga la extensión cilíndrica del ejemplo 1.2.

Al definir una base de parámetros comunes para ambos conjuntos difusos y, tomando en cuenta los elementos de las proyecciones del ejemplo anterior, resulta la siguiente extensión cilíndrica :

$$\begin{aligned} Q_{PE(x)} &= \frac{1}{x_1, z_1} + \frac{1}{x_1, z_2} + \frac{0.8}{x_2, z_1} + \frac{0.8}{x_2, z_2} \\ Q_{PE(z)} &= \frac{0.6}{x_1, z_1} + \frac{1}{x_1, z_2} + \frac{0.6}{x_2, z_1} + \frac{1}{x_2, z_2} \end{aligned} \quad (1.21)$$

## 1.7 SISTEMAS DIFUSOS

Las reglas difusas *if-then* (si-entonces) o sentencias difusas condicionales son expresiones de la forma *if A then B*, donde *A* y *B* son etiquetas de conjuntos difusos caracterizadas por funciones de pertenencia adecuadas. Este tipo de reglas se emplean para capturar los modos imprecisos del razonamiento, que juegan un papel esencial en la habilidad humana para la toma de decisiones en un ambiente de incertidumbre. Un ejemplo que describe un hecho muy simple es:

*If la presión es alta, then el volumen es pequeño*

donde la presión y el volumen son variables lingüísticas, alta y pequeña son valores lingüísticos o etiquetas que están caracterizadas por funciones de pertenencia.

Otro tipo de regla *if-then* difusa, propuesta por Takagi y Sugeno, posee conjuntos difusos involucrados en la premisa. Mediante el uso de este tipo de reglas, podemos describir la fuerza resistiva de un objeto en movimiento como sigue:

*If la velocidad es alta, then fuerza = k\*(velocidad)<sup>2</sup>*

donde, *alta*, que está ubicada en la parte correspondiente a la premisa, es una etiqueta lingüística caracterizada por una función de pertenencia. Sin embargo, la parte consecuente es descrita por una ecuación no lineal de la variable velocidad.

Ambos tipos de reglas *if-then* se han utilizado ampliamente en el modelado y el control.

Los sistemas difusos o de inferencia difusa también se conocen como sistemas basados en reglas difusas, modelos difusos, memorias asociativas difusas (FAM), o controladores difusos cuando se utilizan con fines de control (Aceves López y Aguilar Martin, 2006), (Chen, 2006), (Driankov, Hellendoorn, y Reinfrank, 1996), (Farinwata, Filev, y Langari, 2000), (Hopgood, 2001), (Lagunas Jiménez y Pitalúa Díaz, 2007), (Li, Chen, y Huang, 2001), (Matía, 1994), (Nguyen, Prasad, Walker, y Walker, 2003), (Passino y S., 1998), (Pitalúa Díaz, 2005), (Wang, A Course in Fuzzy Systems and Control, 1997), (Wang, Stable and Optimal Fuzzy Control of Linear Systems, 1998), (Ying, 2000), (Zadeh, 1965), (Zhang, Li, y Zhang, 2001) y (Zilouchian y Jamshidi, 2001). Básicamente un sistema difuso está compuesto de cinco bloques funcionales:

- Una base de reglas que contiene un número de reglas difusas *if-then*.
- Una base de datos que define las funciones de pertenencia de los conjuntos difusos utilizados en las reglas difusas.
- La unidad de toma de decisiones, la cual realiza las operaciones de inferencia basándose en las reglas.
- Una interfaz de *fuzzificación*, la cual transforma las entradas numéricas en grados de pertenencia con valores lingüísticos.
- Una interfaz de *defuzzificación*, la cual transforma los resultados difusos de la inferencia en valores numéricos.

Usualmente, la base de reglas y la base de datos son en conjunto conocidas como la base de conocimiento. La figura 1.5 muestra los bloques descritos anteriormente para un sistema difuso.

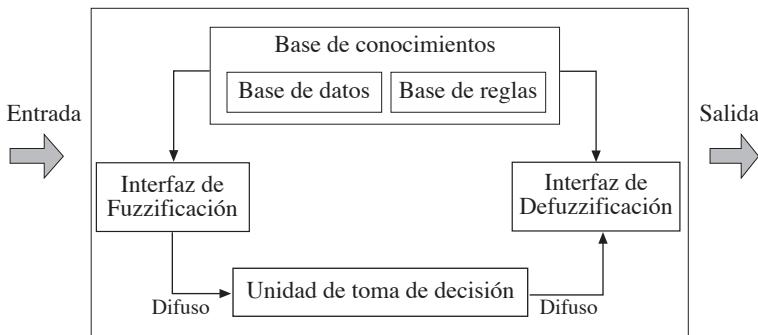


Figura 1.5. Bloques funcionales en un sistema difuso

Las etapas del razonamiento difuso realizadas por un sistema difuso son:

1. Comparar las variables de entrada con las funciones de pertenencia de la premisa para obtener los valores de pertenencia de cada etiqueta lingüística. A esta etapa se le denomina *fuzzificación*.
2. Combinar los valores de pertenencia en la parte correspondiente a la premisa con el fin de obtener los pesos de cada regla.
3. Generar el consecuente calificado dependiendo de la fuerza de disparo (peso) de cada regla.

4. Agregar los consecuentes calificados para producir una salida numérica. A esta etapa se le denomina *defuzzificación*.

Dependiendo de los tipos de razonamiento difuso y de las reglas *if-then* empleadas, la mayoría de los sistemas de inferencia pueden ser clasificados en tres tipos:

*Tipo 1.* La salida es el promedio de los pesos de la salida numérica de cada regla inducida por la fuerza de disparo (el producto o mínimo de los grados de pertenencia relacionados con la premisa) y las funciones de pertenencia de la salida. Las funciones de pertenencia de las salidas utilizadas en el esquema deben ser funciones monotónicas.

*Tipo 2.* La salida difusa es obtenida mediante la operación *max* aplicada a las salidas difusas calificadas (cada una de las cuales es igual al mínimo de las fuerzas de disparo y las funciones de pertenencia de salida de cada regla). Varios esquemas basados en las salidas difusas han sido propuestos para seleccionar la salida numérica final; algunas de ellas son el centroide del área, bisector del área, el criterio del máximo, etcétera.

*Tipo 3.* Corresponde a las reglas difusas *if-then* de Takagi y Sugeno. La salida de cada regla es la combinación lineal de las variables de entrada más un término constante y la salida final es el promedio de los pesos de la salida de cada regla.

La figura 1.6 muestra un sistema de inferencia difusa que utiliza dos reglas y dos entradas para diferentes tipos de reglas y razonamientos difusos como los que se han mencionado anteriormente.

## 1.8 INFERENCIA DIFUSA

Los tipos de inferencia más comunes son la compuesta y la basada en reglas individuales (Driankov, Hellendoorn, y Reinfrank, 1996), (Matía, 1994), (Passino y S., 1998), (Pitalúa Díaz, 2005) y (Wang, A Course in Fuzzy Systems and Control, 1997):

*Compuesta.* Consiste en combinar todas las reglas de la base de conocimientos en una relación difusa en  $X \times Y$ , la que puede ser vista como una sola regla *if-then*.

1. Si las reglas pueden ser vistas como estatutos independientes, entonces la operación de combinación que se emplea es la unión. Pero si las reglas están demasiado relacionadas entonces la intersección es la operación utilizada para la combinación.

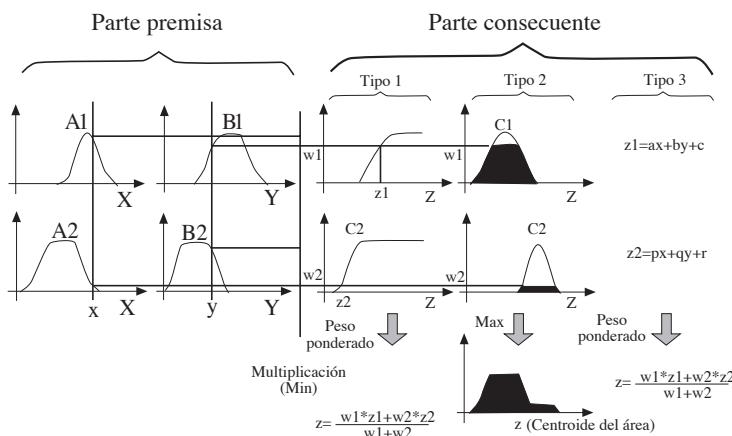


Figura 1.6 Mecanismos de razonamiento difuso y reglas *if-then* comúnmente usadas.

*Basada en reglas individuales.* En este caso cada regla de la base de conocimientos determina un conjunto difuso de entrada y un conjunto difuso de salida. La salida de la unidad de toma de decisiones es la combinación de los conjuntos individuales de salida. La combinación se realiza mediante la unión o la intersección.

## 1.9 FUZZIFICACIÓN

Es el mapeo de un punto real valuado  $x^* \in U \subset R^n$  a un conjunto difuso  $A'$  en  $U$ .

Los tres fuzzificadores más empleados son el singleton, gaussiano y triangular (Driankov, Hellendoorn, y Reinfrank, 1996), (Matía, 1994), (Passino y S., 1998), (Pitalúa Díaz, 2005) y (Wang, A Course in Fuzzy Systems and Control, 1997):

*Fuzzificador Singleton.*

$$\mu_{A'}(x) = \begin{cases} 1 & \text{si } x = x^* \\ 0 & \text{en otro caso} \end{cases} \quad (1.22)$$

*Fuzzificador Gaussiano.*

$$\mu_{A'}(x) = t[e^{-\frac{(x_1 - x_1^*)^2}{a_1}}, \dots, e^{-\frac{(x_n - x_n^*)^2}{a_n}}] \quad (1.23)$$

donde  $a_i$  son parámetros positivos y la norma-t por lo general, se escoge como el producto algebraico o el *min*.

*Fuzzificador Triangular.*

$$\mu_{A^*}(x) = t \left[ 1 - \frac{|x_1 - x_1^*|}{b_1}, \dots, 1 - \frac{|x_n - x_n^*|}{b_n} \right] \quad (1.24)$$

donde  $b_i$  son parámetros positivos y la norma-t por lo general se escoge como el producto algebraico o el *min*.

## 1.10 DEFUZZIFICACIÓN

Es el mapeo de un conjunto difuso  $B'$  en  $V \subset R$ . (el cual es la salida de la unidad de toma de decisiones) a un punto real valuado  $y^* \in V$ .

Los tres defuzzificadores más empleados son el centro de gravedad, promedio de centros y el máximo (Driankov, Helleidoorn, y Reinfrank, 1996), (Matía, 1994), (Passino y S., 1998), (Pitalúa Díaz, 2005) y (Wang, A Course in Fuzzy Systems and Control, 1997):

*Centro de gravedad.*

$$y^* = \frac{\int_{V_a} y \mu_{B'}(y) dy}{\int_{V_a} \mu_{B'}(y) dy} \quad (1.25)$$

Si se maneja  $\mu_B(y)$  como la función de densidad de probabilidad de una variable aleatoria, entonces el defuzzificador del centro de gravedad calcula el valor medio de la variable aleatoria.

$V_\alpha$  se define como:

$$V_\alpha = \{y \in V \mid \mu_{B'}(y) \geq \alpha\}$$

y  $\alpha$  es una constante.

*Promedio de centros:*

$$y^* = \frac{\sum_{i=1}^M \bar{y}^i w_i}{\sum_{i=1}^M w_i} \quad (1.26)$$

donde  $\bar{y}^i$  y  $w_i$  son el centro y altura individual de cada función de membresía ( $i$ ) que forma parte del conjunto difuso.

*Máximo:* Se selecciona un punto en altura ( $B'$ ).

$$\text{altura } (B') = \{y \in V \mid \mu_{B'}(y) = \sup_{y \in V} \mu_{B'}(y)\} \quad (1.27)$$

donde la altura ( $B'$ ) es el conjunto de todos los puntos en  $V$ , en el cual  $\mu_{B'}(y)$  alcanza su valor máximo.

## 1.11 EJEMPLO DE UN SISTEMA DE CONTROL DIFUSO

La unidad compacta FESTO PCS que se muestra en la figura 1.7, es el sistema de tanques usado para este ejemplo.

Para el control difuso del sistema de tanques, se utilizó una estructura de programa basado en flujo de datos conocido como Labview 8.2 de la compañía National Instruments. El diseño de las reglas difusas *if-then* se utilizó para la implementación de los controladores Proporcional-Integral también conocidos como PI (ver Apéndice) en software, en el control del nivel del líquido en el tanque 1 valiéndose del sensor ultrasónico y para el control de la temperatura en el tanque 2 empleando el sensor de temperatura RTD.

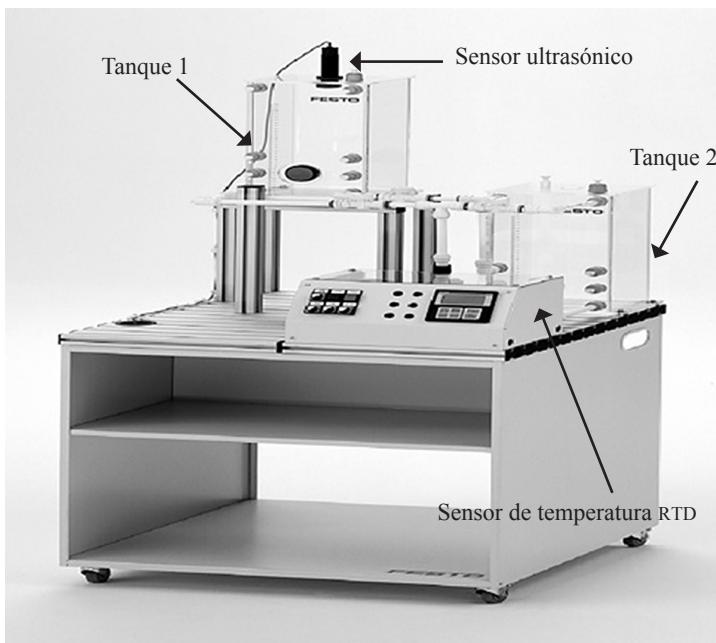


Figura 1.7. Sistema de tanques (Festo Compact Unit PCS).

Al tener las reglas éstas se declaran en el programa Labview, utilizando como defuzzificador el centro de gravedad. Una vez diseñados los controladores difusos, se programan en un dispositivo PAC (Controlador Autómata Programable CFP 2020, de la compañía National Instruments) por medio de la computadora y se ejecutan en tiempo real debido a que el programa funciona como sistema embebido. La figura 1.8 presenta el diagrama esquemático del sistema de control difuso interactuando con el dispositivo PAC.

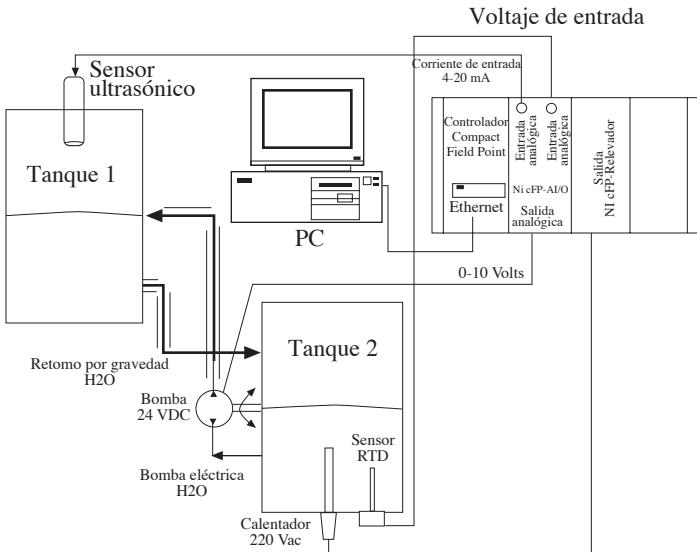


Figura 1.8 Diagrama esquemático del sistema de control difuso de nivel de líquido y temperatura.

### 1.11.1 FUNCIONAMIENTO DE LOS CONTROLADORES DIFUSOS

El objetivo de los controladores PI difusos (ver Figura 1.9) del sistema de tanques, es lograr un control del nivel de líquido en el tanque 1 y controlar la temperatura del líquido en el tanque 2 y demostrar experimentalmente, que es una mejor alternativa de control sin necesidad de obtener un modelo matemático complejo.

Para nuestro caso se utiliza un sensor ultrasónico para medir los cambios de nivel del líquido en el tanque 1, este dispositivo manda una señal analógica del tipo de corriente el cual llega al

módulo de entradas y salidas analógicas del PAC, que sirve como una tarjeta de adquisición de datos, en este caso obtenemos el valor Nivel. Por otro lado, tenemos un sensor RTD de temperatura el cual nos permite obtener los cambios de temperatura del líquido en el tanque 2, esta señal es del tipo de cambio de resistencia, la cual llega a un modulo especial de RTD que puede acondicionar la señal y ser leída por el PAC para obtener el valor de temperatura. El anexo 1 presenta las fotografías de las conexiones realizadas.

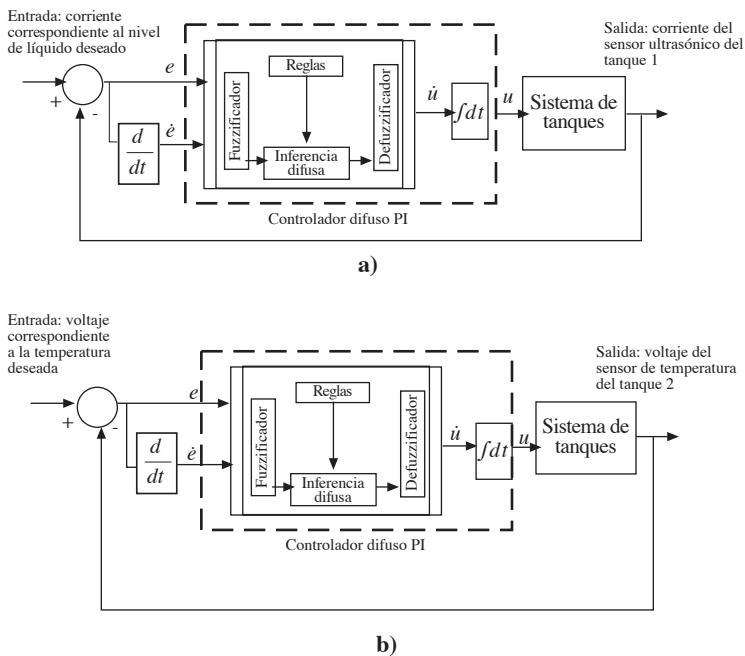


Figura 1.9 Diagrama de bloques del controlador PI difuso:

a) Para el nivel de líquido en el tanque 1; b) Para la temperatura en el tanque 2.

Hasta ahora tenemos nuestras dos señales de entrada del sistema Nivel y Temperatura. Además de contar con los dispositivos actuadores, para el caso del tanque 1 tenemos la bomba DC que se encarga de alimentarlo de líquido proveniente del tanque 2. También se cuenta con una resistencia térmica de 220 Vac cuya función es calentar el líquido del tanque 2. Lo anterior genera la base de reglas difusas que pueden resumirse en las tablas 1.3 y 1.4 (Driankov, Hellendoorn, y Reinfrank, 1996), (Li, Chen, & Huang, 2001), (Matía, 1994), (Passino y S., 1998), (Pitalúa Díaz, 2005), (Wang, A Course in Fuzzy Systems and Control, 1997), (Ying, 2000) y (Zhang, Li, y Zhang, 2001).

Nótese que la salida (parte sombreada) en las tablas 1.3 y 1.4 implica el funcionamiento de la bomba y la resistencia térmica.

Las figuras 1.10 y 1.11 presentan los conjuntos difusos de entrada y de salida de los controladores PI difusos de nivel de líquido y temperatura.

Figura 1.9 Diagrama de bloques del controlador PI difuso:

- a) Para el nivel de líquido en el tanque 1; b) Para la temperatura en el tanque 2.

TABLA 1.3 BASE DE REGLAS DIFUSAS PARA EL CONTROL DE NIVEL DE LÍQUIDO EN EL TANQUE 1

		Error de Nivel ( $e$ )		
		Alto	Medio	Bajo
Cambio en Error de Nivel ( $\dot{e}$ )	Bajando	Apagado	Encendido	Encendido
	OK	Apagado	OK	Encendido
	Subiendo	Apagado	Apagado	Encendido



TABLA 1.4 BASE DE REGLAS DIFUSAS PARA EL CONTROL DE  
TEMPERATURA EN EL TANQUE 2

Cambio en error de temperatura ( $\dot{e}$ )	Error de Temperatura (e)			
	Alto	Medio	Bajo	
	Bajando	Apagado	Apagado	Encendido
	OK	Apagado	OK	Encendido
Subiendo	Encendido	Encendido	Encendido	Encendido

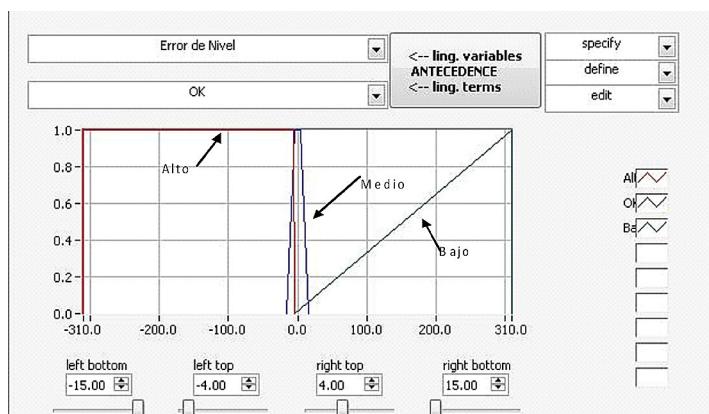


TABLA 1.5 CONDICIONES *IF-THEN* OBTENIDAS DE LA BASE DE REGLAS  
DE LA TABLA 1.3

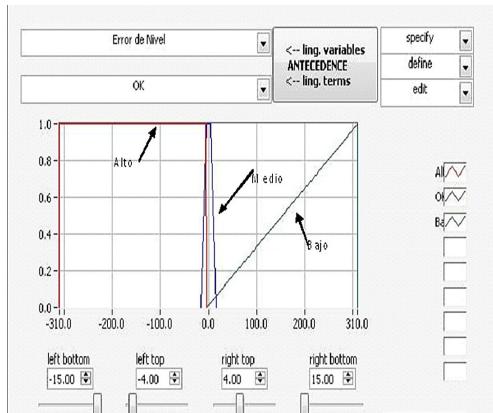
1. If $e$ es Alto y $\dot{e}$ es Bajando, then $u$ es Apagado
2. If $e$ es Alto y $\dot{e}$ es OK, then $u$ es Apagado
3. If $e$ es Alto y $\dot{e}$ es Subiendo, then $u$ es Apagado
4. If $e$ es Medio y $\dot{e}$ es Bajando, then $u$ es Encendido
5. If $e$ es Medio y $\dot{e}$ es OK, then $u$ es OK
6. If $e$ es Medio y $\dot{e}$ es Subiendo, then $u$ es Apagado
7. If $e$ es Bajo y $\dot{e}$ es Bajando, then $u$ es Encendido
8. If $e$ es Bajo y $\dot{e}$ es OK, then $u$ es Encendido
9. If $e$ es Bajo y $\dot{e}$ es Subiendo, then $u$ es Encendido

TABLA 1.6 CONDICIONES *IF-THEN* OBTENIDAS DE LA BASE  
DE REGLAS DE LA TABLA 1.4

1. <i>If e</i> es Alto y <i>è</i> es Bajando, <i>then</i> <i>ù</i> es Apagado
2. <i>If e</i> es Alto y <i>è</i> es OK, <i>then</i> <i>ù</i> es Apagado
3. <i>If e</i> es Alto y <i>è</i> es Subiendo, <i>then</i> <i>ù</i> es Encendido
4. <i>If e</i> es Medio y <i>è</i> es Bajando, <i>then</i> <i>ù</i> es Apagado
5. <i>If e</i> es Medio y <i>è</i> es OK, <i>then</i> <i>ù</i> es OK
6. <i>If e</i> es Medio y <i>è</i> es Subiendo, <i>then</i> <i>ù</i> es Encendido
7. <i>If e</i> es Bajo y <i>è</i> es Bajando, <i>then</i> <i>ù</i> es Encendido
8. <i>If e</i> es Bajo y <i>è</i> es OK, <i>then</i> <i>ù</i> es Encendido
9. <i>If e</i> es Bajo y <i>è</i> es Subiendo, <i>then</i> <i>ù</i> es Encendido



a)



b)

Figura 1.10. Conjuntos difusos de entrada del controlador PI difuso de nivel de líquido: a) Conjunto difuso error de nivel (  $e$  );  
b) Conjunto difuso cambio de error de Nivel (  $\dot{e}$  )

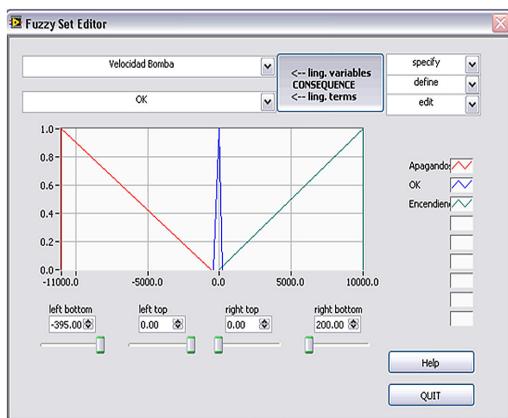
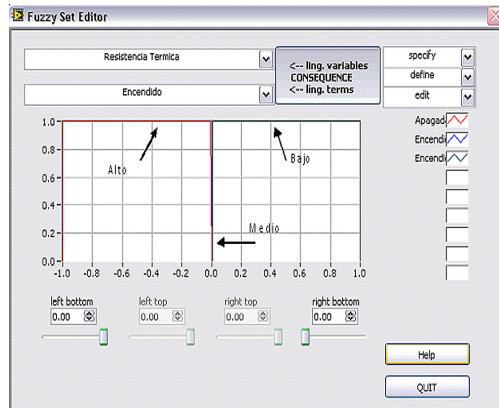
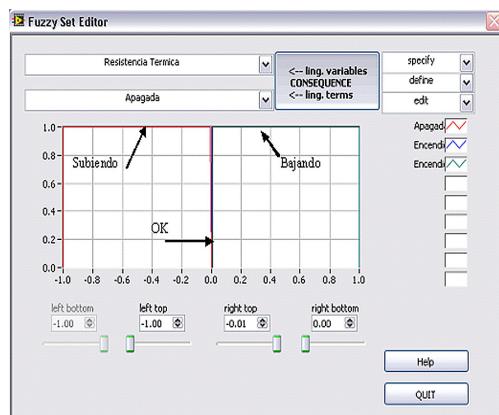


Figura 1.11 Conjunto difuso de salida del controlador PI difuso de nivel de líquido.



a)



b)

Figura 1.12. Conjuntos difusos de entrada del controlador PI difuso de temperatura: a) Conjunto difuso error de temperatura ( $e$ ); b) Conjunto difuso cambio de error de temperatura ( $\dot{e}$ )

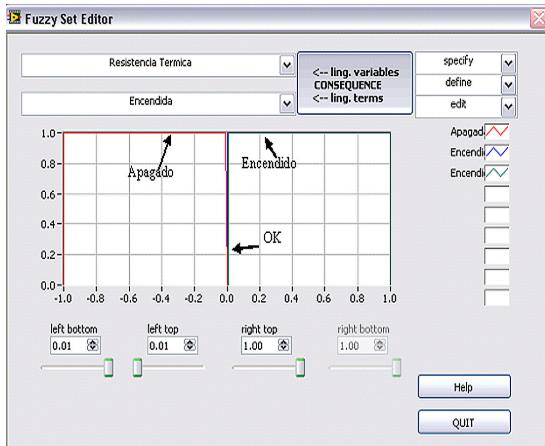


Figura 1.13. Conjunto difuso de salida del controlador PI difuso de temperatura.

