

Universidad de Concepción
Facultad de Ingeniería
Departamento de Ingeniería Eléctrica

Profesor Patrocinante:
Sr. Juan Pablo Segovia V.

“Controlador MIMO Basado en Identificación Neuro-Difusa”

Carlos Roberto Bahamonde Reyes

Informe de Memoria de Título
Para optar al título de

Ingeniero Civil Electrónico

Concepción, Septiembre del 2006

A mi familia...



Agradecimientos

En primer lugar quiero dar gracias a Dios por haberme permitido llegar a esta instancia de mi carrera, por darme la salud y fuerza necesaria para salir adelante en los momentos difíciles.

También quiero agradecer a todas las personas que fueron participes de este proceso en diversas formas, ya sea en el apoyo moral, económico o humano, todos ellos aportaron con su grano de arena, Padre, tíos, amigos, familia en general.

Gracias al profesor Juan Pablo Segovia por creer en mí, por su paciencia y buena voluntad para ayudarme a resolver los obstáculos que se presentaron en este trabajo. Quiero destacar además la buena disposición para impartir conocimientos por parte de profesores como Daniel Sbarbaro, José Espinoza, Hans Rautenberg, entre otros. También quiero dar gracias a don Patricio Orellana por su inmensa buena voluntad y su gran disposición para ayudar.

A los grandes amigos que hice en mi vida universitaria y que siempre me ayudaron en los estudios desinteresadamente, brindándome su honesta amistad, ellos saben quienes son.

Finalmente quiero agradecer a las tres personas que fueron fundamentales en mi carrera, mi madre Gladys, mis abuelos y padres Hernán y Nilda, sin ellos nada de esto hubiera sido posible, gracias por el enorme amor que me han entregado.

Resumen

En esta memoria se propone el estudio de técnicas de modelación y control basadas en lógica difusa del tipo Takagi-Sugeno, para sistemas no lineales multivariables y en particular la identificación de un sistema “TITO” (Two input-Two output) no-lineal.

La modelación del sistema es llevada a cabo en tres etapas. En la primera de ellas, se utiliza una entrada aleatoria para la identificación del sistema y la modelación es realizada en su fase inicial con aprendizaje no supervisado *off-line*. En la segunda fase se excita la planta con una entrada sinusoidal y también con aprendizaje *off-line*. Finalmente se aplica la misma entrada sinusoidal pero se realiza la adaptación *on-line* de los parámetros y se evalúa el desempeño de estas tres etapas. De las simulaciones realizadas se puede apreciar que la identificación del modelo arroja muy buenos resultados y que la modelación en la etapa de validación mejora notablemente con el uso de un algoritmo de identificación con adaptación de los parámetros.

El control neuro-difuso es diseñado en base a controladores lineales locales, los que actúan dependiendo de la zona del espacio de entrada en que se esté operando. Estos controladores locales están basados en el modelo difuso inverso del proceso, los parámetros tanto del antecedente como del consecuente son adaptados con aprendizaje supervisado, o sea “online” con el proceso.

Por último se evalúa el desempeño del controlador neuro-difuso en el mismo sistema propuesto para modelación y además en una aplicación de una Planta de Separación Gas-Líquido. Se puede observar de las simulaciones que el controlador posee una gran robustez y adaptación a los cambios del sistema, además puede lidiar de gran forma con el acoplamiento existente entre las salidas y con el alto grado de no-linealidad que poseen estos sistemas MIMO. Esto es debido a que el modelo tiene características adaptables a través de sus parámetros los que permiten compensar los cambios en los parámetros del sistema y el efecto de las perturbaciones, lo que no es posible alcanzar con técnicas de control convencionales ya que están pensadas para sistemas lineales con comportamiento invariante en el tiempo.

Índice General

Agradecimientos	1
Resumen	2
Introducción	5
1. Sistemas Neuro-Difusos	7
1.1. Sistemas Neuro Difusos.....	7
1.1.1. <i>Ventajas de los Sistemas Neuro-Difusos</i>	8
1.1.2. <i>Limitaciones de los Sistemas Neuro-Difusos</i>	8
1.1.3. <i>Arquitectura del modelo ANFIS</i>	9
1.1.4. <i>Sistemas de aprendizaje</i>	11
2. Modelación Neuro-Difusa	20
2.1. Introducción a los Modelos Neuro-Difusos.....	20
2.2. Modelación	21
2.3. Identificación Difusa.....	23
2.3.1. <i>Identificación Off-Line del Modelo Difuso</i>	25
2.3.2. <i>Adaptación On-Line del Modelo Difuso</i>	27
2.4. Ejemplo de Simulación	28
3. Control Neuro Difuso	36
3.1. Introducción al Control Neuro Difuso	36
3.2. Descripción Matemática del Modelo Difuso	38
3.3. Identificación Modelo Inverso del Proceso	38
3.4. Controlador Difuso Basado en el Modelo Inverso del Proceso	40
3.5. Control Basado en Modelos Neuro Difusos Locales Lineales	40
3.6. Esquemas de Control Neuro Difuso Basados en Modelación	43
3.6.1. <i>Control Híbrido</i>	43
3.6.2. <i>Control con Modelo de Referencia</i>	44

3.6.3. Control Lineal o No lineal Adaptivo.....	45
3.6.4. Control Predictivo	45
3.6.5. Control de Modelo Interno (IMC)	46
3.7. Desarrollo Teórico del Método de Control a Implementar.....	47
3.7.1. Resultados Simulación.....	51
3.8. Control Planta Separación Gas-Líquido	57
3.8.1. Descripción del Sistema a Controlar.....	57
3.8.2. Modelación del Sistema	59
3.8.3. Resultados Simulación.....	61
4. Conclusiones	63
4.1. Conclusiones.....	63
4.2. Comentarios de los Resultados	65
4.3. Trabajos Futuros	65
Bibliografía.....	67
A. Lógica Difusa y Redes Neuronales	70
A.1. Introducción a los Sistemas Difusos	70
A.2. Elementos Básicos de Sistemas Difusos y Lógica Difusa	71
A.2.1. Variable Lingüística.....	71
A.2.2. Conjunto Difuso	71
A.2.3. Funciones de Pertenencia (FP).....	72
A.2.4. Regla o Implicación Difusa.....	75
A.2.5. Sistemas de Inferencia Difusa	76
A.2.6. Fuzzificación	78
A.2.7. Conocimiento Base.....	79
A.2.8. Defuzzificación	79
A.3. Introducción a las Redes Neuronales.....	80
A.3.1. Descripción de las Redes Adaptivas	81
A.3.2. Reglas de Aprendizaje	82
A.3.3. Comparación de las Teorías	83
A.3.4. Integración de las Teorías.....	84

Introducción

El desarrollo de modelos matemáticos [3], [5], [10], [14] de sistemas reales es un tópico de gran importancia en muchas disciplinas de ciencias e ingeniería. Estos modelos pueden ser usados para predicción, simulaciones, análisis del comportamiento del sistema, mejor entendimiento de los mecanismos fundamentales de un sistema, diseño de nuevos procesos o diseño de controladores basados en modelos.

Para sistemas no lineales dinámicos las técnicas convencionales de modelación, identificación y control son difíciles de implementar y algunas veces impracticables, esto es debido a que están pensadas para sistemas lineales con comportamiento invariante en el tiempo, los cuales se asumen completamente conocidos y cuya comprensión es relativamente sencilla. No obstante, en la mayoría de las instancias prácticas, los sistemas a ser controlados son no lineales y las leyes físicas que rigen su comportamiento no son de fácil comprensión, además no se posee un conocimiento previo de éstos. Estos tipos de incertidumbres en los modelos son extremadamente difíciles de manejar con métodos convencionales.

Sin embargo, otras técnicas basadas en lógica difusa [1], [2] han sido usadas recientemente con mayor frecuencia para la modelación de este tipo de procesos complejos. Entre los diferentes tipos de métodos difusos para modelación y control se destaca el método de Takagi-Sugeno [1], [2], [4], [12] el cual consiste en reglas del tipo *if-then* con antecedentes difusos y funciones matemáticas en la parte consecuente.

El objetivo del sistema de identificación es determinar tanto los parámetros no lineales de los antecedentes, como los parámetros lineales del consecuente de las reglas difusas. Estas reglas intentan emular las acciones de control de un operador humano frente a una determinada situación del proceso.

La modelación difusa describe los sistemas estableciendo relaciones entre las variables relevantes del sistema por medio de reglas del tipo *if-then* tales como: “Si la válvula está abierta

entonces la presión decrecerá rápidamente”. A diferencia de los modelos de caja negra como redes neuronales los modelos difusos son transparentes en su estructura para la interpretación y análisis, cuyas soluciones pueden ser fácilmente explicadas a los usuarios.

El uso de un controlador difuso basado en el modelo inverso del proceso es ventajoso puesto que es mucho más fácil obtener información sobre como el proceso responde a una entrada en particular, ya que es capaz de recordar como y por qué el operador responde a estas situaciones particulares. Es decir, recuerda los resultados obtenidos en estados anteriores, o en la observación de situaciones pasadas, para que cuando se vuelva a encontrar en la misma situación, responda mejor y más rápidamente. Esta capacidad de aprendizaje es uno de los principales atributos de un controlador inteligente, lo que significa que éste puede mejorar su ejecución futura debido a que tiene información que ha ganado experimentalmente en el pasado.

En este trabajo se propone el uso de sistemas Neuro-Difusos para observar su desempeño en la modelación y control de sistemas MIMO altamente no-lineales, debido a su gran capacidad de manejar este tipo de sistemas con gran exactitud, eficiencia y relativamente sencilla comprensión.

Sin embargo se debe tener en cuenta que los sistemas Neuro-Difusos tienen algunas limitaciones dentro de las cuales se pueden mencionar: Curso de dimensionalidad, ya que presentan el problema de crecimiento exponencial del tamaño del modelo con respecto al número de variables que poseen, haciendo impráctico su uso para sistemas de más de cuatro dimensiones. Calidad de los datos, debido a que pueden surgir problemas asociados con la cantidad, distribución, y ruido en los datos. Conocimiento previo, ya que en estos sistemas es deseable tener el conocimiento previo y confiable de un experto en el sistema, pues de otro modo pueden existir problemas al no haber descripción cualitativa del sistema.

Finalmente los algoritmos de modelación y control propuestos fueron evaluados mediante la simulación de los sistemas mostrados en los capítulos 2 y 3 mediante el uso del software *Matlab 6.5*.

Capítulo 1

1. Sistemas Neuro-Difusos

En este capítulo se destacan las ventajas de la integración de la lógica Difusa con Redes Neuronales, potenciando su complementación dando como resultado la teoría Neuro-Difusa, que rescata los aspectos positivos de una y mejora las falencias de otra. Se mencionan además las ventajas que pueden obtenerse producto de la fusión de ambas teorías, además de sus limitaciones y en particular, se describe la arquitectura ANFIS. Finalmente se mencionan algunos sistemas de aprendizaje con su respectiva clasificación y se abordan en detalle dos de ellos.

1.1. Sistemas Neuro Difusos

Básicamente un sistema Neuro-Difuso resulta de la integración de los sistemas de Lógica Difusa con la teoría de redes adaptivas (RNA). El objetivo de combinar estas teorías es poder compensar las debilidades de una, con los beneficio de la otra. Estos sistemas se entrenan por medio de un algoritmo de aprendizaje derivado de la teoría de las redes neuronales. El proceso de aprendizaje no se basa en conocimiento, sino en un conjunto de datos de entrenamiento. Los sistemas neuro-difusos son representados básicamente como una red de inferencia similar a una red neuronal. Sin embargo los nodos en estas redes difieren de los utilizados en las redes neuronales en cuanto al tipo de función que realizan y la interpretación que se les da. Es decir, un sistema Neuro-Difuso puede verse como una red neuronal multicapa de avance. La primera capa representa las variables de entrada y las funciones de pertenencia, las reglas difusas se representan mediante enlaces y capas ocultas, la última capa, representa la variable de salida.

Se puede decir entonces que un sistema Neuro-Difuso se considera como la técnica para ajustar a un sistema difuso a partir de datos de entrenamiento para mejorarlo mediante aprendizaje a partir de ejemplos.

1.1.1. Ventajas de los Sistemas Neuro-Difusos

-La estructura neuro-difusa permite el uso de métodos cualitativos y cuantitativos en la construcción de modelos, en la etapa de aprendizaje, en la de funcionamiento, y en la realimentación de conocimiento al usuario, permitiendo así utilizar todo el conocimiento disponible.

- Los métodos cuantitativos se pueden observar en el aprendizaje que los sistemas neuro-difusos pueden realizar, a partir de datos del ambiente, y en su forma matemática de funcionamiento, mientras que los métodos cualitativos se reflejan en el sentido lingüístico que tiene la base de conocimiento de los sistemas difusos y que genera una interacción bilateral sistema-hombre, permitiendo la introducción de conocimiento previo al sistema y la realimentación del estado final del modelo en forma de reglas expresadas lingüísticamente desde el sistema al hombre.

- Las otras ventajas, son las conocidas ya ampliamente y que vienen heredadas de los sistemas difusos y de las RNA, tales como interpretación lingüística, introducción de conocimiento previo, autoaprendizaje, generalización, interpolación, etc.

1.1.2. Limitaciones de los Sistemas Neuro-Difusos

-Curso de la dimensionalidad: Los sistemas difusos presentan el problema del crecimiento exponencial del tamaño del modelo respecto al número de variables que poseen, haciendo impráctico su uso para sistemas de más de cuatro dimensiones

-Calidad de los datos: Existen problemas asociados con la cantidad, distribución y ruido en los datos. Si los datos están mal distribuidos, el aprendizaje puede no ser muy bueno, presentando especialmente problemas de generalización de interpolación si faltan muchos datos en cierta región. Este inconveniente es herencia de los sistemas neuronales y puede ser superado interpolando los datos, haciendo un tratamiento previo o buscando una representación adecuada de los mismos.

-Conocimiento previo: En los sistemas neuro-difusos es deseable tener el conocimiento previo y confiable de un experto en el sistema, pues de otro modo, se debe realizar un modelamiento por "Fuerza Bruta" exclusivamente a través de los datos numéricos empíricos y existirán problemas al no haber descripción cualitativa del sistema, a menos que se apliquen técnicas de minería de datos (descubrimiento de conocimiento) a la base de conocimiento del sistema neuro-difuso.

1.1.3. Arquitectura del modelo ANFIS

Esta arquitectura consiste de un sistema híbrido Neuro-Difuso, el mismo que es funcionalmente equivalente a mecanismo de inferencia Takagi-Sugeno (T-S) [4]. Para un sistema de inferencia T-S de primer orden, un conjunto de reglas difusas *SI-ENTONCES* es el siguiente:

$$\text{Regla 1: Si } x \text{ es } A_1 \text{ e } y \text{ es } B_1, \text{ entonces } z_1 = p_1x + q_1y + r_1 \quad (1.1)$$

$$\text{Regla 2: Si } x \text{ es } A_2 \text{ e } y \text{ es } B_2, \text{ entonces } z_2 = p_2x + q_2y + r_2 \quad (1.2)$$

Donde A_i y B_i son conjuntos fuzzy. La figura 1.1 presenta este mecanismo de inferencia.

Los niveles de activación de las reglas se calculan como $w_i = A_i(x) \cdot B_i(x)$, donde el operador lógico *and* puede ser modelado por una T-norma continua (producto). Las salidas individuales de cada regla son obtenidas como una combinación lineal entre los parámetros del antecedente de cada regla: $z_i = p_i \cdot x + q_i \cdot y + r_i$. La salida de control del modelo z_0 se obtiene por la normalización de los grados de activación de las reglas por la salida individual de cada regla:

$$z_0 = \frac{w_1 \cdot z_1 + w_2 \cdot z_2}{w_1 + w_2} = \frac{\bar{w}_1 \cdot z_1 + \bar{w}_2 \cdot z_2}{\bar{w}_1 + \bar{w}_2} \quad (1.3)$$

Donde \bar{w}_1 y \bar{w}_2 son los valores normalizados de w_1 y w_2 con respecto a la suma ($w_1 + w_2$). La red neuronal híbrida que representa este tipo de inferencia es una red adaptable con 5 capas, donde cada capa representa una operación del mecanismo de inferencia difusa. Esta red se muestra en la figura 1.2.

En esta arquitectura, todos los nodos de una misma capa tienen la misma función (los nodos representados con cuadros son nodos adaptables, es decir, sus parámetros son ajustables). La estructura de la red ANFIS consiste de cinco capas:

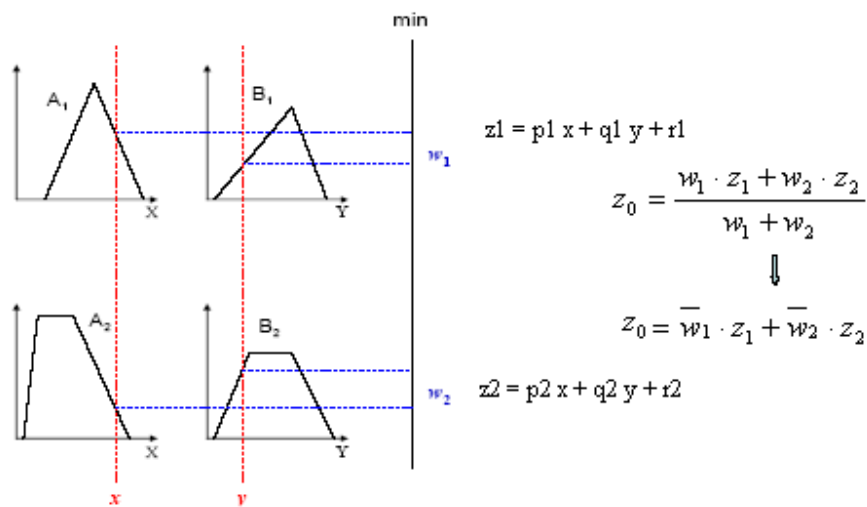


Figura 1.1: Sistema de inferencia Takagi-Sugeno de primer orden.

- Capa 1: Las entradas en esta capa corresponden a las entradas “x” e “y”, y la salida del nodo es el grado de pertenencia para el cual la variable de entrada satisface el término lingüístico asociado a este nodo.

$$O_i^1 = A_i(x) \quad (1.4)$$

- Capa 2: Cada nodo calcula el grado de activación de la regla asociada a dicho nodo. Ambos nodos están representados con una T en figura 1.2, por el hecho de que ellos pueden representar cualquier T-norma para modelar la operación lógica *and*. Los nodos de esta capa son conocidos como nodos de reglas.

$$O_i^2 = w_i = A_i(x) \cdot B_i(x) \quad (1.5)$$

- Capa 3: Cada nodo en esta capa está representado por una “N” en la figura 1.2, para indicar la normalización de los grados de activación. La salida del nodo es el grado de activación normalizado (con respecto a la suma de los grados de activación) de la regla i.

$$O_i^3 = \bar{w}_i = \frac{w_i}{w_1 + w_2} \quad (1.6)$$

- Capa 4: La salida de los nodos corresponde al producto entre el grado de activación normalizado por la salida individual de cada regla.

$$O_i^4 = \bar{w}_i \cdot z_i = \bar{w}_i \cdot (p_i \cdot x + q_i \cdot y + r_i) \quad (1.7)$$

- Capa 5: El único nodo de esta capa calcula la salida total del sistema (agregación) como la suma de todas las entradas individuales de este nodo.

$$O_i^5 = z_0 = \bar{w}_1 \cdot z_1 + \bar{w}_2 \cdot z_2 \quad (1.8)$$

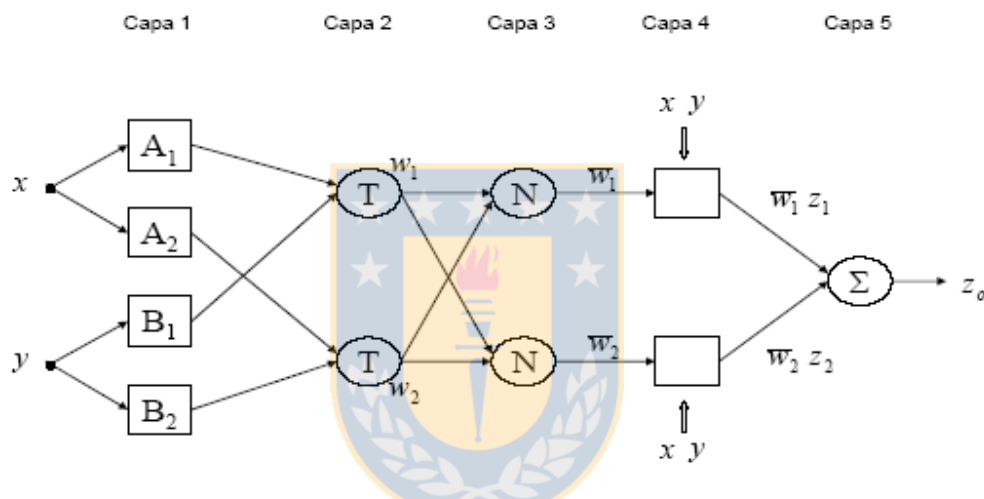


Figura 1.2: Arquitectura ANFIS equivalente para inferencia Takagi-Sugeno de primer orden.

1.1.4. Sistemas de aprendizaje

El aprendizaje en los sistemas neuro-difusos se basa generalmente en el ajuste de los parámetros de los antecedentes o del consecuente de las reglas difusas. Las funciones de pertenencia de los conjuntos difusos son sintetizadas automáticamente por un lazo de adaptación. El ajuste puede aplicarse sobre cualquier variable del sistema difuso como lo son por ejemplo:

-Conjuntos de entrada y salida: Se puede modificar su número, anchura, forma, o ajustar sus posiciones para que el sistema consiga el objetivo.

-Inferencia: Cambiar reglas o realizar una ponderación de importancia de las mismas.

Reducir la anchura en los conjuntos, se asemeja a aumentar la ganancia en un controlador proporcional. El aprendizaje en estos sistemas puede ser clasificado en dos grandes grupos [17]: ***Identificación de la estructura e identificación paramétrica.***

a) Identificación de la estructura

Este grupo de algoritmos identifica la estructura neuro-difusa, es decir, las reglas difusas y su correspondiente partición en el espacio de entrada y salida. Además existen métodos más sofisticados para este grupo de algoritmos, sin embargo, no serán discutidos en este trabajo. En este tipo de aprendizaje se destacan algoritmos más directos y simples como los de agrupamiento de datos que se han difundido de manera exitosa en los últimos años.

- Algoritmos de agrupación de datos

Entre las técnicas más comunes para el agrupamiento de datos se encuentran: método de particionamiento del árbol, fuzzy clustering, y enfoques de optimización de parámetros no lineales. En este caso se mencionarán las técnicas de fuzzy clustering, ya que son las técnicas de agrupamiento *off-line* más frecuentemente usadas en modelación difusa, etapa que se considera como una herramienta previa a la sintonización. El agrupamiento particiona el conjunto de datos de entrada en un número determinado de grupos que presentan características particulares. Esta característica es determinada por la medida de *similitud* [19], la cual es dependiente de los rangos de los vectores de entrada. Esta similitud puede ser entendida como una similitud matemática, definida en algún sentido. Generalmente esta similitud es dada por medio de una *norma distancia*. Se puede definir un cluster entonces, como un grupo o clase de objetos que son más similares a los de su misma clase que a los objetos de otra.

Dentro de esta clasificación, algunos de los algoritmos más destacados son: agrupamiento K-means y K-medoid, algoritmo Fuzzy C-means, algoritmo de Gath-Geva, y el algoritmo de Gustafson-Kessel (G-K). Frecuentemente, los métodos anteriormente mencionados para la obtención de los parámetros de las premisas difusas (fuzzy clustering), son usados solamente para la inicialización del método del descenso de gradiente.

En esta memoria se describirá en detalle solamente el algoritmo de Gustafson-Kessel, dado a que se consideró este último para la modelación del sistema simulado en el capítulo siguiente.

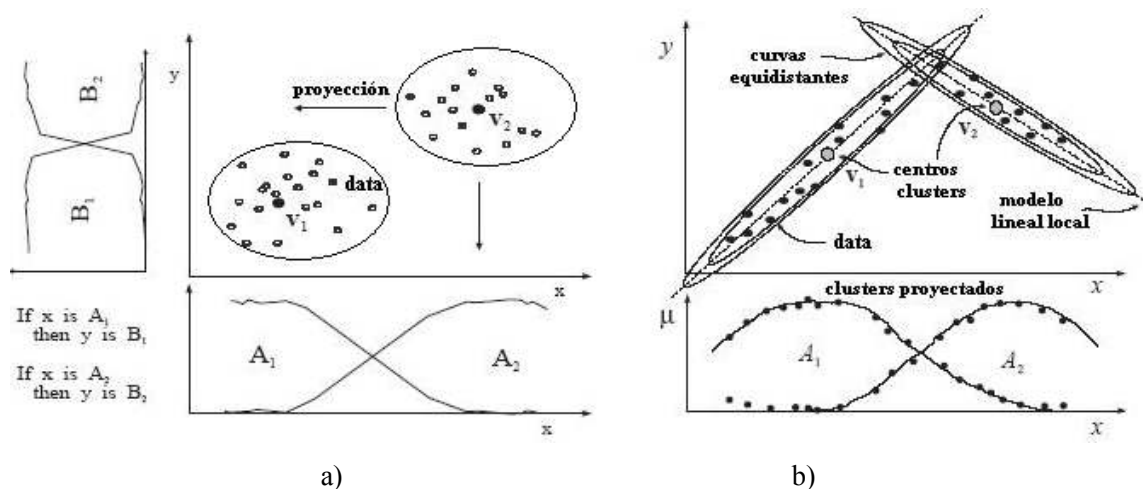


Figura 1.3: Ejemplo de agrupación por fuzzy clustering. a) Reglas basadas en la interpretación de fuzzy clustering. b) Cluster hiper-elipsoidal.

