



UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ciencias y Humanidades

FUNDAMENTOS TEORICOS DE LOGICA DIFUSA (FUZZY LOGIC)
Y SU APLICACION A SISTEMAS ELECTRONICOS

VICTOR ADOLFO CHANG CHANG

Guatemala

1993

FUNDAMENTOS TEORICOS DE LOGICA DIFUSA (FUZZY LOGIC)
Y SU APLICACION A SISTEMAS ELECTRONICOS



UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ciencias y Humanidades

FUNDAMENTOS TEORICOS DE LOGICA DIFUSA (FUZZY LOGIC)
Y SU APLICACION A SISTEMAS ELECTRONICOS

VICTOR ADOLFO CHANG CHANG


Trabajo de investigación presentado para optar
al grado académico de

LICENCIADO EN INGENIERIA ELECTRONICA

Guatemala

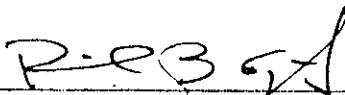
1993

vo. fo. :


(f) 

Doctor Raúl B. González
Asesor

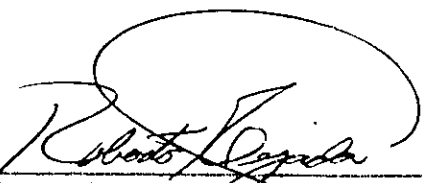
Tribunal:

(f) 

Dr. Raúl B. González

(f) 

Ingeniero Roberto Duke

(f) 

Ingeniero Roberto Tejada

Fecha de aprobación: 21 de julio de 1993

3.	Transitividad	25
4.	Tipos de relaciones	27
F.	Homomorfismo	29
G.	Lógica Difusa	30
1.	Proposiciones	30
2.	Predicados	31
3.	Cuantificadores	31
4.	Conectivos	32
5.	Simbología	33
6.	Reglas de inferencia	34
III.	IMPLEMENTACION EN SISTEMAS REALES	41
A.	Estructura general de un sistema difuso	41
1.	Variables de entrada	43
2.	Reglas de actuación	44
3.	Variables de salida	45
4.	Sistemas adaptativos	49
B.	Redes Neurales	50
1.	Neurona	50
2.	Topologías	54
3.	Leyes de Aprendizaje	56
4.	Aprendizaje no supervisado	57
5.	Aprendizaje supervisado	58
6.	Utilidad de las redes neurales	59
C.	Memorias difusas asociativas	60
1.	Concepto general	61

CONTENIDO

	Páginas
I. INTRODUCCION	1
II. FUNDAMENTOS TEORICOS	5
A. Antecedentes históricos	5
B. Nociones fundamentales de conjuntos difusos	12
1. Conjunto difuso	12
2. Función de membresía	12
3. Cardinalidad	14
4. Variable lingüística	14
C. Operaciones básicas en conjuntos	16
1. Complemento	16
2. Unión	17
3. Intersección	18
4. Operaciones de agregación	18
D. Relaciones	20
1. Producto cartesiano	20
2. Relación difusa	20
3. Relación binaria	20
4. Inversa	21
5. Composición	21
E. Relaciones binarias en un mismo conjunto	24
1. Reflexividad	24
2. Simetría	25

2.	Codificación	62
3.	Sistemas	64
4.	Memorias adaptativas	66
IV.	APLICACIONES	71
A.	Codificación de imágenes de video	71
1.	Introducción	71
2.	Codificación utilizando lógica difusa	79
B.	Metereología	87
C.	Reconocimiento de patrones	90
D.	Productos comerciales	94
E.	Sistemas expertos	98
F.	Sistemas de control	102
G.	Medicina	108
1.	Diagnóstico	108
2.	Otras aplicaciones	112
H.	Implementación en sistemas sencillos	114
I.	Avances recientes	129
V.	BIBLIOGRAFIA	131
	APENDICES	133
A.	Números difusos	133
B.	Características técnicas del decodificador de video	135

LISTA DE TABLAS Y FIGURAS

Tabla	Página
2.1 Primitivas de algunas lógicas trivalentes	8
Figura	
3.1 Estructura típica de un sistema convencional	42
3.2 Estructura típica de un sistema difuso	42
3.3 Ejemplo de proceso difuso	47
4.1 Diagrama de bloques / codificación DCT	75
4.2a-d Mapas de cuantificación	76-77
4.3 Base de información / codificación difusa	80
4.4a-c Imagen Lenna	83-84
4.5a-c Imagen F-14	85-86
4.6 Tipos de cromosomas	93
4.7 Patrón idealizado de clasificación	93
4.8 Inversor difuso	118
4.9 AND difuso	119
4.10 OR difuso	120
4.11 Implementación de función de membresía	121
4.12 Sistema difuso digital	128

I. INTRODUCCION

El concepto de Lógica Difusa (Fuzzy Logic) se originó como una metodología de control digital, combinando elementos de la lógica multivaluada, teoría de probabilidades, inteligencia artificial y redes neurales. En esencia, la lógica difusa trata de simular la forma de pensar del cerebro humano, incorporando la imprecisión inherente en todo sistema físico. Mientras que en la lógica tradicional la pertenencia a un conjunto se toma como una variable con únicamente dos posibles estados (es decir, pertenece o no pertenece), la lógica difusa permite que un elemento dado pueda pertenecer a dos conjuntos que normalmente serían disjuntos. Por ejemplo, un controlador de temperatura ambiental puede interpretar 35°C como "caliente", a la vez que toma 34.9°C como "frío". Estos cambios abruptos suelen provocar inestabilidad en los sistemas de control. La lógica difusa, por el contrario, acepta un número de variables distintas con grados de membresía diversos. En este caso, en vez de tener "frío/caliente" se puede tener "frío /tibio /cálido /caliente". Una temperatura de 15°C puede pertenecer tanto a la categoría "frío", como a la de "tibio". Dependiendo de la aplicación, esta tendrá un grado de membresía distinto en cada categoría.

Básicamente, los sistemas de tipo difuso (fuzzy) toman decisiones con base en entradas que consisten en variables lingüísticas. Estas se procesan por un conjunto de reglas de SI-ENTONCES (Por ejemplo: SI la temperatura es "caliente" ENTONCES prenda el ventilador). En este aspecto, la lógica difusa realiza lo que originalmente se pensaba que haría la inteligencia artificial (AI, por sus siglas en inglés).

En general, la lógica difusa tiene su aplicación en sistemas no lineales, variables en el tiempo y vagamente definidos. Si se tiene un sistema a controlar que está perfectamente descrito por una o varias ecuaciones diferenciales, el camino más sencillo a seguir será la utilización de técnicas de control clásico (basadas en retroalimentaciones proporcionales, integrales y derivadas). Sin embargo, si se tiene un sistema muy complejo, en donde no es fácil reducirlo a un conjunto pequeño de variables, o si la relación entre estas variables no es del todo clara, la lógica en mención suele tener ventajas enormes sobre las técnicas clásicas. A continuación se mencionan algunas de estas ventajas.

A. Procesamiento más sencillo

Debido a que se tiene un número reducido de valores para las variables lingüísticas en vez un rango continuo de valores, el controlador sólo necesita unas cuantas reglas

para procesar los datos y tomar una decisión.

B. Las variables son observables

En la lógica difusa se trabaja únicamente con variables observables (que no siempre son todas las variables medibles), por lo que el modelado de un sistema suele ser más sencillo. Por lo general se estudia la relación entre variables de entrada vrs. variables de salida, agrupando así muchos parámetros internos entre sí. Esto quiere decir que se tienen variables de orden mayor, lo cual conduce a un sistema de orden mayor. Este tipo de sistema es generalmente más estable y exacto.

C. Simplicidad y eficiencia

Debido a que en general se requiere menor cantidad de reglas de mando o software (típicamente se ha observado una reducción de 10:1 en cantidad de software respecto de aplicaciones industriales), el tiempo de diseño se reduce. Además, el período de pruebas se hace más corto, pues todas las variables a controlar son directamente observables.

Nota: Hasta el momento de realizar este trabajo no se cuenta con un consenso acerca de la traducción de el término Fuzzy Logic al español. Literalmente se traduce a "lógica borrosa" o "lógica granulada". Según sus bases teóricas, sin embargo, es más correcto hablar de una "lógica difusa". Este término se emplea en el texto Principios de Inteligencia Artificial y Sistemas Expertos (McGraw-Hill, 1992), cuyo autor es D.W. Rolston para el original en inglés. La obra es traducida al español por el Ing. Alfonso Pérez Gama. En este trabajo se utilizará el término 'fuzzy' entre paréntesis cuando se necesite especificar. Otros términos traducidos del inglés también contendrán el vocablo original entre paréntesis.

II. FUNDAMENTOS TEORICOS

A. Antecedentes históricos

La lógica tradicional de dos valores (verdadero y falso) o bivalente ha evolucionado a lo largo de la historia de la humanidad como la única concepción racional y aceptable. Los primeros pasos hacia la ruptura de este esquema de pensamiento los dio la lógica intuicionista. Esta corriente filosófica se originó entre algunos matemáticos de principios de siglo como respuesta a las exageraciones respecto del empleo del formalismo en la lógica. A la vez, era una tentativa de rescatar a la matemática del naufragio que le acechaba con los descubrimientos de paradojas lógicas irresolubles en el marco bivalente clásico. Entre los principales exponentes de este movimiento se tienen a H. Poincaré, H. Weyl, Brouwer y Heyting. Es entonces que por primera vez se considera la posibilidad de un número de valores de verdad mayor que dos. Trabajos posteriores realizados por Jan Lukasiewicz y Emil Post, que parten, respectivamente, de hipótesis trivalentes y n-valentes, desarrollaron el campo y constituyen las referencias obligadas al hablar del origen de la lógica multivaluada.

Formalmente, se habla de una concepción bivalente de la lógica, la cual se basa en el siguiente principio (llamado

6

semántico o de bivalencia):

" A toda proposición se le puede atribuir uno de dos valores: verdadero o (excluyentemente) falso.",

en donde se entiende por proposición a la unidad estructural lingüística fundamental de la lógica. El anterior principio se puede expresar equivalentemente usando el lenguaje de la teoría de conjuntos:

"Sea A el conjunto de todas las proposiciones. Existen B y C tales que:

$\emptyset \neq B \subset A, \emptyset \neq C \subset A, B \cap C = \emptyset, B \cup C = A$ ",
donde \subset indica contención. En otras palabras, A tiene una partición en dos subconjuntos propios no vacíos. Un subconjunto es el de las proposiciones con valor de verdadero y el otro es el de aquellas que tienen el valor de falso.

El problema surge con proposiciones cuyo valor es indecidible según la concepción bivalente. Por ejemplo: "Mañana lloverá", "Esta oración es falsa", etc. no pueden clasificarse simplemente como verdaderas o falsas. De ahí que surgiera la necesidad de las lógicas polivalentes, las cuales extienden el conjunto de valores lógicos posibles que toma una proposición.

El paso más evidente hacia las lógicas multivaluadas es el de extender la lógica bivalente a una de tres valores. En

todas estas lógicas, es común denotar la veracidad, la falsedad y la indeterminación por los valores $1, 0, \frac{1}{2}$, respectivamente. Además, la negación de una proposición A cualquiera, con valor de verdad $t(A)$ se define como:

$$t(\text{NOT}(A)) = 1 - t(A). \quad (2.1)$$

donde el operador NOT denota negación. El resto de conectivos primitivos se definen de forma diferente en cada tipo de lógica. En la tabla 1 se muestran los valores de algunas primitivas ($+$, $*$, \Rightarrow , \Leftrightarrow) que corresponden a los conectivos Y/O/IMPLICACION/DOBLE IMPLICACION) para 4 de las lógicas trivalentes más conocidas, identificadas por el apellido de su creador.

Tabla 2.1.

Primitivas de algunas lógicas trivalentes

a b	Lukasiewicz				Bochvar				Kleene				Heyting			
	+	*	=>	<=>	+	*	=>	<=>	+	*	=>	<=>	+	*	=>	<=>
0 0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0 ½	0	½	1	½	½	½	½	½	0	½	1	½	0	½	1	0
0 1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
½ 0	0	½	½	½	½	½	½	½	0	½	½	½	0	½	0	0
½ ½	½	½	1	1	½	½	½	½	½	½	½	½	½	½	1	1
½ 1	½	1	1	½	½	½	½	½	½	1	1	½	½	1	1	½
1 0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
1 ½	½	1	½	½	½	½	½	½	½	1	½	½	½	1	½	½
1 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Es importante notar que al eliminar el valor intermedio, todas las lógicas se ajustan a las definiciones clásicas de la lógica bivalente. Esto es una característica importante y necesaria, debido a que se está haciendo una extensión de la lógica clásica; no se está cambiando ésta. Sin embargo, existen "leyes" de la lógica bivalente que ya no se cumplen en ninguna de estas nuevas lógicas. Específicamente, se dejan de cumplir:

-la ley de contradicción: $t[A \text{ AND NOT}(A)] = 0$ (2.2)

-la ley del medio excluido: $t[A \text{ OR NOT}(A)] = 1$ (2.3)

, donde AND representa al conectivo lógico Y (conjunción) y OR (disyunción) representa el conectivo lógico O.

En general, al tener un número de valores lógicos mayor que dos, estas leyes no se han de cumplir. Se puede ver claramente el porqué de esto al hacer una analogía con la teoría de conjuntos. En lógicas no bivalentes esto quiere decir que el conjunto de proposiciones se divide en más de dos subconjuntos, los cuales no necesariamente son disjuntos.

Habiéndose establecido lógicas de tres valores, era de esperarse que se extendiera el estudio a lógicas n-valentes, para valores arbitrarios de $n \geq 2$. En los años treinta de este siglo se desarrollaron algunas lógicas de este tipo. Para cualquier n, los valores de verdad corresponden a números racionales distribuidos uniformemente en el intervalo cerrado $[0,1]$. De esta manera, el conjunto T_n de valores de verdad de una lógica n-valente es:

$$T_n = \{ 0, 1/(n-1), 2/(n-1), \dots, (n-2)/(n-1), 1 \} \quad (2.4)$$

Lukasiewicz fue el primero en extender el rango de valores de verdad hasta incluir todos los números racionales

en $[0,1]$ y, finalmente, todos los reales en ese intervalo.

En la literatura especializada, una lógica n -valente se denota con el símbolo L_n , donde n es la cardinalidad del conjunto de valores de verdad. En este caso, la lógica bivalente es la L_2 ; la lógica de Lukasiewicz que incluye a todos los racionales en $[0,1]$ se denota L_ω . La lógica que incluye todos los valores reales en $[0,1]$ se denota L_1 debido a que la cardinalidad del continuum es Aleph-1 . En general, a cualquier lógica del tipo L_1 se le conoce como lógica difusa (fuzzy) o lógica continua. En este tipo de lógica se emplea el operador generalizado t , el cual está definido así:

$$t: \{\text{Proposiciones}\} \rightarrow [0,1], \quad (2.5)$$

y es el que asigna un valor de verdad real entre 0 y 1 a cualquier proposición lógica.

En la lógica L_1 de Lukasiewicz, él definió los valores de verdad para los siguientes conectores lógicos :

$$t(A \text{ AND } B) = \min(t(A), t(B)) \quad (2.6)$$

$$t(A \text{ OR } B) = \max(t(A), t(B)) \quad (2.7)$$

$$t(\text{NOT}(A)) = 1 - t(A) , \quad (2.8)$$

donde A y B representan dos proposiciones.

Sustituyendo para los valores discretos 0 y 1, se puede

comprobar que los conectivos cumplen con las relaciones clásicas de la lógica bivalente.

Hacia 1937, el filósofo Max Black aplicó esta lógica continua a la teoría de conjuntos, al tratar de explicar los recientes descubrimientos de la mecánica cuántica. Cada elemento en los conjuntos de Black se comportaba análogamente a una proposición en una lógica de tipo continuo.

Luego de este período de intenso trabajo matemático, el tema de lógica multivaluadas no tuvo mayor desarrollo. No fue sino hasta en 1965 que el Dr. Lotfi A. Zadeh publicó el artículo "Fuzzy Sets", en donde desarrolló formalmente una teoría de conjuntos en el ambiente de una lógica polivalente. Fue él quien introdujo el término 'fuzzy' a la literatura técnica. Con este artículo, se generó un nuevo interés en estructuras matemáticas polivalentes. La reciente aparición de productos electrónicos de consumo e industriales basados en esta teoría ha provocado que nuevamente se tome interés por el tema.

B. Nociones fundamentales de conjuntos difusos

El marco teórico de la lógica difusa se origina con una teoría de conjuntos difusos (fuzzy). Esta constituye una extensión de la teoría clásica de conjuntos. En otras palabras, si tomamos la restricción de esta nueva teoría a las condiciones de la teoría clásica, regresamos a ésta.

1. Conjunto difuso. La teoría de conjuntos difusos (fuzzy sets) trata de generalizar la noción de conjunto. En un conjunto no difuso, elementos del universo cumplen con una de dos condiciones: pertenecen o no al conjunto en cuestión. En un conjunto difuso, la pertenencia puede tomar además de estos dos valores, cualquier otro valor intermedio. Así, cada elemento puede tener distinto grado de membresía en este conjunto.

2. Función de membresía. Para denotar formalmente la pertenencia de un elemento a un conjunto difuso, se introduce el concepto de la función de membresía. Esta función se define así:

$$\mu_A : X \rightarrow L, A \subset X, \quad (2.9)$$

donde X es el conjunto universo, A es el conjunto difuso en cuestión y L es un conjunto cualquiera que esté parcialmente

ordenado. Por conveniencia el conjunto L suele ser el rango de valores reales entre 0 y 1 , inclusive (denotado $[0,1]$). Esto permite que la función de membresía pueda ser tratada como una función en los reales. Los límites de este rango corresponden al caso especial de conjuntos no difusos. En este caso, si $\mu_A(x_1)=1$ (es decir, la función de membresía de x_1 en A es igual a 1), el elemento x_1 pertenece al conjunto, mientras que si $\mu_A(x_1)=0$, el elemento no pertenece al conjunto. La posibilidad de que la función de membresía tome valores reales entre 0 y 1 permite que cada elemento tenga un grado de membresía particular en el conjunto. Como se puede ver, la función de membresía simplemente matematiza el concepto de pertenencia y lo extiende para el caso de conjuntos difusos.

Recordando las restricciones que toma el conjunto L , podemos observar que la función de membresía es válida en los conjuntos no difusos, pues en ese caso $L=\{0,1\}$, el cual también es parcialmente ordenado (de hecho es totalmente ordenado) bajo la relación "menor o igual que". El caso en que $L=\{\text{mucho, mas o menos, poco}\}$ también es válido, pues existe un tipo de orden en el conjunto. Aquí, la función de membresía indica en términos lingüísticos la pertenencia al conjunto.

Al emplear $L=[0,1]$, se suele confundir la función de membresía de un conjunto con una probabilidad. A pesar de

que suelen haber analogías, ambos conceptos son totalmente independientes. Por ejemplo, mientras que la suma de probabilidades en el caso discreto (y la integral en el caso continuo) ha de ser igual a la unidad, este requerimiento no aparece para las funciones de membresía.

3. Cardinalidad. En la teoría clásica de conjuntos, la cardinalidad se refiere a la cantidad de elementos en un conjunto. Para un conjunto difuso A en un universo X , se define una cardinalidad escalar (denotada con barras) que consiste en la suma de los valores de la función de membresía de todos los elementos de X en A . Así:

$$A \rightarrow |A| = \sum_{x \in X} \mu_A(x), \quad (2.10)$$

donde R es el conjunto de los reales positivos.

4. Variable lingüística. En la matemática y ciencia basada en la lógica formal se manejan variables, las cuales toman valores numéricos para su fácil manejo. Una variable lingüística es aquella que en vez de tomar valores numéricos, toma valores lingüísticos. Por ejemplo, cuando se maneja la variable 'temperatura' ésta puede tomar un conjunto de valores reales infinito. La mente humana no maneja este conjunto de valores. Esta lo que hace es asignar un nombre

particular a un subconjunto del rango de valores de temperatura. Por ejemplo, en el caso de la variable temperatura, ésta puede tomar los valores 'caliente', 'tibio' y 'frío', en vez de un número real. Esta simplificación permite una facilidad de manejo en casos en que no sea necesaria una gran exactitud.

Más formalmente, una variable lingüística se caracteriza por una quinteta $(x, T(x), U, G, M)$ donde x es el nombre de la variable; $T(x)$ es el conjunto de términos que sirven de valores para x ; U es el universo de estos valores; G es una regla sintáctica (gramática) que genera los nombres y M es una regla semántica que asocia a cada x su significado. En el ejemplo, x es la temperatura, $T(x)$ es el conjunto {'frío', 'tibio', 'caliente'}, U es el conjunto de palabras en español, y G y M corresponden a las reglas del idioma español.