Para concluir se presentan en la figura 1.14 las respuestas de cada sistema de control difuso, junto con su trayectoria lingüística (*linguistic trajectory*) que es el comportamiento gráfico de las reglas difusas como resultado de la activación de cada función de membresía (base de reglas) por la regla difusa en turno (Driankov, Hellendoorn, y Reinfrank, 1996), (Matía, 1994), (Passino y S., 1998), (Pitalúa Díaz, 2005) y (Wang, A Course in Fuzzy Systems and Control, 1997).

La figura 1.14 muestra que las respuestas del sistema de control difuso de nivel de líquido y temperatura, son estables debido a que alcanzan y se establecen en las referencias propuestas (*setpoint*) de 228 cm para el nivel y de 33°C para la temperatura. Lo anterior se comprueba al observar que la trayectoria lingüística generada por cada base de reglas se establece en el rango deseado, localizado

en la celda central OK. Nótese que las celdas que se muestran en la trayectoria lingüística de la figura 1.14 varían su altura y amplitud debido a los rangos establecidos en los conjuntos difusos.

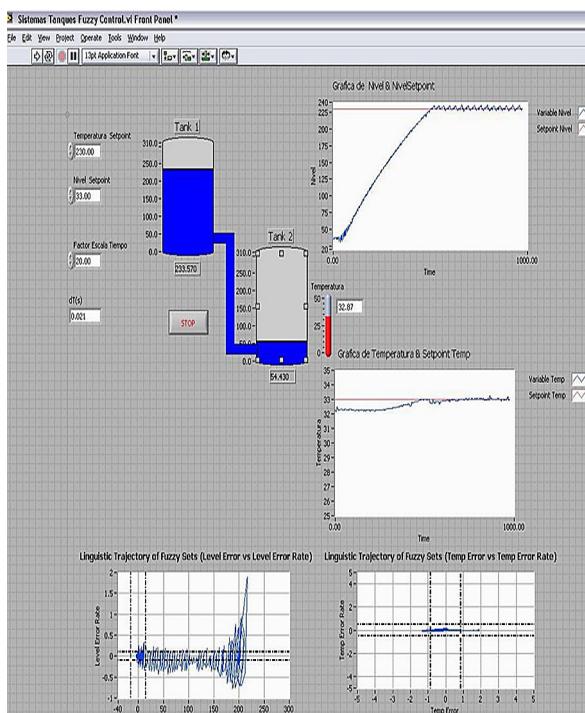


Figura 1.14. Respuesta del sistema de control difuso

## **CAPÍTULO 2. REDES NEURONALES ARTIFICIALES**

### **2.1 FUNDAMENTOS BIOLÓGICOS**

#### *2.1.1 LOS SENTIDOS Y SU ENTORNO*

Al interactuar el hombre con el mundo, recibe una cantidad impresionante de información en diferentes formas, así como también emite información al exterior o dentro de su mismo organismo. Los sonidos que escuchamos, las cosas que vemos, que tocamos y que olemos, es información que llega a nuestro cuerpo captada por los sentidos y se transmite al cerebro para que éste la almacene y procese, dando origen a las emociones, los recuerdos, los pensamientos, la memoria, el juicio, la adopción de decisiones e incluso la conciencia (Tapia, 2001).

También, en el interior del hombre viaja información que le permite a sus órganos regular su funcionamiento. De hecho, la mayor parte de estos procesos pasan desapercibidos, ya que se ejecutan de manera autónoma. Todo esto es posible gracias a

nuestro sistema nervioso, mediante el cual podemos percibir el ambiente que nos rodea, comunicarnos con él y darnos cuenta del lugar que ocupamos en el universo.

### *2.1.2 LAS CÉLULAS ESPECIALIZADAS*

Nuestros sentidos están provistos de órganos que realizan diferentes tareas, las cuales son ejecutadas por células especializadas que reciben información del medio ambiente y la transmiten hacia el cerebro. A este tipo de células las conocemos con el nombre de neuronas.

La neurona es la unidad fundamental de nuestro sistema nervioso. Aunque el funcionamiento de una neurona es relativamente simple, el increíble número de ellas contenidas en nuestro sistema nervioso forma una extensa red que lleva a cabo funciones extremadamente complejas, muchas de las cuales hasta la fecha son una interrogante.

Aunque existe una gran variedad en cuanto a la forma de las neuronas que hace prácticamente imposible realizar una clasificación, es posible establecer tres categorías en base a su funcionamiento (Ramón y Cajal, 1995). Por ejemplo, las neuronas receptoras son las responsables de captar información conocida como estímulos. El calor, la presión, el frío y la textura, son estímulos que reciben las neuronas que corresponden únicamente al sentido del tacto. Las diferentes frecuencias de las ondas sonoras, son información que las neuronas relacionadas con el oído se encargan de recibir. Las partículas volátiles le corresponden a las del olfato y los sabores a las del gusto. Esto significa que las neuronas que pertenecen a un sentido (grupo), no pueden ejercer ninguna influencia sobre otro. De ser así, necesitaríamos del olfato

para escuchar algo o sentiríamos la rigidez de algún objeto con sólo mirarlo, o tal vez al tocar algo podríamos percibir el olor que desprendiese.

Además de percibir el mundo exterior mediante las neuronas receptoras, existe un mundo interno formado por nuestros fluidos, fundamentalmente la sangre y nuestros órganos, cuyo estado es informado mediante neuronas receptoras que igualmente transmiten información a regiones de la médula espinal y al cerebro, las cuales trabajan de forma totalmente independiente de la voluntad.

Otro tipo de neuronas son las motoras, mediante las cuales podemos controlar todos los músculos de nuestro cuerpo. El habla, los movimientos de nuestras extremidades, los reflejos, un simple parpadeo, el constante latido cardiaco; es decir, cualquier movimiento que realice nuestro organismo, sea de manera consciente o inconsciente, depende estrictamente del control que ejercen este tipo de neuronas sobre él. Mediante las neuronas motoras es como nuestro organismo se autorregula, nos permite expresarnos y movernos, de ahí el nombre de estas células.

Por último tenemos a las neuronas de asociación, que relacionan neuronas receptoras entre sí, neuronas receptoras con motoras, neuronas motoras entre ellas y también establecen relación entre grupos de células con funciones asociativas únicamente.

### *2.1.3 ESTRUCTURA DE LAS NEURONAS*

En la figura 2.1, se muestran células que, aunque pertenecen a diferentes sentidos, poseen una estructura básica común (Alcaraz, 2000); en uno de sus extremos se desprenden numerosas ramificaciones cortas que conocemos como dendritas y son las que

cumplen con la función receptora de la información de la célula. En el otro extremo se encuentra una proyección larga y delgada que se llama axón. Esta prolongación se ramifica en su parte terminal abarcando con ello piel, órganos o músculos mediante unos botones terminales. El axón es el encargado de enviar la información procesada por el cuerpo de la neurona –también llamado soma— hacia el cerebro en el caso de las neuronas receptoras o a la parte periférica en el caso de las motoras.

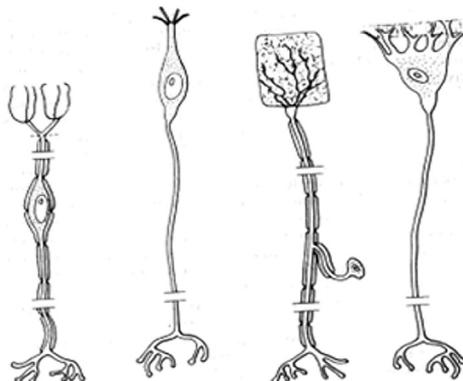


Figura 2.1 Neuronas receptoras. De izquierda a derecha: neurona auditiva, neurona olfatoria, neurona sensible al tacto, neurona que capta el estiramiento de los músculos.

Dentro del soma o cuerpo de la neurona se encuentran nutrientes que llevan a cabo la síntesis necesaria para emitir respuestas provocadas por estímulos. Por último encontramos la unión de las dendritas con el soma, llamada sinápsis, la cual juega un papel fundamental en la recepción de mensajes y en la comunicación entre neuronas con otra.

En la figura 2.2 se muestra, mediante flechas, cómo la información es recibida en las dendritas de la neurona, llegando al cuerpo, donde es procesada para después emitir una señal a través del axón, el cual puede ser muy largo pues llega a medir más de un metro.

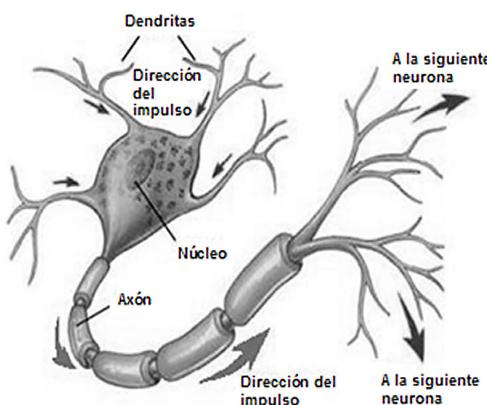


Figura 2.2 Componentes de una neurona motora

#### 2.1.4 COMUNICACIÓN ENTRE NEURONAS

Las neuronas no son una continuación de otra, sino que son unidades individuales separadas las cuales están cubiertas por una membrana que las separa. Pero el extraordinario ordenamiento, al igual que el funcionamiento de nuestro sistema nervioso, es logrado debido a la capacidad de comunicarse entre sí. Al hablar de comunicación entre estas entidades nos referimos a un lenguaje eléctrico o químico.

Tal como se muestra en la figura 2.3, en algunas zonas, las membranas de una neurona se encuentran lo suficientemente cerca para que la corriente eléctrica que circula por el axón supere la distancia que las separa. Esta distancia, que en algunos casos es de dos nanómetros, en muchos otros sitios suele alcanzar hasta los veinte nanómetros y no puede ser superada mediante un brinco de la corriente eléctrica. En este caso las neuronas se valen de los transbordadores, que son compuestos químicos contenidos en esta separación que llevan el nombre de neurotransmisores.

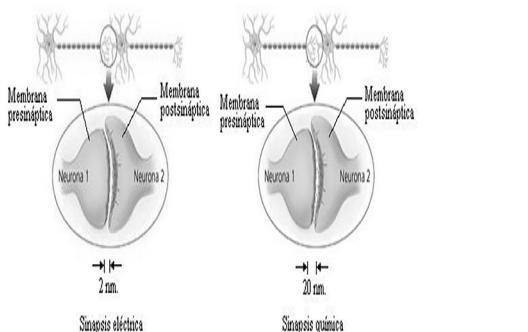


Figura 2.3 Separación sináptica.

De hecho, la mayor parte de la comunicación entre las células que forman nuestro sistema nervioso es de tipo químico. Al ser transmitido un impulso a través del axón, éste llega a los botones terminales en los que se encuentran los neurotransmisores que serán liberados de la membrana presináptica para llegar a la zona receptora de la siguiente neurona, llamada membrana postsináptica. Una vez que se encuentran los neurotransmisores del otro lado, pueden suceder dos cosas: que la neurona emita o no una señal. Esto depende principalmente de lo que ocurra dentro

del cuerpo de la neurona. Si el neurotransmisor llega a excitar lo suficiente la membrana, esta disparará una señal que correrá a través de su axón.

Para ejemplificar el funcionamiento de una neurona y, simplificando notablemente su funcionamiento de una neurona, digamos que una célula recibe ocho impulsos excitadores y cinco impulsos inhibidores. Al sumar algebraicamente los estímulos tendríamos un total de tres impulsos excitadores. Si el umbral que posee la neurona es menor a tres, entonces un impulso correrá por el axón a otras células para provocar un proceso similar al que dio origen el mismo estímulo tal como se muestra en la figura 2.4.

Información que recibe la neurona	Nivel de excitación de la membrana	Envío de señal por el axón
	Bajo (normal)	No
	Ligeramente aumentado	No
	Muy aumentado	Si
	Disminuido	No
	Bajo (normal)	No

Figura 2.4 Proceso de respuesta en una neurona.

El proceso descrito se vuelve todavía más interesante; no obstante que las neuronas forman grupos en base a la función que realizan, una neurona de tipo inhibidor, además de conectarse con otras de su mismo tipo, puede comunicarse con neuronas excitadoras. Esto establece como consecuencia un tipo de autorregulación en mecanismos como los de los músculos. Así por ejemplo, mientras de una neurona se desprende una señal excitadora sobre un músculo, ésta puede llegar a inhibir a otra neurona de su músculo antagonista, permitiéndonos el control adecuado de nuestras extremidades.

Aunado a lo anterior, nos encontramos con otro tipo de neuronas que forman circuitos en el cerebro cuya función es regular finamente los movimientos producidos por los músculos. Gracias a ello es que podemos realizar movimientos extremadamente precisos para las actividades cotidianas. El simple movimiento que hacemos cuando llevamos un vaso a nuestra boca, el control que ejercemos sobre un lápiz cuando escribimos, o los movimientos al caminar o al bailar, son ejemplos de que la comunicación entre neuronas puede regularse mediante la síntesis química y la compleja conexión existente entre ellas.

En la figura 2.5 se muestra cómo el impulso que emite una neurona no es constante y su capacidad de reaccionar varía con el tiempo, hasta volver a su estado de reposo. La comunicación tiene lugar a través de trenes de pulsos por lo que los mensajes se encuentran modulados en frecuencia.

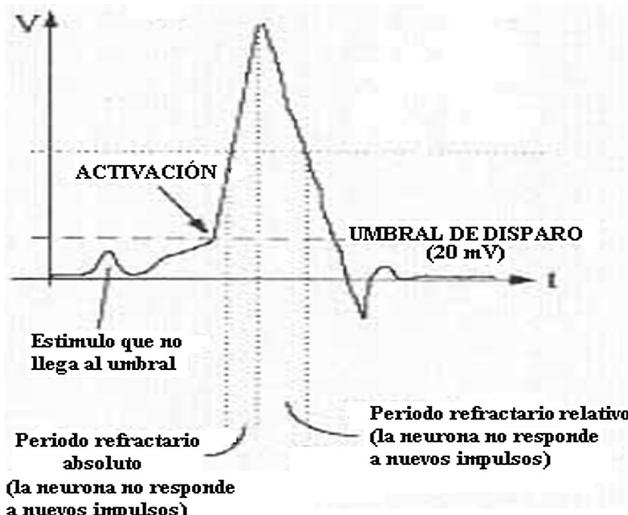


Figura 2.5 Impulso emitido por una neurona.

### 2.1.5 EL APRENDIZAJE HEBBIANO

Para complementar lo que hemos visto hasta ahora sobre los aspectos biológicos de las neuronas y las redes que forman, es necesario analizar como es que las neuronas captan y concentran información, logrando con esto que se origine el proceso de aprendizaje.

En el año 1949, Donald O. Hebb (Hebb, 1975) establece el siguiente postulado: “Cuando un axón de una celda A está suficientemente cerca para conseguir excitar una celda B y repetidamente toma parte en su activación, se lleva a cabo un proceso de crecimiento o cambio metabólico en una o ambas celdas, de tal forma que la eficiencia de A aumenta cuando la celda que va

activar es B". Por celda, Hebb define a un conjunto de neuronas fuertemente conectadas a través de una estructura compleja y la eficiencia, como la intensidad o magnitud de la conexión.

Podemos decir entonces que el fortalecimiento que sucede en la conexión entre neuronas es provocado, en parte, por la incidencia de información de nuestros sentidos. Esto significa que nuestras redes de neuronas responden y se modifican como consecuencia del contacto con nuestro mundo exterior, quedando registradas biológicamente las experiencias que tenemos. Es así como podemos recordar, analizar y reflexionar sobre la información que de manera implícita o explícita haya sido procesada por nuestro cerebro.

## 2.2 REDES NEURONALES ARTIFICIALES

### 2.2.1 ANTECEDENTES HISTÓRICOS

El interés por las redes neuronales data de principios de la década de 1940, como queda de manifiesto en el trabajo de McCulloch y Pitts (McCulloch, 1943). Ellos propusieron modelos neuronales en forma de dispositivos binarios que implican transiciones repentinas 0-1 y 1-0, en el estado de las neuronas, como base para el modelado de sistemas neuronales. Posteriormente, el trabajo de Hebb se basó en modelos matemáticos que intentaron abordar el concepto de aprendizaje mediante el refuerzo y la asociación.

Durante la mitad de la década de 1950 y principios de la de 1970, un tipo de sistemas denominados máquinas de aprendizaje, ideados por Rosenblatt, crearon grandes expectativas entre los investigadores quienes ponían en práctica la teoría de reconocimiento de patrones. El motivo del gran interés por estas máquinas

denominadas perceptrones, fue el desarrollo de demostraciones matemáticas. Entrenaban perceptrones utilizando conjuntos de entrenamiento linealmente separables, que convergían en una solución con un número finito de iteraciones. Las soluciones tomaban la forma de coeficientes de hiperplanos capaces de separar correctamente las clases representadas por los patrones del conjunto de entrenamiento.

Por desgracia, las esperanzas creadas con el descubrimiento de lo que parecía ser un buen modelo teórico del aprendizaje pronto se desvanecieron. El perceptrón y algunas de sus generalizaciones de aquel entonces, eran insuficientes para la mayoría de las tareas de reconocimiento de patrones de significación práctica. Los intentos posteriores de mejora de la potencia de máquinas de tipo perceptrón considerando múltiples niveles de estos dispositivos, aunque eran conceptualmente atractivos, carecían de algoritmos de entrenamiento efectivos como los que habían creado el interés en el propio perceptrón. Nilsson (Nilsson, 1965) resumió el estado de las investigaciones en el campo de las máquinas de aprendizaje a mediados de la década de 1970. Pocos años después, Minsky y Papert presentaron un análisis descorazonador sobre la limitación de las máquinas del tipo perceptrón simple (Minsky y Papert, 1969). Este punto de vista imperó hasta mediados de los ochenta, como se desprende de los comentarios de Simon (Simon, 1986). En esta obra, originalmente publicada en francés en 1984, Simon descarta el perceptrón utilizando el título “Nacimiento y muerte de un mito”.

Los resultados obtenidos por Rumelhart, Hinton y Williams en el desarrollo de nuevos algoritmos de entrenamiento, para perceptrones multicapa, cambiaron las cosas considerablemente (Rumelhart, Hinton, y Williams, 1986). El método básico que emplearon, a menudo considerado como Regla Delta Generalizada

para el Aprendizaje por Retropropagación, ofrece una técnica de entrenamiento efectiva para estas máquinas. Con ésta se ha conseguido que el perceptrón multicapa se constituya uno de los principales modelos de redes neuronales artificiales.

### *2.2.2 EL PERCEPTRÓN SIMPLE*

En 1959 el fisiólogo Frank Rosenblatt publicó los resultados del modelo matemático de una neurona biológica, denominado Perceptrón. Tanto el modelo matemático como su algoritmo de entrenamiento, generaron mucho interés en ese entonces debido a su habilidad para aprender y reconocer patrones sencillos. A continuación se tratarán los aspectos principales de este modelo, pero antes es necesario revisar algunos de los elementos claves de una neurona biológica, vistos en el capítulo anterior, con el propósito de establecer un vínculo entre el fenómeno biológico y el modelo artificial:

- Las neuronas son entidades individuales separadas unas de otras.
- Una neurona es un sistema multientradas por lo que puede recibir varias señales al mismo tiempo.
- La fuerza de unión de sus entradas con su cuerpo puede ser modificada produciendo el fenómeno de aprendizaje.
- La neurona tiene un elemento central de procesamiento (soma) que integra sus entradas realizando una suma algebraica de las señales incidentes.

- Una neurona es un sistema de salida la cual provee un pulso eléctrico (excitación) si la suma algebraica rebasa el umbral de la neurona. Así mismo, no provee ninguna respuesta (inhibición) si no llega a rebasar el umbral de la neurona.

Una neurona biológica toma las señales provenientes de otras neuronas y las modifica de acuerdo a la fuerza de unión de cada entrada (sinápsis) con el cuerpo de la neurona. Esto matemáticamente se expresa de la siguiente forma:

Entradas (dendritas):	$x_0, x_1, \dots x_n$
Fuerza de unión de cada entrada (sinapsis):	$w_0, w_1, \dots w_n$
Entradas modificadas:	$w_0x_0, w_1x_1, \dots w_nx_n$

donde:

$x$  = señal de entrada a la neurona.

$w$  = fuerza de unión da cada entrada con el cuerpo de la neurona.

El cuerpo de la neurona toma las entradas modificadas y realiza una suma algebraica, que es modelada mediante:

$$NA = w_0x_0 + w_1x_1 + \dots + w_nx_n \quad (2.1)$$

donde:

NA = nivel de activación de la neurona.

Para proveer una respuesta a los estímulos de entrada, la neurona hace una comparación de su nivel de activación con un umbral propio, de tal forma que: si el NA es mayor que el umbral, la neurona genera un pulso de salida a través de su axón, y se dice que está excitada. De lo contrario; no genera respuesta alguna, y se dice que esta inhibida.

La respuesta de la neurona es representada mediante:

$$y = f \left[ \sum_{i=0}^{N-1} x_i w_i - \theta \right] \quad (2.2)$$

donde:

$\theta$ .= umbral de la neurona para ser activada.

En forma numérica la respuesta define el estado de excitación mediante una función de activación o función de transferencia, la cual tiene como propósito mapear, escalar o limitar el dominio de la neurona en su entrada a un rango específico a la salida de la misma. Esta respuesta puede ser representada mediante una función conocida como escalón y es representada mediante:

$$y = \begin{cases} +1 & \text{Si } NA > \theta \\ -1 & \text{Si } NA \leq \theta \end{cases} \quad (2.3)$$

La representación gráfica del modelo general de un perceptrón simple puede ser definida de la siguiente forma:

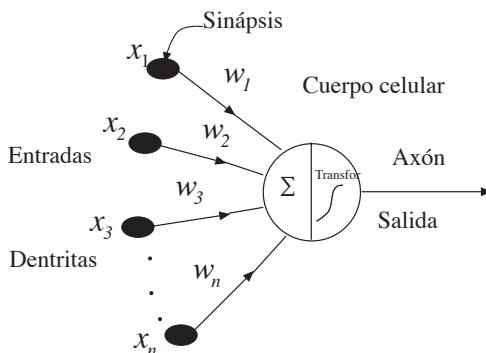


Figura 2.6 Representación gráfica de un perceptrón simple.

Una gráfica de la respuesta o función del nivel de activación es la siguiente:

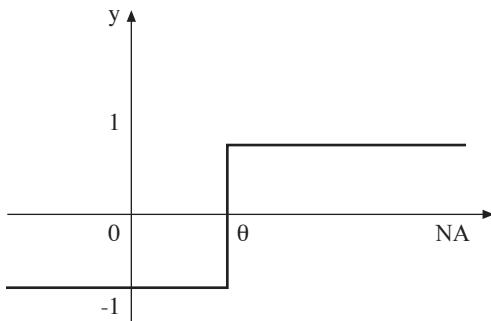


Figura 2.7 Representación gráfica de la función de activación del perceptrón simple.

Un perceptrón simple tiene la capacidad de elegir si una entrada pertenece a una de dos clases. Una técnica muy útil para analizar el comportamiento del perceptrón es graficar en un plano ambas clases. Para esto, lo primero que ha de hacerse es encontrar las características que se deben extraer de cada clase.

La salida del Perceptrón es una función de las entradas  $x_i$ , de los pesos  $w_i$  y del umbral  $\theta$ , es decir, del nivel de activación  $NA$ . En una representación gráfica de la función del nivel de activación, se puede observar cómo ésta puede ser usada como función discriminante  $y$ , de acuerdo al valor que provea, decidir el tipo de clase que representa un par de valores  $x_0, x_1$  dado.

Las propiedades de la función  $NA$  pueden conocerse si ésta se iguala a cero y se obtiene la gráfica que produce  $x_1$  con respecto a  $x_0$ .

$$NA = w_0x_0 + w_1x_1 + \dots + w_nx_n - \theta = 0 \quad (2.4)$$

Por tanto:

$$x_1 = -\frac{w_0}{w_1}x_0 + \frac{\theta}{w_1} \quad (2.5)$$

El resultado equivale a una línea recta con pendiente  $-w_0/w_1$ . En la figura 2.8 se puede observar que el perceptrón separa el espacio de entradas en dos regiones a través de una línea recta, por lo que es capaz de distinguir entre dos tipos de vectores de entrada: los que están arriba (*A*) y los que están por debajo de la línea (*B*).

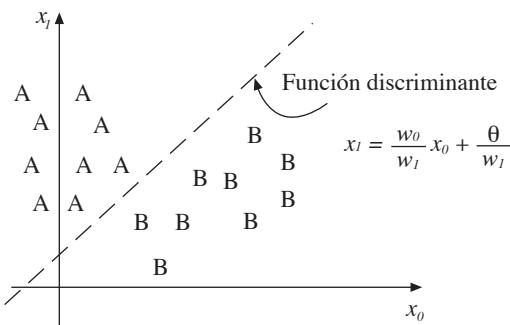


Figura 2.8 Clases linealmente separables.

El análisis anterior demuestra que el modelo del perceptrón simple puede realizar el trabajo de categorización entre dos clases, siempre y cuando éstas sean separables en el espacio característico por una sola línea recta.

El trabajo restante entonces se centra en elegir los valores adecuados para los pesos  $w_i$  y el umbral  $\theta$ , de tal forma que la línea recta esté posicionada justo en el lugar donde las clases a reconocer estén completamente separadas.

### *2.2.3 ALGORITMO DE ENTRENAMIENTO DEL PERCEPTRÓN SIMPLE*

El entrenamiento es un algoritmo que le proporciona a un Perceptrón la capacidad de aprender una determinada tarea. Este proceso consiste en ir ajustando los pesos gradualmente hasta que el vector de salida resultante coincida con el valor de salida deseado.

En base al tipo de aprendizaje podemos establecer dos categorías principales (Nilsson, 1965):

- Aprendizaje supervisado.
- Aprendizaje no supervisado.

El aprendizaje supervisado parte de un vector de entrada del cual se conoce su vector de salida deseado o al menos una aproximación a él. Al par de vectores representando los valores de entrada y salida deseada se le denomina par de entrenamiento. La diferencia o cambio existente entre el vector de salida y el vector de salida deseado, es reducida a través de un algoritmo que va ajustando los pesos hasta que la diferencia entre ambas salidas sea mínima.

En el aprendizaje no supervisado se desconoce la salida y únicamente se proporciona un vector de entrada. Lo que se busca es generar después varios vectores de entrada-salida que sean

consistentes. Es decir, que los pesos se vayan ajustando, poco a poco, a través del reconocimiento de patrones, regularidades, propiedades estáticas, etcétera. Así, las entradas similares producirán el mismo tipo de salida.

En su trabajo, Rosenblatt (Rosenblatt, *Principles of Neurodynamics*, 1959) dedujo un método para ajustar los valores de los pesos del perceptrón de tal forma que éste posicione la línea discriminante en una región adecuada para distinguir entre las clases. A este método de ajuste le llamó Regla de Aprendizaje o Algoritmo de Entrenamiento del Perceptrón.