- **Método Gustafson-Kessel:**

Dado un grupo de observaciones, donde cada observación consiste en n variables medidas, agrupadas en un vector columna n -dimensional $\mathbf{x}_k = [x_{k1}, x_{k2}, \dots, x_{kn}]^T$, $\mathbf{x}_k \in \mathfrak{R}^n$. Un conjunto de N observaciones denotadas por $\mathbf{X} = \{\mathbf{x}_k | k = 1, 2, \dots, N\}$, que es representado como una matriz de $N \times n$:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{Nn} \end{bmatrix} \quad (1.9)$$

El objetivo del agrupamiento es particionar el conjunto de datos \mathbf{X} en c clusters como se muestra en la figura (clases o grupos). Este proceso de agrupamiento permite que los objetos pertenezcan a varios clusters simultáneamente, con diferente grado de pertenencia, cuyos valores van entre 0 y 1. Estos grados de pertenencia son agrupados en una matriz de partición que tiene la siguiente estructura:

$$U = \begin{bmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1c} \\ \mu_{21} & \mu_{22} & \cdots & \mu_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{N1} & \mu_{N2} & \cdots & \mu_{Nc} \end{bmatrix} \quad (1.10)$$

Cuyos elementos μ_{ik} representan el grado de pertenencia de la observación en el cluster i . En un comienzo c se asume conocido, basándose en conocimiento previo, pero puede obtenerse también por prueba y error. La idea central de este algoritmo consiste en asignar cada uno de los objetos de entrada, a uno de los c clusters de tal forma de minimizar un determinado índice, dependiendo del algoritmo utilizado. En el caso de este algoritmo se emplea como índice una *norma distancia adaptiva*, ya que cada cluster tiene su propia matriz norma inducida A_i , la cual conlleva a la siguiente expresión:

$$D_{ikA}^2 = (x_k - v_i)^T A_i (x_k - v_i), 1 \leq i \leq c, 1 \leq k \leq N \quad (1.11)$$

Donde v_i es llamado el prototipo del cluster, que habitualmente es representado por los centros de la partición.

Las matrices A_i son usadas como variables de optimización, permitiendo así que cada cluster adapte la norma distancia a la estructura topológica local de los datos. Si se define $A=(A_1, A_2, \dots, A_c)$, la función objetivo del algoritmo GK queda expresada como:

$$J(X; U, V, A) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m D_{ikA_i}^2 \quad (1.12)$$

Sin embargo la función objetivo no puede ser directamente minimizada con respecto a A_i , pues es lineal en A_i . Esto significa que J puede ser hecho tan pequeño como se desee, simplemente forzando a A_i ser definida positiva. Para obtener una solución factible, A_i debe ser forzada en algún otro sentido. La forma usual de que se cumpla esta condición, es forzar el determinante de A_i , permitiendo de esta forma que la matriz A_i varíe con su determinante fijo, lo que corresponde a la optimización de la figura o forma del cluster, mientras su volumen permanece constante. Lo anterior se puede visualizar a continuación:

$$\|A_i\| = \rho_i, \quad \rho > 0 \quad (1.13)$$

Donde ρ_i es fijado para cada cluster. Utilizando el método de los multiplicadores de Langrange se llega a la siguiente expresión:

$$A_i = [\rho_i \det(F_i)]^{1/n} F_i^{-1} \quad (1.14)$$

Donde F_i es la matriz de covarianza difusa del i -ésimo cluster definida por:

$$F_i = \frac{\sum_{k=1}^N (\mu_{ik})^m (\mathbf{x}_k - \mathbf{v}_i)(\mathbf{x}_k - \mathbf{v}_i)^T}{\sum_{k=1}^N (\mu_{ik})^m} \quad (1.15)$$

Algoritmo:

Inicialización: Elegir el número de clusters $1 < c < N$, el exponente de peso $m > 1$, la tolerancia de termino $\varepsilon > 0$ y la matriz de norma inducida A . Se inicializa la matriz de partición aleatoriamente tal como $U^{(0)} \in [0, 1]$.

Repetir para $l = 1, 2, \dots$

Paso 1 Cálculo de los centros de los clusters

$$\mathbf{v}_i^{(l)} = \frac{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m \mathbf{x}_k}{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m}, \quad 1 \leq i \leq c \quad (1.16)$$

Paso 2 Cálculo de las matrices de covarianza de los clusters

$$F_i = \frac{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m (\mathbf{x}_k - \mathbf{v}_i^{(l)})(\mathbf{x}_k - \mathbf{v}_i^{(l)})^T}{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m} \quad (1.17)$$

Paso 3 Cálculo de las distancias

$$D_{ikAi}^2(\mathbf{x}_k, \mathbf{v}_i) = (\mathbf{x}_k - \mathbf{v}_i^{(l)})^T \left[(\rho_i \det(F_i))^{1/n} F_i^{-1} \right] (\mathbf{x}_k - \mathbf{v}_i^{(l)}) \quad (1.18)$$

Paso 4 Actualización de la matriz de partición

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^c (D_{ikAi}(x_k, v_i) / D_{jkAi}(x_k, v_j))^{2/(m-1)}}, \quad 1 \leq i \leq c, 1 \leq k \leq N \quad (1.19)$$

hasta $\|U^{(l)} - U^{(l-1)}\| < \varepsilon$

b) Estimación paramétrica

Este grupo de algoritmos estiman y adaptan el sistema de parámetros, tales como los incluidos en las funciones de pertenencia, coeficientes polinomiales en el consecuente y otros posibles parámetros.

Entre los más destacados se encuentran: entrenamiento por retro-propagación, que consiste en adaptar los parámetros de las premisas y consecuente encontrando la dirección y magnitud paramétrica que minimice el error cuadrático medio de la salida o índice $J = \frac{1}{2}(y^d - y)^2$. Uno de los métodos más utilizados en esta clasificación es el método del descenso de gradiente. Otros algoritmos de estimación paramétrica son por ejemplo el método de mínimos cuadrados ortogonales, y los métodos de entrenamiento híbrido [19]. Estos últimos llevan este nombre debido a que el propósito de esta estrategia es combinar el desempeño de algoritmos de entrenamiento en dos fases. Esencialmente para los sistemas de inferencia difusa, lo anterior se constituye en separar la adaptación de los parámetros de las funciones de pertenencia dado su parametrización no lineal respecto del índice, con la empleada para los parámetros en el consecuente, que es lineal respecto al índice. Entre estos últimos, se destaca el método del descenso de gradiente (entrenamiento por retropropagación), combinado con un algoritmo de mínimos cuadrados recursivo. A continuación se describirá el algoritmo del descenso de gradiente puesto que será utilizado en el sistema de control simulado en el capítulo sub-siguiente.

- **Método del Descenso de Gradiente.**

Este método consiste en minimizar la función objetivo ε :

$$\varepsilon = \frac{1}{2} \sum_{k=1}^r (y_{zk} - y_k)^2 \quad (1.20)$$

Con

$$y_k = \frac{1}{\sum_{i=1}^p u_{ik}} \sum_{i=1}^p u_{ik} f_{ik} \quad (1.21)$$

$$\begin{aligned} u_{1k} &= \mu_{11}(x_{1k}) \cdot \mu_{12}(x_{2k}) \cdot \dots \cdot \mu_{1M}(x_{Mk}) \\ u_{2k} &= \mu_{21}(x_{1k}) \cdot \mu_{22}(x_{2k}) \cdot \dots \cdot \mu_{2M}(x_{Mk}) \\ &\vdots \\ u_{pk} &= \mu_{p1}(x_{1k}) \cdot \mu_{p2}(x_{2k}) \cdot \dots \cdot \mu_{pM}(x_{Mk}) \end{aligned} \quad (1.22)$$

Donde r es el número de discretizaciones, p el número de reglas, M el número de variables de entrada, y_k la salida del modelo o del controlador, e y_{zk} es la salida deseada. Las funciones de los consequentes de las reglas difusas tienen la forma:



$$f_i = \sum_{j=1}^p p_{ij} x_j + c_i \quad (1.23)$$

$$\begin{aligned} f_1 &= p_{11}x_1 + p_{12}x_2 + \dots + p_{1M}x_M + c_1 \\ f_2 &= p_{21}x_1 + p_{22}x_2 + \dots + p_{2M}x_M + c_2 \\ &\vdots \\ f_p &= p_{p1}x_1 + p_{p2}x_2 + \dots + p_{pM}x_M + c_p \end{aligned} \quad (1.24)$$

Las funciones μ_{pM} son funciones gaussianas como las mostradas en la ecuación (A.4), que consta de los parámetros c y σ , de tal forma que el algoritmo busca sintonizar estos parámetros del modelo difuso en orden a minimizar el error entre la variable deseada y el modelo. Para realizar la actualización de las premisas de las reglas se tiene la siguiente expresión:

$$\begin{aligned} \sigma_{ij}(l+1) &= \sigma_{ij}(l) - \eta_\sigma \frac{\partial \varepsilon}{\partial \sigma_{ij}} \\ c_{ij}(l+1) &= c_{ij}(l) - \eta_c \frac{\partial \varepsilon}{\partial c_{ij}} \end{aligned} \quad (1.25)$$

Donde η_c y η_σ son las tasas de aprendizaje de los parámetros centro y desviación standard respectivamente. Desarrollando las derivadas parciales se llega la siguiente expresión:

$$\sigma_{ij}(l+1) = \sigma_{ij}(l) - \eta_{\sigma} \sum_{k=1}^r 2(y_k - y_{zk}) \frac{\partial y_k}{\partial \sigma_{ij}} \quad (1.26)$$

Donde

$$\frac{\partial y_k}{\partial \sigma_{ij}} = \frac{1}{\sum_{i=1}^p u_{ik}} \left(\sum_{i=1}^p \frac{\partial u_{ik}}{\partial \sigma_{ij}} f_{ik} - y_k \sum_{i=1}^p \frac{\partial u_{ik}}{\partial \sigma_{ij}} \right) \quad (1.27)$$

Finalmente:

$$\sigma_{ij}(l+1) = \sigma_{ij}(l) - \eta_{\sigma} (y_k - y_{zk}) \cdot \left(\frac{f_{ik} - y_k}{\sum_{i=1}^p u_{ik}} \right) \cdot u_{ik} \left(\frac{(x_j - c_{ij})^2}{\sigma_{ij}(l)^3} \right) \quad (1.28)$$

Para los centros:

$$c_{ij}(l+1) = c_{ij}(l) - \eta_c \sum_{k=1}^r 2(y_k - y_{zk}) \frac{\partial y_k}{\partial c_{ij}} \quad (1.29)$$

Donde:

$$\frac{\partial y_k}{\partial c_{ij}} = \frac{1}{\sum_{i=1}^p u_{ik}} \left(\sum_{i=1}^p \frac{\partial u_{ik}}{\partial c_{ij}} f_{ik} - y_k \sum_{i=1}^p \frac{\partial u_{ik}}{\partial c_{ij}} \right) \quad (1.30)$$

Finalmente:

$$c_{ij}(l+1) = c_{ij}(l) - \eta_c (y_k - y_{zk}) \cdot \left(\frac{f_{ik} - y_k}{\sum_{i=1}^p u_{ik}} \right) \cdot u_{ik} \left(\frac{(x_j - c_{ij})}{\sigma_{ij}(l)^2} \right) \quad (1.31)$$

Los parámetros de las funciones f_i son determinados por el algoritmo de mínimos cuadrados recursivos. Luego el objetivo del aprendizaje es que la repuesta real sea igual a la deseada.

Usualmente se utilizan algunas combinaciones de los métodos para determinación de los parámetros del antecedente con métodos para determinación de los parámetros del consecuente, dentro de las cuales se mencionan las siguientes:

- Los parámetros de las premisas y los consecuentes de las reglas difusas son ajustados separadamente. Primero, los parámetros de las premisas son ajustados usando aprendizaje no-supervizado, con algún

método de clustering como por ejemplo el método c-means. Segundo, los parámetros del consecuente son ajustados por medio del método del gradiente descendente (o descenso de gradiente) o el método de mínimos cuadrados.

-Los parámetros son ajustados en dos fases de aprendizaje. Primero, tal como en el esquema anterior, los parámetros de las premisas son ajustados usando aprendizaje no-supervisado. Segundo, todos los parámetros (premisas y consecuentes) son ajustados por medio del método del descenso de gradiente.

-Otra configuración consiste en que primero los parámetros de las premisas son ajustados usando aprendizaje no-supervisado. Finalmente, en cada iteración los parámetros del consecuente son estimados en base al método de mínimos cuadrados y los parámetros del antecedente por medio del método del descenso de gradiente.



Capítulo 2

2. Modelación Neuro-Difusa

En este capítulo se evalúa el desempeño de los sistemas de inferencia difusa en la modelación de sistemas no lineales MIMO con salidas acopladas. En primer lugar se describe la modelación de este sistema aproximado por un sistema MISO. Posteriormente se lleva a cabo la identificación del sistema, lo que es efectuado con un método de agrupación de datos del tipo fuzzy clustering. Esta identificación es realizada con adaptación off-line y on-line para la adaptación de los parámetros de los consecuentes. Finalmente se discuten los resultados experimentales obtenidos con un sistema no lineal MIMO.

2.1. Introducción a los Modelos Neuro-Difusos

La modelación Neuro Difusa hay tenido un relevante crecimiento en las últimas décadas esto es debido a la gran cantidad de propiedades y ventajas que surgen de la integración de dos teorías como son Lógica Difusa y Redes Neuronales. Uno de los principales enfoques de este tipo de modelación se refiere a los modelos de Takagi-Sugeno (T-S) [1], [2]. Estos modelos son ampliamente usados debido a su gran capacidad de manejar sistemas altamente no-lineales, con gran exactitud y relativamente sencilla comprensión.

Con el uso de técnicas convencionales de modelación, la identificación de sistemas complejos se hace algunas veces muy engorrosa e impracticable, debido su alta no-linealidad y comportamiento variante en el tiempo. Estos tipos de incertidumbres en los modelos son extremadamente difíciles de manejar con estos métodos. El uso de estas estrategias puede plantear grandes desventajas, tales como:

-Dificultad de simulación.

-La existencia de alguna solución al problema del control no asegura que ésta pueda ser fácilmente obtenida.

-La implementación del controlador puede resultar de gran complejidad, y suponer un costo computacional excesivo.

-Los parámetros del controlador pueden ser de difícil reajuste, en el caso de variaciones paramétricas de la planta, ya que no se dispone de un conocimiento cualitativo y cuantitativo, sobre el efecto que la modificación de éstos puede provocar sobre el proceso.

La estrategia que se propone para resolver estos problemas es el modelado basado en modelos locales [5], que consiste en dividir un sistema complejo en *subsistemas* más sencillos que puedan ser analizados de forma individual. Si estos subsistemas se eligen además de tipo lineal, resultarán de gran sencillez de simulación y el problema del control podrá ser más fácilmente analizado para cada uno de ellos, según técnicas ampliamente contrastadas, obteniéndose finalmente una ley conjunta estable de control con relativa facilidad.

Otra de las grandes ventajas del uso del razonamiento difuso integrado con la aplicación de modelos locales por tramos [11], [15] es que éstos poseen una gran transparencia en su base de reglas y en su estructura misma lo que hace posible la incorporación de conocimiento previo, además del bajo costo computacional que demandan debido a su intrínseca naturaleza lineal. Tradicionalmente los sistemas difusos fueron construidos solamente basados en conocimiento experto en una forma lingüística. Recientemente ha habido un gran incremento en el uso de información producto de mediciones, ya sea para complementar el conocimiento experto o como la primaria fuente de información.

2.2. Modelación

La modelación e identificación son pasos sumamente importantes en el diseño de un sistema de control [4]. Las aplicaciones típicas de estos modelos son por ejemplo: la simulación, la predicción o el diseño de sistemas de control. Generalmente el proceso de modelación consiste en obtener un modelo paramétrico con el mismo comportamiento del proceso real. Sin embargo, cuando este proceso es complejo, es muy difícil definir las leyes físicas y matemáticas que rigen su comportamiento. En este trabajo se abordará el problema de identificación de un sistema MIMO [13] (Multi Input-Multi Output) con n_i entradas y con n_o salidas. Este sistema puede ser aproximado por un conjunto de modelos difusos discretos MISO (Multi Input-Single Output). Si se considera:

- Dos polinomios A y B definidos por:

$$A = a_0 + a_1 \cdot q + a_2 \cdot q^2 + \cdots + a_{n_A} \cdot q^{n_A} \quad (2.1)$$

$$B = b_0 + b_1 \cdot q + b_2 \cdot q^2 + \cdots + b_{n_B} \cdot q^{n_B} \quad (2.2)$$

Donde n_A y n_B es el orden de los polinomios A y B respectivamente, y “ q ” es el operador de retardo ($q^n y(k) = y(k - n)$).

- Dos enteros m y n , con $n \leq m$ los cuales definen una muestra retardada de una señal discreta en el tiempo como sigue:

$$\{y(k)\}_m^n = [y(k - m), y(k - m - 1), \dots, y(k - n)] \quad (2.3)$$

Los modelos MISO son un tipo de modelo entrada-salida NARX (Non linear Auto Regressive with eXogenous input) definido por:

$$y_l(k + 1) = f_l(x_l(k)), \quad l = 1, 2, \dots, n_o. \quad (2.4)$$

Donde el vector de regresión viene dado por:

$$x_l(k) = \begin{bmatrix} \{y_1(k)\}_0^{n_{yl1}}, \{y_2(k)\}_0^{n_{yl2}}, \dots, \{y_{n_o}(k)\}_0^{n_{yl n_o}}, \\ \{u_1(k)\}_{n_{dl1}}^{n_{ul1}}, \{u_2(k)\}_{n_{dl2}}^{n_{ul2}}, \dots, \{u_{n_i}(k)\}_{n_{dlni}}^{n_{ulni}} \end{bmatrix} \quad (2.5)$$

Además n_y y n_u define el número de salidas y entradas retardadas respectivamente, n_d es el número de retardo puro, n_y es una matriz de $n_o \times n_o$ y n_u, n_d son matrices de $n_o \times n_i$, f_l son funciones no-lineales desconocidas. Los modelos MISO son estimados independientemente, luego para simplificar la notación se omite el índice “ l ” y se abordará sólo el caso MISO. En este enfoque, la salida del proceso puede ser escrita como:

$$y(k + 1) = Ay(k) + Bu(k) + \alpha \quad (2.6)$$

Siendo α el coeficiente de off-set. Los modelos de TS representan una eficiente herramienta para la modelación de un sistema no-lineal MIMO.

2.3. Identificación Difusa

El modelo de T-S ha atraído la atención de muchos investigadores. En efecto, este modelo consiste en reglas *if-then* con antecedentes difusos y funciones matemáticas en la parte consecuente [12]. Los conjuntos difusos del antecedente dividen el espacio de entrada en un número de regiones difusas en donde los distintos modelos locales tienen mayor o menor validez dependiendo de su grado de pertenencia a cada zona difusa., mientras que las funciones del consecuente describen el comportamiento del sistema en esas regiones [5]. Después el modelo final es calculado como la suma ponderada de cada uno de los modelos locales lineales por su respectiva función de validación, la que está relacionada con el grado de pertenencia de cada modelo a las distintas zonas difusas. Las reglas difusas son definidas como:

$$R_i: \quad \text{If } x(k) \text{ is } \Omega_i \text{ then}$$

$$y^i(k+1) = A_i y(k) + B_i u(k) + \alpha_i, \quad i = 1, 2, \dots, K. \quad (2.7)$$

Aquí Ω_i es el conjunto del antecedente difuso de la i -ésima regla. $A = [A_{i1}, \dots, A_{ino}]$, $B = [B_{i1}, \dots, B_{ini}]$ son vectores de polinomios y K es el número de reglas. El antecedente de (2.7) puede ser escrito como:

$$R_i: \quad \text{If } x_1(k) \text{ is } \Omega_{i1} \text{ and ... and } x_p(k) \text{ is } \Omega_{ip}, \text{ then}$$

$$y^i(k+1) = A_i y(k) + B_i u(k) + \alpha_i, \quad i = 1, 2, \dots, K \quad (2.8)$$

Donde

$$p = \sum_{j=1}^{no} n_{yj} + \sum_{j=1}^{ni} n_{uj} + 1 \quad (2.9)$$

Luego la salida global del modelo se calcula como la interpolación de los submodelos activos localmente tal como sigue:

$$y(k+1) = \frac{\sum_{i=1}^K \mu_i(x(k)) y^i(k+1)}{\sum_{i=1}^K \mu_i(x(k))} \quad (2.10)$$

o:

$$y(k+1) = \sum_{i=1}^K y^i(k+1) \Phi_i(x, c_i, \sigma_i) \quad (2.11)$$

Donde $\Phi_i(x, c_i, \sigma_i)$ es la función de validación para las funciones miembros gaussianas con centros c_i y desviación standard σ_i definida como:

$$\Phi_j(x, c_i, \sigma_i) = \frac{\mu_j(x(k))}{\sum_{i=1}^K \mu_i(x(k))} \quad (2.12)$$

Además $y^i(k+1)$ es el i -ésimo modelo local perteneciente al conjunto difuso correspondiente. Estas funciones de pertenencia determinan la transición entre submodelos colindantes. Aquí cada sub-modelo es válido en una región especificada por su correspondiente función de validación Φ_i .

$$\mu_j(x) = \exp\left(-\frac{1}{2} \cdot \left(\frac{x_1 - c_{j1}}{\sigma_{j1}}\right)^2\right) \cdot \exp\left(-\frac{1}{2} \cdot \left(\frac{x_2 - c_{j2}}{\sigma_{j2}}\right)^2\right) \cdot \dots \cdot \exp\left(-\frac{1}{2} \cdot \left(\frac{x_p - c_{jp}}{\sigma_{jp}}\right)^2\right) \quad (2.13)$$

$\mu_j(x(k))$ es el grado de cumplimiento de la regla j . Puesto que la mayoría de los actuales algoritmos de identificación-manejo de datos, asume que la estructura del modelo es conocida, la estructura y el orden del modelo han sido elegidos antes de la identificación. Es por esto que las matrices n_y, n_u , y n_d son determinadas por el usuario en base a conocimiento previo y/o por comparación de diferentes configuraciones basadas en un criterio de prueba y error. Una vez que la estructura es fijada, los parámetros del antecedente son estimados por fuzzy clusterig [19].

Desafortunadamente este método no puede ser aplicado directamente para la identificación de procesos multivariable, sólo un conjunto de sistemas MISO puede ser identificado por fuzzy clustering, es por eso que en esta memoria se realiza la identificación por separado para cada salida.

El procedimiento del método propuesto esta basado en dos etapas. En la primera etapa, llamada identificación *off-line* del modelo fuzzy (difuso), ambos parámetros no-lineales de las funciones miembros (c_i y σ_i), son determinados por el método fuzzy clustering, y los parámetros de los modelos locales son determinados por un algoritmo de mínimos cuadrados ponderados [1]. En la segunda fase, llamada adaptación *on-line* del modelo fuzzy, los parámetros del consecuente de las reglas difusas son adaptados por una versión recursiva del método de mínimos cuadrados ponderados [1].

2.3.1. Identificación Off-Line del Modelo Difuso

Este procedimiento es llevado a cabo en cuatro pasos:

- Construcción de la data de regresión.
- Determinación de los clusters correspondiente a los conjuntos de los submodelos.
- Determinación de las funciones miembros del antecedente a partir de los parámetros del cluster.
- Estimación de los parámetros del consecuente.

-Datos de regresión: Los datos de las muestras disponibles son recolectados en la matriz Z compuesta por la concatenación de la matriz de regresión X y el vector de salida Y :

$$X = \begin{bmatrix} x(k) \\ \vdots \\ x(N-1) \end{bmatrix}, \quad Y = \begin{bmatrix} y(k+1) \\ \vdots \\ y(N) \end{bmatrix}, \quad Z^T = [X \ Y]. \quad (2.14)$$

Donde N es el número de datos muestreados.

- Construcción de los clusters fuzzy: Existen varios algoritmos para la construcción de los clusters fuzzy tales como: el algoritmo C- means [19], el algoritmo Gath-Geva y el algoritmo Gustafson-Kessel (G-K), los cuales fueron mencionados en el capítulo anterior. En este trabajo se utilizó el algoritmo G-K. La idea central de este método es particionar el conjunto de datos Z en N_c clusters, este valor fue escogido en base a un criterio de prueba y error. Este algoritmo entrega como resultado una matriz de partición difusa $U = [\mu_{ik}]_{N_c \times N}$, cuyos elementos $\mu_{ik} \in [0,1]$ representan el grado de la función de pertenencia de la observación (medición) en el cluster i , una matriz prototipo $V = [v_1, \dots, v_{N_c}]$ cuyos elementos corresponden a los centros de la partición, y un conjunto de matrices de covarianza

$F = [F_1, \dots, F_{N_c}]$. Una vez que estas matrices son determinadas, se calculan los parámetros del consecuente A_i, B_i, α_i .