C. Operaciones básicas en conjuntos

1. Complemento. El complemento de una función de membresía se define como una función

$$c: [0,1] \rightarrow [0,1] \quad (2.11)$$

que asigna un valor $c(\mu_A(x))$ a la función de membresía $\mu_A(x)$. Esta función indica el valor de membresía de un elemento x en el conjunto que consiste en la negación del concepto representado por A . Para que una función pueda considerarse un complemento difuso, ésta ha de cumplir las siguientes condiciones:

- a. $c(0)=1$ y $c(1)=0$. En otras palabras, c es una extensión del complemento bivalente (negación).
- b. Para todo $a, b \in [0,1]$, si $a < b$ entonces $c(a) \geq c(b)$. Esto quiere decir que c es monótona no creciente.

Para propósitos prácticos, se suele requerir que se cumplan, además, estas dos condiciones:

- c. c es una función continua.
- d. c es involutiva, es decir $c(c(a)) = a$ para todo $a \in [0,1]$.

La función complemento más sencilla y a la vez la más utilizada es:

$$c(x) = 1 - x \quad (2.12)$$

Se puede observar que esta función cumple con todos los axiomas explicados anteriormente y además es análoga a la negación lógica explicada en la sección II.A.

2. Unión. La unión de dos conjuntos difusos es definida por una función de la forma:

$$u: [0,1] \times [0,1] \rightarrow [0,1], \quad (2.13)$$

la cual, para cada elemento x en el universo, toma como argumento el valor de membresía en dos conjuntos A y B y retorna el valor de la membresía del elemento en el conjunto formado por la unión de A y B .

La función u ha de satisfacer las siguientes condiciones:

- a. $u(0,0) = 0$, $u(0,1) = u(1,0) = u(1,1) = 1$; ha de ser una extensión de la unión bivalente.
- b. $u(a,b) = u(b,a)$; ha de ser conmutativa.
- c. si $a \leq a'$ y $b \leq b'$ entonces $u(a,b) \leq u(a',b')$; ha de ser monótona.
- d. $u(u(a,b),c) = u(a,u(b,c))$; ha de ser asociativa.

Además, es deseable que se cumplan:

- e. u es una función continua.
- f. $u(a,a) = a$; es decir, u es idempotente.

La función original para la unión propuesta por Lotfi Zadeh es:

$$u(a,b) = \max\{a,b\} \quad (2.14)$$

3. Intersección. La intersección se especifica por una función:

$$i: [0,1] \times [0,1] \rightarrow [0,1] \quad (2.15)$$

El argumento de la función es el par consistente en las funciones de membresía de un elemento x en dos conjuntos cualquiera. El valor que devuelve la función es la membresía del elemento en la intersección de ambos conjuntos. Al igual que en la unión, la intersección ha de cumplir ciertas condiciones fundamentales:

- a. $i(1,1) = 1, i(0,1) = i(1,0) = i(0,0) = 0$.
- b. $i(a,b) = i(b,a)$.
- c. si $a \leq a'$ y $b \leq b'$ entonces $i(a,b) \leq i(a',b')$.
- d. $i(i(a,b),c) = i(a,i(b,c))$.

y otros deseables:

- e. i es continua
- f. $i(a,a) = a$

La función de intersección original es:

$$i(a,b) = \min[a,b] \quad (2.16)$$

4. Operaciones de agregación

Las operaciones de agregación tienen la forma

$$h: [0,1]^n \rightarrow [0,1] , \quad (2.17)$$

para algún $n \geq 2$. Estas operaciones combinan varios conjuntos difusos y los transforman en un nuevo conjunto difuso. Al

igual que en la unión e intersección, la operación de agregación actúa sobre las funciones de membresía de cada conjunto. Así, si la operación h actúa sobre A_1, A_2, \dots, A_n para obtener el conjunto A , tenemos:

$$\mu_A(x) = h(\mu_{A_1}(x), \mu_{A_2}(x), \dots, \mu_{A_n}(x)) , \quad (2.18)$$

para todo $x \in X$.

Las operaciones de agregación han de cumplir los siguientes axiomas:

- a. $h(0,0,\dots,0)=0$ y $h(1,1,\dots,1)=1$.
- b. h es monótona no decreciente.

Es usual que también se requiera que la función cumpla con:

- c. h sea una función continua.
- d. h sea simétrica para todos sus argumentos, i.e.:
 $h(a_i | i=1, \dots, n) = h(a_{p(i)} | i=1, \dots, n)$, donde $p(i)$
 es una permutación de $\{1, 2, \dots, n\}$.

De las definiciones anteriores es observable que la unión e intersección son un caso especial de operaciones de agregación. Aunque estas se definen para dos argumentos, se pueden extender a cualquier número debido a la condición que garantiza la asociatividad de las operaciones. Se les menciona por separado debido a que la generalización en operaciones de agregación no es usual en la teoría clásica de conjuntos.

D. Relaciones

Una relación representa la presencia o ausencia de asociación, interacción o interconexión entre los elementos de dos o más conjuntos. Este concepto se puede generalizar de forma que existan grados de relación o interacción entre elementos.

1. Producto cartesiano. El producto cartesiano de una colección de conjuntos no difusos $\{X_1, X_2, \dots, X_n\}$ se representa por $X_1 \times X_2 \times \dots \times X_n$. Los elementos del producto cartesiano de n conjuntos no difusos son n -adas (x_1, x_2, \dots, x_n) tal que $x_i \in X_i$ para $i=1, 2, \dots, n$.

Una relación entre conjuntos no difusos X_1, X_2, \dots, X_n es un subconjunto de su producto cartesiano y se simboliza como $R(X_1, X_2, \dots, X_n)$.

2. Relación difusa (fuzzy). Una relación difusa es un conjunto difuso definido en el producto cartesiano de n conjuntos X_1, X_2, \dots, X_n donde las n -adas (x_1, x_2, \dots, x_n) tienen c/u su grado de membresía dentro de la relación, denotado por $\mu_R(x_1, x_2, \dots, x_n)$, el cual es un número real en el intervalo cerrado $[\emptyset, 1]$.

3. Relación binaria. Una relación definida en dos conjuntos X y Y se conoce como relación binaria y se

representa por $R(X,Y)$. Cuando la relación binaria es difusa, su dominio es el conjunto difuso $\text{dom } R(X,Y)$, cuya función de membresía se define por:

$$\mu_{\text{dom } R}(x) = \max_{y \in Y} \mu_{R(x,y)} \quad (2.19)$$

para todo $x \in X$. El rango de una relación difusa se denota por $\text{ran } R(X,Y)$ y está definido por:

$$\mu_{\text{ran } R}(y) = \max_{x \in X} \mu_{R(x,y)} \quad (2.20)$$

Si cada miembro del dominio de una relación binaria aparece exactamente una vez en R , la relación se llama un mapeo o función. Si $R(X,Y)$ es una función y $\mu_{R(x,y)}(x,y) > 0$ entonces a y se le denomina la imagen de x .

4. Inversa. La inversa de una relación difusa $R(X,Y)$ se denota por $R^{-1}(Y,X)$ y se define por:

$$\mu_{R^{-1}(Y,X)} = \mu_{R(X,Y)}, \quad (2.21)$$

para todo $(x,y) \in X \times Y$.

5. Composición. La composición de dos relaciones $P(X,Y)$ y $Q(Y,Z)$ se denota por:

$$R(X,Z) = P(X,Y) \cdot Q(Y,Z) \quad (2.22)$$

y ha de cumplir las siguientes propiedades:

- a. $P \circ Q$, en general, no es igual a $Q \circ P$
- b. $(P \circ Q)^{-1} = Q^{-1} \circ P^{-1}$
- c. $(P \circ Q) \circ R = P \circ (Q \circ R)$

Al igual que en las operaciones entre conjuntos, se pueden definir composición entre relaciones de distintas formas, siempre que se cumplan las condiciones establecidas. Por ejemplo (suponiendo $R(x,z) = P(x,y) \circ Q(y,z)$), la composición max-min se define por:

$$\mu_R(x,z) = \max_{y \in Y} \min[\mu_P(x,y), \mu_Q(y,z)] \quad (2.23)$$

Otra forma de definir la composición es la max-producto, en la cual:

$$\mu_R(x,z) = \max_{y \in Y} [\mu_P(x,y) * \mu_Q(y,z)] \quad (2.24)$$

Se puede comprobar que ambos tipos de composición satisfacen las tres reglas enunciadas anteriormente.

En general, una relación difusa (fuzzy) en dos conjuntos finitos se puede representar por medio de una matriz, de forma que las filas correspondan a los elementos del primer conjunto y las columnas a los del segundo. Los elementos de la matriz son los grados de membresía de la pareja en la

relación. Esta representación matricial permite que operaciones como la composición de relaciones se efectúe de una manera organizada y con notación definida.

Por ejemplo, sea R una relación entre dos conjuntos dados $X = \{\text{Nueva York, París}\}$ y $Y = \{\text{Honolulu, Nueva York, Los Angeles}\}$ R representa el concepto relacional "muy lejos de". Esta relación se puede representar por la matriz siguiente:

	Nueva York	París
Honolulu	0.7	1
Nueva York	0	0.5
Los Angeles	0.4	0.8

donde cada elemento indica el grado de membresía de la pareja correspondiente en el concepto relacional. Así, Honolulu está definitivamente muy lejos de París, Nueva York definitivamente no está muy lejos de sí misma, Los Angeles no está tan lejos de Nueva York (en este contexto), etc.

E. Relaciones binarias en un mismo conjunto

El caso especial de relaciones binarias en un mismo conjunto se estudia por separado debido a su importancia y aplicabilidad en funciones o mapeos, ordenamientos, etc.

Las relaciones binarias $R(X,X)$ se conocen como grafos dirigidos, debido a que se pueden representar por un grafo sencillo. En este grafo:

- 1) cada elemento del conjunto X se representa por un nodo único,
- 2) Conexiones directas entre nodos indican elementos relacionados con grado de membresía no nulo,
- 3) Cada conexión en el diagrama tiene indicado el valor de membresía de cada par.

Los diferentes tipos de relaciones binarias de la forma $R(X,X)$ se distinguen de acuerdo a si cumplen alguna o algunas de tres propiedades fundamentales: reflexividad, simetría y transitividad. Esto permite la clasificación de las relaciones como de similaridad, compatibilidad u orden.

1. Reflexividad. Para todo $x \in X$, $\mu_R(x,x) = 1$

Si esta propiedad no se cumple para algún x , la relación se conoce como irreflexiva; si no se cumple para algún x se conoce como antirreflexiva.

2. Simetría. Para todo $x, y \in X$, $\mu_R(x, y) = \mu_R(y, x)$

Si esta propiedad no se cumple para algún par $x, y \in X$ distintos, la relación se conoce como asimétrica. Si además se tiene que $\mu_R(x, y) > 0$ y $\mu_R(y, x) > 0$ implica $x=y$ para todo $x, y \in X$, la relación se conoce como antisimétrica.

3. Transitividad. Una relación difusa se conoce como transitiva (o más específicamente max-min transitiva) si y solo si:

$$\mu_R(x, z) \geq \max_{y \in X} \min [\mu_R(x, y), \mu_R(y, z)] \quad (2.25)$$

se satisface para cada par $(x, z) \in X^2$.

Una relación que no cumpla la anterior desigualdad se llama no transitiva. Si además se tiene que

$$\mu_R(x, z) < \max_{y \in X} \min [\mu_R(x, y), \mu_R(y, z)] \quad (2.26)$$

para todo $(x, z) \in X^2$, la relación se llama antitransitiva.

La transitividad en relaciones difusas, al igual que la composición, se puede definir de distintas maneras. La única condición es que al tomar las funciones de membresía bivaluadas, se regrese a la definición clásica de transitividad. Por ejemplo, otra definición aceptable es la transitividad max-producto, en donde

$$\mu_R(x,z) \geq \max_{y \in X} [\mu_R(x,y) * \mu_R(y,z)] \quad (2.27)$$

para todo $(x,z) \in X^2$.

a. Clausura transitiva. La clausura transitiva de una relación difusa $R(X,X)$ se define como aquella relación $R_T(X,X)$ que es transitiva, contiene a $R(X,X)$ y sus elementos tienen los grados de membresía mínimos que permiten que se cumplan las dos condiciones anteriores. Para determinar la clausura transitiva $R_T(X,X)$ de una relación $R(X,X)$ se emplea el siguiente algoritmo:

- *1. $R' = R \cup (R \circ R)$
- *2. si $R' \neq R$ entonces haga $R = R'$ y repita el paso 1.
- *3. Fin del algoritmo: $R_T = R'$.

Es importante recalcar que la composición y unión empleadas han de ser compatibles con la definición de transitividad utilizada. Por ejemplo, si se usa la definición max-min de transitividad, se ha de utilizar la composición max-min y el operador max para la unión.

4. Tipos de relaciones.

a. Relación de similaridad. Una relación binaria $R(X,X)$ difusa que sea reflexiva, simétrica y transitiva se conoce como relación de similaridad. En el caso especial de relaciones no difusas, la relación se conoce como de equivalencia.

b. Relación de Proximidad. Una relación binaria $R(X,X)$ difusa que es reflexiva y simétrica se conoce como una relación de proximidad. El análogo en relaciones no difusas se conoce como relaciones de compatibilidad o de tolerancia.

c. Relación de Orden. Una relación binaria $R(X,X)$ difusa que es reflexiva, antisimétrica y transitiva (de cualquier tipo) se llama un ordenamiento parcial difuso.

Nuevamente, los conceptos esbozados con anterioridad son simplemente una generalización de los ya conocidos para relaciones no difusas. La relación de similaridad se interpreta como lazo de similitud cuantificado por la función de membresía. Obviamente, al ser el grado de membresía igual a la unidad se tiene la relación de equivalencia. De esta

analogía se desprende que una relación de similaridad induce en el conjunto una partición en clases de similaridad. A diferencia del caso no difuso (non-fuzzy) estas clases constituyen conjuntos que en general no son disjuntos.

Las relaciones de orden permiten organizar los diferentes elementos de un conjunto en arreglos definidos. Esto permite que, por ejemplo, las variables lingüísticas puedan ser valuadas al igual que variables numéricas, pues se tiene un orden en el conjunto de valores posibles.

F. Homomorfismo

Supongamos dos relaciones binarias difusas $R(X,X)$ y $Q(Y,Y)$. Una función $h:X \rightarrow Y$ se llama un homomorfismo de (X,R) a (Y,Q) si

$$\mu_R(x_1, x_2) \leq \mu_Q[h(x_1), h(x_2)] \quad (2.28)$$

para todo $x_1, x_2 \in X$.

Si la función h es además biyectiva (es decir inyectiva y sobreyectiva), se le llama un isomorfismo. Si Y está contenido en X (recuérdese que X y Y son conjuntos no difusos) entonces h se llama un endomorfismo. Una función que sea tanto un isomorfismo como un endomorfismo se conoce como un automorfismo.

El concepto fundamental de homomorfismo implica que las dos relaciones están ligadas entre sí por medio de una función (que es un caso especial de una relación). Al tener un isomorfismo, los elementos de X esencialmente son transformados a los elementos de Y manteniendo las propiedades de la relación R en la relación Q .

G. Lógica difusa (Fuzzy Logic)

Teniendo los conceptos fundamentales de la teoría de conjuntos difusos, se puede hacer una presentación más formal de la lógica difusa. Es conveniente aclarar que esta es una lógica de tipo semántico, en donde interesa el significado de los símbolos. La lógica difusa pues, trata de extender el alcance de la lógica bivalente clásica a un contexto semántico. Si bien existen analogías entre una y otra (de hecho, la lógica bivalente se considera un caso especial de la difusa), es imposible presentar esta lógica como un sistema formal, en la forma que se hace para el caso bivalente.

1. Proposiciones. En la lógica bivalente tradicional se manejan las llamadas proposiciones, las cuales son aserciones o declaraciones expresadas en algún lenguaje y cuyo valor ha de ser verdadero o falso. La proposición contiene un sujeto y un predicado. En la lógica difusa el concepto de proposición es más flexible, pues el conjunto de valores de verdad es infinito. Los valores que toma la proposición son subconjuntos difusos de este conjunto de valores. Por ejemplo, si el conjunto de valores de verdad es el rango de números reales entre 0 y 1 inclusive, el valor 'muy verdadero' se puede considerar como un subconjunto difuso del rango. Los valores cercanos a 1 tendrán un grado

de membresía alto y los valores cercanos a \emptyset tendrán un grado de membresía bajo o nulo.

2. Predicados. Los predicados de la lógica clásica (P en la proposición x es P) están limitados en el sentido de que su definición semántica ha de ser perfectamente clara, es decir no han de existir ambigüedades con respecto de su definición. Por el contrario, un predicado difuso puede tener una definición que no es del todo clara, siempre que se pueda representar como un subconjunto difuso de un universo dado. Por ejemplo 'viejo' es un predicado difuso que no sería aceptable en la lógica clásica. Sin embargo 'nonagenario' está perfectamente definido, por lo que es un predicado aceptable en la lógica clásica. Por extensión, ambos predicados son permisibles en la lógica difusa (ya que suponemos que la lógica clásica es un caso especial de la difusa).

3. Cuantificadores. El uso de cuantificadores también se extiende en la lógica difusa. Mientras que en la lógica bivalente (y la mayoría de lógica polivalentes finitas) se aceptan sólo dos cuantificadores ('para todo' y 'existe'), la lógica difusa permite una variedad de cuantificadores que caracterizan de una forma imprecisa la cardinalidad de un conjunto. Por ejemplo 'muchos'.

'algunos', 'mil y pico', etc. son cuantificadores válidos en este tipo de razonamiento. A la vez, existen los llamados modificadores difusos, los cuales corresponden a los adverbios en gramática. Por ejemplo, 'bastante', 'más o menos', 'un poco', etc. representan modificadores difusos. Estos modificadores, combinados con predicados difusos, permiten la valoración de una variable lingüística. Así, la variable lingüística 'edad' puede tomar los valores 'joven', 'bastante joven', 'viejo', 'un poco viejo', etc.

4. Conectivos. Las proposiciones se pueden combinar utilizando los conectivos lógicos. Se puede definir una función lógica que tome un número determinado de variables lógicas y les asigne un valor de verdad particular para cada combinación de valores de verdad de las variables involucradas. El total de funciones posibles depende del número de variables y del número de valores de verdad. En general, se define un conjunto de primitivas, las cuales son funciones de una o dos variables, con base en las cuales se construyen las demás. Un conjunto de primitivas se llama completo si puede generar todas las funciones de n variables, con n finito, se pueden generar con una combinación finita de estas primitivas. Es evidente por la naturaleza misma de la lógica difusa que ésta no tiene un conjunto finito de primitivas. Por esta razón, al trabajar con un número finito

de primitivas, estamos teniendo acceso solamente a un subconjunto de todas las funciones lógicas posibles. Por lo general se suele definir un conjunto de primitivas análogo a un conjunto de primitivas de lógica clásica. Esto se hace por conveniencia pues la idea es extender las posibilidades de la lógica bivalente a razonamientos no precisos.

5. Simbología. En lógica difusa, el concepto de proposición se suele interpretar como una colección de restricciones de tipo difuso aplicadas a un conjunto de variables. Debido a la naturaleza del lenguaje algunas de las restricciones y variables están implícitas en una proposición. La representación del significado de una proposición consiste entonces en un proceso que haga explícitas la restricciones y variables. Esto se logra representando a la proposición en una forma canónica:

$$Q X \text{ es } A \quad (2.29)$$

en donde Q es un cuantificador difuso, X es una colección de variables y A es un predicado difuso. Una vez se tiene esta forma canónica, se pueden trabajar silogismos, las cuales permiten deducciones lógicas. Uno de estos es el silogismo de intersección/producto, expresado:

$$\begin{array}{l} Q_1 \text{ A's son B's} \\ Q_2 \text{ (A y B)'s son C's} \end{array}$$

$$(Q_1 * Q_2) \text{ A's son (B y C)'s} \quad , \quad (2.30)$$

donde $(Q_1 * Q_2)$ representa un cuantificador obtenido con base en Q_1 y Q_2 . Por ejemplo:

$$\begin{array}{l} \text{muchos jóvenes son estudiantes} \\ \text{la mayoría de estudiantes jóvenes son solteros} \end{array}$$

$$\text{Casi todos los jóvenes son estudiantes y solteros,} \quad (2.31)$$

es una regla de inferencia admisible, con la convención de que 'Casi todos' equivale a una combinación de los cuantificadores 'muchos' y 'la mayoría'. En general, esta operación equivale a una operación de agregación entre conjuntos difusos, y se asocia a una operación aritmética difusa. De hecho, los valores de las variables lingüísticas mencionados anteriormente son ejemplos de lo que se llama un número difuso. Este se define como un conjunto difuso convexo y normal (i.e., que tiene al menos un elemento con grado de membresía unitario) y se puede visualizar como una extensión del concepto de intervalo de confianza.