El algoritmo de aprendizaje consta de los siguientes cinco pasos (Simon, 1986):

- *Paso 1.* Inicialización de pesos y umbral. Ajuste de  $w_i(0)$  ( $0 \leq i \leq N-1$ ) y  $\theta$  a un valor aleatorio. Donde  $w_i(t)$  es el peso de la entrada  $i$  a un tiempo  $t$  y  $\theta$  es el umbral del nodo de salida.
- *Paso 2.* Presentar las nuevas entradas y las salidas deseadas. Sabiendo cual es la respuesta que se desea que aprenda el perceptrón para un vector de entrada dado  $x_0, x_1, \dots, x_{N-1}$ , se presenta este vector de entrada al perceptrón y se propaga hasta la salida  $d(t)$ .
- *Paso 3.* Calcular el error actual. Se calcula el error, restando la salida proveída por el perceptrón  $y(t)$  a la salida que se desea que aprenda  $d(t)$ .
- *Paso 4.* Adaptar pesos.  $w_i(t+1) = w_i(t) + [d(t)-y(t)]x_i(t)$ ,  $0 \leq i \leq N-1$

$$d(t) \begin{cases} +1 & \text{Si la entrada es de la clase A} \\ -1 & \text{Si la entrada es de la clase B} \end{cases} \quad (2.6)$$

Una vez que se tiene el valor del error, se hace el ajuste de cada uno de los pesos, tomando su valor actual y sumándole, algebraicamente el error multiplicado por su respectiva entrada y por un valor de ganancia  $\eta$ , llamado tasa de aprendizaje. Este procedimiento se repite para cada par entrada-salida que forman el conjunto inicial de información que el perceptrón aprenderá; este conjunto es llamado conjunto de entrenamiento. Es importante mencionar que los pesos no deben ser ajustados si el perceptrón realizó una salida correcta.

- *Paso 5.* Volver al paso 2.

El hecho de tomar cada par de entrada-salida, que forman el conjunto de entrenamiento y, aplicar este procedimiento, es llamado una época. Este es un método iterativo de tal forma que se repetirá durante tantas épocas como sea necesario, es decir, hasta que el error tienda a cero o sea menor de cierto valor establecido como criterio de paro.

#### *2.2.4 APPLICACIÓN DE UN PERCEPTRÓN SIMPLE A UN PROBLEMA DE CLASIFICACIÓN.*

Este problema es planteado de la siguiente forma: ¿Es el perceptrón capaz de realizar la multiplicación lógica de dos entradas?, es decir, si el perceptrón simple es capaz de simular una compuerta AND.

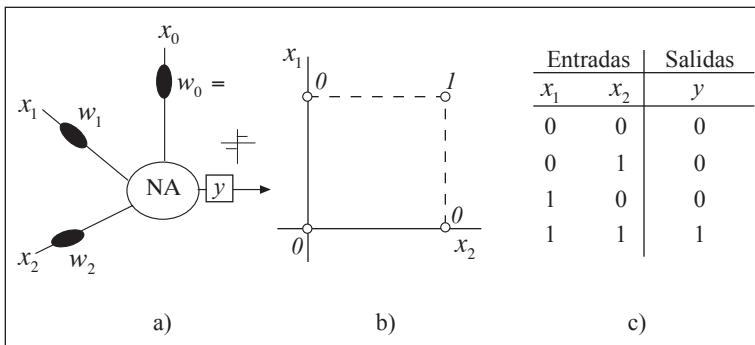


Figura 2.9 Perceptrón aplicado al problema de la multiplicación lógica.

a) Representación del modelo con sus entradas, salidas y pesos; b) Mapa de características; c) Tabla de entradas y salidas que el perceptrón debe aprender.

Una compuerta AND de dos entradas provee como resultado un 1 lógico sólo si las dos entradas son 1, de lo contrario, provee un cero lógico. La representación del problema utilizando un perceptrón, así como la tabla de entradas-salidas que éste debe aprender y su respectivo mapa de características para la AND lógica se muestra en la figura 2.9. Como se observa en dicha figura, ha sido añadida una entrada  $x_0$  con un valor de 1. Esta entrada tiene asociado un peso  $w_0$ , que es ajustado durante el proceso de aprendizaje al igual que los demás pesos, y hace la función del umbral que se ajusta de forma automática durante el entrenamiento.

En la siguiente tabla se muestra, a manera de resumen, el proceso de entrenamiento de un perceptrón simple para emular la compuerta AND lógica.

TABLA 2.1 ENTRENAMIENTO DE UNA NEURONA PARA RESOLVER EL PROBLEMA DE LA AND LÓGICA

	$x_0$	$x_1$	$x_2$	$w_0$	$w_1$	$w_2$	$NA$	$d(t)$	$y(t)$	$e(t)$	Observaciones
Época 1	1	0	0	1.5	0.5	1.5	1.5	0	1	-1	Ajuste de pesos
	1	0	1	0.5	0.5	1.5	2.0	0	1	-1	Ajuste de pesos
	1	1	0	-0.5	0.5	0.5	0.0	0	0	0	Los pesos permanecen
	1	1	1	-0.5	0.5	0.5	0.5	1	1	0	Los pesos permanecen
Época 2	1	0	0	-0.5	0.5	0.5	-0.5	0	0	0	Los pesos permanecen
	1	0	1	-0.5	0.5	0.5	0.0	0	0	0	Los pesos permanecen

Debido a que los ajustes de los pesos proveen una respuesta correcta para cada par de entrenamiento, podemos decir que el perceptrón ha aprendido a diferenciar entre las dos clases.

En la figura 2.10, se puede observar cómo, al ir realizando el algoritmo de aprendizaje, la función discriminante, que es una recta con pendiente  $-w_2/w_1$ , va moviéndose en el plano característico conforme se van ajustando los pesos hasta colocarse en una posición en la que separa las clases. Cabe mencionar que no es necesario llegar hasta un valor de cero en el error. Dependiendo de la aplicación, se puede fijar que la función discriminante pueda colocarse dentro de un rango dado. Si el entrenamiento se lleva a cabo hasta que el error sea cero, el perceptrón o la red se comportarán de forma completamente selectiva para el problema que fueron entrenados.

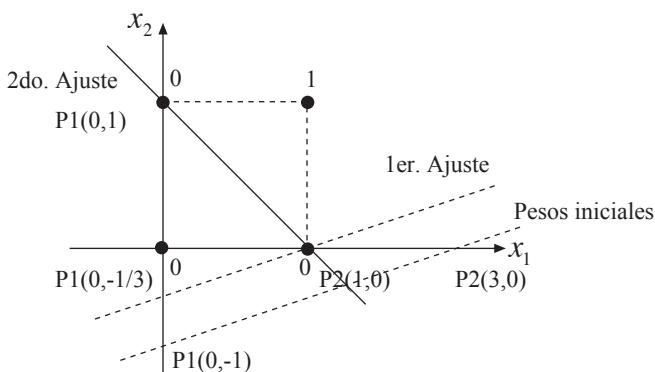


Figura 2.10 Función discriminante moviéndose durante el entrenamiento.

Otro aspecto importante es el hecho de que el entrenamiento de un perceptrón puede necesitar de varias épocas para el aprendizaje de un problema, por lo que en muchas ocasiones la experiencia del diseñador cobra una importancia relevante. Por ejemplo, el elegir una tasa de aprendizaje demasiado pequeña puede provocar que se necesiten un número de épocas muy elevado, o si por el contrario se elige una tasa muy grande, es posible que la función discriminante se mueva de manera muy brusca, ocasionando con ello que se pase por alto el intervalo de valores correctos para resolver el problema y, por ende, necesitar un número de iteraciones también muy elevado.

En este caso después de realizar dos ajustes a los pesos durante el entrenamiento, los resultados para los cuales la salida no produce error son los siguientes:

$$w_0 = -0.5, w_1 = 0.5, w_2 = 0.5$$

Entonces la función discriminante que separa las dos clases correctamente puede ser representada mediante la recta:

$$x_1 = -x_2 + 1$$

De la función discriminante obtenemos que es una recta con pendiente  $m = -1$  que pasa por los puntos  $P1(0,1)$  y  $P2(1,0)$ .

### *2.2.5 DISEÑO DE UN PERCEPTRÓN MEDIANTE DISPOSITIVOS ELECTRÓNICOS*

Partiendo de los resultados anteriores es posible llevar el modelo teórico a la construcción de un circuito con amplificadores operacionales. Como hemos visto hasta aquí, el comportamiento de

una neurona desde un enfoque general, puede representarse mediante operaciones matemáticas muy simples, como la suma, las multiplicaciones y la comparación.

Un dispositivo electrónico que realiza dichas operaciones sin ningún problema es el amplificador operacional (Coughlin y Driscoll, 1987) en donde cada operación depende únicamente de la configuración en la que se conecte el amplificador.

Sabemos que una neurona recibe la información mediante sus dendritas y multiplica cada uno de los valores de las entradas por una fuerza de unión (peso). Entonces podemos representar una entrada y multiplicarla por un factor mediante un amplificador inversor con una ganancia específica. El valor de la ganancia a lazo cerrado del amplificador, representa la fuerza de unión (sinápsis) de la neurona en sus entradas (dendritas). Este valor está determinado en el amplificador por:

$$\Delta V = -\frac{R_f}{R_i} \quad (2.7)$$

donde:

$\Delta V$  = Ganancia de voltaje a lazo cerrado del amplificador.

$R_f$  = Resistencia de retroalimentación.

$R_i$  = Resistencia de entrada.

El diagrama simbólico del amplificador inversor es el siguiente:

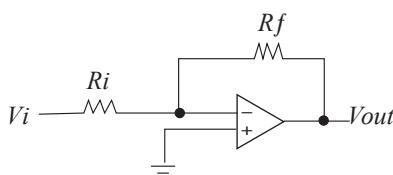


Figura 2.11 Amplificador inversor. Equivalente electrónico de la entrada (dendrita) ponderada por la fuerza de unión (sinápsis).

El procesamiento que realiza el cuerpo de una neurona es la suma de todas las entradas ponderadas. Esta operación podemos representarla mediante un sumador inversor (ver Figura 2.12), donde el voltaje de salida está dado por:

$$V_{out} = -[V_1 + V_2] \quad \text{si} \quad Rf = Ri = R_1 = R_2 \quad (2.8)$$

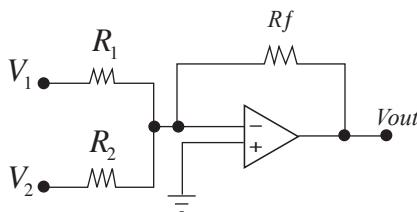


Figura 2.12 Amplificador sumador inversor.

Finalmente, el resultado de la unidad de procesamiento se compara con el umbral de la neurona. Si el resultado de la suma es mayor que el umbral, entonces la neurona emitirá una señal de salida, en caso contrario no lo hará. La representación puede realizarse mediante un comparador de nivel no inversor.

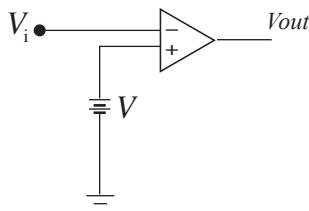


Figura 2.13 Comparador de nivel no inversor. Equivalente electrónico del umbral y la respuesta de una neurona artificial.

La suma de voltajes, que fue la respuesta obtenida en la configuración de la figura 2.12, entra al comparador representada por  $V_i$  en la figura 2.13.

La respuesta emitida por el comparador de nivel ( $V_{out}$ ) depende de la magnitud del voltaje  $V$ . Esta respuesta puede tener únicamente dos valores:  $V_{sat+}$  y  $V_{sat-}$ . Resumiendo lo dicho tenemos:

$$V_{out} = \begin{cases} V_{sat^+} & \text{si } Vi > V \\ V_{sat^-} & \text{si } Vi \leq V \end{cases} \quad (2.9)$$

donde:

$Vi$  representa la suma de los valores de entrada ponderadas.

$V$  representa el umbral de la neurona.

$V_{out} V_{sat+}$  representa el disparo de la neurona cuando es excitada.

$V_{sat-}$  representa la respuesta de la neurona cuando es inhibida.

Al realizar un circuito con las tres configuraciones anteriores tenemos una neurona artificial que realiza multiplicaciones lógicas.

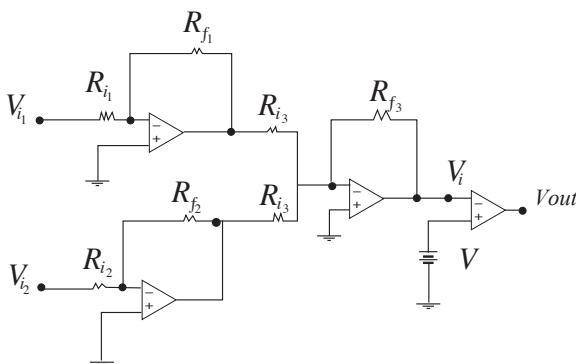


Figura 2.14 Neurona artificial con amplificadores operacionales.

## 2.3 REDES NEURONALES ARTIFICIALES

### 2.3.1 CLASES NO SEPARABLES POR UNA LÍNEA RECTA

Como se mencionó anteriormente, el perceptrón simple está muy limitado en cuanto a los problemas que puede resolver en el mundo real ya que la mayoría de los problemas de significación práctica son sistemas multiclasses. Un ejemplo en el cual se puede observar esta limitante del perceptrón simple es intentar resolver el problema de la operación lógica XOR (OR-exclusiva).

En la tabla de la figura 2.15 es posible ver cómo el par de entradas 0 0 y 1 1 producen como salida un cero lógico, mientras que el resto produce un 1 lógico. En el mapa de características de dicha figura se muestra que una línea recta no es suficiente para separar las clases y poder identificarlas.

Entradas		Salidas
$x_1$	$x_2$	$Y$
0	0	0
0	1	1
1	0	1
1	1	0

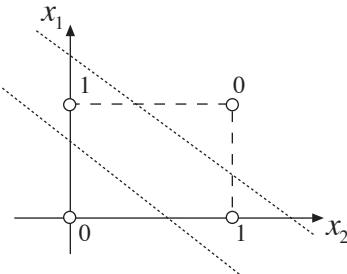


Figura 2.15 Operación lógica XOR. Izquierda: Tabla de entradas y salidas de la función lógica XOR. Derecha: Mapa de características representando la salida que debe de proveer cada par  $x_1, x_2$ .

Para resolver este problema se utiliza el recurso que se tiene del perceptrón que genera una línea recta separando en dos regiones el espacio característico y, un segundo perceptrón que genera otra línea recta, de tal forma que entre ambas dividan el espacio característico en tres regiones.

La salida de estos perceptrones se utiliza como entrada para un tercer perceptrón ubicado en la capa de salida. Este último integra las salidas de los anteriores y provee como respuesta un 1 lógico si el vector de entrada se encuentra dentro de la región formada por las dos líneas rectas y, provee un cero lógico si el vector de entrada se encuentra fuera de ellas.

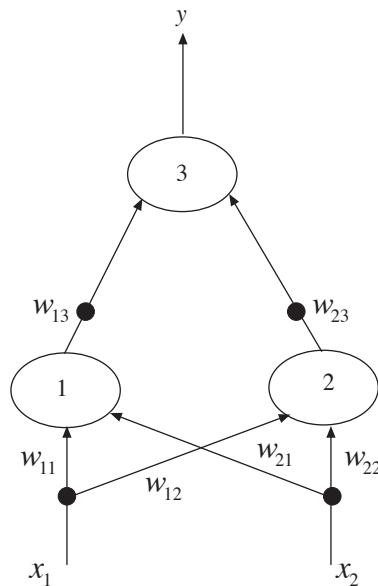


Figura 2.16 Perceptrón para la operación lógica XOR.

La configuración del digrama anterior es un sistema neuronal que puede resolver el problema de la XOR. Las neuronas 1 y 2 definen las dos líneas rectas que separan el espacio característico en tres regiones, mientras que la neurona 3, toma la respuesta de las dos anteriores y ofrece como respuesta una de dos clases, la que se encuentra dentro del espacio de las dos líneas y la que se encuentra fuera de él.

### *2.3.2 PERCEPTRONES MULTICAPA*

La solución a los problemas de reconocimiento de clases en un espacio característico, las cuales no son separables por una línea recta, y aun más, problemas de identificación multiclasas son los sistemas multineuronas o Redes Neuronales Artificiales (RNA). Aunque existen diferentes tipos de RNA en base a su estructura, como las de Hopfield, de Hamming, los clasificadores Carpenter-Grossberg, o las RNA de Kohonen (Rumelhart, Hinton, y Williams, 1986), en este trabajo nos enfocaremos al modelo perceptrón debido a su simplicidad y eficiencia en lo que a problemas de clasificación se refiere.

La RNA consta de una arquitectura genérica en la que las neuronas están dispuestas en forma de capas. La primera capa se denomina de entrada, posteriormente se encuentra la capa oculta y al final una capa de salida (Lippmann, 1987).

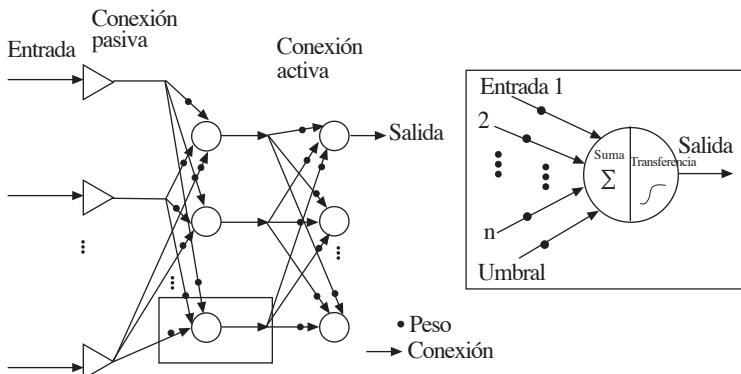


Figura 2.17 Modelo genérico del perceptrón multicapa.

En un perceptrón multicapa las entradas son conectadas sólo a las neuronas de la capa oculta donde, en éstas últimas, las funciones que definen el nivel de activación y la salida están dadas por:

$$NA_i = \sum_{i=1}^n x_i w_i \quad (2.10)$$

$$y = f(NA_i) = f\left(\sum_{i=1}^n x_i w_i\right) \quad (2.11)$$

Las neuronas de la última capa son de salida y proporcionan la respuesta de la red. El nivel de activación y la salida de estas neuronas son definidas mediante:

$$NA_j = \left( \sum_{j=1}^n x_{ij} w_{ij} \right) \quad (2.12)$$

$$y = f(NA_j) = f\left(\sum_{j=1}^n x_{ij} w_{ij}\right) \quad (2.13)$$

	Función	Rango	Gráfica
<b>Identidad</b>	$y = x$	$[-\infty, +\infty]$	
<b>Escalón</b>	$y = sign(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
<b>Lineal a tramos</b>	$y = \begin{cases} -1, & \text{si } x < -l \\ x, & \text{si } -l \leq x \leq l \\ +1, & \text{si } x > l \end{cases}$	$[-1, +1]$	
<b>Sigmoide</b>	$y = \frac{1}{1+e^{-x}}$ $y = \tanh(x)$	$[0, +1]$ $[-1, +1]$	
<b>Gaussiana</b>	$y = A e^{-\beta x^2}$	$[0, +1]$	
<b>Sinusoidal</b>	$y = A \sin(ax + \varphi)$	$[-1, +1]$	

Figura 2.18 Funciones de activación.

Anteriormente se mencionó que el potencial eléctrico de una neurona varía con respecto al tiempo, por lo que la salida generada, que es una función del nivel de activación, puede ser definida no sólo mediante una función escalón. Entre las funciones de activación más utilizadas podemos encontrar las mostradas en la figura 2.18.

Como observamos, el aprendizaje consiste en ajustar los pesos para que las salidas producidas por la red sean iguales o lo más parecidas posible a los valores deseados. Esto implica que el resultado deseado es minimizar el error producido por la red. Para ello es posible utilizar un procedimiento conocido como Algoritmo de Retroproyagación (Lippmann, 1987).

### *2.3.3 ALGORITMO DE ENTRENAMIENTO DE RETROPROPAGACIÓN*

Al hablar de redes de retropropagación o redes de propagación hacia atrás, se hace referencia a un algoritmo de aprendizaje mas que a una arquitectura determinada. Durante mucho tiempo no se dispuso de algoritmos para entrenar redes multicapa, y como las redes de una capa estaban muy limitadas en cuanto a lo que eran capaces de representar, el campo de las RNA estaba estancado. La invención y perfeccionamiento del algoritmo de retropropagación dio un gran impulso al desarrollo de este campo, ya que tiene un buen fundamento matemático que ha expandido enormemente el rango de problemas donde se aplican las RNA, sobre todo aquellos en los que la relación entre entradas y salidas es no lineal y los datos de entrenamiento son abundantes.

La retropropagación consiste en diseminar el error hacia atrás, es decir, de la capa de salida hacia la capa de entrada, pasando por las capas ocultas intermedias y ajustando los pesos de las conexiones con el fin de reducir dicho error.

En la figura 2.19, se muestra que el error es una función de los pesos de la red los cuales deben ser ajustados. El proceso de ajuste de los pesos en la dirección que se muestra en dicha figura es llamado descendencia de gradiente.

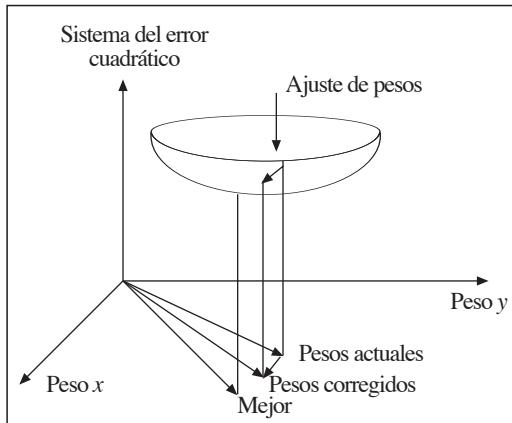


Figura 2.19 Descendencia de gradiente.

El error total que la red produce se puede medir en las neuronas de salida (Werbos, 1990), y está definido por:

$$E = \frac{(d_j - y_j)^2}{2} \quad (2.14)$$

Para reducir el error es necesario modificar los pesos que se conectan a las neuronas de salida de la red. Pero, al modificar estos pesos, se obliga a que las salidas de las neuronas de la capa oculta desarrollen una respuesta distinta de la que produjeron. Además, se debe tomar en cuenta que las salidas de estas neuronas ocultas dependen del valor de los pesos que conectan las entradas de la red con éstas. Por lo que estos pesos deben ser también modificados. Entonces, al reducir el error en las neuronas de la capa de salida de la red, se debe reducir el error en la capa de

las neuronas de entrada. El error que cada neurona de la capa oculta produce, se obtiene al calcular la modificación de los pesos para reducir el error de la capa de salida. Por lo tanto, el error es propagado de manera distribuida desde la capa de salida hasta la capa de entrada. Debido a esta propagación hacia atrás del error es que éste es llamado algoritmo de retropropagación del error o simplemente retropropagación.

A continuación se describen los pasos que integran el algoritmo de retropropagación:

- *Paso 1.* Inicializar los pesos. Ajustar todos los pesos a un valor aleatorio pequeño.
- *Paso 2.* Presentar las entradas y las salidas deseadas. Presentar el vector de entrada  $x_1, x_2, \dots, x_n$  y especificar las salidas deseadas  $d_1, d_2, \dots, d_n$ .
- *Paso 3.* Calcular la salida que genera la red.  $y_j(t) = f[\sum w_{ij}(t) y_i(t)]$ , donde la salida es una función continua y monótonamente creciente.
- *Paso 4.* Ajuste de los pesos. a) Si se trata de un peso de una sinápsis que se conecta con una neurona de la capa de salida, actualizar el peso utilizando la siguiente expresión:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(d_j - y_j)f'(NA_j)y_i \quad (2.15)$$

donde:

$w_{ij}(t+1)$  es el nuevo valor del peso.

$w_{ij}(t)$  es el valor del peso actual.

$d_j$  es el valor deseado para la salida de la neurona  $j$ .

$y_i$  es el valor de la salida en la neurona  $j$  antes de actualizar el peso.

$\alpha$  es la tasa de aprendizaje.

$f'(NA_j)$  es la derivada de la función de salida de la neurona  $j$ .  
 $y_i$  es el valor de salida de la neurona  $i$  de la capa anterior, de cuya conexión se está ajustando el peso.

b) Si se trata de un peso de alguna sinapsis que se conecta a una neurona de la capa oculta; actualizar el peso utilizando la siguiente expresión:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha f'(NA_j) x_i \sum[(d_j - y_j) f'(NA_j) w_j] \quad (2.16)$$

donde:

$w_{ij}(t+1)$  es el nuevo valor del peso  $w_{ij}$ .

$w_{ij}(t)$  es el valor actual del peso  $w_{ij}$ .

$\alpha$  es la tasa de aprendizaje.

$f'(NA_i)$  es la derivada de la función de salida de la neurona  $i$ .

$x_i$  es la entrada a la sinápsis de la neurona  $i$ .

$d_j$  es el valor deseado para la salida de cada una de las neuronas  $j$ .

$y_j$  es el valor obtenido en las neuronas  $j$ .

$f(NA_j)$  es la función de salida de las neuronas  $j$ .

$w_i$  es el valor del peso de las neuronas  $j$ .

- *Paso 5.* Volver al paso 2.

Este algoritmo ha sido probado con un gran cantidad de problemas tales como el de la XOR (Rosenblatt, *Principles of Neurodynamics*, 1959), reconocimiento de voz (Sejnowski y Rosenberg, 1986), (Pelling, Moore, y Tomlinson, 1986), (Elman, 1987) y reconocimiento de imágenes (Rosenblatt, *Two Theorems of Statistical Separability in Perceptron*, 1958). Una demostración del poder de este algoritmo fue realizada por Sejnowski (Sejnowski & Rosenberg, 1986), en la cual entrenó un perceptrón con 120 neuronas en la capa oculta y más de 20 000 pesos para transformar

un mapa de palabras inglesas en texto por medio de los fonemas correspondientes.

A continuación se muestra un ejemplo de los valores de los pesos obtenidos al aplicar el algoritmo de aprendizaje para que una RNA pueda resolver el problema de la XOR:

TABLA 2.2 PESOS OBTENIDOS DURANTE EL ENTRENAMIENTO DE UNA RNA PARA RESOLVER EL PROBLEMA DE LA XOR

$w_{13}=6.103170$	$w_{24}= 4.148270$
$w_{14}=6.111240$	$w_{35}= 8.401280$
$w_{23}=6.111240$	$w_{45}= -9.068840$

Para los que el valor de los umbral de las neuronas de la red son:

$$\theta_3 = -2.63774$$

$$\theta_4 = -6.36146$$

$$\theta_5 = -3.83855$$

Si sustituimos cada uno de los valores de la red en las ecuaciones que representan el nivel de activación para cada uno de los perceptrones, es posible obtener la gráfica de sus funciones discriminantes y dibujarlas en un espacio característico.

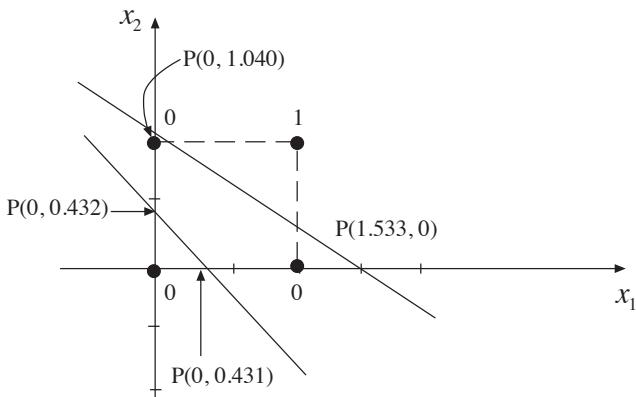


Figura 2.20 Funciones discriminantes para el problema de la XOR.