-Determinación de las Funciones Miembros a partir de los Parámetros del Cluster: En este trabajo se usaron funciones de membresía Gaussianas para representar los conjuntos difusos Ω_{ij} :

$$\Omega_{ij}(x_j(k)) = \exp\left(-\frac{1}{2} \cdot \left(\frac{x_j - c_{ij}}{\sigma_{ij}}\right)^2\right) \quad (2.15)$$

Esta elección lleva a la siguiente formula compacta:

$$\Omega_{ij}(x_j(k)) = \exp\left(-\frac{1}{2} (x(k) - c_i^x)^T (F_j^{xx})^{-1} (x(k) - c_i^x)\right) \quad (2.16)$$

Con $c_i^x = [c_{1i}, \dots, c_{pi}]$ el vector de centros y $(F_j^{xx})^{-1}$ es la inversa de la matriz conteniendo las varianzas en su diagonal:



$$F_j^{xx} = \begin{bmatrix} \sigma_{1j}^2 & 0 & \dots & 0 \\ 0 & \sigma_{2j}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_{pj}^2 \end{bmatrix} \quad (2.17)$$

- Estimación de los Parámetros del Consecuente: Estos parámetros en cada regla son estimados separadamente por método de mínimos cuadrados ponderados por la minimización del siguiente criterio:

$$\min_{\theta} \left(\frac{1}{N} (Y - X_e \theta_i)^T Q_i (Y - X_e \theta_i) \right) \quad (2.18)$$

Donde $X_e = [X \ 1]$ es la matriz de regresión extendida por una columna unitaria, Q_i es una matriz que contiene los valores de la función de validación Φ_i del i -ésimo modelo local para cada muestra:

$$Q_i = \begin{bmatrix} \Phi_i(x(1), c_i, \sigma_i) & 0 & \dots & 0 \\ 0 & \Phi_i(x(2), c_i, \sigma_i) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \Phi_i(x(N), c_i, \sigma_i) \end{bmatrix} \quad (2.19)$$

Finalmente la estimación de los parámetros es llevada a cabo por la siguiente expresión:

$$\theta_i = [X_e^T Q_i X_e]^{-1} X_e^T Q_i Y \quad (2.20)$$

2.3.2. Adaptación On-Line del Modelo Difuso

Es necesario una adaptación on-line, para obtener un buen modelo capaz de describir el proceso en una gran cantidad de puntos de operación, para ser usado en un esquema de control adaptivo [4]. Generalmente, los modelos difusos de TS obtenidos por clustering, tienen parámetros consecuentes constantes, como lo indica la expresión (2.20), pero en este caso los parámetros son actualizados, lo que significa que en cada instante “ k ” se puede obtener un modelo TS:

$$y^i(k+1) = A_i(k)y(k) + B_i(k)u(k) + \alpha_i(k) \quad (2.21)$$

En esta fase, las premisas de las reglas son mantenidas fijas, y solamente los consecuentes de las reglas son adaptadas para cada modelo local, por una versión recursiva del algoritmo de mínimos cuadrados ponderados con factor de olvido λ :

$$\theta_j(k) = \theta_j(k-1) + \delta_j(k)(y(k) - x^T(k)\theta_j(k-1)) \quad (2.22)$$

$$\delta_j(k) = \frac{P_j(k-1)x(k)}{x^T(k)P_j(k-1)x(k) + \lambda / \Phi_j(x(k), c_j, \sigma_j)} \quad (2.23)$$

$$P_j(k) = \frac{1}{\lambda} [I - \delta_j(k)x^T(k)]P_j(k-1) \quad (2.24)$$

En la expresión (2.22), el vector de parámetros θ_j es el mismo que para la identificación off-line en (2.20), pero en este caso es actualizado por la suma del vector de corrección de las estimaciones antiguas $\theta_j(k-1)$. En (2.23) y (2.24), λ es el factor de olvido de las mediciones viejas, Φ_j es el peso del dato actual, δ_j la ganancia de corrección y P_j es la matriz de ganancia de adaptación.

2.4. Ejemplo de Simulación

En esta memoria se considerará un sistema MIMO descrito por las siguientes ecuaciones:

$$y_1(k+1) = \frac{y_1(k)}{1 + y_2^2(k)} + u_1(k) \quad (2.25)$$

$$y_2(k+1) = \frac{y_1(k)y_2(k)}{1 + y_2^2(k)} + u_2(k) \quad (2.26)$$

Las entradas son u_1 y u_2 , las salidas son y_1 e y_2 . El proceso de identificación es llevado a cabo con entradas aleatorias para u_1 y u_2 uniformemente distribuidas en el intervalo $[-1 \ 1]$, y se usaron tres clusters (o número de reglas) para cada salida. Las señales de entrada se muestran en la figura 2.1. A partir de los datos de entrada se pueden obtener los datos de las salidas de acuerdo a las ecuaciones (2.25) y (2.26). Una vez que los datos entrada-salida están disponibles, se calculan las matrices U , V y F de acuerdo a las ecuaciones (2.16) y (2.17), mediante el algoritmo de Gustafson-Kessel. Luego se determinan los parámetros del consecuente de cada regla por medio del algoritmo de mínimos cuadrados ponderados de acuerdo a (2.20). Las respuestas de la planta y la identificación del modelo son mostradas en la figura (2.2). Para validar el modelo establecido se aplicó el siguiente vector de entrada:

$$[\sin(2\pi k / 25), \cos(2\pi k / 25)]^T \quad (2.27)$$

Las respuestas de la planta y de la identificación del modelo para estas entradas se muestran en la figura (2.3). Estas respuestas son obtenidas de los modelos locales calculados por fuzzy clustering sin adaptación de los parámetros del consecuente. Se puede observar de la figura (2.3), que las salidas estimadas no pueden seguir a las salidas del proceso y el error resultante es bastante grande. Para mejorar la calidad del modelo fuzzy establecido, los parámetros de los consecuentes de las reglas son adaptados por un algoritmo de mínimos cuadrados recursivos con un factor de olvido $\lambda=0.99$ de acuerdo a (2.22). La figura (2.4) muestra las respuestas de la planta y la identificación del modelo para el vector de entradas sinusoidales pero con adaptación. Para establecer un índice de exactitud del modelo se define una función VAF la cual calcula el la *Varianza Estimada entre dos señales (Variance Accounted for between two signals (VAF))*, como sigue a continuación:

$$VAF = 100\% \left[1 - \frac{\text{var}(y_1 - y_2)}{\text{var}(y_1)} \right]^T \quad (2.28)$$

Donde y_1 es la salida del proceso e y_2 es la salida del modelo. El VAF de dos señales iguales es 100%. Si las señales difieren, el VAF es más bajo. La tabla (1.2) muestra los índices de desempeño del modelo en las fases de identificación, validación sin adaptación y validación con adaptación. De esta tabla se puede observar que el modelo fuzzy TS adaptado es más preciso que el modelo no adaptado.

	Identificación	Validación sin adaptación	Validación con adaptación
VAF-y1(%)	96.43	53.79	85.81
VAF-y2(%)	96.38	46.61	73.62

Tabla 1.1: Comparación de la exactitud de predicción del modelo difuso TS en las tres fases.

Para verificar la aplicabilidad de este método a procesos reales, se debe mostrar la evolución de los parámetros de los polinomios A_i , B_i y α_i durante la adaptación. En efecto los parámetros del polinomio B_i corresponden a la ganancia de los controles u_1 y u_2 , los cuales no deben cambiar sus valores en grandes proporciones.

Las reglas son conclusiones lineales de las entradas del sistema, por ejemplo para la regla i :

$$\begin{aligned} y_{1i}(k+1) &= a_{i1p1}y_1(k) + a_{i2p1}y_2(k) + b_{i1p1}u_1(k) + \alpha_{ip1} \\ y_{2i}(k+1) &= a_{i1p2}y_1(k) + a_{i2p2}y_2(k) + b_{i1p2}u_2(k) + \alpha_{ip2} \end{aligned} \quad (2.29)$$

De la figura (2.6) a la figura (2.8) se muestra la evolución de los parámetros a_{i1p1} , a_{i2p1} , b_{i1p1} , α_{ip1} , de la salida y_1 para las tres reglas (clusters). De la figura (2.9) a la figura (2.11) se muestra la evolución de los parámetros a_{i1p2} , a_{i2p2} , b_{i1p2} , α_{ip2} , de la salida y_2 también para las tres reglas.

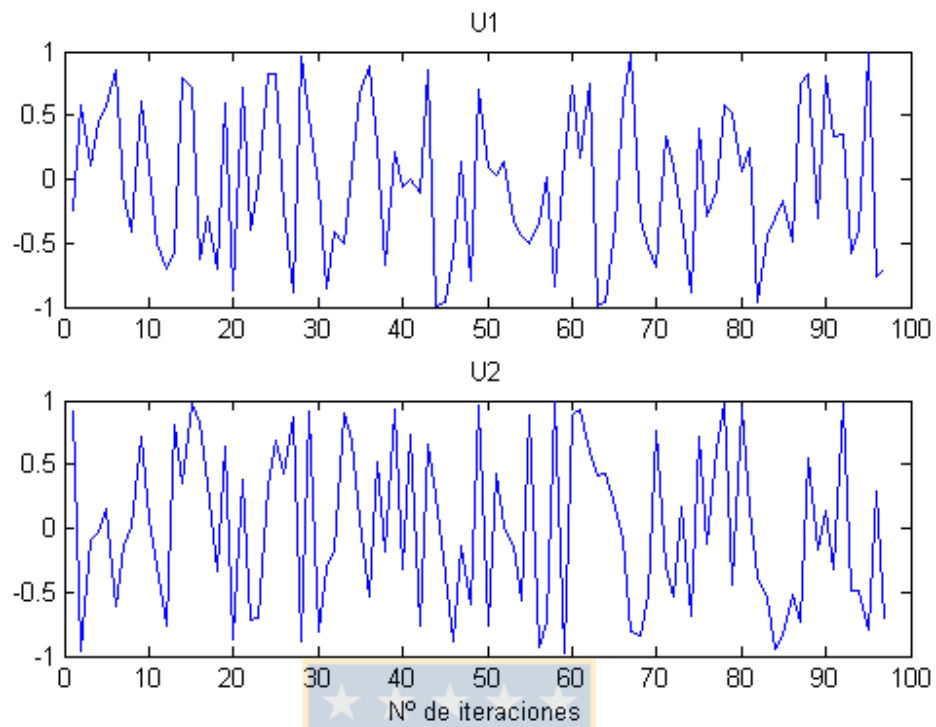


Figura 2.1: Entradas aleatorias u_1 y u_2 para la identificación.

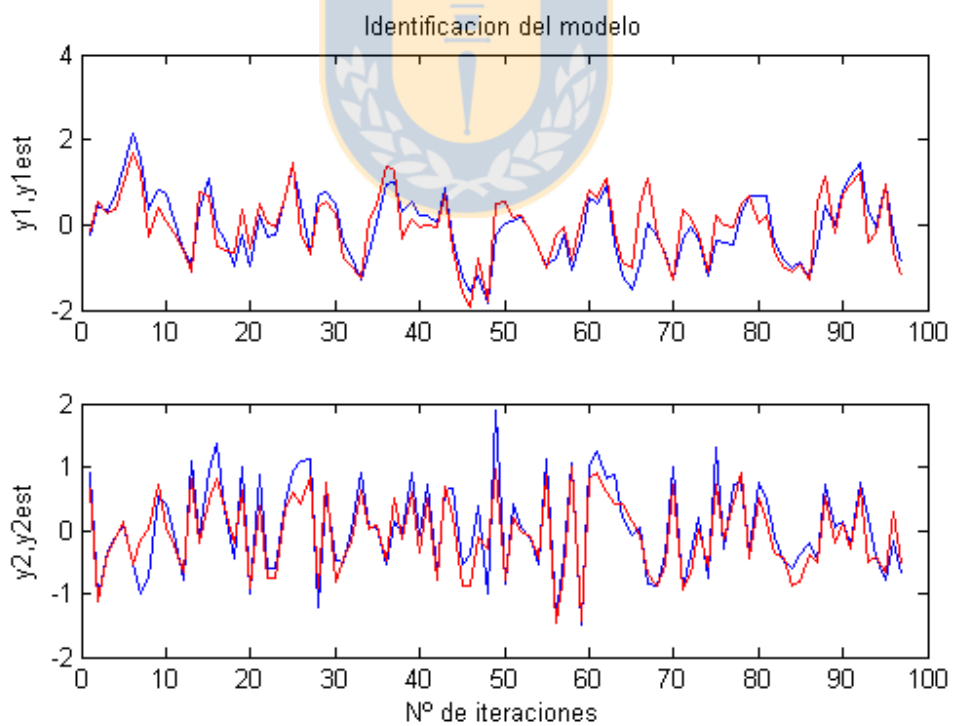


Figura 2.2: Comparación de las salidas del proceso (línea azul), con las salidas del modelo (línea roja) para la identificación.

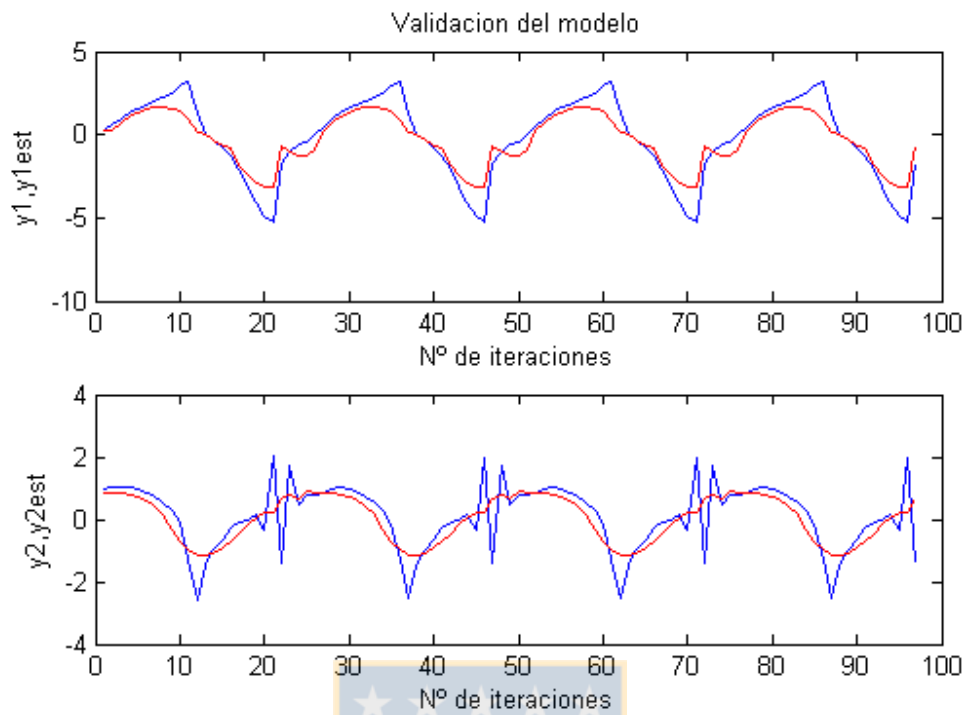


Figura 2.3: Comparación de las salidas del proceso (línea azul), con las salidas del modelo (línea roja) para la validación

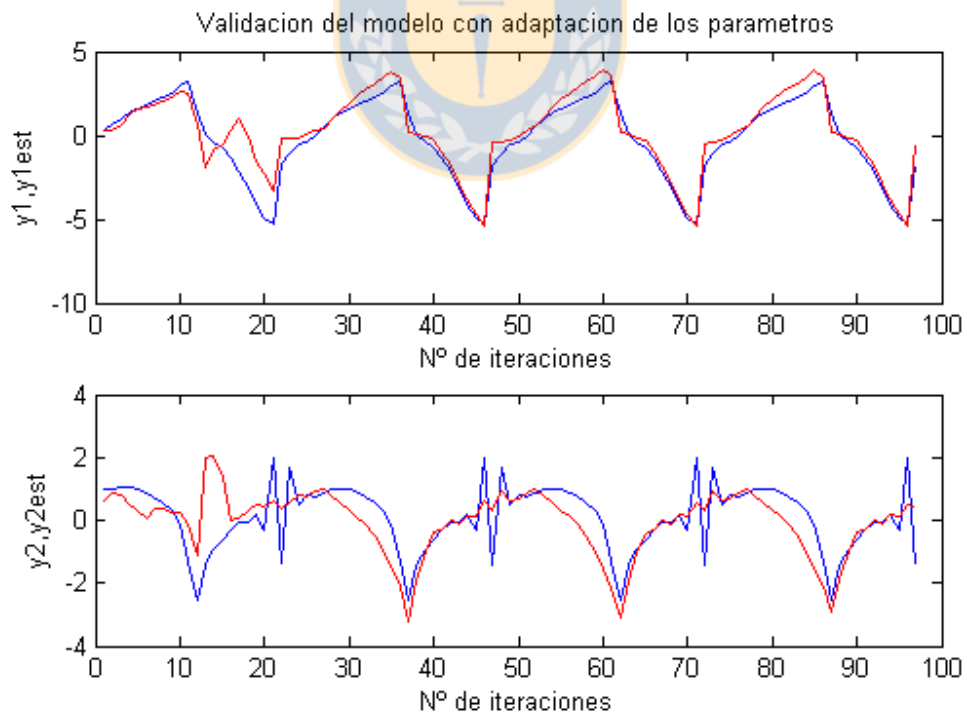


Figura 2.4: Comparación de las salidas del proceso (línea azul), con las salidas del modelo (línea roja) para la validación con adaptación de los parámetros.

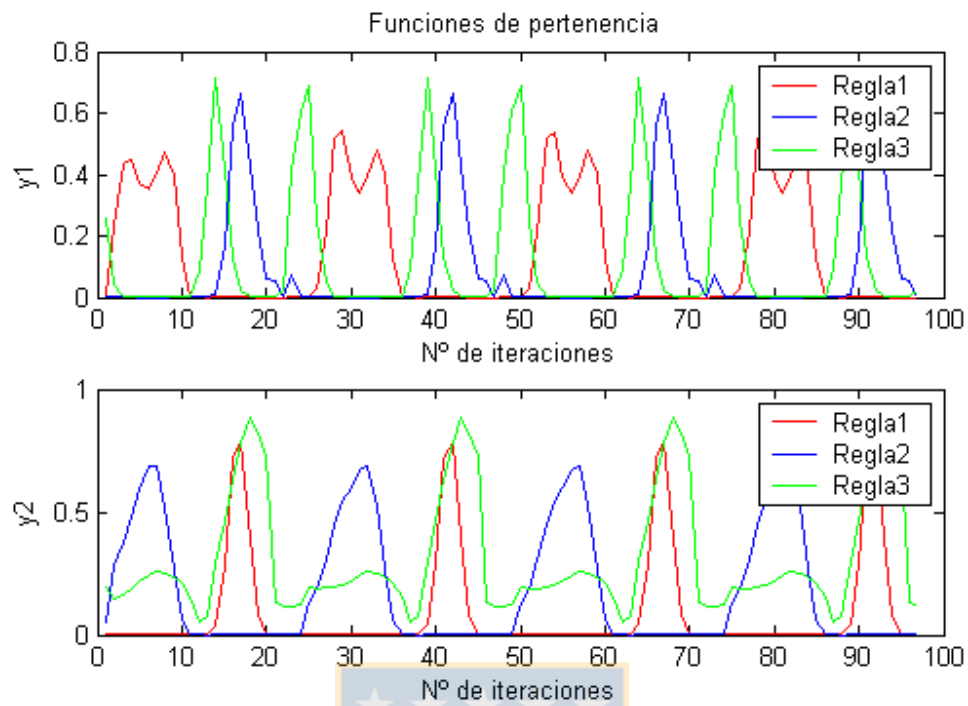


Figura 2.5: Funciones de pertenencia de las dos salidas para las tres reglas.