6. Reglas de inferencia. En la lógica difusa, las premisas y conclusiones de una regla de inferencia se

expresan generalmente en la forma canónica. Esta representación enfatiza el hecho de que cada premisa es un tipo de restricción sobre una variable y que la conclusión es una restricción inducida que depende de las restricciones originales.

Las reglas de inferencia de la lógica difusa se pueden clasificar en reglas categóricas y reglas disposicionales. Las reglas categóricas son aquellas que no contienen cuantificadores difusos. Las reglas disposicionales son aquellas que contienen un cuantificador difuso en al menos una de sus premisas. El cuantificador puede estar contenido implícitamente en la oración.

a. principio de contención

$$\begin{array}{l} X \text{ es } A \\ A \text{ c } B \end{array}$$

$$X \text{ es } B$$

b. principio de contención disposicional

$$\begin{array}{l} \text{usualmente } X \text{ es } A \\ A \text{ c } B \end{array}$$

$$\text{usualmente } X \text{ es } B$$

c. Regla conjuntiva

$$\begin{array}{l} X \text{ es } A \\ X \text{ es } B \end{array}$$

$$X \text{ es } (A \cap B),$$

donde la intersección se define en términos del AND lógico. Así $\mu_{A \cap B}(u) = \mu_A(u) \text{ AND } \mu_B(u)$, haciendo que los valores de membresía equivalgan a valores de verdad (ver intersección de conjuntos).

d. Producto cartesiano

$$\begin{array}{l} X \text{ es } A \\ Y \text{ es } B \end{array}$$

$$(X, Y) \text{ es } (A \times B),$$

donde (X, Y) es una variable binaria y $A \times B$ es el producto cartesiano de A y B .

e. Regla de proyección

$$(X, Y) \text{ es } R$$

$$X \text{ es } \pi R,$$

donde πR es la proyección de la relación R en el dominio de X , definida por

$$\mu_{\pi R}(u) = \sup_v \mu_R(u, v). \quad (2.32)$$

f. Regla de composición

$$\begin{array}{l} X \text{ es } A \\ (X, Y) \text{ es } R \end{array}$$

$$Y \text{ es } A \cdot R ,$$

donde $A \cdot R$ es la composición de la relación unaria A con la relación binaria R , definida por:

$$\mu_{A \cdot R}(v) = \sup_u (\mu_A(u) \text{ AND } \mu_R(u)) \quad (2.33)$$

Esta es una forma alternativa de la definición de la composición max-min, enunciada en el inciso D de este capítulo.

g. Modus ponens generalizado

$$\begin{array}{l} X \text{ es } A \\ (Y \text{ es } C) \text{ si } (X \text{ es } B) \end{array}$$

$$Y \text{ es } A$$

h. Modus ponens disposicional

$$\begin{array}{l} \text{usualmente } (X \text{ es } A) \\ \text{usualmente } (Y \text{ es } B \text{ si } X \text{ es } A) \end{array}$$

$$\text{usualmente}^2 (Y \text{ es } B),$$

donde 'usualmente²' es un cuantificador que corresponde a operar el cuantificador 'usualmente' sobre sí mismo.

Aquí 'usualmente' es un número difuso y 'usualmente²' corresponde a ese número elevado al cuadrado. No interesa en este caso como se define la elevación al cuadrado más que como una modificación en la función de membresía del conjunto 'usualmente'.

i. Principio de extensión. Este principio es uno de los más importantes en la lógica difusa, pues provee el mecanismo para el cálculo restricciones inducidas. Sea U un universo del cual es parte la variable X . Esta variable está restringida por un predicado difuso en la proposición ' X es A '. Asuma que f es un mapeo de U a V , de forma que X está mapeada en $f(X)$. Entonces, de acuerdo al principio de extensión, las restricciones inducidas en $f(X)$ están dadas por $f(A)$. Formalmente:

$$\frac{X \text{ es } A}{f(X) \text{ es } f(A)}$$

donde $\mu_{f(A)}(v) = \sup_{u \in U} \mu_A(u)$ con $v = f(u)$. El símbolo \sup se refiere a la función supremo, que devuelve el valor de la mínima cota superior de un conjunto.

El principio de extensión se relaciona íntimamente con el concepto de homomorfismo expresado con anterioridad. De esta

forma, se puede ver una proposición como una relación entre un conjunto de sujetos y un conjunto de predicados. Se puede entonces hablar de proposiciones homomórficas entre sí, es decir, que son equivalentes bajo una transformación.



III. IMPLEMENTACION EN SISTEMAS

Desde el control de ciclos de calefacción en una lavadora automática hasta la codificación digital de imágenes de video, los sistemas de tipo difuso se emplean en las situaciones más variadas. Sin embargo, todos estos sistemas son básicamente similares. Se puede describir una estructura general de bloques que engloba todas las aplicaciones existentes. La diferencia entre cada aplicación la darán las reglas específicas (software) y el tipo de entradas y salidas que tenga el sistema.

A. Estructura general de un sistema difuso.

Podemos generalizar la estructura de cualquier sistema real como uno que acepta entradas y las procesa de alguna forma para obtener así salidas deseadas. Los sistemas difusos, pues, han de tener este tipo de estructura. La diferencia entre un sistema difuso y uno de tipo convencional es la forma en que procesan las señales. Para esto, un sistema difuso emplea la lógica difusa como su principal herramienta, manejando variables lingüísticas, procesando conjuntos difusos, etc. En las figuras 3.1. y 3.2. se pueden observar las estructuras típicas de un sistema de tipo convencional y uno difuso.

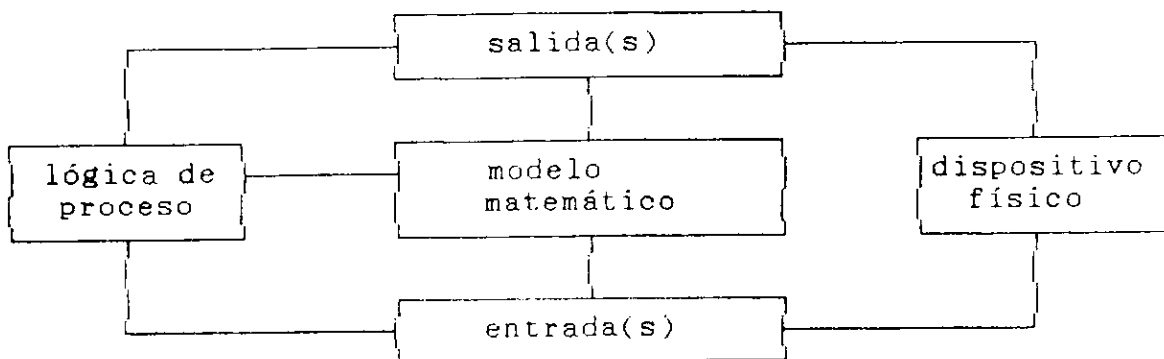


Fig. 3.1. Estructura típica de un sistema convencional

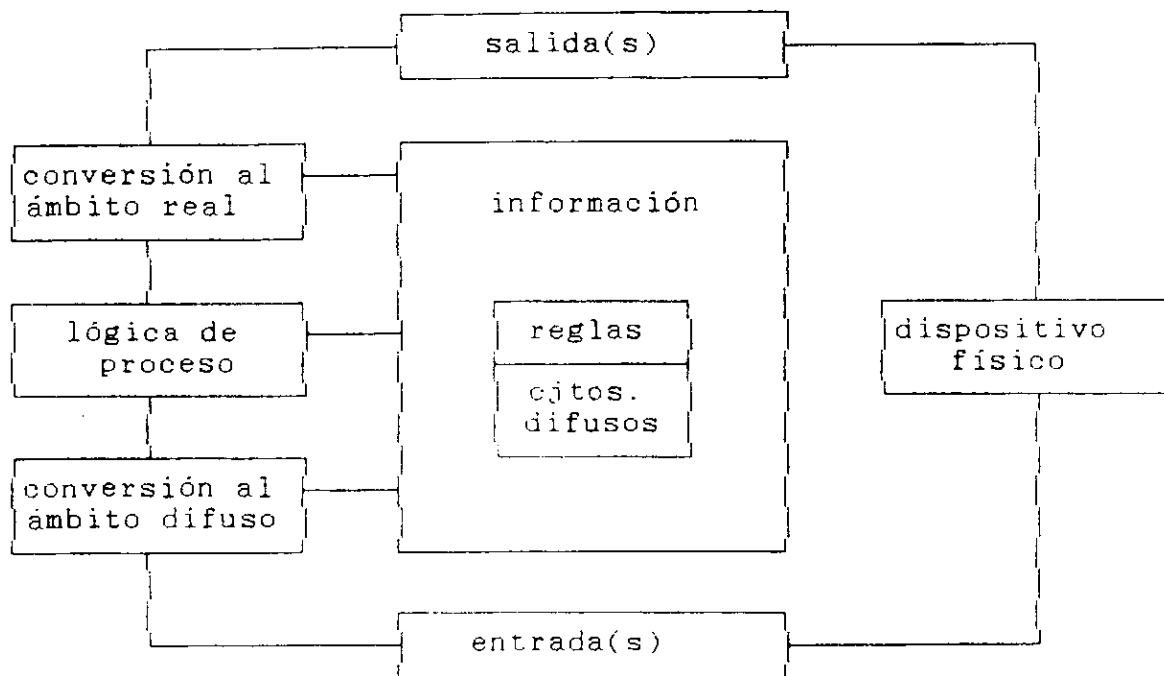


Fig. 3.2. Estructura típica de un sistema difuso.

Como se puede observar, la diferencia básica estriba en el tipo de procesamiento que sufren las entradas. En un sistema difuso, las entradas se categorizan en conjuntos difusos por medio de la conversión al ámbito difuso, o sea como valores posibles de las variables lingüísticas. Esto permite su procesamiento utilizando la lógica difusa. La salida del sistema ha de ser no difusa por lo que se ha de convertir nuevamente al ámbito convencional. Esta situación es análoga al procesamiento digital de señales. En este caso, las entradas analógicas se convierten a digital y se procesan de esta forma, pero la(s) salida(s) vuelve(n) a ser analógica(s).

Mientras que en un sistema convencional el procesamiento depende de un riguroso modelo matemático (una serie de ecuaciones diferenciales, por ejemplo), los sistemas difusos basan su procesamiento en una base de datos consistente en información y reglas de actuación.

1. VARIABLES DE ENTRADA. Las variables de entrada suelen ingresar en un formato no difuso. La labor de conversión al ámbito difuso consiste en categorizar un valor específico dentro de uno o más conjuntos difusos. Estos conjuntos son los valores que toma una variable lingüística. Por ejemplo, si tenemos una entrada de temperatura ambiental de un valor específico, digamos 15°C, la conversión al ámbito

difuso ha de asignarle un grado de membresía en conjuntos como "frío", "tibio", "caliente". El grado de membresía depende de las características que tenga cada conjunto.

Debido a que los conjuntos difusos no son disjuntos, la temperatura de 15°C es tanto "fría" como "tibia". En cada uno de estos conjuntos, la temperatura en cuestión tiene su propio grado de membresía.

La intersección entre los diversos conjuntos es siempre deseable. De hecho se puede argumentar que es imprescindible para que tenga sentido la implementación de un sistema difuso. De no ser por la existencia de estas intersecciones, se vuelve al caso de un sistema no difuso.

2. Reglas de Actuación. Una vez se tienen las entradas en un formato difuso, corresponde el procesamiento de estas para determinar una salida. Las reglas de actuación consisten en inferencias de tipo difuso que dan como resultado un conjunto difuso de salida. Por lo general estas reglas de actuación son del tipo

$$\text{SI } X \text{ es } A \text{ ENTONCES } Y \text{ es } B , \quad (3.1)$$

en donde X denota una variable de entrada, A denota un conjunto difuso que sirve como valor para una variable lingüística, Y es una variable de salida y B su valor difuso

correspondiente. Las proposiciones "X es A" y "Y es B" pueden ser compuestas. Esto quiere decir que pueden ser varias proposiciones independientes unidas por uno o más conectivos lógicos. Por ejemplo, una regla puede decir "SI la temperatura ambiente es t₁ a Y la humedad relativa es alta, entonces entregue poca potencia al calentador". La salida es evaluada en función de las entradas. Note que se ha de utilizar la operación Y (conjunción) para este proceso. La salida puede no ser única, pues más de una regla puede ser evaluada en un instante dado debido al traslape entre conjuntos. De hecho, se puede decir que todas las reglas se evalúan en un instante dado, activándose sólo las salidas de las que el antecedente tiene un valor de verdad no nulo. Además, la salida es de tipo difuso, pues "poca potencia" corresponde a un conjunto difuso de la variable lingüística "potencia".

3. Variables de Salida. Cuando se han procesado las variables de entrada con las reglas de actuación, se obtienen salidas difusas. El proceso de convertir este tipo de salida a un formato no difuso se conoce como la técnica de centroides(1). El resultado que se obtiene es una salida no difusa que el dispositivo real acepta perfectamente.

Para ejemplificar lo mencionado anteriormente, considere un controlador automático de un automóvil. Este automóvil ha

de llegar a un destino predeterminado atravesando un camino con valles y colinas. El controlador intenta optimizar el tiempo de viaje mediante el monitoreo de velocidad y distancia al destino, reaccionado acorde. Sus tres entradas son velocidad actual, aceleración y distancia al punto de destino. Supondremos que la única salida es la potencia que entrega el motor.

La figura 3.3 muestra gráficamente cómo el sistema toma las variables de entrada y las procesa de acuerdo a las reglas de actuación. En este caso supondremos que el vehículo está subiendo una colina bastante inclinada, la cual se halla relativamente cerca del destino. Las funciones de membresía asignan los valores observados a cada variable. Nótese que en general una entrada no difusa corresponde a dos posibles valores difusos. Las tres reglas que se activan son las listadas de entre las que conforman el conjunto de reglas de actuación. En todas las reglas, el conjunto de salida parcial queda acotado por el valor de verdad del antecedente. Esto se hace para mantener el valor de verdad de la implicación. En el antecedente, cuando sea necesario, se opera la disjunción con la función mínimo, propuesta por Zadeh.

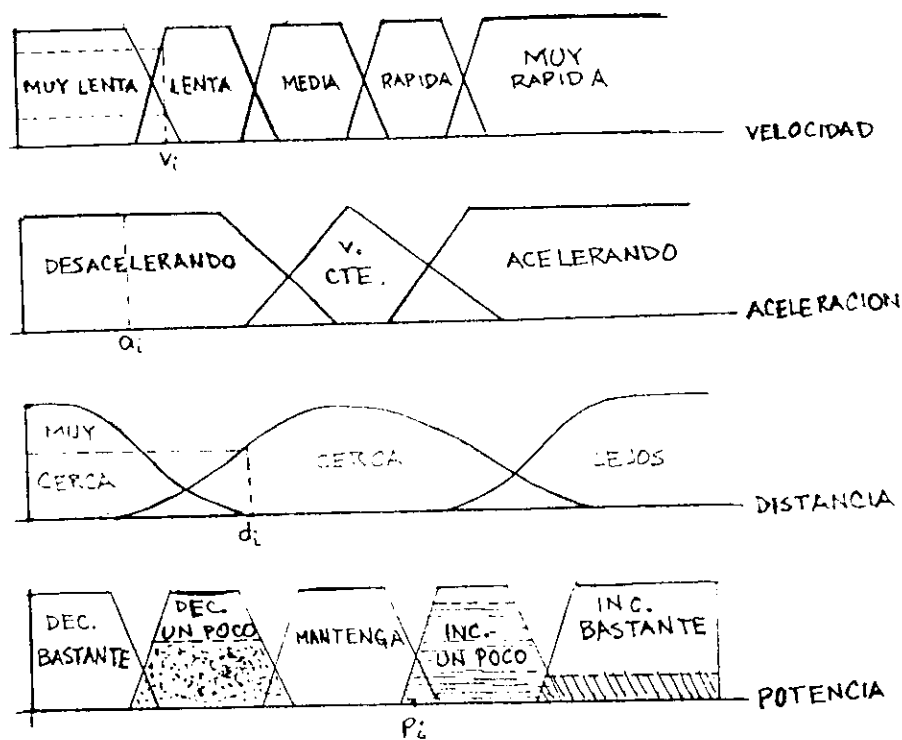


Figura 3.3.

Los valores de las variables lingüísticas se muestran gráficamente. Las reglas en cuestión son:

- * SI velocidad es MUY LENTA y aceleración es DESACELERANDO ENTONCES INCREMENTE potencia BASTANTE.
- * SI velocidad es LENTA y aceleración es DESACELERANDO ENTONCES INCREMENTE potencia UN POCO.
- * SI distancia es CERCA ENTONCES DECREMENTE potencia UN POCO.

La salida se calcula como el centroide de las áreas sombreadas.

La conversión al ámbito difuso consiste en la determinación de las funciones de membresía. Por lo general, en un gráfico de la variable vrs. el grado de membresía, las funciones de membresía tienen forma trapezoidal, como en el ejemplo anterior. Esto es para simplificar el sistema. La existencia de un traslape entre estas funciones es lo que hace que estos sistemas tengan cambios suaves y, en general, se comporten más eficientemente que los sistemas convencionales. Una regla de tipo empírico que se ha establecido es que el traslape entre conjuntos ha de corresponder a aproximadamente el 25% de ellos.

El procesamiento de las reglas de actuación o inferencias corresponde a un tipo de procesamiento en paralelo. Debido a que cada variable de entrada suele activar varias inferencias, todas estas han de ser procesadas antes de determinar la salida. La velocidad de procesamiento de un sistema se mide entonces en inferencias difusas por segundo o FLIPS (por Fuzzy Logic Inferences Per Second). En la actualidad existen microprocesadores que manejan lógica difusa y uno de sus parámetros lo constituye su velocidad en FLIPS. Togai Infralogic, una compañía estadounidense, desarrolló a finales de los años ochenta un procesador difuso digital, el FC110. Este es capaz de ejecutar 50 kiloFLIPS por segundo y básicamente es un procesador con arquitectura RISC de 8 bits que ejecuta 10 millones de

instrucciones por segundo. Las reglas de actuación y funciones de membresía se almacenan en memorias externas.

4. Sistemas adaptativos. Suele ser el caso que un sistema necesite ser "programado" una sola vez y todo su comportamiento futuro esté definido en este "programa". Sin embargo, existen sistemas difusos que necesitan adaptarse a cambios continuos en las condiciones en que funciona. Los sistemas adaptativos modifican su actuación (i.e. sus reglas o inferencias) según se necesite. Es en estos casos que se aprovechan las características de las redes neurales, las cuales se presentan a continuación.

B. Redes Neurales

En este trabajo se manejará el concepto de red neural como aquel sistema artificial que simula el comportamiento neural de un organismo biológico. Más específicamente, una red neural está constituida de neuronas agrupadas en una topología particular (es decir una forma particular de interconexión entre neuronas), con una ley de aprendizaje que permite adaptar cada neurona a condiciones específicas. A continuación se presentan los conceptos básicos de redes neurales. Esta presentación no es en ningún caso exhaustiva y sólo intenta una definición un tanto operacional del concepto, en lo que concierne a su aplicabilidad en sistemas difusos.

1. Neurona. En el contexto de redes neurales artificiales, una neurona es un elemento que se comporta como una función. Una neurona transduce una entrada no acotada $x(t)$ en el tiempo t a una salida acotada $S(x(t))$. La función S es en general monótona no decreciente, lo cual equivale a que $dS/dt \geq 0$ para todo t , si existe la derivada. Esto quiere decir que un incremento en la activación de entrada sólo puede incrementar la señal de salida o dejarla constante. En la práctica esto implica que las funciones tienen una cota superior o valor de saturación.

Una representación más específica de una neurona asume

que tiene n entradas y una salida. Esta salida ha de ser una función de la combinación lineal de sus entradas. Así, si llamamos y a la salida y x_i , $i = 1, \dots, n$ a sus entradas,

$$y = S\left(\sum_{i=1}^n w_i x_i\right) \quad (3.2)$$

La función S toma diferentes formas, algunas de las cuales tienen un nombre particular

- a. función logística. Esta es una de las más empleadas. Se define así:

$$S(x) = (1 + e^{-cx})^{-1}, \quad (3.3)$$

donde c es una constante positiva. En este caso, S es monótona creciente.