### 2.3.4. DISEÑO DE UNA RED NEURONAL PARA EL PROBLEMA DE LA XOR LÓGICA MEDIANTE DISPOSITIVOS ELECTRÓNICOS

El análisis para construir una RNA parte de la premisa siguiente: “Una red neuronal es un arreglo de nodos o unidades procesadoras simples, las cuales representan el cuerpo de las neuronas. Las unidades se encuentran conectadas entre sí y actúan como los axones y las dendritas. Las conexiones entran a las unidades procesadoras multiplicadas por un factor, el cual es un valor analógico que representa la fuerza de conexión o sinápsis” (Rumelhart, Hinton, y Williams, 1986), (Lippmann, 1987). Esto nos lleva a pensar que al igual que en el modelo matemático, si conectamos tres perceptrones simples para resolver el problema de la XOR, podemos reproducir tres neuronas artificiales construidas con amplificadores operacionales.

Un aspecto muy importante en la construcción de la RNA con amplificadores operacionales (ver Figura 2.21) es el ajuste de los valores que corresponden a cada una de las sinapsis de la red. Para realizar este ajuste es posible utilizar resistencias variables de  $10\text{ k}\Omega$  como valor máximo, reduciendo así el valor de las sinapsis a 10 % de su valor original. Esta pequeña modificación no altera en forma alguna el comportamiento de la red y presenta la ventaja de poder utilizar una mayor escala en los voltajes de entrada. Por lo tanto, los valores de las resistencias pueden ser:

TABLA 2.3 VALORES DE RESISTENCIAS PARA EL PERCEPTRÓN MULTICAPA ELECTRÓNICO

<b>Resistencia</b>	<b>Valor (<math>\text{k}\Omega</math>)</b>	<b>Resistencia</b>	<b>Valor (<math>\text{k}\Omega</math>)</b>	<b>Resistencia</b>	<b>Valor (<math>\text{k}\Omega</math>)</b>
R1	10k $\Omega$	R11	10k $\Omega$	R21	10k $\Omega$
R2	10k $\Omega$	R12	4.148k $\Omega$	R22	10k $\Omega$
R3	6.103k $\Omega$	R13	10k $\Omega$	R23	10k $\Omega$
R4	6.111k $\Omega$	R14	10k $\Omega$	R24	110k $\Omega$
R5	10k $\Omega$	R15	10k $\Omega$	R25	10k $\Omega$
R6	10k $\Omega$	R16	10k $\Omega$	R26	10k $\Omega$
R7	10k $\Omega$	R17	10k $\Omega$		
R8	10k $\Omega$	R18	8.401k $\Omega$		
R9	10k $\Omega$	R19	10k $\Omega$		
R10	6.111k $\Omega$	R20	9.068k $\Omega$		

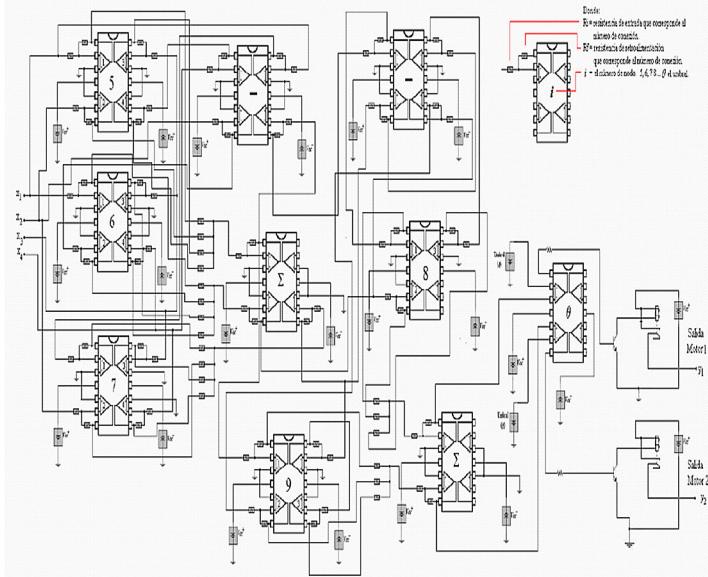


Figura 2.21 Diseño de la RNA con amplificaciones operacionales.



## CAPÍTULO 3. SISTEMAS NEURODIFUSOS

### 3.1 ANTECEDENTES DE LOS SISTEMAS NEURODIFUSOS

El término *sistemas neurodifusos* se refiere a la combinación de las *redes neuronales* y de los *sistemas difusos* y, la idea de efectuar esta combinación es aprovechar la capacidad de aprendizaje de las primeras y la estructura basada en reglas de los modelos difusos (Ruspini, Bonissone & Pedrycz, Handbook of Fuzzy Computation, 1998), (Von Altrock, Neurofuzzy Technologies, 2000).

Las técnicas difusas han sido reconocidas como una herramienta poderosa para obtener modelos de sistemas que no son fáciles de modelar convencionalmente debido a: la falta de conocimiento preciso acerca del mismo, las características fuertemente no lineales o las variantes en el tiempo. La naturaleza de los modelos difusos basados en reglas permiten utilizar la información expresada en oraciones o frases de lenguaje natural, lo cual hace que el modelo sea accesible para la interpretación y el análisis cuantitativo. A nivel computacional, los modelos difusos pueden interpretarse como estructuras matemáticas flexibles, similares a

las redes neuronales, que pueden aproximar una larga clase de sistemas no lineales y con un grado de precisión deseado (Wang, Adaptive Fuzzy Systems and Control, Design and Stability Analysis, 1994).

En el diseño de sistemas de control basados en el modelo, el modelado y la identificación del proceso puede acaparar la mayor parte del tiempo y elevar los costos (Richalet, 1996). Para identificar efectivamente con sistemas no lineales, se requiere una estructura del modelo matemáticamente tratable y técnicas de estimación robustas. Un requisito natural es que el modelo no solamente logre predecir las salidas del sistema con suficiente precisión sino que también, proporcione intuiciones relativas al funcionamiento del mismo.

Un modelo no sólo puede emplearse para una situación dada sino que puede adaptarse a cambios en los parámetros de diseño y condiciones de operación. El empleo de modelos lineales (para sistemas lineales o casi lineales), hace posible relacionar o unir el entorno matemático con el significado físico de la estructura y los parámetros del modelo. Con métodos no lineales estos dos tópicos llegan a ser mutuamente exclusivos.

Por otra parte, las aproximaciones del tipo “*caja negra*” basados en redes neuronales, *wavelets* o funciones de base radial, proporcionan modelos no lineales bastante generales con capacidades de aproximación satisfactorias así como técnicas de estimación paramétrica eficientes. Sin embargo, la estructura y los parámetros de éstos raramente tienen algún significado físico. Dichos modelos no pueden ser utilizados para analizar el comportamiento de los sistemas, no pueden ser modificados cuando se cambia de una escala del proceso a otro y, en consecuencia, no proveen soluciones convenientes para la práctica industrial.

Las aproximaciones con modelos del tipo “*caja blanca*” basados en las leyes de los elementos y principios de conservación, nos llevan a modelos con buena interpretación física y propiedades de escalamiento, desafortunadamente, la derivación de éstos es tediosa, consumen demasiado tiempo y encarecen el proceso. La mayoría de los parámetros no pueden ser estimados a partir de datos y la incertidumbre significativa en dichos parámetros y en los niveles de estructura, no permiten que éstos puedan utilizarse para predicciones precisas.

Existe una gama de técnicas de modelado que combinan las ventajas de los dos tipos de aproximaciones antes descritas, de tal modo que las partes conocidas del sistema se modelan utilizando los principios físicos, en tanto que las partes desconocidas o menos seguras, se aproximan con cajas negras. Estos métodos son conocidos como modelos semifísicos o del tipo “*caja gris*”. Los modelos difusos exhiben algunas propiedades de las técnicas basadas en modelos del tipo “*caja gris*” y su construcción envuelve métodos basados en lógica difusa y razonamiento aproximado, pero también pueden originarse en las áreas de las redes neuronales, análisis de datos y la identificación convencional de sistemas.

Si no existe un conocimiento *a priori* para obtener un modelo difuso, las reglas y las funciones de pertenencia pueden extraerse a partir de datos. Varias técnicas han sido propuestas en la literatura, tales como *fuzzy clustering*, métodos de aprendizaje neuronal, mínimos cuadrados ortogonales, aprendizaje inductivo o razonamiento evidencial y la identificación neurodifusa. La selección de una técnica en particular depende de la información inicial disponible.

Una variedad de esquemas de control utilizan modelos difusos, ejemplos de algunos de ellos lo constituyen el control inverso, control con modelo interno y el control predictivo basado en modelos difusos (Babuška y Verbruggen , 1996). Estos modelos pueden extenderse mediante el uso de modelos neurodifusos.

El modelado de sistemas basado en principios físicos, el cual se conoce como modelado por conocimiento, en ocasiones resulta muy complejo y su grado de dificultad depende del proceso a considerar. Por el contrario, un sistema de inferencia difusa emplea reglas difusas *if-then* que pueden modelar los aspectos cualitativos del conocimiento humano y los procesos de razonamiento sin requerir de un análisis cuantitativo preciso. Este modelado o identificación difusa, inicialmente explorada por Takagi y Sugeno (Takagi & Sugeno, 1989), ha encontrado numerosas aplicaciones prácticas en las áreas de control, predicción e inferencias. Sin embargo, algunos aspectos fundamentales que atanen a estas aproximaciones son específicamente los siguientes:

- No existen métodos estándares para transformar el conocimiento o la experiencia humana en una base de reglas y datos de un sistema de inferencia difusa.
- Existe la necesidad de aplicar métodos efectivos para sintonizar las funciones de pertenencia (MF) de tal modo que se pueda minimizar el error de medición de la salida u optimizar un índice de desempeño.

Bajo esta perspectiva, surge la arquitectura llamada ANFIS que puede servir de base para construir un conjunto de reglas *if-then* con funciones de pertenencia apropiadas para generar los pares entrada-salida deseados, (Jang, 1993) y (Jang, 1995).

### 3.2 REDES ADAPTABLES, ARQUITECTURA Y ALGORITMO BÁSICO DE APRENDIZAJE

Una red adaptable es un superconjunto de todas las clases de redes neuronales prealimentadas con capacidad de aprendizaje supervisado. Como su nombre lo indica, es una red cuya estructura consiste de nodos y ligas direccionales mediante las cuales dichos nodos son conectados y una parte de éstos son adaptables, lo que significa que sus salidas dependen de los parámetros asociados a ellos y, las reglas de aprendizaje especifican cómo pueden ser cambiados los parámetros a fin de minimizar un error de medición prescrito.

La regla básica de aprendizaje de las redes adaptables está basada en el gradiente descendente y la regla de la cadena propuesta por Werbos en los años setenta (Werbos, *Beyond regression: New tools for prediction and analysis in the behavioral sciences*, 1974).

Una red adaptable es una red multicapa prealimentada, en la cual cada nodo desempeña una función particular (función nodal) sobre las señales que llegan a él y posee un conjunto de parámetros pertenecientes al mismo. Las fórmulas para las funciones nodales pueden variar de nodo a nodo, y la selección de cada función nodal depende de la función entrada-salida que la red adaptable debe llevar a cabo. Las ligas en una red ANFIS solamente indican la dirección del flujo de señales entre los nodos, no hay pesos asociados con estas ligas. Para denotar la diferencia de capacidades en una red adaptable se utilizan nodos cuadrados y circulares. Un nodo cuadrado (nodo adaptable) tiene parámetros asociados en tanto que un nodo circular (nodo fijo) no los tiene. La figura 1.3 muestra la arquitectura de una red ANFIS.

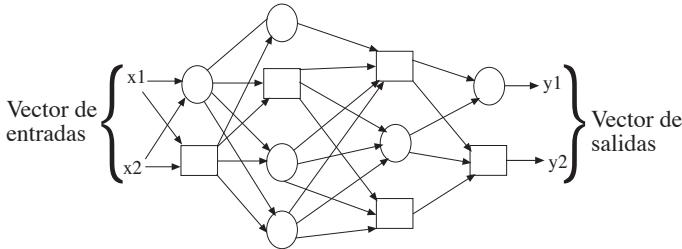


Figura 3.1 Arquitectura de una red ANFIS.

El conjunto de parámetros de una red adaptable es la unión de los conjuntos de parámetros de cada uno de sus nodos. De acuerdo con lo anterior, para lograr un mapeo entrada-salida deseado, estos parámetros se actualizan con base en los datos de entrenamiento proporcionados y en el procedimiento de aprendizaje basado en el gradiente, el cual se describe a continuación.

Supongamos que una red adaptable posee  $L$  capas y que la capa  $l$  ( $l=0,1,\dots,L$ ;  $l=0$  representa la capa de entrada) tiene  $N(l)$  nodos. Bajo estas consideraciones la salida y la función del nodo  $i$  ( $i=1,\dots,N(l)$ ) de la capa  $l$  pueden representarse como  $x_{b,i}$  y  $f_{b,i}$  respectivamente. Sin pérdida de generalidad, se asume que no hay saltos en las ligas, esto es, las ligas no conectan capas no consecutivas.

Dado que la salida de un nodo depende de las señales que llegan a él y del conjunto de parámetros asociados, es posible emplear la siguiente expresión general para la función del nodo  $f_{b,i}$ :

$$x_{l,i} = f_{l,i}(x_{l-1,1}, \dots, x_{l-1,N(l-1)}, \alpha, \beta, \gamma, \dots) \quad (3.1)$$

donde  $\alpha, \beta, \gamma$ , etcétera, son los parámetros pertenecientes a este nodo.

Suponiendo que el conjunto de datos de entrenamiento tiene  $P$  elementos, es posible definir un criterio de error (o función de energía) para el  $p$ -ésimo ( $1 \leq p \leq P$ ) elemento de entrada de los datos de entrenamiento como la suma de errores cuadráticos de las salidas:

$$E_p = \sum_{k=1}^{N(L)} (d_k - x_{L,k})^2 \quad (3.2)$$

donde  $d_k$  es el  $k$ -ésimo componente del  $p$ -ésimo vector de salida deseado, y  $x_{L,k}$  es el  $k$ -ésimo componente del vector de salida real producido por la presentación del  $p$ -ésimo vector de entrada. En consecuencia, el error de medición total es:

$$E = \sum_{p=1}^P E_p \quad (3.3)$$

Para desarrollar un procedimiento de aprendizaje basado en el gradiente descendente en  $E$  sobre el espacio de parámetros, primeramente debemos calcular la variación del error o señal de error como la derivada del error de medición con respecto a la salida del nodo  $i$  en la capa  $l$ . En símbolos se escribe:

$$\varepsilon_{l,i} = \frac{\partial^+ E_p}{\partial x_{l,i}} \quad (3.4)$$

La expresión anterior se conoce como derivada ordenada. Su diferencia con la derivada ordinaria radica en la forma en que visualizamos la función que vamos a diferenciar. Para la salida de un nodo interno  $x_{l,i}$  (donde  $l \neq L$ ), la derivada parcial  $\partial E_p / \partial x_{l,i}$

es igual a cero dado que  $E_p$  no depende directamente de  $x_{l,i}$ . Sin embargo, es obvio que  $E_p$  depende de  $x_{l,i}$  indirectamente, dado que un cambio en  $x_{l,i}$  se propagará a través de caminos indirectos a la capa de salida y esto produce un cambio correspondiente en el valor de  $E_p$ . Por ello,  $\epsilon_{l,i}$  puede verse como la razón o proporción de esos dos cambios cuando se hacen infinitesimales.

La señal de error para la salida del  $i$ -ésimo nodo (en la capa  $L$ ) puede calcularse directamente:

$$\epsilon_{L,i} = \frac{\partial^+ E_p}{\partial x_{L,i}} = \frac{\partial E_p}{\partial x_{L,i}} = -2(d_i - x_{L,i}) \quad (3.5)$$

si  $E_p$  está definida como en (3.2). Para los nodos internos de la  $i$ -ésima posición de la capa  $l$ , la señal de error puede obtenerse mediante la regla de la cadena:

$$\epsilon_{l,i} = \frac{\partial^+ E_p}{\partial x_{l,i}} = \sum_{m=1}^{N(l+1)} \frac{\partial^+ E_p}{\partial x_{l+1,m}} \times \frac{\partial f_{l+1,m}}{\partial x_{l,i}} = \sum_{m=1}^{N(l+1)} \epsilon_{l+1,m} \times \frac{\partial f_{l+1,m}}{\partial x_{l,i}} \quad (3.6)$$

donde  $0 \leq l \leq L-1$ . Esto es, la variación del error en un nodo interno puede expresarse como la combinación lineal de las variaciones de error de los nodos en la siguiente capa. Además, las dos últimas expresiones pueden emplearse para toda  $0 \leq l \leq L$  y  $1 \leq i \leq N(l)$ . Puesto que las señales de error son obtenidas secuencialmente desde la capa de salida hasta la de entrada, este modelo de aprendizaje se denomina: Regla de Aprendizaje por Retropropagación, (Rumelhart, Hinton y Williams, ,1986).

El vector gradiente se define como la derivada del error de medición con respecto a cada parámetro y se aplica nuevamente la regla de la cadena para encontrarlo. Si  $\alpha$  es un parámetro del  $i$ -ésimo nodo en la capa  $l$ , tenemos que:

$$\frac{\partial^+ E_p}{\partial \alpha} = \frac{\partial^+ E_p}{\partial x_{l,i}} \times \frac{\partial f_{l,i}}{\partial \alpha} = \varepsilon_{l,i} \times \frac{\partial f_{l,i}}{\partial \alpha} \quad (3.7)$$

Si el parámetro  $\alpha$  es común en los diferentes nodos, entonces la ecuación (3.7) puede rescribirse en una forma más general

$$\frac{\partial^+ E_p}{\partial \alpha} = \sum_{x^* \in S}^n \frac{\partial^+ E_p}{\partial x^*} \times \frac{\partial f^*}{\partial \alpha} \quad (3.8)$$

donde  $S$  es el conjunto de nodos que contienen a  $\alpha$  como parámetro y  $f^*$  es la función nodal para calcular  $x^*$ . Entonces, la derivada del error total de medición con respecto a  $\alpha$  es:

$$\frac{\partial^+ E}{\partial \alpha} = \sum_{p=1}^P \frac{\partial^+ E_p}{\partial \alpha} \quad (3.9)$$

La fórmula de actualización para el parámetro genérico  $\alpha$  es:

$$\Delta \alpha = -\eta \frac{\partial^+ E}{\partial \alpha} \quad (3.10)$$

en la cual  $\eta$  es la velocidad de aprendizaje que puede ser expresada mediante:

$$\eta = \frac{k}{\sqrt{\sum_\alpha \left( \frac{\partial^+ E}{\partial \alpha} \right)^2}} \quad (3.11)$$

donde  $k$  es el tamaño del paso, es decir, la longitud de cada transición gradiente en el espacio de parámetros.

### 3.3 SISTEMAS DE INFERENCIA DIFUSA BASADOS EN REDES ADAPTABLES

Funcionalmente no hay muchas restricciones en cuanto a las funciones nodales de una red adaptable, excepto la diferenciabilidad por partes. Estructuralmente, la única limitación de la configuración de la red es que debe ser de tipo prealimentada. Salvo estas dos restricciones mínimas, las aplicaciones de las redes adaptables son varias en diversas áreas. Una clase de redes adaptables que funcionalmente son equivalentes a los sistemas de inferencia difusa, es la arquitectura conocida como ANFIS la cual se estudia en este capítulo.

Para efectuar el análisis de esta estructura se supondrá, por simplicidad, que el sistema de inferencia difuso en consideración tiene dos entradas  $x$  y  $y$ , una salida única  $z$  y que posee la arquitectura mostrada en la figura 3.1. También supone que la base de reglas contiene dos sentencias difusas *if-then* del tipo Takagi y Sugeno.

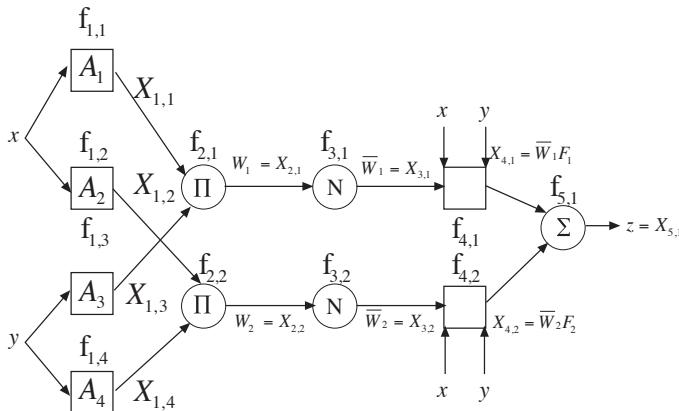


Figura 3.2 Red de ANFIS de dos entradas.

Las reglas a emplear y las funciones nodales se describen a continuación:

*Regla 1.* If  $x$  is  $A_1$  and  $y$  is  $B_1$  then  $F_1 = p_1x + q_1y + r_1$

*Regla 2.* If  $x$  is  $A_2$  and  $y$  is  $B_2$  then  $F_2 = p_2x + q_2y + r_2$ .

*Capa 1.* Cada nodo  $i$  en esta capa es un nodo cuadrado (adaptable) cuya salida está dada por:

$$\begin{aligned} x_{l,i} &= f_{1,i} = \mu_{A_i}(x) && \text{para } i = 1, 2 \\ x_{l,i} &= f_{1,i-2} = \mu_{B_{i-2}}(x) && \text{para } i = 3, 3 \end{aligned} \quad (3.12)$$

donde  $x$  (o  $y$ ) es la entrada al nodo  $i$ , y  $A_i$  (o  $B_{i-2}$ ) es la etiqueta lingüística (pequeño, largo, etcétera) asociada con dicha función nodal. Usualmente, se eligen las funciones de pertenencia con forma de campana con un máximo igual a 1 y un mínimo de cero, por ejemplo,  $A_i$  puede estar caracterizada por la función campana generalizada:

$$\mu_{A_i}(x) = \frac{1}{1 + [(\frac{x - c_i}{a_i})^2]^{b_i}} \quad (3.13)$$

donde  $\{a_i, b_i, c_i\}$  es el conjunto de parámetros. La figura 3.3 muestra que cada uno de estos parámetros tiene un significado físico:  $c$  determina el centro de la función correspondiente de pertenencia,  $a$  es la mitad de su anchura y  $b$  (junto con  $a$ ) controla la pendiente en el punto donde el valor de MF es 0.5 y cuando los valores de estos parámetros cambian, las funciones con forma de campana también lo hacen. A los parámetros involucrados en esta capa se les conoce como parámetros de la premisa. Cabe señalar que en esta capa también pueden emplearse otro tipo de funciones continuas o diferenciables por tramos como las trapezoidales y las triangulares, respectivamente.

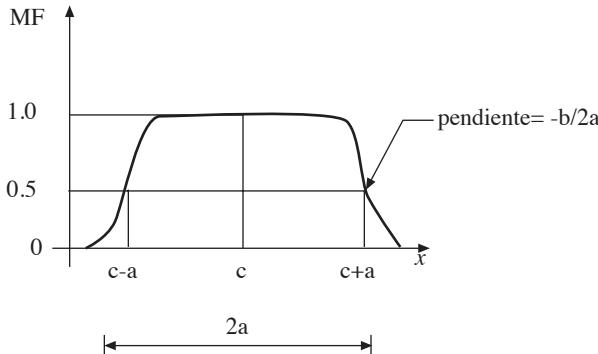


Figura 3.3 Función de pertenencia tipo campana generalizada.

*Capa 2.* Cada nodo de esta capa es un nodo circular etiquetado con la letra griega  $\Pi$ , cuya función es multiplicar las señales que llegan a él y enviar el producto a la salida. Para nuestro caso, tenemos:

$$x_{2,i} = w_i = \mu_{A_i}(x) \times \mu_{B_i}(y), i = 1, 2. \quad (3.14)$$

donde la salida de cada nodo representa la fuerza de disparo de una regla.

*Capa 3.* Cada nodo en esta capa es del tipo circular y está etiquetado con la letra  $N$ . Aquí, el  $i$ -ésimo nodo calcula la relación de la fuerza de disparo (o peso) de la  $i$ -ésima regla respecto a la suma de las fuerzas de disparo de todas las reglas y su salida está dada por:

$$x_{3,i} = \bar{w}_i = \frac{x_{2,i}}{w_1 + w_2}, i = 1, 2. \quad (3.15)$$

Por conveniencia, las salidas de esta capa se denominan **fuerzas de disparo normalizadas**.

*Capa 4.* Cada nodo  $i$  en esta capa es cuadrado y con función nodal

$$x_{4,i} = \bar{w}_i F_i, \quad i = 1, 2. \quad (3.16)$$

en nuestro caso dado que  $F_i = p_i x + q_i y + r_i$ , resulta obvio que podemos escribir:

$$x_{4,i} = \bar{w}_i (p_i, q_i, r_i, x, y), \quad i = 1, 2. \quad (3.17)$$

donde  $\bar{w}_i$  es la salida de la capa 3 y  $\{p_i, q_i, r_i\}$  es el conjunto de parámetros. Los parámetros de esta capa son conocidos como parámetros consecuentes.

*Capa 5.* El nodo individual de esta capa es circular y está etiquetado con la letra  $\Sigma$  y calcula la salida total como la suma de todas las señales que llegan a él, esto es

$$x_{5,1} = \sum_i \bar{w}_i f_{4,i} \sum x_{4,i}, \quad i = 1, 2. \quad (3.18)$$

Con esto, hemos construido una red adaptable que funcionalmente es equivalente a un sistema de inferencia difuso tipo 3.

### 3.4 ALGORITMO DE APRENDIZAJE HÍBRIDO

En la arquitectura ANFIS de la figura 3.1, se observa que cuando se fijan los valores de los parámetros de la premisa, la salida total puede expresarse como la combinación lineal de los parámetros consecuentes. En símbolos, la salida puede expresarse como:

$$z = x_{5,1} = x_{3,1} (p_1 x + q_1 y + r_1) + x_{3,2} (p_2 x + q_2 y + r_2)$$

o bien;

$$\begin{aligned} z = x_{5,1} &= p_1(xx_{3,1}) + q_1(yx_{3,1}) + r_1 x_{3,1} + \\ &p_2(xx_{3,2})q_2(yx_{3,2}) + r_2 x_{3,2} \end{aligned} \quad (3.19)$$

La red ANFIS en consideración posee solo una salida, esto es:

$$\text{salida} = F(\vec{I}, S) \quad (3.20)$$

donde  $\vec{I}$  es el vector de variables de entrada y  $S$  es el conjunto de parámetros. Si existe una función  $H$ , tal que la función compuesta  $H \circ F$  es lineal en algunos de los elementos de  $S$ , entonces estos elementos pueden identificarse mediante el método de mínimos cuadrados. Si el conjunto de parámetros  $S$  puede descomponerse como:

$$S = S_1 \oplus S_2 \quad (3.21)$$

donde  $\oplus$  representa la suma directa tal que  $H \circ F$  sea lineal en los elementos de  $S_2$ , entonces al aplicar  $H$  a la expresión para la salida, tenemos:

$$H(\text{salida}) = H \circ F(\vec{I}, S) \quad (3.22)$$

que es lineal en los elementos de  $S_2$ . Con los valores de  $S_1$  dados y tomando  $P$  datos de entrenamiento en la última expresión se obtiene una ecuación matricial de la forma:

$$A\theta = B \quad (3.23)$$

donde  $\theta$  es un vector desconocido cuyos elementos son parámetros en  $S_2$ . Esta ecuación representa el problema de mínimos cuadrados lineales y la mejor solución para  $\theta$ , que minimice  $\|A\theta - B\|^2$ , es el estimador de mínimos cuadrados  $\theta^*$ :

$$\theta^* = (A^T A)^{-1} A^T B \quad (3.24)$$

donde  $A^T$  es la matriz transpuesta de  $A$  y  $(A^T A)^{-1} A^T$  es la pseudoinversa de  $A$  si  $A^T A$  es no singular.