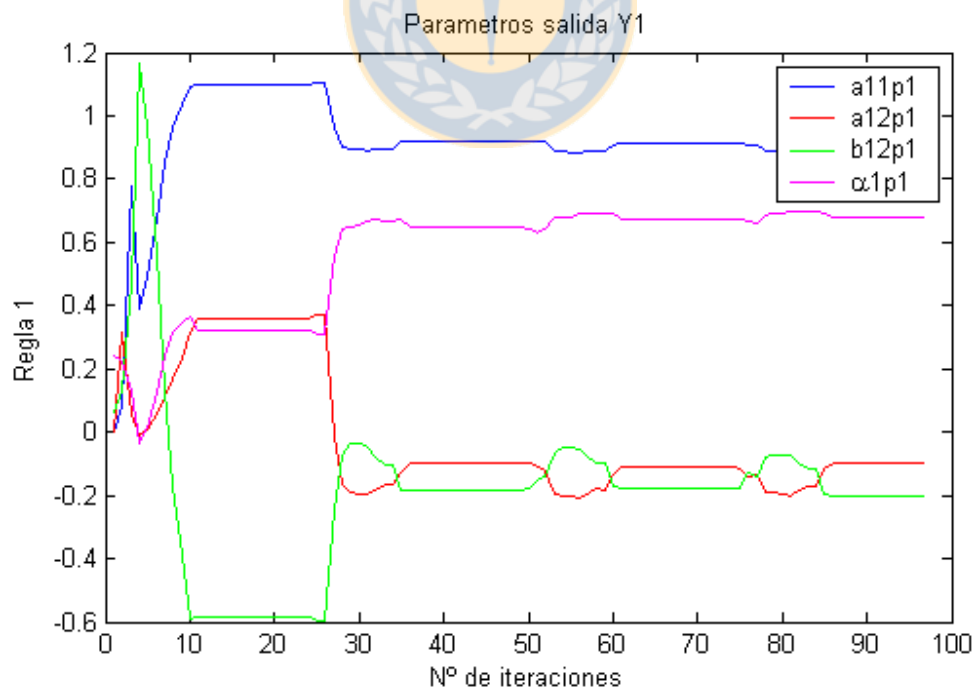


Figura 2.6: Evolución de los parámetros de la salida y_1 regla 1

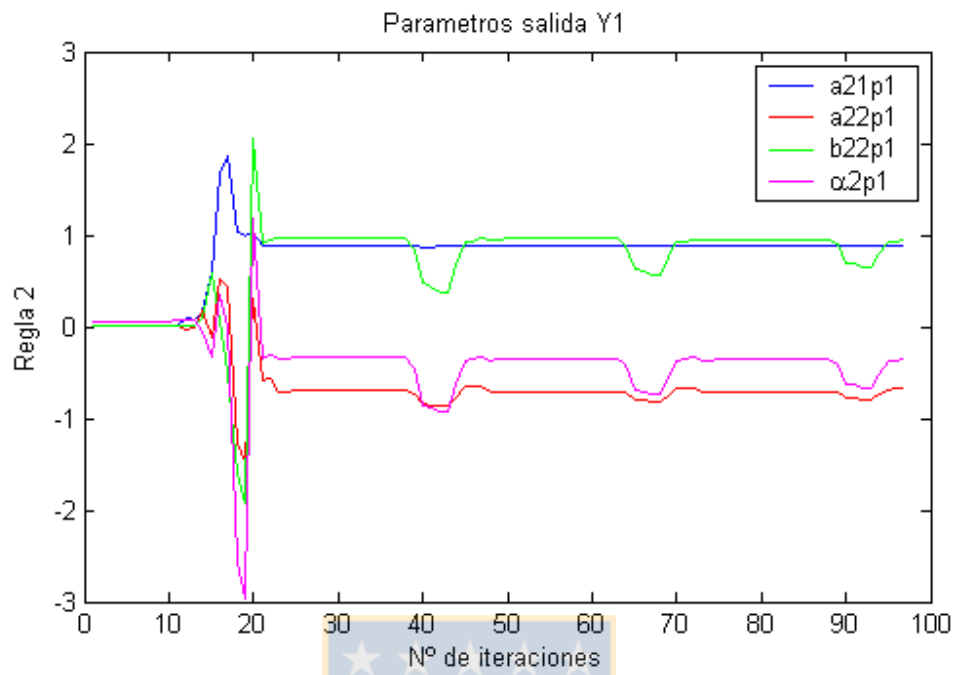


Figura 2.7: Evaluación de los parámetros de la salida y_1 regla 2

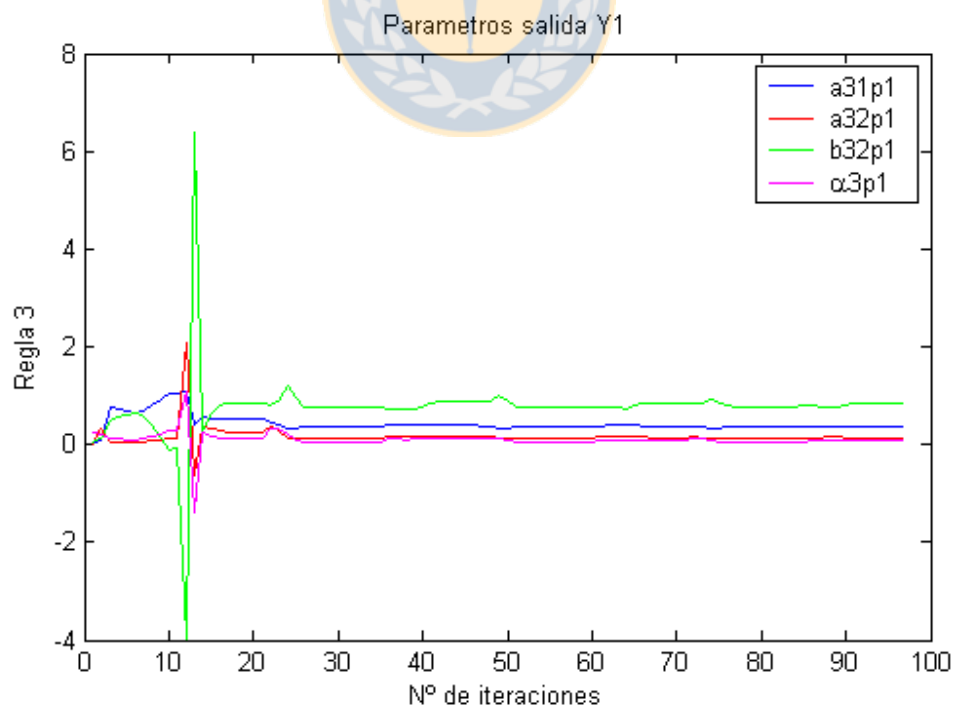


Figura 2.8: Evaluación de los parámetros de la salida y_1 regla 3

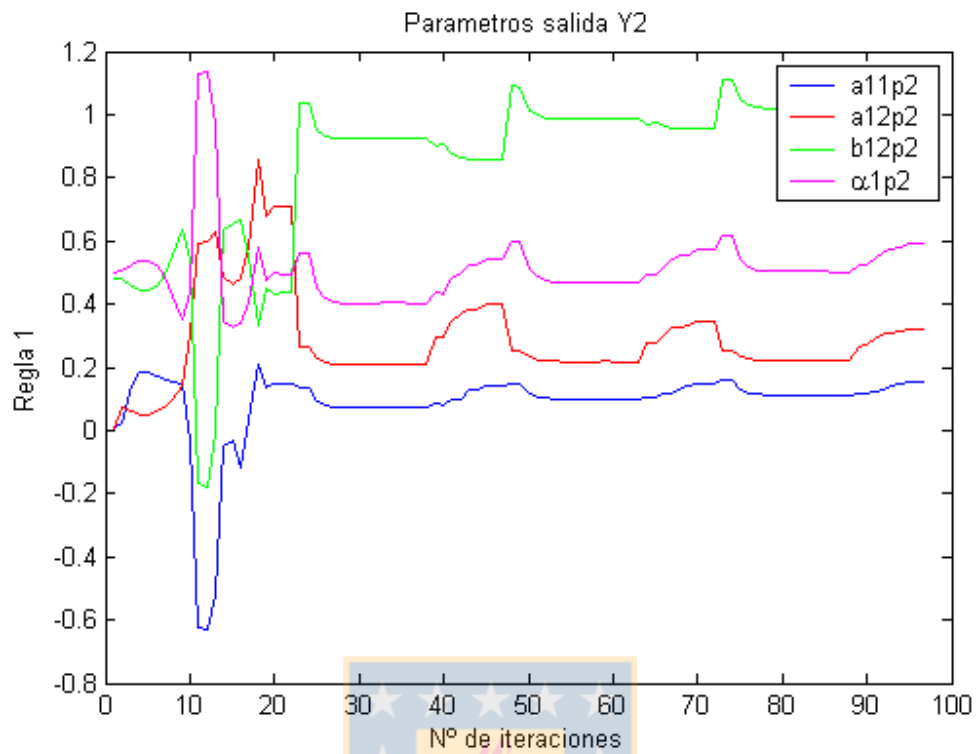


Figura 2.9: Evaluación de los parámetros de la salida y_2 regla 1

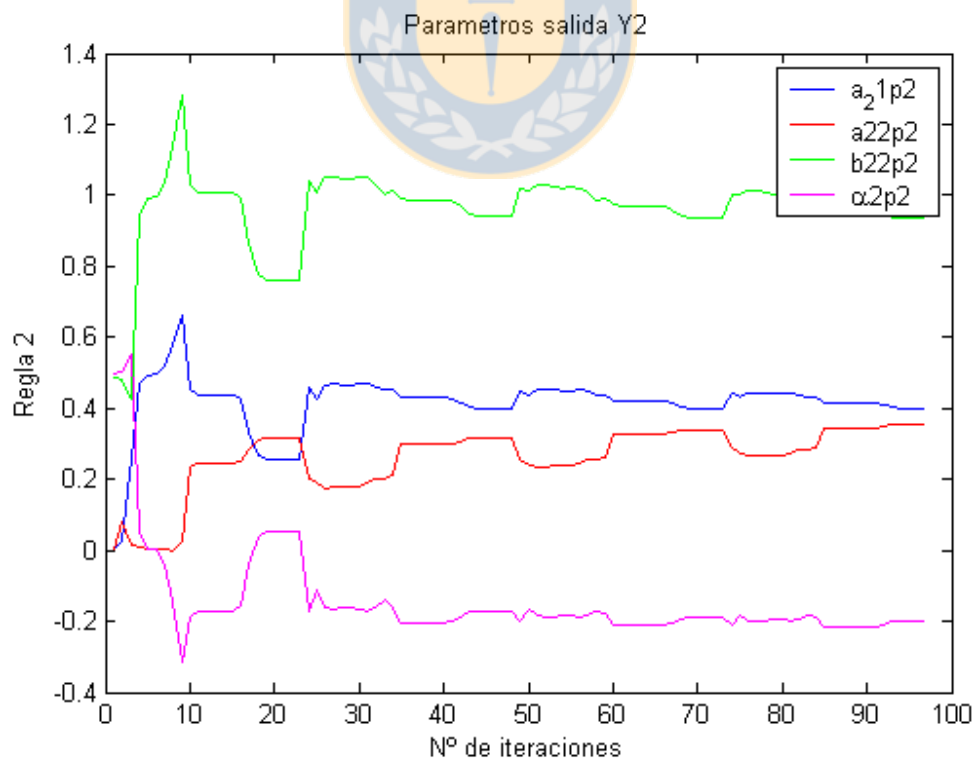


Figura 2.10: Evaluación de los parámetros de la salida y_2 regla 2

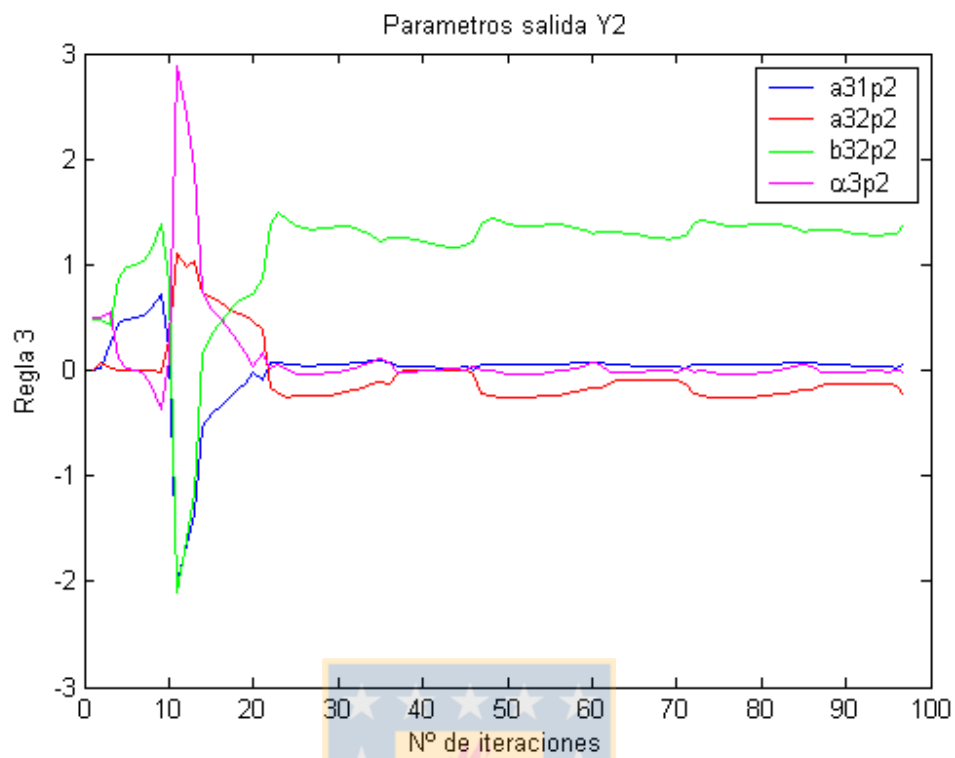


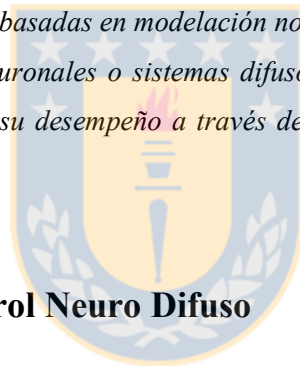
Figura 2.11: Evaluación de los parámetros de la salida y_2 regla 3

En estas últimas figuras se puede notar que los parámetros lineales varían hasta la muestra 30 aproximadamente. Después de este valor su variación es débil y prácticamente constante.

Capítulo 3

3. Control Neuro Difuso

En este capítulo se presentan técnicas de control neuro difuso para sistemas no lineales MIMO con salidas acopladas. En una primera fase, se describe la identificación del modelo inverso del proceso, que luego es utilizado directamente como controlador. En segundo lugar se estudia el control neuro difuso basado en modelos locales lineales describiendo dos arquitecturas posibles para este tipo de control. Luego se mencionan algunos esquemas de control emergentes producto de la combinación de técnicas de control basadas en modelación no lineal convencional, las cuales han sido adaptadas para el uso de redes neuronales o sistemas difusos. Finalmente se propone una de estas estrategias específicas y se evalúa su desempeño a través de algunos resultados experimentales con sistemas MIMO no lineales.



3.1. Introducción al Control Neuro Difuso

En los recientes años, han aparecido promisorios enfoques especialmente en el campo de redes neuronales y lógica difusa, los cuales han sido aplicados exitosamente. Cuando el bloque de control de un lazo de realimentación está compuesto por una red neuronal o un sistema de inferencia difusa, entonces se habla de un sistema de control neuronal o difuso respectivamente. El control neuro-difuso se refiere al diseño de controladores basados en lógica difusa que emplean técnicas de redes neuronales para adaptarse y para establecer objetivos de control. Como resultado de la combinación de ambos sistemas, se pueden destacar algunas características exclusivas de controladores neuro-difusos:

1. Capacidad de aprendizaje.
2. Operación paralela.
3. Poseen una forma estructurada del conocimiento (lo que permite incorporar información previa).
4. Buena integración con otros métodos de control.

Se debe considerar además que un controlador neuro-difuso generalmente contiene un gran número de parámetros, lo que implica una mayor versatilidad frente a características no-lineales de una planta, que un controlador lineal, por lo que son considerados superiores a estos últimos.

Los esquemas de control no lineal avanzados (control neuro-difuso), están generalmente basados en modelos no lineales. Para la aplicación de esta estrategia de control se debe elegir un adecuado modelo de la dinámica del proceso. El uso de modelos no lineales en el diseño de control impone demandas de certeza sobre la estructura del modelo mismo. Para un modelo de un proceso dado existen varias arquitecturas de control no lineal y técnicas de diseño.

Para los esquemas basados en modelos no-lineales, se pueden distinguir dos enfoques: el de modelo directo y el de modelo inverso [17] [18]. El esquema basado en el modelo inverso utiliza este último directamente como controlador [21]. El esquema de control basado en el modelo directo utiliza el modelo de la planta y luego sintetiza la ley de control usando un cálculo optimización/inversión predefinido. De los dos esquemas anteriores se abordará solamente el control basado en el modelo inverso.

Dependiendo de las aplicaciones, se pueden distinguir dos tipos de aprendizaje [21]: En el aprendizaje *off-line* (o aprendizaje por lotes), la actualización de los parámetros toma lugar solo después de que todos los datos de entrenamiento han sido presentados (época o barrido). Si el proceso es variante en el tiempo, o si no se tiene suficiente información para construir un buen modelo del proceso antes de la aplicación, entonces se deberían usar algoritmos de control adaptivo *on-line*. En este aprendizaje los parámetros son actualizados inmediatamente después de que un par de datos entrada-salida ha sido presentado. Generalmente primero se suele aplicar el aprendizaje *off-line* y basado en estos resultados se desarrolla un algoritmo de identificación *on-line* para sistemas con características cambiantes.

Entre los métodos que se han aplicado para determinar los parámetros de las funciones del antecedente, el método del gradiente descendiente o descenso de gradiente [2] [20] permite una gran exactitud en la identificación. El método del gradiente probablemente es el mejor algoritmo de aprendizaje ya que es una técnica idealmente adecuada para aprendizaje *on-line* instantáneo. Esto se debe a que es simple, tiene bajos requerimientos de memoria, y bajo costo computacional.

Para procesos variantes en el tiempo, el aprendizaje *off-line* puede ser usado solamente para la inicialización del modelo difuso [21], luego es necesario una adaptación *on-line* para las reglas.

3.2. Descripción Matemática del Modelo Difuso

El modelo a utilizar para realizar el control, es del tipo Takagi-Sugeno, este puede ser visto como un mapeo no lineal entre el modelo de entrada y el espacio de salida: $y=f(\varphi)$, donde “y”, denota la salida del modelo y “ φ ” el vector de regresión conteniendo las entradas del modelo difuso. Este modelo difuso puede ser formulado con un conjunto de reglas como sigue:

$$R_i: \text{If } x_1(k) \text{ is } \mu_{i1}(x_1) \text{ and ... and } x_p(k) \text{ is } \mu_{ij}(x_j), \text{ then } y = f(\varphi) \quad (3.1)$$

Donde $\mu_{ij}(x_j)$ es el i-ésimo antecedente difuso referido a la j-ésima variable de entrada, cuya función de pertenencia es presentada en la ecuación (A.4), que consta de los parámetros c_i y σ_i , y $\varphi = [x_1, \dots, x_j]$ es el vector de regresión. Además cuando f es de primer orden se puede expresar como:

$$f_i(\varphi) = p_{in}x_n + \dots + p_{i1}x_1 + p_{i0} \quad , \text{ con } i = 1, 2, \dots, c \quad (3.2)$$

Aquí el modelo fuzzy es lineal en los parámetros p_{ij} , con $j=1, 2, \dots, n$. y no lineal con respecto a los parámetros $a_{ij} = (c_{ij}, \sigma_{ij})$. Los parámetros p_{ij} pueden ser definidos por algoritmos de optimización lineal, mientras que los parámetros del antecedente deberían ser optimizados en alguna forma no-lineal.

3.3. Identificación Modelo Inverso del Proceso

Por simplicidad, el algoritmo será descrito y analizado para el caso de un sistema SISO (single input-single output), sin embargo la estrategia puede ser extendida al caso de un sistema MIMO. El sistema puede ser descrito por la siguiente representación discreta:

$$y(k+1) = F(y(k), \dots, y(k-n_y), u(k-\delta), \dots, u(k-n_u)) \quad (3.3)$$

Donde $y(k)$, es la salida de la planta, $u(k)$ es la señal de control discreta para el instante k , $F(\cdot)$ es una función no lineal desconocida y las constantes n_y, n_u definen el orden del proceso.

Asumiendo que:

$$\frac{\partial y(k+1)}{\partial u(k-\delta+1)} \neq 0 \quad (3.4)$$

Mientras $\partial y(k+1)/\partial u(k-i+1) = 0$, para $i < \delta$, el grado relativo del sistema será δ . Por sucesivas iteraciones de la ecuación (3.3) es posible encontrar un mapeo de entrada salida como el que sigue:

$$y(k+\delta) = \tilde{F}(y(k), \dots, y(k-n_y), u(k), \dots, u(k-n_u)) \quad (3.5)$$

Para este tipo de procesos, el inverso de la planta puede ser formulado como:

$$u(k) = \tilde{F}^{-1}(y(k+\delta), y(k), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u)) \quad (3.6)$$

o:

$$u(k-\delta) = \tilde{F}^{-1}(y(k), y(k-\delta), \dots, y(k-n_y-\delta), u(k-1-\delta), \dots, u(k-n_u-\delta)) \quad (3.7)$$

En la arquitectura de control adaptativa propuesta, el método de identificación/adaptación se utiliza para aprender el modelo inverso de la planta $u_m(k-\delta) = f(\varphi_m(k), \Theta)$, por ajuste de los pares de datos $\{u(k-\delta), \varphi_m(k)\}$ con la minimización del error de control de salida:

$$E(k) = \frac{1}{2} e_u^2 = \frac{1}{2} (\tilde{F}^{-1}(\varphi_m(k)) - f(\varphi_m(k), \Theta))^2 = \frac{1}{2} (u(k-\delta) - u_m(k-\delta))^2 \quad (3.8)$$

Donde el vector de entrada del modelo difuso es:

$$\varphi_m(k) = [y(k), y(k-\delta), \dots, y(k-n_y-\delta), u(k-1-\delta), \dots, u(k-n_u-\delta)]^T \quad (3.9)$$

y $\Theta = \{a_{ij}, p_{ij}\}$ representa el vector de parámetros.

3.4. Controlador Difuso Basado en el Modelo Inverso del Proceso

Este algoritmo de control asume que para un proceso inverso estable existe una única ley de control que permite un seguimiento cuasi perfecto:

$$u(k) = \tilde{F}^{-1}\left(r(k + \delta), y(k), \dots, y(k - n_y), u(k - 1), \dots, u(k - n_u)\right) \quad (3.10)$$

Donde $r(k+\delta)$ es el set-point del controlador. En control directo, el modelo identificado del proceso inverso es usado como controlador directamente, porque la acción de control es explícitamente calculada como la salida del modelo, que fue entrenada para representar la dinámica inversa de la planta. Sin embargo el modelo difuso inverso de la planta y el controlador difuso directo $u(k) = f(\varphi_c(k), \Theta)$, son idénticos en su estructura y parámetros. La única diferencia que hay es en el vector de entradas. Considerando que (3.9) es el vector de entrada para el modelo inverso del proceso, entonces:

$$\varphi_c(k) = \left[r(k + \delta), y(k), \dots, y(k - n_y), u(k - 1), \dots, u(k - n_u) \right]^T \quad (3.11)$$

Es el vector de entrada del controlador difuso. En la figura (3.1) se muestra un diagrama esquemático del sistema de control planteado, donde se puede observar que el controlador fuzzy mapea su vector de entrada $\varphi_c(k)$ en $u(k)$, con $u(k) = f(\varphi_c(k), \Theta)$ y el modelo inverso de la planta mapea $\varphi_m(k)$ en $u_m(k-\delta)$, con $u_m(k-\delta) = f(\varphi_m(k), \Theta)$.

En general, la inversión del modelo no-lineal del proceso no tiene solución analítica y deben ser aplicados métodos numéricos. Aquí, modelos locales lineales pueden de nuevo ser ventajosos puesto que los submodelos locales pueden ser invertidos sencillamente.

3.5. Control Basado en Modelos Neuro Difusos Locales Lineales

Hay básicamente dos formas para diseñar controladores para estructura de modelos locales, la estructura basada en linealización, y el enfoque basado en modelos locales. Ambos enfoques se pueden visualizar en la figura (3.2). Para el enfoque basado en linealización, el modelo no lineal global es linealizado en cada instante de muestreo, en el punto de operación dinámico actual, y subsecuentemente se diseña un controlador lineal. Este enfoque puede ser aplicado a cualquier tipo de modelo no lineal. Sin embargo, la linealización de las estructuras de los modelos locales es computacionalmente muy simple debido a su linealidad intrínseca. El segundo enfoque consiste en el

uso de estructuras con modelos locales. Aquí los controladores locales son diseñados para los submodelos individuales, y la salida de control es luego calculada como una interpolación de los controladores localmente válidos de acuerdo al punto de operación actual. Ambos enfoques producen un controlador no lineal globalmente, resultando en un enfoque adaptivo de parámetros. Considerando que el método basado en linealización generalmente produce resultados ligeramente mejores, el controlador local puede ser diseñado off-line. Consecuentemente el enfoque basado en el diseño de modelos locales tiene menos demanda computacional sobre el hardware en operación en tiempo real.

La estrategia que propone el modelado y control basados en modelos locales [11] [14] consiste en dividir un sistema complejo en *subsistemas* más sencillos que puedan ser analizados de forma individual. La clave, por tanto, consiste en la elección de aquellos parámetros adecuados que permitan realizar dicha partición. Estos subsistemas tienen un rango de validez limitado utilizando una base de reglas borrosa que permite la selección de los modelos adecuados a la situación en que se encuentra el sistema, es decir, cada regla corresponde a un estado o punto de trabajo del sistema.

Uno de los problemas que surgen a la hora de utilizar esta técnica es la selección del número de modelos locales que se deben emplear y la identificación de los distintos modelos locales utilizados. Una de las alternativas es construir el sistema difuso a partir de la experiencia y del conocimiento a priori que se tiene del comportamiento del proceso. Otra opción es la utilización de datos experimentales y extraer a partir de éstos la estructura del sistema de reglas difusas (extracción de reglas) y los parámetros de los modelos locales (identificación paramétrica). Este subsistema selecciona la combinación más adecuada de entre los distintos controladores diseñados para varios puntos de trabajo.

Dicho de otra forma, se trata de un sistema difuso del tipo Takagi-Sugeno, donde los consecuentes de las reglas son los distintos controladores diseñados para cada punto de trabajo. Los antecedentes, por su parte, son un conjunto de variables observadas que proporcionan la información necesaria para determinar de forma difusa el régimen de trabajo del proceso.

Por otra parte, se hace necesario un elemento supervisor que sea de algún modo capaz de integrar de nuevo todos estos *subsistemas* en el sistema original. Cuanto mayor sea el número de modelos locales elegidos, más sencillo será encontrar un controlador que funcione adecuadamente para cada uno de los regímenes. Sin embargo, esta propiedad se enfrenta a la denominada maldición de la dimensionalidad: una partición uniforme lleva a que el número de particiones crezca de forma exponencial con el número de variables.

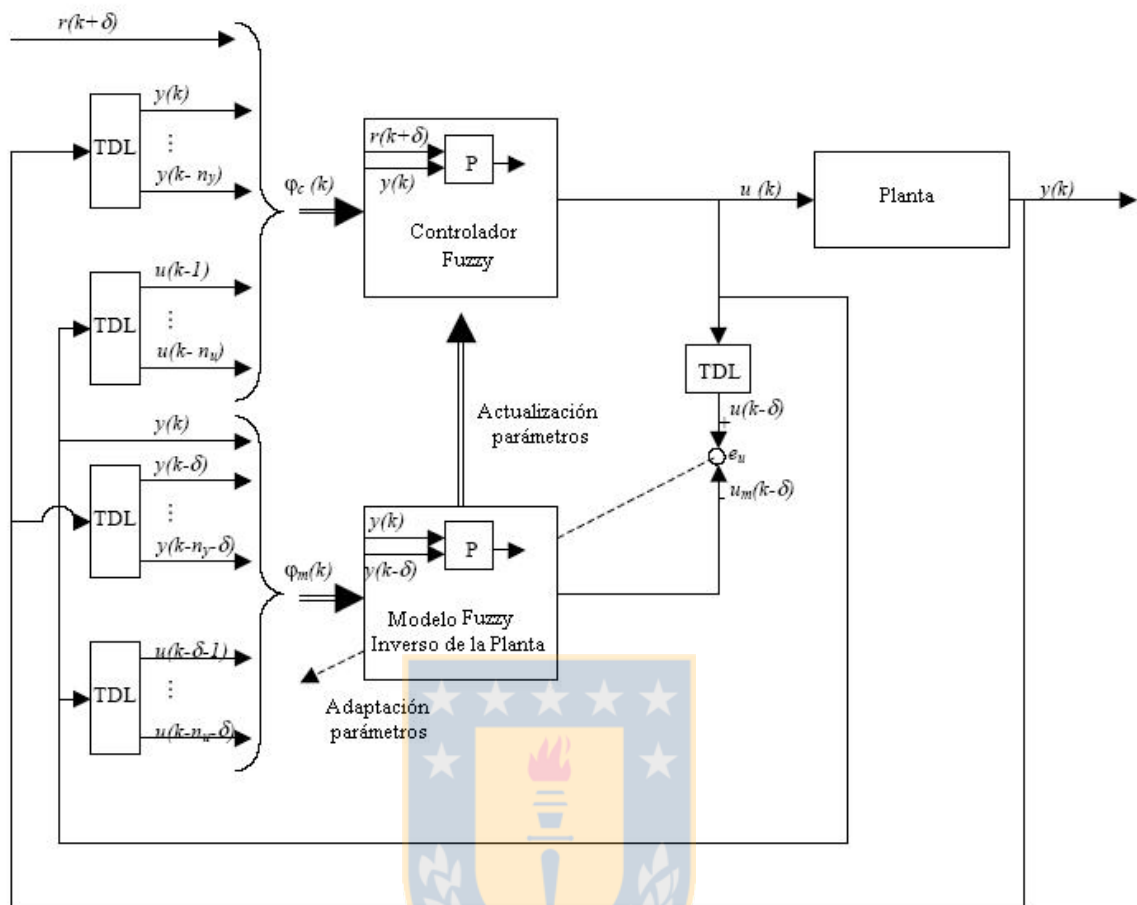


Figura 3.1: Arquitectura de control basada en el modelo inverso del proceso. Donde TDL denota un bloque de retardo en el tiempo.

Las arquitecturas con múltiples modelos poseen grandes ventajas, puesto que pueden ser aplicadas directamente técnicas de diseño de controladores lineales bien conocidas y sencillas de aplicar, explotando la linealidad local de los submodelos.

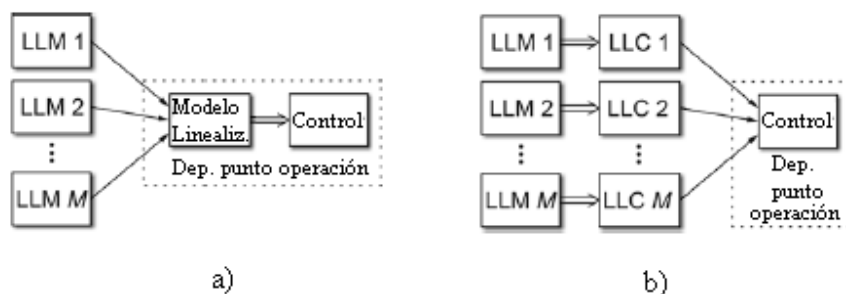


Figura 3.2: Diseño de control. a) Basado en linealización, b) basado en modelos locales

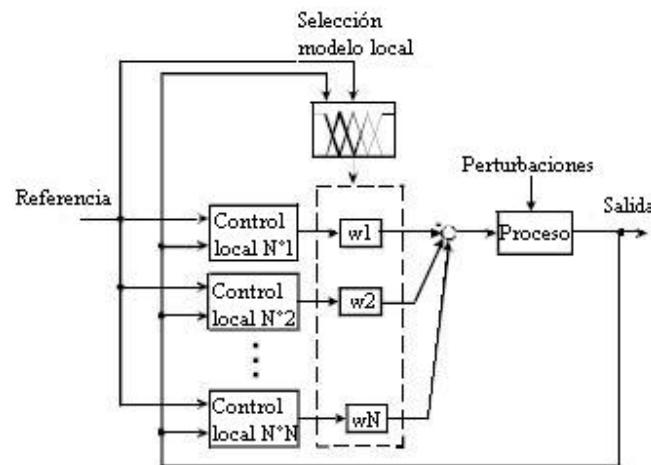


Figura 3.3: Arquitectura de control basada en modelos locales lineales.

3.6. Esquemas de Control Neuro Difuso Basados en Modelación

En los recientes años se han investigado diferentes técnicas de diseño de control no lineal basadas en lógica difusa y redes neuronales. La mayoría de ellas son técnicas de control basadas en modelación no lineal convencional, las cuales han sido adaptadas para el uso de redes neuronales o sistemas difusos. Entre estos enfoques se incluyen por ejemplo:

- Control realimentado clásico
- Linealización realimentada.
- Control modelo referencia
- Control predictivo.
- Control realimentado con modelo inverso.

Algunas de estas estructuras son descritas brevemente a continuación:

3.6.1. Control Híbrido

Fue pensado en un principio como un esquema de aprendizaje *off-line* para las leyes de control deseadas, el cual es llamado también *aprendizaje generalizado*. De esta forma el control neuronal es capaz de manejar la planta sobre un rango de operación sin inestabilidad. Se puede utilizar un período de *aprendizaje especializado* on-line, proveído por el controlador en base a redes neuronales. Una alternativa de esta configuración es mostrada en la figura (3.2), donde se observa que es posible utilizar un controlador lineal de ganancia fija en paralelo con la ley de control neuronal. Esta ley de control de

ganancia fija es elegida primero para estabilizar la planta. La planta es luego manejada sobre el rango de operación con la red neuronal sintonizada on-line para mejorar el control. Este esquema ha sido usado frecuentemente para sistemas de control de vuelo.