- b. función umbral. Esta equivale a una función escalón.

$$S(x_{k+1}) = \begin{cases} 1 & \text{si } x_{k+1} > T \\ S(x_k) & \text{si } x_{k+1} = T \\ \emptyset & \text{si } x_{k+1} < T \end{cases} \quad (3.4)$$

para un valor T de umbral. El subíndice k indica pasos discretos temporales. Este tipo de funciones halla

aplicación en sistemas de tipo digital.

c. función tangente hiperbólica.

$$\begin{aligned} S(x) &= \tanh(cx) \\ &= (e^{cx} - e^{-cx}) / (e^{cx} + e^{-cx}) \end{aligned} \quad (3.5)$$

Esta función es monótona no decreciente y bipolar (toma valores positivos y negativos).

d. función umbral lineal. Esta función aproxima mejor el comportamiento real de una neurona ante una función umbral:

$$S(x) = \begin{cases} 1 & \text{si } cx \geq 1 \\ 0 & \text{si } cx < 0 \\ cx & 0 \leq cx < 1 \end{cases}, \quad (3.6)$$

la cual equivale a la expresión más sencilla

$$S(x) = \min(1, \max(0, cx)) \quad (3.7)$$

Esta función permite modelar el retardo existente en cualquier dispositivo mediante la variación del parámetro c .

e. función umbral exponencial. Esta es similar a la anterior

$$S(x) = \min(1, e^{cx}) \quad (3.8)$$

Esta función provee a las redes neurales de una convergencia rápida a patrones de señales.

- f. función distribución exponencial. Esta provee otro tipo de función umbral:

$$S(x) = \min(1, 1 - e^{-cx}) \quad (3.9)$$

- g. función de razón polinomial. Esta es una familia de funciones de la forma,

$$S(x) = \max(0, x^n / (c + x^n)), \quad n > 1 \quad (3.10)$$

Cuando la activación es positiva, $S(x)$ es monótona creciente.

- h. funciones de señales codificadas en pulsos. Al igual que en el campo de los circuitos electrónicos, suele ser deseable que la información esté codificada en pulsos. Así, es la presencia o ausencia de los pulsos la que contiene la información y cualquier degradación que sufran en su transmisión puede ser obviada. El procesamiento de pulsos implica que la neurona muestrea su entrada. En el tiempo de muestreo, una neurona cuenta, suma o integra los pulsos más recientes. Esto equivale a una

memoria que se actualiza constantemente. Más concretamente, esto asemeja un pre-almacenamiento (buffer) de memoria en un sistema digital.

La función que caracteriza a la neurona se puede representar entonces de la siguiente forma:

$$S_1(t) = \int_{-\infty}^t x_1(s) e^{a-t} ds, \quad (3.11)$$

donde x_1 es una función que puede valer sólo 1 ó 0 (presencia o ausencia de pulsos).

2. Topologías. Una red neural consiste en una interconexión de neuronas. Básicamente podemos distinguir dos tipos de interconexión: la de cadena abierta y la retroalimentada. El caso de cadena abierta implica que la salida (o salidas) de la red es un mapeo directo de las entradas. Si las funciones de cada neurona están determinadas y no varían en el tiempo, el mapeo está completamente determinado.

El caso de redes con retroalimentación suele llamarse recursivo. Aquí las neuronas reciben como entradas una combinación lineal de las salidas del resto de neuronas que conforman la red incluyéndose a sí misma (no importa si en realidad no reciban entradas de todas las otras neuronas,

pues se les representa con un peso de cero).

La expresión matemática de una red neural particular se puede llevar a cabo por medio de una matriz de conexión. Los elementos m_{ij} de esta matriz corresponden a los pesos de la salida de la neurona i a la entrada de la neurona j .

El comportamiento del sistema en el tiempo ha de describirse por medio de un conjunto de ecuaciones diferenciales de primer orden. Uno de los modelos más utilizados es el llamado modelo de Hopfield continuo, que corresponde a una red de n neuronas, descritas por n ecuaciones de la forma:

$$C_i x_i' = -x_i/R_i + \sum_{j \neq i} m_{ij} g(x_j) + u_i, \quad i = 1, 2, \dots, n \quad (3.12)$$

donde C_i se conoce como la constante de tiempo de la membrana (es análoga a una capacitancia en un circuito eléctrico), R_i es la resistencia de membrana, los w_{ij} son los elementos de la matriz de conexión y los u_i son las entradas a la red. La variable primada (x_i') corresponde a la derivada en el tiempo de x_i .

Así como éste, existen otros sistemas de ecuaciones diferenciales propuestos por distintos especialistas en la materia.

3. Leyes de Aprendizaje. Una red neural puede ser estática, es decir que los pesos de interconexión (es decir los coeficientes de la combinación lineal) no varíen en el tiempo. Sin embargo, es deseable que una red pueda aprender con base en cambios en el sistema. Por esto se le provee de una ley de aprendizaje, la cual modifica estos pesos de interconexión de manera que se configure la red de la manera adecuada. Definiremos operacionalmente el aprendizaje como un proceso que codifica información. Un sistema aprende un patrón si lo codifica en su estructura, es decir, en los pesos de las interconexiones. El sistema cambia esta estructura a medida que va aprendiendo. Podemos extender este concepto diciendo que un sistema ha aprendido el par (x_1, y_1) si responde con y_1 cuando el sistema es activado con x_1 .

Los tipos de aprendizaje en una red neural se pueden clasificar como supervisados y no supervisados. En el aprendizaje supervisado se asume que se conoce la salida deseada de la red. Con base en una función de error se ajustan los pesos hasta que se obtenga la salida correspondiente. En el aprendizaje no supervisado no se conoce la salida deseada. El aprendizaje se basa en conjuntos de valores de entrada y salida. Los algoritmos empleados en este tipo de aprendizaje intentan localizar patrones en la entrada.

En los siguientes apartados examinamos estos tipos de aprendizaje.

4. Aprendizaje no supervisado. Las leyes de aprendizaje no supervisado consisten en ecuaciones diferenciales que expresan la variación de las componentes de la matriz de conexión en el tiempo.

a. Ley de Aprendizaje Hebbiana de señal (Signal Hebbian learning law). Esta ley correlaciona señales cercanas entre sí:

$$m_{1j}' = -m_{1j} + S_1(x_1)S_j(x_j), \quad (3.13)$$

donde S es la función correspondiente de cada neurona.

b. Ley de aprendizaje competitiva. Esta ley fue propuesta por Grossberg en 1969.

$$m_{1j}' = S_j(y_j)[S_1(x_1) - m_{1j}] \quad (3.14)$$

En este caso, la función $S_j(y_j)$ es de tipo discreto y bivalente (vale solo 0 o 1) y sirve para modular la diferencia entre el valor actual de conexión y el valor obtenido por la entrada.

c. Ley de aprendizaje Hebbiana diferencial. Esta correlaciona tanto la velocidad de la señal como la señal misma:

$$m_{1j}' = -m_{1j} + S_1(x_1)S_j(y_j) + S_1'(x_1)S_j'(y_j) \quad (3.15)$$

Como se puede ver, esta es una variación de la ley Hebbiana mencionada anteriormente. La velocidad con que cambian las señales es un factor adicional en la determinación del cambio en los valores de conexión.

d. Ley de aprendizaje diferencial competitivo.

Introducida por Kosko en 1990, esta combina el aprendizaje competitivo y el Hebbiano diferencial:

$$m_{1j}' = S_j'(y_j)[S_1(x_1) - m_{1j}] \quad (3.16)$$

Esta ley obliga a aprender solo cuando hay un cambio. La velocidad de señal S_j' se comporta como un refuerzo para determinar si hay o no cambio.

5. Aprendizaje supervisado. El aprendizaje supervisado intenta aproximar una función desconocida $f: X \rightarrow Y$ con base en muestras $(x_1, y_1), \dots, (x_m, y_m)$ por medio de la minimización del error. Este error se define como salida

deseada menos la salida real. La supervisión consiste en una señal de aprendizaje omnipresente, una señal de error externa que sirve para modificar el sistema.

Existe una serie de algoritmos diseñados para efectuar el aprendizaje supervisado. Rosenblatt introdujo el concepto de perceptron(2), en donde se estima un error lineal con base en errores instantáneos. El algoritmo de error cuadrático mínimo (o LMS por Least Mean-Squared) de Widrow estima el error cuadrático medio de una red lineal con un error cuadrático instantáneo. La regla Delta o algoritmo de propagación en reversa (backpropagation algorithm) modifica los pesos después de cada presentación de los vectores de entrada; su mayor utilidad es el reconocimiento de patrones. La presentación de estos algoritmos es bastante extensa, por lo que se omitirá en este trabajo. De cualquier forma esta presentación no será necesaria para conjuntar la lógica difusa con las redes neurales. Para mayor información se puede consultar Kosko, 1992 (ver Bibliografía).

6. Utilidad de las redes neurales. Debido a su naturaleza, las redes neurales no necesitan un modelo matemático para operar. Ellas se ajustan, ya sea con o sin supervisión, de forma que se obtengan salidas adecuadas a las entradas en cada momento. En otras palabras, no se define un modelo rígido sobre el cual basan su accionar sino que

aprenden con base en ejemplos.

Por otra parte, las redes neurales procesan información numérica (en oposición a información sintáctica). Esto permite que se implementen redes neurales en circuitos integrados electrónicos u ópticos. Estas propiedades distinguen a las redes neurales (al igual que los sistemas difusos) de las implementaciones de inteligencia artificial convencional, las cuales favorecen el uso de procesamiento simbólico.

C. Memorias Difusas Asociativas

El estudio de la lógica difusa requiere el uso de razonamientos proposicionales y el cálculo de predicados. Aunque estas técnicas son perfectamente aplicables y bien establecidas teóricamente, su empleo es en cierto modo engorroso y limitado. Hasta ahora, las técnicas mencionadas se han empleado en sistemas expertos en un intento de lograr inteligencia artificial.

Sin embargo, en la lógica difusa los predicados se pueden considerar como un tipo especial de conjunto difuso. Esto nos permite visualizar razonamientos lógicos como asociaciones entre conjuntos. Este punto de vista permite trasladar la información a un formato numérico (recuerde las funciones de membresía) y no sintáctico como se haría en un

razonamiento lógico convencional. Este formato numérico permite que se modifiquen fácilmente sistemas difusos, empleando diversas técnicas como la de las redes neurales, de las cuales se habló con anterioridad.

1. Concepto General. Un conjunto difuso se representa numéricamente por su función de membresía. En un universo con subconjuntos, podríamos localizar a sus elementos en el conjunto $[\emptyset, 1]^n$. Entonces, supongamos un sistema difuso de esta entrada y una salida expresada en términos de p subconjuntos difusos. El sistema se puede representar sencillamente como una transformación:

$$S: I^n \rightarrow I^p \quad (3.17)$$

Una transformación continua de la forma anterior que mapea bolas de conjuntos difusos en I^n a bolas en I^p representa a un sistema difuso real. A este tipo de sistemas se les llamará memorias difusas asociativas (o más simplemente FAMs, por Fuzzy Associative Memories).

La FAM más sencilla codifica una regla o asociación (A_1, B_1) . Por ejemplo, suponga un control de tráfico con la regla "Si el tráfico hacia el norte es denso entonces mantenga la luz verde más tiempo". La asociación (DENSO, MAS TIEMPO) codifica esta regla eficientemente. En general, un

sistema de FAMs codifica y procesa en forma paralela un banco de m reglas FAM $(A_1, B_1), \dots, (A_m, B_m)$.

Es importante recalcar que los conjuntos de entrada A_i pueden ser el producto de operaciones de agregación entre conjuntos sencillos. Generalmente esta operación de agregación es la conjunción. Así, la asociación $(A^1, A^2; B)$ representa la regla

$$\text{"SI } E^1 \text{ es } A^1 \text{ Y } E^2 \text{ es } A^2 \text{ ENTONCES } S \text{ es } B\text{"}, \quad (3.18)$$

donde E^1 y E^2 son variables de entrada y S es una variable de salida.

2. Codificación. Las FAMs se comportan como transformaciones de conjuntos difusos a conjuntos difusos. Esto implica que existe una relación entre los elementos de estos conjuntos. La relación ha de codificarse de alguna forma en el sistema. Mediante la cuantificación de los subconjuntos difusos de entrada y salida formamos vectores con un número finito de componentes que representan un subconjunto difuso. Llamemos A al vector de entrada y B al vector de salida. Entre estos vectores existe una relación de la forma:

$$A \cdot M = B, \quad (3.19)$$

donde el operador \circ denota la operación composición max-min mencionada en el capítulo II.

La determinación de la matriz M constituye el problema básico de la codificación de información en un sistema difuso. Por lo general estos sistemas emplean una codificación conocida como de correlación mínima. En este caso:

$$m_{ij} = \min(a_i, b_j), \quad (3.20)$$

donde los a_i pertenecen a A y los b_j pertenecen a B .

Otro tipo de codificación emplea la operación producto:

$$m_{ij} = a_i b_j \quad (3.21)$$

En sistemas reales generalmente se escoge este tipo de codificación debido a que se preserva mayor información (i.e., el valor de los elementos de la matriz depende de ambos valores y no sólo de uno, como es el caso de la correlación del mínimo).

Considere un sistema cualquiera de m entradas y n salidas. Supongamos que la entrada E^i puede tomar como valor cualquiera de r_i subconjuntos difusos (valores de la variable lingüística) y que la salida S_j puede tomar s_j posibles subconjuntos. Se pueden definir

$$k = \prod_{i=1}^m \pi r_i * \prod_{j=1}^n \pi s_j \quad (3.22)$$

posibles celdas FAM para especificar el sistema. La mayoría de las veces no se utilizan todas las posibles celdas, debido a la naturaleza del sistema, condiciones externas, etc.

Es interesante notar que, dadas las condiciones enumeradas en el párrafo anterior, se pueden tener hasta 2^k posibles sistemas difusos diferentes. El número de sistemas implementables en la realidad es, por supuesto, una fracción mínima de este número.

3. Sistemas. Un sistema difuso almacena y procesa reglas de tipo FAM en paralelo. La salida es entonces un resultado de varios procesos parciales correspondientes a cada regla. Es decir, cada inferencia da como resultado un conjunto difuso B_j' (en este caso la variable primada no denota la derivada). La salida completa es de la forma:

$$B = \sum_j w_j B_j' \quad (3.23)$$

donde los w_j son coeficientes escalares (o pesos como se les conoce). Los sistemas adaptativos emplean diversas técnicas para escoger los coeficientes w_j y, por lo tanto, definir el sistema en cada instante. Formalmente, se dice que un

sistema es adaptativo si $dS/dt \neq 0$ para algún t , donde S es la función que define el sistema.

Para obtener la salida deseada, el conjunto difuso B ha de ser convertido a un valor escalar. Uno de los métodos empleados consiste en tomar el elemento con máximo valor de membresía en el conjunto. Otro método, más utilizado, computa la salida escalar como el centroide del conjunto agregado B :

$$y = \frac{\int_{-\infty}^{\infty} ym(y)dy}{\int_{-\infty}^{\infty} m(y)dy} \quad (3.24)$$

siempre que existan las integrales.

En implementaciones reales suelen utilizarse elementos que procesan datos en forma discreta (que es el caso de los microprocesadores digitales). Debido a esto, los conjuntos de entrada y salida se caracterizan no por una función sino por un conjunto discreto de elementos con sus correspondientes grados de membresía. Las reglas actúan sobre un conjunto discreto de valores, los cuales dan como resultado otro conjunto discreto de valores. Así, las integrales de la expresión anterior se convierten en sencillas sumatorias:

$$y = \frac{\sum_{j=1}^p y_j m_B(y_j)}{\sum_{j=1}^p m_B(y_j)} \quad (3.25)$$

donde se llama B al conjunto de salida, el cual está cuantificado en p elementos, llamados y_j , los cuales tienen un grado de membresía $m_B(y_j)$.

Por cuestión de simplificación, las variables difusas toman valores de forma trapezoidal simétrica (como en el ejemplo), por lo que la expresión para la salida no difusa es bastante sencilla. Esto permite que se implemente esta conversión en algoritmos sencillos para microprocesadores digitales.

4. Memorias adaptativas. Una FAM adaptativa (AFAM por sus siglas en inglés) es un mapeo dependiente del tiempo entre conjuntos difusos. Estas FAMs pueden generar reglas a partir de muestras. Considere un sistema $S: I^n \rightarrow I^p$. Una regla FAM tiene la forma "SI X es A_1 ENTONCES Y es B_1 " o equivalentemente $\langle A_1, B_1 \rangle$. Suponga que r conjuntos difusos A_1, \dots, A_r cuantifican el universo de entrada X. Los s conjuntos difusos B_1, \dots, B_s cuantifican el universo de salida Y. En general r, s y la forma de los subconjuntos

han de ser dados por el usuario.

El total de combinaciones entrada-salida produce un total de $r*s$ celdas FAM, las cuales denotaremos como F_{ij} , en el espacio $I^n \times I^p$ (producto cartesiano). Estas celdas han de traslaparse una sobre otra, pues pares contiguos de conjuntos A_i y A_{i+1} , y B_i y B_{i+1} se traslapan. Esto provoca que las F_{ij} no cubran todo el espacio $I^n \times I^p$. En general, habrá puntos con mayor concentración de F_{ij} , los cuales indican una regla de actuación.

Una forma alternativa de generar reglas de actuación consiste en trabajar el espacio $X \times Y$ en vez de $I^n \times I^p$. En este caso, los conjuntos X y Y son de tipo discreto, donde $X = \{x_1, \dots, x_n\}$ y $Y = \{y_1, \dots, y_p\}$. Esto facilita la implementación digital, pues en vez de funciones se almacenan conjuntos de valores con sus correspondientes grados de membresía. Un conjunto de muestras para aprendizaje de la forma $(x^1, y^1), (x^2, y^2), \dots, (x^k, y^k)$ entra al sistema. Estas muestras generan k vectores m_1, \dots, m_k en $X \times Y$. El aprendizaje los distribuye a diferentes celdas FAM F_{ij} . La acumulación de muestras en una celda particular F_{ij} sirve para activar los m_{ij} particulares de una red neural que estructuralmente es análoga al sistema difuso. Mientras las neuronas de la red tengan una señal de activación monótona creciente, los pesos de conexión en donde debe haber reglas tenderán a un máximo. La red varía su matriz de conexión de

forma que, a mayor número de muestras de aprendizaje correspondientes a una celda Fig, mayor valor tiene el peso de interconexión en la red neural y más fácilmente se determinan reglas de actuación.

La facilidad del procesamiento de tipo difuso se manifiesta en la forma de manejar la información. Un sistema difuso cuantifica la descripción de un conjunto difuso en forma numérica exacta. El procesamiento que se le da a estas cantidades exactas es de tipo difuso. La forma de procesar las cantidades (reglas) se puede determinar experimentalmente o por medio de redes neurales. La diferencia entre éstas y los sistemas difusos es que éstas codifican información de tipo numérico exacto. Así, la determinación de la estructura del sistema se realiza de forma no difusa, pero el procesamiento en sí es el que requiere esta imprecisión.

Notas:

(1) La técnica de centroides es la más utilizada en aplicaciones reales. Sin embargo existen otras formas de conversión del ámbito difuso al preciso. La más sencilla asume que la salida consiste en un solo conjunto difuso y la salida precisa (es decir, no difusa), es el elemento central del conjunto con un valor más alto de membresía.

(2) El perceptron es básicamente una red neural sin retroalimentación con una neurona de salida. Esto es, se puede visualizar a un perceptron como una caja negra que tiene varias entradas y una salida, en donde las primeras no dependen de la última. Su aplicación la halla en reconocimiento y separación de patrones. Una explicación más detallada la halla en el libro de Kosko, 1992.

IV. APLICACIONES

A continuación se presentan algunas de las aplicaciones más importantes de la lógica difusa relacionadas con la electrónica.

A. Codificación de imágenes de video.

La transmisión de imágenes de video en forma digital reduce la posibilidad de errores pero requiere un mayor ancho de banda. La capacidad limitada de los canales existentes, aunada a la investigación en el campo de televisión de alta definición, motiva el estudio de técnicas de compresión de imágenes. Estas técnicas intentan minimizar el número de bits necesarios para representar una imagen y reconstruirla con el mínimo de distorsión.

1. Introducción. Las imágenes de video se componen de pixeles (del inglés "pixels" o "picture elements") o cuadritos. Cada pixel tiene asociado un nivel de voltaje que corresponde a su luminancia, en donde la máxima luminancia corresponde al blanco, la mínima al negro y los valores intermedios son tonos de gris. En este caso se estudiarán imágenes en blanco y negro para simplificar (La crominancia o color se genera de manera distinta, pues no se trabaja de

pixel en pixel). En general, los niveles de luminancia de los pixeles se correlacionan bastante en una secuencia normal de imágenes. Al codificar una imagen se trata de representarla con datos que no están correlacionados. Mientras menos correlación haya en los datos, existe menos redundancia y se tiene la información esencial de la imagen. La codificación de imágenes puede ser predictiva o transformada. En el primer caso, se busca redundancia en los datos para predecir un comportamiento similar en imágenes subsiguientes. En la codificación transformada, los datos se transforman a una forma en la que exista un mínimo de correlación. Se descartan los coeficientes con bajo valor para así comprimir la imagen. A continuación examinaremos un tipo de codificación de imágenes por transformada desde el punto de vista clásico y el de la lógica difusa. Para simplificar asumiremos que se tiene una imagen de video en blanco y negro. El término $x(m,n)$ representa el valor de luminancia del pixel de la fila m y columna n . La transformada de coseno discreta (DCT por sus siglas en inglés) para un conjunto de $N \times N$ pixeles se define así:

$$X(u,v) = \frac{4c(u)c(v)}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \cos \frac{(2m+1)u\pi}{2N} \cos \frac{(2n+1)v\pi}{2N} \quad (4.1)$$

y la inversa:

$$x(m,n) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} c(u)c(v)X(u,v) \cos \frac{(2m+1)u\pi}{2N} \cos \frac{(2n+1)v\pi}{2N} \quad (4.2)$$

donde $c(0)=1/\sqrt{2}$ y $c(k)=1$ para $k=1,2,\dots,N-1$. Nótese que esta es una transformada bidimensional.