Particularmente tenemos que en nuestro caso, al considerar  $P$  datos de entrada, tenemos que:

$$A = \begin{bmatrix} (xx_{3,1})_1 & (yx_{3,1})_1 & (x_{3,1})_1 & (xx_{3,2})_1 & (yx_{3,2})_1 & (x_{3,2})_1 \\ (xx_{3,1})_2 & (yx_{3,1})_2 & (x_{3,1})_2 & (xx_{3,2})_2 & (yx_{3,2})_2 & (x_{3,2})_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ (xx_{3,1})_P & (yx_{3,1})_P & (x_{3,1})_P & (xx_{3,2})_P & (yx_{3,2})_P & (x_{3,2})_P \end{bmatrix} \quad (3.25)$$

$$B = \begin{bmatrix} f_{d_1} \\ f_{d_2} \\ f_{d_p} \end{bmatrix} \quad (3.26)$$

$$\theta = [p_1 \quad q_1 \quad r_1 \quad p_2 \quad q_2 \quad r_2]^T \quad (3.27)$$

Otra opción es hacer uso de las fórmulas de mínimos cuadrados recursivos (Jang & Sun, Neuro Fuzzy Modeling and Control, 1995):

$$\theta_{i+1} = \theta_i + S_{i+1} a_{i+1} (b_{i+1}^T - a_{i+1}^T \theta_i) \quad (3.28)$$

$$S_{i+1} = S_i - \frac{S_i a_{i+1} a_{i+1}^T S_i}{1 + a_{i+1}^T S_i a_{i+1}}, i = 0, 1, \dots, P-1 \quad (3.29)$$

donde:

$$\theta^* = \theta_P$$

$a_{i+1}^T$  =  $i$ -ésimo vector fila de la matriz  $A$

$b_{i+1}^T$  =  $i$ -ésimo elemento de  $B$

$S_{i+1}$  =  $i$ -ésima matriz de covarianza

Las condiciones necesarias para iniciar el algoritmo son:

$$\theta_0 = 0$$

$S_0 = \gamma I$ , donde  $\gamma$  es un valor positivo e  $I$  es la matriz de identidad de dimensión  $M \times M$ .

De acuerdo con lo anterior es posible combinar el método del gradiente y el estimador de mínimos cuadrados para actualizar los parámetros en una red adaptable. En el aprendizaje híbrido a ser aplicado en el modo fuera de línea, cada iteración está compuesta de un paso hacia delante y otro hacia atrás. En el paso hacia delante, cuando un vector de entrada se ha presentado, se calculan las salidas de los nodos en la red, capa por capa, mientras la fila correspondiente en las matrices  $A$  y  $B$  son obtenidas. Este proceso es realizado para todas las entradas de datos de entrenamiento para formar por completo  $A$  y  $B$ ; entonces los parámetros en  $S_2$  son identificados mediante la fórmula (3.23) o las fórmulas de mínimos cuadrados recursivos dadas en (3.28) y (3.29). Después que los parámetros en  $S_2$  han sido identificados, se calcula el error de medición para cada entrada de datos de entrenamiento. En el paso hacia atrás, las señales de error se propagan desde la salida hacia la entrada y el vector gradiente es acumulado para cada entrada de datos de entrenamiento. Al finalizar el paso hacia atrás para todos los datos de entrenamiento, los parámetros en  $S_1$  son actualizados por el método del gradiente.

Para valores fijos dados de los parámetros en  $S_1$  los parámetros hallados en  $S_2$  poseen la garantía de que serán los que corresponden al punto óptimo global en el espacio paramétrico  $S_2$  debido a la selectividad del error cuadrático medio. El aprendizaje híbrido no sólo reduce la dimensión del espacio de búsqueda del método del gradiente sino que también reduce sustancialmente el tiempo necesario para alcanzar la convergencia.

### 3.5 APROXIMACIONES NEURODIFUSAS

#### 3.5.1 *EL ANFIS COMO UN APROXIMADOR UNIVERSAL*

La presente sección explica una propiedad singular consistente en que cuando el número de reglas no está restringido, un modelo Sugeno de orden cero posee una capacidad de aproximación ilimitada para reproducir una función no lineal arbitraria sobre un conjunto compacto. Este hecho es intuitivamente razonable. Sin embargo, para dar una prueba matemática, se necesita explicar el Teorema de Stone-Weierstrass, (Kantarovich and Akilov, Functional Analysis, 1982) y (Royden, Real Analysis, 1968).

Sea el dominio  $D$  un espacio compacto de dimensión  $N$ , y sea  $F$  un conjunto de funciones reales continuas en  $D$  que satisfacen los siguientes criterios:

1. *Función idéntica.* La constante  $f(x) = 1$  está en  $F$ .
2. *Separabilidad.* Para dos puntos cualesquiera  $x_1 \neq x_2$  en  $D$ , existe una  $f$  in  $F$  tal que  $f(x_1) \neq f(x_2)$

3. *Cerradura algebráica.* Si  $f$  y  $g$  son dos funciones cualesquiera en  $F$ , entonces  $fg$  y  $af + bg$  están en  $F$  para dos números reales cualesquiera  $a$  y  $b$ .

Entonces  $f$  es denso en  $C(D)$ , el conjunto de funciones reales continuas en  $D$ . En otro caso, para algún  $\varepsilon > 0$  y cualquier función  $g$  en  $C(D)$ , existe una función  $f$  en  $F$  tal que  $|g(x) - f(x)| < \varepsilon$  para toda  $x \in D$ .

### 3.5.1.1 Función idéntica

La primera hipótesis del teorema de Stone-Weierstrass requiere que el sistema de inferencia difuso sea capaz de calcular la función idéntica. Es evidente que calcular esta función consiste en colocar la parte consecuente de cada regla igual a 1. Un sistema de inferencia difuso con una sola regla basta para satisfacer este requerimiento.

### 3.5.1.2 Separabilidad

La segunda hipótesis del teorema requiere que el sistema de inferencia difuso sea capaz de calcular funciones que posean valores diferentes para puntos diferentes. Obviamente, esto es alcanzable por cualquier sistema difuso con parámetros apropiados.

### 3.5.1.3 Cerradura algebraica (aditiva)

La tercera hipótesis del teorema requiere que el sistema de inferencia difusa sea invariante bajo la suma y la multiplicación. Supóngase que se tienen dos sistemas de inferencia difusa  $S$  y  $\hat{S}$ , cada una de ellas con dos reglas, y la salida final de cada sistema se especifica como:

$$S : z = \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2} \quad (3.30)$$

$$\hat{S} : \hat{z} = \frac{\hat{w}_1 \hat{f}_1 + \hat{w}_2 \hat{f}_2}{\hat{w}_1 + \hat{w}_2} \quad (3.31)$$

Entonces se tiene:

$$\begin{aligned} az + b\hat{z} &= a \frac{w_1 f_1 + w_2 f_2}{w_1 + w_2} + b \frac{\hat{w}_1 \hat{f}_1 + \hat{w}_2 \hat{f}_2}{\hat{w}_1 + \hat{w}_2} \\ &= \frac{w_1 \hat{w}_1 (af_1 + bf_1) + w_1 \hat{w}_2 (af_1 + bf_2) + w_2 \hat{w}_1 (af_2 + bf_1) + w_2 \hat{w}_2 (af_2 + bf_2)}{w_1 \hat{w}_1 + w_1 \hat{w}_2 + w_2 \hat{w}_1 + w_2 \hat{w}_2} \end{aligned}$$

Por lo tanto, es posible construir un sistema de inferencia difusa de tres reglas que calcule  $az + b\hat{z}$  donde la fuerza de disparo y la salida de cada regla está definida por  $w_i \hat{w}_j$  y  $af_i + bf_j$  ( $i, j = 1$  ó  $2$ ), respectivamente.

### 3.5.1.4 Cerradura algebraica (multiplicativa)

En este último caso debemos concluir si el teorema de Stone-Weierstarss puede aplicarse al modelo difuso Sugeno de orden cero. El producto de las salidas de dos sistemas de inferencia difusa  $z$  y  $\hat{z}$  puede expresarse como:

$$\hat{z} = \frac{w_1 \hat{w}_1 f_1 \hat{f}_1 + w_1 \hat{w}_2 f_1 \hat{f}_2 + w_2 \hat{w}_1 f_2 \hat{f}_1 + w_2 \hat{w}_2 f_2 \hat{f}_2}{w_1 \hat{w}_1 + w_1 \hat{w}_2 + w_2 \hat{w}_1 + w_2 \hat{w}_2} \quad (3.32)$$

por lo que es claro que es posible construir un sistema de inferencia difusa que calcule  $\hat{z}$ , donde la fuerza de disparo y la salida de cada regla están definidas por  $w_i \hat{w}_j$  y  $f_i \hat{f}_j$  ( $i, j = 1$  o  $2$ ), respectivamente.

Con base en la descripción precedente se concluye que las arquitecturas ANFIS que calculan  $az + b\hat{z}$  y  $z\hat{z}$  son de la misma clase que  $S$  y  $\hat{S}$ . Si y solamente sí, las funciones de pertenencia empleadas son invariantes bajo la multiplicación. Una clase de funciones de pertenencia que satisfacen la propiedad es la función de pertenencia Gaussianas escalada

$$\mu_{A_i}(x) = k_i e^{-\left(\frac{x-c_i}{a_i}\right)^2} \quad (3.33)$$

Otra clase de funciones que permanecen invariantes bajo el operador producto son las funciones de pertenencia de conjuntos clásicos, los cuales suponen valores de 0 ó 1. Las funciones de pertenencia de esta clase pueden verse como un caso especial de la función campana generalizada dada en (3.13) con parámetro  $b$  próximo a  $\infty$ , o como una función trapezoidal con  $a = b$  y  $c = d$ .

Por lo tanto, con una clase apropiada de funciones de pertenencia, un modelo Sugeno de orden cero puede satisfacer los cuatro criterios del teorema de Stone-Weierstrass. Esto es, para algún  $\varepsilon > 0$  y alguna función real  $g$ , existe un modelo difuso Sugeno de orden cero tal que  $|g(\vec{x}) - \hat{S}(\vec{x})| < \varepsilon$  para toda  $\vec{x}$  en un conjunto compacto. El argumento precedente de la capacidad de aproximación universal aplica para otros tipos de modelos difusos, puesto que el modelo Sugeno es un caso especial del modelo difuso Mandani, el modelo Tsukamoto y otros modelos Sugeno de alto orden (Jang, Sun y Mizutani, Neurofuzzy and Soft Computing, 1998).

Sin embargo, debemos ser precavidos en cuanto a la aceptación de esta idea, puesto que no se ha hecho mención de cómo construir el modelo Sugeno de acuerdo a un conjunto de entrenamiento dado; el teorema de Stone-Weierstrass proporciona solamente un teorema de existencia pero no un método constructivo.

### 3.6 APROXIMACIÓN DE UNA FUNCIÓN NO LINEAL

Esta sección presenta resultados de simulación de la arquitectura ANFIS para el modelo difuso Sugeno. En esta aplicación el ANFIS se utiliza para modelar una función no lineal.

#### 3.6.1 CONSIDERACIONES PRÁCTICAS

En un sistema de inferencia difusa convencional el número de reglas se determina por un experto que está familiarizado con el sistema a modelar. En el siguiente ejemplo, el conocimiento del experto no está disponible y el número de funciones de pertenencia

asignado a cada variable de entrada se selecciona empíricamente. Esta situación es similar a la da las redes neuronales donde no existe un mecanismo efectivo para determinar el número de neuronas ocultas necesarias para alcanzar un desempeño adecuado.

### *3.6.2 MODELADO DE UNA FUNCIÓN SINC DE DOS ENTRADAS*

En este ejemplo se utiliza el ANFIS para modelar una función *sinc* bidimensional definida por:

$$z = \text{sinc}(x, y) = \frac{\text{sen}(x)}{\text{sen}(y)} \quad (3.34)$$

Se emplea el algoritmo de entrenamiento híbrido. El algoritmo se programó en código de MatLab. Se utilizan datos distribuidos en el rango de entrada [-10, 10] x [-10,10]. El ANFIS empleado contiene 25 reglas con cinco funciones de pertenencia por entrada. Se calcularon 30 parámetros para los antecedentes y 75 para los consecuentes. Las siguientes gráficas muestran los resultados alcanzados utilizando 20 épocas de entrenamiento.

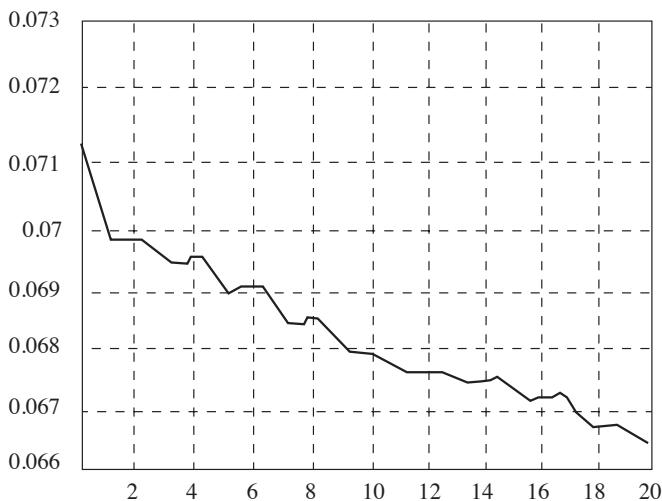


Figura 3.4 Comportamiento del error cuadrático medio.

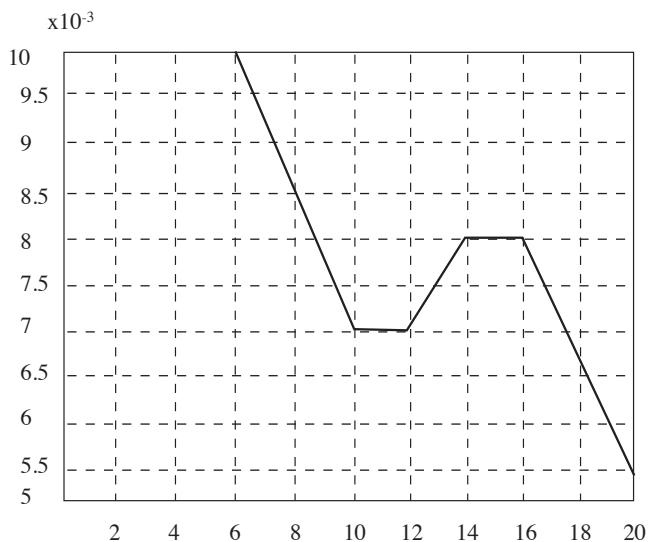


Figura 3.5 Comportamiento del tamaño de pas,  $K$ .

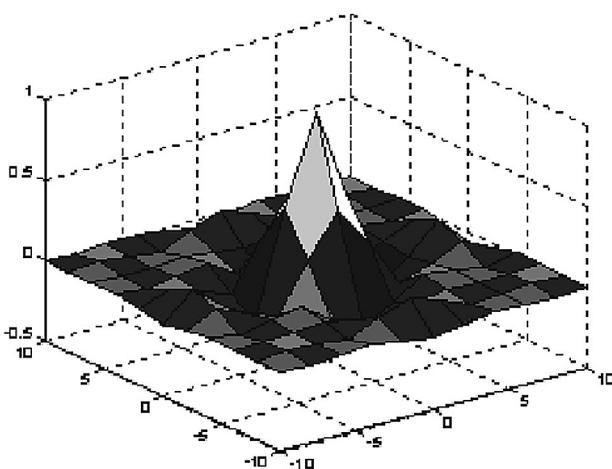


Figura 3.6 Gráfica de la superficie esperada.

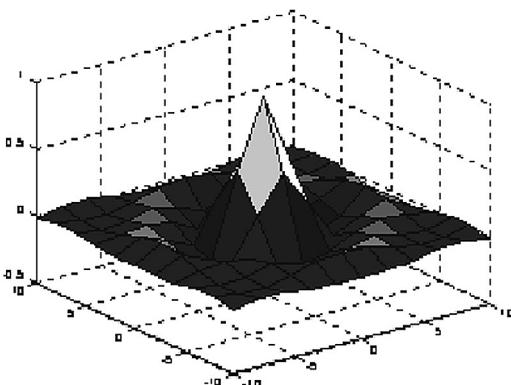


Figura 3.7 Gráfica de la superficie alcanzada.

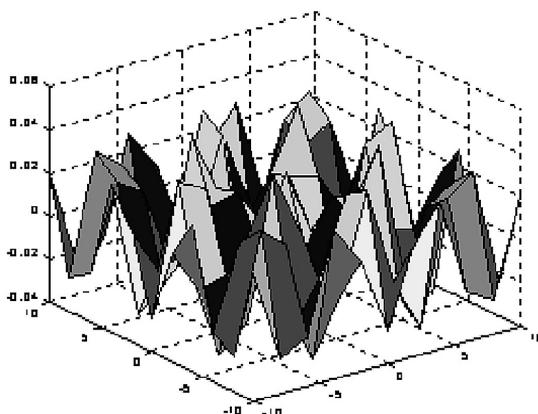


Figura 3.8 Diferencia entre la superficie alcanzada y deseada.

### 3.7 ESQUEMAS DE CONTROL BASADOS EN MODELOS NEURODIFUSOS

#### 3.7.1 INTRODUCCIÓN

Existe una gran variedad de esquemas de control convencional cuyo diseño se basa en modelos. Ejemplos de tales esquemas son el control inverso, el control con modelo interno y el control predictivo. El principio de operación de los dos primeros consiste en que la señal de control aplicada al proceso proviene de un modelo del sistema inverso del proceso. El tercero, utiliza un modelo del proceso que predice su comportamiento en un horizonte determinado. El modelo forma parte de una función de costo a optimizar con la finalidad de calcular la señal de control.

Cuando se diseñan controladores difusos se necesita adquirir el conocimiento a partir del operador que controla el proceso. Usualmente existen dos tipos de información que pueden obtenerse de un operador humano: información lingüística e información numérica.

La información lingüística puede obtenerse mediante una entrevista al operador. Ésta permite alcanzar acciones o decisiones de control como un conjunto de reglas difusas que pueden obtenerse después de pruebas de ensayo y error. La información numérica permite relacionar las señales de los sensores, observadas por el operador humano y, las acciones de control correspondientes como un conjunto de pares de datos entrada-salida. Estos pares de datos pueden emplearse como un conjunto de entrenamiento para construir un controlador difuso. Los sistemas de inferencia difusa basados en redes neuronales adaptables (ANFIS) pueden utilizar la información lingüística así como los pares de datos entrada-salida

Con base en lo anterior, la aplicación de los esquemas de control basados en el modelo puede extenderse mediante el uso de modelos neurodifusos, principalmente cuando no se dispone del modelo matemático del proceso a controlar. Si un modelo neurodifuso reemplaza al modelo convencional en estos esquemas se da origen al término esquemas de control basados en modelos neurodifusos.

### *3.7.2 APRENDIZAJE INVERSO EN EL DISEÑO DE CONTROLADORES (CONTROL NEURODIFUSO INVERSO)*

El desarrollo del aprendizaje inverso, también conocido como aprendizaje general para el diseño de controladores neurodifusos envuelve dos fases: en la fase de aprendizaje se utiliza una

técnica de identificación en línea o fuera de línea para modelar la dinámica inversa de la planta<sup>1</sup>. El modelo neurodifuso obtenido, el cual representa la dinámica de la planta, se usa para generar las acciones de control en la fase de aplicación. Ambas fases pueden proceder de manera simultánea, de aquí que este método sea compatible con el esquema clásico de control adaptable. El diagrama de bloques que corresponde al método de aprendizaje inverso se muestra en la figura 3.9.

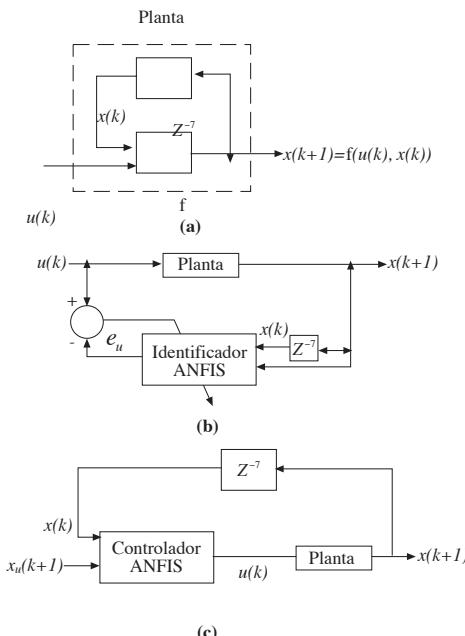


Figura 3.9 Diagrama de bloques para el método de aprendizaje inverso: (a) Planta; (b) Fase de entrenamiento y (c) Fase de aplicación.

<sup>1</sup> Antes de usar el aprendizaje inverso debemos asegurarnos de que el sistema a controlar tenga un inverso único y que éste exista.

Por hipótesis, se supone que el orden de la planta se conoce y que:

$$x(k+1) = f(x(k), u(k)) \quad (3.35)$$

donde  $x(k+1)$  representa el estado en el instante  $k+1$ ,  $x(k)$  es el estado actual y  $u(k)$  es la señal de control en el tiempo  $k$ . Por motivos pedagógicos, la señal de control es un escalar.

Similarmente, el estado en el instante  $k+2$  está dado por:

$$x(k+2) = f(x(k+1), u(k+1)) = f(f(x(k), u(k)), u(k+1)) \quad (3.36)$$

En general, se tiene que:

$$x(k+n) = F(x(k), U) \quad (3.37)$$

donde  $n$  es el orden de la planta;  $F$  es una función múltiple compuesta de  $f$ ;  $U$  es la secuencia de control desde  $k$  hasta  $k+n-1$ , lo cual es igual a  $[u(k), u(k+1), \dots, u(k+n-1)]^T$ . La ecuación precedente indica que la secuencia de control dada desde el instante  $k$  hasta  $k+1$  moverá el estado de la planta de  $x(k)$  a  $x(k+n)$  en exactamente  $n$  instantes de tiempo. Además se supone que la dinámica inversa del sistema existe, esto es,  $U$  puede expresarse como una función explícita de  $x(k)$  y  $x(k+n)$ :

$$U = G(x(k), x(k+n)) \quad (3.38)$$

Esta última ecuación dice que existe una secuencia única de control  $U$  que, especificada por el mapeo  $G$ , que puede llevar la planta del estado de  $x(k)$  a  $x(k+n)$  en  $n$  instantes de tiempo.

Por lo tanto, el problema del aprendizaje inverso consiste en determinar el mapeo inverso  $G$ . No obstante que en la fase de entrenamiento se minimiza el error  $\|U - \hat{U}\|^2$  no se garantiza la minimización de todo el error del sistema  $\|x_d(k) - x(k)\|^2$  en la fase de aplicación.

La figura 3.10 muestra el diagrama de control neurodifuso inverso, el cual es similar a los que se encuentran en la literatura de control convencional.

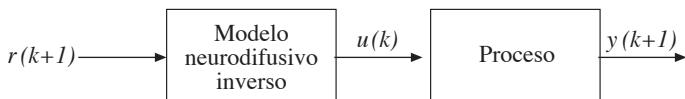


Figura 3.10 Control neurodifuso inverso.

Dado que el esquema de control está en lazo abierto, el control neurodifuso inverso trabaja solamente si no hay perturbaciones o errores de modelado. Los enfoques basados en la inversión pueden aplicarse solamente a sistemas estables con fase mínima (sistemas cuya dinámica inversa es estable).

### 3.7.3 CONTROL CON MODELO NEURODIFUSO INTERNO

Las perturbaciones y los errores de modelado causan problemas en los esquemas de lazo abierto. El esquema de control con modelo interno es una vía efectiva para compensar estos problemas. Un esquema de control neurodifuso con modelo interno, consiste de tres partes: el controlador basado en un modelo inverso del proceso, el modelo difuso y, un filtro de retroalimentación. La figura 3.11 ilustra el esquema correspondiente.

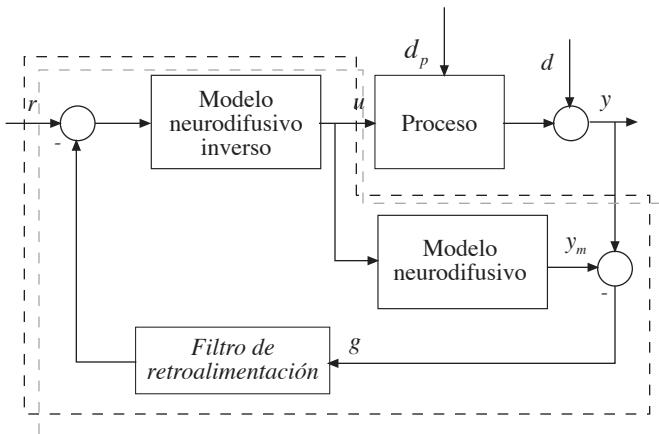


Figura 3.11 Esquema de control modelo neurodifuso interno.

El propósito de que el modelo neurodifuso trabaje en paralelo con el proceso es sustraer el efecto de la acción de control de la salida del proceso. Si las salidas predichas y las salidas medidas del proceso son iguales, el error es cero y el control trabaja en la configuración prealimentada. Si una perturbación actúa sobre la salida del proceso, la señal de retroalimentación es igual a la influencia de la perturbación y no se ve afectada por los efectos de la acción de control. Esta señal simplemente se sustrae de la referencia y el controlador trabaja en lazo cerrado.

Con un modelo perfecto, el esquema de control con modelo interno, es capaz de cancelar el efecto de perturbaciones aditivas de salidas no medidas sin presentar las desventajas de los controladores prealimentados. Sin embargo, esto no es cierto para perturbaciones aditivas de entrada o perturbaciones de carga.

Las dos propiedades básicas del esquema de control con modelo interno son: la estabilidad inherente y el control perfecto. La estabilidad inherente significa que si el controlador y el proceso son estables entrada-salida y un modelo perfecto del proceso está disponible, el sistema en lazo cerrado es estable entrada-salida. Si el sistema no es estable entrada-salida, pero puede estabilizarse mediante la retroalimentación, el control con modelo interno puede aun aplicarse. Control perfecto significa que si el controlador es un modelo inverso exacto, y el sistema en lazo cerrado es estable, entonces el control está libre de errores y en consecuencia  $y(k) = r(k)$ , para toda  $k$ . El control sin *offsets* de estado estacionario es alcanzado para referencias asintóticamente constantes.

Sin embargo, en la práctica, el modelo nunca es una representación exacta del proceso. La señal retroalimentada contiene entonces los efectos de perturbaciones no medidas y de errores de modelado, llegando a ser una verdadera retroalimentación.