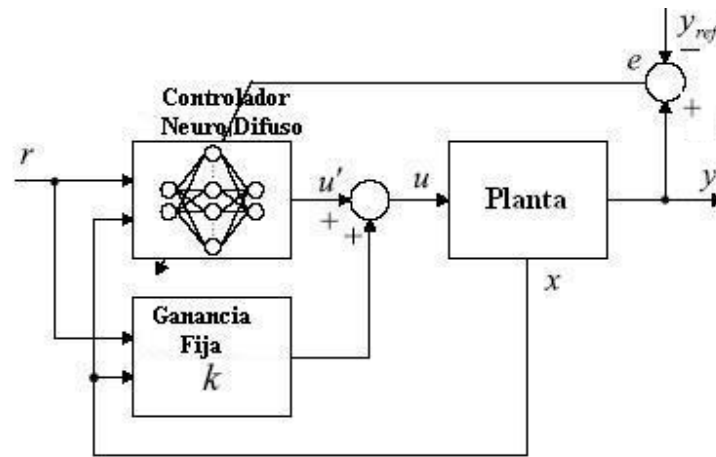


Figura 3.4: Esquema control híbrido

3.6.2. Control con Modelo de Referencia

El esquema de control presentado anteriormente no considera la capacidad de seguimiento. En este esquema, la ejecución deseada del sistema en lazo cerrado es especificada a través de un modelo de referencia, el cual es definido por los pares entrada salida $\{r(t), y_r(t)\}$. Como muestra la figura 3.5, el sistema de control intenta hacer que la salida de la planta siga a la salida del modelo de referencia asintóticamente. En este esquema, el error entre la planta y la salida del modelo de referencia es usado para ajustar los pesos del controlador neuronal.

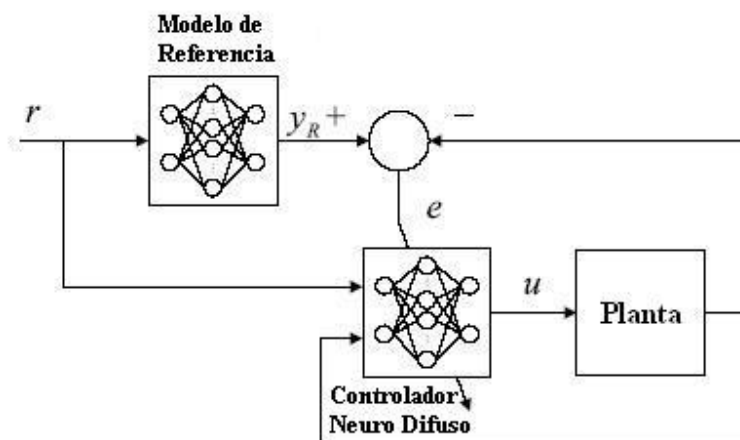


Figura 3.5: Esquema de control modelo de referencia.

3.6.3. Control Lineal o No lineal Adaptivo

Este enfoque puede ser usado no solamente en control no lineal, sino también como una parte de un controlador para plantas lineales. Si el proceso tiene comportamiento variante en el tiempo, o si hay perturbaciones que no pueden ser medidas y que afectan la salida del proceso, es deseable tener una componente adaptiva en el lazo de control. En el control basado en el modelo, éste último es adaptado online y el controlador es adaptado de acuerdo a esto. Para modelos no lineales, esto usualmente requiere técnicas de adaptación no lineales. Sin embargo, esos parámetros que han sido estimados usando métodos de mínimos cuadrados lineales, pueden ser adaptados con técnicas de adaptación lineales como algoritmos de mínimos cuadrados recursivos. El costo del error de seguimiento es evaluado de acuerdo a algún índice de rendimiento. El resultado es luego usado como base para ajustar los pesos de las conexiones de la red neuronal. Estos pesos son ajustados on-line usando aprendizaje por retropropagación. El esquema de control correspondiente es mostrado en la figura 3.6. En aplicaciones prácticas, este diseño de control explota las ventajas de las redes neuronales y la robustez de la teoría de control adaptivo.

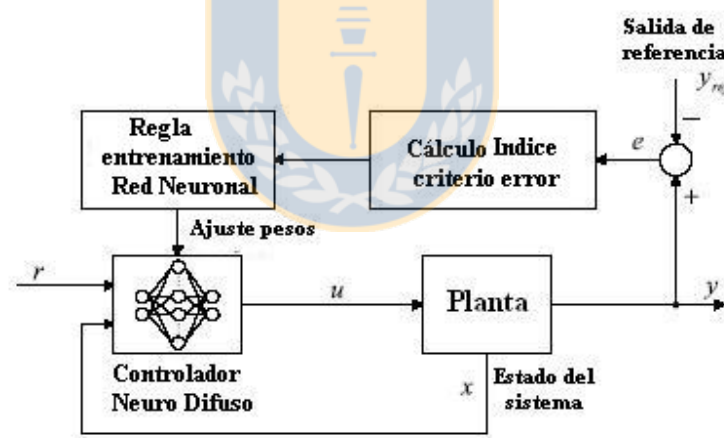


Figura 3.6: Esquema de control adaptivo.

3.6.4. Control Predictivo

Dentro de la familia de los métodos de control óptimo y predictivo, la reciente técnica de horizonte ha sido introducida como una ley de control realimentado muy factible computacionalmente. En este enfoque, una red neuronal provee la predicción de la respuesta futura de la planta sobre un horizonte especificado. Las predicciones suministradas por la red son luego pasadas en una rutina de

optimización numérica, la cual intenta minimizar un criterio de rendimiento especificado en el cálculo de una adecuada señal de control. Esta variable manipulada es determinada por optimización de la función de costo siguiente:

$$J = \sum_{j=N_1}^{N_2} (\hat{y}(k+j) - r(k+j))^2 + \beta \sum_{j=0}^{N_u-1} \Delta u(k+j)^2 \quad (3.10)$$

Donde el primer termino penaliza la desviación de la salida del modelo $\hat{y}(k+j)$ de la trayectoria de la referencia sobre el horizonte de predicción $[N_1 N_2]$. El segundo término evalúa el esfuerzo de control para los N_u valores de la señal de control optimizada. Para el cálculo de los valores de la señal predictiva $\hat{y}(k+j)$, se usa el modelo del proceso. Los controladores predictivos usualmente llevan a buenos resultados. Puesto que la determinación de la variable manipulada es solamente basada en el modelo del proceso, la ejecución de control depende fuertemente del modelo. El problema de costo computacional puede ser superado en parte por uso de controladores predictivos generalizados, los cuales pueden estar basados en arquitecturas de modelos locales. Estos controladores son mas comúnmente aplicados al control de la industria química, y su esquema puede ser visualizado en la figura 3.7.

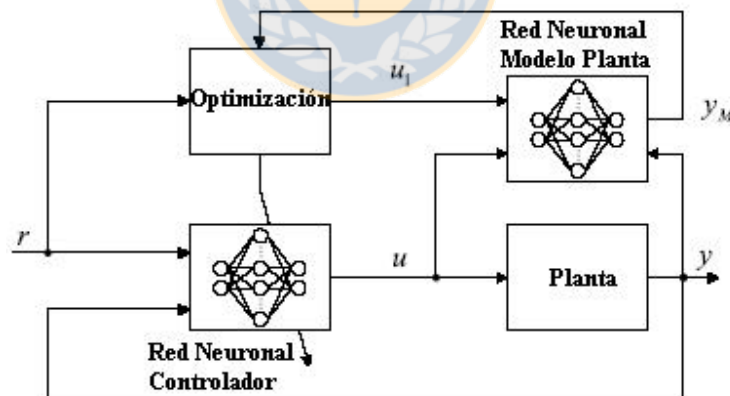


Figura 3.7: Esquema de control predictivo.

3.6.5. Control de Modelo Interno (IMC)

Otra técnica de control basada en el modelo que es menos cara computacionalmente que el control predictivo es el enfoque del IMC. La estructura del IMC es algo diferente que el lazo de control

realimentado standard, ya que posee diferentes propiedades comparado con los sistemas realimentados convencionales. Por ejemplo, en el caso nominal cuando el modelo es idéntico al proceso, el lazo de realimentación es igual a cero tal que el sistema pasa a ser un lazo abierto, y por tanto pueden surgir problemas de inestabilidad.

Específicamente, en este esquema se enfatiza el rol del modelo directo e inverso del sistema. Como muestra la figura 3.8, tanto el modelo directo e inverso del sistema son usados directamente como elementos dentro del lazo de realimentación. La red NN_1 es primero entrenada off-line para emular las dinámicas de la planta controlada directamente. Durante la operación online, el error entre el modelo y la salida de la planta es usado como una señal realimentada y pasada al control neural NN_2 . El efecto de NN_1 es restar el efecto de la señal de control de la salida de la planta, es decir, la señal realimentada es solamente la influencia debido a perturbaciones, lo cual no puede ser realizado por un controlador pre-alimentado tradicional. El IMC ha sido minuciosamente examinado y se ha demostrado que posee gran robustez (Hunt y Sbarbaro-Hofer, 1991). Este enfoque puede ser extendido al diseño de autopilotos bajo perturbaciones externas.

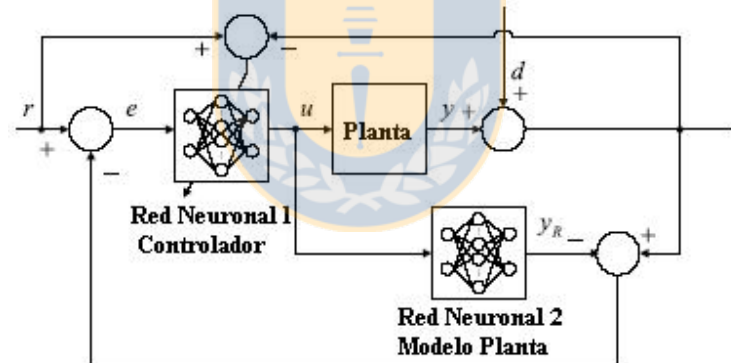


Figura 3.8: Esquema de control modelo interno.

3.7. Desarrollo Teórico del Método de Control a Implementar

La estrategia de control que se utilizó esta memoria es una combinación de algunas arquitecturas mencionadas anteriormente como por ejemplo control adaptivo, predictivo y control de modelo interno (IMC), heredando por tanto las ventajas y robustez de éstos. Esta estrategia integrada consiste en un esquema de control neuro-difuso basado en el modelo inverso del proceso para tres regímenes de operación distinta. El sistema de control, es ejercido empleando de nuevo la base de

$$u(k) = \frac{\sum_{i=1}^c \mu_i(x(k))u_i}{\sum_{i=1}^K \mu_i(x(k))} \quad (3.12)$$

o:

$$u(k) = \sum_{i=1}^c u_i(k)\Phi_i(x, c_i, \sigma_i) \quad (3.13)$$

Donde $\mu_j(x(k))$ es el producto de las funciones de pertenencia de las variables de entrada que son del tipo gaussiana, como muestra la expresión (2.15). La función de validación $\Phi_i(x, c_i, \sigma_i)$ es de la modelación expresada en la ecuación (2.12). La deducción del controlador se puede visualizar mejor con el desarrollo más detallado del modelo del sistema MIMO presentado anteriormente.

El sistema está regido por las siguientes ecuaciones:

$$y_1(k+1) = \frac{y_1(k)}{1 + y_2^2(k)} + u_1(k) \quad (3.14)$$

$$y_2(k+1) = \frac{y_1(k)y_2(k)}{1 + y_2^2(k)} + u_2(k) \quad (3.15)$$

De donde se ve la clara dependencia en cada salida de su respectivo valor auto-regresivo, de la otra salida, y de su respectiva entrada. Por lo tanto el modelo directo del proceso MIMO, para el régimen de operación correspondiente a la regla i que se plantea tiene la forma:

$$\begin{aligned} y_{1i}(k+1) &= a_{i1p1}y_1(k) + a_{i2p1}y_2(k) + b_{i1p1}u_1(k) + \alpha_{ip1} \\ y_{2i}(k+1) &= a_{i1p2}y_1(k) + a_{i2p2}y_2(k) + b_{i1p2}u_2(k) + \alpha_{ip2} \end{aligned} \quad (3.16)$$

Como se puede observar este modelo planteado anteriormente es un modelo predictivo-adaptivo con un horizonte de predicción igual a 1. La red adaptable de la figura 3.9 provee la predicción de la respuesta futura de la planta sobre el horizonte especificado. Esta red adapta los parámetros entrenables como son los centros y anchos (c_{ij}, σ_{ij}) de las funciones de pertenencia con el método del descenso del gradiente como fue descrito en el capítulo 1, que busca minimizar el índice:

$$\varepsilon = \frac{1}{2} \sum_{k=1}^r (y_{zk} - y_k)^2 \quad (3.17)$$

Siendo y_{zk} el set-point en este caso e y_k la salida del proceso.

El costo del error de seguimiento es evaluado de acuerdo a este índice de rendimiento. El resultado es luego usado como base para ajustar los pesos de las conexiones de la red adaptable puesto que al adaptar estos parámetros se está adaptando los pesos de validación $\Phi_i(x, c_i, \sigma_i)$ de cada sub-modelo.

Los parámetros del consecuente son estimados usando un método de mínimos cuadrados recursivos como fue mostrado en detalle en el capítulo anterior.

Finalmente, si se requiere calcular el modelo inverso, basta sólo con realizar una sencilla invertibilidad discreta sobre la expresión indicada anteriormente. La inversión en este caso consiste en un simple despeje de la variable de control u , dado a la simplicidad intrínseca que tienen los modelos locales lineales:

$$\begin{aligned} u_{1i}(k) &= \frac{y_{1i}(k+1) - (a_{i1p1}y_1(k) + a_{i2p1}y_2(k) + \alpha_{ip1})}{b_{i1p1}} \\ u_{2i}(k) &= \frac{y_{2i}(k+1) - (a_{i1p2}y_1(k) + a_{i2p2}y_2(k) + \alpha_{ip2})}{b_{i1p2}} \end{aligned} \quad (3.18)$$

Aquí es donde puede verse la utilidad del empleo de modelos locales lineales, ya que esto permite el planteamiento a su vez de leyes de control locales lineales estables. Por consiguiente si se desea obtener el controlador basado en el modelo inverso se reemplaza $y(k+1)$ por $r(k+1)$ (referencia o set-point) y se tiene:

$$\begin{aligned} u_{1i}(k) &= \frac{r_1(k+1) - (a_{i1p1}y_1(k) + a_{i2p1}y_2(k) + \alpha_{ip1})}{b_{i1p1}} \\ u_{2i}(k) &= \frac{r_2(k+1) - (a_{i1p2}y_1(k) + a_{i2p2}y_2(k) + \alpha_{ip2})}{b_{i1p2}} \end{aligned} \quad (3.19)$$

En el control basado en el modelo, éste último es adaptado online y el controlador es adaptado de acuerdo a esto.

Finalmente la salida de control global está dada por la suma ponderada de cada uno de los controladores locales por su respectiva función de validación como lo muestra la expresión (3.13).

La estrategia de control Neuro Difusa planteada combinada con el control predictivo-adaptivo usualmente lleva a buenos resultados puesto que es bastante versátil ante procesos que tiene comportamiento variante en el tiempo, o si hay perturbaciones externas que no pueden ser medidas y que afectan la salida del proceso. Esto es debido a que el modelo tiene características adaptables a través de sus parámetros los que permiten compensar los cambios en los parámetros del sistema y el efecto de las perturbaciones. Se puede decir entonces que los parámetros son los encargados de “absorber” estos efectos.

Otro aspecto de gran relevancia en este tipo modelación es el concepto de la *transición de los submodelos*. Existen dos tipos de transición: una transición dura (hard switching) o una transición suave (smooth transition) entre los submodelos locales. En el primer caso (también conocido como Discretización lógica), solamente un modelo local es válido dentro de cada subregión, es decir, cada punto de operación tiene un modelo que lo representa exactamente como una función determinística de dicho punto de operación. En este caso no existe sobreposición entre los modelos, y éstos son locales sólo dentro de los puntos de operación utilizados para el desarrollo de los mismos.

Para el segundo caso (utilizado en este trabajo), la idea es proveer un cambio gradual entre el comportamiento de los submodelos colindantes. En este caso al igual que en el anterior se generan los modelos adecuados para ciertos puntos de operación de manera que cumplan con los requisitos establecidos para ellos. La principal diferencia con la discretización lógica, es que al pasar de una zona de operación a otra, este paso se realiza en forma más suave y a través de funciones de pertenencia.

3.7.1. Resultados Simulación

El sistema a ser controlado es el mismo utilizado en el capítulo anterior regido por las ecuaciones (2.25) y (2.26). Los resultados de las simulaciones son mostrados a continuación:

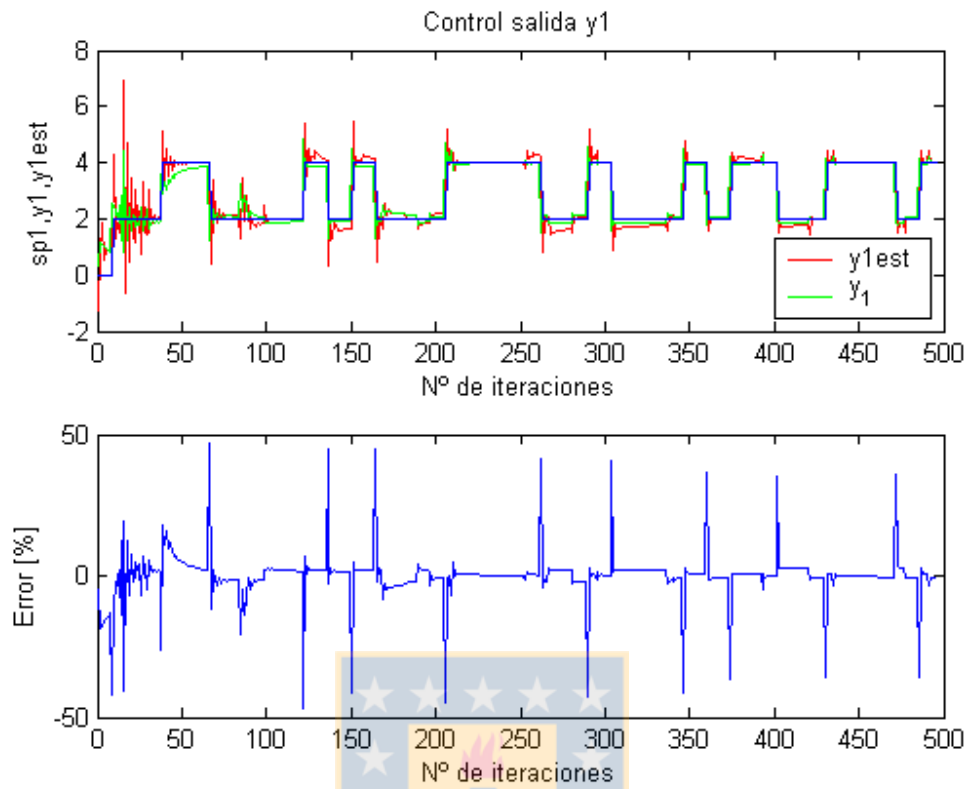


Figura 3.10: Respuesta de control salida y_1 y su respectivo error de seguimiento.

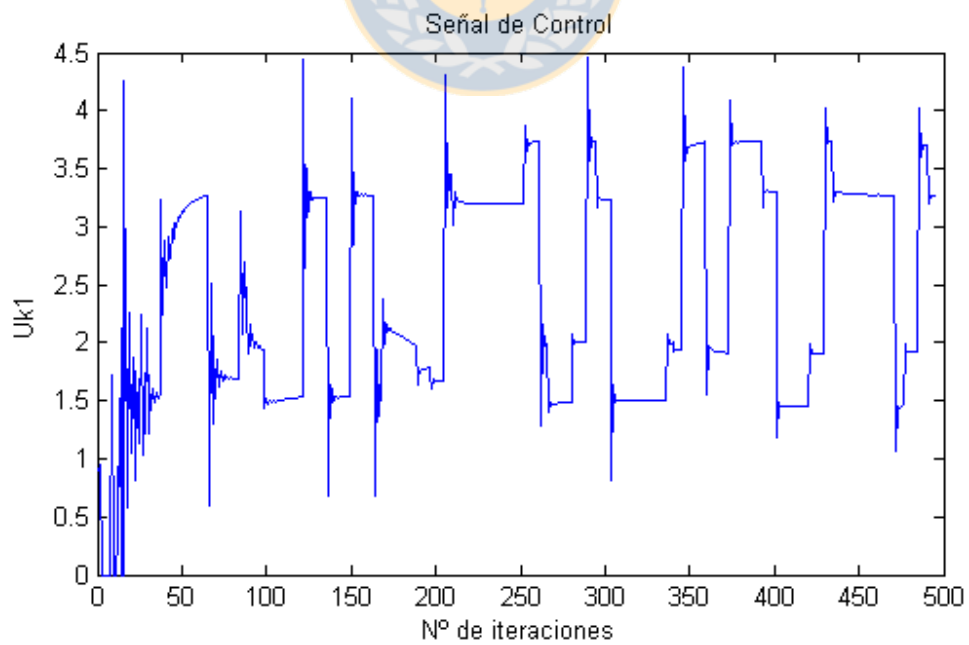


Figura 3.11: Señal de control U_{k1} .

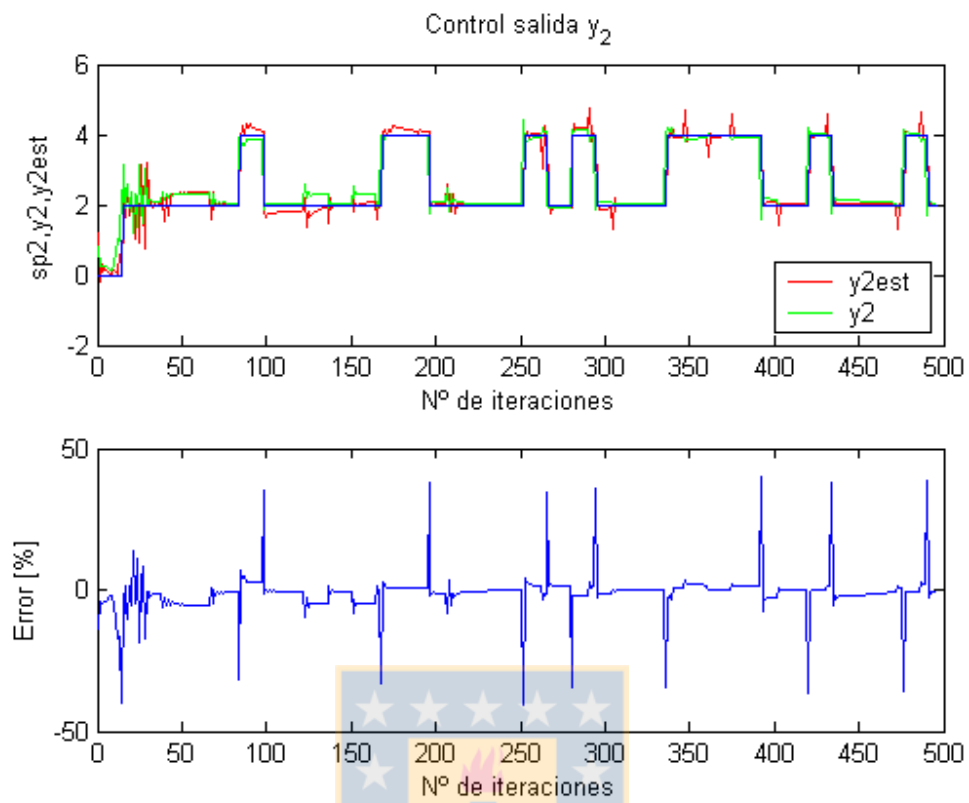


Figura 3.12: Respuesta de control salida y_2 y su respectivo error de seguimiento.

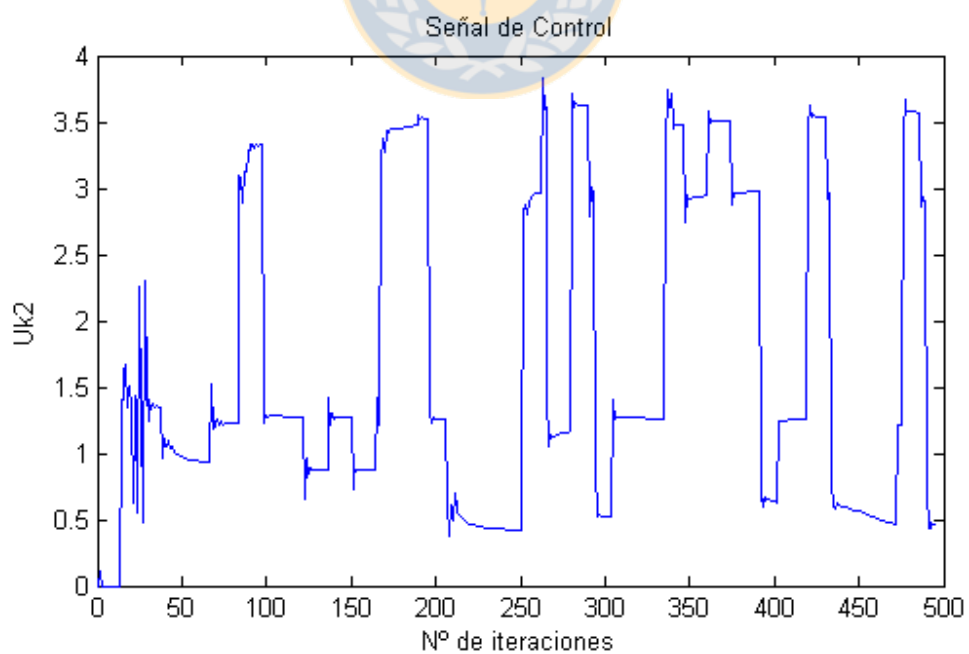


Figura 3.13: Señal de control U_{k2} .