En la implementación, una imagen de video se divide en varias subimágenes de $N \times N$ píxeles y cada una de estas se transforma con la DCT. Definimos también la energía AC y DC de una subimagen de la siguiente forma:

$$\text{energía DC} = X^2(0,0) \quad (4.3)$$

$$\text{energía AC} = \left[\sum_u \sum_v X^2(u,v) \right] - X^2(0,0) \quad (4.4)$$

La energía DC es una medida de la intensidad del fondo de la imagen. Cuando una subimagen mantiene una intensidad de fondo uniforme, la mayor parte de la energía se concentra alrededor del término $X(0,0)$. La energía AC es una medida de la variación entre píxeles que existe en un cuadro dado. Una imagen con grandes variaciones en tonalidad distribuye su energía en el espectro de $X(u,v)$.

La técnica clásica está representada por el sistema adaptativo de codificación por transformada, propuesto por Chen y Smith en 1977. El sistema de Chen divide una imagen en un número de subimágenes. Luego se ordenan las subimágenes con base en su energía AC. A continuación se

divide este ordenamiento en cuatro clases de subimágenes, digamos 1, 2, 3 y 4. La clase 1 es la que tiene las subimágenes con mayor "actividad", es decir con mayor energía AC. La cuarta clase es la que tiene subimágenes con menor energía AC. Se utiliza la varianza en cada pixel para computar cuatro mapas de bits o mapas de cuantización, uno para cada clase de subimágenes. El mapa de cuantización es uno-a-uno con la subimagen pero no corresponde a esta en sí. Es decir, una subimagen de 16 x 16 pixeles se codifica con un mapa de cuantización de 16 x 16 elementos, pero cada elemento no corresponde a un pixel. Cada elemento del mapa indica el número de bits utilizados para cuantificar el valor real $X(u,v)$. Se asignan más bits a imágenes con varianza grande y menos bits a imágenes con varianzas pequeñas. Los coeficientes de la transformada se normalizan antes de cuantificarse y convertirse a digital.

En el receptor es necesaria cierta información adicional para decodificar la imagen comprimida. Esta información incluye la forma de clasificación, los coeficientes de normalización y los cuatro mapas de cuantificación. En la figura 4.1. se muestra un esquema general de la codificación y decodificación por medio de la transformada discreta de coseno.

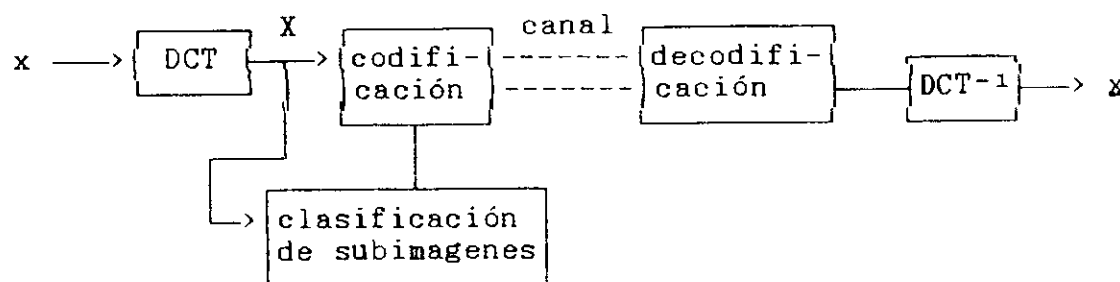


Figura 4.1. Diagrama de bloques de codificación por medio de la transformada discreta de coseno.

Los mapas de bits o de cuantificación utilizados con más frecuencia son de 16×16 , correspondientes a subimágenes de 16×16 píxeles. Estos se muestran en las figuras 4.2.a-d, y corresponden a un mapeo de aproximadamente 1 bit/píxel, el cual equivale a una compresión de 8:1 (pues normalmente se tienen 8 bits por cada píxel). Es necesario aclarar que estos mapas corresponden no a la subimagen en sí, sino a las transformadas discretas de coseno. Es decir, no es $x(1,1)$ sino $X(1,1)$ el que se cuantiza en 6 bits para una subimagen clasificada en la clase 1. Para compresiones aún mayores (por ejemplo, de 0.5 bit/píxel o 16:1) los mapas difieren (tienen mayor cantidad de ceros). En general, la energía DC ($X(0,0)$) siempre se codifica con el máximo número de bits, ocho en este caso.

```

8 7 6 5 4 3 3 2 2 1 1 1 0 0 0 0
7 6 6 5 4 3 3 2 2 1 1 1 0 0 0 0
6 5 5 4 4 3 3 2 2 1 1 1 0 0 0 0
5 5 4 4 4 3 3 3 2 2 1 1 1 0 0 0
4 4 4 3 3 3 3 2 2 2 1 1 1 0 0 0
3 3 3 3 3 2 2 2 2 2 1 1 1 1 0 0
3 3 3 3 3 2 2 2 2 1 1 1 1 1 0 0
2 2 2 3 2 2 2 1 1 1 1 1 1 1 0 0
2 2 2 2 3 2 2 1 1 1 1 1 1 1 0 0
2 2 2 2 2 2 1 1 1 1 1 1 1 0 0 0
1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0

```

Figura 4.2.a
 Compresión 8:1
 Mapa de cuantificación
 para la clase 1
 de subinágenes.

```

8 6 5 4 4 3 2 2 2 1 1 1 0 0 0 0
6 5 5 4 4 3 2 2 2 1 1 1 0 0 0 0
5 5 4 4 4 3 3 2 2 1 1 1 0 0 0 0
4 4 4 3 3 3 3 2 2 2 1 1 0 0 0 0
3 3 3 3 3 3 3 2 2 2 1 1 1 0 0 0
3 3 3 3 2 2 2 2 2 2 1 1 1 1 0 0
2 2 2 2 2 2 2 2 2 2 1 1 1 1 0 0
2 2 2 3 3 2 1 1 1 1 1 1 1 1 0 0
2 2 2 3 4 3 2 1 1 1 1 1 0 0 0 0
1 2 1 2 3 2 1 1 1 1 1 0 1 0 0 0
1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0
1 1 1 1 1 1 1 1 2 2 1 0 0 0 0 0

```

Figura 4.2.b
 Compresión 8:1
 Mapa de cuantificación
 para la clase 2

```

8 5 4 3 3 2 2 1 1 1 1 0 0 0 0 0
5 5 4 4 3 3 2 2 1 1 0 0 0 0 0 0
4 4 3 3 3 3 2 2 1 1 1 0 0 0 0 0
3 3 3 3 3 2 2 2 1 1 1 0 0 0 0 0
3 3 3 2 2 2 2 1 1 1 1 0 0 0 0 0
2 2 2 2 2 2 1 1 1 1 1 0 0 0 0 0
2 2 2 2 2 1 1 1 1 1 1 0 0 0 0 0
2 1 1 1 2 1 1 1 1 1 0 0 0 0 0 0
1 1 1 2 2 2 1 1 1 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
1 1 1 0 1 1 1 1 0 0 0 0 0 0 0 0
1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0

```

Figura 4.2.c .