Para errores de modelado grandes, el desempeño del control se deteriora y puede producir problemas de estabilidad. Una diferencia importante entre sistemas lineales y no lineales, consiste en que para los primeros, puede suponerse que las perturbaciones actúan aditivamente en la salida, mientras que para los no lineales, las perturbaciones no medidas que actúan sobre el proceso generalmente permiten diferenciar entre el modelo y el proceso. El filtro en la retroalimentación se introduce para filtrar el ruido de medición de la salida y estabilizar el lazo mediante la reducción de la ganancia. Con sistemas y modelos no lineales el filtro debe diseñarse empíricamente.

### 3.7.4 CONTROL PREDICTIVO BASADO EN EL MODELO NEURODIFUSO

El Control Predictivo Basado en el Modelo (CPBM), ha sido utilizado en más de 2 000 aplicaciones industriales relacionadas con la refinación, petroquímica, química, pulpa y papel así como en industrias de procesamiento de alimentos (Richalet, 1996), (Piché, Sayyar-Rodsari, Jonson y Gerules, 2000). En el CPBM, un modelo predice las salidas del proceso sobre un horizonte finito de predicción especificado. Una estrategia de control óptimo se obtiene minimizando una función de costo cuadrática, la cual usualmente incluye la diferencia entre la predicción y la salida deseada o referencia, así como una penalidad en los cambios del control. Las restricciones sobre el control y otras variables del proceso pueden formar parte del problema de optimización.

Dado que el CPBM requiere un modelo confiable del proceso, el modelado y la identificación constituyen la mayor parte del diseño de un controlador predictivo. El CPBM convencional utiliza modelos de predicción lineales para estimar las salidas futuras del proceso. Para plantas no lineales, los modelos lineales no son lo suficientemente buenos. La derivación de un modelo del proceso basado en las leyes de elementos y los principios de conservación no es sencilla resultando una actividad cara y lenta.

Pueden lograrse ahorros significativos en tiempo y dinero mediante la aplicación de modelos basados en redes neuronales y lógica difusa. La ventaja de los modelos difusos es que el conocimiento impreciso *a priori*, expresado en términos de reglas difusas, puede incluirse en el modelo inicial y refinarse usando datos del proceso.

Además puede crearse un modelo neurodifuso a partir de datos entrada-salida obtenidos en planta mediante la sintonización adecuada de una red tipo ANFIS, de tal modo que las acciones de control óptimo puedan calcularse mediante optimización numérica, utilizando al modelo como un predictor. Las entradas al proceso, así como las variables en él involucradas, pueden sujetarse a operaciones de restricción incorporadas en una rutina de optimización.

El CPBM es una metodología general para resolver problemas de control en el dominio del tiempo (Babuška , *An Overview of Fuzzy Modeling and Model-Based Fuzzy Control*, 1999). El esquema correspondiente se muestra en la figura 3.12 y está basado en tres conceptos principales:

1. El uso de un modelo para predecir salidas futuras del proceso en instantes de tiempo discretos, sobre un horizonte de predicción.
2. El cómputo de una secuencia de acciones futuras de control sobre un horizonte especificado para minimizar una función objetivo o de costo dada, de tal forma que la salida predicha del proceso es tan próxima como sea posible a la señal de referencia deseada.
3. La estrategia de horizonte móvil, para que solamente la primera acción de control en la secuencia sea aplicada, los horizontes son desplazados hacia el futuro y la optimización es repetida.

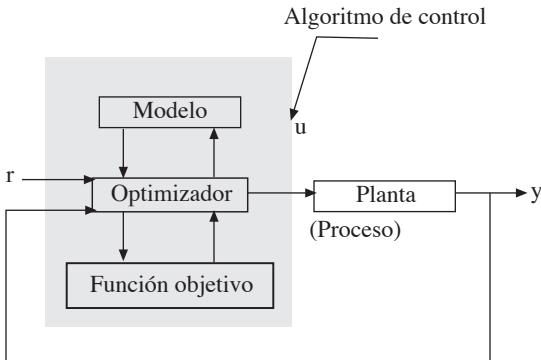


Figura 3.12 Esquema de control predictivo basado en el modelo CPBM.

Las salidas futuras del proceso son predichas sobre un horizonte de predicción  $H_p$  utilizando para ello un modelo del proceso. Los valores de salida predichos, denotados como  $\hat{y}(k+i)$  para  $i = 1, \dots, H_p$  dependen del estado del proceso en el instante actual  $k$  (por ejemplo, para un modelo entrada-salida representado por una colección de entradas y salidas pasadas) y en las señales futuras de control,  $u(k+i)$  para  $i = 0, \dots, H_c - 1$ , donde  $H_c$  es el horizonte de control. Si  $H_c$  se selecciona de modo que  $H_c < H_p$ , la señal de control es manipulada solamente mientras el horizonte de control permanezca constante, esto es:  $u(k+i) = u(k + H_c - 1)$  para  $i = H_c, \dots, H_p - 1$ . La figura 3.13 ilustra este principio.

La secuencia de señales futuras de control  $u(k+i)$  para  $i = 0, \dots, H_c - 1$  es calculada mediante la optimización de la función de costo, para llevar y mantener la salida del proceso, tan cerca como sea posible, de la trayectoria de referencia dada  $r$ , la cual puede ser el propio punto de ajuste, o bien, alguna versión filtrada de él.

En su mayoría las funciones objetivo o de costo utilizadas son modificaciones de la siguiente función cuadrática (Clarke, Mohtadi y Tuffs, 1987):

$$J = \sum_{i=1}^{H_p} \alpha_i (r(k+i) - \hat{y}(k+i))^2 + \sum_{i=1}^{H_c} \beta_i \Delta u(k+i-1)^2 \quad (3.39)$$

El primer término se utiliza para minimizar la desviación cuadrática de la salida del proceso con respecto a la referencia, en tanto que el segundo término representa la penalidad del esfuerzo de control. El último término también puede estar dado por la señal de control propiamente dicha, o bien, por alguna versión filtrada de ésta dependiendo del problema. Los vectores  $\alpha_i$  y  $\beta_i$  definen los pesos del error de la salida así como del esfuerzo de control. Las restricciones impuestas a los niveles y velocidades de la entrada de control pueden especificarse como una parte del problema de optimización. Generalmente puede utilizarse alguna otra variante de la función de costo, pero para una función de costo cuadrática, una lineal, un modelo invariante en el tiempo y en ausencia de restricciones, es posible obtener una solución analítica del problema de optimización. Además, cuando sea necesario, es aconsejable emplear optimización numérica.

El principio de horizonte móvil, consiste en que solamente la señal de control  $u(k)$  es aplicada al proceso (ver Fig. 3.13). En el siguiente instante (muestra), la salida del proceso  $y(k+1)$  está disponible y, tanto la optimización como la predicción, pueden repetirse con los datos actualizados. La acción de control calculada en el instante  $k+1$  generalmente será diferente de la que ha sido calculada en el paso  $k$ , dado que la información actual del proceso está disponible (Sousa, Babuška, Bruijn y Verbruggen, 1996).

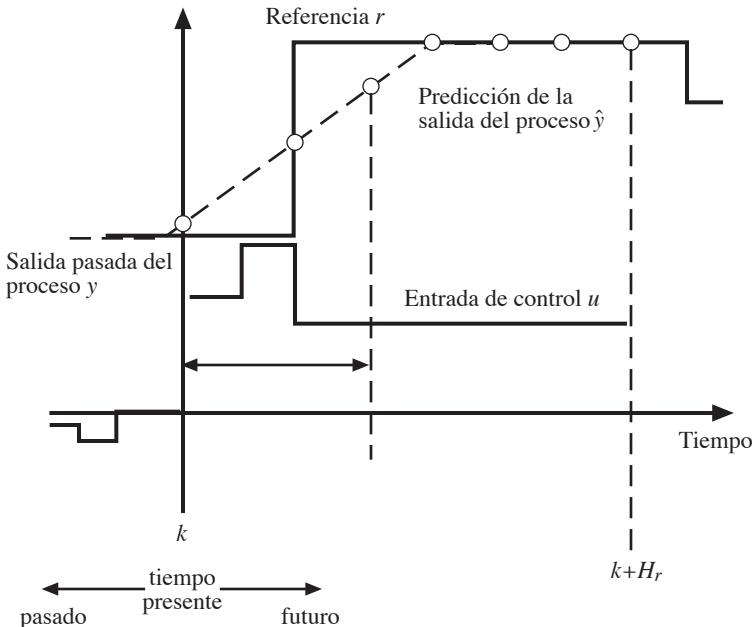


Figura 3.13 El principio básico del control predictivo.

Un modelo neurodifuso actuando como un predictor numérico puede acoplarse en el esquema de control que se ha mostrado en la figura 3.9. Dado que este tipo de modelos son en general no lineales, el problema de optimización en el CPBM es inherentemente no lineal y generalmente no convexo. El camino más simple para la minimización es utilizar alguna técnica de optimización numérica, como el método de Nelder-Mead (Nelder y Mead, 1965) o bien, la programación cuadrática secuencial (Gill, Murria y Wright, *Practical Optimization*, 1981). Sin embargo, estos algoritmos requieren una potencia computacional que podría ser

un obstáculo serio para la programación en tiempo real. Además la convergencia a un mínimo global no está garantizada y el algoritmo puede quedar atrapado por algún mínimo local, lo que puede resultar en acciones de control no deseados y en un pobre desempeño. Así mismo, el número de iteraciones requeridas para alcanzar la solución puede diferir drásticamente de muestra a muestra. Con frecuencia es necesaria la finalización prematura de las iteraciones, para cumplir con las restricciones en tiempo real, lo cual repercute en la obtención de soluciones de calidad inferior y en acciones de control inadecuadas. Actualmente se investigan esquemas alternativos de optimización para resolver el problema satisfactoriamente, encontrar un mínimo global y satisfacer los requerimientos en tiempo real.

Otra posibilidad puede ubicarse en modelos no lineales de arquitectura específica tales como los modelos afines no lineales. Por ejemplo, un modelo entrada-salida Takagi-Sugeno, que no involucre las entradas en la parte antecedente, es una forma no lineal afín. Para modelos afines las técnicas de linealización entrada-salida pueden aplicarse para obtener un modelo globalmente linealizado del proceso (Roubos, Babuska, Bruijn y Verbruggen, 1998). La idea central de esta técnica radica en que, si el proceso es de fase mínima, es posible linealizarlo mediante una ley de retroalimentación estática no lineal. Adicionalmente, si el modelo no lineal del proceso es un modelo entrada-salida afín, y asumiendo que no hay errores de modelación, el esquema de linealización es exacto, de modo tal que el sistema lineal resultante es válido en el rango completo de operación. Esta última, es la diferencia principal con respecto a la linealización clásica con expansiones de Taylor, donde el modelo lineal resultante solamente es válido en la vecindad del punto de operación nominal.

Generalmente el problema de optimización consiste en encontrar el máximo o el mínimo de una función objetivo. Ésta puede representar el costo, energía o beneficio en términos de las variables de interés del proceso que será analizado. El modelo del proceso y las restricciones describen las relaciones entre las variables de interés.

Los problemas típicos de ingeniería involucran procesos que pueden representarse mediante ecuaciones matemáticas o por datos experimentales, de tal modo que se establece un criterio de desempeño tal que se minimice una función de costo. El objetivo de optimizar consiste en hallar los valores de las variables del proceso con las cuales se obtenga el mejor valor para el criterio de desempeño. Los elementos descritos (proceso o modelo y el criterio de desempeño) constituyen el problema de optimización.

En la operación de plantas los beneficios surgen del desempeño de las mismas, esto lleva a la reducción de: contaminantes, consumo de energía, costos de mantenimiento y operación entre otros más.

### *3.7.5 CASO DE ESTUDIO*

Con respecto a los esquemas de control descritos en las secciones precedentes se describirá una aplicación de control inverso neurodifuso. El lector puede consultar (Ruz Hernández, Sintonización óptima de un controlador difuso para el calentamiento y presurización del generador de vapor de una unidad termoeléctrica, 2001), (Ruz, Suárez, Shelomov y Villavicencio, 2003) y (Pacheco, Ruz y Shelomov) para analizar aplicaciones que conciernen a control con modelo interno y control predictivo neurodifusos.

Supóngase que una planta se describe mediante la siguiente ecuación dinámica discreta:

$$y(k+1) = \frac{y(k)u(k)}{1 + y^2(k)} - \tan(u(k)) \quad (3.40)$$

donde  $y(k)$  y  $u(k)$  son el estado y la acción de control, respectivamente, en el tiempo  $k$ . Se supone que la dinámica de la planta se desconoce y que se va a construir un ANFIS que mapee cada par de datos de entrada  $[y(k), y(k+1)]$  hacia una acción de control deseada  $u(k)$ .

Este mapeo no se expresa fácilmente como una fórmula analítica, sólo si la ecuación que gobierna la dinámica de la planta se conoce.

Para entrenar el ANFIS se requiere colectar pares de datos de entrenamiento. Esto se hace seleccionando las entradas  $u(k)$ ,  $k = 1, \dots, 101$ , como números aleatorios uniformemente distribuidos entre  $-1$  y  $1$  y empleando la ecuación (3.40) (con  $y(1) = 0$ ) para generar cien pares de datos de entrenamiento de la forma  $[y(k), y(k+1); u(k)]$  con  $k = 1, \dots, 100$ . La figura 3.14 muestra las secuencias de entrada y salida del sistema a controlar.

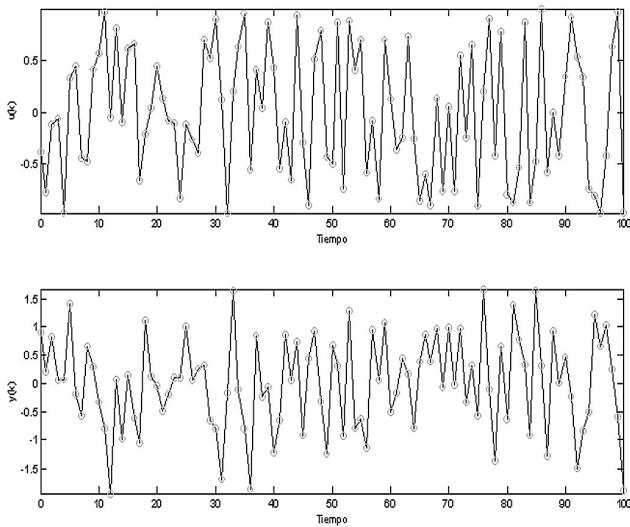


Figura 3.14 Colecta de datos de entrenamiento para el control inverso, donde la gráfica superior corresponde a  $u(k)$  y la inferior a  $y(k)$ .

La salida deseada está dada por la ecuación:

$$y_d(k) = 0.6 \sin(2\pi / 250) + 0.2 \sin(2\pi k / 50) \quad (3.41)$$

y el controlador ANFIS alcanza un buen desempeño, tal como se observa en la figura 3.15, donde la gráfica superior del lado izquierdo, indica las salidas deseada y actual, de la planta y la superior de la derecha muestra la diferencia entre ellas. En la parte inferior a la izquierda se muestran las funciones de pertenencia iniciales para  $y(k+1)$  y a la derecha las finales. Se consideraron 30 épocas (Jang, *Neurofuzzy and Soft Computing*, 1998) empleando comandos del Toolbox de Fuzzy Logic o la herramienta *anfisedit*.

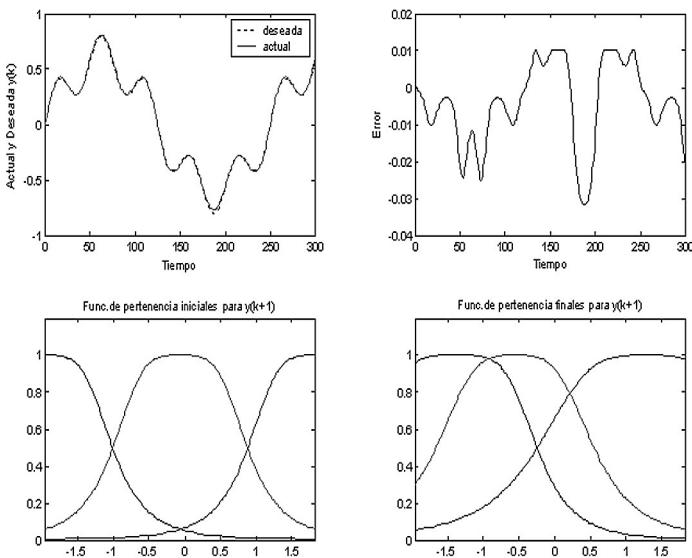


Figura 3.15 Desempeño del control inverso neurodifuso y funciones de pertenencia de  $y(k+1)$ .



## CAPÍTULO 4. ALGORITMOS GENÉTICOS

### 4.1 ANTECEDENTES

Los algoritmos genéticos están inspirados en los procesos evolutivos que se presentan en la naturaleza, donde la idea central es la supervivencia de los individuos más aptos y la modificación constante de sus descendientes para adaptarse al entorno donde habitan. Estos principios son en parte emulados por los algoritmos genéticos, cuando se usan para resolver problemas de optimización (Coello, Van Veldhuizen y Lamont, 2002), (Doyle, Francis y Tannenbaum, 1992), (Coello, 1999) y (Holland, 1992). La técnica de solución de problemas basada en los algoritmos genéticos supone que la solución potencial de cualquier problema es un individuo, que puede ser representado por medio de un conjunto de parámetros. Estos parámetros son vistos como los genes de un cromosoma y pueden ser estructurados por medio de una cadena de valores en forma binaria. En control automático, muchos problemas pueden ser planteados como problemas de optimización, con una o más funciones objetivo, sujetos a restricciones. Así por

ejemplo, en un problema de control retroalimentado, se podría plantear como función objetivo a la integral del valor absoluto de la señal del error (índice de desempeño). En este caso, la función objetivo contiene los parámetros de la planta y los parámetros del controlador, que son los que se desean encontrar (Mo, Coelho, Krohling & Fleming, 2003). Cada uno de los individuos de la población (cromosomas) sobre los cuales opera el algoritmo genético, contiene a los parámetros del controlador que se desea diseñar. Estos individuos son evaluados por medio de la función objetivo con el fin de asignarles un valor que les califique su aptitud dentro del problema de optimización. Después de que los individuos son evaluados, el proceso continúa con la etapa de selección, donde se mantienen y reproducen los individuos más aptos para que posteriormente, se realicen los procesos genéticos de cruzamiento y mutación en los cuales los individuos, al cruzarse, generan descendientes con nuevas características que dan variedad a la población. La mutación ocurre con una tasa de probabilidad baja por lo que sólo algunos de los individuos de la población son alterados. Sin embargo, estos pequeños cambios son suficientes para que el algoritmo pueda explorar todo el espacio de búsqueda y evitar converger hacia un mínimo o un máximo local.

El proceso completo se realiza  $n$  veces con una población inicial, creada en forma aleatoria, con  $m$  individuos (Gp, Liu; Jb, Yang; JF, 2003). El número de iteraciones, así como el tamaño de la población, son propuestas por el usuario. A continuación se detallan cada una de las partes que componen el proceso genético.

## 4.2 CODIFICACIÓN

En contraste con lo que sucede en la estrategia evolutiva y en la programación evolutiva. Los algoritmos genéticos trabajan con cadenas de bits de extensión fija ( $l$ ). Frecuentemente los algoritmos genéticos se aplican en problemas de control óptimo con parámetros que varían dentro de un rango continuo de valores. El proceso de codificación consiste en determinar el número de bits para representar a cada parámetro, tomando como base los valores mínimo y máximo de los parámetros, así como la resolución deseada. El número de bits para codificar un parámetro se puede encontrar a partir de la siguiente ecuación:

$$R = \frac{U - V}{2^I - 1} \quad (4.1)$$

donde:

$V$ = mínimo valor del rango del parámetro

$U$ = máximo valor del rango del parámetro

$I$ = número de bits necesarios para codificar el parámetro

$R$ = resolución deseada.

*Ejemplo 4.1.* Determinar el número de bits necesarios para codificar un parámetro contenido dentro del intervalo [-5,5] con una resolución de 0.1.

Aplicando la ecuación 4.1 se tiene como resultado  $I=6.6582$ , sin embargo, el número de bits debe ser un valor entero, entonces se toma el número inmediato superior, que es en este caso siete. Dado que se modifica el valor de la resolución deseada, el siguiente paso es recalcular el valor de la resolución con el nuevo valor de  $I$ ; después del cambio  $I$ , los resultados son los siguientes:

Resolución	0.0787
1	7 bits

El resultado anterior se puede visualizar en la tabla 4.1.

TABLA 4.1 RESULTADO DEL EJEMPLO 4.1

V. Parámetro Real	V. Codificado
-5	0000000
-4.9213	0000001
-4.8426	0000010
:	:
.	.
5	1111111

El número total de bits que conforman cada una de las cadenas binarias, es la suma de los números de bits de cada parámetro, conocido como ancho de la cadena. El número de bits por cada parámetro y el ancho de la cadena serán necesarios para inicializar la población, que es el siguiente paso del proceso genético.

#### 4.3 INICIALIZACIÓN DE LA POBLACIÓN

La población de cadenas binarias, que representan a los parámetros, se genera aleatoriamente con una distribución uniforme, sobre el espacio de búsqueda, a partir del conocimiento del número

de bits de cada parámetro, el ancho de la cadena y el tamaño de la población. El tamaño de la población está en función de la complejidad del problema a resolver, del ancho de las cadenas de la población y de los recursos de procesamiento. Con un tamaño pequeño existe poca probabilidad de encontrar la solución y con poblaciones muy grandes se utiliza mucho tiempo de proceso.

*Ejemplo 4.2.* Inicializar una población de tamaño cinco, considerando los datos del ejemplo 4.1.

Dado que la población es creada aleatoriamente, se podría tener una población como la siguiente

Población inicial	Número de cadena
1 0 1 0 0 1 0	1
0 0 1 1 1 0 1	2
0 1 1 0 1 1 0	3
1 1 1 0 1 0 0	4
1 1 0 0 1 1 1	5

Ahora se tienen los datos necesarios para la siguiente fase.

#### 4.4 DECODIFICACIÓN

El proceso de decodificación convierte cada uno de los individuos de la población (cadenas binarias), a sus respectivos valores reales, para ser evaluados a través de una función objetivo. A los individuos representados por sus valores reales se les conoce como fenotipos. La decodificación se puede realizar por medio del

código binario estándar o mediante la aplicación de otros códigos por ejemplo: el *código Grey*, el cual presenta la ventaja principal de que para valores de parámetros solución muy cercanos entre sí, existen pocos cambios que representan a esos números, lo cual se puede tomar como una ventaja sobre el código binario estándar en la solución de muchos problemas de optimización . A continuación se presenta la ecuación 4.2, para el caso binario estándar.

$$Val_{Par(i)} = u_i + \frac{v_i - u_i}{2^{l_x} - 1} \left( \sum_{j=0}^{l_x-1} ai(l_x - j)2^j \right) \quad (4.2)$$

Para el caso de decodificación mediante *código Grey* se tiene:

$$Val_{Par(i)} = u_i + \frac{v_i - u_i}{2^{l_x} - 1} \left( \sum_{j=0}^{l_x-1} (\oplus b_{ik}) 2^j \right) \quad (4.3)$$

donde la relación entre el código binario estándar y el *código Grey* está dada por la siguiente expresión:

$$b_i = \begin{cases} a_i, & \text{si } i = 1 \\ a_{i-1} \oplus a_i, & \text{si } i > 1 \end{cases}$$

Código binario estándar a C. Grey

$$a_i = \bigoplus_{j=1}^i b_j$$

C. Grey a binario estándar

Para mostrar la aplicación de esta fase de decodificación se retoman los resultados del ejemplo 4.1, utilizando el código binario estándar y mediante la ecuación 4.2, se obtienen los valores reales de los parámetros (fenotipos). Los cuales son mostrados en la columna tres de la siguiente tabla:

TABLA 4.2 VALORES REALES DE LOS PARÁMETROS

Población	Cadena	Valor Real
1 0 1 0 0 1 0	1	1.4534
0 0 1 1 1 0 1	2	-2.7177
0 1 1 0 1 1 0	3	-0.7502
1 1 1 0 1 0 0	4	4.1292
1 1 0 0 1 1 1	5	3.1061

## 4.5 EVALUACIÓN

Se evalúa la función objetivo tomando como argumentos cada uno de los elementos de la población (parámetros), es decir, los valores reales ya decodificados de cada una de las cadenas binarias. Durante este proceso se asigna una calificación a cada elemento de la población, que indica su grado de aptitud.

## 4.6 REPRODUCCIÓN

La etapa de reproducción está formada por tres fases a saber: selección cruzamiento y mutación. Dentro de esta etapa se puede hacer uso de un parámetro adicional que es conocido como elitismo, que permite conservar los mejores elementos a través de las  $n$  generaciones.

## 4.7 SELECCIÓN

Este proceso permite seleccionar, en cada uno de los ciclos evolutivos, a los elementos de la población con mejor valor de aptitud. Los métodos de selección más usados son los siguientes:

### 4.7.1 TORNEO (*TOURNAMENT*)

Puede resolver problemas de optimización (de minimización o de maximización). Un par de elementos de la población son seleccionados aleatoriamente y el individuo que tenga mejor calificación, es copiado en un lugar de almacenamiento conocido como *mating pool*. El proceso se repite hasta que este arreglo es llenado con los mejores elementos.

### 4.7.2 SELECCIÓN DE LA RUEDA DE LA RULETA (*ROULETTE WHEEL*)

Es una de las técnicas de selección más usada. Este mecanismo permite seleccionar de manera aleatoria a los mejores individuos, tomando como base alguna medida de aptitud. Se obtiene la suma ( $F_s$ ) de los valores de aptitud de cada uno de los individuos de la población actual, donde los individuos son transformados en segmentos contiguos en el intervalo  $[0, F_s]$ . Por ejemplo, la circunferencia de la rueda de la ruleta en la figura 4.1, es la suma de los valores de aptitud de cinco individuos. El individuo cinco es el que tiene el valor de desempeño más alto y es el que ocupa el intervalo más grande, en contraparte, el individuo uno es el que

tiene el desempeño más pobre y por lo tanto, ocupa el espacio más pequeño de la rueda de la ruleta. Para seleccionar un individuo, primero se genera un número de forma aleatoria, en el intervalo  $[0, F_s]$ , entonces se selecciona un individuo cuyo segmento cubre el número aleatorio. El proceso se repite hasta que se alcanza el número de elementos deseado.

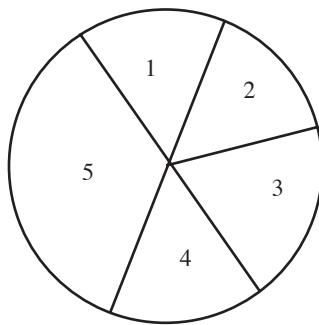


Figura 4.1 Selección de la Rueda de la Ruleta

#### 4.8 CRUZAMIENTO

Aunque la rutina de selección podría calificarse también como un operador. Los operadores de cruzamiento y mutación constituyen la herramienta esencial que permite dotar a los descendientes de las mejores características de los padres. Para ilustrar la manera como trabaja el operador de cruzamiento, se muestra un mecanismo de cruzamiento de un punto en la figura 4.2. Como puede verse se fija aleatoriamente un punto de cruzamiento. Las porciones de los dos cromosomas que se encuentran a la derecha del punto de entrecruzamiento, son intercambiadas para generar

la descendencia. Una tasa funcional ( $p_c$ ) con un valor típico de entre 0.6 y 1.0 es utilizada normalmente como la probabilidad de entrecruzamiento.