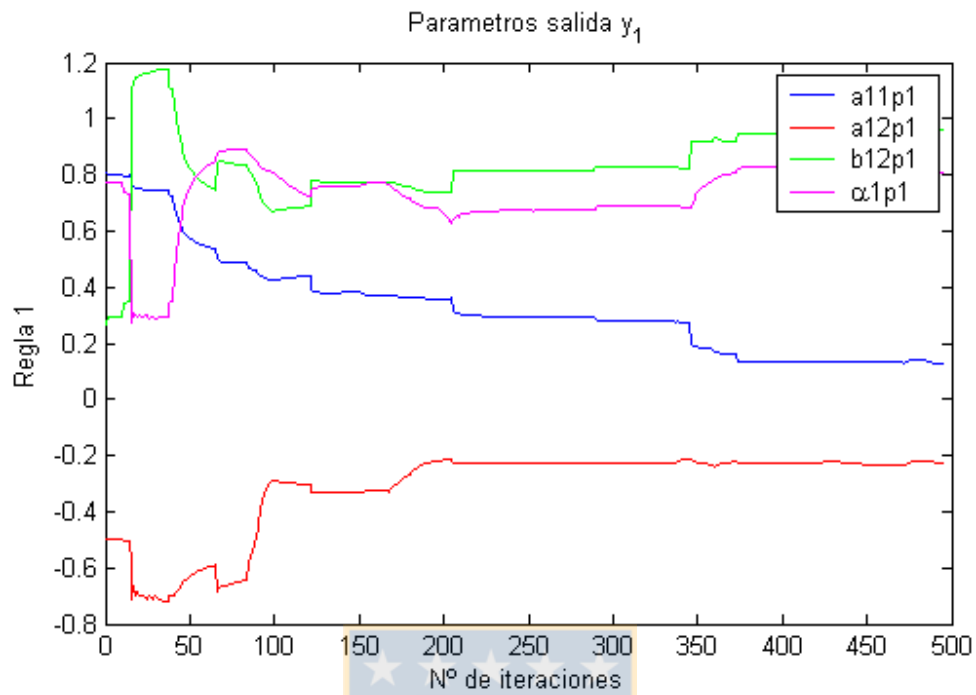


Figura 3.14: Evaluación de los parámetros de la salida y_1 regla 1.

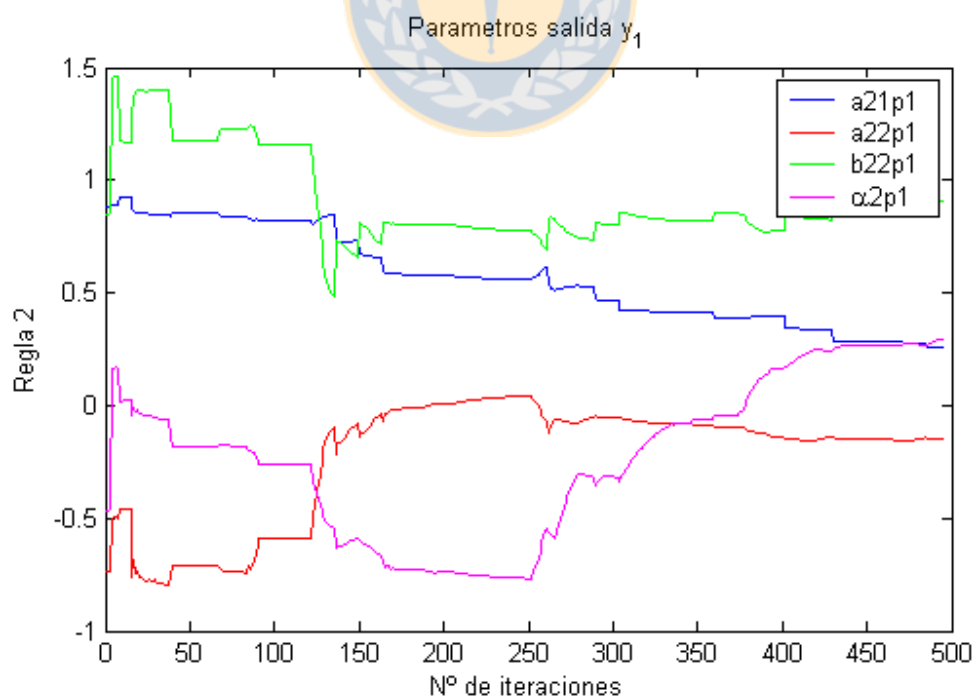


Figura 3.15: Evaluación de los parámetros de la salida y_1 regla 2.

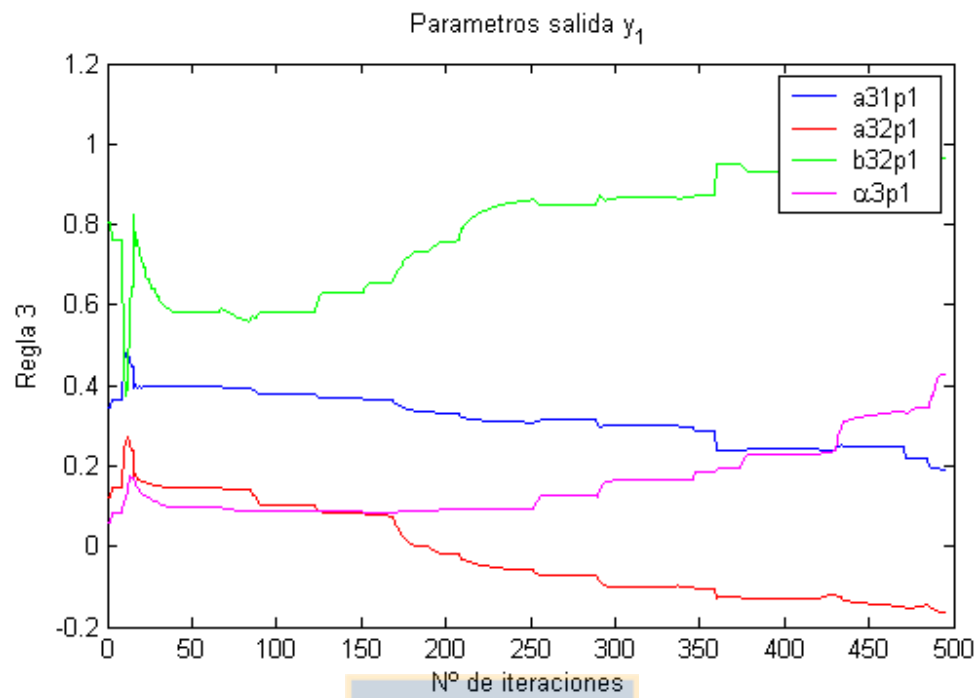


Figura 3.16: Evaluación de los parámetros de la salida y_1 regla 2.

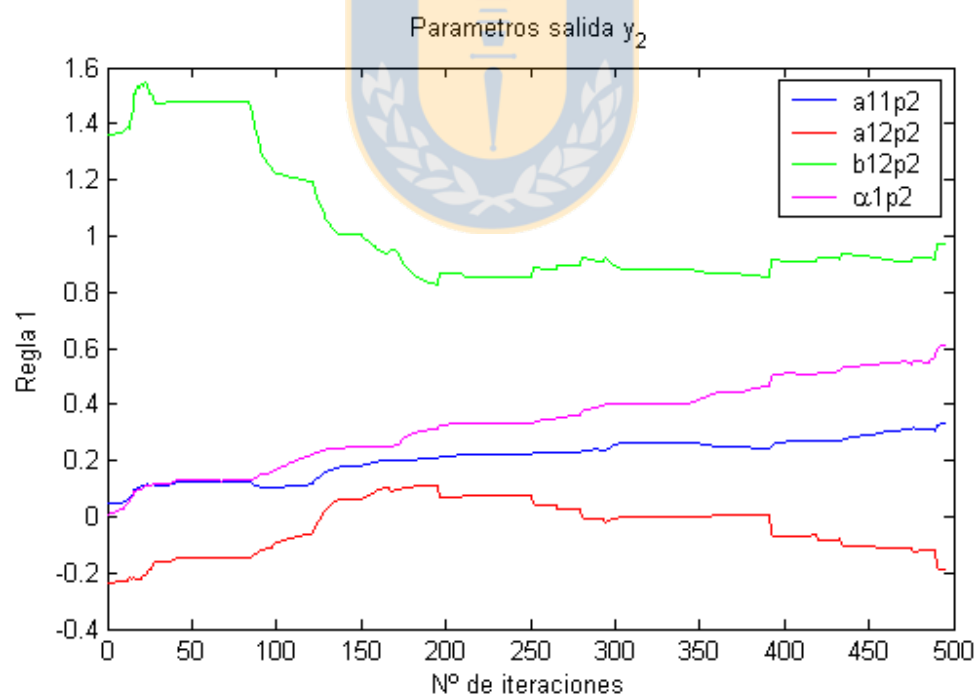


Figura 3.17: Evaluación de los parámetros de la salida y_2 regla 1.

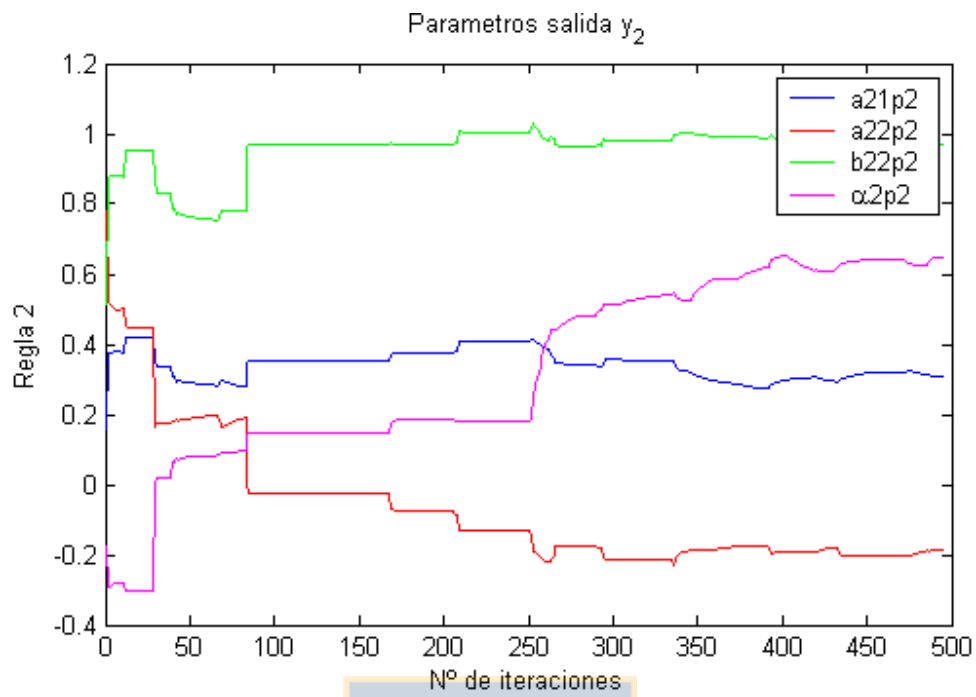


Figura 3.15: Evaluación de los parámetros de la salida y_2 regla 2.

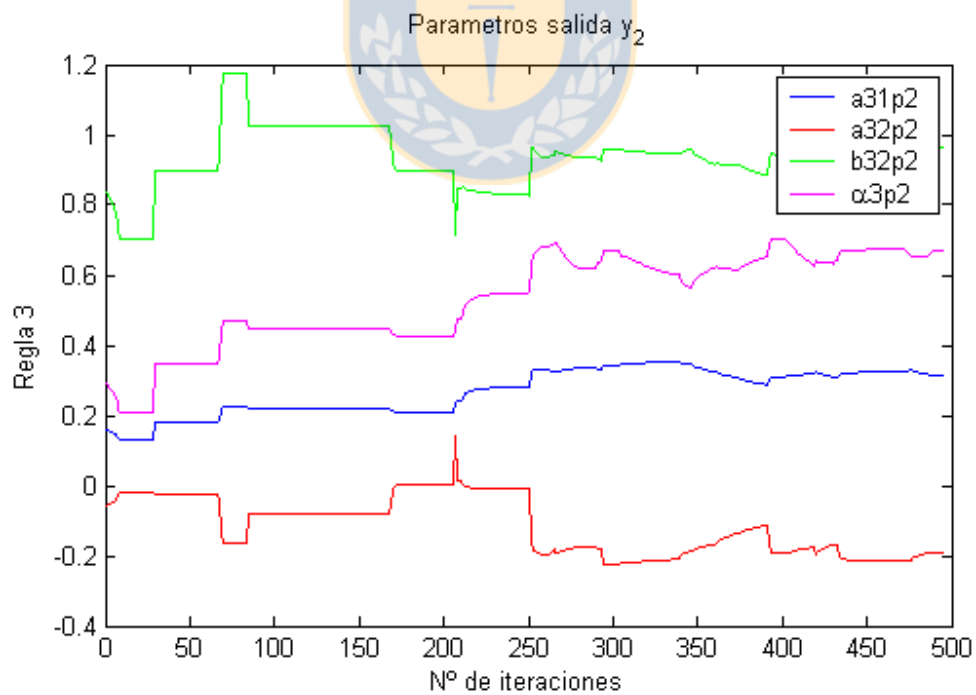


Figura 3.16: Evaluación de los parámetros de la salida y_2 regla 3.

3.8. Control Planta Separación Gas-Líquido

3.8.1. Descripción del Sistema a Controlar

El sistema a controlar consiste en un sub-proceso dentro de una instalación semi-industrial, la cual es usada para la reducción de NO_x efluente de gases y tratamiento de aguas con residuos tecnológicos por medio de reducción con CO_2 contenido en el flujo de gases. El rol de la unidad de separación es capturar el vapor de los gases que estén bajo la presión baja desde los canales efluentes a través de la corriente de agua y llevarlos justo hasta la presión alta para la etapa de neutralización (sentido descendente). La unidad de separación es mostrada en la figura 3.17.

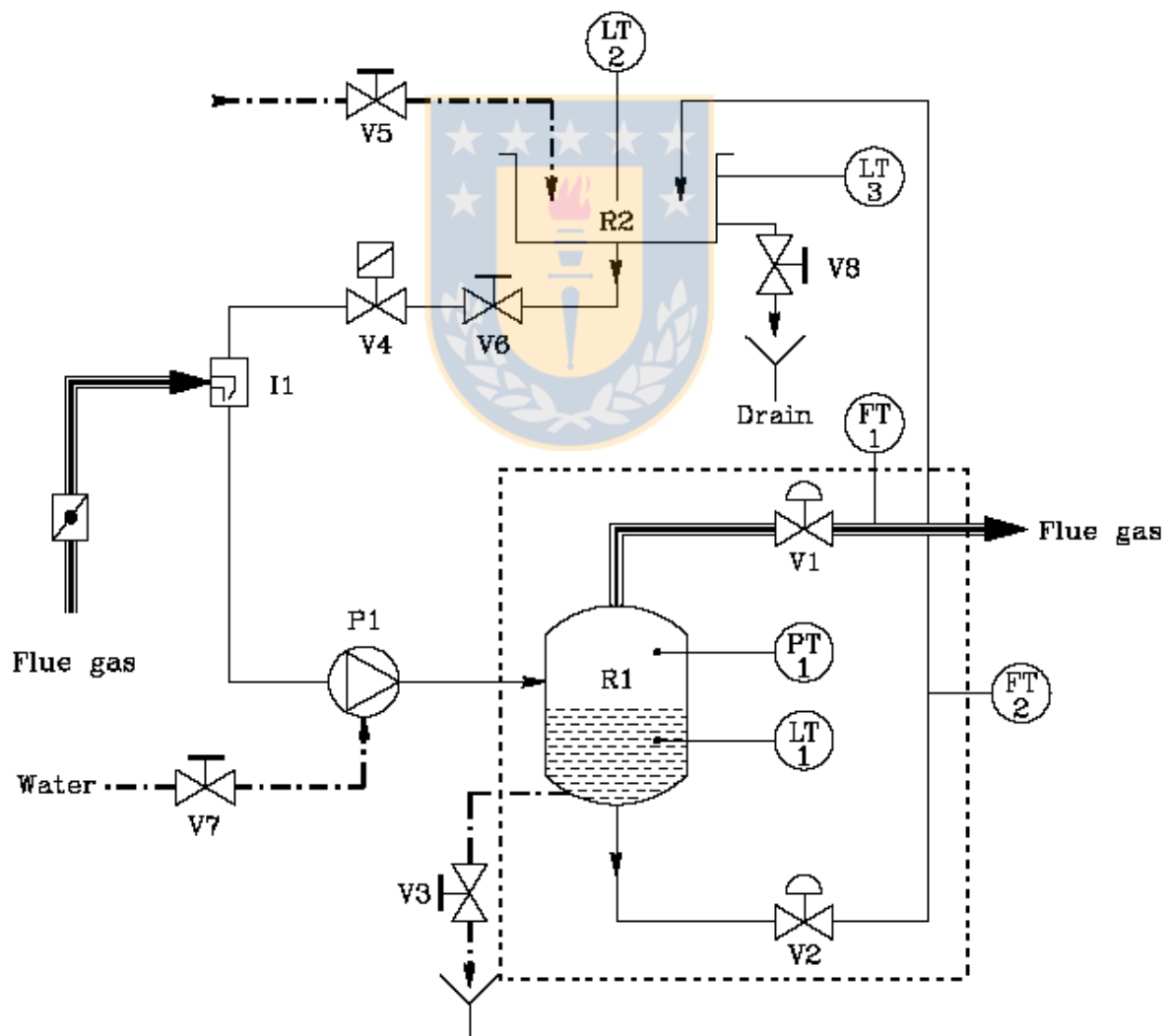


Figura 3.17: Unidad de separación gas-líquido.

Los vapores de los gases que vienen de los canales efluentes son reunidos por el flujo de agua dentro del tanque de circulación de agua a través del inyector I_1 . El flujo de agua es generado por la bomba P_1 . La velocidad de la bomba es mantenida constante. La bomba alimenta la mezcla de agua y gas dentro del separador R_1 donde el gas es separado del agua. Por lo tanto el gas acumulado en R_1 forma una clase de “amortiguador del gas entrante” con la presión interna aumentada. La tarea de esta presión es “soplar”, o dicho de otra forma expulsar fuera de R_1 el vapor del gas hacia la próxima unidad de neutralización. Por otra parte el “amortiguador” obliga al agua a circular hacia el deposito R_2 . La cantidad de agua en el circuito es constante. Si por alguna razón adicional se necesita agua, se utiliza la trayectoria de abastecimiento de agua de la válvula V_5 .

Para efectos de simplificación y como una forma de acotar el sistema a un proceso del tipo TITO (Two-Input Two-Output), se realizaron ciertas modificaciones al sistema de tal forma que se tendrá como objetivo controlar el sub-sistema comprendido dentro del rectángulo con líneas punteadas en la figura 3.17. Este sub-sistema es mostrado a continuación:

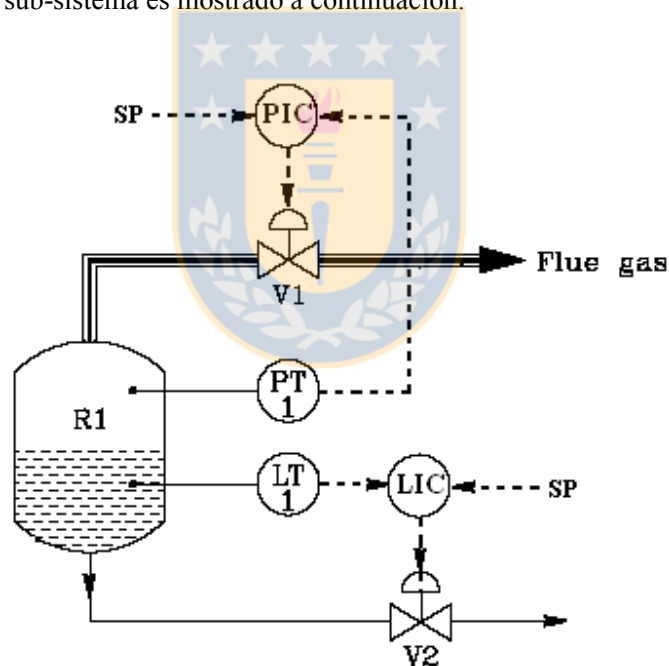


Figura 3.18: Esquema típico (lazos desacoplados) diagrama P&I del proceso a controlar.

El control de presión se llevará a cabo a través de la válvula de salida de aire V_1 y para el nivel se utilizará la válvula de descarga de agua V_2 . Esto es debido principalmente a que el modelo está desarrollado considerando estos actuadores como variables de entrada del sistema.

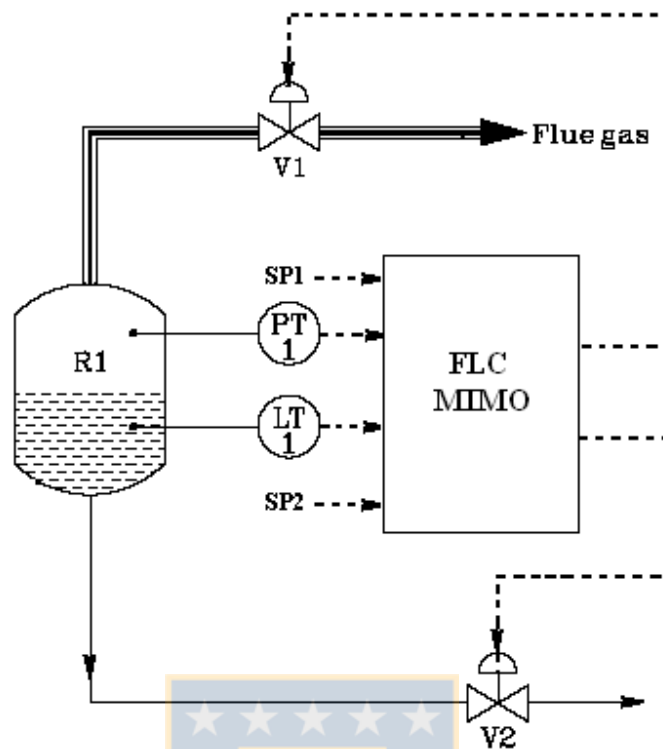


Figura 3.19: Esquema Fuzzy Logic Control del proceso MIMO a controlar.

Del diagrama de la figura 3.19 se desprende que es posible controlar la presión interna y el nivel del estanque para la evaluación del sistema Neuro-Difuso, ya que constituyen en su conjunto un sistema MIMO (en este caso TITO) en donde ambas variables poseen un gran acoplamiento, además de la alta no linealidad que representan ambos lazos. El esquema del sistema

3.8.2. Modelación del Sistema

El modelo con algunas simplificaciones puede ser descrito por las siguientes expresiones:

$$\text{Válvula } V_1: \quad K_1 = K_{01} \cdot R_{V1}^{v_1-1} = 75.1 \cdot 46.1^{v_1-1} \left[l/(s\sqrt{\text{bar}}) \right] \quad (3.20)$$

$$v_1 = \begin{cases} 1 & r_1 > 1 \\ 0 & r_1 < 0 \\ r_1 & \text{otro caso} \end{cases} \quad (3.21)$$

$$\text{Válvula } V_2: \quad K_2 = K_{02} \cdot R_{V_2}^{v_2-1} = 0.742 \cdot 75.66^{v_2-1} \left[l/(s\sqrt{\text{bar}}) \right] \quad (3.22)$$

$$v_2 = \begin{cases} r_{2\max} & r_2 > r_{2\max} \\ 0 & r_2 < 0 \\ r_2 & \text{otro caso} \end{cases} \quad (3.23)$$

Con $r_{2\max} = 0.8625$

$$\text{Flujo aire a través de Válvula } V_1: \quad \Phi_1 = K_1 \sqrt{p_1} \left[l/s \right] \quad (3.24)$$

$$\text{Flujo agua a través de Válvula } V_2: \quad \Phi_2 = K_2 \sqrt{p_1 + K_w \cdot h_1} \left[l/s \right] \quad (3.25)$$

$$\text{Flujo agua hacia el separador } R_1: \quad \Phi_w = 0.1644 \left[l/s \right] \quad (3.26)$$

$$\text{Flujo aire hacia el separador } R_1: \quad \Phi_{\text{aire}} = \Phi_{\text{aire}0} + \Phi_{\text{aire}1} \cdot p_1 = 6.46 - 1.615 \cdot p_1 \left[l/s \right] \quad (3.27)$$

El cambio de presión del aire dentro de R_1 es:

$$\frac{dp_1}{dt} = \frac{1}{V} \left[p_0 (\Phi_{\text{aire}} - \Phi_1) K_F + (p_0 + p_1) (\Phi_w - \Phi_2) K_F \right] \left[\text{bar} / s \right] \quad (3.28)$$

El cambio de nivel del agua en R_1 es:

$$\frac{dh_1}{dt} = \frac{1}{S_1} (\Phi_w - \Phi_2) \cdot K_F \left[m/s \right] \quad (3.29)$$

El volumen de aire dentro de R_1 es:

$$V = S_1 (h_{R1} - h_1) = S_1 (2.25 - h_1) \left[m^3 \right] \quad (3.30)$$

Donde:

$$\begin{aligned}
K_w &= 0.0981 & [bar / m] \\
K_F &= 1e^{-3} & [m^3 / l] \\
p_0 &= 1.033 & [bar] \\
\Phi_w &= 0.1644 & [l / s] \\
S_1 &= 0.312 & [m^2]
\end{aligned} \tag{3.31}$$

De la ecuación (3.28) se puede ver claramente que el cambio de la variable de salida presión p_1 (dp_1) depende de las variables p_1 , h_1 , V_1 y V_2 . De la ecuación (3.29) se puede observar además que el cambio de la altura h_1 (dh_1) depende de las variables p_1 , h_1 , y V_1 . Por lo tanto el modelo Takagi-Sugeno (T-S) que se propone para ambas salidas correspondiente a la regla “i” es el siguiente:

$$\begin{aligned}
p_{1i}(k+1) &= a_{i1p1}p_1(k) + a_{i2p1}h_1(k) + b_{i1p1}V_1(k) + b_{i2p1}V_2(k) + \alpha_{ip1} \\
h_{1i}(k+1) &= a_{i1p2}p_1(k) + a_{i2p2}h_1(k) + b_{i1p2}V_2(k) + \alpha_{ip2}
\end{aligned} \tag{3.32}$$

Luego se desprende que el modelo (T-S) inverso queda como sigue:

$$\begin{aligned}
V_{1i}(k) &= \frac{p_{1i}(k+1) - (a_{i1p1}p_1(k) + a_{i2p1}h_1(k) + \alpha_{ip1})}{b_{i1p1}} \\
V_{2i}(k) &= \frac{h_{1i}(k+1) - (a_{i1p2}p_1(k) + a_{i2p2}h_1(k) + \alpha_{ip2})}{b_{i1p2}}
\end{aligned} \tag{3.33}$$

Luego si se desea obtener el controlador basado en el modelo inverso se reemplaza $p_{1i}(k+1) = r_1(k+1)$ y $h_{1i}(k+1) = r_2(k+1)$ (referencias de presión y altura respectivamente). Los parámetros de las premisas y los consecuentes difusos son estimados con el método del gradiente y un algoritmo de mínimos cuadrados como se explico anteriormente. Finalmente la salida de control global está dada por la suma ponderada de cada uno de los controladores locales por su respectiva función de validación como lo muestra la expresión (3.13).