Compresión 8:1. .
 Mapa de cuantificación
 para la clase 3. .

```

8 3 3 2 1 1 0 0 0 0 0 0 0 0 0 0
3 3 2 2 1 1 1 1 0 0 0 0 0 0 0 0
2 2 2 1 1 1 1 1 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

Figura 4.2.d .

Compresión 8:1 .
 Mapa de cuantificación
 para la clase 4. .

Es importante recalcar que el nivel de compresión está dado para una imagen supuesta que tiene una distribución aleatoria de las luminancias de sus pixeles. En la realidad, el nivel de compresión depende de cómo se distribuyen estas luminancias, de acuerdo a cada imagen.

En Guatemala, la industria de televisión por cable utiliza equipo especial para la decompresión de imágenes de video. La imagen es enviada en forma digital al satélite, modulada en QPSK. La compresión se realiza utilizando la transformada discreta de coseno. La clasificación de subimágenes se realiza con un algoritmo patentado por la compañía Compression Labs, Inc. La recepción se lleva a cabo por medio del receptor M6 de esa compañía, conocido por su nombre comercial Spectrum Saver Integrated Receiver Decoder. En nuestro país, existen varias compañías de cable que tienen estos aparatos, para la recepción de algunos canales de satélite que ya emplean transmisión digital comprimida. En el apéndice B se tienen algunas características técnicas de este aparato. Por otra parte, la compresión de imágenes con DCT está en fase de investigación para su utilización, primero en la transmisión digital, para eventualmente llegar a un formato de televisión de alta definición. La NHK de Japón ha presentado un sistema de transmisión digital multicanal mediante la compresión de la señal por medio de la transformada discreta de coseno.

2. Codificación utilizando lógica difusa. Un sistema de FAMS puede generar reglas con base en el aprendizaje no supervisado. El sistema de codificación emplea siempre la DCT y los mapas de cuantificación. La diferencia se halla en la forma de clasificar las subimágenes. Estas se pueden clasificar en cuatro clases difusas (fuzzy) representadas por los valores difusos AL (alto), MA (medio alto), MB (medio bajo), y BA (bajo). Al igual que el sistema de Chen, la subimagen AL se codifica con más bits y la BA con menos bits. Los cuatro conjuntos difusos GR (grande), MD (mediano), PQ (pequeño) y MP (muy pequeño) sirven para valorar la variable lingüística T, que representa la potencia AC de una imagen. Además, se introduce la variable L, que denota la potencia AC de baja frecuencia y toma los valores PQ (pequeño) y GR (grande).

Sea la subimagen de $m \times m$ píxeles, con m par. Las variables T y L se definen numéricamente así:

$$T = \sum_{u=0}^{m-1} \sum_{v=0}^{m-1} |X(u,v)| - |X(0,0)| \quad (4.5)$$

$$L = \sum_{u=0}^{m-1} \sum_{v=0}^{m-1} |X(u,v)| - |X(0,0)|, \quad (4.6)$$

donde $| \cdot |$ representa el valor absoluto.

A continuación corresponde estimar reglas FAM del tipo $(T_1, L_1; B_1)$ con base en datos de entrenamiento. Para cubrir todas las posibilidades, se han de determinar $4 \times 2 = 8$ reglas de actuación (por los 4 posibles valores de T_1 y los 2 de L_1). Una red neural que se comporta de acuerdo al aprendizaje diferencial competitivo clasifica vectores generados a partir del sistema de Chen y determina 5 reglas. Las otras tres se determinan por sentido común para obtener la siguiente base de información:

		T			
		GR	MD	PQ	MP
L	GR	AL	AL	MB	BA
	PQ	AL	MA	MB	BA

Figura 4.3.

Por ejemplo, la primer celda sintetiza la regla:

"SI la potencia AC es GRANDE y la potencia de baja frecuencia es GRANDE ENTONCES clasifique la subimagen en la clase ALTA",

y así sucesivamente.

La diferencia básica entre este esquema de clasificación

y el de Chen estriba en el hecho de que esta última tiene que ordenar las subimágenes y luego las clasifica, mientras que la FAM no necesita el ordenamiento previo. Esto reduce los cálculos requeridos del sistema pues, una vez entrenado, este clasifica las subimágenes sin cálculos numéricos.

Experimentos realizados por Kosko y Kong demostraron que la implementación difusa de este sistema de clasificación de imágenes es mucho más eficiente que el algoritmo original de Chen. El sistema difuso requiere menos bits por pixel para mantener una calidad de imagen comparativa. Esta, por supuesto, es una apreciación subjetiva, pero la utilización de parámetros numéricos de comparación le dan la razón al sistema difuso. Por ejemplo, se define un error de reconstrucción:

$$r(m,n) = x(m,n) - \underline{x}(m,n), \quad (4.7)$$

donde \underline{x} representa el valor calculado. La varianza de este error es:

$$\sigma^2 = \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} r^2(m,n) \quad (4.8)$$

Con este valor se calcula la relación señal a ruido de la imagen:

$$\text{SNR} = 10 \log (D^2/\sigma^2) \text{ dB}, \quad (4.9)$$

donde D es la diferencia entre el máximo y el mínimo valor de luminancia posibles.

En los ejemplos mostrados en las figura 4.4.a-c. y 4.5.a-c., las SNR son consistentemente más altas al utilizar la FAM en vez del sistema clásico de Chen.



Fig. 4.4a. imagen original "Lenna"



Figura 4.4b
codificación
con lógica
difusa



Figura 4.4c
codificación
con técnica
Chen



Fig. 4.5a. imagen original "F-16"

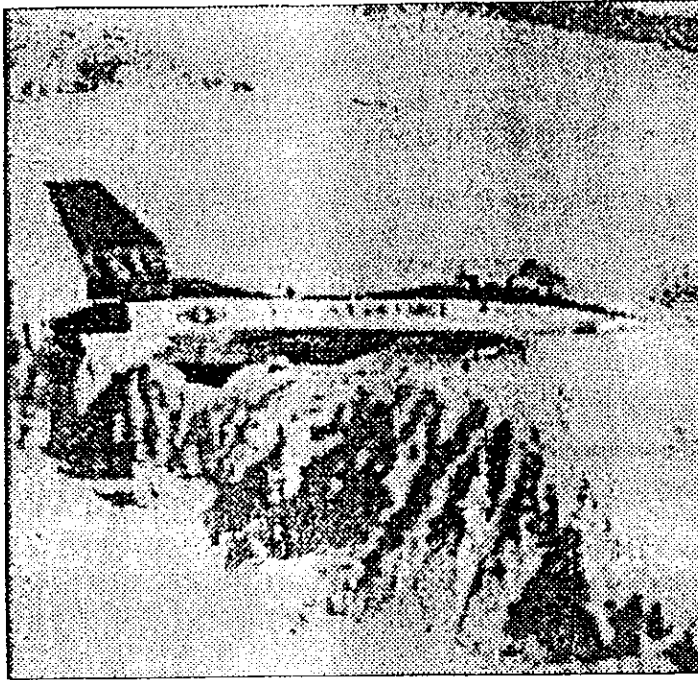


Figura 4.5b
codificación
con lógica
difusa



Figura 4.5c.
codificación
con técnica
Chen

B. Metereología

La ciencia de la metereología maneja sistemas extremadamente complejos. Por esta razón, las descripciones y predicciones metereológicas siempre se han hecho con base en términos lingüísticos tales como "clima caliente", "sequía", "presión baja", etc. La aplicación de conjuntos difusos en esta ciencia trata de aprovechar la característica de imprecisión para manejar adecuadamente la complejidad inherente.

Las aplicaciones más exitosas de la lógica difusa en metereología son las que se sustentan en un examen de patrones climatológicos en el tiempo. Con base en estas estadísticas se busca entender los ciclos de cambios climatológicos que ocurren en regiones geográficas determinadas. En uno de estos estudios, se recopilaron datos que cubren aproximadamente un siglo sobre la precipitación pluvial de un área.

La idea de estos estudios es el examen de cambios entre períodos de sequía y humedad. Cada período se caracteriza por niveles anuales de precipitación similares entre sí. Los puntos de cambio entre sequía y humedad se caracterizan por la aparición de disimilitud entre niveles consecutivos de precipitación. El punto exacto de cambio no puede ser determinado con absoluta precisión. Para acomodar la

ambigüedad entre las fronteras de "sequía" y "humedad", se construye una relación difusa de similaridad entre los valores anuales de precipitación. De esta forma se agrupan los años correspondientes a niveles de precipitación similares. La relación se forma calculando un índice de similaridad s'_{ij} entre precipitaciones p_i y p_j de esta forma:

$$s'_{ij} = 1 - |p_i - p_j| \quad (4.10)$$

Los índices se normalizan de forma que tengan valores en el intervalo $[0,1]$ por medio de la fórmula:

$$s_{ij} = (s'_{ij} - s'_{\min}) / (s'_{\max} - s'_{\min}), \quad (4.11)$$

donde s'_{\max} y s'_{\min} son los índices de similaridad máximos y mínimos del grupo de datos, respectivamente. Estos índices normalizados corresponden a los grados de membresía de una relación R , definida del conjunto de precipitaciones anuales a sí mismo. De esta forma:

$$\mu_R(p_i, p_j) = s_{ij} \quad (4.12)$$

Es fácil ver que la relación R es reflexiva pues $s'_{ii} = 1 - |p_i - p_i| = 1$. Además, como el mínimo del valor absoluto es 0, el valor máximo de s'_{ij} es 1. Esto implica que $s'_{ii} = s'_{\max}$.

siempre que s_{ij} (o sea s_{\max}) no sea igual a s_{\min} (lo cual correspondería a una región sin cambios).

También se puede ver que R es simétrica pues s_{ij} es igual a s_{ji} (ya que podemos intercambiarlos dentro del valor absoluto). Entonces, $\mu_R(p_i, p_j) = s_{ij} = s_{ji} = \mu_R(p_j, p_i)$.

Sin embargo, la relación R no suele ser transitiva. Para esto se calcula R_t , la clausura transitiva de R . En este caso, R_t es transitiva por lo que es de similitud. Tomando un valor arbitrario $\alpha \in [0, 1]$ como parámetro de similitud, determinamos que parejas p_i, p_j son similares en grado α . Esto permite agrupar los datos y representarlos gráficamente. Los cambios entre períodos se pueden apreciar fácilmente y permiten que estos se investiguen y puedan ser predichos matemáticamente.

C. Reconocimiento de Patrones.

Uno de los campos en que halla mayor aplicación la lógica difusa es en el de reconocimiento de patrones y agrupación de objetos. El reconocimiento de patrones engloba una serie de técnicas encaminadas a detectar ritmos repetitivos en conjuntos de datos. La aplicación de lógica difusa, y más específicamente conjuntos difusos a este campo es una consecuencia natural debido a que los patrones empleados no tienen una definición muy precisa. El ser humano es capaz de reconocer similitudes y clasificar patrones como caracteres escritos a mano, caras, voces, etc., los cuales tienen un rango de variación bastante amplio.

Por ejemplo, una aplicación involucra el reconocimiento y clasificación de cromosomas dentro de tres distintas categorías: mediana, submediana y acrocéntrica. Un ejemplo de cada tipo se muestra en la figura 4.6. El esquema de clasificación se basa en la relación entre la longitud de los brazos del cromosoma y la longitud total de su cuerpo. Es muy difícil identificar fronteras bien marcadas entre los tres tipos. Por lo tanto, Lee en 1975 implementó una técnica difusa de reconocimiento de patrones. En esta se comparan los ángulos y largos de brazos de los cromosomas con los identificados en el esqueleto idealizado de la figura 4.7. Primero se efectúa una prueba de simetría, en donde se define un cromosoma simétrico como aquel cuyos lados cumplen con que

$a_1 = a_2$, $a_3 = a_4$ y cuyos ángulos cumplen $A_{2i-1} = A_{2i}$ para $1 \leq i \leq 4$. El grado de membresía de cada cromosoma x en el conjunto difuso S de cromosomas simétricos se calcula por:

$$\mu_S(x) = 1 - (1/720^\circ) \sum_{i=1}^n |A_{2i-1} - A_{2i}| \quad (4.13)$$

donde los ángulos A_i se dan en grados sexagesimales. Mientras más simétrico es un cromosoma, más cercano a la unidad es su grado de membresía. Posteriormente, se clasifica un cromosoma en tres categorías difusas, llamadas mediano (M), submediano (SM) y acrocéntrico (AC). La función de membresía del cromosoma x en el conjunto M es:

$$\mu_M(x) = \mu_S(x) [1 - (|a_1 - a_4| + |a_2 - a_3|) / a_T], \quad (4.14)$$

donde $a_T = a_1 + a_2 + a_3 + a_4 + a_5$. La función de membresía en SM está dada por:

$$\mu_{SM}(x) = \mu_S(x) [1 - a_{SM} / 2a_T], \quad (4.15)$$

donde $a_{SM} = \min[(|a_1 - 2a_4| + |a_2 - 2a_3|), (|2a_1 - a_4| + |2a_2 - a_3|)]$

Finalmente, la función de membresía en el conjunto AC es:

$$\mu_{AC}(x) = \mu_S(x) [1 - a_{AC} / 4a_T], \quad (4.16)$$

donde $a_{AC} = \min[(|a_1 - 4a_4| + |a_2 - 4a_3|), (|4a_1 - a_4| + |4a_2 - a_3|)]$.

Cada cromosoma tiene un grado de membresía en cada uno de los tres conjuntos M, SM y AC. Este cromosoma se clasifica en una categoría determinada si alcanza un valor máximo de membresía en ésta. Si este valor máximo no fuera lo suficientemente grande (podemos definir un umbral de aceptación), se puede interpretar como que el cromosoma en particular no pertenece a ninguna de las tres categorías. Si el valor máximo fuera igual para más de una categoría, se puede aplicar un sistema de prioridades en la clasificación.

Una ventaja de esta implementación es que se trabaja con la forma de la figura (el esqueleto básico), lo cual hace que ésta no sea afectada por rotaciones, traslaciones, expansiones o contracciones. De esta forma se trabaja con el objeto en sí y no con éste en un contexto previamente definido.

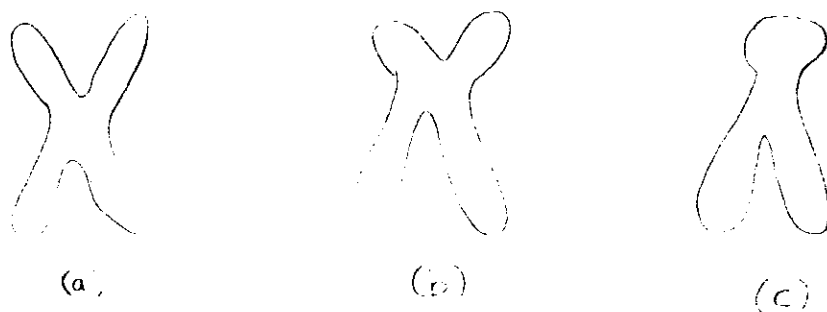


Figura 4.6. Tipos de cromosomas
 a) mediano b) submediano c) acrocéntrico

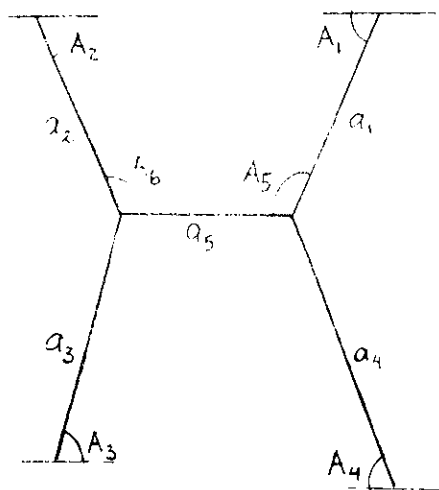


Figura 4.7. Patrón idealizado para la clasificación de cromosomas.

D. Productos comerciales.

Desde hace más de una década que compañías electrónicas en países desarrollados han lanzado al mercado productos en los que interviene la lógica difusa. A continuación se listan algunos de ellos y la forma en que implementan este tipo de lógica en su funcionamiento.

1. Acondicionadores de aire. El modelo CS-XG de la corporación Matsushita incluye 4608 distintas formas de funcionamiento, dependiendo de las variables externas. La compañía Mitsubishi ofrece un modelo que reduce en un 20% el consumo de energía eléctrica comparado con aparatos tradicionales.

2. Lavadoras. La serie Aisaigo Day Fuzzy de Matsushita incluye dos sensores ópticos que determinan la calidad de la ropa y la cantidad de suciedad. Además, es capaz de detectar la cantidad de ropa y el tipo de detergente (líquido o en polvo) y con base en estos datos selecciona el mejor de alrededor de 600 ciclos diferentes de lavado.

3. Aviación. La compañía de aviación Rockwell de los Estados Unidos ofrece un controlador de giro y torsión en aeroplanos, controlado por lógica difusa.

4. Elevadores. La compañía Otis, probablemente la más grande en el mundo en su campo, tiene un modelo que controla su accionar con base en lógica difusa. Un esfuerzo conjunto de Fujitec y Toshiba ha logrado un elevador que evalúa la cantidad de personas esperando frente al aparato. De esta forma, tratan de reducir tiempos de espera y optimizar el servicio.

5. Electrónica de Consumidor. La Sanyo Fisher y la Canon, dos grandes de la industria electrónica, ofrecen al consumidor videocámaras que automáticamente determinan el enfoque y la iluminación adecuada para la imagen presente en pantalla. El modelo H800 de la Canon utiliza un dispositivo de carga acoplada (charge coupled device o CCD) que mide la claridad de una imagen en seis regiones distintas de la pantalla, determinando así si se halla en foco. La cámara también mide la variación en el enfoque (pues la cámara se mueve) y se ajusta para evitar al máximo los sobretiros. La salida consiste en la posición del lente de enfoque, controlada por un motor. Se emplean únicamente 13 reglas de actuación, las cuales ocupan 1.1 kilobytes de memoria.

La Sony Corporation tiene un televisor que ajusta su brillantez, color y contraste automáticamente. De la misma compañía existe una computadora de bolsillo (PCT-500) la cual

elimina el teclado. El ingreso de datos se lleva a cabo por medio de un lapicero que marca sobre una superficie sensora. Por medio de lógica difusa se analizan los patrones de escritura para determinar la entrada. El computador reconoce hasta 3000 diferentes caracteres japoneses como entrada.

6. Transporte colectivo. El sistema de tren subterráneo de Sendai controla la aceleración y desaceleración de sus vehículos. En pruebas realizadas ha mostrado ser tan eficiente que los pasajeros de pie no han necesitado de agarraderas.

7. Automóviles. El conocido modelo Justy de Subaru emplea una transmisión electrónica continuamente variable (electronic continuously variable transmission o ECVT). Esta ajusta continuamente la posición de una faja que varía la palanca de transmisión y por ende la aceleración y desempeño del automóvil. Las reglas de actuación de este sistema se determinaron con base en las respuestas dadas por distintos usuarios a un cuestionario referente al desempeño de vehículos de prueba.

8. Inversiones. La Yamaichi Securities de Japón ha desarrollado un programa que maneja y recomienda diferentes inversiones en la bolsa de valores. El programa

se basa en el comportamiento pasado de determinadas acciones y ha manejado portafolios de 350 millones de dólares con bastante éxito para la compañía.

3. Aplicaciones industriales. La General Electric, en colaboración con el departamento de defensa de los E.E.U.U. ha desarrollado un controlador para el crecimiento de cristales de arseniuro de galio, material muy utilizado para circuitos integrados a gran escala (Very Large Scale Integrated Circuits o VLSIs). Por otra parte, el departamento de investigaciones de la Marina de los E.E.U.U. ha desarrollado un controlador para la formación de aleaciones. Este manipula variables tales como la condición de la superficie, velocidad de deposición de la aleación, temperatura y flujo del metal líquido. Con este controlador se aseguran de buenas aleaciones en sus buques.

La compañía Fuji emplea lógica difusa en la producción de sus rollos para fotografía. El sistema controla el flujo de materiales en la línea de producción. Toshiba tiene un controlador de sistemas de ventilación para túneles de tránsito automovilístico.

E. Sistemas expertos.

Un sistema experto computarizado modela el proceso de razonamiento de un experto humano en un campo específico de conocimiento. Esto permite que un inexperto en la materia pueda acceder y consultar el sistema. Un sistema experto es más que una tabla que hace corresponder una pregunta con su respuesta. Un sistema experto incluye cierto tipo de "entendimiento" de las preguntas y es capaz de realizar inferencias no triviales a partir de la información que contiene y que se le presenta.

La mayoría de sistemas expertos computarizados están diseñados para interactuar directamente con el usuario en forma de diálogo. El usuario provee los parámetros del problema en cuestión y el sistema experto provee de información, juicios o consejos relevantes. Usualmente el sistema es capaz de explicar el proceso de razonamiento empleado para llegar a las conclusiones. Algunos campos sobre los cuales se han diseñado sistemas expertos incluyen el área de diagnóstico y tratamiento médico, química, localización de averías en máquinas, exploración geológica, control de calidad e inversiones en la bolsa de valores.

La mayoría de sistemas expertos consisten de una base de conocimientos específica y de un conjunto de algoritmos de razonamiento, conocidos en conjunto como un aparato de inferencias (inference engine). Algunos sistemas consisten

de una base de conocimientos a la cual se le pueden agregar otras, permitiendo la creación de diferentes "expertos". Además, el sistema ha de tener algún medio de adquisición de conocimiento, que le permita actualizar su base de información constantemente.

Los datos, relaciones, juicios, opiniones, etc. contenidos en la base de conocimientos del sistema experto usualmente manifiestan grados de imprecisión e incertidumbre. La administración de esta incertidumbre en el diseño de un sistema experto es de importancia vital para que este pueda efectuar inferencias que no sean triviales. La utilidad de los conjuntos difusos para llevar a cabo esta labor es evidente y continua siendo estudiada en la actualidad.

El aparato de inferencia de un sistema experto opera con base en una serie de reglas de producción, las cuales conectan antecedentes con sus consecuencias, premisas con sus conclusiones o condiciones con sus acciones. Generalmente son de la forma SI A, ENTONCES B. Estas reglas son fácilmente agregadas, modificadas o removidas de la base correspondiente. Actualmente se emplean dos métodos para evaluar estas reglas de producción. La primera corresponde a la regla de inferencia *modus ponens*. En este caso, los datos se proveen al sistema experto, el cual los utiliza para evaluar las reglas de producción y determinar las posibles conclusiones. Un método alternativo corresponde a la regla

de inferencia modus tollens, la cual es

$$\text{NOT B AND (A} \Rightarrow \text{B) } \Rightarrow \text{NOT A} \quad (4.17)$$

En este caso, el sistema experto busca la información especificada en el antecedente de la regla SI-ENTONCES. Esta información se ha de hallar en la base de conocimiento, en el consecuente de otras reglas, o la ha de proveer el usuario. Este último método es más rápido pues sólo las reglas que lleven al objetivo final son evaluadas. Además, cuando hay información que es difícil de proveer y sólo probablemente necesaria, el método en cuestión es más aplicable.

Las reglas empleadas por expertos humanos son frecuentemente de una naturaleza imprecisa. La necesidad de que existan únicamente dos valores de verdad o el requerimiento de que se satisfagan exactamente los antecedentes de una regla de tipo SI-ENTONCES no parece natural en el contexto del razonamiento humano. Debido a esto se han realizado investigaciones exhaustivas en la aplicación de lógica difusa en el proceso de inferencias. Algunos sistemas expertos bastante exitosos se basan en el método de la regla modus tollens mencionado anteriormente. El sistema hace preguntas cada vez más específicas al usuario, las respuestas de las cuales permiten asignar un valor de verdad a cada una de diferentes posibles respuestas a la pregunta original.

Estas posibles respuestas constituyen un conjunto difuso, donde los grados de membresía de cada una representan la validez de éstas bajo las reglas de producción difusas. Una de las preocupaciones principales en el diseño de sistemas de almacenamiento de información es la eficiencia de éstos. Los sistemas han de actuar lo suficientemente rápido para que puedan interactuar con usuarios humanos, independientemente de la cantidad de información contenida. La utilización de bases de datos difusas y sistemas expertos utilizando lógica difusa es más eficiente con la implementación en circuitos integrados diseñados específicamente para ello. En el apartado I (i latina) de este capítulo se mencionan algunos de estos circuitos integrados.

F. Sistemas de Control.

El esquema general de un sistema de control basado en lógica difusa fue presentado en el capítulo anterior. En este apartado examinamos algunos ejemplos específicos que están relacionados con ingeniería de control.

El propósito del control automático es alcanzar y mantener un estado de un proceso o sistema mediante el monitoreo de ciertas variables de estado y tomando las acciones de control adecuadas. El diseño de un controlador automático efectivo requiere de un modelo matemático preciso del sistema o proceso involucrado. Para una gran cantidad de procesos complejos, la construcción de este modelo es difícil debido a comportamientos no lineales y variables en el tiempo o imprecisión en las mediciones disponibles. En estos casos es conveniente la implementación de lógica difusa en los controladores en cuestión.