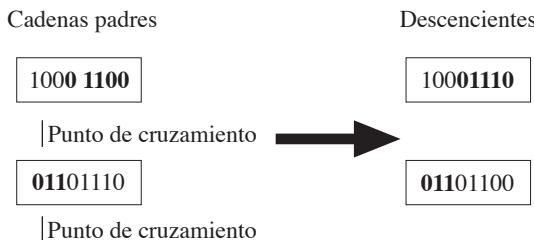


Figura 4.2 Proceso de cruzamiento.

## 4.9 MUTACIÓN

El operador genético de mutación se aplica a cada descendiente, de forma individual, después del ejercicio de entrecruzamiento. Este proceso altera cada bit aleatoriamente con una pequeña probabilidad ( $p_m$ ) con un valor típico de menos de 0.1. Para ejemplificar este proceso en la figura 4.3 se muestra a un individuo antes y después de ser mutado.

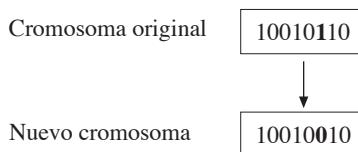


Figura 4.3 Mutación de un bit.

## 4.10 ALGORITMO GENÉTICO SIMPLE

Después de mencionar cada una de las etapas principales de un algoritmo genético simple conviene presentar, de manera esquemática, a los bloques que intervienen en el proceso genético así como el orden en la ejecución del mismo. En la figura 4.4 se muestra el algoritmo genético.

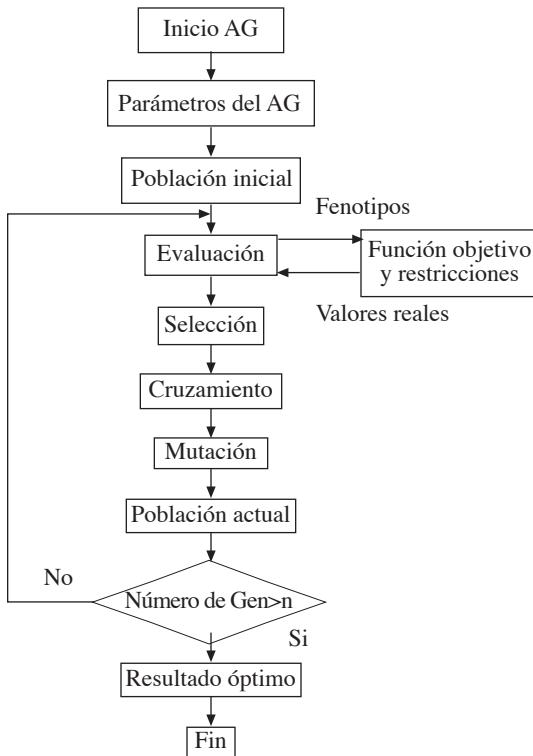


Figura 4.4 Algoritmo Genético Simple.

## 4.11 APLICACIÓN DE UN ALGORITMO GENÉTICO A UN PROBLEMA DE CONTROL

Diseño de un controlador basado en las acciones de control Proporcional, Integral y Derivativo (PID).

Para el diseño del controlador PID propuesto en este ejemplo se considera un sistema lineal e invariante en el tiempo, de una entrada-una salida (SISO). El esquema de control se muestra en la figura 4.5, el cual contiene el modelo de la planta  $G(s)$ , la variable de salida (variable controlada  $Y(s)$ ), la variable de referencia (*set point*  $R(s)$ ), la perturbación de entrada  $v(s)$ , la variable de error  $E(s)$  y la variable de control  $u(s)$ .

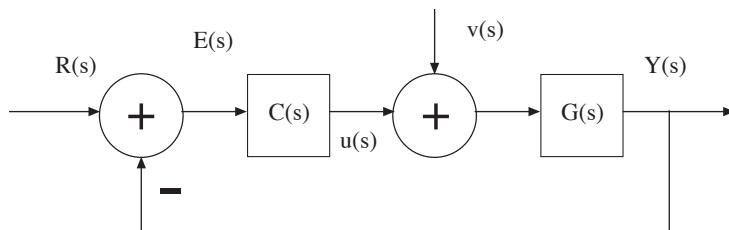


Figura 4.5 Esquema de control utilizado.

El modelo del controlador PID, representado por  $C(s)$ , es presentado por medio de la ecuación 4.4.

$$C(s) = K_p \left( 1 + \frac{1}{T_i s} + \frac{T_d s}{1 + T_f s} \right) \quad (4.4)$$

donde,  $K_p$  denota a la constante proporcional,  $T_i$  el tiempo integral,  $T_d$  el tiempo derivativo y  $T_f$  denota la constante de tiempo del filtro pasa bajas. En esta aplicación  $T_f=0.1 T_d$ .

Se propone el modelo que incluye un filtro pasa bajas, en la acción derivativa, por dos razones, la primera: Este atenúa el ruido de medición, que en los procesos reales se presenta al medir la variable de salida. La segunda porque el modelo tradicional del controlador  $C(s)=K_p+K_i/s+K_d s$  corresponde a una función impropia.

El modelo del proceso seleccionado esta dado por:

$$G(s) = \frac{e^{-5s}}{(s+1)^3}$$

donde el retardo  $e^{-5s}$  se puede sustituir por la aproximación de Padé de Primer orden, obteniéndose la siguiente función de transferencia:

$$G(s) = \frac{-2.5s + 1}{(2.5s + 1)(s + 1)^3}$$

#### *4.11.1 ESPECIFICACIONES DE DISEÑO*

Las especificaciones de diseño del controlador PID, son las siguientes:

- Seguimiento de la señal de referencia.
- Atenuación de la perturbación de carga.

- Robustez ante incertidumbres del modelo de la planta.
- Atenuación del ruido de medición.

#### 4.11.2 FUNCIÓN OBJETIVO

Se propone como función objetivo el siguiente índice de desempeño (Kristiansson y Lennartson, 2006):

$$\left| \frac{1}{s} G_{yv} \right|_{\infty} \quad \text{Índice de desempeño donde } \|\cdot\|_{\infty} \text{ denota la norma } H_{\infty} \quad (4.5)$$

La norma  $H_{\infty}$  es el valor máximo de la respuesta de magnitud de la gráfica de Bode (D.H. Griffel, 2002) y (Doyle, Francis & Tannenbaum, 1992).

$$G_{yv} = \frac{G(s)}{1 + G(s)C(s)}$$

Y la función de transferencia  $G_{yv}$ :

$$\left| \frac{1}{s} G_{yv} \right|_{\infty} \quad \text{Índice de desempeño donde } \|\cdot\|_{\infty} \text{ denota la norma } H_{\infty} \quad (4.6)$$

En la ecuación 4.6 están incluidos tanto los parámetros de la planta, los cuales son conocidos, como los parámetros del controlador ( $K_p$ ,  $T_b$ ,  $T_d$  y  $T_f$ ). El índice de desempeño (función objetivo), será minimizado por el algoritmo genético dentro del problema de optimización, arrojando como resultado, los valores de los parámetros del controlador. La función de transferencia  $G_{yv}$ ,

relaciona a  $Y(s)$  como variable de salida, y  $v(s)$  como variable de entrada. El producto de  $I/s$  por  $G_{yv}$  corresponde realmente a la variable  $Y(s)$ , ya que  $v(s)=I/s$ . Lo que quiere decir es que se desea minimizar el efecto de la perturbación  $v$  sobre la salida  $y$ , en el rango de frecuencia  $0 \leq \omega < \infty$ .

Las señales de referencia  $y$  y de perturbación de carga son del tipo escalón así, los valores de interés son: sobrepaso máximo, error en estado estacionario, tiempo de establecimiento y tiempo de decaimiento entre otros.

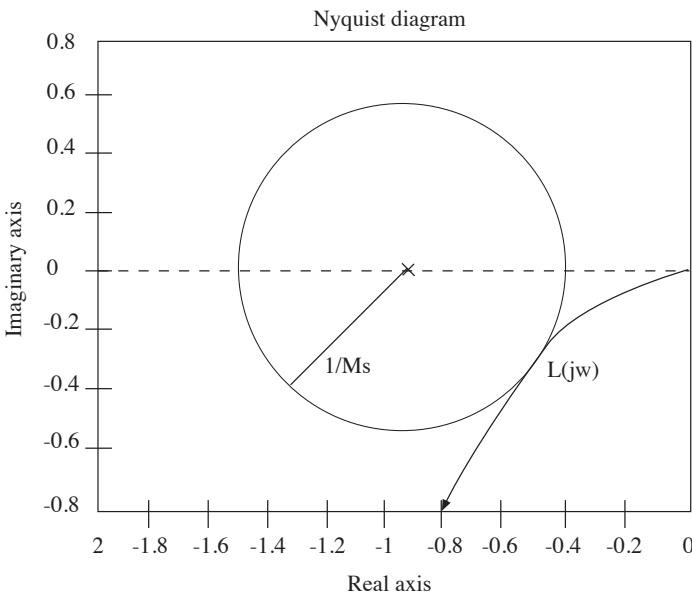
#### *4.11.3 RESTRICCIONES DEL PROBLEMA DE OPTIMIZACIÓN*

En el planteamiento del problema de optimización se proponen dos restricciones. La restricción 1, se muestra en la ecuación 4.7.

$$M_s = |S(s)| \max_{0 \leq \omega < \infty} |S(s)| \leq \beta \quad (4.7)$$

donde:

$$S(s) = \frac{1}{1 + G(s)C(s)} \quad (4.8)$$

Figura 4.6 Curva de Nyquist  $L(s)$  y definición de  $1/M_s$ .

La restricción uno, contiene a la función de transferencia  $S(s)$ , llamada función de sensibilidad, que asegura que el controlador diseñado sea capaz de minimizar las posibles variaciones en las dinámicas de la planta (variación de parámetros). Este criterio esta expresado en términos de la curva de Nyquist, de la función de transferencia de lazo abierto  $L(s)=C(s)G(s)$ , como se muestra en la figura 4.6. Así, el valor de  $M_s$  es el reciproco de la distancia más corta, de la curva de Nyquist y el punto critico -1. Se proponen valores de  $M_s$  en el rango de 1.3 a 2. (Åström y Hägglund, 1995).

La restricción dos se utiliza para acotar, a valores preestablecidos, la señal de control  $u$ . Esto es muy importante en la práctica dado que si se obtienen valores muy grandes de  $u$ , esto podría ocasionar la saturación de los actuadores, ocasionando deficiencias en el funcionamiento del proceso o inclusive, en casos extremos, la inestabilidad. Los valores máximos permitidos para la restricción dos, varían de acuerdo a la planta en particular. La restricción dos se presenta por medio de la ecuación 4.9.

$$\|G_{ur}(s)\|_{\infty} \leq \alpha \quad (4.9)$$

donde:

$$G_{ur}(s) = \frac{C(s)}{1 + G(s)C(s)} \quad (4.10)$$

#### 4.11.4 PARÁMETROS DEL ALGORITMO GENÉTICO SIMPLE

El algoritmo genético simple, esta hecho en código MatLab® e incluye las funciones de: codificación, creación de población inicial, decodificación, evaluación, selección, cruzamiento y mutación. El algoritmo genético requiere de los siguientes parámetros para su operación:

1. Número de parámetros
2. Tamaño de la población
3. Rango máximo de los parámetros
4. Rango mínimo de los parámetros
5. Resolución de los parámetros
6. Número de generaciones

7. Probabilidad de cruzamiento
8. Probabilidad de mutación
9. Número de puntos de cruzamiento

Además de los parámetros enlistados, se requiere proporcionar al *AG* la siguiente información:

Estrategia de selección: Torneo

Elitismo: Si

Problema de optimización: minimizar

#### *4.11.5 PROCESO DE OPTIMIZACIÓN*

El proceso de optimización completo se presenta en la figura 4.7, ella se aprecia que el algoritmo genético requiere de datos iniciales para formar la población de individuos, codificados como cadenas binarias (cromosomas), en donde cada uno de estos individuos contiene a los tres parámetros del controlador PID. Como ya se mencionó al inicio de este capítulo, se requiere obtener un valor real de cada uno de los cromosomas (fenotipos), para que estos sean pasados como argumento a la función de MatLab® que contiene a la función objetivo y a las restricciones.

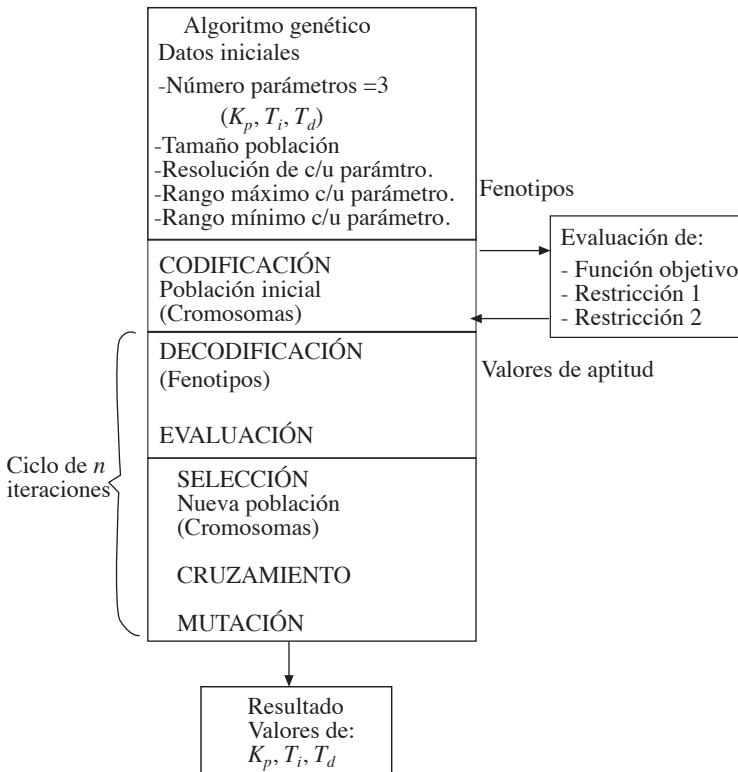


Figura 4.7 Proceso de optimización.

Cada uno de los individuos que son evaluados a través de la función objetivo, sujeto a las restricciones, son regresados al proceso genético para que, en base a su valor de aptitud, sean seleccionados los mejores individuos, y así formar la nueva población. También se observa que después de que el proceso se realiza  $n$  veces, se obtiene como resultado al individuo mejor

calificado de la última generación. Se espera que el individuo (parámetros del controlador PID), que se obtiene como resultado final, corresponda a un mínimo global de la función objetivo. La función objetivo y las restricciones están implantadas en código MatLab®.

#### *4.11.6 RESULTADOS*

Los resultados se obtuvieron por medio del software MatLab®-Simulink. Los resultados que se presentan corresponden a los parámetros del algoritmo genético, parámetros de los controladores PID y las gráficas donde se muestran: la evolución del algoritmo genético a través de las generaciones, la respuesta temporal de los procesos a las señales de referencia y perturbación de carga, las señales de control y las gráficas de Nyquist, donde se muestra el concepto de  $M_s$ . A continuación se presentan los resultados de las cuatro plantas de prueba propuestas.

En la tabla 4.3 se muestran los parámetros del algoritmo genético que fueron utilizados para el diseño del controlador PID aplicado a la planta del ejemplo.

TABLA 4.3 PARÁMETROS DEL ALGORITMO GENÉTICO DE LA PLANTA.

PARÁMETROS DEL ALGORITMO GENÉTICO			
Número de generaciones	Tamaño de la población	Probabilidad de cruzamiento	Probabilidad de mutación
50	25	0.85	0.04

En la figura 4.8 se muestra, cómo la función objetivo (valor de aptitud), es minimizada conforme el número de generaciones del proceso genético se incrementa.

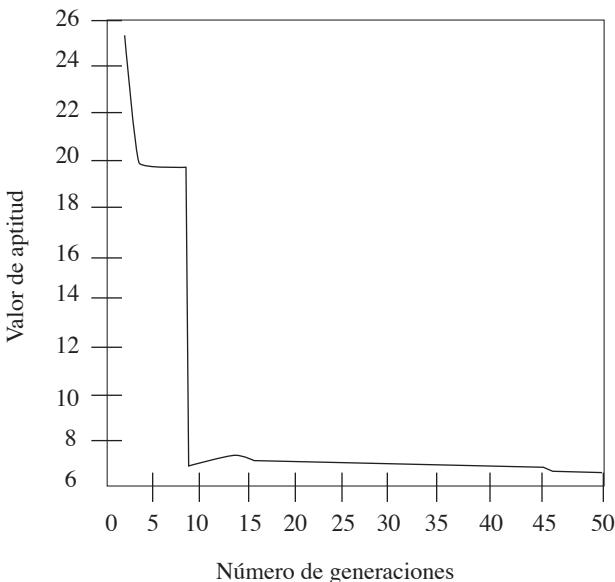


Figura 4.8 Valor de aptitud contra número de generaciones del proceso.

La tabla 4.4 contiene los valores de los parámetros del controlador PID sintonizado mediante el método de optimización, así como los valores de la función objetivo y las restricciones. En la misma figura 4.8 se presentan los parámetros del controlador obtenidos mediante el método de Ziegler-Nichols.

Los resultados de la simulación del proceso se muestran figura 4.9 donde se aprecia un mejor desempeño del PID diseñado con el método de optimización, con respecto al de Ziegler-Nichols.

TABLA 4.4 PARÁMETROS DEL CONTROLADOR PID DE LA PLANTA DEL EJEMPLO

Parámetros	Método de optimización						Método de Ziegler-Nichols					
	Funció n	Objetivo	$M_s$	Restricción	$\left\  \frac{1}{s} G_{vp} \right\ _\infty$	$\left\  S \right\ _\infty$	$\left\  G_{nr} \right\ _\infty$	$K_p$	$T_i$	$T_d$		
K <sub>p</sub>	K <sub>i</sub>	K <sub>d</sub>	T <sub>i</sub>	T <sub>d</sub>	T <sub>f</sub>	$\left\  \frac{1}{s} G_{vp} \right\ _\infty$	$\left\  S \right\ _\infty$	K <sub>p</sub>	T <sub>i</sub>	T <sub>d</sub>		
PID	0.64	0.18	1.04	3.53	1.62	0.1Td	6.36	1.94	7.07	0.75	7.9	2

El PID de Ziegler-Nichols presenta una respuesta oscilatoria y un mayor tiempo de establecimiento. También se puede ver en la gráfica la respuesta inversa al inicio del cambio de referencia y de perturbación de carga, dado que el retardo que presenta la planta se approximó por el modelo de Padé de primer orden, esto adiciona al modelo un cero en el semiplano derecho del plano complejo “*s*” (modelo de fase no-mínima).

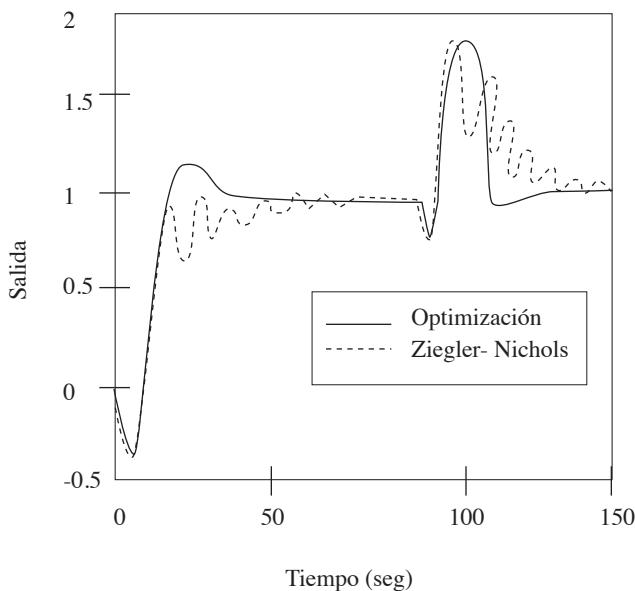


Figura 4.9 Respuesta temporal debido a las señales de referencia y perturbación de carga.

La señal de control del proceso que se obtiene con el método de optimización, es de menor magnitud y sin oscilaciones, comparada con la de Ziegler-Nichols, esto se puede constatar por medio de la figura 4.10.

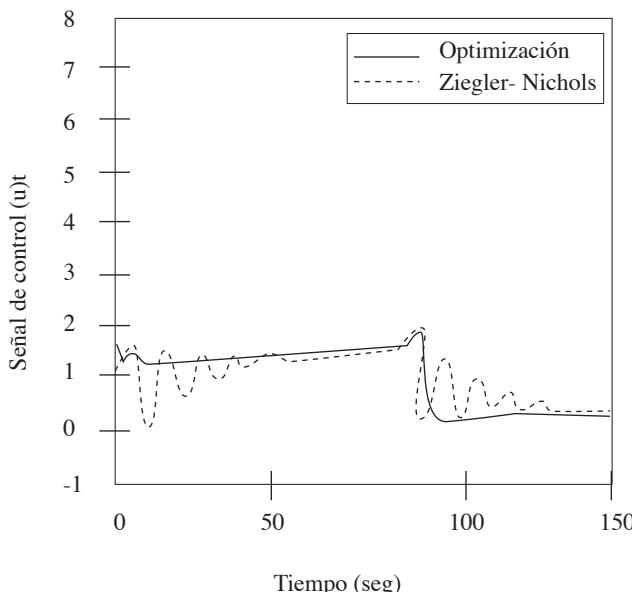


Figura 4.10 Señal de control  $u$ .

En la figura 4.11, se muestran la curva de Nyquist del proceso ( $L(jw)$ ), y  $I/M_s$  que corresponde a la distancia más corta, de la curva de Nyquist y el punto critico -1. El valor de  $M_s$  se encuentra en la tabla 4.4 y corresponde a la restricción 1.

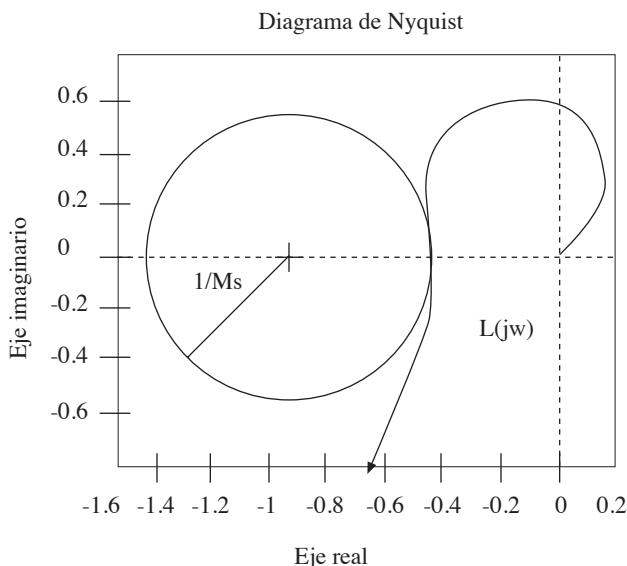


Figura 4.11 Significado grafico de  $M_s$ . y curva de Nyquist.



## **ANEXO. CONCEPTOS UTILIZADOS EN LA TEORÍA DE LOS SISTEMAS DE CONTROL**

*Sistemas de control de lazo cerrado y de lazo abierto.* Un sistema de control se define como la interconexión de componentes que forman la configuración de un sistema que proporcionará una respuesta de salida deseada. Se genera una señal proporcional al error entre la respuesta deseada y la real. La utilización de esta señal, para controlar el proceso, produce una secuencia de operaciones de lazo cerrado que se conoce como sistema retroalimentado. La principal diferencia entre los sistemas de lazo cerrado y los de lazo abierto es la generación y utilización de la señal de error. El sistema de lazo cerrado, cuando opera correctamente, lo hace de forma que el error se reduce a un valor mínimo.

*El principio de realimentación.* La idea de realimentación, ó retroalimentación, es muy simple y, sin embargo, extremadamente poderosa. Ésta ha tenido una profunda influencia en la tecnología actual. La aplicación del principio de realimentación ha obtenido mayores avances en control, comunicaciones e instrumentación. Muchos proyectos han sido concebidos sobre esta idea. El principio de realimentación se podría expresar de la siguiente manera:

Se incrementa la variable manipulada ( $u$ ) cuando la variable del proceso ( $y$ ) es mas pequeña que la señal de entrada ( $y_{sp}$ ) y disminuye la variable manipulada cuando la variable del proceso es mayor que la señal de entrada. Este tipo de realimentación es llamada realimentación negativa, porque la variable manipulada ( $u$ ) se mueve en dirección opuesta a la variable del proceso ( $y$ ). El principio de realimentación puede ser ilustrado por el diagrama de bloques mostrada en la figura A.1. El diagrama a bloques muestra que el proceso y el controlador son conectados en un lazo de realimentación cerrado. La presencia del bloque, con signo negativo en el lazo de retroalimentación, indica que la realimentación es negativa. La razón por la que el sistema de realimentación es de gran interés, es que la realimentación, al ser un proceso de lazo cerrado, minimiza los efectos que pudiera sufrir la variable del proceso contra los disturbios y variaciones que pudieran presentarse.

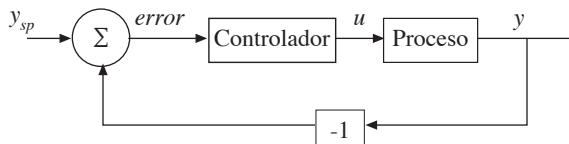


Figura A.1 Diagrama a bloques de un proceso retroalimentado con controlador.

En su configuración más elemental, un sistema retroalimentado contiene los bloques y señales que se muestran en la figura A.2.

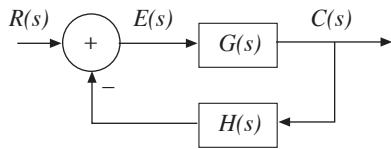


Figura A.2. Sistema retroalimentado.

La señal  $E_a(s)$  es una medida del error de sistema, y  $E_a(s) = R(s) - C(s)$  cuando  $H(s) = 1$ . La salida del sistema está representada por  $C(s)$ ;  $R(s)$  denota la señal de entrada;  $G(s)$  representa la función de transferencia del proceso bajo control y  $H(s)$  simboliza la función de transferencia del sensor con el cual se obtiene la señal de salida. La salida del sistema de lazo cerrado es:

$$C(s) = G(s)E_a(s) = G(s)[R(s) - H(s)C(s)]$$

$$C(s) = G(s)R(s) - G(s)H(s)C(s)$$

$$C(s) + G(s)H(s)C(s) = G(s)R(s)$$

$$C(s)[1 + G(s)H(s)] = G(s)R(s)$$

$$C(s) = \frac{G(s)}{1 + G(s)H(s)} R(s) \quad (\text{A.1})$$

De la ecuación A.1, se obtiene la función de transferencia de lazo cerrado  $G_{LC}(s)$  del sistema.

$$G_{LC}(s) = \frac{C(s)}{R(s)} = \frac{G(s)}{1 + GH(s)} \quad (\text{A.2})$$

La señal actuante del error se puede obtener sustituyendo  $C(s)=G(s) E_a(s)$  en la ecuación A.2

$$G(s)E_a(s) = \frac{G(s)}{1 + GH(s)} R(s)$$

$$E_a(s) = \frac{G(s)}{G(s)[1 + GH(s)]} R(s)$$

$$E_a(s) = \frac{1}{1 + GH(s)} R(s) \quad (\text{A.3})$$

Se puede apreciar en la ecuación A.3, que si se desea reducir la magnitud del error, entonces la magnitud de  $1+GH(s)$  debe ser mucho mayor que la unidad en el intervalo de  $s$  bajo consideración.