3.8.3. Resultados Simulación

Los resultados de las simulaciones son mostrados a continuación:

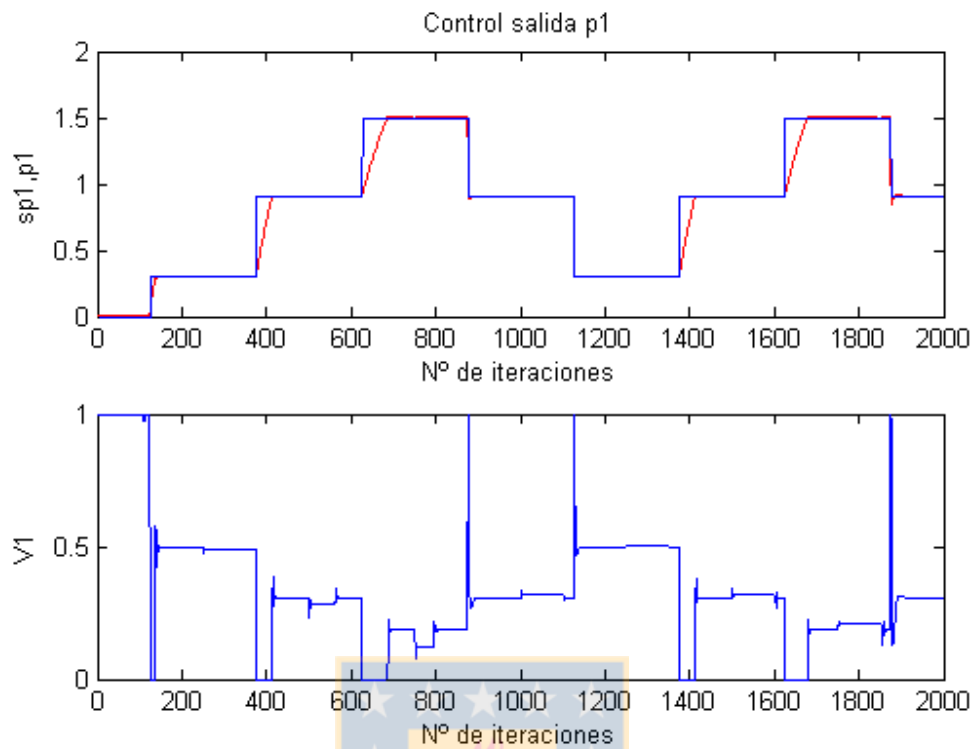


Figura 3.20: Respuesta de control salida p_1 y su respectiva acción de control V_1 .

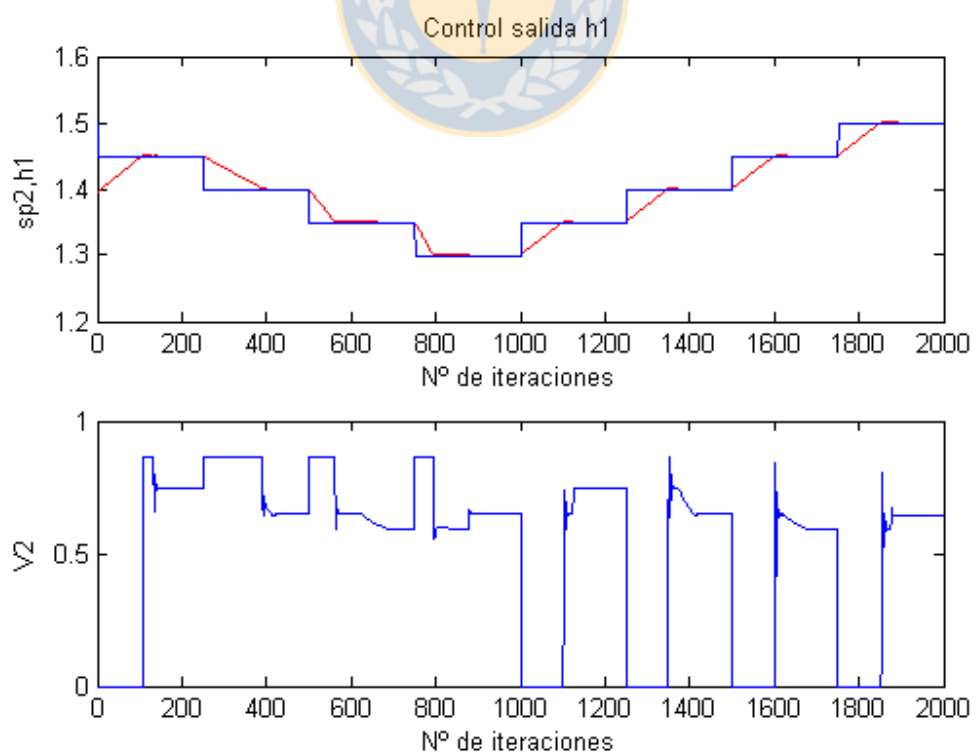


Figura 3.21: Respuesta de control salida h_1 y su respectiva acción de control V_2 .

Capítulo 4

4. Conclusiones

4.1. Conclusiones

El modelado y control neuro-difuso basado en modelos locales se presenta como una alternativa válida para su aplicación a sistemas complejos, dando generalmente lugar a algoritmos de control de grandes prestaciones y de gran eficiencia en cuanto a su implementación. Se ha destacado que el enfoque con modelos locales lineales tiene considerables ventajas sobre otros enfoques difusos o neuronales.

Estas ventajas que pueden obtenerse del empleo de esta representación son numerosas, ya que simplifican tanto el análisis de la dinámica de estos sistemas, como la síntesis de controladores estables. Una de las más destacables es que poseen una rápida adaptación para cambios en el ambiente, lo que los hace apropiados y una herramienta flexible para el diseño de sistemas no-lineales de control y modelación.

La modelación difusa, a diferencia de la modelación de caja negra como redes neuronales, provee una interfaz de gran transparencia en su estructura para la interpretación y análisis del diseñador u operador del sistema de control. Esta transparencia en su base de reglas y en su estructura misma hace posible la incorporación de conocimiento previo, además del bajo costo computacional que demandan debido a su intrínseca naturaleza lineal.

Este aspecto permite además, extraer información sobre el comportamiento no-lineal local del proceso. Para procesos dinámicos las ganancias locales, polos y ceros pueden ser obtenidos automáticamente de los modelos locales. Además puede mejorar la calidad, confiabilidad, capacidad de extrapolación y tiempo de entrenamiento significativamente.

Sin embargo a menudo se puede suscitar un conflicto entre la interpretabilidad y precisión, ya que también se pierde un poco de esta última a medida que los modelos son mas sencillos, de manera

que se debe manejar de forma adecuada la construcción de éstos para no descuidar y mantener un equilibrio entre estos dos objetivos. Los algoritmos propuestos en este trabajo intentan reducir la brecha entre exactitud e interpretabilidad en la modelación y el control neuro-difuso.

Paralelamente al esquema de control, el conocimiento previo disponible es un importante factor que influye en la elección de una arquitectura de modelo adecuada. En adición al diseño de control, otros aspectos juegan un importante rol para la implementación del controlador, dependiendo de la aplicación específica. Entre estos se incluyen por ejemplo la capacidad de implementación en tiempo real, y bajos requerimientos de recursos computacionales.

No obstante, muchas veces el conocimiento es inadecuado, en el sentido de que no siempre está disponible un modelo inverso exacto al momento de la implementación. En operación continua, la ejecución del modelo inverso y del controlador mejora debido al aprendizaje de las reglas y al incremento de la experiencia (datos entrada-salida) sobre el proceso. Al comienzo de la operación, algunos problemas pueden surgir en el diseño de control puesto que el error de modelación generalmente afecta la exactitud de los resultados de control. Por lo tanto la capacidad de seguimiento y de estabilidad del sistema en lazo cerrado depende en gran medida de la exactitud del modelo inverso.

Cabe resaltar la sencillez de implementación en tiempo real de cualquier algoritmo de control diseñado a partir de leyes de control locales, ya que éste se limita a la realización de una inferencia difusa de la ganancia de control de realimentación. Lo anterior se pudo constatar al revisar la bibliografía, ya que estas estrategias de control han sido usadas ampliamente usadas en diversos tipos de aplicaciones.

Otro aspecto importante se refiere a las limitaciones naturales de la red de inferencia difusa ante una pobre excitación, ya que en este caso, no es posible extraer suficiente información para caracterizar procesos complejos. Por lo tanto la señal de excitación debe poseer algún grado de variabilidad para llegar a modelos locales más óptimos.

Otra de las grandes desventajas del uso de estos sistemas de control inteligentes es que su ejecución no es realizable en régimen transiente. Este problema debería ser abordado con el uso de una estrategia de control híbrido, la cual combina un controlador inteligente con un controlador convencional. En efecto, en la mayoría de los casos, no hay implementadas soluciones neuro-difusas puras, sino más bien soluciones híbridas, donde el control inteligente es usado para mejorar el control convencional. De esta forma se podría tener como resultado un control más inteligente y autónomo.

Por otra parte, la adaptación del algoritmo basado en el método del descenso de gradiente compensa las incertidumbres y los cambios en la dinámica de los procesos. La habilidad de aprendizaje on-line del modelo permite que el controlador basado en éste sea usado en aplicaciones donde no existe el conocimiento del proceso a controlar o este proceso esta sujeto a cambios en sus características dinámicas.

4.2. Comentarios de los Resultados

Como se pudo observar de los resultados, la identificación del modelo arroja muy buenos resultados puesto que la señal de entrada que se utiliza es una señal que cubre casi todas las zonas de operación del sistema, sin embargo esto no es muy factible de realizar en aplicaciones en tiempo real dado al gran desgaste que este tipo de entradas implicaría por parte de los actuadores.

La validación por su parte puede mejorar notablemente si la excitación es más variada. No obstante se utilizó una entrada sinusoidal para esta etapa, puesto que es mucho más cercana a las entradas que se pueden encontrar en aplicaciones en tiempo real. Además los resultados de la modelación en la etapa de validación mejoran drásticamente con el uso de un algoritmo de identificación con adaptación (recursivo) de los parámetros.

Si se realiza una etapa de modelación previa al control que identifique la estructura de la red de inferencia, debido a las excelentes cualidades de generalización de éstas, se genera una adecuada acción de control en todas las zonas de operación donde el modelo es válido.

De los gráficos del sistema en lazo cerrado siguiendo una referencia, se puede observar que el controlador posee una gran robustez y adaptación a los cambios del sistema, además puede lidiar de gran forma con el acoplamiento existente entre las salidas y con el alto grado de no-linealidad que poseen los sistemas MIMO presentados, lo que se puede desprender de las gráficas del error.

4.3. Trabajos Futuros

En general, los resultados obtenidos son bastante satisfactorios, sin embargo se puede mejorar aún más la obtención de los modelos difusos Takagi-Sugeno, por ejemplo mediante la determinación automática del número de clusters o por algún método gráfico para visualizar la distribución de los datos de entrada, y de esta forma poder determinar en forma más exacta el número adecuado de clases

(clusters) a utilizar. También se pueden mejorar los resultados por la obtención de un valor más óptimo del factor de olvido, o por el uso de un factor de olvido variable, etc.

Para validar de manera más amplia la estrategia de control, se podría someter a prueba el Sistema Neuro-Difuso con otros sistemas no-lineales (N-L) de gran complejidad tales como el modelo de un espesador de relave, sistemas N-L de excitación crítica, en donde la salida no depende linealmente de la acción de control, sistemas N-L de oscilación caótica, o sistemas N-L de dinámica crítica tanto en la parte auto regresiva como en la excitación.

Sería interesante además ver el desempeño del sistema de control implementado, en alguna plataforma de programación en tiempo real tal como un PLC (Programable Logic Controller) con un sistema SCADA/HMI, o tal vez en un sistema de control DCS (Distributed Control System) como el delta V, ambos existentes en el *Laboratorio de Control de Procesos del Departamento de Ingeniería Eléctrica*.



Bibliografija

- [1] Amine Trabelsi, Frederic Lafont, Mohamed Kamoun, Gilles Enea. "Identification of nonlinear multivariable systems by adaptive fuzzy takagi-sugeno Model". *International Journal of Computational Cognition*, Volume 2, Number 3, Pages 137–153, September 2004.
- [2] Vesna Ranković, Ilija Nikolić. "Application of the takagi-sugeno fuzzy controller for solving the robots' inverse kinematics problem". *Facta Universitatis, Series: Mechanics, Automatic Control and Robotics* Vol.3, No 15, 2003, pp. 1039 - 1054, April 03, 2002.
- [3] Robert Babuska. "Data-Driven Fuzzy Modeling: Transparency and Complexity Issues". Delft University of Technology, Faculty of Information Technology and Systems Control Engineering Laboratory, Mekelweg 4, P.O. Box 5031, 2600 GA Delft The Netherlands.
- [4] Takagi, T. and Sugeno, M. (1985), "Fuzzy Identification of Systems and Its Applications to Modeling and Control", *IEEE Tran. on Systems, Man, and Cybernetics*, Vol. SMC-15, No. 1, pp. 116-132.
- [5] A. Fink, S. Töpfer, R. Isermann. "Nonlinear model-based control with local linear neuro-fuzzy models", 27 September 2002.
- [6] R. Babuska, HB Vebruggen, "An overview of fuzzy modeling for control", *Control Engineering Practice*, 4(11):1593-1606, 1996.
- [7] R. Babuska, HB Vebruggen, "Identification of composite linear models via fuzzy clustering", In *Proceedings European Control Conference 4 (1995)*, Rome, Italy, pp 1593-1606.
- [8] D.E. Gustafson, W.C. Kessel, "Fuzzy clustering with fuzzy covariance matrix", In *Proceedings IEEE CDC, San Diego (1979)*, pp 761-766.

- [9] F. Lafont, J.F. Balmat, "Fuzzy logic to the identification and the command of the multidimensional systems", *International Journal of Computational Cognition*, 2(2), pp 21-47, 2004.
- [10] O. Nelles, A. Fink, R. Babuska, M. Setnes, "Comparison of Two Construction Algorithms for Takagi-Sugeno Fuzzy Models", *International Journal of Applied Mathematics and Computer Science*, 10(4): 835-855, 2000.
- [11] O. Nelles, A. Fink, R. Isermann, "Local Linear Model Trees (LOLIMOT) Toolbox for Nonlinear System Identification", 12th IFAC Symposium on System Identification (SYSID), Santa Barbara, USA, 2000.
- [12] M. Sugeno, G.T. Kang. "Structure identification of fuzzy model, *Fuzzy Sets and Systems*" , 28: 15-33, 1987.
- [13] Robert Babuska. "Fuzzy Systems, Modeling and Identification". Delft University of Technology, Department of Electrical Engineering Control Laboratory, Mekelweg 4, P.O. Box 5031, 2600 GA Delft, The Netherlands.
- [14] Pablo Carbonell. "Metodología para la selección de modelos locales borrosos en el espacio de regímenes de trabajo". Dpto. Ingeniería de Sistemas y Automática Universidad Politécnica de Valencia Pza Ferrándiz i Carbonell, 2; 03801 ALCOI.
- [15] José L. Díez, José L. Navarro y A. Sala. "Algoritmos de Agrupamiento en la Identificación de Modelos Borrosos". *Revista Iberoamericana de Automática e Informática Industrial*, Vol. 1, Núm.2, Julio 2004, pp. 32-41.
- [16] Mamdani E., Assilian S. "An Experiment in linguistic Synthesis with a Fuzzy Logic Controller", *International Journal of Man-Machine Studies*, 1-13, 1975.
- [17] Pedro A. Ascencio O. "Sistemas Neuro-Difusos en Modelación y Control de Procesos No Lineales". Memoria de Título, Universidad de Concepción, Depto. Ing. Eléctrica, 1998.
- [18] Mauricio A. Quintana M. "Implementación de un Controlador Neuro-Difuso en Tiempo Real Sobre PLC". Memoria de Título, Universidad de Concepción, Depto. Ing. Eléctrica, 1998.

[19] Balazs Balasko, Janos Abonyi and Balazs Feil, “Fuzzy Clustering and Data Analysis Toolbox”, Department of Process Engineering University of Veszprem P.O.Box 158 H-8200, Veszprem, Hungary

[20] Tomothy J. Ross. “Fuzzy Logic with Engineering Applications”, Second Edition 2004 John Wiley & Sons Ltd.

[21] János Abonyi, Hans Andersen, Lajos Nagy, Ferenc Szeifert Inverse, “Fuzzy-Process-Model Based Direct Adaptive Control”. Department of Chemical Engineering Cybernetics University of Veszprém.

[22] Alexandra Grancharova, Tor A. Johansen, Juš Kocijan. “Explicit model predictive control of gas-liquid separation plant via orthogonal search tree partitioning”. Department of Engineering Cybernetics, Norwegian University of Science and Technology 7491 Trondheim Norway.



Anexo A

A. Lógica Difusa y Redes Neuronales

En este capítulo se expone en primer lugar, la teoría de la lógica difusa, haciendo mención a sus principales elementos, conceptos básicos y definiciones de relevancia. En una segunda fase se hace una introducción y breve descripción de las redes neuronales (o redes adaptivas) explicando sus principales aspectos. Luego se hace una comparación entre ambas teorías, destacando ventajas y limitaciones de ambos sistemas.

A.1. Introducción a los Sistemas Difusos

La lógica difusa fue investigada por primera vez por Lofti A. Zadeh [9], quien introdujo el concepto de conjunto difuso bajo el que reside la idea de que los elementos bajo los que se construye el pensamiento humano no son numéricos sino lingüísticos. Estos términos lingüísticos, son inherentemente menos precisos que los datos numéricos pero en muchas ocasiones aportan información más útil para el razonamiento humano. El aspecto central de los sistemas difusos, es que a diferencia de los que se basan en lógica clásica, tienen la capacidad de reproducir aceptablemente los modos usuales del razonamiento, considerando que la certeza de una proposición, es una cuestión de grado. Este contraste se puede visualizar en la figura A.1.

El primer ejemplo usado por para ilustrar el concepto de conjunto difuso fue el conjunto “hombres altos”. Según la teoría de la lógica clásica el conjunto “hombres altos” es un conjunto al que pertenecerían los hombres con una estatura mayor a un cierto valor, que se puede establecer en 1.80 mt, por ejemplo, y todos los hombres con una altura inferior a este valor quedarían fuera del conjunto. Así se tendría que un hombre que mide 1.81 mt. de estatura pertenecería al conjunto hombres altos, y en cambio un hombre que mida 1.79 mt. de altura ya no pertenecería a ese conjunto. Sin embargo no parece muy lógico decir que un hombre es alto y otro no cuando su estatura difiere en dos centímetros. De tal forma que la lógica difusa plantea que un hombre con estatura menor 1.80 mt igual pertenece al

conjunto de hombre altos pero con un grado de pertenencia que va entre 0 y 1 mientras que la lógica clásica se asigna un grado de pertenencia “0” a los hombres con estatura menor a 1.80 mt y grado de pertenencia “1” a los hombres con estatura mayor a 1.80 mt.

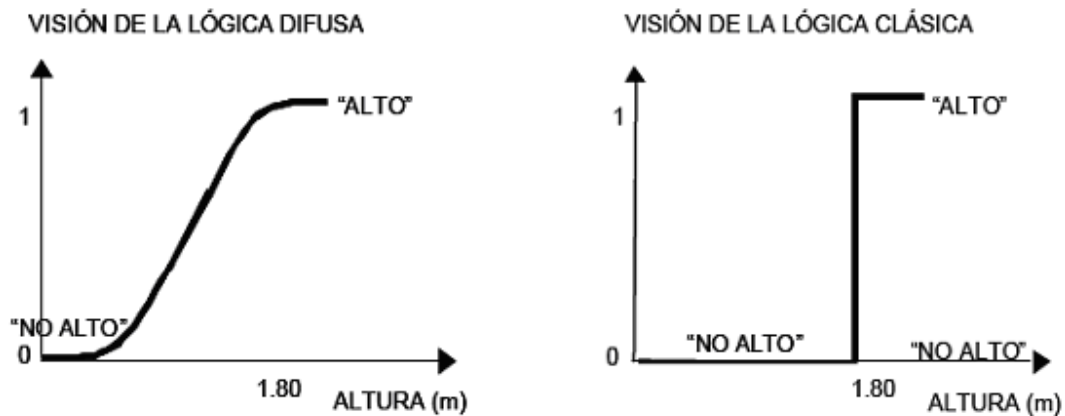


Figura A.1: Lógica clásica versus lógica difusa

A.2. Elementos Básicos de Sistemas Difusos y Lógica Difusa

A.2.1. Variable Lingüística

Uno de los primeros conceptos que se debe manejar para comprender de buena forma la lógica difusa es el de variable lingüística. Esta se utiliza cotidianamente para representar expresiones humanas de calificativos usados normalmente, tales como, la temperatura de un reactor, estatura de una persona, velocidad de un vehículo etc. Por ejemplo si se refiere a ésta última, se podría definir los valores: baja, media y alta para la variable lingüística “velocidad”. Una variable lingüística puede ser caracterizada por los siguientes elementos: $(x, X, T(x))$, donde x es el nombre de la variable, X es el universo de discurso o rango en que se encuentra la variable, y T es una transformación que asocia a cada valor lingüístico de x un valor correspondiente $T(x)$ que denota el conjunto difuso en X .

A.2.2. Conjunto Difuso

Sea X una colección de objetos denotados por x , entonces un Conjunto Difuso (FS, Fuzzy Set) $A \in X$ es definido por el siguiente conjunto de pares ordenados:

$$A = \{(x, \mu_A(x)) \mid x \in X\} \quad (\text{A.1})$$

Donde $\mu_A(x) : X \rightarrow [0,1]$ se denomina función de pertenencia (MF, Membership Function) del conjunto difuso A y X representa el **universo de discurso**. La MF transforma cada elemento de X en un grado de pertenencia entre 0 y 1.

A.2.3. Funciones de Pertenencia (FP)

De lo mencionado anteriormente se puede deducir que un conjunto difuso queda completamente caracterizado por su función de pertenencia, ésta puede ser definida en forma continua o discreta, unidimensional o bidimensional. La forma más adecuada de expresarlas es mediante expresiones matemáticas parametrizadas con un conjunto mínimo de valores característicos. Estos parámetros deben permitir una adaptación a través de algún criterio coherente. Con respecto a la elección de estas FP hay que considerar tres aspectos importantes: Distribución, forma y cantidad de las FP a utilizar. Dependiendo de estas tres características va a depender la sensibilidad y convergencia que tenga el modelo y/o controlador difuso. A continuación se describen algunas de las más comunes.

Definición A.2.3.1 Funciones de Pertenencia Triangulares

Una FP triangular queda definida mediante tres parámetros $\{a, b, c\}$ de la siguiente manera:

$$\mu_A(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & a \leq x \leq b \\ 0, & c \leq x \end{cases} \quad (\text{A.2})$$

El conjunto de parámetros $\{a, b, c\}$ con $(a < b < c)$ determinan las coordenadas en x de los tres vértices de la FP triangular.

Definición A.2.3.2 Funciones de Pertenencia Trapezoidales

Una FP triangular se define mediante cuatro parámetros $\{a, b, c, d\}$ como se muestra:

$$\mu_A(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases} \quad (\text{A.3})$$

El conjunto de parámetros $\{a, b, c, d\}$ con $(a < b < c < d)$ determinan las coordenadas en x de los tres vértices de la FP trapezoidal definida.

Las FP triangulares y trapezoidales han sido ampliamente utilizadas en aplicaciones en tiempo real, sin embargo debido a que están compuestas por líneas rectas que concluyen en vértices, las derivadas no existen en estos puntos.

Para algunos casos las funciones triangulares no son apropiadas y el uso de ellas en estas situaciones puede provocar que los valores de la función de pertenencia no sean los deseados. Además, en la modelación de sistemas dinámicos el uso de funciones triangulares puede aproximar su comportamiento escasamente a un grado de precisión. Generalmente, cuando hay que representar variables borrosas en finanzas, mercado, y otros modelos de negocio, la combinación de curvas con forma de campana, sigmoideas y para representar los puntos intermedios y extremos respectivamente, proveen una mejor forma para representar regiones borrosas individuales. Para la modelación de procesos por ejemplo son muy utilizadas las FP gaussianas, esto se puede atribuir a que la transición entre una zona y otra es más suave. Para el caso de un modelo basado en modelos locales, estas funciones permiten un cambio de comportamiento más gradual entre los distintos submodelos al pasar de una región a otra.

Definición A.2.3.3 *Funciones de Pertenencia Gaussianas*

Este tipo de funciones quedan determinadas por dos parámetros: $\{c, \sigma\}$, como sigue a continuación:

$$\mu_A(x) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \quad (\text{A.4})$$

Donde c representa el centro de la FP y σ la desviación estándar que determina su anchura.