1. Control de Tráfico. El control automático de señales de tráfico basadas en un modelo estocástico ha sido utilizado desde hace varios años en algunas ciudades. Simulaciones computarizadas de implementaciones de lógica difusa han mostrado que se mejoran los tiempos de retardo promedio en comparación con sistemas convencionales. Las reglas de control difusas imitan el protocolo utilizado por un operador humano para decidir los intervalos de tiempo de

luces verdes opuestas. Por ejemplo, si hay pocos carros esperando por una luz verde en una calle, el operador puede retardar el cambio de la luz en la avenida hasta que el tráfico en ésta sea más bajo o hasta que la luz roja en la calle haya estado encendida bastante tiempo. Los términos en negrillas constituyen los nombres de conjuntos difusos definidos en el universo de número de carros (pocos, más bajo) y tiempo (bastante).

La aplicación de reglas de actuación también ha de depender de las condiciones de tráfico existentes. Por ejemplo, el hecho de que hayan algunos carros esperando una luz roja provocará distintas respuestas dependiendo si es de noche o de día. El control óptimo ha de permitir flexibilidad en el conjunto de reglas.

Mamdani y Pappis presentaron en 1977 un controlador de lógica difusa para una intersección de calle y avenida, ambas de una vía. Las reglas empleadas tienen la forma

SI	T = medio
Y	A = ma (medio)
Y	Q = me (pocos)
ENTONCES	E = medio

EN OTRO CASO,

SI	T = mucho
Y	A = ma (muchos)
Y	Q = me (medio)
ENTONCES	E = mucho

Las variables de entrada T, A y Q representan el tiempo desde el último cambio, el número de carros que pasan por la luz verde y el número de carros esperando en la vía opuesta, respectivamente. La variable de salida E representa la extensión de tiempo que transcurre antes de cambiar la luz. Los símbolos μ_m y μ_{me} equivalen a los términos más que y menos que, respectivamente. Los términos como más que muchos y menos que poco se modelan como un conjunto difuso en su universo correspondiente. Si el grado de membresía de la variable x en el conjunto A (donde A representa un conjunto como muchos, pocos, etc.) está dado por $\mu_A(x)$, el grado de membresía de x en el conjunto más que A está dado por

$$\mu_{ma}(A)(x) = \begin{cases} 0 & x \leq x_0 \\ 1 - \mu_A(x) & x > x_0 \end{cases} \quad (4.18)$$

donde x_0 es el elemento con máximo valor de membresía en A. El grado de membresía de x en el conjunto menos que A está dado por

$$\mu_{me}(A)(x) = \begin{cases} 0 & x \geq x_0 \\ 1 - \mu_A(x) & x < x_0 \end{cases} \quad (4.19)$$

Cada diez segundos un conjunto diferente de 5 reglas de control es evaluado cada segundo para determinar el tiempo de mantenimiento del estado actual de las luces. La detección de las variables de entrada se lleva a cabo por medio de transductores de presión que detectan el paso de un automóvil. El controlador mantiene la luz en un estado un máximo de 57 segundos. Para determinar la salida no difusa se utiliza una tabla de decisión que asigna a cada valor de duración un grado de confianza con el que éste puede ser tomado dadas las condiciones actuales. El controlador escoge el tiempo de duración que maximiza el nivel de confianza. Simulaciones de este controlador comparado con uno que no utiliza lógica difusa muestran que el primero reduce los tiempos de espera para cada automóvil. Para llegar a este punto, sin embargo, se tuvieron que modificar las reglas de actuación a prueba y error.

El concepto de este controlador fue extendido a dos intersecciones por Nakatsuyama, Nagahashi y Nishizuka en 1984. Nuevamente, el controlador difuso tuvo como resultado menores tiempos de espera para los automóviles. A partir de estos resultados, los investigadores mencionados anteriormente introdujeron un controlador de fase de lógica difusa. El controlador de fase sirve para coordinar intersecciones consecutivas, provocando cambios con retardos fijos. Esto minimiza el tiempo de recorrido para el automóvil promedio.

De esta forma, si la luz en la intersección 1 cambia de rojo a verde en el tiempo t , la luz en la intersección 2 cambia a verde en el tiempo $t+n$, donde n es un retardo computado con base en las condiciones de tráfico actuales. De la misma forma se calcula el retardo m para cambiar la luz verde a amarillo.

Los controladores mencionados (el de lógica difusa y el de fase) tienen distintos resultados, dependiendo de las condiciones de tráfico. Para maximizar la eficiencia del sistema, se incorporó una regla de actuación adicional, que selecciona el tipo de controlador (sea el de fase o el otro) con base en las condiciones de tráfico existentes.

2. Control de aeroplanos. El acercamiento y aterrizaje de un aeroplano involucra el maniobramiento del aparato para la manutención de una trayectoria adecuada hacia la pista de aterrizaje. Esta trayectoria es provista al aviador por medio de un sistema de aterrizaje por instrumentos, el cual transmite dos señales de radio al aeroplano que le sirven de referencia. El piloto aviador se basa en estas señales para corregir la posición y velocidad del aparato. Las correcciones se efectúan por aproximaciones sucesivas con base en reglas no escritas conocidas por todo aviador experto.

Un controlador difuso puede simular fácilmente estas reglas de forma que el aterrizaje pueda ser efectuado automáticamente. Los valores de entrada son la velocidad de descenso, la desviación de la velocidad requerida y la desviación de la trayectoria dada por las señales de radio. Las salidas son el cambio en la velocidad del motor y el cambio en la elevación del avión. Larkin en 1985 presentó un modelo de controlador automático en donde todas las variables toman cinco posibles valores difusos: negativo grande, negativo medio, cambio insignificante, positivo medio y positivo grande. Las reglas de actuación las proporcionó un aviador experimentado. La salida no difusa se determina con base en el centroide de las salidas difusas.

G. Medicina

La imprecisión y la incertidumbre juegan un papel muy importante en el campo de la medicina. Por esta razón la teoría de conjuntos difusos ha tendido bastante aplicación. Usualmente estas aplicaciones están implementadas por medio de una computadora.

1. Diagnóstico. El proceso de diagnóstico médico es el que ha sido estudiado con mayor frecuencia. Con el incremento en la cantidad de información disponible a los profesionales médicos, el proceso de clasificación de diferentes conjuntos de síntomas bajo una categoría y la determinación de la acción médica correcta se hace cada vez mas difícil.

Una enfermedad se puede manifestar de diferentes formas en distintos pacientes y etapas de desarrollo. Además, un síntoma en particular puede ser común a diferentes enfermedades. Una descripción médica adecuada de síntomas y enfermedades utiliza términos lingüísticos y es necesariamente vaga. Tanto el conocimiento teórico del profesional como su conocimiento de la historia clínica del paciente constituyen fuentes de imprecisión e incertidumbre. Errores en los exámenes médicos, imprecisiones en procedimientos tales como rayos X y ultrasonido, exageraciones de pacientes en cuanto a su historial médico,

etc. constituyen sólo algunos de los tantos factores que complican el diagnóstico. Desde hace años han surgido modelos de diagnóstico basados en conjuntos difusos. En el modelo presentado por Sánchez (1979), el conocimiento médico del profesional está representado como una relación difusa entre síntomas y enfermedades. Así, dado un conjunto difuso A de los síntomas observados en el paciente y la relación R representando el conocimiento médico que relaciona los síntomas (tomados del universo S) con las enfermedades (tomadas del conjunto E), se obtiene un conjunto difuso de posibles enfermedades B del paciente por medio de la regla de composición

$$B = A \cdot R \quad (4.20)$$

ó

$$\mu_B(e) = \max_{s \in S} [\min(\mu_A(s), \mu_R(s,e))] \quad (4.21)$$

para todo $e \in E$. Los grados de membresía de los síntomas observados en el conjunto difuso A representan el grado de certeza de la presencia de un síntoma o la severidad de éste. Los grados de membresía en el conjunto difuso B denotan el grado de certeza con el que podemos asignar un diagnóstico, en particular a un paciente.

Este modelo es elaborado por Adlassnig (1982) para el diseño de CADIAG-2 un sistema computarizado de diagnóstico. El modelo propone la existencia de dos tipos de relaciones

El modelo propone la existencia de dos tipos de relaciones entre síntomas y enfermedades: una relación de ocurrencia y una de confirmabilidad. La primera provee información acerca de la tendencia o frecuencia de apareamiento de un síntoma cuando la enfermedad dada está presente. La segunda relación describe el poder de discriminación implícito en el síntoma para confirmar la presencia de la enfermedad. La distinción entre ocurrencia y confirmabilidad es útil porque un síntoma dado puede ocurrir con una enfermedad dada pero también puede ocurrir con otras, disminuyendo así su poder como factor discriminante. Otro síntoma bien puede ser raro en un tipo de enfermedad, pero su presencia puede constituir una confirmación casi segura de la presencia de la enfermedad. Las relaciones se construyen basadas en conocimiento de los pacientes. Debido a que las descripciones empleadas suelen utilizar modificadores verbales tales como "muy" (por ejemplo, la aparición de ronchas es muy frecuente en casos de SIDA) se define una operación que modela este modificador así:

$$\mu_{\text{muy } A}(x) = \mu_A(x), \quad (4.22)$$

donde A es un conjunto difuso cualquiera.

Con base en la representación matricial de cada relación, y la composición mencionada al principio de este apartado, se pueden construir matrices que representan la indicación de

ocurrencia y la indicación de confirmabilidad (componiendo la relación pacientes-síntomas con la relación síntomas-ocurrencia y la relación síntomas-confirmabilidad, respectivamente). Pueden también determinarse relaciones de no ocurrencia y no confirmabilidad, tomando el negado de las relaciones correspondientes.

Una aplicación específica en diagnóstico médico involucra la determinación de estenosis (angostamiento) de arterias que transportan sangre al miocardio (la pared muscular central del corazón). Esta es una de las principales causas de ataques cardíacos. Por lo tanto es importante determinar cuán severo es un bloqueo en las arterias y cuáles de éstas son las anormales.

Utilizando talio 201 se puede determinar la distribución de flujo sanguíneo en el corazón. Sin embargo, la relación entre esta distribución y la localización de estenosis en las arterias no está claramente definida. Cios, Shin y Goodenday (1991) diseñaron un sistema de diagnóstico de tipo difuso (fuzzy) que permite determinar la estenosis arterial con base en el análisis de la distribución de flujo sanguíneo. Se generaron 15 reglas de actuación (en este caso de clasificación) con base en experiencias médicas anteriores. Los investigadores experimentaron con distintos tipos de reglas hasta llegar a la estructura óptima. En ésta, se tienen tres

antecedentes y un consecuente. Con más antecedentes, las reglas eran muy específicas e incrementaban grandemente el número de reglas. Con menos antecedentes se tenían reglas muy generales que no efectuaban adecuadamente la labor de diagnóstico.

2. Otras aplicaciones. Además de ser herramienta útil en el diagnóstico de enfermedades, la lógica difusa se utiliza en procesos tales como la administración adecuada de anestesia durante operaciones. En un artículo escrito por Meier, et. al, (1992) se presenta el diseño de un controlador que efectúa esta labor.

Una de las labores principales del anestesista durante cirugía es el control del efecto de la anestesia. Este efecto no es medible fácilmente; en la práctica se evalúa con base en presión sanguínea, ritmo circulatorio y otras señales clínicas como el tamaño de la pupila, actividad motriz, etc. El control adecuado de los niveles de anestesia es, en resumidas cuentas, un asunto muy delicado. Se han propuesto varios métodos para la simplificación de la medición del efecto de la anestesia, pero los anestesistas aún utilizan la presión sanguínea como parámetro para la dosificación de anestésico inhalados. El trabajo de Meier, et al., consistió en desarrollar un controlador encargado de medir la presión media arterial de los pacientes. De esta forma, el

anestesiista puede concentrarse en otras labores, tales como la administración de la droga, control de ventilación, etc. En el trabajo de Meier, la presión arterial promedio se retroalimenta al sistema y se compara con una presión teórica. Las variables de entrada al controlador difuso son el error, su derivada y su integral. Aquí se toman en cuenta disturbios tales como incisiones en la piel (las cuales provocan cambios rápidos en la presión sanguínea) y mediciones de otros aparatos electrónicos. La salida del controlador es la concentración del agente anestésico entregada al paciente. Las funciones de membresía empleadas no son trapezoidales sino en forma de campana gaussiana (se basan en datos obtenidos de operaciones reales). Las características de control fueron simuladas en computadora para determinar las reglas de actuación óptimas. La implementación se lleva a cabo por medio de una computadora personal que colecta los datos de entrada, los despliega, controla el proceso y almacena los datos.

Este controlador se ha empleado experimentalmente en 11 casos de cirugía abdominal. Los supervisores no necesitaron intervenir en el control de la concentración de anestésico. De hecho, el controlador mostró ser más efectivo que el control manual. Experimentos con más pacientes y en mayor diversidad de condiciones presuponen la implementación futura de estos controladores en hospitales alrededor del mundo.

H. Implementación en sistemas sencillos

Así como los sistemas basados en la lógica digital se basan en un número reducido de componentes, es posible identificar bloques básicos para la construcción de sistemas de lógica difusa. La forma más sencilla consiste en construir las funciones elementales de la lógica difusa utilizando componentes electrónicos comunes. A continuación se presenta el diseño de un inversor, la conjunción y la disyunción de la lógica difusa. Por estar trabajando con un intervalo continuo de valores, es de esperarse que estos circuitos contengan componentes analógicos (amplificadores operacionales por ejemplo). Estos circuitos trabajan con niveles TTL, siendo 5 V. el '1' lógico y 0 V. el '0' lógico. Los valores intermedios se corresponden linealmente. Así 2.5 V. han de equivaler al '1/2' lógico. Por supuesto, estos circuitos tienen como caso especial los de la lógica bivalente.

1. Inversor. Como se recordará del capítulo II, el inversor obedece la siguiente ecuación:

$$t(\text{NOT}(A)) = 1 - t(A) \quad (4.23)$$

donde A es una proposición y t es un operador que devuelve el valor lógico de una proposición. Un amplificador diferencial

común puesto en configuración de sustractor simula la función del inversor. En la figura 4.8. se puede apreciar el circuito. Se puede ver que es necesaria la introducción de voltajes diferentes de alimentación (V_{cc} , $-V_{ee}$). Esto generalmente no es problema en aplicaciones grandes.

2. Conjunción (AND lógico). La conjunción lógica se trabaja con el operador min (mínimo).

$$t(A \text{ AND } B) = \min(t(A), t(B)) \quad (4.24)$$

Un ejemplo de compuerta AND para dos entradas se muestra en la figura 4.9. En este caso, los dos valores sirven de entrada a un comparador (como lo es el ECG 834). Los voltajes de alimentación al comparador han de ser 5 V. y tierra. Con esto, se obtienen 5 V. a la salida del comparador si la entrada negativa es la menor. Estos 5 V. activan el interruptor analógico (en este caso ECG 4016B) que permite el paso de esta entrada hacia la salida. En el caso contrario, el interruptor analógico de la otra entrada es el activado. La utilización de un inversor digital (como el ECG 7404) garantiza que siempre habrá únicamente un interruptor analógico activado, siendo la salida el mínimo de las dos entradas.

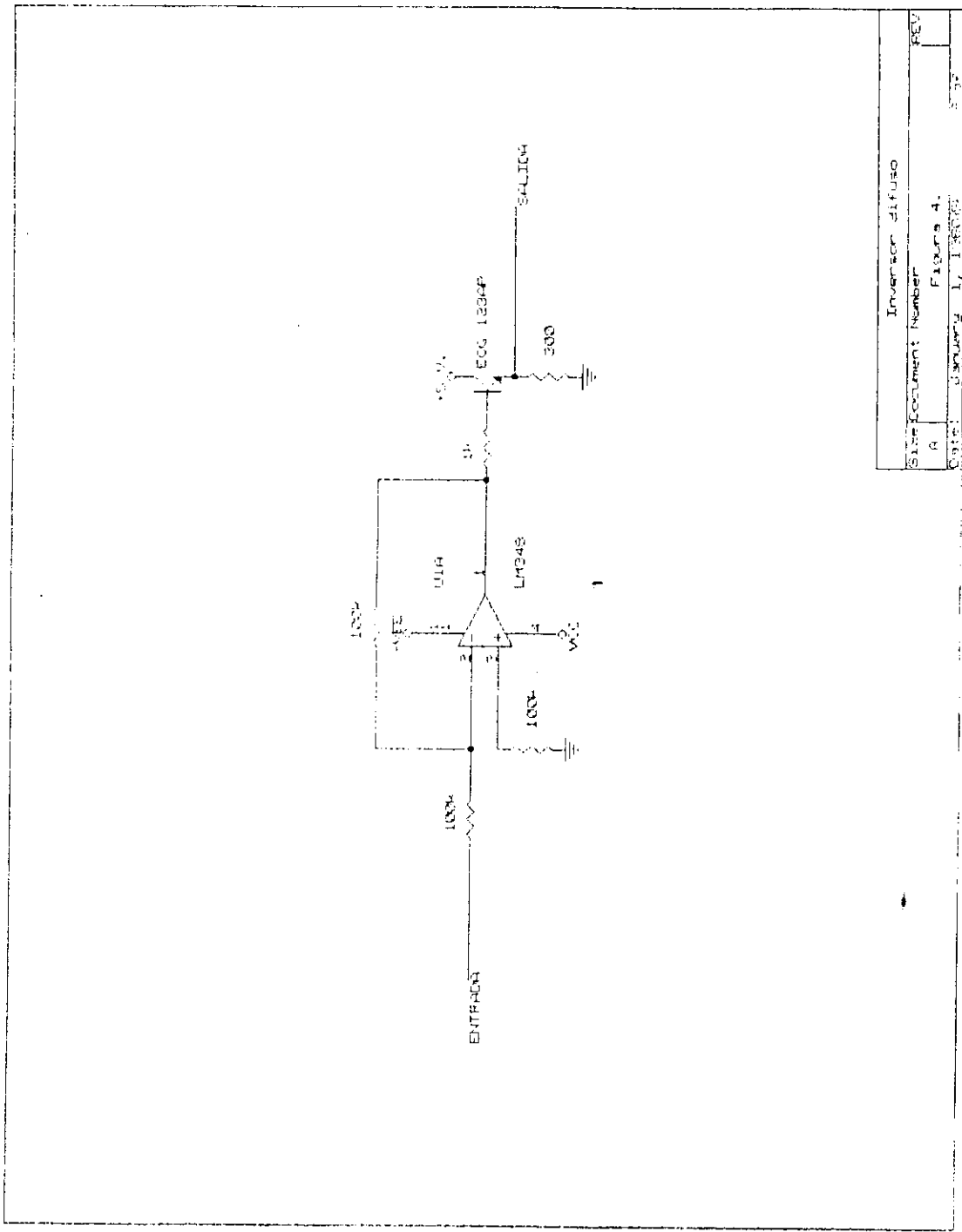
3. Disjunción (OR lógico). El caso de la disjunción es similar al de la conjunción. En este caso se utiliza la función max (máximo).

$$t(A \text{ OR } B) = \max (t(A), t(B)) \quad (4.25)$$

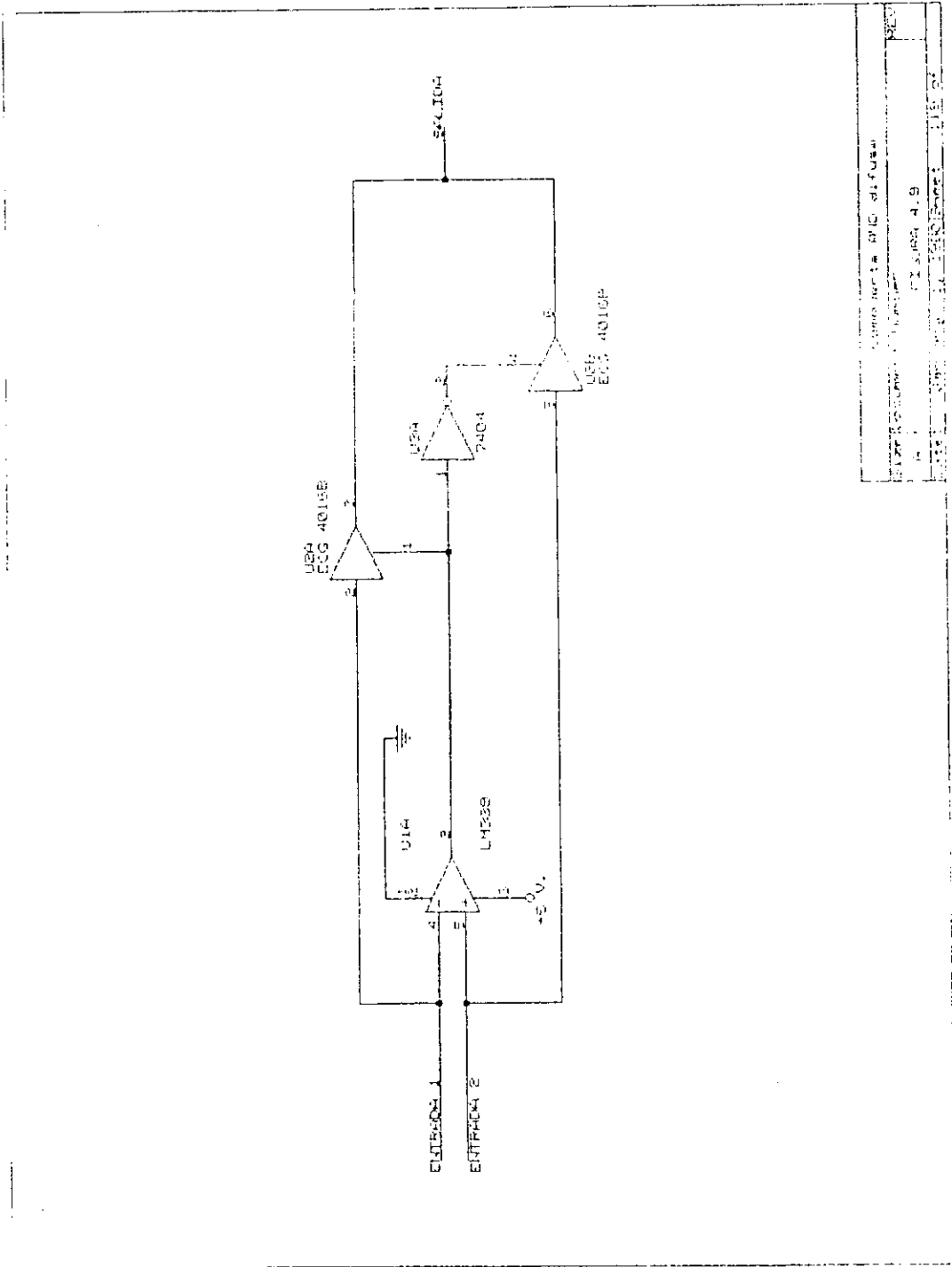
El circuito que simule esta función ha de ser similar al de la conjunción. En este caso, sin embargo, se ha de activar la salida que sea máxima. Esto se puede lograr cambiando de lugar al inversor digital o invirtiendo el orden de las entradas. La figura 4.10 muestra el circuito.

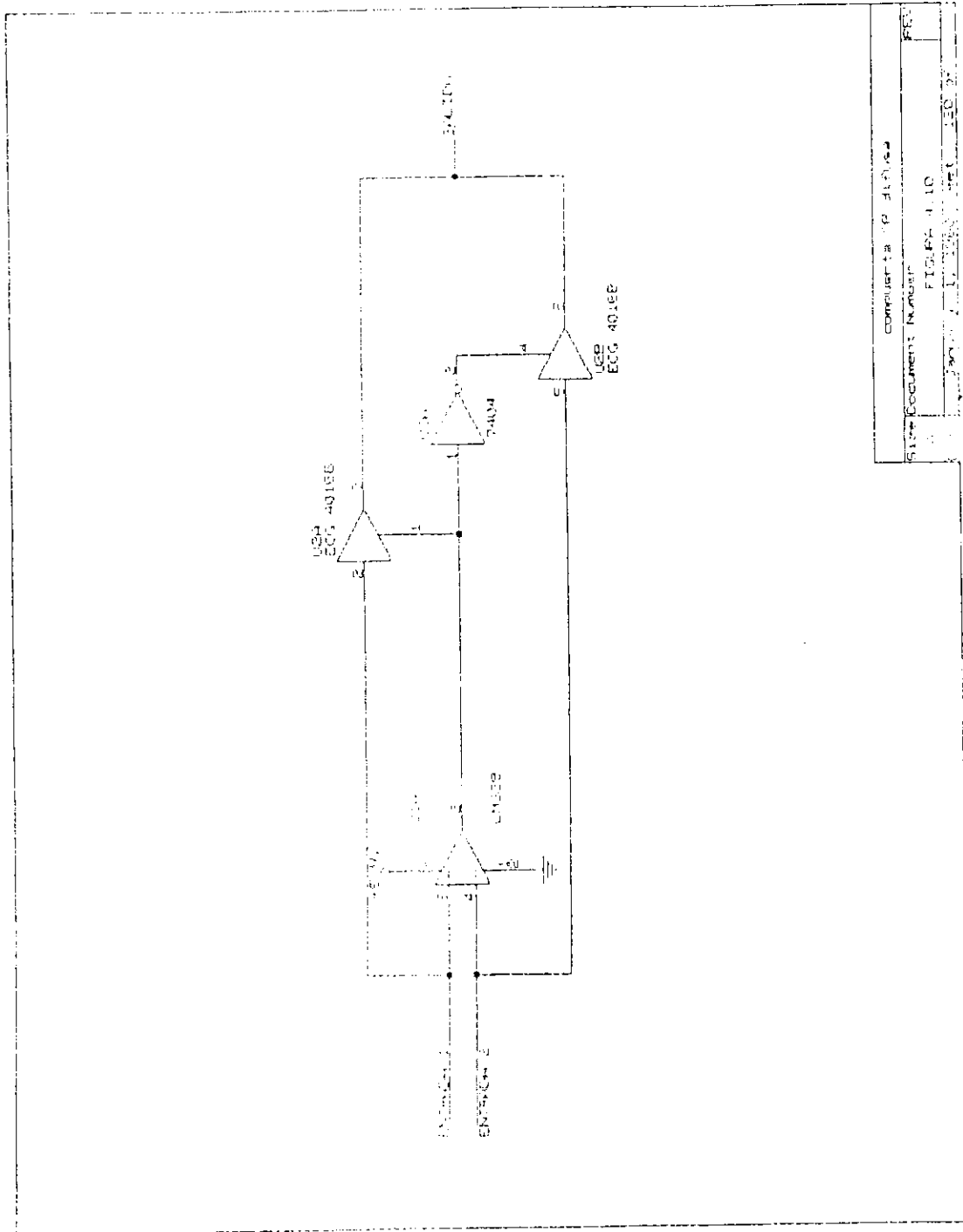
Con los circuitos anteriores se puede trabajar en lógica difusa a nivel de proposiciones. Para poder trabajar con funciones de membresía ha de existir alguna medio que las almacene. La implementación más sencilla consiste en almacenar las funciones en memorias digitales. Para esto, se ha de convertir cada entrada primero a un valor de voltaje (por medio de transductores) y luego a un valor digital, al cual le corresponde un valor de membresía en un valor de variable lingüística (por ejemplo, 15 °C puede tener un grado de membresía 0.8 en el conjunto "tibio"). En la figura 4.11. se observa esta implementación. El convertidor analógico a digital puede ser de cualquier tipo y configurado de forma

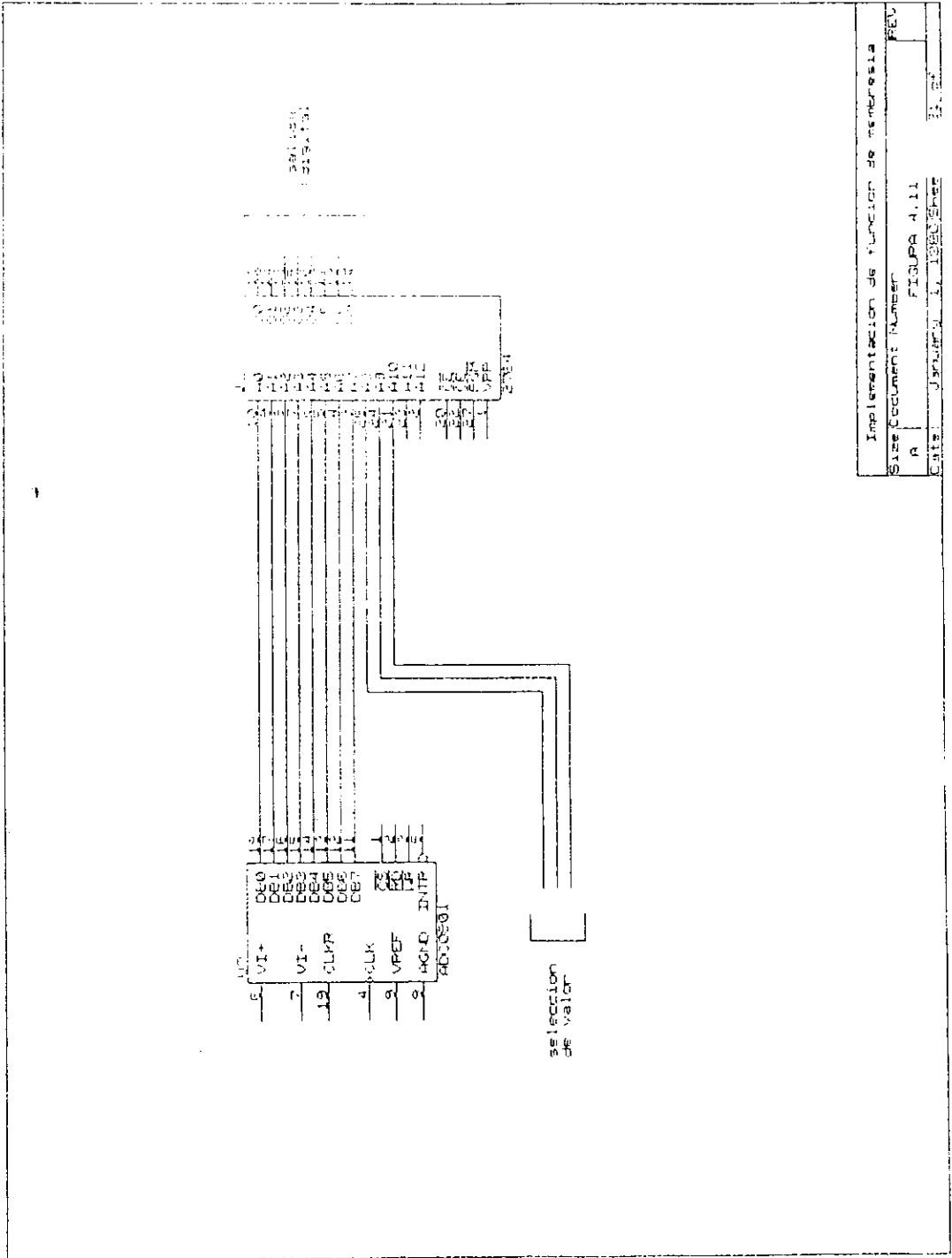
que efectúe conversiones continuamente. Si se codifican los valores con n bits, la memoria ha de tener al menos $n+3$ bits para direccionamiento (debido a que cada variable lingüística suele tener entre 3 y 7 valores posibles). Los valores de selección corresponden al valor difuso (fuzzy) correspondiente, ordenados de menor a mayor. Por tener un rango de valores continuos a convertir, se pueden almacenar secuencialmente valores de función de membresía en las posiciones de memoria correspondientes. Por ejemplo, si estamos midiendo temperaturas de 0°C a 100°C , cuantificándolas en 8 bits, y clasificándolas en los valores "frío", "tibio", "caliente", "hirviendo" (codificados como 000, 001, 010, 011, respectivamente), el valor de membresía de 50°C en "caliente" se halla en la posición 0011 1000 0000 de memoria. Aquí, los 8 bits inferiores corresponden al valor digitalizado de la entrada y los 4 superiores al valor difuso en cuestión. De esta forma, se codifica digitalmente un conjunto complicado de funciones de membresía, correspondientes a los valores de una variable lingüística con dos circuitos integrados de uso común.



Inversor difuso	
Sheet Number:	Figura 4.
REV	
DATE:	January 17, 1992







La implementación de un sistema de control basado en lógica difusa utilizando componentes electrónicos comunes es bastante sencilla. El almacenamiento de funciones de membresía mostrado anteriormente permite que la evaluación de reglas la haga un microprocesador digital. Con un puerto de entrada, uno de salida y uno o dos de líneas de control se puede implementar un sistema de control generalizado para cualquier aplicación (si se tiene más de una salida se incrementan los puertos de salida). Un circuito típico simplificado se muestra en la figura 4.12. Se asume que existen transductores adecuados que convierten señales de voltaje o corriente en magnitudes no eléctricas y viceversa. En la figura se muestran sólo los circuitos y las conexiones más importantes. Se excluye circuitería de decodificación que depende de los componentes y microprocesador utilizados. El convertidor analógico-digital mostrado acepta hasta 8 entradas diferentes. En caso se requirieran más entradas, simplemente se agregan convertidores y se modifica la circuitería de decodificación correspondiente. Para el ejemplo supondremos que se tienen 3 entradas y 1 salida (un número de valores típico de un sistema de control sencillo). El componente U3 es una memoria de sólo lectura (o ROM por sus siglas en inglés) que contiene la información de las funciones de membresía de los diferentes valores de entrada y de salida. Los bits de direccionamiento (A_{12} , A_{11} , ..., A_1 ,

A₀) contienen un byte de información de forma que:

- * A₁₂A₁₁ = 11 corresponde a la salida del sistema
- * A₁₂A₁₁ = 00, 01, 10 corresponde a las entradas 1, 2, 3, respectivamente.

- * A₁₀A₉A₈ codifica uno de hasta ocho posibles valores difusos de la variable codificada en A₁₂A₁₁ (por ejemplo, para la variable temperatura, "congelado" sería 000, "frío" sería 001, "tibio" sería 010, "caliente" sería 011, etc.)

- * Los restantes 8 bits codifican lo siguiente:

Si es una variable de entrada: la dirección contiene el valor de la variable (que viene del A/D), y la memoria contiene el dato del grado de membresía en el conjunto difuso indicado por A₁₀A₉A₈.

Si es una variable de salida: la dirección contiene un grado de membresía (que viene del microprocesador) y la memoria devuelve el valor digital del área acotada por este grado de membresía.

El circuito integrado U8 es un registro de 8 bits que

permite el ingreso de la parte baja de la dirección en U3 por parte del microprocesador. Este registro habilita su salida cuando se están determinando las áreas acotadas en las funciones de membresía. Por esta razón su habilitación es invertida respecto del convertidor, por medio de U7.

Supongamos que se tiene una aplicación particular. El control de la velocidad con que se desplaza un tren recibe entradas de cantidad de personas, velocidad instantánea y aceleración instantánea para determinar la potencia entregada al motor principal. Asumiremos que los valores de cantidad de personas son "bajo", "regular", "alto" y "sobrecarga"; los de velocidad son "baja", "media", "alta"; y los de aceleración son "desacelerando", "nulo" y "acelerando". Así, para cantidad de personas tendremos los siguientes conjuntos difusos: cant_baj, cant_reg, cant_alt y cant_sob. Para velocidad tendremos los conjuntos vel_baj, vel_med y vel_alt. Para aceleración tendremos los conjuntos acel_neg (desacelerando), acel_nul (velocidad constante) y acel_pos (acelerando). La salida es la potencia entregada al motor del elevador y tendrá los valores pot_nul (no entregar potencia), pot_baj (entregar poca potencia), pot_med (entregar regular potencia) y pot_alt (entregue el máximo de potencia).

La cantidad de pasajeros se mide con base en la

diferencia de peso existente entre el tren en ese instante y el tren vacío. La velocidad y aceleración también pueden convertirse a señales eléctricas con transductores adecuados.

Las reglas de actuación en este caso las determinamos con un poco de sentido común, de forma que:

```
SI cant_baj y vel_baj ENTONCES pot_baj
SI cant_reg y acel_neg ENTONCES pot_med
SI cant_alt y acel_neg ENTONCES pot_alt
SI vel_alt y acel_nul ENTONCES pot_nul
SI vel_med y acel_pos ENTONCES pot_baj
SI cant_sob y vel_med ENTONCES pot_med
SI vel_baj o acel_neg ENTONCES pot_alt
```

Probablemente se necesite algunas reglas más para tener un sistema más robusto, pero en este caso las siete reglas cubren todos los valores difusos posibles de salida y entrada. Además, con excepción de la salida pot_nul (correspondiente a no entregar potencia al motor), todos los valores posibles de salida están representados equitativamente. Así, el programa que ejecuta el microprocesador corresponde al siguiente algoritmo (en pseudo-código):

```

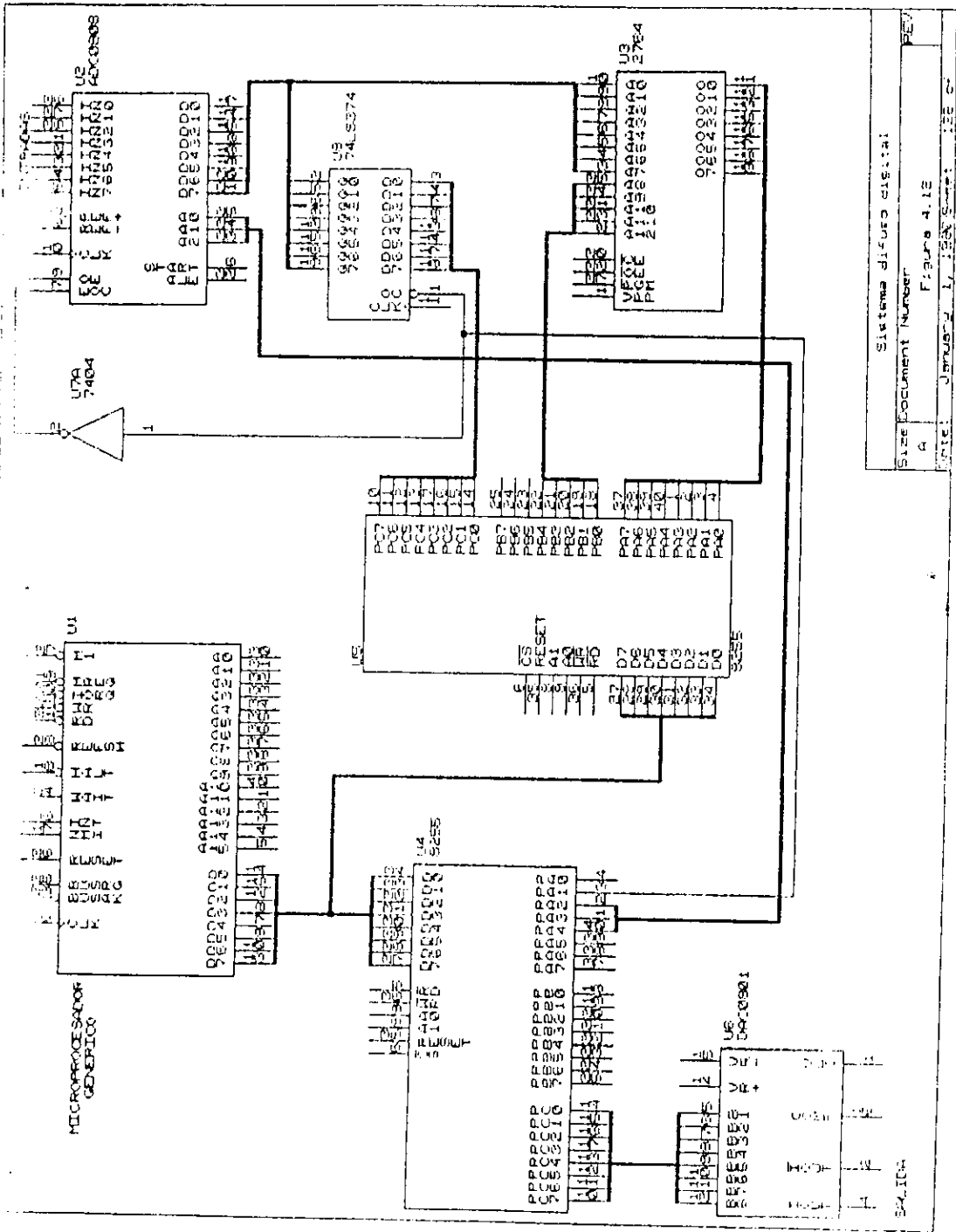
* 1. Leer cant_baj, cant_reg, cant_alt, cant_sob
* 2. Leer vel_baj, vel_med, vel_alt
* 3. Leer acel_neg, acel_nul, acel_pos
* 4. p1:= min (cant_baj, vel_baj)
* 5. p2:= min (cant_reg, acel_neg)
* 6. p3:= min (cant_alt, acel_neg)
* 7. p4:= min (vel_alt, acel_nul)
* 8. p5:= min (vel_med, acel_pos)
* 9. p6:= min (cant_sob, vel_baj)
* 10. p7:= max (vel_baj, acel_neg)
* 11. Leer a1,...,a7
* 12. A:= kb*a1+km*a2+ka*a3+kn*a4+kb*a5+km*a6+ka*a7
* 13. B:= a1+a2+a3+a4+a5+a6+a7
* 14. out:= A/B
* 15. Escribir out al puerto correspondiente
* 16. Regresar a 1.

```

Los pasos 1, 2 y 3 permiten la adquisición de los datos de entrada. Cada orden de "Leer" corresponde a una escritura al puerto de control para seleccionar la entrada y valor difuso correspondiente. Los pasos del 4 al 10 evalúan las reglas de actuación. El paso 11 determina las áreas acotadas por los grados de membresía de cada regla de actuación. Los pasos del 12 al 14 calculan la salida con base en la técnica de centroide, donde kb, km, ka, kn son los valores de

potencia correspondientes al centro de las áreas delimitadas por las funciones de membresía de potencia baja, media, alta y nula, respectivamente.

Como se puede ver, el soporte lógico (software) necesario para un sistema difuso es reducido y sencillo. Simplemente se limita a tomar datos y evaluarlos con funciones sencillas (max, min). El número de operaciones aritméticas complejas (división, multiplicación) es reducido e incluso se podría efectuar externamente por medios analógicos. En este caso se encomendó la tarea al microprocesador para simplificar el soporte físico (hardware) empleado.



Systems disk digital
 Size Document Number
 4 Figure 4.12
 Date: January 17, 1988 Sheet: 22 of 22

I. Avances recientes

La aplicación de la lógica difusa en diferentes productos y procesos ha incrementado constantemente durante los últimos años. Para el año 1992 se habían proyectado ganancias de alrededor de 350 millones de dólares relacionadas con lógica difusa. Japón tiene alrededor del 80% del mercado mundial, pues el gobierno ha apoyado el desarrollo de esta tecnología desde sus inicios. El consorcio más grande dedicado a la investigación en este campo es el laboratorio internacional de ingeniería difusa (Laboratory for International Fuzzy Engineering) localizado en Yokohama. Este consorcio fue organizado por el ministerio de comercio internacional e industria del Japón y agrupa alrededor de 45 compañías japonesas y subsidiarias japonesas de compañías estadounidenses y europeas. Incluye gigantes como Hitachi Ltd., Sony Corporation, IBM de Japón y Thomson de Japón.

En Estados Unidos el desarrollo no ha sido tan impresionante pero poco a poco surge una gran cantidad de proyectos relacionados con la lógica difusa. Motorola, en colaboración con Aptronix, introdujo el año pasado un ambiente de programación e inferencias difusas, más conocido como FIDE (por Fuzzy Inference Development Environment). Existe ya una gran cantidad de programas diseñados para trabajar en este ambiente. Por aparte, Motorola ofrece herramientas más específicas. Existen generadores de código

especiales para trabajar con los microcontroladores 68HC05 y 68HC11 (que son microprocesadores de uso específico para control de sistemas).

Existen otros microprocesadores que trabajan con lógica difusa. El primero de ellos data de los años setenta y fue desarrollado por Togai Infralogic, una compañía estadounidense. La únicas operaciones que incluía eran el máximo y el mínimo, utilizadas en conjuntos difusos. Yamakawa diseñó y fabricó en 1987 un microprocesador que trabaja en forma analógica y es capaz de trabajar en paralelo. El mismo Yamakawa presentó recientemente su neurona difusa (fuzzy neuron) la cual es una neurona que acepta conjuntos difusos como entrada y genera conjuntos difusos de salida. Esta neurona difusa puede ser la base de sistemas difusos más grandes. También en Japón, Kaoru Hirota ha diseñado un balancín (flip-flop) difuso, el cual incluye el balancín binario como caso especial. Basado en este diseño se está discutiendo la posibilidad de desarrollar una computadora totalmente difusa (fuzzy), conteniendo el proceso de información en forma binaria como un caso especial.

V. BIBLIOGRAFIA

- Bueno, E. Lógica polivalente. La Habana, Editorial de Ciencias Sociales. 180 pp.
1976
- Cios, K., I. Shin y L. Goodenday. "Using Fuzzy Sets to Diagnose Coronary Artery Stenosis". IEEE computer magazine. 24 (3): 57-63.
1991
- Coughlin, B. y F. Briscoe. Circuitos integrados lineales y amplificadores operacionales. México, D.F., Prentice-Hall Hispanoamericana, S.A. 394 pp.
1987
- Cox, E. "Fuzzy Fundamentals". IEEE spectrum. (E.E.U.U.) 29 (10): 58-61.
1992
- Dougherty, E. y C. Giardina. Mathematical methods for artificial intelligence and autonomous systems. Nueva Jersey, Prentice-Hall, Inc. 446 pp.
1988
- Kaufman, M. y A. Seidman. Manual para ingenieros y técnicos en electrónica. México, D. F., McGraw-Hill/ Interamericana de México. 629 pp.
1988
- Kaufmann, A. y M. Gupta. Introduction to fuzzy arithmetic. New York, Van Nostrand Reinhold. 361 pp.
1991
- Klir, G. y T. Folger. Fuzzy sets, uncertainty, and information. New Jersey, Prentice-Hall, Inc. 355 pp.
1988
- Kosko, B. Neural networks and fuzzy systems. New Jersey, Prentice-Hall, Inc. 449 pp.
1992
- Lin, C. y C. Lee. "Neural-Network-Based Fuzzy Logic Control and Decision System". IEEE transactions on computers. 40 (12): 1320-1336.
1991
- Mano, M. Lógica digital y diseño de computadores. México, D. F., Prentice-Hall Hispanoamericana, S.A. 635 pp.
1982
- Meier, R.; J. Nieuwland, A. Zbinden y S. Hacisalihzade. "Fuzzy logic control of human blood pressure during anesthesia". IEEE control systems magazine. 12 (10): 12-16.
1992

- Moore, K. "Artificial neural networks". IEEE potentials.
1992 (E.E.U.U.) 11 (1): 23-28.
- Negoita, C. y D. Ralescu. Simulation, knowledge-based
1987 computing, and fuzzy statistics. New York, Van
Nostrand Reinhold. 158 pp.
- Rolston, D. W. Principios de Inteligencia Artificial y
1992 Sistemas Expertos. Bogotá, McGraw-Hill. 177 pp.
- Schwartz, D. y G. Klir. "Fuzzy logic flowers in Japan".
1992 IEEE spectrum. 29 (7): 32-37.
- Self, K. "Designing with fuzzy logic". IEEE spectrum.
1990 27 (11): 42-44, 105.
- Suger, E., B. Morales y L. Pinot. Introducción a la
1981 matemática moderna. México, D. F., Editorial
Limusa. 289 pp.
- Tanaka, T., T. Tsutsui y R. Watanabe. "Digital multi-channel
1993 television signal transmission system". 47th
Annual Broadcast Engineering Conference
Proceedings (Las Vegas).
- Zadeh, L.A. "Fuzzy logic". IEEE computer. 21 (4): 83-92.
1988
- Zimmerman, H. "Modelling flexibility, vagueness and
1988 uncertainty in operations research". Investigación
operativa (Brasil). 1 (1): 7-34.

APENDICE A

Números Difusos (Fuzzy Numbers)

Sea A un subconjunto difuso de el conjunto de los números reales (representado por R). Este subconjunto se llama convexo si para todo x_1, x_2 que pertenecen a R se cumple

$$\mu_A[\tau x_1 + (1-\tau)x_2] \geq \min [\mu_A(x_1), \mu_A(x_2)] \quad (A.1)$$

para todo $\tau \in [0,1]$. De hecho, el lado derecho de la desigualdad se puede substituir por cualquier operación que cumpla con las condiciones de una intersección en conjuntos difusos.

Además, un subconjunto es normal si y solo si

$$\max_{x \in R} \mu_A(x) = 1 \quad (A.2)$$

En este caso se puede substituir el operador \max por cualquier otro que cumpla con las condiciones para la unión de conjuntos difusos.

Definimos un número difuso en R como un subconjunto de R , el cual es convexo y normal. De esta forma, el número difuso es una generalización de un intervalo de confianza.

El número difuso difiere de una variable aleatoria debido a su subjetividad. Mientras que la variable aleatoria es producto de una medición objetiva, el número difuso es una valoración subjetiva. Esta valoración depende generalmente de el contexto en el cual se esté trabajando.

Así, el número difuso es simplemente un valor que puede tomar una variable lingüística. Al llamar "número" al subconjunto simplemente se mantiene el esquema de conceptos correspondiente a una variable no difusa (cuyos valores son números no difusos).

APPENDIX A
SYSTEM SPECIFICATIONS

Interface (Encoder input and IRD output)	NTSC composite (RS-170) baseband or RGB component
Vertical Resolution	Display = 480 lines
Horizontal Resolution	Display = 368 pixels
Compression Type	Discrete Cosine Transformation (DCT) with motion compensation using proprietary CLI technology
Video Data Compression Rate	1.73/1.95/4.36 Mbps
Channel Data Rate	2.93568/3.3/6.6 Mbps

Interface (outputs)	Stereo Monaural (RF Channel 3, 4)
Audio Data Rate	Dolby™ digital audio at 200 Kbps/channel (nominal)
Dynamic Range	>70 dB
Distortion	<.3%
Frequency Response	±3 dB, 20 Hz to 15 KHz
Channel Separation	40 dB @ 1kHz

Interface	RS-232
Serial Data	1200, 2400, 4800, 9600, 19200 baud
Substitute for Audio	400 Kbps (nominal)
Substitute for Video (when video source is removed)	0 to 4.36 Mbps

Transmission Interface

Error Correction	Any combination of 16 bit errors in a 1056-bit block
------------------	--