*Acciones de control proporcional, integral y derivativo.* Además de la acción de 2 posiciones o de encendido-apagado, las acciones de control proporcional, integral, y derivativo son acciones de control básicas que se encuentran en los controladores automáticos industriales. Para cada acción de control la relación entre la salida del controlador  $u(s)$  y la señal de error del actuador  $e(s)$  se

establece por una función de transferencia de forma específica. En lo que sigue, se ilustrarán funciones de transferencia  $u(s)/e(s)$  de acción de control proporcional, acción de control proporcional integral, acción de control proporcional derivativo y acción de control proporcional integral derivativo.

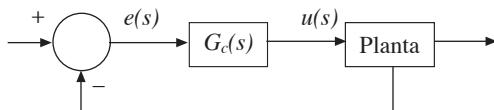


Figura A.3 Diagrama de bloques de un sistema de control.

En relación con la función de trasferencia del controlador  $G_c(s)$ , mostrado en la figura anterior, para la acción de control proporcional  $u(s)$  y  $e(s)$  están relacionados por:

$$\frac{u(s)}{e(s)} = G_c(s) = K_p \quad (\text{A.4})$$

donde  $K_p$  se llama ganancia proporcional.

Para la acción de control integral, la relación entre  $u(s)$  y  $e(s)$

$$\frac{u(s)}{e(s)} = G_c(s) \frac{k_i}{s} \quad (\text{A.5})$$

donde  $K_i$  es una constante.

Para la acción de control proporcional integral (*PI*),  $u(s)$  y  $e(s)$  están relacionadas por:

$$\frac{u(s)}{e(s)} = G(s) = K_p \left( 1 + \frac{1}{T_i s} \right)$$

$$u(s) = K_p e(s) + K_p \left( \frac{1}{T_i s} \right) e(s) \Rightarrow u = K_p e + \frac{K_p}{T_i} \int e dt \Rightarrow \boxed{\dot{u} = \dot{e} + \frac{T_p}{T_i} e}$$
(A.6)

donde  $K_p$  es la ganancia proporcional y  $T_i$  es una constante llamada tiempo integral.

Para la acción de control proporcional derivativo (*PD*),  $M(s)$  y  $E(s)$  están relacionadas por:

$$\frac{M(s)}{E(s)} = G_c(s) = K_p (1 + T_d s) \quad (A.7)$$

donde  $K_p$  es la ganancia proporcional y  $T_d$  es una constante llamada tiempo derivativo.

En forma similar, para la acción de control proporcional integral derivativo (*PID*),  $M(s)$  y  $E(s)$  están relacionadas por:

$$\frac{M(s)}{E(s)} = G_c K_p \left( 1 + T_d S + \frac{1}{T_i s} \right) \quad (A.8)$$

*Sensibilidad de los sistemas de control a las variaciones de los parámetros.* Un sistema de lazo cerrado detecta los cambios en la salida debido a los cambios del proceso e intenta corregir la salida. La sensibilidad de un sistema de control a las variaciones de los parámetros es de importancia primordial. Una ventaja importante de un sistema de control con retroalimentación es su capacidad para reducir la sensibilidad del sistema.

En el caso de circuito cerrado, si  $GH(s) > 1$  para todas las frecuencias complejas de interés, entonces  $E_d(s)$  es mínimo o cero, y la salida se puede aproximar por la ecuación A.9.

$$C(s) \equiv \frac{1}{H(s)} R(s) \quad (\text{A.9})$$

Es decir, la salida sólo está afectada por  $H(s)$ , que puede ser una constante. Si  $H(s)=1$ , tenemos el resultado deseado, esto es, la salida es igual a la entrada. Sin embargo, antes de utilizar este método para todos los sistemas de control, debemos observar que el requisito de que  $G(s), H(s)$  ó  $GH(s) > 1$ , puede hacer que la respuesta del sistema sea altamente oscilatoria y aun inestable, siendo ésta, la principal desventaja de los sistemas retroalimentados.

Pero el hecho de que conforme aumentamos la magnitud de la función de transferencia del circuito  $G(s)$   $H(S)$  reducimos el efecto de  $G(s)$  en la salida, es un concepto por demás útil. Por tanto, la primera ventaja de un sistema con retroalimentación es que el efecto de la variación de los parámetros del proceso ( $G(s)$ ), se reduce.

Con objeto de ilustrar el efecto de las variaciones de los parámetros se considerará un cambio en el proceso, de forma que el nuevo proceso sea  $G(s)+\Delta(G(s))$ . Entonces en el caso de lazo abierto, el cambio en la salida es:

$$\Delta C(s) = \Delta G(s)R(s)$$

En el sistema de lazo cerrado, se tiene:

$$C(s) + \Delta C(s) = \frac{G(s) + \Delta G(s)}{1 + G(s)H(s) + \Delta G(s)H(s)} R(s) \quad (\text{A.10})$$

$$\frac{G(s)R(s)}{1 + G(s)H(s)} + \Delta C(s) = \frac{[G(s) + \Delta G(s)]R(s)}{1 + G(s)H(s) + \Delta G(s)H(s)}$$

$$\Delta C(s) = \frac{[G(s) + \Delta(s)]R(s)}{1 + G(s)H(s) + \Delta G(s)H(s)} - \frac{G(S)R(s)}{1 + G(s)H(s)}$$

$$\Delta G(s) = \frac{\Delta G(s)R(s)}{[1 + G(s)H(s) + \Delta G(s)H(s)][1 + G(s)H(s)]}$$

$$\Delta C(s) = \frac{\Delta G(s)R(s)}{[1 + GH(s) + \Delta GH(s)][1 + GH(s)]}$$

cuando  $GH(s) > \Delta GH(s)$ , como es el caso frecuente, el cambio en la salida debido a la variación de  $G(s)$  se presenta en la ecuación A.11:

$$\Delta C(s) = \frac{\Delta G(s)}{(1 + GH(s))^2} = R(s) \quad (\text{A.11})$$

Se observa que el cambio en la salida del sistema del circuito cerrado, se reduce por el factor  $1+GH(s)$ , el cual desempeña un papel muy importante en las características de los sistemas de control con retroalimentación.

Formalmente, la sensibilidad del sistema, denotada por  $S$ , se define como: la relación del cambio porcentual en la función de transferencia del sistema respecto al cambio porcentual en la función del proceso  $G(s)$ . La función de transferencia de sistema es  $G_{LC}(s)$ , conocida también como  $T(s)$ . La sensibilidad se define como:

$$S = \frac{\Delta T(s) / T(s)}{\Delta G(s) / G(s)} \quad (\text{A.12})$$

En el límite, para pequeños cambios de incremento, la ecuación A.12 es equivalente a la ecuación A.13

$$S = \frac{\partial T / T}{\partial G / G} = \frac{\partial \ln T}{\partial \ln G} \quad (\text{A.13})$$

Claramente, por la ecuación:  $\Delta C(s) = \Delta G(s)R(s)$  la sensibilidad del sistema de circuito abierto es igual a uno. La sensibilidad del circuito cerrado se obtiene fácilmente usando la ecuación A.8.

La función de transferencia del sistema de circuito cerrado es:

$$T(s) = \frac{G}{1 + GH}$$

Por tanto, la sensibilidad del sistema con retroalimentación es:

$$S = \frac{\partial T}{\partial G} * \frac{G}{T} = \frac{1}{(1 + GH)^2} * \frac{G}{G / (1 + GH)} = \frac{1}{1 + GH(s)} \quad (\text{A.14})$$

La sensibilidad del sistema con retroalimentación a los cambios en el elemento de retroalimentación  $H(s)$  es:

$$S_H^T = \frac{\partial T}{\partial H} * \frac{H}{T} = \left( \frac{G}{1 + GH} \right)^2 * \frac{-H}{G / (1 + GH)} = \frac{-GH}{1 + GH} \quad (\text{A.15})$$

Cuando  $GH$  es grande, la sensibilidad tiende a la unidad y los cambios en  $H(s)$  afectan directamente a la respuesta de salida. Por tanto, es importante usar componentes de retroalimentación que no varíen con los cambios del ambiente o que puedan mantenerse constantes.

Con frecuencia, la función de transferencia del sistema  $T(s)$  es una función racional de la forma:

$$T(s) = \frac{G}{1 + GH}$$

Donde  $\alpha$  es un parámetro que puede estar sujeto a variaciones debidas al ambiente. Por tanto, podemos obtener la sensibilidad a en función de  $\alpha$  escribiendo ahora la ecuación:

$$S = \frac{\partial T / T}{\partial G / G} = \frac{\partial \ln T}{\partial \ln G} \text{ como: } S_\alpha^T = \left[ \frac{\partial \ln T}{\partial \ln \alpha} = \left. \frac{\partial \ln N}{\partial \ln \alpha} \right]_{\alpha_0} - \left. \frac{\partial \ln D}{\partial \ln \alpha} \right]_{\alpha_0} = S_\alpha^N - S_\alpha^D \quad (\text{A.16})$$

Donde  $\alpha_0$  es el valor nominal del parámetro. La capacidad para reducir el efecto de la variación en los parámetros de un sistema de control, agregando un lazo de retroalimentación, es una ventaja importante de los sistemas de control con retroalimentación. Para diseñar sistemas de lazo abierto,  $G(s)$  debe seleccionarse cuidadosamente con el objeto de cumplir las especificaciones deseadas. Sin embargo, un sistema de lazo cerrado conduce a que  $G(s)$  sea especificada con menos exactitud, ya que la sensibilidad a los cambios o errores en  $G(s)$  se reduce por la ganancia del circuito  $1+GH(s)$ . Este beneficio de los sistemas de circuito cerrado es una gran ventaja para los amplificadores electrónicos de la industria de las comunicaciones.

*Índices de funcionamiento.* Un índice de funcionamiento es una medida cuantitativa del desempeño de un sistema y se elige de forma tal que resalte las especificaciones importantes del sistema.

Si el propósito es mejorar el diseño de un sistema o diseñar un sistema de control adaptable, se debe elegir y medir un índice de funcionamiento.

Entonces el sistema se considera <<sistema de control óptimo>> cuando sus parámetros se ajustan de forma tal que el índice alcanza un valor extremo, comúnmente un valor mínimo. Para que un índice de funcionamiento resulte útil, siempre debe ser un número positivo o cero. Entonces el mejor sistema se define como aquel que minimiza este índice. Un índice de funcionamiento adecuado es la integral del error cuadrático (*ICE*), que se expresa como:

$$I_1 = \int_0^T e^2(t)dt \quad (\text{A.17})$$

El límite superior  $T$  es un tiempo finito elegido de manera arbitraria, de forma que la integral tienda a un valor en estado estacionario. Por lo general, es conveniente elegir  $T$  como el tiempo de estabilización  $T_s$ .

Este criterio discriminará entre los sistemas excesivamente amortiguados y los que presentan poco amortiguamiento. El valor mínimo de la integral representa un compromiso entre el valor de amortiguamiento y otros valores.

*Optimización.* Generalmente, los problemas de optimización son tratados por la rama de las matemáticas denominada investigación de operaciones, aunque prácticamente todas las ramas de la ciencia y la ingeniería necesitan tratar con problemas de este tipo en algún momento. Por ejemplo, en teoría de juegos se trata de maximizar la probabilidad de ganar y en reconocimiento de patrones de minimizar el error de clasificación de un patrón desconocido (como una imagen de satélite digitalizada, o un canal procesado de una señal de un electroencefalograma).

El problema de optimización no siempre está formulado de una forma tan clara. En muchos casos, la forma de la función  $F$  no se conoce, y debe de aproximarse mediante polinomios o combinaciones de funciones conocidas. Por ello habría que hallar los coeficientes de los polinomios o funciones que hacen que la función calculada se acerque lo más posible a la función objetivo. Cuando el problema se reduce a calcular una serie de coeficientes, se habla de optimización paramétrica.

En control industrial se plantean también problemas de optimización, como por ejemplo mantener en funcionamiento una máquina dentro de su régimen óptimo, por ejemplo. Cada máquina suele tener una serie de parámetros variables y lo que se desea optimizar, es habitualmente, la calidad del producto final o la rapidez a la hora de producirlo.

No siempre el espacio de búsqueda completo contiene soluciones válidas. En algunos casos, los valores de las variables se sitúan dentro de un rango más allá del cual la solución es inválida. Se trata entonces de un problema de optimización con restricciones. En este caso, el problema consiste en maximizar  $F(x)$  dentro del subespacio. Un ejemplo de este problema es el de optimización de los horarios de clase de una institución de enseñanza, hay que disponerla de forma que un profesor o alumno no deba estar en dos sitios a la vez, puede que el número de horas libres entre clases sea mínimo, y que se cumplan las preferencias de todos los implicados. En este caso, la optimización se reduce a cumplir todas las restricciones.

*Método analítico.* Si existe la función  $F$  es de una sola variable y se puede derivar dos veces en todo su rango, se pueden hallar todos sus máximos, sean locales o globales. Sin embargo, la mayoría de

las veces no se conoce la forma de la función  $F$ , y si se conoce, no tiene por qué ser diferenciable ni siquiera una vez. Incluso el tratamiento analítico para funciones de más de una variable es complicado.

*Métodos exhaustivos, aleatorios y heurísticos.* Los métodos exhaustivos recorren todo el espacio de búsqueda quedándose con la mejor solución y, los heurísticos utilizan reglas para eliminar zonas del espacio de búsqueda, se suelen utilizar en juegos para examinar y podar el árbol de posibilidades a partir de la jugada actual.

En los métodos aleatorios se va muestreando el espacio de búsqueda acotando las zonas que no han sido exploradas; se escoge la mejor solución, y, además, se da el intervalo de confianza de la solución encontrada.



## BIBLIOGRAFÍA

- Aceves-López, A., & Aguilar-Martin, J. (2006). *A simplified version of Mamdani's fuzzy controller: The Natural Logic Controller*. *IEEE Trans. on fuzzy systems*, 16-30.
- Alcaraz, Romero & Manuel, Víctor. (2000). *Estructura y funcionamiento del sistema nervioso*. Guadalajara, Jalisco: Trillas.
- Anzurez Marín, J. y Pitalúa Díaz, N. (2009). Detección y Aislamiento Robusto de Fallas: *Observadores con Entradas Desconocidas (UIO)*, Revista DYNA-Universidad Nacional de Colombia. 209-217 pp.
- Anzurez-Marin, J., Pitalúa-Díaz, N., Cuevas-Silva, O. and Villar-García, J. (2008). *Unknown Inputs Observers Design for Fault Detections in a Two-Tank Hydraulic System*, CERMA - IEEE, Cuernavaca México. 373-378 pp.
- Åström, K.J. and T. Hägglund (1995). *PID Controllers: Theory, Design, and Tuning*. Second Edition, Instrument Society of America, Research Triangle Park, NC.

- Babuška, R., & Verbruggen, H. B. (1996). *An Overview of Fuzzy Modeling for Control*, Control Eng. Practice, Vol. 4, No. 11, 1593-1606.
- Babuška, R., (1999). An Overview of Fuzzy Modeling and Model-Based Fuzzy Control, World Scientific Series in Robotics and Intelligent Systems- Vol. 23, Fuzzy Logic Control Advances in Applications, Part I:Tutorials pp. 3-36, World Scientific Publishing Co. Pte. Ltd.
- Chen, C. (2006). *Analog and Digital Control System Design*. USA: Oxford University Press.
- Clarke, D. W., Mohtadi, C., & Tuffs P.S. (1987) Generalized Predictive Control, *Automatica*, Vol. 23, No. 2, pp. 137-160.
- Coello-Coello Carlos A., David A. Van Veldhuizen, and Gary B. Lamont (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers.
- Coello-Coello Carlos A (1999).An Update Survey of Evolutionary-Based Multiobjective Optimization Techniques: State of the Art and Future Trends. *Congress on Evolutionary Computation. Washington, D.C. IEEE Service Center*, pp. 3-13.
- Coughlin, R., & Driscoll, F. (1987). *Circuitos integrados lineales y amplificadores operacionales*. México: Prentice Hall.
- Doyle John C; Francis Bruce A; Tannenbaum Allen R (1992). *Feedback Control Theory*. Macmillan Publishing Company, USA.
- D.H. Griffel (2002). *Applied Functional Analysis*, Dover Publications. Inc. USA.
- Driankov, D., Hellendoorn, H., & Reinfrank, M. (1996). *An Introduction to Fuzzy Control*. USA: Springer.
- Elman, J. (1987). *Learning the Hidden Structure of Speech*. University of California at San Diego.

- Farinwata, S., Filev, R., & Langari, R. (2000). *Fuzzy Control Synthesis and Analysis*. New Jersey, USA: John Wiley and Sons.
- Gill, P.E., Murray, W., & Wright, M.H., (1981). *Practical Optimization*, Great Britain, Academic Press.
- Gp, Liu; Jb, Yang; JF (2003). Whidborne. *Multiobjective Optimisation and Control*. Research Studies Press LTD, Great Britain.
- Hammerstrom, Dan. (June, 1993). *Neural Networks at Work*. *IEEE Spectrum*, 26 - 32 pp.
- Hammerstrom, Dan. (July, 1993). *Working with Neural Networks*. *IEEE Spectrum*, 46 -53 pp.
- Hebb, D. O. (1975). *The organization of the behavior*. New York: John Wiley & Sons USA.
- Hinton, G. E. (1992). How Neural Networks Learn from Experience. *Scientific American Magazine*. 144-151 pp.
- Holland John H (1992). *Adaptation in Natural and Artificial Systems*, First MIT Press Edition, USA.
- Hopgood,A. (2001). *Intelligent System for Engineers and Scientist*. USA: CRC Press.
- Jang, J.-S. R. (1993). ANFIS: Adaptive-Network-based Fuzzy Inference System”, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, Num. 3, 665-685 pp.
- Jang, J.-S. R. , & Sun, C. – T. (1995). Neuro Fuzzy Modeling and Control, *Proceedings of the EEE*, Vol. 83, Num.3., 378-406 pp.
- Jang, J.-S. R. , Sun, C. – T., & Mizutani, E. (1998) Neurofuzzy and Soft Computing, Prentice Hall.
- Kantarovich L. V. and Akilov G. P. (1982), *Functional Analysis*, Pergamon, Oxford, Second Edition, 1982.

- Kristiansson Birgitta and Lennartson Bengt (2006). Robust Tuning of PI and PID Controllers Using Derivative Action despite Sensor Noise. *IEEE Control Systems Magazine*, Febrero, Vol. 26 No. 1, pp 55-69.
- Lagunas-Jiménez, R., & Pitalúa-Díaz, N. (2007). Tuning Fuzzy Control Rules via Genetic Algorithms. *CERMA-IEEE*. Cuernavaca, México: IEEE.
- Li, H., Chen, C., & Huang, H. (2001). *Fuzzy Neural Intelligent Systems*. CRC Press. USA.
- Lippmann, R. (1987). An introduction to Computing with Neural Nets. *IEEE ASSP Magazine*, 4-22 pp.
- Matía, E. F. (1994). *Análisis y diseño de sistemas de control de procesos basados en lógica borrosa, tesis de doctorado*. ETSII. Madrid, España.
- McCulloch, W. S. (1943). A logical Calculus of the ideas Imminent in Nervous Activity. *Bulletin of Mathematical Biophysics* 5: 115-33 pp.
- Minsky, M., & Papert, S. (1969). *Perceptrons: An introduction to Computational Geometry*. Massachusetts: MIT Press.
- Mo Jamshidi, Leandro dos Santos Coelho, Renato A. Krohling and Peter J. Fleming (2003). *Robust Control Systems with Genetic Algorithms*. CRC Press.
- Nelder, J.A., & Mead R. (1952) A Simplex Method for Function Minimization, *J. Computer*, Vol. 7, pp. 308-313.
- Nguyen, H. T., Prasad, N. R., Walker, C. L., & Walker, E. A. (2003). *A First Course in Fuzzy and Neural Control*. USA: Chapman & Hall / CRC .
- Nilsson, N. J. (1965). *Foundations of the Trainable Pattern-Classifying Systems*. New York: McGraw Hill.

- Pacheco Richard, J. G., Ruz Hernández, J. A., & Shelomov E., Técnicas neurodifusas aplicadas al control del equipo didáctico de la bola y la viga, *Memorias del X Seminario anual de Automática, electrónica industrial e instrumentación, SAEI*, Vigo, España, 2003.
- Passino, M. K., & S., Y. (1998). *Fuzzy Control*. USA: Addison Wesley.
- Pelling, S., Moore, R., & Tomlinson, M. (1986). The Multi-Layer Perceptron as a Tool for Speech Pattern Processing Research. *Proceedings of IA Autumn Conference on Speech and Hearing*
- Peña, Antonio. (1998). *La membrana de las células*. México: Fondo de Cultura Económica.
- Piché, S., Sayyar-Rodsari, B., Johnson D., & Gerules, M. (2000). Nonlinear Model Predictive Control Using Neural Networks, *IEEE Control Systems Magazine*, Vol. 20, Núm. 3, 53-62.
- Pitalúa-Díaz, N. (2005). *Estabilización global de un sistema difuso continuo Takagi-Sugeno por medio de su modelo discreto*, tesis de doctorado. Guadalajara, México: Cinvestav.
- Ramón y Cajal, S. (1995). *Histology of the Nervous System of Man and Vertebrates*. New York: Oxford University Press.
- Richalet, J. , (1993). Industrial Applications of Model Based Predictive Control, *Automatica*, Vol. 29, Num. 5, 1251-1274 pp.
- Rosenblatt, Frank. (1958). Two theorems of Statistical Separability in Perceptron. In *Mechanization of thought Processes* , Vol.1.
- Rosenblatt, Frank. (1959). *Principles of Neurodynamics*. New York: Spartan Books.

- Roubos, J. A., Babuska, R., Bruijn, P.M., & Verbruggen H. B. (1998) Predictive Control by local linearization of a Takagi-Sugeno Fuzzy Model, *Proceedings of the IEEE International Conference on Fuzzy Systems*, 37-42 pp.
- Royden, H. L. (1968), *Real Analysis*, Macmillan, New York.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). *Learning internal representations by error propagation*, in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1, Rumelhart, D.E. & James, L., McClelland, Eds. Cambridge, MA: MIT Press, chap. 8, 318-362 pp.
- Rumelhart, D., Hinton, G., & Williams, R. (1986). *Learning Internal Representation by Error Propagation*. Massachusetts: MIT Press.
- Ruspini, E., Bonissone, P. & Pedrycz, W. (1998). *Handbook of Fuzzy Computation*, Taylor and Francis Group.
- Ruz Hernández, J. A., (2001). *Sintonización óptima de un controlador difuso para el calentamiento y presurización del generador de vapor de una unidad termoeléctrica*, Tesis de Maestría en Ingeniería de Control, Facultad de Ingeniería de la Universidad Autónoma del Carmen.
- Ruz Hernández, J. A., Suárez, D. A., Shelomov, E. & Villavicencio, A.,(2003). Predictive Control Based on an Auto-Regressive Neuro-Fuzzy Model Applied to the Steam Generator Startup Process at a Fossil Power Plant”, *Revista de Computación y Sistemas*, Vol. 6, No. 3, 204-212 pp.
- Sejnowski, T., & Rosenberg, C. (1986). *NETtalk: A Parallel Network Taht Learns to read Aloud*. Cambridge, MA, USA: John Hopkins University, 661 - 672 pp.
- Simon, Jean-Claude. (1986). *Pattern and Operators: The foundations of Data Representations*. New York: McGraw-Hill.

- Sousa, J. M., Babuška, R. , Bruijn, P. , & Verbruggen, H. B. (1996). Comparison of Convencional and Fuzzy Predictive Control, *Proceedings of the Fifth IEEE International Conference on Fuzzy Systems*, 1782-1787 pp.
- Takagi, T., & Sugeno, M. (1985). Fuzzy Identification of systems and its applications to modeling and control, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 15, 116-132 pp.
- Tapia, R. (2001). *Las células de la mente*. México: Fondo de Cultura Económica.
- Von Altrock, C. (2000). *Introduction to Neurofuzzy Technologies*, Tutorial and Workshop available in [www.fuzzytech.com](http://www.fuzzytech.com).
- Wang, L. X. (1997). *A Course in Fuzzy Systems and Control*. USA: Prentice Hall.
- Wang, L. X. (1998). Stable and Optimal Fuzzy Control of Linear Systems. *IEEE Transactions on Fuzzy Systems*, 137-143 pp.
- Wang, L.-X. (1994). *Adaptive Fuzzy Systems and Control, Design and Stability Analysis*, New Jersey, Prentice-Hall.
- Werbos P. (1974) Beyond regression: New tools for prediction and analysis in the behavioral sciences, Ph. D. Dissertation, Harvard University, USA.
- Werbos, Paul. (1990). Backpropagation trough Time. *Proceedings of the IEEE*, Vol. 78, No. 10.
- Ying, H. (2000). *Fuzzy Control and Modeling*. New York, USA: IEE Press Series on Biomedical Engineering.
- Zadeh, L. A. (1965). Fuzzy Sets. *Information and Control Journal*, 338-353 pp.
- Zhang, J., Li, R., & Zhang, P. (2001). Stability Analysis and Systematic Design of Fuzzy Control Systems. *Fuzzy Sets and Systems Journal*. 65-75 pp.

Zilouchian, A., & Jamshidi, M. (2001). Intelligent Control Systems using Soft Computing Methodologies. USA: CRC Press.

## ÍNDICE

Agradecimientos .....	7
Prólogo.....	9
Introducción .....	11
Capítulo 1. Sistemas difusos .....	17
1.1 Antecedentes .....	17
1.2 Teoría de conjuntos difusos .....	18
1.3 Funciones de membresía típicas .....	20
1.4 Operaciones entre conjuntos difusos .....	23
1.5 Relaciones difusas .....	25
1.6 Operaciones con relaciones difusas.....	26
1.7 Sistemas difusos .....	29
1.8 Inferencia difusa .....	33
1.9 Fuzzificación.....	34
1.10 Defuzzificación.....	35
1.11 Ejemplo de un sistema de control difuso.....	36
Capítulo 2. Redes neuronales artificiales .....	47
2.1 Fundamentos biológicos .....	47
2.2 Redes neuronales artificiales .....	56

2.3 Redes neuronales artificiales .....	73
Capítulo 3. Sistemas neurodifusos.....	87
3.1 Antecedentes de los sistemas neurodifusos .....	87
3.2 Redes adaptables, arquitectura y algoritmo básico de aprendizaje.....	91
3.3 Sistemas de inferencia difusa basados en redes adaptables .....	96
3.4 Algoritmo de aprendizaje híbrido .....	99
3.5 Aproximaciones neurodifusas.....	103
3.6 Aproximación de una función no lineal .....	107
3.7 Esquemas de control basados en modelos neurodifusos.....	111
Capítulo 4. Algoritmos genéticos.....	129
4.1 Antecedentes .....	129
4.2 Codificación.....	131
4.3 Inicialización de la población .....	132
4.4 Decodificación .....	133
4.5 Evaluación .....	135
4.6 Reproducción.....	135
4.7 Selección.....	136
4.8 Cruzamiento.....	137
4.9 Mutación.....	138
4.10 Algoritmo genético simple.....	139
4.11 Aplicación de un algoritmo genético a un problema de control .....	140
Anexo. Conceptos utilizados en la teoría de los sistemas de control .....	155
Bibliografía .....	169



*Introducción a los sistemas inteligentes* se terminó de imprimir en diciembre de 2009 en Laredo Impresores, Calzada Macristy de Hermosillo y esquina con Montes de Oca #958 Col. República Mexicana, C.P. 21250, Mexicali, Baja California. La edición estuvo al cuidado del Departamento de Editorial Universitaria de la Secretaría de Rectoría e Imagen Institucional de la Universidad Autónoma de Baja California. En la composición tipográfica se utilizó la familia Times New Roman. El tiraje consta de 300 ejemplares.