Definición A.2.3.4 *Funciones de Pertenencia del tipo Campana*

Esta FP se caracteriza por medio de tres parámetros $\{a, b, c\}$, como se muestra a continuación:

$$\mu_A(x) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \quad (\text{A.5})$$

Donde c define el centro, a define el ancho, y b define las pendientes de los puntos de cruce de la FP. El parámetro b generalmente es positivo. Si fuera de valor negativo, indicaría que la forma de la campana sería invertida.

Puesto que estas dos últimas funciones mencionadas poseen características de ser diferenciables en todos sus puntos (propiedad de suavidad), han llegado a ser cada vez más usadas para especificar conjuntos difusos. Las funciones gaussianas son bien conocidas en probabilidad y estadística, y tienen importantes propiedades entre las cuales se tiene que no cambian su forma al multiplicarse por otra (el producto de dos gaussianas es una gaussiana con un factor de escalamiento). La FP tipo campana tiene un parámetro más que la gaussiana, por ende tiene un grado más de libertad para ajustar para ajustar sus pendientes en los puntos de cruce.

Definición A.2.3.5 *Funciones de Pertenencia del tipo Sigmoide*

A pesar de que las FP gaussianas y de forma de campana tienen una mayor suavidad no permiten especificar FP asimétricas. Sin embargo una FP sigmoide puede ser abierta por la izquierda o por la derecha, y el producto o la diferencia absoluta de dos permite definir FP asimétricas. Estas FP se definen de la siguiente forma:

$$\mu_A(x) = \frac{1}{1 + e^{-a(x-c)}} \quad (\text{A.6})$$

Donde c es el punto de cambio de concavidad y a entrega el valor de la pendiente tanto en signo como en magnitud, obteniéndose para $a > 0$ una sigmoide abierta hacia la derecha y para $a < 0$ una abierta a la izquierda. Además dentro de las propiedades de la sigmoide está la posibilidad de generar FP cerradas.

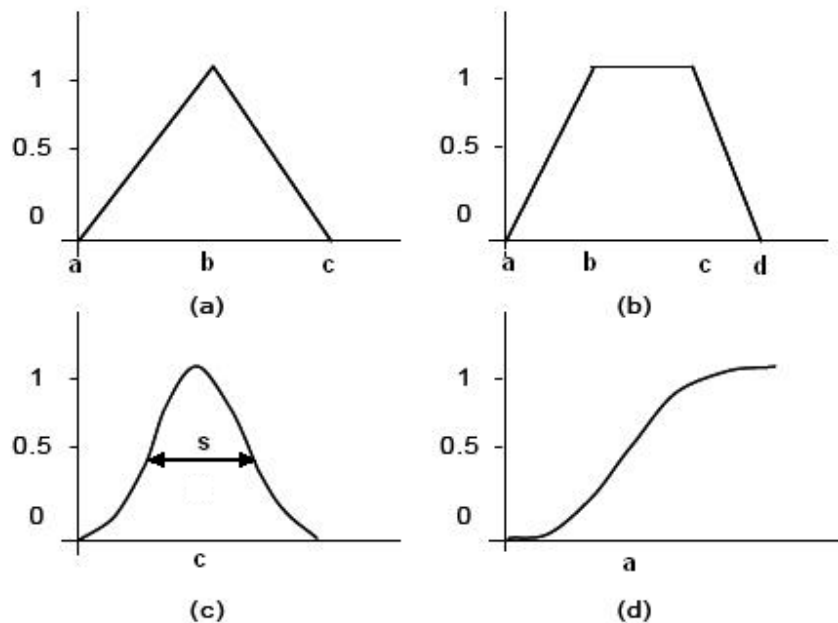


Figura A.2: Algunas de las funciones de pertenencia más habituales: (a) triangular, (b) trapezoidal, (c) gaussiana, y (d) sigmoideal.

A.2.4. Regla o Implicación Difusa

En el razonamiento difuso una regla difusa representa una función de implicación como una generalización de lo planteado en la lógica clásica como $A \rightarrow B$. Dicha implicación es referida como declaraciones difusas **if-then** que se caracterizan por la siguiente estructura:

$$R_1: \text{if (antecedente) then (consecuente)} \quad (A.7)$$

Donde el *antecedente* o premisa puede contener la combinación de proposiciones difusas por medio de los conectivos *and* y *or*, y el *consecuente* puede ser una proposición compuesta o normalmente una asignación atómica. Dichas reglas en el ámbito del control de sistemas describen una relación *causal* entre las variables de estado y las acciones de control. Como un ejemplo se puede considerar un sistema MISO de dos entradas y una salida:

$$R: \text{if } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } A_2 \text{ then } u \text{ is } B \quad (A.8)$$

Si los conjuntos difusos A_1 , A_2 , B son identificados por sus correspondientes funciones de pertenencia $\mu_{A_1}(x_1)$, $\mu_{A_2}(x_2)$ y $\mu_B(u)$, la relación R equivalente a la *regla difusa* puede ser construida por la siguiente expresión:

$$\mu_R(x_1, x_2, u) = I(T(\mu_{A_1}(x_1), \mu_{A_2}(x_2)), \mu_B(u)) \quad (\text{A.9})$$

Donde la conjunción está basada en la *norma T* (ver [18]) y la implicación difusa por la función binaria *I*. Esta función *I* recibe el nombre de **Implicación Difusa** y consta de múltiples representaciones de acuerdo a las equivalencias de los conectivos utilizados.

Para expresar el conocimiento que entregan las reglas difusas de forma completa normalmente se precisa de varias reglas, que se agrupan formando lo que se conoce como una **Base de Reglas**, es decir el conjunto de reglas que expresan las relaciones conocidas entre antecedentes y consecuentes. La base de reglas se puede representar como una matriz o tabla de las reglas que la componen, esto es más conocido como una **Memoria Asociativa Difusa (MAD)**. Estas MAD representan la consecuencia de cada regla definida para cada combinación de dos entradas. Las MAD permiten realizar una representación gráfica clara de las relaciones entre dos variables lingüísticas de entrada y la variable lingüística de salida, pero requiere que se indique explícitamente todas las reglas que se pueden formar con estas dos variables de entrada. Cuando el número de conjuntos de cada una de las particiones de entrada crece, las MAD se hacen difícilmente manejables.

A.2.5. Sistemas de Inferencia Difusa

Un sistema de inferencia difusa es una entidad de cómputo basada en los conceptos de la teoría de conjuntos difusos, reglas del tipo *si...entonces* y el razonamiento difuso. El campo de aplicación de este tipo de lógica ha tenido un gran incremento en múltiples disciplinas entre las cuales una de las más utilizadas es el control automático, específicamente en la **modelación difusa**. Los sistemas de inferencia difusa más conocidos o pueden ser clasificados en dos grandes grupos:

- **Esquemas puros:** Son aquellos en que los espacios de entrada y salida son de características difusas, es decir los valores del antecedente tanto como del consecuente son conjuntos difusos.
- **Esquemas modificados:** En general los valores considerados como variables de entrada o salida no son de naturaleza difusa, por lo cual son necesarias transformaciones.

Este trabajo estará enfocado los esquemas modificados, ya que los valores no deben poseer características de conjunto, sino más bien de valores puntuales de las variables observadas. Este último esquema mencionado puede ser visualizado en el diagrama de la Figura A.3. Además es posible realizar una nueva clasificación de los sistemas de inferencia difusa, en la cual encontramos tres grupos de los cuales se mencionaran los dos más importantes a continuación.

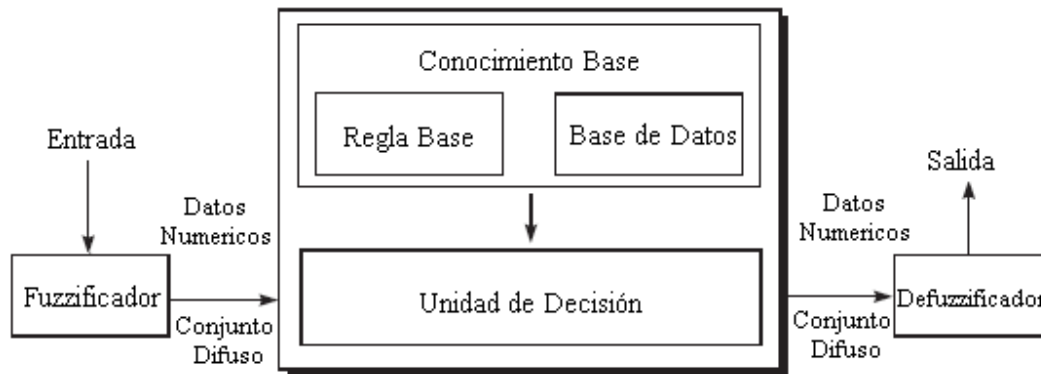


Figura A.3: Diagrama representativo de los esquemas modificados de inferencia difusa.

Definición A.2.5.1. *Sistemas de Inferencia del 1^{er} tipo:*

Este sistema fue propuesto por primera vez por Mamdani [17] (1975) para realizar un controlador difuso que estabiliza un sistema en torno a un punto de trabajo, y tiene la siguiente estructura:

$$R_k : \text{if } x_1 \text{ is } A_1^k \text{ and } \dots \text{ and } x_n \text{ is } A_n^k \text{ then } y_k \text{ is } B_k \quad (\text{A.10})$$

Donde A_n^k y B_k son los conjuntos difusos en los universos de discurso $U \subset \mathfrak{R}$ y $V \subset \mathfrak{R}$. $x = (x_1, \dots, x_n) \in U \times \dots \times U_n$ e $y \in V$ son variables lingüísticas de entrada y salida respectivamente.

Definición A.2.5.2. *Sistemas de Inferencia del 2^o tipo:*

Este tipo de sistema fue propuesto por Takagi y Sugeno (T-S) [4], y es conocido como modelo difuso de T-S. Este consiste en un desarrollo sistemático que genera reglas difusas desde un conjunto de datos de entrada y salida. La función de salida es una combinación lineal compuesta de polinomios sobre las variables de entrada. La estructura de este tipo de sistemas se muestra en la siguiente ecuación:

$$R_k : \text{if } x_1 \text{ is } A_1^k \text{ and } \dots \text{ and } x_n \text{ is } A_n^k \text{ then } y_k = f_k(x) \quad (\text{A.11})$$

$$\text{Donde} \quad f_k(x) = \theta_1^k \cdot x_1 + \dots + \theta_n^k \cdot x_n + \mathcal{G}^k \quad (\text{A.12})$$

Siendo $x = (x_1, \dots, x_n)$ el vector de entradas y $\Theta = (\theta_1^k, \dots, \theta_n^k, \mathcal{G}^k)$ el vector paramétrico del modelo con \mathcal{G}^k un valor escalar que representa el coeficiente de *offset*. Cuando f es igual a una

constante, el modelo se denomina de orden cero y es un caso especial equivalente al sistema de inferencia del 1^{er} tipo donde el consecuente es puntual o pre-defuzzificado. Para encontrar el valor representativo de la salida del método de inferencia se utiliza el cálculo de un promedio ponderado entre cada uno de los valores denominados y_k y sus respectivos pesos denominados w_k , como se muestra en la figura A.5. Esto puede ser expresado de la siguiente forma:

$$y = \frac{\sum_{k=1}^M w_k \cdot y_k}{\sum_{k=1}^M w_k} = \sum_{k=1}^M \bar{w}_k \cdot y_k \quad (\text{A.13})$$

Donde M es el número de reglas luego $k = 1, 2, \dots, M$ y \bar{w}_k corresponde a el factor de peso normalizado por la sumatoria de todos los pesos involucrados. Los sistemas difusos descrito con n entradas x_i y una sola salida “ y ”, se conocen como MISO (Multiple Input-Single Output), mientras que los que tienen varias salidas se denominan MIMO (Multiple Input-Multiple Output). Para estos últimos sistemas se puede generalizar el formato anterior de las reglas o bien descomponerlo en sistemas de tipo MISO.

A.2.6. Fuzzificación

La etapa de fuzificación consiste en transformar las mediciones de las variables de entrada no difusas en conjuntos difusos. Para esto el fuzificador mapea el vector de entrada $x = (x_1, \dots, x_n) \in X$ en un conjunto difuso $A \in X$. Existen a lo menos dos formas de realizar este mapeo:

Definición A.2.6.1. Fuzificación Singleton:

Es el método de fusificación más utilizado, principalmente en sistemas de control y consiste en considerar los propios valores discretos como conjuntos difusos. Dicho de otra forma, para cada valor de entrada x se define un conjunto de A que lo soporta con función de pertenencia $\mu_A(x')$ de manera que:

$$\mu_A(x) = \begin{cases} 1, & x' = x \in U \\ 0, & x' \neq x \in U \end{cases} \quad (\text{A.14})$$

Definición A.2.6.2. *Fuzificación No Singleton:*

Tiene como valor máximo 1 y decrece en forma uniforme respecto de dicho punto. Un ejemplo de ella es la forma gaussiana descrita anteriormente:

$$\mu_A(x) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \quad (\text{A.15})$$

A.2.7. Conocimiento Base

El conocimiento base es todo aquel conocimiento previo sobre el proceso en análisis. Este se constituye principalmente de dos módulos:

- **Regla base:** Consiste en una colección de reglas difusas *if – then* como las que se muestran en la relación (A.16):

$$R_k : \text{if } x_1 \text{ is } A_1^k \text{ and ... and } x_n \text{ is } A_n^k \text{ then } y_k \text{ is } B_k \quad (\text{A.16})$$

Que genera una relación R^k según el método de implicación difusa elegido para todo $k = \overline{1, M}$. Luego la regla base representa el conocimiento ordenado, que en general poseen los operadores respecto a los estados del proceso.

- **Base de datos:** Posee la información suficiente sobre las funciones de pertenencia (número, distribución, forma, etc.) de los conjuntos difusos usados en las correspondientes reglas difusas.

A.2.8. Defuzzificación

El desfusor realiza la función contraria al difusor, es decir, transforma un conjunto difuso sobre el universo V , en un valor no difuso $y \in V$. El difusor tiene como entradas valores concretos de las variables de entrada y como salida grados de pertenencia de los conjuntos difusos (entres 0 y 1). La entrada al bloque desfusor es el conjunto de salida, resultado del bloque de inferencia y la salida es un valor numérico concreto representativo del conjunto. Para realizar este cálculo existen varios métodos de los cuales se mencionarán solamente tres:

- **Método del máximo:** Se define como:

$$y = \arg \sup_{y \in V} (\mu_B(y)) \quad (\text{A.17})$$

Es decir corresponde al punto en que $\mu_B(y)$ alcanza su valor máximo.

- **Método del Promedio de los Máximos:** Es el promedio de todas las acciones de control máximas. En el caso discreto se expresa de la siguiente forma:

$$y = \sum_{j=1}^l \frac{y_j}{l} \quad (\text{A.18})$$

Donde y_j el valor en el que la función de pertenencia alcanza su valor máximo, y “ l ” es el número de valores que se encuentran en el máximo.

- **Método del centro de Áreas o Centroide:** Es el método comúnmente más usado. Este método ubica el centro de gravedad de la distribución de la acción de control, también se puede interpretar como el promedio ponderado. Para el caso de un universo discreto se tiene:

$$y = \frac{\sum_{j=1}^n \mu_B(y_j) \cdot y_j}{\sum_{j=1}^n A_j(\mu_B(y_j))} \quad (\text{A.19})$$

Donde y_j se asume el valor medio de la regla de salida, “ n ” es el número de discretizaciones del universo de discurso, y A_j es el área del conjunto difuso.

A.3. Introducción a las Redes Neuronales

¿Qué es una Red Neuronal?

Es un sistema de procesamiento de información que tiene ciertas características de comportamiento inspiradas en el conocimiento actual que tenemos sobre el funcionamiento de las neuronas biológicas. Son por tanto modelos computacionales paralelos que constan de unidades de proceso (neuronas) adaptivas y masivamente interconectadas.

La aplicación de redes neuronales artificiales (RNA) ha llamado significativamente la atención en varias disciplinas, tales como procesamiento de señales, identificación y control. El éxito de las redes neuronales es principalmente atribuido a sus características:

- (1) Estructura paralela con etapas distribuidas y procesamiento cantidad masiva de información.
- (2) Capacidad de aprendizaje, lo que es posible por ajuste de los pesos de las interconexiones de las redes.

La primera característica habilita a la red neuronal para procesar gran cantidad de información en tiempo real. La implicación de la segunda característica es que las dinámicas no-lineales de un sistema pueden ser aprendidas e identificadas directamente por una red neuronal artificial. La red puede también adaptarse a cambios en el ambiente y toma decisiones a pesar de la incertidumbre en las condiciones de operación.

Las RNA se parecen al cerebro en dos aspectos:

- La red adquiere el conocimiento mediante un **proceso de aprendizaje**.
- Los pesos de las conexiones (**pesos sinápticos**) entre las unidades de proceso se utilizan para **almacenar el conocimiento**.

El procedimiento utilizado para el proceso de aprendizaje se llama **algoritmo de aprendizaje** y permite modificar los pesos sinápticos de la red neuronal para conseguir un objetivo deseado. Estas redes se llaman neuronales por estar inspiradas en la neurociencia pero no pretenden ser buenos modelos de neuronas biológicas. Se busca sobre todos su **capacidad computacional**.

A.3.1. Descripción de las Redes Adaptivas

Una red adaptiva es una red multicapa en la cual cada nodo ejecuta una función particular (función de nodo) sobre las señales de llegada, usando un conjunto de parámetros específicos a este nodo. La forma de las funciones de nodo pueden variar de un nodo a otro, y la elección de cada función de nodo depende sobre todo de cual función la red adaptiva esta diseñada para implementar.

Para reflejar diferentes capacidades adaptivas, se usan nodos circulares y cuadrados en una red adaptiva. Un nodo cuadrado (nodo adaptivo) tiene parámetros modificables mientras que un nodo circular (nodo fijo) no tiene este tipo de parámetros. El conjunto de parámetros de una red adaptiva es

la unión del conjunto de parámetros de cada nodo adaptivo. En orden para obtener un mapeo de entrada - salida deseado, estos parámetros son actualizados de acuerdo a los **datos de entrenamiento** dados y un procedimiento de actualización que se siguen para ajustar los parámetros para optimizar el desempeño de la red, se les conoce comúnmente como *Algoritmos de Aprendizaje*. El desempeño de una red se determina midiendo el error de la salida deseada y la salida de la red bajo las mismas condiciones de entrada. A esta variable se le denomina la *Medida del Error* y puede tomar diferentes formas para diferentes aplicaciones. En general, una regla de aprendizaje se deriva al aplicar una técnica de optimización específica a una medida de error dada.

Las redes adaptivas se pueden clasificar en dos categorías de acuerdo al tipo de enlaces que tengan: **Redes de Avance (FN, Feedforward Networks)** y **Redes Recurrentes (RN, Recurrent Networks)**. En la figura A.4 se puede observar un ejemplo típico de la primera categoría, dado que la salida de cada nodo se propaga hacia la salida total. Si existe algún lazo de realimentación, la red es RN.

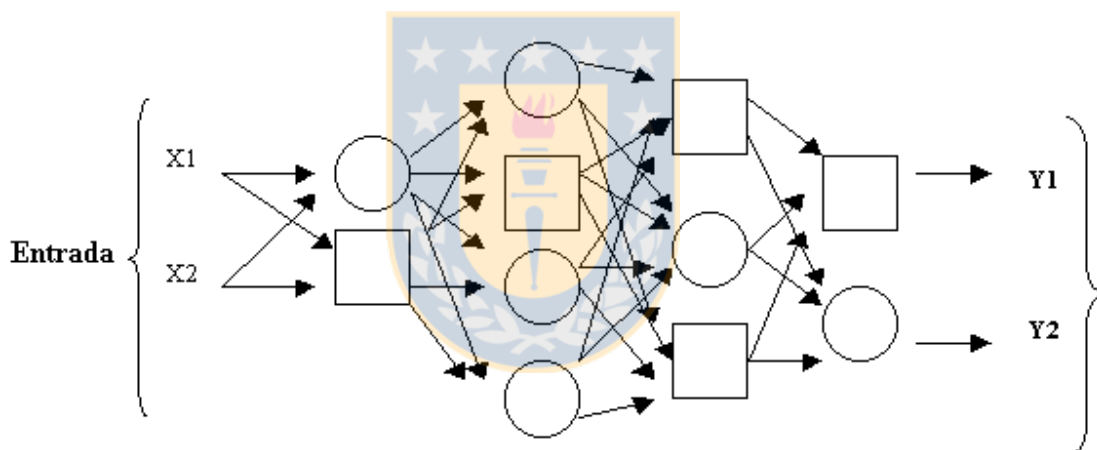


Figura A.4: Ejemplo de red adaptiva

A.3.2. Reglas de Aprendizaje

Una de las características más importantes de las redes neuronales es su capacidad de aprender interactuando con su entorno o con alguna fuente de información. El aprendizaje de la red es un proceso adaptivo mediante el cual se van modificando los pesos sinápticos de la red para mejorar su comportamiento. De manera general, una red neuronal va a modificar su peso sináptico w_{ij} correspondiente a la conexión de la neurona i con la neurona j mediante una regla de aprendizaje de la forma:

$$w_{ij}(k+1) = w_{ij}(k) + \Delta w_{ij}(k) \quad (\text{A.20})$$

Es decir, el nuevo valor del peso sináptico se obtiene sumándole una cantidad (modificación) al valor antiguo. Entre los métodos de aprendizaje más comunes se pueden mencionar los siguientes:

- **Aprendizaje supervisado:** En este tipo de aprendizaje se dispone de un conjunto de patrones de entrenamiento para los que se conoce perfectamente la salida deseada de la red. Un objetivo para diseñar la regla de aprendizaje supervisada es generalmente minimizar el error entre las salidas (respuestas) de la red y las salidas (respuestas) deseadas.

De esta forma se tendrán reglas de aprendizaje basadas en la corrección del error, como la regla de retropropagación del error, el algoritmo de mínimos cuadrados, y el método del descenso del gradiente, muy utilizados en problemas de clasificación y predicción.

- **Aprendizaje no supervisado:** Se le conoce también con los nombres de competitivo o autoorganizado, en el que se dispone de un conjunto de patrones de entrenamiento pero no se conocen las salidas deseadas de la red. La red por sí misma buscará su comportamiento más adecuado atendiendo a cierto criterio y encontrará estructuras o prototipos en el conjunto de patrones de entrenamiento. Como ejemplo de este aprendizaje se tiene la regla de aprendizaje competitivo no supervisado, utilizada en problemas de agrupación de patrones y obtención de prototipos, y los algoritmos de agrupación de datos [21].

A.3.3. Comparación de las Teorías

Teóricamente las redes neuronales y los sistemas difusos (fuzzy) son equivalentes, pero en la práctica cada uno tiene sus propias ventajas y desventajas, de las cuales se puede desprender que existe complementariedad en diversos tópicos. Aunque estos sistemas tienen una perspectiva similar, existen claras diferencias en su estructura:

- Los sistemas de redes neuronales poseen un gran número de elementos de decisión altamente interconectados, los cuales demuestran la habilidad para aprender y generalizar a partir de información de entrenamiento. En cambio los sistemas difusos, implementan decisiones, en base al grado de cumplimiento de las variables de entrada y un número limitado de precondiciones o implicaciones lingüísticas.

- Los sistemas neuronales poseen valores sinápticos distribuidos en múltiples capas, los cuales son complicados de interpretar, ya sea con respecto a los valores en las interconexiones tanto como en

describir como cada una de ellas influye en el sistema (modelo de caja negra), como también se torna difícil determinar la topología y tamaño de la red para un problema particular, de incorporar conocimiento previo y dimensionar. Además el aprendizaje es usualmente lento y no garantizado.

-Por otra parte los sistemas difusos son estimadores numéricos estructurados, basado en el conocimiento experto y el razonamiento partir de términos lingüísticos. Lo anterior permite afrontar determinados estados con bastante robustez, siempre y cuando la base del conocimiento cubra en su totalidad los eventos dinámicos. Sin embargo la ausencia de mecanismos transparentes de aprendizaje, que permitan afrontar las variaciones no presupuestadas, hacen que estos sistemas disminuyan su desempeño.

-Los sistemas difusos además tienen una pobre capacidad de aprendizaje ya que es difícil al operador humano de sintonizar tanto las reglas difusas como las funciones de pertenencia. Además no siempre se puede contar con la presencia de un experto.

A.3.4. Integración de las Teorías

Los sistemas híbridos que combinan lógica difusa, redes neuronales, algoritmos genéticos y sistemas expertos, proporcionan los métodos más eficientes para resolver una gran variedad de problemas. Cada una de esas técnicas tiene propiedades computacionales particulares (por ejemplo: habilidad de aprender) que las hace óptimas para resolver ciertos problemas. Uno de estos sistemas híbridos corresponde a los sistemas Neuro-Difusos, que combinan las técnicas de redes neuronales artificiales y las técnicas de inferencia difusa.

El ámbito de coexistencia entre las teorías o visión integrista, se respalda al menos desde dos razonamientos evidentes, los cuales pueden ser definidos como **unificación** y **potenciación**, los que se describen a continuación:

Unificación: Este aspecto se refiere a que los elementos y conceptos que estas presentan no se pueden desvincular para establecer una única visión. Para evidenciar lo dicho anteriormente se debe responder a dos preguntas:

¿Cuándo es apropiado usar lógica difusa? Cuando el proceso involucra interacción humana ya sea para descripción o intuición, cuando el modelo matemático no existe o existe dificultad en las evaluaciones en tiempo real, cuando se tiene imprecisión en los datos, y necesariamente cuando existen reglas claras de decisión asociadas a la dinámica.

¿Cuándo es apropiado usar redes neuronales? Cuando se requiere establecer transformaciones no lineales de manera automática, cuando se deben tomar decisiones a partir de datos masivos, cuando se requiere optimizar en corto tiempo, y necesariamente cuando se requieren adaptaciones en línea de acuerdo a la evolución del medio.

Lo expresado anteriormente deja en evidencia que el conocimiento experto y el aprendizaje optimizado son necesarios mutuamente. Esto es dado que el conocimiento humano necesita herramientas autónomas para optimizar las decisiones y por otro lado un sistema numérico necesita ser transparente para ser guiado o validado por el diseñador o evaluador.

Potenciación: Este aspecto consiste en potenciar ambas teorías dado a la naturaleza complementaria que éstas presentan, de manera de rescatar las virtudes de una y mejorar las deficiencias de la otra.

